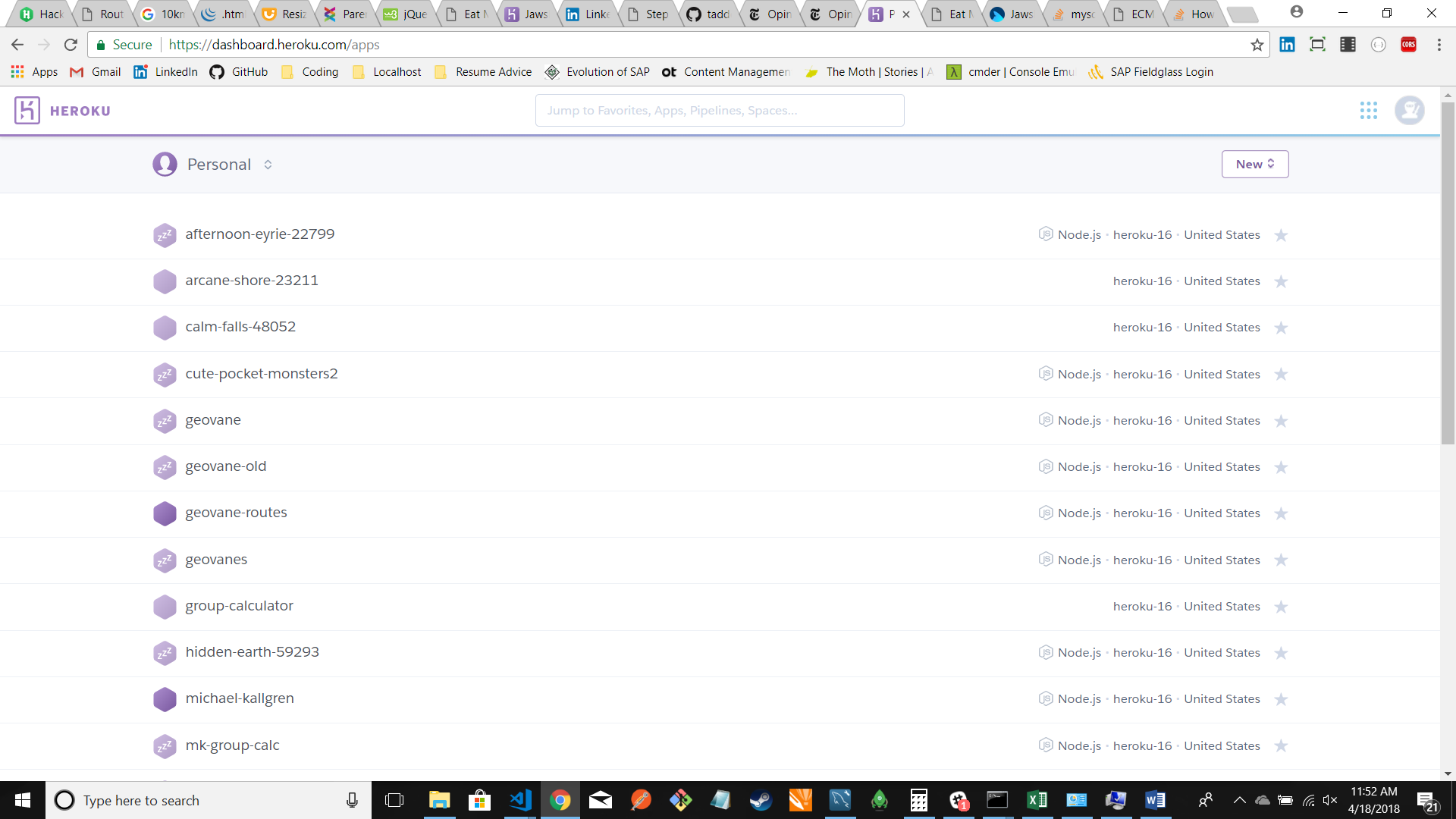
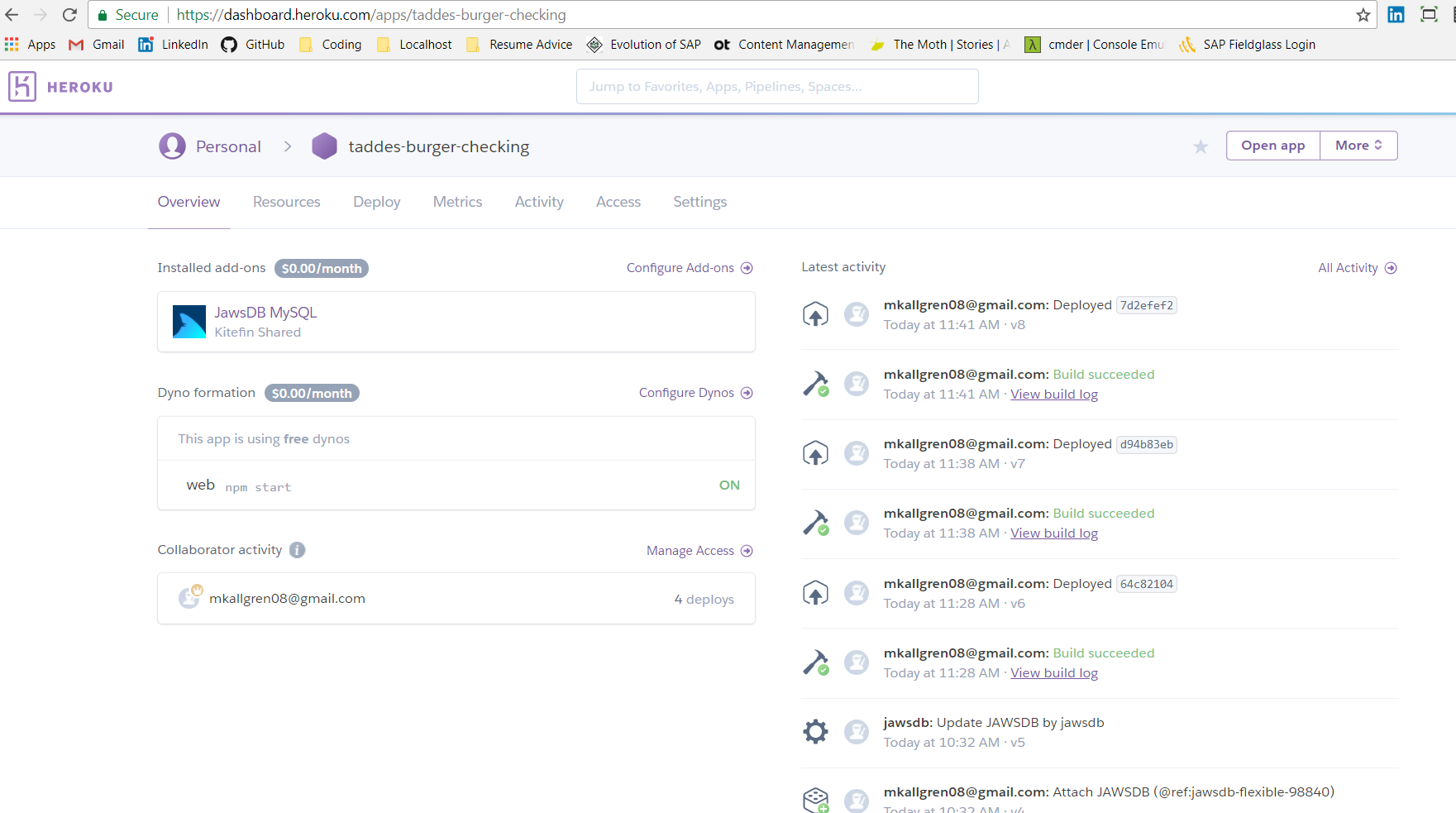
**Deploying and Seeding a Remote SQL DB via Heroku and MySQL Workbench**

**Step 1:** Create your Heroku app using the “heroku create [app-name]” CLI code

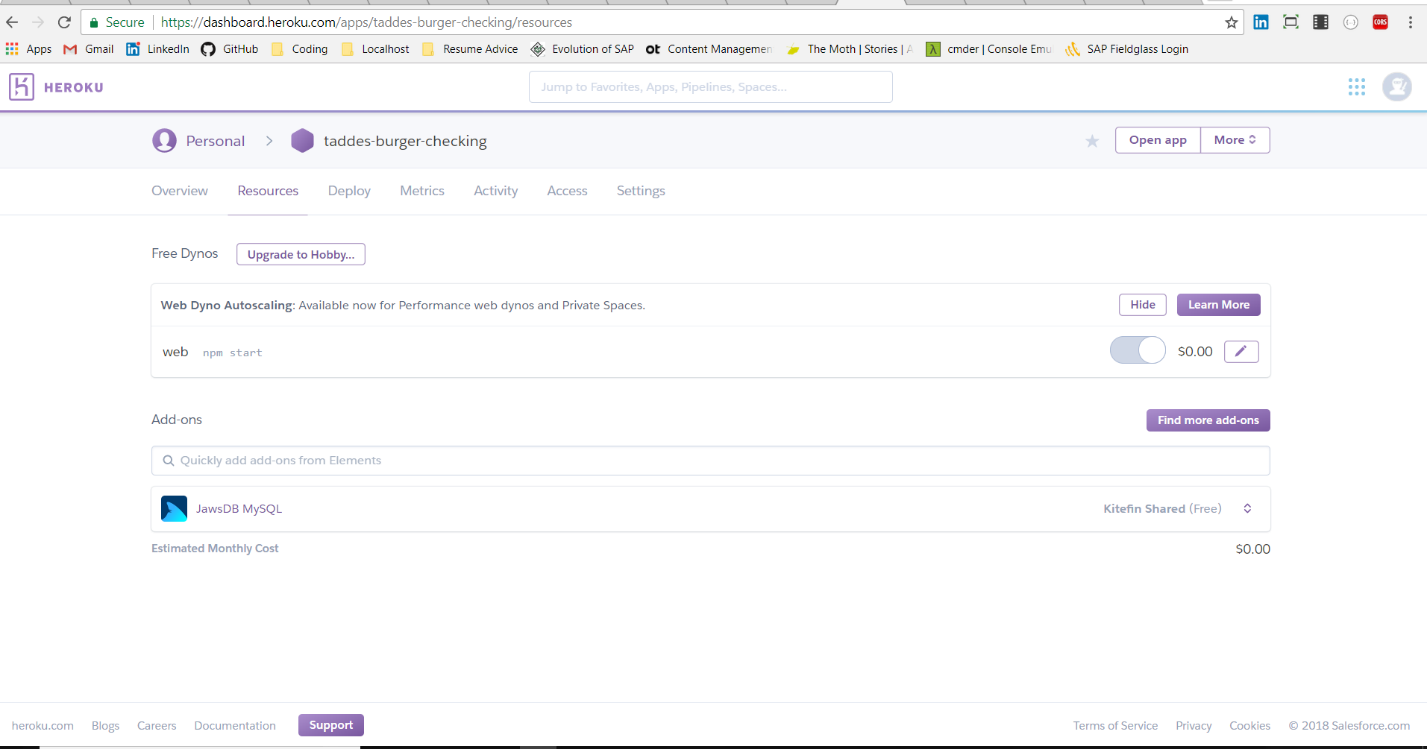
**Step 2:** Go to your Heroku dashboard at heroku.com:



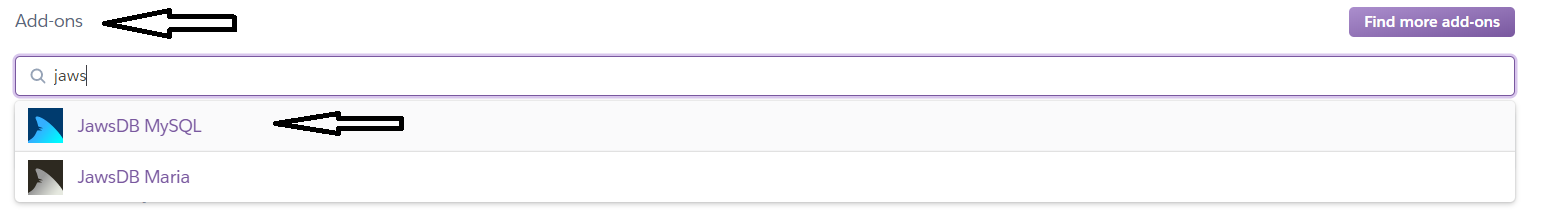
**Step 3:** Select your app:



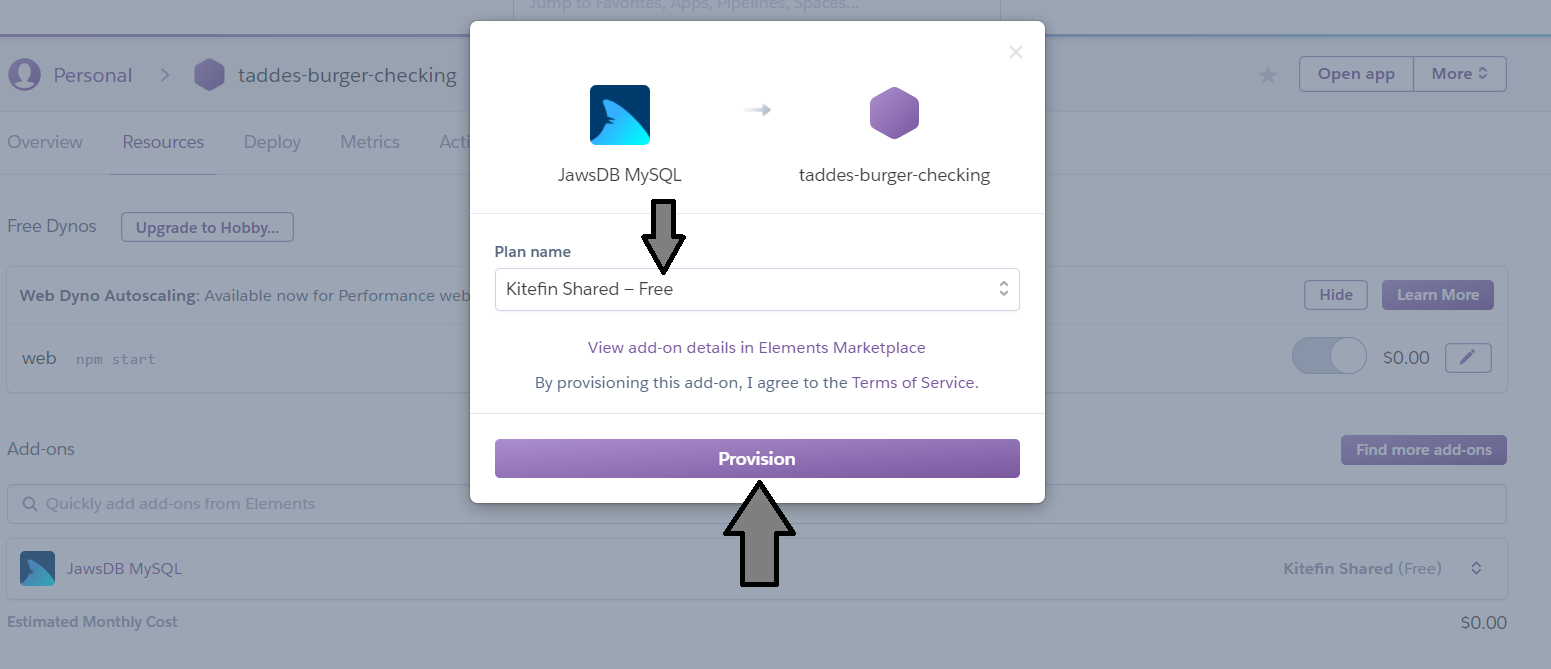
**Step 4:** Click the ‘Resources’ tab:



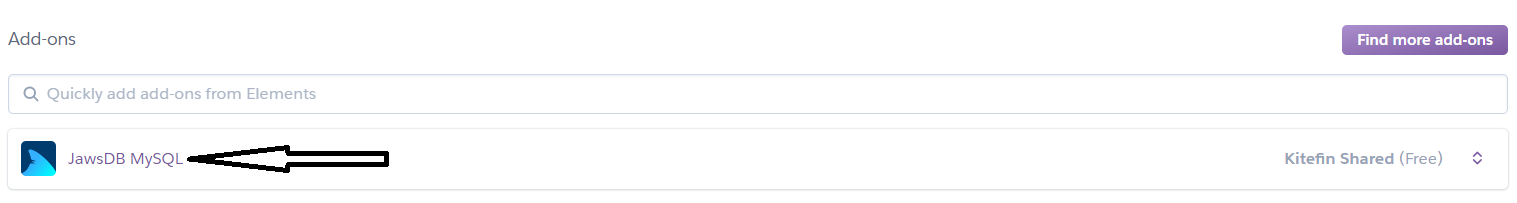
**Step 5:** Under ‘Add-ons’ search for ‘JawsDB MySQL’:



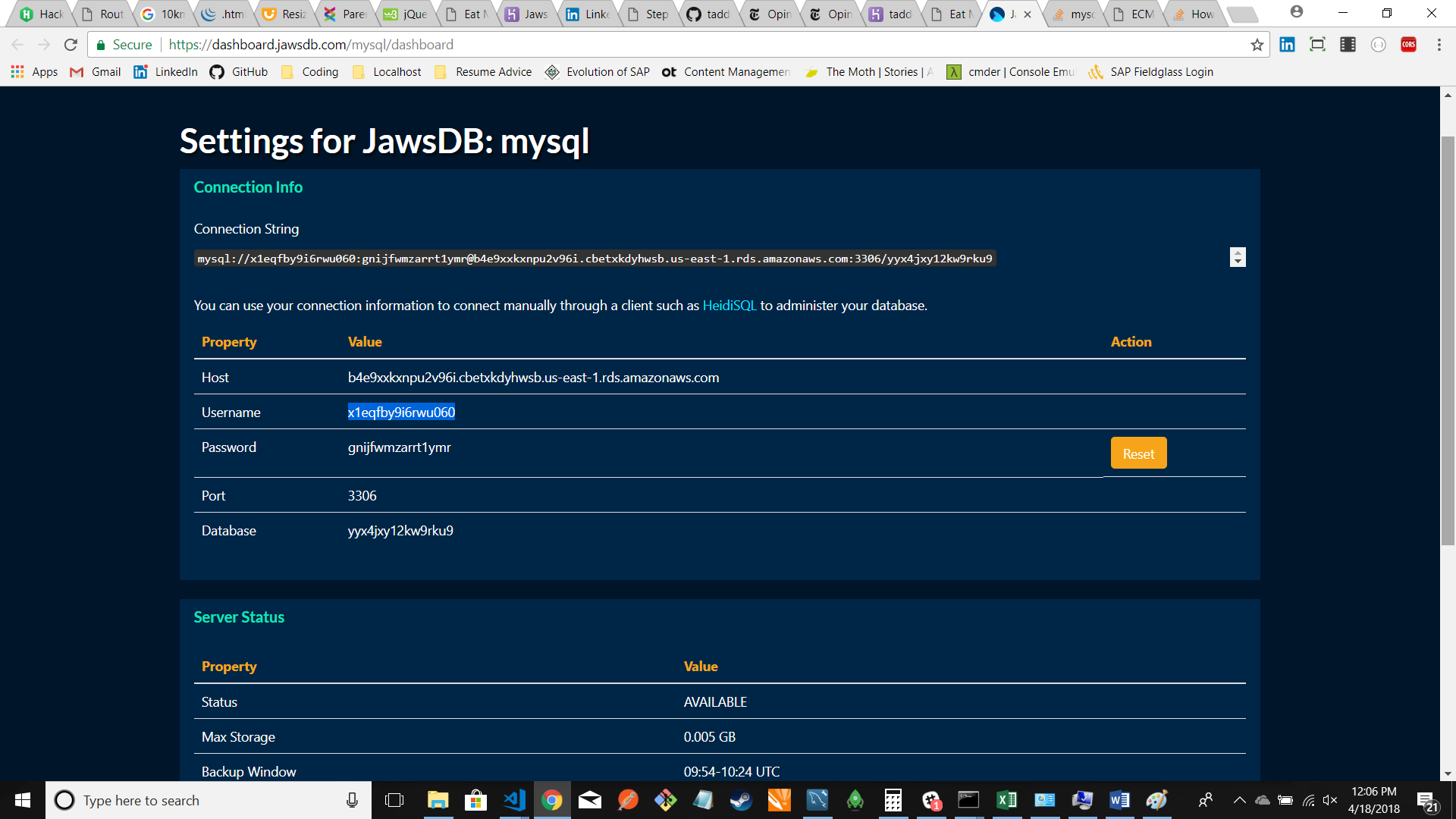
**Step 6**: Select the ‘Kitefin Shared – Free’ option and click ‘Provision’:



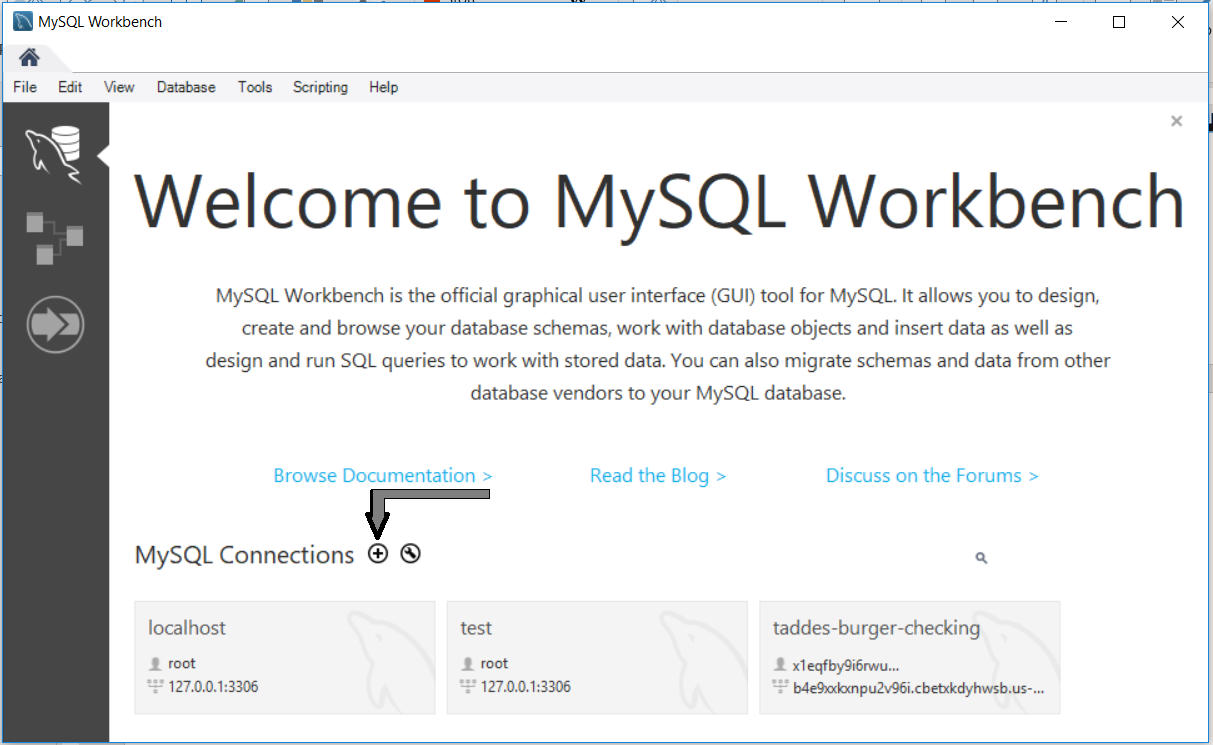
**Step 7:** Click on the ‘JawsDB MySQL’ link under the search bar in ‘Add-ons’; you’ll be taken to a new page:



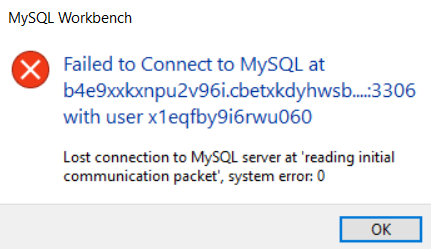
**Step 8:** Make sure the page you’re on looks something like this:



**Step 9:** Open MySQL Workbench and click the icon to make a new connection

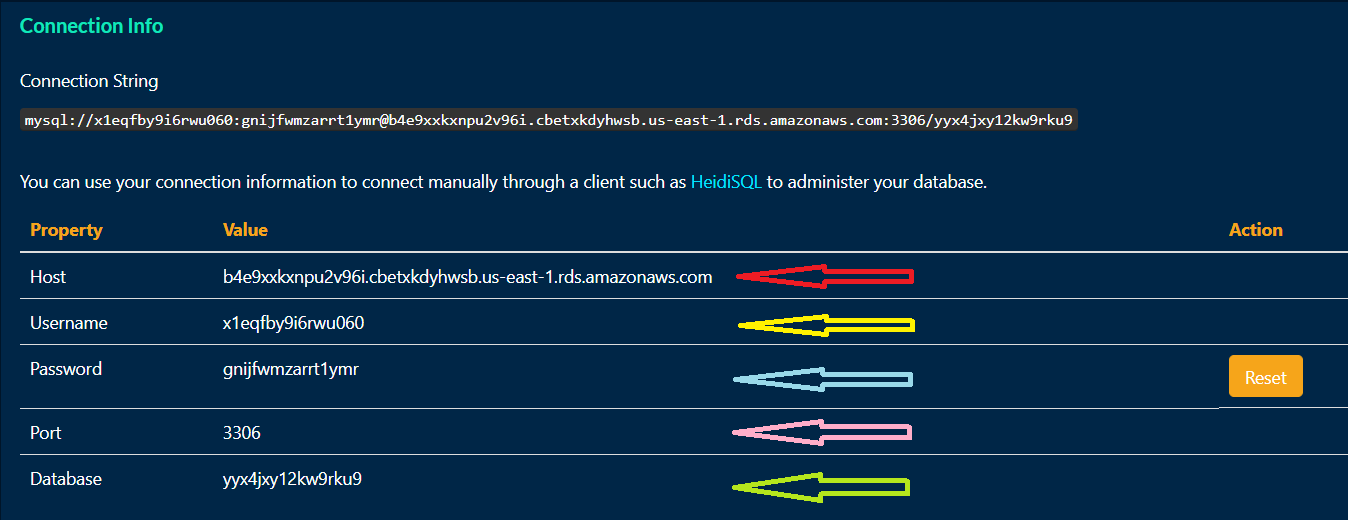


**Step 10:** Using the connection data from your JawsDB connection page, fill in the fields of your new MySQL Connection, and then test the connection:

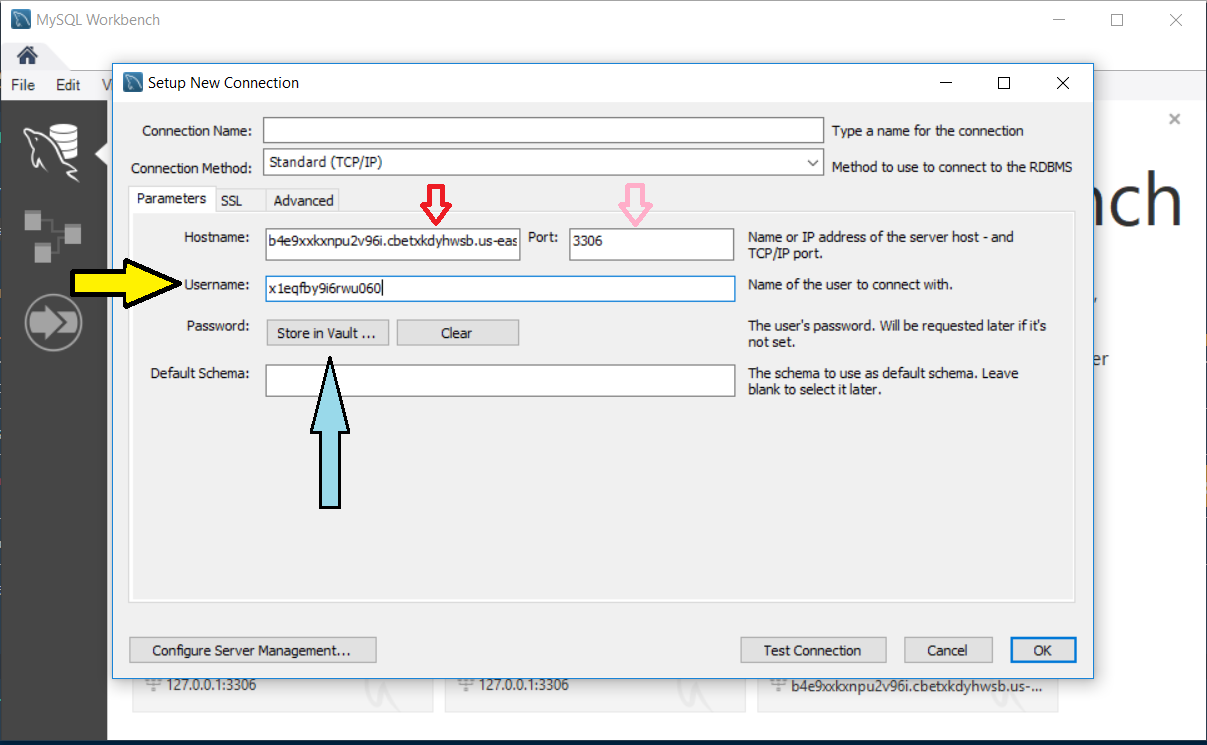
NOTE: Be sure that the network you’re on will allow a connection to a remote db. If your connection fails due to an error message like this -

--- all you need to do is get onto a network that allows you access.

JawsDB page:



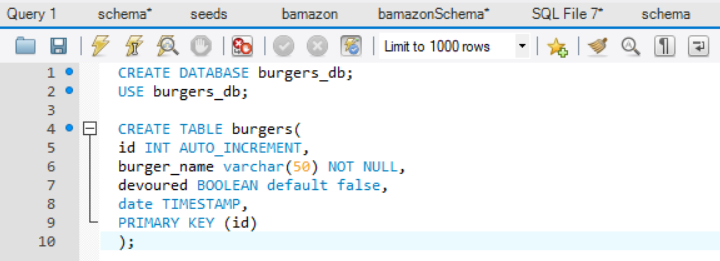
MySQL Workbench – New Connection:



**NOTE:** Don’t worry about the data base (the light green arrow) – that comes later.

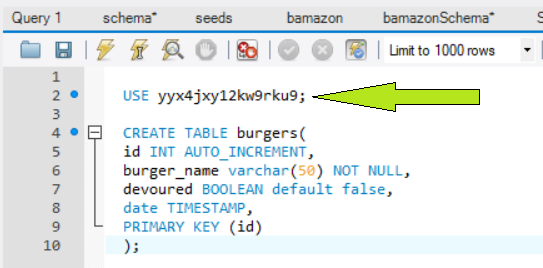
**Step 11:** Open your remote database and run your schema and seed files in MySQL Workbench(with one exception – I’ll note that in a second):

schema.sql (wrong):



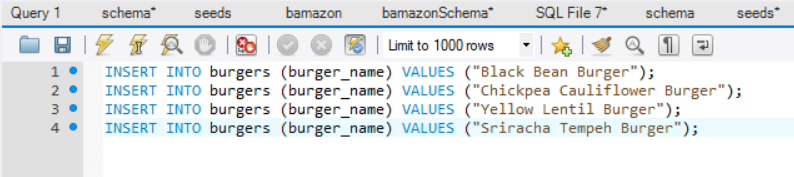
This schema is wrong because it A) tries to create a database in the remote database (one already exists!) and B) it tries to use the wrong database.

schema.sql (right):



Remember the light green arrow from before (with the Database info from the JawsDB webpage)? This is where that data is used.

seeds.sql:

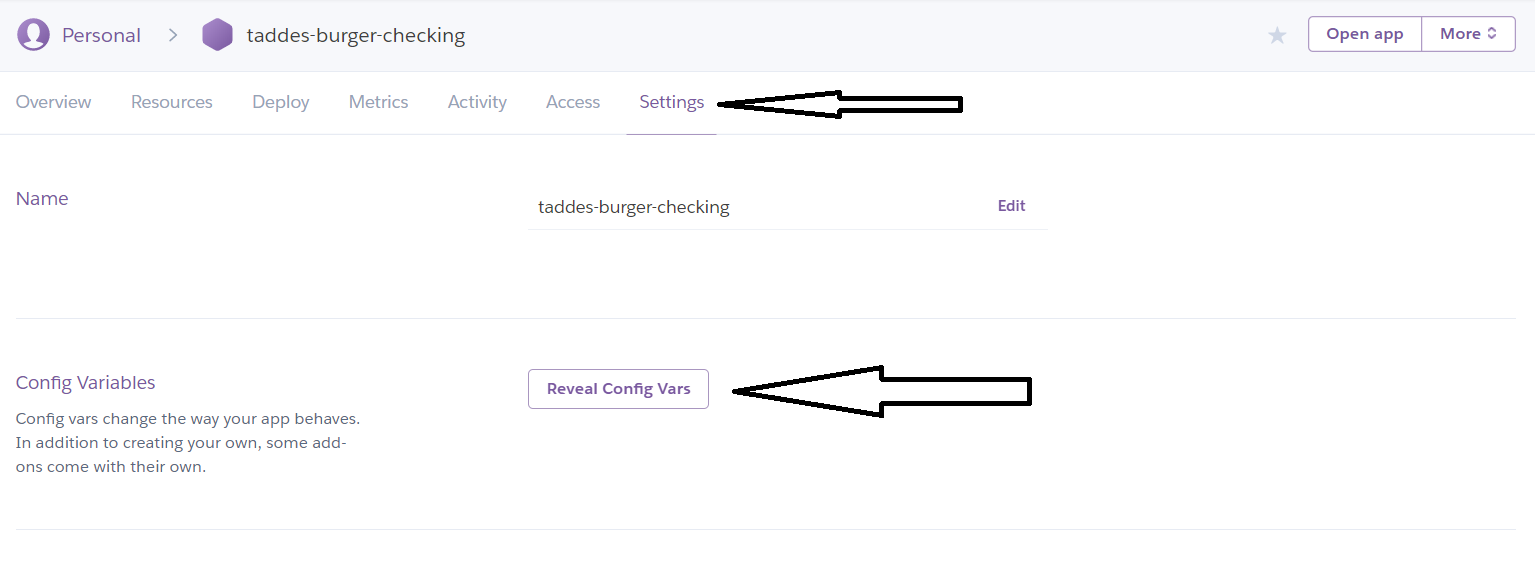


Run the seeds.sql as you normally would in MySQL Workbench

**Step 12:** Make sure that your MySQL connection uses process.env.JAWSDB\_URL to connect in your javascript file:



This works because when you provisioned your JawsDB add-on (Step 6), it created a Config Var in Heroku called “JAWSDB\_URL” that contains all the data you need to connect to your app’s database. If you would like to see your app’s remote Config Vars, go to the Settings Tab on your app’s Heroku page and click the button that says “Reveal Config Vars”



And after revealing the Config Vars, you can add custom ones to your application (similar to adding custom environment variables on your local machine):

