

Docker has many powerful features that you can use to implement a container networking setup that meets your needs, but sometimes you will need to be able to locate and diagnose issues. In this lesson, we will discuss some basic troubleshooting techniques. We will talk about some ways of accessing useful log data, and we will discuss using the `netshoot` image to test from within various parts of your Docker networks.

Relevant Documentation

- <https://success.docker.com/article/troubleshooting-container-networking>

Lesson Reference

1. Run a container and access the container logs.

```
docker run --name log-container busybox echo Here is my container log!
docker logs log-container
```

2. Run a service and access the service logs.

```
docker service create --name log-svc --replicas 3 -p 8080:80 nginx
curl localhost:8080
docker service logs log-svc
```

3. Check the Docker daemon logs.

```
sudo journalctl -u docker
```

4. Create a custom network with a container, and attach a netshoot container to the network for troubleshooting.

```
docker network create custom-net
docker run -d --name custom-net-nginx --network custom-net nginx
```

5. Inject a netshoot container into another container's network sandbox.

```
docker run --rm --network custom-net nicolaka/netshoot curl custom-net-nginx:80
docker run --rm --network container:custom-net-nginx nicolaka/netshoot curl localhost:80
```

6. Inject a netshoot container into the sandbox of a container that uses the `none` network.

```
docker run -d --name none-net-nginx --network none nginx
docker run --rm --network container:none-net-nginx nicolaka/netshoot curl localhost:80
```