Mozout

Generated on 21/10/2026 15:59

KNOWLEDGE BASE · PUBLIC DOCUMENT

Getting inside a running container: exec, cp and what not to edit there

Node.js, Python & Docker

CODE	VERSION	COUNTRY · LANGUAGE
RLX-INT-KB-3438	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/3438

Getting inside a running container: exec, cp and what not to edit there

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



The container is running and you need to look inside: a file, a variable, run one of the application's commands. **The short answer:** `docker exec` opens a command line (or runs a command) *inside* a container that is already up, and `docker cp` copies files between the server and the container. They are for **looking and diagnosing**. They are not for maintaining the application: whatever you change in there is lost when the container is recreated.

The commands you use most

To…	Command
Open a command line in the container	`docker exec -it name sh` (Alpine images only have `sh`; others may have `bash`). Leave with `exit`.
Run a single command and leave	`docker exec name ls -la /app`
Enter as root, if the image runs as another user	`docker exec -u root -it name sh`
Do the same in a Compose service	`docker compose exec service sh` (in the project folder)
Copy a file from the container to the server	`docker cp name:/app/app.log ./app.log`
Copy from the server into the container	`docker cp ./config.json name:/app/config.json`
Watch usage live	`docker stats` (memory and processor per container)
See a container's processes	`docker top name`

Diagnosing, step by step

1. **Confirm it is up:** `docker ps`. `exec` only works on running containers; for one that fell over, read the logs: **what to look at when it will not start**.

2. **Go in and look:**

```
docker exec -it name sh
ls -la
cat /app/config.json
exit
```

Check that the files are where the application expects them, with the right owners.

3. **Test the network from inside** (if the image has the tools): `getent hosts db` shows whether another service's name resolves. Small images often have neither `curl` nor `ping`.

4. **Bring out the files you need** with `docker cp` to read them at leisure on the server, instead of reading them on screen.

What you change inside the container is lost.
If you edit a file with `exec` and then recreate the container (`docker compose up -d` with a new image, for example), the change disappears. The real fix is made in the code or the configuration, and the image rebuilt, or the file is mounted from outside: **volumes and bind mounts**. To update without losing data: **updating a Docker application**.

Mind the secrets on screen.
`Docker exec name env` prints every variable, including passwords and keys. Do not paste that output into a support ticket or a forum without removing whatever is secret.

Whoever can use `docker` controls the server.
Belonging to the `docker` group is, in practice, being root. Give that access only to those who should have it: **Docker precautions**.

NEED A HAND?

The container runs, but the server stopped answering or SSH will not let you in? That is ours: tell us what you see.

Open a support ticket →

SEE ALSO

- Your first container, and what to look at when it will not start
- Docker volumes or bind mounts: where your data lives and how to back it up
- Updating a Docker application without losing data

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-3438 · v2026.10.10 · Generated on 21/10/2026 15:59