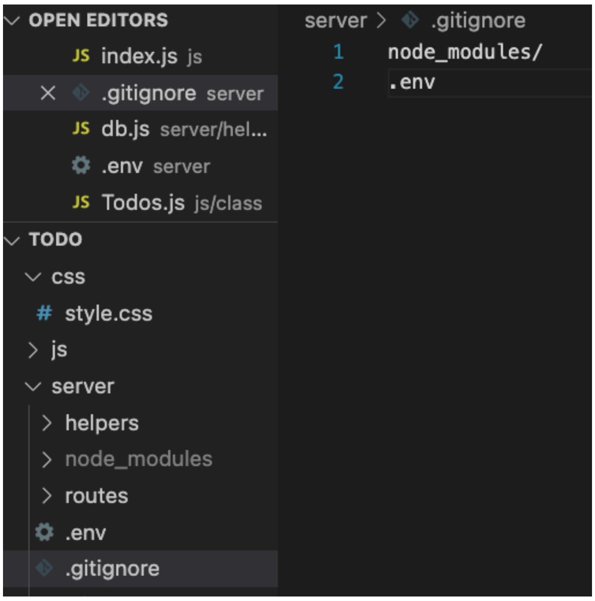
Exercise. Todo (part 8) – Deployment

Finally, web site app is deployed to the server. In this example, a service available on https.//render.com is used. This is a relatively simple service that can be used for deploying HTML/CSS/Javascript front-end, NodeJS backend, and Postgres database. Render.com can be used for free. Create a free account at https://render.com. This exercise makes use of instructions available at https://www.dotenv.org/docs/languages/nodejs/render.

Key learnings for this part are:

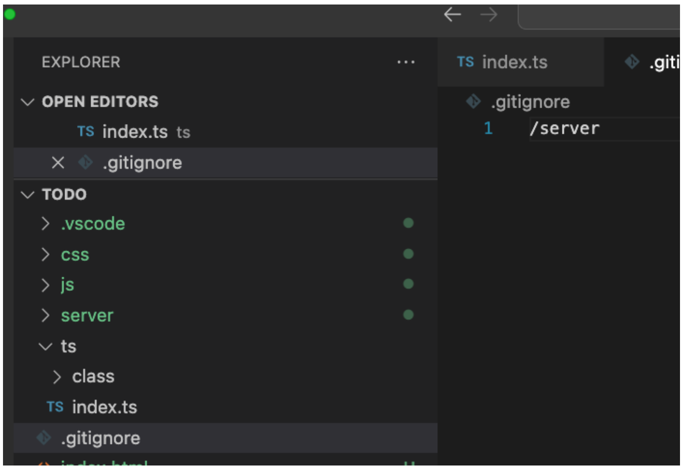
- Create an account for a service
- Pushing frontend and backend code into separate repositories for deployment
- Deploying PostgreSQL database
- Creating env file for production on Render
- Deploying NodeJS server
- Deploying front-end using HTML, CSS and JavaScript

Before starting deployment, make sure, that you have pushed your front-end and back-end code to **SEPARATE** repositories. Create a .gitignore file under the server folder with the following content. Initialize the repository, commit the server folder to git, and push it to the Github server.



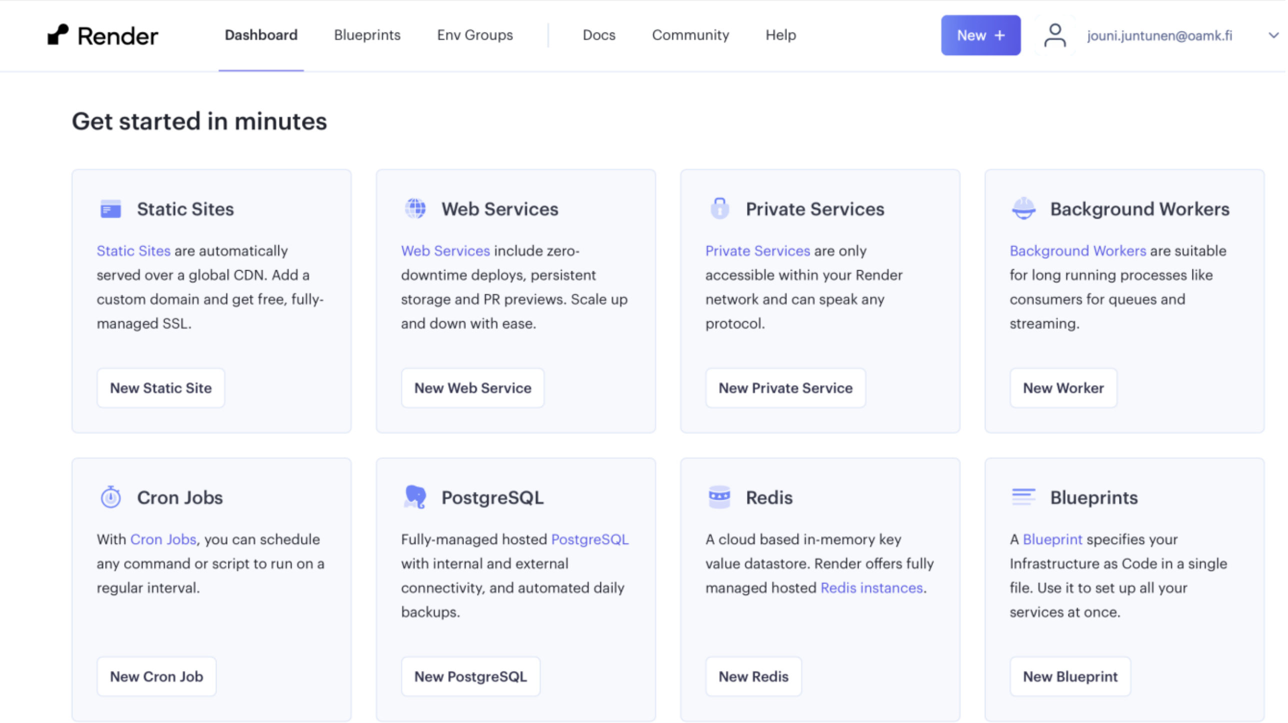
```
node_modules/
.env
```

Create .gitignore to the root folder as well. Since client code is pushed into a separate repository, the server folder is ignored. Initialize the repository, commit the root folder to git, and push it to the Github server.



```
/server
```

Open render.com. As described earlier, the client, backend, and database are deployed separately into different services on the Render server.



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

Background Workers

Background Workers are suitable for long running processes like consumers for queues and streaming.

New Worker

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.

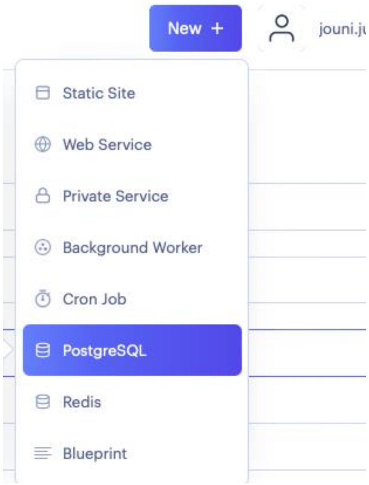
New Redis

Blueprints

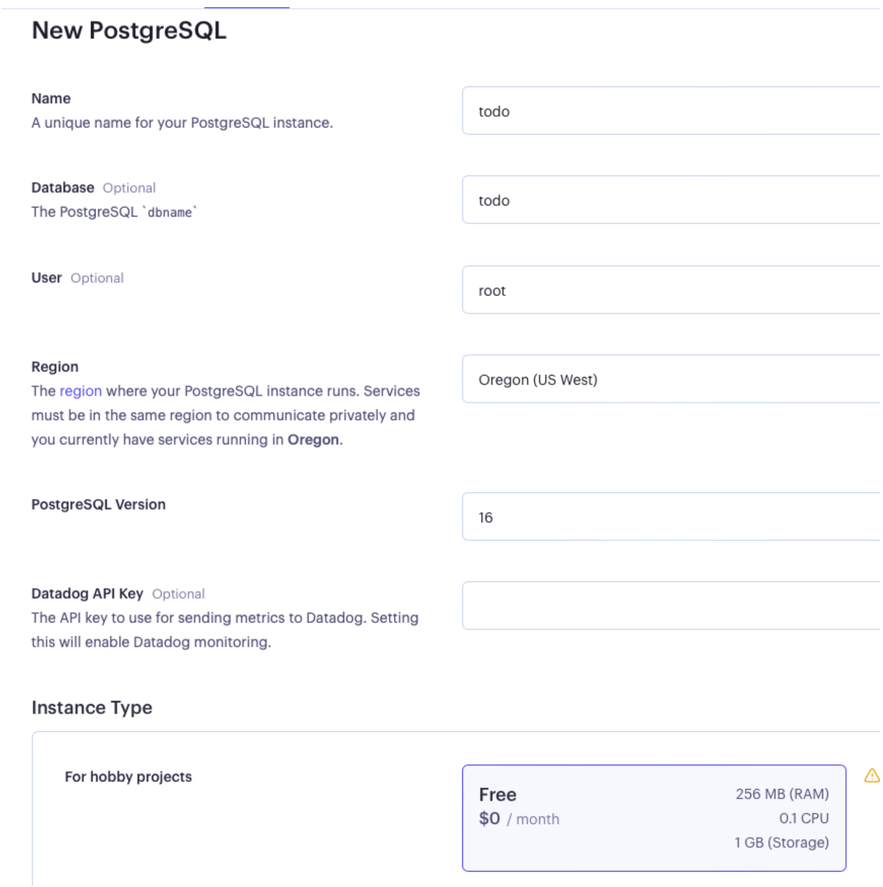
A Blueprint specifies your Infrastructure as Code in a single file. Use it to set up all your services at once.

New Blueprint

Start by deploying database. Select New and PostgreSQL.



Provide name, database name, and user name (which cannot be postgres) and hit Create database button.



New PostgreSQL

Name
A unique name for your PostgreSQL instance.
todo

Database Optional
The PostgreSQL `dbname`
todo

User Optional
root

Region
The region where your PostgreSQL instance runs. Services must be in the same region to communicate privately and you currently have services running in **Oregon**.
Oregon (US West)

PostgreSQL Version
16

Datadog API Key Optional
The API key to use for sending metrics to Datadog. Setting this will enable Datadog monitoring.

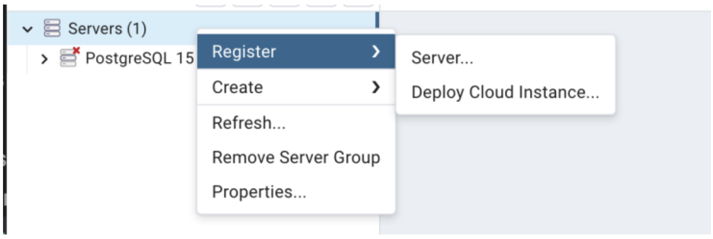
Instance Type

For hobby projects

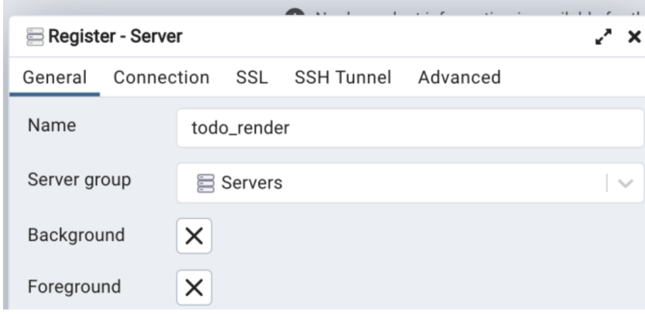
Free
$0 / month
256 MB (RAM)
0.1 CPU
1 GB (Storage)

Wait until the database gets created to get the required information and credentials for connecting to the database. This might take a while (free service from render.com is rather slow).

After database creation is ready, open PGAdmin4 on your workstation. Register server from render.com.



Provide name.



Register - Server

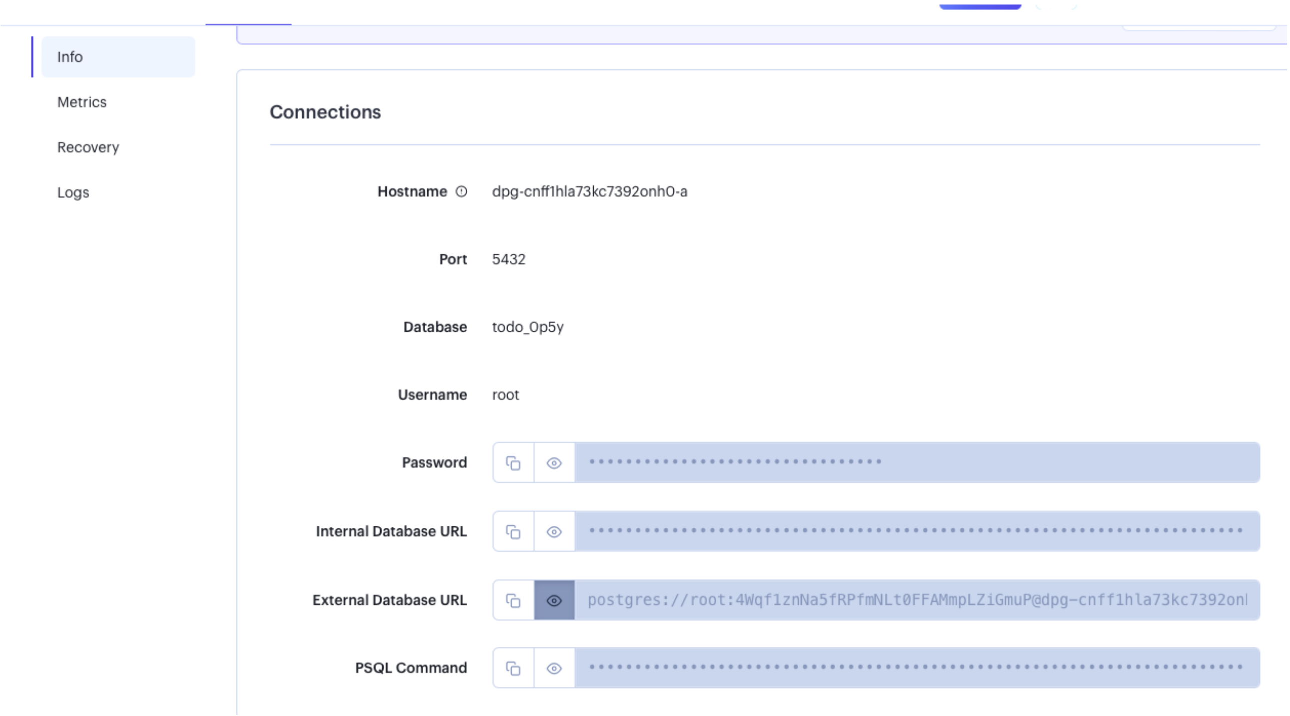
Name: todo_render

Server group: Servers

Background: ☒

Foreground: ☒

Copy external url for PostgreSQL database from render.com. Select info from the left side menu and press the eye icon to reveal text for External database URL.



Connections

Hostname: dpg-cnff1hla73kc7392onh0-a

Port: 5432

Database: todo_0p5y

Username: root

Password: ••••••••••••••••••••••••••••••••

Internal Database URL: ••

External Database URL: postgres://root:4Wqf1znNa5fRPfmNLt0FFAMmpLZiGmuP@dpg-cnff1hla73kc7392onl

PSQL Command: ••

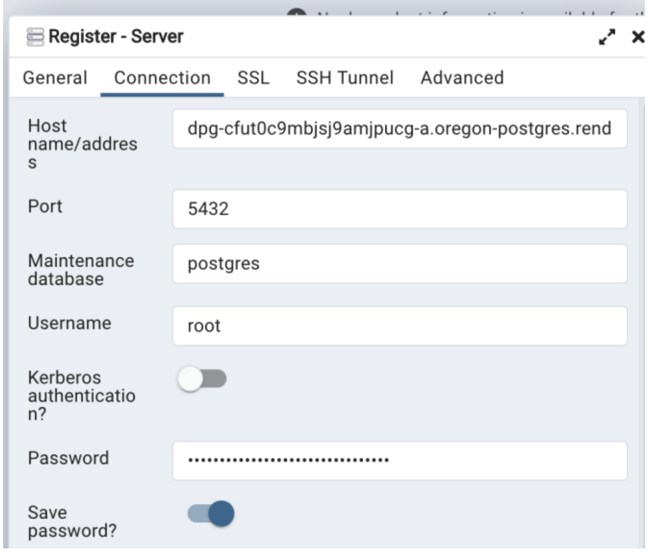
The original address is as follows (this is an example, the string will be slightly different for you).

postgres://root:8fs76r7K1bSNwk1npEyQiS2Sbe4UIwkl@dpg-cfut0c9mbjsj9amjpucg-a.oregon-postgres.render.com/todo_1bob

The address has some extra data and the final value should be as follows (cut off the beginning and end from the original string).

dpg-cfut0c9mbjsj9amjpucg-a.oregon-postgres.render.com

Provide that as host name, configure port, username and password (which can be copied from render.com under info the same way than External database URL).



Register - Server

Host name/address: dpg-cfut0c9mbjsj9amjpucg-a.oregon-postgres.rend

Port: 5432

Maintenance database: postgres

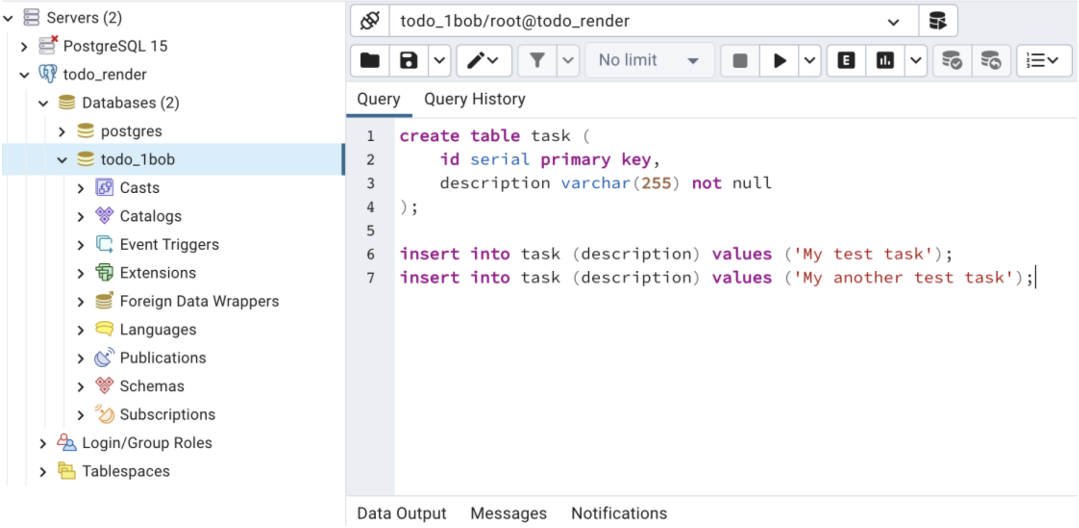
Username: root

Kerberos authentication?: ○

Password: ••••••••••••••••••••••••••••••••

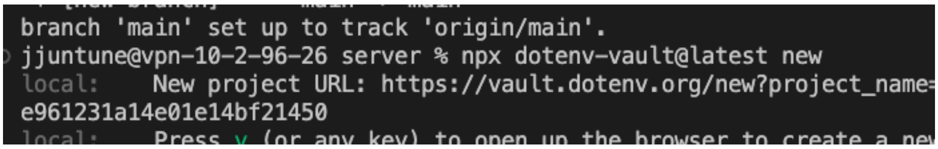
Save password?: ●

Create connection. After the server is connected, open the database and Query tool. Create a table with some test data.



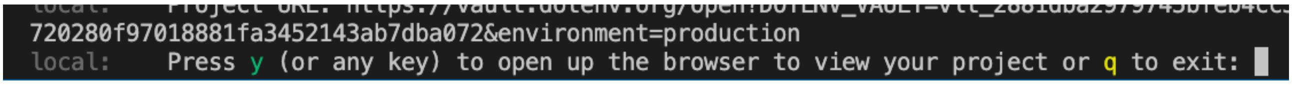
```sql
create table task (
    id serial primary key,
    description varchar(255) not null
);

insert into task (description) values ('My test task');
insert into task (description) values ('My another test task');
```

Next .env file production is created. First, you need to run *npx dotenv-vault@latest new* command under server folder.

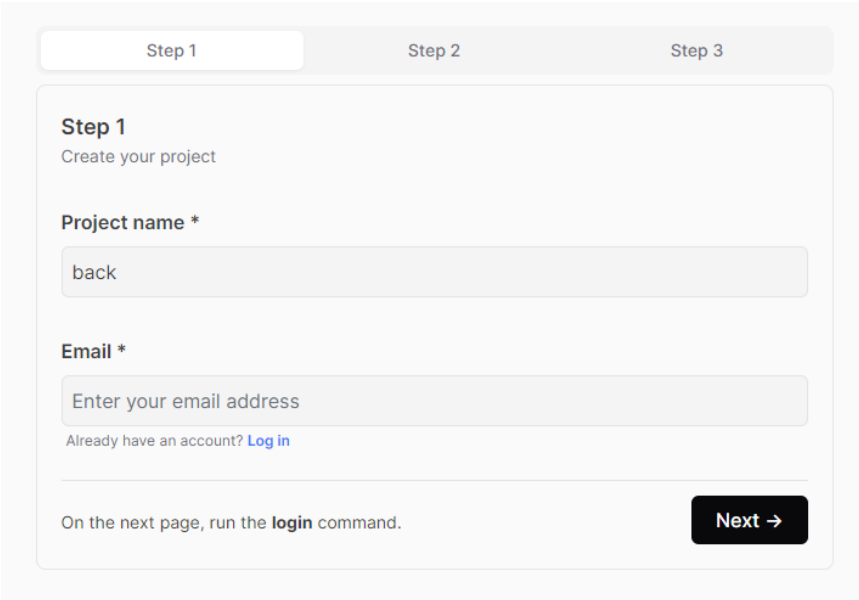


```
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
jjuntune@vpn-10-2-96-26 server % npx dotenv-vault@latest new
local:    New project URL: https://vault.dotenv.org/new?project_name=
e961231a14e01e14bf21450
local:    Press y (or any key) to open up the browser to create a new
```

While running commands you might be are asked to press y to open browser. Do so if asked.



```
local:    Project URL: https://vault.dotenv.org/open?DOTENV_VAULT=vlt_2881dba29797435bfeb4cc
720280f97018881fa3452143ab7dba072&environment=production
local:    Press y (or any key) to open up the browser to view your project or q to exit:
```



Step 1
Create your project

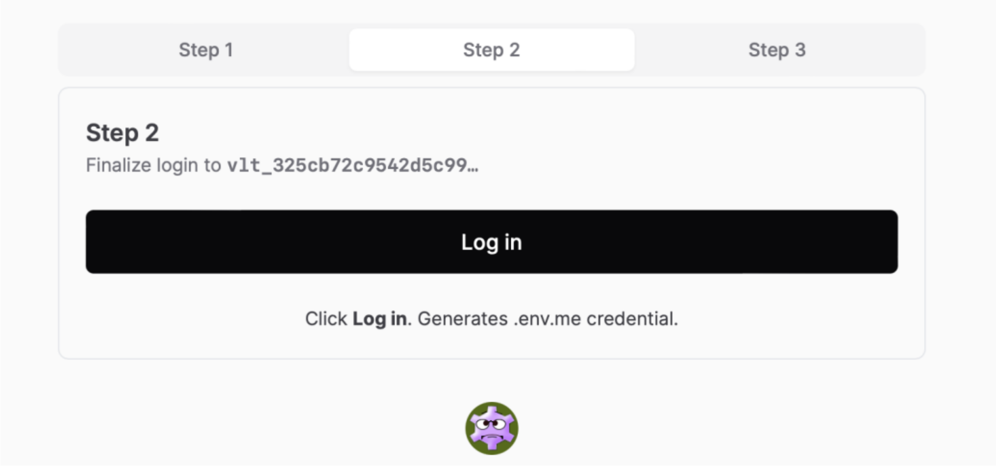
Project name *
back

Email *
Enter your email address

Already have an account? Log in

On the next page, run the **login** command.

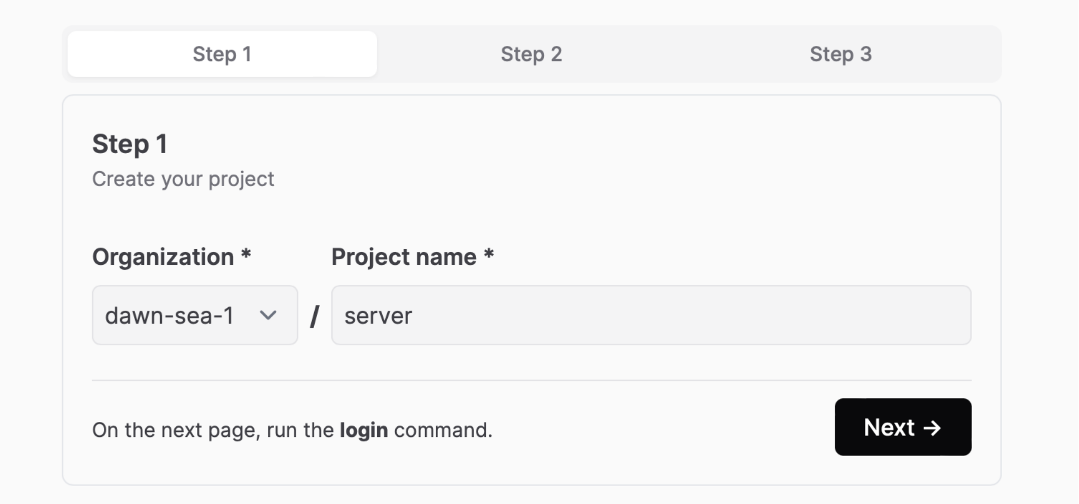
You may also need to provide login information.



Step 2
Finalize login to vlt_325cb72c9542d5c99…

Click **Log in**. Generates .env.me credential.

After running latest new command, the browser is opened with some information. Press Next.



Step 1

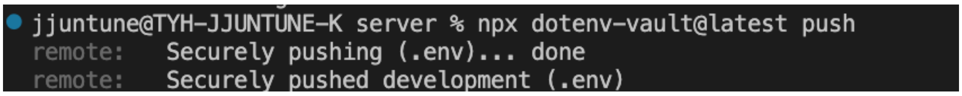
Create your project

Organization * : dawn-sea-1

Project name * : server

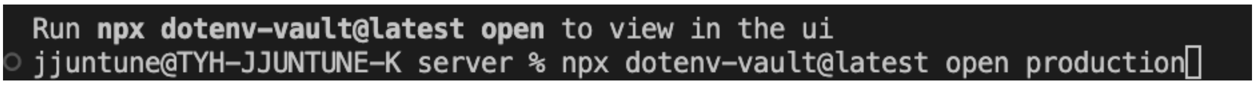
On the next page, run the **login** command.

Next push local .env file to dotenv server.



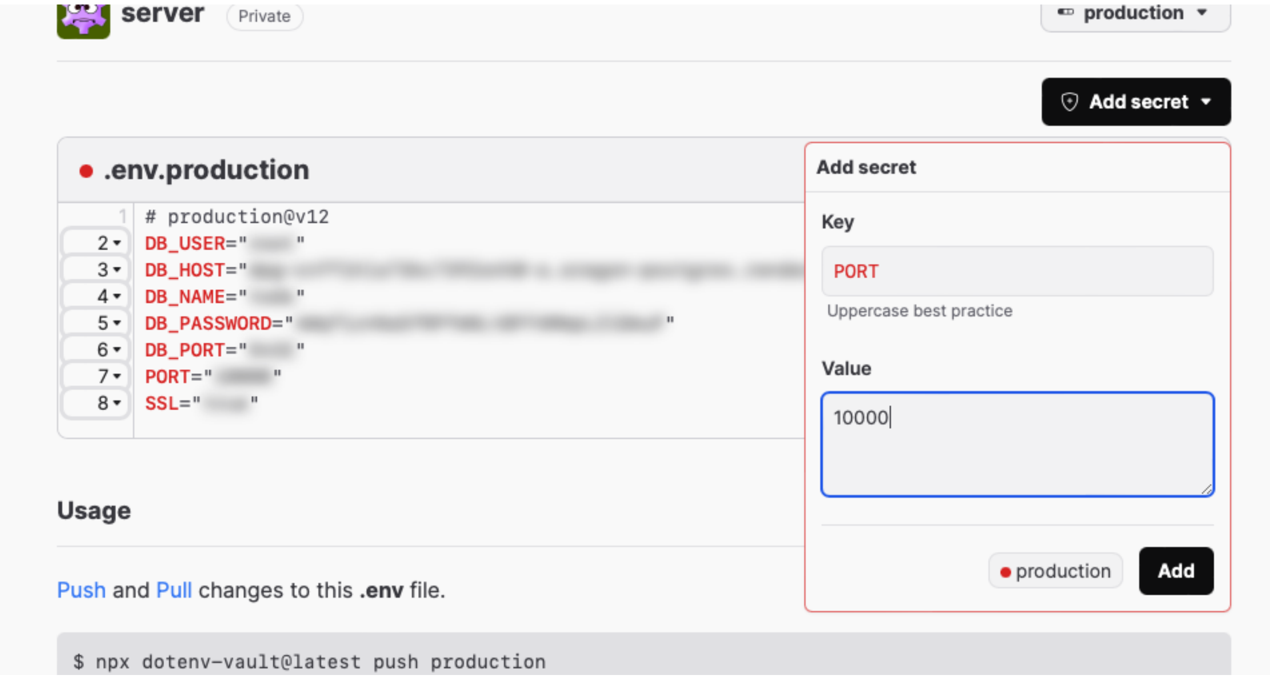
```
jjuntune@TYH-JJUNTUNE-K server % npx dotenv-vault@latest push
remote:   Securely pushing (.env)... done
remote:   Securely pushed development (.env)
```

Open the graphical editing tool in a browser running *npx dotenv-vault@latest open production* command.



```
Run npx dotenv-vault@latest open to view in the ui
jjuntune@TYH-JJUNTUNE-K server % npx dotenv-vault@latest open production
```

Edit correct values to keys. You may also remove (arrow down on the left side of the key with the line number) and add required keys (Select Add secret).



```
# production@v12
DB_USER="[redacted]"
DB_HOST="[redacted]"
DB_NAME="[redacted]"
DB_PASSWORD="[redacted]"
DB_PORT="[redacted]"
PORT="[redacted]"
SSL="[redacted]"
```

Key: PORT

Value: 10000

Push and Pull changes to this **.env** file.

```
$ npx dotenv-vault@latest push production
```

Correct keys are something as follows.

DB_USER: root (or whatever value you provided while creating database on Render)

DB_HOST: the same value that was provided for connection in pgAdmin (e.g. dpg-cnff1hla73kc7392onh0-a.oregon-postgres.render.com)

DB_NAME: Name of the database, you may check this also from pgAdmin (e.g. todo_1b0b)

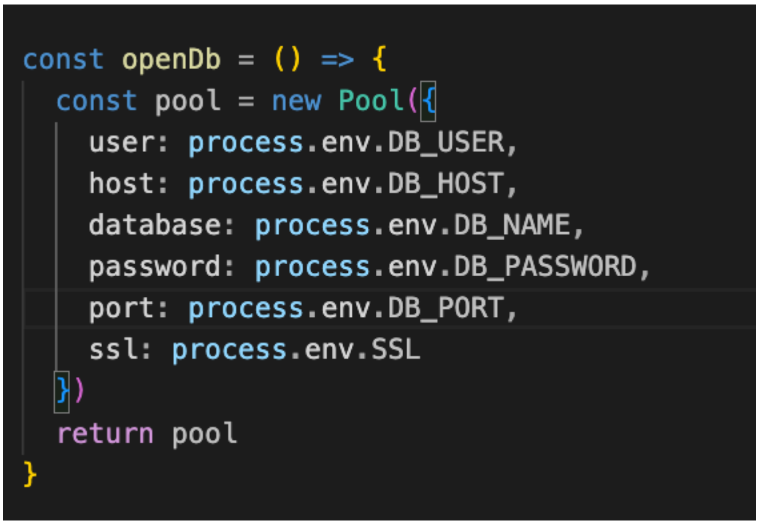
DB_PASSWORD: Postgres password on Render (was used to login to the database using pgAdmin)

DB_PORT: On Render this is by default 5432

PORT: On Render this is by default 10000

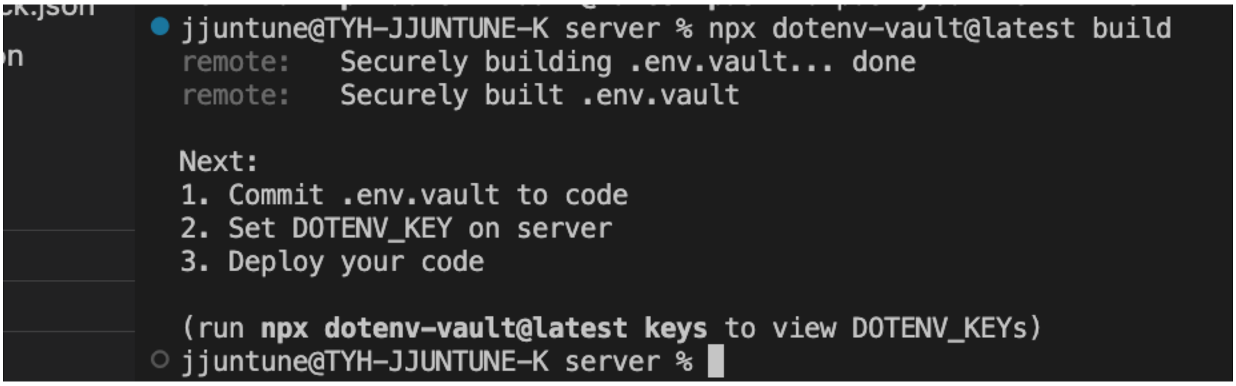
SSL: true

Edit database connection details on your backend code. Database connection requires SSL to be set to true in order to work.



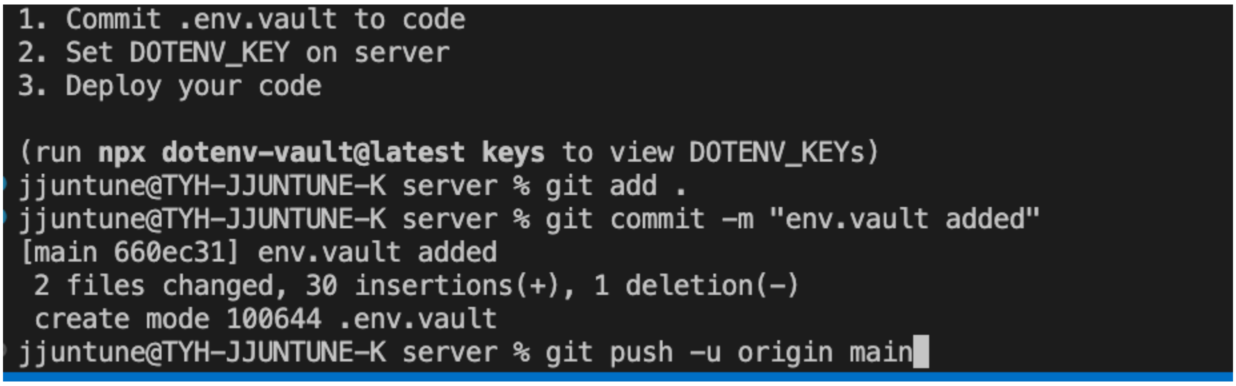
```
const openDb = () => {
  const pool = new Pool({
    user: process.env.DB_USER,
    host: process.env.DB_HOST,
    database: process.env.DB_NAME,
    password: process.env.DB_PASSWORD,
    port: process.env.DB_PORT,
    ssl: process.env.SSL
  })
  return pool
}
```

Build env.vault file.



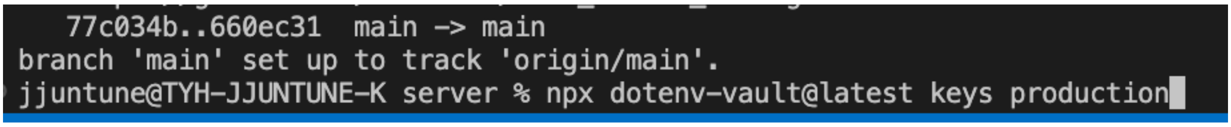
```
● jjuntune@TYH-JJUNTUNE-K server % npx dotenv-vault@latest build
  remote:   Securely building .env.vault... done
  remote:   Securely built .env.vault

  Next:
  1. Commit .env.vault to code
  2. Set DOTENV_KEY on server
  3. Deploy your code

  (run npx dotenv-vault@latest keys to view DOTENV_KEYs)
○ jjuntune@TYH-JJUNTUNE-K server %
```

Add, commit, and push your latest server code to the git. Make sure, that env.vault is committed.



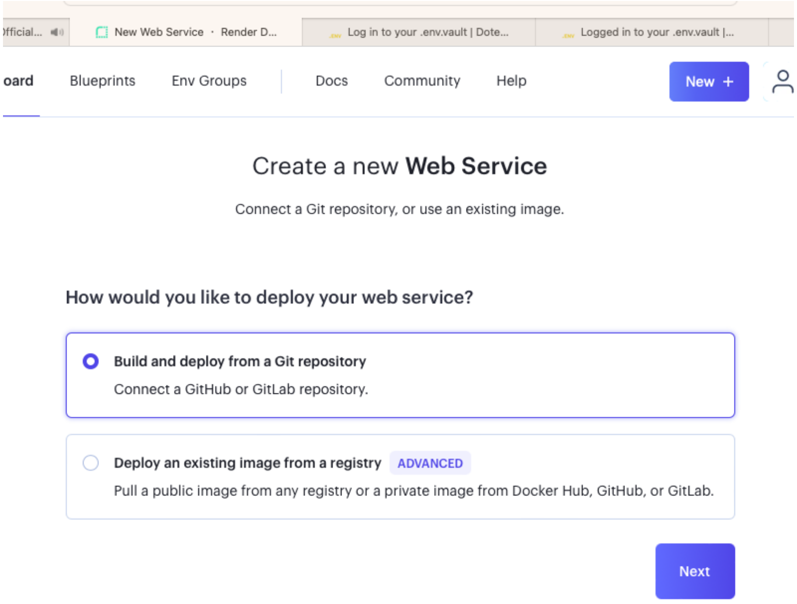
```
1. Commit .env.vault to code
2. Set DOTENV_KEY on server
3. Deploy your code

(run npx dotenv-vault@latest keys to view DOTENV_KEYs)
jjuntune@TYH-JJUNTUNE-K server % git add .
jjuntune@TYH-JJUNTUNE-K server % git commit -m "env.vault added"
[main 660ec31] env.vault added
 2 files changed, 30 insertions(+), 1 deletion(-)
 create mode 100644 .env.vault
jjuntune@TYH-JJUNTUNE-K server % git push -u origin main
```

Run the following command to enable Render to use defined keys from Vault. Command will generate a key which should be stored since you will need it later.



```
   77c034b..660ec31  main -> main
branch 'main' set up to track 'origin/main'.
jjuntune@TYH-JJUNTUNE-K server % npx dotenv-vault@latest keys production
```

Create new web service on render.com by selectin New -> Web Service.


Official...
New Web Service · Render D...
Log in to your .env.vault | Dote...
Logged in to your .env.vault |...
oard
Blueprints
Env Groups
Docs
Community
Help
New +

Create a new **Web Service**

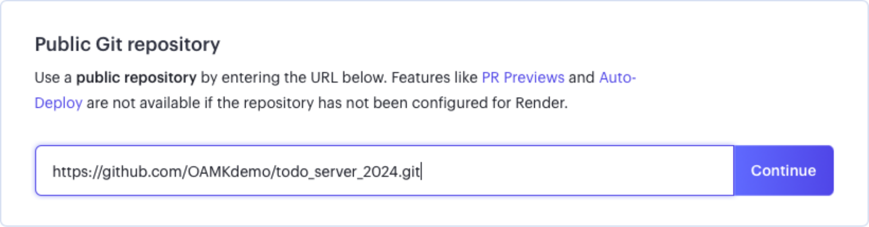
Connect a Git repository, or use an existing image.

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.

Provide public git address to the backend code and hit Continue.

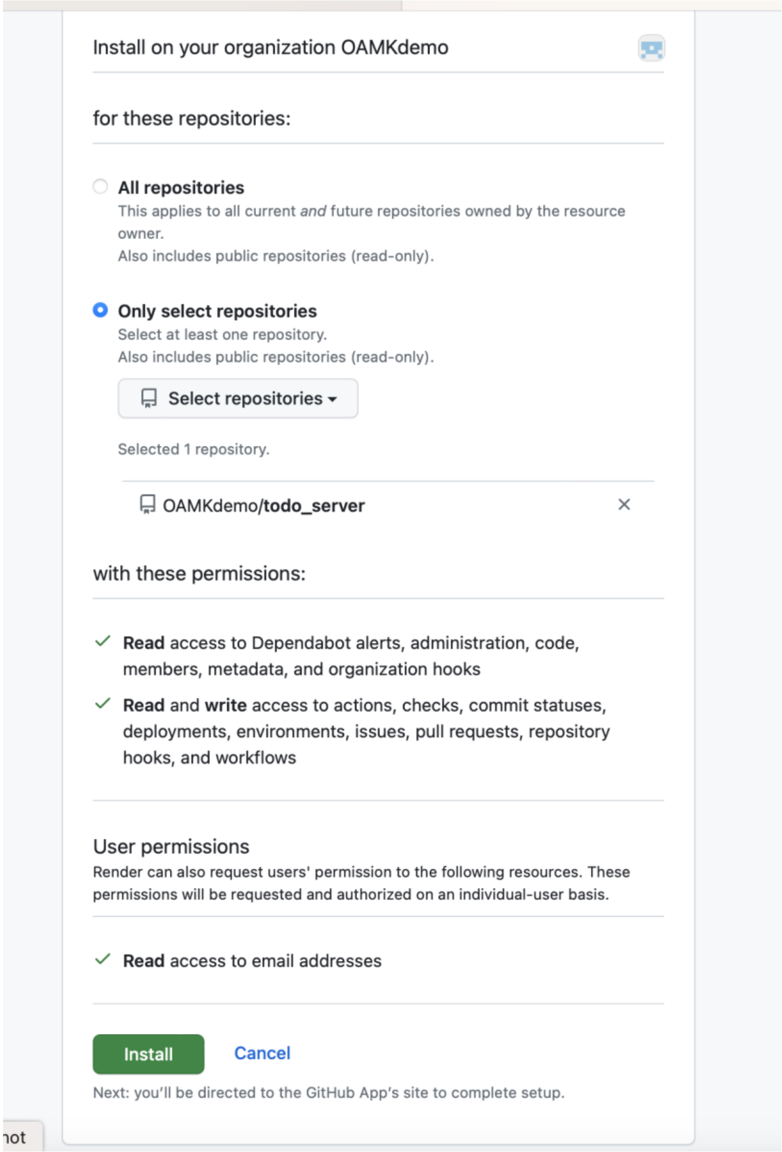


Public Git repository

Use a **public repository** by entering the URL below. Features like PR Previews and Auto-Deploy are not available if the repository has not been configured for Render.

https://github.com/OAMKdemo/todo_server_2024.git

You may be prompted to install integration for Render on Github. Follow the instructions to make integration.



Install on your organization OAMKdemo

for these repositories:

○ **All repositories**
This applies to all current *and* future repositories owned by the resource owner.
Also includes public repositories (read-only).

● **Only select repositories**
Select at least one repository.
Also includes public repositories (read-only).

Selected 1 repository.

OAMKdemo/**todo_server** ×

with these permissions:

✓ **Read** access to Dependabot alerts, administration, code, members, metadata, and organization hooks

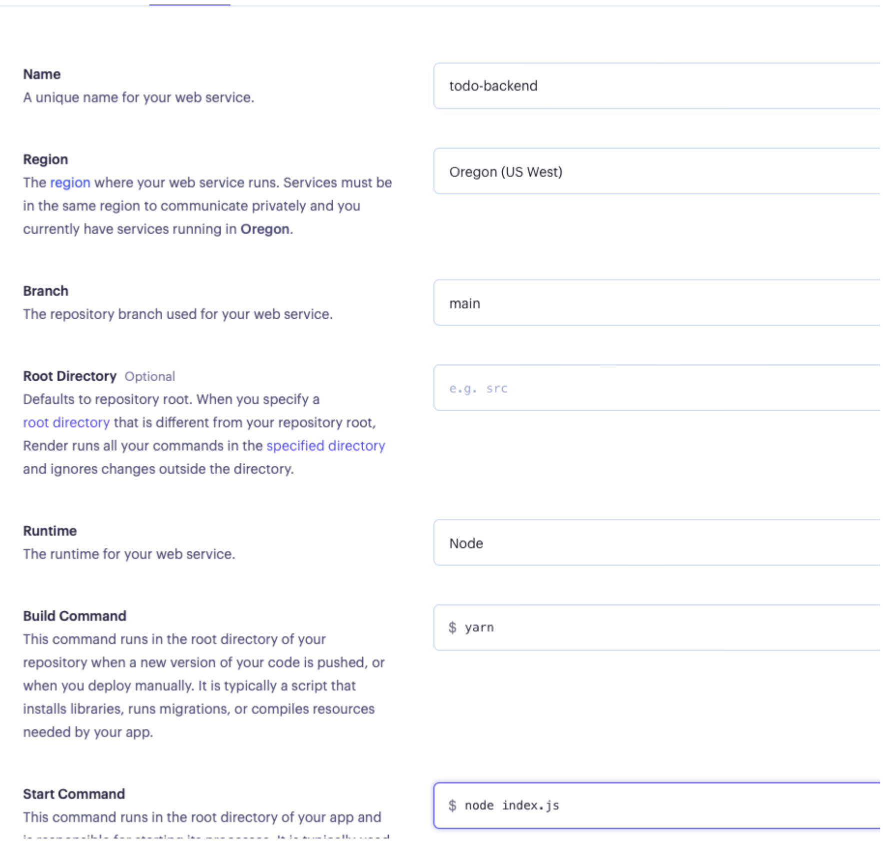
✓ **Read** and **write** access to actions, checks, commit statuses, deployments, environments, issues, pull requests, repository hooks, and workflows

User permissions
Render can also request users' permission to the following resources. These permissions will be requested and authorized on an individual-user basis.

✓ **Read** access to email addresses

Next: you'll be directed to the GitHub App's site to complete setup.

Provide required information. Make sure, that start command is *node index.js*. Build command can be default (npm i) and need not to be changed.



Name
A unique name for your web service.
todo-backend

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

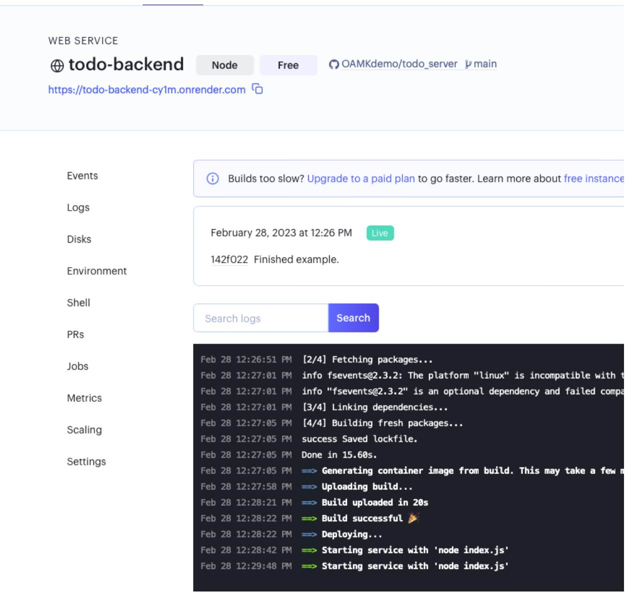
Root Directory Optional
Defaults to repository root. When you specify a root directory that is different from your repository root, Render runs all your commands in the specified directory and ignores changes outside the directory.
e.g. src

Runtime
The runtime for your web service.
Node

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.
$ yarn

Start Command
This command runs in the root directory of your app and
$ node index.js

Select free instance type and press Create Web Service(blue) button (on the bottom of the page).



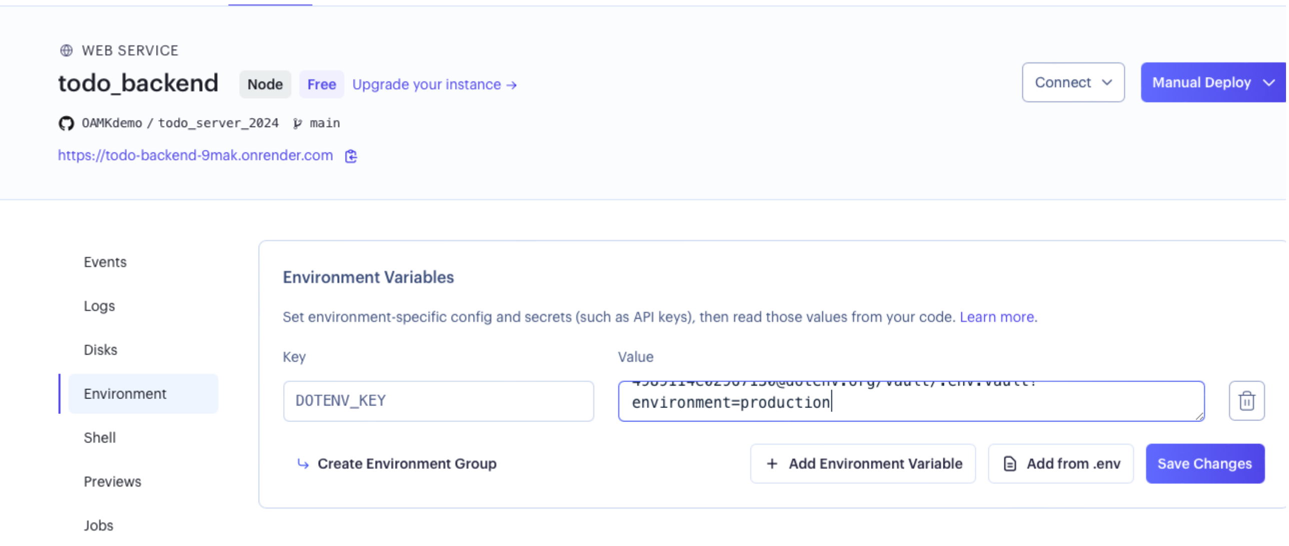
WEB SERVICE
todo-backend Node Free OAMKdemo/todo_server main
https://todo-backend-cy1m.onrender.com

Builds too slow? Upgrade to a paid plan to go faster. Learn more about free instance

February 28, 2023 at 12:26 PM Live
142f022 Finished example.

```
Feb 28 12:26:51 PM  [2/4] Fetching packages...
Feb 28 12:27:01 PM  info fsevents@2.3.2: The platform "linux" is incompatible with t
Feb 28 12:27:01 PM  info "fsevents@2.3.2" is an optional dependency and failed compa
Feb 28 12:27:01 PM  [3/4] Linking dependencies...
Feb 28 12:27:05 PM  [4/4] Building fresh packages...
Feb 28 12:27:05 PM  success Saved lockfile.
Feb 28 12:27:05 PM  Done in 15.60s.
Feb 28 12:27:05 PM  ==> Generating container image from build. This may take a few m
Feb 28 12:27:58 PM  ==> Uploading build...
Feb 28 12:28:21 PM  ==> Build uploaded in 20s
Feb 28 12:28:22 PM  ==> Build successful 🎉
Feb 28 12:28:22 PM  ==> Deploying...
Feb 28 12:28:42 PM  ==> Starting service with 'node index.js'
Feb 28 12:29:48 PM  ==> Starting service with 'node index.js'
```

Add required DOT_ENV key under Environment. Use the that was generated by the *npx dotenv-vault@latest keys production* command.

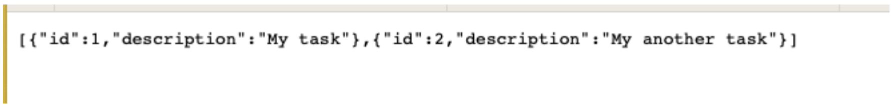


Environment Variables

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

Key	Value
DOTENV_KEY	environment=production

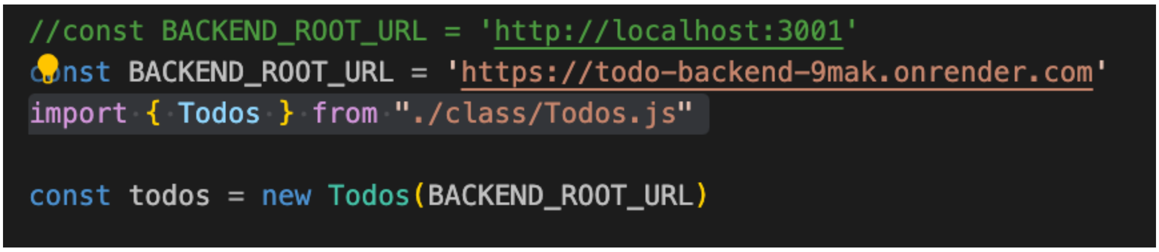
Wait that service gets deployed. Test out that you can call the backend (using a browser or you might use REST Client as well). The backend is returning simple JSON data.



```
[{"id":1,"description":"My task"},{"id":2,"description":"My another task"}]
```

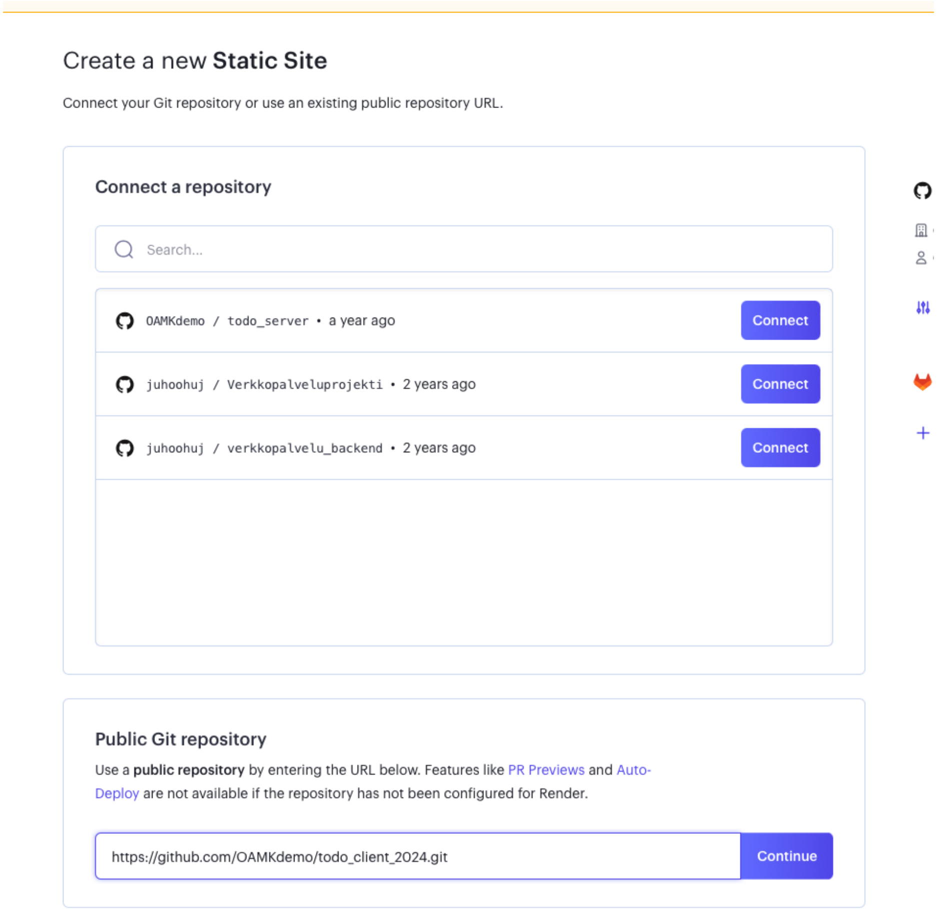
Do **NOT** proceed if you are not able to run the backend. Check the instructions, that you have made all configurations correctly. When database and backend services are running you are ready to deploy the client.

Change the backend url for the client. Again, you can test the website running it locally (and remember it might be quite slow). Now front-end (client) is working on your workstation but the backend is called from render.com.



```
//const BACKEND_ROOT_URL = 'http://localhost:3001'
const BACKEND_ROOT_URL = 'https://todo-backend-9mak.onrender.com'
import { Todos } from "./class/Todos.js"

const todos = new Todos(BACKEND_ROOT_URL)
```

Install frontend as Static Site, since it contains only HTML, CSS and JavaScript Provide public git address containing front-end code (and of course push your code to git before that).



Create a new **Static Site**

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

Connect a repository

Search...

OAMKdemo / todo_server • a year ago — Connect

juhoohuj / Verkkopalveluprojekti • 2 years ago — Connect

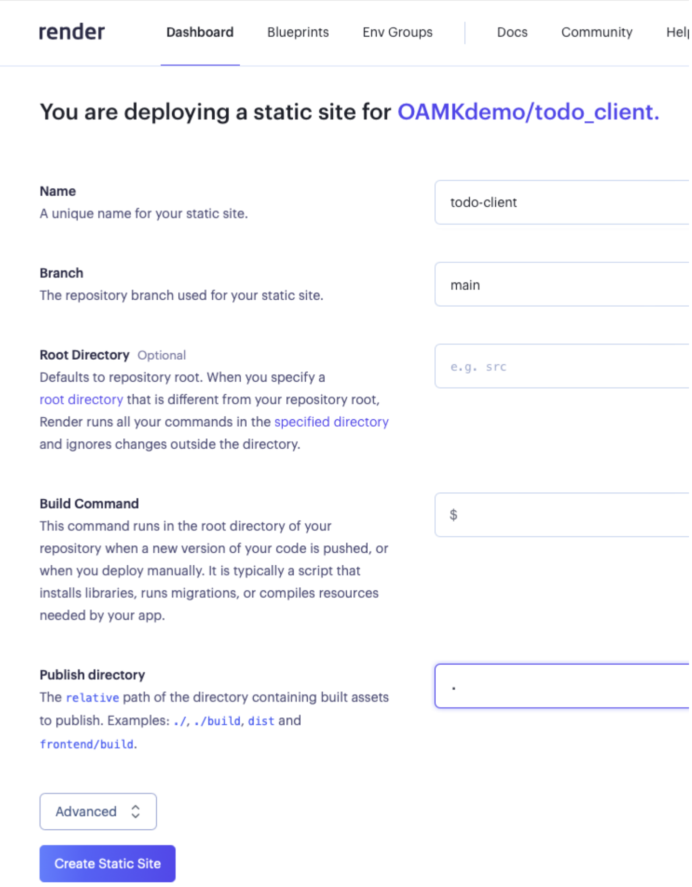
juhoohuj / verkkopalvelu_backend • 2 years ago — Connect

Public Git repository

Use a **public repository** by entering the URL below. Features like PR Previews and Auto-Deploy are not available if the repository has not been configured for Render.

https://github.com/OAMKdemo/todo_client_2024.git

Give a name and publish directory, which is a dot (.). Press Create Static Site button.



render Dashboard Blueprints Env Groups Docs Community Hel

You are deploying a static site for OAMKdemo/todo_client.

Name
A unique name for your static site.
todo-client

Branch
The repository branch used for your static site.
main

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

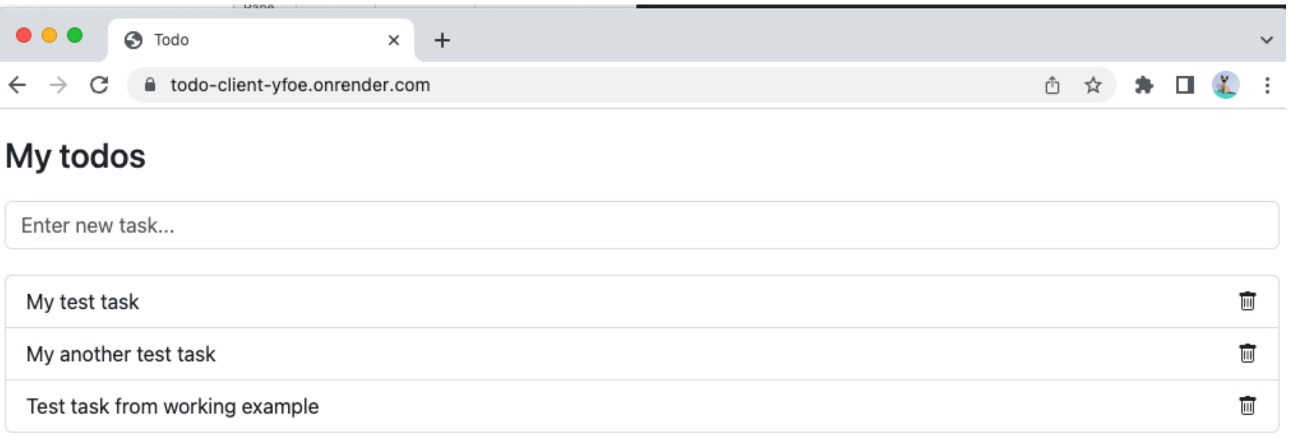
Build Command
This command runs in the root directory of your repository when a new version of your code is pushed, or when you deploy manually. It is typically a script that installs libraries, runs migrations, or compiles resources needed by your app.
$

Publish directory
The `relative` path of the directory containing built assets to publish. Examples: `./`, `./build`, `dist` and `frontend/build`.
.

Advanced

Create Static Site

After deployment is finished, front-end should also work.



Todo

todo-client-yfoe.onrender.com

My todos

Enter new task...

My test task

My another test task

Test task from working example