Poznajemy Gitea

Opowieść o Gicie i Gitea oraz mapa książki.

- Gitea: Git z filiżanką herbaty

Gitea: Git z filiżanką herbaty

Poziom: podstawy **Dotyczy: Gitea 1.27** **Czas: ~15 min**

Ta strona otwiera książkę o Gitea: skąd wziął się Git, dlaczego firma taka jak Zielona w ogóle potrzebuje własnego serwera do kodu i co czeka na kolejnych stronach.

Skróty użyte na tej stronie (1)

Skrót	Co znaczy
MIT	MIT License: bardzo liberalna licencja wolnego oprogramowania, pozwala na dowolne użycie i modyfikację kodu

Skąd się wzięło

> **Pojęcie: Git.** Program do śledzenia zmian w plikach, głównie w kodzie źródłowym: zapisuje kto, kiedy i co zmienił, pozwala wrócić do dowolnej wcześniejszej wersji i pozwala kilku osobom pracować równolegle nad tym samym projektem bez nadpisywania swojej pracy.

Git napisał Linus Torvalds w 2005 roku, twórca jądra Linuksa. Do tego czasu społeczność jądra korzystała z komercyjnego programu BitKeeper, użyczanego deweloperom bezpłatnie przez firmę BitMover. Gdy BitMover wycofał tę bezpłatną licencję po sporze o inne narzędzie, Torvalds w kilka tygodni napisał od zera własny system, dopasowany do jednej rzeczy: obsługi bardzo dużego projektu z tysiącami współpracujących osób. Nazwał go Git, co w brytyjskim slangu oznacza nieznośnego, upierdliwego typa. Sam Torvalds żartował później, że skoro nazywa programy swoim imieniem (Linux od Linusa), tym razem wybrał nazwę pasującą do niego samego.

Git sam w sobie działa lokalnie, na dysku jednej osoby. Do pracy zespołowej potrzebny jest wspólny punkt, do którego wszyscy wysyłają i z którego pobierają zmiany: serwer Git. Pierwsze publiczne serwisy tego typu (GitHub od 2008 roku) postawiły na hosting w chmurze, ale firma, która chce trzymać swój kod u siebie, we własnej sieci, bez wysyłania go na zewnątrz, potrzebuje własnego serwera.

> **Pojęcie: Gitea.** Lekki serwer Git z własnym interfejsem webowym: konta użytkowników, organizacje, przegląd zmian (pull requesty), integracje przez webhooki. Funkcjonalnie

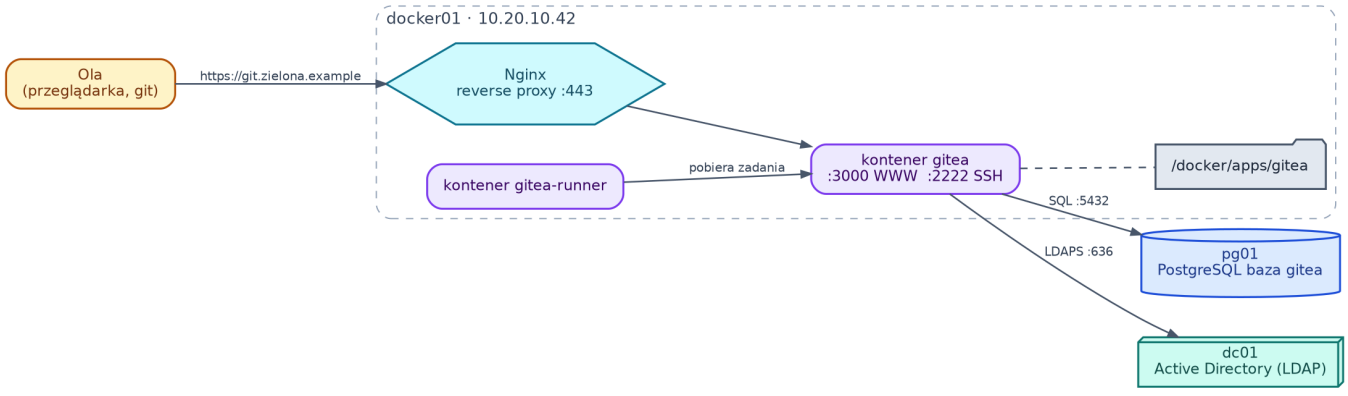
> odpowiada GitLabowi czy wewnętrznemu GitHubowi, ale wymaga ułamka ich zasobów sprzętowych.

Gitea nie powstała od zera. W 2014 roku Lunny Xiao stworzył Gogs, lekką alternatywę dla GitLaba napisaną w języku Go. Gogs szybko zyskał popularność, ale repozytorium projektu kontrolowała jedna osoba, a decyzje o kierunku rozwoju zapadały bez udziału reszty współpracujących programistów. W listopadzie 2016 roku część społeczności Gogs odłączyła się i założyła Gitea: ten sam kod na start, ale z otwartym, wspólnym zarządzaniem projektem zamiast jednego decydenta. Nazwa to gra słów: Git plus tea, czytana z twardym „g", z hasłem „Git with a cup of tea", czyli „Git z filiżanką herbaty".

Ciekawostka: w październiku 2022 roku dwaj główni opiekunowie Gitea, Lunny Xiao i Matti Ranta, założyli spółkę Gitea Limited (dziś działającą pod nazwą CommitGo), żeby sprzedawać płatny hosting i wersję rozszerzoną. Domena i znak towarowy projektu trafiły do tej spółki bez pełnej zgody społeczności, co części programistów się nie spodobało. Serwis Codeberg odpowiedział jeszcze w 2022 roku własnym forkiem o nazwie Forgejo, na początku bliźniaczo podobnym do Gitea, od początku 2024 roku rozwijanym już niezależnie. Sama Gitea zostaje wolnym oprogramowaniem na licencji MIT niezależnie od tego sporu, a Zielona korzysta właśnie z niej.

Jak to działa w jednym obrazku

Zespół Oli otwiera Giteę w przeglądarce pod adresem `git.zielona.example`. Ruch trafia najpierw na Nginksa na `docker01`, opisanego na stronie Reverse proxy i równoważenie obciążenia, a dopiero stamtąd do kontenera Gitea. Sama Gitea trzyma metadane (konta, uprawnienia, opisy zgłoszeń) w bazie PostgreSQL na `pg01`, a logowanie sprawdza w Active Directory na `dc01`. Osobny kontener, gitea-runner, opisany na stronie części 2 Gitea Actions: runner i pierwszy workflow, pobiera z Gitea zadania do automatycznego budowania kodu.



Architektura Gitea w Zielonej: przeglądarka, Nginx, kontener Gitea, baza i Active Directory (kliknięcie otwiera pełny rozmiar)

Dlaczego Zielona

> **Z ŻYCIA WZIĘTE**
>
> Piątek, przed wdrożeniem Gitea. Zespół Oli trzyma cały kod cennika na udziale sieciowym `\\fs01\IT\Programy\cennik_final_v3_NAPRAWDE.zip`. Około 16:00 dwie osoby z zespołu, pracując nad tym samym plikiem, zapisują go niemal w tej samej minucie: jedna wersja nadpisuje drugą, a poranna poprawka Tomka znika bez śladu. Nikt nie wie, czyja wersja jest ostatnia, dopóki Ola nie porówna plików ręcznie w Wordzie.

Taka sytuacja powtarzała się w Zielonej regularnie, zanim zespół programistów dostał własny serwer Git. Nazwa pliku `cennik_final_v3_NAPRAWDE.zip` mówi sama za siebie: bez systemu kontroli wersji każdy trzyma swoją kopię i ręcznie dogaduje się z resztą, co jest aktualne. Gitea rozwiązuje to jednym mechanizmem: każda zmiana trafia jako osobny commit z autorem i datą, dwie osoby mogą pracować na tym samym pliku równolegle na osobnych gałęziach, a scalenie zmian, opisane na stronie Pierwsze repozytorium: od pustego projektu do pierwszego pusha, pilnuje, żeby żadna wersja nie zniknęła po cichu.

Do tego dochodzi kontrola dostępu: Ola widzi i zatwierdza wszystko w organizacji swojego zespołu, Tomek jako nowa osoba dostaje dokładnie te uprawnienia, jakie zespół mu przyzna, a odejście kogoś z firmy to jedno kliknięcie usuwające dostęp, zamiast zmiany hasła do współdzielonego udziału sieciowego.

Mapa tej książki

Książka o Gitea składa się z dziesięciu stron w dwóch częściach. Część 1 (ta) prowadzi od instalacji do pierwszego pusha, część 2 zajmuje się automatyzacją i utrzymaniem.

Strona	Na jakie pytanie odpowiada
Gitea: Git z filiżanką herbaty	Skąd wzięła się Gitea i po co ją mamy
Instalacja Gitea w Dockerze	Jak postawić Giteę w kontenerze, w laboratorium i w produkcji
Instalacja Gitea bez Dockera	Jak postawić Giteę jako usługę systemd, bez kontenerów
Pierwsze uruchomienie: kreator, administrator, reverse proxy	Jak przejść kreator instalacji i wystawić Giteę przez Nginksa

Strona	Na jakie pytanie odpowiada
Logowanie kontem z Active Directory	Jak podłączyć logowanie przez LDAP zamiast kont lokalnych
Pierwsze repozytorium: od pustego projektu do pierwszego pusha	Jak założyć repozytorium i wykonać pierwszy commit, push i pull request
Gitea Actions: runner i pierwszy workflow	Jak uruchomić automatyczne buildy po każdym pushu
Pierwszy kontener: obraz budowany w Gitea i wdrożony na docker01	Jak z kodu w Gitea zrobić działający kontener na serwerze
Kopia zapasowa i aktualizacja Gitea	Jak bezpiecznie zrobić backup i podnieść wersję
Gitea: częste problemy	Co robić, gdy coś nie działa

Powiązane: Reverse proxy i równoważenie obciążenia · Active Directory: konta, grupy, jednostki organizacyjne

Źródła: en.wikipedia.org/wiki/Gitea · en.wikipedia.org/wiki/Forgejo · forgejo.org/compare-to-gitea · blog.gitea.com