SETTING UP REDIS FOR LOCAL USE

Redis is a memory-based data structure store that we will be using extensively throughout the course. Redis' website can be found here: https://redis.io/

While Redis is very powerful and widely used, it only supports the Linux platform. Normally this would not be a problem, since almost all web servers in the world are running Linux. However, for us to work with it while we develop on Windows we will need a workaround.

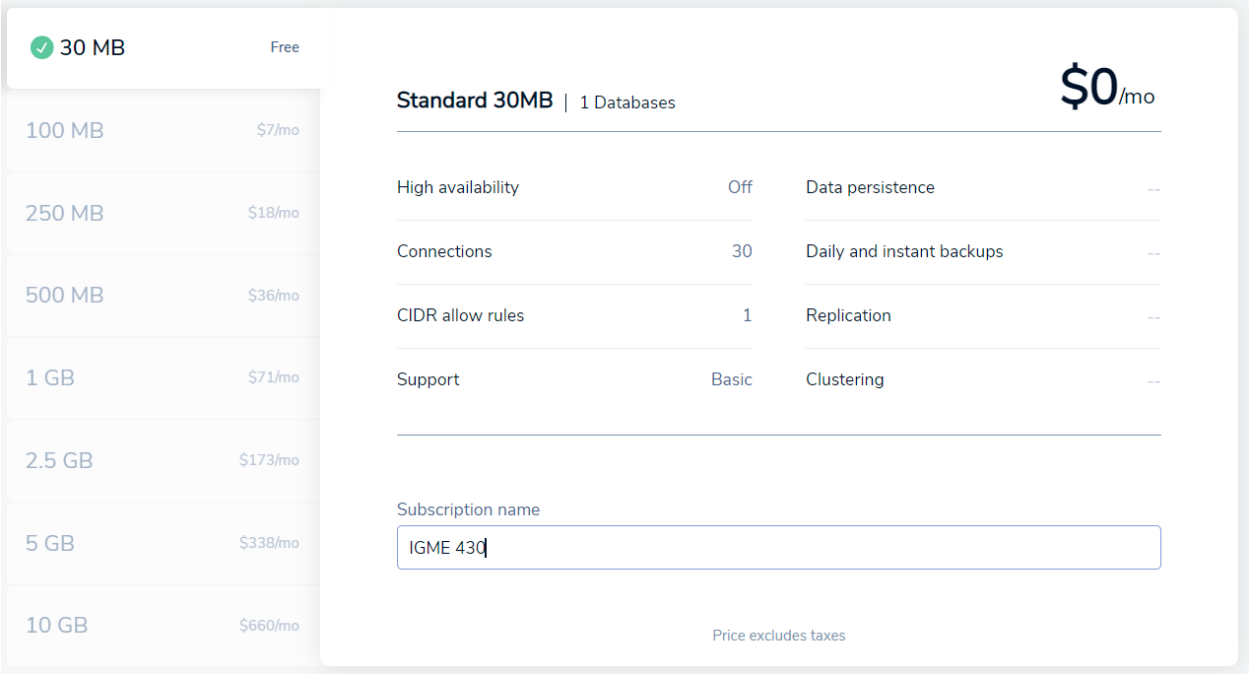
Luckily, cloud-based Redis servers exist and can be used up to a limit for free. This guide will walk you through setting up an account on Redis Labs: Redis Cloud so that you can use it for local development.

Contents

SETTING UP REDIS FOR LOCAL USE

Creating an Account with Redis Labs

1. Navigate to https://redis.com/try-free/
2. Select “Sign up with Github” and authenticate using your GitHub account. Alternatively, you can use “Sign up with Google” and use your RIT account. Remember that you will need to use @g.rit.edu to do this properly.
3. Follow the steps to create your account.
4. Click the “Create your subscriptions” button. On the new screen, click “Fixed Plans” and scroll down. Keep all the options at the default (AWS, US East, 30MB / Free). For “Subscription Name” enter whatever you like.
5. Create a name for your subscription (this could be anything), and select the “30mb Free” plan.

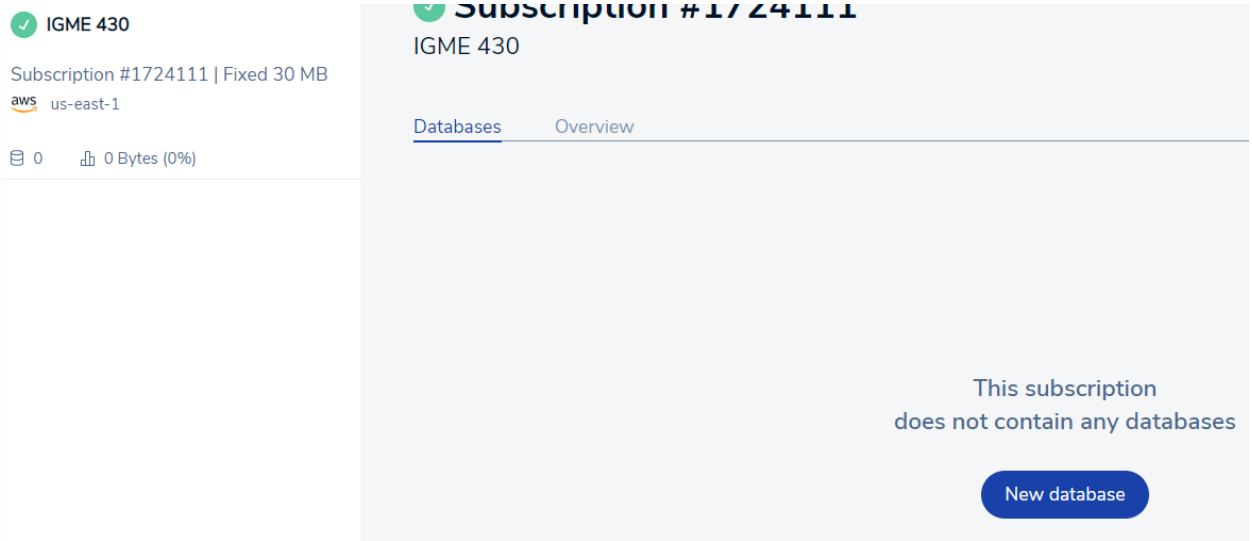


6. Click Create Subscription. This will bring you to a new screen for you subscription.

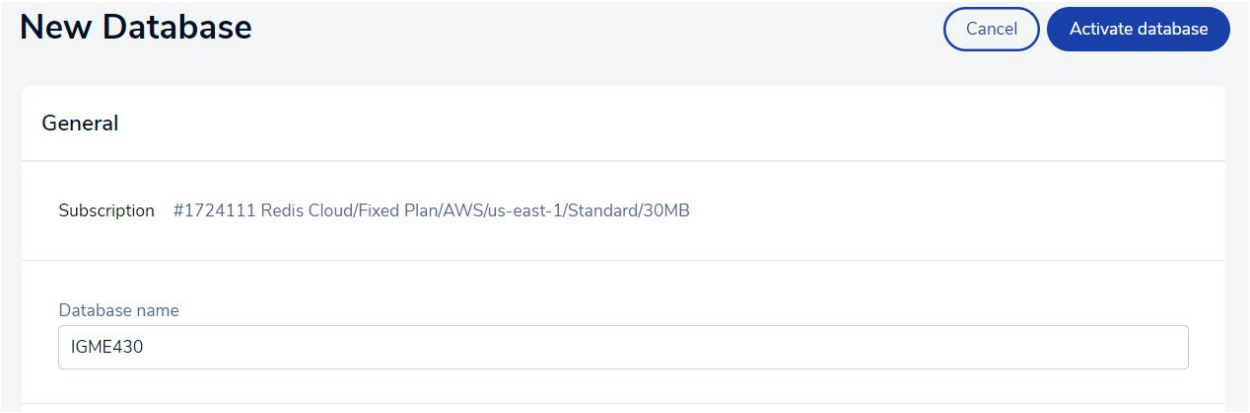
---CONTINUE TO NEXT PAGE---

SETTING UP REDIS FOR LOCAL USE

7. Now we need to create a database. Click the “New Database” button at the center of the screen.



8. Enter a name for this Database. This, again, can be anything you want. Keep all other options as the defaults. Then click “Activate Database” in the upper right corner.



9. After a few moments, your “public endpoint” should populate with a URI. Copy this down somewhere. Scroll down to the “Security” section. There should be a “Default User Password”. Press the “Copy” button, and paste it in the same place you have written down the URI.

    a. The URI will look something like this (note this is not a real link):

    redis-10020.c15.us-east-1-2-.ec2.cloud.redislabs.com:10020

SETTING UP REDIS FOR LOCAL USE

The number at the end following the colon is the port number

b. Your password will look like a string of 32 random letters and numbers

Using your Redis Labs database from code

Once you have the information from above, you can plug it into your code anywhere that you need to use Redis. The string must follow the following format:

redis://default:YOUR_PASSWORD@YOUR_ENDPOINT_WITH_PORT

Below is a sample of how that might look:

```
const redisURL = process.env.REDISCLOUD_URL ||
  'redis://default:a2jj92f3n928n3f09823f98n34f@redis-10020.c15.us-east-1-2.ec2.cloud.redislabs.com:10020';
  //              |---------PASSWORD--------| |--------------------HOSTNAME--------------------| |PORT|

let redisClient = redis.createClient({
  legacyMode: true,
  url: redisURL,
});
redisClient.connect().catch(console.error);
```

Once we talk about config files, we can move this information there to prevent people from seeing it in our code.

Using your Redis Labs database on Heroku

Once you have the proper URL from above, you can make use of Heroku's config vars to store the URL. This is useful if you want to swap out the endpoint without redeploying your server code.

Open your Heroku app, click "Settings", scroll down and click "Reveal Config Vars" and enter your REDISCLOUD_URL.