

# Extra notes - Heroku & MongoDB

- Dr Nick Hayward

## Contents

- basic install for Heroku
- Heroku and MongoDB
- Heroku addon MongoDB - caveats
- Heroku with custom mLab MongoDB
- rename Heroku app from cli

## basic install for Heroku

Heroku is a hosted cloud service for publication of web apps &c. in a variety of languages, including Node.js.

Further details available at Heroku website,

- <https://www.heroku.com>

Sign up for a free account at the above site.

Then, we need to install Heroku's CLI toolbelt app,

- <https://toolbelt.heroku.com>

For OS X, the simplest option is to use Homebrew,

```
brew install heroku
```

and then check install version,

```
heroku --version
```

CLI help documentation can then be found using the following command,

```
heroku help
```

Reference guide for the toolbelt CLI,

- <https://devcenter.heroku.com/articles/heroku-cli>

## Heroku and MongoDB

We can also add a hosted instance of MongoDB, which we can then associate with our hosted app on Heroku.

For MongoDB, we can use a service such as mLab,

- <https://www.mlab.com/>

We need to create a remote instance of MongoDB for our app's data. On Heroku, we install an add-on for *mLab MongoDB*, which is a hosted service for MongoDB.

Further details on Heroku add-ons,

- <https://elements.heroku.com/addons>

To install this add-on we can run the following command in our project's CWD,

```
heroku create //create a new heroku app if necessary
heroku addons:create mongolab:sandbox //install add-on for mLab MongoDB with free
sandbox plan
```

mLab offers a free sandbox option, which is installed for our app.

### Heroku addon MongoDB - caveats

If we add a DB using the addon for mLab, Heroku will create the DB with mLab, and then configure the instance of the DB with the current app on Heroku.

i.e. it creates a special account with mLab just for the addon for the current project. This will also set the config environment variables for the hosted app so it can connect to the DB on mLab.

### Heroku with custom mLab MongoDB

Heroku can also be configured to use an existing DB on mLab. This allows us to create the DB with mLab, set a custom username and password, and manage each DB through a single account, for example.

On mLab, we can login and create a DB. This will then give us a URL for the DB on mLab, e.g.

- `mongodb://:@ds328353.mlab.com:27773/`

Then, we need to ensure that this DB has at least one user associated on mLab under the *Users* tab.

Once we've configured mLab, we can then pass these config settings to Heroku, e.g.

```
heroku config:set MONGODB_URI=mongodb://<dbuser>:
<dbpassword>@ds328353.mlab.com:27773/<dbname>
```

With this command, we are storing the DB credentials in a config variable on Heroku. This variable is then accessed in Node.js through the `process` object as a property of the environment, e.g.

```
process.env.MONGODB_URI
```

### rename Heroku app from cli

We can rename and update a Heroku app using the CLI toolbelt, again in the root directory of the enclosing git repository

```
heroku apps:rename new_name --app current_name
```

This will update the name locally, and on the Heroku servers. It will also update the git repository name for `heroku push` commands.