

PW - Deployment Process

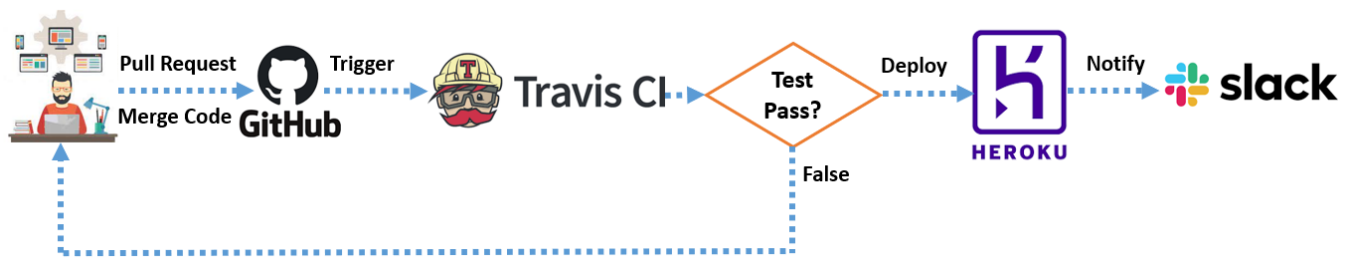
1. Introduction:

This page will show the process of how to deploy the project.

2. Github repository link:

<https://github.com/trundn/COMP90082-Assignment>

3. CI/CD Pipeline



3.1. Travis CI configuration file

```



language: node_js
node_js:
  - '14'
cache:
  - directories:
    - - node_modules
script:
  - npm run build
branches:
  - only:
    - - sprint2
notifications:
  - slack: $SLACK_KEY
deploy:
  - provider: heroku
  - skip_cleanup: true
  - api_key: $HEROKU_API_KEY
  - app: comp90082-pure-and-lazy
  - on:
    - - branch: sprint2


```

3.2. Travis CI environment variables

Environment Variables

Customize your build using environment variables. For secure tips on generating private keys [read our documentation](#)

HEROKU_API_KEY		Available to all branches	
----------------	---	---------------------------	---

 If your secret variable has special characters like `&`, escape them by adding `\` in front of each special character. For example, `ma&w!doc` would be entered as `ma\&w\!doc`.

3.3. Heroku environment variables

Config Vars

Config vars change the way your app behaves. In addition to creating your own, some add-ons come with their own.

Config Vars

[Hide Config Vars](#) 

3.4. Auth0 configuration

3.4.1. Application settings

Allowed Callback URLs

http://localhost:4200, https://comp90082-pure-and-lazy.herokuapp.com

After the user authenticates we will only call back to any of these URLs. You can specify multiple valid URLs by comma-separating them (typically to handle different environments like QA or testing). Make sure to specify the protocol (`https://`) otherwise the callback may fail in some cases. With the exception of custom URI schemes for native clients, all callbacks should use protocol `https://` . You can use [Organization URL](#) parameters in these URLs.

Allowed Logout URLs

http://localhost:4200, https://comp90082-pure-and-lazy.herokuapp.com

Allowed Web Origins

```
http://localhost:4200, https://comp90082-pure-and-lazy.herokuapp.com
```

Comma-separated list of allowed origins for use with Cross-Origin Authentication, Device Flow, and web message response mode, in the form of `<scheme> "://" <host> [":" <port>]`, such as `https://login.mydomain.com` or `http://localhost:3000`. You can use wildcards at the subdomain level (e.g.: `https://*.contoso.com`). Query strings and hash information are not taken into account when validating these URLs.

3.4.2. Script for Auth Pipeline Rule

⚡ Getting Started

📈 Activity

📄 Applications >

🔑 Authentication >

🏢 Organizations NEW

👤 User Management >

🎨 Branding >

🛡 Security >

🔧 Actions >

🔗 Auth Pipeline >

Rules

Hooks

Rules + CREATE

Custom Javascript snippets that run in a secure, isolated sandbox in the Auth0 service as part of your authentication pipeline. [Learn more](#)

⚠ We recommend you to explore Actions, our newest extensibility product now in General Availability. Actions gives you many of the same functions as Rules and more. Get started today in [Actions](#). [Learn more about Actions](#).

TRY ALL RULES WITH... ⌵ 🔄 REFRESH

Indicate user needs to complete registration details and redirect to registration completion page 🟢 ⋮

```

/**
 * Auth0 Rule: Indicate user needs to complete registration details and redirect to reg
 */
function (user, context, callback) {
  // Short-circuit if the user is signed up already, is using a refresh token,
  // or is using silent auth
  if (context.stats.loginsCount > 1 || context.protocol === 'oauth2-refresh-token' ||
    context.request.query.prompt === 'none') {
    return callback(null, user, context);
  }

  // Set registration status
  user.user_metadata = user.user_metadata || {};
  user.user_metadata.registration_complete = false;
  auth0.users.updateUserMetadata(user.user_id, user.user_metadata)
    .then(function() {
      callback(null, user, context);
    })
    .catch(function(err) {
      callback(err);
    });
}

```

4. Run on Developer Environment

Config the environment variables

Nodejs package an NPM package

For the windows platform and MacOS platform

Nodejs download website: <https://nodejs.org/en/download/>

Version check command:

node -v

Install MongoDB Application

Download MongoDB Community

install link: <https://www.mongodb.com/try/download/community>

Connect to MongoDB

Connection link:

<mongodb+srv://trungnd:Pass4eportfolio@cluster0.xjxw5.mongodb.net/pureandlazydb?retryWrites=true&w=majority>

Install instruction:

Clone the project to local

Enter the command into the terminal:

```
git clone https://github.com/trundn/COMP90082-Assignment.git
```

Switch to project directory:

```
cd COMP90082-Assignment
```

Install dependencies:

```
npm install
```

Setting up the environmental file

Create local file name as .env to set up MongoDB connect file.

copy variable into .env file: MONGODB_PASSWORD = 'Pass4eportfolio'

Lunch the project:

```
npm start
```

Lunch the website

Open the browser and enter the URL localhost:4200