

## Program szkolenia:

# Jenkins - Continuous Delivery pipelines

### Informacje:

<b>Nazwa:</b>	<b>Jenkins - Continuous Delivery pipelines</b>
<b>Kod:</b>	<b>Jenkins-CI</b>
<b>Kategoria:</b>	DevOps i narzędzia
<b>Odbiorcy:</b>	testerzy, admini, developerzy, DevOps
<b>Czas trwania:</b>	2-3 dni
<b>Forma:</b>	50% wykłady / 50% warsztaty

---

Nauż się tworzyć potoki wdrożeniowe (delivery pipelines) z wykorzystaniem Jenkins Pipelines. Dowiedz się, jak zbudować skalowalne środowisko CI/CD oparte o Jenkinsa.

W myśl zasady *automatyzujemy tylko to co potrafimy zrobić manualnie* szkolenie rozpoczniemy od manualnego stworzenia jobów i widoków w Jenksie aby następnie stopniowo przekształcić je we w pełni zautomatyzowany potok wdrożeniowy zapisany w kodzie.

Z każdym kolejnym krokiem będziemy poznawać i wykorzystywać coraz bardziej zaawansowane mechanizmy Jenkinsa i Jenkins Pipelines.

Po szkoleniu będziesz potrafił/a m.in.:

- zaprojektować i zapisać w kodzie (pipeline as code) potok wdrożeniowy and gt; (delivery pipeline)
- stworzyć, opublikować i użyć bibliotek/i współdzielonej aby ułatwić zarządzanie potokami wdrożeniowymi na poziomie Twojej organizacji
- skalować infrastrukturę CI/CD z wykorzystaniem rozproszonej architektury Jenkinsa
- poruszać się po ekosystemie Jenkinsa i dobierać rozwiązania oraz narzędzia odpowiednie do wymagań, które chcesz zrealizować
- stosować najlepsze praktyki w projektowaniu i implementacji potoków wdrożeniowych oraz unikać powszechnych błędów i pułapek

### Zalety szkolenia:

- holistyczne podejście do tematu tworzenia potoków wdrożeniowych (delivery pipelines)
- skupienie na praktycznych aspektach CI/CD
- przekrojowe ćwiczenia praktyczne
- zalecane wzorce i praktyki

## Szczegółowy program:

### 1. Wprowadzenie do pracy z Jenkinsem (aby bez problemu zanurzyć się w automatyzację potoków wdrożeniowych)

1.1. job, pipeline i widoki

1.2. przestrzeń robocza

1.3. pluginy

1.4. konfiguracja Jenkinsa

### 2. Ekosystem Jenkinsa (aby sprawnie i bezpiecznie korzystać z Jenkinsa)

### 3. Continuous Integration, Continuous Delivery, Continuous Deployment w teorii i praktyce (aby mieć bazę pod tworzenie i automatyzację potoków)

### 4. Jenkins Pipelines (aby zastąpić manualnie tworzone i rozwijane potoki wdrożeniowe zautomatyzowanym rozwiązaniem zgodnym z podejściem pipeline as code)

4.1. różnice i podobieństwa między podejściem deklaratywnym i podejściem skryptowym

4.2. tworzenie potoków wdrożeniowych w podejściu deklaratywnym

4.3. ekosystem Jenkins Pipelines

4.4. kompozycja potoku wdrożeniowego

4.5. manualna promocja potoku wdrożeniowego

4.6. Jenkinsfile i podejście pipeline as code

4.7. biblioteki współdzielone

4.8. najlepsze praktyki

### 5. Rozproszona architektura Jenkinsa (aby skalować infrastrukturę CI/CD i umożliwić równoległe wykonywanie wielu potoków wdrożeniowych)

5.1. Jenkins Controller

5.2. egzekutory, węzły i agenty

5.3. etykiety

5.4. agenty chmurowe

## 6. Automatyzacja i projektowanie potoków wdrożeniowych (aby samodzielnie projektować i tworzyć potoki wdrożeniowe)

6.1. projektowanie potoków wdrożeniowych

6.2. optymalizacja potoków wdrożeniowych

6.3. migracja istniejących potoków wdrożeniowych do podejścia pipeline as code

6.4. najlepsze praktyki