Render Web Hosting Guide

This guide will walk you through deploying your Python web application on Render using the free Hobby plan.

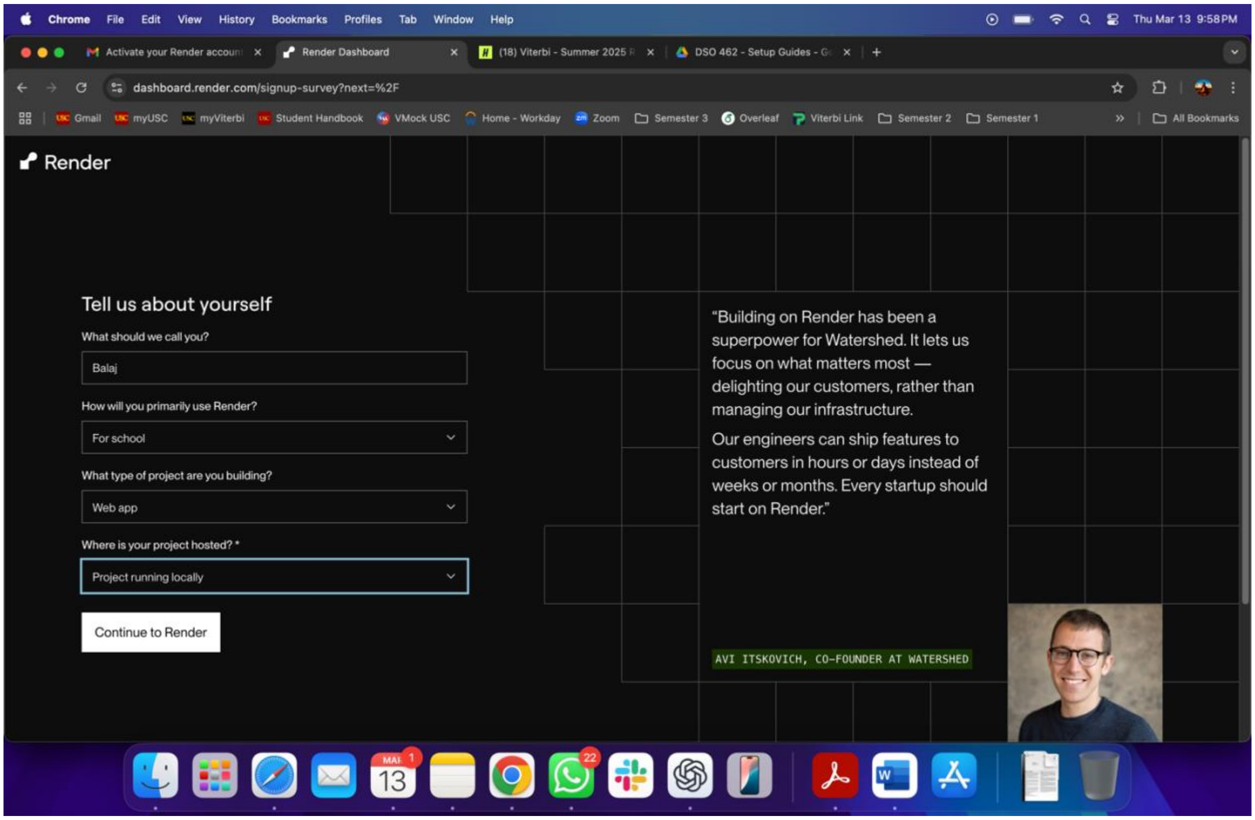
Step 0: Prepare Your Application for Deployment

Before deploying your application, ensure the following files are in your project directory:

1. **requirements.txt** – This file should list all the Python libraries your application depends on. Create it by running: `pip freeze > requirements.txt`
2. **Procfile** – This file tells Render how to run your application. Create a new file named Procfile (without any extension) and add the following line: `web: gunicorn app:app`
    a. Replace app:app if your main application file has a different name. The format is filename_without_extension:flask_app_variable_name.

Step 1: Sign Up and Set Up Render

1. Go to Render and sign in with your **GitHub** account.
2. Authorize Render to access your GitHub repositories.
3. Confirm your email (this should match your GitHub email).
4. Verify your email by clicking the confirmation link sent to your inbox.
5. In the **“Tell Us About Yourself”** section, select:
    - Enter your name
    - **Primary use:** School
    - **Building:** A web app
    - **Current hosting:** Locally



Tell us about yourself

What should we call you?

Balaj

How will you primarily use Render?

For school

What type of project are you building?

Web app

Where is your project hosted? *

Project running locally

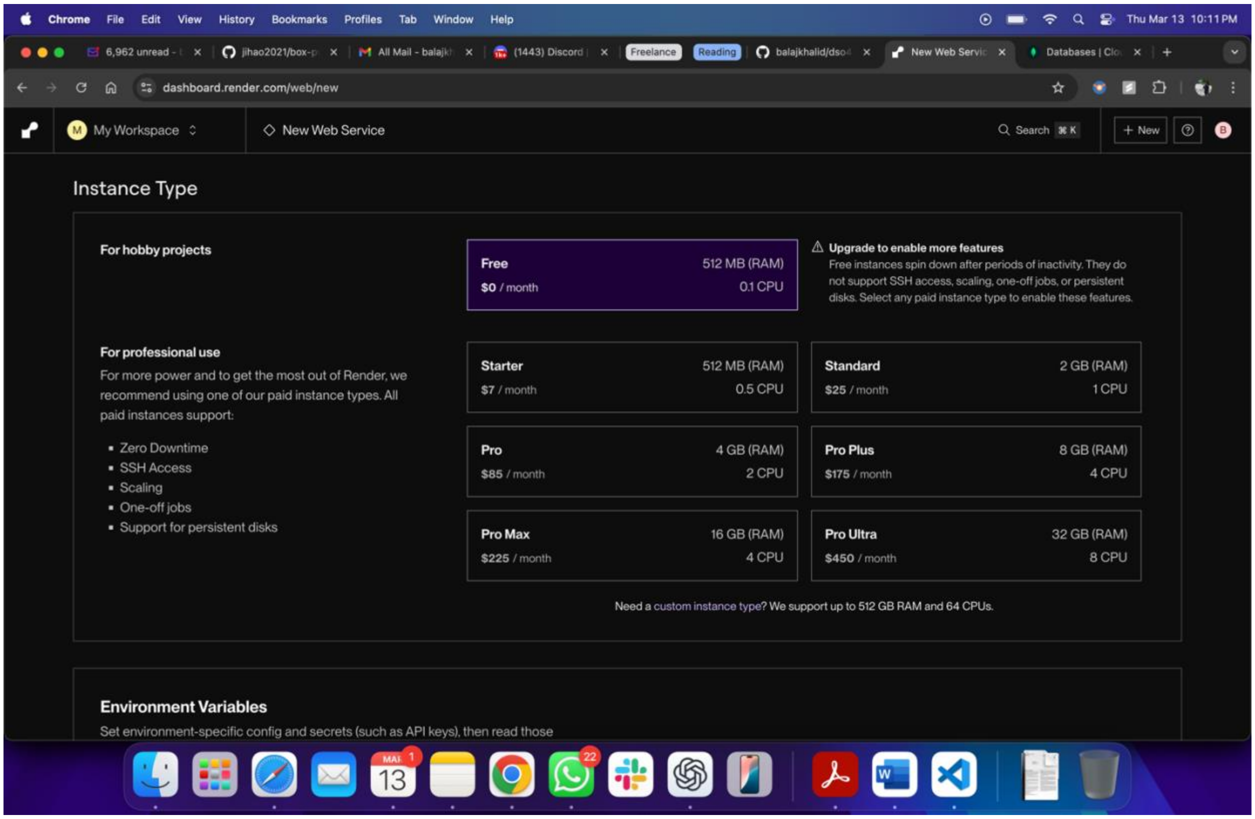
"Building on Render has been a superpower for Watershed. It lets us focus on what matters most — delighting our customers, rather than managing our infrastructure.

Our engineers can ship features to customers in hours or days instead of weeks or months. Every startup should start on Render."

AVI ITSKOVICH, CO-FOUNDER AT WATERSHED

6. Click **Continue to Render**.

Step 2: Create a New Web Service



Instance Type

For hobby projects

Free	512 MB (RAM)	$0 / month	0.1 CPU

⚠ Upgrade to enable more features
Free instances spin down after periods of inactivity. They do not support SSH access, scaling, one-off jobs, or persistent disks. Select any paid instance type to enable these features.

For professional use
For more power and to get the most out of Render, we recommend using one of our paid instance types. All paid instances support:

- Zero Downtime
- SSH Access
- Scaling
- One-off jobs
- Support for persistent disks

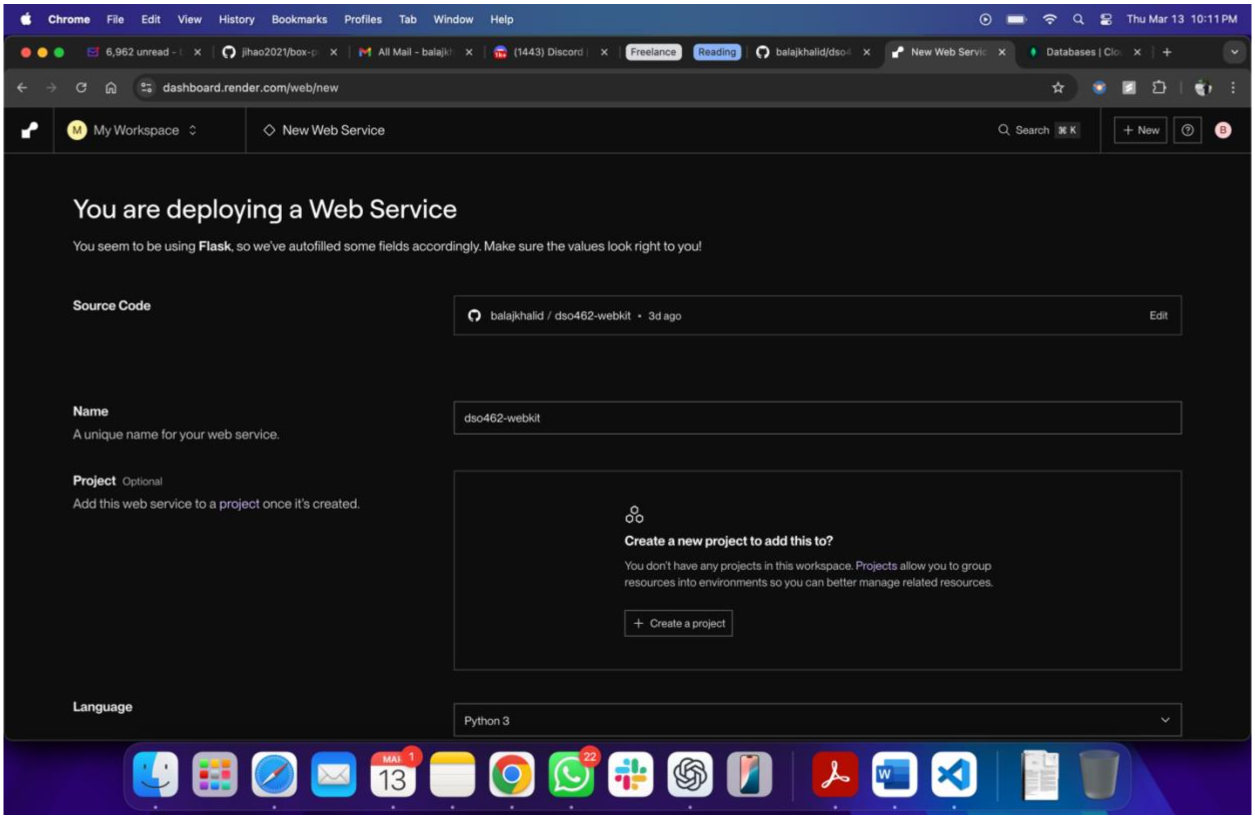
Starter	512 MB (RAM)	$7 / month	0.5 CPU
Standard	2 GB (RAM)	$25 / month	1 CPU
Pro	4 GB (RAM)	$85 / month	2 CPU
Pro Plus	8 GB (RAM)	$175 / month	4 CPU
Pro Max	16 GB (RAM)	$225 / month	4 CPU
Pro Ultra	32 GB (RAM)	$450 / month	8 CPU

Need a custom instance type? We support up to 512 GB RAM and 64 CPUs.

Environment Variables
Set environment-specific config and secrets (such as API keys), then read those

1. Select **Hobby (Free Plan)** and click **Select Plan**.
2. Under **Service Type**, choose **Web Services**.
3. In the **Configure** tab:
   - Connect your GitHub account.
   - Select the repository where your web app code is stored. (You can choose “All repositories” or manually select one.)
   - Click **Next**.

Step 3: Configure Deployment Settings



You are deploying a Web Service

You seem to be using **Flask**, so we've autofilled some fields accordingly. Make sure the values look right to you!

Source Code: balajkhalid / dso462-webkit • 3d ago — Edit

Name: dso462-webkit

A unique name for your web service.

Project Optional

Add this web service to a project once it's created.

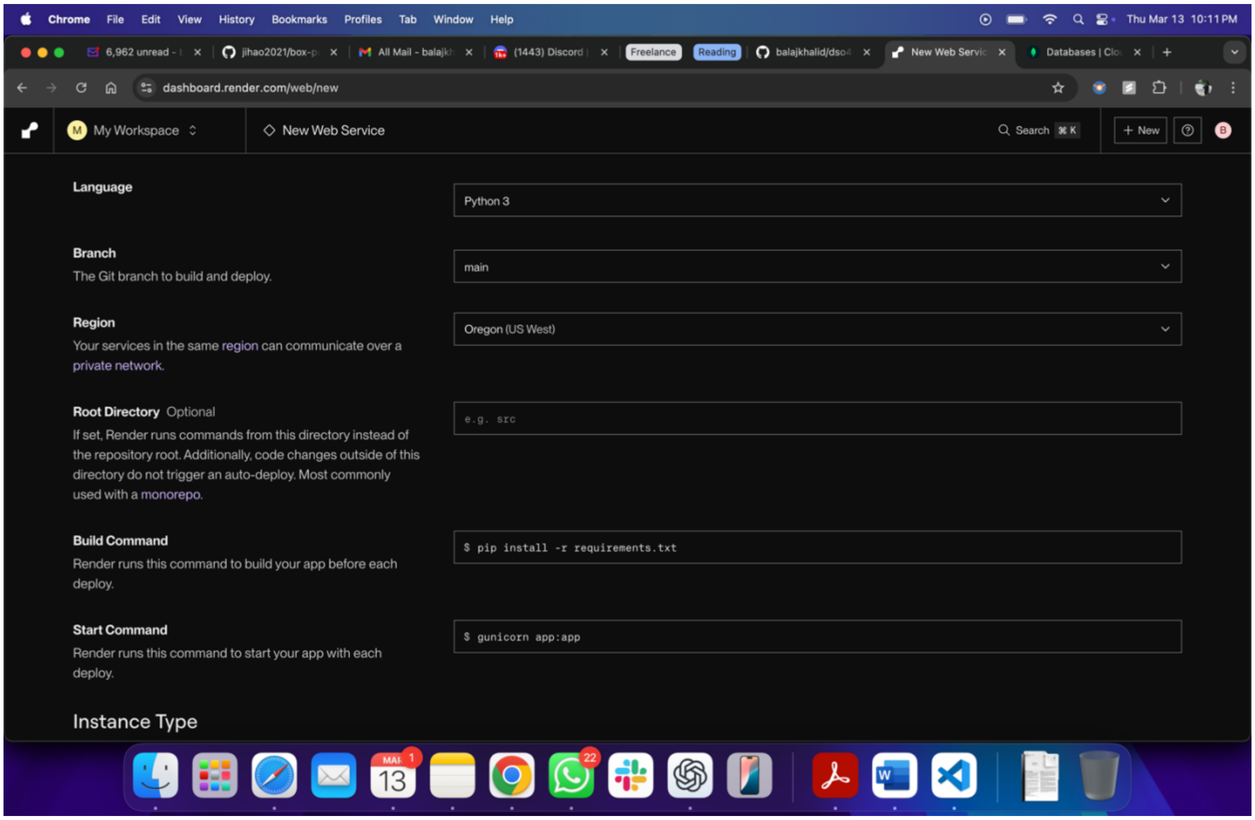
Create a new project to add this to?

You don't have any projects in this workspace. Projects allow you to group resources into environments so you can better manage related resources.

+ Create a project

Language: Python 3

1. **Service Name:** Choose a name (e.g., dso462-webkit).
2. **Runtime:** Select **Python 3**.
3. **Branch:** Choose **main**.
4. **Region:** Select **US West** (or the closest region for better performance).
5. **Build Command:** `pip install -r requirements.txt`
6. **Start Command:** `gunicorn app:app`
    a. • Ensure your Flask app's entry point is correct.
7. **Instance Type:** Select **Free**.



Language
Python 3

Branch
The Git branch to build and deploy.
main

Region
Your services in the same region can communicate over a private network.
Oregon (US West)

Root Directory Optional
If set, Render runs commands from this directory instead of the repository root. Additionally, code changes outside of this directory do not trigger an auto-deploy. Most commonly used with a monorepo.
e.g. src

Build Command
Render runs this command to build your app before each deploy.

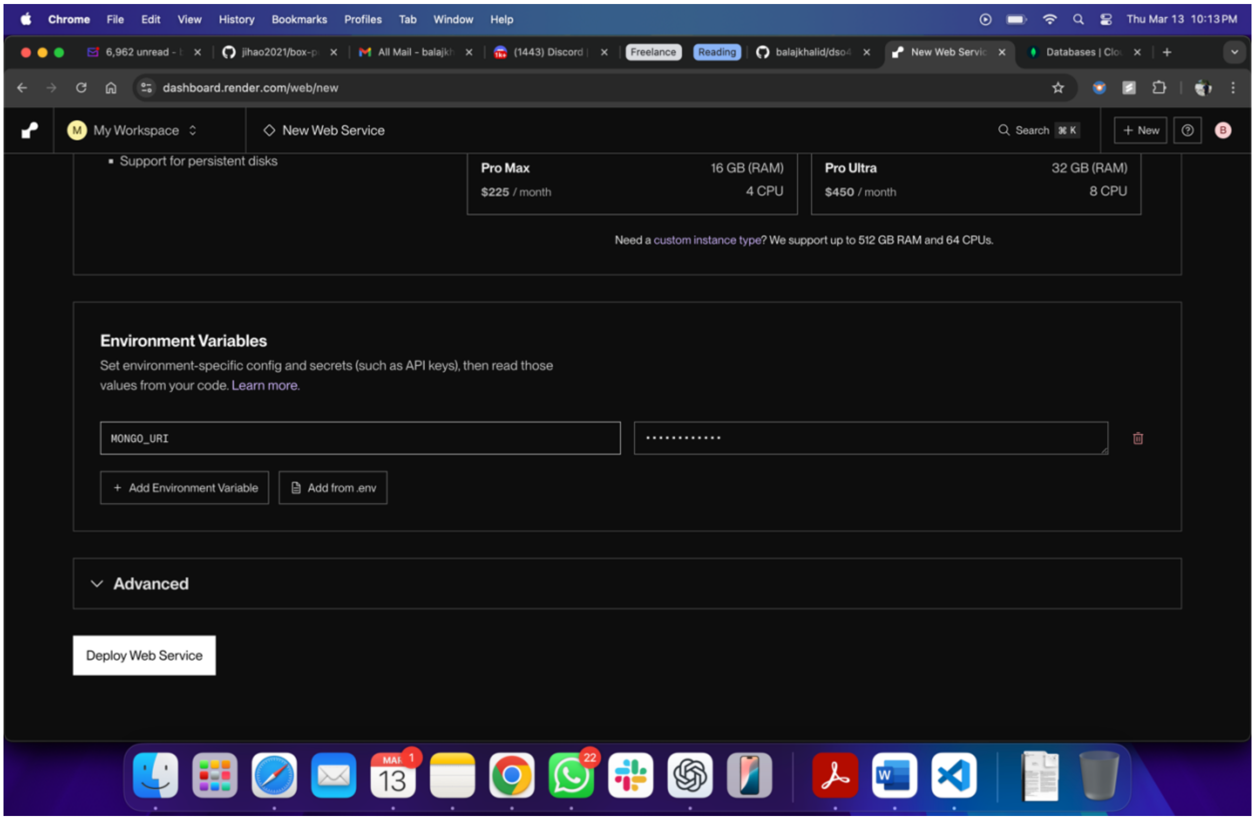
```
$ pip install -r requirements.txt
```

Start Command
Render runs this command to start your app with each deploy.

```
$ gunicorn app:app
```

Instance Type

Step 4: Add Environment Variables



- Support for persistent disks

Pro Max	16 GB (RAM)	Pro Ultra	32 GB (RAM)
$225 / month	4 CPU	$450 / month	8 CPU

Need a custom instance type? We support up to 512 GB RAM and 64 CPUs.

Environment Variables
Set environment-specific config and secrets (such as API keys), then read those values from your code. Learn more.

MONGO_URI	············

1. Under **Environment Variables**, add your MongoDB connection string (MONGODB_URI).
2. Do not enclose it in quotes ("").
3. Alternatively, you can upload an .env file containing your environment variables.

Step 5: Deploy Your Web Server

1. Click **Deploy Web Server**.
2. Wait for Render to build and deploy your app.
3. Once deployment is complete, you will see a **Live URL** where your application is hosted.

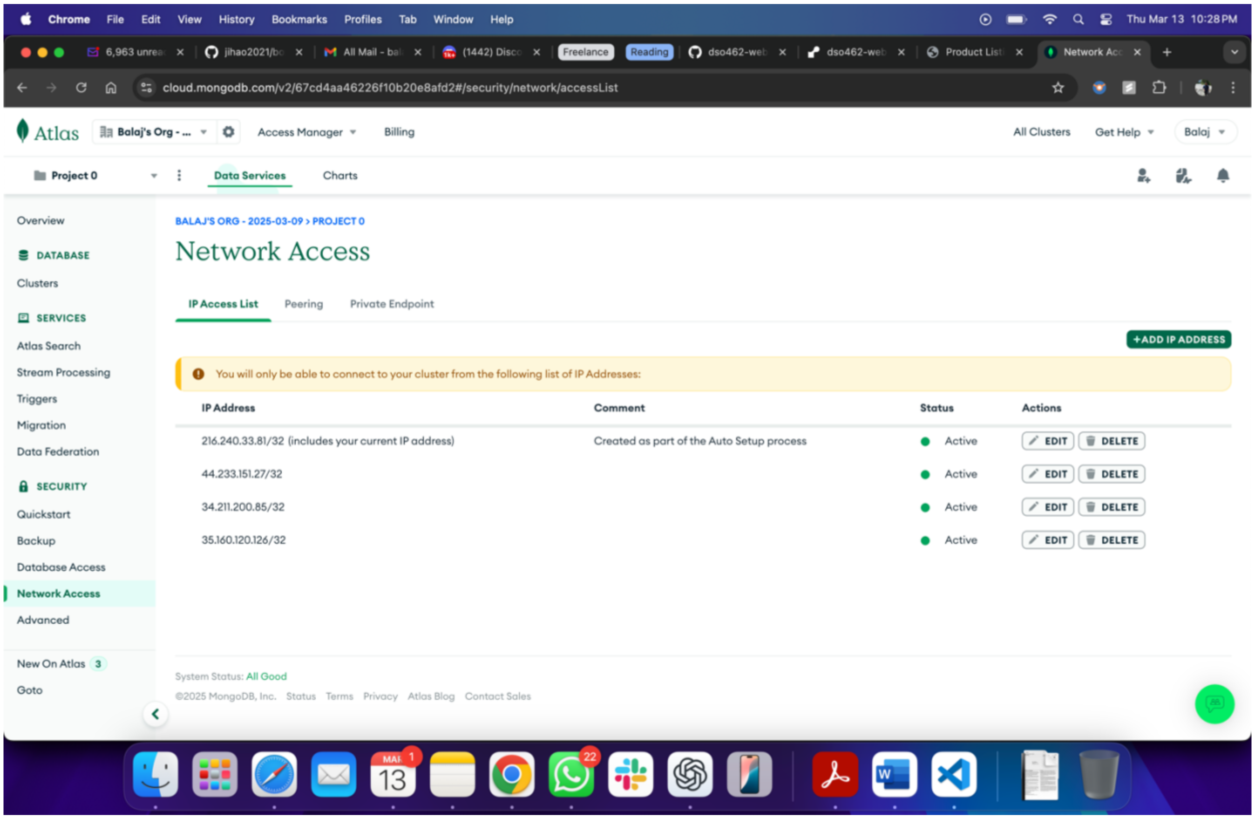
Step 6: Whitelist Render's IPs in MongoDB

To allow your Render-hosted app to access your MongoDB database, you need to whitelist Render's IP addresses.

1. On Render, click on the Connect button to view a list of IP addresses used by your service.
2. Go to MongoDB Atlas and whitelist these IPs:

    - Navigate to Project → Network Access.
    - Click on Add IP Address and enter each Render IP address separately.
    - Click Save to apply the changes.

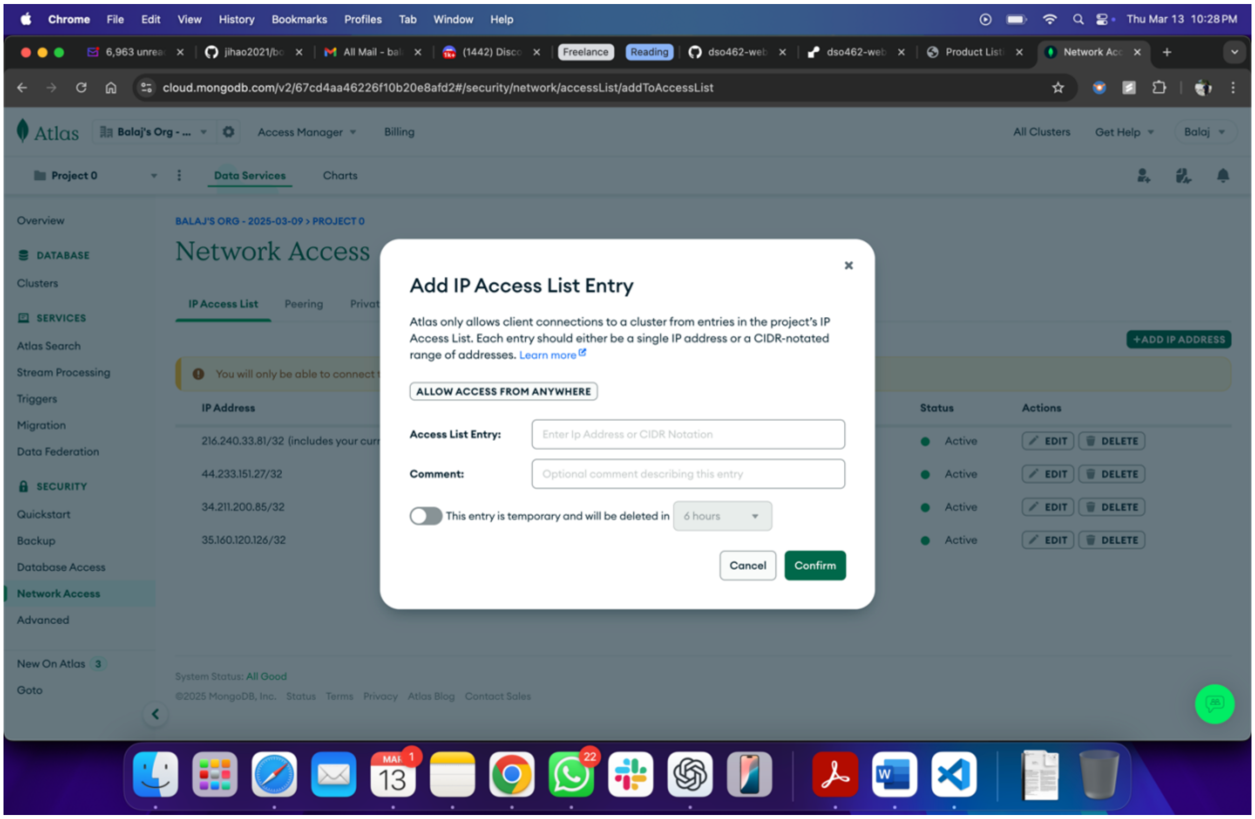
Once this is done, MongoDB will allow connections from your Render-hosted application.



Network Access

You will only be able to connect to your cluster from the following list of IP Addresses:

IP Address	Comment	Status	Actions
216.240.33.81/32 (includes your current IP address)	Created as part of the Auto Setup process	● Active	EDIT DELETE
44.233.151.27/32		● Active	EDIT DELETE
34.211.200.85/32		● Active	EDIT DELETE
35.160.120.126/32		● Active	EDIT DELETE


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Thu Mar 13 10:28 PM
cloud.mongodb.com/v2/67cd4aa46226f10b20e8afd2#/security/network/accessList/addToAccessList
ALLOW ACCESS FROM ANYWHERE
Cancel
Confirm

Add IP Access List Entry

Atlas only allows client connections to a cluster from entries in the project's IP Access List. Each entry should either be a single IP address or a CIDR-notated range of addresses. Learn more

Access List Entry: Enter Ip Address or CIDR Notation

Comment: Optional comment describing this entry

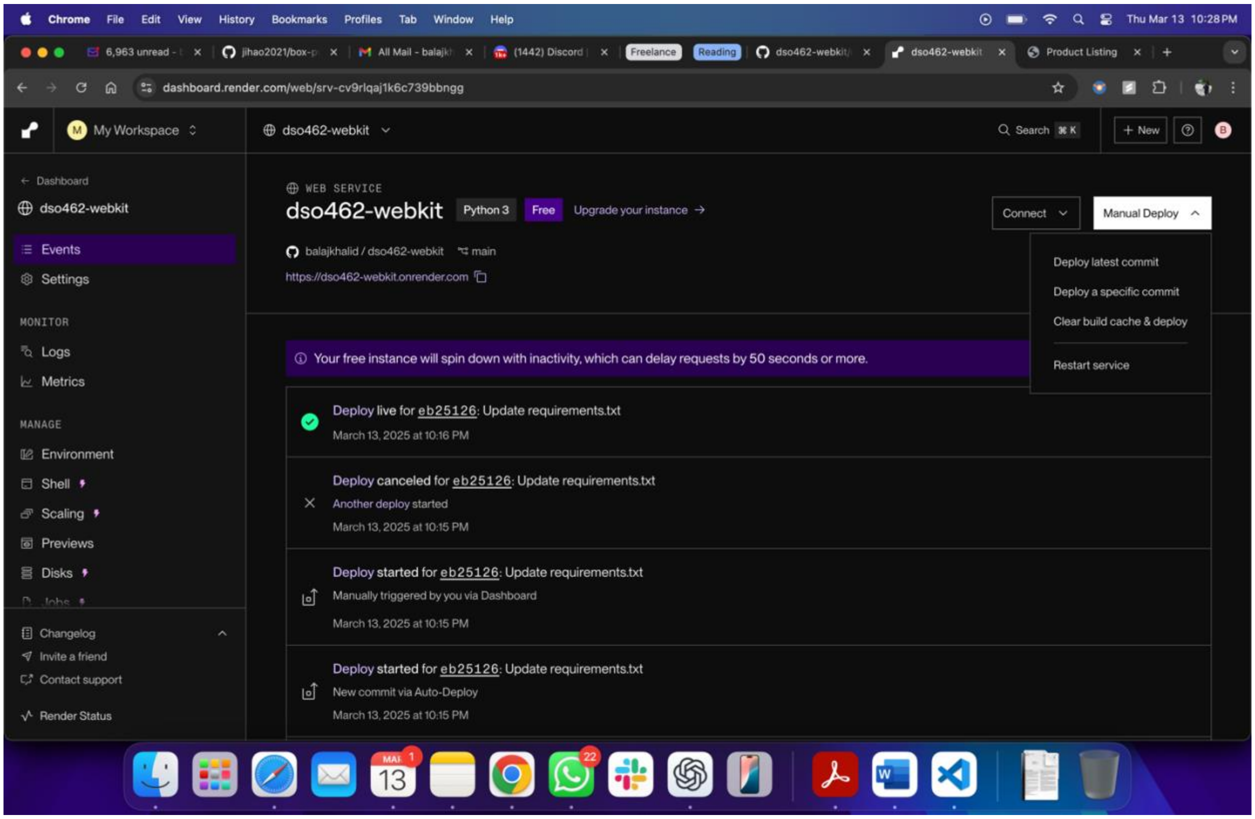
This entry is temporary and will be deleted in 6 hours

Future Updates: Deploying New Code Changes

If you want to update your site with new code in the future:

1. Go to Render and select your project.
2. Navigate to Manual Deploy.
3. Click Deploy Latest Commit.
4. Wait a few minutes for the changes to be applied.

Your app will be updated with the latest code from your GitHub repository.



Your free instance will spin down with inactivity, which can delay requests by 50 seconds or more.

Deploy live for eb25126: Update requirements.txt
March 13, 2025 at 10:16 PM

Deploy canceled for eb25126: Update requirements.txt
Another deploy started
March 13, 2025 at 10:15 PM

Deploy started for eb25126: Update requirements.txt
Manually triggered by you via Dashboard
March 13, 2025 at 10:15 PM

Deploy started for eb25126: Update requirements.txt
New commit via Auto-Deploy
March 13, 2025 at 10:15 PM