zypper: odpowiedniki na SLES i openSUSE

Poziom: podstawy **Dotyczy: SLES/openSUSE** **Czas: ~20 min**

`zypper` to menedżer pakietów SUSE — nadbudówka nad tym samym RPM-em co na RHEL, ale z własną składnią, własnym pojęciem „łatki” i jedną komendą, której nie ma odpowiednika w `dnf`: `zypper ps` pokazuje, które uruchomione procesy wciąż trzymają biblioteki usunięte podczas ostatniej aktualizacji.

Skróty użyte na tej stronie (11)

Skrót	Co znaczy
DNS	Domain Name System: książka telefoniczna internetu, zamienia nazwy na adresy IP
GPG	GNU Privacy Guard: program do podpisywania i szyfrowania plików oraz sprawdzania podpisów paczek
GRUB	Grand Unified Bootloader: program, który jako pierwszy startuje po włączeniu komputera i podaje jądru, co ma uruchomić
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
SAP	Systeme, Anwendungen und Produkte in der Datenverarbeitung: niemiecki producent oprogramowania klasy ERP, właściciel silnika HANA i przejętego wcześniej Sybase ASE
SLES	SUSE Linux Enterprise Server: płatna dystrybucja Linuksa firmy SUSE, konkurent Rocky/RHEL
SMT	Simultaneous Multithreading: wielowątkowość, w której rdzeń obsługuje dwie kolejki instrukcji naraz (u Intela Hyper-Threading)
SP	Special Publication: oznaczenie serii dokumentów NIST, tu wytycznych o hasłach SP 800-63B
SUSE	nazwa niemieckiej firmy i jej dystrybucji Linuksa, od niemieckiego skrótu Software und System Entwicklung
URL	Uniform Resource Locator: adres strony lub zasobu

Fiszka

TYLKO ODCZYT

Wyświetlenie listy repozytoriów: `zypper lr`

```
sudo zypper lr
```

Pokazujemy wszystkie skonfigurowane repozytoria z numerem, aliasem, nazwą i stanem (włączone, sprawdzanie GPG, odświeżanie automatyczne). Pierwsza komenda do uruchomienia, gdy `zypper refresh` albo instalacja zachowuje się nieoczekiwanie.

Przykład:

```
#  | Alias                                | Name                                 |
Enabled | GPG Check | Refresh
---+--------------------------------------+--------------------------------------
+---------+-----------+--------
1  | SLE-Module-Basesystem15-SP5-Pool     | SLE-Module-Basesystem15-SP5-Pool     |
Yes     | (r ) Yes  | Yes
2  | SLE-Module-Basesystem15-SP5-Updates  | SLE-Module-Basesystem15-SP5-Updates  |
Yes     | (r ) Yes  | Yes
3  | SLE-Product-SLES15-SP5-Pool          | SLE-Product-SLES15-SP5-Pool          |
Yes     | (r ) Yes  | Yes
4  | SLE-Product-SLES15-SP5-Updates       | SLE-Product-SLES15-SP5-Updates       |
Yes     | (r ) Yes  | Yes
```

GPG = GNU Privacy Guard: program do podpisywania i szyfrowania plików oraz sprawdzania podpisów paczek · **zypper** = menedżer pakietów dystrybucji SUSE i openSUSE

TYLKO ODCZYT

Wyświetlenie repozytoriów razem z adresami URL: `zypper lr -u`

```
sudo zypper lr -u
```

To samo co wyżej, plus kolumna z URL każdego repozytorium — przydatne, gdy trzeba sprawdzić, czy repozytorium wskazuje na właściwy serwer SCC/RMT.

Przykład:

```
#  | Alias                          | Name                           | Enabled | GPG
Check | Refresh | URI
```

```
---+--------------------------------+--------------------------------+---------
+----------+---------+-----------------------------------------
1  | SLE-Module-Basesystem15-SP5-Pool | SLE-Module-Basesystem15-SP5-Pool | Yes   | (r )
Yes  | Yes     | https://updates.suse.zielona.example/SUSE/Products/SLE-Module-
Basesystem/15-SP5/x86_64/product/
```

URL = Uniform Resource Locator: adres strony lub zasobu · **zypper** = menedżer pakietów dystrybucji SUSE i openSUSE

TYLKO ODCZYT

Odświeżenie metadanych repozytoriów: `zypper ref`

```
sudo zypper ref
```

Odpowiednik `apt update` / `dnf check-update` — ściągamy aktualne metadane ze wszystkich włączonych repozytoriów, niczego nie instalując. Skrót od `zypper refresh`.

Przykład:

```
Repository 'SLE-Module-Basesystem15-SP5-Updates' is up to date.
Retrieving repository 'SLE-Product-SLES15-SP5-Updates' metadata
...................[done]
Building repository 'SLE-Product-SLES15-SP5-Updates' cache
.......................[done]
All repositories have been refreshed.
```

zypper = menedżer pakietów dystrybucji SUSE i openSUSE

TYLKO ODCZYT

Aktualizacja pakietów bez zmiany wersji dystrybucji: `zypper up`

```
sudo zypper up
```

Podnosimy zainstalowane pakiety do najnowszych wersji dostępnych w **aktualnie zarejestrowanej** wersji produktu — czyli zwykłe patchowanie w obrębie tego samego SP (service pack). Skrót od `zypper update`.

Przykład:

```
The following 14 packages are going to be upgraded:
  kernel-default openssl-1_1 sudo systemd ...
14 packages to upgrade.
Overall download size: 89.3 MiB. Already cached: 0 B.
After the operation, additional 3.1 MiB will be used.
Continue? [y/n/v...? shows all options] (y): y
```

□□ **SP** = Special Publication: oznaczenie serii dokumentów NIST, tu wytycznych o hasłach SP 800-63B · **zypper** = menedżer pakietów dystrybucji SUSE i openSUSE

□□ **TYLKO ODCZYT**

Aktualizacja dystrybucji: `zypper dup`

```
sudo zypper dup
```

Nie jest to „mocniejsza wersja” `zypper up` — to inna operacja. `dup` dopasowuje zainstalowane pakiety **dokładnie** do tego, co jest w aktualnie włączonych repozytoriach, ignorując wersje zainstalowane. Jeśli repozytoria wskazują już na nowszy SP albo inną wersję produktu, `dup` zainstaluje tę nową wersję pakietów, nawet jeśli oznacza to usunięcie starych i podmianę zależności. Na SLES-ie używa się `dup` niemal wyłącznie przy migracji między wersjami (np. SP4 → SP5), nie do rutynowego patchowania — do tego służy `zypper up` albo `zypper patch`.

Przykład:

```
Computing distribution upgrade...
The following 6 packages are going to be upgraded:
  kernel-default openssl-1_1 systemd systemd-sysvinit ...
The following NEW package is going to be installed:
  suseconnect-ng
The following package is going to be REMOVED:
  yast2-migration-sle
```

□□ **SP** = Special Publication: oznaczenie serii dokumentów NIST, tu wytycznych o hasłach SP 800-63B · **zypper** = menedżer pakietów dystrybucji SUSE i openSUSE

□□ **TYLKO ODCZYT**

Instalacja pakietu: `zypper in`

```
sudo zypper in NazwaPakietu
```

Instalujemy pakiet z zależnościami. `in` to oficjalny skrót `zypper install`, akceptowany na równi z pełną nazwą.

Przykład:

```
Resolving package dependencies...
The following NEW package is going to be installed:
  nginx
Overall download size: 620.1 KiB.
(1/1) nginx-1.25.3-150500.1.9.x86_64.rpm ..............[done]
```

zypper = menedżer pakietów dystrybucji SUSE i openSUSE

OSTROŻNIE · NIEODWRACALNE

Usunięcie pakietu: `zypper rm`

```
sudo zypper rm NazwaPakietu
```

Usuwamy pakiet. Skrót `zypper remove`. Jeśli usuwany pakiet jest wymagany przez inny — `zypper` zapyta, czy usunąć kaskadowo, zamiast robić to po cichu.

Przykład:

```
The following package is going to be REMOVED:
  nginx
1 package to remove.
```

zypper = menedżer pakietów dystrybucji SUSE i openSUSE

TYLKO ODCZYT

Wyszukiwanie pakietu: `zypper se`

```
zypper se NazwaPakietu
```

Przeszukujemy nazwy i podsumowania pakietów we wszystkich włączonych repozytoriach. Nie wymaga uprawnień administratora.

Przykład:

```
S  | Name           | Summary                      | Type
---+----------------+------------------------------+--------
i  | postgresql16   | PostgreSQL 16 client         | package
   | postgresql16-server | PostgreSQL 16 database server | package
```

zypper = menedżer pakietów dystrybucji SUSE i openSUSE

TYLKO ODCZYT

Wyświetlenie informacji o pakiecie: `zypper if`

```
zypper if NazwaPakietu
```

Skrót od `zypper info` — wersja, rozmiar, repozytorium źródłowe, zależności i opis pakietu.

Przykład:

```
Information for package postgresql16-server:
---------------------------------------------
Repository     : SLE-Module-Server-Applications15-SP5-Pool
Name           : postgresql16-server
Version        : 16.4-150500.3.9.1
Installed      : No
Status         : not installed
Summary        : The PostgreSQL Server
```

zypper = menedżer pakietów dystrybucji SUSE i openSUSE

TYLKO ODCZYT

Wyświetlenie listy pakietów: `zypper pa`

```
zypper pa --installed-only | head -5
```

Skrót od `zypper packages`. Bez flag pokazuje wszystkie dostępne pakiety ze wszystkich repozytoriów; `--installed-only` zawęża do tego, co faktycznie jest w systemie.

Przykład:

```
S | Repository | Name          | Version              | Arch
--+------------+---------------+----------------------+-------
i | @System    | bash          | 5.2.15-150500.4.6.1  | x86_64
i | @System    | glibc         | 2.31-150300.61.1     | x86_64
```

zypper = menedżer pakietów dystrybucji SUSE i openSUSE

TYLKO ODCZYT

Sprawdzenie procesów trzymających usunięte biblioteki: `zypper ps`

```
sudo zypper ps
```

Unikalna komenda tej rodziny narzędzi — `dnf` nie ma bezpośredniego odpowiednika. Po aktualizacji biblioteki (np. `libssl`, `glibc`) stary plik `.so` fizycznie znika z dysku, ale proces, który go już wczytał do pamięci przed aktualizacją, dalej go trzyma — jądro nie zwalnia i-node dopóki ostatni proces go używa. `zypper ps` przegląda `/proc/*/maps` wszystkich procesów i pokazuje, które z nich mapują pliki, których już nie ma w bieżącej wersji na dysku. To jest dokładnie lista usług, które trzeba zrestartować, żeby faktycznie zaczęły korzystać z nowej, załatanej wersji biblioteki — bez restartu proces działa dalej ze starym, dziurawym kodem w pamięci, mimo że dysk pokazuje już nową wersję.

Przykład:

```
The following running processes use deleted files:


PID   | Command | Service       | Files
------+---------+---------------+----------------------------------
1842  | nginx   | nginx.service | /usr/lib64/libssl.so.1.1 (deleted)
2011  | sshd    | sshd.service  | /usr/lib64/libcrypto.so.1.1 (deleted)
```

glibc = GNU C Library: podstawowa biblioteka systemowa Linuksa, od której wersji zależy, czy dany program (na przykład serwer P4) w ogóle się uruchomi · **zypper** = menedżer pakietów dystrybucji SUSE i openSUSE

TYLKO ODCZYT

Instalacja łatek: `zypper patch`

```
sudo zypper patch
```

Instalujemy wyłącznie oficjalne łatki SUSE (patrz *Pojęcie: łatka* niżej) — czyli aktualizacje bezpieczeństwa i poprawki błędów opublikowane przez SUSE jako osobne, śledzone jednostki. To jest komenda zalecana na produkcyjnym SLES-ie do rutynowego patchowania, bardziej precyzyjna niż `zypper up`, bo instaluje dokładnie to, co jest opisane w konkretnym biuletynie bezpieczeństwa.

Przykład:

```
Loading repository data...
Reading installed packages...
5 patches needed (3 security patches)
Retrieving patch SUSE-SLE-Module-Basesystem-15-SP5-2026-1421 ...[done]
Applying patch SUSE-SLE-Module-Basesystem-15-SP5-2026-1421 ...
```

SUSE = nazwa niemieckiej firmy i jej dystrybucji Linuksa, od niemieckiego skrótu Software und System Entwicklung · **zypper** = menedżer pakietów dystrybucji SUSE i openSUSE

TYLKO ODCZYT

Wyświetlenie listy dostępnych łatek: `zypper lp`

```
zypper lp
```

Skrót od `zypper list-patches` — pokazuje każdą dostępną łatkę osobno, z kategorią i poziomem istotności, zamiast zbiorczej liczby jak `patch-check`.

Przykład:

```
Name                                          | Category | Severity  | Status | Summary
----------------------------------------------+----------+-----------+--------
+--------------------------------
SUSE-SLE-Module-Basesystem-15-SP5-2026-1421    | security | important | needed | Security
update for openssl-1_1
```

zypper = menedżer pakietów dystrybucji SUSE i openSUSE

TYLKO ODCZYT

Sprawdzenie liczby dostępnych łatek: `zypper patch-check`

```
zypper patch-check
```

Szybkie sprawdzenie „ile łatek czeka” bez pobierania ich treści — dobre do skryptów monitorujących, które mają tylko zgłosić alarm, jeśli liczba przekroczy próg.

Przykład:

```
Loading repository data...
Reading installed packages...
5 patches needed (3 security patches)
```

zypper = menedżer pakietów dystrybucji SUSE i openSUSE

TYLKO ODCZYT

Dodanie repozytorium: `zypper ar`

```
sudo zypper ar https://updates.suse.zielona.example/repo/Module-Basesystem
NazwaRepozytorium
```

Skrót od `zypper addrepo`. Pierwszy argument to URL, drugi to alias, pod którym repozytorium będzie widoczne w `zypper lr`.

Przykład:

```
Adding repository 'SLE-Module-Basesystem' [done]
Repository 'SLE-Module-Basesystem' successfully added
```

URL = Uniform Resource Locator: adres strony lub zasobu · **zypper** = menedżer pakietów dystrybucji SUSE i openSUSE

TYLKO ODCZYT

Włączenie i wyłączenie repozytorium: `zypper mr -e` / `-d`

```
sudo zypper mr -d NazwaRepozytorium
sudo zypper mr -e NazwaRepozytorium
```

Skrót od `zypper modifyrepo`. `-d` (disable) wyłącza repozytorium bez usuwania jego konfiguracji, `-e` (enable) włącza je z powrotem — wygodniejsze niż `rr`/`ar`, gdy chcemy tylko czasowo wyłączyć jedno repozytorium.

Przykład:

```
Repository 'SLE-Module-Basesystem' has been successfully disabled.
```

zypper = menedżer pakietów dystrybucji SUSE i openSUSE

TYLKO ODCZYT

Usunięcie repozytorium: `zypper rr`

```
sudo zypper rr NazwaRepozytorium
```

Skrót od `zypper removerepo` — trwałe usunięcie wpisu repozytorium, w przeciwieństwie do tymczasowego wyłączenia przez `mr -d`.

Przykład:

```
Removing repository 'SLE-Module-Basesystem' ...................[done]
Repository 'SLE-Module-Basesystem' has been removed.
```

zypper = menedżer pakietów dystrybucji SUSE i openSUSE

TYLKO ODCZYT

Zablokowanie pakietu: `zypper al`

```
sudo zypper al NazwaPakietu
```

Skrót od `zypper addlock`. Odpowiednik `apt-mark hold` / `dnf versionlock add` — blokuje pakiet na aktualnej wersji, `zypper up` / `zypper dup` / `zypper patch` go pomijają.

Przykład:

```
Specified lock has been successfully added.
```

zypper = menedżer pakietów dystrybucji SUSE i openSUSE

TYLKO ODCZYT

Wyświetlenie listy blokad: `zypper ll`

```
zypper ll
```

Skrót od `zypper locks` — pokazuje, które pakiety są aktualnie zablokowane.

Przykład:

```
#  | Name           | Type
---+----------------+--------
1  | kernel-default | package
```

zypper = menedżer pakietów dystrybucji SUSE i openSUSE

TYLKO ODCZYT

Usunięcie blokady: `zypper rl`

```
sudo zypper rl NazwaPakietu
```

Skrót od `zypper removelock` — zdejmuje blokadę nałożoną przez `al`.

Przykład:

```
1 lock has been successfully removed.
```

zypper = menedżer pakietów dystrybucji SUSE i openSUSE

TYLKO ODCZYT

Sprawdzenie statusu rejestracji: `SUSEConnect -s`

```
SUSEConnect -s
```

Sprawdzamy, czy system jest zarejestrowany w SCC (SUSE Customer Center) albo lokalnym RMT, jakie moduły są aktywowane i do kiedy ważna jest subskrypcja każdego z nich. Pierwsza komenda do sprawdzenia, gdy `zypper ref` nagle zaczyna zwracać błędy autoryzacji na repozytoriach SUSE.

Przykład:

```
[{"identifier":"SLES","version":"15.5","status":"Registered","subscription_status":"ACTI
VE","expires_at":"2027-01-01 00:00:00 UTC"}]
```

SUSE = nazwa niemieckiej firmy i jej dystrybucji Linuksa, od niemieckiego skrótu Software und System Entwicklung

TYLKO ODCZYT

Rejestracja produktu: `SUSEConnect -r KOD -e EMAIL`

```
sudo SUSEConnect -r XXXXXXXXXXXXXXXX -e it@zielona.example
```

Rejestrujemy system w SCC (albo w RMT, jeśli skonfigurowany jako serwer rejestracji) podanym kodem rejestracyjnym i adresem e-mail administratora. Bez ważnej rejestracji repozytoria aktualizacji SUSE (`*-Updates`) są niedostępne — widać wtedy tylko repozytoria instalacyjne (`*-Pool`).

Przykład:

```
Registering system to SUSE Customer Center


Announcing system to https://scc.suse.com ...
Activating SLES 15.5 x86_64 ...
-> Adding service to repositories ...
Successfully registered system
```

SUSE = nazwa niemieckiej firmy i jej dystrybucji Linuksa, od niemieckiego skrótu Software und System Entwicklung

TYLKO ODCZYT

Wyświetlenie dostępnych modułów i rozszerzeń: `SUSEConnect --list-extensions`

```
SUSEConnect --list-extensions
```

> **Pojęcie: moduł SLES.** Moduł to opcjonalny, oddzielnie aktywowany pakiet repozytoriów wewnątrz jednej licencji SLES — na przykład „Basesystem”, „Server Applications”, „Containers”, „Development Tools”. W przeciwieństwie do RHEL-a, gdzie większość oprogramowania leży w jednym dużym repozytorium BaseOS/AppStream, SLES dzieli zawartość na moduły, z których część jest aktywna od razu po rejestracji, a część trzeba dołączyć osobną komendą. `--list-extensions` pokazuje, co jest dostępne w ramach aktualnej rejestracji i co jeszcze wymaga aktywacji.

Przykład:

```
AVAILABLE EXTENSIONS AND MODULES

    Containers Module 15 SP5 x86_64
    Activate with: SUSEConnect -p sle-module-containers/15.5/x86_64

    Development Tools Module 15 SP5 x86_64
    Activate with: SUSEConnect -p sle-module-desktop-applications/15.5/x86_64
```

SLES = SUSE Linux Enterprise Server: płatna dystrybucja Linuksa firmy SUSE, konkurent Rocky/RHEL

TYLKO ODCZYT

Usunięcie repozytoriów po wyrejestrowaniu: `SUSEConnect --cleanup`

```
sudo SUSEConnect --cleanup
```

Porządkujemy nim system po `SUSEConnect --de-register` (albo po zmianie kodu rejestracyjnego) — usuwamy repozytoria i usługi, które zostały po starej rejestracji i inaczej zostałyby osierocone w konfiguracji `zypper`.

Przykład:

```
Removing service Basesystem_Module_15_SP5_x86_64 ...
Removing service SLES_15_SP5_x86_64 ...
Cleanup finished.
```

ZMIENIA SYSTEM

Aktualizacje na SLE Micro: `transactional-update`

```
sudo transactional-update
```

Na SLE Micro i innych wariantach z systemem plików tylko-do-odczytu (`/usr` w Btrfs, snapshoty pod korzeniem) zwykły `zypper` nie modyfikuje systemu na żywo. `transactional-update` tworzy nowy snapshot Btrfs, instaluje w nim wszystkie zaległe aktualizacje, a zmianę aktywuje dopiero po restarcie — jeśli coś pójdzie nie tak, wraca się do poprzedniego snapshotu przez GRUB, zamiast ręcznie naprawiać system.

Przykład:

```
Checking for newer version of transactional-update ...
Executing 'zypper --non-interactive patch --with-interactive -l'
Creating new default snapshot: 47
Snapshot 47 created.
transactional-update finished
Please reboot your machine to activate the changes and avoid data loss.
```

Uważaj

- `zypper dup` na SLES-ie sięgasz po migrację wersji, nie rutynowe patchowanie — do tego jest `zypper up` / `zypper patch`.
- Zablokowany pakiet (`zypper al`) jest pomijany przez `up`, `dup` i `patch` naraz — sprawdź `zypper ll`, jeśli aktualizacja pomija coś bez wyjaśnienia.
- `zypper ps` warto uruchamiać po **każdej** aktualizacji obejmującej biblioteki systemowe, nie tylko po aktualizacji jądra.

GRUB = Grand Unified Bootloader: program, który jako pierwszy startuje po włączeniu komputera i podaje jądru, co ma uruchomić

Z życia wzięte

Z ŻYCIA WZIĘTE

Poniedziałek. Pilotaż SAP na hana01 dostaje aktualizację openssl, a pan Zbigniew pyta, czy wszystko na pewno działa poprawnie. Tomek uruchamia zypper ps i widzi, że proces bazy wciąż trzyma w pamięci starą, usuniętą już z dysku wersję biblioteki, więc dopiero restart usługi kończy sprawę naprawdę.

Po co to istnieje

Pojęcie: wzorzec (pattern). Wzorzec to nazwany zestaw pakietów instalowanych razem jako jedna logiczna całość — na przykład wzorzec `Minimal` czy `Server-Applications` dociąga dziesiątki pakietów jedną komendą (`zypper in -t pattern Server-Applications`), tak jak grupy pakietów (`@grupa`) w `dnf group install` po stronie RHEL-a. Różnica jest w tym, że na SUSE wzorce są używane szerzej — instalator openSUSE/SLES opiera dobór oprogramowania właśnie na wzorcach, nie na pojedynczych zaznaczeniach.

`zypper` powstał dla SUSE Linux z tego samego powodu, dla którego Debian dostał `apt`, a Red Hat `yum`: sam RPM (który SUSE dzieli z RHEL-em) rozwiązuje instalację pojedynczej paczki, ale nie zna internetu ani grafu zależności między setkami repozytoriów. `zypper` opiera się na bibliotece `libzypp` i silniku SAT-solver (satysfakcjonowanie ograniczeń logicznych, ten sam rodzaj algorytmu co w `dnf`), który liczy najlepszy zestaw pakietów spełniający żądanie, priorytety repozytoriów i blokady.

Kluczowa różnica wobec świata RHEL-a to obecność **łatek** jako osobnego bytu. **Pojęcie: łatka (patch).** Łatka to metadana SUSE opisująca konkretną poprawkę — jeden numer, jedna data, jedna lista pakietów, które trzeba podnieść, żeby ta konkretna poprawka bezpieczeństwa albo błędu została zastosowana. Zamiast myśleć „zaktualizuj pakiet X do wersji Y”, administrator SLES-a może myśleć „zainstaluj łatkę SUSE-SLE-Module-Basesystem-15-SP5-2026-1421”, co jest bezpośrednim odniesieniem do biuletynu bezpieczeństwa SUSE. RHEL ma podobny mechanizm (errata), ale nie jest on tak mocno wbudowany w codzienne polecenia `dnf` jak `zypper patch` w `zypper`.

Analogia

`zypper` i `dnf` to **dwa różne panele sterowania tym samym silnikiem**.

Pod maską — silnikiem — jest RPM: ten sam format pakietu, ta sama baza `/var/lib/rpm`, te same narzędzia niskopoziomowe (`rpm -q`, `rpm -V`). Ale kokpit, którym kierujesz tym silnikiem, różni się między producentami samochodów. `dnf` ma swój układ przycisków (`dnf upgrade`, `dnf autoremove`), `zypper` ma inny (`zypper up`, `zypper packages --unneeded`) — te same funkcje, inne nazwy, inne skróty klawiszowe.

Jeden przycisk w kokpicie SUSE nie ma odpowiednika u konkurencji: `zypper ps` to kontrolka na desce rozdzielczej, która zapala się, gdy silnik (biblioteki systemowe) został wymieniony, ale skrzynia biegów (uruchomione procesy) wciąż pracuje na starej przekładni. `dnf` takiej kontrolki nie ma wprost — trzeba by ręcznie zaglądać pod maskę (`lsof +c0 | grep DEL` albo `needs-restarting` z dodatkowego pakietu `dnf-utils`), żeby zobaczyć to samo.

Jak to działa

Rozwiń: jak to działa pod spodem

> **Pojęcie: usługa rejestracji SCC/RMT.** SUSE Customer Center (SCC) to chmurowa usługa SUSE, która autoryzuje dostęp do repozytoriów aktualizacji na podstawie ważnej

> subskrypcji — bez zarejestrowania systemu (`SUSEConnect -r`) repozytoria `*-Pool` (instalacyjne, z nośnika) są dostępne, ale `*-Updates` (poprawki bezpieczeństwa i błędów) już nie. RMT (Repository Mirroring Tool, następca dawnego SMT) to lokalny serwer proxy dla organizacji, które nie chcą, żeby każdy serwer łączył się bezpośrednio z internetem po aktualizacje — RMT rejestruje się raz w SCC, ściąga repozytoria lokalnie, a serwery wewnątrz sieci rejestrują się do RMT zamiast do SCC.

Priorytety repozytoriów działają w `zypper` inaczej niż mogłoby się wydawać z liczby przy każdym wpisie: repozytoria o **tym samym** priorytecie dzielą się nim równo (żadne nie wygrywa automatycznie), a wygrywa repozytorium z **niższą** liczbą priorytetu — analogicznie do `nice` w planowaniu procesów, gdzie mniejsza liczba znaczy wyższe pierwszeństwo. Domyślny priorytet to 99; jeśli chcesz, żeby jedno repozytorium zawsze wygrywało przy konflikcie wersji tego samego pakietu, obniżasz jego priorytet poniżej pozostałych.

`zypper up` i `zypper dup` różnią się dokładnie tym, względem czego liczą różnicę. `up` bierze zainstalowaną wersję pakietu i szuka nowszej w tym samym repozytorium/module, z zachowaniem ograniczenia „nie usuwaj, nie podmieniaj architektury, nie zmieniaj vendora pakietu” — bardzo blisko `apt upgrade`. `dup` olewa to, co jest zainstalowane, i synchronizuje cały system z tym, co *aktualnie* oferują włączone repozytoria — jeśli w międzyczasie podmieniłeś repozytoria na nowszy SP, `dup` zainstaluje dokładnie ten SP, nawet kosztem usunięcia pakietów, które w nowej wersji już nie istnieją. Stąd twarda zasada: `dup` na produkcyjnym SLES-ie włącza się świadomie, przy zaplanowanej migracji, nie w cotygodniowym cronie patchującym.

Krok po kroku

1. Sprawdzenie stanu rejestracji przed jakąkolwiek aktualizacją.

```
SUSEConnect -s
```

```
"status": "Registered",
"subscription_status": "ACTIVE"
```

Jeśli `subscription_status` nie jest `ACTIVE` (np. `EXPIRED`), `zypper ref` na repozytoriach `*-Updates` zwróci błąd autoryzacji, zanim jeszcze dojdziesz do właściwego patchowania. Naprawa zwykle sprowadza się do ponownej rejestracji nowym kodem:

```
sudo SUSEConnect --cleanup
sudo SUSEConnect -r XXXXXXXXXXXXXXXX -e it@zielona.example
```

2. Rutynowe patchowanie na produkcji.

```
sudo zypper ref
zypper lp
sudo zypper patch
```

Odświeżenie metadanych, podgląd, co konkretnie zostanie zainstalowane (z numerem biuletynu i poziomem istotności), i dopiero instalacja. To jest odpowiednik `dnf upgrade --security` po stronie RHEL-a, tylko z dokładniejszą granulacją.

3. `zypper ps` po każdej aktualizacji obejmującej biblioteki.

```
sudo zypper patch
sudo zypper ps
```

```
PID   | Command | Service       | Files
------+---------+---------------+---------------------------------
1842  | nginx   | nginx.service | /usr/lib64/libssl.so.1.1 (deleted)
```

Kolumna `Service` mówi wprost, jaką jednostkę systemd zrestartować, żeby proces przeładował nową bibliotekę:

```
sudo systemctl restart nginx.service
sudo zypper ps
```

```
No processes using deleted files found.
```

Pusty wynik po restarcie potwierdza, że nic w systemie nie trzyma już starych, usuniętych bibliotek w pamięci.

4. Blokada pakietu na czas testów kompatybilności.

```
sudo zypper al kernel-default
zypper ll
sudo zypper up
```

```
Not updating kernel-default (locked)
```

5. Migracja do nowego service packa — jedyny uzasadniony przypadek na `dup`.

```
sudo zypper lr
# repozytoria przełączone na nowy SP (ręcznie albo przez `zypper migration`)
```

```
sudo zypper ref
sudo zypper dup
```

Przed tym krokiem repozytoria muszą już wskazywać na docelowy SP — `dup` synchronizuje system z tym, co *aktualnie* widzi w repozytoriach, więc kolejność (najpierw przełączenie repozytoriów, potem `dup`) ma znaczenie.

Częste błędy

Rutynowe użycie `zypper dup` zamiast `zypper up`.

```
The following 4 packages are going to be REMOVED:
  yast2-network yast2-nfs-client ...
```

Ktoś przyzwyczajony do myślenia „`dup` to `up`, tylko mocniejsze” odpala je w zwykłym oknie patchowania i dostaje listę pakietów do usunięcia, bo repozytoria (przypadkiem albo przez wcześniejszą, niedokończoną zmianę) wskazywały już na inny SP. Naprawa: na rutynowe patchowanie zawsze `zypper up` albo `zypper patch`, `dup` tylko przy świadomej migracji wersji, po sprawdzeniu `zypper lr -u`.

Pominięcie `zypper ps` po aktualizacji bibliotek.

Aktualizacja `openssl` przechodzi bez błędu, restart maszyny nie jest zaplanowany w oknie serwisowym, a usługa `nginx` cicho działa dalej na starej, podatnej wersji `libssl` załadowanej do pamięci przed aktualizacją — mimo że `rpm -q openssl-1_1` na dysku pokazuje już nową wersję. Wykrycie i naprawa:

```
sudo zypper ps
sudo systemctl restart nginx.service
```

Repozytorium z błędem GPG po zmianie klucza podpisującego.

```
Signature verification failed for repomd.xml
Warning: Signature verification failed for file 'repomd.xml' from repository 'SLE-Module-Basesystem15-SP5-Updates'.
Warning: File 'repomd.xml' from repository 'SLE-Module-Basesystem15-SP5-Updates' is unsigned, continue? [yes/no] (no):
```

Zdarza się po zmianie klucza podpisującego repozytorium (np. po stronie RMT) albo po przeterminowanym kluczu w lokalnej bazie zaufanych kluczy. Odpowiadanie „yes” za każdym razem obchodzi problem, ale nie go rozwiązuje — właściwa naprawa to ponowny import aktualnego klucza:

```
sudo rpm --import /etc/zypp/repos.d/klucz-repo.asc
sudo zypper ref
```

Rejestracja wygasła, a repozytoria `*-Updates` znikają bez wyraźnego komunikatu o przyczynie.

```
Repository 'SLE-Module-Basesystem15-SP5-Updates' is invalid.
[SLE-Module-Basesystem15-SP5-Updates|https://updates.suse.zielona.example/...] Valid metadata
not found at specified URL
```

Objaw wygląda jak problem sieciowy, ale przyczyną jest wygasła subskrypcja. Sprawdzenie zawsze zaczyna się od statusu rejestracji, nie od diagnostyki sieci:

```
SUSEConnect -s
```

□□ Ćwiczenia

> Ćwiczenia poniżej wymagają maszyny z SLES-em albo openSUSE i uprawnień administratora (`sudo`) — zarządzanie repozytoriami i pakietami systemowymi nie da się bezpiecznie odizolować w katalogu `/tmp`.

Ćwiczenie 1 — sprawdzenie, co trzyma usuniętą bibliotekę

```
sudo zypper in -y sl
sudo zypper rm -y sl
```

Zainstaluj i usuń dowolny mały pakiet (np. `sl`), a następnie uruchom `sudo zypper ps` mimo że akurat nie było żadnej aktualizacji bibliotek. Dlaczego wynik najpewniej będzie pusty, w przeciwieństwie do sytuacji po aktualizacji `openssl` czy `glibc`?

Odpowiedź

`zypper ps` szuka procesów, które mają w pamięci zamapowany plik **usunięty z dysku**, ale wciąż otwarty przez działający proces (typowe dla bibliotek współdzielonych `.so`, które proces wczytuje raz przy starcie i trzyma przez cały czas działania). Usunięcie małego, samodzielnego

pakietu jak `sl` nie dotyka żadnej biblioteki używanej przez inne, działające procesy — nie ma więc kogo wykazać na liście. Efekt pojawia się dopiero przy aktualizacji pakietów takich jak `openssl`, `glibc` czy `libssh2`, z których korzystają długo działające usługi.

Ćwiczenie 2 — blokada i jej wpływ na trzy różne komendy aktualizujące

```
sudo zypper al sl
zypper ll
sudo zypper up
sudo zypper patch
sudo zypper rl sl
```

`sl` nie jest realistycznym kandydatem do blokady (nie ma aktualizacji bezpieczeństwa), więc dla tego ćwiczenia wybierz pakiet, który akurat ma dostępną aktualizację w `zypper lp`, i powtórz sekwencję. Czy blokada nałożona przez `zypper al` obowiązuje jednakowo dla `up`, `dup` i `patch`?

Odpowiedź

Tak — blokada zapisana przez `zypper al` jest globalna względem wszystkich trzech komend aktualizujących. `up`, `dup` i `patch` sprawdzają tę samą listę blokad przed policzeniem planu operacji i pomijają zablokowany pakiet niezależnie od tego, którą z trzech komend uruchomisz.

Pytania kontrolne

1. Czym różni się `zypper up` od `zypper dup`?

Odpowiedź

`zypper up` podnosi zainstalowane pakiety do nowszych wersji w obrębie tego samego produktu/SP, nigdy nie usuwając ani nie podmieniając architektury pakietu. `zypper dup` synchronizuje cały zestaw zainstalowanych pakietów z tym, co aktualnie oferują włączone repozytoria, nawet kosztem usunięcia pakietów — używany świadomie przy migracji między wersjami/SP, nie do rutynowego patchowania.

2. Do czego służy `zypper ps` i dlaczego nie ma bezpośredniego odpowiednika w `dnf`?

Odpowiedź

`zypper ps` pokazuje procesy, które wciąż mają w pamięci zamapowane pliki bibliotek usunięte z dysku podczas aktualizacji — czyli dokładną listę usług wymagających restartu, żeby zacząć korzystać z nowej wersji. `dnf` nie ma tej komendy wbudowanej wprost; najbliższy odpowiednik to dodatkowe narzędzie `needs-restarting` z pakietu `dnf-utils` albo ręczne przeszukanie `lsof`.

3. Kiedy warto użyć `zypper dup` na SLES-ie?

Odpowiedź

Wyłącznie przy świadomej migracji wersji — na przykład przejściu z jednego service packa na kolejny, po przełączeniu repozytoriów na docelową wersję. Rutynowe patchowanie produkcyjnego serwera powinno używać `zypper up` albo `zypper patch`.

4. Co pokazuje `SUSEConnect -s` i dlaczego warto to sprawdzić przed diagnozowaniem błędów repozytoriów jako problemu sieciowego?

Odpowiedź

Pokazuje status rejestracji systemu w SCC/RMT dla każdego aktywowanego modułu, w tym czy subskrypcja jest aktywna i do kiedy ważna. Repozytoria `*-Updates` przestają odpowiadać poprawnie, gdy subskrypcja wygaśnie — objaw wygląda jak problem sieci albo DNS, ale przyczyną jest wygasła rejestracja, więc to pierwsza rzecz do sprawdzenia, zanim zacznie się diagnozować sieć.

Powiązane: dnf: instalacja, aktualizacja, historia, wycofywanie zmian · apt i dpkg: odpowiedniki na Debianie i Ubuntu · Standardowa aktualizacja systemu produkcyjnego · systemd i systemctl — jednostki, zależności, stany

Źródła: `man 8 zypper` · `man 8 SUSEConnect` · SUSE Documentation: „Package Management with Zypper" · OneNote: SLES.txt („Listowanie repozytoriów", „Rejestrowanie SLES")

Revision #15
Created 2026-09-18 06:57:00 UTC by Tomasz Konieczny
Updated 2026-10-03 19:09:38 UTC by Tomasz Konieczny