

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.

render


Sign up for Render



GitHub



GitLab



Google

OR

Email

your@email.com

Password

correct horse battery staple

☐ Email me about Render product updates

COMPLETE SIGN UP

By signing up you agree to our [terms of service](#).

Already have an account?

Sign in

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.

← → ↺ ⌂ dashboard.render.com

render

Dashboard

Blueprints


Env Groups


Docs


Community


Help

New +










Overview


Get up and running 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

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

render Dashboard Blueprints Env Groups Docs Community Help New +

New PostgreSQL

Name

Database

User

Region The region where your Database runs.

PostgreSQL Version

Datadog API Key

Please [enter your payment information](#) to select a plan with higher limits.

| Instance Type | RAM | CPU | Storage | Price |
|---------------------------------------|--------|--------|---------|---------------|
| <input checked="" type="radio"/> Free | 256 MB | Shared | 1 GB | \$0 / month |
| <input type="radio"/> Starter | 256 MB | Shared | 1 GB | \$7 / month |
| <input type="radio"/> Standard | 1 GB | 1 CPU | 16 GB | \$20 / month |
| <input type="radio"/> Pro | 4 GB | 2 CPU | 96 GB | \$95 / month |
| <input type="radio"/> Pro Plus | 8 GB | 4 CPU | 256 GB | \$185 / month |

Need a custom plan? We support up to 512 GB RAM, 64 CPUs, and 5 TB storage.

free instance type limits."/>

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.

render Dashboard Blueprints Env Groups Docs Community Help New +

Overview

Search services

| NAME | STATUS |
|-------------------------------|-----------|
| postgresql-deployment-example | Available |

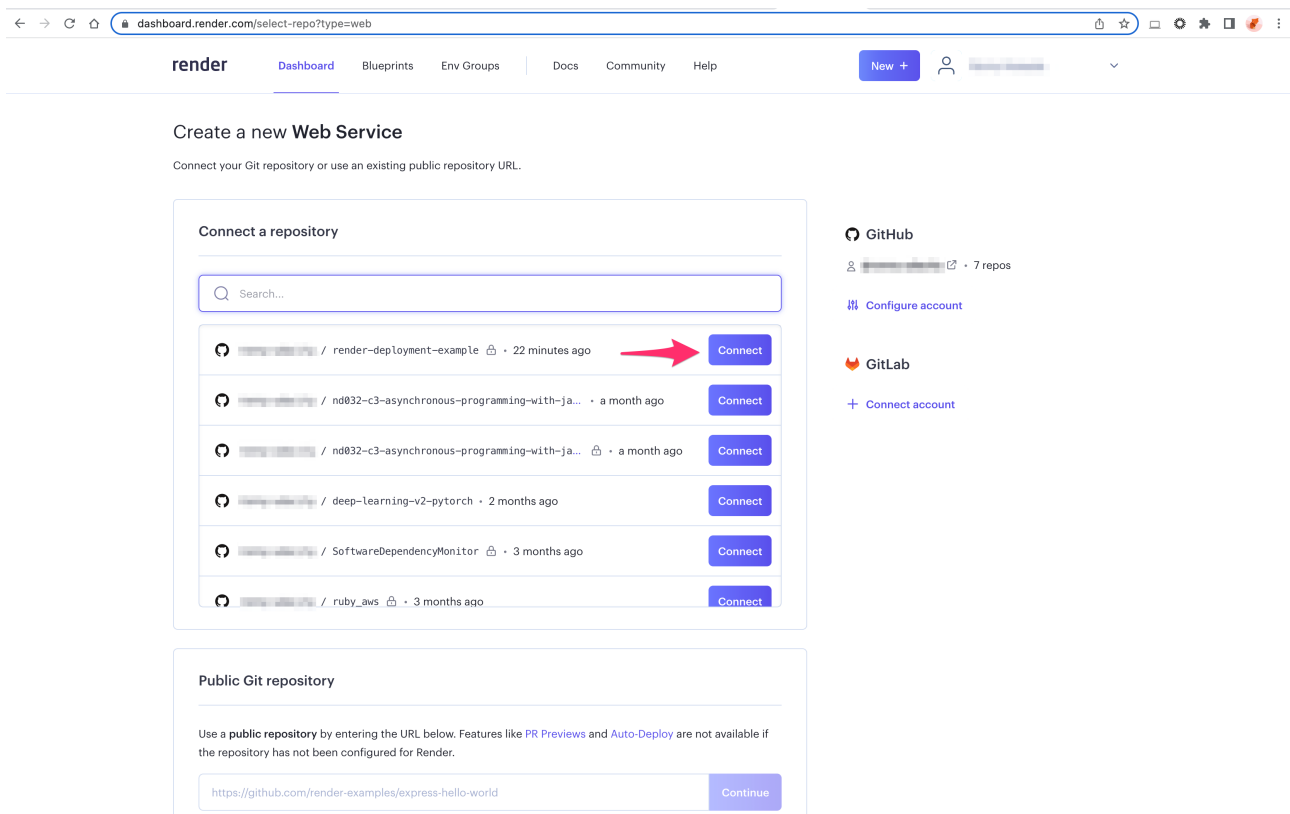
Web Service

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.](#)

- Static Site
- Web Service
- Private Service
- Background Worker
- Cron Job
- PostgreSQL
- Redis
- Blueprint

Create a Web Service

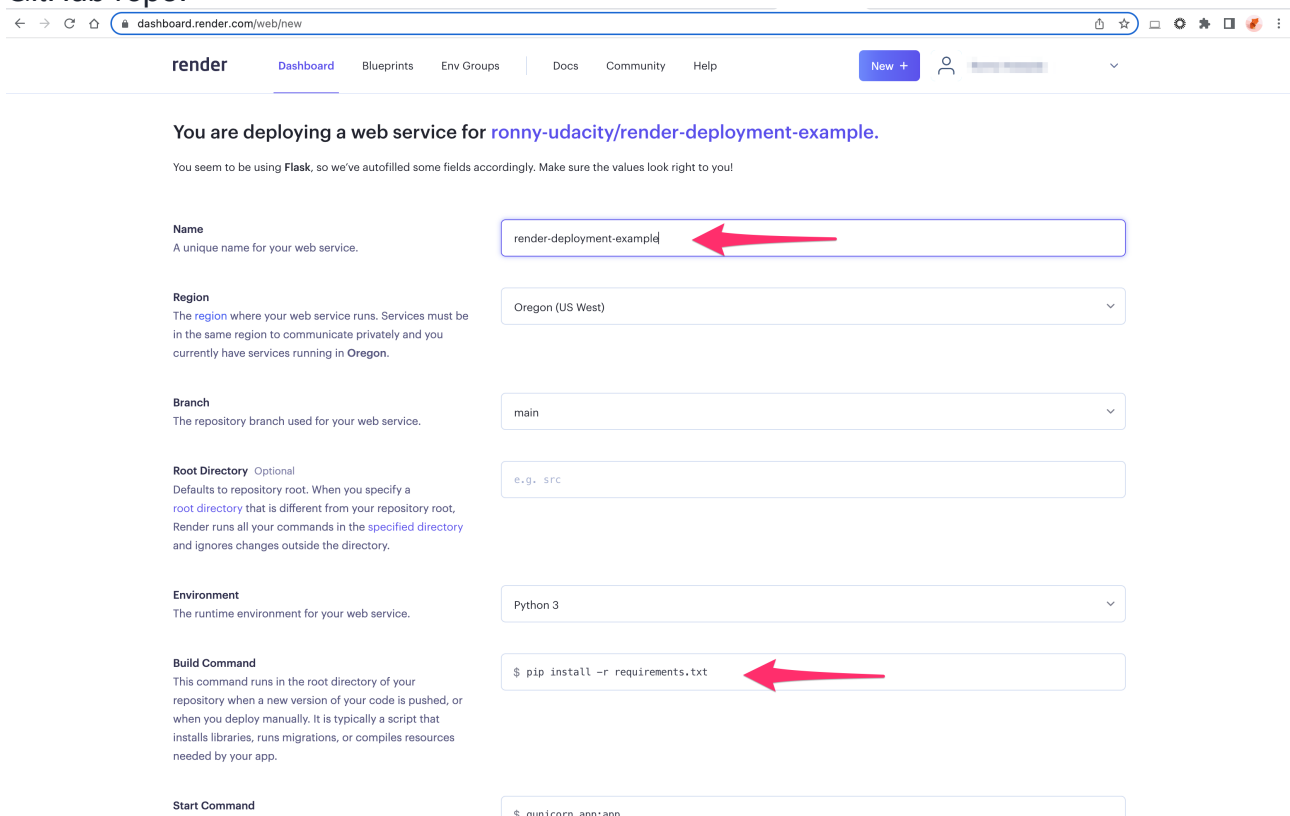
Connect your Flask app from GitHub or GitLab repo to the Web Service



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.



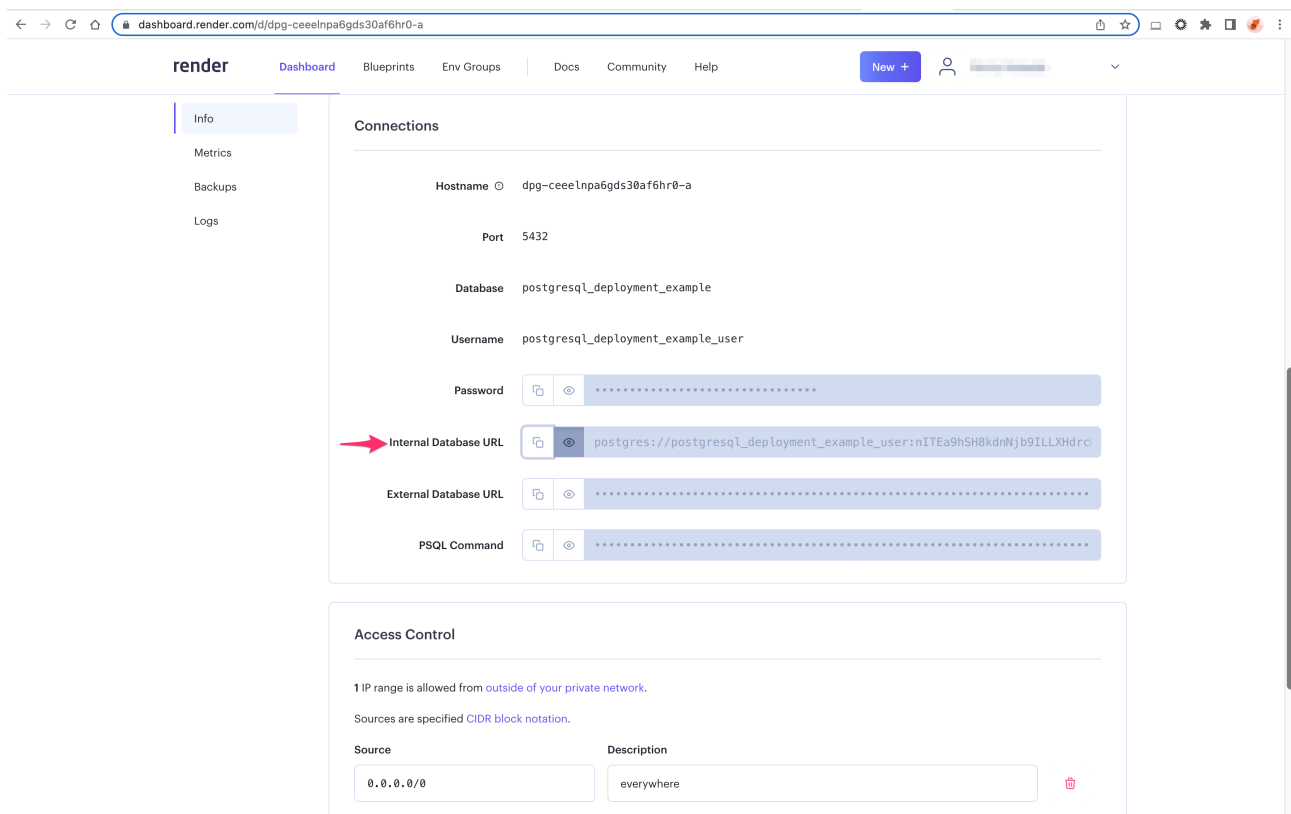
Create a new Web Service

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.

render-deployment-example.onrender.com
Hello!!!! You are doing great in this Udacity project.

Deployed Flask app - root directory

render-deployment-example.onrender.com/coolkids
Be cool, man, be cooooo! You're almost a FSND grad!

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.