

OPINION

How to Deploy Streamlit on Heroku

For Endless Possibilities in Data Science



Sohaib Ahmad

[Follow](#)

Jul 1 · 5 min read ★



Photo by Kevin Ku on Unsplash

In a previous post, I predicted that the popularity of `Flask` would really take a hit once `Streamlit` comes more into the mainstream. I also made the comment that I would never use Flask again.

I still stand by both of these comments.

In that time, I've made four different machine learning projects, all of which are being used by family and friends on a daily basis, including:

1. A COVID Dashboard for my local friends and family that focuses in on our local area
2. A simple application for restaurants to take bookings (help out my fathers business)
3. A small game involving a face mask recognition system for my nephew

None of these are for financial gain, rather, they're made exactly for what I've always enjoyed about artificial intelligence: they're fun, novel and creative projects that just cool.

I can now develop, code and deploy novel applications in less than a couple of hours.

For those who don't know, `Heroku` is a platform as a service (PaaS) that enables developers to build, run, and operate applications entirely in the cloud. It's super useful when you want to build something small that scales, but more importantly, it's really helpful for small pet projects.

The free tier is something that I really value and would really recommend readers to deploy more projects so that they're free for everyone to play with. What's the point of making cool stuff if you can't show anyone?





Photo by Paul Green on Unsplash

In what follows, I'll take you through the different steps required. The first or second time it may be a bit slow, but after, you'll fly through it.

Note: the following instructions are only verbose to a degree to help you deploy simple applications, without much thought of scalability etc. My advice is targeted at individuals who want to build small applications, so if you expect to have 100m users, this is not really the tutorial for you

Firstly: make sure you have a `Heroku` login! Use this link and navigate through making a free tier. If you struggle here: I'll be very disappointed.

Requirement .txt

Make sure that your `terminal` is in your project folder. When you launch your project into the cloud, you need to create a `requirements.txt` file so that the server knows what to download to run your code. The library `pipreqs` can autogenerate requirements (it's pretty handy) so I'd recommend installing it as follows:

```
pip install pipreqs
```

Then once it's downloaded, just step out of the folder, run the following command, and in the folder, you should find your `requirements.txt` file.

```
pipreqs <directory path>
```

It should contain libraries and their versions in the following format:

```
numpy==1.15.0
pandas==0.23.4
streamlit==0.60.0
xlrd==1.2.0
```

setup.sh and Procfile

Now, the next step is a little bit messy but bear with me. You need to set up two more things: a `setup.sh` file, and a `Procfile`.

The `setup.sh` file contains some commands to set the problem on the `Heroku` side, so create a `setup.sh` file (you can use the `nano` command) and save the following in that file (change the email in the middle of the file to your correct email)

```
mkdir -p ~/.streamlit/
echo "\
[general]\n\
email = \"your@domain.com\"\n\
" > ~/.streamlit/credentials.toml
echo "\
[server]\n\
headless = true\n\
enableCORS=false\n\
port = $PORT\n\
"
```

Nice!

Now, make (and I'm cheating a little bit here), but make a procfile using the command:

```
nano Procfile
```

and from there, you want to insert the following piece of code (*remember to replace `[name-of-app].py` to whatever your app is called. Most of mine are just `app.py`*)

```
web: sh setup.sh && streamlit run [name-of-app].py
```

Moving the files across with Git

Heroku builds systems using Git and it's insanely easy to get set up. Git is a version control system and runs as default on a lot of operating systems. Check if you have it, if not, install it.

Once you're happy with your installation, you'll need to make a git repository in your project folder. To do this, make sure you are in your project folder and run the following command:

```
git init
```

This initialises a git repository in your project folder and you should see something like the following print out:

```
Initialized empty Git repository in /Users/...
```

Nice!

The first time you do this you'll need to click [here](#) and install the `Heroku CLI`. We're using the free version of `Heroku` which is great but naturally has drawbacks as it doesn't have certain desirable features configured. The features are more useful for larger projects (like SSL and scalability, also our machines tend to go to sleep if they are idle for more than 30 minutes) but hey, it's free!

Once you've downloaded the `Heroku CLI`, run the following login command:

```
C:\Users...> heroku login
```

This opens up a browser window, from which you can log in.

Once you're in, it's time to create your cloud instance. Run the following command

```
C:\Users...> heroku create
```

and you'll see that Heroku will create some oddly named instance (don't worry, it's just how it is):

```
Creating app... done, 🍷 true-poppy-XXXX  
  
https://true-poppy-XXXXXX.herokuapp.com/ |  
https://git.heroku.com/true-poppy-XXXXXX.git
```

So `Heroku` created an app called 'true-poppy' for me. Not sure why, but I'll take it. Now all that's remaining is to move the code across, so in our project folder we run the following commands:

```
git add .  
git commit -m "Enter your message here"  
git push heroku master
```

Once it's merged, the `Heroku` application will start downloading and installing everything on the server side. This takes a couple of minutes but if all is good, then you should something like:

```
remote: Verifying deploy... done.
```

Now if you run the following:

```
heroku ps:scale web=1
```

Your job is done! If you copy the url it gives you in the command line into your browser, you'll see that you can now run your application online. You can even check it on your mobile phone, it's perfect!

. . .

Streamlit and Heroku make a phenomenal combination. Once you've done the actual hard-work of creating a machine learning model that makes sense and generates sensible results, the difficult part should never be the deployment. You should want people to play with a working product. `Streamlit` takes care of so much of the aesthetic and `Heroku` takes care of the rest.

Yes, there are several drawbacks to the above methodology in that it's limited in scope, design and scalability but I challenge you to find a quicker way of deploying a half-decent and fully functional MVP that users can interact with. Even better, I'll even do a race!

I really recommend these pragmatic approaches because without users interacting with your product — you'll just never know if it's any good.

Give it a go. Surprise yourself.

. . .

Thanks for reading! If you have any messages, please let me know!

Keep up to date with my latest articles here!

Sign up for The Daily Pick

By Towards Data Science

Hands-on real-world examples, research, tutorials, and cutting-edge techniques delivered Monday to Thursday. Make learning your daily ritual. [Take a look](#)

Get this newsletter

Emails will be sent to
kevin.zhangdongzhao@gmail.com.
[Not you?](#)

[Programming](#) [Coding](#) [Data Science](#) [Software Development](#) [Python](#)

[About](#) [Help](#) [Legal](#)

Get the Medium app



A button that says 'Download on the App Store', and if clicked it will lead you to the iOS App store



A button that says 'Get it on, Google Play', and if clicked it will lead you to the Google Play store