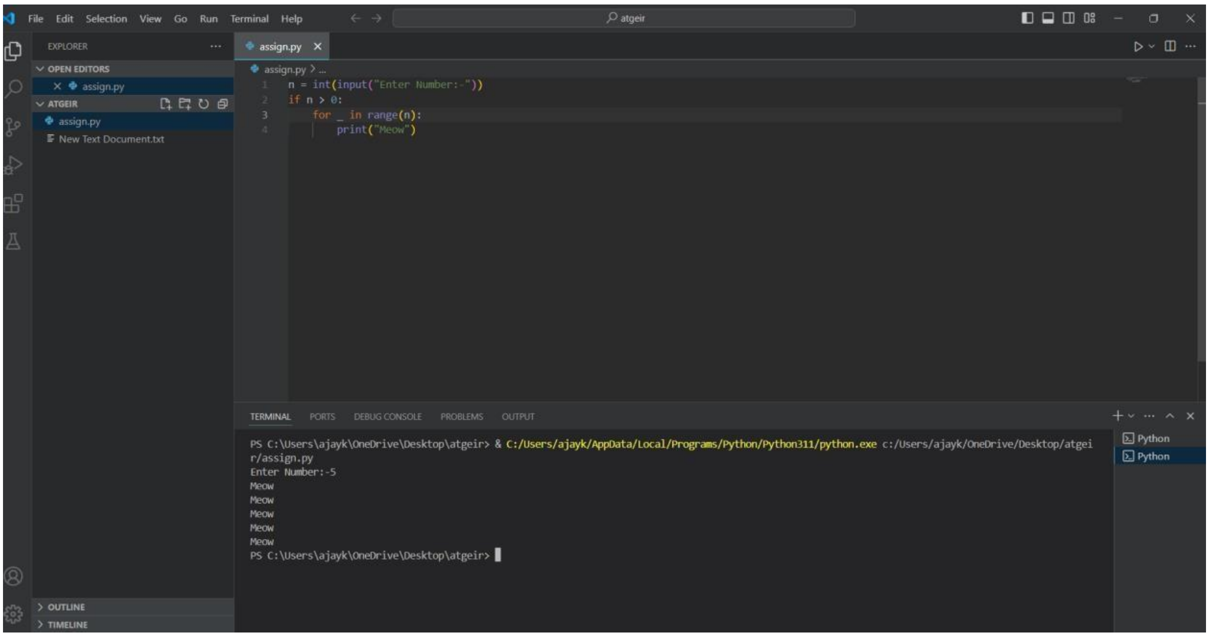
Assignement_Day1

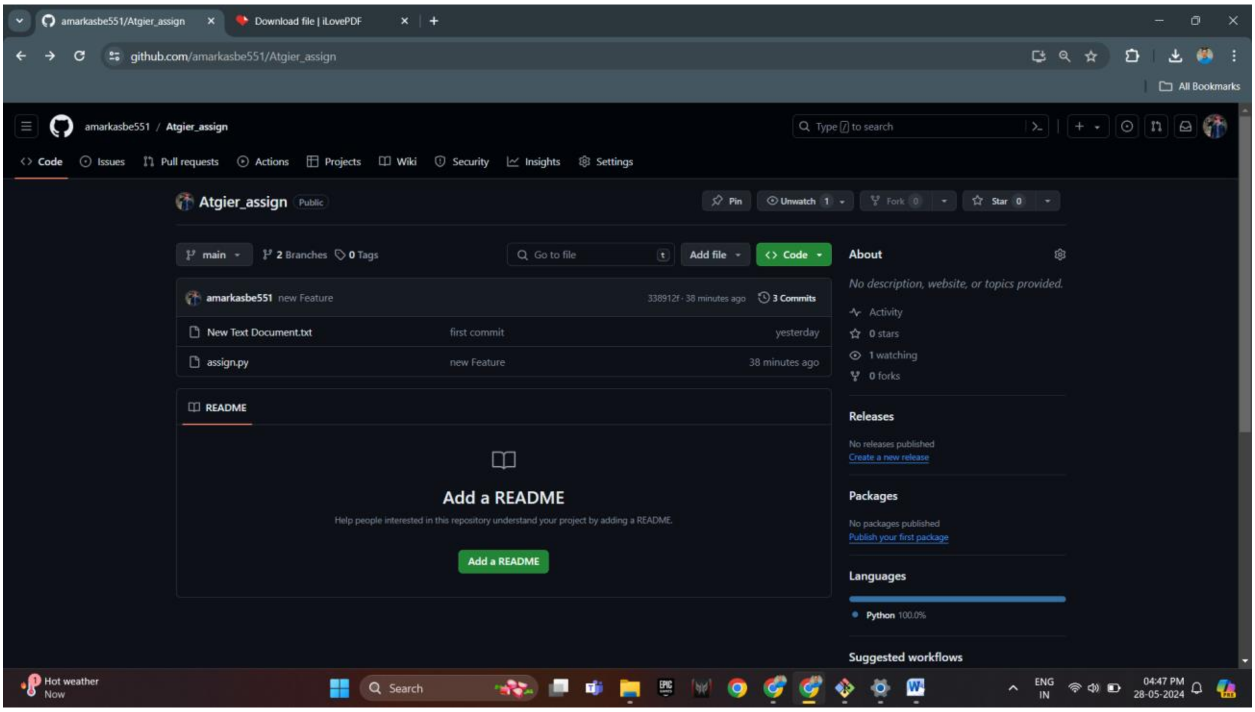
- Setup your own IDE.



```python
n = int(input("Enter Number:-"))
if n > 0:
    for _ in range(n):
        print("Meow")
```

```
PS C:\Users\ajayk\OneDrive\Desktop\atgeir> & C:/Users/ajayk/AppData/Local/Programs/Python/Python311/python.exe c:/Users/ajayk/OneDrive/Desktop/atgei
r/assign.py
Enter Number:-5
Meow
Meow
Meow
Meow
Meow
PS C:\Users\ajayk\OneDrive\Desktop\atgeir> █
```

- Setup your own GITHUB account, create your repository, create a file and commit it to the master branch.
  Create a new branch and commit a file to the repository.
  Merge both the branches.



Atgier_assign Public

main · 2 Branches · 0 Tags

amarkasbe551 new Feature · 338912f · 38 minutes ago · 3 Commits

New Text Document.txt	first commit	yesterday
assign.py	new Feature	38 minutes ago

README

Add a README

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

Add a README

About

No description, website, or topics provided.

Activity

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

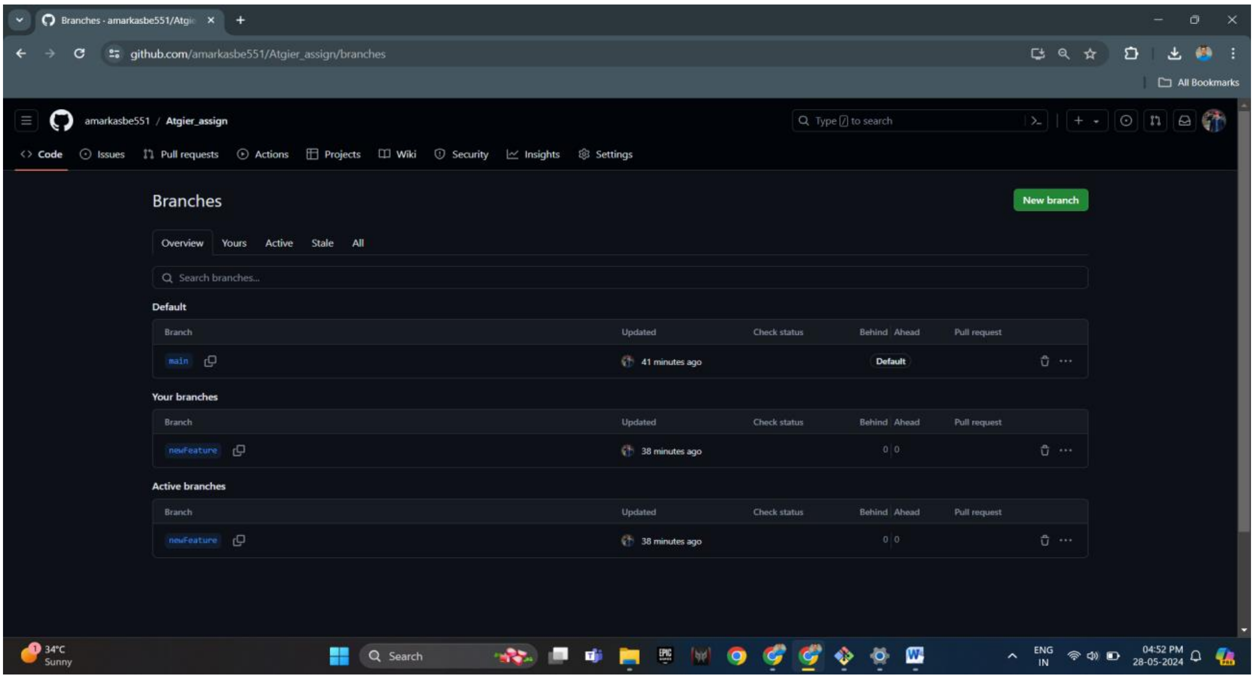
No packages published

Publish your first package

Languages

Python 100.0%

Suggested workflows



Branches

Default

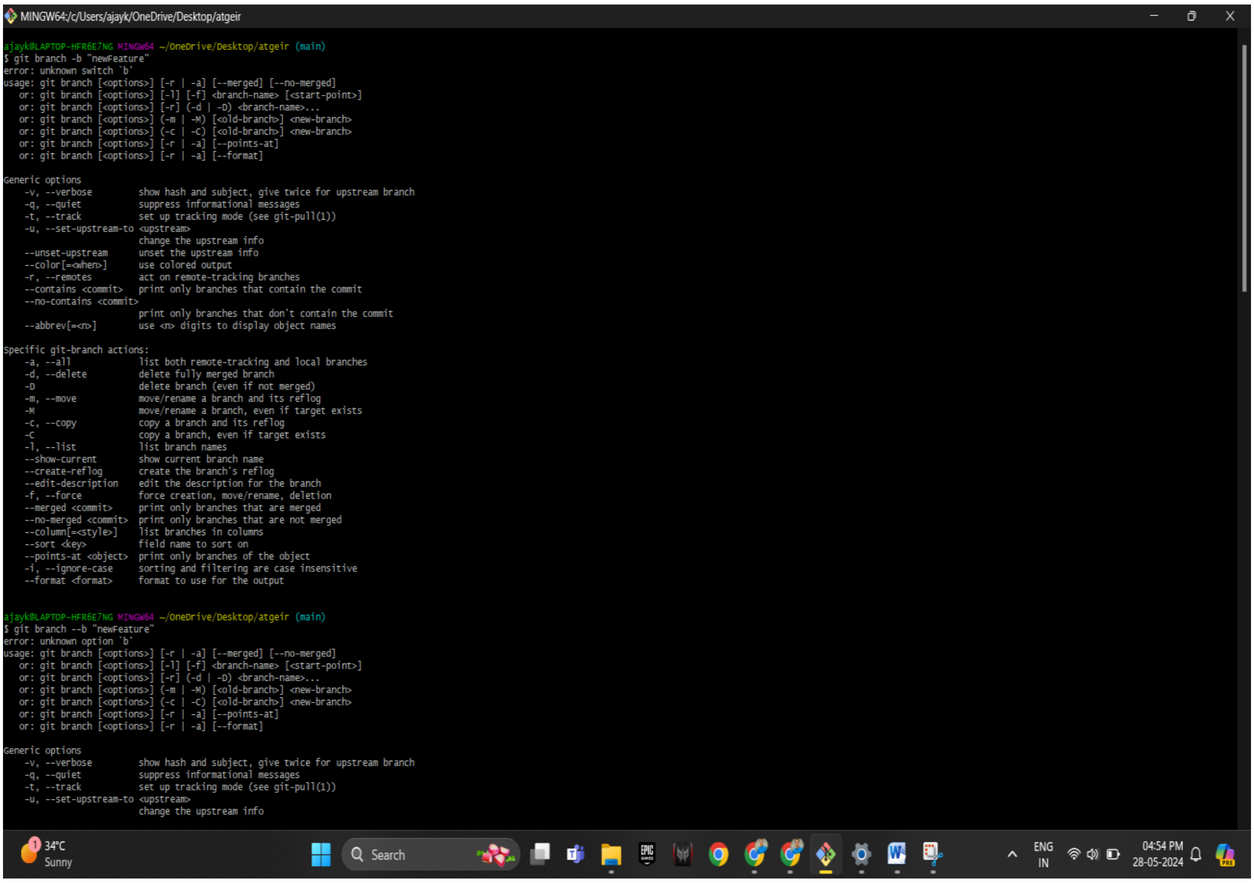
Branch	Updated	Check status	Behind | Ahead	Pull request
main	41 minutes ago		Default	

Your branches

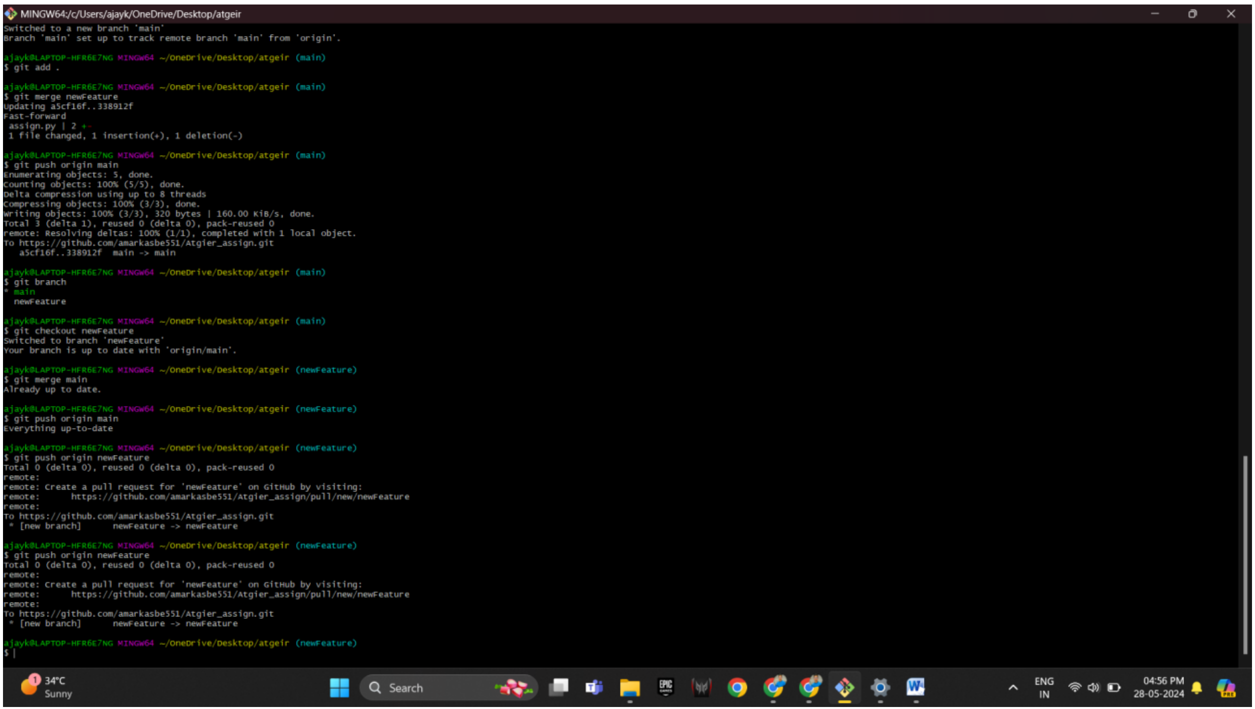
Branch	Updated	Check status	Behind | Ahead	Pull request
newFeature	38 minutes ago		0 | 0	

Active branches

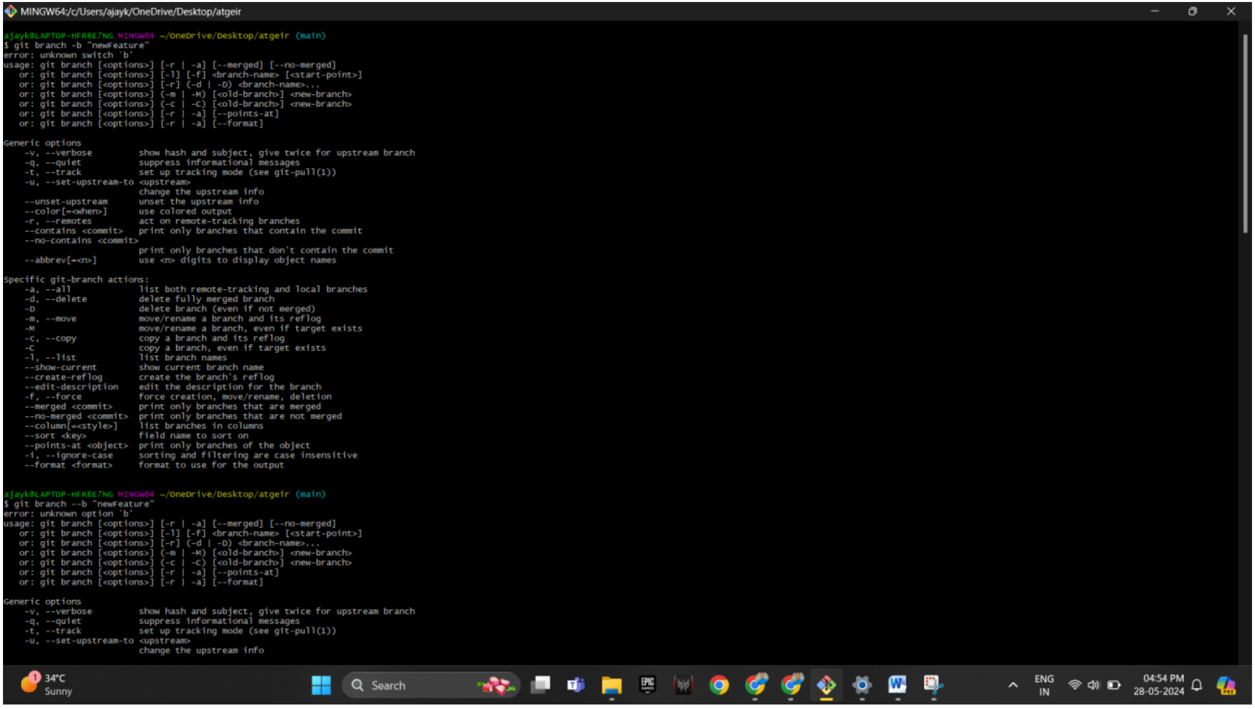
Branch	Updated	Check status	Behind | Ahead	Pull request
newFeature	38 minutes ago		0 | 0	



```
ajayk@LAPTOP-HFR6E7NG MINGW64 ~/OneDrive/Desktop/atgeir (main)
$ git branch -b "newFeature"
error: unknown switch `b'
usage: git branch [<options>] [-r | -a] [--merged] [--no-merged]
   or: git branch [<options>] [-l] [-f] <branch-name> [<start-point>]
   or: git branch [<options>] [-r] (-d | -D) <branch-name>...
   or: git branch [<options>] (-m | -M) [<old-branch>] <new-branch>
   or: git branch [<options>] (-c | -C) [<old-branch>] <new-branch>
   or: git branch [<options>] [-r | -a] [--points-at]
   or: git branch [<options>] [-r | -a] [--format]

Generic options
    -v, --verbose         show hash and subject, give twice for upstream branch
    -q, --quiet           suppress informational messages
    -t, --track           set up tracking mode (see git-pull(1))
    -u, --set-upstream-to <upstream>
                          change the upstream info
    --unset-upstream      unset the upstream info
    --color[=<when>]      use colored output
    -r, --remotes         act on remote-tracking branches
    --contains <commit>   print only branches that contain the commit
    --no-contains <commit>
                          print only branches that don't contain the commit
    --abbrev[=<n>]        use <n> digits to display object names

Specific git-branch actions:
    -a, --all             list both remote-tracking and local branches
    -d, --delete          delete fully merged branch
    -D                    delete branch (even if not merged)
    -m, --move            move/rename a branch and its reflog
    -M                    move/rename a branch, even if target exists
    -c, --copy            copy a branch and its reflog
    -C                    copy a branch, even if target exists
    -l, --list            list branch names
    --show-current        show current branch name
    --create-reflog       create the branch's reflog
    --edit-description    edit the description for the branch
    -f, --force           force creation, move/rename, deletion
    --merged <commit>     print only branches that are merged
    --no-merged <commit>  print only branches that are not merged
    --column[=<style>]    list branches in columns
    --sort <key>          field name to sort on
    --points-at <object>  print only branches of the object
    -i, --ignore-case     sorting and filtering are case insensitive
    --format <format>     format to use for the output


ajayk@LAPTOP-HFR6E7NG MINGW64 ~/OneDrive/Desktop/atgeir (main)
$ git branch --b "newFeature"
error: unknown option `b'
usage: git branch [<options>] [-r | -a] [--merged] [--no-merged]
   or: git branch [<options>] [-l] [-f] <branch-name> [<start-point>]
   or: git branch [<options>] [-r] (-d | -D) <branch-name>...
   or: git branch [<options>] (-m | -M) [<old-branch>] <new-branch>
   or: git branch [<options>] (-c | -C) [<old-branch>] <new-branch>
   or: git branch [<options>] [-r | -a] [--points-at]
   or: git branch [<options>] [-r | -a] [--format]

Generic options
    -v, --verbose         show hash and subject, give twice for upstream branch
    -q, --quiet           suppress informational messages
    -t, --track           set up tracking mode (see git-pull(1))
    -u, --set-upstream-to <upstream>
                          change the upstream info
```

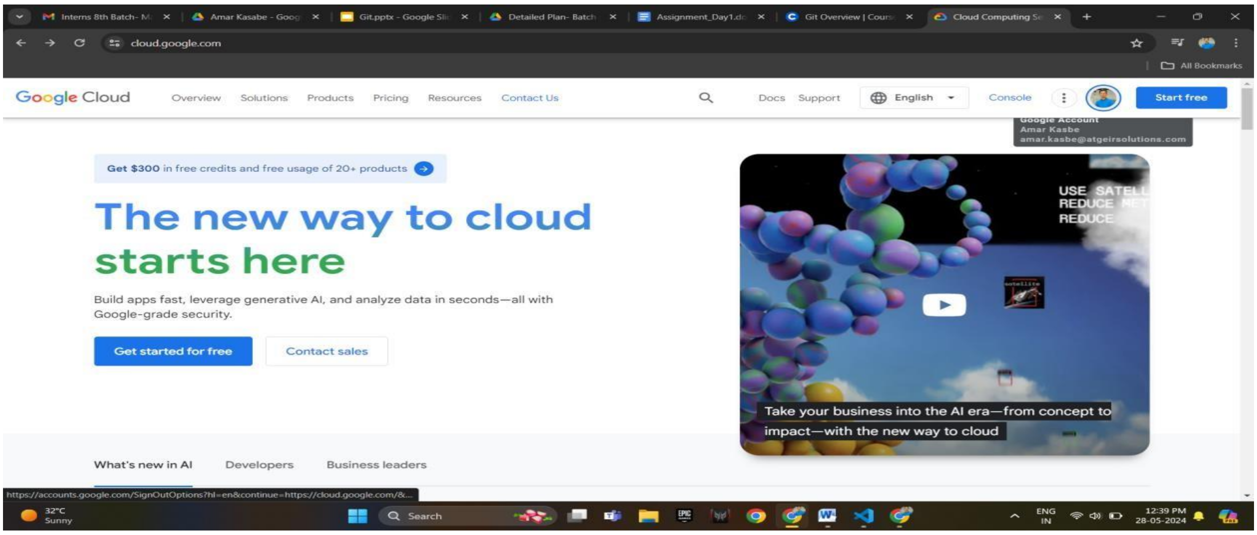


```
Switched to a new branch 'main'
Branch 'main' set up to track remote branch 'main' from 'origin'.

ajayk@LAPTOP-HFR6E7NG MINGW64 ~/OneDrive/Desktop/atgeir (main)
$ git add .

ajayk@LAPTOP-HFR6E7NG MINGW64 ~/OneDrive/Desktop/atgeir (main)
$ git merge newFeature
Updating a5cf16f..338912f
Fast-forward
 assign.py | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

ajayk@LAPTOP-HFR6E7NG MINGW64 ~/OneDrive/Desktop/atgeir (main)
$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 320 bytes | 160.00 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/amarkasbe551/Atgier_assign.git
   a5cf16f..338912f  main -> main

ajayk@LAPTOP-HFR6E7NG MINGW64 ~/OneDrive/Desktop/atgeir (main)
$ git branch
* main
  newFeature

ajayk@LAPTOP-HFR6E7NG MINGW64 ~/OneDrive/Desktop/atgeir (main)
$ git checkout newFeature
Switched to branch 'newFeature'
Your branch is up to date with 'origin/main'.

ajayk@LAPTOP-HFR6E7NG MINGW64 ~/OneDrive/Desktop/atgeir (newFeature)
$ git merge main
Already up to date.

ajayk@LAPTOP-HFR6E7NG MINGW64 ~/OneDrive/Desktop/atgeir (newFeature)
$ git push origin main
Everything up-to-date

ajayk@LAPTOP-HFR6E7NG MINGW64 ~/OneDrive/Desktop/atgeir (newFeature)
$ git push origin newFeature
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'newFeature' on GitHub by visiting:
remote:      https://github.com/amarkasbe551/Atgier_assign/pull/new/newFeature
remote:
To https://github.com/amarkasbe551/Atgier_assign.git
 * [new branch]      newFeature -> newFeature

ajayk@LAPTOP-HFR6E7NG MINGW64 ~/OneDrive/Desktop/atgeir (newFeature)
$ git push origin newFeature
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'newFeature' on GitHub by visiting:
remote:      https://github.com/amarkasbe551/Atgier_assign/pull/new/newFeature
remote:
To https://github.com/amarkasbe551/Atgier_assign.git
 * [new branch]      newFeature -> newFeature

ajayk@LAPTOP-HFR6E7NG MINGW64 ~/OneDrive/Desktop/atgeir (newFeature)
$ |
```



```
ajayk@LAPTOP-HFR6E7NG MINGW64 ~/OneDrive/Desktop/atgeir (main)
$ git branch -b "newFeature"
error: unknown switch `b'
usage: git branch [<options>] [-r | -a] [--merged] [--no-merged]
   or: git branch [<options>] [-l] [-f] <branch-name> [<start-point>]
   or: git branch [<options>] [-r] (-d | -D) <branch-name>...
   or: git branch [<options>] (-m | -M) [<old-branch>] <new-branch>
   or: git branch [<options>] (-c | -C) [<old-branch>] <new-branch>
   or: git branch [<options>] [-r | -a] [--points-at]
   or: git branch [<options>] [-r | -a] [--format]

Generic options
    -v, --verbose         show hash and subject, give twice for upstream branch
    -q, --quiet           suppress informational messages
    -t, --track           set up tracking mode (see git-pull(1))
    -u, --set-upstream-to <upstream>
                          change the upstream info
    --unset-upstream      unset the upstream info
    --color[=<when>]      use colored output
    -r, --remotes         act on remote-tracking branches
    --contains <commit>   print only branches that contain the commit
    --no-contains <commit>
                          print only branches that don't contain the commit
    --abbrev[=<n>]        use <n> digits to display object names

Specific git-branch actions:
    -a, --all             list both remote-tracking and local branches
    -d, --delete          delete fully merged branch
    -D                    delete branch (even if not merged)
    -m, --move            move/rename a branch and its reflog
    -M                    move/rename a branch, even if target exists
    -c, --copy            copy a branch and its reflog
    -C                    copy a branch, even if target exists
    -l, --list            list branch names
    --show-current        show current branch name
    --create-reflog       create the branch's reflog
    --edit-description    edit the description for the branch
    -f, --force           force creation, move/rename, deletion
    --merged <commit>     print only branches that are merged
    --no-merged <commit>  print only branches that are not merged
    --column[=<style>]    list branches in columns
    --sort <key>          field name to sort on
    --points-at <object>  print only branches of the object
    -i, --ignore-case     sorting and filtering are case insensitive
    --format <format>     format to use for the output


ajayk@LAPTOP-HFR6E7NG MINGW64 ~/OneDrive/Desktop/atgeir (main)
$ git branch --b "newFeature"
error: unknown option `b'
usage: git branch [<options>] [-r | -a] [--merged] [--no-merged]
   or: git branch [<options>] [-l] [-f] <branch-name> [<start-point>]
   or: git branch [<options>] [-r] (-d | -D) <branch-name>...
   or: git branch [<options>] (-m | -M) [<old-branch>] <new-branch>
   or: git branch [<options>] (-c | -C) [<old-branch>] <new-branch>
   or: git branch [<options>] [-r | -a] [--points-at]
   or: git branch [<options>] [-r | -a] [--format]

Generic options
    -v, --verbose         show hash and subject, give twice for upstream branch
    -q, --quiet           suppress informational messages
    -t, --track           set up tracking mode (see git-pull(1))
    -u, --set-upstream-to <upstream>
                          change the upstream info
```

Create your GCP account.



Get $300 in free credits and free usage of 20+ products

The new way to cloud starts here

Build apps fast, leverage generative AI, and analyze data in seconds—all with Google-grade security.

Get started for free

Contact sales

Take your business into the AI era—from concept to impact—with the new way to cloud