Jenkins: lokaj, który buduje kod od 2004 roku

Poziom: podstawy **Dotyczy: Jenkins LTS 2.568.3** **Czas: ~15 min**

Ta strona otwiera książkę o Jenkinsie: skąd wziął się serwer, który od dwudziestu lat buduje cudzy kod, dlaczego zmienił nazwę w połowie życia i dlaczego w Zielonej stoi dziś obok nowszego TeamCity zamiast go zastąpić.

Skróty użyte na tej stronie (2)

Skrót	Co znaczy
CI	Continuous Integration: automatyczne budowanie i testowanie kodu po każdej zmianie; w ITIL także configuration item, czyli element konfiguracji: każdy składnik usługi zapisany w CMDB
LTS	Long Term Support: wersja z długim wsparciem poprawkami

Skąd się wzięło

> **Pojęcie: serwer CI.** Program, który po każdej zmianie w repozytorium sam pobiera kod, kompiluje go, uruchamia testy i zgłasza wynik, zamiast czekać, aż ktoś zrobi to ręcznie na swoim komputerze. CI to skrót od Continuous Integration, czyli ciągłej integracji: im częściej zespół łączy swoje zmiany i sprawdza je automatycznie, tym szybciej wychodzi na jaw, że dwie osoby napisały coś, co się wzajemnie wyklucza.

Projekt zaczął się w 2004 roku, kiedy Kohsuke Kawaguchi, programista pracujący wtedy w Sun Microsystems, miał dość tego, że jego zespół codziennie wychodził z pracy, nie wiedząc, czy to, co właśnie wpisali do repozytorium, w ogóle się kompiluje. Napisał sobie prosty serwer, który pilnował tego za nich, i nazwał go Hudson. Pierwsze publiczne wydanie wyszło w lutym 2005 roku.

Hudson szybko wyszedł poza jeden zespół w Sun Microsystems: w 2008 roku dostał nagrodę Duke's Choice Award na konferencji JavaOne i stał się jednym z najpopularniejszych serwerów CI w świecie Javy, obok starszego CruiseControl. Kłopoty zaczęły się razem z przejęciem Sun Microsystems przez Oracle. Nowy właściciel chciał mocniej kontrolować kierunek rozwoju projektu i w grudniu 2010 roku zgłosił nazwę „Hudson" jako swój znak towarowy, co społeczności współtwórców,

przyzwyczajonej do decydowania wspólnie, się nie spodobało.

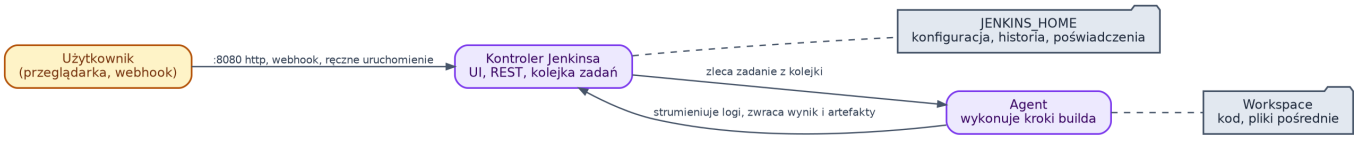
11 stycznia 2011 roku Kawaguchi zaproponował społeczności prostą rzecz: skoro sporu o nazwę i kontrolę nad projektem nie da się rozwiązać polubownie, zmieńmy nazwę i rozwijajmy dalej pod nią, niezależnie od Oracle. Głosowanie 29 stycznia 2011 roku przeszło zdecydowaną większością. Oracle ogłosił własną kontynuację Hudsona kilka dni później, ale to rozwidlenie pod nazwą Jenkins skupiło większość dotychczasowych współtwórców i całą energię społeczności. Sam Hudson trafił w maju 2011 roku pod skrzydła Fundacji Eclipse, a w 2017 roku został oficjalnie uznany za zakończony.

Ciekawostka: nazwę „Jenkins" Kawaguchi wybrał świadomie z tego samego szufladkowego pomysłu co „Hudson": to kolejne angielskie imię kojarzone z kamerdynerem, tylko niezajęte przez żaden inny projekt. Logo Jenkinsa, zaprojektowane przez studio Frontside Bureau, pokazuje wprost tę metaforę: wąsatego lokaja w niebieskiej marynarce i czerwonej muszce, który „wykonuje zlecone mu zadanie", tak jak prawdziwy kamerdyner wykonuje polecenia domu. Stąd tytuł tej strony: Jenkins od dwudziestu lat jest właśnie takim cyfrowym lokajem, tylko zamiast podawać herbatę, buduje kod.

Rok po rozwidleniu, w 2010, dwaj przedsiębiorcy, Sacha Labourey i Francois Dechery, założyli firmę CloudBees, która jeszcze w tym samym roku przejęła firmę Kawaguchiego, InfraDNA. Kawaguchi objął w CloudBees stanowisko dyrektora technicznego w 2014 roku, a sama firma jest dziś głównym komercyjnym opiekunem projektu: sprzedaje wsparcie, wersję rozszerzoną i chmurową, ale sam Jenkins zostaje wolnym oprogramowaniem, tak jak trafił do Zielonej.

Jak to działa w jednym obrazku

Jenkins rozdziela dwie role: kontroler przyjmuje zlecenia i trzyma konfigurację, a agent wykonuje faktyczny build na swojej maszynie. Ten podział, opisany szczegółowo na stronie Architektura Jenkinsa: kontroler, agenci, workspace, jest sercem wszystkiego, co Jenkins robi od dwudziestu lat, niezależnie od tego, czy kontroler stoi na gołym serwerze, czy w kontenerze.



Kontroler przyjmuje żądania i zleca zadania agentowi, trzymając konfigurację w JENKINS_HOME, a agent pracuje w swoim workspace (kliknięcie otwiera pełny rozmiar)

Dlaczego Zielona

> **Z ŻYCIA WZIĘTE**
>
> Ola pamięta jeszcze czasy, kiedy jej zespół trzymał skrypty budujące na własnych laptopach, uruchamiane ręcznie przed każdym wydaniem. Jenkins trafił na `ci01` jako jeden z pierwszych porządnych narzędzi w firmie, na długo przed tym, jak ktokolwiek w Zielonej usłyszał słowo „DevOps". Dziś, kiedy zespół Oli ocenia TeamCity jako nowocześniejszą alternatywę, nikt nie ma odwagi ruszyć trzydziestu istniejących pipeline'ów Jenkinsa, które po prostu działają.

Tak Jenkins i TeamCity wylądowały obok siebie na jednym serwerze `ci01`: nie dlatego, że Zielona nie potrafi się zdecydować, tylko dlatego, że migracja działającego systemu produkcyjnego bez wyraźnej korzyści rzadko jest priorytetem numer jeden. Nowe projekty zespół Oli częściej zakłada w TeamCity, ale stare, sprawdzone Jenkinsfile zostają tam, gdzie są, dopóki nikt nie znajdzie dobrego powodu, żeby je przepisać. To bardzo typowa sytuacja w firmie, która działa od 1998 roku: obok najnowszego rozwiązania zawsze stoi coś starszego, co wciąż zarabia na swoje utrzymanie.

Mapa tej książki

Książka o Jenkinsie prowadzi od architektury, przez dwa warianty instalacji i pracę z agentami, aż po pipeline'y, bezpieczeństwo, logowanie domenowe i utrzymanie.

Strona	Na jakie pytanie odpowiada
Jenkins: lokaj, który buduje kod od 2004 roku	Skąd wziął się Jenkins i dlaczego wciąż stoi obok TeamCity
Architektura Jenkinsa: kontroler, agenci, workspace	Z czego składa się Jenkins i gdzie faktycznie wykonują się buildy
Instalacja i aktualizacja Jenkinsa	Jak postawić kontroler jako usługę systemd, bez kontenerów
Agenci: dodanie, diagnostyka, utrzymanie	Jak dodać agenta po SSH i naprawić najczęstsze awarie połączenia
Instalacja Jenkinsa w Dockerze	Jak postawić kontroler i agenta w kontenerach, z reverse proxy
Pipeline: Jenkinsfile, etapy, artefakty	Jak zapisać proces budowania w pliku trzymanym razem z kodem
Poświadczenia i bezpieczeństwo	Jak bezpiecznie trzymać sekrety i kto co może kliknąć
Jenkins: logowanie kontem z Active Directory	Jak podłączyć logowanie domenowe zamiast zakładania kont lokalnych

Strona	Na jakie pytanie odpowiada
Utrzymanie i diagnostyka	Co sprawdzać regularnie, żeby dysk się nie zapełnił, a kolejka nie urosła

Powiązane: Świat przykładów: Zielona · TeamCity

Źródła: en.wikipedia.org/wiki/Jenkins_(software) · en.wikipedia.org/wiki/Hudson_(software) · en.wikipedia.org/wiki/CloudBees · kohsuke.org, wpis „Bye bye Hudson, hello Jenkins" · jenkins.io/download/lts

---

Revision #23
Created 2026-09-28 16:57:44 UTC by Tomasz Konieczny
Updated 2026-10-26 14:14:53 UTC by Tomasz Konieczny