Nginx: serwer, który powstał, żeby obsłużyć dziesięć tysięcy połączeń

Poziom: podstawy **Dotyczy: Nginx, cała książka** **Czas: ~15 min**

Ta strona otwiera książkę o Nginksie: skąd wziął się serwer, który dziś stoi przed niemal każdą aplikacją webową w Zielonej, jaki problem miał rozwiązać na starcie i co czeka na kolejnych stronach.

Skróty użyte na tej stronie (7)

Skrót	Co znaczy
C10K	problem C10K, dziesięć tysięcy jednoczesnych połączeń: pytanie postawione w 1999 roku, jak jeden serwer ma obsłużyć taki ruch bez klękania pod własnym ciężarem, od którego zaczyna się historia Nginksa
FTP	File Transfer Protocol: stary protokół przesyłania plików, bez szyfrowania
HTTP	Hypertext Transfer Protocol: protokół stron WWW
IT	Information Technology: dział albo praca związana z komputerami i infrastrukturą firmy
TLS	Transport Layer Security: szyfrowanie połączeń, np. w HTTPS
WebSocket	stałe, dwukierunkowe połączenie przeglądarki z serwerem, przez które Home Assistant na żywo odświeża panele; serwer pośredniczący musi przekazywać nagłówki Upgrade i Connection
WWW	World Wide Web: strony i aplikacje otwierane w przeglądarce po HTTP albo HTTPS

Skąd się wzięło

Pod koniec lat dziewięćdziesiątych popularne wtedy serwery WWW, głównie Apache, uruchamiały osobny proces albo wątek systemowy na każde przychodzące połączenie. Przy paruset jednoczesnych użytkownikach to jeszcze działało, ale rosnące portale internetowe zaczęły przyjmować tysiące połączeń naraz, a każdy dodatkowy proces zjadał pamięć i czas procesora na

samo przełączanie między nimi.

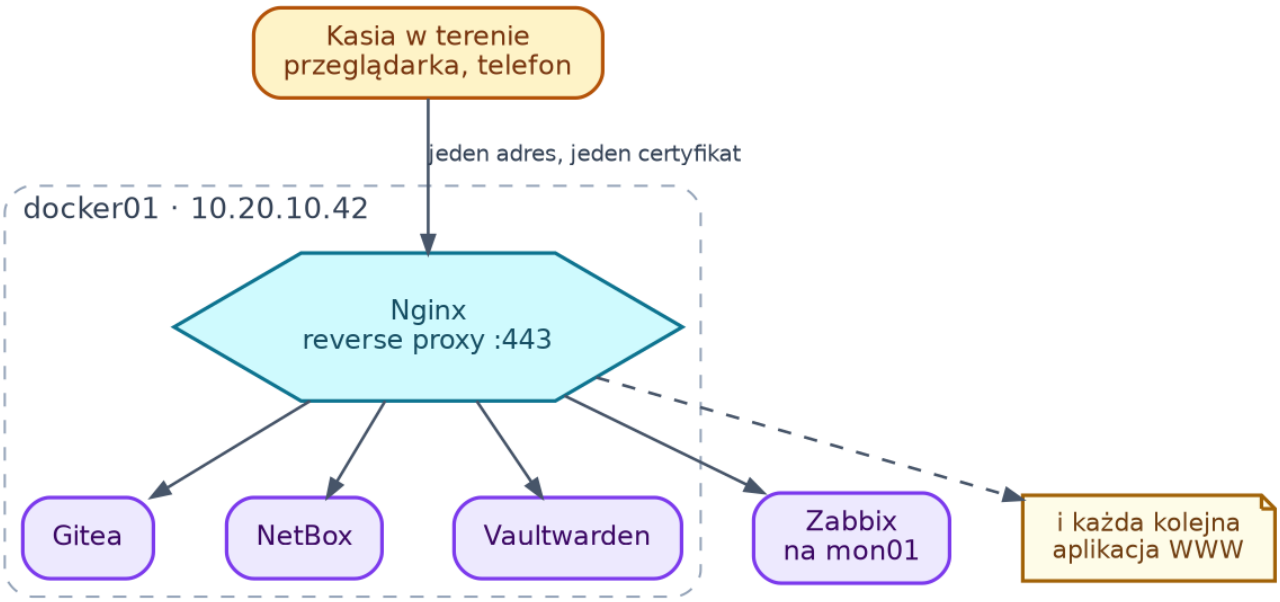
> **Pojęcie: problem C10K.** Nazwa ukuta w 1999 roku przez inżyniera Dana Kegela, opisująca pytanie, jak jeden serwer ma obsłużyć dziesięć tysięcy jednoczesnych połączeń bez klękania pod własnym ciężarem. Kegel sformułował ją po tym, jak zobaczył serwer FTP `cdrom.com` obsługujący dziesięć tysięcy klientów naraz na jednym łączu gigabitowym. Nazwa przylgnęła do całej fali nowych serwerów zdarzeniowych, które zamiast procesu na połączenie trzymały niewielką liczbę procesów obsługujących tysiące połączeń jednocześnie.

Igor Sysoev, rosyjski programista pracujący wtedy dla Ramblera, jednego z największych rosyjskich portali internetowych, zmagał się z dokładnie tym problemem na co dzień. Zaczął pisać własny serwer HTTP jako projekt poboczny już w 2002 roku, właśnie w odpowiedzi na C10K: architektura zdarzeniowa zamiast procesu na połączenie, mały narzut pamięci na jednoczesne żądanie. W 2004 roku wydał go publicznie pod nazwą Nginx, w skrócie od angielskiego „engine x". Do września 2008 roku sam Rambler obsługiwał nim już 500 milionów żądań dziennie.

Nginx rozchodził się dalej niezależnie od twórcy, a w lipcu 2011 roku Sysoev razem z Maximem Konovalovem założyli firmę Nginx, Inc., żeby sprzedawać wsparcie i produkty komercyjne wokół darmowego serwera. W marcu 2019 roku firmę kupił F5 Networks za 670 milionów dolarów, przejmując rozwój flagowego produktu razem z zespołem, choć sam serwer Nginx pozostał darmowy i otwarty jak wcześniej.

Jak to działa w jednym obrazku

Zamiast tworzyć nowy proces systemowy na każde połączenie, Nginx trzyma niewielką liczbę procesów roboczych, z których każdy obsługuje tysiące połączeń jednocześnie, przełączając się między nimi tylko wtedy, gdy któreś faktycznie ma dane do przeczytania albo wysłania. Ten sam mechanizm pozwala w Zielonej jednemu Nginksowi na `docker01` stać przed wieloma aplikacjami naraz, zamiast wystawiać każdą z nich osobno na świat.



Nginx jako reverse proxy przed Gitea, NetBoksem, Vaultwardenem i Zabbiksem w Zielonej (kliknięcie otwiera pełny rozmiar)

Dlaczego Zielona

> **Z ŻYCIA WZIĘTE**
>
> Wiosna 2022 roku. Zespół IT Zielonej trzyma wtedy Giteę na porcie 3000, a NetBoksa na porcie 8000, Vaultwardena jeszcze nie ma. Każda aplikacja ma własny certyfikat TLS, własny port do zapamiętania i własny wpis w regułach FortiGate. Ola pyta, dlaczego link do repozytorium wygląda jak `https://docker01.zielona.example:3000`, zamiast po prostu `https://git.zielona.example`.

Odpowiedzią był reverse proxy, opisany szerzej na stronie Reverse proxy i równoważenie obciążenia. Jeden Nginx na `docker01` przejął port 443 dla wszystkich aplikacji naraz, każdej przydzielając własną nazwę domenową zamiast numeru portu, i własny certyfikat TLS bez trzymania go osobno w każdej aplikacji. Zespół IT dodaje dziś nową aplikację do Zielonej, dopisując kilka linii do konfiguracji Nginksa, zamiast otwierać kolejny port na zaporze i tłumaczyć użytkownikom nowy adres.

Drugi powód jest równie praktyczny: Nginx radzi sobie z ruchem, który same aplikacje obsługują gorzej, jak długie połączenia WebSocket używane przez powiadomienia w Vaultwardenie, albo przekazywanie żądań do kilku instancji tej samej aplikacji naraz przy większym ruchu. Dwuosobowy zespół nie ma czasu uczyć każdej aplikacji z osobna obsługi TLS i równoważenia ruchu, więc robi to

raz, w jednym miejscu, przed nimi wszystkimi.

Mapa tej książki

Strona	Na jakie pytanie odpowiada
Nginx: serwer, który powstał, żeby obsłużyć dziesięć tysięcy połączeń	Skąd wziął się Nginx i po co stoi przed każdą aplikacją w Zielonej
Instalacja Nginx na Rocky Linux 9 i w Dockerze	Jak postawić Nginksa z pakietu albo w kontenerze
Architektura Nginx, procesy, konteksty, pliki konfiguracyjne	Z czego Nginx się składa i co znaczy przeładowanie konfiguracji
Bloki server i location	Który blok obsłuży dane żądanie i w jakiej kolejności
Reverse proxy i równoważenie obciążenia	Jak przekazać ruch do aplikacji i rozłożyć go na kilka serwerów
TLS w Nginx	Jak skonfigurować szyfrowane połączenie i sprawdzić, co serwer faktycznie podaje
Logi i diagnostyka	Gdzie szukać przyczyny błędu 502, 504 albo 413
Nginx: kopia konfiguracji, certyfikatów i bezpieczna aktualizacja	Jak bezpiecznie zaktualizować Nginksa i nie zgubić certyfikatów

Powiązane: Reverse proxy i równoważenie obciążenia · Instalacja Nginx na Rocky Linux 9 i w Dockerze · Konwencja Dockera w Zielonej

Źródła: en.wikipedia.org/wiki/Nginx · en.wikipedia.org/wiki/Igor_Sysoev · en.wikipedia.org/wiki/C10k_problem · blog.nginx.org/blog/celebrating-20-years-of-nginx · thehackernews.com/2019/03/f5-networks-acquires-nginx.html · kegel.com/c10k.html

---

Revision #26
Created 2026-09-28 17:15:28 UTC by Tomasz Konieczny
Updated 2026-10-26 14:16:07 UTC by Tomasz Konieczny