

Resolving Network Route Conflicts Between Docker and LiveNX

QUICK GUIDE

Overview

Issue: Docker creates network routes that interfere with the local network configured on LiveNX.

Basic Knowledge Required to Understand This Issue

What is Docker and Click House, and how are they related to each other?

- **Docker:**
 - A tool that packages applications into containers.
 - Containers include everything needed to run an application (code, libraries, etc.).
 - Makes it easy to deploy and run applications consistently across different systems.
- **ClickHouse:**
 - A fast database designed for analyzing large volumes of data quickly.
- **Relationship:**
 - Docker can run ClickHouse inside a container.
 - This helps with easy deployment, consistent operation, and managing ClickHouse across different systems or environments.

Context

- Docker was introduced with ClickHouse in LiveNX version 24.2.
- After installing LiveNX 24.2, a new interface and route related to Docker appear on the machine because ClickHouse runs in a Docker container.
- When you run the `ip route` command, you might see an entry like this:
`172.17.0.0/16 dev docker0 proto kernel scope link src 172.17.0.1 linkdown`

Explanation:

- This route is for the Docker network.
- Traffic destined for the IP range `172.17.0.0/16` will be handled by the `docker0` network interface.
- `proto kernel` means this route was added automatically by the kernel.
- `scope link` means the route is directly connected to the local link.
- `src 172.17.0.1` specifies the source IP address for this route.
- `linkdown` indicates that the `docker0` interface is currently down or inactive.

Summary:

- Traffic for the Docker network `172.17.0.0/16` is routed through the `docker0` interface, which is currently down.
- **Ensure that LiveNX devices do not use IP addresses** in the `172.17.0.0/16` range to avoid issues. Devices in this range may show as down in LiveNX, and flows may not appear.

To Fix The Issue

How do you change the default network IP range assigned to Docker?

1. Edit the `/etc/docker/daemon.json` file.
2. Find a non-conflicting IP range.
3. Add the following configuration:
`"bip": "non.conflicting.ip.address/here"`
4. The updated file might look like this:

```
admin@livenx:~$ cat /etc/docker/daemon.json
{
  "data-root": "/data/docker",
  "bip": "10.5.0.1/16"
}
```

Note For the `bip` parameter, use an IP address/netmask, not a subnet/netmask. It should include the gateway IP address for the bridge network and not end with `.0`.

5. Restart Docker with the command:
`sudo systemctl restart docker`
6. After editing, verify with the `ip route` command to ensure the Docker IP range has changed.