opsloom

Menu On this page

This guide will walk you through the steps to get your application up and running.

Local Environment Setup

Using env.example as a reference, create a .env file with all the same keys and environment variables. Set the ENV variable to "dev."

Python Environment Setup

Before you go further, ensure you are using Python 3.12 for optimal performance. If you go into the next steps with a different version, you may have to restart this guide to change the version.

We use uv for package management. You can install it on Mac/Linux with the following command:

```
curl -LsSf https://astral.sh/uv/install.sh | sh
```

Like other Python package managers, uv creates a virtual environment in your project. To install the dependencies from pyproject.toml in the virtual environment, please run:

```
uv sync
```

Initializing the Database

Opsloom uses Postgres to store user and conversation data. It also uses the PGVector extension for Postgres to store embeddings for RAG AI assistants.

First, start a Postgres container based on the pgvector image. You can do this by running the script in db/start_postgres.sh, or you can connect to an external AWS RDS instance. The default username and password are configurable in the db/start_postgres.sh file.

Next, run the initial database scripts using alembic. The following command will run all of the scripts under the db/versions/tables folder:

```
uv run alembic upgrade head
```

Initializing the API

Opsloom's API is written in Python, using the FastAPI framework. You can initialize the API server with the following commands:

```
uv sync
uv run uvicorn server:app --reload --port 8080
```

To invoke the API without using the frontend, you will need to download Bruno and supply it with the files in the /docs/api_docs folder.

Running the Frontend Server

The frontend runs on Vite and React. You can initialize it with the following commands:

```
cd frontend
npm install
npm run start
```

Run Documentation Server (Optional)

You can run the documentation server by calling these commands from the project's root directory:

```
npm i
npm run docs:dev
```

Congratulations! Your program should now be running end-to-end. If you sign up with a new account, you'll be greeted with our out-of-the-box assistant. Take a look at the Creating Your First Assistant section for steps on creating your first custom assistant.

Previous page
Architecture

Next page
Creating Your First Assistant