

Deploying your app on Heroku

Deploying your Node Express app with Heroku

Heroku is an easy and free way to quickly deploy your web app. You can use any git repo with your app's source code and use it to push your project to Heroku, which will give you a URL to connect to your web app. This page will tell you how to deploy your Node/Express app to Heroku, since that is what we are using in this course. Heroku can work with other web servers and frameworks.

Make a Heroku account

Go to <https://signup.heroku.com/> (<https://signup.heroku.com/>) to get a free account. You will need this account/password to login to the Heroku command line interface.

Install the Heroku command line interface

Go to <https://devcenter.heroku.com/articles/heroku-cli#download-and-install> (<https://devcenter.heroku.com/articles/heroku-cli#download-and-install>) and install the Heroku Command Line Interface (CLI). You should be able to access heroku through your terminal by running the command

```
$ heroku
```

Create a Heroku app and push your code

1. Navigate to your app's top level directory on your local machine (or where your Express server - likely *server.js* - file is).
2. Make sure your Express server is listening on the `PORT` environment variable when it's not on localhost. As in our lecture notes, the port listening code of your app should look something like this:

```
const port = process.env.PORT || 5000
app.listen(port, () => {
  log(`Listening on port ${port}...`)
})
```

3. Check that your package.json file tells Heroku to start your Express server in the "scripts" object :

```
"scripts": {
  "start": "node server.js"
},
```

4. If your app is already version controlled by git (for example, if it is already on GitHub), skip this step. Otherwise, initialize a git repo and commit your code:

```
$ git init
$ git add .
```

```
$ git commit -m "initial app commit"
```

5. Create a heroku app using the CLI (you may be asked to sign in through the browser):

```
$ heroku create  
<heroku will generate an app and give you the URL>
```

You can check that the heroku remotes were created:

```
$ git remote -v  
<it should show heroku as a remote>
```

Now, push the latest commit to the heroku remote's master/main branch:

```
$ git push heroku  
<heroku will build the app and show you the status>  
# can also try `git push heroku master` or `git push heroku main` if there are branch issues
```

6. Once the push is successful, use the Heroku URL for your app to access your deployed web server.
7. Whenever you make further changes to your app, commit those changes and push them to the Heroku remote when you are ready to deploy

The instructions above reflect how we deployed our apps in lecture. For more detail about this process, visit the official page about [Heroku and git](https://devcenter.heroku.com/articles/git) [_ \(https://devcenter.heroku.com/articles/git\)](https://devcenter.heroku.com/articles/git).