CSCI 4140 – Tutorial 9

Deploying Node.js Applications on Heroku

Matt YIU, Man Tung (mtyiu@cse)
SHB 118

Office Hour: Wednesday, 3-5 pm

2016.03.29

Prerequisite

- We will start with an **Express application**
  - Please follow the instructions on the tutorial slides "**Installing Node.js and Express on [Windows|Linux or Mac]**", pp. 19-21 for creating an application skeleton
  - If you don't use an Express application, you need to figure out where to configure the server's listen IP address and port number
- We will deploy the application on **Heroku**
- **Install Heroku toolbelt: https://toolbelt.heroku.com/**

Prerequisite

- Install Heroku toolbelt: https://toolbelt.heroku.com/
  - Once installed, you'll have access to the heroku command from your command shell. Log in using the email address and password you used when creating your Heroku account:

```
$ heroku login
Enter your Heroku credentials.
Email: adam@example.com
Password (typing will be hidden):
Authentication successful.
```

Step 1. Create new app

- Now go to the directory of your app and type:
  - **Note:** `socket-io-chatroom` is the name of your app and is optional
  - Remember to change the app name!

```
$ heroku create socket-io-chatroom
Creating socket-io-chatroom... done, stack is cedar-14
https://socket-io-chatroom.herokuapp.com/ |
https://git.heroku.com/socket-io-chatroom.git
```

Step 2. Include a package.json file

- All Node.js applications should include a **package.json** file in the **root of their project**
  - Since we are using **Express application generator** to create our application skeleton, this file is automatically generated
- Heroku runs "npm start" when you deploy your app so make sure that the startup script in **scripts.start** is correct

```
{
  "name": ”socket-io-chat",
  "version": "0.0.0",
  "private": true,
  "scripts": {
    "start": "node ./bin/www"
  },
 "dependencies": {
    "body-parser": "~1.12.0",
    "cookie-parser": "~1.3.4",
    "debug": "~2.1.1",
    "express": "~4.12.2",
    "jade": "~1.9.2",
    "morgan": "~1.5.1",
    "serve-favicon": "~2.2.0",
    "socket.io": "^1.4.5"
  }
}
```

Sample package.json file

Step 3. Edit the startup script (`bin/www`)

- Your app should create an HTTP server at the port specified by Heroku
  - The port is available as an environment variable (`process.env.PORT`)
  - If you use the default startup script generated by Express application generator, the port number is already set up for you
  - If you use your own startup script, please check if you change the port number as `process.env.PORT`

Step 4. `git commit` and push to Heroku

- Now back to the root directory of your application
- In case you did not create the Git repository…

```
$ git init
Initialized empty Git repository in
/Users/mtyiu/Development/nodejs-Heroku/.git/
$ git add .
```

- Commit your code changes:

```
$ git commit -a -m "<Your commit message>"
```

Step 5. `git commit` and push to Heroku

- We are ready to push the code to Heroku:

```
$ git push heroku master
```

- When it is done, you can visit your website using the URL:

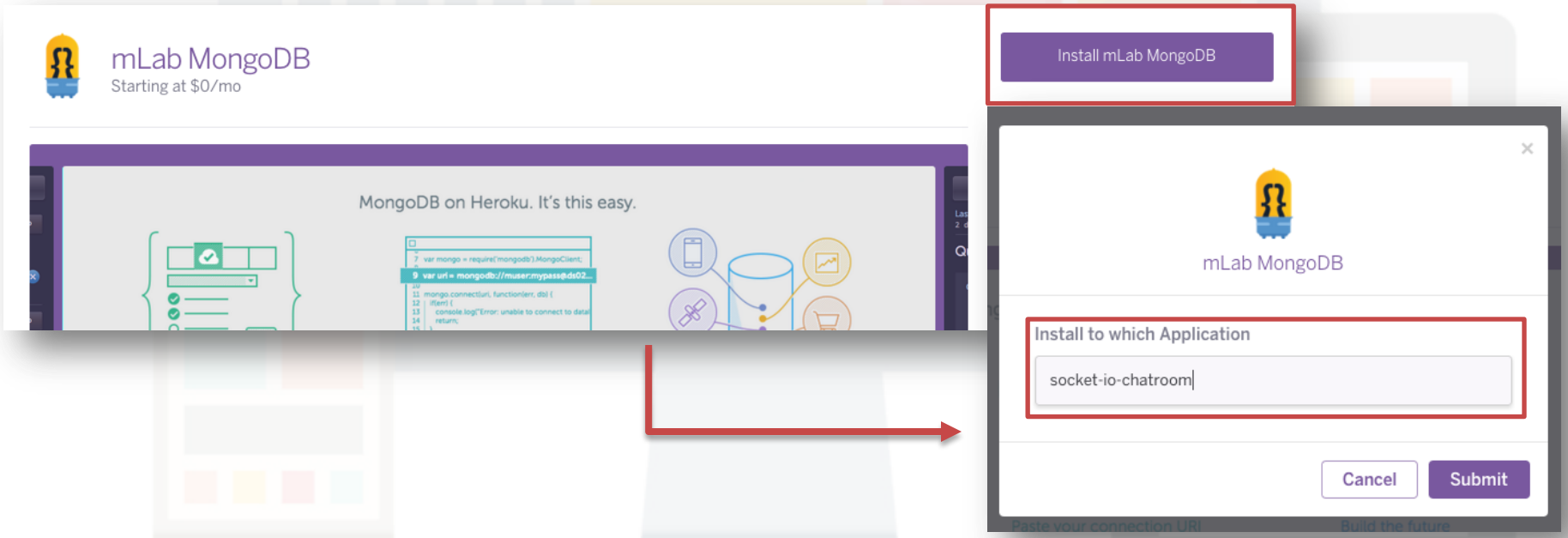
  `https://[APP_NAME].herokuapp.com/`

- Socket.IO chat room is deployed at https://socket-io-chatroom.herokuapp.com/

- Done!

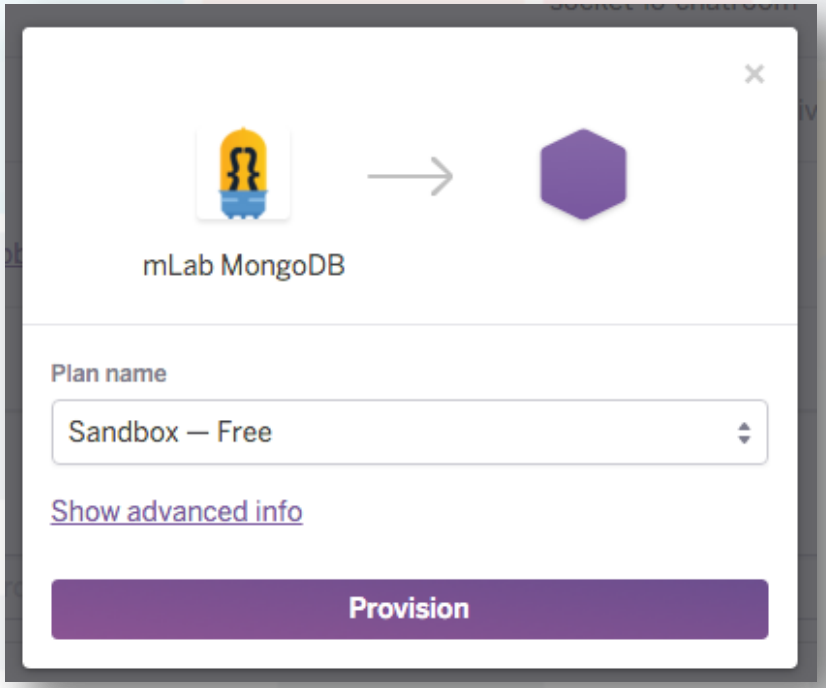
Add-ons on Heroku

- Heroku provides many add-ons that can be integrated with your app easily: https://elements.heroku.com/addons
  - E.g., mLab MongoDB for a MongoDB, ClearDB MySQL
- To install an add-on, go to the above page and select it



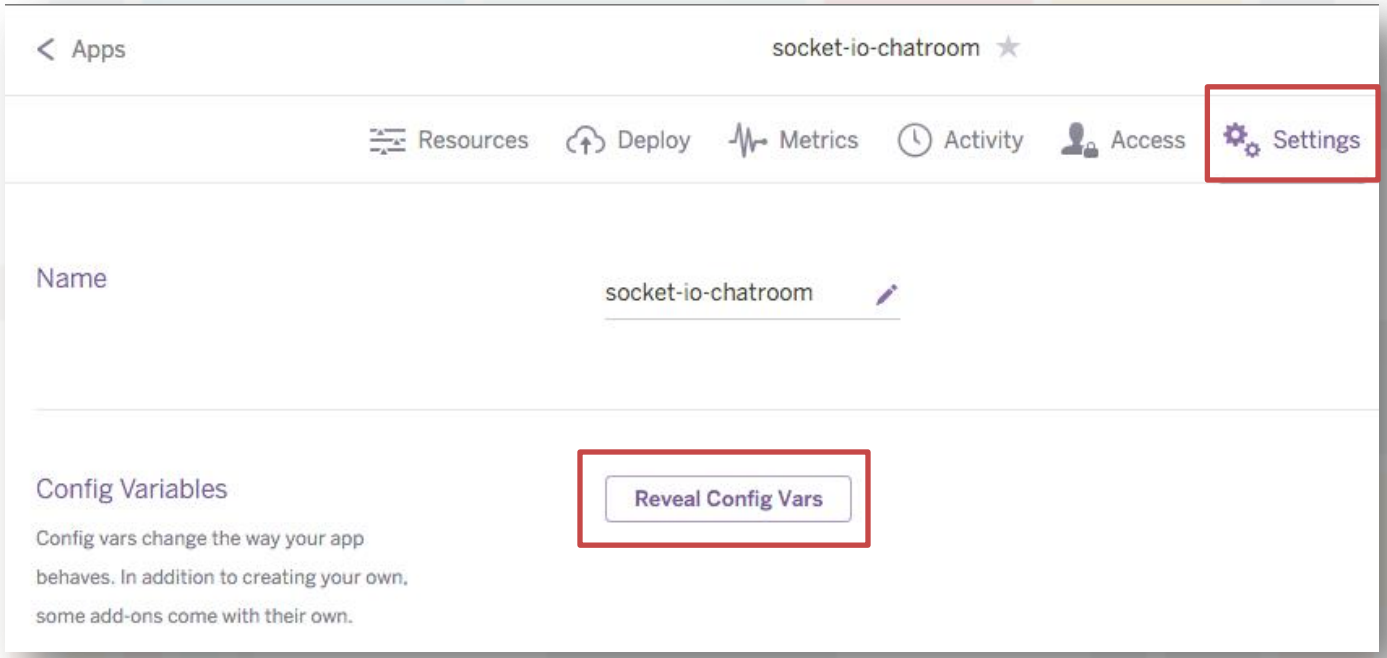
Add-ons on Heroku

- Select the free plan and click “Provision”



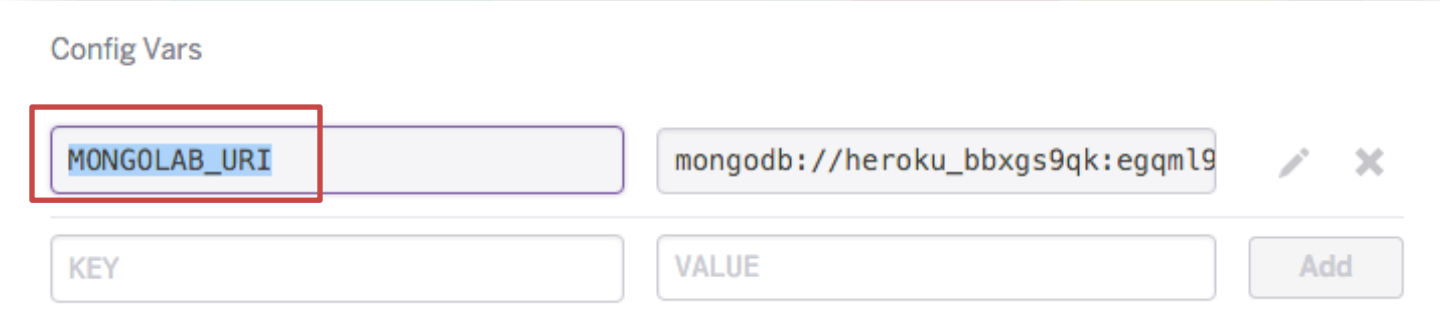
Add-ons on Heroku

- To find the configuration variables for using the add-ons, go to Heroku Dashboard and select your app → **Settings** → **Reveal Config Vars**



Add-ons on Heroku

- To find the configuration variables for using the add-ons, go to Heroku Dashboard and select your app → **Settings** → **Reveal Config Vars**



- To access the value of the variables (e.g., `MONGOLAB_URI`), use the `process.env` object (e.g., **`proecss.env.MONGOLAB_URI`**)

Add-ons on Heroku

- To help you run your app locally, the Heroku toolbelt provides a "`heroku local`" command

```
$ heroku local
```

- To set up the environment variables as you run "`heroku local`", include an "**`.env`**" file (Note: Do not include it in the git repository)
  - E.g.,

```
# Copy Heroku config var to your local .env
$ heroku config:get MONGOLAB_URI -s >> .env
```

- Open http://localhost:5000 with your web browser – now your app can access all Heroku add-ons!

More about Heroku

- To access the logs for your app:

```
$ heroku logs
```

- To login to the shell of your app:

```
$ heroku run bash
```

More about Heroku

- To further configure your deployment, Heroku defines a mechanism called **Procfile** to control your applicataion's **dynos**
  - A dyno is a lightweight Linux container that runs a single user-specified command
  - More information is available at https://devcenter.heroku.com/articles/procfile
- Free dynos have some limitations:
  - Each free dyno sleeps after 30 minutes of inactivity
  - **Each free dyno must sleep 6 hours in a 24 hour period**
  - Due to this limitation, you should develop your app locally, and deploy it when you finish a milestone
  - More details: https://www.heroku.com/pricing

References

- Getting Started on Heroku with Node.js:
  - https://devcenter.heroku.com/articles/getting-started-with-nodejs
- Heroku Dev Center:
  - https://devcenter.heroku.com/categories/reference

– End –