SKY@TAMU Maintenance Documentation

Table of Contents

Overall Note

This document serves as a tutorial on how to set up the application in Heroku. This is already done, but if significant changes are made to the website, a Heroku reset may be required.

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 of use.

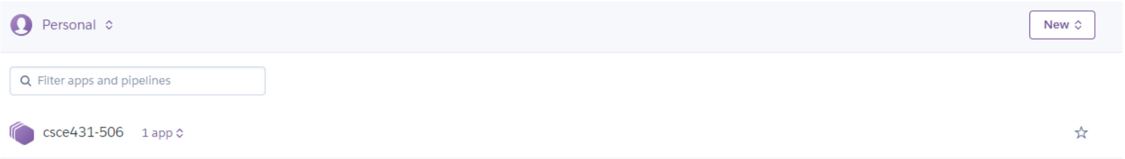
Heroku

What is Heroku and why is it significant?

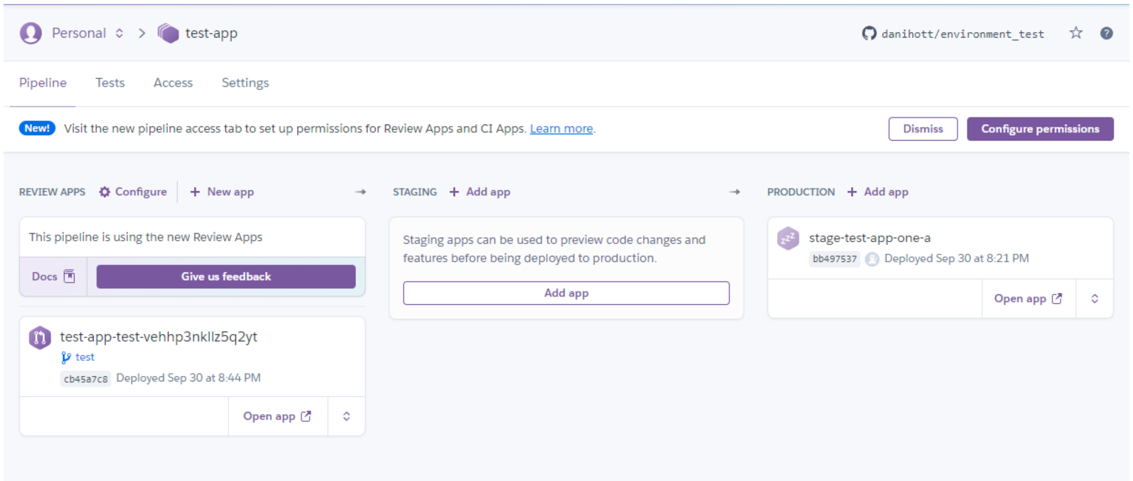
Heroku is a service used to host the SKY@TAMU Website. It is the centralized location which brings all components required to run the application together. This is the **most** important service that keeps this application running for SKY.

Access Application

1. Visit https://id.heroku.com/login
2. Sign in with @tamu.edu credentials provided during the application transfer process
3. Click on the application. Should be "Skyattamu" or something similar



You should be directed to the page below:



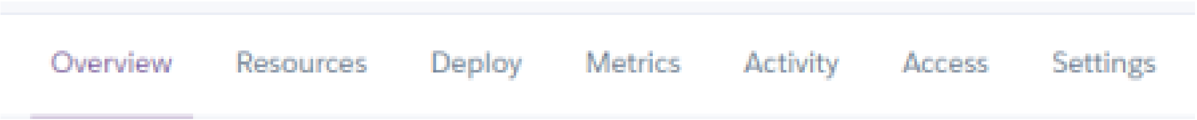
Visit the new pipeline access tab to set up permissions for Review Apps and CI Apps. Learn more.

This pipeline is using the new Review Apps

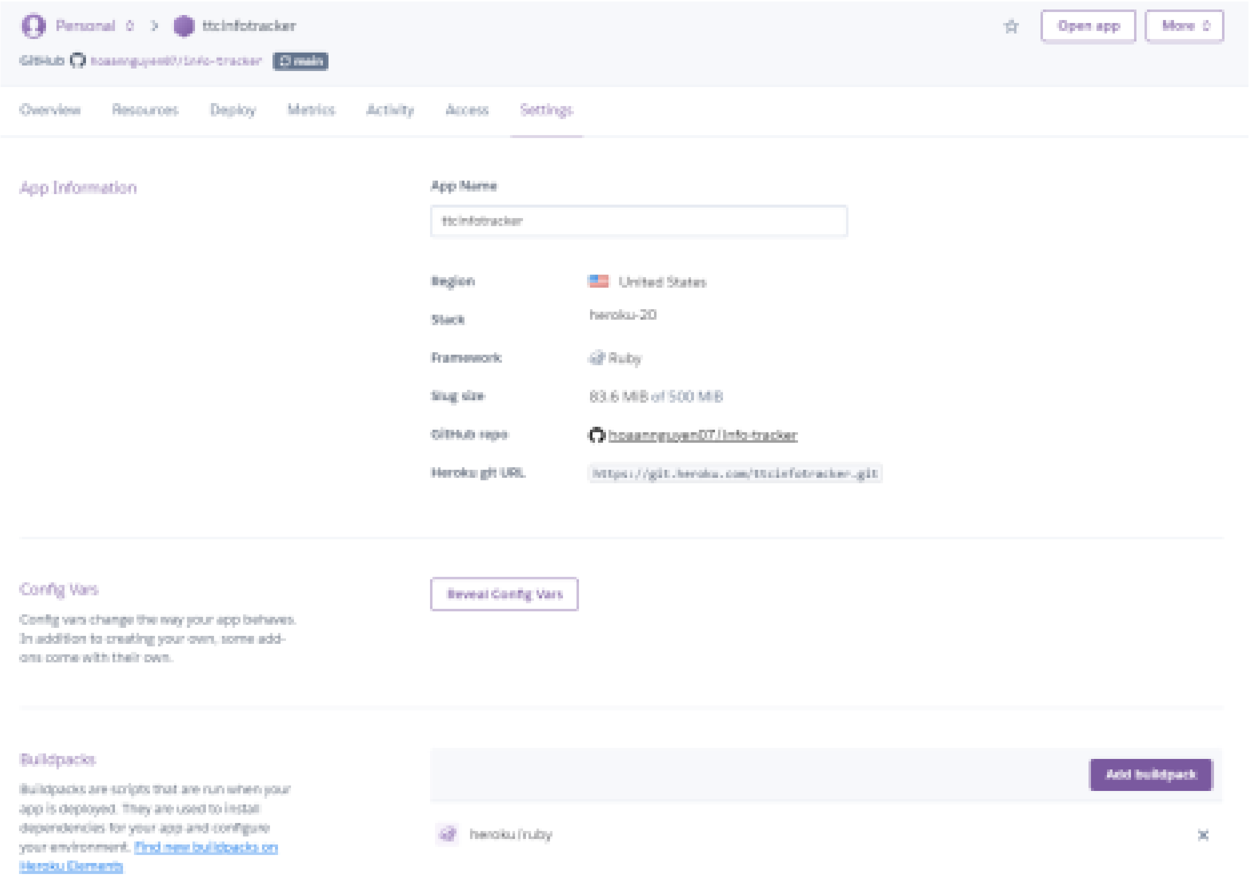
Staging apps can be used to preview code changes and features before being deployed to production.

Get Link to Application

1. Click on the application name under “Production”
   a. 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: hoaanmguyen07/info-tracker

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

Config Vars

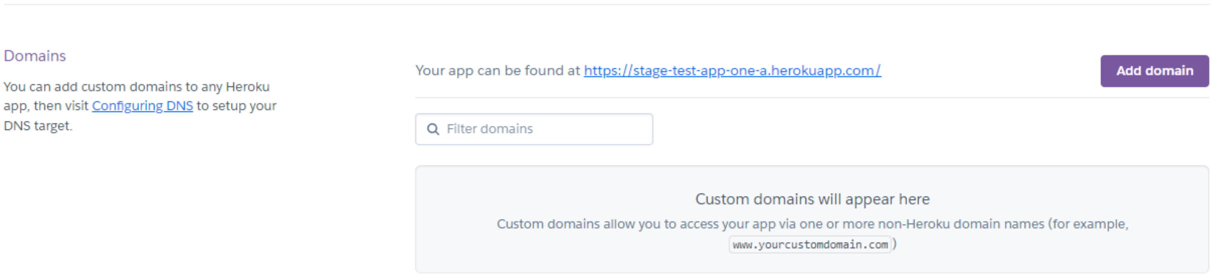
Config vars change the way your app behaves. In addition to creating your own, some add-ons come with their own.

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://stage-test-app-one-a.herokuapp.com/

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`)

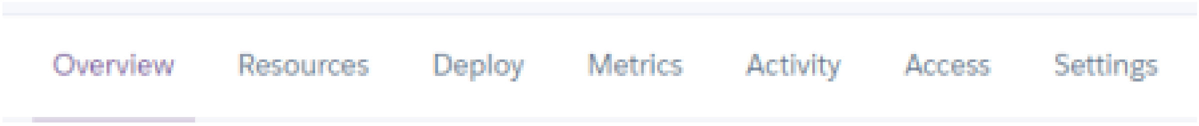
Application Specifics

Below are specifics on how the application is run 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 no additional work needs to be done.

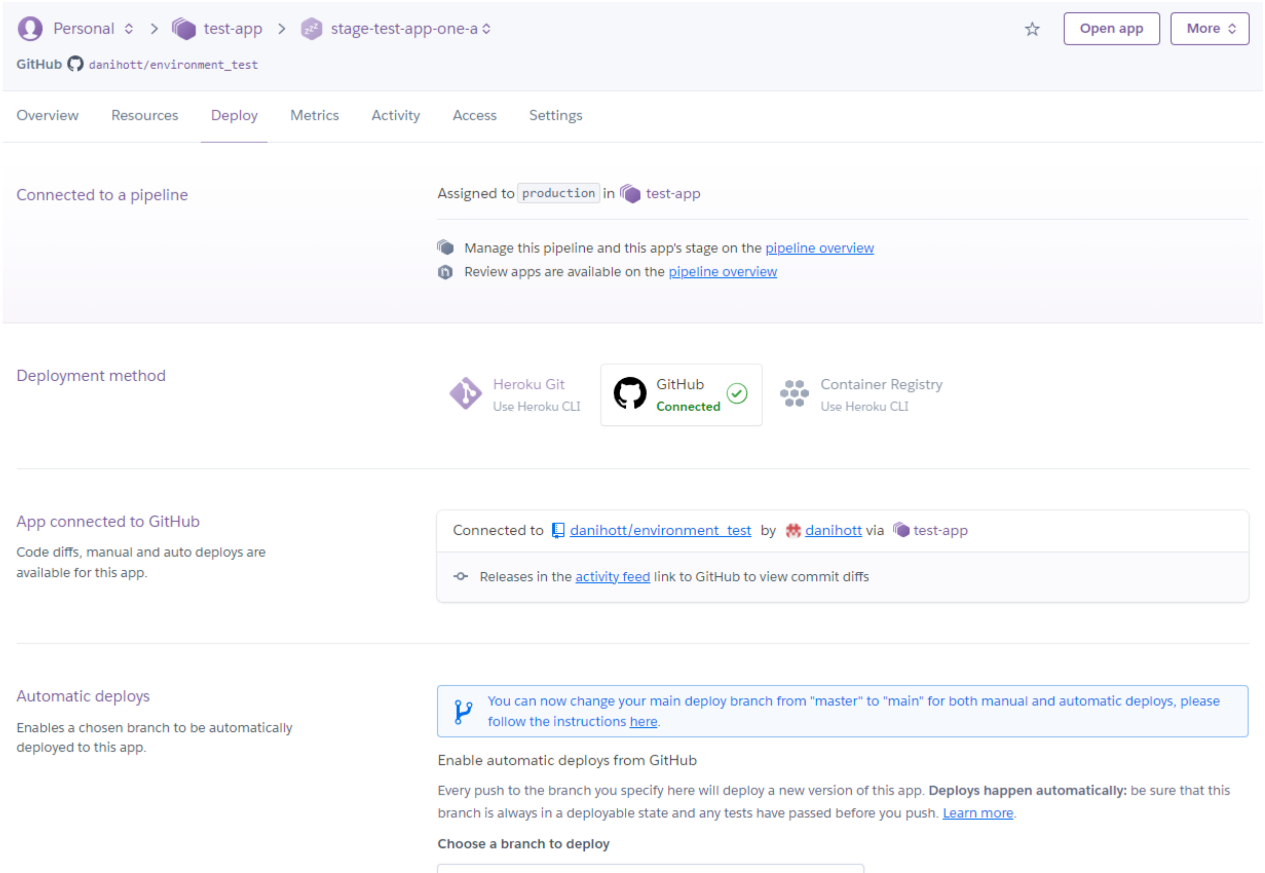
Application Deployment

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:



Connected to a pipeline

Assigned to `production` in test-app

Manage this pipeline and this app's stage on the pipeline overview

Review apps are available on the pipeline overview

Deployment method

App connected to GitHub

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

Connected to danihott/environment_test by danihott via test-app

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

Automatic deploys

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

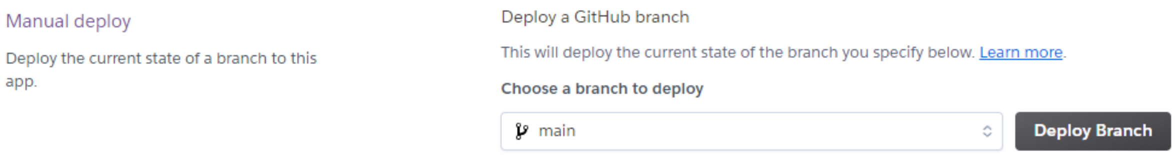
You can now change your main deploy branch from "master" to "main" for both manual and automatic deploys, please follow the instructions here.

Enable automatic deploys from GitHub

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

Choose a branch to deploy

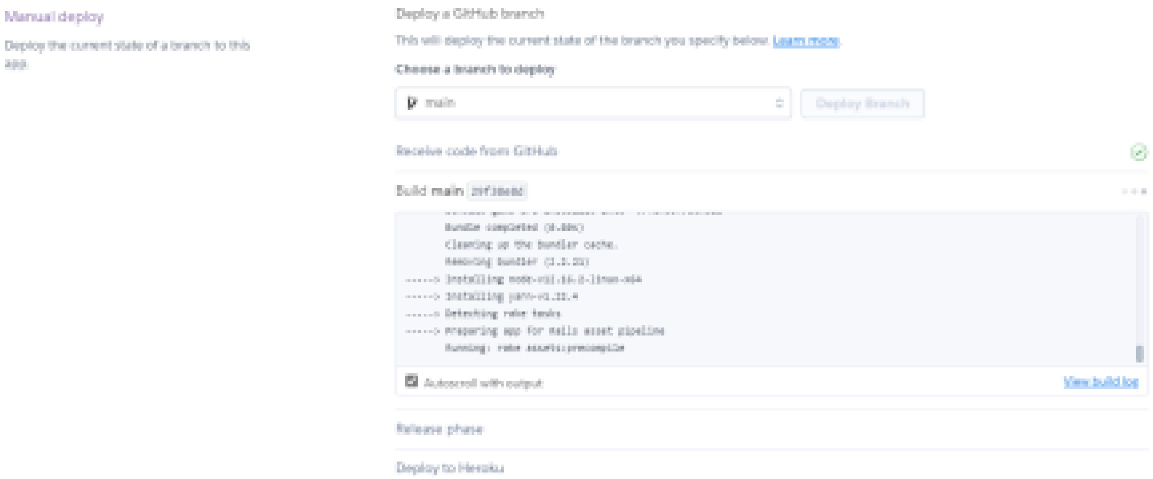
2. Scroll to the bottom under “Manual Deploy” and click on ‘***Deploy Branch***’



Deploy the current state of a branch to this app.

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

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



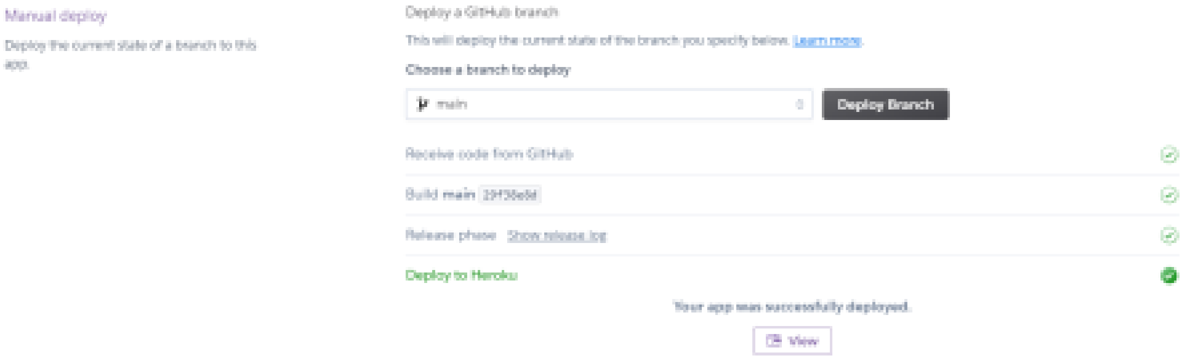
Deploy the current state of a branch to this app.

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

```
       Bundle completed (0.00s)
       Cleaning up the bundler cache.
       Removing bundler (2.1.23)
-----> 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
```

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



Deploy the current state of a branch to this app.

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

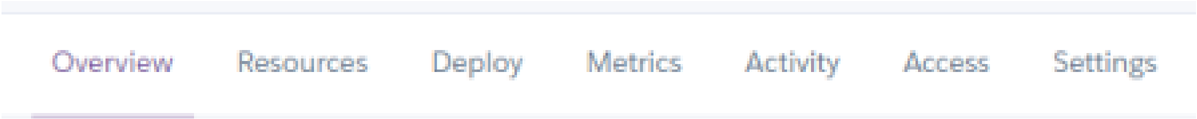
Your app was successfully deployed.

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.

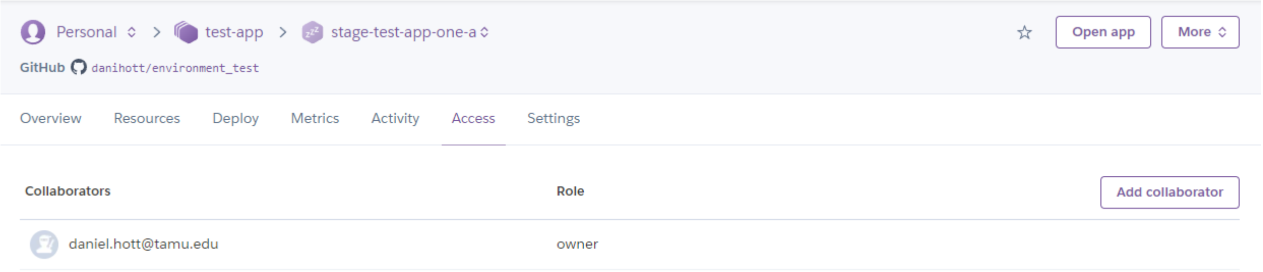
Add Access

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 '***skyattamu***'. 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:



Collaborators	Role
daniel.hott@tamu.edu	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 **'skyattamu'** app.

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…***'

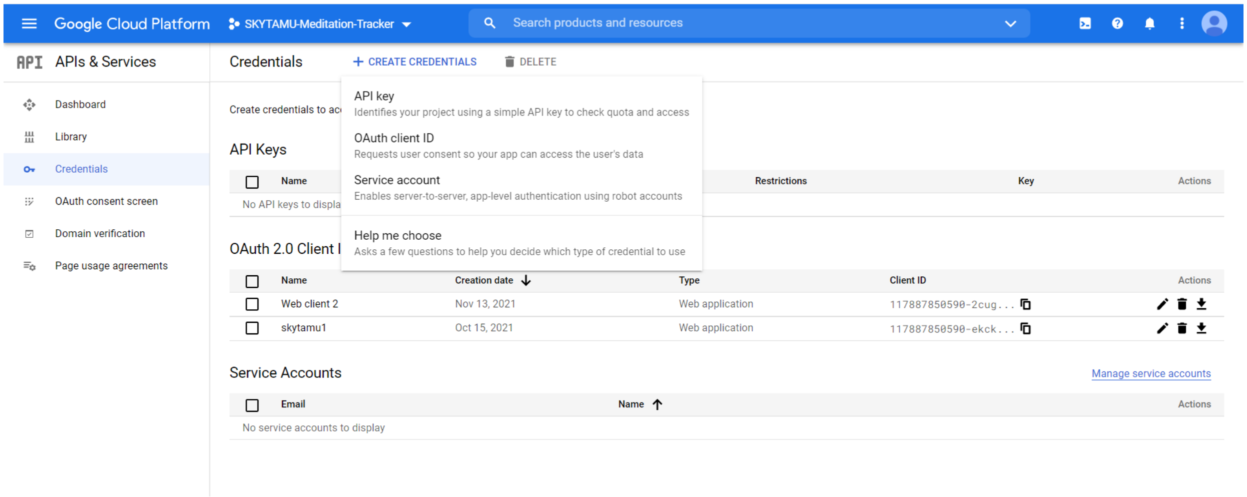
Google API

What is Google API and why is it significant?

Google API is a service the SKY@TAMU Website 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***'



Identifies your project using a simple API key to check quota and access

Requests user consent so your app can access the user's data

Enables server-to-server, app-level authentication using robot accounts

Asks a few questions to help you decide which type of credential to use

Create credentials to ac

API Keys

□	Name	Restrictions	Key	Actions
	No API keys to displa			

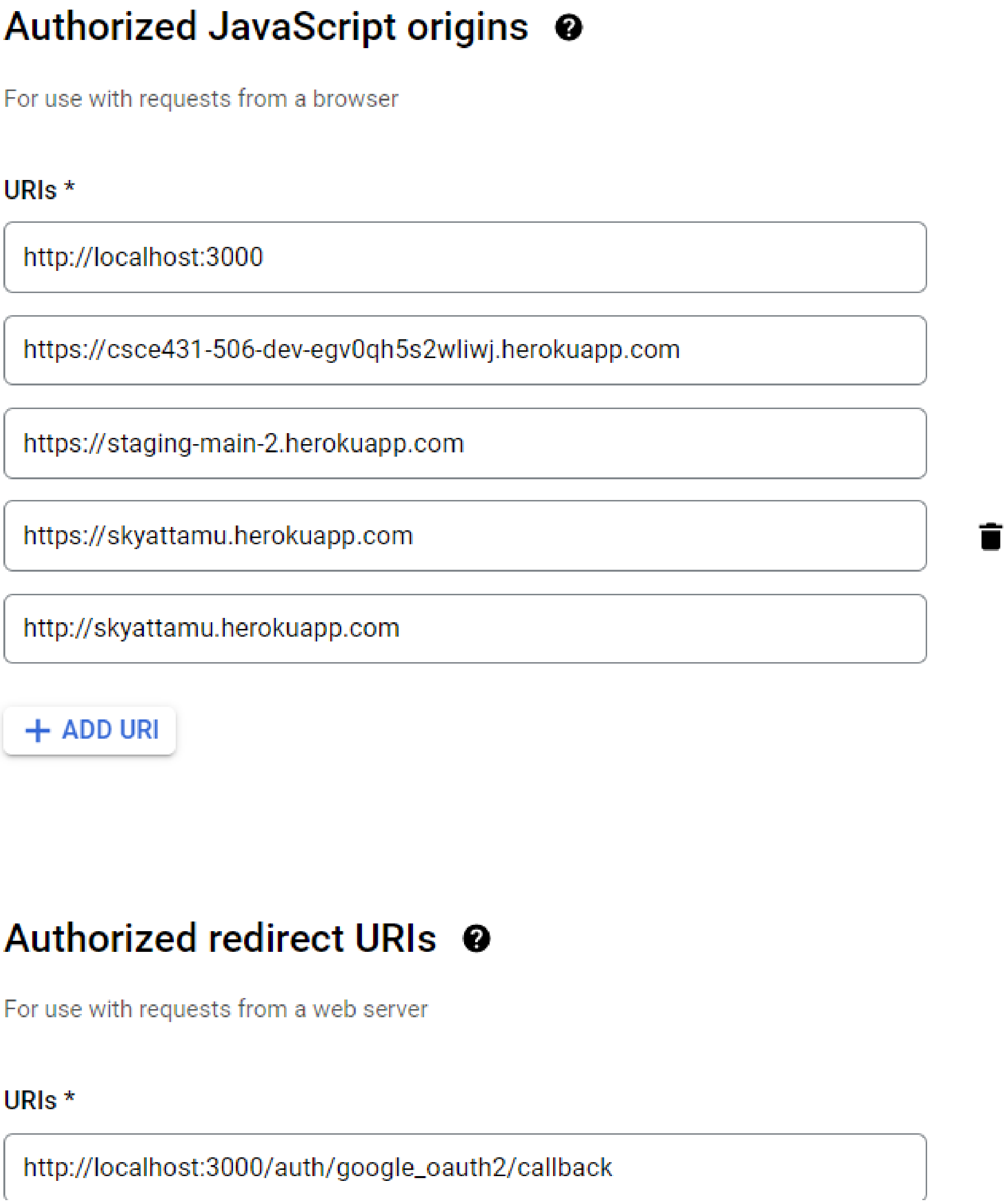
OAuth 2.0 Client I

□	Name	Creation date ↓	Type	Client ID	Actions
□	Web client 2	Nov 13, 2021	Web application	117887850590-2cug...	
□	skytamu1	Oct 15, 2021	Web application	117887850590-ekck...	

Service Accounts

□	Email	Name ↑	Actions
	No service accounts to display		

    c. Under **"Application Type"**, choose Web Application
    d. Click on '***Add URI***' under **"Authorized redirect URIs"**



Authorized JavaScript origins

For use with requests from a browser

URIs *

http://localhost:3000

https://csce431-506-dev-egv0qh5s2wliwj.herokuapp.com

https://staging-main-2.herokuapp.com

https://skyattamu.herokuapp.com

http://skyattamu.herokuapp.com

Authorized redirect URIs

For use with requests from a web server

URIs *

http://localhost:3000/auth/google_oauth2/callback

e. Add the in the URL that will be called back to

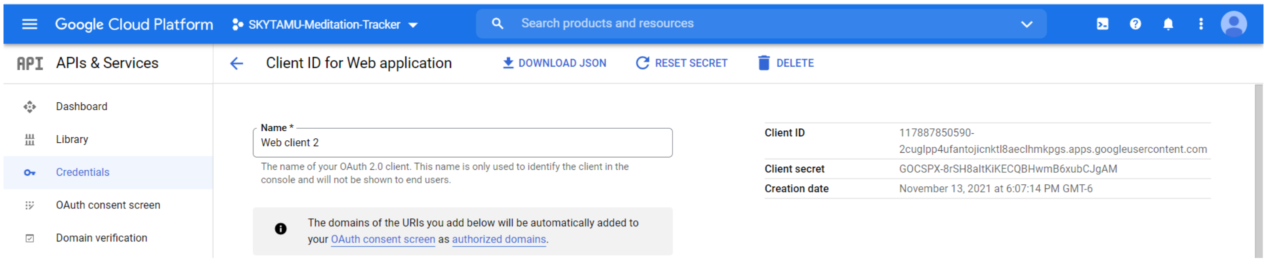
i. Should be: URL of application (ending with a '/') + 'admins/auth/google_oauth2/callback'

ii. Example:
https://skyattamu.herokuapp.com/admins/auth/google_oauth2/callback 1. This is the call back URI for the current SKY@TAMU website

2. 'https://skyattamu.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



Client ID for Web application

Name *
Web client 2

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.

Client ID	117887850590-2cuglpp4ufantojicnktl8aeclhmkpgs.apps.googleusercontent.com
Client secret	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
Creation date	November 13, 2021 at 6:07:14 PM GMT-6

You will need to make a small code change if you ever need to change the keys.