

Download and install git program from internet

Check the installed version:

```
Last login: Sun Sep  3 10:03:25 on ttys000
prince@Princes-MacBook-Air ~ % git -v
git version 2.39.2 (Apple Git-143)
prince@Princes-MacBook-Air ~ %
```

## DAY 7

- *Git & GitHub*

- **Git is like a time machine for your code.**
- It is a tool that keeps a record of every version of your code, so you can always go back to a previous state if something goes wrong.
- **Install Git:** If you haven't already, download and install Git on your computer. You can get it from the official Git website:  
<https://git-scm.com/downloads>
- If you want to work with git in your project →
- Run **git init** inside the root folder of your project
- This command tells Git to start tracking changes in your project folder.

```
git status
```

- After making changes to your project (e.g., writing code), you'll want to save those changes in Git.

```
git add .
```

- The **.** means "add all changes." You can replace it with specific file names if needed.

- *gitignore*

- The **.gitignore** file is a special configuration file used in Git repositories to specify files and directories that should be ignored by Git.
- These ignored files and directories won't be tracked by Git or included in version control.
- **Create .gitignore File**

```
◆ .gitignore
1  # Ignore node_modules directory
2  node_modules/
3
4  # Other entries...
5
6
```

- This saves a snapshot of your project's current state.

```
git commit -m "Initial commit"
```

- If you want to collaborate with others or back up your code online, you can create a remote repository on platforms like GitHub
- **Link Your Local and Remote Repositories**
- If you created a remote repository, you can link it to your local one

```
git remote add origin https://github.com/yourusername/hotels.git
```

- **Push Changes to Remote**
- To send your local commits to the remote repository, use the git push command

```
git push -u origin master
```

- **Pull Changes**
- If you're collaborating with others, you can fetch their changes and merge them into your code using `git pull`.

## Create repository on github after creating your account.

Perform the following commands in the vs terminal to upload the project in the repository

**...or push an existing repository from the command line**

```
git remote add origin git@github.com:Prince-1501/node_hotels.git
git branch -M main
git push -u origin main
```

- *Host MongoDB database*

- **Now we are running locally MongoDB database.**
- All data operation is performed in a local database, so let's host our database server and make our DB online presence
- MongoDB Atlas provides a Free cluster for users where you can host your database for free.
- MongoDB Atlas offers a cloud-based platform for hosting MongoDB databases
- The free tier allows developers to explore and experiment with the database without incurring any costs.
- <https://www.mongodb.com/atlas/database>
- Create an account for free ( I already have an account )
- Show Step-by-step Process to host MongoDB Atlas

Create / login on monogdb atlas

structure  
scale, from  
ery.

arch, real-  
e with a

Company


Email\*

Password\*

☐ I agree to the **Terms of Service** and **Privacy Policy**.

Create your Atlas account

or

 Sign up with Google

Sign in

AtlasPrince's Org ...Access ManagerBilling

Project 0Data ServicesApp ServicesCharts

DEPLOYMENTDatabaseData LakeSERVICESDevice SyncTriggersData APIData FederationSearchStream Processing

PRINCE'S ORG - 2023-02-12 > PROJECT 0

Database Deployments

Find a database deployment...

Edit ConfigCreate

productsConnectView MonitoringBrowse Collections...

FREE

SHARED

Monitoring for products is Paused

Monitoring will automatically resume when you connect to your cluster.

Visit the documentation for more info.

VERSION	REGION	CLUSTER TIER	TYPE	BACKUPS	LINKED APP SERVICES	ATLAS SQL	ATLAS SEARCH
6.0.9	AWS / Mumbai (ap-south-1)	M0 Sandbox (General)	Replica Set - 3 nodes	Inactive	None Linked	Connect	Create Index

AtlasPrince's Org ...Access ManagerBilling

Project 0Data ServicesApp ServicesCharts

DEPLOYMENTDatabaseData LakeSERVICESDevice SyncTriggersData APIData FederationSearchStream Processing

PRINCE'S ORG - 2023-02-12 > PROJECT 0

Database Deployments

Find a database deployment...

Edit Config

Create a database

Choose your cloud provider, region, and specs.

Build a Database

Once your database is up and running, live migrate an existing MongoDB database into Atlas with our Live Migration Service.

AtlasPrince's Org ...Access ManagerBilling

Project 0Data ServicesApp ServicesCharts

DEPLOYMENTDatabaseData LakeSERVICESDevice SyncTriggersData APIData FederationSearchStream Processing

Create a database user using a username and password. Users will be given the read and write to any database privilege by default. You can update these permissions and/or create additional users later. Ensure these credentials are different to your MongoDB Cloud username and password. You can manage existing users via the Database Access Page.

Username

helloworld

Password

helloworld

Autogenerate Secure Password

Copy

Create User

Authentication Type

Password

EDIT

REMOVE

- ✓ Set up connection security
- ✓ Choose a connection method
- 3 Connect

## Connecting with MongoDB for VS Code

### 1. Install MongoDB for VS Code.

In [VS Code](#), open "Extensions" in the left navigation and search for "MongoDB for VS Code." Select the extension and click install.

### 2. In VS Code, open the Command Palette.

Click on "View" and open "Command Palette."

Search "MongoDB: Connect" on the Command Palette and click on "Connect with Connection String."

### 3. Connect to your MongoDB deployment.

Paste your connection string into the Command Palette.

```
mongodb+srv://helloworld:<password>@cluster0.cjppq8mh.mongodb.net/
```

Replace **<password>** with the password for the **helloworld** user.

When entering your password, make sure all special characters are [URL encoded](#). [↗](#)

### 4. Click "Create New Playground" in MongoDB for VS Code to get started.

[Learn more about Playgrounds](#) [↗](#)

#### RESOURCES

[Connect to MongoDB through VSCode](#) [↗](#)

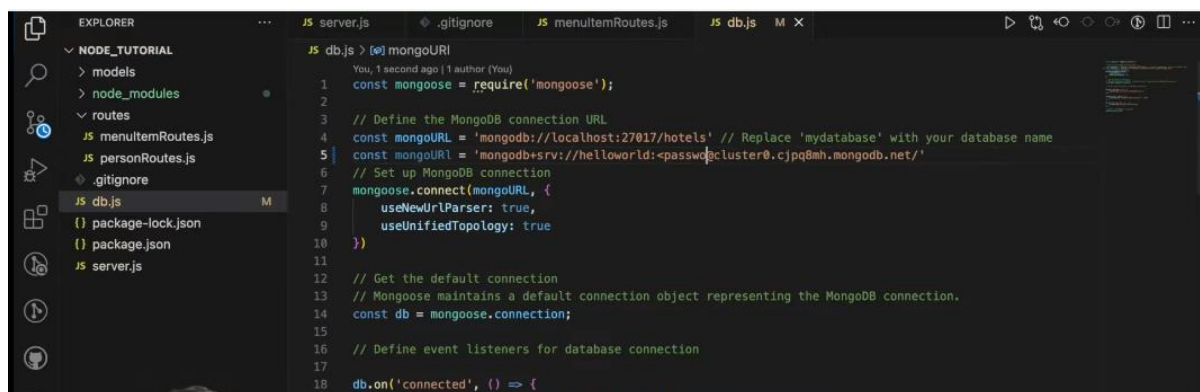
[Access your Database Users](#) [↗](#)

[Explore your data with playgrounds](#) [↗](#)

[Troubleshoot Connections](#) [↗](#)

```
mongodb+srv://helloworld:<password>@cluster0.cjppq8mh.mongodb.net/
```

Monodb url:



The screenshot shows the VS Code editor with the file explorer on the left. The file explorer shows a project named 'NODE\_TUTORIAL' with files like 'models', 'node\_modules', 'routes', 'menulitemRoutes.js', 'personRoutes.js', '.gitignore', 'db.js', 'package-lock.json', 'package.json', and 'server.js'. The 'db.js' file is selected and open in the editor. The code in 'db.js' shows the MongoDB connection string being used: 'mongodb+srv://helloworld:<password>@cluster0.cjppq8mh.mongodb.net/'. The code also shows the mongoose connection setup and event listeners for the database connection.

```
1 const mongoose = require('mongoose');
2
3 // Define the MongoDB connection URL
4 const mongoURL = 'mongodb://localhost:27017/hotels' // Replace 'mydatabase' with your database name
5 const mongoURL = 'mongodb+srv://helloworld:<password>@cluster0.cjppq8mh.mongodb.net/'
6 // Set up MongoDB connection
7 mongoose.connect(mongoURL, {
8   useNewUrlParser: true,
9   useUnifiedTopology: true
10 })
11
12 // Get the default connection
13 // Mongoose maintains a default connection object representing the MongoDB connection.
14 const db = mongoose.connection;
15
16 // Define event listeners for database connection
17
18 db.on('connected', () => {
```



- *Dotenv*

- The `dotenv` module in Node.js is used to manage configuration variables and sensitive information in your applications.
- It's particularly useful for keeping sensitive data like API keys, database connection strings, and other environment-specific configurations separate from your code.

```
npm install dotenv
```

- **Create a `.env` File**
- This is where you'll store your environment-specific configuration variables.
- format `VAR_NAME=value`.



```
PORT=3000
API_KEY=your-api-key
DB_CONNECTION_STRING=your-db-connection-string
```

- In your server file (usually the main entry point of your application), require and configure the `dotenv` module.

```
require('dotenv').config();
```

- **Access Configuration Variables:**

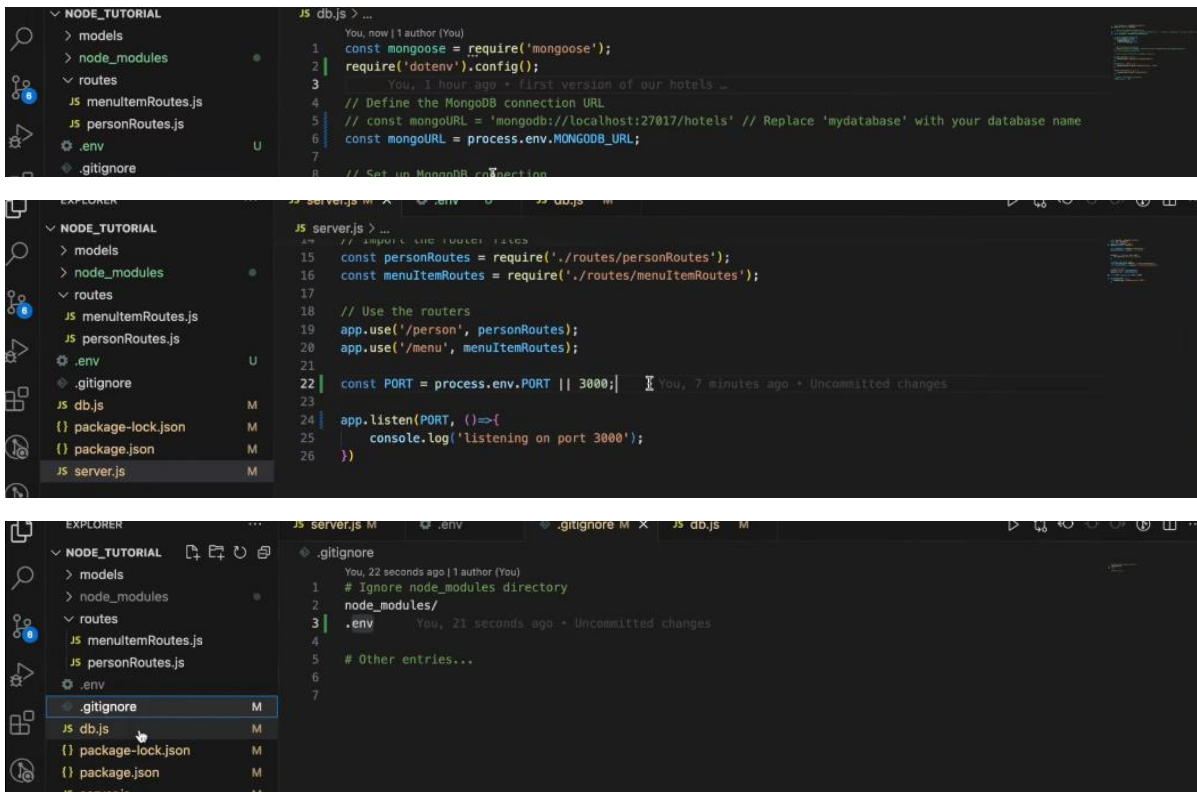
```
const port = process.env.PORT || 3000; // Use 3000 as a default if PORT is not defined
const apiKey = process.env.API_KEY;
const dbConnectionString = process.env.DB_CONNECTION_STRING;
```

- Remember to keep your `.env` file secure and never commit it to a public

version control system like Git, as it may contain sensitive information. Typically, you should include the `.env` file in your project's `.gitignore` file to prevent accidental commits.

- *Test MongoDB Cluster Postman*

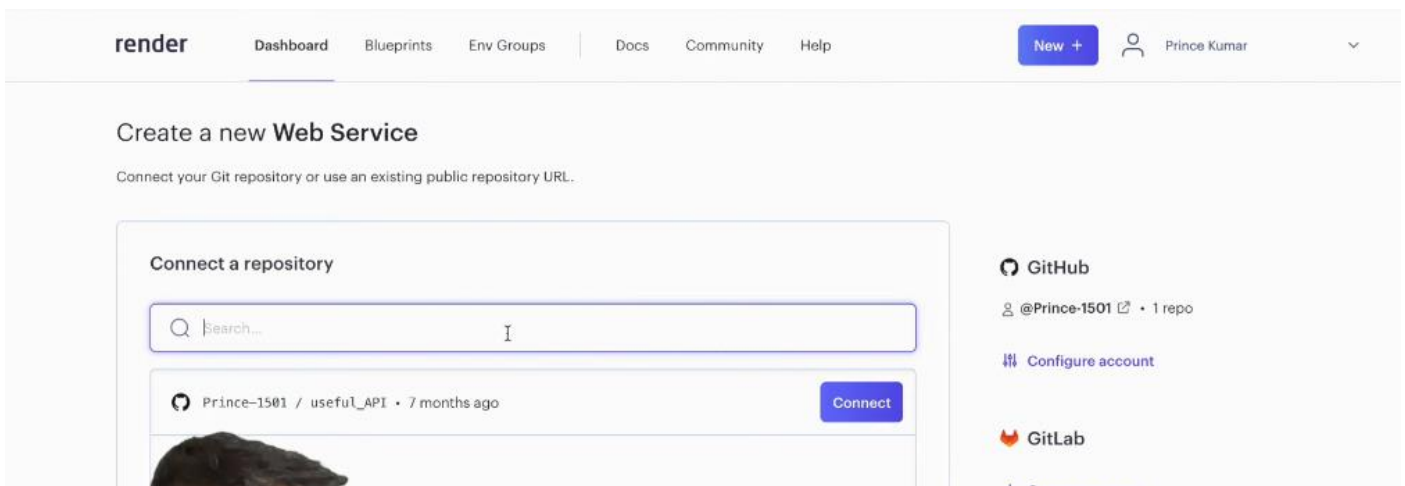
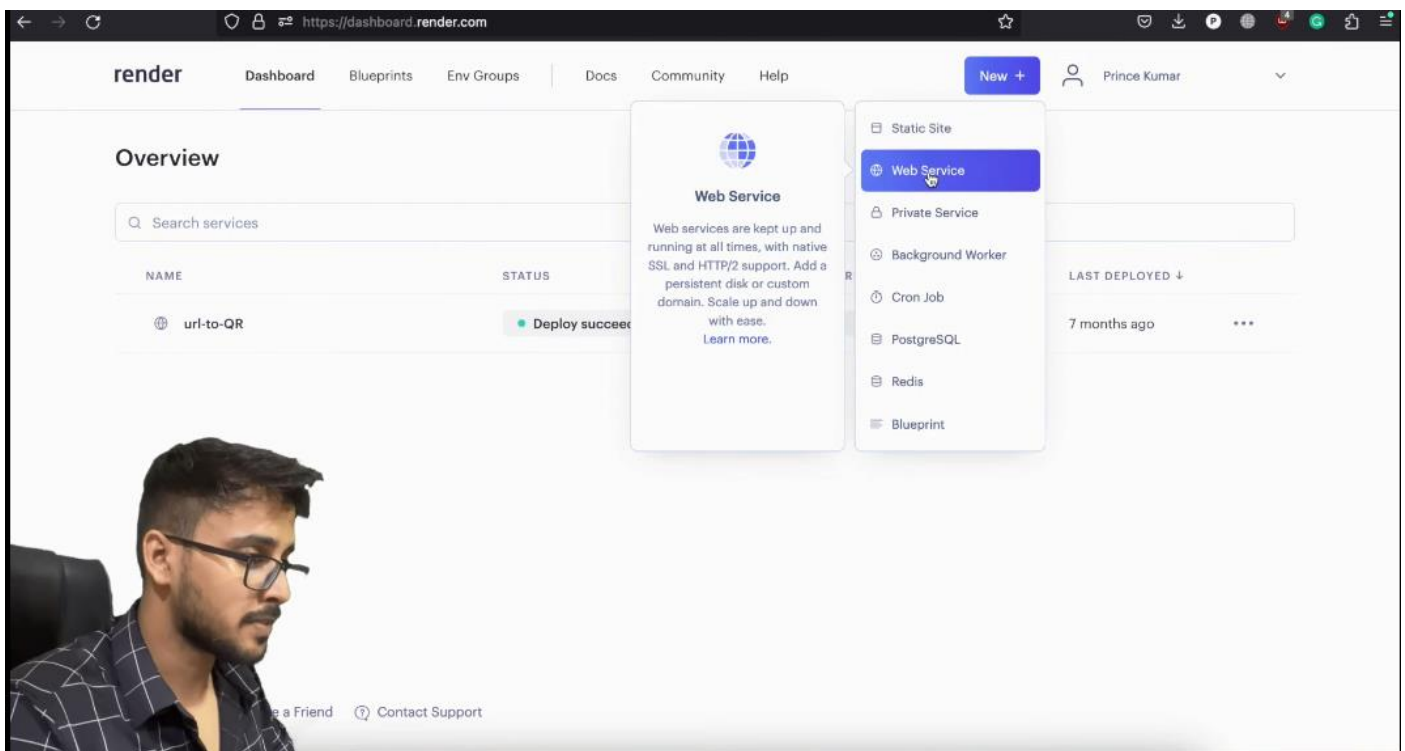
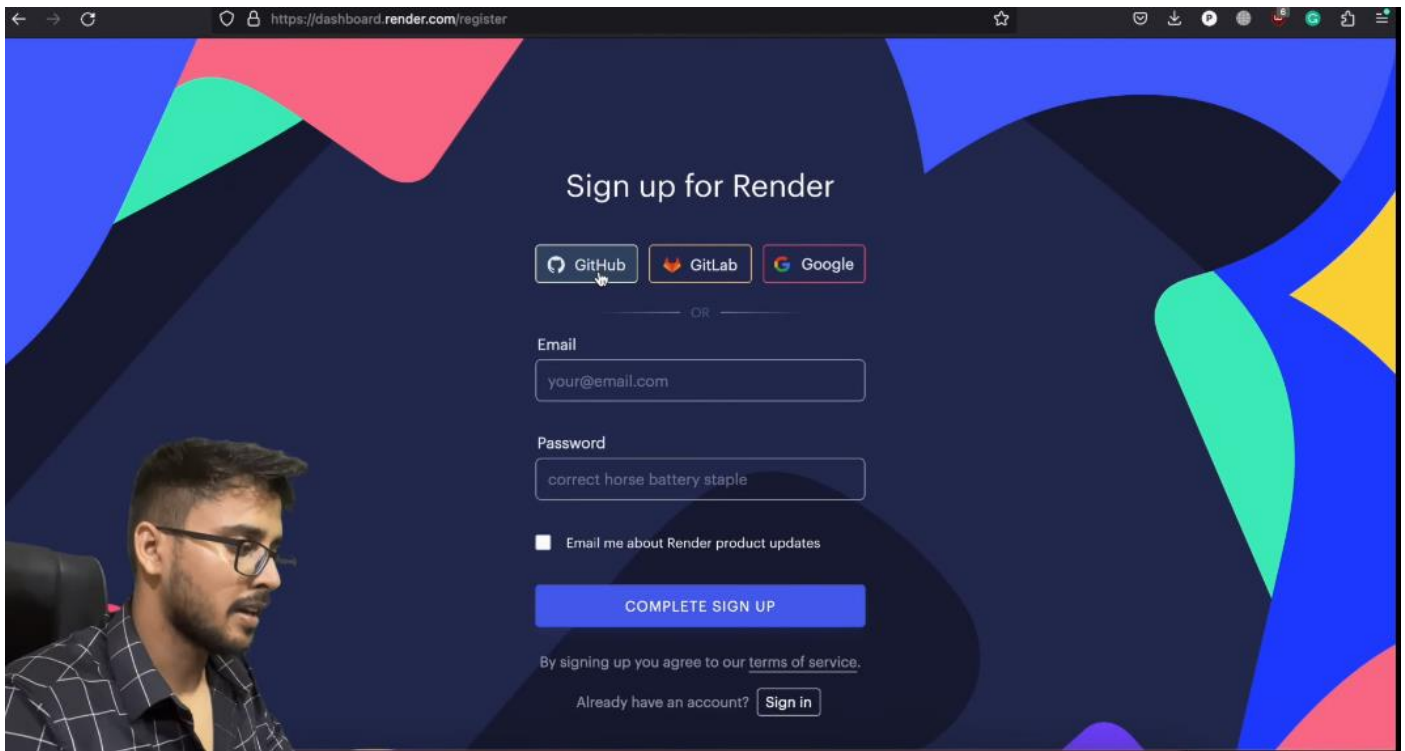
- Now we can test the MongoDB Cluster and check whether our data is present or not in the online DB



### • Host NodeJS Server

- Now we are going to host our server so that our Application or Endpoints is accessible to all the user over the Internet.
- We are using localhost and our endpoints are only accessible within our computer

- We have to make it publicly available, so there are lots of company who helps us to make our application run 24\*7
- Like, AWS, Google Cloud, etc. but these charge too much amount for our application
- So we are going to use some free services to host our nodeJS application, which lots of company provides for developer purpose.
- Like, Heroku, Netlify, Render, etc
-





render

DashboardBlueprintsEnv GroupsDocsCommunityHelp

New +Prince Kumar

you are deploying a web service for prince-1501/node\_hotels.

Name

A unique name for your web service.

hotels

Region

The region where your web service runs. Services must be in the same region to communicate privately and you currently have services running in Oregon.

Oregon (US West)

Branch

The repository branch used for your web service.

main

Root Directory

Default: /  
When you specify a root directory, the build command runs from your repository root, and the start command runs in the specified directory.

e.g., src

Runtime

The runtime for your web service.

Node

Build Command

This command runs in the root directory of your repository when a new version of your code is pushed, or when you deploy manually. It is typically a script that installs libraries, runs migrations, or compiles resources needed by your app.

\$ npm install

Start Command

This command runs in the root directory of your app and is responsible for starting the application. It is typically used to start the application. It can access environment variables by you in Render.

\$ node server.js

render

DashboardBlueprintsEnv GroupsDocsCommunityHelp

New +Prince Kumar

Render runs all your commands in the specified directory and ignores changes outside the directory.

Runtime

The runtime for your web service.

Node

Build Command

This command runs in the root directory of your repository when a new version of your code is pushed, or when you deploy manually. It is typically a script that installs libraries, runs migrations, or compiles resources needed by your app.

\$ npm install

Start Command

This command runs in the root directory of your app and is responsible for starting the application. It is typically used to start the application. It can access environment variables by you in Render.

\$ node server.js

Node index.js //for server app

Npm start //for react app

Please enter your payment information to select an instance type with higher limits.

render

DashboardBlueprintsEnv GroupsDocsCommunityHelp

New +Prince Kumar

Use environment variables to store API keys and other configuration values and secrets. You can access them in your code like regular environment variables, for example with `os.getenv()` in Python or `process.env` in Node.

key

value

Generate

Add Environment Variable

You can store secret files (like `.env` or `.npmrc` files and private keys) in Render. These files can be accessed during builds and in your code just like regular files.

All secret files you create are available to read at the root of your repo (or Docker context). They are also available to load by absolute path at `/etc/secrets/<filename>`.

Add Secret File

If network error occurs during fetching data from mongodb through this online url,

Add the ip address 0.0.0.0 in the whitelist of ip address on the monogdb atlas server through database->security->network access