Deploying React Front-end to Netlify

Heroku free goes into sleep if there is no traffic in 30 minutes. If you access the web server again after 30 minutes, it takes 5 to 10 seconds to wake up.

Deploy React front-end to Netlify and Node.js back-end to Heroku. Netlify is a web hosting service. We can connect our GitHub repository with Netlify and update the website on pushing source to GitHub. We can use Netlify free tire and it does not go into sleep. The React front-end page is always displayed quickly!

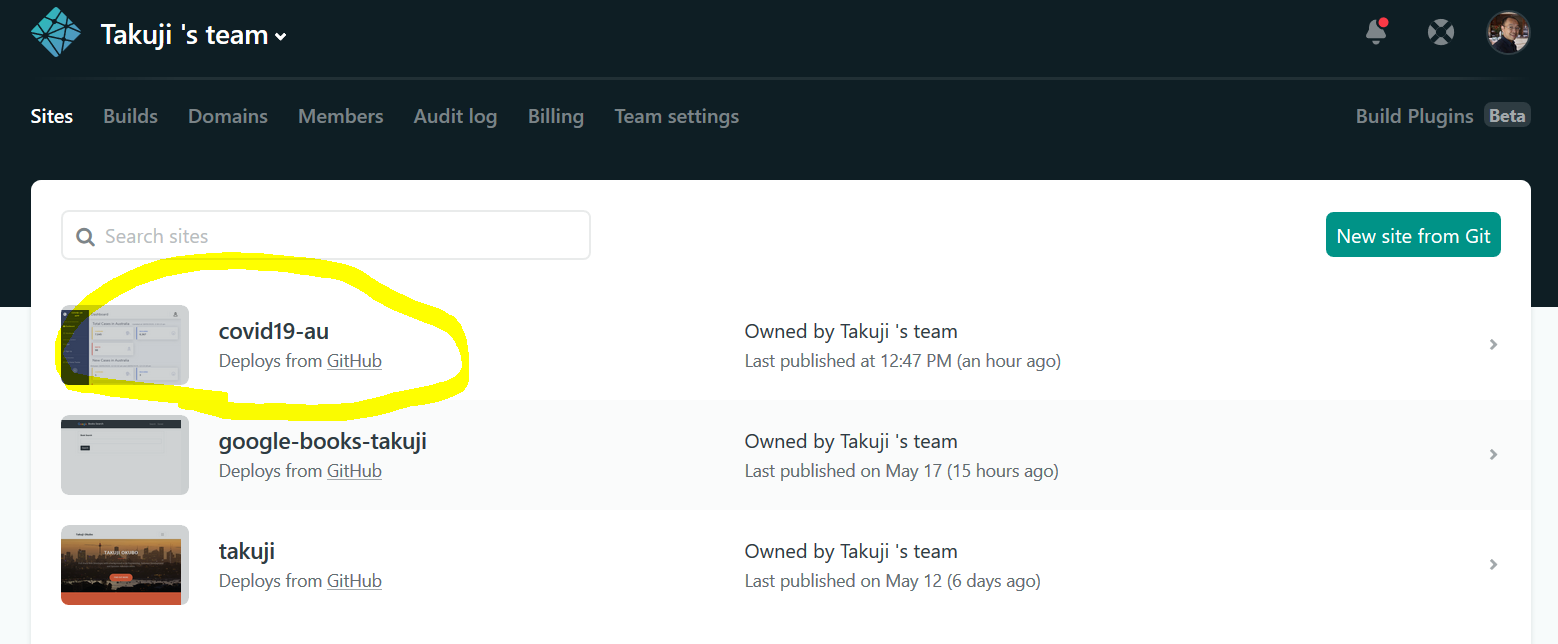
# Steps to Deploy React Front-end to Netlify

1. Sign up Netlify.
2. Sign in Netlify.
3. Click “New site from Git” button. Link a GitHub repositor with Netlify.

See this article.

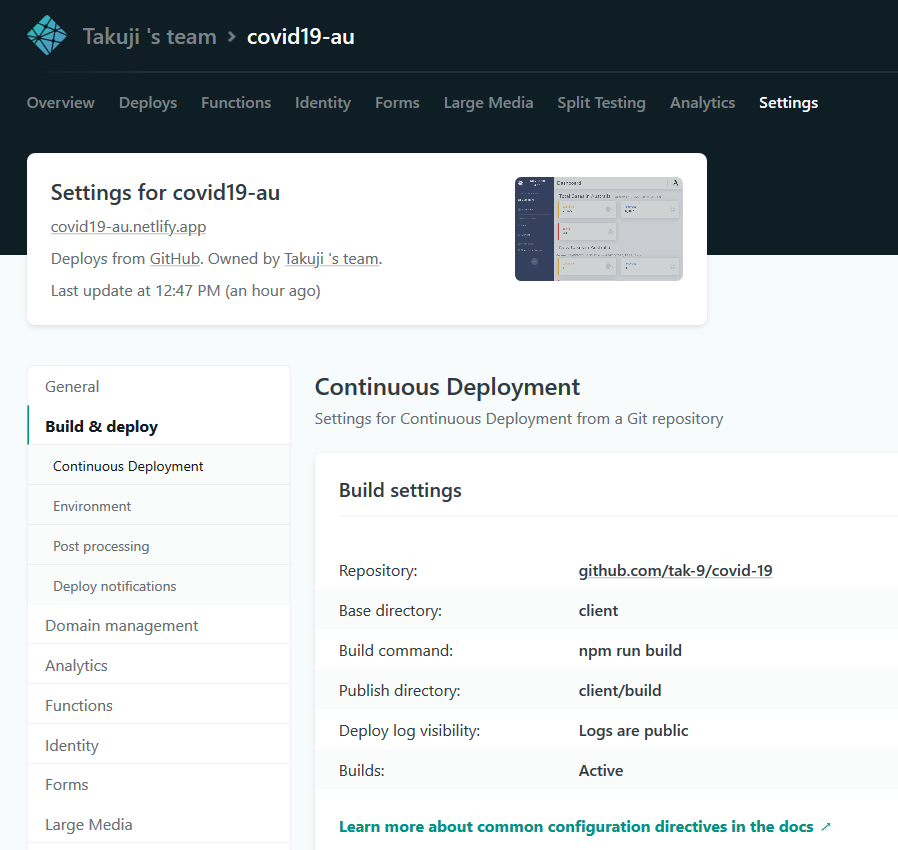
<https://www.netlify.com/blog/2016/09/29/a-step-by-step-guide-deploying-on-netlify/>

1. Choose a new site.

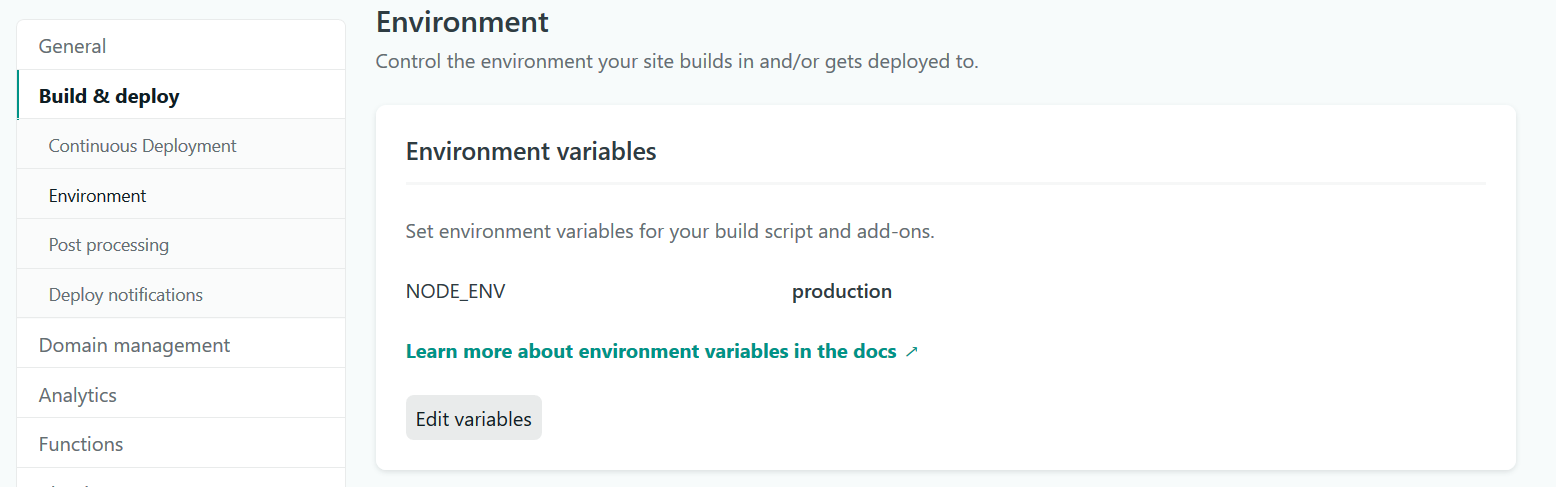


1. Choose a project > Settings > Continuous Deployment

* Base directory: client
* Build command: npm run build
* Publish directory: client/build



# Set Environment Variable



# Modify React Application

./util/env.js

export const serverUrl = (process.env.NODE\_ENV === "production") ? "https://covid19-au.herokuapp.com" : "http://localhost:3001";

Calling API

import { serverUrl } from '../util/env';

        axios.post(serverUrl + '/api/user/', {

            username: this.state.username,

            password: this.state.password

        })

            .then(response => {

# Modify Express

Add Netlify hostname to cors origin to allow API access from Netlify.

const cors = require("cors");

app.use(

  cors({

    origin: ["http://localhost:3000", "https://covid19-au.netlify.app"],// allow to server to accept request from different origin

    methods: "GET,HEAD,PUT,PATCH,POST,DELETE",

    credentials: true // allow session cookie from browser to pass through

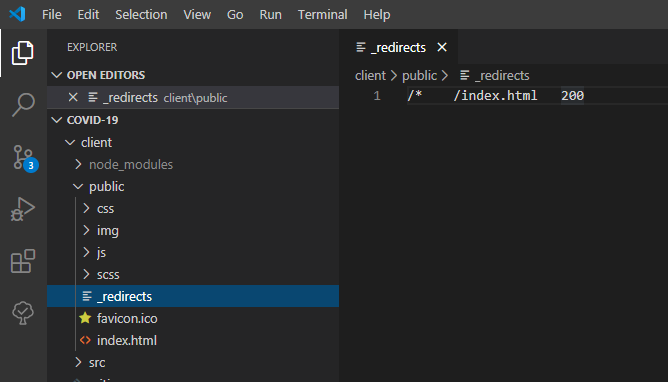
  })

);

# Modify Application for BrowserRouter

* create \_redirects in React’s public directory
* See this page for details.

<https://medium.com/@rexben/how-to-fix-page-not-found-on-netlify-with-react-router-dom-e0520692be5>



# Optional

In the landing page, useEffect() or componentDidMount(), make a dummy access to heroku to wake up Heroku server. The Heroku server should be up when a user click a button or do something.

  useEffect(()=>{

        // Wakeup Heroku when accessed from Netlify

        if (window.location.hostname === "covid19-au.netlify.app") {

            console.log("Accessing https://covid19-au.herokuapp.com/img/login.jpg This is only for netlify");

            axios.get("https://covid19-au.herokuapp.com/img/login.jpg");

        }

    }

  );

Wakeup Heroku when opening landing page.

# Troubleshooting

<https://community.netlify.com/t/support-guide-my-site-deploy-fails-unless-netlifys-build-cache-is-cleared/3035>

*Fairly rare but worth mentioning: You have a package.json in your repo root, but you are building a subdirectory of your repo and have NOT appropriately set the base directory in your build settings. That can lead to some confusing caching situations since we’ll only sometimes use that package.json (when there is nocache).*

Base directory must be set. Set as client

If you notice error message of module is missing in the Netlify deployment log file, run “npm install *package-name* –save” in the React client directory.