Repozytoria: konfiguracja, EPEL, priorytety

Poziom: podstawy **Dotyczy: RHEL/Rocky 9–10, różnice RHEL/Rocky 7–8 zaznaczone**

Czas: ~18 min

Repozytorium mówi `dnf`, gdzie szukać pakietów. EPEL dokłada do RHEL-a i Rocky'ego setki popularnych narzędzi, których oficjalne repozytoria świadomie nie zawierają — ta strona pokazuje, jak je włączyć, nie psując przy tym reszty systemu.

Skróty użyte na tej stronie (5)

Skrót	Co znaczy
EPEL	Extra Packages for Enterprise Linux: dodatkowe repozytorium pakietów dla RHEL i klonów
GPG	GNU Privacy Guard: program do podpisywania i szyfrowania plików oraz sprawdzania podpisów paczek
RHEL	Red Hat Enterprise Linux: komercyjna dystrybucja Linuksa
RPM	RPM Package Manager: format paczek i narzędzie instalujące oprogramowanie w RHEL/Rocky, dnf korzysta z niego pod spodem
URL	Uniform Resource Locator: adres strony lub zasobu

Fiszka

TYLKO ODCZYT

Wyświetlenie aktualnie włączonych repozytoriów: `dnf repolist`

```
dnf repolist
```

> **Pojęcie: repozytorium pakietów.** Serwer (albo katalog lokalny) z zestawem pakietów i metadanymi opisującymi ich wersje oraz zależności. `dnf` nie wie samo z

siebie, gdzie szukać oprogramowania — musimy mu wskazać repozytoria, a każde z nich reprezentuje osobny plik `.repo` w `/etc/yum.repos.d/`.

Przykład:

```
repo id                                 repo name
appstream                               Rocky Linux 9 - AppStream
baseos                                  Rocky Linux 9 - BaseOS
extras                                  Rocky Linux 9 - Extras
```

dnf = Dandified YUM: menedżer pakietów RHEL/Rocky

TYLKO ODCZYT

Wyświetlenie repozytoriów wraz z wyłączonymi: `dnf repolist --all`

```
dnf repolist --all
```

Pokazujemy wszystko, co jest zdefiniowane w `/etc/yum.repos.d/`, niezależnie od tego, czy jest aktywne. Kolumna `status` mówi, które są `enabled`, a które `disabled`.

Przykład:

```
repo id                                 repo name                                status
appstream                               Rocky Linux 9 - AppStream                enabled
baseos                                  Rocky Linux 9 - BaseOS                   enabled
epel                                    Extra Packages for Enterprise Linux 9    disabled
crb                                     Rocky Linux 9 - CRB                      disabled
```

dnf = Dandified YUM: menedżer pakietów RHEL/Rocky

TYLKO ODCZYT

Wyświetlenie pełnych szczegółów repozytoriów: `dnf repolist -v`

```
dnf repolist -v
```

Rozwinięta wersja: adres `baseurl` lub `mirrorlist`, liczba pakietów, status GPG. Przydaje się, gdy trzeba sprawdzić, skąd dokładnie repozytorium pobiera dane, bez otwierania pliku `.repo`.

Przykład:

```
Repo-id            : epel
Repo-name          : Extra Packages for Enterprise Linux 9 - x86_64
Repo-status        : enabled
Repo-baseurl       : https://download.fedoraproject.org/pub/epel/9/Everything/x86_64/
Repo-pkgs          : 15234
Repo-size          : 32 G
```

GPG = GNU Privacy Guard: program do podpisywania i szyfrowania plików oraz sprawdzania podpisów paczek · **dnf** = Dandified YUM: menedżer pakietów RHEL/Rocky

ZMIENIA SYSTEM

Dodanie repozytorium z URL-a: `dnf config-manager --add-repo`

```
dnf config-manager --add-repo https://repo.zielona.example/moje-repo.repo
```

Pobieramy plik `.repo` spod wskazanego adresu i zapisujemy go w `/etc/yum.repos.d/`. Szybszy sposób niż ręczne tworzenie pliku, jeśli dostawca repozytorium publikuje gotowy plik konfiguracyjny.

dnf = Dandified YUM: menedżer pakietów RHEL/Rocky

ZMIENIA SYSTEM

Włączenie i wyłączenie repozytorium: `dnf config-manager --set-enabled` / `--set-disabled`

```
dnf config-manager --set-enabled crb
dnf config-manager --set-disabled epel
```

Zmieniamy pole `enabled` w pliku `.repo` na trwałe — od tej pory repozytorium jest brane pod uwagę (albo pomijane) przy każdej kolejnej komendzie `dnf`, bez potrzeby dopisywania flag za każdym razem.

dnf = Dandified YUM: menedżer pakietów RHEL/Rocky

TYLKO ODCZYT

Budowa pliku `/etc/yum.repos.d/nazwa.repo`

```
cat /etc/yum.repos.d/epel.repo
```

Przykład:

```
[epel]
name=Extra Packages for Enterprise Linux 9 - $basearch
baseurl=https://download.fedoraproject.org/pub/epel/9/Everything/$basearch
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-9
```

Pola, które warto znać osobno:

- **`name`** — czytelna nazwa wyświetlana w `dnf repolist`, nie wpływa na działanie.
- **`baseurl`** — bezpośredni adres katalogu z pakietami; `$basearch` i `$releasever` są podstawiane automatycznie na architekturę i wersję systemu.
- **`mirrorlist`** — alternatywa dla `baseurl`: adres pliku z listą serwerów lustrzanych, spośród których `dnf` sam wybiera najszybszy. Repozytorium ma zwykle jedno albo drugie, rzadko oba naraz.
- **`enabled`** — `1` włącza repozytorium, `0` wyłącza; to samo pole zmienia `--set-enabled` / `--set-disabled`.
- **`gpgcheck`** — `1` wymusza sprawdzenie podpisu każdego pakietu przed instalacją; `0` wyłącza sprawdzanie (unikamy tego na produkcji).
- **`gpgkey`** — ścieżka albo URL do klucza publicznego, którym repozytorium podpisuje pakiety.
- **`priority`** — liczbowy priorytet repozytorium używany przez wtyczkę `dnf-plugin-priorities`; niższa liczba = wyższy priorytet. Bez tej wtyczki pole jest ignorowane.
- **`module_hotfixes`** — opisany niżej, osobno, bo dotyczy tylko repozytoriów z pakietami modularnymi.

URL = Uniform Resource Locator: adres strony lub zasobu · **cat** = concatenate: sklej i wypisz pliki

ZMIENIA SYSTEM

Podstawowa instalacja EPEL: `dnf install epel-release`

```
dnf install epel-release
```

> **Pojęcie: EPEL.** *Extra Packages for Enterprise Linux* — repozytorium utrzymywane przez społeczność Fedory, dostarczające pakiety, które nie trafiają do oficjalnych repozytoriów RHEL/Rocky, bo Red Hat świadomie ich nie wspiera komercyjnie albo

dlatego, że są zbyt niszowe dla dystrybucji korporacyjnej. Pakiet `epel-release` instaluje tylko plik `.repo` i klucz GPG — nie instaluje żadnego prawdziwego oprogramowania, tylko otwiera drzwi do repozytorium, z którego można je potem pobrać.

Przykład:

```
Zależności rozwiązane.
================================================================================
 Pakiet         Architektura  Wersja      Repozytorium   Rozmiar
================================================================================
Instalowanie:
 epel-release   noarch        9-7.el9     extras         16 k
```

EPEL = Extra Packages for Enterprise Linux: dodatkowe repozytorium pakietów dla RHEL i klonów · **RHEL** = Red Hat Enterprise Linux: komercyjna dystrybucja Linuksa · **GPG** = GNU Privacy Guard: program do podpisywania i szyfrowania plików oraz sprawdzania podpisów paczek · **dnf** = Dandified YUM: menedżer pakietów RHEL/Rocky

TYLKO ODCZYT

Ręczna instalacja EPEL z pliku RPM (offline albo bez dostępu do internetowego repo)

```
rpm -ivh https://dl.fedoraproject.org/pub/epel/epel-release-latest-9.noarch.rpm
```

Odpowiednik `dnf install epel-release`, ale przez bezpośredni adres URL do pliku `.rpm` — działa nawet wtedy, gdy `epel-release` nie jest jeszcze widoczny w żadnym skonfigurowanym repozytorium (np. świeżo postawiony system bez `extras`). `rpm -ivh` (`i` — install, `v` — verbose, `h` — pasek postępu z `#`) instaluje sam pakiet bez rozwiązywania zależności z sieci — dla `epel-release`, który zależności praktycznie nie ma, to nie problem.

Przykład:

```
Retrieving https://dl.fedoraproject.org/pub/epel/epel-release-latest-9.noarch.rpm
warning: /var/tmp/rpm-tmp.XXXXXX: Header V4 RSA/SHA256 Signature, key ID 3228467c: NOKEY
Verifying...                          ################################# [100%]
Preparing...                          ################################# [100%]
Updating / installing...
   1:epel-release-9-7.el9             ################################# [100%]
```

Ten sam wzorzec działa dla instalacji offline: pobieramy plik `.rpm` na maszynie z dostępem do internetu, przenosimy go (np. `scp`) na docelowy serwer bez dostępu do sieci zewnętrznej, i uruchamiamy `rpm -ivh /sciezka/do/epel-release-latest-9.noarch.rpm` lokalnie — sam plik wystarczy, nic więcej nie trzeba pobierać z sieci.

EPEL = Extra Packages for Enterprise Linux: dodatkowe repozytorium pakietów dla RHEL i klonów · **RPM** = RPM Package Manager: format paczek i narzędzie instalujące oprogramowanie w RHEL/Rocky, dnf korzysta z niego pod spodem · **URL** = Uniform Resource Locator: adres strony lub zasobu · **rpm** = RPM Package Manager: format i narzędzie paczek RHEL

ZMIENIA SYSTEM

Warianty per dystrybucja i wersja

RHEL wymaga dodatkowo repozytorium CodeReady Builder (CRB), Rocky Linux — jego odpowiednika o innej nazwie:

RHEL 9:

```
subscription-manager repos --enable codeready-builder-for-rhel-9-$(arch)-rpms
dnf install https://dl.fedoraproject.org/pub/epel/epel-release-latest-9.noarch.rpm
```

Rocky Linux 9:

```
dnf config-manager --set-enabled crb
dnf install epel-release
```

RHEL 8:

```
subscription-manager repos --enable codeready-builder-for-rhel-8-$(arch)-rpms
dnf install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
```

Rocky Linux 8:

```
dnf config-manager --set-enabled powertools
dnf install epel-release
```

RHEL 7 (starsze środowiska, `yum` zamiast `dnf`):

```
subscription-manager repos --enable rhel-*-optional-rpms \
                           --enable rhel-*-extras-rpms \
                           --enable rhel-ha-for-rhel-*-server-rpms
```

```
yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
subscription-manager repos --disable=rhel-7-server-eus-optional-rpms
```

CentOS 7:

```
yum install epel-release
```

RHEL = Red Hat Enterprise Linux: komercyjna dystrybucja Linuksa · **dnf** = Dandified YUM: menedżer pakietów RHEL/Rocky · **yum** = Yellowdog Updater Modified: starszy menedżer pakietów RHEL

TYLKO ODCZYT

CRB / PowerTools jako wymóg EPEL

Repozytorium **CRB** (RHEL 9, wcześniej `codeready-builder`) albo **PowerTools** (Rocky/RHEL 8) zawiera pakiety deweloperskie i biblioteki, od których zależy część pakietów w EPEL. Bez włączenia CRB/PowerTools instalacja niektórych pakietów z EPEL zakończy się błędem brakującej zależności, mimo że sam EPEL jest poprawnie skonfigurowany — dlatego procedura instalacji EPEL na RHEL/Rocky 8–9 zawsze zaczyna się od włączenia jednego z tych dwóch repozytoriów.

EPEL = Extra Packages for Enterprise Linux: dodatkowe repozytorium pakietów dla RHEL i klonów · **RHEL** = Red Hat Enterprise Linux: komercyjna dystrybucja Linuksa

ZMIENIA SYSTEM

Jednorazowe użycie repozytorium: `dnf --enablerepo=epel install`

```
dnf --enablerepo=epel install NazwaPakietu
```

Włączamy repozytorium `epel` tylko na czas tej jednej komendy, nie zmieniając trwale pola `enabled` w pliku `.repo`. To jest sposób, w jaki EPEL zwykle trzyma się na produkcji: repozytorium zostaje skonfigurowane i wyłączone (`enabled=0`), a włączamy je punktowo, per komenda, kiedy naprawdę trzeba coś z niego zainstalować.

EPEL = Extra Packages for Enterprise Linux: dodatkowe repozytorium pakietów dla RHEL i klonów · **dnf** = Dandified YUM: menedżer pakietów RHEL/Rocky

TYLKO ODCZYT

Ręczny import klucza GPG: `rpm --import`

```
rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-9
```

> **Pojęcie: klucz GPG i podpis pakietu.** Każdy pakiet w rzetelnie utrzymywanym repozytorium jest podpisany kluczem prywatnym dostawcy; `dnf`/`rpm` sprawdza ten podpis kluczem publicznym, żeby potwierdzić, że pakiet pochodzi od zadeklarowanego źródła i nie został podmieniony po drodze. `rpm --import` wgrywa klucz publiczny do lokalnej bazy zaufanych kluczy — bez tego kroku pierwsza instalacja z nowego repozytorium zapyta o potwierdzenie importu klucza albo, przy `-y` i braku wcześniejszego importu, zakończy się błędem weryfikacji podpisu.

Przykład:

```
(brak wyjścia przy powodzeniu)
```

GPG = GNU Privacy Guard: program do podpisywania i szyfrowania plików oraz sprawdzania podpisów paczek · **rpm** = RPM Package Manager: format i narzędzie paczek RHEL

TYLKO ODCZYT

Wyświetlenie listy zaimportowanych kluczy: `rpm -qa gpg-pubkey`

```
rpm -qa gpg-pubkey*
```

Klucze GPG w bazie RPM są traktowane jak specyficzny rodzaj „pakietu" o nazwie `gpg-pubkey`. Sprawdzamy tą komendą, jakie klucze system już zna i im ufa.

Przykład:

```
gpg-pubkey-3228467c-6a2a5a15
gpg-pubkey-fd431d51-4ae0493b
```

`rpm -qi gpg-pubkey-3228467c-6a2a5a15` pokaże pełne dane klucza — właściciela i datę importu.

GPG = GNU Privacy Guard: program do podpisywania i szyfrowania plików oraz sprawdzania podpisów paczek · **RPM** = RPM Package Manager: format paczek i narzędzie instalujące oprogramowanie w RHEL/Rocky, dnf korzysta z niego pod spodem · **rpm** = RPM Package Manager: format i narzędzie paczek RHEL

ZMIENIA SYSTEM

Instalacja bez weryfikacji podpisu: `dnf install --nogpgcheck`

```
dnf install --nogpgcheck ./pakiet-bez-podpisu.rpm
```

Ostrzeżenie: wyłącza sprawdzanie podpisu GPG dla tej instalacji. Legalny użytek to pakiet zbudowany wewnętrznie, który świadomie nigdy nie był podpisany — ale ta sama flaga otwiera drzwi do zainstalowania czegokolwiek podmienionego po drodze, bez żadnego ostrzeżenia poza tym jednym razem. Nie używamy tego jako rutynowego obejścia błędu „NOKEY" — zamiast tego importujemy właściwy klucz przez `rpm --import`.

GPG = GNU Privacy Guard: program do podpisywania i szyfrowania plików oraz sprawdzania podpisów paczek · **dnf** = Dandified YUM: menedżer pakietów RHEL/Rocky

TYLKO ODCZYT

Wyświetlenie pełnej konfiguracji repozytorium: `dnf config-manager --dump`

```
dnf config-manager --dump epel
```

Wypisujemy wszystkie efektywne ustawienia repozytorium — łącznie z wartościami domyślnymi, których nie ma jawnie w pliku `.repo`. Sięgamy po to do diagnozowania, dlaczego repozytorium zachowuje się inaczej, niż wynikałoby z samego pliku konfiguracyjnego.

dnf = Dandified YUM: menedżer pakietów RHEL/Rocky

TYLKO ODCZYT

Przeszukanie tylko jednego repozytorium: `dnf repoquery --repo=epel`

```
dnf repoquery --repo=epel --all | grep NazwaPakietu
```

Ograniczamy wynik `repoquery` do jednego wskazanego repozytorium, ignorując pozostałe — przydatne, żeby sprawdzić, czy dany pakiet naprawdę pochodzi z EPEL, a nie z jakiegoś innego repozytorium o podobnej nazwie pakietu.

Przykład:

```
htop-0:3.2.2-2.el9.x86_64
```

EPEL = Extra Packages for Enterprise Linux: dodatkowe repozytorium pakietów dla RHEL i klonów · **dnf** = Dandified YUM: menedżer pakietów RHEL/Rocky

◫◫ ZMIENIA SYSTEM

Priorytet repozytorium

```
dnf install dnf-plugin-priorities
```

> **Pojęcie: priorytet repozytorium.** Liczba (domyślnie 99, im niżej tym ważniejsze) decydująca, z którego repozytorium `dnf` weźmie pakiet, jeśli ten sam pakiet w tej samej wersji istnieje w kilku repozytoriach naraz. Bez wtyczki priorytetów `dnf` bierze zwykle nowszą wersję niezależnie od repozytorium — z priorytetami możemy wymusić, że np. `baseos` zawsze wygrywa z `epel` dla pakietów, które występują w obu.

```
[epel]
name=Extra Packages for Enterprise Linux 9
priority=99
```

◫◫ **dnf** = Dandified YUM: menedżer pakietów RHEL/Rocky

◫◫ PLIK KONFIGURACYJNY

`module_hotfixes` — wyjątek dla repozytoriów z pakietami modularnymi

```
[epel]
name=Extra Packages for Enterprise Linux 9
module_hotfixes=1
```

Repozytoria bazowe RHEL/Rocky (`baseos`, `appstream`) korzystają z modułów (patrz strona o `dnf` — moduły i strumienie). EPEL nie jest modularny, ale czasem publikuje pakiet w tej samej nazwie i wersji co pakiet dostarczany przez moduł z `appstream` — bez `module_hotfixes=1` `dnf` odrzuci taki pakiet z EPEL jako „ukryty" przez filtr modularności, nawet gdy jawnie o niego poprosimy. Ustawienie `module_hotfixes=1` mówi `dnf`, żeby dla tego konkretnego repozytorium zignorował filtrowanie modularne.

◫◫ **RHEL** = Red Hat Enterprise Linux: komercyjna dystrybucja Linuksa · **EPEL** = Extra Packages for Enterprise Linux: dodatkowe repozytorium pakietów dla RHEL i klonów

◫◫ TYLKO ODCZYT

Wszystkie flagi i komendy

Komenda / pole	Co robi

`dnf repolist [--all\	-v]`	lista repozytoriów (włączone/wszystkie/zestaw filtrami)
`dnf config-manager --add-repo URL`	dodaje repozytorium z pliku `.repo`	
`dnf config-manager --set-enabled/--set-disabled`	trwale włącza/wyłącza repozytorium	

`dnf config-manager --dump`	pełna efektywna konfiguracja repozytorium
`baseurl` / `mirrorlist`	źródło pakietów: adres bezpośredni albo lista mirrorów
`gpgcheck` / `gpgkey`	weryfikacja podpisu i klucz publiczny
`priority`	kolejność wyboru przy konflikcie tego samego pakietu (wymaga wtyczki)
`module_hotfixes`	wyłącza filtr modularności dla repozytorium
`dnf install epel-release`	instalacja EPEL z repozytorium `extras`
`rpm -ivh .../epel-release-latest-N.noarch.rpm`	ręczna/offline instalacja EPEL
`dnf --enablerepo=epel install`	jednorazowe użycie repozytorium bez trwałej zmiany
`rpm --import klucz`	import klucza GPG
`rpm -qa gpg-pubkey*`	lista zaufanych kluczy
`dnf install --nogpgcheck`	instalacja bez weryfikacji podpisu (ostrożnie)
`dnf repoquery --repo=nazwa`	przeszukanie jednego repozytorium

Uważaj

- Włączenie CRB/PowerTools jest wymagane przed instalacją wielu pakietów z EPEL na RHEL/Rocky 8–9 — pominięcie tego kroku daje mylący błąd o brakującej zależności, niezwiązany pozornie z EPEL.
- `--nogpgcheck` wyłącza jedyną linię obrony przed podmienionym pakietem — traktuj jako ostatnią deskę ratunku, nie nawyk.
- Adres `epel-release-latest-9.noarch.rpm` zawsze wskazuje na *najnowszą* wersję pakietu `epel-release` dla danej głównej wersji systemu — nie ma osobnych adresów dla każdej podwersji (9.2, 9.4 itd.).

EPEL = Extra Packages for Enterprise Linux: dodatkowe repozytorium pakietów dla RHEL i klonów · **RHEL** = Red Hat Enterprise Linux: komercyjna dystrybucja Linuksa

Z życia wzięte

Z ŻYCIA WZIĘTE

Środa. Tomek chce zainstalować htop na nowym serwerze pg01, ale dnf nie widzi takiego pakietu w żadnym włączonym repozytorium. Dopiero po włączeniu EPEL i wymaganego CRB pakiet się znajduje, a Tomek zapamiętuje, żeby EPEL trzymać domyślnie wyłączone i włączać

> je tylko na czas pojedynczej instalacji.

Po co to istnieje

Oficjalne repozytoria RHEL (`baseos`, `appstream`) są utrzymywane i wspierane komercyjnie — Red Hat testuje każdy pakiet, gwarantuje zgodność wsteczną i odpowiada za łatanie luk. To ma cenę: żeby coś trafiło do tych repozytoriów, musi przejść proces certyfikacji i utrzymania, który dla wielu użytecznych, ale niszowych narzędzi (`htop`, `screen`, `jq`, `tmux`) po prostu się nie opłaca. EPEL wypełnia tę lukę — te same pakiety, zbudowane zgodnie ze standardami RHEL, ale bez formalnego wsparcia Red Hata, utrzymywane przez społeczność Fedory.

Dlatego administratorzy produkcyjni często trzymają EPEL wyłączony na stałe i włączają go punktowo (`--enablerepo=epel`) tylko na czas instalacji konkretnego pakietu: chcą korzystać z bogactwa EPEL, ale nie chcą, żeby `dnf update` bez pytania podmienił pakiet z oficjalnego repozytorium na wersję z EPEL, gdyby nazwa się kiedyś pokryła.

Analogia

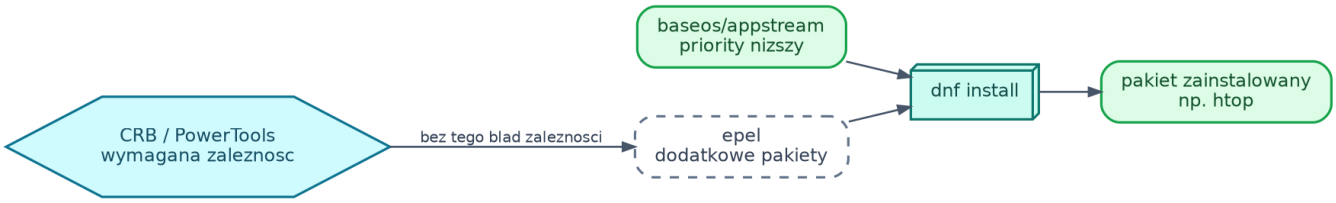
Repozytorium to **hurtownia z katalogiem**, a `baseurl`/`mirrorlist` to jej adres.

Oficjalne repozytoria RHEL to hurtownia produkcyjna z certyfikatem jakości na każdym produkcie i telefonem serwisowym, pod który można zadzwonić, jeśli coś się zepsuje. EPEL to sąsiedni magazyn prowadzony przez fachowców-wolontariuszy: towar solidny, często dokładnie ten sam co u konkurencji o wyższej cenie, ale bez papierowego certyfikatu i bez numeru serwisowego. Priorytet repozytorium to reguła na wejściu do magazynu: „jeśli oba magazyny mają ten sam produkt, bierzemy zawsze z certyfikowanego, chyba że certyfikowany go nie ma".

Klucz GPG to pieczęć producenta na opakowaniu. Zanim wypuścisz towar na halę (zainstalujesz pakiet), sprawdzasz pieczęć — jeśli jest naruszona albo jej nie ma, produkt mógł zostać podmieniony po drodze przez kogoś trzeciego. `--nogpgcheck` to decyzja „wpuszczam towar bez sprawdzania pieczęci" — czasem uzasadniona (towar własnej produkcji, nigdy nie miał pieczęci), ale nigdy nie powinna być procedurą standardową.

Jak to działa

Kiedy `dnf` szuka pakietu, przegląda metadane wszystkich **włączonych** repozytoriów po kolei i buduje jedną wspólną listę dostępnych pakietów. Jeśli ten sam pakiet w tej samej wersji występuje w dwóch repozytoriach naraz, o wyborze decyduje:



Instalacja pakietu z EPEL wymaga dodatkowo repozytorium CRB, inaczej zależności się nie domkną. (kliknięcie otwiera pełny rozmiar)

1. **Priorytet** (`priority` w pliku `.repo`), jeśli wtyczka `dnf-plugin-priorities` jest zainstalowana — niższa liczba wygrywa.
2. Bez priorytetów: `dnf` bierze zwykle repozytorium przetwarzane jako pierwsze w kolejności alfabetycznej nazw plików `.repo`, co jest zachowaniem niedeterministycznym z punktu widzenia administratora i lepiej się na nim nie opierać.

Filtr modularności (dotyczy repozytoriów bazowych RHEL/Rocky, patrz strona o `dnf`) działa niezależnie od priorytetów: jeśli pakiet o danej nazwie jest częścią modułu w `appstream`, a EPEL próbuje dostarczyć pakiet o tej samej nazwie spoza modułu, `dnf` domyślnie ukrywa wersję z EPEL, żeby nie kolidowała z systemem modułów. `module_hotfixes=1` w definicji repozytorium wyłącza to ukrywanie dla tego jednego repozytorium.

Weryfikacja GPG działa na dwóch poziomach: `repo_gpgcheck` sprawdza podpis samych metadanych repozytorium (czy lista pakietów nie została podmieniona), a `gpgcheck` sprawdza podpis każdego pobieranego pakietu osobno. Klucz publiczny, którym `dnf` weryfikuje podpisy, musi wcześniej trafić do lokalnej bazy — albo automatycznie przy pierwszej instalacji z danego repozytorium (z pytaniem o potwierdzenie odcisku klucza), albo ręcznie przez `rpm --import`.

Krok po kroku

1. Sprawdź, co jest już włączone.

```
dnf repolist
```

2. Na Rocky Linux 9 włącz CRB — wymóg EPEL.

```
dnf config-manager --set-enabled crb
```

3. Zainstaluj EPEL.

```
dnf install epel-release
```

```
Instalowanie:
 epel-release   noarch   9-7.el9   extras   16 k
```

4. Sprawdź, że repozytorium się pojawiło — i od razu je wyłącz, jeśli chcesz trzymać je „na żądanie".

```
dnf repolist --all | grep epel
dnf config-manager --set-disabled epel
```

```
epel     Extra Packages for Enterprise Linux 9 - x86_64     enabled
```

5. Od teraz instaluj z EPEL punktowo, tylko kiedy potrzebujesz konkretnego pakietu.

```
dnf --enablerepo=epel install htop
```

Repozytorium włącza się tylko na czas tej komendy — kolejne `dnf update` go nie dotknie.

Częste błędy

Instalacja pakietu z EPEL na RHEL/Rocky 8–9 kończy się błędem brakującej zależności.

```
Błąd:
 Problem: pakiet ghostscript-9.54.0-5.el9.x86_64 z instalacji wymaga
  libijs.so.0()(64bit), którego nie można wybrać.
```

Pakiet z EPEL zależy od biblioteki, która żyje w repozytorium CRB (RHEL 9) lub PowerTools (Rocky/RHEL 8), a to repozytorium nie jest włączone. Naprawa: `dnf config-manager --set-enabled crb` (Rocky 9) lub `--set-enabled powertools` (Rocky 8), albo odpowiedni `subscription-manager repos --enable codeready-builder-for-rhel-9-$(arch)-rpms` na RHEL, zanim ponowisz instalację.

Import klucza pominięty, a `-y` w komendzie ukrywa problem cicho.

```
dnf install -y ./pakiet-wewnetrzny.rpm
```

```
Błąd:
 GPG check FAILED
```

Pakiet nie ma zaufanego klucza w lokalnej bazie, a `-y` nie potrafi automatycznie potwierdzić importu nieznanego klucza — instalacja po prostu się nie wykona. Naprawa: `rpm --import /sciezka/do/klucza.pub` przed instalacją, albo, jeśli pakiet naprawdę nigdy nie był podpisany, świadome i jednorazowe `--nogpgcheck`.

EPEL zostawiony trwale włączony podmienia pakiet z oficjalnego repozytorium.

```
$ dnf list installed | grep screen
screen.x86_64    4.9.1-4.el9    @epel
```

Pakiet, który powinien pochodzić z `baseos` albo `appstream`, ląduje z `@epel`, bo repozytorium EPEL zostało trwale włączone bez ustawionych priorytetów, a `dnf update` w pewnym momencie wybrał nowszą wersję z EPEL. Zwykle nieszkodliwe, ale zaskakujące przy audycie i utrudniające odtworzenie „czystego" stanu RHEL. Naprawa: trzymaj EPEL wyłączony na stałe i włączaj go tylko `--enablerepo=epel` per instalacja, albo skonfiguruj priorytety tak, żeby `baseos`/`appstream` zawsze wygrywały.

Ręczna instalacja z linku, który wygasł albo wskazuje starą wersję.

```
rpm -ivh https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
```

```
error: %pre(epel-release-8-19.el8.noarch) scriptlet failed
```

Skrypt instalacyjny `epel-release` dla wersji 8 odmawia instalacji na systemie w wersji 9 (albo odwrotnie) — numer wersji w adresie URL (`epel-release-latest-9` vs `-8`) musi zgadzać się z główną wersją systemu docelowego. Naprawa: sprawdź `cat /etc/os-release` albo `rpm -E %rhel` przed pobraniem linku i użyj adresu z właściwym numerem.

□□ Ćwiczenia

> Wszystko poniżej wykonujesz na własnej maszynie, w katalogu `/tmp`. Ćwiczenia, które wymagałyby instalowania pakietów na produkcji, są zastąpione odpowiednikiem z `repoquery` i `--assumeno`.

Ćwiczenie 1 — sprawdzenie repozytoriów bez zmiany systemu

```
cd /tmp
dnf repolist --all > repo-przed.txt
cat repo-przed.txt
rpm -qa gpg-pubkey* > klucze-przed.txt
cat klucze-przed.txt
```

Ile repozytoriów masz włączonych, a ile wyłączonych? Ile kluczy GPG jest już zaimportowanych?

Odpowiedź

Liczba zależy od konkretnego systemu — na świeżym Rocky Linux 9 typowo widać `appstream`, `baseos`, `extras` jako `enabled` oraz `crb` jako `disabled`. Liczba kluczy `gpg-pubkey` odpowiada liczbie różnych dostawców pakietów, którym system już ufa — zwykle co najmniej klucz Rocky/RHEL i ewentualnie klucz EPEL, jeśli był kiedyś importowany.

Ćwiczenie 2 — suchy przebieg instalacji z EPEL

```
cd /tmp
dnf repoquery --repo=epel --available 2>/dev/null | grep -i '^jq-' || echo "EPEL niedostępny lub niewłączony w tym systemie"
dnf --assumeno --enablerepo=epel install jq
```

Co pokazuje `dnf --assumeno`, zanim cokolwiek zainstaluje?

Odpowiedź

`--assumeno` zatrzymuje się na etapie pokazania planu transakcji (jaki pakiet, w jakiej wersji, z jakiego repozytorium, jaki rozmiar pobierania) i automatycznie odpowiada „nie" na pytanie o potwierdzenie — dzięki temu widzisz dokładnie, co by się stało, bez ryzyka przypadkowej instalacji. Jeśli repozytorium `epel` nie jest w ogóle skonfigurowane w systemie, `dnf` zgłosi błąd braku repozytorium zamiast planu transakcji.

Pytania kontrolne

1. Zainstalowałeś `epel-release`, ale próba `dnf install htop` kończy się błędem o brakującej bibliotece z pakietu, którego nazwa sugeruje repozytorium `crb`. Co zrobiłeś nie tak?

Odpowiedź

Najpewniej pominięto włączenie repozytorium CRB (RHEL 9) lub PowerTools (Rocky/RHEL 8), które jest wymagane przez część pakietów z EPEL. Naprawa: `dnf config-manager --set-enabled crb` (albo `powertools`), a potem ponowna próba instalacji.

2. Dlaczego administratorzy często trzymają EPEL wyłączony na stałe i włączają go tylko przez `--enablerepo=epel`?

Odpowiedź

Żeby zwykłe `dnf update` nigdy nie podmieniło pakietu z oficjalnego, wspieranego komercyjnie repozytorium na wersję z EPEL, gdyby nazwy się pokryły — a jednocześnie zachować możliwość skorzystania z bogactwa pakietów EPEL, kiedy naprawdę są potrzebne, bez trwałej zmiany konfiguracji systemu.

3. Co robi `module_hotfixes=1` i kiedy jest potrzebne?

Odpowiedź

Wyłącza dla danego repozytorium filtr modularności, który normalnie ukrywa pakiety spoza modułu, jeśli pakiet o tej samej nazwie jest już dostarczany przez moduł w repozytoriach bazowych (`appstream`). Jest potrzebne, gdy repozytorium takie jak EPEL (niemodularne) publikuje pakiet w tej samej nazwie co pakiet modularny — bez tego ustawienia `dnf` odrzuci pakiet z EPEL jako zasłonięty przez moduł, mimo że jawnie o niego poprosisz.

4. Masz plik `epel-release-latest-9.noarch.rpm` pobrany wcześniej na maszynie z dostępem do internetu. Jak zainstalujesz go na serwerze bez dostępu do sieci zewnętrznej?

Odpowiedź

Przenosisz plik na docelowy serwer (np. `scp`) i instalujesz lokalnie: `rpm -ivh /sciezka/do/epel-release-latest-9.noarch.rpm`. Ponieważ `epel-release` praktycznie nie ma własnych zależności, `rpm` poradzi sobie bez kontaktu z jakimkolwiek repozytorium — po tej instalacji plik `.repo` i klucz GPG EPEL są już na miejscu, choć samo repozytorium nadal wymaga dostępu do sieci, żeby cokolwiek z niego zainstalować.

Powiązane: dnf: instalacja, aktualizacja, historia, wycofywanie zmian · Kernele: lista, usuwanie starych, brak miejsca na /boot · Diagnostyka konfliktów i dzielenie aktualizacji na części

Źródła: `man 5 dnf.conf` · `man 8 dnf-config-manager` · dokumentacja EPEL (Fedora Project) · materiały wewnętrzne — procedura instalacji EPEL na RHEL/Rocky

---

Revision #15
Created 2026-09-18 06:56:53 UTC by Tomasz Konieczny
Updated 2026-10-03 19:09:31 UTC by Tomasz Konieczny