Table of Contents

1. Heroku

You will host your own dedicated server and database on Heroku.

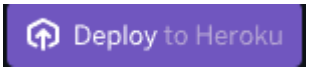
1. Goto Herku and signup for a free account: Heroku | Sign up
    a. Students (enrolled in college, middle school, or high school)
        i. Make your GitHub account a student account (free) if it is not one already: https://education.github.com/discount_requests/pack_application
        ii. Click "Get the student offer" button on Heroku's page and follow the directions: Heroku is for Students
2. Download Heroku CLI on your personal computer. The instructions for your respective OS can be found here: The Heroku CLI
    a. Technically, Heroku allows you to view logs and run commands from your Heroku Dashboard, but you are installing the CLI on your computer just in case you need to do more with your configuration

2. Server and Dashboard - Backend as a Service (BaaS)

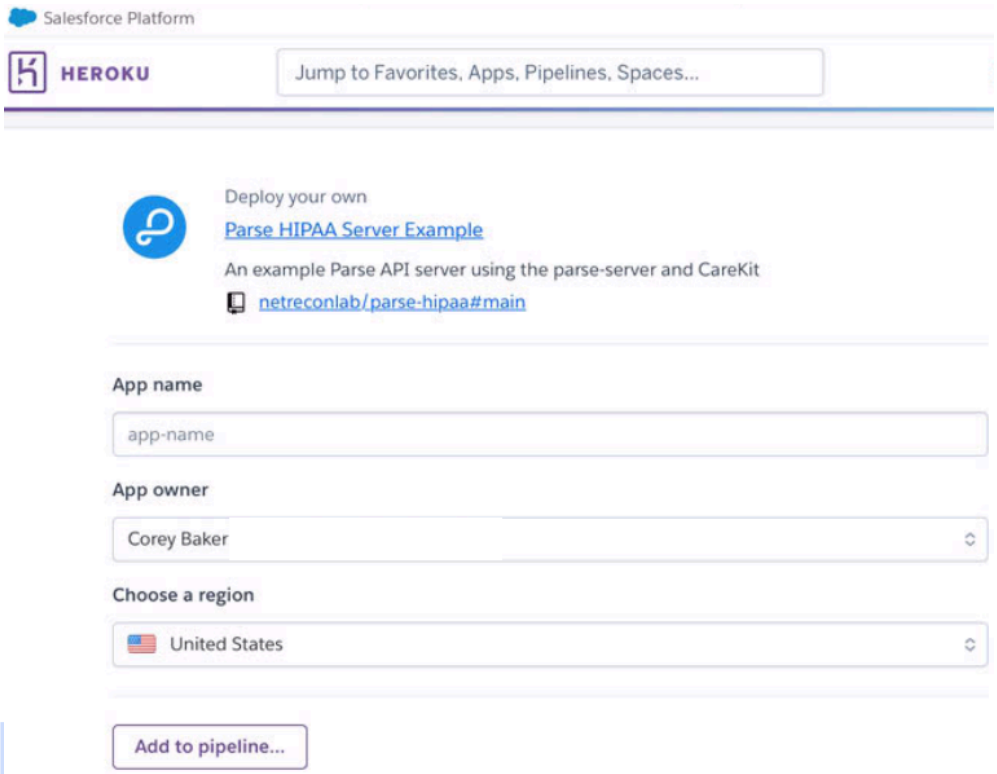
Your BaaS is based on nodejs, specifically it's a parse-server.

2.1 Deploying your own Parse-HIPAA Server on Heroku

1. Goto the parse-hipaa github repo and click the button (if this doesn't work in Safari, try another browser):



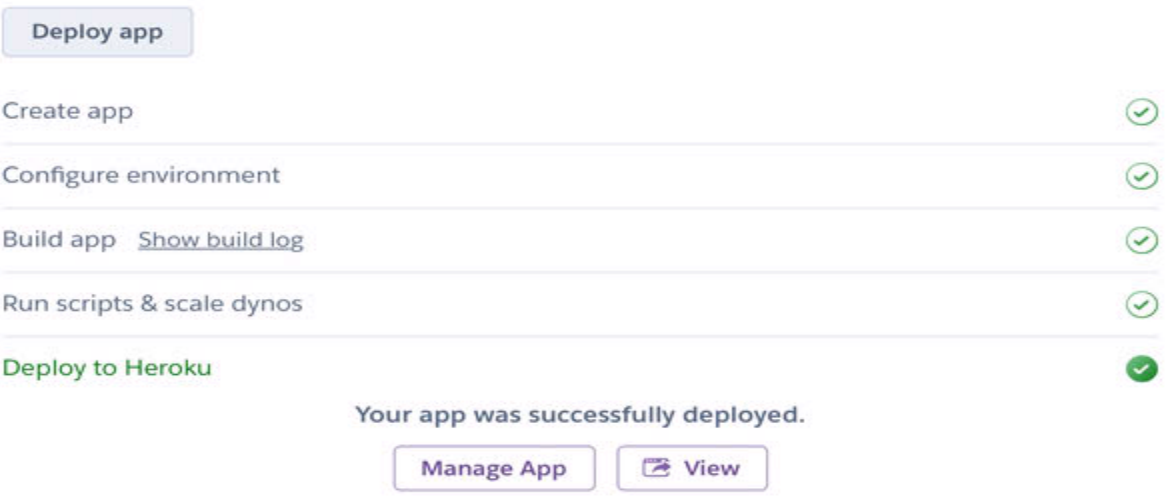
2. You should see a screen that looks like the following:



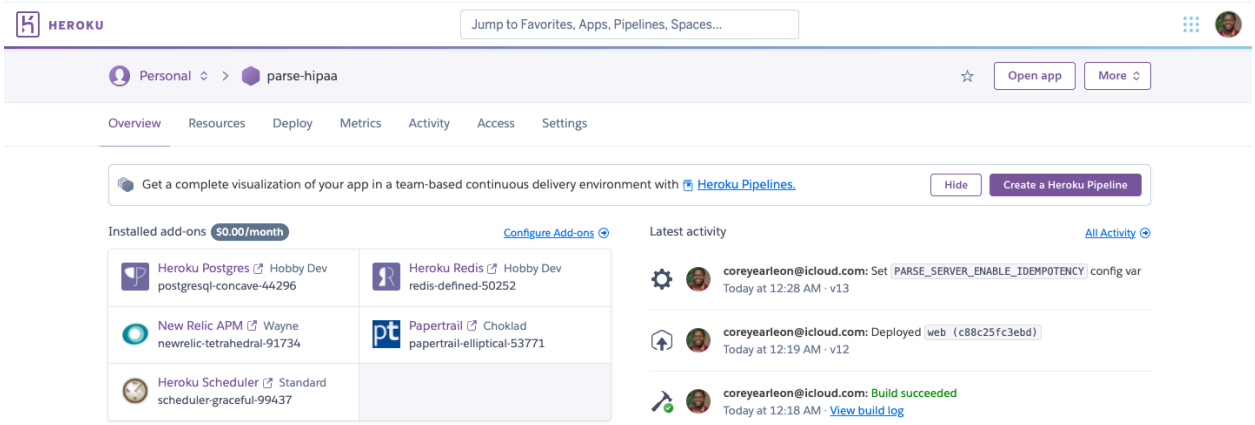
3. Create an ***App name*** of your choice. Note, this will be referred to as **YOUR_APP_NAME** for the rest of this document
4. Under the ***Config vars*** section, set **NEW_RELIC_APP_NAME** to reflect **YOUR_APP_NAME**
5. Now you will create a username and password to log into your dashboard. This should be something you plan to remember or save to your Keychain
    a. Set your username in **PARSE_DASHBOARD_USERNAMES**
    b. Set your password in **PARSE_DASHBOARD_USER_PASSWORDS**
    c. Note that you should be entering the hash of your password (see notes on the environment variable). Type your password into this site to get a hash

6. You should leave all other ***Config vars*** as they are… Scroll to the bottom of the page and click on ***Deploy app***

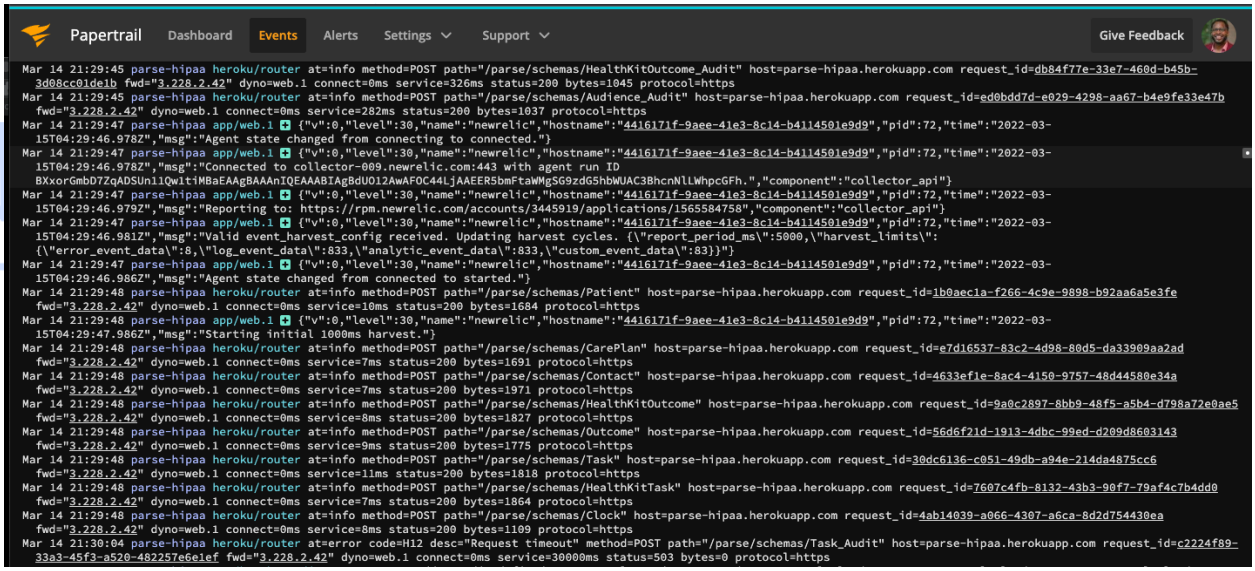
7. This will take a couple of minutes to deploy, but when it's finished, you should see all green checkmarks with the message, "Your app was successfully deployed". Click ***Manage App***



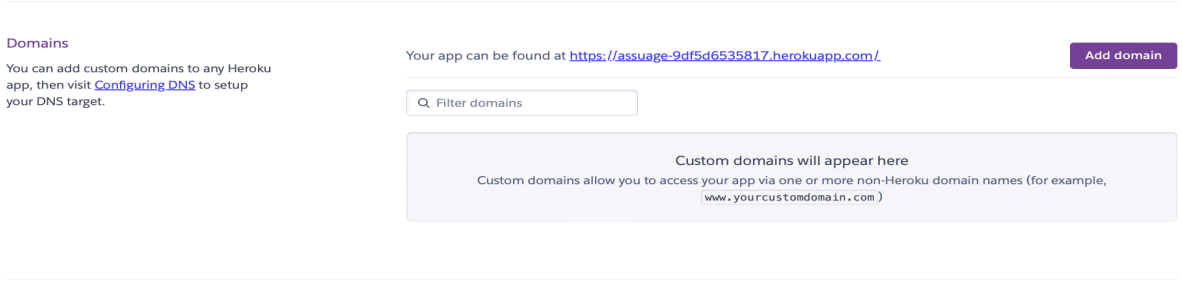
8. You should now be on your apps main screen:



9. Click on **Papertrail** under *Installed add-ons*. Accept the user agreement and you should now be able to view your logs

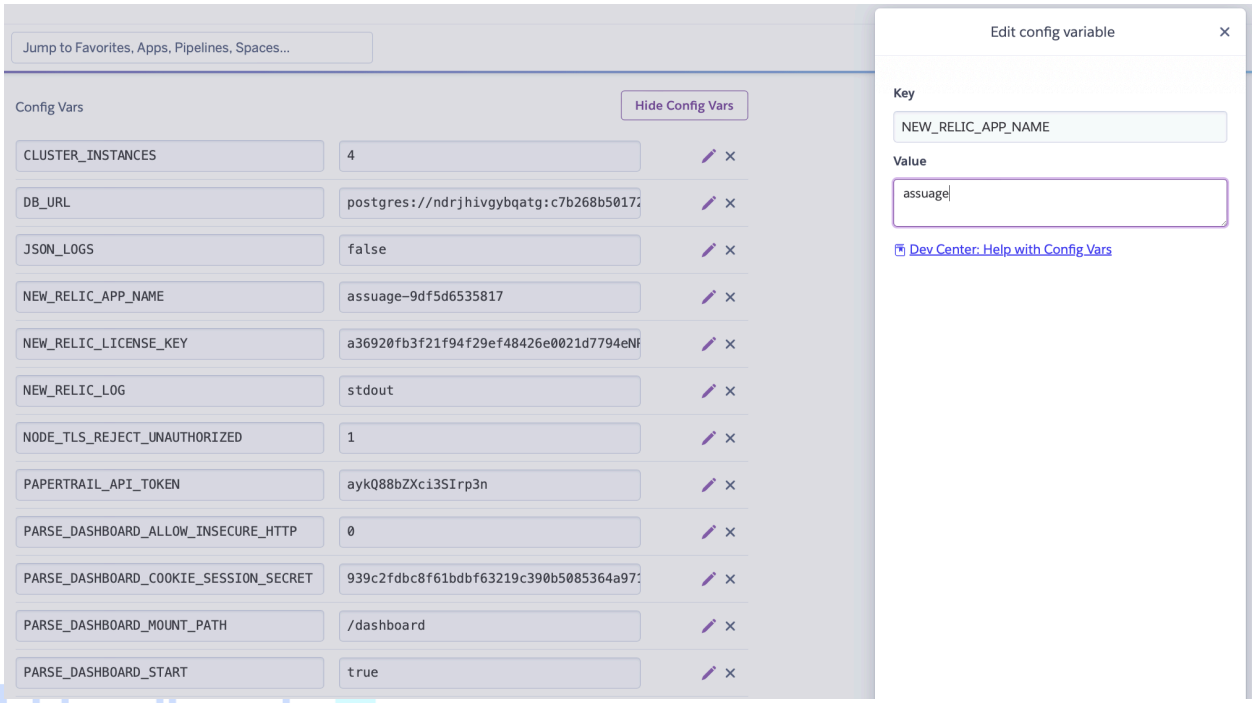


10. Go back to your Heroku Dashboard and goto ***Settings***. Scan the page for the **Domains** section and you should see the link to access your parse-hipaa server:

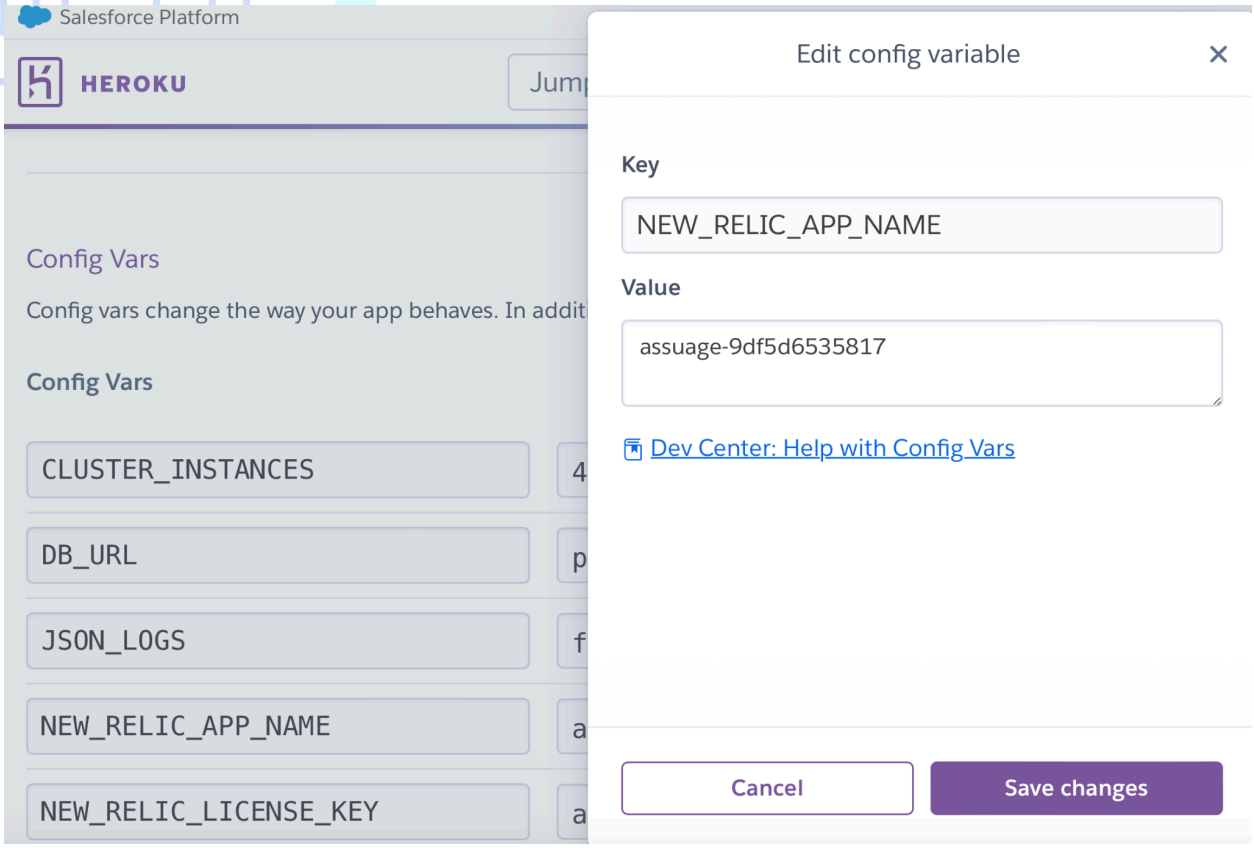


11. Copy the name of your app and the numbers & letters following it. For example, from the image above, I would copy “assuage-9df5d6535817”.
12. Then go up to NEW_RELIC_APP_NAME in Config Vars. Click the pencil icon to edit this variable. Currently, the value is set to “assuage”.The screen should look like the

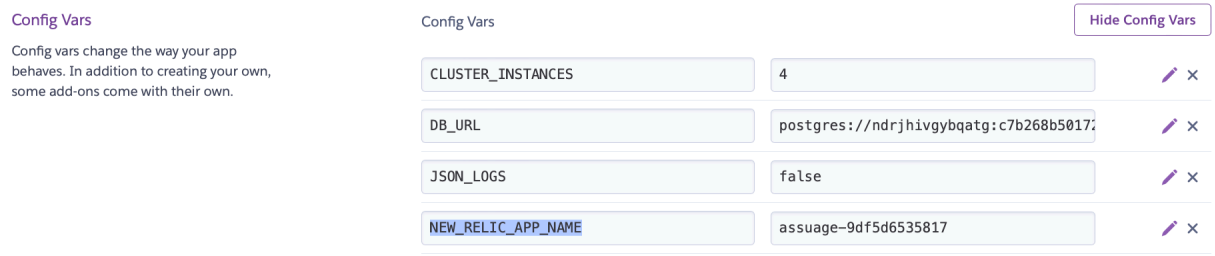
following:



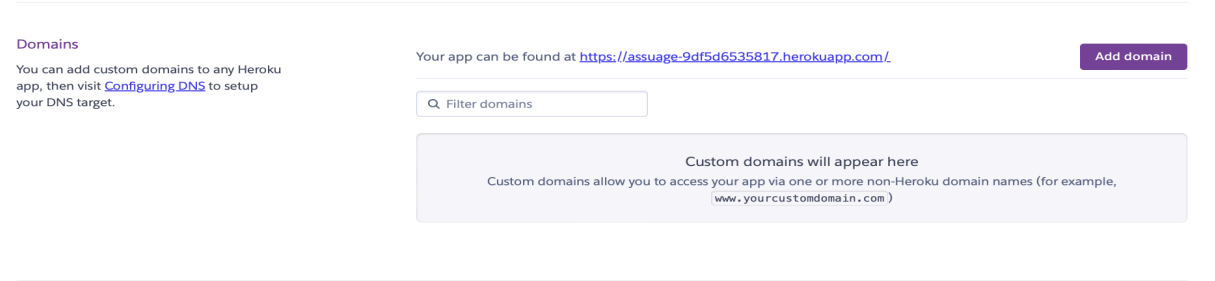
13. Then paste into the value box. In this example, the screen will look like the following:



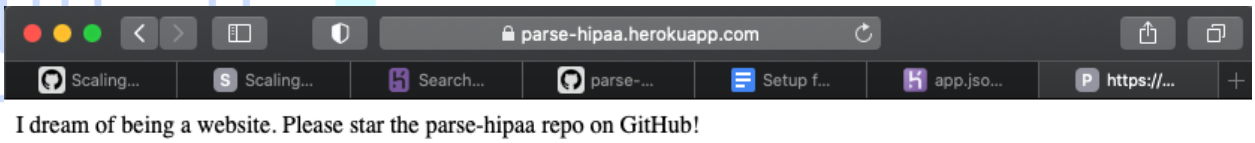
14. Then select save changes. In this case, the NEW_RELIC_APP_NAME will now look like this:



15. Go down to the **Domains** section again and you should see the link to access your parse-hipaa server:

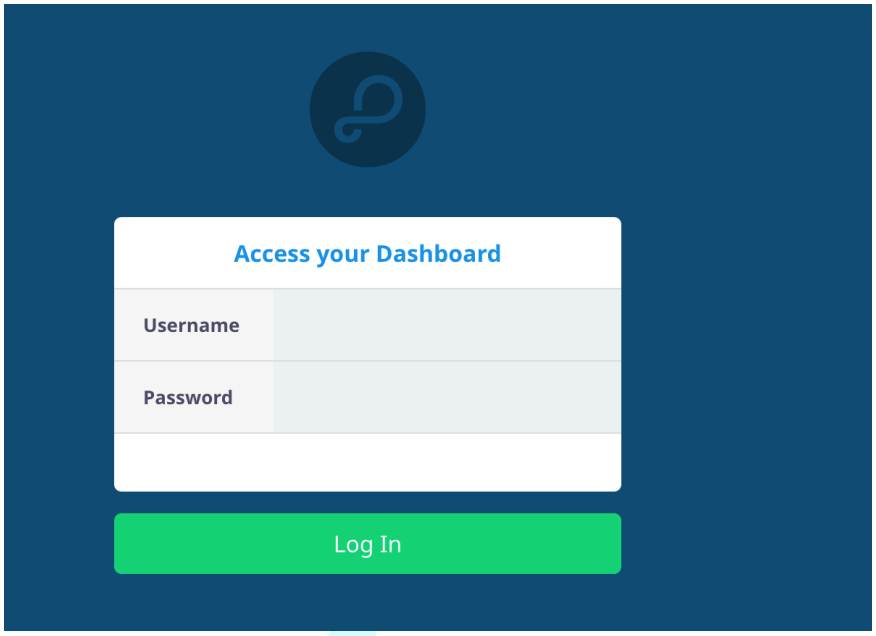


16. Click the link and if your server is working you will see the following:

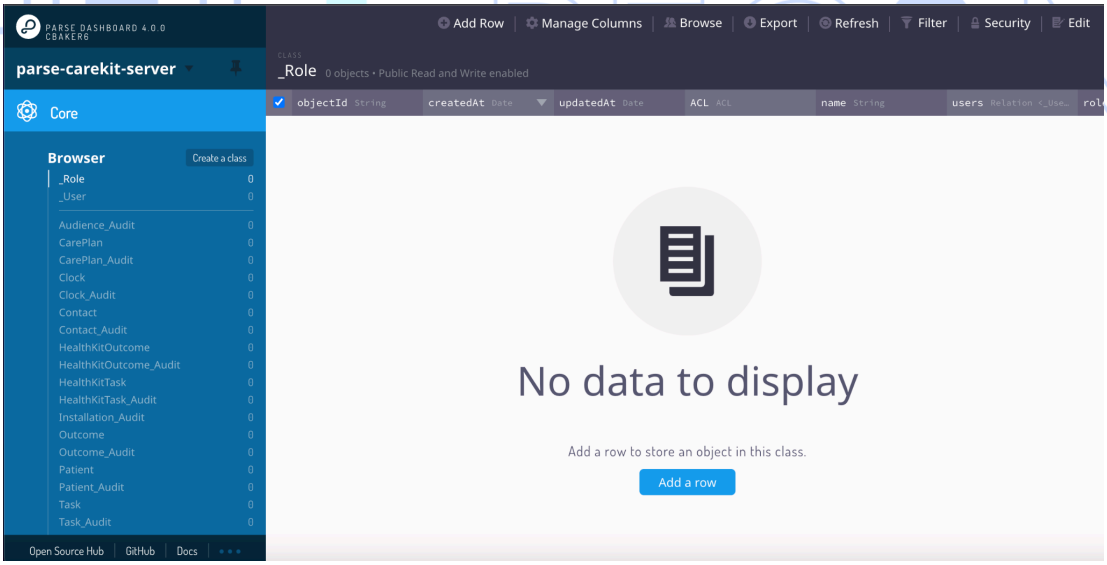


17. Edit the URL in your browser by appending “/dashboard”. Your new url should look like the follow, “https://YOUR_APP_NAME.herokuapp.com/dashboard”. If your dashboard is

working properly you will see the following:

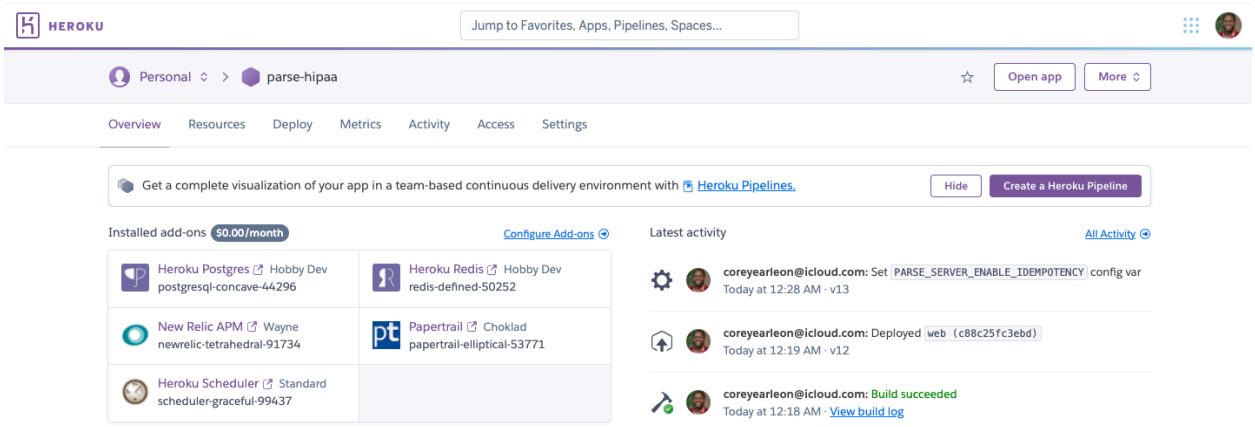


18. Use the values entered for **PARSE_DASHBOARD_USERNAME and the “original” (not the hash) of PARSE_DASHBOARD_USER_PASSWORD to login**
19. If your dashboard is correctly configured to connect to your server, you will see the following:

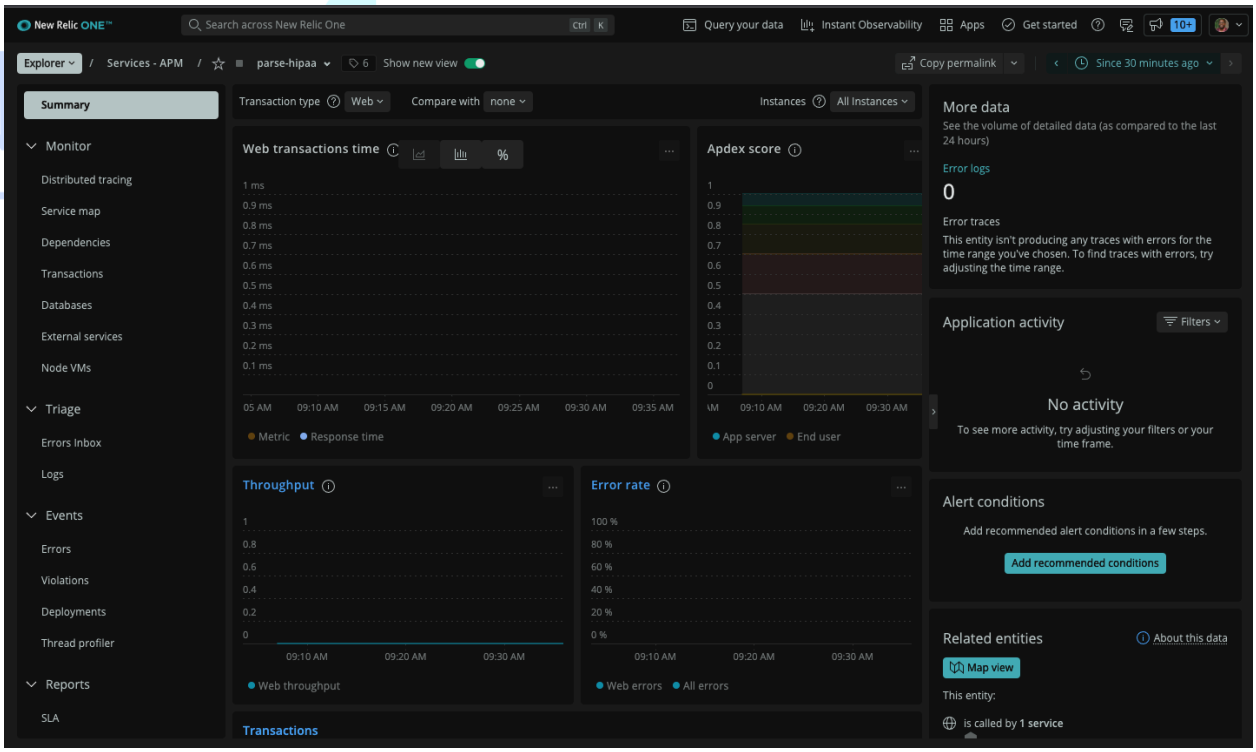


2.2 Monitor Your Servers Performance

1. Under *Installed add-ons*, you will see that your server comes with New Relic APM



2. You can click the New Relic APM link in the Heroku Dashboard and it will automatically create an account (will send you an email about your new account) along with taking you to your servers monitoring page:



3. You can use New Relic in the future to monitor the performance of your Parse HIPAA server

3. Database (PostgreSQL)

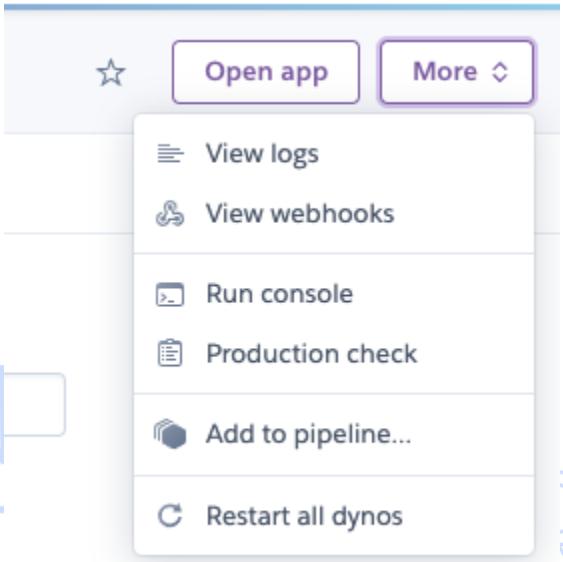
Your database was created during the aforementioned deployment process of your Heroku based parse-hipaa server. Your database is PostgreSQL 13+ and can be scaled when needed

using Heroku's services. The following will take you through the proper steps to configure your database using some one-time pre-installed scripts on your parse-hipaa server:

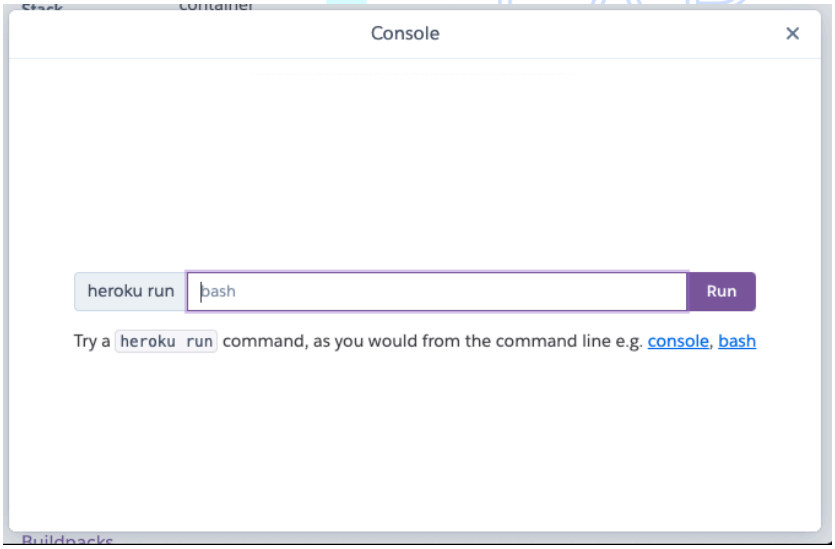
3.1 Indexing your Database (ParseCareKit based apps only)

To keep your database running efficiently, you should run a script to add a number of useful indexes to your database:

1. Back on dashboard.heroku.com, look on the top right corner for button **More** that's next to the *Open app* button. Click **More** and then click **Run console**



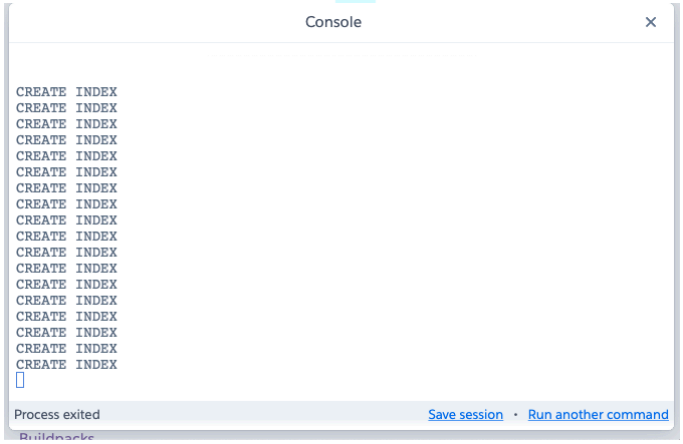
2. You should now see a screen at the bottom of your browser that is similar to a command terminal



3. Paste the following command in the *bash* box, "./scripts/setup-parse-index.sh" and then press the **Run** button

4. While the script is running, you will see the following

5. If the script ran correctly, you will see the following



3.2 Setting up Postgis Extensions for Geolocation Queries

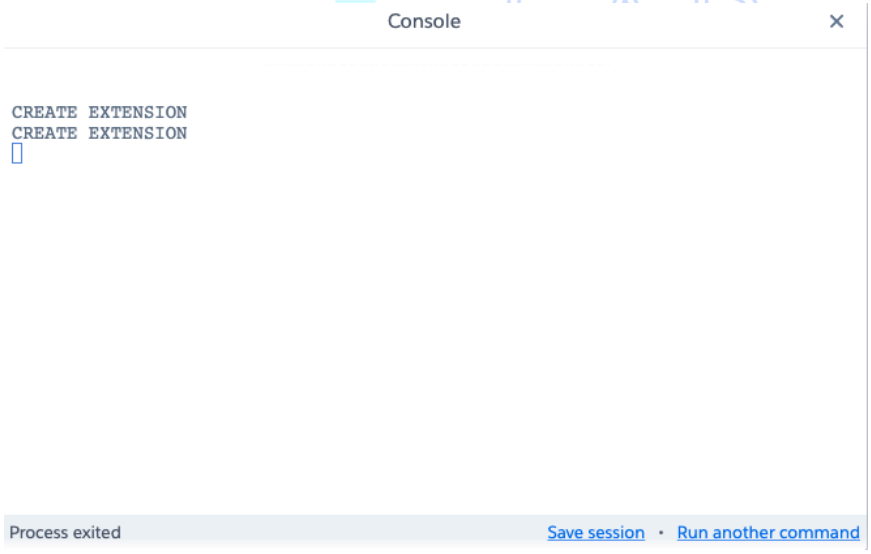
CS485G students and those running Heroku Free Services should **SKIP** this step as it adds 5k+ rows to your database and the free services only offers 10k rows

1. Assuming you still have the Console open from the previous step, click the **Run another command** link on the bottom right of the *Console* window

2. Paste the following command in the *bash* box, “./scripts/setup-dbs.sh” and then press the **Run** button

3. While the script is running, you will see the following

4. If the script ran correctly, you will see the following

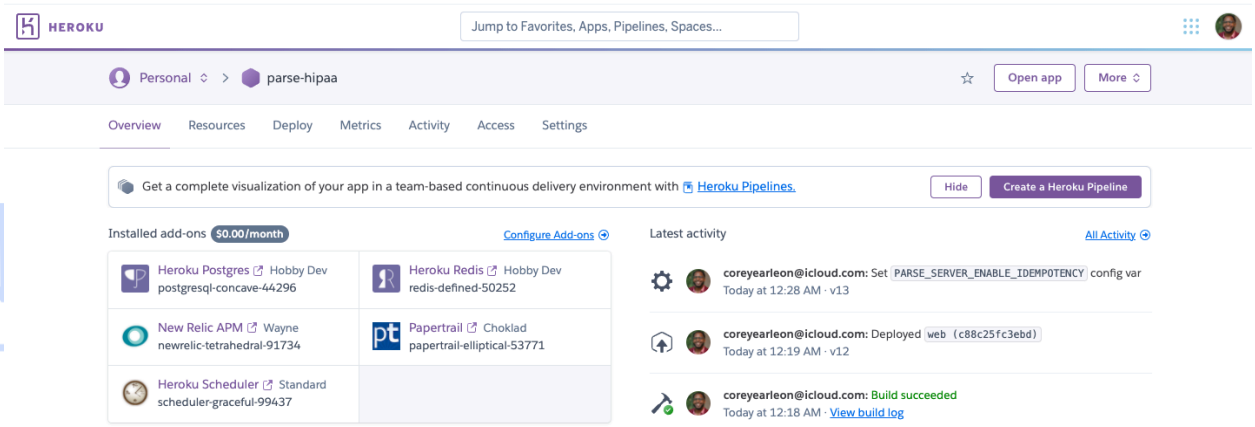


4. Setup Scheduler and Idempotency on Heroku

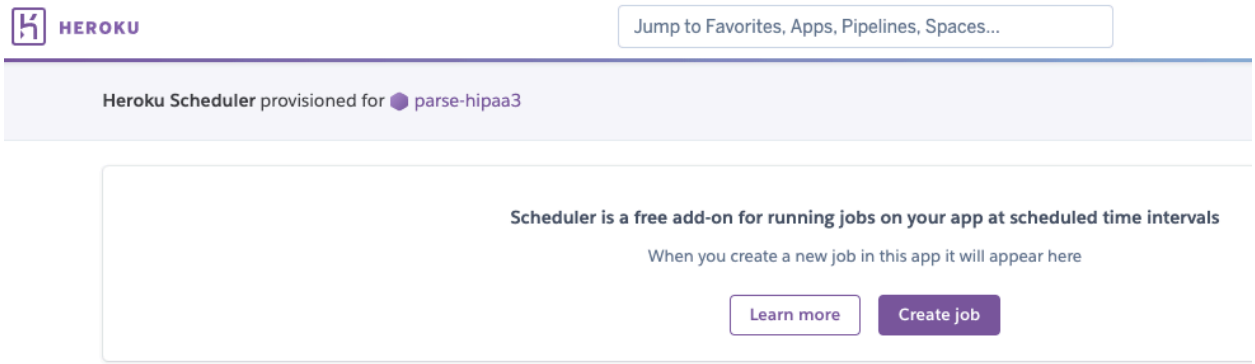
Idempotency is used to dedup identical requests made due to network intermittency. For example, if a user was being created for the first time over a bad network connection, it's possible for that user to be created twice when idempotency is not used. This is because the request to create the user may be sent/received twice due to bad network connectivity. Your server and apps are able to remedy the aforementioned issues, but you need to enable some additional configurations:

4.1 Heroku Scheduler

1. In your Heroku Dashboard of your app, go to *Overview->Installed add-ons* and click **Heroku Scheduler**

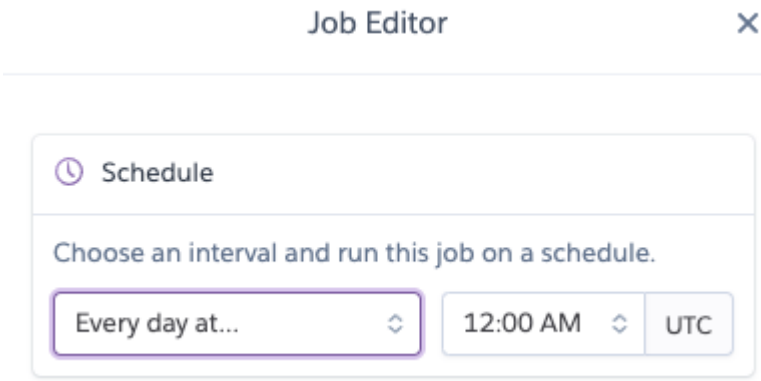


2. Then click the **Create job** button



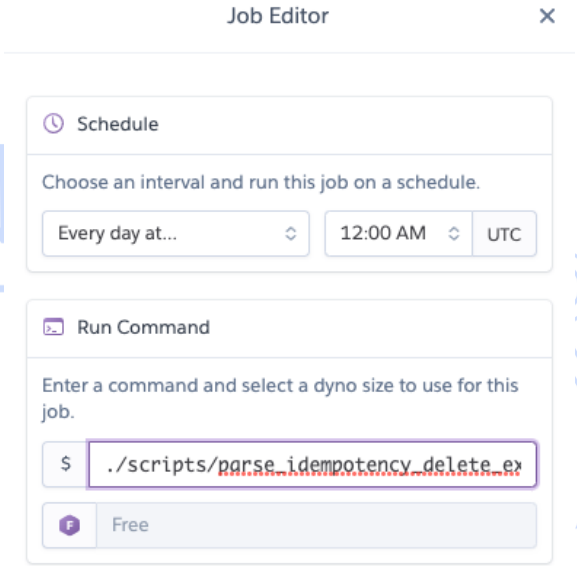
3. In the *Job Editor* window, under *Schedule* choose **Every day at…** in the drop down menu. You can leave the default time, or select a different time if you like.
    a. Scheduling "everyday" is okay while your app is in the *development* as you are using Heroku's free service and want to conserve your dyno minutes, but as your

app moves to *production* and app users increase you should move towards,

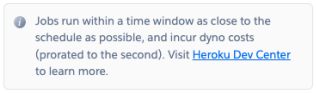


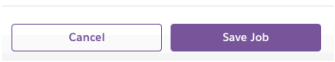
Every 10 minutes

4. Under *Run Command,* enter the following command, "./scripts/parse_idempotency_delete_expired_records.sh"

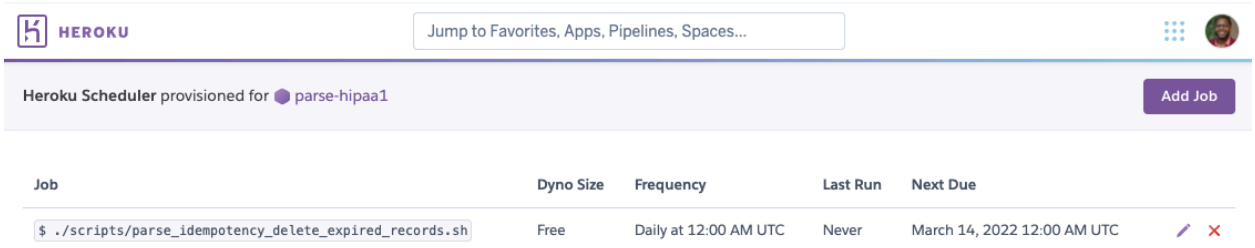


5. Then press the **Save Job** button at the bottom



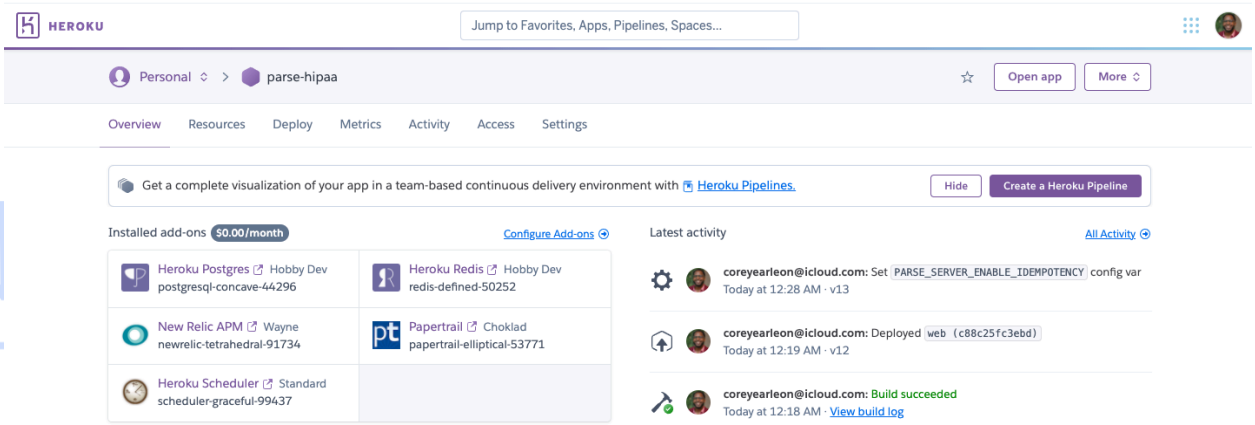


6. The new scheduled job should appear now and will run automatically at the scheduled time

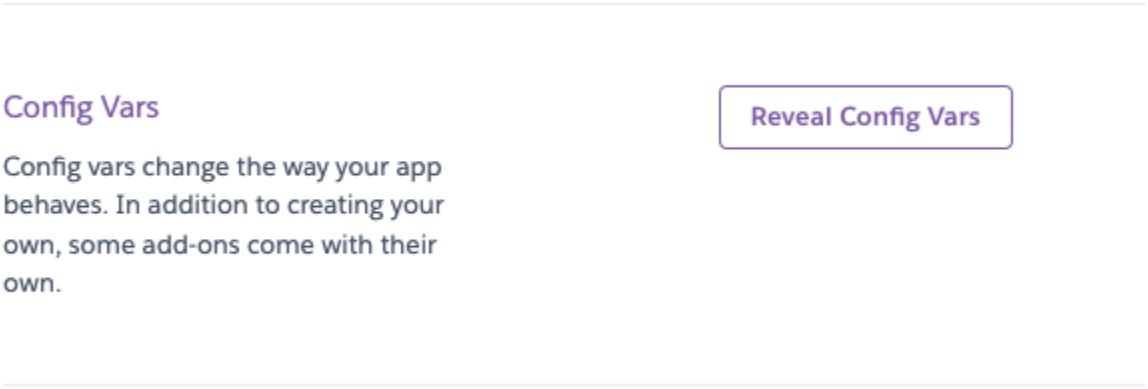


4.2 Enabling Idempotency on parse-hipaa

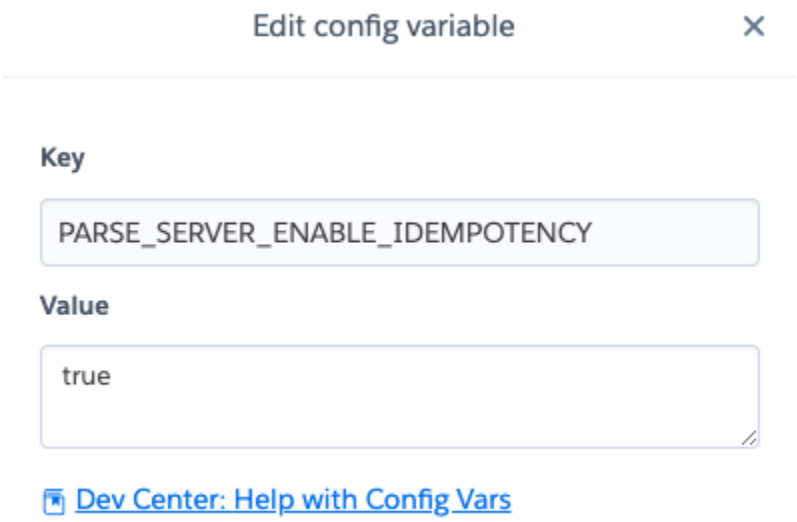
1. Go back to the previous window of your Heroku Dashboard and click on **Settings**



2. Click on **Reveal Config Vars**



3. Look for the config var, “PARSE_SERVER_ENABLE_IDEMPOTENCY”, click edit and change it’s string value from *false to true*



4. Give your server a couple minutes to restart and check that you can access your server by visiting “https://YOUR_APP_NAME.herokuapp.com/dashboard”

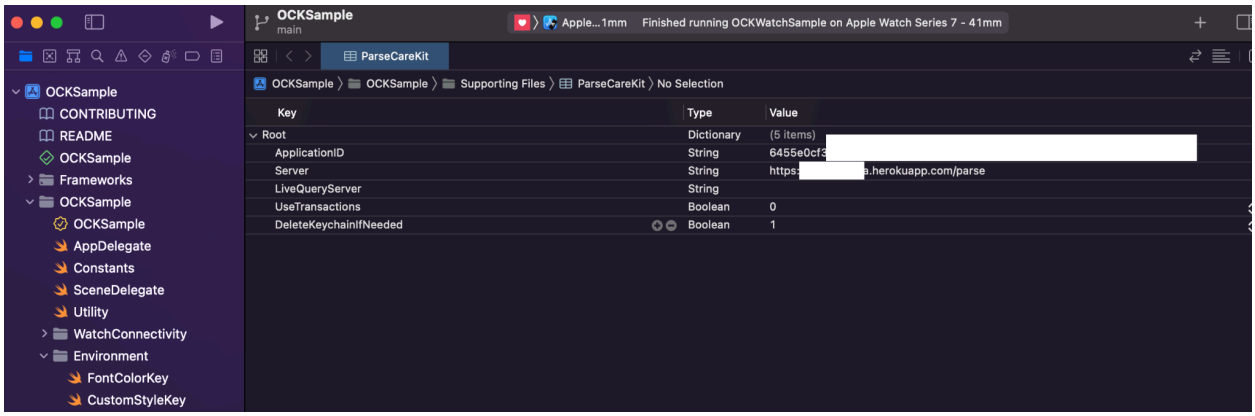
5. Client Applications

Now that your parse-hipaa server is configured on Heroku, you need to edit your client application with the values of your new **PARSE_SERVER_APPLICATION_ID** and “https://YOUR_APP_NAME.herokuapp.com/parse”. Depending on what your application is based on, follow the steps below.

5.1 CareKitSample-ParseCareKit Based Apps (CS 485G apps)

1. If you are building on your personal mac, be sure to install swiftlint using brew. Directions can be found here: https://github.com/realm/SwiftLint#using-homebrew
    a. Be sure to follow any warnings or suggestions in the terminal about adding lines to your environment if needed after running: *brew install swiftlint*
    b. Students using the macs on campus should already have swiftlint installed and can proceed to the next step

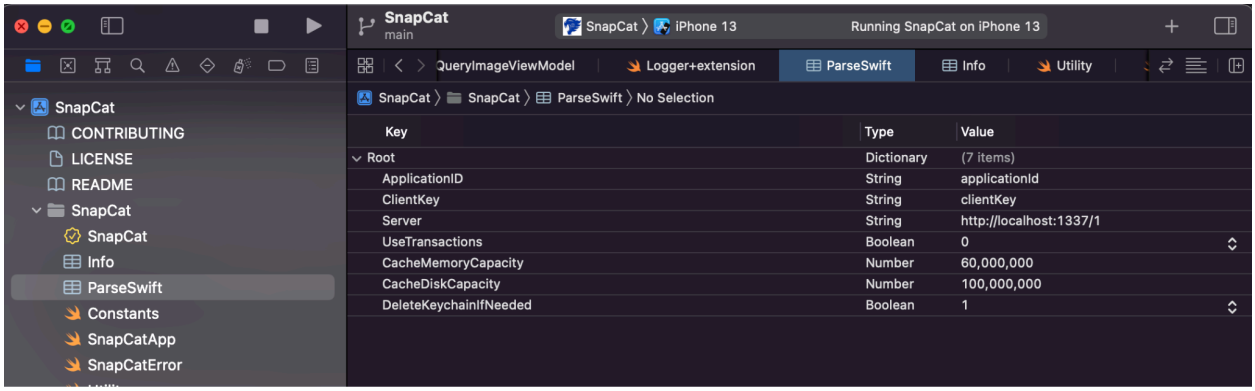
2. Open your project in Xcode and goto, OCKSample->OCKSample->Supporting Files->ParseCareKit. You should see the following:



3. In the row for ***ApplicationID***, put the value of your **PARSE_SERVER_APPLICATION_ID** in the **Value** column
4. In the row for ***Server***, put the value of your "https://YOUR_APP_NAME.herokuapp.com/parse" in the **Value** column
5. You can now build and run your application and you should be able to login and save data. You can view all saved data via the Parse HIPAA Dashboard you created in the previous steps
6. You're finished with the setup, you **DON'T** need to do any other steps, ***HAPPY CODING!!!***

5.2 SnapCat Based Apps

1. If you are building on your personal mac, be sure to install swiftlint using brew. Directions can be found here: GitHub - realm/SwiftLint: A tool to enforce Swift style and conventions.
    a. Students using the macs on campus should already have swiftlint installed and can proceed to the next step
2. Open your project in Xcode and goto, SnapCat->SnapCat->ParseSwift. You should see the following:



3. In the row for *ApplicationID*, put the value of your **PARSE_SERVER_APPLICATION_ID** in the **Value** column

4. In the row for *Server,* put the value of your "https://YOUR_APP_NAME.herokuapp.com/parse" in the **Value** column
5. You can now build and run your application and you should be able to login and save data. You can view all saved data via the Parse HIPAA Dashboard you created in the previous steps
6. You're finished with the setup, you **DON'T** need to do any other steps, ***HAPPY CODING!!!***

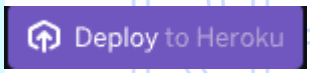
5.3 ParseSwift Based Apps that are not ParseCareKit or SnapCatBased, along with other Parse SDK's

Depending on your SDK, you need to edit your client SDK with the values of your new **PARSE_SERVER_APPLICATION_ID** and "https://YOUR_APP_NAME.herokuapp.com/parse".

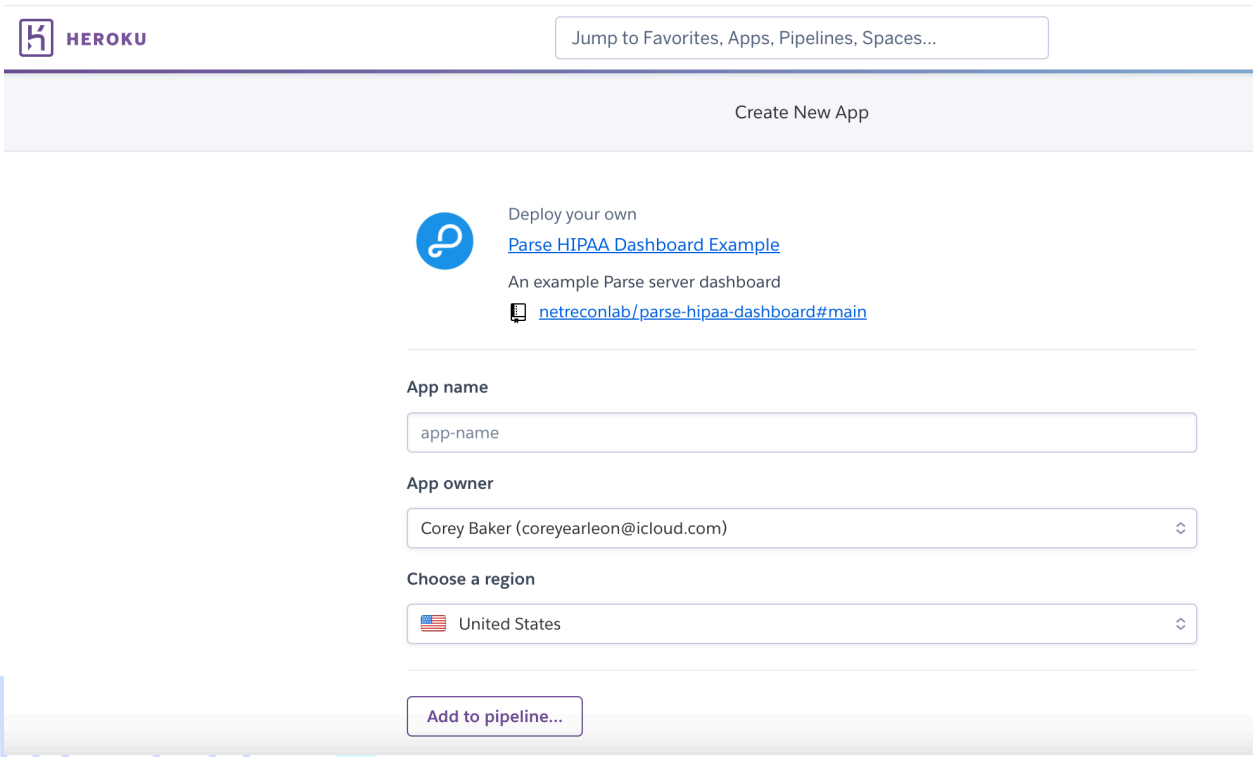
6. Standalone Dashboard

This is only for the standalone parse-hipaa-dashboard. If you configured a parse-hipaa-server in *2. Server - Backend as a Server*, you should skip this section.

Your BaaS is based on nodejs, specifically parse-dashboard.

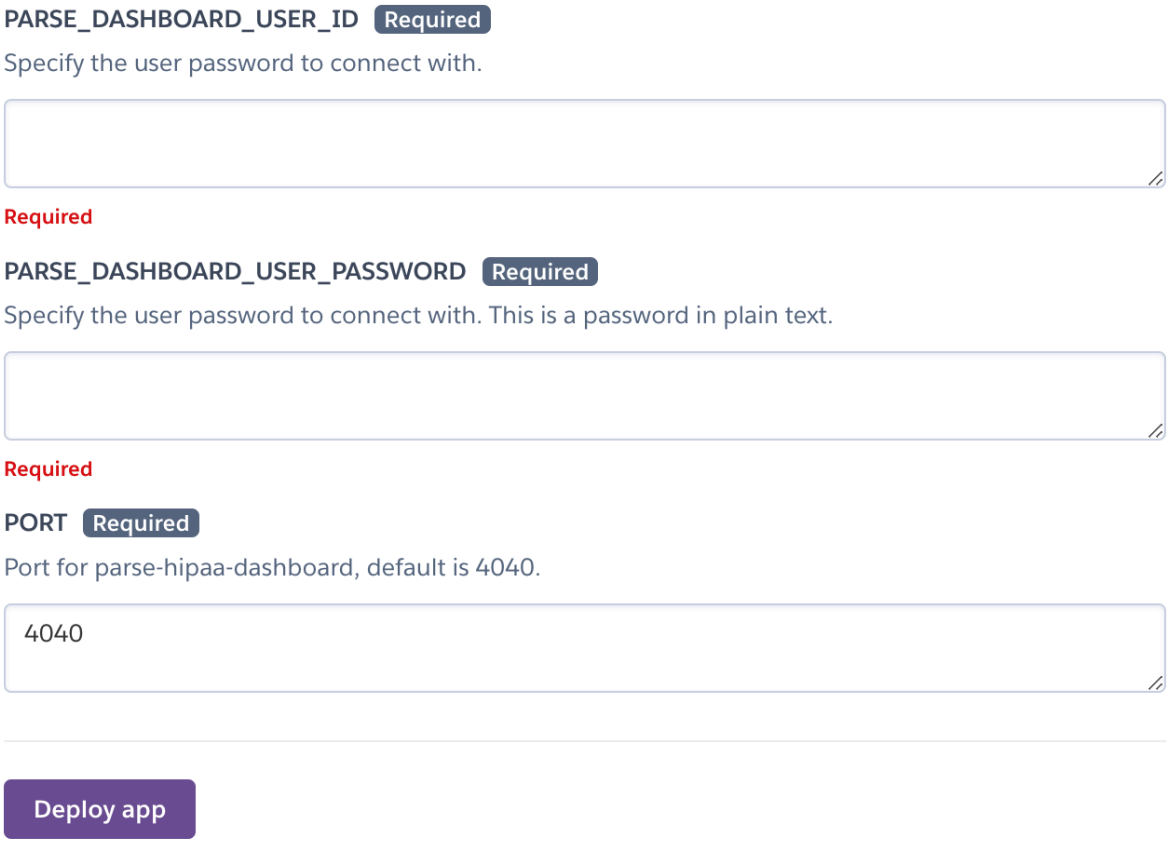
1. Goto the parse-hipaa-dashboard GitHub repo and click the button (if this doesn't work in Safari, try another browser): 

2. You should see a screen that looks like the following:

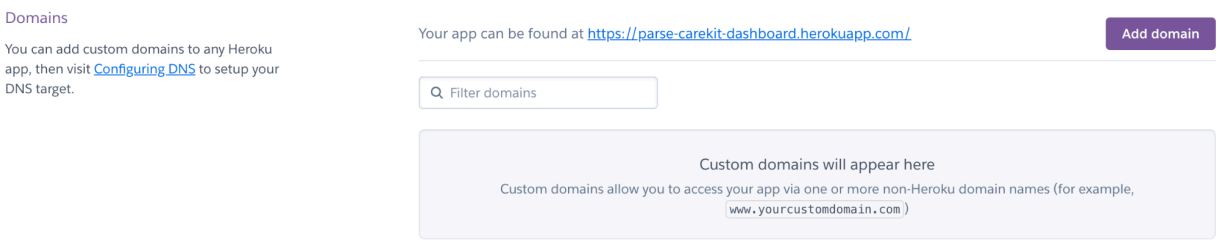


3. Use YOUR_APP_NAME from your server setup, but append it with “dashboard. It should look like, “YOUR_APP_NAME-dasboard”
4. Next, you will need to link your new Parse HIPAA Dashboard to your server. You will accomplish this by making some of the environment config vars in your Parse HIPAAA Dashboard match your server. Under the ***Config vars*** section, set the following (left side is your config var for your Parse HIPAA Dashboard, the right side is the VALUE for the respective config var on your server):
    a. **PARSE_DASHBOARD_APP_ID = PARSE_SERVER_APPLICATION_ID**
    b. **PARSE_DASHBOARD_APP_NAME = YOUR_APP_NAME**
    c. **PARSE_DASHBOARD_PRIMARY_KEY = PARSE_SERVER_PRIMARY_KEY**
    d. **PARSE_DASHBOARD_SERVER_URL = PARSE_SERVER_URL**
    e. **PARSE_DASHBOARD_GRAPHQL_SERVER_URL = (read below)**
        i. **This is the same as PARSE_SERVER_URL, accept you should delete “/parse” at the end of the string and add “/graphql”**
        ii. **Your graphql url should look something like: https://yourappname.herokuapp.com/graphql**
5. Now you will create a username and password to log into your dashboard. This should be something you plan to remember or save to your Keychain
    a. Set your username in **PARSE_DASHBOARD_USERNAME**
    b. Set your password in **PARSE_DASHBOARD_USER_PASSWORD**
        i. Note that you should be entering the hash of your password (see notes on the environment variable). Type your password into this site to get a hash

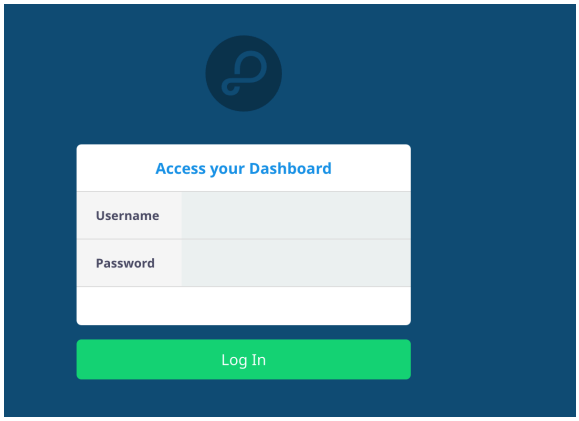
6. Scroll and tap ***Deploy app***



7. When finished, click ***Manage App***. Go to ***Settings***. Scan the page for ***Domains*** and you should see the link to access your Dashboard:



8. Click the link and if your dashboard is working you will see the following:



9. Use the values entered for **PARSE_DASHBOARD_USERNAME and the "original" (not the hash) of PARSE_DASHBOARD_USER_PASSWORD to login**
10. If your dashboard is correctly configured to connect to your server, you will see the following: