Your First Render Deploy

Run your web app in minutes.

Welcome! Let's get up and running on Render.

This tutorial uses free Render resources—no payment required. All you need is a GitHub repo with the web app you want to deploy (GitLab and Bitbucket work too).

> **Want to deploy an example app using a particular language or framework?**
>
> Check out our quickstarts.

1. Sign up

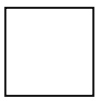
Signing up is fast and free:

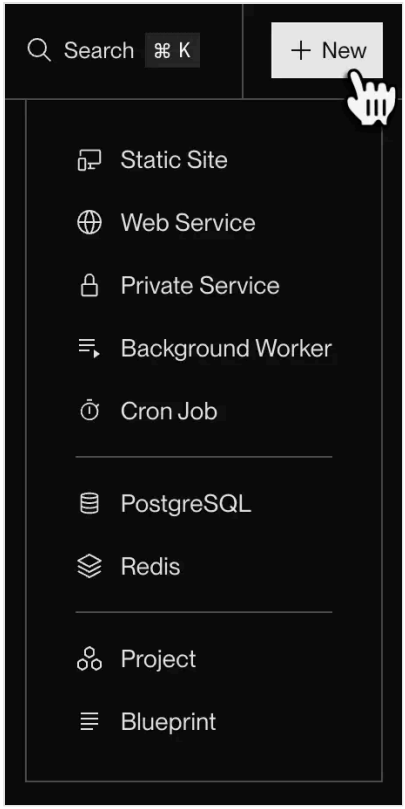
Sign up for Render

2. Choose a service type

To deploy to Render, you create a **service** that pulls, builds, and runs your code.

1. Launch the Render Dashboard.
2. In the top-right corner, open the **+ New** dropdown:





Here you select a **service type**.

For this tutorial, choose **Web Service** or **Static Site**:

SERVICE TYPE	DESCRIPTION	COMMON FRAMEWORKS
Web Service	Choose this if your web app runs any server-side code. The app also needs to listen for HTTP requests on a port. Full-stack web apps, API servers, and mobile backends are all web services.	Express, Next.js, Fastify, Django, FastAPI, Flask, Rails, Phoenix
Static Site	Choose this if your web app consists entirely of static content (mostly HTML/CSS/JS). Blogs, portfolios, and documentation sets are often (but not *always*) static sites.	Create React App, Vue.js, Hugo, Docusaurus, Next.js static exports

You can deploy either of these service types for free on Render.

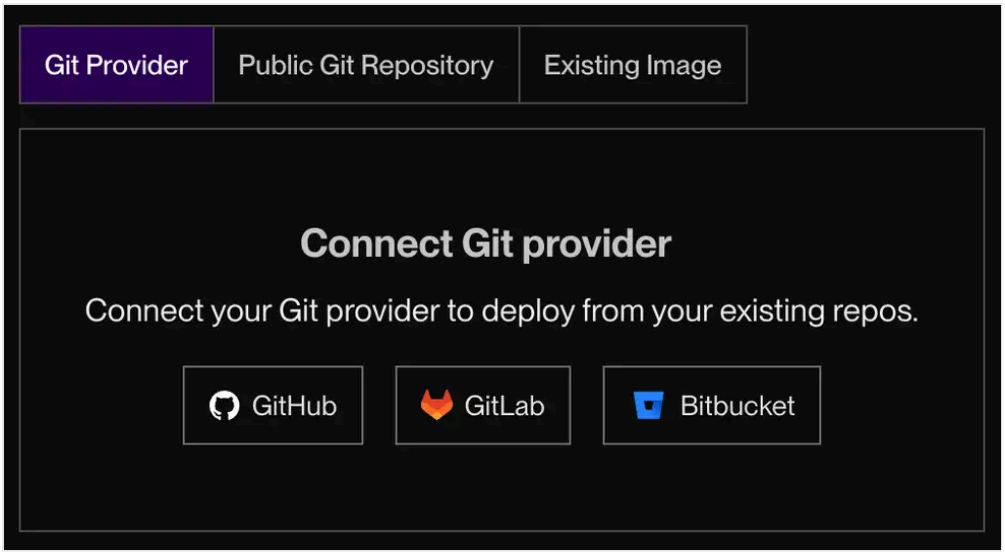
Free web services "spin down" after 15 minutes of inactivity.

They spin back up when they next receive incoming traffic. Learn more about free instance limitations.

3. Link your repo

After you select a service type, the service creation form appears.

1. First, connect your GitHub/GitLab/Bitbucket account to Render:



After you connect, the form shows a list of all the repos you have access to:

2. Select the repo that contains your web app and click **Connect**.
   The rest of the creation form appears.

4. Configure deployment

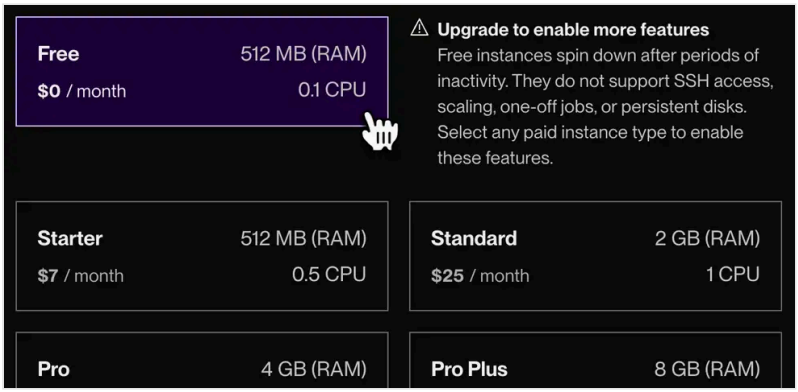
Complete the service creation form to define how Render will build and run your app.

Click the tab for your service type to view important field details:

Web Service | Static Site

Important web service fields

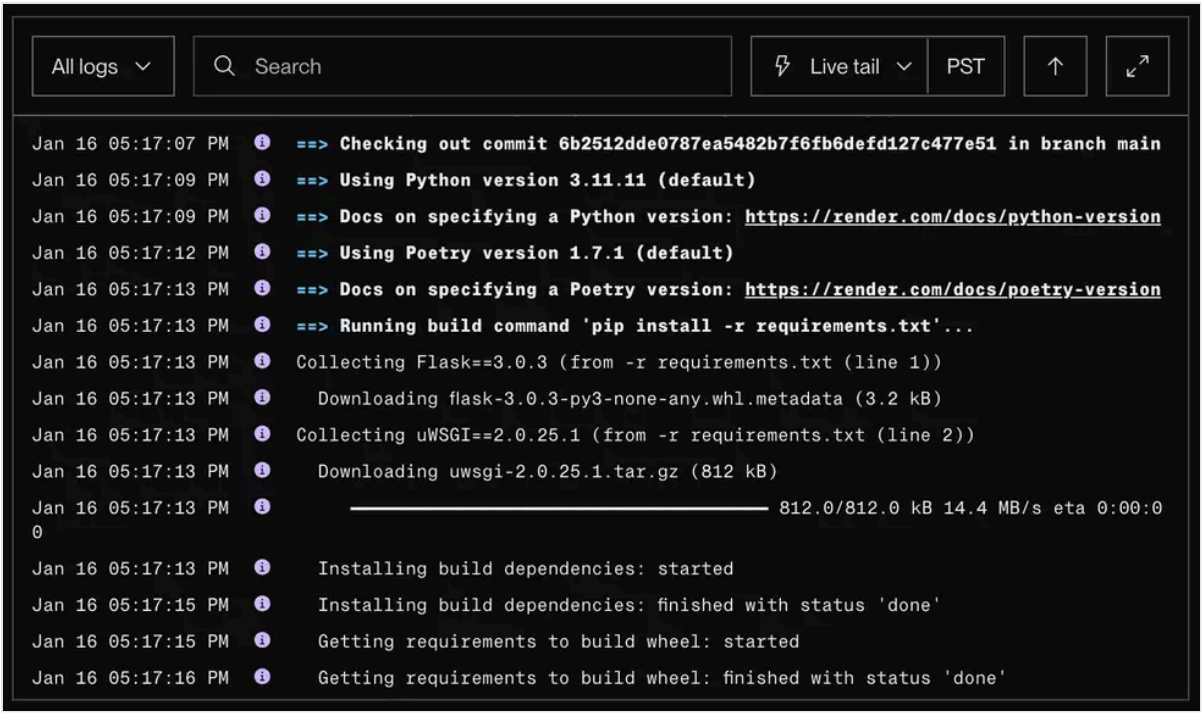
FIELD	DESCRIPTION
Branch	Your service only deploys commits on the branch you specify, such as `main`. Render can automatically redeploy your app whenever you push changes to this branch.
Root Directory	Deploying from a monorepo? Specify the subdirectory that represents your application root. Your build and start commands will run from this directory.
Language	If your app's programming language isn't listed in this dropdown, you can still deploy using the `Docker` runtime if you build your app from a `Dockerfile`.
Build Command	This is the command that Render will use to build your app from source. Common examples include: Node.js | Python | Ruby BASH `npm install` You can also use `yarn` or `bun`. This usually resembles the command you run locally to install dependencies and perform any necessary compilation.
Start Command	This is the command that Render will use to start your app. Common examples include:

FIELD	DESCRIPTION
	Node.js Python Ruby BASH `npm start` You can also use `yarn` or `bun`. For some frameworks, this might differ from the command you run locally to start your app. For example, a Flask app might use `flask run` locally but `gunicorn` for production.
Instance Type	This determines your service's RAM and CPU, along with its cost. Choose the **Free** instance type to deploy for free: 
Environment Variables	These will be available to your service at both build time and runtime. If you forget any, you can always add them later and redeploy.

When you're done, click the **Deploy** button at the bottom of the form. Render kicks off your first deploy.

5. Monitor your deploy

Render automatically opens a log explorer that shows your deploy's progress:



Follow along as the deploy proceeds through your build and start commands.

- **If the deploy completes successfully,** the deploy's status updates to **Live** and you'll see log lines like these:

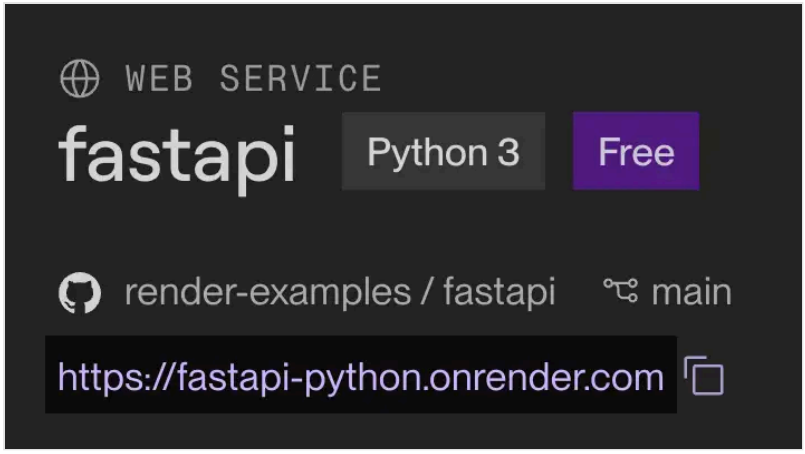
```bash
# Web service
==> Deploying...
==> Running 'npm start' # (or your start command)
==> Your service is live 🎉

# Static site
==> Uploading build...
==> Your site is live 🎉
```

- **If the deploy fails,** the deploy's status updates to **Failed**. Review the log feed to help identify the issue.
  - Also see Troubleshooting Your Deploy for common solutions.
  - After you identify the issue, push a new commit to your linked branch. Render will automatically start a new deploy.

6. Open your app

After your app deploys successfully, you're ready to view it live.

Every Render web service and static site receives a unique `onrender.com` URL. Find this URL on your service's page in the Render Dashboard:



Click the URL to open it in your browser. Your service will serve the content for its root path.

Congratulations! You've deployed your first app on Render 🎉

When you're ready, check out recommended next steps.

Next steps

Connect a datastore

Render provides fully managed Postgres and Key Value instances for your data needs. Both provide a Free instance type to help you get started.

> **Free Render Postgres databases expire 30 days after creation.**
>
> You can upgrade to a paid instance at any time to keep your data. Learn more about free instance limitations.

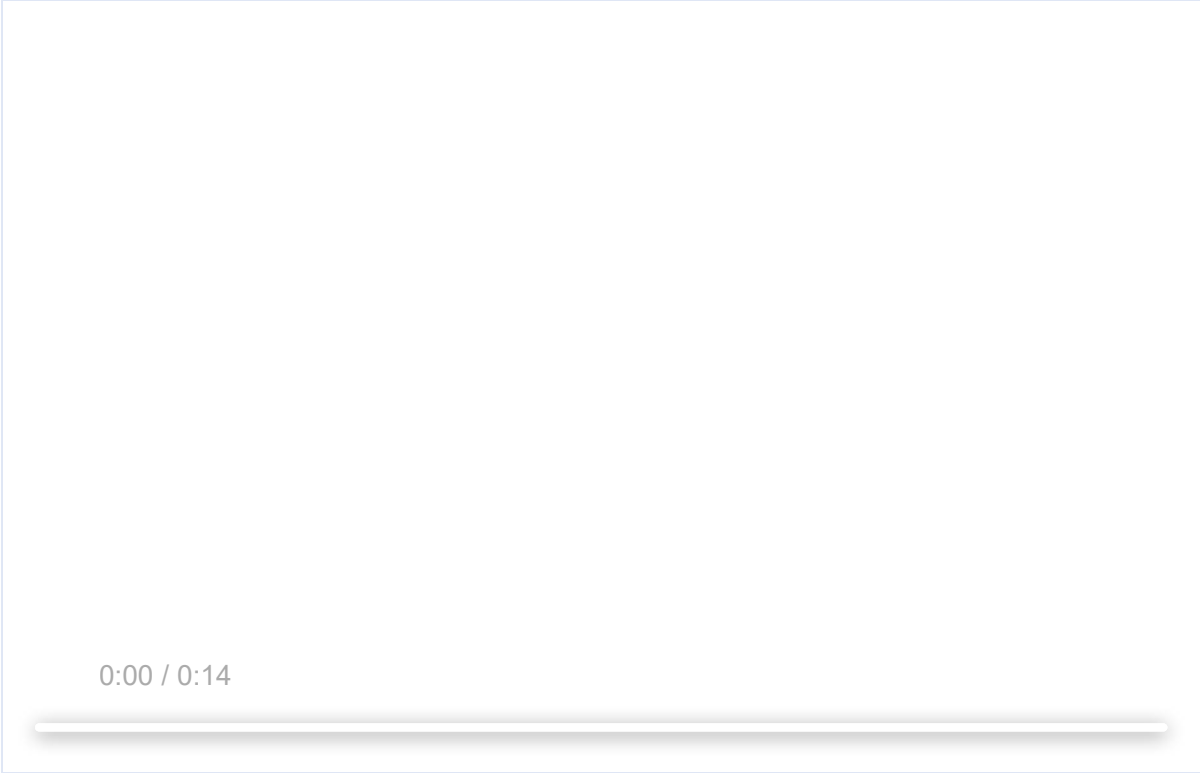
Learn how to create datastores and connect them to your app:

- Render Postgres databases
- Render Key Value instances

Paid services can also attach a disk for persistence of local filesystem data (by default, local filesystem changes are lost with each deploy).

Install the Render CLI

The Render CLI helps you manage your Render services right from your terminal. Trigger deploys, view logs, initiate psql sessions, and more.



Get started with the Render CLI.

Add a custom domain

Each Render web service and static site receives its own `onrender.com` URL. You can also add your own custom domains to these service types. Learn how.

Learn about operational controls

Deploying your app is just the beginning. Check out a few of the ways you can manage and monitor your running services on Render:

- Scaling your instance count
- Analyzing service metrics
- Rolling back a deploy
- Enabling maintenance mode

Note that some of these capabilities require running your service on a paid instance type.

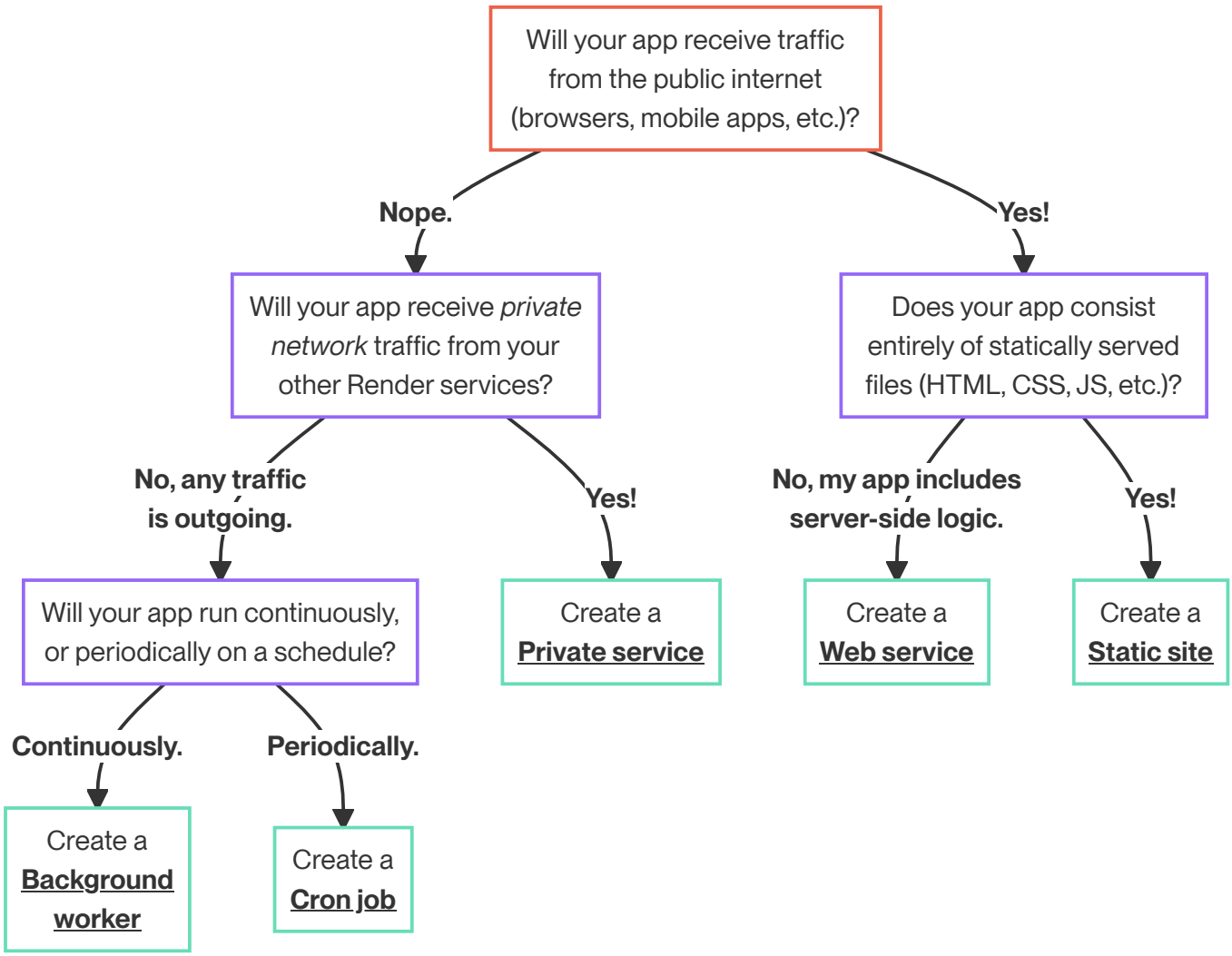
Explore other service types

In addition to supporting web services and static sites, Render offers a variety of other service types to support any use case:

SERVICE TYPE	DESCRIPTION
Private services	Run servers that aren't reachable from the public internet.
Background Workers	Offload long-running and computationally expensive tasks from your web servers.
Cron Jobs	Run periodic tasks on a schedule you define.

Note that free instances are not available for these service types.

Use this flowchart to help determine which service type is right for your use case:



X LinkedIn GitHub