Table Tennis Information Tracker Maintenance Documentation

Table of Contents

Overall Note

Everything in this document has been set up and there is no need to change anything. This document is only to give you the basics of the tools used to run the Table Tennis Information Tracker Application. If you plan on updating the application and adding in new feature, please reach out to us so we can give you a more comprehensive run-down of the specifics needed to do you want to do.

IMPORTANT NOTE:

This application **WILL** go dormant if not used for an extended period of time (like long breaks such as Winter or Summer). This will cause the whole app to basically be erased. The easiest way to mitigate this is to have one person log in at least once every few weeks. A single use will make the app come back to life and not erase its data due to lack to use.

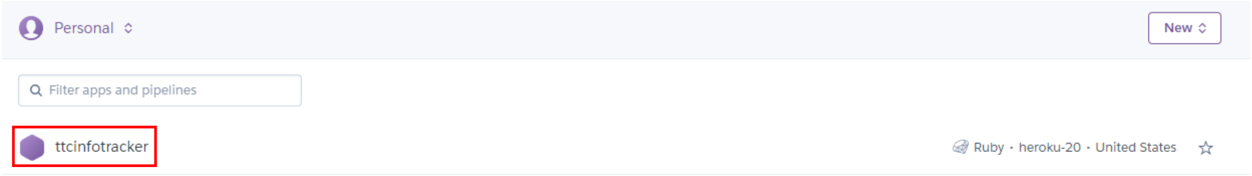
Heroku

What is Heroku and why it significant?

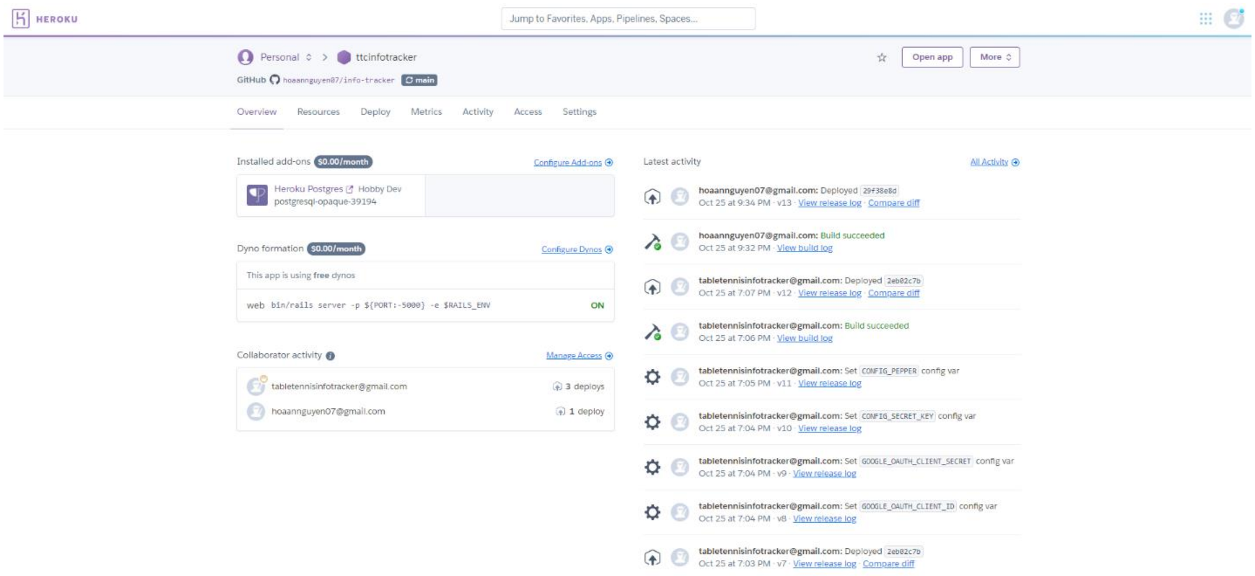
Heroku is a service used to host the Table Tennis Information Tracker application. It is the centralized location which brings all components required to run the application together. This is the **most** important service that keeps this application running for the Table Tennis Club.

Access Application

1. Visit https://id.heroku.com/login
2. Sign in with credentials provided during the application transfer process
3. Click on '***ttcinfotracker***'



You should be directed to the page below:



GitHub `hoaannguyen07/info-tracker` main

Installed add-ons $0.00/month

Heroku Postgres Hobby Dev
postgresql-opaque-39194

Dyno formation $0.00/month

This app is using **free** dynos

```
web bin/rails server -p ${PORT:-5000} -e $RAILS_ENV                ON
```

Collaborator activity

tabletennisinfotracker@gmail.com	3 deploys
hoaannguyen07@gmail.com	1 deploy

Latest activity

hoaannguyen07@gmail.com: Deployed `29f38e8d`
Oct 25 at 9:34 PM · v13 · View release log · Compare diff

hoaannguyen07@gmail.com: Build succeeded
Oct 25 at 9:32 PM · View build log

tabletennisinfotracker@gmail.com: Deployed `2eb02c7b`
Oct 25 at 7:07 PM · v12 · View release log · Compare diff

tabletennisinfotracker@gmail.com: Build succeeded
Oct 25 at 7:06 PM · View build log

tabletennisinfotracker@gmail.com: Set `CONFIG_PEPPER` config var
Oct 25 at 7:05 PM · v11 · View release log

tabletennisinfotracker@gmail.com: Set `CONFIG_SECRET_KEY` config var
Oct 25 at 7:04 PM · v10 · View release log

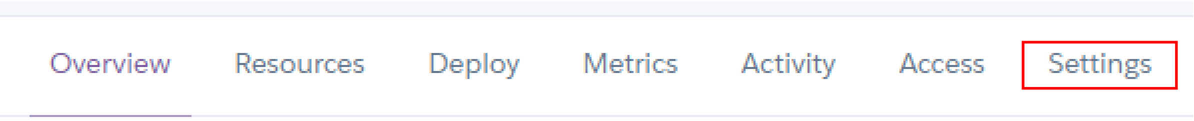
tabletennisinfotracker@gmail.com: Set `GOOGLE_OAUTH_CLIENT_SECRET` config var
Oct 25 at 7:04 PM · v9 · View release log

tabletennisinfotracker@gmail.com: Set `GOOGLE_OAUTH_CLIENT_ID` config var
Oct 25 at 7:04 PM · v8 · View release log

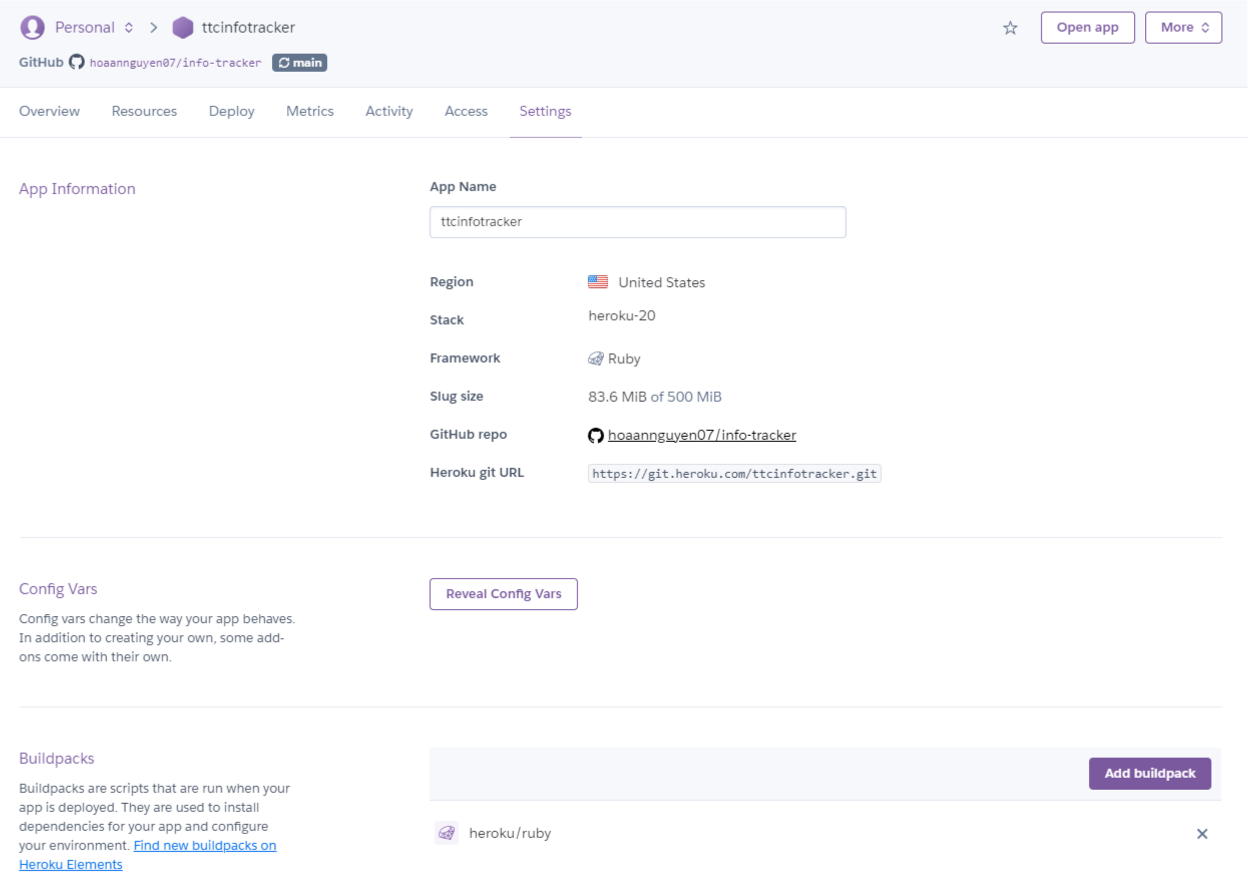
tabletennisinfotracker@gmail.com: Deployed `2eb02c7b`
Oct 25 at 7:03 PM · v7 · View release log · Compare diff

Get Link to Application

1. Go to the app on Heroku
2. There are 2 ways to get the link:
    a. In Heroku itself
        i. Go to the '***Settings***' tab



You should be directed to the page below:



App Information

App Name: ttcinfotracker

Region: United States

Stack: heroku-20

Framework: Ruby

Slug size: 83.6 MiB of 500 MiB

GitHub repo: hoaannguyen07/info-tracker

Heroku git URL: `https://git.heroku.com/ttcinfotracker.git`

Config Vars

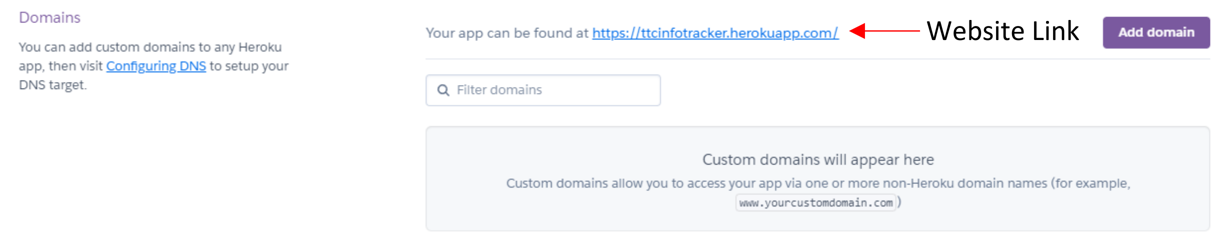
Config vars change the way your app behaves. In addition to creating your own, some add-ons come with their own.

Buildpacks

Buildpacks are scripts that are run when your app is deployed. They are used to install dependencies for your app and configure your environment. Find new buildpacks on Heroku Elements

heroku/ruby

ii. Scroll to the bottom until you can see the '***Domains***' section



Domains

You can add custom domains to any Heroku app, then visit Configuring DNS to setup your DNS target.

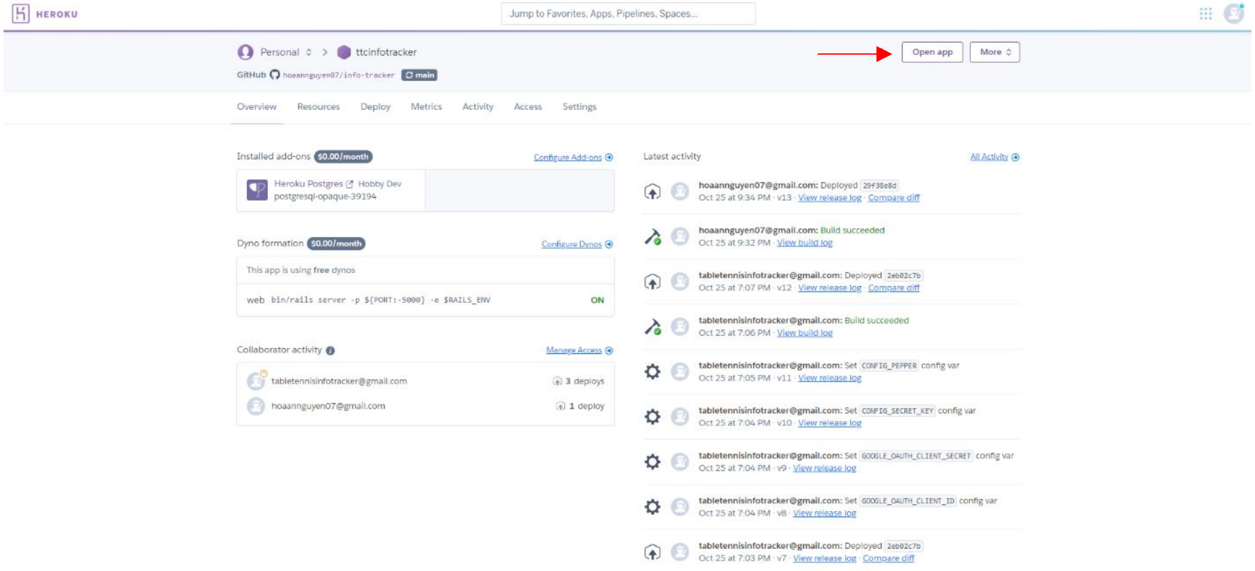
Your app can be found at https://ttcinfotracker.herokuapp.com/ ← Website Link

Custom domains will appear here

Custom domains allow you to access your app via one or more non-Heroku domain names (for example, `www.yourcustomdomain.com`)

b. Open the Application itself and get the URL

   i. Click on '***Open App***' button (can be accessed from anywhere in the app in Heroku



GitHub hoaannguyen07/info-tracker main

Installed add-ons $0.00/month

Heroku Postgres Hobby Dev
postgresql-opaque-39194

Dyno formation $0.00/month

This app is using **free** dynos

```
web bin/rails server -p ${PORT:-5000} -e $RAILS_ENV    ON
```

Collaborator activity

tabletennisinfotracker@gmail.com 3 deploys
hoaannguyen07@gmail.com 1 deploy

Latest activity

hoaannguyen07@gmail.com: Deployed 29f38e8d
Oct 25 at 9:34 PM · v13 · View release log · Compare diff

hoaannguyen07@gmail.com: Build succeeded
Oct 25 at 9:32 PM · View build log

tabletennisinfotracker@gmail.com: Deployed 2eb02c7b
Oct 25 at 7:07 PM · v12 · View release log · Compare diff

tabletennisinfotracker@gmail.com: Build succeeded
Oct 25 at 7:06 PM · View build log

tabletennisinfotracker@gmail.com: Set CONFIG_PEPPER config var
Oct 25 at 7:05 PM · v11 · View release log

tabletennisinfotracker@gmail.com: Set CONFIG_SECRET_KEY config var
Oct 25 at 7:04 PM · v10 · View release log

tabletennisinfotracker@gmail.com: Set GOOGLE_OAUTH_CLIENT_SECRET config var
Oct 25 at 7:04 PM · v9 · View release log

tabletennisinfotracker@gmail.com: Set GOOGLE_OAUTH_CLIENT_ID config var
Oct 25 at 7:04 PM · v8 · View release log

tabletennisinfotracker@gmail.com: Deployed 2eb02c7b
Oct 25 at 7:03 PM · v7 · View release log · Compare diff

   ii. You will be directed to the Table Tennis Information Tracker application on a new tab and then you can copy the URL

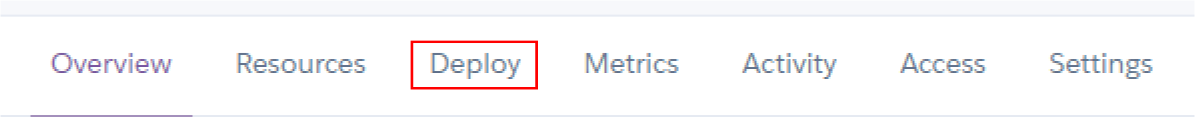
Application Specifics

Below are specifics on how the application is ran and all the different topics we, as developers, think you should be aware of in case you want to work on the application in the future. That being said, everything has been set up by us before the transfer of ownership of this application and additional work needs to be done. If there are additional questions in the future, you can reach out to us for help.

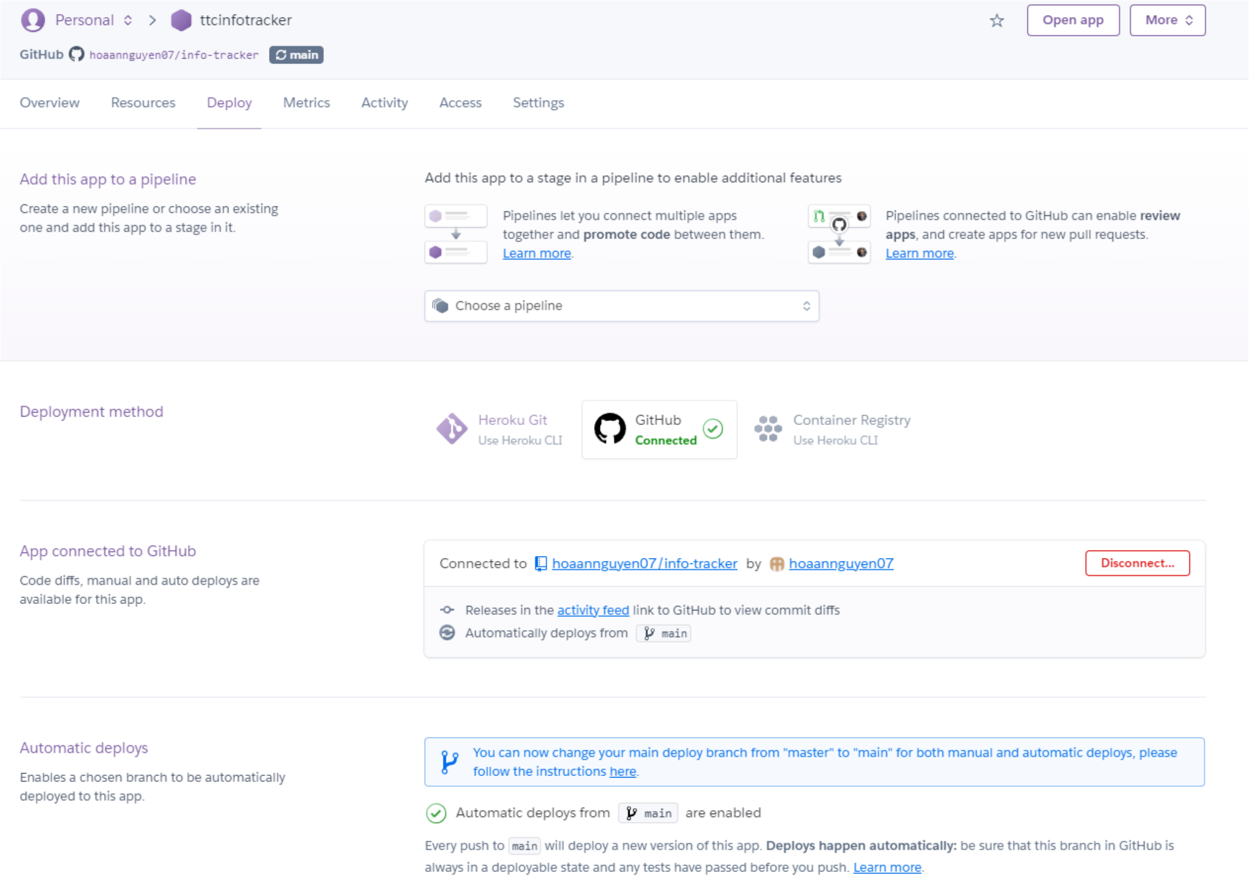
Application Deployment

We have set up the application to automatically deploy whenever the 'main' branch of the repository has been updated (through pull requests on GitHub). However, if there is ever a need to manually re-deploy the application, then follow the steps bellow:

1. Click on the '***Deploy***' tab



You should be directed to the page below:



Add this app to a pipeline

Create a new pipeline or choose an existing one and add this app to a stage in it.

Add this app to a stage in a pipeline to enable additional features

Pipelines let you connect multiple apps together and **promote code** between them. Learn more.

Pipelines connected to GitHub can enable **review apps**, and create apps for new pull requests. Learn more.

Deployment method

App connected to GitHub

Code diffs, manual and auto deploys are available for this app.

Connected to hoaannguyen07/info-tracker by hoaannguyen07

Releases in the activity feed link to GitHub to view commit diffs

Automatically deploys from `main`

Automatic deploys

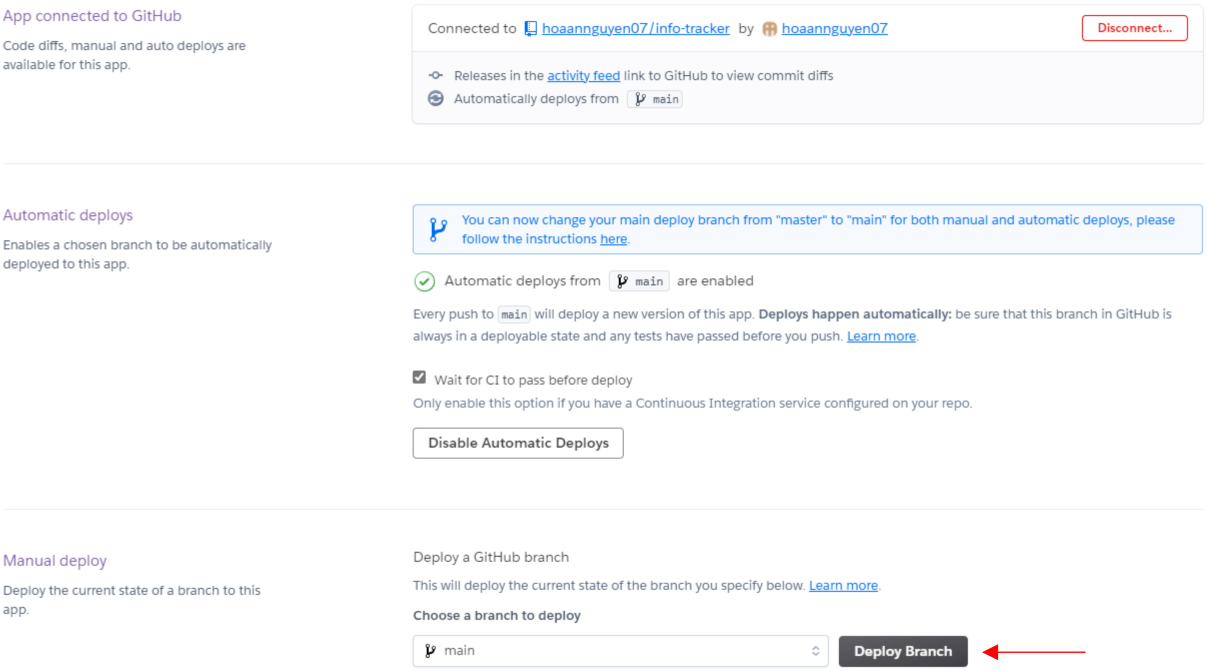
Enables a chosen branch to be automatically deployed to this app.

You can now change your main deploy branch from "master" to "main" for both manual and automatic deploys, please follow the instructions here.

✓ Automatic deploys from `main` are enabled

Every push to `main` will deploy a new version of this app. **Deploys happen automatically:** be sure that this branch in GitHub is always in a deployable state and any tests have passed before you push. Learn more.

2. Scroll to the bottom and click on ‘***Deploy Branch***’



App connected to GitHub

Code diffs, manual and auto deploys are available for this app.

Connected to hoaannguyen07/info-tracker by hoaannguyen07

Disconnect...

Releases in the activity feed link to GitHub to view commit diffs

Automatically deploys from `main`

Automatic deploys

Enables a chosen branch to be automatically deployed to this app.

You can now change your main deploy branch from "master" to "main" for both manual and automatic deploys, please follow the instructions here.

✓ Automatic deploys from `main` are enabled

Every push to `main` will deploy a new version of this app. **Deploys happen automatically:** be sure that this branch in GitHub is always in a deployable state and any tests have passed before you push. Learn more.

☑ Wait for CI to pass before deploy

Only enable this option if you have a Continuous Integration service configured on your repo.

Disable Automatic Deploys

Manual deploy

Deploy the current state of a branch to this app.

Deploy a GitHub branch

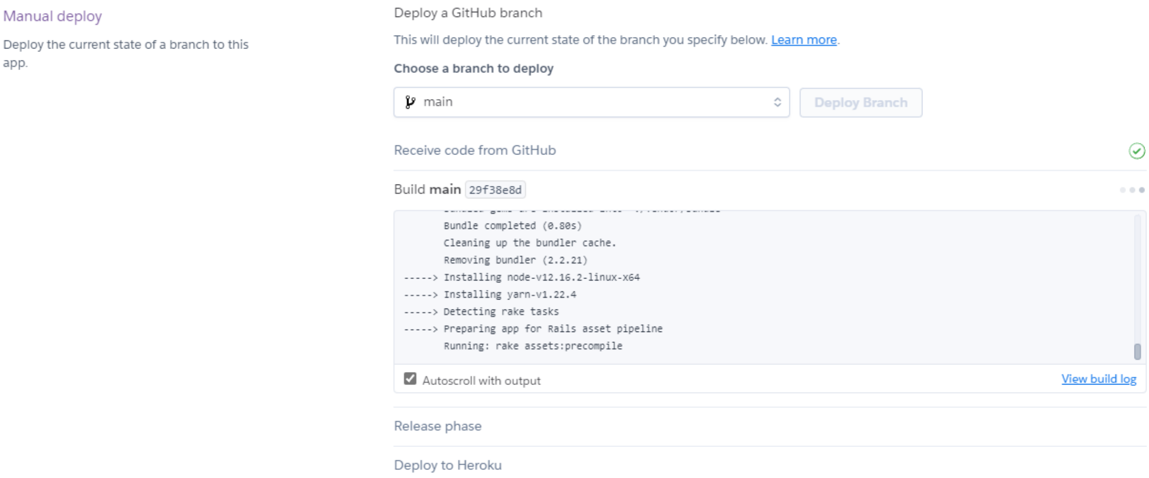
This will deploy the current state of the branch you specify below. Learn more.

Choose a branch to deploy

main

Deploy Branch

3. Application will start building and get ready for re-deployment



Manual deploy

Deploy the current state of a branch to this app.

Deploy a GitHub branch

This will deploy the current state of the branch you specify below. Learn more.

Choose a branch to deploy

main

Deploy Branch

Receive code from GitHub ✓

Build **main** `29f38e8d`

```
       Bundle completed (0.80s)
       Cleaning up the bundler cache.
       Removing bundler (2.2.21)
-----> Installing node-v12.16.2-linux-x64
-----> Installing yarn-v1.22.4
-----> Detecting rake tasks
-----> Preparing app for Rails asset pipeline
       Running: rake assets:precompile
```

☑ Autoscroll with output

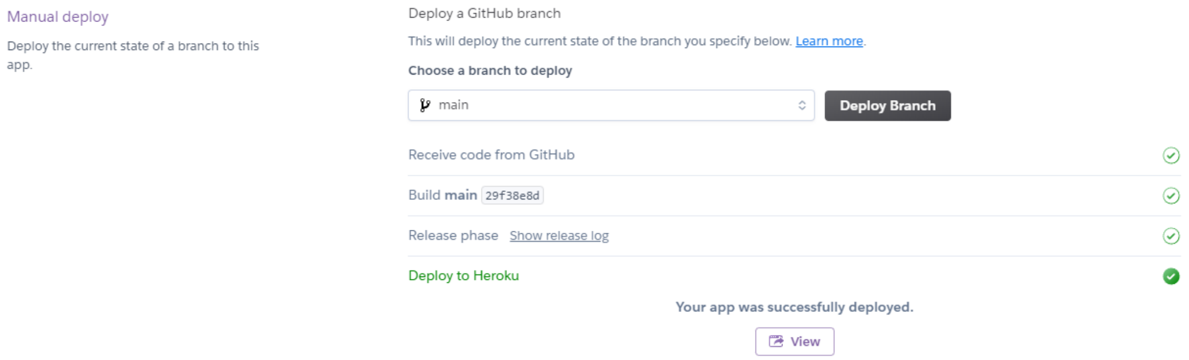
View build log

Release phase

Deploy to Heroku

You can click on ‘***View build log***’ to see what it is actually doing but this will mostly be for future developers to use

4. When build is done and re-deployment is successful, you should see something like this


Manual deploy

Deploy the current state of a branch to this app.

Deploy a GitHub branch

This will deploy the current state of the branch you specify below. Learn more.

Choose a branch to deploy

main

Receive code from GitHub

Build main 29f38e8d

Release phase Show release log

Deploy to Heroku

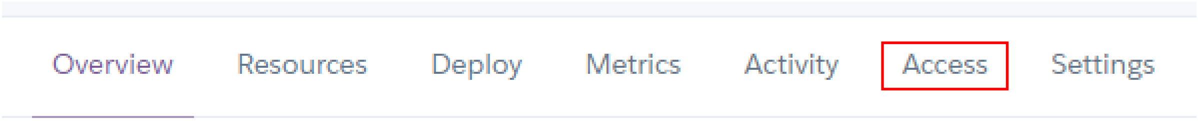
Your app was successfully deployed.

5. If the build failed, you could go to the ‘***Activity***’ tab and view the logs of the build and re-deployment to see what went wrong
    a. There are times that the building phase would succeed but the deployment phase would fail

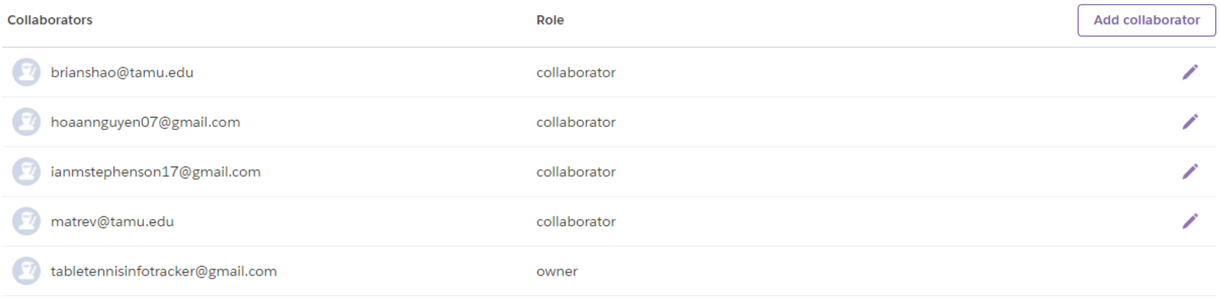
Add Access

If you have people you want to give access to in order to further develop this project, you can give their Heroku accounts access to the ‘***ttcinfotracker***’. This will allow them to view and make changes to this app on their Heroku account.

1. Go to the ‘*Access*’ tab



You should see a list of names who have access to this app on Heroku. It should look similar to this image:



Collaborators	Role
brianshao@tamu.edu	collaborator
hoaannguyen07@gmail.com	collaborator
ianmstephenson17@gmail.com	collaborator
matrev@tamu.edu	collaborator
tabletennisinfotracker@gmail.com	owner

2. Click on ‘***Add collaborator***’
3. Add the email address of the Heroku account you want to give access to and click ‘*Save Changes*’
4. This will send an email to the person you want to give access to, telling them that they have been given access to the application on Heroku. When they log into Heroku with their credentials, they should see the ‘***ttcinfotracker***’ app.

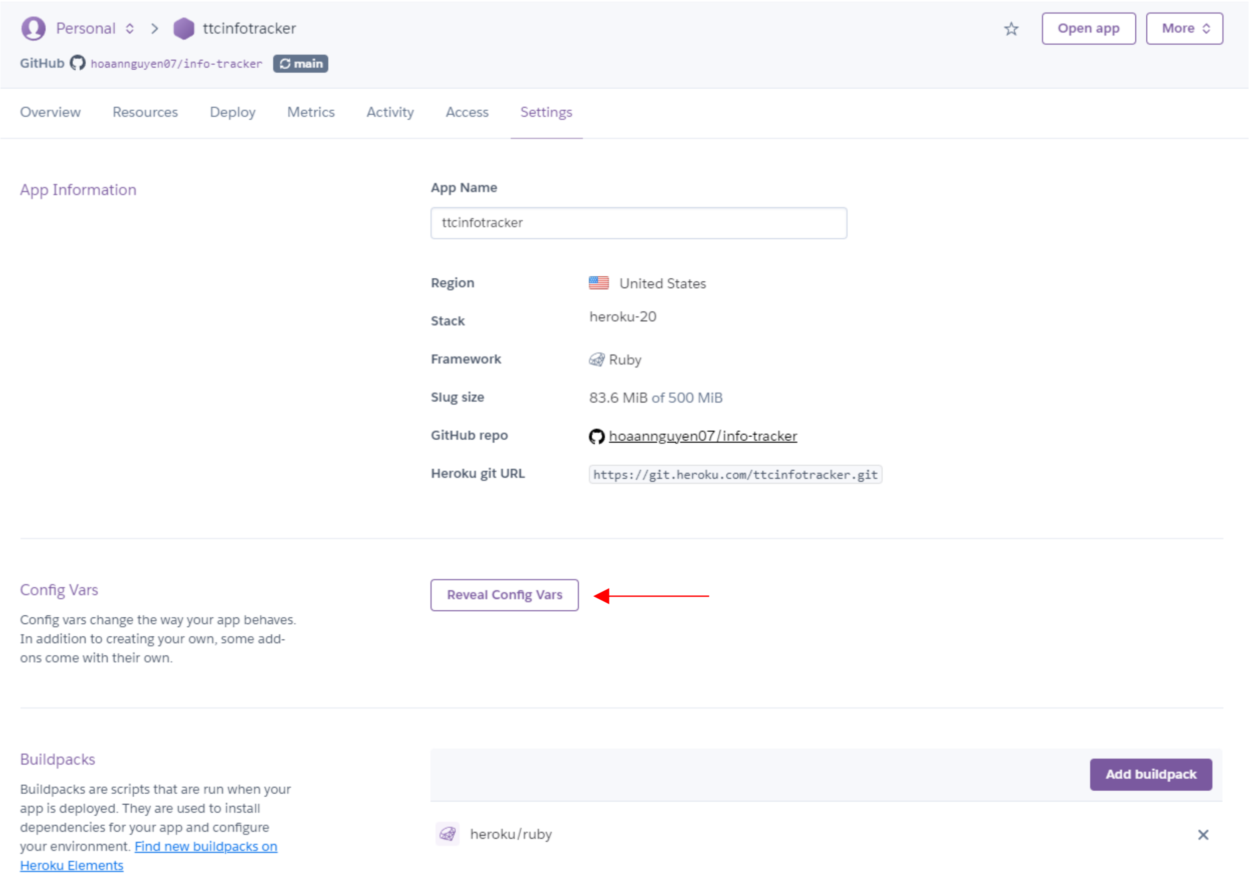
Environment Variables

There are 4 environment variables you should be aware of, and they are critical to the running of this application. These are the secret keys to connect to the Google API and Azure Storage Blob, which will be expanded upon later in this document. The Google API and Azure Storage Blob environment variables come in pairs: Client ID and Client Secret, which is why there are 4 in total. These keys have been made and no changes will be required whatsoever unless you decide to do away with Google Authentication or change Azure Storage Blobs.

Note: These keys are confidential and will cause a major security breach if known publicly. It is important that you do not share these keys unless necessary. During development, it is best practice to make new keys to locally test and make sure the production app (app currently used by users) has a separate set of keys that only it uses. Make sure you know what you are doing if you are going to change a key.

To view these keys, follow the step bellow:

1. Go to the ‘***Settings***’ tab
2. Click on ‘***Reveal Config Vars***’ in the **“Config Vars”** section



GitHub hoaannguyen07/info-tracker main

App Name: ttcinfotracker

Region: United States

Stack: heroku-20

Framework: Ruby

Slug size: 83.6 MiB of 500 MiB

GitHub repo: hoaannguyen07/info-tracker

Heroku git URL: `https://git.heroku.com/ttcinfotracker.git`

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

Buildpacks are scripts that are run when your app is deployed. They are used to install dependencies for your app and configure your environment. Find new buildpacks on Heroku Elements

heroku/ruby

Transfer Application

In case you want to transfer the ownership of this application to someone else's Heroku account, you use the follow steps:

Note: make sure that you have added that person as a collaborator (see the Add Access section for more information)

1. Go to the '***Settings'*** tab
2. Scroll to the **"Transfer Ownership"** section
3. Choose the Heroku account you want to transfer the ownership of the application to
4. Click '***Transfer App…***'

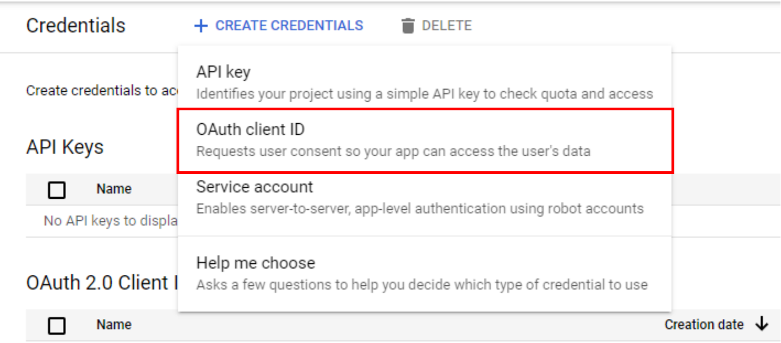
Google API

What is Google API and why it significant?

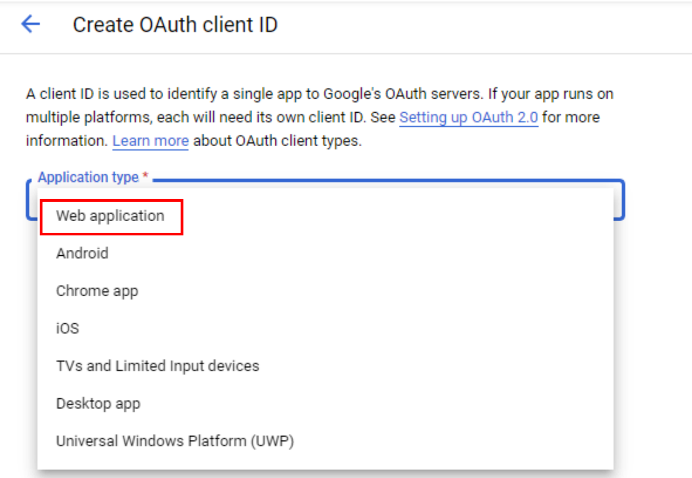
Google API is a service the Table Tennis Information Tracker application uses to allow members to sign into the application (authentication system). The authentication of the application has been delegated to a third-party authenticator to ensure the highest possible level of security. This is the reason for the Google Client ID and Secret keys included in the set of environment variables. This application has been registered with Google to allow it to use Google's authentication system.

In order to view keys or create new keys (possibly for development purposes):

1. Visit https://console.developers.google.com/cloud-resource-manager
2. Log into your desired account
3. Navigate to the **"Credentials"** section (in dropdown menu) under the **"APIs & Services"** tab on the sidebar
    a. If you are logged in with credentials made for this application, then you will see the Table Tennis Information Tracker Application under **"OAuth 2.0 Client IDs"**
        i. Upon clicking on it, you can view the client id & secret
4. To create keys to test locally, you need not be logged into the Google account used for this application. Any account will work.
5. To create a new a new set of keys, follow these steps:
    a. On the **"Credentials"** tab, click on '***Create Credentials***' at the top
    b. Select '***OAuth client ID***'

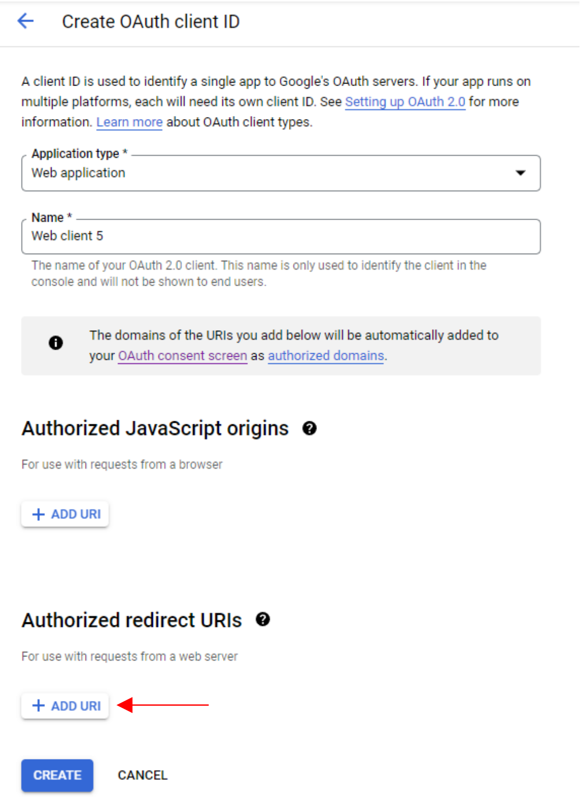

Credentials
+ CREATE CREDENTIALS
DELETE
Create credentials to ac
API Keys
Name
No API keys to displa
OAuth 2.0 Client I
Name
Creation date ↓
API key
Identifies your project using a simple API key to check quota and access
OAuth client ID
Requests user consent so your app can access the user's data
Service account
Enables server-to-server, app-level authentication using robot accounts
Help me choose
Asks a few questions to help you decide which type of credential to use

c. Under **“Application Type”**, choose Web Application



A client ID is used to identify a single app to Google's OAuth servers. If your app runs on multiple platforms, each will need its own client ID. See Setting up OAuth 2.0 for more information. Learn more about OAuth client types.

d. Click on ‘***Add URI***’ under **“Authorized redirect URIs”**



A client ID is used to identify a single app to Google's OAuth servers. If your app runs on multiple platforms, each will need its own client ID. See Setting up OAuth 2.0 for more information. Learn more about OAuth client types.

Application type *
Web application

Name *
Web client 5

The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.

The domains of the URIs you add below will be automatically added to your OAuth consent screen as authorized domains.

Authorized JavaScript origins

For use with requests from a browser

Authorized redirect URIs

For use with requests from a web server

e. Add the in the URL that will be called back to
   i. Should be: URL of application (ending with a ‘/’) + ‘admins/auth/google_oauth2/callback’
   ii. Example: https://ttcinfotracker.herokuapp.com/admins/auth/google_oauth2/callback
      1. This is the call back URI for the current Table Tennis Information Tracker production application

        2. ‘https://ttcinfotracker.herokuapp.com/’ is the URL of the app and the callback is ‘admins/auth/google_oauth2/callback’

   f. Click ‘***Save***’
   g. Use the Client ID and Secret (have to click on the app you just created again) as environment variables
      i. The environment variables are: GOOGLE_OAUTH_CLIENT_ID and GOOGLE_OAUTH_CLIENT_SECRET for the client id and secret, respectively

Azure

What is Azure and why it significant?

Azure is Microsoft’s Cloud platform service that the application uses to store images in the photo gallery. The use of secret keys for Azure is the same as Google API. However, there is no need to use Azure for development so there is no need to create new keys like in Google API. There is no need to worry about Azure unless the amount of photos in gallery is too large or you want to change the Azure Bucket housing all the application’s images. In the case of have too many photos, causing the bucket to run out of space, the easiest way to remedy this is to delete photos from the gallery. As for changing Azure Buckets, that requires setting up a new Azure Bucket and getting the keys for that. However, something to note is that if you switch buckets, by default, the images stored in the current own will not transfer over unless you transfer it over yourself, which will require research on your end.