NetworkManager i nmcli: konfiguracja trwała

Poziom: podstawy

Dotyczy: RHEL/Rocky 9–10 i SLES (NetworkManager domyślnie), Ubuntu Server zwykle przez Netplan

Czas: ~20 min

*`nmcli` konfiguruje sieć **trwale** — w przeciwieństwie do `ip` z poprzedniej strony, zapisuje ustawienia do plików, które NetworkManager odtwarza po każdym restarcie. Cena tej trwałości: zmiana zapisana `nmcli c mod` nie działa, dopóki jej nie zastosujesz.*

Skróty użyte na tej stronie (15)

Skrót	Co znaczy
CLI	Command Line Interface: praca przez wpisywanie komend w terminalu
DHCP	Dynamic Host Configuration Protocol: usługa, która sama rozdaje komputerom adresy IP
DNS	Domain Name System: książka telefoniczna internetu, zamienia nazwy na adresy IP
IEEE	Institute of Electrical and Electronics Engineers: międzynarodowa organizacja inżynierów, która m.in. wydaje normy i przyznaje kamienie milowe w historii techniki
IPv4	Internet Protocol wersja 4: adresy zapisane czterema liczbami z kropkami, na przykład 10.20.10.77
KVM	Keyboard, Video, Mouse: konsola albo przełącznik, który daje jedną klawiaturę, ekran i mysz do wielu serwerów (w Linuksie KVM to też technika wirtualizacji)
MAC	Media Access Control: unikalny adres fizyczny karty sieciowej, przypisany fabrycznie
MTU	Maximum Transmission Unit: największy pakiet, jaki łącze przepuści bez dzielenia, zwykle 1500 bajtów w Ethernecie, a przy PPPoE 1492
RHEL	Red Hat Enterprise Linux: komercyjna dystrybucja Linuksa
SLES	SUSE Linux Enterprise Server: płatna dystrybucja Linuksa firmy SUSE, konkurent Rocky/RHEL
SSH	Secure Shell: szyfrowane zdalne logowanie do serwera
VLAN	Virtual LAN: wydzielona logicznie sieć na tym samym sprzęcie
VPN	Virtual Private Network: szyfrowany tunel do sieci firmowej

Skrót	Co znaczy
Wi-Fi	Sieć bezprzewodowa: wygodna w biurze, ale w serwerowni dokłada kolejną możliwą przyczynę awarii
YAML	YAML Ain't Markup Language: format plików konfiguracyjnych oparty na wcięciach

Fiszka

TYLKO ODCZYT

Wyświetlenie statusu sieci jednym poleceniem: nmcli

```
nmcli
```

Bez argumentów wypisuje zwięzłe podsumowanie: stan sieci ogólnie, każde urządzenie z jego aktywnym połączeniem, adresem i trasą. Dobry pierwszy odruch na nieznanej maszynie, zaraz obok `ip -br a`.

Przykład:

```
ens192: connected to biuro-static
        "Intel I219-V"
        ethernet (e1000e), 52:54:00:AA:BB:CC, hw, mtu 1500
        ip4 default
        inet4 10.20.10.42/24
        route4 0.0.0.0/0
        route4 10.20.10.0/24

lo: unmanaged
        "lo"
        loopback (unknown), 00:00:00:00:00:00, sw, mtu 65536

DNS configuration:
        servers: 10.20.10.11
        interface: ens192

Use "nmcli device show" to get complete information about known devices and
"nmcli connection show" to get an overview on active connection profiles.
```

nmcli = NetworkManager command line interface: konfiguracja sieci

TYLKO ODCZYT

Wyświetlenie wyniku z pełnym formatowaniem: nmcli -p

```
nmcli -p d show NazwaInterfejsu
```

`-p` (pretty) dodaje nagłówki sekcji i separatory, które domyślny tryb pomija dla łatwiejszego przetwarzania skryptami. Wygodniejsze do czytania ręcznego, gorsze do `grep`/`awk`.

Przykład:

```
===============================================================================
                        Device details (ens192)
===============================================================================
GENERAL.DEVICE:                         ens192
GENERAL.TYPE:                           ethernet
GENERAL.STATE:                          100 (connected)
GENERAL.CONNECTION:                     biuro-static
-------------------------------------------------------------------------------
IP4.ADDRESS[1]:                         10.20.10.42/24
IP4.GATEWAY:                            10.20.10.1
IP4.DNS[1]:                             10.20.10.11
===============================================================================
```

nmcli = NetworkManager command line interface: konfiguracja sieci

TYLKO ODCZYT

Wypisanie urządzeń sieciowych: nmcli d status

```
nmcli d status
```

> **Pojęcie: połączenie (connection) kontra urządzenie (device).** Urządzenie to fizyczny albo wirtualny interfejs sieciowy widoczny w jądrze — to samo, co pokazuje `ip link`. Połączenie to zapisany w NetworkManagerze **profil ustawień** (adres, brama, DNS, VLAN…), który można podpiąć pod urządzenie. Jedno urządzenie może mieć kilka profili połączeń zdefiniowanych (np. „biuro-dhcp" i „biuro-static"), ale w danej chwili

aktywny jest najwyżej jeden. To rozróżnienie tłumaczy, dlaczego polecenia `nmcli` dzielą się na grupę `device` i grupę `connection` — to dwa różne obiekty, mimo że w mowie potocznej mówi się o obu „interfejs".

Przykład:

```
DEVICE  TYPE      STATE                   CONNECTION
ens192  ethernet  connected               biuro-static
lo      loopback  unmanaged               --
```

`unmanaged` przy `lo` znaczy, że NetworkManager świadomie nie zarządza tym interfejsem — pętla zwrotna nie potrzebuje konfiguracji.

DNS = Domain Name System: książka telefoniczna internetu, zamienia nazwy na adresy IP · **VLAN** = Virtual LAN: wydzielona logicznie sieć na tym samym sprzęcie · **nmcli** = NetworkManager command line interface: konfiguracja sieci

TYLKO ODCZYT

Wypisanie listy profili połączeń: nmcli c show

```
nmcli c show
```

Pojęcie: profil połączenia. Nazwany komplet ustawień sieciowych zapisany trwale przez NetworkManager: metoda adresacji (DHCP czy statyczna), adres, brama, serwery DNS, ustawienia VLAN/bond/team, i informacja, do jakiego urządzenia ma się podpinać. Profil istnieje niezależnie od tego, czy jest w danej chwili aktywny — może leżeć „na zapas", gotowy do włączenia poleceniem `nmcli c up`.

Przykład:

```
NAME           UUID                                  TYPE      DEVICE
biuro-static   3f9a1b2c-4d5e-4f60-8a1b-9c8d7e6f5a4b  ethernet  ens192
biuro-dhcp     7a2b3c4d-5e6f-4071-9b2c-ad9e8f7a6b5c  ethernet  --
```

Profil `biuro-dhcp` istnieje, ale nie jest przypisany do żadnego urządzenia w tej chwili (`DEVICE: --`) — jest zdefiniowany, ale nieaktywny.

DHCP = Dynamic Host Configuration Protocol: usługa, która sama rozdaje komputerom adresy IP · **DNS** = Domain Name System: książka telefoniczna internetu, zamienia nazwy na adresy IP · **VLAN** = Virtual LAN: wydzielona logicznie sieć na tym samym sprzęcie · **nmcli** = NetworkManager

command line interface: konfiguracja sieci

□□ TYLKO ODCZYT

Wyświetlenie pełnych szczegółów profilu: nmcli c show NazwaPolaczenia

```
nmcli c show NazwaPolaczenia
```

Wszystkie ustawienia zapisane w profilu, dziesiątki linii — adresacja IPv4/IPv6, MTU, ustawienia 802.1X, kolejkowanie. Praktyczne użycie: filtrowanie przez `grep`.

```
nmcli c show NazwaPolaczenia | grep ipv4
```

Przykład:

```
ipv4.method:                            manual
ipv4.addresses:                         10.20.10.42/24
ipv4.gateway:                           10.20.10.1
ipv4.dns:                               10.20.10.11
ipv4.dns-search:                        --
```

□□ **IPv4** = Internet Protocol wersja 4: adresy zapisane czterema liczbami z kropkami, na przykład 10.20.10.77 · **MTU** = Maximum Transmission Unit: największy pakiet, jaki łącze przepuści bez dzielenia, zwykle 1500 bajtów w Ethernecie, a przy PPPoE 1492 · **nmcli** = NetworkManager command line interface: konfiguracja sieci

□□ TYLKO ODCZYT

Aktywacja i dezaktywacja profilu: nmcli c up / nmcli c down

```
sudo nmcli c up NazwaPolaczenia
```

Podpina profil pod jego urządzenie i w pełni renegocjuje połączenie — dla Ethernetu oznacza to krótkie zerwanie i ponowne podniesienie łącza. `nmcli c down NazwaPolaczenia` robi odwrotność: odłącza profil od urządzenia, zostawiając interfejs bez adresu.

Przykład:

```
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/4)
```

□□ **nmcli** = NetworkManager command line interface: konfiguracja sieci

□□ TYLKO ODCZYT

Trwała zmiana ustawień IPv4: nmcli c mod

```
sudo nmcli c mod NazwaPolaczenia \
  ipv4.method manual \
  ipv4.addresses AdresIP/24 \
  ipv4.gateway AdresIP \
  ipv4.dns AdresIP
```

> (gdzie kolejne `AdresIP` to: adres tego interfejsu, brama domyślna oraz serwer DNS)

Zapisuje cztery właściwości naraz do pliku profilu na dysku: `ipv4.method manual` przełącza z DHCP na adres statyczny, `ipv4.addresses` i `ipv4.gateway` ustawiają adres i bramę, `ipv4.dns` nadpisuje serwery DNS (kolejne adresy dodajemy przecinkiem albo powtarzając flagę z `+ipv4.dns`). Polecenie **milczy przy powodzeniu i nic nie zmienia w działającej sieci** — więcej w sekcji *Jak to działa*. Powrót do DHCP: `nmcli c mod NazwaPolaczenia ipv4.method auto` (i wyczyszczenie `ipv4.addresses` przez `nmcli c mod NazwaPolaczenia ipv4.addresses ""`).

□□ **IPv4** = Internet Protocol wersja 4: adresy zapisane czterema liczbami z kropkami, na przykład 10.20.10.77 · **DNS** = Domain Name System: książka telefoniczna internetu, zamienia nazwy na adresy IP · **DHCP** = Dynamic Host Configuration Protocol: usługa, która sama rozdaje komputerom adresy IP · **nmcli** = NetworkManager command line interface: konfiguracja sieci

□□ TYLKO ODCZYT

Utworzenie nowego profilu od zera: nmcli c add type ethernet

```
sudo nmcli c add type ethernet con-name NazwaPolaczenia ifname NazwaInterfejsu
```

Tworzy nowy profil o podanej nazwie (`con-name`), przypisany domyślnie do trybu DHCP, powiązany z konkretnym urządzeniem (`ifname`). Dopiero teraz istnieje jako plik na dysku — nie jest jeszcze aktywny, dopóki nie zrobimy `nmcli c up NazwaPolaczenia`.

Przykład:

```
Connection 'biuro-dhcp' (7a2b3c4d-5e6f-4071-9b2c-ad9e8f7a6b5c) successfully added.
```

DHCP = Dynamic Host Configuration Protocol: usługa, która sama rozdaje komputerom adresy IP · **nmcli** = NetworkManager command line interface: konfiguracja sieci

TYLKO ODCZYT

Usunięcie profilu: nmcli c del

```
sudo nmcli c del NazwaPolaczenia
```

Kasuje profil trwale — jeśli był aktywny, interfejs traci konfigurację natychmiast. Ostrzeżenie w *Częste błędy*: usunięcie profilu, przez który jesteśmy podłączeni zdalnie, ucina sesję bez pytania.

nmcli = NetworkManager command line interface: konfiguracja sieci

TYLKO ODCZYT

Zastosowanie zmian bez zrywania łącza: nmcli d reapply

```
sudo nmcli d reapply NazwaInterfejsu
```

Alternatywa dla `nmcli c up`: próbuje wcielić bieżące ustawienia profilu **bez** pełnego zestawienia połączenia od nowa — łącze nie mruga. Działa dla większości zmian (adres, DNS, trasy), ale nie dla wszystkich (np. zmiana `ipv4.method` z `auto` na `manual` bywa ignorowana przez `reapply` i wymaga pełnego `c up`).

Przykład:

```
Connection successfully reapplied to device 'ens192'.
```

DNS = Domain Name System: książka telefoniczna internetu, zamienia nazwy na adresy IP · **nmcli** = NetworkManager command line interface: konfiguracja sieci

TYLKO ODCZYT

Wyświetlenie informacji ogólnych: nmcli g

```
nmcli g
```

Najkrótszy możliwy status: czy sieć w ogóle jest włączona, stan łączności, stan trybu samolotowego dla urządzeń bezprzewodowych.

Przykład:

```
STATE      CONNECTIVITY  WIFI-HW  WIFI     WWAN-HW  WWAN
connected  full          enabled  enabled  enabled  enabled
```

nmcli = NetworkManager command line interface: konfiguracja sieci

TYLKO ODCZYT

Globalne wyłączenie sieci: nmcli n off

```
sudo nmcli n off
```

Odłącza wszystkie zarządzane interfejsy naraz — odpowiednik trybu samolotowego. `nmcli n on` przywraca. Rzadko używane na serwerach (odcina zdalny dostęp), przydatne przy testach w środowisku lokalnym z konsolą fizyczną.

nmcli = NetworkManager command line interface: konfiguracja sieci

TYLKO ODCZYT

Uruchomienie interfejsu tekstowego: nmtui

```
sudo nmtui
```

Pełnoekranowy interfejs w terminalu (na bazie biblioteki `newt`) do tych samych operacji — edycji profili, aktywacji połączeń, zmiany nazwy hosta — bez zapamiętywania składni `nmcli`. Wygodny przy pierwszej konfiguracji świeżo zainstalowanej maszyny przez konsolę, niepraktyczny do automatyzacji.

TYLKO ODCZYT

Przegląd plików profili: /etc/NetworkManager/system-connections/

```
sudo ls -l /etc/NetworkManager/system-connections/
sudo cat /etc/NetworkManager/system-connections/NazwaPolaczenia.nmconnection
```

> **Pojęcie: plik keyfile.** Format przechowywania profilu połączenia NetworkManagera — zwykły plik tekstowy w stylu INI, z sekcjami `[connection]`, `[ipv4]`, `[ethernet]`. To

właśnie ten plik tworzy, czyta i modyfikuje `nmcli` pod spodem; edycja ręczna działa równie dobrze, o ile po niej wykonamy `nmcli c reload`, żeby NetworkManager zauważył zmianę na dysku.

Przykład:

```
-rw-------. 1 root root 285 wrz 10 11:02 biuro-static.nmconnection


[connection]
id=biuro-static
uuid=3f9a1b2c-4d5e-4f60-8a1b-9c8d7e6f5a4b
type=ethernet
interface-name=ens192

[ipv4]
method=manual
address1=10.20.10.42/24,10.20.10.1
dns=10.20.10.11;

[ipv6]
method=auto
```

Uprawnienia `600` (tylko root) nie są przypadkiem — plik może zawierać hasła (np. do sieci Wi-Fi albo VPN) i NetworkManager świadomie blokuje odczyt zwykłym użytkownikom.

Wi-Fi = Sieć bezprzewodowa: wygodna w biurze, ale w serwerowni dokłada kolejną możliwą przyczynę awarii · **VPN** = Virtual Private Network: szyfrowany tunel do sieci firmowej · **ls** = list: wypisz zawartość katalogu · **cat** = concatenate: sklej i wypisz pliki

TYLKO ODCZYT

Łączenie interfejsów w bonding lub team

```
sudo nmcli c add type bond con-name bond0 ifname bond0 bond.options "mode=active-backup"
sudo nmcli c add type ethernet con-name bond0-ens192 ifname ens192 master bond0
sudo nmcli c add type ethernet con-name bond0-ens224 ifname ens224 master bond0
sudo nmcli c mod bond0 ipv4.addresses AdresIP/24 ipv4.gateway AdresIP ipv4.method manual
sudo nmcli c up bond0
```

Bonding łączy kilka fizycznych kart sieciowych w jeden logiczny interfejs — dla odporności na awarię (`active-backup`: druga karta przejmuje ruch, gdy pierwsza padnie) albo dla większej przepustowości (`802.3ad`/LACP, wymaga wsparcia po stronie przełącznika). `team` to nowsza, bardziej modularna alternatywa o podobnym celu (`nmcli c add type team`), rzadziej spotykana na nowych wdrożeniach niż bonding.

nmcli = NetworkManager command line interface: konfiguracja sieci

TYLKO ODCZYT

Utworzenie interfejsu VLAN: nmcli c add type vlan

```
sudo nmcli c add type vlan con-name NazwaInterfejsu.30 ifname NazwaInterfejsu.30 dev NazwaInterfejsu id 30
sudo nmcli c mod NazwaInterfejsu.30 ipv4.addresses AdresIP/24 ipv4.method manual
sudo nmcli c up NazwaInterfejsu.30
```

Tworzy wirtualny interfejs znakowany tagiem VLAN 30 na fizycznej karcie `NazwaInterfejsu`. Ruch na tym interfejsie jest oznaczany nagłówkiem IEEE 802.1Q — przełącznik po drugiej stronie musi mieć port skonfigurowany jako trunk z dopuszczonym VLAN-em 30, inaczej ramki zostaną odrzucone po drodze.

VLAN = Virtual LAN: wydzielona logicznie sieć na tym samym sprzęcie · **IEEE** = Institute of Electrical and Electronics Engineers: międzynarodowa organizacja inżynierów, która m.in. wydaje normy i przyznaje kamienie milowe w historii techniki · **nmcli** = NetworkManager command line interface: konfiguracja sieci

TYLKO ODCZYT

Wszystkie grupy poleceń `nmcli`

Grupa	Skrót	Dotyczy
`general`	`g`	ogólny status demona i łączności
`device`	`d`	interfejsy widziane przez jądro
`connection`	`c`	zapisane profile ustawień
`networking`	`n`	globalne włączenie/wyłączenie sieci
`radio`	`r`	tryb samolotowy Wi-Fi/WWAN

Uważaj

- `nmcli c mod` zapisuje profil na dysk, ale **nie zmienia** działającej sieci — trzeba `nmcli c up` albo `nmcli d reapply`.
- `nmcli c del` na aktywnym profilu odłącza interfejs natychmiast, bez potwierdzenia.
- Nazwa połączenia (`con-name`) i nazwa urządzenia (`ifname`) to dwa osobne pola — łatwo je pomylić w poleceniach, które przyjmują oba.

Wi-Fi = Sieć bezprzewodowa: wygodna w biurze, ale w serwerowni dokłada kolejną możliwą przyczynę awarii

Z życia wzięte

Z ŻYCIA WZIĘTE

Czwartek. Po aktualizacji na docker01 interfejs sieciowy wraca z innym adresem niż wcześniej, bo NetworkManager przypisał konfigurację do nowej nazwy urządzenia. Tomek poprawia to przez nmcli, wiążąc profil połączenia na stałe z adresem MAC karty, zamiast polegać na nazwie, która może się zmienić przy kolejnej aktualizacji.

Po co to istnieje

Pojęcie: profil połączenia. (zdefiniowane w Fiszce wyżej — kluczowe pojęcie tej strony: zapisany, nazwany komplet ustawień sieciowych, niezależny od tego, czy jest aktualnie użyty).

`ip` z poprzedniej strony zmienia stan sieci natychmiast, ale nietrwale. NetworkManager istnieje po to, żeby ktoś **pamiętał** tę konfigurację między restartami, umiał ją reprezentować jako nazwane profile (dom, biuro, VPN firmowy — dla laptopa; „biuro-static", „biuro-dhcp" — dla serwera) i odtwarzał właściwy profil automatycznie przy starcie systemu albo podłączeniu kabla. Jest to demon działający w tle (`NetworkManager.service`), z którym `nmcli` rozmawia przez D-Bus — sam `nmcli` nic nie robi bez działającego demona.

Na RHEL/Rocky i SLES to domyślny mechanizm zarządzania siecią od kilku dużych wersji wstecz. Ubuntu Desktop też używa NetworkManagera, ale **Ubuntu Server domyślnie korzysta z Netplan**, które renderuje pliki YAML do backendu `systemd-networkd` (rzadziej: do samego

NetworkManagera) — to osobny świat konfiguracji, z innymi plikami i inną składnią, opisany na poprzedniej stronie. Polecenia `nmcli` mają sens tylko tam, gdzie faktycznie działa `NetworkManager.service`.

Analogia

Jeśli `ip` to dyspozytor na hali (poprzednia strona), NetworkManager to **recepcja hotelu z księgą rezerwacji**.

Profil połączenia to rezerwacja: zapisana, nazwana, czeka w księdze niezależnie od tego, czy gość aktualnie śpi w pokoju. `nmcli c add` to złożenie nowej rezerwacji. `nmcli c mod` to zmiana szczegółów istniejącej rezerwacji — numeru pokoju, terminu — ale **sama zmiana wpisu w księdze nie przenosi gościa, który już śpi w starym pokoju**. Trzeba go obudzić i przeprowadzić: to jest `nmcli c up` (pełne wymeldowanie i zameldowanie na nowo, z przerwą) albo `nmcli d reapply` (dyskretna zmiana bez budzenia gościa — działa tylko dla drobniejszych zmian).

Urządzenie (`device`) to sam pokój — istnieje niezależnie od tego, kto go aktualnie rezerwuje. Jeden pokój może mieć w księdze kilka różnych rezerwacji na różne terminy (kilka profili), ale gość śpi w nim tylko jeden na raz.

Jak to działa

Rozwiń: jak to działa pod spodem

`nmcli` to nakładka CLI na demona `NetworkManager`, z którym komunikuje się przez **D-Bus** — międzyprocesowy mechanizm komunikacji w Linuksie. Sam `nmcli` nie modyfikuje bezpośrednio stanu jądra (tym zajmuje się demon, wołając pod spodem te same mechanizmy `netlink`, co `ip`) — jest tylko klientem wysyłającym żądania.



Polecenie nmcli działa przez NetworkManager, który zarządza profilem połączenia i urządzeniem sieciowym. (kliknięcie otwiera pełny rozmiar)

Kluczowy mechanizm do zapamiętania: **profil ma dwa stany — zapisany i zastosowany, i to są dwie różne rzeczy.**

1. `nmcli c mod` zapisuje nowe wartości do pliku `.nmconnection` na dysku (`/etc/NetworkManager/system-connections/` na RHEL/Rocky 9+ i SLES). To jest operacja czysto plikowa plus powiadomienie demona o zmianie w bazie profili.
2. Dopóki nie zajdzie **aktywacja** — `nmcli c up`, `nmcli d reapply`, albo automatyczna reaktywacja po restarcie usługi — jądro nadal ma stare ustawienia. `ip a` w tym momencie pokazuje starą konfigurację, mimo że plik na dysku już mówi co innego.

Różnica między `c up` a `d reapply`: `c up` przechodzi przez pełny cykl aktywacji — DHCP renegocjuje dzierżawę, interfejs Ethernet na chwilę traci i odzyskuje łącze, trasy są budowane od nowa. `d reapply` próbuje wcielić zmiany „w locie", bez tego cyklu — szybciej i bez przerwy w łączności, ale nie każda zmiana ustawień da się zastosować w ten sposób (NetworkManager sam decyduje, co potrafi dociągnąć na żywo, a co wymaga pełnej reaktywacji, i w drugim przypadku zwykle trzeba i tak użyć `c up`).

Na RHEL/Rocky 9 i nowszych zniknęły stare pliki `/etc/sysconfig/network-scripts/ifcfg-*`, znane z RHEL 7/8 — cała konfiguracja jest teraz w formacie keyfile pod `/etc/NetworkManager/system-connections/*.nmconnection`. Kto szuka `ifcfg-ens192` na RHEL 9, nie znajdzie niczego — to najczęstsza pomyłka administratorów migrujących ze starszych wersji, rozwinięta w *Częste błędy*.

Krok po kroku

1. Zobacz urządzenia i przypisane im profile.

```
nmcli d status
```

```
DEVICE  TYPE      STATE                   CONNECTION
ens192  ethernet  connected               biuro-dhcp
lo      loopback  unmanaged               --
```

2. Przygotuj docelowy profil ze statycznym adresem, nie ruszając jeszcze aktywnego połączenia.

```
sudo nmcli c add type ethernet con-name biuro-static ifname ens192 \
  ipv4.method manual ipv4.addresses 10.20.10.42/24 ipv4.gateway 10.20.10.1 ipv4.dns
10.20.10.11
```

```
Connection 'biuro-static' (3f9a1b2c-4d5e-4f60-8a1b-9c8d7e6f5a4b) successfully added.
```

Nowy profil istnieje na dysku, ale interfejs nadal korzysta z `biuro-dhcp` — `ip -br a` w tym momencie nic nie zmieni.

3. Przełącz na nowy profil.

```
sudo nmcli c up biuro-static
ip -br a
```

```
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/7)
ens192           UP             10.20.10.42/24
```

Dopiero teraz `ip -br a` (z poprzedniej strony) pokazuje nowy adres — to weryfikacja niezależnym narzędziem, że zmiana faktycznie doszła do jądra.

4. Sprawdź, czy zmiana przetrwa restart, bez uciekania się do prawdziwego restartu.

```
sudo cat /etc/NetworkManager/system-connections/biuro-static.nmconnection | grep -A4
'\[ipv4\]'
```

```
[ipv4]
method=manual
address1=10.20.10.42/24,10.20.10.1
dns=10.20.10.11;
```

Skoro ustawienia leżą w pliku, a nie tylko w pamięci jądra, przetrwają restart — inaczej niż w przypadku samego `ip a add` z poprzedniej strony.

5. Zmień DNS bez zrywania łącza.

```
sudo nmcli c mod biuro-static ipv4.dns "10.20.10.11,10.20.10.12"
sudo nmcli d reapply ens192
nmcli c show biuro-static | grep ipv4.dns
```

```
Connection successfully reapplied to device 'ens192'.
ipv4.dns:                               10.20.10.11,10.20.10.12
```

Łącze Ethernet ani na moment nie zgasło — `reapply` dociągnął zmianę DNS bez pełnej reaktywacji.

Częste błędy

Zmiana zapisana, ale nigdy niezastosowana.

```
sudo nmcli c mod biuro-static ipv4.addresses 10.20.10.48/24
ip -br a
```

```
ens192           UP             10.20.10.42/24
```

Adres w pliku profilu to już `10.20.10.48/24`, ale jądro wciąż ma stary `10.20.10.42/24` — `nmcli c mod` **nigdy** nie stosuje zmiany od razu. Naprawa: `nmcli c up biuro-static` albo `nmcli d reapply ens192` zaraz po każdej modyfikacji, którą trzeba zobaczyć na żywo.

Szukanie plików `ifcfg-*` na RHEL 9+.

```
$ ls /etc/sysconfig/network-scripts/
ls: cannot access '/etc/sysconfig/network-scripts/': No such file or directory
```

Katalog ze starymi plikami `ifcfg-ens192` zniknął wraz z migracją na format keyfile w RHEL 8→9. Naprawa: konfiguracja jest teraz w `/etc/NetworkManager/system-connections/*.nmconnection`; `nmcli c show nazwa` daje ten sam wgląd bez znajomości formatu pliku.

Usunięcie aktywnego profilu przez zdalną sesję.

```
sudo nmcli c del biuro-static
```

```
Connection 'biuro-static' (3f9a1b2c-4d5e-4f60-8a1b-9c8d7e6f5a4b) successfully deleted.
```

Jeśli `biuro-static` był profilem, przez który akurat trwa sesja SSH, połączenie ucina się w momencie wykonania tej komendy — bez potwierdzenia i bez możliwości cofnięcia z tej samej sesji. Naprawa: przed usunięciem aktywnego profilu upewnij się, że masz inną drogę dostępu (konsola, drugi interfejs, KVM) albo najpierw aktywuj zastępczy profil (`nmcli c up`), a dopiero potem usuń stary.

Pomylenie nazwy połączenia z nazwą urządzenia.

```
sudo nmcli c up ens192
```

```
Error: Connection activation failed: No suitable connection found: 'ens192'.
```

`ens192` to nazwa urządzenia, a `nmcli c up` oczekuje nazwy **połączenia** (`con-name`), która może być zupełnie inna, np. `biuro-static`. Naprawa: `nmcli c show` pokazuje, jak faktycznie nazywa się profil,

który chcesz aktywować; jeśli naprawdę chcesz działać na urządzeniu, właściwe polecenie to `nmcli d connect ens192` (aktywuje ostatnio używany albo domyślny profil dla tego urządzenia).

Zmiana `ipv4.method` zignorowana przez `reapply`.

```
sudo nmcli c mod biuro-dhcp ipv4.method manual ipv4.addresses 10.20.10.42/24
sudo nmcli d reapply ens192
ip -br a
```

```
ens192           UP             10.20.10.112/24
```

Adres nadal pochodzi z DHCP — zmiana metody adresacji to jedna z tych, których `reapply` nie potrafi wcielić bez pełnej reaktywacji. Naprawa: `nmcli c up biuro-dhcp` zamiast `reapply`, gdy zmienia się sam sposób adresacji, nie tylko wartości w jej obrębie.

□□ Ćwiczenia

> Wszystko poniżej wykonujesz na **interfejsie testowym typu `dummy`**, tworzonym tylko na czas ćwiczenia — realne karty sieciowe maszyny (np. ta, przez którą jesteś połączony przez SSH) pozostają nietknięte. Wymaga `sudo`.

Ćwiczenie 1 — profil na interfejsie, który nic nie może zepsuć

```
sudo nmcli c add type dummy ifname dummy0 con-name test-dummy \
  ipv4.method manual ipv4.addresses 192.0.2.50/24
sudo nmcli c up test-dummy
ip -br a show dummy0
nmcli c show test-dummy | grep -E 'ipv4.method|ipv4.addresses|GENERAL.STATE'
```

Sprawdź, czy adres widoczny w `ip -br a` zgadza się z tym zapisanym w profilu. Teraz posprzątaj:

```
sudo nmcli c del test-dummy
sudo ip link del dummy0 2>/dev/null
```

Odpowiedź

Adres `192.0.2.50/24` powinien być widoczny w obu miejscach — `nmcli c up` wykonał pełną aktywację, więc zmiana z pliku profilu trafiła do jądra od razu. Gdyby zamiast `nmcli c up` użyć tylko `nmcli c add` (bez aktywacji), `ip -br a` pokazałby interfejs `dummy0` bez żadnego adresu — profil istniałby, ale nie byłby zastosowany, dokładnie ten sam mechanizm co w *Częste błędy* wyżej.

Ćwiczenie 2 — zobacz różnicę między zapisaniem a zastosowaniem na własne oczy

```
sudo nmcli c add type dummy ifname dummy1 con-name test-dummy2 \
  ipv4.method manual ipv4.addresses 192.0.2.60/24
sudo nmcli c up test-dummy2
ip -br a show dummy1

sudo nmcli c mod test-dummy2 ipv4.addresses 192.0.2.61/24
echo "--- zaraz po nmcli c mod, przed reapply ---"
ip -br a show dummy1

sudo nmcli d reapply dummy1
echo "--- po reapply ---"
ip -br a show dummy1

sudo nmcli c del test-dummy2
sudo ip link del dummy1 2>/dev/null
```

W którym momencie `ip -br a` pokazuje nowy adres `192.0.2.61/24`?

Odpowiedź

Dopiero po `nmcli d reapply dummy1`. Zaraz po `nmcli c mod` interfejs nadal pokazuje stary adres `192.0.2.60/24` — plik profilu już ma nową wartość, ale jądro jeszcze nie. To jest dokładnie mechanizm opisany w sekcji *Jak to działa*: zapisanie i zastosowanie to dwa osobne kroki.

Pytania kontrolne

1. Czym różni się urządzenie (`device`) od połączenia (`connection`) w terminologii NetworkManagera?

Odpowiedź

Urządzenie to fizyczny albo wirtualny interfejs widziany przez jądro (to, co pokazuje `ip link`). Połączenie to zapisany profil ustawień — nazwany komplet adresu, bramy, DNS i innych parametrów — który można podpiąć pod urządzenie. Jedno urządzenie może mieć zdefiniowanych kilka profili połączeń, z których aktywny jest najwyżej jeden naraz.

2. Wykonałeś `nmcli c mod biuro-static ipv4.dns 10.20.10.12` i `cat /etc/resolv.conf` nadal pokazuje stary serwer DNS. Co przegapiłeś?

Odpowiedź

Aktywację zmiany — `nmcli c mod` tylko zapisuje nową wartość do profilu na dysku, nie stosuje jej. Potrzeba `nmcli c up biuro-static` albo `nmcli d reapply <urządzenie>`, żeby NetworkManager przepisał nowy DNS do działającej konfiguracji (i do `/etc/resolv.conf`, jeśli tak jest skonfigurowany).

3. Na serwerze z RHEL 9 szukasz pliku `/etc/sysconfig/network-scripts/ifcfg-ens192` i go nie ma. Czy to oznacza uszkodzoną konfigurację sieci?

Odpowiedź

Nie. RHEL 9 (i nowsze) przechowuje profile NetworkManagera w formacie keyfile, pod `/etc/NetworkManager/system-connections/*.nmconnection`, a nie w starym formacie `ifcfg-*` znanym z RHEL 7/8. Brak tego pliku jest normalny; konfigurację sprawdzasz przez `nmcli c show` albo bezpośrednio przez plik `.nmconnection`.

4. Czym różni się `nmcli c up` od `nmcli d reapply` i kiedy `reapply` może nie zadziałać?

Odpowiedź

`nmcli c up` przechodzi przez pełny cykl aktywacji połączenia — z przerwą w łączności (renegocjacja DHCP, ponowne zestawienie łącza). `nmcli d reapply` próbuje wcielić bieżące ustawienia profilu bez tej przerwy. `reapply` nie zadziała dla zmian, których NetworkManager nie potrafi dociągnąć „na żywo" — klasyczny przykład to zmiana `ipv4.method` z `auto` na `manual`, która wymaga pełnej reaktywacji przez `c up`.

Powiązane: ip: adresy, trasy, interfejsy · Kto nasłuchuje i kto z kim gada: ss, lsof, netstat · firewalld: strefy, porty, usługi, rich rules

Źródła: `man 1 nmcli` · `man 5 nm-settings-keyfile` · `man 1 nmtui` · OneNote „Ubuntu" (Netplan — zarządzanie siecią, jako kontrast wobec NetworkManagera)

Revision #17
Created 2026-09-17 21:43:15 UTC by Tomasz Konieczny
Updated 2026-10-03 19:08:57 UTC by Tomasz Konieczny