

KNOWLEDGE BASE · PUBLIC DOCUMENT

Updating a Docker application without losing data

Node.js, Python & Docker

CODE

RLX-MZ-KB-887

VERSION

v2026.10.10

COUNTRY · LANGUAGE

Mozambique · English

MozOut Mozambique

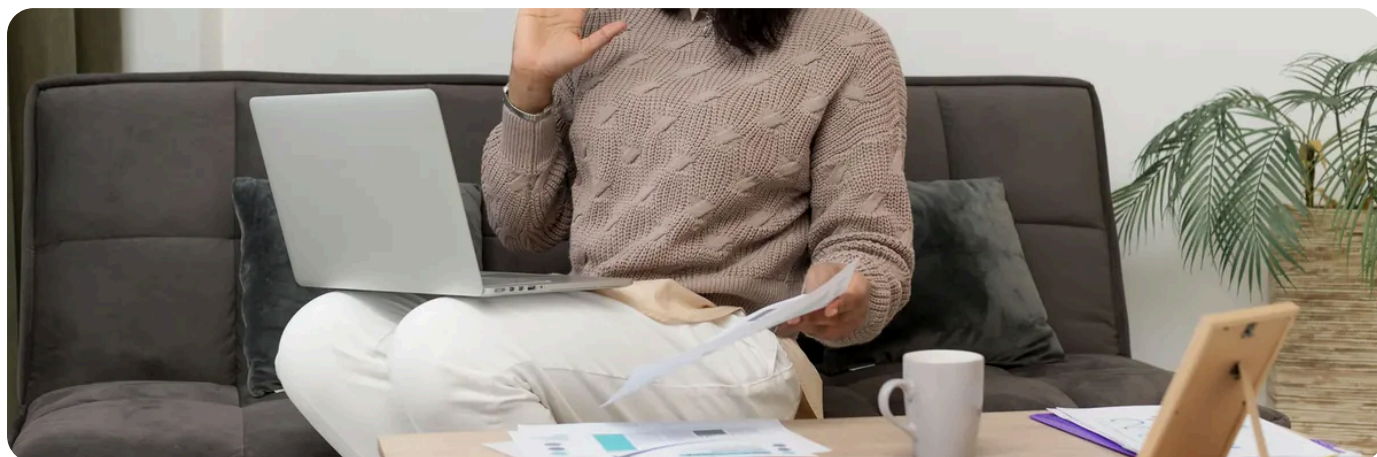
RLX Holdings, S.A. · Group RLX Holdings

Tax ID (NUIT) 401 191 607 · Legal entity no. (NUEL) 105 045 974 · Economic activity code (CAE) 62022

<https://www.meu.mozout.co.mz/knowledgebase/887>

Updating a Docker application without losing data

📁 Node.js, Python & Docker 📅 Created on **October 10, 2026** ⌚ 3 min read



Updating an application in Docker is, as a rule, pulling the new image and recreating the container. The data survives **if it lives in a volume**, and that is where everything is won or lost: the container is disposable, the volume is not. With a backup made first and a pinned version, updating is a calm procedure that you can undo. This article assumes a VPS with Docker and a Compose project ([example](#)).

🔄 The procedure

1 Read the notes for the new version. The project publishes what changed and what needs extra steps (migrations, renamed variables). Jumping several versions at once is where problems tend to appear.

2 Dump the database:

```
docker compose exec -T db pg_dump -U appuser appdb > backup.sql
```

The `-T` is needed so the `>` does not receive junk from a terminal. For MariaDB or MySQL the equivalent is `mysqldump`. See [backups and restores](#).

3 Keep the configuration file and the `.env` too somewhere off the server. Copying a volume and the whole VPS are covered in [backing up a VPS](#).

4 Pin the version you are moving to in the file (`image: name:1.2.3` rather than `:latest`). That way you know exactly what you installed and can go back.

5 Pull and recreate:

```
docker compose pull
docker compose up -d
```

Compose only recreates the services whose image changed. The volumes stay.

6 **Read the logs** and check the application started and the migrations ran: `docker compose logs -f -tail=100`. Open the application and try a real operation.

7 **Clean up old images** only when you are happy: `docker image prune`. Until you do, going back is just changing the number and repeating step 5.

? What deletes data and what does not

Command	What it does	The data?
<code>docker compose pull + up -d</code>	Pulls images and recreates the containers that changed.	Stays, if it is in volumes.
<code>docker compose down</code>	Stops and removes the containers and the network.	Stays. Volumes are not touched.
<code>docker compose down -v</code>	Does the same and removes the volumes too.	Lost.
<code>docker volume prune</code> and <code>docker system prune --volumes</code>	Remove volumes that no container is using (depending on version and options, only anonymous ones or named ones too).	Lost , if the containers are stopped or removed.
Files written outside a volume	They live in the container layer.	Lost when the container is recreated.

! **An “unused” volume is a volume with no running container.**
After a `docker compose down`, your database volume has no container attached, and a `prune` run at that moment can take it. Before any `prune` that includes volumes, look at the list with `docker volume ls` and have a backup. Otherwise, do not use `--volumes`.

! **A database’s major version is not changed by changing the number.**
Going, say, from one major version of PostgreSQL to the next needs a dump (`pg_dump`) and a restore into a new volume. For minor updates within the same major version, the procedure above is enough.

💡 **Rehearse with a copy.**
If the application matters, restore the backup into a new folder with another project name, update there, and only then touch what is live. And note: updating Docker and the application is yours on an unmanaged VPS: **how far our support goes.**

🔊 NEED A HAND?

Something went wrong during the update and the server stopped answering? If it is the VPS, the network or access, we deal with it; tell us what you saw.

[📄 Open a support ticket →](#)**🔗 SEE ALSO**[Backing up with mysqldump and pg_dump, and restoring the copy](#)[Backing up a VPS: snapshots or files](#)[Docker filling the disk: images, volumes and logs](#)**mozout****MozOut Mozambique****RLX Holdings, S.A. · Group RLX Holdings****TAX ID (NUIT)** 401 191 607**LEGAL ENTITY NO. (NUEL)** 105 045 974**ECONOMIC ACTIVITY
CODE (CAE)** 62022**REGISTERED OFFICE** Bairro 7 de Setembro
Vilankulo 1304 · CEP 0411-02, Inhambane, Mozambique
CONTACT info@mozout.co.mz · +258 84 635 1111 · +258 86 635 1111 ·
www.mozout.co.mz**OPENING HOURS** Monday to Friday - 08 AM to 04 PM
Saturday & Sunday - ClosedOnline version of this
document

RLX-MZ-KB-887 · v2026.10.10 · Generated on 21/10/2026 17:40