



Mark Miyashita

negativetwelve

[Student \(/resume/#education\)](#), [Software Engineer \(/resume/#work\)](#), [Teacher \(/cs61a\)](#), [Aspiring Entrepreneur \(/blog\)](#)

Install Heroku and Deploy Your App

- Introduction
- Installation
- Deploying Your App
 - Ruby on Rails
 - Node.js
 - Django

Introduction

So, you've finished adding the final touches to your app and now you want to put it on the Internet for everyone else to use. How do I do that?!

Well, there are several options available to you at this point, most of them are free. The most popular free option is to use Heroku and I'm going to be walking through the steps to set up Heroku on your computer and deploy your first app.

First, let's talk a little about what Heroku is. Heroku is a cloud application platform. In one sentence, Heroku allows you to serve your web applications using their servers and allows other computers to access your site at the specified url. Heroku has mastered the art of simplicity, and setting up an app and deploying has never been easier. Let's dive into the steps necessary to do this.

Since Heroku is a service, we're first going to have to sign up for an account. You can do that here (<https://id.heroku.com/signup/devcenter>).

Installation

- First, we need to download the Heroku Toolbelt (<https://toolbelt.heroku.com/>).

The Heroku Toolbelt install a bunch of command line tools that allow you to easily create and edit apps straight from your own computer. It mainly gives us access to the **heroku** command that allows us to do a lot of different heroku-specific commands.

- Next, let's log in to our Heroku accounts using the command line. Run the following command:

```
$ heroku login
Enter your Heroku credentials.
Email: example@example.com
Password:
Could not find an existing public key.
Would you like to generate one? [Yn]
Generating new SSH public key.
Uploading ssh public key /Users/mark/.ssh/id_rsa.pub
```

If prompted to generate a public key, go ahead and enter "Y" and proceed to do so.

That's it for installation! Now we just need to figure out how to deploy our apps using the new command line interface that we just installed. Go on to the next section if you're ready.

Deploying Your App

Heroku can deploy a bunch of apps and you can find different tutorials listed on their website here (<https://devcenter.heroku.com/articles/quickstart#step-4-deploy-an-application>). I'm going to outline the three most popular: Ruby on Rails, Node.js, and Django.

Ruby on Rails

If you already have an existing Rails app, you don't need to complete this first step.

Create a new app:

```
rails my_new_app  
cd my_new_app
```

Heroku uses a PostgreSQL database so we're going to have to edit our **Gemfile** to reflect that. Change this line:

```
gem 'sqlite3'
```

To this:

```
gem 'pg'
```

Make sure everything works locally by running:

```
bundle install
```

If you aren't using Git already, you definitely should be. Git allows you to quickly keep track of changes and to work easily and collaboratively with others. Let's initialize our app to use git.

```
git init  
git add .  
git commit -m "initial commit"
```

Now let's deploy our app to Heroku. First, let's create a new app using the Heroku Toolbelt that we installed earlier. The argument below, "my-app-name", is the name that we are giving our app. Feel free to choose anything here, it will eventually become the url for your app.

```
$ heroku create my-app-name  
Creating my-app-name... done, stack is cedar  
http://my-app-name.herokuapp.com/ | git@heroku.com:my-app-name.git  
Git remote heroku added
```

What the command did was create a new app with a name that you specified. If we want to give our app a random name, we can run the command: **heroku create** without an optional name and heroku will create a name for us. Let's push our code to heroku and deploy app.

```
$ git push heroku master
Counting objects: 67, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (52/52), done.
Writing objects: 100% (67/67), 86.33 KiB, done.
Total 67 (delta 5), reused 0 (delta 0)

-----> Heroku receiving push
-----> Rails app detected
-----> Installing dependencies using Bundler version 1.1
    Checking for unresolved dependencies.
    Unresolved dependencies detected.
    Running: bundle install --without development:test --path vendor/bundle --deployment
    Fetching source index for http://rubygems.org/
    Installing rake (0.8.7)
    ...
    Installing rails (3.0.5)
    Your bundle is complete! It was installed into ./vendor/bundle
-----> Rails plugin injection
    Injecting rails_log_stdout
    Injecting rails3_serve_static_assets
-----> Discovering process types
    Procfile declares types -> (none)
    Default types for Rails -> console, rake, web, worker
-----> Compiled slug size is 8.3MB
-----> Launching... done, v5
    http://my-app-name.herokuapp.com deployed to Heroku

To git@heroku.com:my-app-name.git
* [new branch]      master -> master
```

Now that our app has been pushed to Heroku, we need to initialize our database. Then we can now open our app!

```
heroku run rake db:reset
heroku open
```

Now, whenever we want to push updates to our app, we run the following sequence.

```
git add .  
git commit -m "updates for this commit"  
git push heroku master
```

And our app will be updated and the changes can be seen immediately at our url.

Node.js

If you already have an existing Node app, you don't have to complete this first step.

Create a new folder and make a new file and add the following contents:

```
mkdir node_app  
cd node_app  
touch web.js
```

We're going to create a simple web server that uses Express and displays "Hello World!"

web.js

```
var express = require("express");  
var app = express();  
app.use(express.logger());  
  
app.get('/', function(request, response) {  
  response.send('Hello World!');  
});  
  
var port = process.env.PORT || 5000;  
app.listen(port, function() {  
  console.log("Listening on " + port);  
});
```

Now let's create another file **package.json** that will contain information about the dependencies of our app.

package.json

```
{
  "name": "node-example",
  "version": "0.0.1",
  "dependencies": {
    "express": "3.1.x"
  },
  "engines": {
    "node": "0.10.x",
    "npm": "1.2.x"
  }
}
```

Now use the command **npm** to install our dependencies.

```
npm install
```

Lastly, we're going to make a **Procfile** that is going to declare what processes we want to run once we deploy.

```
web: node web.js
```

You can now start your app using Foreman with the following command.

```
foreman start
```

If you aren't using Git already, you definitely should be. Git allows you to quickly keep track of changes and to work easily and collaboratively with others. Let's initialize our app to use git.

```
git init
git add .
git commit -m "initial commit"
```

Now let's deploy our app to Heroku. First, let's create a new app using the Heroku Toolbelt that we installed earlier. The argument below, "my-app-name", is the name that we are giving our app. Feel free to choose anything here, it will eventually become the url for your app.

```
$ heroku create my-app-name
Creating my-app-name... done, stack is cedar
http://my-app-name.herokuapp.com/ | git@heroku.com:my-app-name.git
Git remote heroku added
```

What the command did was create a new app with a name that you specified. If we want to give our app a random name, we can run the command: **heroku create** without an optional name and heroku will create a name for us. Let's push our code to heroku and deploy app.

```
$ git push heroku master
Counting objects: 343, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (224/224), done.
Writing objects: 100% (250/250), 238.01 KiB, done.
Total 250 (delta 63), reused 0 (delta 0)

-----> Node.js app detected
-----> Resolving engine versions
    Using Node.js version: 0.10.3
    Using npm version: 1.2.18
-----> Fetching Node.js binaries
-----> Vendoring node into slug
-----> Installing dependencies with npm
    ....
    Dependencies installed
-----> Building runtime environment
-----> Discovering process types
    Procfile declares types -> web

-----> Compiled slug size: 4.1MB
-----> Launching... done, v9
    http://my-app-name.herokuapp.com deployed to Heroku

To git@heroku.com:my-app-name.git
* [new branch]      master -> master
```

Once we have finished deploying our app, we can access it by entering in the following command.

```
heroku open
```

Django

Django instructions are a little more complex and I believe that the Heroku instructions can explain it better than I can. For now, view the instructions here (<https://devcenter.heroku.com/articles/django>). I will update this post in the future.

1 Comment **Mark Miyashita****1 Login** ▾ **Recommend** 1  **Share****Sort by Best** ▾

Join the discussion...




LOG IN WITH

OR SIGN UP WITH DISQUS **ljnissen** • 3 years ago

This is a very nice post. I especially like your attention to detail.

Thank you for sharing!

^ | ▾ • Reply • Share ▸

 **Subscribe**  **Add Disqus to your site** Add DisqusAdd  **Privacy**

© 2014 Mark Miyashita

Github (<http://github.com/negativetwelve>) · TA Ratings(<https://hkn.eecs.berkeley.edu/coursesurveys/instructor/7762>) · Facebook(<https://www.facebook.com/markmiyashita12>) · LinkedIn (<http://www.linkedin.com/pub/mark-miyashita/38/63/49/>) · Google+ (<https://plus.google.com/101180624276428786239?rel=author>)