Deployment

1. Docker Compose Services

Our `docker-compose.yml` runs four key services

Service	Image/Build	Role
`frontend`	`./frontend/...`	Nginx server; handles port 80 and proxies `/api` to the backend.
`backend`	`./backend/...`	The Spring Boot REST API.
`postgres`	`postgres:15`	Relational database.
`postgres-backup`	`./db_backup`	Custom service that runs automated DB dumps via crontab.

2. Maintenance Commands

To help a new dev manage the site, here are the main commands:

- **Start everything:** `docker compose up -d`
- **Stop everything:** `docker compose down` (you can add -v flag to delete all volumes)
- **View Backend Logs:** `docker compose logs -f backend`
- **Check Backups:** Logs are stored in `./db_backup/logs` and actual dumps are in `./db_backup/backups`.
- **Build**: When in frontend/ or backend/ route directories use: `docker build . -t roboteamtwente/website-{name, e.g. "frontend"}:{version}`
- **Pushing**: First login to RoboTeam docker account on your device. Then -> `docker push roboteamtwente/website-{name, e.g. "frontend"}:{version}`

3. Deployment

When you are done developing, please consider building the images of services and push them. After that you are able to ssh into our VPS that hosts a lot of products (main landing page included).

```
ssh user@h2960363.stratoserver.net
```

You have to add your user to that VPS's whitelist beforehand in order to login.
In there:

1. `sudo -i` to switch to superuser.
2. `cd docker` to the main directory where all containers are started from.
3. `vim docker-compose.yml` and edit the used version of desired service(s) to the newest one.
4. `docker pull roboteamtwente/name:version` pull desired updated images
5. `docker compose up -d nameOfTheContainer` start the container(s)

Revision #3
Created 2026-04-14 14:15:06 UTC by Illia Guzerya
Updated 2026-04-14 15:28:47 UTC by Dmytro Khorsun