



MERN Projects for Beginners

Create Five Social Web Apps Using
MongoDB, Express.js, React, and Node

Nabendu Biswas

Apress®

MERN Projects for Beginners

**Create Five Social Web Apps
Using MongoDB, Express.js,
React, and Node**

Nabendu Biswas

Apress®

MERN Projects for Beginners: Create Five Social Web Apps Using MongoDB, Express.js, React, and Node

Nabendu Biswas
Bhopal, India

ISBN-13 (pbk): 978-1-4842-7137-7

<https://doi.org/10.1007/978-1-4842-7138-4>

ISBN-13 (electronic): 978-1-4842-7138-4

Copyright © 2021 by Nabendu Biswas

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Louise Corrigan
Development Editor: James Markham
Coordinating Editor: Jessica Vakili

Distributed to the book trade worldwide by Springer Science+Business Media New York, 1 NY Plaza, New York, NY 10014. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/978-1-4842-7137-7. For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

Table of Contents

About the Author ix

About the Technical Reviewer xi

Chapter 1: MERN Deployment Setup..... 1

 The MERN Stack at a Glance..... 1

 Firebase Hosting Initial Setup 2

 MongoDB Setup 6

 Creating a New Project..... 6

 Database User and Network Access 12

 Deploying the Back End to Heroku..... 16

 Deploying the Front End to Firebase 23

 Install Node.js and npm 25

 Summary..... 31

Chapter 2: Building a Dating App with MERN 33

 Firebase Hosting Initial Setup 34

 React Basic Setup..... 34

 Creating a Header Component..... 36

 Creating the Dating Cards Component..... 39

 Creating the Swipe Buttons Component 44

 Initial Back-End Setup 47

 MongoDB Setup 49

 Initial Route Setup..... 49

 Database User and Network Access 50

 MongoDB Schema and Routes..... 51

 Integrating the Back End with the Front End 56

TABLE OF CONTENTS

| | |
|--|-----------|
| Deploying the Back End to Heroku..... | 58 |
| Deploying the Front End to Firebase..... | 59 |
| Summary..... | 59 |
| Chapter 3: Building a Short Video App with MERN | 61 |
| Firebase Hosting Initial Setup | 62 |
| React Basic Setup..... | 63 |
| Creating a Video Component..... | 63 |
| Creating a Video Footer Component..... | 69 |
| Creating a Video Sidebar Component..... | 75 |
| Making Components Dynamic | 78 |
| Initial Back-End Setup | 81 |
| MongoDB Setup..... | 83 |
| Initial Route Setup | 83 |
| Database User and Network Access..... | 84 |
| MongoDB Schema and Routes | 85 |
| Integrating the Back End with the Front End | 90 |
| Deploying the Back End to Heroku..... | 93 |
| Deploying the Front End to Firebase..... | 94 |
| Summary..... | 94 |
| Chapter 4: Building a Messaging App with MERN | 95 |
| Firebase Hosting Initial Setup | 96 |
| React Basic Setup..... | 97 |
| Creating a Sidebar Component | 97 |
| Creating a Sidebar Chat Component..... | 105 |
| Creating a Chat Component | 108 |
| Creating a Chat Footer Component..... | 114 |
| Initial Back-End Setup | 117 |
| MongoDB Setup | 119 |
| Initial Route Setup..... | 119 |

| | |
|---|------------|
| Database User and Network Access | 120 |
| MongoDB Schema and Routes..... | 121 |
| Configuring Pusher | 126 |
| Adding Pusher to the Back End..... | 130 |
| Adding Pusher to the Front End | 133 |
| Integrating the Back End with the Front End | 135 |
| Additional Setup..... | 141 |
| Creating a Login Component..... | 148 |
| Adding Google Authentication | 151 |
| Using Redux and Context API | 155 |
| Using Redux Data in Other Components | 159 |
| Deploying the Back End to Heroku..... | 166 |
| Deploying the Front End to Firebase | 167 |
| Summary..... | 168 |
| Chapter 5: Building a Photo-Based Social Network with MERN | 169 |
| Firebase Hosting Initial Setup | 170 |
| React Basic Setup..... | 172 |
| Creating a Header Component..... | 172 |
| Creating a Post Component | 173 |
| Making Components Dynamic | 177 |
| Firebase Authentication Setup | 180 |
| Creating a Modal for Signup | 182 |
| Sign up with Firebase | 187 |
| Sign in with Firebase | 189 |
| Adding Posts and Images..... | 191 |
| Initial Back-End Setup | 197 |
| MongoDB Setup | 199 |
| Initial Route Setup..... | 199 |
| Database User and Network Access | 200 |

TABLE OF CONTENTS

| | |
|--|------------|
| MongoDB Schema and Routes..... | 201 |
| Integrating the Back End with the Front End | 206 |
| Configuring Pusher | 213 |
| Adding Pusher to the Back End..... | 213 |
| Adding Pusher to the Front End | 215 |
| Hiding Secrets..... | 217 |
| Deploying the Back End to Heroku..... | 218 |
| Deploying the Front End to Firebase | 219 |
| Summary..... | 220 |
| Chapter 6: Build a Popular Social Network with MERN | 221 |
| Firebase Hosting Initial Setup | 222 |
| React Basic Setup..... | 223 |
| Adding a Styled Component..... | 223 |
| Creating a Header Component..... | 224 |
| Creating Sidebar Components | 229 |
| Creating a Feed Component..... | 232 |
| Adding a Widget..... | 237 |
| Creating a Messenger Component..... | 241 |
| Creating a Post Component | 245 |
| Google Authentication Setup..... | 250 |
| Creating a Login Component..... | 251 |
| Using Redux and Context API | 255 |
| Using Redux Data in Other Components | 258 |
| Initial Back End Setup..... | 261 |
| MongoDB Setup | 263 |
| Initial Route Setup..... | 263 |
| Database User and Network Access | 264 |
| Storing Images in MongoDB | 265 |
| MongoDB Schema and Routes..... | 271 |

| | |
|---|------------|
| Integrating the Back End with the Front End | 272 |
| Configuring Pusher | 277 |
| Adding Pusher to the Back End..... | 277 |
| Adding Pusher to the Front End | 279 |
| Deploying the Back End to Heroku..... | 280 |
| Deploying the Front End to Firebase | 280 |
| Deploying the Front End to Firebase | 281 |
| Summary..... | 282 |
| Index..... | 283 |

About the Author

Nabendu Biswas is a full-stack JavaScript developer who has been working in the IT industry for the past 16 years. He has worked for some of the world's top development firms and investment banks. Nabendu is a tech blogger who publishes on DEV Community (dev.to), Medium (medium.com), and The Web Dev(TWD) (thewebdev.tech). He is an all-around nerd who is passionate about everything JavaScript, React, and Gatsby. You can find him on Twitter [@nabendu82](https://twitter.com/nabendu82).

About the Technical Reviewer

Alexander Nnakwue is a self-taught software engineer with experience in back-end and full-stack engineering. With experience spanning more than four years, he loves to solve problems at scale. Currently, he is interested in startups, open source web development, and distributed systems. In his spare time, Alexander loves watching soccer and listening to all genres of music.

CHAPTER 1

MERN Deployment Setup

Welcome to *MERN Projects for Beginners*, where you learn to build awesome web apps using the MERN (MongoDB, Express, React, Node.js) framework. This stack is in high demand in the startup sector because you can make a fully functional web app using it. A front-end engineer who knows HTML, CSS, and React can quickly learn Node.js and MongoDB and build a fully production-ready web app.

In this book, you learn how to host a back end using Node.js code in Heroku. The front-end site uses React code and Firebase hosting. It is also hosted through a cloud database called MongoDB Atlas. Most of the hosting setups are the same in the next five chapters, so it won't be repeated in most chapters.

The MERN Stack at a Glance

Before installing Firebase, let's discuss the basics of the technologies involved in the MERN stack.

- **MongoDB** is an open source document based on the NoSQL database. It is different from traditional relational databases that store data in tables. It stores data in JSON-like documents. It is highly scalable and performance-oriented and thus suited for modern-day web apps.
- **React** is the most popular open source JavaScript library for building a website's or web app's front end or user interface. It is developed and maintained by Facebook.
- **Node.js** lets developers write server-side code using JavaScript. It integrates very well with React or Angular at the front end and with MongoDB for databases.
- **Express** is a framework of Node.js, and through it, you can create API endpoints, which are the basis of any back-end server-side code.

Firestore Hosting Initial Setup

You need a Google account to work with Firestore. Go to <https://firebase.google.com> and click **Go to console** in the top-right corner. You must be logged in to your Google account to do so, as seen in Figure 1-1.

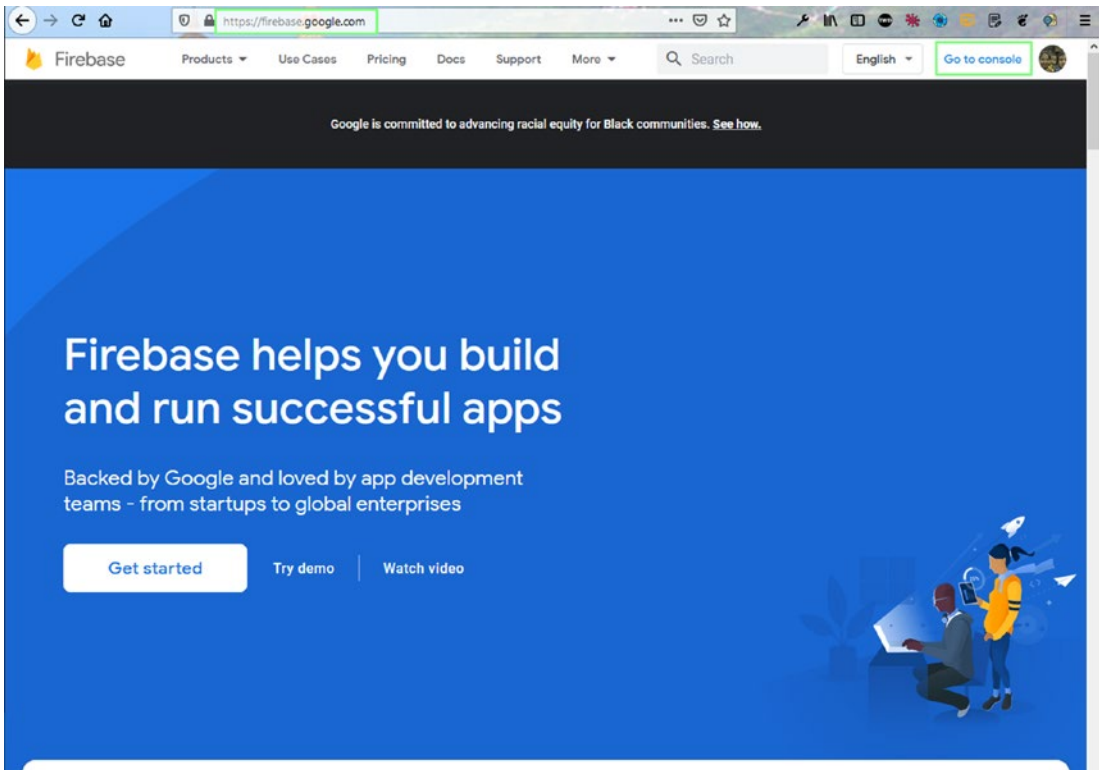


Figure 1-1. Firestore console caption

Click the **Add project** link on the page, as seen in Figure 1-2.

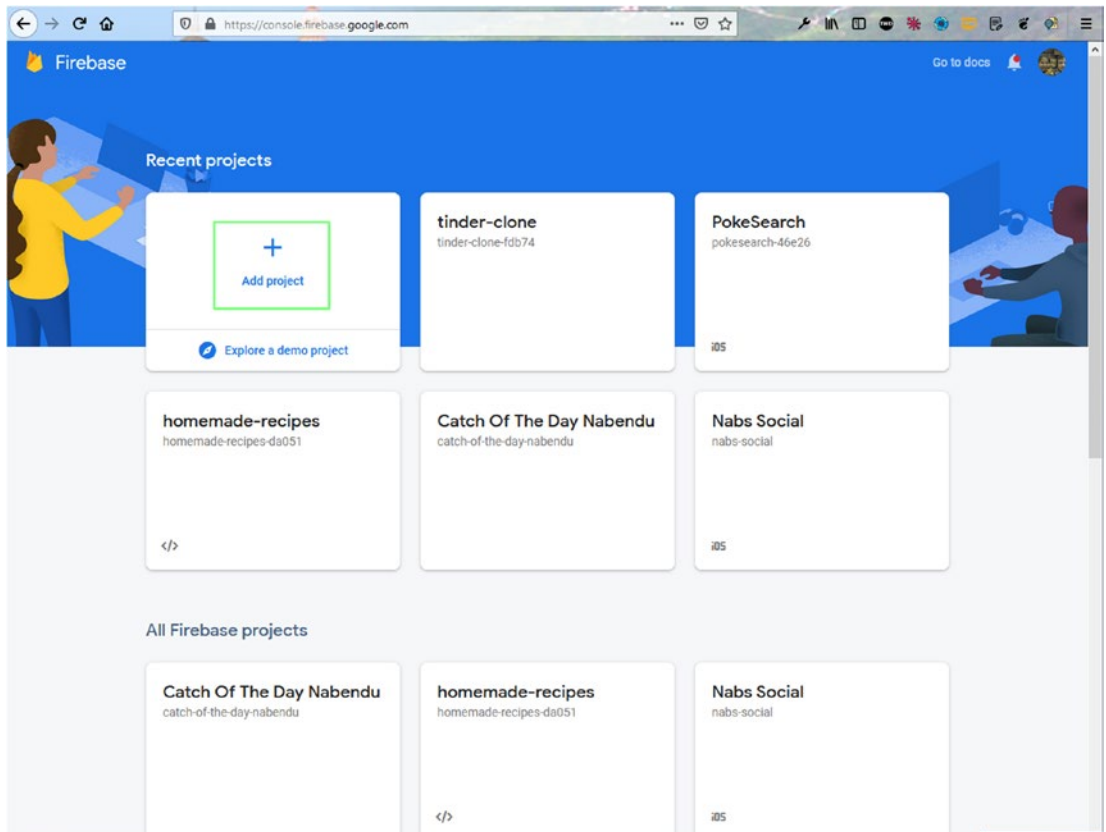


Figure 1-2. Add project

On this page, name the project **dating-app-mern**, and then click the **Continue** button, as seen in Figure 1-3. Note that this is just an installation instruction. You start building the app in the next chapter.

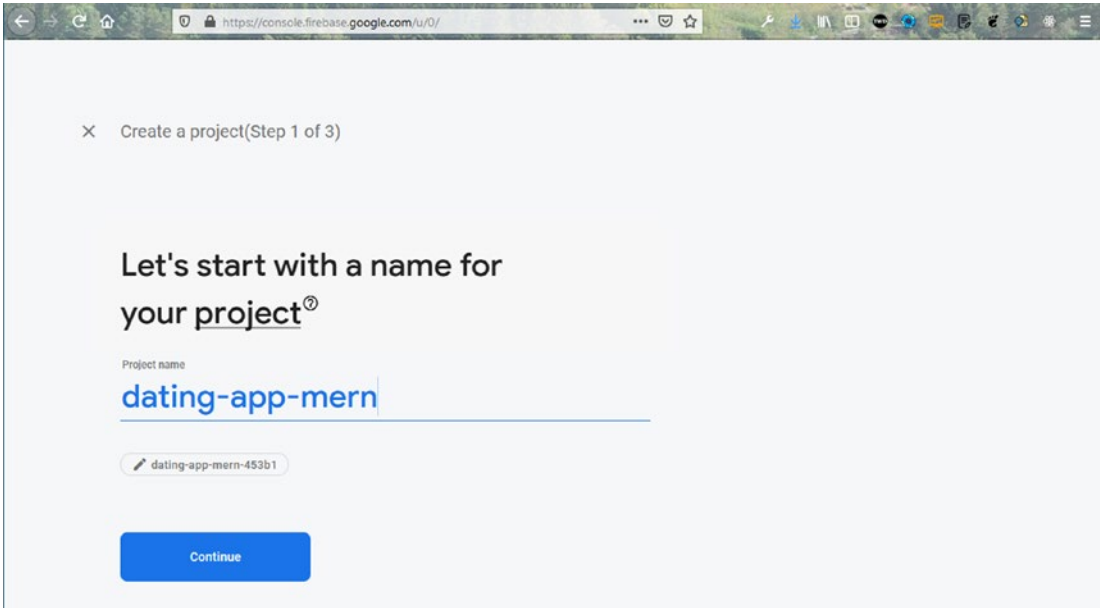


Figure 1-3. App name

On the next page, click the **Create project** button, as seen in Figure 1-4.

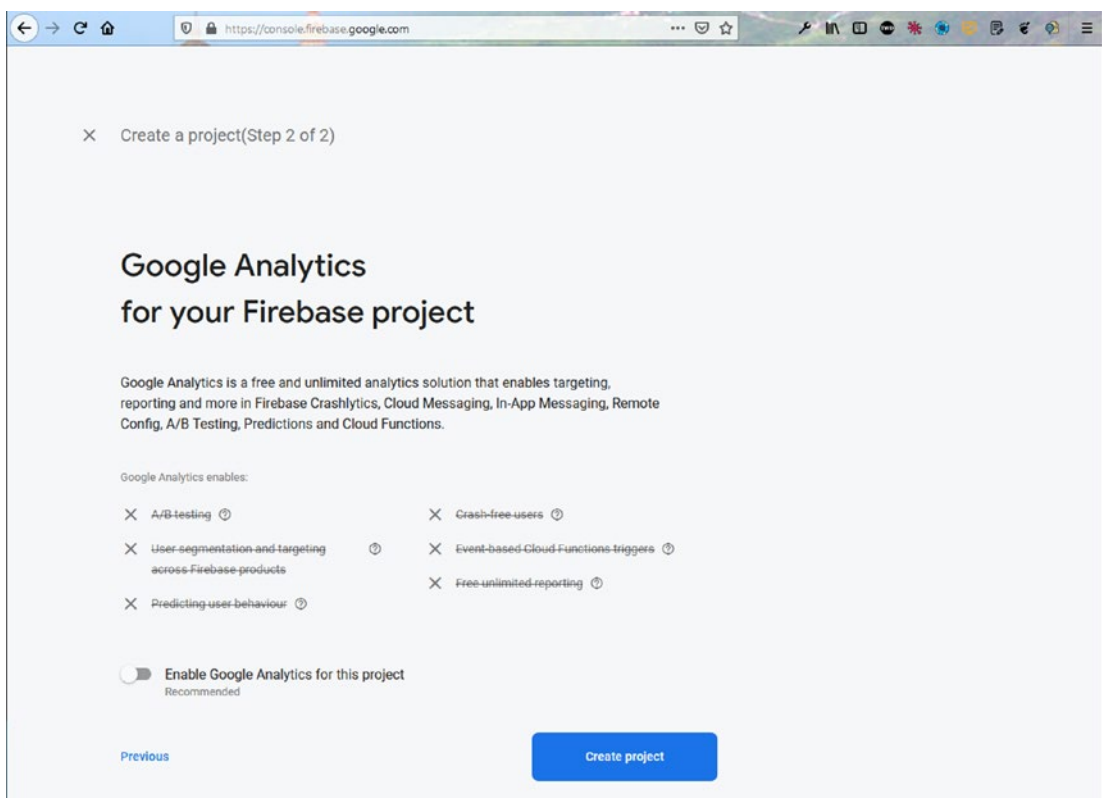


Figure 1-4. *Create project*

It takes some time to create the project, as seen in Figure 1-5.

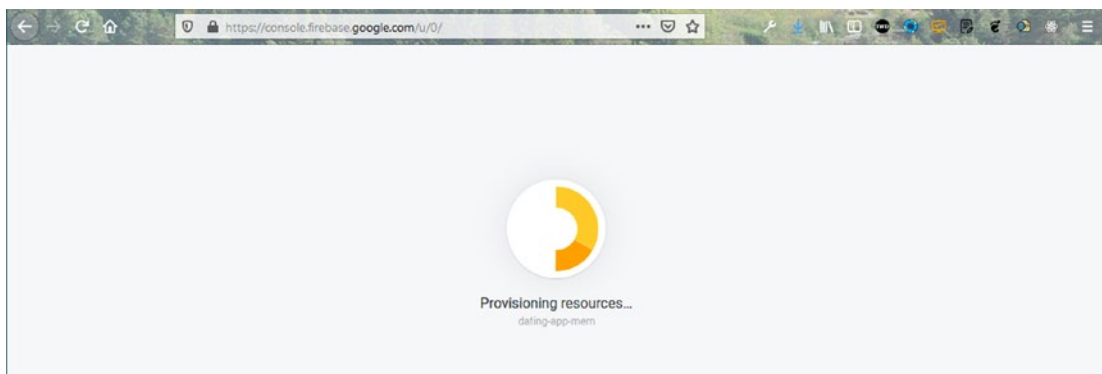


Figure 1-5. *Project created*

MongoDB Setup

MongoDB is the database that you work with on the cloud. It is also known as MongoDB Atlas. This is easier to work with than setting up on a local machine. Go to www.mongodb.com and log in or create a new account.

Creating a New Project

After logging in, you see a screen similar to the one shown in Figure 1-6. Click the **New Project** button.

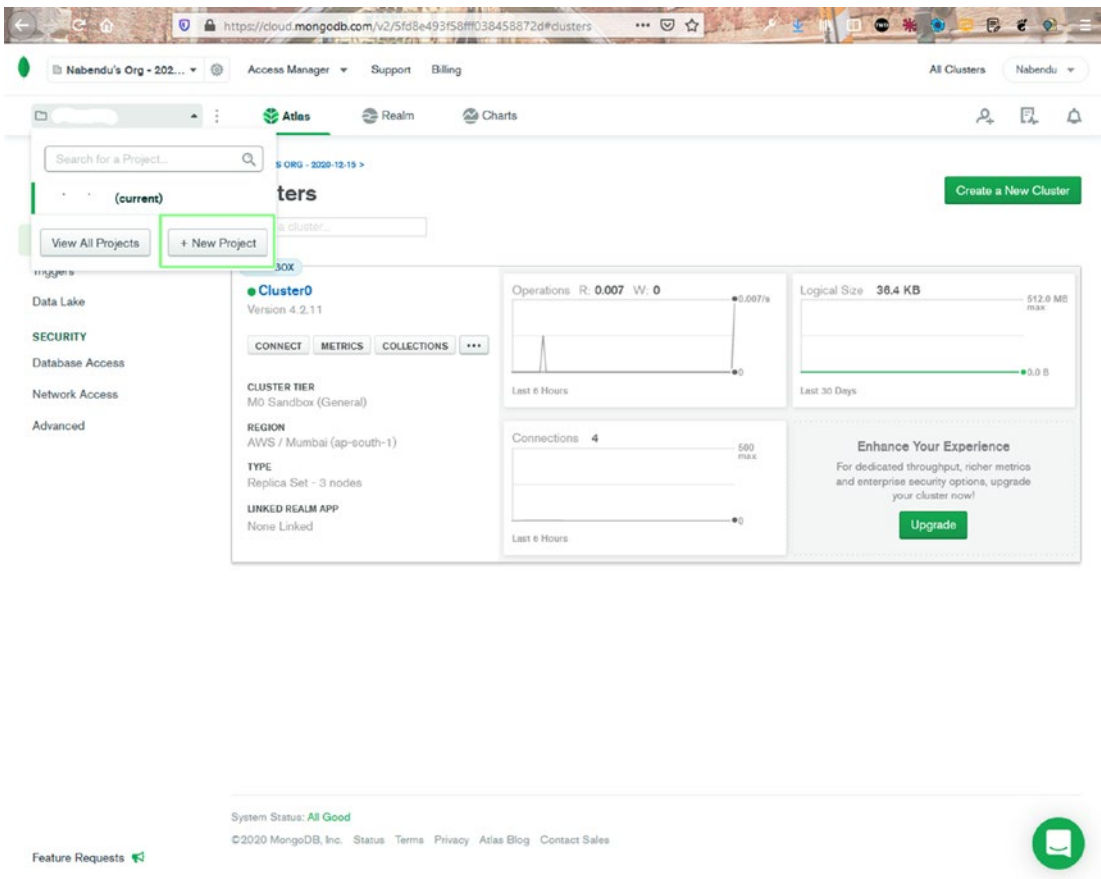


Figure 1-6. MongoDB new project

Name your project **dating-app-mern**, and then click the **Next** button, as seen in Figure 1-7.

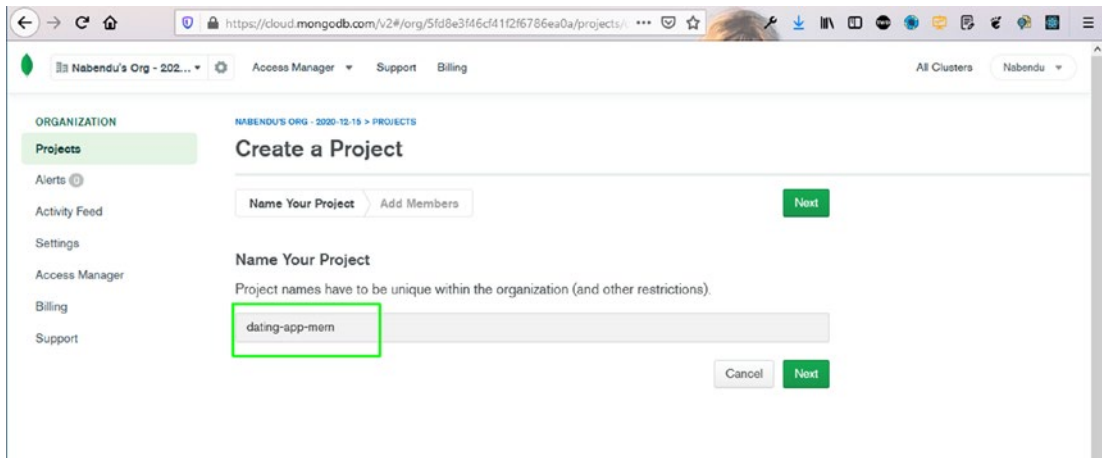


Figure 1-7. Project name

On the next screen, click the **Create Project** button, as seen in Figure 1-8.

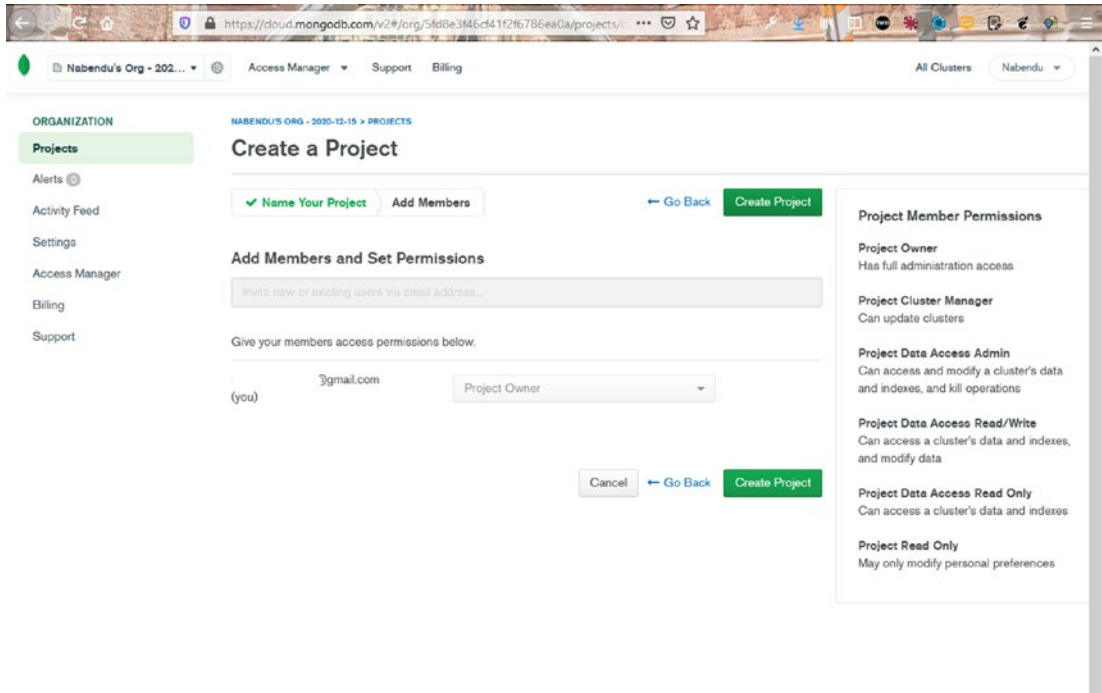


Figure 1-8. MongoDB Create Project

On the next screen, click the **Build a Cluster** button, as seen in Figure 1-9.

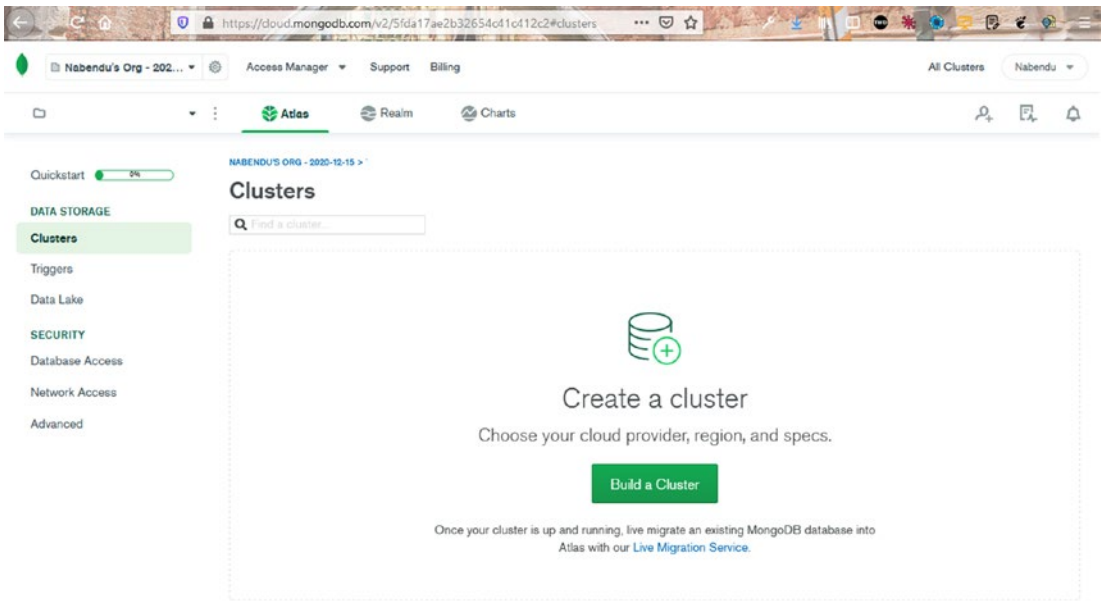


Figure 1-9. *Build a Cluster*

On the next screen, select the Free tier, as seen in Figure 1-10.

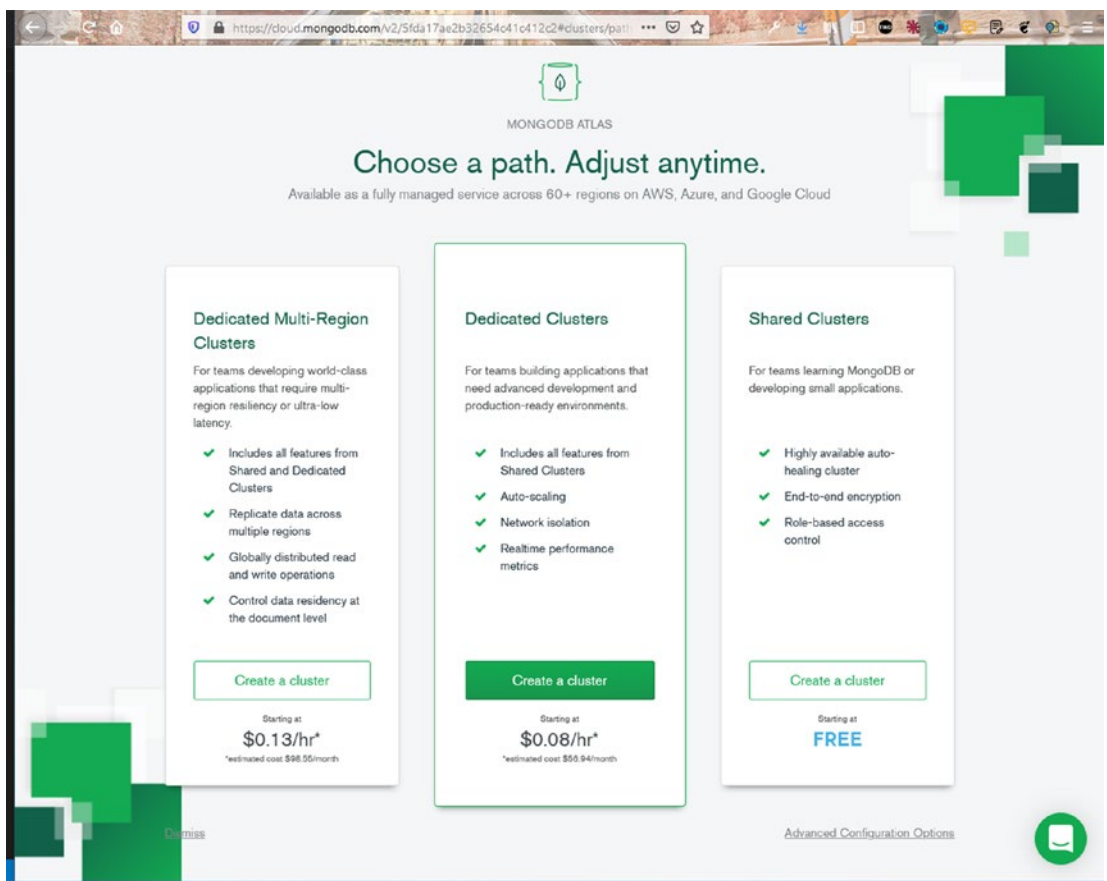


Figure 1-10. Free tier

On the next screen, you need to choose the AWS region in which to create the database. (I chose Mumbai because I live in India, and this gives me low latency.) Afterward, click the **Create Cluster** button, as seen in Figure 1-11.

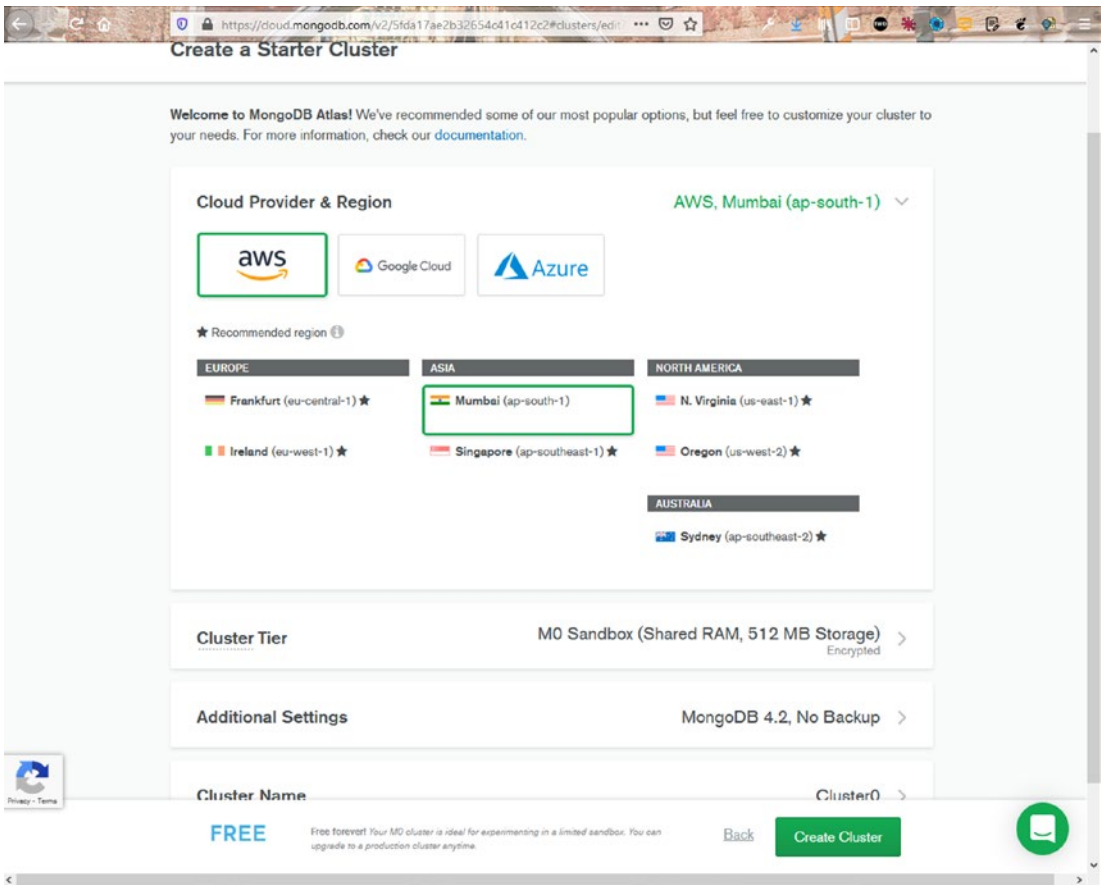


Figure 1-11. Choose region

The next screen shows that the cluster has been created, which takes time. You can go back and create your first API endpoint, as seen in Figure 1-12.

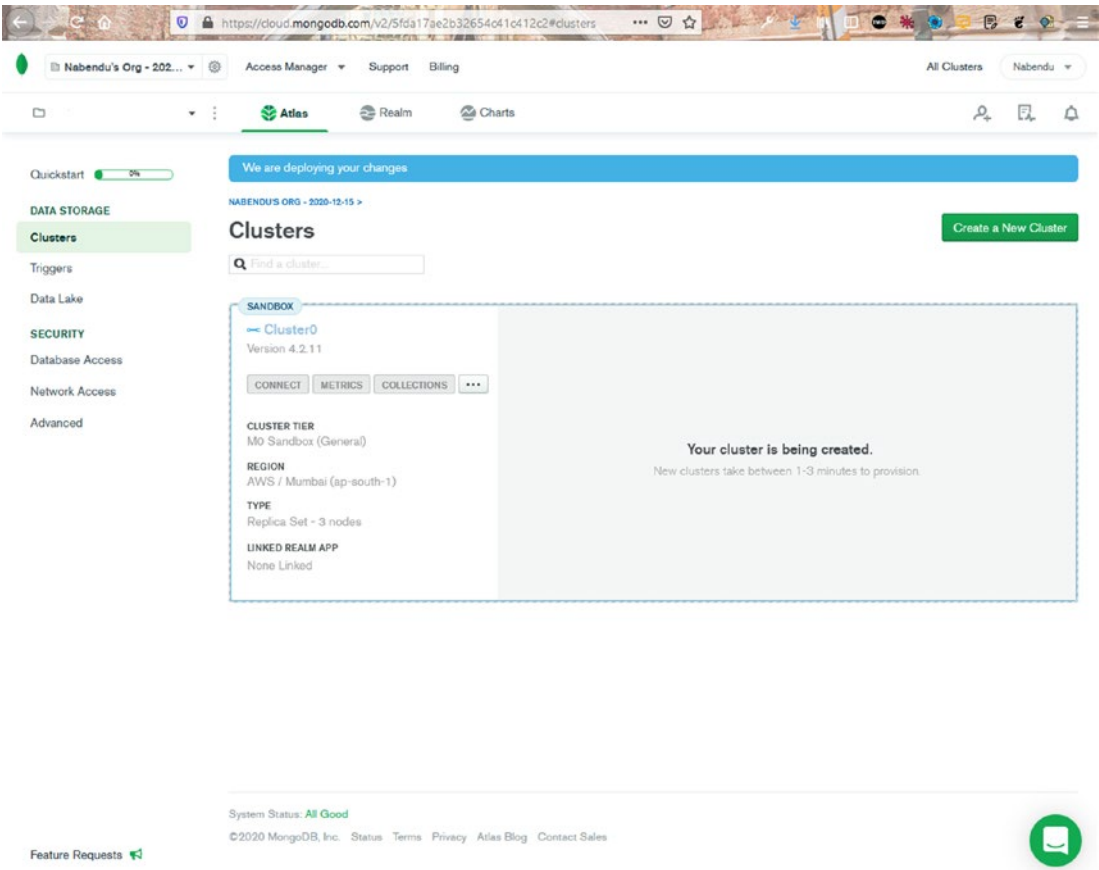


Figure 1-12. Cluster created

Database User and Network Access

To create a user in MongoDB, click the **Database Access** tab and then the **Add New Database User** button, as seen in Figure 1-13.

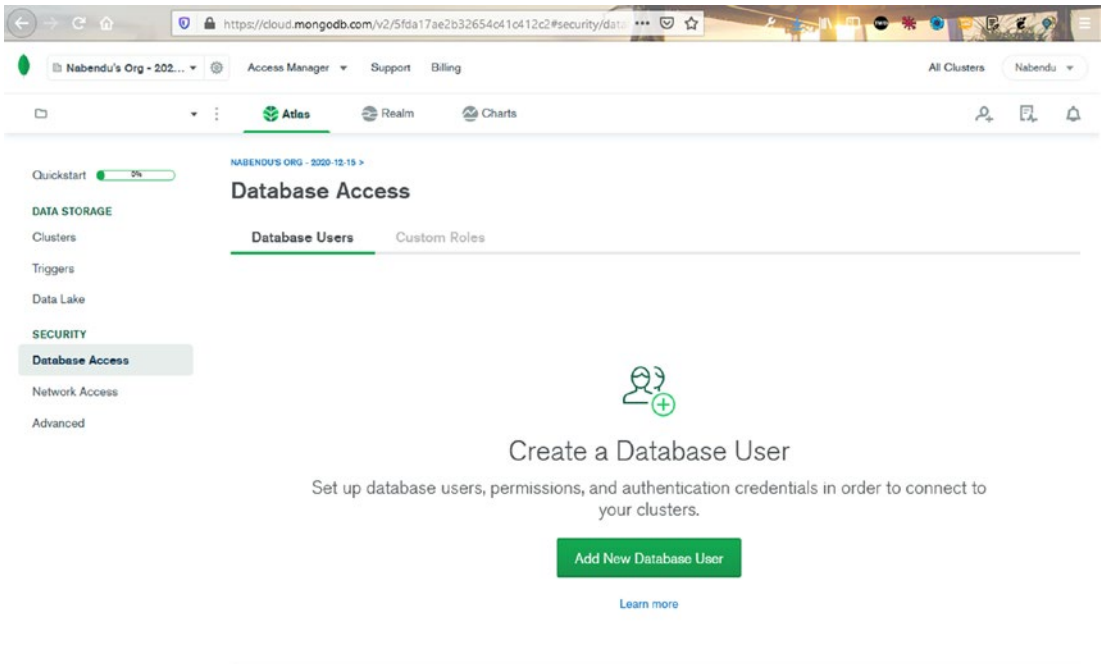


Figure 1-13. Create database user

On the next screen, you need to enter a username and a password, as seen in Figure 1-14. You must remember both. Next, scroll down and click the **Add User** button.

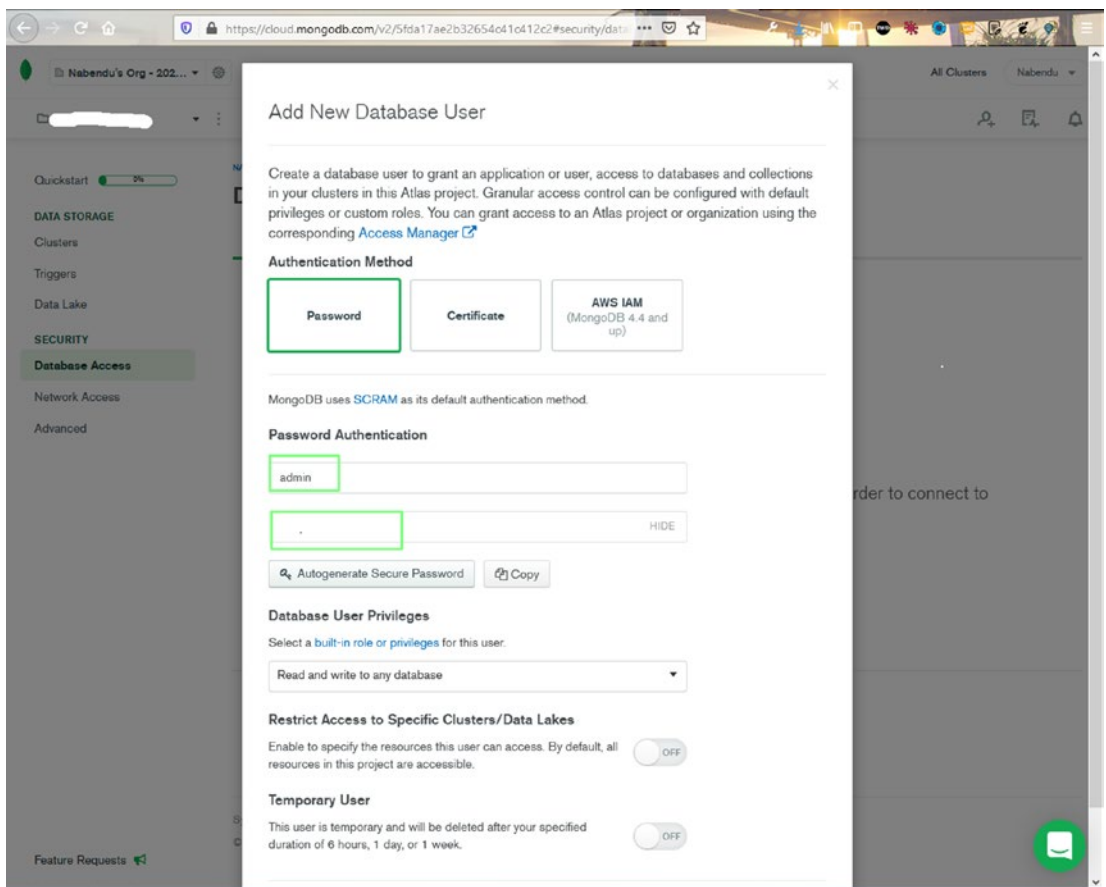


Figure 1-14. Add user

Next, go to the **Network Access** tab and click the **Add IP Address** button, as seen in Figure 1-15.

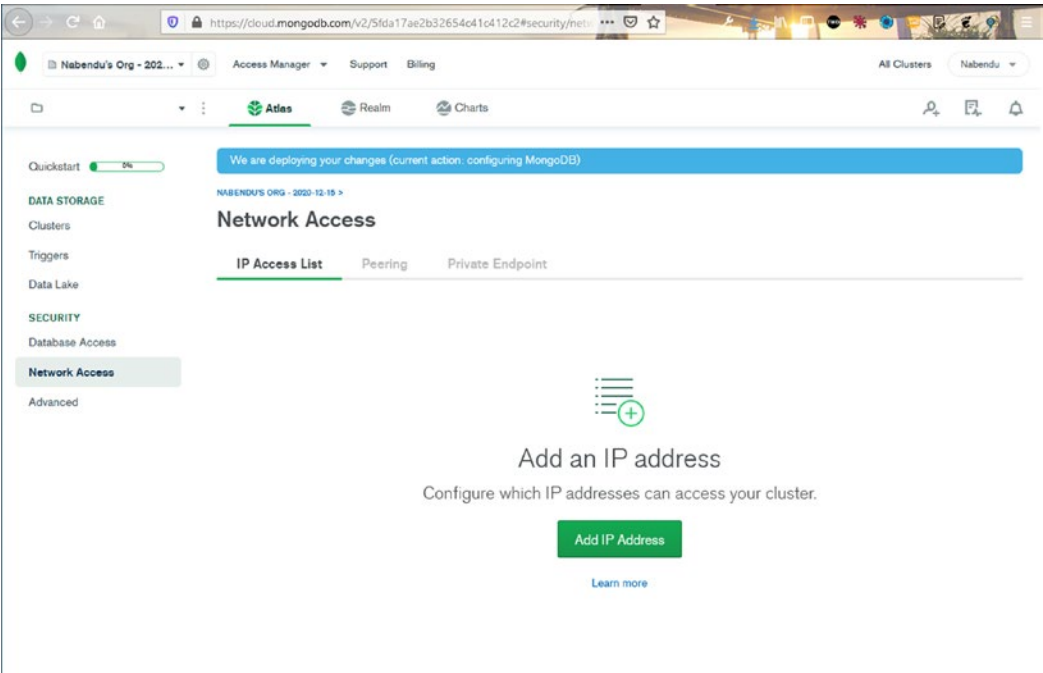


Figure 1-15. Network access

In the popup window, click the **ALLOW ACCESS FROM ANYWHERE** button and then click the **Confirm** button, as seen in Figure 1-16.

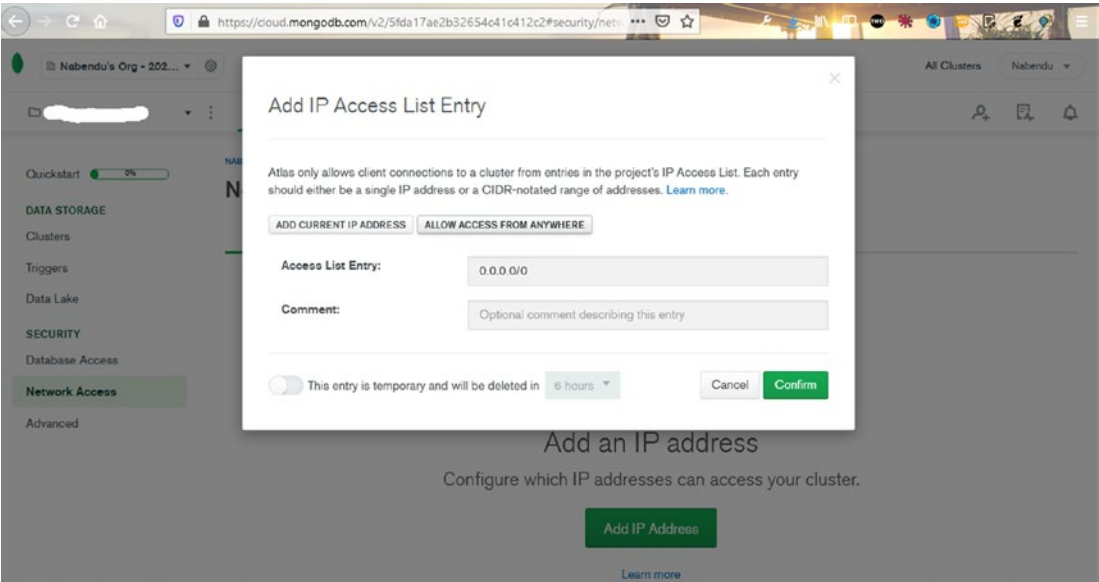


Figure 1-16. Allow access

Next, return to the **Cluster** tab and click the **CONNECT** button, which opens the popup window shown in Figure 1-17. Click the **Connect your application** tab.

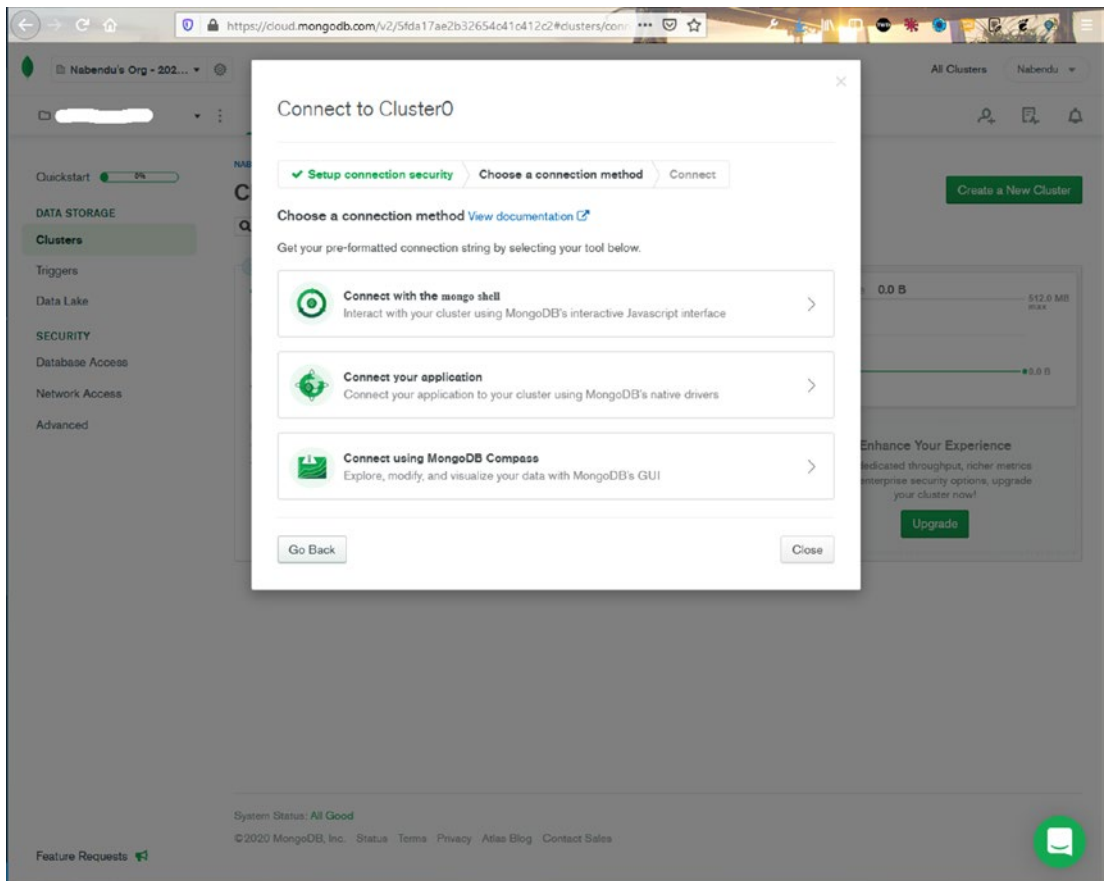


Figure 1-17. *Connect your application*

Copy the connection URL by clicking the **Copy** button, as seen in Figure 1-18.

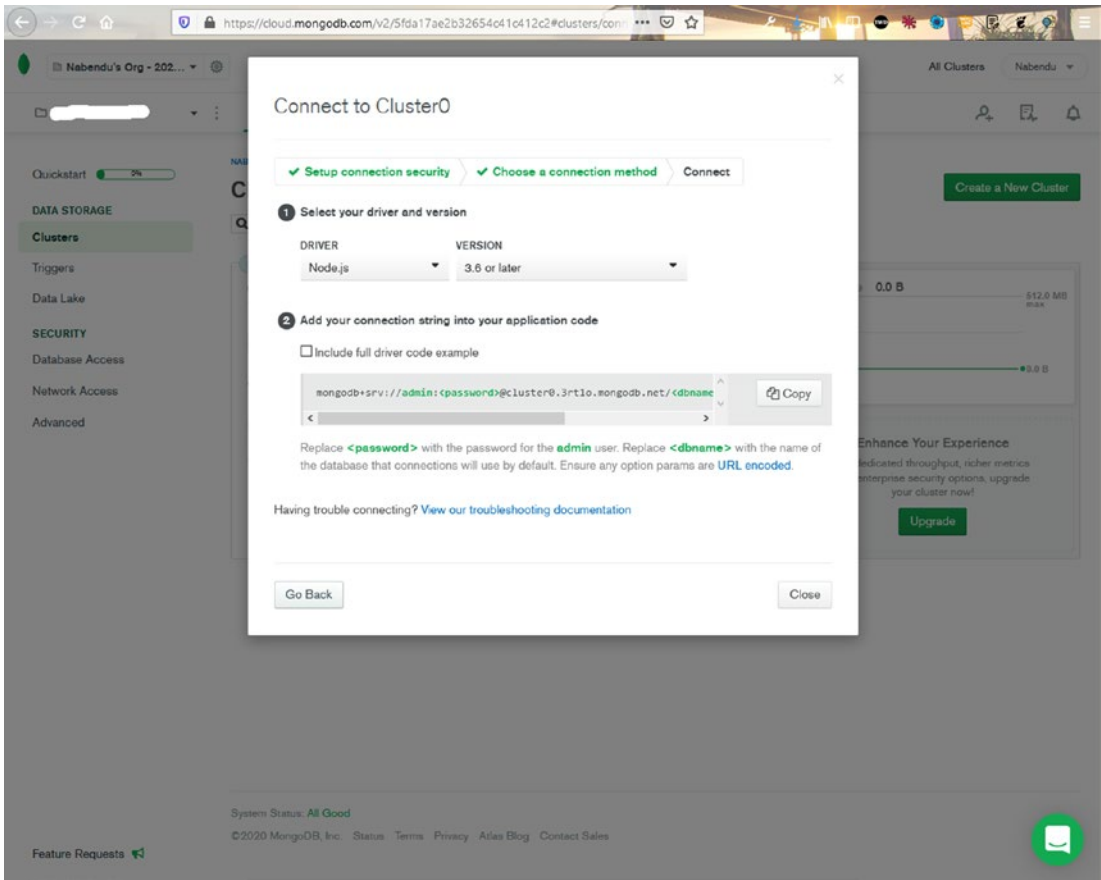


Figure 1-18. Connection string

Deploying the Back End to Heroku

Once you complete the back-end code, go to www.heroku.com to deploy the back end. Log in to your Heroku account, click the **New** drop-down menu, and then click the **Create new app** button, as seen in Figure 1-19. You can also do this from the command line using the Heroku CLI, but that is not covered here.

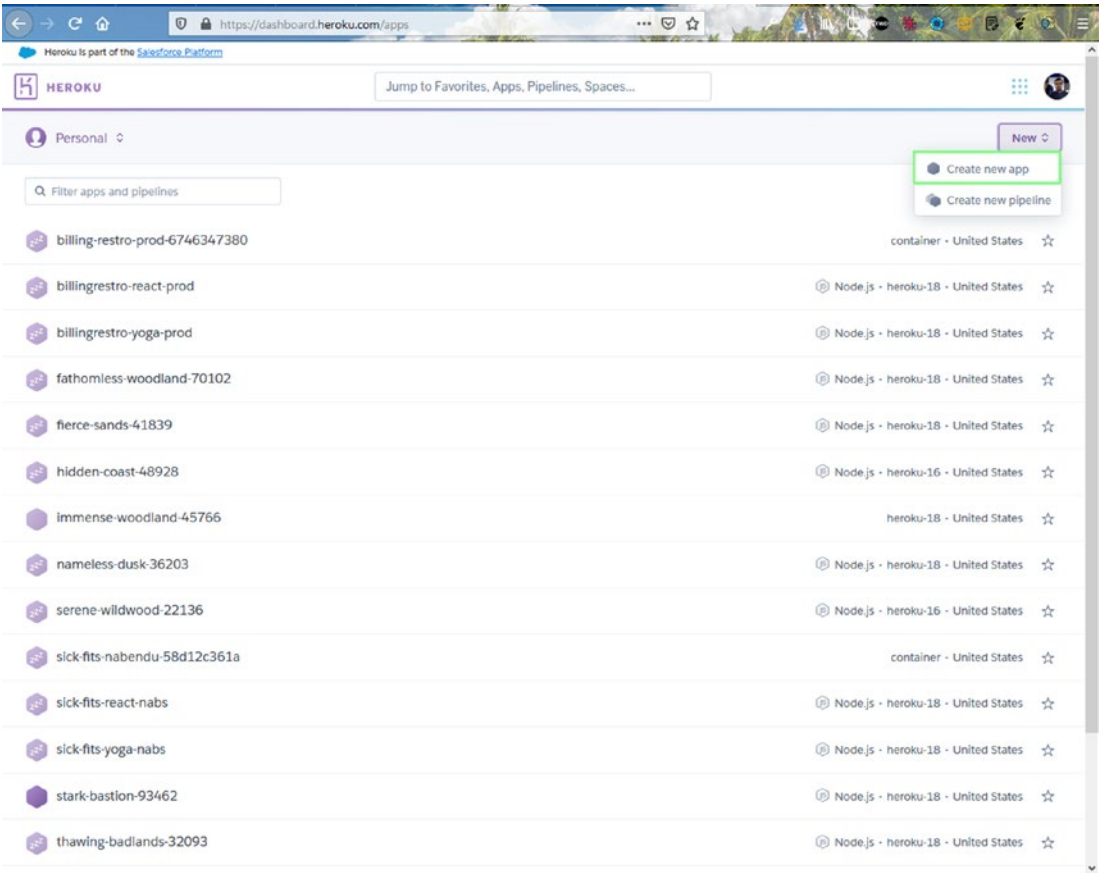


Figure 1-19. Heroku login

Next, name the app and click the **Create app** button, as seen in Figure 1-20.

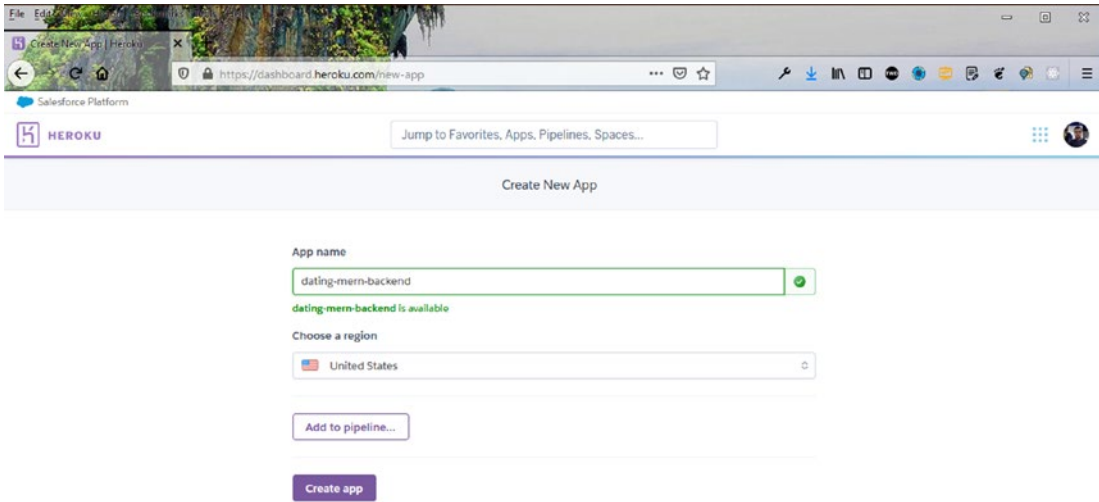


Figure 1-20. Heroku app name

The next screen shows all the commands to deploy your app, but you need the Heroku CLI. Click the link and follow the instructions to install it on your operating system, as seen in Figure 1-21.

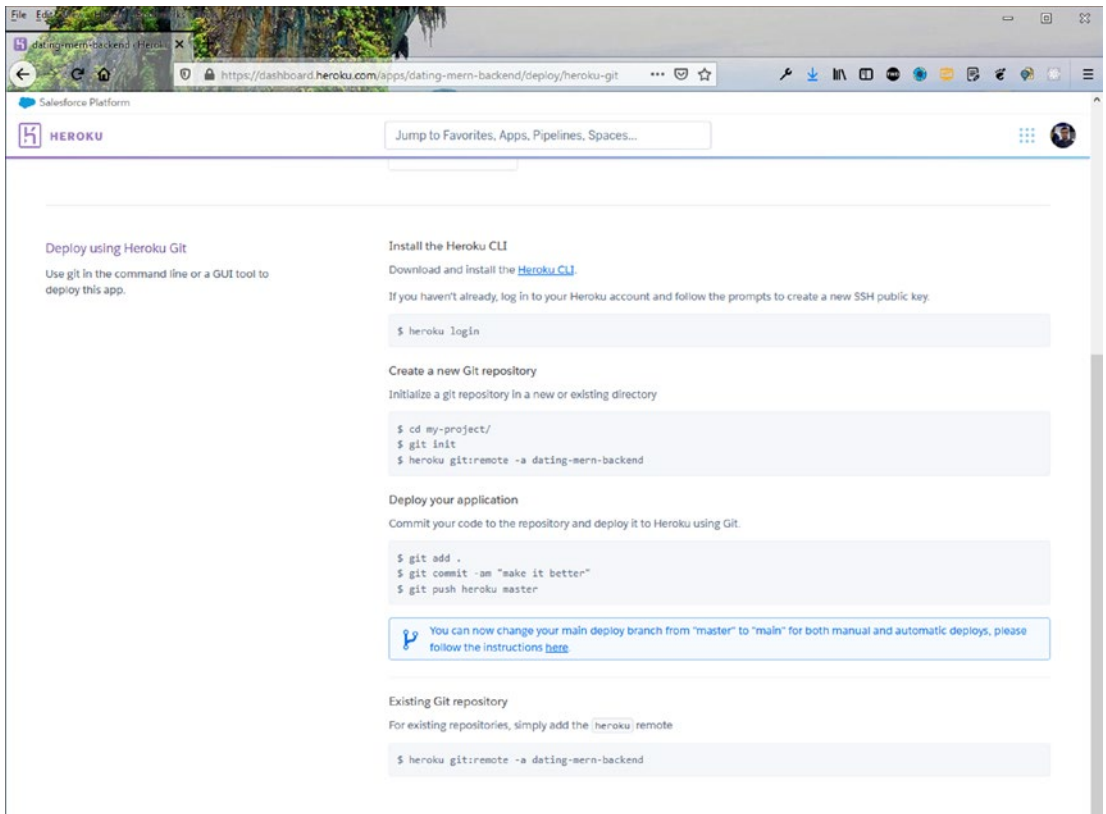


Figure 1-21. Heroku instructions

Run the `heroku login` command in the backend folder. You are asked for permission to open the browser. This command asks you to press any key to open in the browser.

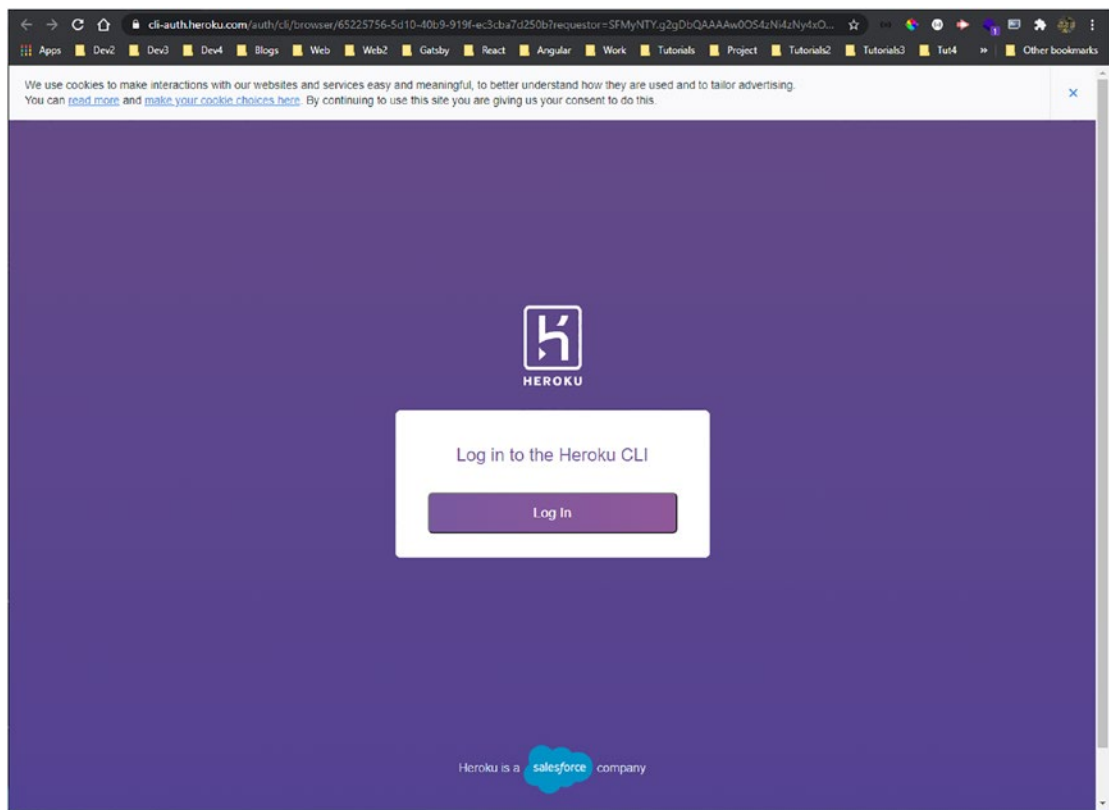


Figure 1-22.

Here, you can log in with your credentials, as seen in Figure 1-23.