Introduction to Render Cloud Platform

Render is a unified cloud platform to build and run apps and websites. Render provides all services in one place, including web services, static sites, background workers, cron jobs, Dockerfiles, private services, PostgreSQL, and Redis.

Since Heroku discontinued the free-tier account, Render is an alternative that provides free-tier services for small projects and hobbyists. See Render's pricing plans here.

You will learn how to deploy a Flask app and Postgres database on Render Console in the following steps:

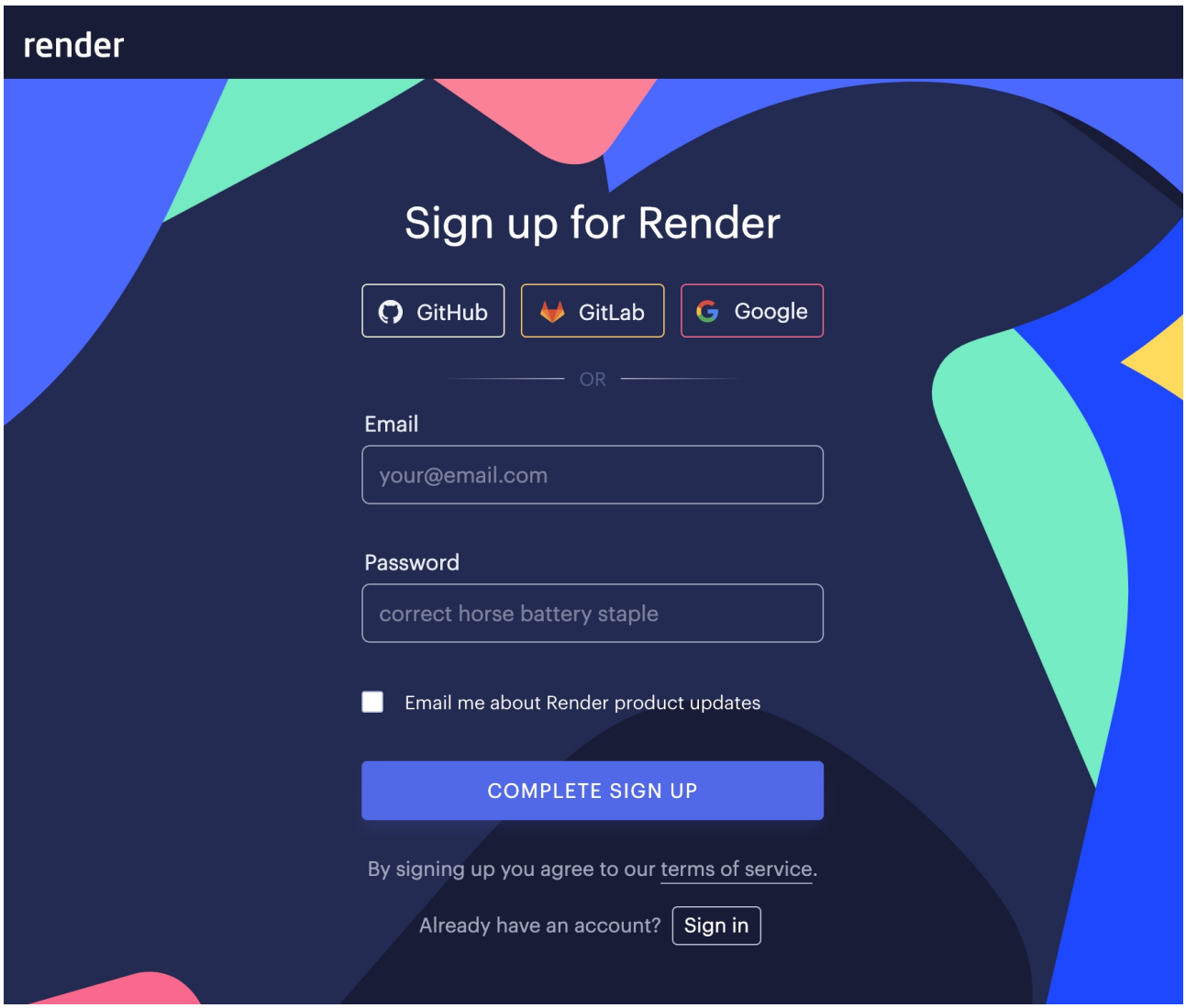
1. Create a Render account
2. Set up a Database Service with Postgres
3. Deploy a Flask app with Render's Web Service

You can download or clone the Flask app example for the exercise below from this GitHub repo.

After you complete this exercise, please suspend or delete the services to avoid any charges.

Create a Render Account

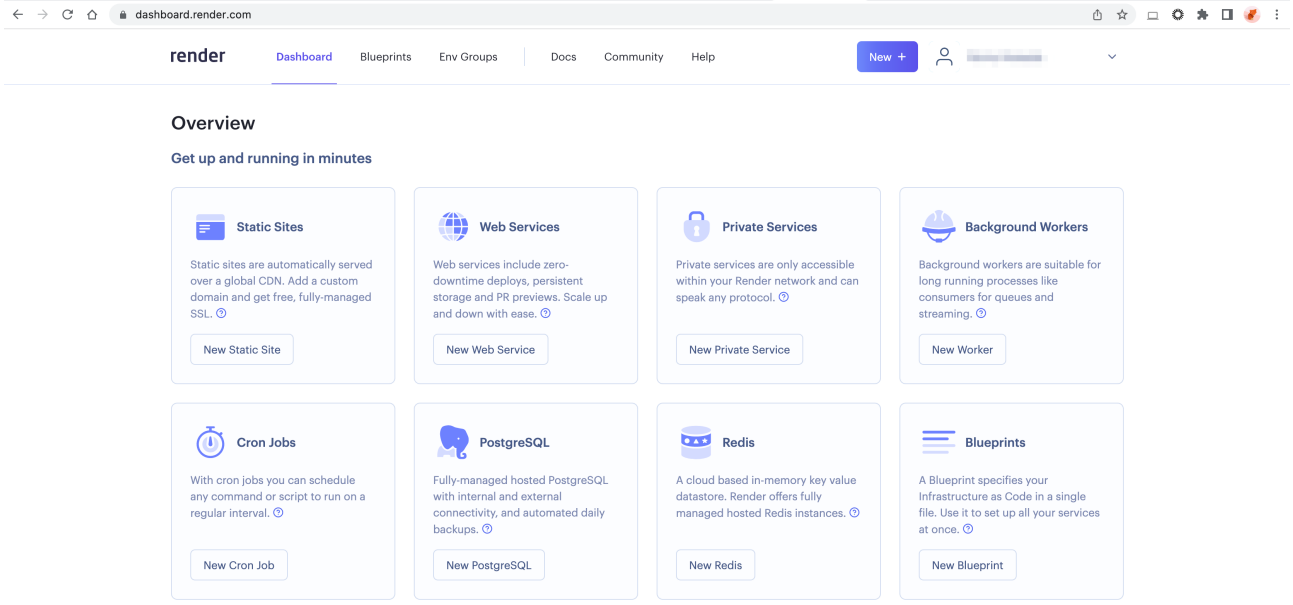
From the Render.com landing page, click the "Get Started" button to open the sign-up page. You can create an account by linking your GitHub, GitLab, or Google account or provide your email and password.



Registration Page

Set up a Database Service with Postgres

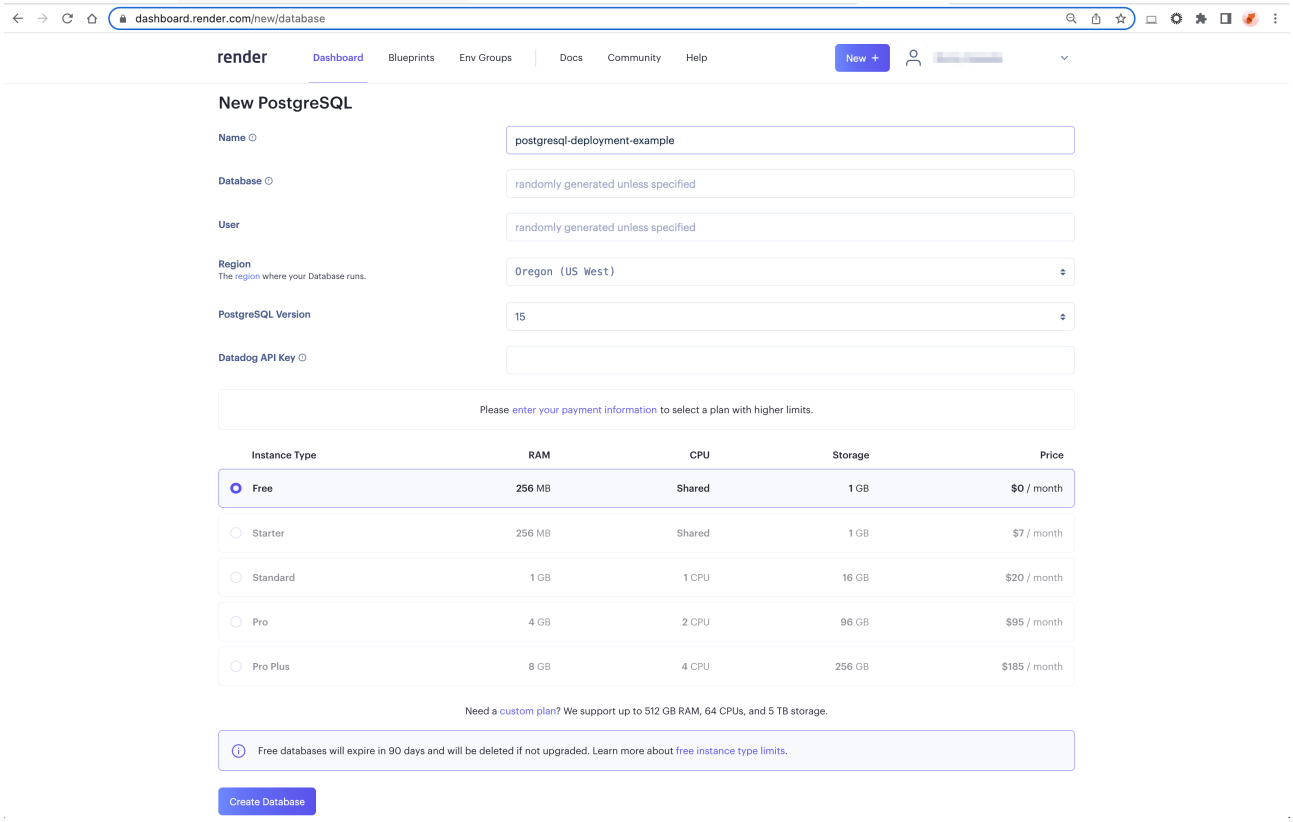
Once you are logged in, you will be redirected to the Render Dashboard. Click the `New Postgres` button to set up a Postgres cloud database.



Render Dashboard

On the "New Postgres" page:

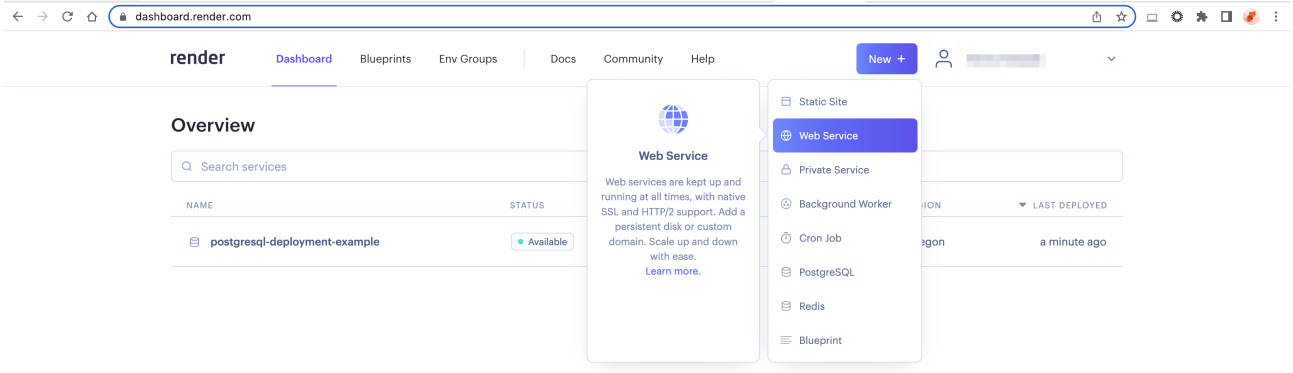
1. Provide a name for the new database service: `postgres-deployment-example`
2. Select an instance type: `Free`
3. Click `Create Database` button



Create Postgres Database

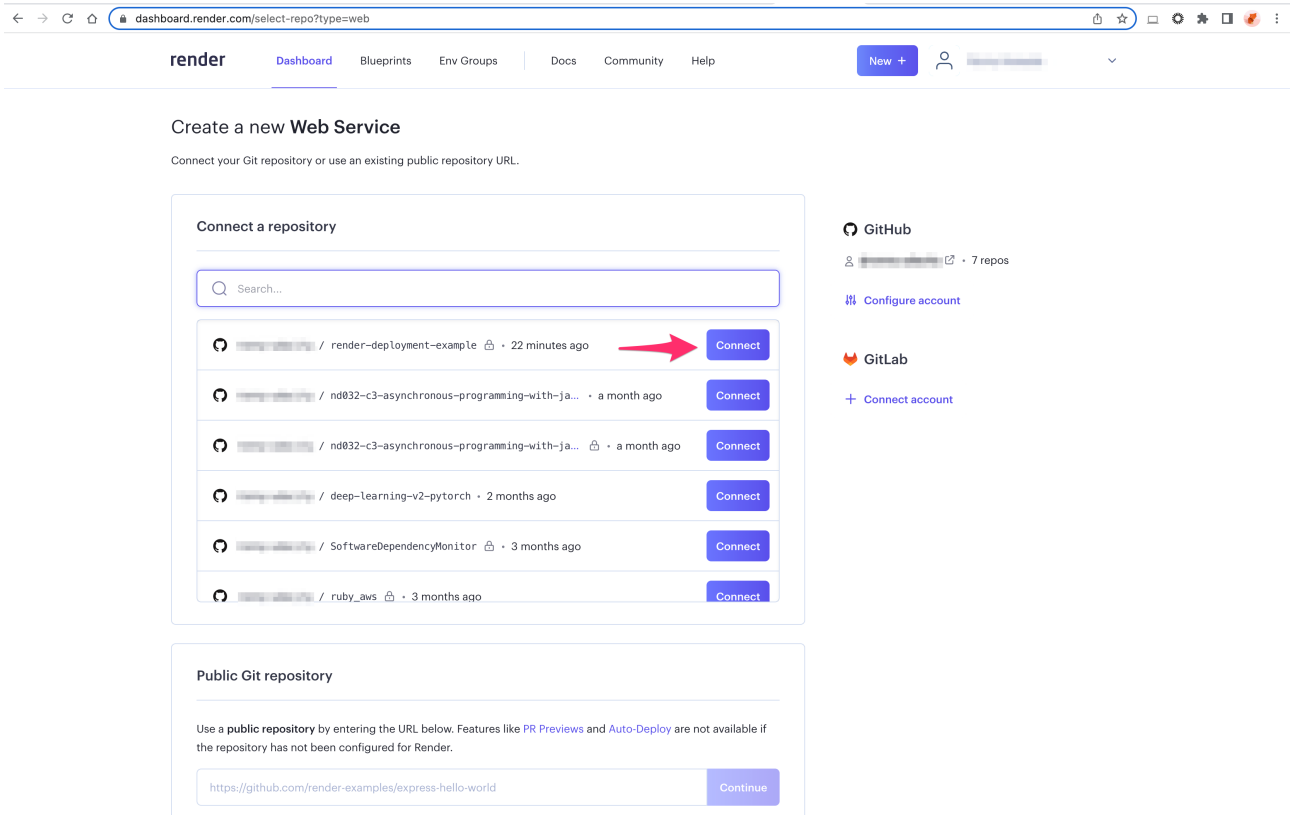
Deploy Apps with Render's Web Service

Once the database is set up, we can go back to Render Dashboard and create a new `Web Service`.



Create a Web Service

Connect your Flask app from GitHub or GitLab repo to the Web Service

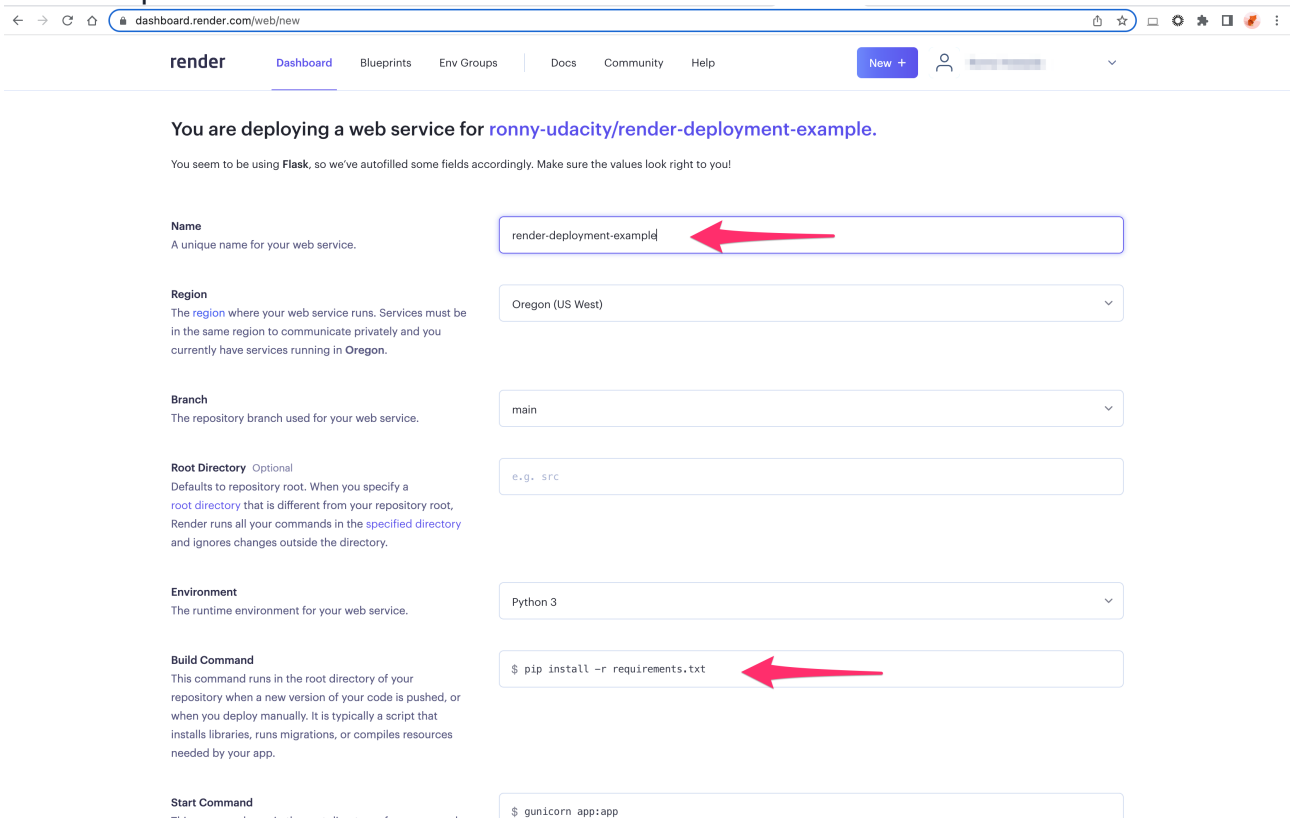


Connect a Flask app from GitHub or GitLab repo to Render Web Service

On the "New Web Service" page:

1. Provide a name for the new database service: `render-deployment-example`
2. Select an instance type: `Free`
3. Enter the build command: `pip install -r requirements.txt`

Note: Render will install the dependencies from the "requirements.txt" provided in the GitHub repo.



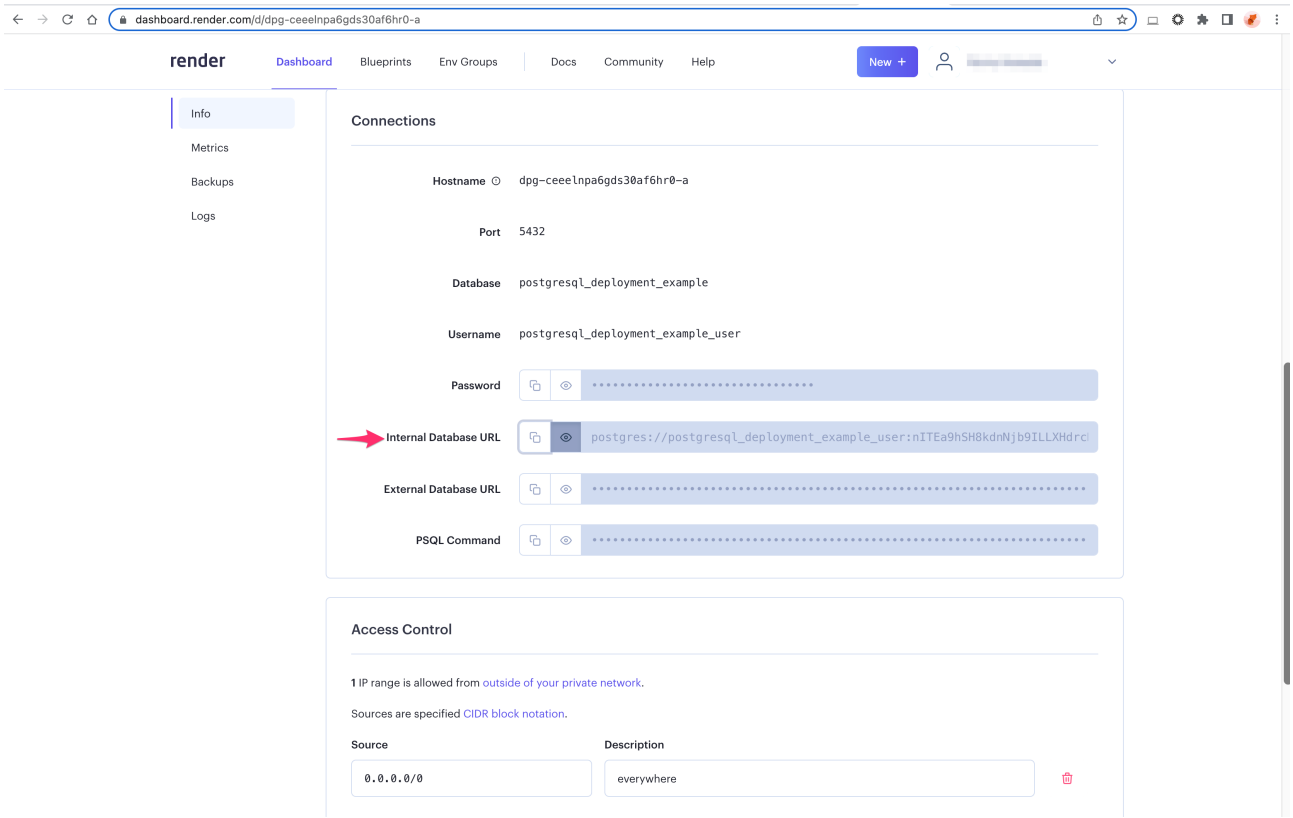
Create a new Web Service

Connect the Database Service and Web Service

Before you click `Create Web Service`, you will need to connect the Postgres service so your Flask app can read and write data to the Postgres database. To connect the services, you can copy the Postgres database URL and paste it into the environment variables within the web service

Copy Postgres Database URL

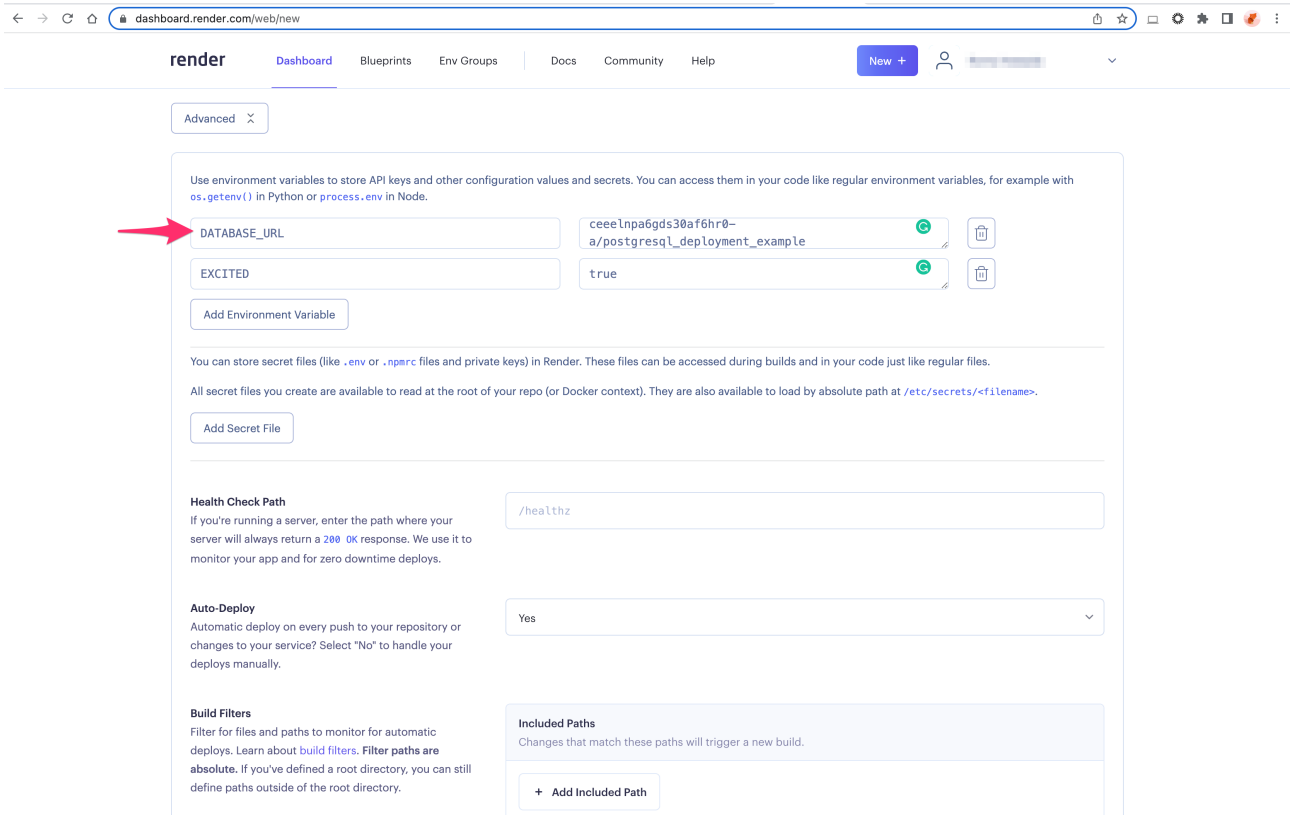
From the Postgres service (name: "postgres-deployment-example"), click the "Info" side navigation and copy the `Internal Database URL` from the Connections page.



Copy Postgres Database URL

Paste the Database URL in the Web Service Environment Variable

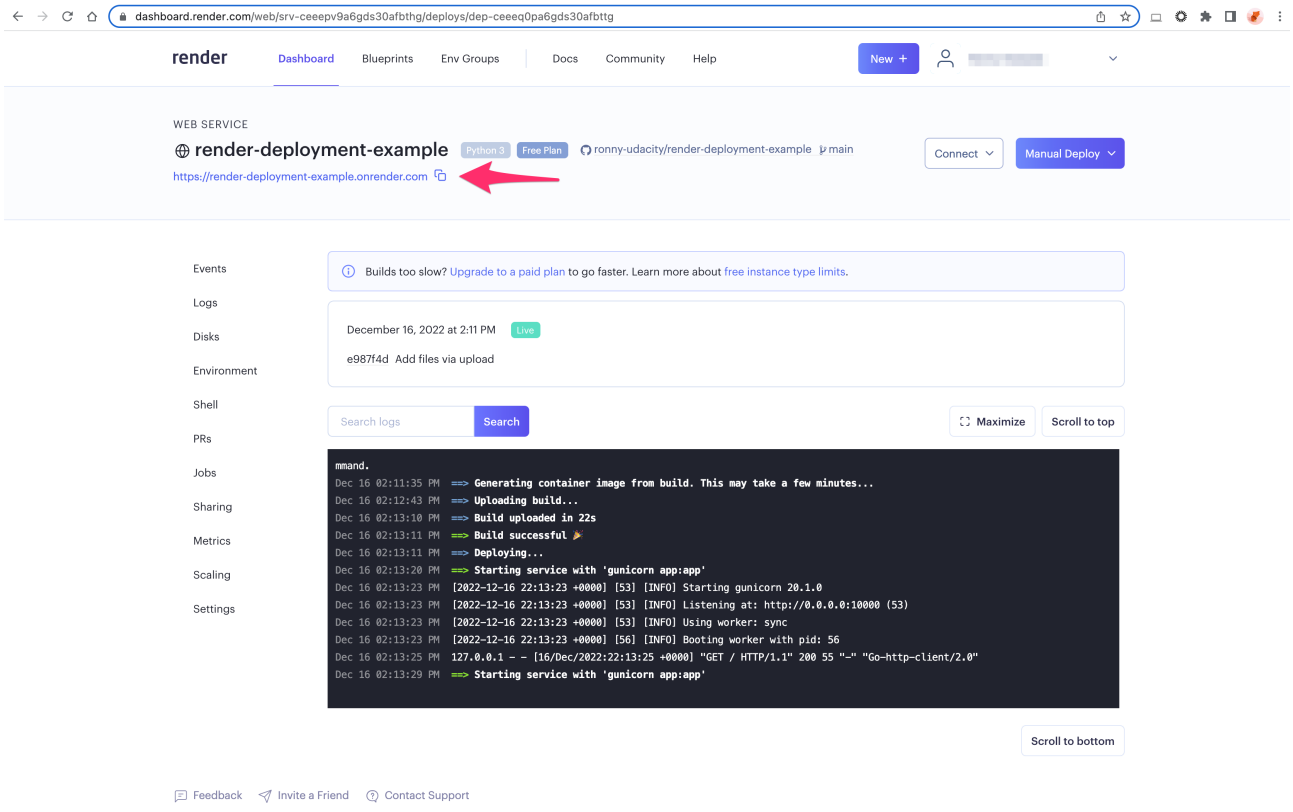
From the web service (name: "render-deployment-example"), create an environment variable with the key: `DATABASE_URL` and value: `the <Database URL> copied from the Postgres service`.

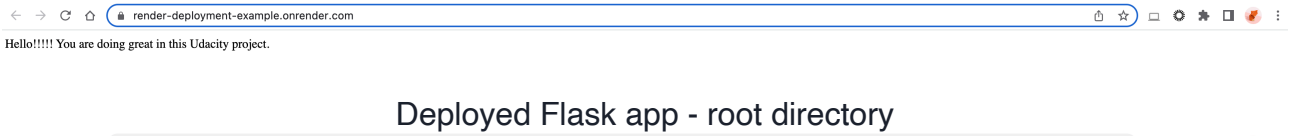


Paste DATABASE_URL into Web Service environment variable

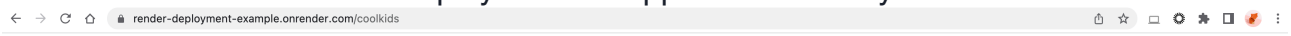
Note: The Flask app will use the second environment variable ("EXCITED: true"). You can store any other credentials for your apps by adding the environment variables.

After the Web Service is ready, you can open your Flask app on the browser by clicking the `App URL`.





Deployed Flask app - root directory



Deployed Flask app - /coolkids directory

After you complete this exercise, please suspend or delete the services to avoid any charges.

Render CLI

At this time of writing, Render is working on the `render-cli` in the alpha version. Please refer to the announcement page if you would like to test and provide feedback about the CLI.