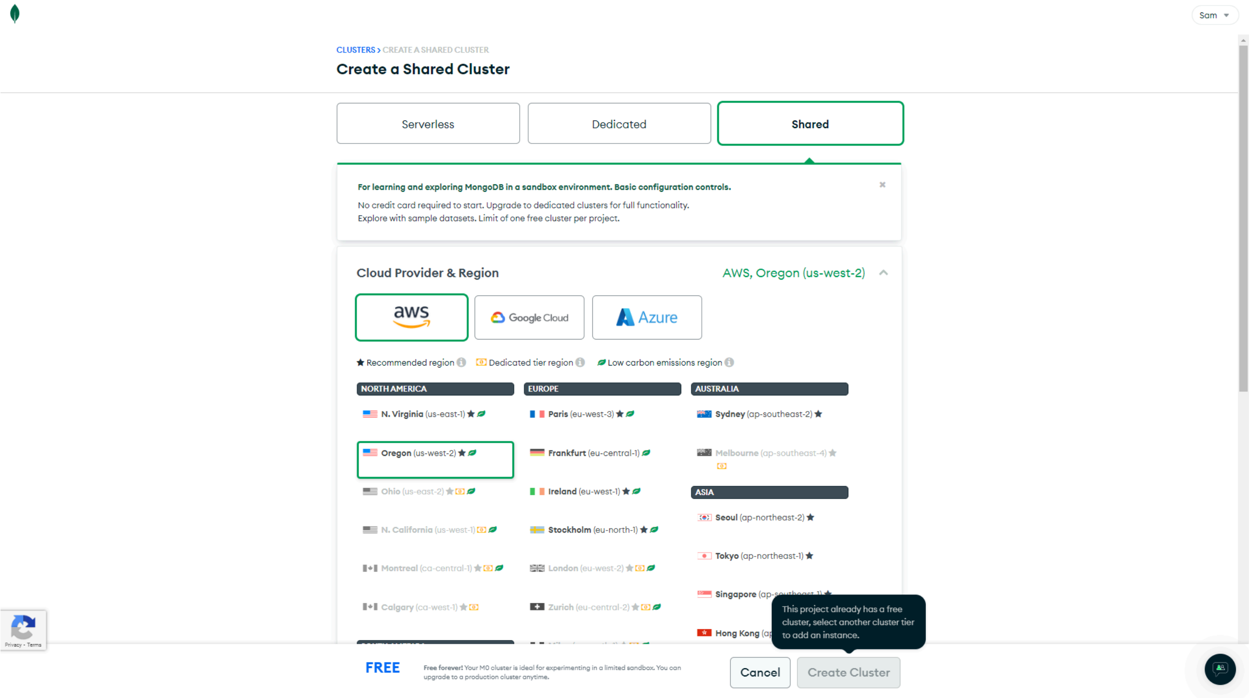
Vroom Room Deployment Guide

1. Get MongoDB database up and running
    a. Create an account with MongoDB Atlas
    b. Deploy a cluster via “Create deployment”
        i. M0 sandbox free tier in the shared tab is sufficient
        ii. Default configuration is fine



CLUSTERS > CREATE A SHARED CLUSTER

Create a Shared Cluster

For learning and exploring MongoDB in a sandbox environment. Basic configuration controls.

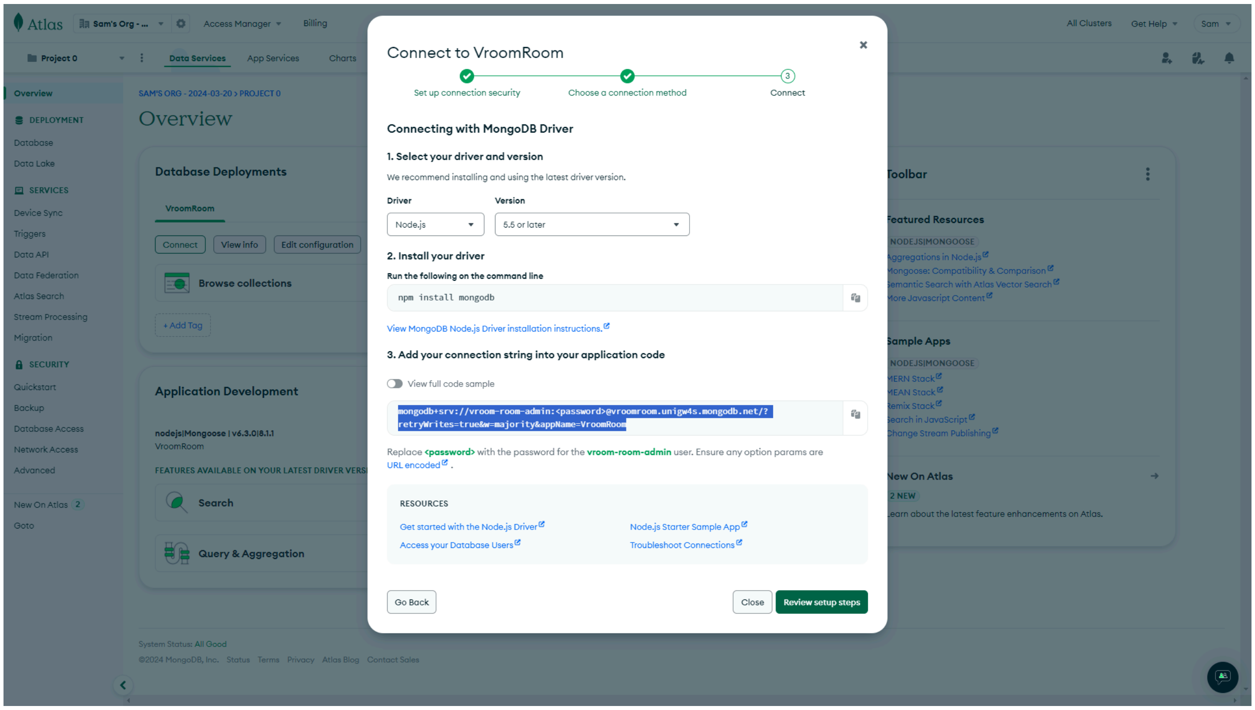
No credit card required to start. Upgrade to dedicated clusters for full functionality.
Explore with sample datasets. Limit of one free cluster per project.

Cloud Provider & Region: AWS, Oregon (us-west-2)

This project already has a free cluster, select another cluster tier to add an instance.

FREE Free forever! Your M0 cluster is ideal for experimenting in a limited sandbox. You can upgrade to a production cluster anytime.

    c. Create new admin database user with password Authentication method under database access
    d. Add 0.0.0.0/0 under network access
    e. Click “connect” under new databases ⟶ Drivers ⟶ Copy Connection String with filled in password



Connect to VroomRoom

Connecting with MongoDB Driver

1. Select your driver and version

We recommend installing and using the latest driver version.

Driver: Node.js

Version: 5.5 or later

2. Install your driver

Run the following on the command line

```
npm install mongodb
```

View MongoDB Node.js Driver installation instructions.

3. Add your connection string into your application code

```
mongodb+srv://vroom-room-admin:<password>@vroomroom.unigw4s.mongodb.net/?
retryWrites=true&w=majority&appName=VroomRoom
```

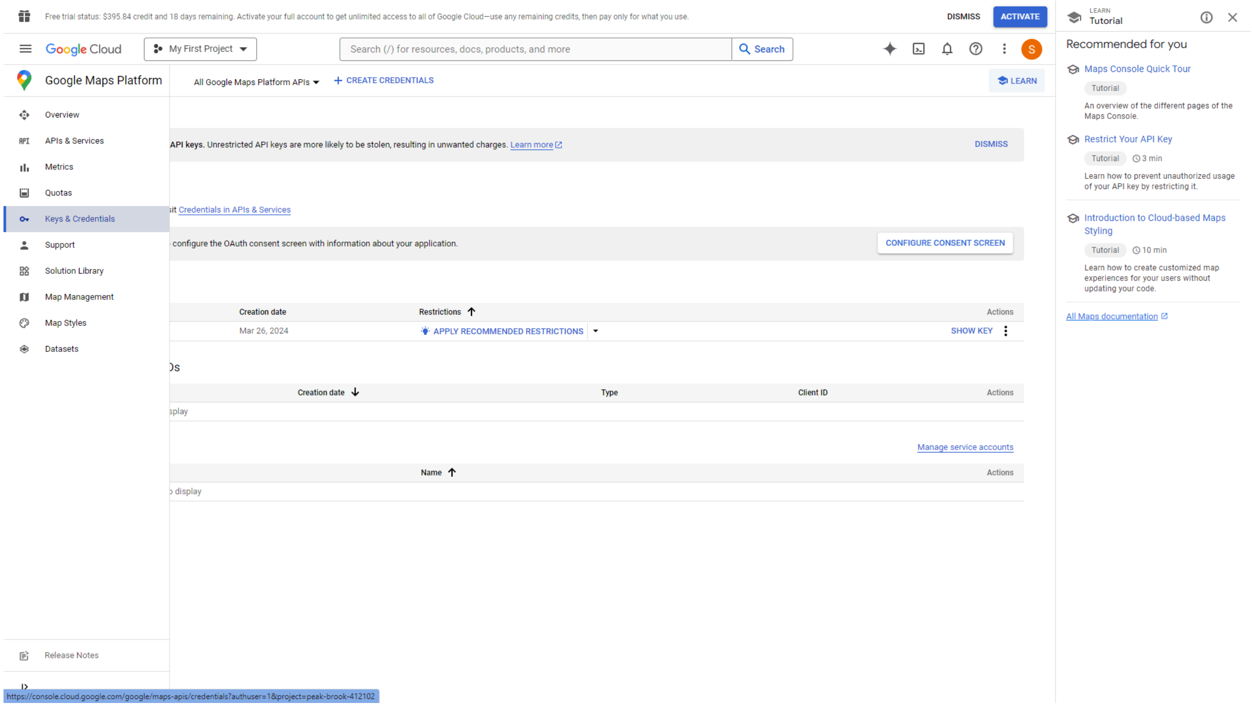
Replace <password> with the password for the vroom-room-admin user. Ensure any option params are URL encoded .

RESOURCES

Get started with the Node.js Driver
Node.js Starter Sample App
Access your Database Users
Troubleshoot Connections
Go Back
Close
Review setup steps

   f. Save result for later

2. Setup Google Maps API
   a. Create Google Cloud account
   b. Enable Places API, Directions API, Maps Javascript API
   c. Create new API credentials



Free trial status: $395.84 credit and 18 days remaining. Activate your full account to get unlimited access to all of Google Cloud—use any remaining credits, then pay only for what you use.

API keys. Unrestricted API keys are more likely to be stolen, resulting in unwanted charges. Learn more

it Credentials in APIs & Services

configure the OAuth consent screen with information about your application.

Creation date	Restrictions	Actions
Mar 26, 2024	APPLY RECOMMENDED RESTRICTIONS	SHOW KEY

Ds

Creation date	Type	Client ID	Actions
splay			

Name	Actions
o display	

LEARN
Tutorial

Recommended for you

Maps Console Quick Tour
Tutorial
An overview of the different pages of the Maps Console.

Restrict Your API Key
Tutorial 3 min
Learn how to prevent unauthorized usage of your API key by restricting it.

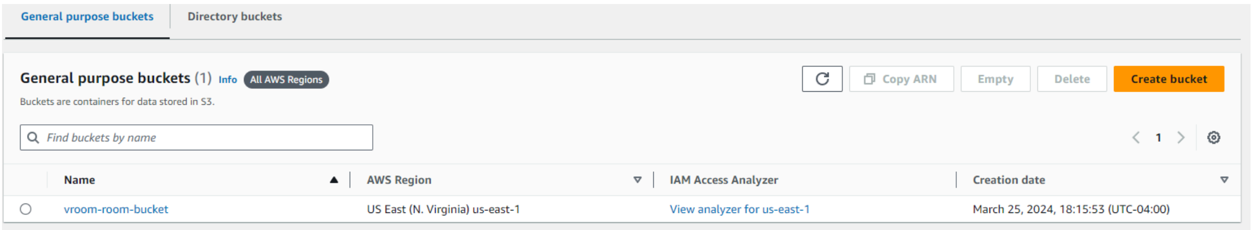
Introduction to Cloud-based Maps Styling
Tutorial 10 min
Learn how to create customized map experiences for your users without updating your code.

All Maps documentation

https://console.cloud.google.com/google/maps-apis/credentials?authuser=1&project=peak-brook-412102

   d. Save key for later

3. Setup AWS Bucket
   a. Create AWS account
   b. Create new IAM user with S3 privileges



General purpose buckets (1) Info All AWS Regions

Buckets are containers for data stored in S3.

	Name	AWS Region	IAM Access Analyzer	Creation date
○	vroom-room-bucket	US East (N. Virginia) us-east-1	View analyzer for us-east-1	March 25, 2024, 18:15:53 (UTC-04:00)

   c. Use that account to create new S3 bucket
      i. Allow public access
      ii. Allow ACL write access
   d. Save access key ID and access secret key in a secure location for later

4. Test Locally
   a. Clone directory from https://github.com/Samyuul/Group7-Capstone-Carpool
   b. In /frontend/routes/http-commons, in line 5, add baseURL: process.env.REACT_APP_URL_PROXY
   c. Create new .env file in /frontend with
      i. REACT_APP_GOOGLE_API_KEY = …
      ii. REACT_APP_URL_PROXY = http://localhost:8080
   d. Create new .env file in root directory with
      i. MONGO_URI = …
      ii. AWS_BUCKET_NAME = …
      iii. AWS_ACCESS_KEY = …
      iv. AWS_ACCESS_KEY_SECRET = …
      v. NODE_ENV = production
   e. Fill out .env files with the values obtained from steps 1 - 3
   f. In root directory, create build via “npm run local-build”
   g. Run build via “npm run local–start”
   h. Validate that the website works by going to http://localhost:8080

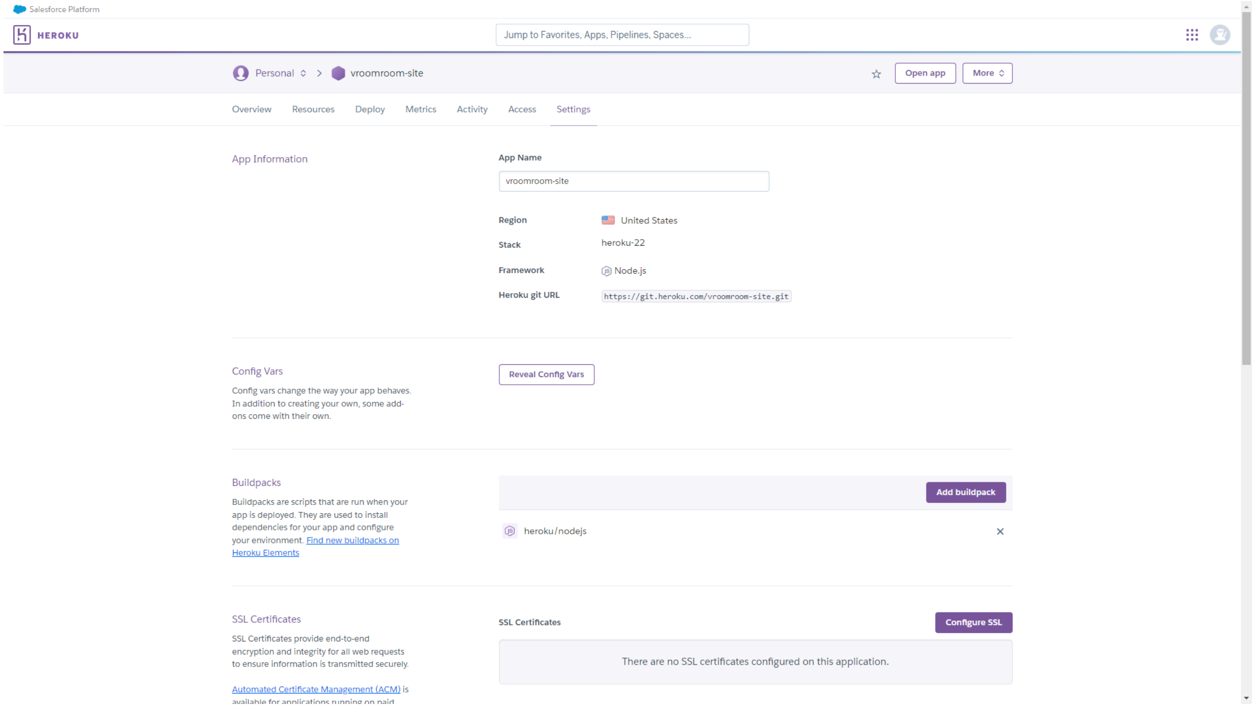
5. Deploy to Heroku
   a. Install heroku
   b. Login to heroku via CLI “heroku login”
   c. Create new app via “heroku create -a name_app”
   d. Remove the line you added in step 4 b)

e. Deploy via git push heroku main

f. Login into Heroku on browser

g. Go to newly created project settings

h. Go to config variables and add the .env variables in step 4



App Information

App Name: vroomroom-site

Region: United States

Stack: heroku-22

Framework: Node.js

Heroku git URL: `https://git.heroku.com/vroomroom-site.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/nodejs

SSL Certificates

SSL Certificates provide end-to-end encryption and integrity for all web requests to ensure information is transmitted securely.

Automated Certificate Management (ACM) is available for applications running on paid

SSL Certificates

There are no SSL certificates configured on this application.