

Jenkins - tak dla automatyzacji!

Tomasz Jurczyk

Politechnika Wrocławska

21 kwietnia 2012

Agenda

- 1 Wstęp - o czym jest ta prezentacja?

Agenda

- 1 Wstęp - o czym jest ta prezentacja?
- 2 Idea Continuous Integration

Agenda

- 1 Wstęp - o czym jest ta prezentacja?
- 2 Idea Continuous Integration
- 3 Jenkins - podstawowe narzędzie pracy

- 1 Wstęp - o czym jest ta prezentacja?
- 2 Idea Continuous Integration
- 3 Jenkins - podstawowe narzędzie pracy
- 4 Techniczne aspekty konfiguracji

- 1 Wstęp - o czym jest ta prezentacja?
- 2 Idea Continuous Integration
- 3 Jenkins - podstawowe narzędzie pracy
- 4 Techniczne aspekty konfiguracji
- 5 Budowanie rozproszone

- 1 Wstęp - o czym jest ta prezentacja?
- 2 Idea Continuous Integration
- 3 Jenkins - podstawowe narzędzie pracy
- 4 Techniczne aspekty konfiguracji
- 5 Budowanie rozproszone
- 6 Podsumowanie

- 1 Wstęp - o czym jest ta prezentacja?
- 2 Idea Continuous Integration
- 3 Jenkins - podstawowe narzędzie pracy
- 4 Techniczne aspekty konfiguracji
- 5 Budowanie rozproszone
- 6 Podsumowanie
- 7 Słowem podsumowania - warto?

- 1 Wstęp - o czym jest ta prezentacja?
- 2 Idea Continuous Integration
- 3 Jenkins - podstawowe narzędzie pracy
- 4 Techniczne aspekty konfiguracji
- 5 Budowanie rozproszone
- 6 Podsumowanie
- 7 Słowem podsumowania - warto?
- 8 AMA

Continuous Integration?

Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily - leading to multiple integrations per day. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible. Many teams find that this approach leads to significantly reduced integration problems and allows a team to develop cohesive software more rapidly. This article is a quick overview of Continuous Integration summarizing the technique and its current usage.

Zasady CI'a

- 1 Kod w kontroli wersji

Zasady CI'a

- 1 Kod w kontroli wersji
- 2 Automatyizacja budowania

Zasady CI'a

- 1 Kod w kontroli wersji
- 2 Automatyizacja budowania
- 3 Automatyczne testowanie

Zasady CI'a

- 1 Kod w kontroli wersji
- 2 Automatyizacja budowania
- 3 Automatyczne testowanie
- 4 Codziennie dostarczanie kodu

Zasady CI'a

- 1 Kod w kontroli wersji
- 2 Automatyizacja budowania
- 3 Automatyczne testowanie
- 4 Codziennie dostarczanie kodu
- 5 Dostarczenie = build

- 1 Kod w kontroli wersji
- 2 Automatyizacja budowania
- 3 Automatyczne testowanie
- 4 Codziennie dostarczanie kodu
- 5 Dostarczenie = build
- 6 Szybkość, szybkość, szybkość...

- 1 Kod w kontroli wersji
- 2 Automatyizacja budowania
- 3 Automatyczne testowanie
- 4 Codziennie dostarczanie kodu
- 5 Dostarczenie = build
- 6 Szybkość, szybkość, szybkość...
- 7 Staging

- 1 Kod w kontroli wersji
- 2 Automatyizacja budowania
- 3 Automatyczne testowanie
- 4 Codziennie dostarczanie kodu
- 5 Dostarczenie = build
- 6 Szybkość, szybkość, szybkość...
- 7 Staging
- 8 Zmiany przez pryzmat buildów

- 1 Kod w kontroli wersji
- 2 Automatyizacja budowania
- 3 Automatyczne testowanie
- 4 Codziennie dostarczanie kodu
- 5 Dostarczenie = build
- 6 Szybkość, szybkość, szybkość...
- 7 Staging
- 8 Zmiany przez pryzmat buildów
- 9 Przejrzystość

Czy warto się w to bawić?

Czy warto się w to bawić?

Zdecydowanie tak!

Czy warto się w to bawić?

Zdecydowanie tak!

- 1 Szybka informacja, które dostarczenie powoduje problem

Czy warto się w to bawić?

Zdecydowanie tak!

- 1 Szybka informacja, które dostarczenie powoduje problem
- 2 Deweloperzy mogą doglądać na bieżąco stan oprogramowania

Czy warto się w to bawić?

Zdecydowanie tak!

- 1 Szybka informacja, które dostarczenie powoduje problem
- 2 Deweloperzy mogą doglądać na bieżąco stan oprogramowania
- 3 Zespolenie budowania z testami jednostkowymi

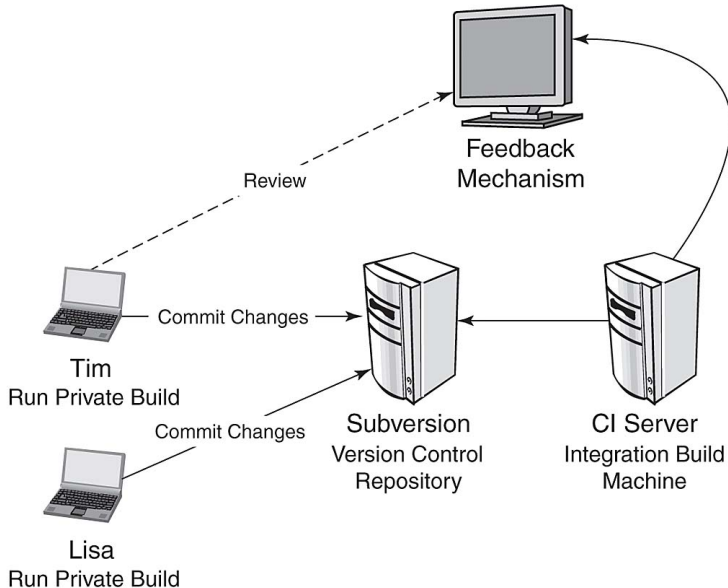
Czy warto się w to bawić?

Zdecydowanie tak!

- 1 Szybka informacja, które dostarczenie powoduje problem
- 2 Deweloperzy mogą doglądać na bieżąco stan oprogramowania
- 3 Zespolenie budowania z testami jednostkowymi
- 4 Reasumując - dzięki temu możemy tworzyć szybciej i lepiej

CI - jak to działa?

CI - jak to działa?



- 1 Wystartowany w 2007 roku jako alternatywa dla CC

- 1 Wystartowany w 2007 roku jako alternatywa dla CC
- 2 Poprzednio znany jako Hudson (split 11 stycznia 2011)

- 1 Wystartowany w 2007 roku jako alternatywa dla CC
- 2 Poprzednio znany jako Hudson (split 11 stycznia 2011)
- 3 Dostarcza systemy dla SCMu potrzebne do budowania

- 1 Wystartowany w 2007 roku jako alternatywa dla CC
- 2 Poprzednio znany jako Hudson (split 11 stycznia 2011)
- 3 Dostarcza systemy dla SCMu potrzebne do budowania
- 4 Licencja: MIT

- 1 Wystartowany w 2007 roku jako alternatywa dla CC
- 2 Poprzednio znany jako Hudson (split 11 stycznia 2011)
- 3 Dostarcza systemy dla SCM-u potrzebne do budowania
- 4 Licencja: MIT
- 5 Cotygodniowe releasey + LTS

Instalacja?

Instalacja?

Nic prostszego!

Instalacja?

Nic prostszego!

- 1 `wget http://mirrors.jenkins-ci.org/war/latest/jenkins.war`

Instalacja?

Nic prostszego!

- 1 `wget http://mirrors.jenkins-ci.org/war/latest/jenkins.war`
- 2 `java -jar jenkins.war`

Instalacja?

Nic prostszego!

- 1 `wget http://mirrors.jenkins-ci.org/war/latest/jenkins.war`
- 2 `java -jar jenkins.war`
- 3 Efekt?

O co tutaj chodzi?

O co tutaj chodzi?

- 1 Buildy, joby, widoki...

O co tutaj chodzi?

- 1 Buildy, joby, widoki...
- 2 Przyjazne "Changes"

O co tutaj chodzi?

- 1 Buildy, joby, widoki...
- 2 Przyjazne "Changes"
- 3 Aspekty pozwalające automatyzować

O co tutaj chodzi?

- 1 Buildy, joby, widoki...
- 2 Przyjazne "Changes"
- 3 Aspekty pozwalające automatyzować
- 4 Pluginy

Zbudujmy coś!

Jak to tak naprawdę wygląda?

Konfiguracja

1 Konfiguracja przez WWW

- 1 Konfiguracja przez WWW
- 2 Konfiguracja przez XML

- 1 Konfiguracja przez WWW
- 2 Konfiguracja przez XML
- 3 API

① \$url/api

- 1 \$url/api
 - 1 XML

- 1 \$url/api
 - 1 XML
 - 2 JSON

- 1 \$url/api
 - 1 XML
 - 2 JSON
 - 3 Python

- 1 \$url/api
 - 1 XML
 - 2 JSON
 - 3 Python
- 2 <http://pypi.python.org/pypi/jenkinsapi>

- 1 \$url/api
 - 1 XML
 - 2 JSON
 - 3 Python
- 2 <http://pypi.python.org/pypi/jenkinsapi>
- 3 <https://github.com/kellyrob99/Jenkins-api-tour>

- 1 \$url/api
 - 1 XML
 - 2 JSON
 - 3 Python
- 2 <http://pypi.python.org/pypi/jenkinsapi>
- 3 <https://github.com/kellyrob99/Jenkins-api-tour>
- 4 <https://launchpad.net/python-jenkins>

- 1 \$url/api
 - 1 XML
 - 2 JSON
 - 3 Python
- 2 <http://pypi.python.org/pypi/jenkinsapi>
- 3 <https://github.com/kellyrob99/Jenkins-api-tour>
- 4 <https://launchpad.net/python-jenkins>

Ilość API = zaleta czy wada?

Obecnie dostępnych ponad 300 pluginów.

Obecnie dostępnych ponad 300 pluginów.

- 1 Startowanie buildów

Obecnie dostępnych ponad 300 pluginów.

- 1 Startowanie buildów
- 2 Parametry

Obecnie dostępnych ponad 300 pluginów.

- 1 Startowanie buildów
- 2 Parametry
- 3 Artefaktowanie

Obecnie dostępnych ponad 300 pluginów.

- 1 Startowanie buildów
- 2 Parametry
- 3 Artefaktowanie
- 4 Raporty

Obecnie dostępnych ponad 300 pluginów.

- 1 Startowanie buildów
- 2 Parametry
- 3 Artefaktowanie
- 4 Raporty
- 5 Powiadomienia

Obecnie dostępnych ponad 300 pluginów.

- 1 Startowanie buildów
- 2 Parametry
- 3 Artefaktowanie
- 4 Raporty
- 5 Powiadomienia
- 6 Procesy promocji buildów

Obecnie dostępnych ponad 300 pluginów.

- 1 Startowanie buildów
- 2 Parametry
- 3 Artefaktowanie
- 4 Raporty
- 5 Powiadomienia
- 6 Procesy promocji buildów
- 7 Zarządzanie użytkownikami i rolami

Obecnie dostępnych ponad 300 pluginów.

- 1 Startowanie buildów
- 2 Parametry
- 3 Artefaktowanie
- 4 Raporty
- 5 Powiadomienia
- 6 Procesy promocji buildów
- 7 Zarządzanie użytkownikami i rolami

Przy wyborze i instalacji pluginów wymagana rozważa!

A co w przypadku, kiedy jedna maszyna to za mało?

- 1 Uruchamianie slave'ów łatwym sposobem

- 1 Uruchamianie slave'ów łatwym sposobem
- 2 Sprawne grupowanie

- 1 Uruchamianie slave'ów łatwym sposobem
- 2 Sprawne grupowanie
- 3 Jenkins po części monitoruje maszyny

- 1 Uruchamianie slaveów łatwym sposobem
- 2 Sprawne grupowanie
- 3 Jenkins po części monitoruje maszyny
- 4 Różnorodność systemowa slaveów

Takie rozwiązanie niesie za sobą kilka wad:

Takie rozwiązanie niesie za sobą kilka wad:

- 1 Algorytm rozdzielania zadań do slave'ów nie do końca dopracowany

Takie rozwiązanie niesie za sobą kilka wad:

- 1 Algorytm rozdzielania zadań do slave'ów nie do końca dopracowany
- 2 Przy dużej ilości projektów duża ilość informacji *przerzucana* przez sieć

Takie rozwiązanie niesie za sobą kilka wad:

- 1 Algorytm rozdzielania zadań do slave'ów nie do końca dopracowany
- 2 Przy dużej ilości projektów duża ilość informacji *przerzucana* przez sieć
- 3 Czasami stabilność...

Możliwości Jenkinsa (zalety):

Możliwości Jenkinsa (zalety):

- 1 Środowisko Continuous Integration

Możliwości Jenkinsa (zalety):

- 1 Środowisko Continuous Integration
- 2 Integracja z najpopularniejszymi systemami kontroli wersji

Możliwości Jenkinsa (zalety):

- 1 Środowisko Continuous Integration
- 2 Integracja z najpopularniejszymi systemami kontroli wersji
- 3 Dostępne API - możliwość automatyzacji utrzymania

Możliwości Jenkinsa (zalety):

- 1 Środowisko Continuous Integration
- 2 Integracja z najpopularniejszymi systemami kontroli wersji
- 3 Dostępne API - możliwość automatyzacji utrzymania
- 4 Budowanie rozproszone

Możliwości Jenkinsa (zalety):

- 1 Środowisko Continuous Integration
- 2 Integracja z najpopularniejszymi systemami kontroli wersji
- 3 Dostępne API - możliwość automatyzacji utrzymania
- 4 Budowanie rozproszone
- 5 Pluginy

Możliwości Jenkinsa (zalety):

- 1 Środowisko Continuous Integration
- 2 Integracja z najpopularniejszymi systemami kontroli wersji
- 3 Dostępne API - możliwość automatyzacji utrzymania
- 4 Budowanie rozproszone
- 5 Pluginy
- 6 Integracja z powiadomieniami

Możliwości Jenkinsa (zalety):

- 1 Środowisko Continuous Integration
- 2 Integracja z najpopularniejszymi systemami kontroli wersji
- 3 Dostępne API - możliwość automatyzacji utrzymania
- 4 Budowanie rozproszone
- 5 Pluginy
- 6 Integracja z powiadomieniami
- 7 LTS

Wady:

Wady:

- 1 Stabilność

Wady:

- 1 Stabilność
- 2 Niektóre pluginy niewspierane

Konkurencja

- 1 CruiseControl
- 2 BuildBot
- 3 Bamboo
- 4 TimeCity
- 5 Continuum

Warto?

Warto?

Oczywiście, że tak!

?