109-2 Web Programming

Deployment Tutorial

Presenter 陳柏志

2021/6/28

Table of Contents

Deploy on Personal PC / Work Station

- Execute your code directly on your computer
- Need fixed IP you may need to set up for this, e.g. port forwarding

  140.112.218.2
- The computer must be always on
- Your Deployed service 網址 http://140.112.218.2:3000
- Cool Dev boards that can be used as servers:

Raspberry pi

ESP32

Deploy on GCP

- Professional
- $300 USD creditable amount
- GPU available
- Servers in Taiwan
- Google APIs

Deploy on GCP

- Go to https://cloud.google.com/

- Add credit card information
- 新增專案

- Go to Compute Engine ➔ VM 執行個體

Deploy on GCP

- Setup VM

Deploy on GCP

- Install npm, git and tmux

```
sudo apt-get install npm
```

```
sudo apt-get install git
```

```
sudo apt-get install tmux
```

```
git clone …
```

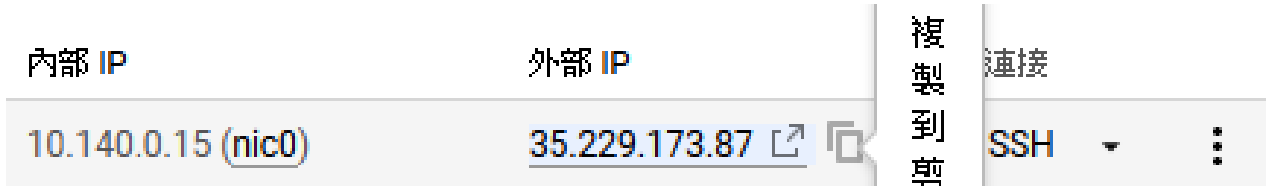
Deploy on GCP

- Use tmux to run code

```
tmux new -s “webserver”          // create a named tmux session
sudo npm start
Ctrl + b  // hot ket  ➔  d       // leave the tmux session
tmux ls                          // list tmux sessions
tmux a -t “webserver”            // enter “webserver” session
tmux kill-session -t “webserver” // kill the “webserver” session
Ctrl + b  // hot ket  ➔  %       // split window
```

Deploy on GCP

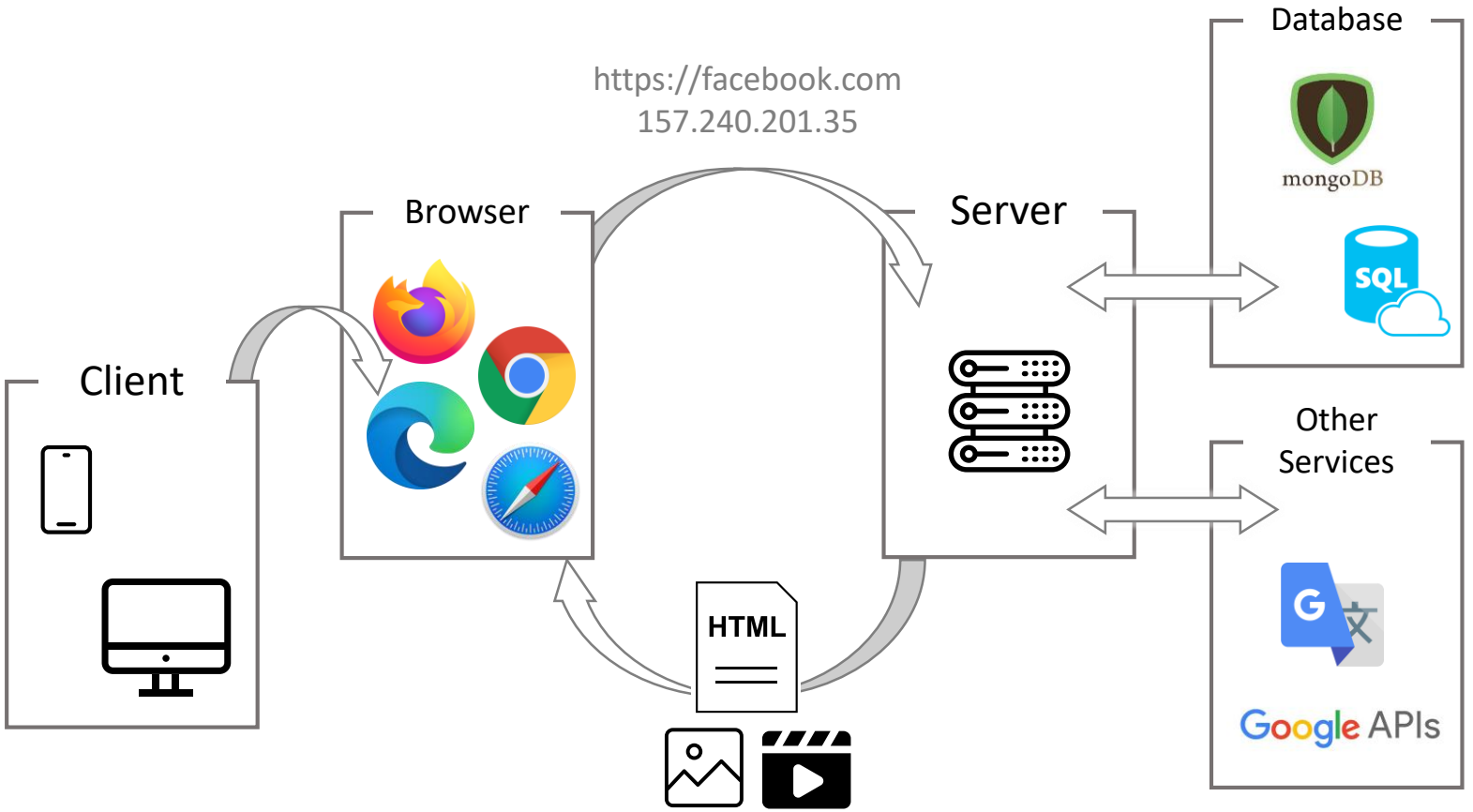
- Check your IP



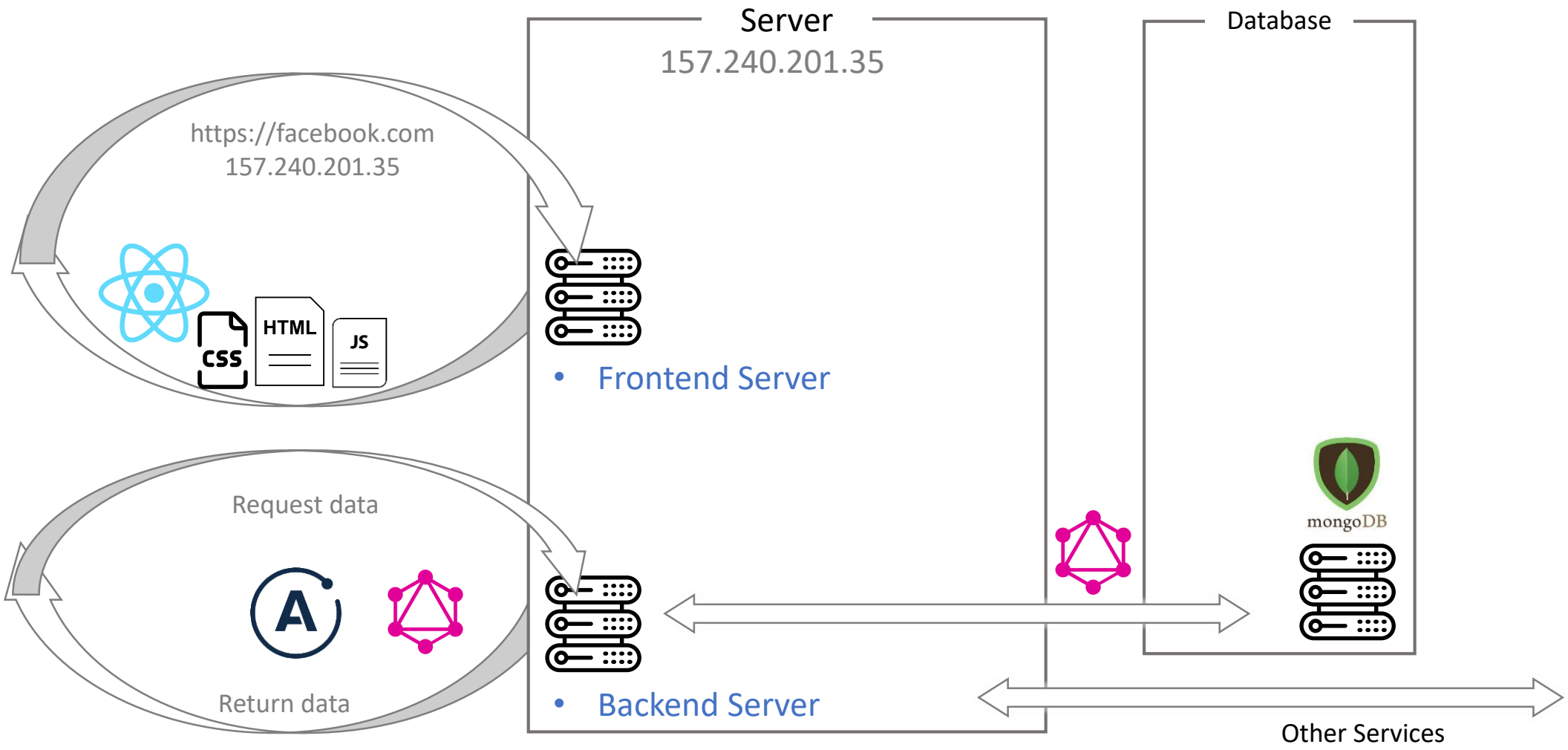
- Your Deployed service 網址

http://35.229.173.87:3000

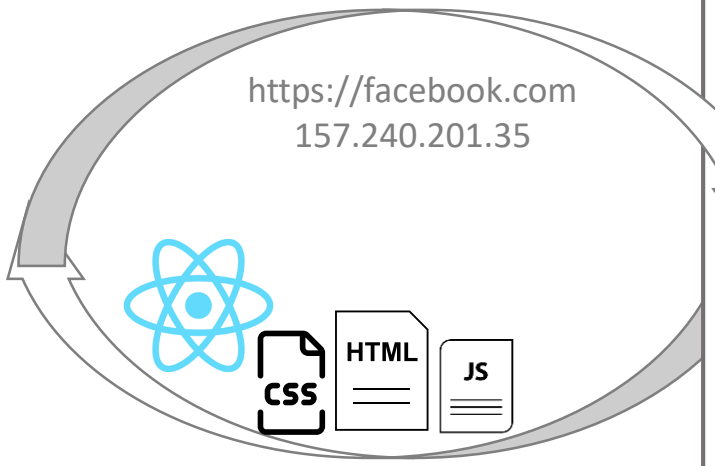
Frontend , Backend & Database



Frontend , Backend & Database



Frontend , Backend & Database



Server

- ipconfig / ifconfig

```
IPv4 位址 . . . . . . . . . . . . : 192.168.0.116
```

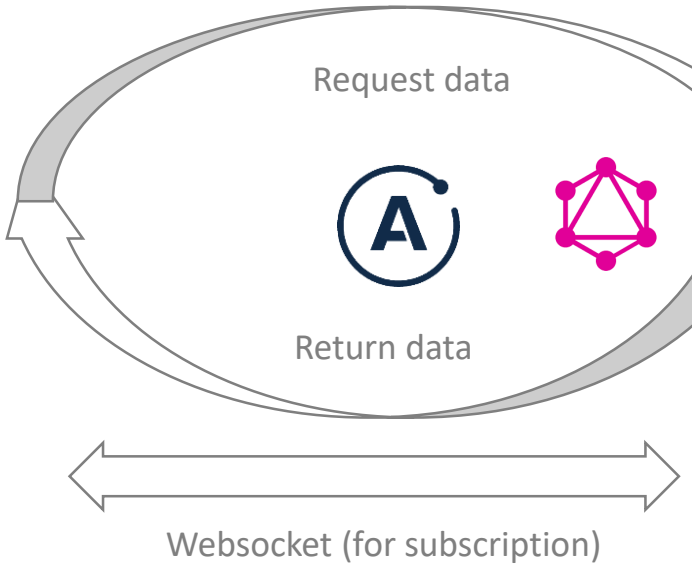
- Frontend Server
  - npm start / yarn start
  - In package.json:

```
"scripts": {
  "start": "react-scripts start",
  "predeploy": "npm run build",
  "deploy": "gh-pages -d build",
```

  - localhost: 3000

- 22: ssh
- 80: http
- 443: https

Frontend , Backend & Database



Server

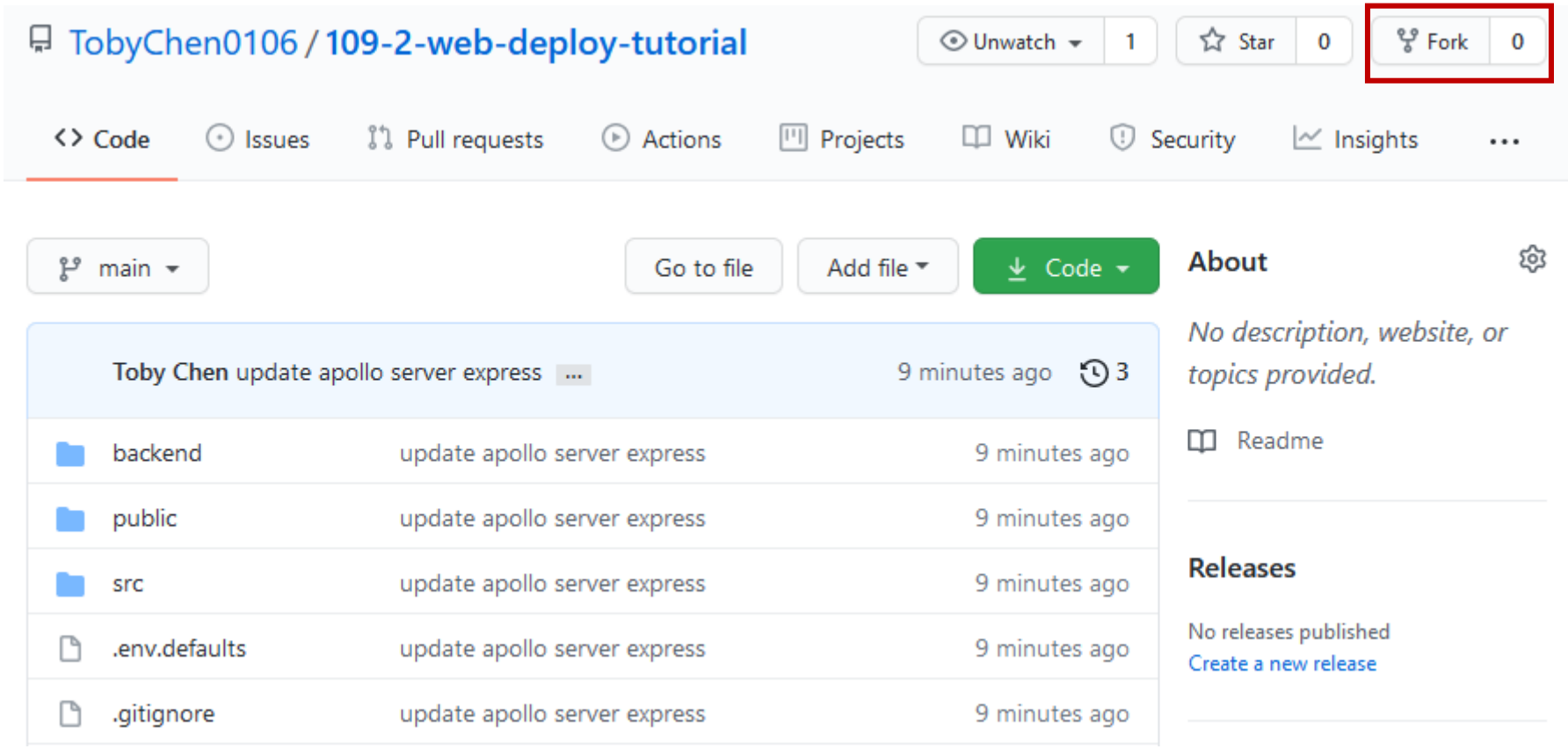
- Backend Server
  - npm run server / yarn server
  - In package.json:

```
"scripts": {
  "server": "nodemon src/index.js --ext j
  "test": "npx cypress run --browser fire
},
```

  - localhost: 5000

Build React App and Express Server

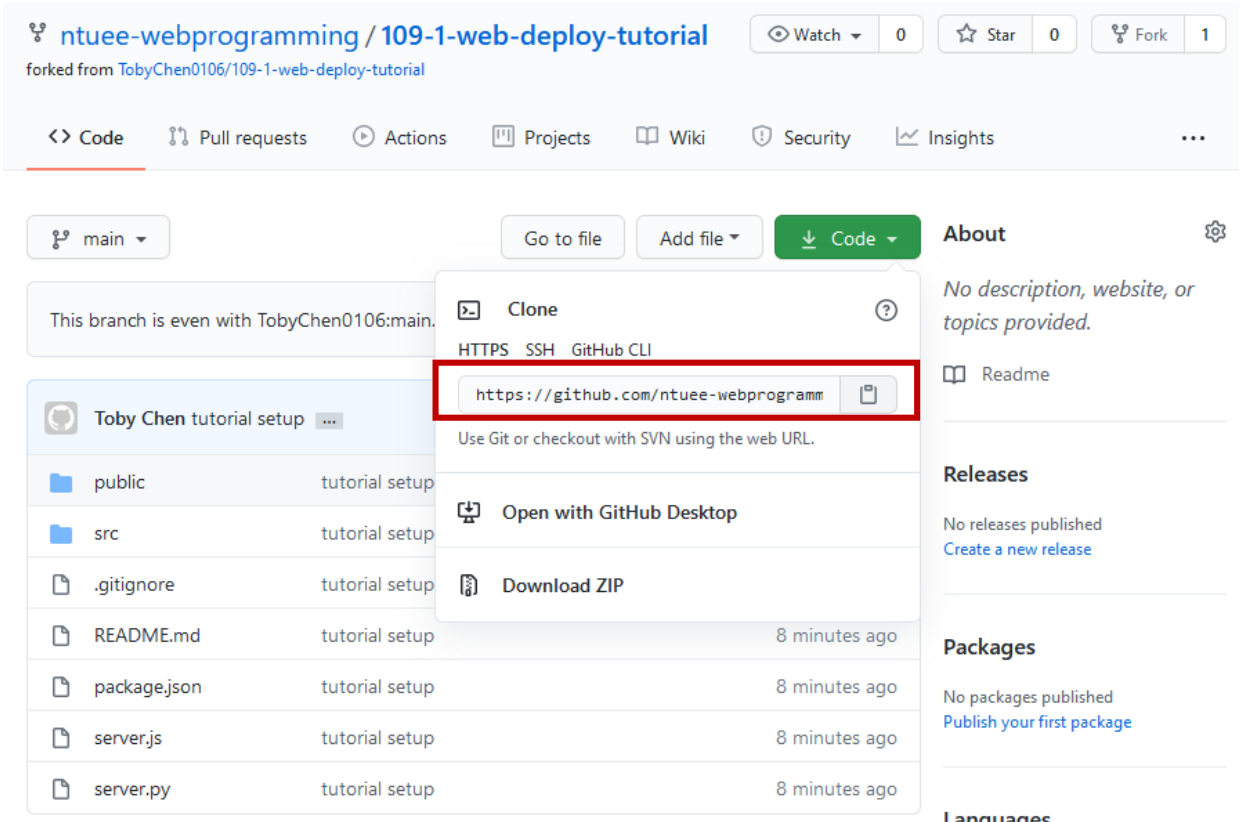
- Fork or clone the example code here: http://toby.best/deploy-tutorial



Build React App and Express Server

- Clone the code from your repository

```
git clone https://github.com/...
```



Build React App and Express Server

- In `package.json`

```
"scripts": {
  "start": "node server.js",
  "build": "react-scripts build",
  "start_react": "react-scripts start"
},
```

- Fill .env file

```
2  MONGO_URL=
3  PORT=
```

- Run `npm install` / `yarn install`
- Run `npm run build` / `yarn build`
- Run `npm start` / `yarn start`

*Using node v14.15.4

Build React App and Express Server

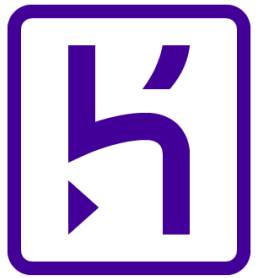

```
19  const __dirname = dirname(fileURLToPath(import.meta.url));
20  const port = process.env.PORT || 80;
21
22  const typeDefs = importSchema("./backend/schema.graphql");
23  const pubsub = new PubSub();
24  const app = express();
25
26  app.use(cors());
27  app.use("/api", apiRoute);
28  app.use(bodyParser.json());
29  app.use(express.static(path.join(__dirname, "build")));
30  app.get("/*", function (req, res) {
31    res.sendFile(path.join(__dirname, "build", "index.html"));
32  });
33
34  const server = new ApolloServer({ ···
45  });
46
47  server.applyMiddleware({ app });
48  const httpServer = http.createServer(app);
49  server.installSubscriptionHandlers(httpServer);
50
51  mongo.connect();
52
53  httpServer.listen(port, () => {
54    console.log(
55      `🚀 Server Ready at ${port}! 🚀`
56    );
57    console.log(
58      `Graphql Port at ${port}${server.subscriptionsPath}`
59    );
60  });
```

- In server.js
- Use express to serve:
  - build/index.html
  - RESTful APIs
  - Apollo server
- Connect to Mongodb
- Serve at given port

*Using node v14.15.4

Deploy on Heroku

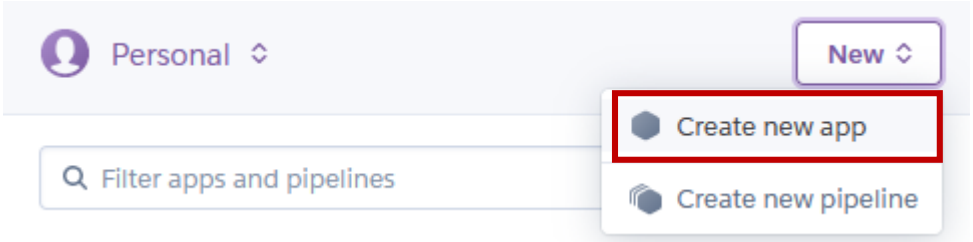
- Includes backend server
- Database api
- 5 free apps
- 500 free dyno hours per month
- Your app falls asleep in 30 minutes if no one ping it
- Awaking the app takes a while (~30s)



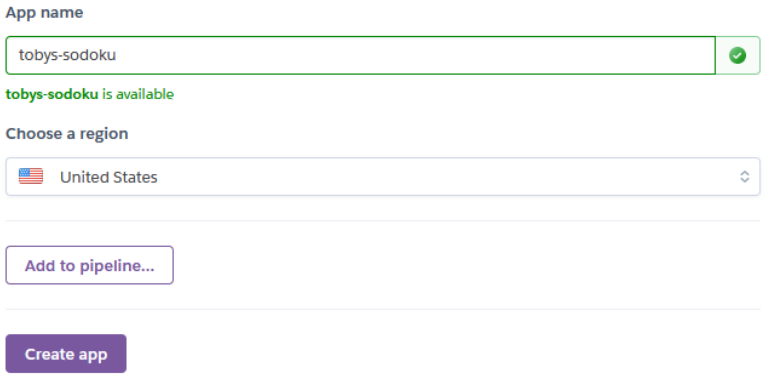
https://www.heroku.com/

Deploy on Heroku

- Register and login
- Click "New" ➔ "Create new app"

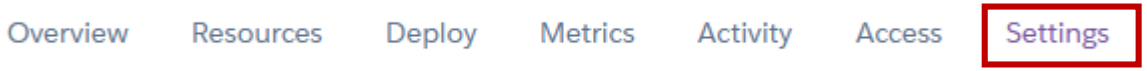


- Create App

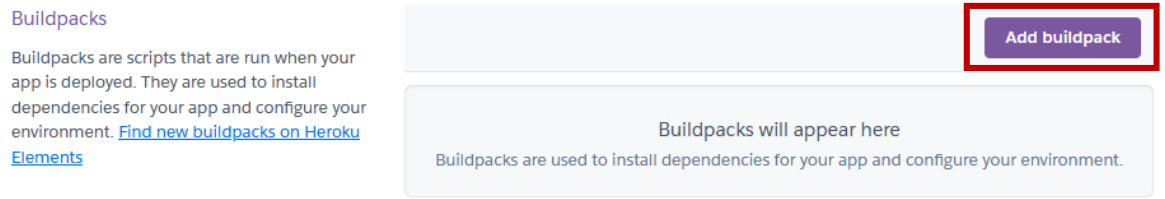


Deploy on Heroku

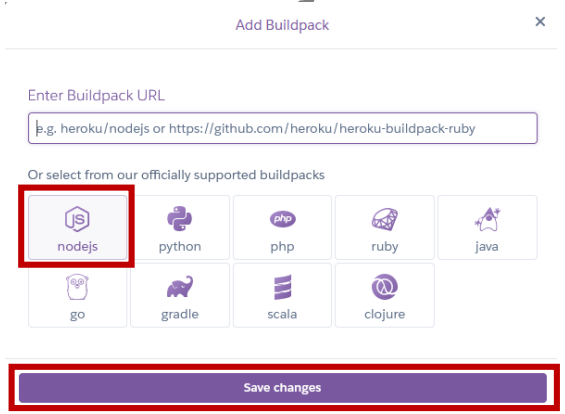
- Go to Settings section



- Select “Add buildpack”

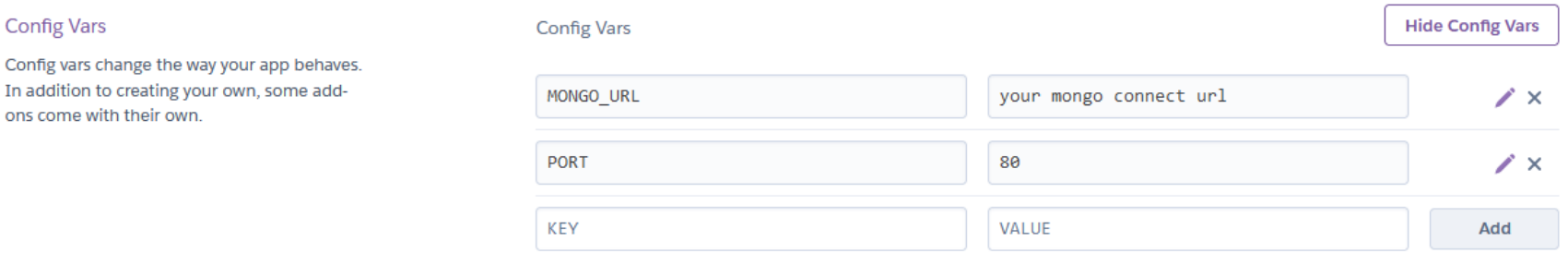


- Choose “node.js”



Deploy on Heroku

- Setup Variable



Deploy on Heroku

- Create a new branch for Heroku deploy (optional)

```
git checkout -b heroku
```

- Check current branch

```
git branch
```

```
* heroku
  main
```

- Add and commit your code

```
git add .
```

```
git commit –m “deploy on heroku”
```

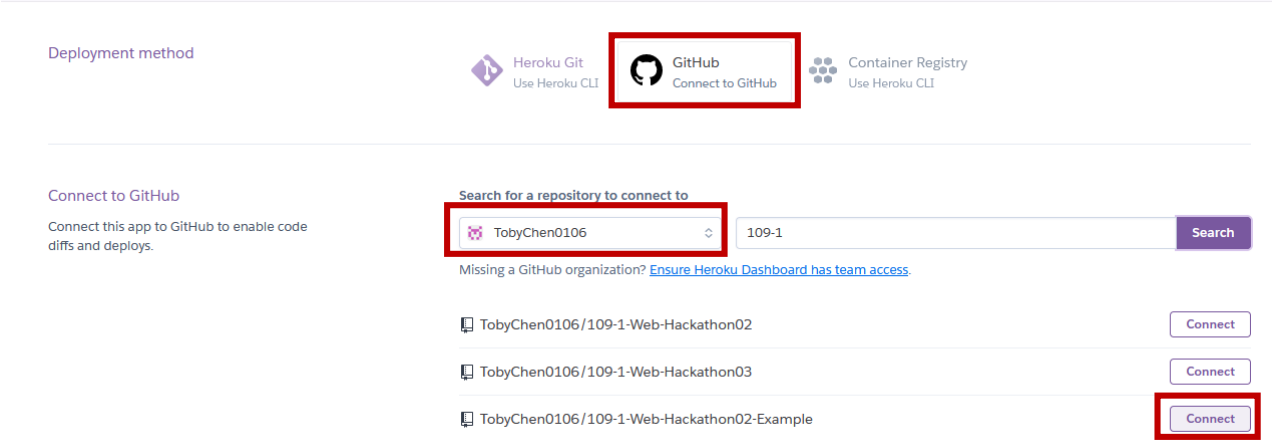
- Push your code

```
git push origin heroku
```

Deploy on Heroku

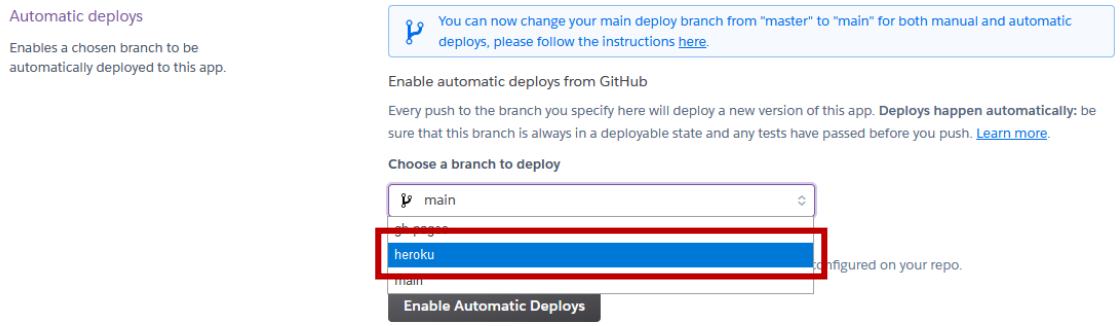
- Go to Deploy section

- Select Github, connect to Github and connect to your repo

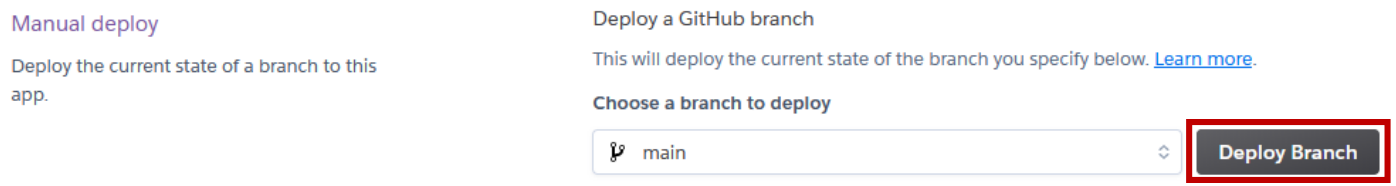


Deploy on Heroku

- Select branch



- Deploy

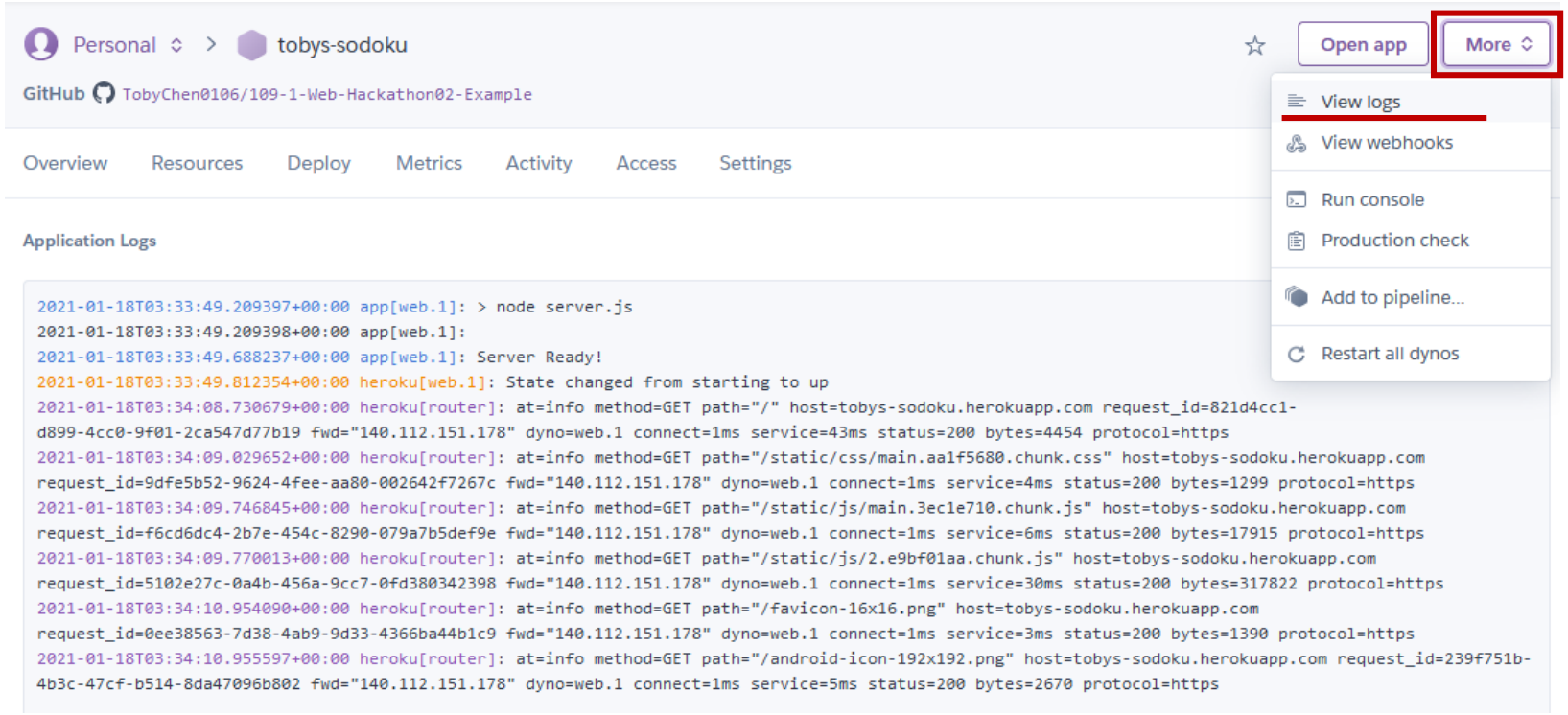


- What the Heroku node.js buildpack does:

git clone ⇒ npm install ⇒ npm run build ⇒ npm start

Deploy on Heroku

- Check logs



Deploy on Heroku

- Keep your app awake
  - Import the code:

```
import wakeUpDyno from "./backend/route/wakeUpDyno.js";
```

  - Call it in server.js

```
httpServer.listen(port, () => {
  const DYNO_URL = "https://my_app_A.heroku.com/";
  wakeUpDyno(DYNO_URL);
  console.log(`🚀 Server Ready at ${port}! 🚀`);
  console.log(`Graphql Port at ${port}${server.subscriptionsPath}`);
});
```

Other COOL Platforms

- GitHub Pages
  - Frontend only
- AWS
- Azure
- Firebase

Some other COOL stuff

- Buy your own domain:
  - COOL

    140.112.218.2 ⟹ haha.toby.best

  - Create your own mail server

    hi@toby.best

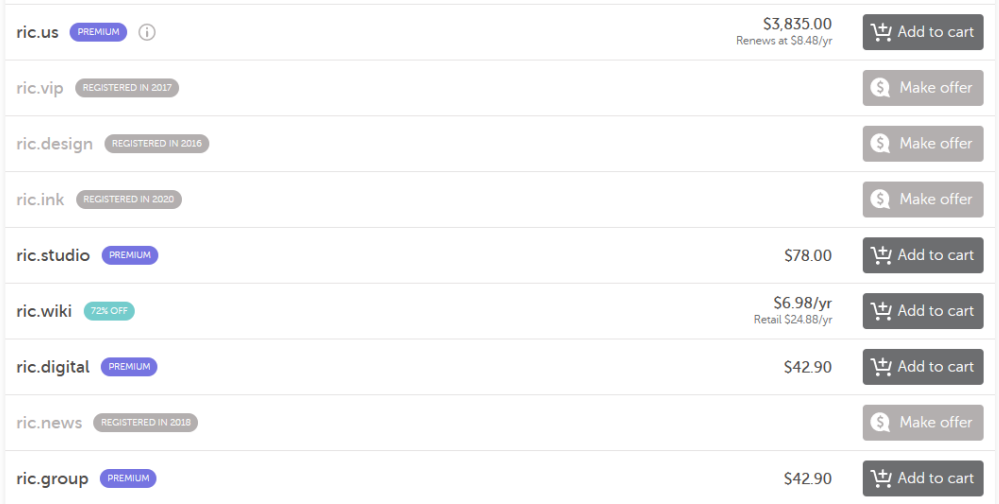
  - Do something COOL

https://ap.www.namecheap.com/

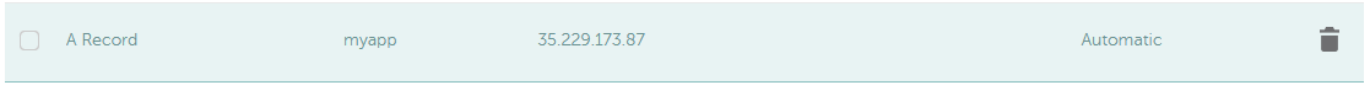
https://tld-list.com/

Some other COOL stuff

- Buy your own domain:



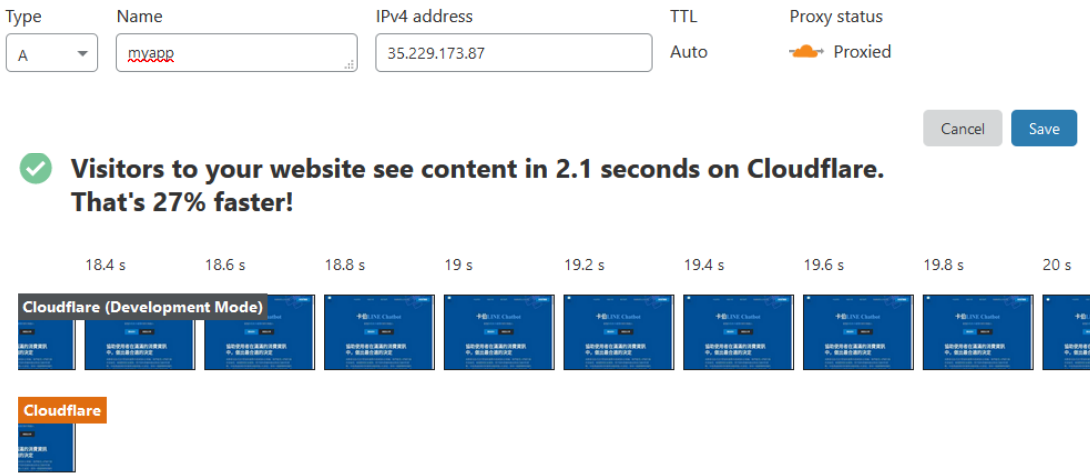
- Setup A Record



Some other COOL stuff

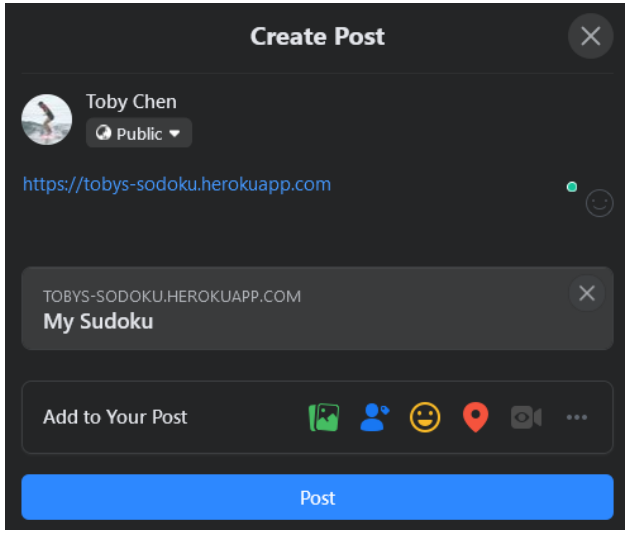
- Using https

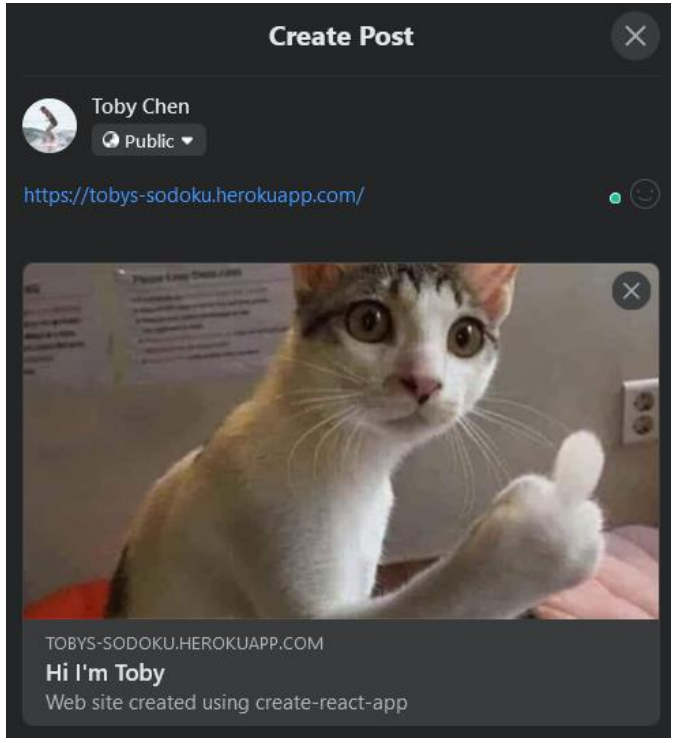
  - Require an SSL certificate and it is signed by a CA
  - Certbot: https://certbot.eff.org/
  - Use Cloudefare: https://www.cloudflare.com
    - Simple access to https
    - Firewall
    - Speedup your website



Some other COOL stuff

- Favicon https://www.favicon-generator.org/
  - Update favicon files and `manifest.json`
- OG property https://ogp.me/





Some other COOL stuff

- OG property https://ogp.me/

63 lines (55 sloc) | 2.99 KB — Raw | Blame

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8" />
  <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <meta name="theme-color" content="#000000" />
  <meta name="description" content="Web site created using create-react-app" />

  <meta property="og:title" content="Hi I'm Toby" />
  <meta property="og:image" content="https://s.keepmeme.com/files/en_posts/20200825/389052dac5820aeb2660ed9fb1bc1c22cat-showing-middle-finger-meme.jpg" />

  <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
  <!--
      manifest.json provides metadata used when your web app is installed on a
```

- OG image design principle:
  https://www.kapwing.com/resources/what-is-an-og-image-make-and-format-og-images-for-your-blog-or-webpage/

Friendly Reminders

- Deadline at 9pm, Thursday, 07/01
- Try to Deploy ASAP
- Write your README properly

Dear all:

如先前公告之期末專題說明 (https://bit.ly/3xJuP0e), 請大家注意底下事項：

1. Deadline 為 9pm, Thursday, 07/01.

在那之前，需要繳交多項資料，列舉如下：

2. 將 code push 到 Ceiba, 並按照上項規定將報告寫在 README.md

3. 完成服務部署 (deployment)

4. 拍攝 6 mins 之 demo/說明影片，上傳至雲端服務/空間

5. 至 FB 社團發表專題介紹

6. 至 https://bit.ly/3fyvADe Google From，填寫上述 2-5 項之連結，以利助教收集大家的期末專題資訊。

7. 此外，請記得以組為單位，將組員互評填寫在 "wp1092/final/eval.txt" 這個檔案

以上，請參考專題說明 (https://bit.ly/3xJuP0e)之說明，如有問題，請來信： eewebprogramming@googlegroups.com

祝大家期末專題順利！

- Ric

Dear all:

有鑒於上學期有不少同學的「安裝」與「測試」說明沒有寫清楚，導致被助教判定為 compile error 或是有些功能沒有被看到，進而造成分數的損失，在此特別提醒大家，務必在 README.md 寫清楚：

1. 如何在 localhost 安裝之詳細步驟

我們會按照你提供的步驟在我們的 local machine 進行安裝。
如果因為說明不全而導致我們安裝或是執行失敗，我們不會主動腦補去看你們的 code 來搞清楚怎麼安裝/執行，in such case, 你很可能會被判定為「無法 compile/執行」，而遭受底下其中之一的 penalty:
* 缺如 .env 等設定檔案 (2% ~ 5%)
* 無法 compile, 但後來有改好 (2% ~ 30%)
* 無法 compile, 但還是改不好 (個案處理)
* 可以 compile 但不能在 local repo work，只能從 deployed 版本評分 (10% up)
* Readme 沒照規定寫安裝方式/服務內容/心得/deploy連結/demo連結 (2% ~ 10%)

2. 如何在 localhost 執行與建議之測試方式

我們會按照你提供的功能說明進行測試。
所以如果你的功能說明不全、或是有些功能未完成但卻沒有指出來、或者是有些需要特別指引才能執行的功能，請務必說明清楚，或者是提供對應的測資。
如果因為說明不完整而導致我們沒有測到，影響到分數，請自行負責。

大家加油！

- Ric

Thanks for Your Attention

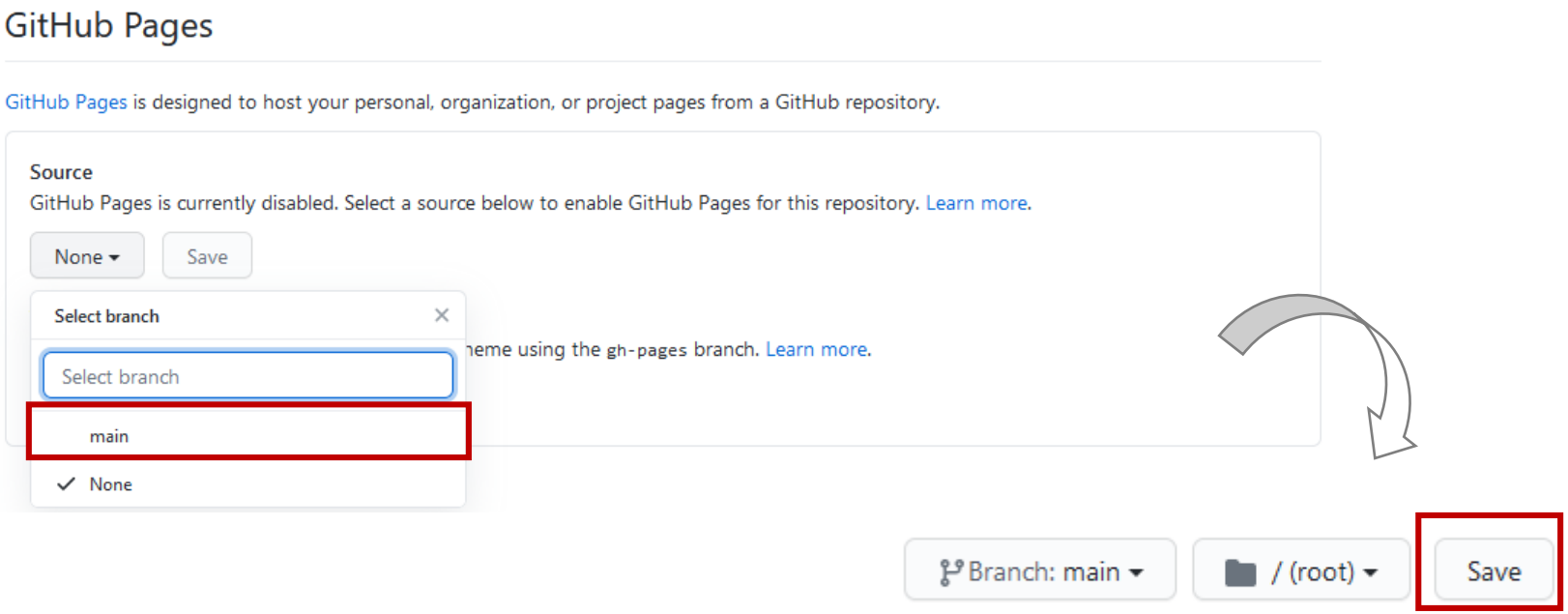
Deployment Tutorial

Presenter 陳柏志

2021/6/28

Appendix - Deploy on GitHub Pages

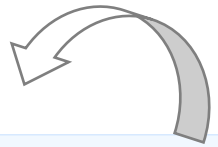
- Run `npm install gh-pages --save-dev` / `yarn add gh-pages --save-dev`
- Go to your GitHub Repo
  - settings ➔ options ➔ GitHub Pages

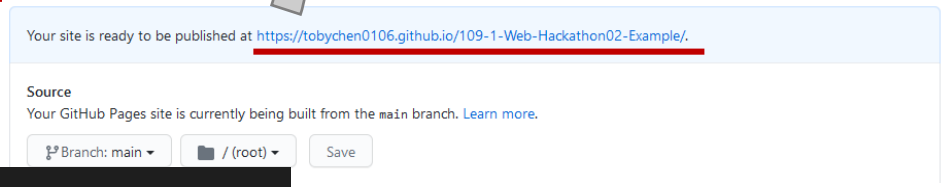


Appendix - Deploy on GitHub Pages

- Add homepage attribute in package.json

```
{
  "name": "my-sudoku",
  "version": "0.1.0",
  "private": true,
  "homepage": "https://tobychen0106.github.io/109-1-Web-Hackathon02-Example/",
```





- Add attributes in scripts in package.json

```
"scripts": {
  "start": "react-scripts start",
  "predeploy": "npm run build",
  "deploy": "gh-pages -d build",
  "start_server": "node server.js",
  "build": "react-scripts build"
},
```

“predeploy” : “npm run build”
“deploy” : ”gh-pages –d build”

```
"scripts": {
  "start": "react-scripts start",
  "predeploy": "yarn build",
  "deploy": "gh-pages -d build",
  "start_server": "node server.js",
  "build": "react-scripts build"
},
```

“predeploy” : “yarn build”
“deploy” : ”gh-pages –d build”

Appendix - Deploy on GitHub Pages

- Run `git add .`
- Run `git commit -m “for github-pages deployment”`
- Run `npm run deploy` / `yarn run deploy`

Appendix - Deploy on GitHub Pages

- Go to the Github Pages options: