



Jenkins

Mariusz Pawłowski

Agenda

Agenda

- Czym jest Continuous Integration
- Instalacja i konfiguracja Jenkinsa
- Zadania

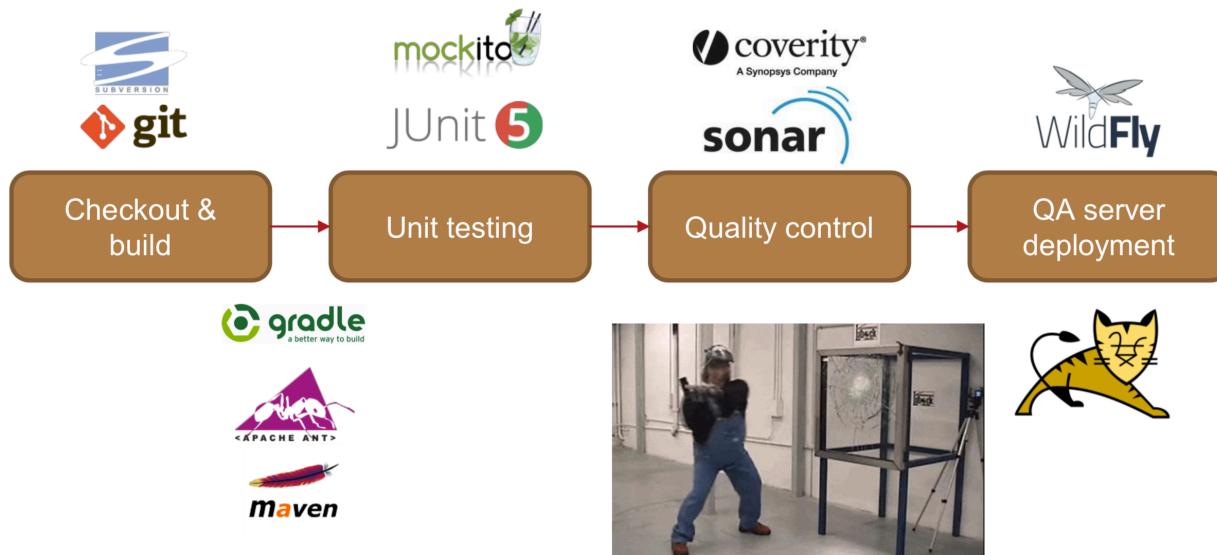
Continuous integration

Continuous integration

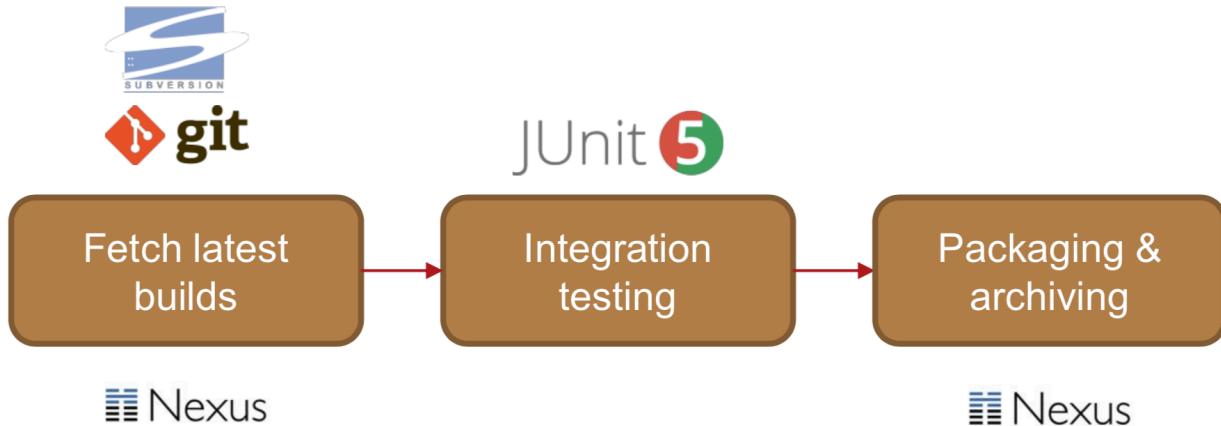
Praktyki stosowane w trakcie procesu wytwarzania oprogramowania polegające na automatyzacji procesów wytwarzania oprogramowania zawierającej regularnie wykonywane:

- Weryfikację zmian w repozytorium
- Analizę Clean Code
- Budowanie/Kompilację
- Instalowanie wersji w repozytorium artefaktów
- Deployowanie na maszynach testowych/stagingowych
- Wykonywanie testów jednostkowych/integracyjnych

Continuous integration



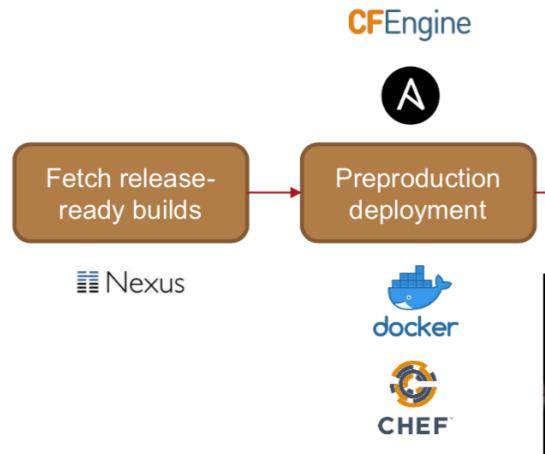
Continuous integration



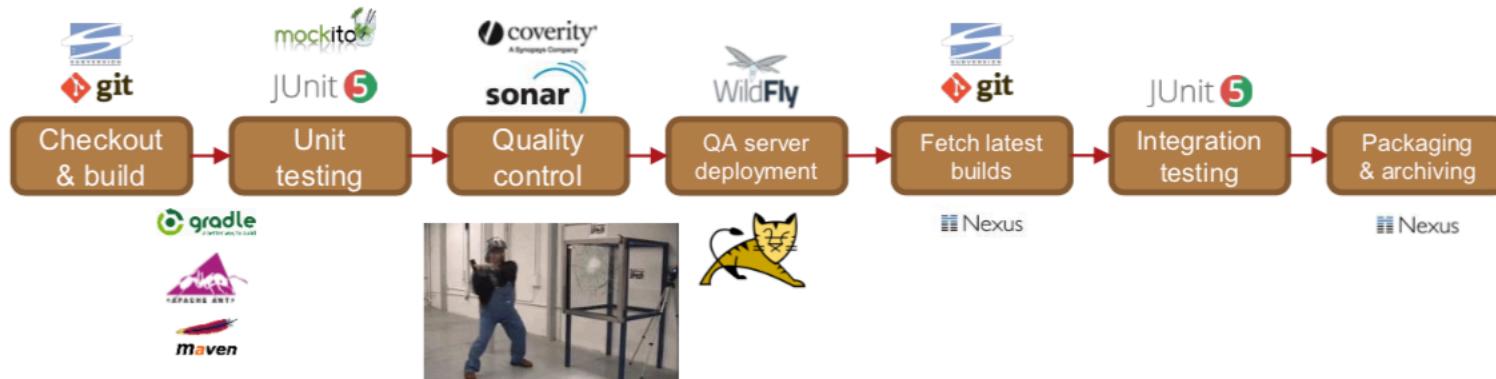
Continuous delivery

- Dyscyplina w procesie programowania. Zapewnienie stanu, w którym w dowolnym momencie oprogramowanie może zostać releasowane w dowolnej istniejącej wersji.
- Priorytety oraz podejście do realizacji zadań zespołu sprowadza się do utrzymania wersji deployowalnej.

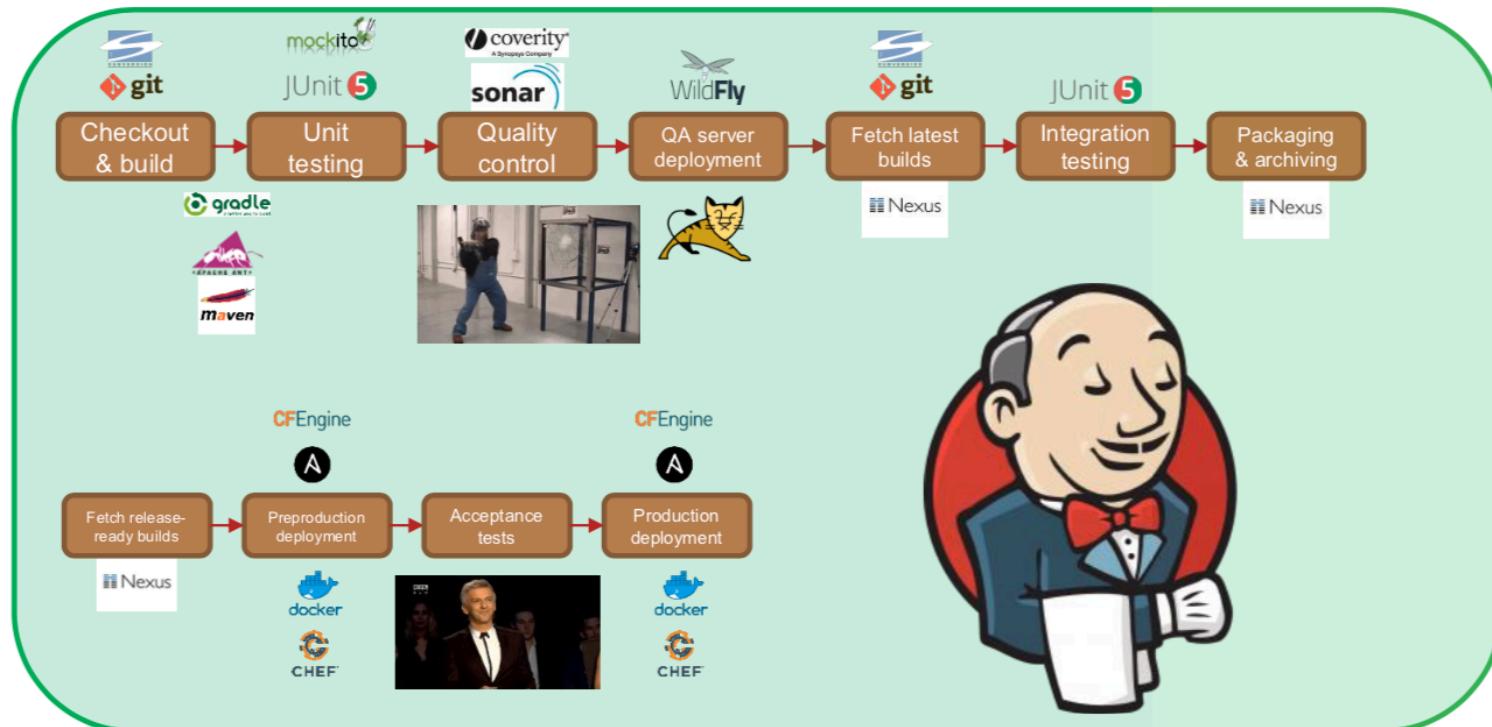
Continuous delivery



Automatyzacja procesu



Rola Jenkinsa



Cel warsztatów

- Przygotowanie niezależnych procesów CI oraz CD
- Aktualizacja kodu z repozytorium
- Zbudowanie aplikacji
- Wykonanie testów
- Wykonanie deploymentu
- Poinformowanie całego zespołu o statusie zadania wraz z informacją o ostatnim autorze felernego builda – budowanie odpowiedzialności zarówno za wytwarzany osobiście kod jak również odpowiedzialności zespołowej.

Jenkins

Czym jest Jenkins

- Narzędzie open-source służące do automatyzacji cyklu życia kodu wytwarzanego przez deweloperów.
- Architektura Jenkinsa opiera się o pokaźny zbiór pluginów, które można konfigurować wedle własnych potrzeb.

Korzyści z korzystania z Jenkinsa

- Możliwość kompilacji, integracji, uruchomienia aplikacji w warunkach podobnych do produkcyjnych
- Przyspieszenie procesów testów oraz weryfikacji poprawności kodu
- Utrzymanie spójnej formy aplikacji oraz poprawnej integracji wszystkich jej modułów

Zasada działania

Działanie Jenkinsa opiera się o pracę zadaniową (jobs)

- Job uruchamiany jest dla skonfigurowanego przez administratora (dewelopera) projektu
- Job może być uruchamiany ręcznie, automatycznie cyklicznie lub może być triggerowany na wystąpienie zdarzenia (np. commit)
- Projekty mogą być uzależnione od siebie, wówczas poprawne zakończenie jednego zadania uruchomi kolejne – zależne.

Zasada działania

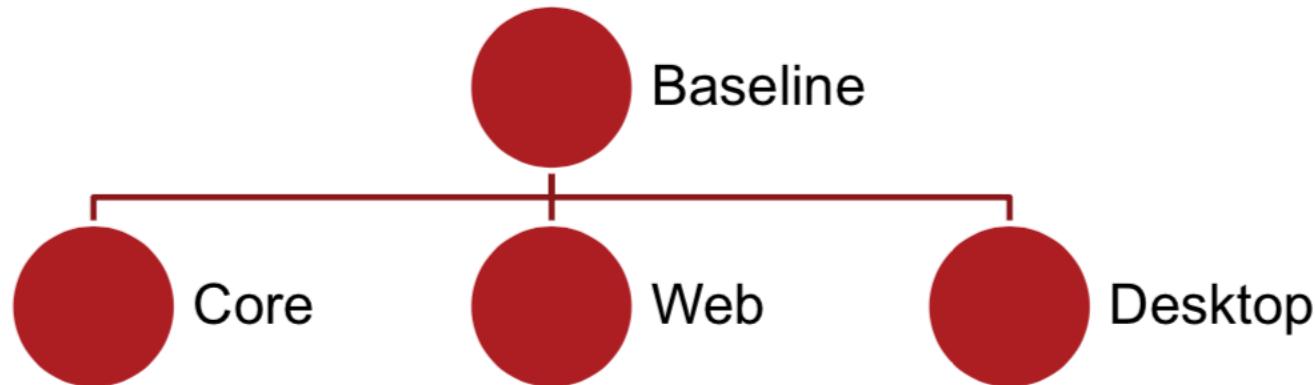
- Istnieją trzy statusy zadania (SUCCESS, UNSTABLE, FAILURE):

 **Zadanie #56 (2017-12-29 15:41:52)**

 **Zadanie #59 (2017-12-29 15:47:19)**

 **Zadanie #70 (2017-12-29 12:59:14)**

Przykładowa architektura



Instalacja Jenkinsa

- Istnieją różne sposoby instalacji Jenkinsa. Na potrzeby warsztatu wybierzemy ten najwygodniejszy – obraz dockerowy. Niestety, oficjalny obraz nie zawiera w sobie instalacji Mavena co jest nam potrzebne do budowania naszych projektów. W związku z tym, przygotujemy nowy obraz, na własne potrzeby w oparciu o ten oficjalny.

Instalacja Jenkinsa

Podstawowy, oficjalny obraz Jenkinsa:

```
#docker pull jenkins/jenkins  
  
#docker run --name jenkins -p 0.0.0.0:9090:8080 -p 0.0.0.0:50000:50000 -v /  
opt/jenkins:/var/jenkins_home jenkins/jenkins
```

Ale nie będziemy z niego korzystać bezpośrednio.

Instalacja Jenkinsa

- Przygotowanie obrazu dockerowego polega na stworzeniu pliku Dockerfile który zawiera instrukcje w jaki sposób budować obraz.
- Nas będzie interesowało dodanie instalacji Maven do już istniejącego obrazu Jenkinsa.

Instalacja Jenkinsa

Przejdź do katalogu domowego: #cd

Utwórz plik o nazwie Dockerfile i dodaj treść:

```
FROM jenkins/jenkins:lts
```

```
USER root
```

```
ENV JENKINS_OPTS --httpPort=1 --httpsPort=9090
```

```
RUN apt-get update && apt-get install -y maven
```

Instalacja Jenkinsa

W katalogu, gdzie istnieje nowy plik Dockerfile wykonujemy komendę budowania nowego obrazu o nazwie jenkins-maven:

```
#docker build -t jenkins-maven .
```

Sprawdzamy czy obraz istnieje:

```
#docker images
```

Instalacja Jenkinsa

Dla celów warsztatów wykorzystamy konfigurację Jenkinsa w dockerze.
Przygotowujemy dane i uruchamiamy aplikacje:

```
#mkdir -p /opt/jenkins  
#chmod 777 /opt/jenkins  
  
# docker run --name jenkins-maven  
-p 0.0.0.0:9090:9090 -p 0.0.0.0:50000:50000 -v /opt/jenkins:/var/jenkins_home  
-v /opt/wildfly:/opt/wildfly -d jenkins-maven
```

<https://localhost:9090>

Instalacja Jenkinsa

Dla celów warsztatów wykorzystamy konfigurację Jenkinsa w dockerze. Przygotowujemy dane uruchamiamy aplikację:

```
#mkdir -p /home/<username>/jenkins  
#chmod 777 /home/<username>/jenkins  
  
#mkdir -p /home/<username>/wildfly  
#chmod 777 /home/<username>/wildfly
```

```
# sudo docker run --name jenkins-maven -p 9090:9090 -p 50000:50000  
-v /home/<username>/jenkins:/var/jenkins_home  
-v /home/<username>/wildfly:/opt/wildfly jenkins-maven  
https://localhost:9090
```

Instalacja Jenkinsa

Zaczynamy

Odblokuj Jenkinsa

Aby zapewnić, że Jenkins jest bezpiecznie uruchomiony przez administratora, hasło zostało zapisane do pliku logów ([nie masz pewności, gdzie go znaleźć?](#)) oraz w pliku na serwerze:

`/var/jenkins_home/secrets/initialAdminPassword`

Skopiuj hasło z jednej z powyższych lokalizacji i wklej poniżej.

Hasło administratorskie:

Instalacja Jenkinsa

Zaczynamy

Dostosuj Jenkinsa

Wtyczki rozszerzające Jenkinsa o dodatkowe funkcjonalności, które zaspokajają wiele potrzeb.

Zainstaluj sugerowane wtyczki

Zainstaluj wtyczki, które społeczność Jenkinsa uznała za najbardziej przydatne.

Wybierz wtyczki do instalacji

Wybierz i zainstaluj wtyczki najbardziej dopasowane do Twoich potrzeb.

Instalacja Jenkinsa

- Nie zmieniamy domyślnej konfiguracji pluginów. Wybieramy po prostu „Instaluj”.
- Na koniec instalacji pluginów, wypełnij formularz dodawania głównego administratora systemu.
- Oraz wybierz „Zapisz i zakończ”
- Login i hasło zapamiętaj na cały czas trwania kursu!

Instalacja wtyczek

Instalacja wtyczek

Folders Plugin	OWASP Markup Formatter Plugin	Build Timeout	Credentials Binding Plugin	jQuery UI plugin
✓ Folders Plugin	✓ OWASP Markup Formatter Plugin	✓ Build Timeout	✓ Credentials Binding Plugin	** Jackson 2 API Plugin
✓ Timestamper	✓ Workspace Cleanup Plugin	✓ Ant Plugin	✓ Gradle Plugin	** JavaScript GUI Lib: ACE Editor
Pipeline	○ GitHub Branch Source Plugin	○ Pipeline: Git Step	✓ Pipeline: Stage View Plugin	** Pipeline: Groovy
○ Git plugin	○ Subversion Plug-in	○ SSH Slaves plugin	✓ Matrix Authorization Strategy Plugin	** Pipeline: Input Step
○ PAM Authentication plugin	○ LDAP Plugin	○ Email Extension Plugin	✓ Mailer Plugin	** Pipeline: Stage Step

jqGrid UI plugin
** Jackson 2 API Plugin
** JavaScript GUI Lib: ACE Editor
** Pipeline: Git Step
** Pipeline: Groovy
** Pipeline: Input Step
** Pipeline: Stage Step
** Pipeline: Stage View Plugin
** Pipeline: REST API Plugin
** Pipeline: Shared Script GUI Lib: Handlebars bundle plugin
** JavaScript GUI Lib: Moment.js bundle plugin
Pipeline: Stage View Plugin
** Pipeline: Build Step
** Pipeline: Model API
** Pipeline: Declarative Extended Pipeline API
** Jenkins Apache HttpClient Components Client 4.x API Plugin
** Jenkins Slaves Plugin
** Display URL API
Jenkins Mailer Plugin
Matrix Authorization Strategy Plugin
- Jenkins Jsch dependency plugin
** Jenkins Glt client plugin
** Jenkins Glt server Plugin
** Pipeline: Shared Groovy Libraries
** - zależności wymagane

Jenkins 2.101

Instalacja Jenkinsa

- Jeśli mamy problem z aplikacją Jenkinsa, jej działaniem możemy wymusić z endpointu restowego jego restart. Pierwszy zrestartuje aplikację ubijając wszystkie uruchomione zadania, drugi zrestartuje go dopiero po zakończeniu wszystkich trwających zadań.
- <https://domena:9090/restart>
<https://domena:9090/safeRestart>

Konfiguracja Jenkinsa

Jenkins

Najnowsze aktualizacje | Wykonawcy | Kolejka zadań | Zarządzaj Jenkinsem | Historia zadań | Użytkownicy | Nowy Projekt

Szukaj...

Zarządzaj Jenkinsem

 [Skonfiguruj system](#)
Konfiguruj ustawienia globalne i ścieżki.

 [Konfiguruj ustawienia bezpieczeństwa](#)
Zabezpiecz Jenkinsa: decyduj, kto ma do niego dostęp.

 [Configure Credentials](#)
Konfiguruj dostawcę i typów kredencjalnych.

 [Globalne narzędzia do konfiguracji](#)
Konfiguruj narzędzia, ścieżki do nich i automatyczne instalatory.

 [Odczytaj ponownie konfigurację z dysku](#)
Porzuć wszystkie dane załadowane w pamięci i załaduj wszystko z systemu plików. Opcja użyteczna gdy zmodyfikowano pliki konfiguracyjne bezpośrednio z dysku.

 [Zarządzaj wtyczkami](#)
Dodaj, usuń, włącz lub włącz wtyczki które mogą rozszerzyć funkcjonalność Jenkinsa.

 [Informacje o systemie](#)
Wyświetla wiele środowiskowych informacji pomocnych przy rozwiązywaniu problemów.

 [Dziennik systemowy](#)
Dziennik systemowy gromadzi wywoływanie `java.util.logging` powiązane z Jenkinsem.

Kolejka zadań

Nie ma zadań w kolejce

Status wykonawców zadań

1 Bezcenny
2 Bezcenny

Konfiguracja Jenkinsa



[Statystyki obciążenia](#)

Sprawdź obciążenie zasobów systemowych i dowiedz się, czy nie potrzebujesz więcej maszyn do uruchamiania zadań.



[Wiersz poleceń Jenkinsa](#)

Zarządzaj Jenkinsem poziomu z wiersza poleceń lub systemu.



[Konsola skryptów](#)

Wykonuje dowolny skrypt pomocny do celów administracyjnych, usuwania usterek i diagnostyki.



[Zarządzaj węzłami](#)

Dodawaj, usuwaj, kontroluj i monitoruj różne węzły, na których Jenkins uruchamia zadania.



[Informacje o Jenkinsie](#)

Sprawdź wersję i licencję



[Zarządzanie starymi danymi](#)

Usuwanie danych po nieużywanych lub starszych wersjach wtyczek.



[Zarządzaj użytkownikami](#)

Dodawaj, usuwaj i modyfikuj użytkowników, którzy mogą się logować do Jenkinsa



[Przygotuj do wyłączenia](#)

Zatrzyma wykonanie nowych buildów, tak by system mógłbyć bezpiecznie wyłączony.

Konfiguracja Jenkinsa

- **Dodanie pluginów**
- Jenkins->ZarządzanieJenkinsem
- Zarządzaj wtyczkami
- Dostępne
- Znajdź i zaznacz „Git Plugin”, „GitHub Plugin”, „Maven Integration plugin”
- Wybierz „Zainstaluj bez ponownego uruchamiania”
- Po zainstalowaniu zaznacz checkbox „Uruchom ponownie Jenkinsa, gdy instalacja się zakończy i żadne zadanie nie będzie wykonywane”
- Poczekaj na ponowne uruchomienie się Jenkinsa
- Jeśli pluginy nie są widoczne na liście „Dostępne” prawdopodobnie są już zainstalowane. Zweryfikuj to na liście „Zainstalowane”.

Konfiguracja Jenkinsa

- Globalna konfiguracja JDK
 - 1. Jenkins->Zarządzanie Jenkinsem
 - 2. Globalne narzędzia konfiguracji
 - 3. W sekcji JDK wybierz Dodaj
 - 4. Odznacz Install automatically
 - 5. Jako nazwę ustaw:Default
 - 6. Jako JAVA_HOME ustaw:/docker-java-home

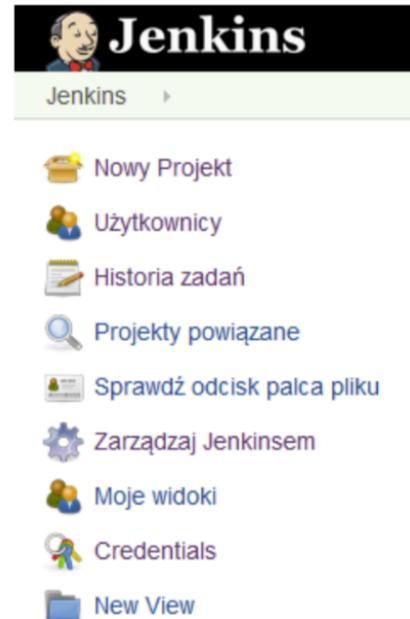
Konfiguracja Jenkinsa

- Globalna konfiguracja maven
 - 1. Jenkins->Zarządzanie Jenkinsem
 - 2. Globalne narzędzia konfiguracji
 - 3. W sekcji Maven wybierz Dodaj
 - 4. Odznacz Install automatically
 - 5. Jako nazwę ustaw:Default
 - 6. Jako MAVEN_HOME ustaw:/usr/share/maven

Jobs

Tworzenie nowego projektu

Aby utworzyć nowy
Jenkinsowy projekt
wystarczy wybrać z menu
po lewej „Nowy projekt”



Tworzenie nowego projektu

Następnie podać jego nazwę, wybrać typ, ewentualnie jeśli projekt jest podobny do już istniejącego możemy skopiować jego całą konfigurację.

Podaj nazwę projektu

Pole wymagane

Ogólny projekt
To jest podstawowa funkcja Jenkinsa. Jenkins stworzy projekt łączący dowolny SCM z dowolnym systemem budującym, może to być również wykorzystane do czegoś innego niż budowanie oprogramowania.

Maven project
Stwórz projekt maven. Jenkins wykorzysta twoje pliki POM by znacznie zredukować konfigurację.

Pipeline
Orchestrates long-running activities that can span multiple build slaves. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

Github Organization
Scans a GitHub organization (or user account) for all repositories matching some defined markers.

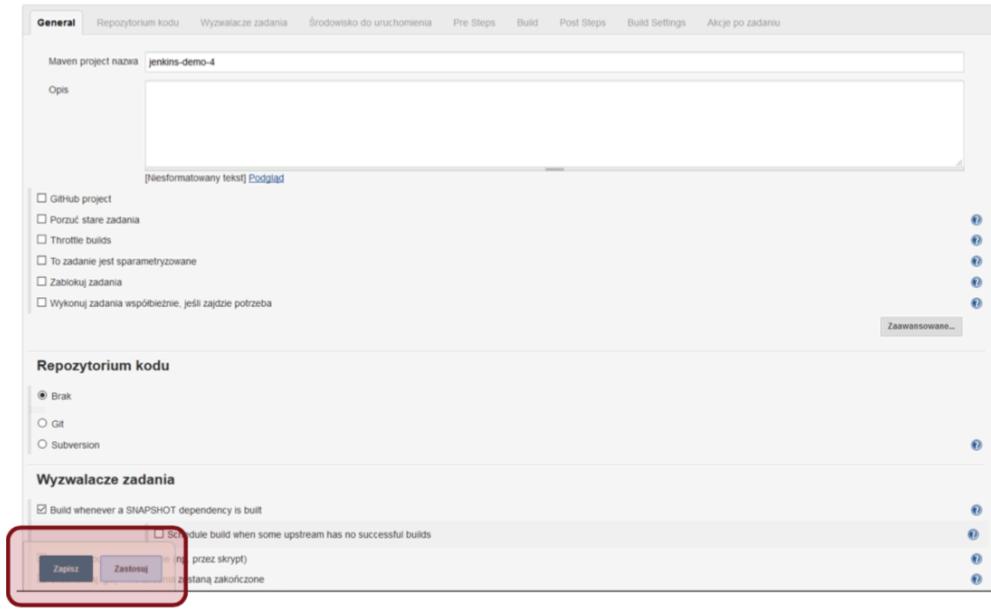
Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

Jeśli chcesz stworzyć nowy projekt na podstawie istniejącego, możesz użyć tej opcji:

 Kopiuj z Podaj nazwę

Tworzenie nowego projektu

W trzecim kroku można dokonać szczegółowej konfiguracji projektu. Zapis odbywa się przez przycisku w dolnym panelu widoku.



Tworzenie nowego projektu

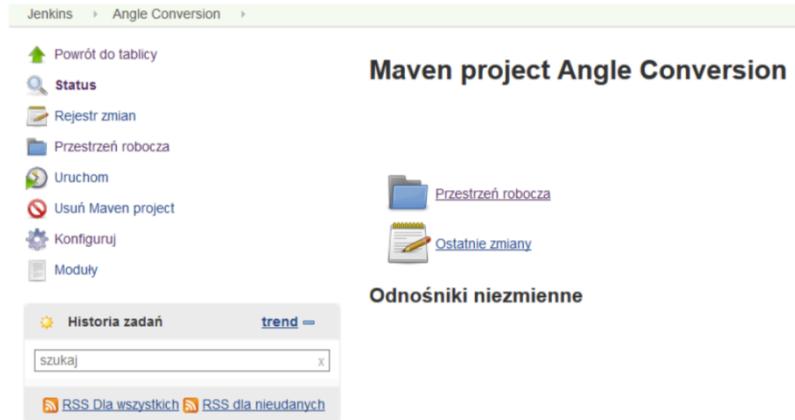
- Dodaj nowy projekt **Maven** o nazwie Angle Conversion
- Pomiń konfigurację trzeciego kroku, zapisując projekt w bazie Jenkinsa.
- Przejdz nadashboard(wybierz głównego Jenkinsa). Twój projekt powinien pojawić się na liście.

Tworzenie nowego projektu

S	P	Nazwa ↓	Ostatni sukces	Ostatni błąd	Czas trwania	
		Angle Conversion	—	—	nd.	

Tworzenie nowego projektu

Menu projektu jest widoczne po lewej stronie. Jest to miejsce, z którego możemy przejść do dalszej konfiguracji projektu, przejrzeć zmiany, buildy oraz przestrzeń roboczą. Istnieje możliwość również skasowania projektu.



The screenshot shows the Jenkins interface for the 'Angle Conversion' Maven project. On the left, there is a sidebar with the following options:

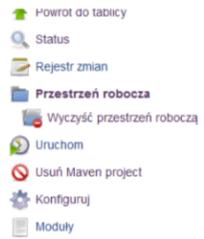
- Powrót do tablicy
- Status
- Rejestr zmian
- Przestrzeń robocza
- Uruchom
- Usuń Maven project
- Konfiguruj
- Moduły

Below the sidebar, there is a search bar labeled 'szukaj' and two RSS feed links: 'RSS Dla wszystkich' and 'RSS dla nieudanych'. To the right of the sidebar, there is a section titled 'Maven project Angle Conversion' with a link to 'Przestrzeń robocza' and another link to 'Ostatnie zmiany'. At the bottom, there is a section titled 'Odnośniki niezmienne'.

Tworzenie nowego projektu

Dla nowego projektu przestrzeń robocza nie istnieje. Aby ją utworzyć należy wykonać pierwszy build.

Zakończy się on porażką jednak utworzy przestrzeń roboczą.



Error: no workspace

There's no workspace for this project. Possible reasons are:

1. The project was renamed recently and no build was done under the new name.
2. The agent this project has run on for the last time was removed.
3. The workspace directory (`/var/jenkins_home/workspace/Angle Conversion`) is removed outside Jenkins.
4. The workspace was wiped out and no build has been done since then.

Run a build to have Jenkins create a workspace.

Przestrzeń robocza Angle Conversion na master



Brak plików w katalogu

Tworzenie nowego projektu

- Każdy projekt Jenkinsie posiada odpowiadający mu katalog na dysku. Wszystkie przestrzenie robocze znajdują się na hoście: workspace/
- Projekt zapisuje konfigurację na dysku. Wszystkie tworzone pliki, checkoutowane projekty zostaną zapisane w katalogu projektu.
- Uruchom projekt Angle Conversion w celu utworzenia przestrzeni roboczej.

Przygotowanie repozytorium kodu

- W repozytorium:
<https://github.com/infoshareacademy/jdqz1-jenkins>
- istnieje branch: **mp**
Proszę o utworzenie własnych imiennych branchy pochodzących od brancha **mp**. Od tej pory proszę pracować ze swoim branchem!
- `#git clone [url]`
`#git checkout mp`
`#git checkout -b <initials>`
`#git push --set-upstream origin <initials>`

Konfiguracja istniejącego projektu

Aby zmienić konfigurację istniejącego projektu, należy przejść do jego przestrzeni. Z menu po lewej stronie wybrać: Konfiguruj



Konfiguracja istniejącego projektu - sekcja general

- Github project – konfiguracja spięcia Githuba z Jenkinsem
- Porzuć stare zadania – opcja weryfikuje czy stare zadania powinny być kasowane (w celu zwolnienia miejsca na dysku). Konfiguracja pozwala na ustalenie limitu zadań wg ich liczby oraz wieku. Nie ustawiamy żadnych wartości.
- Throttle builds – pilnuje liczby zadań w danej jednostce czasu.
- To zadanie jest sparametryzowane – oznaczenie tej opcji wymusi na użytkowniku podanie wartości parametrów zanim job zostanie uruchomiony. Pozostawiamy wyłączone.

Konfiguracja istniejącego projektu - sekcja general

- Zablokuj zadania – czasowe wyłączenie możliwości aktywacji zadania. Np. z powodu czasowej niedostępności repozytorium kodu.
- Wykonuj zadania współbieżnie, jeśli zajdzie potrzeba – pozwala na równoległe wykonywanie zadań. Pozostawiamy wyłączone.
- Cichy okres – opóźnienie uruchomienia zadania.
- Liczba ponowień – liczba odwołań do repozytorium kodu w przypadku problemów z dostępem.
- Block build when upstream project is building – czeka z uruchomieniem zadania do chwili aż wszystkie nadzędne projekty zostaną zakończone.

Konfiguracja istniejącego projektu - sekcja general

- Block build when downstream project is building - czeka z uruchomieniem zadania do chwili aż wszystkie podrzędne projekty zostaną zakończone.
- Nazwa wyświetlana – nazwa wykorzystywana tylko do prezentacji projektu w GUI Jenkinsa
- Trzymaj logi konsoli projektów zależnych

Konfiguracja istniejącego projektu - sekcja general

- Skonfiguruj URL do projektu GitHub (GitHub project/Project url) na: <https://github.com/infoshareacademy/jdqz1-jenkins>
- Skonfiguruj projekt tak aby pozostawał tylko 10 ostatnich zadań.

Konfiguracja istniejącego projektu

Dla opcji GIT:

- Repository URL – URL do projektu w repozytorium
- Credentials – dane dostępowe do repozytorium. Wybierając domyślnego providera jako Jenkinsa mamy możliwość bezpiecznie przechowywać i używać danych w komunikacji z repozytorium
- Branch Specifier – definicja branch'a, z którego mają odbywać się checkouty aktualizujące kod - */<branch_name>
- Repository browser – wybieramy domyślną przeglądarkę repozytorium, zgodną z używanym przez nas systemem

Konfiguracja istniejącego projektu

Dla opcji GIT:

- Zdefiniuj odwołanie do repozytorium kodu na: <https://github.com/infoshareacademy/jdqz1-jenkins>
- Użyj swojego brancha stworzonego w jednym z poprzednich zadań
- Stwórz pakiet danych dostępowych z wykorzystaniem swojego użytkownika i hasła do githuba
- Wybierz odpowiedni repository browser
Zapisz zmiany oraz uruchom zadanie.

Konfiguracja istniejącego projektu

Wyzwalacze

- Konfiguracja dotycząca startu nowego zadania. Dla przykładu wyzwalaczem może być nowy commit (wymaga szerszej konfiguracji również po stronie repozytorium) lub na przykład build może odbywać się cyklicznie.

Cykl definiujemy wg harmonogramu. Opis jak to robić znajdziemy w pomocy kontekstowej Jenkinsa.

Jeśli job ma się odpalać, np. co 15 minut, definicją będzie:
H/15 * * * *

Konfiguracja istniejącego projektu

Wyzwacze

- Skonfiguruj automatyczne, cykliczne wywołanie zadania co 5 minut.



Dziękuję!

Pytania?