

Dokument roboczy projekt ZSI Moduł 4

Arkadiusz Kałuża, Adam Kierat

Przygotowanie dokumentu roboczego zawierającego zapisy dotyczące punktów wyróżnionych we wzorze dokumentu roboczego dostępnego na platformie zdalnej edukacji.

- **A1. Ustanowienie współdzielonego repozytorium**

Ustanowienie współdzielonego repozytorium na platformie github.com Link:
<https://github.com/arcziko9/ZarzadzanieSystemamiInformatycznymiPOLSL>

- **B1. Ustalenie z prowadzącym tematu projektu.**

Sekcja wybrała projekt numer 4.4:

a. Automatyzacja w procesie wytwarzania oprogramowania Jenkins

- **B3. Przygotowanie notatki z ASD: 3 pozycje, 3 najważniejsze wyróżniki, linki do witryn zawierających szczegóły.**

- a. Jenkins jest serwerem/systemem do automatyzacji zadań z obszarów rozwoju oprogramowania takich jak Continuous Integration i Continuous Delivery nie wymagających zaangażowania człowieka. Samo oprogramowanie Jenkins jest rozwijane jako Open Source i wydawane na licencji MIT. Pierwsza wersja wydana została w 2005 roku pod nazwą Hudson i rozpoczęła rewolucję w dziedzinie zwinnego prowadzenia projektów programistycznych.

Przydatne linki:

- * <https://ujeb.se/e5QRA>
- * <https://ujeb.se/66VgS>
- * <https://ujeb.se/Y3Ilk>

- **C1.Opracowanie koncepcji realizacji projektu.**

- Jenkins początkowo został zaprojektowany do celów Continuous Integration dla projektów pisanych w języku Java, ale jego uniwersalność pozwoliła szybko rozszerzyć możliwe zastosowania na inne języki programowania oraz automatyzację innych zadań.

Wszechstronne zastosowania Jenkinsa:

- * Pobieranie kodu źródłowego z dowolnego systemu kontroli wersji, np. AccuRev, CVS, Subversion, Git, Mercurial, Perforce, TD/OMS, ClearCase, RTC.
- * Wyzwalanie zadania za pomocą narzędzi wspierających budowanie oprogramowania, np. Apache Ant, Apache Maven, sbt, jak również natywne skrypty powłoki, np. Linux sh, bash, csh, zsh, Windows batch commands, PowerShell.

- * Zadania mogą być wyzwalane ręcznie, skrypcem, przez zmianę w repozytorium kodu, słowo kluczowe umieszczone w komentarzu do zmiany w repozytorium kodu, przez zintegrowany system zewnętrzny lub zdarzenie zakończenia inne go zadania.
- * Zadanie może składać się z wielu kroków, które mogą obejmować takie aspekty budowania oprogramowania jak: kompilowanie, linkowanie, lintowanie, testowanie automatyczne, analiza statyczna kodu, migracja baz danych, deployment aplikacji serwerowych, publikacja aplikacji klienckich, etc.
- * Za pomocą pluginów możliwa jest prezentacja wyników wykonywanych zadań w postaci raportów WWW, dzięki czemu oprócz zero-jedynkowej odpowiedzi na pytanie “czy zadanie się powiodło?” Jenkins pozwala na pełen przegląd stanu projektu, np. za pomocą raportów błędów, problemów z jakością kodu, pokrycia kodu testami, złożoności obliczeniowej kodu, wydajności, etc.

Instalacja na dowolnej platformie

Serwer Jenkins można zainstalować na dowolnej platformie i systemie operacyjnym, które obsługują Java Runtime Environment (JRE), które to środowisko jest jedynym wymaganiem komponentem do pracy serwera. Oczywiście, w zależności od tego jakie zadania mają być realizowane przez serwer Jenkins, inne narzędzia również mogą być potrzebne, np. git do pobierania kodu źródłowego z repozytorium, czy Apache Ant do wykonywania zadań budowania.

Architektura Master-Slave

Serwer Jenkins pracuje w architekturze Master-Slave. Oznacza to, że jest jeden (lub więcej) serwer Master który, rozdysponowuje zadania pomiędzy agentów Slave. W najprostszej instalacji zarówno serwer jak i agenci znajdują się na tej samej maszynie. Jest to dobre rozwiązanie dla małych zespołów wytwarzających aplikację dla pojedynczej platformy docelowej.

W bardziej wymagających wdrożeniach, gdy mamy do czynienia z większymi zespołami lub bardziej rozbudowanymi projektami, w szczególności gdy oprogramowanie powstaje na różne platformy, różne systemy operacyjne, wtedy na osobnej maszynie instalowany jest serwer Jenkins, a na osobnych maszynach wyposażonych w docelowe platformy i systemy operacyjne instalowani są agenci Slave.

Wbudowany serwer HTTP

Serwer Jenkins wyposażony jest w wbudowany serwer HTTP obsługujący interfejs WWW oraz REST API, możliwa jest instalacja na dowolnym systemie operacyjnym obsługującym stos TCP/IP. Ze względów bezpieczeństwa jednak, mocno zalecane jest skonfigurowanie serwera Jenkins za sprawdzonym serwerem Reverse Proxy takim jak Apache czy Nginx. W tym celu polecana jest instalacja na dowolnej dystrybucji systemu Linux co zapewni najbezpieczniejszą i najstabilniejszą konfigurację.

Jenkins w kontenerze Docker

Dostępny jest również kontener Docker systemu Jenkins do instalacji i uruchomienia serwera w środowisku skonteneryzowanym za pomocą dockerów. Rozwiązanie to polecane jest do testów, a w przyszłości być może również do produkcyjnego zastosowania.

Konfiguracja zadań

Obecnie zadania w systemie Jenkins można skonfigurować na dwa podstawowe sposoby. Przy czym każdy z nich można użyć w innym scenariuszu działania. W pierwszym z nich konfiguracja odbywa się w sposób klasyczny za pomocą interfejsu WWW w którym wybieramy i konfigurujemy 3 aspekty projektu-zadania.

Przygotowanie projektu,

czyli skąd i jaki kod źródłowy pobrać, jak przygotować środowisko do budowy oprogramowania, w jaki sposób zadanie jest wyzwalane i w jaki sposób przechowywane są wyniki zadania.

Budowanie,

czyli właściwa część zadania zawiera konfigurację kolejnych kroków w procesie budowania aplikacji na podstawie pobranego wcześniej kodu źródłowego.

Zadania po zakończeniu budowania.

Jest to lista zadań które zostaną wykonane w zależności od tego czy proces budowania powiódł się, czy też nie. Należą do nich takie typowe czynności jak: wysłanie powiadomienia, upload paczki na serwer, wygenerowanie raportów. Również w tym miejscu można skonfigurować czy i jakie kolejne projekty-zadania mają zostać uruchomione.

Pipeline

Drugi sposób konfiguracji nazwany Pipeline obsługuje 2. i 3. aspekt konfiguracji w sposób bliższy programistom, a mianowicie za pomocą pliku tekstowego umieszczonego w repozytorium. W tym przypadku za pomocą interfejsu WWW projekt konfigurowany jest tylko wstępnie, natomiast główna konfiguracja znajduje się w pliku tekstowym w repozytorium kodu razem z kodem budowanej aplikacji. Dzięki temu rozwiązaniu uzyskano kontrolę i wersjonowanie zmian w procesie budowania aplikacji, a jednocześnie sędowano odpowiedzialność za proces budowania oprogramowania na programistów, którzy je wytwarzają.

Podsumowanie

Serwer automatyzacji Jenkins pozwala w istotny sposób usprawnić prawdopodobnie każdy proces wytwarzania oprogramowania. Dzięki swojej mnogości funkcji i bogatej bazie pluginów jest kompletnie niezależny od procesu, języka programowania, platformy sprzętowej czy systemu operacyjnego. Możliwości konfiguracji i dostępność pluginów w pierwszej chwili może przytłaczać i faktycznie wdrożenie serwera Jenkins zazwyczaj wymaga poświęcenia odpowiedniej ilości czasu i zaangażowania zespołu. Z drugiej strony, jest to najbardziej rozbudowany i uniwersalny system na rynku wspierający programistów w ich codziennej pracy i usprawniający proces i jakość wytwarzanego oprogramowania.

System pluginów

Przegląd funkcjonalności rozszerzanych przez pluginy

- Pobieranie kodu źródłowego – podstawowym systemem wersjonowania kodu systemu Jenkins jest git, ale dzięki pluginom możliwe jest pobranie kodu z praktycznie każdego systemu CVS.
- Uprawnienia – domyślnie Jenkins posiada prosty system uprawnień opierający się na zwykłej autentykacji użytkowników i autoryzacji kilku ról. Natomiast za pomocą pluginów możliwe jest rozbudowanie autentykacji o integrację z innymi dostawcami, np. OAuth, LDAP, PAM oraz rozbudowanie systemu uprawnień do bardzo szczegółowego.
- Kontrola zdalnych agentów – domyślnie Jenkins zarządza zdalnymi agentami za pomocą specjalnej aplikacji na dedykowanym porcie, ale dostępne są pluginy umożliwiające kontrolę również nad systemami zdalnymi które nie są wyposażone w agenta Jenkins lub z gdy przestał on działać, np. za pomocą SSH.
- Odczyt i interpretacja komunikatów z narzędzi – jak już wcześniej wspomniano, Jenkins potrafi interpretować wyniki z uruchamianych narzędzi i prezentować je w postaci raportów za pomocą interfejsu WWW.
- Odczyt i interpretacja komentarzy zawartych w kodzie – specjalnie na potrzeby developerów Java opracowano plugin generujący dokumentację na podstawie komentarzy Javadoc.
- Powiadomienia – Jenkins dysponuje również bogatą listą pluginów wysyłających powiadomienia za pomocą maila, wywołując zdalne zapytanie REST API, lub wysyłając wiadomość za pomocą komunikatora.
- Integracja z zewnętrznymi aplikacjami – oprócz możliwości wywołania zapytania REST API zewnętrznej aplikacji, najczęściej używane przez programistów aplikacje i serwisy doczekały się dedykowanych pluginów wspierających integrację, np. GitHub, GitLab, BitBucket, JIRA, Bugzilla, etc.