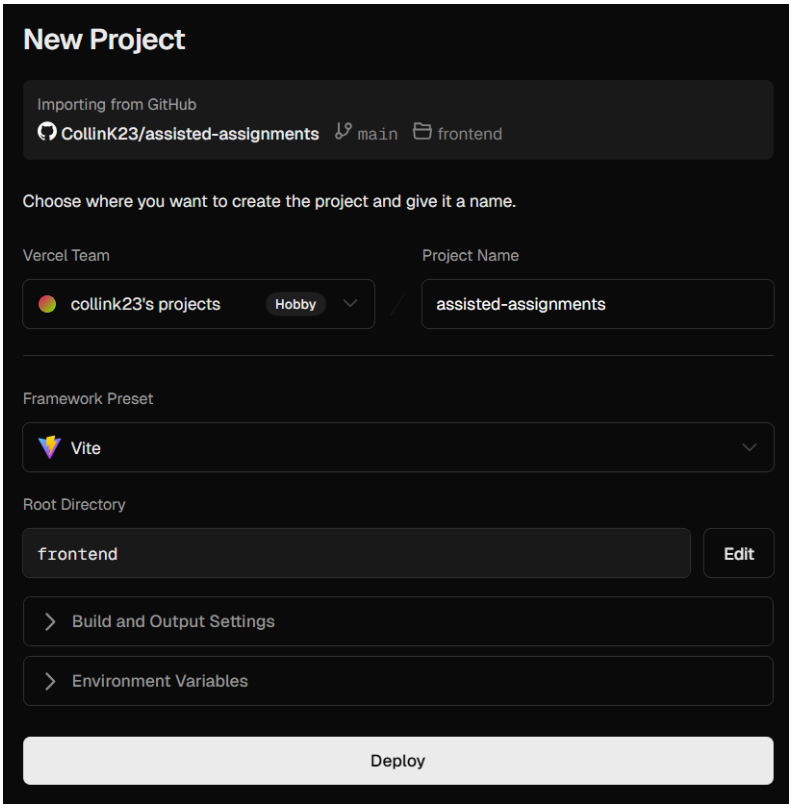
Deployment.md

Deployment

This setup works for the current project, but more scalable deployment methods may be necessary if user demand grows.

1. Frontend Deployment

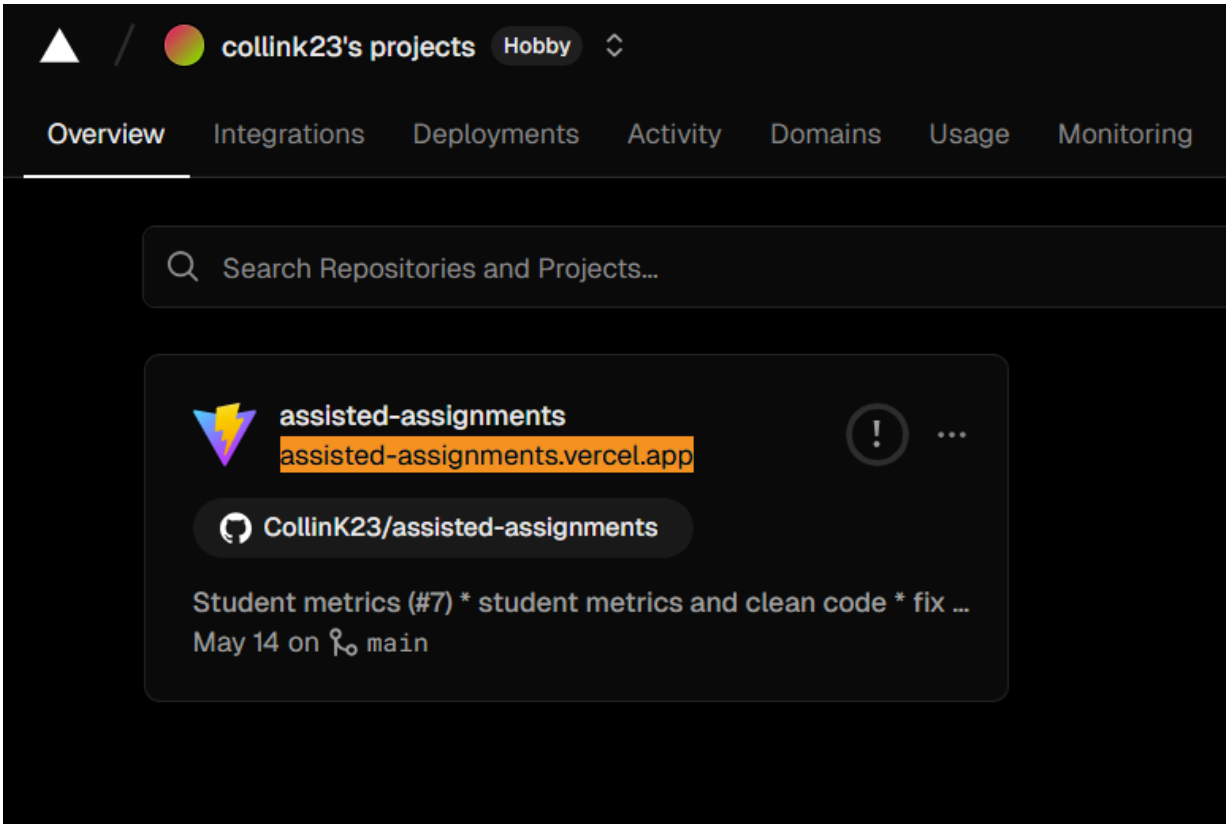
The current Frontend is deployed on Vercel. To deploy the Frontend on Vercel, create an account and link a GitHub Account. Start a new project, and select this GitHub Repository. Under Framework Preset select Vite, in Root Directory select frontend. In the Environment Variables Section, copy the variables from the frontend .env file.



Notes:

- You may want to set up a different Auth0 environment for production and development.
- `VITE_API_BASE_URL will` be updated after setting up Backend Deployment.

- Vercel will automatically redeploy the frontend when changes are pushed to the main branch.
- You can now access the website using the URL shown in your Vercel Dashboard (highlighted in the screenshot below).



2. Redis & Database Deployment

Currently, the project uses a Neon PostgreSQL database in production.

Database Deployment: In the Vercel project Dashboard, navigate to Storage > Create Database > Neon > Continue > Choose a Region > Select Installation plan > Continue > Create

Redis Cache Deployment: In the Vercel project Dashboard, navigate to Storage > Create Database > Redis > Continue > Choose a Region > Select a plan > Continue > Create

Notes:

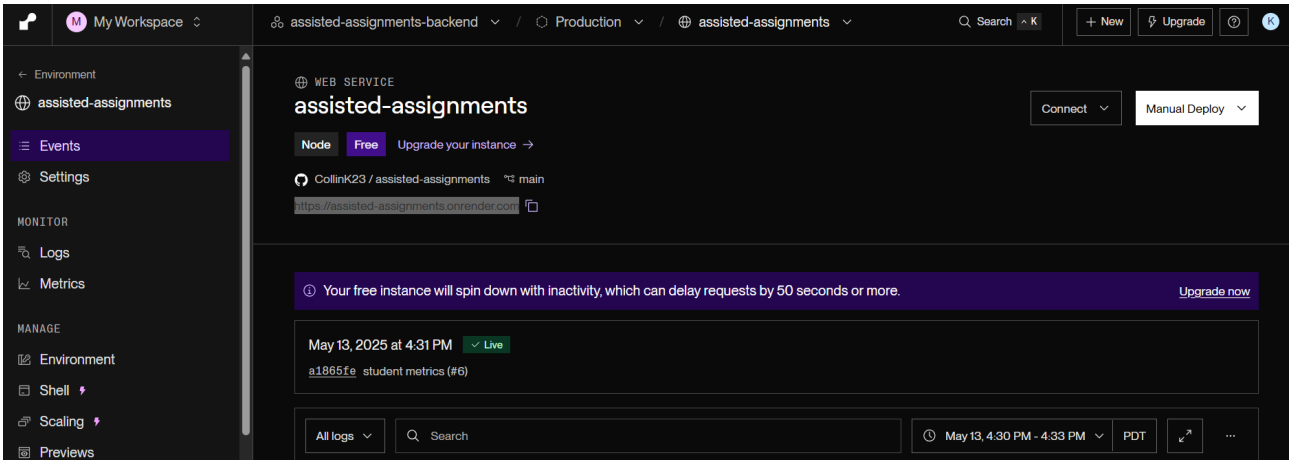
- The values for `REDIS_URL` and `DATABASE_URL` will be needed for the backend deployment.

3. Backend Deployment

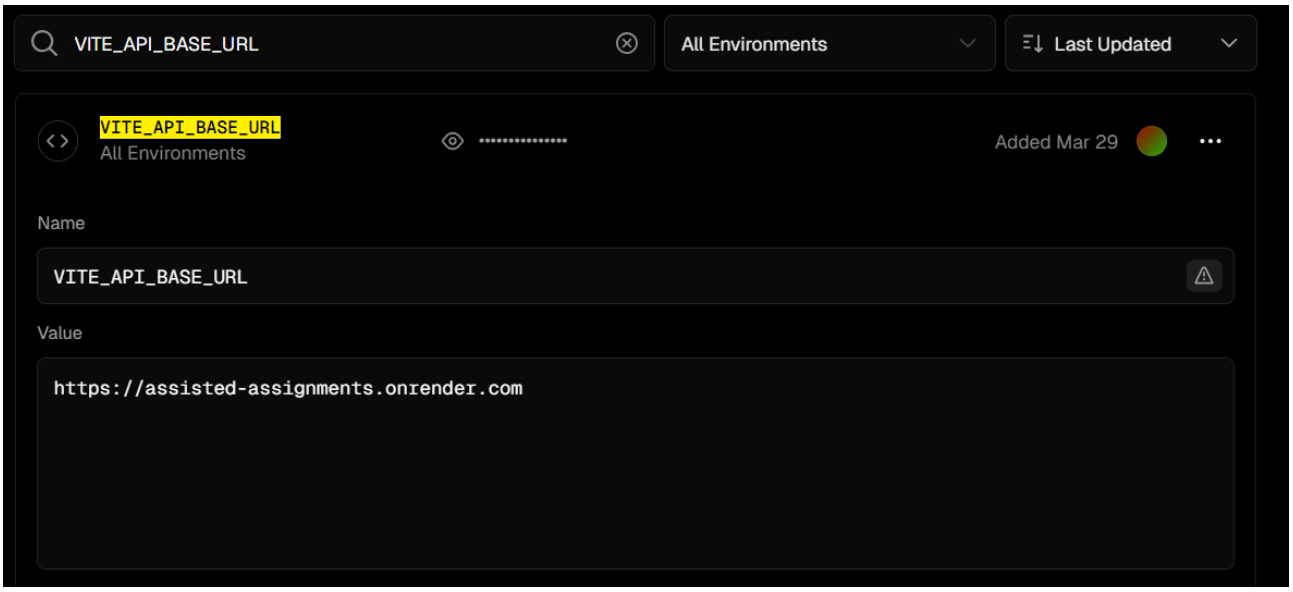
The current Backend is deployed on Render. To deploy the Backend on Vercel, add a new Web Service, connect GitHib as a Git provider, and select this GitHub Repository. In the new web service: create a name, for language select `Node`, for branch enter `main`, select a region, for the root diretory enter `backend`, for the build command enter `npm install`, and for the run command enter `npm start`. Select an instace type. In environment variables, copy the variables from the backend .env file and update the following values:

```
NODE_ENV=production
REDIS_URL= *The value for the one created in Vercel*
DATABASE_URL = "The value for the one created in Vercel*
```

Deploy the service. Once deployment is complete, you'll receive an onrender.com link to access it.



To connect the Frontend with the backend, in the Vercel dashboard: Navigate to Settings > Environment Variables > Find VITE_API_BASE_URL > Edit > Update the value to the onrender.com link that was created in Render.



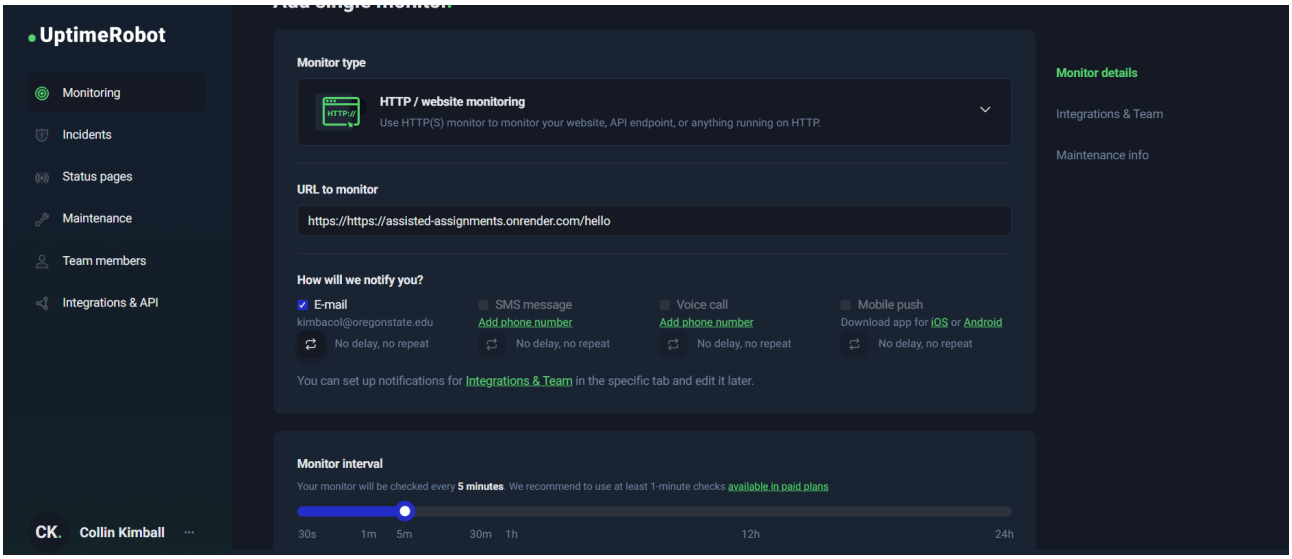
Notes:

- In Render's free plan, the backend spins down after 15 minutes of inactivity, which can cause a delay when it's accessed again as the server needs to restart.

4. Setup Backend Uptime Monitoring (Optional)

Using a service such as UptimeRobot, we can get around the issue with Render's backend spin-down by sending periodic requests to keep the server awake.

To do this add a HTTP/Website monitor to your backend onrender.com link and add `/hello` at the end (e.g., https://assisted-assignments.onrender.com/hello). For monitor Interval select any time <= 10 minutes to keep the server active.


UptimeRobot
Monitoring
Incidents
Status pages
Maintenance
Team members
Integrations & API
CK. Collin Kimball
Monitor type
HTTP / website monitoring
Use HTTP(S) monitor to monitor your website, API endpoint, or anything running on HTTP.
URL to monitor
https://https://assisted-assignments.onrender.com/hello
How will we notify you?
E-mail
kimbacol@oregonstate.edu
No delay, no repeat
SMS message
Add phone number
No delay, no repeat
Voice call
Add phone number
No delay, no repeat
Mobile push
Download app for iOS or Android
No delay, no repeat
You can set up notifications for Integrations & Team in the specific tab and edit it later.
Monitor interval
Your monitor will be checked every 5 minutes. We recommend to use at least 1-minute checks available in paid plans
30s 1m 5m 30m 1h 12h 24h
Monitor details
Integrations & Team
Maintenance info