Optional: Git + Heroku commands

**If you are curious how the below command works, read on...**

1. $ git push heroku master

There are always few other steps to execute: Installing Git and Heroku, creating a local Git repo, signing-up to heroku, log-in heroku via command-line, creating heroku handle to hosting point (*explained in PART 2*)

**1. A local Git repository:**

1. $ git add .
2. $ git commit -m "my first commit"
3. Created initial commit 5df2d09: my first commit
4. 44 files changed, 8393 insertions(+), 0 deletions(-)
5. create mode 100644 README
6. create mode 100644 Procfile
7. create mode 100644 app/controllers/source\_file
8. ...

**2. Have sign-up(ed) for Heroku and logged-in via command-line:**

1. $ heroku login // or you can do this online
2. Enter your Heroku credentials.
3. Email: user@example.com
4. Password:
5. Could not find an existing public key.
6. Would you like to generate one? [Yn]
7. Generating new SSH public key.
8. Uploading ssh public key /Users/adam/.ssh/id\_rsa.pub

**PART 2: but what does *heroku* and *master* indicate?**

It is more of a Git question than Heroku - Heroku is a hosting platform, which depends on Git (Distributed Version Control System) for deployment.

The basic concept of 'push' is pushing some thing (file, app, ..) we have locally (in our working machine) to somewhere else, in this case to a remote repository (remote machine).

In Git before using 'push' we create a remote (handle) which acts as a reference to a remote repository (Complete URL), we do so using the following command:

1. $ git remote add <remote-name-of-our-choice> <URL-where-you-be-pushing-yourapp>

The basic structure of 'push' command is:

1. $ git push <remote-name> <branch>

So $ git push heroku master is actually pushing your code/app/file (from some local Git repo) to a remote repo 'heroku' .

wondering when this 'heroku' remote got created, it was added when you executed ***$ heroku create***

1. $ heroku create
2. Creating stark-fog-398... done, stack is cedar
3. http://stark-fog-398.herokuapp.com/ | git@heroku.com:stark-fog-398.git
4. Git remote heroku added

Do notice the last line "**Git remote heroku added**".

to make it more clear, here's a Git command to check/output all the remotes: $ git remote -v will display something similar to the following

1. $ git remote -v
2. heroku git@heroku.com:somerepo.git (fetch)
3. heroku git@heroku.com:somerepo.git (push)

*So we can assume that the following command was executed (implicitly) somewhere, when you did $ heroku create* , hence creating the heroku remote to some heroku repo (url)\*

1. $ git remote add heroku git@heroku.com:somerepo.git

- Thank you to Nabeel Ahmed for this great explanation!