

What's the difference between Network Initscript and NetworkManager in RHEL 7?

Cuestión

- RHEL 7 now has increased support for NetworkManager, so which should I use?
- Is Network Initscript going away now that NetworkManager is on the rise?
- When is Network Initscript functionality preferred? When is NetworkManager preferred?

Medio Ambiente

Red Hat Enterprise Linux 7

Resolución

There are two networking methodologies available for use in RHEL 7. This KBASE details what they are, when you should use one or the other, and what the future holds for each.

What is Network Initscript and NetworkManager?

NetworkManager is a suite of cooperative network management tools designed to make networking simple and straightforward by eliminating the need to manually edit network configuration files by hand. NetworkManager provides a flexible management interface with GUI, CLI, and TUI options for managing local, remote, or even headless systems. It provides support for broad array of network interface types including Ethernet, InfiniBand IPoIB, VLANs, Bridges, Bonds, Teams, WiFi, WiMAX, WWAN, Bluetooth, VPN, and more.

Network initscript is a basic network interface start/stop framework that is part of the initscripts package.

When should I use Network Initscript?

Currently, NetworkManager does not support the following device types:

- ISDN
- 56k modems
- Open vSwitch
- IPSec
- IPv6-related tunnels (SIT)
- VXLAN encapsulation

However, some of these device types are planned to be supported in future releases of NetworkManager in RHEL 7.

When should I use NetworkManager?

- For managing InfiniBand iPoIB interfaces
The network init script will also work, but system startup times are likely to be much slower.
- Whenever an easy to use network configuration and management interface is required or preferred for configuring network devices
Using the network init script requires administrators to have knowledge of the ifcfg file format and edits need to be done by hand in order to configure or even make simple changes to an interface configuration (introducing the chance to make configuration errors) Unlike the network init script, NetworkManager simplifies network configuration and provides flexible interface options using either the GUI, CLI, or TUI interfaces.

While RHEL 5/6 has device hotplug support (udev rule that runs the ifup script for newly created devices), it has been disabled for RHEL 7 since it can result in race conditions when initializing newly found devices.

Additional Information and Known Issues

- NetworkManager is now the default mechanism for RHEL 7. If NetworkManager is disabled by the administrator, then the fallback would be to use init scripts for network interface management.
- Systemd does not allow stopping of a service once it's already been stopped. This means you're no longer able to stop network services, bring up only a single or even just a few network interfaces and then use the network init script to perform the necessary clean-up/stopping of those active interfaces. e.g., "service network stop; ifup eth0; service network stop" would no longer work.
- When transitioning from NetworkManager to using the network init script, the default gateway parameter in the interface's ifcfg file will be depicted as 'GATEWAY0'. In order for the ifcfg file to be compatible with the network init script, this parameter must be renamed to 'GATEWAY'. This limitation will be addressed in an upcoming release of RHEL7.
- Certain interface bonding configuration options as defined by the BONDING_OPTS parameter in the interface's ifcfg file may not be compatible with NetworkManager.

What does the future hold for each?

Due to the increased focus in NetworkManager, Red Hat may at some point deprecate Network Init script in a future major release. In the meantime, no new features or RFEs will be accepted for Network Init script in the RHEL 7 timeframe. New features and RFEs are planned in NetworkManager. Regardless, defects and bug fixes will be allowed for both technologies.

The network init script methodology was originally designed to be synchronously loaded. This is not a problem for RHEL 5/6 where services are started one after another, so the system loads in a predictable way. Starting in RHEL 7, the boot process is highly parallelized (due to systemd) and it's the duty of services to cope with and dynamically react to any changes in the system. As such, the network init script was not intended to be loaded this way and a number of timing issues have been uncovered where network devices may not start after system initialization. The only solution to this problem is to slow down and serialize service startup to reliably bring-up these types of devices when using the network init script. However, doing so comes at a price, a huge impact on boot times (negating one of the major benefits of systemd) and may sometimes cause the system to appear unresponsive or stalled during boot up while waiting for certain devices to come up.