These instructions are for Windows 7 and above, so follow the instructions for creating a Heroku App in README\_heroku.md, and use these instructions to supplement and help resolve issues. In general, login into the heroku CLI with the Windows cmd.exe prompt, and then use whatever terminal emulator you prefer.

Create a GitHub repository for your app before or wait until confirming the application a name during the Heroku create process. After the GitHub repository is created run the following command from Git-Bash.

git clone *https://github.com/chrisg4github/Project-GW.git*

**Heroku Commands and processes:**

C:\Users\Chrisg\BootCampClasswork\Project-GW>heroku login

Enter your Heroku credentials:

Email: chrisguthrie1600@comcast.net

Password: \*\*\*\*\*\*\*\*\*\*\*\*

Logged in as chrisguthrie1600@comcast.net

C:\cd Users\Chrisg\BootCampClasswork\Project-GW>

C:\Users\Chrisg\BootCampClasswork\Project-GW>heroku create

Creating app... done, nameless-sea-62811

https://nameless-sea-62811.herokuapp.com/ | https://git.heroku.com/nameless-sea-

62811.git

C:\Users\Chrisg\BootCampClasswork\Project-GW>heroku rename Project-GW

Renaming nameless-sea-62811 to Project-GW... !

! Name must start with a letter and can only contain lowercase letters,

! numbers, and dashes.

C:\Users\Chrisg\BootCampClasswork\Project-GW>heroku rename project-gw

Renaming nameless-sea-62811 to project-gw... done

https://project-gw.herokuapp.com/ | https://git.heroku.com/project-gw.git

Git remote heroku updated

! Don't forget to update git remotes for all other local checkouts of the

! app.

C:\Users\Chrisg\BootCampClasswork\Project-GW>git remote -v

heroku https://git.heroku.com/project-gw.git (fetch)

heroku https://git.heroku.com/project-gw.git (push)

origin https://github.com/chrisg4github/Project-GW.git (fetch)

origin https://github.com/chrisg4github/Project-GW.git (push)

# Create the postgres database hobby-dev is a free database level – this is an extremely size limited (in terms of record count and bytes) database.

C:\Users\Chrisg\BootCampClasswork\Project-GW>heroku addons:create heroku-postgresql:hobby-dev

Creating heroku-postgresql:hobby-dev on project-gw... free

Database has been created and is available

! This database is empty. If upgrading, you can transfer

! data from another database with pg:copy

Created postgresql-animated-62565 as DATABASE\_URL

Use heroku addons:docs heroku-postgresql to view documentation

# List any heroku addons you have set up

C:\Users\Chrisg\BootCampClasswork\Project-GW>heroku addons

Add-on Plan Price State

--------------------------------------------- --------- ----- -------

heroku-postgresql (postgresql-animated-62565) hobby-dev free created

+- as DATABASE

The table above shows add-ons and the attachments to the current app (project-gw

) or other apps.

# List information about the database set up

C:\Users\Chrisg\BootCampClasswork\Project-GW>heroku pg:info

=== DATABASE\_URL

Plan: Hobby-dev

Status: Available

Connections: 0/20

PG Version: 9.6.5

Created: 2017-11-14 18:52 UTC

Data Size: 7.1 MB

Tables: 0

Rows: 0/10000 (In compliance)

Fork/Follow: Unsupported

Rollback: Unsupported

Continuous Protection: Off

Add-on: postgresql-animated-62565

# Check the contents of the DATABASE\_URL

C:\Users\Chrisg\BootCampClasswork\Project-GW>heroku config

=== project-gw Config Vars

DATABASE\_URL: postgres://zveqkcpjroxbce:44a0357267af92609c0117ccd9c262d8746e69c5

980140d843cdfef853069cb8@ec2-50-17-253-3.compute-1.amazonaws.com:5432/d7b3o1itpode38

Exit heroku.

Log into the local database:

# Local verification of the database gw\_db

Chrisg@Chrisg-PC MINGW64 ~/bootcampclasswork/Project-GW (master)

$ psql -d gw\_db -U postgres

Password for user postgres:

psql (9.6.4)

WARNING: Console code page (437) differs from Windows code page (1252)

8-bit characters might not work correctly. See psql reference

page "Notes for Windows users" for details.

Type "help" for help.

gw\_db=# \dt

*Put list of tables here.*

gw\_db=# \q (to quit psql)

**Push local postgres to heroku postgres:** *(push commands didn't work)*

**# Log into heroku to push the local database to heroku database**

C:\Users\Chrisg\BootCampClasswork\Project-GW>heroku login

Enter your Heroku credentials:

Email: chrisguthrie1600@comcast.net

Password: \*\*\*\*\*\*\*\*\*\*\*\*

Logged in as chrisguthrie1600@comcast.net

C:\Users\Chrisg\BootCampClasswork\Project-GW>heroku pg:push gw\_db DATABASE\_URL --app project-gw

heroku-cli: Pushing gw\_db ---> postgresql-animated-62565

'env' is not recognized as an internal or external command,

operable program or batch file.

**Command to push database to heroku with user id and password:**

PGUSER=postgres PGPASSWORD='password' heroku pg:push gw\_db DATABASE\_URL --app project-gw

**# View heroku postgres log**

heroku logs -p postgres -t

**# Failure output of pg:push attempts:**

$ heroku pg:push gw\_db DATABASE\_URL --app project-gw

heroku-cli: Pushing gw\_db ---> postgresql-animated-62565

pg\_restore: [archiver] input file is too short (read 0, expected 5)

$ PGUSER=postgres PGPASSWORD='password' heroku pg:push gw\_db DATABASE\_URL --app project-gw

heroku-cli: Pushing gw\_db ---> postgresql-animated-62565

pg\_dump: [archiver (db)] connection to database "gw\_db" failed: FATAL: password authentication failed for user "postgres"

pg\_restore: [archiver] input file is too short (read 0, expected 5)

**Use one of these dump commands from a terminal emulator or create a dump file from pgAdmin:**

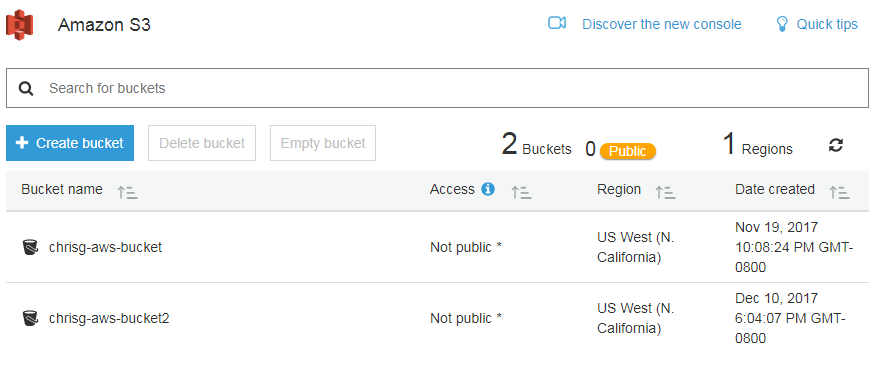
**Examples of pg\_dump commands from terminal emulator:**

1. PGPASSWORD='mypassword' pg\_dump -Fc --no-acl --no-owner -U postgres gw\_db > gw\_db\_12102017.dump
2. pg\_dump -Fc --no-owner -U postgres gw\_db > gw\_db\_12102017.dump
3. pg\_dump -Fc gw\_db > gw\_db\_12102017.dump

**Examples of pg\_restore from terminal emulator:**

1. pg\_restore --verbose --clean --no-acl --no-owner -U postgres -d gw\_db gw\_db\_1142017.dump
2. heroku pg:backups:restore 'https://s3.amazonaws.com/me/items/3H0q/mydb.dump' DATABASE\_URL
3. pg\_restore -U <username> -d <dbname> -1 <filename>.dump
4. PGPASSWORD=<PASS> pg\_restore --verbose --clean --no-acl --no-owner -h <HOSTNAME> -U <USER> -d <DATABASE> -p <PORT><FILEPATH>

**After the dump file is created, create an AWS account and use it to store the database dump for use in the pg\_backup:restore process. To get the pg\_backup:restore to work, I had to make the dump file "Public", but since Amazon warns against this, I remove the "Public" access after the restore.**

****

**When I had issues with Heroku, I opened a ticket and they responded with a day or two.**

Ticket #530343 - Not able to move my local postgres database to Heroku.

**Heroku Support said to use the pg\_backup:restore process:**

heroku pg:backups:restore 'https://s3-us-west-1.amazonaws.com/chrisg-aws-bucket/gw\_db\_11142017.dump' DATABASE\_URL

**Postgres database table test Hawaii install -**

Chrisg@Chrisg-PC MINGW64 ~/bootcampclasswork/Project-GW (master)

$ heroku pg:backups:restore 'https://s3-us-west-1.amazonaws.com/chrisg-aws-bucket/gw\_db\_11142017.dump' DATABASE\_URL

! WARNING: Destructive Action

! This command will affect the app project-gw

! To proceed, type project-gw or re-run this command with --confirm

! project-gw

>project-gw

Starting restore of https://s3-us-west-1.amazonaws.com/chrisg-aws-bucket/gw\_db\_11142017.dump to postgresql-animated-62565... done

Use Ctrl-C at any time to stop monitoring progress; the backup will continue restoring.

Use heroku pg:backups to check progress.

Stop a running restore with heroku pg:backups:cancel.

Restoring... done

**Postgres GW database table install -**

$ heroku pg:backups:restore 'https://s3-us-west-1.amazonaws.com/chrisg-aws-bucket2/gw\_db\_12102017\_Terminal.dump' DATABASE\_URL

! WARNING: Destructive Action

! This command will affect the app project-gw

! To proceed, type project-gw or re-run this command with --confirm

! project-gw

> project-gw

Starting restore of https://s3-us-west-1.amazonaws.com/chrisg-aws-bucket2/gw\_db\_12102017\_Terminal.dump to postgresql-animated-62565... done

Use Ctrl-C at any time to stop monitoring progress; the backup will continue restoring.

Use heroku pg:backups to check progress.

Stop a running restore with heroku pg:backups:cancel.

Restoring... done

$ heroku pg:psql

--> Connecting to postgresql-animated-62565

\dt

List of relations

Schema | Name | Type | Owner

--------+---------------------------+-------+----------------

public | city\_country\_mthly\_temps | table | zveqkcpjroxbce

public | country\_mthly\_temps | table | zveqkcpjroxbce

public | country\_yearly\_co2 | table | zveqkcpjroxbce

public | country\_yearly\_temps | table | zveqkcpjroxbce

public | global\_mthly\_temps | table | zveqkcpjroxbce

public | state\_country\_mthly\_temps | table | zveqkcpjroxbce

(6 rows)

select count(\*) from global\_mthly\_temps;

count

-------

1992

(1 row)

\q

$

**Heroku postgres (too many rows for hobby-dev level):**

There are size and row restrictions for hobby-dev level, after which you have to pay, so I deleted most of the test data from the test Hawaii tables.

**Other Heroku postgres information about the database:**

**Get the heroku postgres credentials using the command:**

C:\Users\Chrisg\BootCampClasswork\Project-GW>heroku pg:credentials:url

Connection information for default credential.

**Connection info string:**

"dbname=d7b3o1itpode38 host=ec2-50-17-253-3.compute-1.amazonaws.com port=5432

user=zveqkcpjroxbce password=44a0357267af92609c0117ccd9c262d8746e69c5980140d843

cdfef853069cb8 sslmode=require"

**Connection URL *aka* DATABASE\_URL:**

postgres://zveqkcpjroxbce:44a0357267af92609c0117ccd9c262d8746e69c5980140d843c

dfef853069cb8@ec2-50-17-253-3.compute-1.amazonaws.com:5432/d7b3o1itpode38

**# Pushing local repo to GitHub repo and then to the Heroku repo:**

To deploy your app to Heroku, you typically use the git add, commit, push commands

to push the code from your local repository’s master branch to your heroku remote.

$ git add .

$ git commit -m "Move new files to GitHub"

$ git push -u origin master

**Then push the files to Heroku master:**

$ git push heroku master

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**# Receive this error message when logging into Heroku CLI, but I was able to log into the CLI and perform commands. To clean this up, uninstall and then reinstall Heroku CLI.**

C:\Users\Chrisg\BootCampClasswork\Project-GW>heroku login

! Error parsing plugin

! C:\Users\Chrisg\AppData\Local\heroku\client\node\_modules\heroku-pg

! Unexpected token ;

! Error parsing plugin

! C:\Users\Chrisg\AppData\Local\heroku\client\node\_modules\heroku-pipelines

! Error parsing

! C:\Users\Chrisg\AppData\Local\heroku\client\node\_modules\heroku-pipelines\

package.json:

! Unexpected token m in JSON at position 1

Enter your Heroku credentials:

Email:

**Once the application is installed and working this is the URL:**

**Heroku GW Visualization app:**

https://project-gw.herokuapp.com/

**Once the GW temperature and CO2 tables were loaded from the dump file, there were too many rows for the hobby-dev level. I deleted most of the rows.**

$ heroku pg:psql

--> Connecting to postgresql-animated-62565

delete from city\_country\_mthly\_temps

where m\_date < '2012-01-01';

DELETE 226155

delete from country\_mthly\_temps

where m\_date < '2012-01-01';

DELETE 536755

select count(\*) from country\_yearly\_temps where year < 2012;

count

-------

45117

(1 row)

delete from country\_yearly\_temps

where year < 2012;

DELETE 45117

delete from state\_country\_mthly\_temps where m\_date < '2012-01-01';

DELETE 615067

delete from country\_yearly\_co2 where year < 2012;

DELETE 16572

select count(\*) from country\_yearly\_co2;

count

-------

660

(1 row)

select count(\*) from state\_country\_mthly\_temps;

count

-------

2892

(1 row)

select count(\*) from country\_yearly\_temps;

count

-------

241

(1 row)

select count(\*) from country\_mthly\_temps;

count

-------

2892

(1 row)

select count(\*) from city\_country\_mthly\_temps;

count

-------

1200

(1 row)

select count(\*) from global\_mthly\_temps;

count

-------

1992

(1 row)

commit;

WARNING: there is no transaction in progress

COMMIT

\q

$