



Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'netplan-generate.8'

\$ man netplan-generate.8

netplan-generate(8) netplan-generate(8)

NAME

netplan-generate - generate backend configuration from netplan YAML files

SYNOPSIS

netplan [--debug] generate -h | --help

netplan [--debug] generate [--root-dir ROOT_DIR] [--mapping MAPPING]

DESCRIPTION

netplan generate converts netplan YAML into configuration files understood by the backends (systemd-networkd(8) or NetworkManager(8)). It does not apply the generated configuration.

You will not normally need to run this directly as it is run by netplan apply, netplan try, or at boot.

Only if executed during the systemd initializing phase (i.e. "Early bootup, before basic.target is reached"), will it attempt to start/apply the newly created service units.

Requires feature: generate-just-in-time

For details of the configuration file format, see netplan(5).

OPTIONS

-h, --help

Print basic help.

--debug

Print debugging output during the process.

--root-dir ROOT_DIR

Instead of looking in `{lib,etc,run}/netplan`, look in `/ROOT_DIR/{lib,etc,run}/netplan`

--mapping MAPPING

Instead of generating output files, parse the configuration files and print some internal information about the device specified in MAPPING.

HANDLING MULTIPLE FILES

There are 3 locations that netplan generate considers:

? `/lib/netplan/*.yaml`

? `/etc/netplan/*.yaml`

? `/run/netplan/*.yaml`

If there are multiple files with exactly the same name, then only one will be read. A file in `/run/netplan` will shadow - completely replace - a file with the same name in `/etc/netplan`. A file in `/etc/netplan` will itself shadow a file in `/lib/netplan`.

Or in other words, `/run/netplan` is top priority, then `/etc/netplan`, with `/lib/netplan` having the lowest priority.

If there are files with different names, then they are considered in lexicographical order

- regardless of the directory they are in. Later files add to or override earlier files.

For example, `/run/netplan/10-foo.yaml` would be updated by `/lib/netplan/20-abc.yaml`.

If you have two files with the same key/setting, the following rules apply:

? If the values are YAML boolean or scalar values (numbers and strings) the old value is overwritten by the new value.

? If the values are sequences, the sequences are concatenated - the new values are appended to the old list.

? If the values are mappings, netplan will examine the elements of the mappings in turn using these rules.

SEE ALSO

`netplan(5)`, `netplan-apply(8)`, `netplan-try(8)`, `systemd-networkd(8)`, `NetworkManager(8)`

AUTHORS

Daniel Axtens (<daniel.axtens@canonical.com>).

`netplan-generate(8)`