Forking

Contents

Overview

In layman's terms, forking is creating a copy of an existing repository under your account.

Forking a repository allows you to freely experiment with the existing code base, without the fear of breaking the project.
Forking also gives you the ability to contribute to a project, by adding additional functionality.

Proposing a change

Imagine you are using someone's project and you find a bug. Usually, you would raise an issue for this; however, forking means you would be able to attempt a fix yourself.
The steps would be:

1. Fork the repository
2. Create a new branch
3. Fix the issue
4. Submit a pull request to the owner of the original project

If the owner approves your fix, your work should then be merged into the original repository.

Your idea

If a project is under public license that allows you to freely use it, you could use the project as a starting point for
your own project. For example, if you found a web application that you liked, and it was under the public license,
you could fork the project and add any additional functionality. This would then be yours to freely use or distribute.

Tutorial

Forking a repository

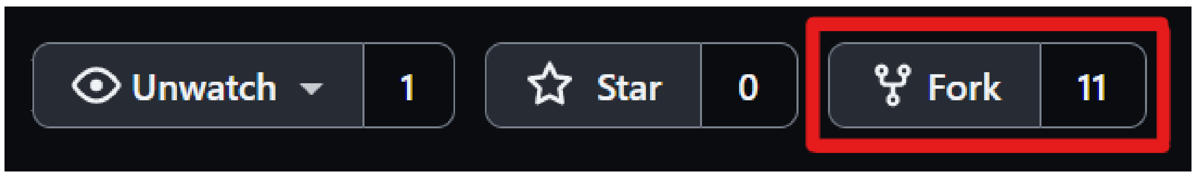
You will now: fork a remote repository, configure upstream (to get updates from the original repository) and propose a change to the owner

1. Make sure you're logged into your GitHub account

2. Go to any public repository available. This one for example.

3. To fork the repository, click on `Fork` (in the top-right of the page):

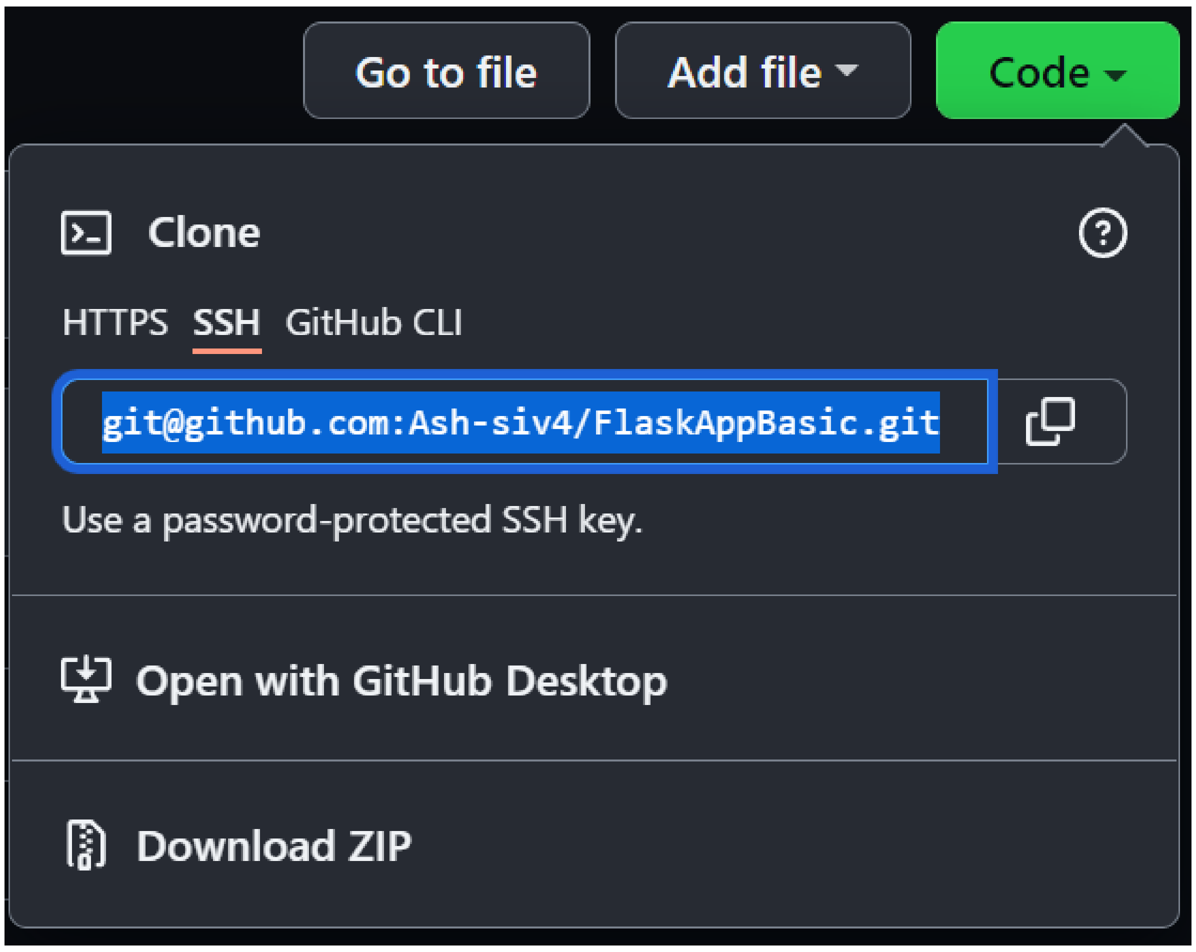
Express-Testing
Networking
Security
Cloud Fundamentals
AWS Foundations
AWS Intermediate
Linux
DevOps
Jenkins Introduction
Jenkins Pipeline
Markdown
IDE Cheatsheet



This will create a copy of the repository under your account.

4. You will then be redirected to your account's version of the repository

5. Click on the green `Code` button, and **copy** the `SSH` URL



git@github.com:Ash-siv4/FlaskAppBasic.git

Use a password-protected SSH key.

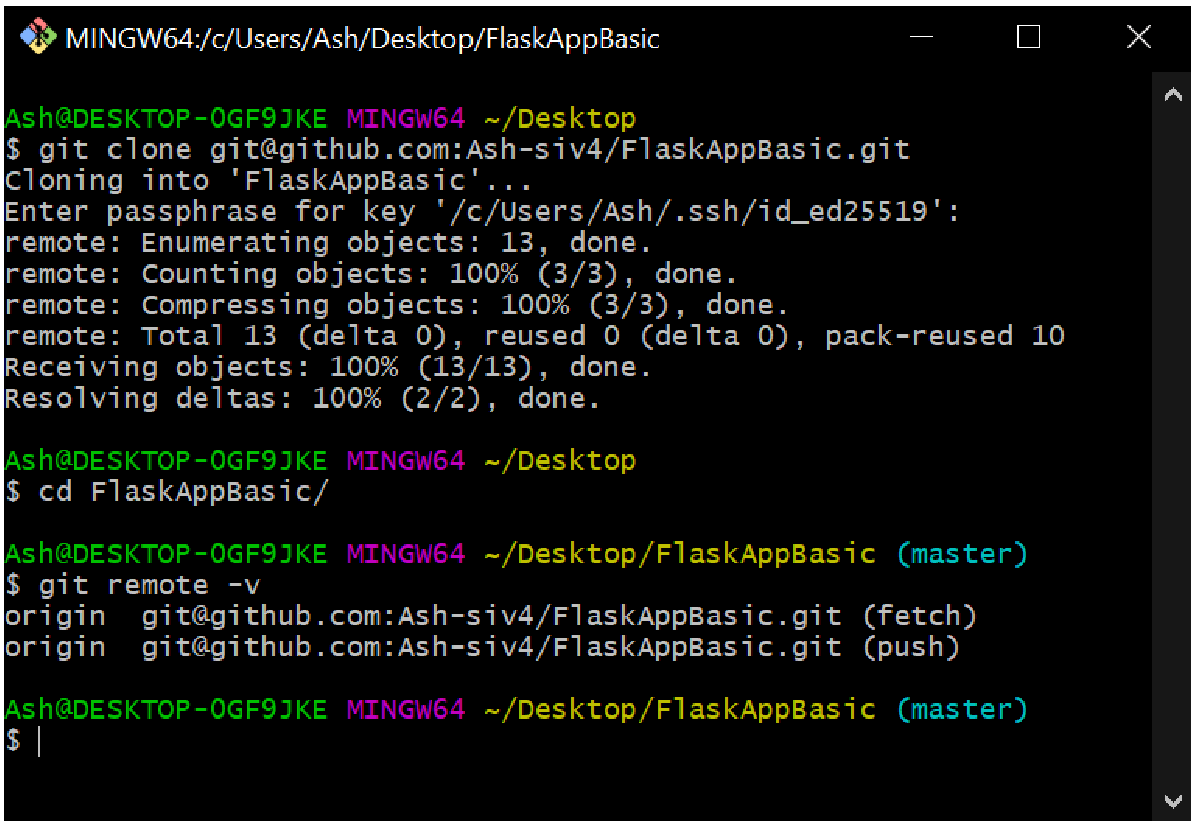
6. Open a Git Bash terminal in the location you want to clone your project into

7. Run the command `git clone [URL]`, but replace [URL] with the repository URL you copied in the previous step

8. Change directory to the project you just cloned

9. Run the following command, to confirm that you have cloned the correct repository: `git remote -v`.

> You should get a similar output for `fetch` and `push`, but the URL will be pointing to your repository



```
Ash@DESKTOP-OGF9JKE MINGW64 ~/Desktop
$ git clone git@github.com:Ash-siv4/FlaskAppBasic.git
Cloning into 'FlaskAppBasic'...
Enter passphrase for key '/c/Users/Ash/.ssh/id_ed25519':
remote: Enumerating objects: 13, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 13 (delta 0), reused 0 (delta 0), pack-reused 10
Receiving objects: 100% (13/13), done.
Resolving deltas: 100% (2/2), done.

Ash@DESKTOP-OGF9JKE MINGW64 ~/Desktop
$ cd FlaskAppBasic/

Ash@DESKTOP-OGF9JKE MINGW64 ~/Desktop/FlaskAppBasic (master)
$ git remote -v
origin  git@github.com:Ash-siv4/FlaskAppBasic.git (fetch)
origin  git@github.com:Ash-siv4/FlaskAppBasic.git (push)

Ash@DESKTOP-OGF9JKE MINGW64 ~/Desktop/FlaskAppBasic (master)
$ |
```

Updating forked repository from original

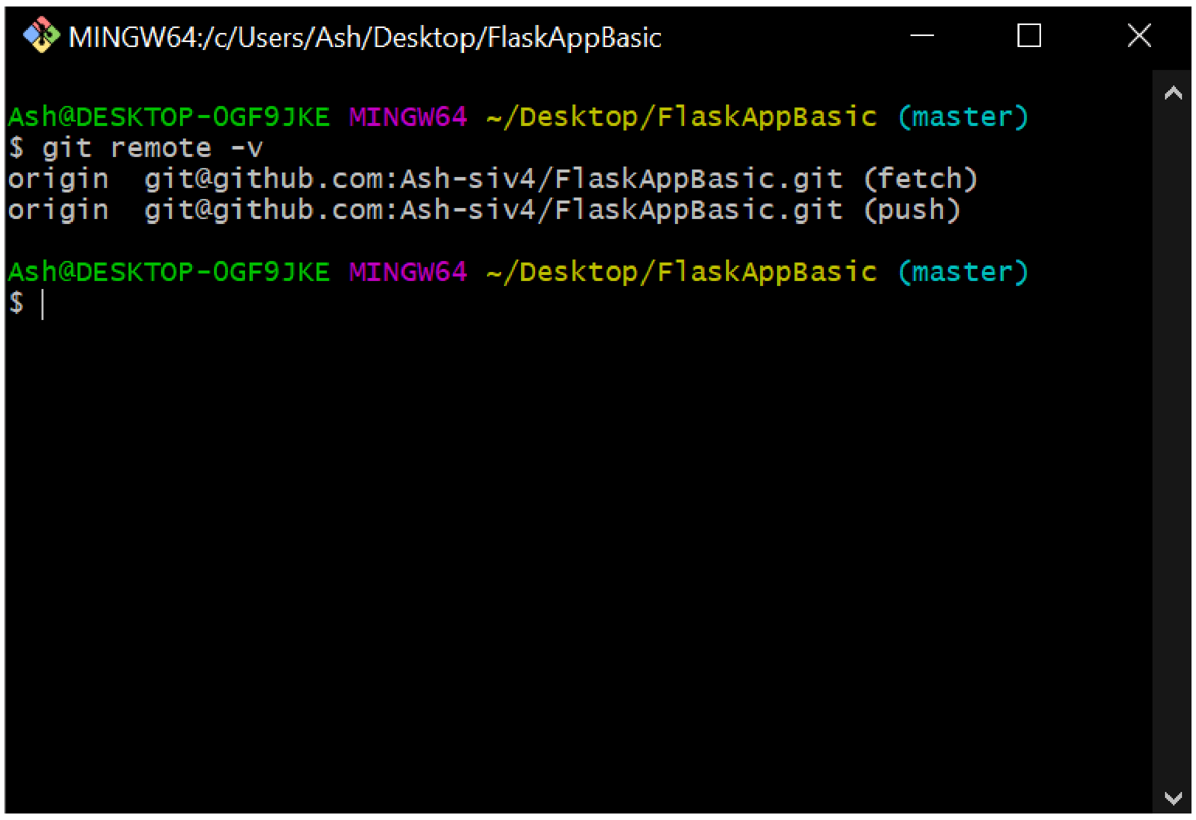
Now you will set up your local project to receive updates from the original repository. This is required, so that when
the owner of the original repository adds new functionality, bug fixes etc., you

will be able to get these changes as well.

1. Open your Bash terminal, in the root directory of your project. You should be able to see that you're on the main branch.

   Next, you want to execute `git remote -v` to make sure that the upstream to the original repository is not set up yet.

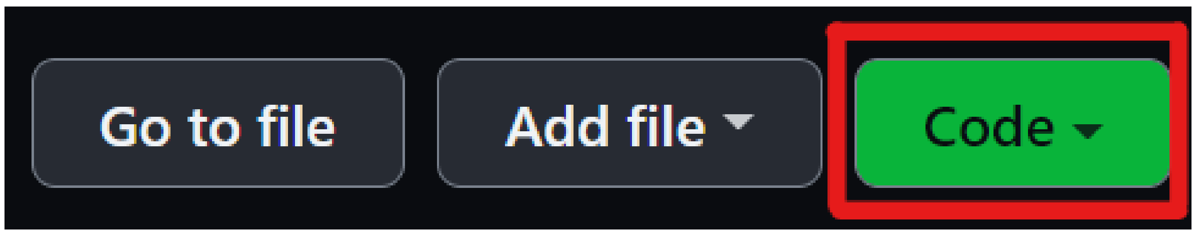
   You should see a similar output to this:

   ```
   Ash@DESKTOP-OGF9JKE MINGW64 ~/Desktop/FlaskAppBasic (master)
   $ git remote -v
   origin  git@github.com:Ash-siv4/FlaskAppBasic.git (fetch)
   origin  git@github.com:Ash-siv4/FlaskAppBasic.git (push)

   Ash@DESKTOP-OGF9JKE MINGW64 ~/Desktop/FlaskAppBasic (master)
   $ |
   ```

2. In your browser, go to the original git repository and copy the repository URL.

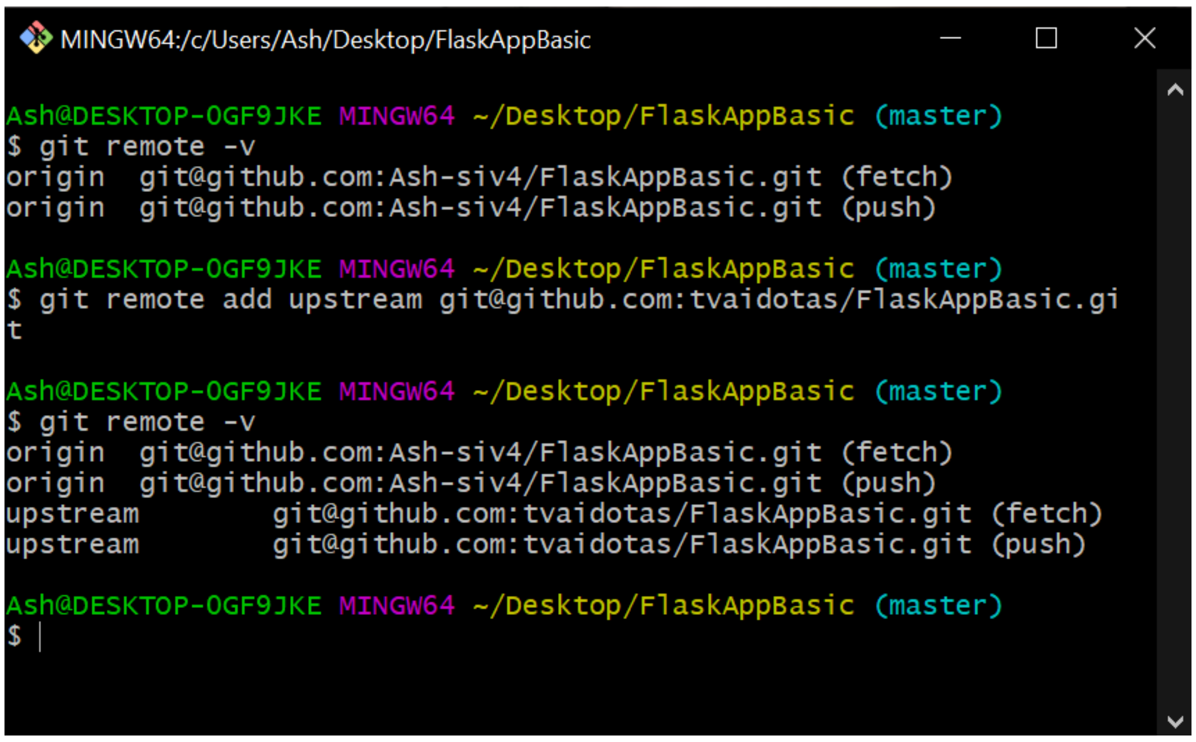
   You can do this by clicking on the green `Code` button and copying the `SSH` URL:

3. Execute the following command in your Bash terminal, but replace the [URL] with the one you just copied: `git remote add upstream [URL]`

4. Next, you want to check that this has worked, which you can do by executing the `git remote -v` command.

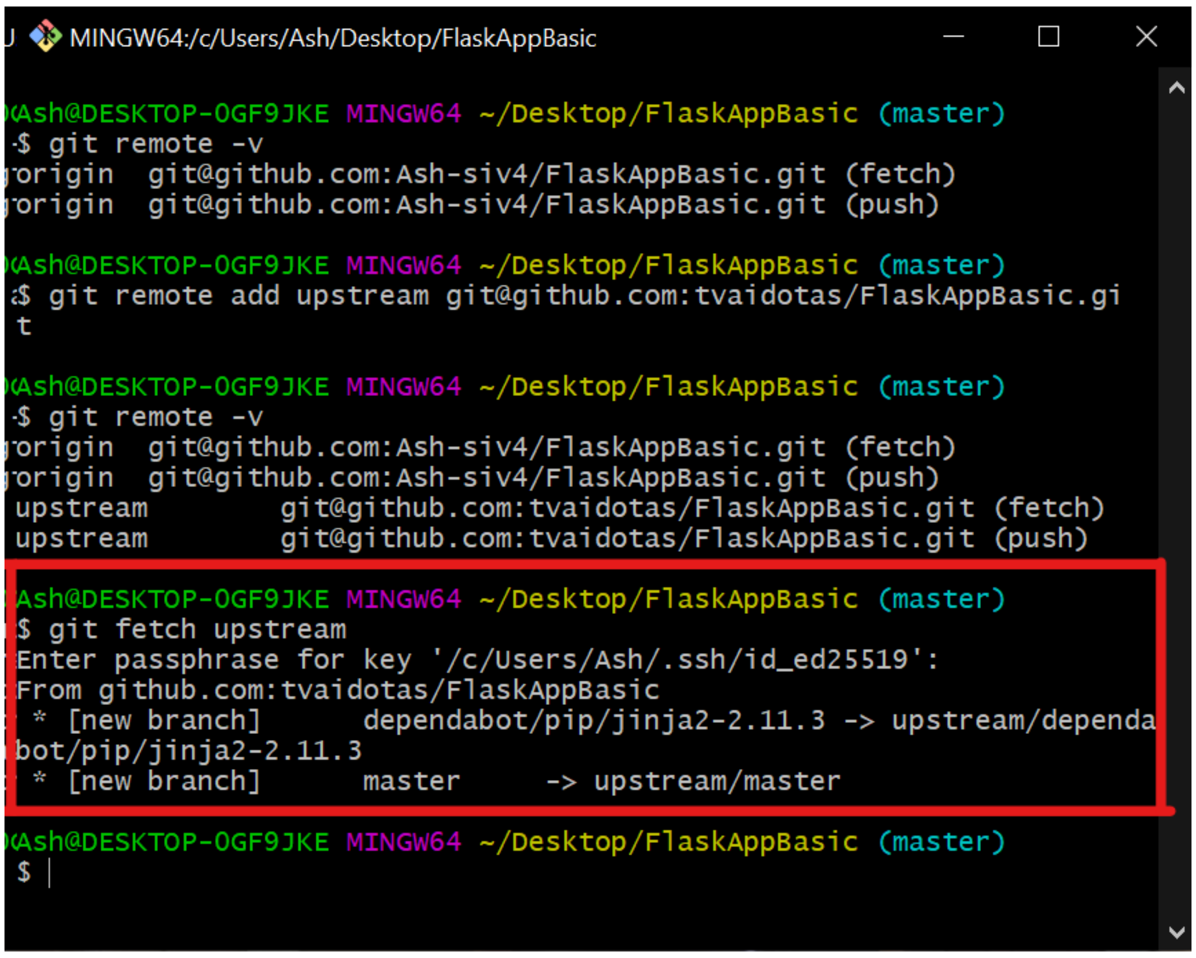
   The output should look similar to this:

   ```
   Ash@DESKTOP-OGF9JKE MINGW64 ~/Desktop/FlaskAppBasic (master)
   $ git remote -v
   origin  git@github.com:Ash-siv4/FlaskAppBasic.git (fetch)
   origin  git@github.com:Ash-siv4/FlaskAppBasic.git (push)

   Ash@DESKTOP-OGF9JKE MINGW64 ~/Desktop/FlaskAppBasic (master)
   $ git remote add upstream git@github.com:tvaidotas/FlaskAppBasic.gi
   t

   Ash@DESKTOP-OGF9JKE MINGW64 ~/Desktop/FlaskAppBasic (master)
   $ git remote -v
   origin  git@github.com:Ash-siv4/FlaskAppBasic.git (fetch)
   origin  git@github.com:Ash-siv4/FlaskAppBasic.git (push)
   upstream        git@github.com:tvaidotas/FlaskAppBasic.git (fetch)
   upstream        git@github.com:tvaidotas/FlaskAppBasic.git (push)

   Ash@DESKTOP-OGF9JKE MINGW64 ~/Desktop/FlaskAppBasic (master)
   $ |
   ```

5. Now you will pull the changes from the original repository into yours, which you can do by executing the `git fetch upstream` command in your Git Bash terminal.

   The output will depend on whether there are new changes or not:



```
Ash@DESKTOP-0GF9JKE MINGW64 ~/Desktop/FlaskAppBasic (master)
$ git remote -v
origin  git@github.com:Ash-siv4/FlaskAppBasic.git (fetch)
origin  git@github.com:Ash-siv4/FlaskAppBasic.git (push)

Ash@DESKTOP-0GF9JKE MINGW64 ~/Desktop/FlaskAppBasic (master)
$ git remote add upstream git@github.com:tvaidotas/FlaskAppBasic.gi
t

Ash@DESKTOP-0GF9JKE MINGW64 ~/Desktop/FlaskAppBasic (master)
$ git remote -v
origin  git@github.com:Ash-siv4/FlaskAppBasic.git (fetch)
origin  git@github.com:Ash-siv4/FlaskAppBasic.git (push)
upstream        git@github.com:tvaidotas/FlaskAppBasic.git (fetch)
upstream        git@github.com:tvaidotas/FlaskAppBasic.git (push)

Ash@DESKTOP-0GF9JKE MINGW64 ~/Desktop/FlaskAppBasic (master)
$ git fetch upstream
Enter passphrase for key '/c/Users/Ash/.ssh/id_ed25519':
From github.com:tvaidotas/FlaskAppBasic
 * [new branch]      dependabot/pip/jinja2-2.11.3 -> upstream/dependa
bot/pip/jinja2-2.11.3
 * [new branch]      master     -> upstream/master

Ash@DESKTOP-0GF9JKE MINGW64 ~/Desktop/FlaskAppBasic (master)
$ |
```

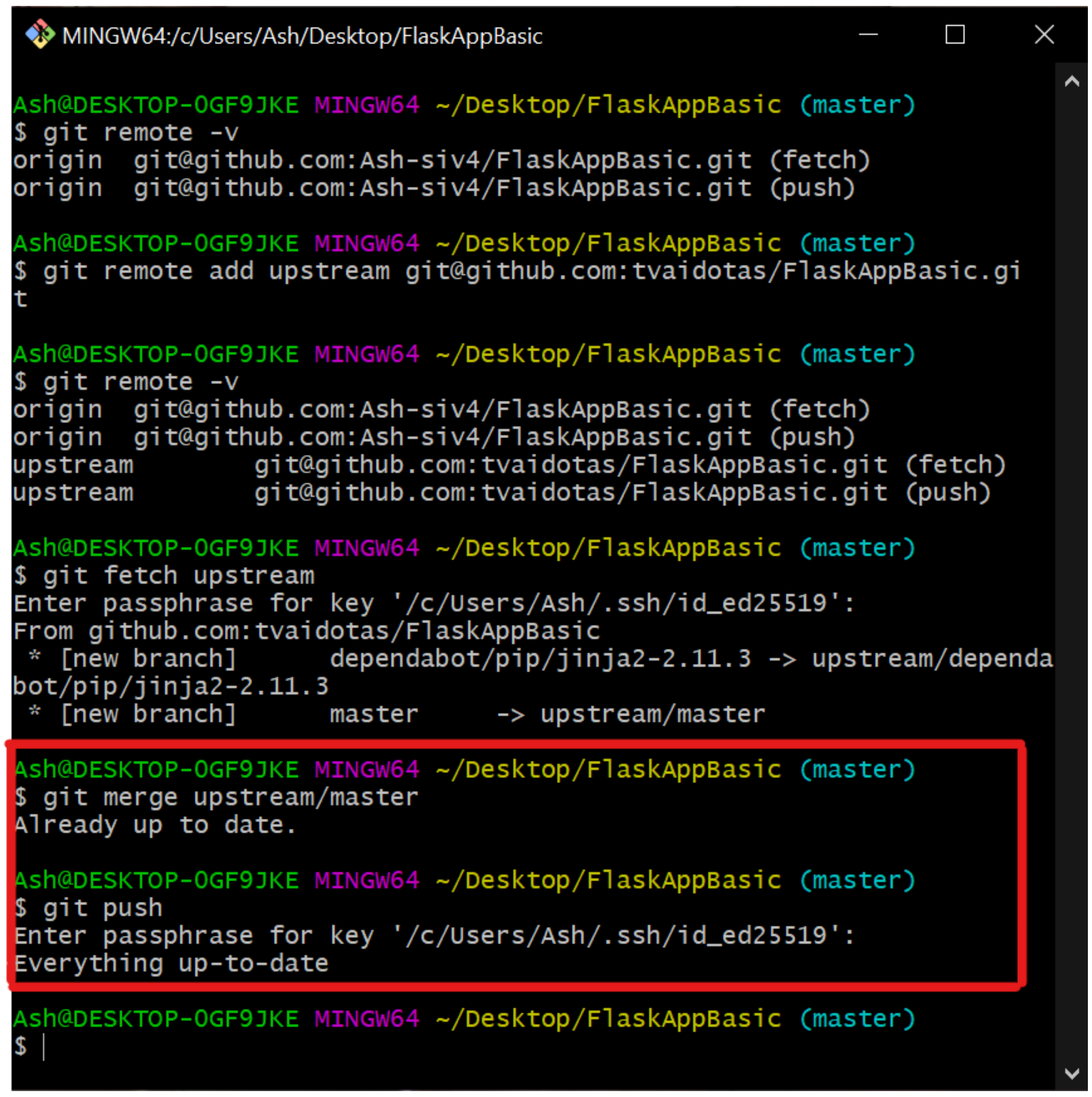
6. You will want to update the main branch, so we will merge the *upstream/main* into the *local origin/main*.

   You can do this by executing the `git merge upstream/[name_of_main_branch` command.

   > Example command: **git merge upstream/main**

   Keep in mind that there may be merge conflicts that you will need to resolve.

   If you have resolved the merge conflicts, or if there were none, you should *push* to update your *origin/main* (or *origin/master*) branch.



```
Ash@DESKTOP-0GF9JKE MINGW64 ~/Desktop/FlaskAppBasic (master)
$ git remote -v
origin  git@github.com:Ash-siv4/FlaskAppBasic.git (fetch)
origin  git@github.com:Ash-siv4/FlaskAppBasic.git (push)

Ash@DESKTOP-0GF9JKE MINGW64 ~/Desktop/FlaskAppBasic (master)
$ git remote add upstream git@github.com:tvaidotas/FlaskAppBasic.gi
t

Ash@DESKTOP-0GF9JKE MINGW64 ~/Desktop/FlaskAppBasic (master)
$ git remote -v
origin  git@github.com:Ash-siv4/FlaskAppBasic.git (fetch)
origin  git@github.com:Ash-siv4/FlaskAppBasic.git (push)
upstream        git@github.com:tvaidotas/FlaskAppBasic.git (fetch)
upstream        git@github.com:tvaidotas/FlaskAppBasic.git (push)

Ash@DESKTOP-0GF9JKE MINGW64 ~/Desktop/FlaskAppBasic (master)
$ git fetch upstream
Enter passphrase for key '/c/Users/Ash/.ssh/id_ed25519':
From github.com:tvaidotas/FlaskAppBasic
 * [new branch]      dependabot/pip/jinja2-2.11.3 -> upstream/dependa
bot/pip/jinja2-2.11.3
 * [new branch]      master     -> upstream/master

Ash@DESKTOP-0GF9JKE MINGW64 ~/Desktop/FlaskAppBasic (master)
$ git merge upstream/master
Already up to date.

Ash@DESKTOP-0GF9JKE MINGW64 ~/Desktop/FlaskAppBasic (master)
$ git push
Enter passphrase for key '/c/Users/Ash/.ssh/id_ed25519':
Everything up-to-date

Ash@DESKTOP-0GF9JKE MINGW64 ~/Desktop/FlaskAppBasic (master)
$ |
```

Exercises

There are no exercises for this module.