

Deployment

Helen Yang

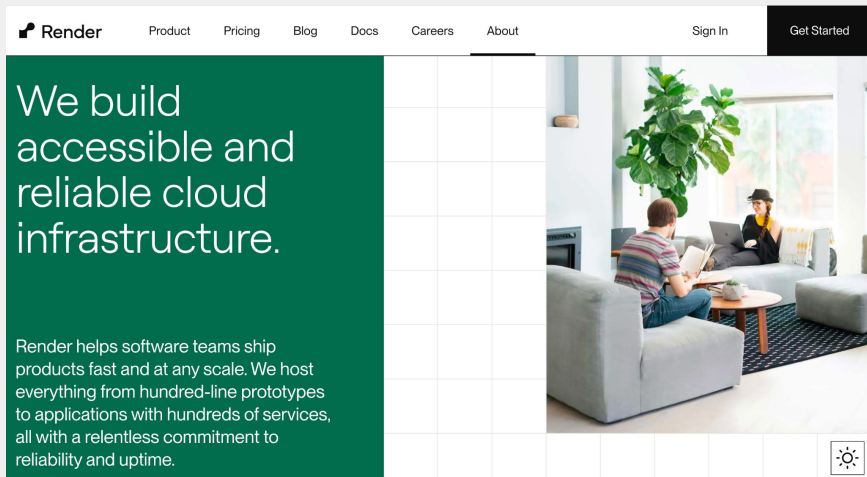
So you've made a web app...
Now what?

Deployment

- We need to deploy! What is deployment?
 - Making your web app accessible to the world!
 - Really just getting somebody else to run your server.js
 - `npm run start`
 - Now any time someone wants to access your site, they can visit a url, which will open up a client
 - `localhost:5050 -> yourwebsite.com`

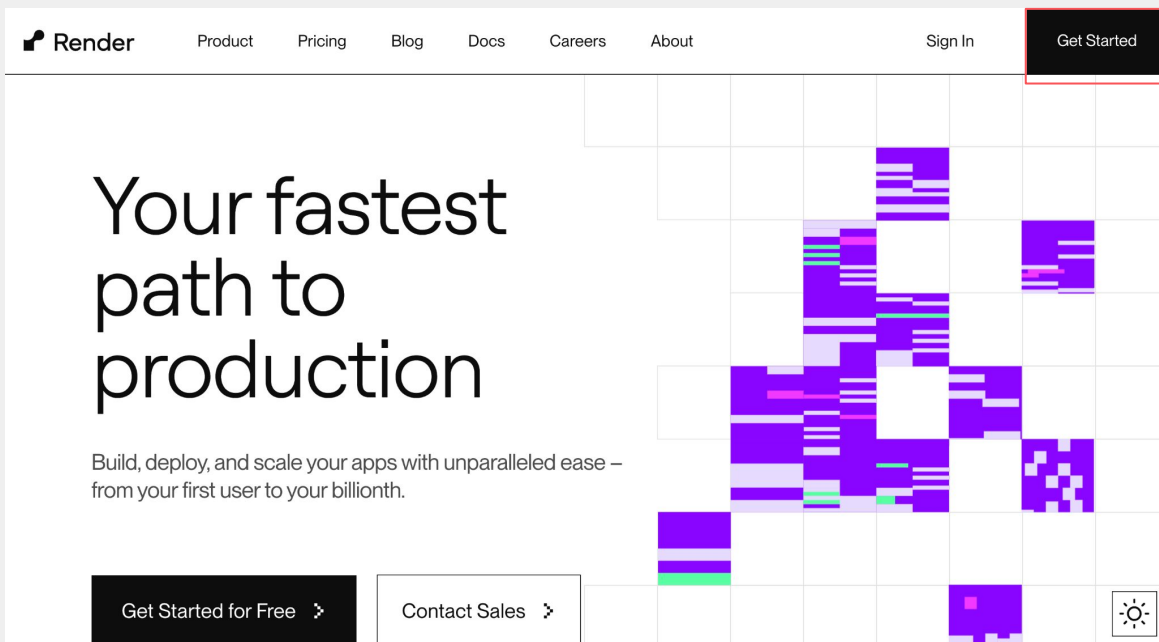
Deployment

- We're going to use Render to deploy.
- `yourappname.onrender.com`
- `http://catbook-5bwk.onrender.com/`



Part 1: Create a Render Account


1. Go to <https://www.render.com/> and click “Get Started” in the top right.





Part 1: Create a Render Account

2. Fill out the account creation info, verify email, and fill out form to create account.

Sign up for Render

 GitHub

 GitLab

 Google

OR

Email

Password

☐ Email me about Render product updates

COMPLETE SIGN UP

Tell us about yourself.

What should we call you?

How will you primarily use Render?

What framework do you primarily use?

What code repository do you use?

Where did you first hear about us?
*

CONTINUE TO RENDER

Deployment: Visual Walkthrough
Only 1 person per team needs to do this

Deployment: visual walkthrough

The screenshot shows the Render dashboard with a navigation bar at the top. The navigation bar includes the Render logo, links to Dashboard, Blueprints, Env Groups, Docs, Community, and Help, a 'New +' button, and a user profile dropdown for 'weblab-staff@mit.edu'. The main content area is titled 'Get started in minutes' and features a grid of service cards. A user menu is open on the right, with the 'weblab-staff' option highlighted by a red rectangle.

Render Dashboard | Blueprints | Env Groups | Docs | Community | Help | New + | weblab-staff@mit.edu

Get started in minutes

- Static Sites**
Static Sites are automatically served over a global CDN. Add a custom domain and get free, fully-managed SSL.
[New Static Site](#)
- Web Services**
Web Services include zero-downtime deploys, persistent storage and PR previews. Scale up and down with ease.
[New Web Service](#)
- Private Services**
Private Services are only accessible within your Render network and can speak any protocol.
[New Private Service](#)
- Cron Jobs**
With Cron Jobs, you can schedule any command or script to run on a regular interval.
[New Cron Job](#)
- PostgreSQL**
Fully-managed hosted PostgreSQL with internal and external connectivity, and automated daily backups.
[New PostgreSQL](#)
- Redis**
A cloud based in-memory key value datastore. Render offers fully managed hosted Redis instances.
[New Redis](#)
- Blueprints**
A Blueprint specifies your Infrastructure as Code in a single file. Use it to set up all your services at once.
[New Blueprint](#)

User Menu:

- weblab-staff@mit.edu
- weblab-staff** (highlighted)
- New Team
- Account Settings
- Billing
- Sign Out

Deployment: visual walkthrough

The screenshot displays the Render dashboard interface. At the top, the navigation bar includes the Render logo, links to Dashboard, Blueprints, Env Groups, Docs, Community, and Help, a 'New +' button, and a user profile for 'weblab-staff@mit.edu'. The main content area is titled 'Get started in minutes' and features a grid of service cards. A 'New' dropdown menu is open, listing options: Static Site, Web Service (highlighted), Private Service, Background Worker, Cron Job, PostgreSQL, Redis, and Blueprint. The visible cards include:

- Static Sites**: Automatically served over a global CDN. Add a custom domain and get free, fully-managed SSL. Button: **New Static Site**.
- Web Services**: Include zero-downtime deploys, persistent storage and PR previews. Scale up and down with ease. Button: **New Web Service**.
- Web Service** (expanded): Web services are kept up and running at all times, with native SSL and HTTP/2 support. Add a persistent disk or custom domain. Scale up and down with ease. [Learn more.](#) Button: **New Private Service**.
- Background Workers**: Workers are long running queues and streaming. Button: **New Worker**.
- Cron Jobs**: With [Cron Jobs](#), you can schedule any command or script to run on a regular interval.
- PostgreSQL**: Fully-managed hosted [PostgreSQL](#) with internal and external connectivity, and automated daily backups.
- Redis**: A cloud based in-memory key value datastore. Render offers fully managed hosted [Redis instances](#).
- Blueprints**: A [Blueprint](#) specifies your Infrastructure as Code in a single file. Use it to set up all your services at once.

Deployment: visual walkthrough

Render Dashboard Blueprints Env Groups Docs Community Help New + weblab-staff@mit.edu

Create a new Web Service

Connect a Git repository, or use an existing image.

How would you like to deploy your web service?

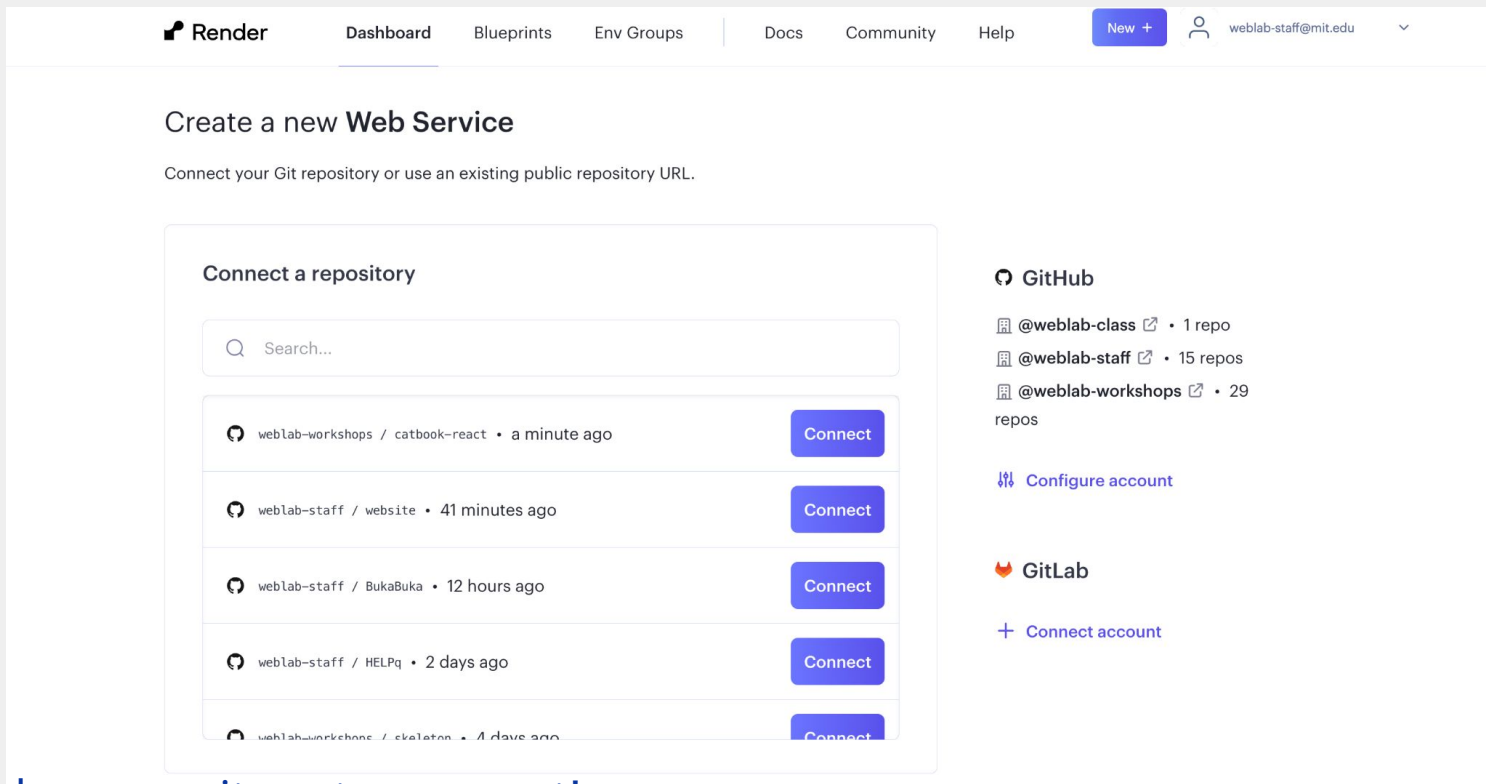
- ☒ **Build and deploy from a Git repository**
Connect a GitHub or GitLab repository.
- ☐ **Deploy an existing image from a registry** ADVANCED
Pull a public image from any registry or a private image from Docker Hub, GitHub, or GitLab.

Next

Feedback Invite a Friend Contact Support

Select “Build and deploy from a Git repository”

Deployment: visual walkthrough








Render Dashboard Blueprints Env Groups Docs Community Help New + weblab-staff@mit.edu




Create a new Web Service

Connect your Git repository or use an existing public repository URL.

Connect a repository

 weblab-workshops / catbook-react • a minute ago	Connect
 weblab-staff / website • 41 minutes ago	Connect
 weblab-staff / BukaBuka • 12 hours ago	Connect
 weblab-staff / HELPq • 2 days ago	Connect
 weblab-workshops / skelton • 4 days ago	Connect

GitHub

-  @weblab-class • 1 repo
-  @weblab-staff • 15 repos
-  @weblab-workshops • 29 repos

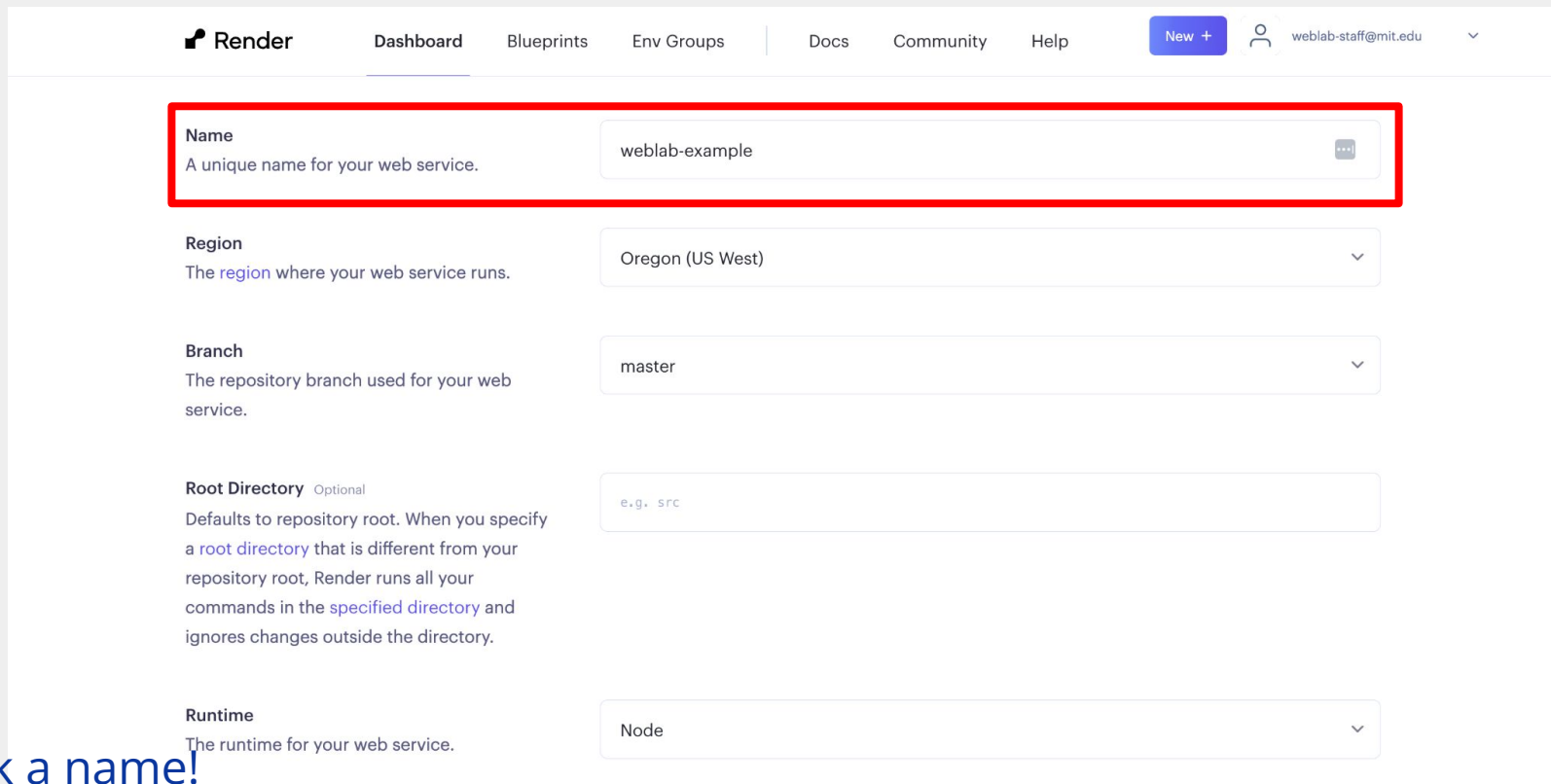
[Configure account](#)

GitLab

[+ Connect account](#)

Pick a repository to connect!

Deployment: visual walkthrough



The screenshot shows the Render deployment configuration interface. The top navigation bar includes the Render logo, links to Dashboard, Blueprints, Env Groups, Docs, Community, and Help, a 'New +' button, and a user profile for 'weblab-staff@mit.edu'. The main form contains several fields: 'Name' (highlighted with a red box), 'Region', 'Branch', 'Root Directory', and 'Runtime'. The 'Name' field is labeled 'Name' and 'A unique name for your web service.' with the value 'weblab-example'. The 'Region' field is labeled 'Region' and 'The region where your web service runs.' with the value 'Oregon (US West)'. The 'Branch' field is labeled 'Branch' and 'The repository branch used for your web service.' with the value 'master'. The 'Root Directory' field is labeled 'Root Directory' (Optional) and 'Defaults to repository root. When you specify a root directory that is different from your repository root, Render runs all your commands in the specified directory and ignores changes outside the directory.' with the value 'e.g. src'. The 'Runtime' field is labeled 'Runtime' and 'The runtime for your web service.' with the value 'Node'.

Name
A unique name for your web service.

weblab-example

Region
The region where your web service runs.

Oregon (US West)

Branch
The repository branch used for your web service.

master

Root Directory Optional
Defaults to repository root. When you specify a root directory that is different from your repository root, Render runs all your commands in the specified directory and ignores changes outside the directory.

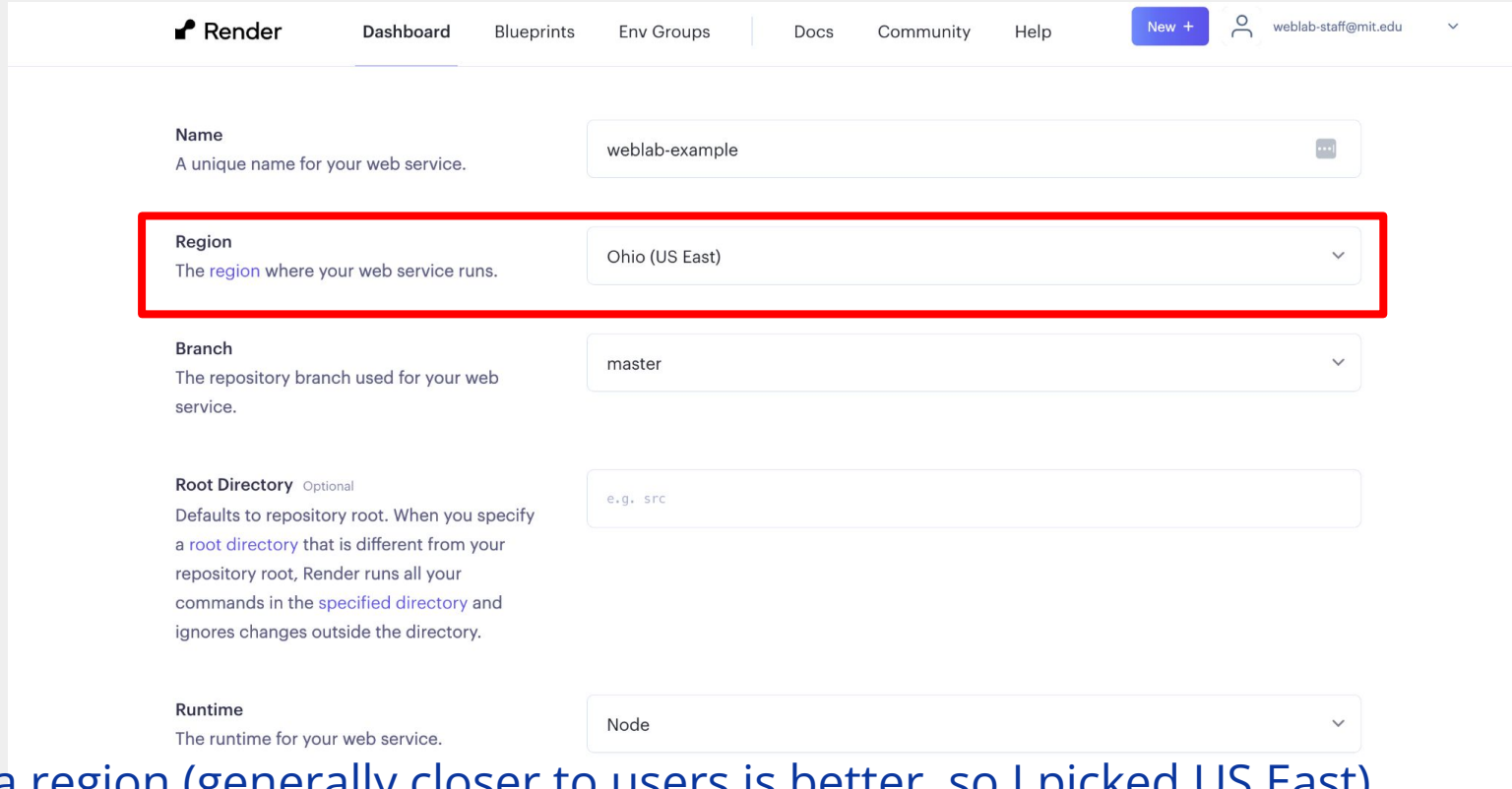
e.g. src

Runtime
The runtime for your web service.

Node

Pick a name!

Deployment: visual walkthrough

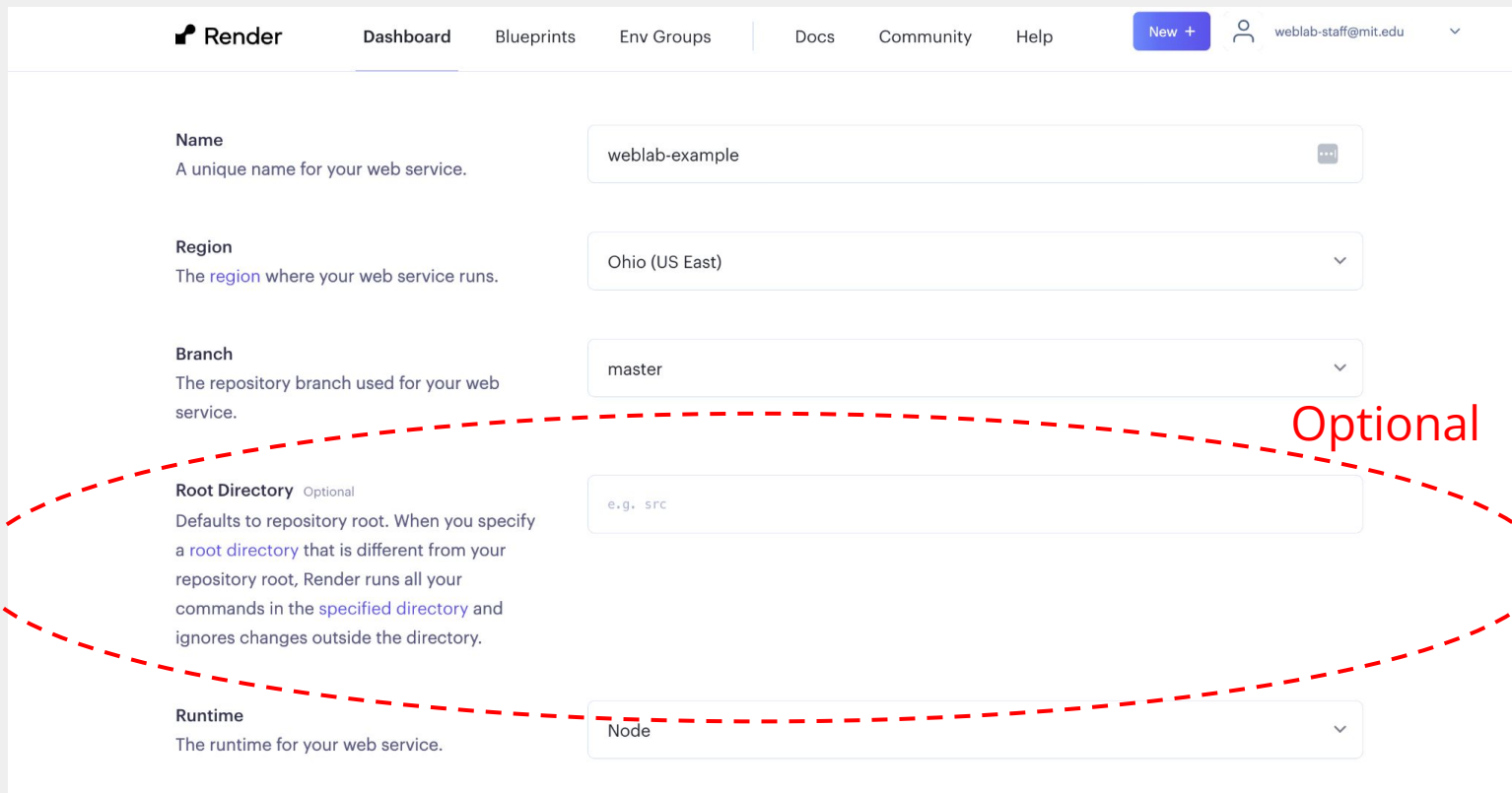


The screenshot shows the Render deployment configuration interface. The 'Region' field is highlighted with a red rectangular box. The configuration includes the following fields:

- Name:** A unique name for your web service. Value: `weblab-example`
- Region:** The [region](#) where your web service runs. Value: `Ohio (US East)`
- Branch:** The repository branch used for your web service. Value: `master`
- Root Directory:** Optional. Defaults to repository root. When you specify a [root directory](#) that is different from your repository root, Render runs all your commands in the [specified directory](#) and ignores changes outside the directory. Value: `e.g., src`
- Runtime:** The runtime for your web service. Value: `Node`

Pick a region (generally closer to users is better, so I picked US East)

Deployment: visual walkthrough



The screenshot shows the Render deployment configuration interface. A red dashed circle highlights the 'Root Directory' and 'Runtime' sections. The 'Root Directory' section is labeled 'Optional' and includes a text input field with the placeholder 'e.g., src'. The 'Runtime' section is a dropdown menu currently set to 'Node'.

Render Dashboard Blueprints Env Groups Docs Community Help New + weblab-staff@mit.edu

Name
A unique name for your web service.
weblab-example

Region
The [region](#) where your web service runs.
Ohio (US East)


Branch
The repository branch used for your web service.
master



Root Directory Optional
Defaults to repository root. When you specify a [root directory](#) that is different from your repository root, Render runs all your commands in the [specified directory](#) and ignores changes outside the directory.
e.g., src

Runtime
The runtime for your web service.
Node


Optional

Deployment: visual walkthrough


 **Render**

[Dashboard](#) [Blueprints](#) [Env Groups](#) | [Docs](#) [Community](#) [Help](#) [New +](#)  [weblab-staff@mit.edu](#) 


Name
A unique name for your web service.

weblab-example 

Region
The [region](#) where your web service runs.

Ohio (US East) 


Branch
The repository branch used for your web service.

master 

Root Directory Optional
Defaults to repository root. When you specify a [root directory](#) that is different from your repository root, Render runs all your commands in the [specified directory](#) and ignores changes outside the directory.


e.g., `src`

Runtime
The runtime for your web service.



Node 

Pick Node for your runtime.

Deployment: visual walkthrough

 **Render**

[Dashboard](#) [Blueprints](#) [Env Groups](#) | [Docs](#) [Community](#) [Help](#)

[New +](#)  [weblab-staff@mit.edu](#) 

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 && npm run build
```

Start Command

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

```
$ npm start
```


Instance Type



For hobby projects	Free \$0 / month	512 MB (RAM) 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
---------------------------	----------------------------	-------------------------	--

Set Build

Command to
npm install &&
npm run build

Deployment: visual walkthrough

 **Render**

[Dashboard](#) [Blueprints](#) [Env Groups](#) | [Docs](#) [Community](#) [Help](#) [New +](#)  [weblab-staff@mit.edu](#) 

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 && npm run build
```

Start Command

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

```
$ npm start
```

Instance Type

For hobby projects	Free \$0 / month	512 MB (RAM) 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
---------------------------	----------------------------	-------------------------	--

Set Start
Command to
npm start

Deployment: visual walkthrough

The screenshot shows the Render dashboard with the 'Instance Type' section. A red box highlights the Starter, Standard, Pro, and Pro Plus tiers. The Starter tier is highlighted with a blue border. Below the instance types, there is a section for 'Environment Variables' with a form to add a new variable.

environment variables defined by you in Render.

Instance Type

For hobby projects

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

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) 0.5 CPU	Standard	2 GB (RAM) 1 CPU
\$7 / month		\$25 / month	

Pro	4 GB (RAM) 2 CPU	Pro Plus	8 GB (RAM) 4 CPU
\$85 / month		\$175 / month	

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

Need a [custom instance type](#)? We support up to 512 GB RAM and 64 CPUs.

Environment Variables Optional

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

Select any tier
under \$200!

Deployment: visual walkthrough

The screenshot shows the Render dashboard with the 'Environment Variables' section highlighted by a red rectangle. The dashboard includes a navigation bar with links to Dashboard, Blueprints, Env Groups, Docs, Community, and Help. A 'New +' button and a user profile icon are also present. The main content area displays pricing for 'Pro Max' (\$225/month, 16 GB RAM, 4 CPU) and 'Pro Ultra' (\$450/month, 32 GB RAM, 8 CPU). Below this, there is a section for 'Environment Variables' with a description: 'Set environment-specific config and secrets (such as API keys), then read those values from your code. [Learn more.](#)'. Two variables are listed: 'MONGO_SRV' and 'SESSION_SECRET', each with a 'value' field, a 'Generate' button, and a trash icon. A '+ Add Environment Variable' button is at the bottom of the list. Below the red rectangle, there is an 'Advanced' dropdown and a 'Create Web Service' button. At the very bottom, there are links for 'Feedback', 'Invite a Friend', and 'Contact Support'.

Render Dashboard

Navigation: Dashboard, Blueprints, Env Groups, Docs, Community, Help

User: webiab-staff@mit.edu

One-off jobs
Support for persistent disks

Pro Max 16 GB (RAM) 4 CPU
\$225 / month

Pro Ultra 32 GB (RAM) 8 CPU
\$450 / month

Need a [custom instance type](#)? We support up to 512 GB RAM and 64 CPUs.

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

MONGO_SRV value [Generate](#)

SESSION_SECRET value [Generate](#)

+ Add Environment Variable

Advanced

Create Web Service

Feedback Invite a Friend Contact Support

If you want, set environment variables MONGO_SRV and SESSION_SECRET but I'll explain environmental variables in a bit – we can skip this and do it later

Deployment: visual walkthrough

The screenshot shows the Render dashboard with the following elements:

- Navigation Bar:** Render logo, Dashboard (active), Blueprints, Env Groups, Docs, Community, Help, New + button, and user profile (weblab-staff@mit.edu).
- Deployment Options:**
 - One-off jobs
 - Support for persistent disks
 - Pro Max:** 16 GB (RAM), 4 CPU, \$225 / month
 - Pro Ultra:** 32 GB (RAM), 8 CPU, \$450 / month
- Custom Instance Note:** Need a [custom instance type](#)? We support up to 512 GB RAM and 64 CPUs.
- Environment Variables:**
 - Optional section: Set environment-specific config and secrets (such as API keys), then read those values from your code. [Learn more.](#)
 - Variables: MONGO_SRV, SESSION_SECRET
 - Each variable has a "value" input field, a "Generate" button, and a trash icon.
 - + Add Environment Variable button
- Advanced:** A dropdown menu.
- Create Web Service:** A blue button highlighted with a red rectangle.
- Create!** Red text next to the highlighted button.
- Footer:** Feedback, Invite a Friend, Contact Support.

Boom! We've deployed!

Or did we?

If success...

If should see something like this on your dashboard after deploying, you should be good to go for now!

Ungrouped Services

Active 1

Suspended 0

All 1

SERVICE NAME

STATUS

TYPE

RUNTIME

REGION

LAST DEPLOYED ↓



catbook



Deployed

Web Service


Node

Oregon

2 hours ago



If success...

 **Render**

Dashboard

Blueprints


Env Groups

Docs


Community

Help

New +

 weblab-staff

▼

 WEB SERVICE


catbook


Node


Standard

Connect ▼

Manual Deploy ▼

 weblab-workshops / catbook-react

 deploy

<https://catbook-5bwk.onrender.com> 

Events

Logs

Disks

Environment


Shell


Previews


Jobs


Metrics


Scaling

 Autoscailing configured to scale from 1 to 2 instances with 70% target CPU utilization and 70% target memory utilization.
January 8, 2024 at 11:21 AM

 Autoscailing configured to scale from 1 to 2 instances with 60% target CPU utilization and 60% target memory utilization.
January 8, 2024 at 11:21 AM

 Instance type changed from Free to Standard
January 8, 2024 at 11:18 AM

 **Deploy live for det1bb5c:** regenerate package-lock.json
January 8, 2024 at 11:18 AM



Otherwise... check logs to debug!!

Events

Logs

Disks

Environment

Shell

Previews

Jobs

Metrics

Scaling

Settings

January 8, 2024 at 12:42 PM Failed

Exited with status 1 while building your code.

97b9bcf Add files via upload

Rollback

All logs ▾ Search 🔍 Jan 8, 12:41 PM - 12:44 PM ▾ EST ⬆ ⚙

Jan 8 12:43:30 PM ⓘ at /opt/render/project/src/node_modules/babel-loader/lib/index.js:59:103 {

Jan 8 12:43:30 PM ⚠️ opensslErrorStack: ['error:03000086:digital envelope routines::initialization error'],

Jan 8 12:43:30 PM ⓘ library: 'digital envelope routines',

Jan 8 12:43:30 PM ⓘ reason: 'unsupported',

Jan 8 12:43:30 PM ⓘ code: 'ERR_OSSL_EVP_UNSUPPORTED'

Jan 8 12:43:30 PM ⓘ }

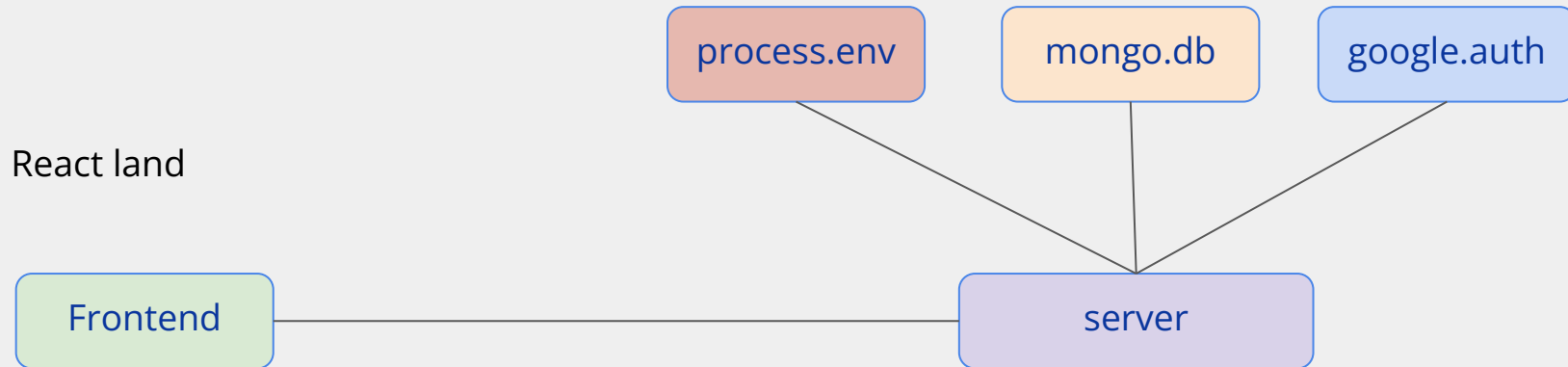
Jan 8 12:43:30 PM ⓘ

Jan 8 12:43:30 PM ⓘ Node.js v20.10.0

Jan 8 12:43:30 PM ⓘ ==> Build failed 🚫

Setting environment variables


Understanding `process.env`



Understanding `process.env`

- `process.env` serves as a sort of **global variable** for our **backend** files to access
- We can use it to hide sensitive info

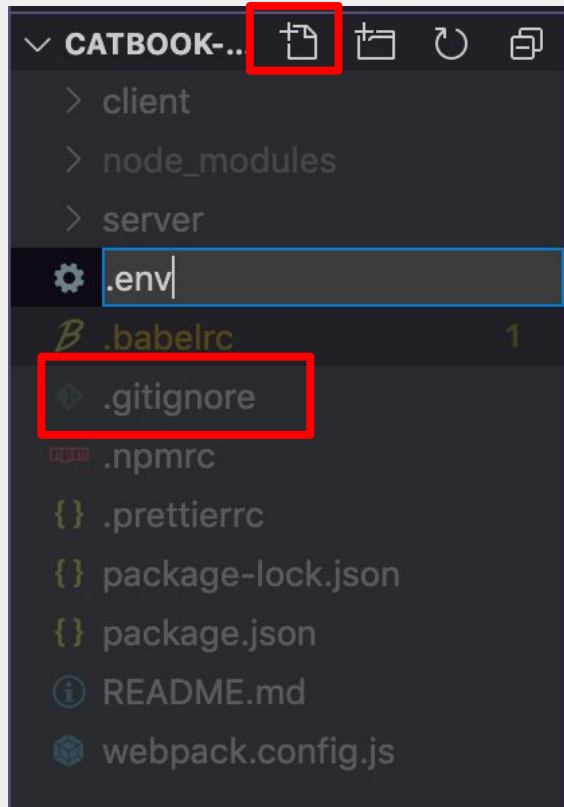
Understanding process.env



```
34 // Server configuration below
35 const mongoConnectionURL =
36 | "mongodb+srv://weblab:jAT4p5IAgYWQgR@catbook-y1ndp.mongodb.net/test?retryWrites=true&w=majority";
37 const dbName = "catbook";
```

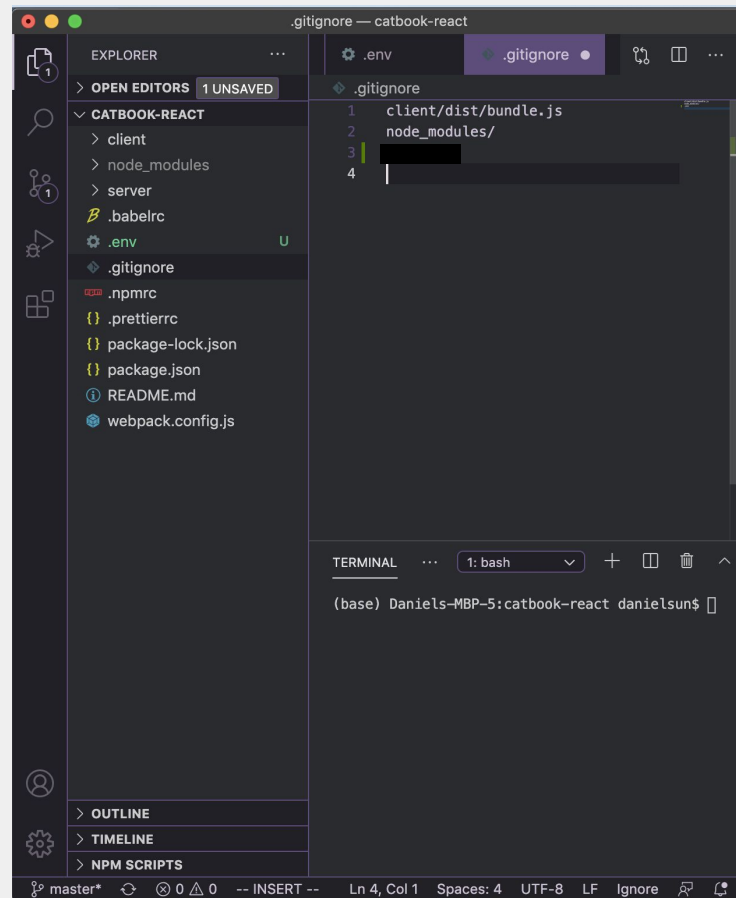
How to use `process.env`

- Step 1: Add a file named `.env` to your project's outermost folder
- Step 2: Add `.env` to your project's `.gitignore` file



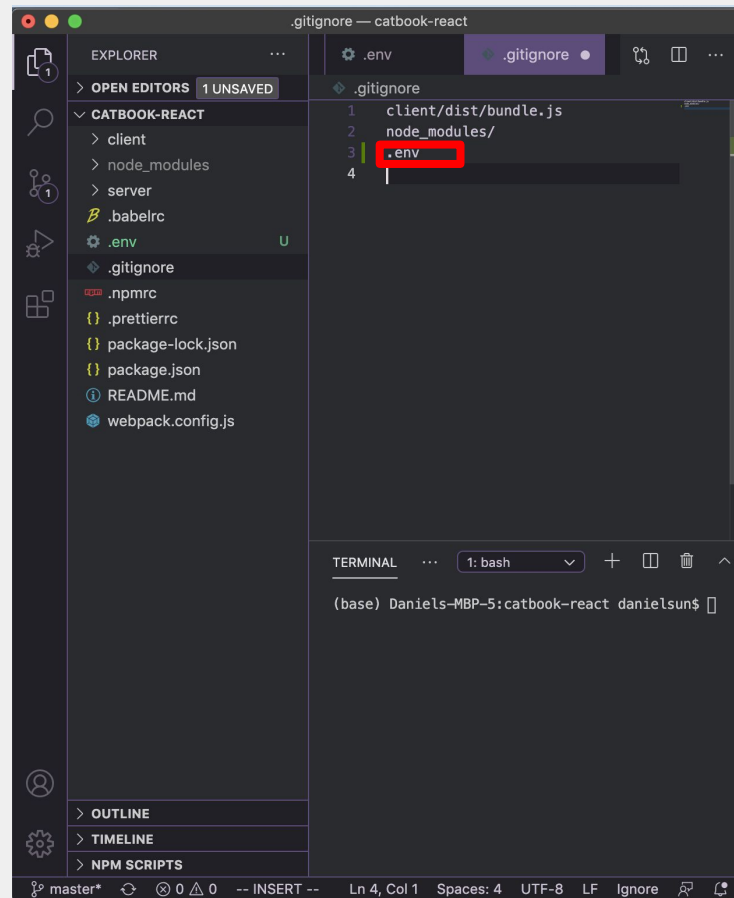
How to use `process.env`

- Step 1: Add a file named `.env` to your project's outermost folder
- Step 2: Add `.env` to your project's `.gitignore` file



How to use `process.env`

- Step 1: Add a file named `.env` to your project's outermost folder
- Step 2: Add `.env` to your project's `.gitignore` file



How to use `process.env`

- Step 3: Add relevant *sEcREts* to the `.env` file

```
1  SESSION_SECRET="session-secret"  
2  MONGO_SRV="mongodb+srv://username:password@catbook-ylndp.mongodb.net/test?retryWrites=true&w=majority"
```

- Note! Everyone has to make their own `.env` file!

How to use `process.env`

- Step 4: Install the dotenv package!
 - Use `npm install dotenv`
- Step 5: Add the following line near the top of `server.js`
 - `require('dotenv').config();`
- Step 6: Use `process.env.VAR_NAME` where needed!

How to use process.env

```
// Server configuration below
const mongoConnectionURL =
  "mongodb+srv://weblab:AT4po55IAgYWQgP@catbook-ylnp.mongodb.net/test?retryWrites=true&w=majority"
```

```
1 SESSION_SECRET="session-secret"
```

```
2 MONGO_SRV="mongodb+srv://username:password@catbook-ylnp.mongodb.net/test?retryWrites=true&w=majority"
```

```
// set up a session, which will persist login data across requests
```

```
app.use(
  session({
    secret: "session-secret",
    resave: false,
    saveUninitialized: false,
  })
);
```

How to use process.env

```
// Server configuration below
// TODO change connection URL after setting up your team database
const mongoConnectionString = process.env.MONGO_SRV;
// TODO change database name to the name you chose
const dbName = "FILL_ME_IN";
```

```
// set up a session, which will persist login data across requests
app.use(
  session({
    secret: process.env.SESSION_SECRET,
    resave: false,
    saveUninitialized: false,
  })
);
```

Deployment: again!

- After making these changes, always remember to:
 - `git add <file>`
 - `git commit -m "message"`
 - `git push`
- After that, we can redeploy!

Setting environment
variables on Render

Understanding environmental variables on Render

- Render doesn't actually have access to our .env files!
- To fix this, we can set Render environment variables
 - Render version of a .env file

How to set environment variables on Render

The screenshot shows the Render dashboard for a service named 'catbook'. The sidebar on the left contains navigation links: Events, Logs, Disks, Environment (highlighted with a red box), Shell, Previews, Jobs, Metrics, and Scaling. The main content area shows the 'Environment Variables' configuration panel, which is also highlighted with a red box. This panel includes a title 'Environment Variables', a description 'Set environment-specific config and secrets (such as API keys), then read those values from your code. [Learn more.](#)', and a table with two columns: 'Key' and 'Value'. The table contains two entries: 'MONGO_SRV' and 'SESSION_SECRET', both with masked values. To the right of each entry is a trash icon. At the bottom of the panel are three buttons: 'Create Environment Group' (with a link icon), '+ Add Environment Variable', and 'Save Changes'.

Render

Dashboard Blueprints Env Groups Docs Community Help

New + W weblab-staff

WEB SERVICE

catbook Node Standard

Connect Manual Deploy

weblab-workshops / catbook-react deploy

<https://catbook-5bwk.onrender.com>

Events

Logs

Disks

Environment

Shell

Previews

Jobs

Metrics

Scaling

Environment Variables

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

Key	Value
MONGO_SRV
SESSION_SECRET

Create Environment Group + Add Environment Variable Save Changes

Secret Files

How to set environment variables on Render

The screenshot shows the Render dashboard with the 'Environment' tab selected in the left sidebar. The main content area is divided into two sections: 'Environment Variables' and 'Secret Files'.

Environment Variables: This section allows you to set environment-specific configuration and secrets. It features a table with two columns: 'Key' and 'Value'. Two variables are listed: 'MONGO_SRV' and 'SESSION_SECRET', both with masked values. There are buttons to 'Create Environment Group', 'Add Environment Variable', and 'Save Changes'.

Secret Files: This section is highlighted with a red rectangle. It allows you to store plaintext files containing secret data, such as a .env file or a private key. It includes instructions on how to read these files during builds and at runtime. There is a button to 'Add Secret File'.

You can
also
upload
your .env
file here!



Deployment: again!

- After setting environment variables on Render, redeploy again!
 - Head to Render, dashboard for your web service, then click Manual Deploy.



Dashboard

Blueprints

Env Groups

Docs

Community

Help

New +

W

weblab-staff

WEB SERVICE

catbook

Node

Standard

Connect

Manual Deploy

weblab-workshops / catbook-react






deploy

<https://catbook-5bwk.onrender.com>



Going back to Google Auth

OAuth 2.0 Client IDs

<input type="checkbox"/>	Name	Creation date ↓	Type	Client ID	Actions
<input type="checkbox"/>	Web client 1	Jan 19, 2024	Web application	519761739942-5fsi...	<div>a. Edit your cred</div> <div>  </div>

Edit OAuth client

[Manage service accounts](#)

Service Accounts

Authorized JavaScript origins ?

For use with requests from a browser

URIs 1 *
http://localhost:5050

URIs 2 *
http://localhost

URIs 3 *
http://weblabproject.onrender.com

+ ADD URI

Add your deployed URL from render

Make sure to add this, or you will not be able to log in on your deployed site!

Success! (hopefully)

Things might not work out.

Recap

- Link your app to Render through Github
- Create .env file and add to your .gitignore
 - Remember .env contains stuff that github should not know about
- Move sensitive data into the .env file
 - Including your mongo srv and session secret
 - Access them in backend code with `process.env.VARIABLE_NAME`
- Make sure Render knows about your .env variables
 - Open Environment on dashboard and put them in. You don't need quotes for strings
- Click deploy

Now to the engineers
behind Render!