Nginx: kopia konfiguracji, certyfikatów i bezpieczna aktualizacja

Poziom: operacyjny **Dotyczy: Nginx 1.24+ z pakietu albo w Dockerze**

Czas: ~25 min

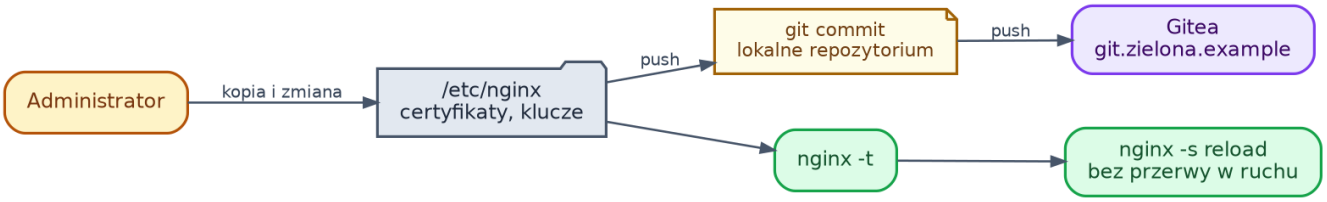
Nginx sam w sobie nie trzyma żadnych danych do odtworzenia, cała jego wartość leży w plikach konfiguracyjnych i certyfikatach. Kopiujemy je do Gitea tak samo jak każdy inny kod, a aktualizację pakietu robimy zawsze z testem konfiguracji i drogą powrotną gotową, zanim jej użyjemy.

Skróty użyte na tej stronie (2)

Skrót	Co znaczy
IT	Information Technology: dział albo praca związana z komputerami i infrastrukturą firmy
TLS	Transport Layer Security: szyfrowanie połączeń, np. w HTTPS

Zanim zaczniemy

Wymaganie	Uwaga
Dostęp do `/etc/nginx` (wariant pakietowy) albo `/docker/apps/nginx` (wariant Docker)	konto z prawem odczytu tych ścieżek na hoście
Konto w Gitea i repozytorium na konfigurację	patrz Gitea po pierwsze uruchomienie i uprawnienia
Znajomość obecnej wersji Nginksa	`nginx -v` albo `docker compose images` w wariancie kontenerowym



Kopia /etc/nginx i certyfikatów trafia przez git commit i push do Gitea, a każda zmiana kończy się nginx -t i dopiero potem nginx -s reload (kliknięcie otwiera pełny rozmiar)

Uważaj: klucze prywatne certyfikatów TLS to dane wrażliwe na równi z hasłami. Repozytorium w Gitea, do którego trafiają, musi mieć zawężony dostęp tylko dla zespołu IT, dokładnie tak jak repozytoria z `.env` innych aplikacji w Zielonej.

Przebieg

1. Co kopiujemy

> **Pojęcie: konfiguracja Nginksa jako kod.** Cały stan Nginksa, poza samym binarnym pakietem albo obrazem, mieści się w kilku plikach tekstowych: nic nie trzeba eksportować z bazy danych ani zatrzymywać usługi, żeby zrobić spójną kopię. Trzymanie tych plików w Gicie, zamiast tylko na dysku serwera, daje historię zmian i możliwość cofnięcia złej edycji jednym `git revert`, zamiast odtwarzania z kopii binarnej.

Co	Gdzie w wariancie pakietowym	Gdzie w wariancie Docker
Główny plik konfiguracyjny	`/etc/nginx/nginx.conf`	`/docker/apps/nginx/nginx.conf`
Bloki `server` poszczególnych aplikacji	`/etc/nginx/conf.d/*.conf`	`/docker/apps/nginx/conf.d/*.conf`
Certyfikaty i klucze prywatne	`/etc/nginx/ssl/`	`/docker/apps/nginx/certs/`
Plik Compose (tylko wariant Docker)	nie dotyczy	`/docker/compose/nginx/compose.yaml`

2. Pierwsze wypchnięcie konfiguracji do Gitea

```
cd /docker/apps/nginx
git init
```

```
git remote add origin https://git.zielona.example/it/nginx-docker01.git
```

Plik `.gitignore` wykluczający klucze prywatne z historii repozytorium, jeśli decydujemy się trzymać certyfikaty osobno od reszty konfiguracji:

```
certs/*.key
```

Uważaj: jeśli klucze prywatne mimo wszystko mają trafić do repozytorium, upewniamy się, że repozytorium jest prywatne i zespół ma świadomość, że każdy z prawem odczytu widzi wtedy też klucze. Bezpieczniejszym wariantem jest osobne repozytorium tylko na certyfikaty, z węższym gronem osób mających dostęp.

```
git add nginx.conf conf.d/
git commit -m "Konfiguracja Nginx z docker01"
git push -u origin main
```

3. Kopia po każdej zmianie

```
cd /docker/apps/nginx
git add -A
git commit -m "Opis zmiany: nowy blok server dla ipa.zielona.example"
git push
```

Przykład:

```
[main a1b2c3d] Opis zmiany: nowy blok server dla ipa.zielona.example
 1 file changed, 12 insertions(+)
```

4. Aktualizacja pakietu z testem i wycofaniem

Przed aktualizacją zapisujemy obecną wersję, żeby mieć punkt odniesienia:

```
rpm -q nginx
```

```
sudo dnf update nginx
```

Po aktualizacji sprawdzamy składnię, zanim cokolwiek przeładujemy:

```
sudo nginx -t
```

Uważaj: `dnf update` podmienia binarkę na dysku, ale działający proces nadal używa starej, wczytanej do pamięci przy starcie, dopóki nie wykonamy przeładowania. Dopiero `nginx -t` zakończone sukcesem daje zielone światło na `sudo systemctl restart nginx`. Jeśli test się nie powiedzie, stara wersja binarki wciąż działa bez przerwy w ruchu, mamy czas na spokojną poprawkę.

Wycofanie do poprzedniej wersji pakietu, jeśli nowa okazała się niekompatybilna:

```
sudo dnf downgrade nginx
sudo nginx -t && sudo systemctl restart nginx
```

5. Aktualizacja obrazu w wariancie Docker

```
cd /docker/compose/nginx
```

`compose.yaml`, zmiana samego tagu wersji:

```
services:
  nginx:
    image: nginx:1.30.6
```

```
docker compose pull
docker compose exec nginx nginx -t
docker compose up -d
```

Test składni wykonujemy na jeszcze działającym starym kontenerze przed `docker compose up -d`, żeby złapać błąd składniowy w zamontowanej konfiguracji, zanim w ogóle spróbujemy podmienić obraz.

6. Odnawianie certyfikatów i przeładowanie bez przerwy

Po podmianie plików certyfikatu i klucza w `/etc/nginx/ssl/` (albo `/docker/apps/nginx/certs/` w wariancie Docker), Nginksa informujemy o zmianie przeładowaniem, nigdy restartem:

```
sudo nginx -t && sudo nginx -s reload
```

W wariancie Docker:

```
docker compose exec nginx nginx -t && docker compose exec nginx nginx -s reload
```

`reload` wczytuje nowy plik certyfikatu do nowych procesów roboczych, jednocześnie zostawiając stare procesy przy życiu do zakończenia obsługiwanych już żądań, opisane szerzej na stronie Architektura Nginx, procesy, konteksty, pliki konfiguracyjne. Żaden aktywny klient nie widzi przerwy w połączeniu.

Weryfikacja

```
openssl s_client -connect NazwaDomeny:443 -servername NazwaDomeny </dev/null 2>/dev/null |
openssl x509 -noout -dates
```

Przykład po odnowieniu certyfikatu:

```
notBefore=Sep 28 00:00:00 2026 GMT
notAfter=Dec 27 00:00:00 2026 GMT
```

```
sudo systemctl status nginx | grep Active
```

Przykład:

```
Active: active (running) since Mon 2026-09-28 10:02:11 UTC; 3s ago
```

Wycofanie zmian

Cofnięcie ostatniej zmiany konfiguracji z historii Gita, zamiast ręcznego odtwarzania z pamięci:

```
cd /docker/apps/nginx
git log --oneline -5
git revert a1b2c3d
sudo nginx -t && sudo nginx -s reload
```

Gdy coś nie zadziała

Objaw	Przyczyna	Co robimy
Po `dnf update nginx` usługa nie startuje	nowa wersja odrzuca dyrektywę usuniętą albo zmienioną między wersjami	`nginx -t` pokazuje dokładną linię błędu, poprawiamy albo wycofujemy pakiet komendą `dnf downgrade`
`nginx -s reload` nie widzi nowego certyfikatu	podmieniono plik pod inną ścieżką niż wskazana w `ssl_certificate`	sprawdzamy ścieżkę w bloku `server`, potwierdzamy `openssl x509 -noout -dates` na pliku wskazanym w konfiguracji
`git push` do Gitea odrzucony	konflikt z równoległą zmianą innego administratora	`git pull --rebase`, rozwiązujemy konflikt ręcznie, dopiero potem `git push`
Po `docker compose up -d` nowej wersji obrazu kontener nie startuje	konfiguracja zamontowana `:ro` używa dyrektywy nieobsługiwanej przez nową wersję obrazu	`docker compose logs nginx`, cofamy tag obrazu w `compose.yaml` do poprzedniej działającej wersji

Powiązane: Architektura Nginx, procesy, konteksty, pliki konfiguracyjne · TLS w Nginx · Gitea · Instalacja Nginx na Rocky Linux 9 i w Dockerze

Źródła: nginx.org, dokumentacja poleceń `-s reload` i `-t` · `man 8 nginx` · docs.gitea.com, rozdział „Basic Git" · `man 1 git-revert`

Revision #24
Created 2026-09-28 17:15:31 UTC by Tomasz Konieczny
Updated 2026-10-26 14:16:17 UTC by Tomasz Konieczny