# Troubleshooting a network service

Bert Van Vreckem

## Contents

Last modified: 2015-01-22 10:42:26

A checklist for troubleshooting a network service running on RHEL/CentOS 7. Basic knowledge of TCP/IP is assumed (addressing, port numbers, main protocols, etc.)

## 1 TL;DR Checklist

1. Link layer
   - Check the cable/port LEDs
2. Network layer
   - `ip a`
   - `ip r` (+ ping default gw)
   - `cat /etc/resolv.conf` (+ `dig www.google.com @a.b.c.d +short`)
3. Transport layer
   - `sudo systemctl status SERVICE.service`

    - `sudo systemctl start SERVICE.service`
    - `sudo systemctl enable SERVICE.service`
  - `sudo firewall-cmd –list-all`
    - `sudo firewall-cmd –add-service=SERVICE –permanent`
    - `sudo firewall-cmd –add-port=PORT/tcp –permanent`
    - `sudo systemctl restart firewalld`

4. Application layer
   - `sudo journalctl -f -u SERVICE.service`
   - `sudo systemctl restart SERVICE.service` (after each config file change)

# 2 General guidelines

A few best practices when setting up and troubleshooting network services

- **Automate** your tests or at least use a **checklist** (Shell script, Ansible playbook, Serverspec, etc.)
- Keep your checklist **up-to-date** as you learn new things
- Be **thorough**, don't skip steps (e.g checking the cables)
- **Error messages** usually give a clue of where to look. Google them.
- Work in **small steps** and verify every step.
- Don't assume. **Test**.
- Keep a **backup** copy of the original configuration, and the latest working version.
- Always **validate the syntax** of config files before applying them
- Know what **logs** to look at
- Open a separate terminal that shows the **logs in realtime** (`journalctl -f`)

## 2.1 A bottom-up approach

In this troubleshooting guide, we propose a **bottom-up** approach following the layers of the TCP/IP protocol stack:

1. Link layer: cables, network ports, etc.
2. Internet layer: ip address configuration, routing and DNS
3. Transport layer: service, network ports, firewall settings
4. Application layer: configuration, etc.

It is important to follow the same procedure systematically. When you skip steps, or start to troubleshoot too high up in the stack, you may miss the cause of the fault.

# 3 Link layer

In this phase, we check the network hardware, specifically network cables and ports.

- Is the power on?
- Is the network cable plugged in?
  - In VirtualBox, go to the VM settings, Network, select the active interfaces, click "Advanced" and make sure the checkbox "Cable connected" is checked.
- Are the Ethernet port LEDs on, both on the machine and on the switch?
  - If not, test the cable

    - Some switches have a different color for e.g. FastEthernet (100Mbps) and Gigabit Ethernet. Check whether you see the expected color.

- Use the command `ip link`
    - `UP`: ok, the interface is connected
    - `NO-CARRIER`: no signal on this interface

# 4 Network layer

The network layer is responsible for being able to communicate with other hosts on the network. In order to be able to communicate, each host should have three settings configured correctly:

1. The network interface should have an **IP address** assigned
2. A **default gateway** should be set
3. A **DNS server** should be set

Before "reaching out" to other hosts, first check local settings.

## 4.1 IP address

The IP address may have been set automatically (DHCP), or manually. Check this in `/etc/sysconfig/network-scripts/ifcfg-IFACE`, with IFACE the name of the network interface.

Use the command `ip address` (or shortcut `ip a`) to list the IP addresses for each interface.

You should know the expected value, if not the exact IP, at least the network range or network IP and network mask.

- Many home routers have an IP range of 192.168.0.0/24, 192.168.1.0/24
- On a VirtualBox NAT interface, the IP address of the VM should be 10.0.2.15/8
- On a VirtualBox host-only interface, the IP address of the VM depends on how you configured this network. The default host only network has IP range 192.168.56.0/24, with a DHCP server handing out addresses starting at 192.168.56.101.

Possible problems and causes (automatic IP assignment with DHCP):

- No IP
    - The DHCP server is unreachable
    - The DHCP server won't give an IP to this host
- IP looking like 169.254.x.x
    - No DHCP server could be reached, and a "link-local" address was assigned
- IP not in the expected range
    - You may have accidentally set a fixed IP previously, and forgot to set it to DHCP again…

Possible problems and causes (manual IP setting):

- IP not in the expected range
    - Mistake in IP address assignment, check the configuration file
- Correct IP, but "network unreachable"
    - Check the network mask, this should be identical for all hosts on the LAN

### 4.2 Default gateway

Usually, a host is connected to a LAN through a switch. Network traffic to the outside world goes through a router, connected to the same LAN. Every host on the LAN should know this router, the “default gateway”.

Use the command `ip route` (or shortcut `ip r`). There should be a line starting with `default via x.y.z.w`.

Possible problems and causes (automatic IP assignment with DHCP):

- No default gateway set
    - there is most probably also a problem with the IP address of this host, fix this first
    - If you manage the DHCP server, maybe it’s configured badly?
- Unexpected default gateway address
    - You may have accidentally set the default gateway manually previously and forgot to set it to DHCP again
    - If you manage the DHCP server, maybe it’s configured badly?

### 4.3 DNS server

In order to be able to resolve host names to IP addresses, every host should be able to contact a DNS server. View the file `/etc/resolv.conf`. It usually has a header that mentions it was generated automatically, and should have one or more lines starting with `nameserver`.

```
# Generated by NetworkManager
```

Possible problems and causes are equivalent to those with the default gateway setting.

### 4.4 Check LAN connection

If the previous settings are correct, you can check whether other hosts on the LAN can be reached.

- Ping the default gateway: `ping a.b.c.d`
- Ping another known host on the LAN
- Check DNS name resolution: `dig www.google.com @e.f.g.h +short`

**Be aware** that `ping` (and other network troubleshooting tools like the `traceroute` family) may not always work. Some system administrators will block ICMP traffic on routers, rendering the results useless. A command like `ping www.google.com` (for some the first command they try in case of network connection problems) is not very suitable, in that it depends on too many things to work at once:

- the host’s network settings should be correct
- routing should work
- DNS should be available
- no router between this host and the target should block ICMP
- …

## 5 Transport layer

In the transport layer, we’ll check whether the network service is actually running, what port it uses, and whether the firewall allows traffic on that port. An example for `httpd` is given, but this can be applied to other services.

### 5.1 Service and port

- Is the service running? `sudo systemctl status httpd.service`
  - Expected output: `active (running)`
  - If the output contains: `inactive (dead)`, start the service, and if necessary, make sure it starts automatically on boot

    ```
    sudo systemctl start httpd.service
    sudo systemctl enable httpd.service
    ```

- What port is the service using? `sudo ss -tlnp` (list TCP (`-t`) server (`-l`) port numbers (`-n`) with the process behind them (`-p`). The `-p` option requires root, hence the `sudo`)
  - The expected output depends on the service and on how you configured it. The httpd service usually listens on port 80 (HTTP) or 443 (HTTPS), but the port number may have been set to a non-standard. Check `/etc/services` for standard port numbers for well-known network services.

### 5.2 Firewall setting

Does the firewall allow traffic on the service? `sudo firewall-cmd –list-all`.

```
$ sudo firewall-cmd --list-all
[sudo] password for USER:
public (default, active)
  interfaces: enp0s3 enp0s8
  sources:
  services: dhcpv6-client mdns samba-client ssh http https
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:
```

Check the output for the following items:

- The network interface that the service listens on is listed
- The service name is listed.
  - The value should be one listed by `firewall-cmd –get-services`.
  - Remark that the service name for `firewalld` is not necessarily equal to the service name for `systemd`. E.g. BIND is called `named.service` by `systemd`, while it is referred to as `dns` by `firewalld`.
- If the service name is not present, the port numbers used by the service should be listed (e.g. when using a non-standard port, or a service not known by `firewalld`)

## 6 Application layer

On this phase, we check whether the application is configured correctly, and whether it is available to clients and responds correctly to requests.

### 6.1 Log files

Check the log, either using `journalctl`, or by looking at the log file in `/var/log`. The former is standard, the latter may be needed for services that keep a log file not managed/recognized by `systemd`.

Open a separate terminal with the relevant logs opened and watching for changes (`-f` option): - `sudo journalctl -f -u httpd.service` - `sudo tail -f /var/log/httpd/error_log`

### 6.2 Configuration files

Check the configuration file, somewhere in `/etc/`, e.g. `/etc/httpd/httpd.conf`. First, create a backup of the current configuration file(s), and if possible the default one (as created when installing the service)

- Check the configuration file for errors (depends on the service, read the man-page or documentation)
- Validate the syntax of the configuration file. Most services have a command that does this.
  - e.g. for `httpd`: `apachectl configtest`
- After making changes, restart the service
  - `sudo systemctl restart httpd.service`

### 6.3 Availability

You can start checking availability on the loopback interface, but it is important to repeat this from another host. The loopback interface is not firewalled, while physical network interfaces are.

- Do a port scan from another host on the lan, e.g.
  - `sudo nmap -sS -p 80,443 HOST` (perform a TCP SYN scan on port 80 and 443)
- Use a test tool or client software to check availability of the service, e.g.
  - `wget http://HOST/`, `wget https://HOST/`
  - `curl http://HOST/`, `curl https://HOST/`