Mozout

Generated on 15/11/2026 00:39

KNOWLEDGE BASE · PUBLIC DOCUMENT

Docker volumes or bind mounts: where your data lives and how to back it up

Docker & Containers

CODE	VERSION	COUNTRY · LANGUAGE
RLX-INT-KB-3450	v2026.10.10	International · English

MozOut International
RLX Holdings (Pty) Ltd. · Group RLX Holdings
Tax no. (SARS) 943 907 6234 · Registration no. (CIPC) 2025/667953/07

https://www.my.mozout.com/knowledgebase/3450

Docker volumes or bind mounts: where your data lives and how to back it up

Docker & Containers Created on **October 10, 2026** 2 min read



What a container writes inside itself disappears when it is deleted. For data to survive, you connect it to somewhere outside the container, in two ways: a **volume** (managed by Docker) or a **bind mount** (a server folder you choose). **The short answer:** for a database, use a volume; for code or configuration files you want to edit by hand on the server, a bind mount.

Which to choose

	Volume	Bind mount
What it is	An area Docker manages (under `/var/lib/docker`).	A folder on the server, at the path you choose.
Written in Compose as	`db_data:/var/lib/postgresql/data`	`./config:/app/config`
Good for	Databases and application data.	Configuration and files you edit on the server.
The catch	You do not see it in a folder by hand; `down -v` deletes it.	File owners and permissions have to suit the container's user.

The full set-up of an application with its database is in **Docker Compose: an application with its database**.

Seeing where the data is, and making a copy

1 **List the volumes** and see where one is:

```
docker volume ls
docker volume inspect volume-name
```

Compose prefixes the project name to the volume name (`project_db_data`). That is the full name.

2 **For a database, make a logical copy** rather than copying the files while it runs: `pg_dump` or `mysqldump` inside the container. The files of a database that is writing can end up inconsistent. See **backing up with mysqldump and pg_dump**.

3 **For other data, copy the volume to a file** with a temporary container:

```
docker run --rm -v project_data:/data -v "$PWD":/backup alpine tar czf /backup/data.tar.gz -C /data .
```

This creates `data.tar.gz` in the folder you are in. To restore, the same with `tar xzf` onto an empty volume.

4 **Keep the copy off the server.** A copy on the same disk does not protect you from the disk failing: **backing up a VPS**.

! **Watch the owners on a bind mount.**
If the container runs as a user that does not own the server folder, writing fails with “permission denied” even though everything is “mounted”. Set the folder’s owner to the container’s user, instead of putting everything at 777. And `docker compose down -v` deletes the volumes, meaning the data: without `-v` it only removes the containers.

Give volumes clear names
in Compose and write down what each holds. Six months from now nobody remembers which volume is the database, and deleting the wrong one is the expensive mistake: **clearing the disk without losing data**.

RECOMMENDED
Need a server with room and backups for your containers? Have a look at our VPS plans.
See the VPS servers →

SEE ALSO

- Docker Compose: an application with its database
- Updating a Docker application without losing data
- Docker filling the disk: images, volumes and logs

MozOut International
RLX Holdings (Pty) Ltd. · Group RLX Holdings

TAX NO. (SARS)	943 907 6234
REGISTRATION NO. (CIPC)	2025/667953/07
REGISTERED OFFICE	1111 Olive Street Saint Louis 63101, Missouri, United States
CONTACT	info@mozout.com · +27 78 823 3325 · www.mozout.com
OPENING HOURS	Monday to Friday - 08 AM to 04 PM Saturday & Sunday - Closed

Online version of this document

RLX-INT-KB-3450 · v2026.10.10 · Generated on 15/11/2026 00:39