interweb

Generated on 15/11/2026 00:13

KNOWLEDGE BASE · PUBLIC DOCUMENT

Docker containers that cannot reach each other or the host: networks explained

Docker & Containers

CODE	VERSION	COUNTRY · LANGUAGE
RLX-AO-KB-1102	v2026.10.10	Angola · English

Interweb Angola
RLX Angola, Lda. · Group RLX Holdings
Tax ID (NIF) 500 260 5724 · Commercial registration no. 26464-25/250724

https://www.meu.interweb.ao/knowledgebase/1102

Docker containers that cannot reach each other or the host: networks explained

Docker & Containers Created on **October 10, 2026** 3 min read



An application in Docker “cannot see” the database, or two projects cannot talk to each other, or a container cannot reach a service running on the server itself. **The short answer:** each container is a separate “computer” with its own network. In it, `localhost` means *the container itself*, not the server and not its neighbour. To talk to another container you use the **service name**, on a network they both share.

The three cases, and the right address

You want to connect…	Use this address	Condition
Two services in the same Compose file	The service name (`db` , `redis` …) and the internal port.	Nothing to do: Compose creates one network per project and puts the services on it.
Containers from different Compose projects	The service name, on the shared network.	An external network both join (see below).
A container to a service running on the server	`host.docker.internal`	On Linux, add it to the service (see below) and make the server’s service listen on an address the container can reach.

Two projects on one network, step by step

1. **Create a network that belongs to no project:**

```
docker network create shared
```

2 **In both `compose.yaml` files**, attach the service to the network, declared as external:

```
services:
  app:
    image: my-app
    networks:
      - shared

networks:
  shared:
    external: true
```

3 **Bring both projects up** and, from one of them, call the other by its service name. To confirm: `docker network inspect shared` lists the connected containers.

Reaching a service on the server itself

For a container to talk to, say, a database installed on the server, on Linux add this to the service in Compose:

```
extra_hosts:
  - "host.docker.internal:host-gateway"
```

and use `host.docker.internal` as the address. The server's service has to listen on an address the container can reach, not only on `127.0.0.1`, which the container never sees. When you do that, do not open the service to the world: restrict it with the firewall. See **ports and firewall on a VPS**.

Service names only work on networks you created or Compose did.
Two containers started with `docker run` on the default network do not resolve each other by name. Put them on a network of their own (`docker network create`, `--network`) or use Compose. When the database connection fails, the first suspect is `localhost` in the configuration: **an application with its database**.

To diagnose
, go into the container and ask for the name to be resolved: `docker exec -it app getent hosts db`. If it returns an address, the network is right and the error is the port or the credentials. If it returns nothing, the two do not share a network: **getting inside a container**.

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

SEE ALSO

- Docker Compose: an application with its database
- Ports and firewall on a VPS: why your app is not reachable
- Nginx as a reverse proxy in front of a container

interweb

Interweb Angola

RLX Angola, Lda. · Group RLX Holdings

TAX ID (NIF)	500 260 5724
COMMERCIAL REGISTRATION NO.	26464-25/250724
REGISTERED OFFICE	Rua Direita do Patriota, Bairro Dangereux Belas, Luanda, Angola
CONTACT	info@interweb.ao · +244 942 011 191 · www.interweb.ao
OPENING HOURS	Monday to Friday - 08 AM to 04 PM Saturday & Sunday - Closed

Online version of this document

RLX-AO-KB-1102 · v2026.10.10 · Generated on 15/11/2026 00:13