



## Kompleksowe wprowadzenie do DevOps - szkolenie w formie warsztatowej

Numer usługi 2026/02/14/202247/3335107

5 535,00 PLN brutto  
4 500,00 PLN netto  
158,14 PLN brutto/h  
128,57 PLN netto/h  
332,00 PLN cena rynkowa ⓘ

JSYSTEMS SPÓŁKA  
Z OGRANICZONĄ  
ODPOWIEDZIALNOŚ  
CIĄ

Brak ocen dla tego dostawcy

- 📄 Usługa szkoleniowa
- 📺 zdalna w czasie rzeczywistym
- 🕒 35:00 h
- 📅 14.12.2026 do 18.12.2026

## Informacje podstawowe

<b>Kategoria</b>	Informatyka i telekomunikacja / Administracja IT i systemy komputerowe
<b>Grupa docelowa usługi</b>	Skierowane do początkujących DevOpsów, administratorów systemów, programistów oraz osób zainteresowanych automatyzacją, konteneryzacją i wdrażaniem aplikacji w nowoczesnych środowiskach.
<b>Minimalna liczba uczestników</b>	6
<b>Maksymalna liczba uczestników</b>	15
<b>Data zakończenia rekrutacji</b>	10-12-2026
<b>Forma prowadzenia usługi</b>	zdalna w czasie rzeczywistym
<b>Liczba godzin usługi</b>	35
<b>Podstawa uzyskania wpisu do BUR</b>	Certyfikat systemu zarządzania jakością wg. ISO 9001:2015 (PN-EN ISO 9001:2015) - w zakresie usług szkoleniowych

## Cel

### Cel edukacyjny

Po ukończeniu tego szkolenia uczestnik będzie potrafił:

- wykorzystywać skrypty shellowe na przykładzie basha,
- zagadnienia wstępne z zakresu GIT i Ansible, Dockera - tworzenia własnych obrazów,
- repozytorium obrazów, Docker Compose, Kubernetesa, logowania, monitoringu,
- koncepcje pracy z wieloma środowiskami i dobrych praktyk.

## Efekty uczenia się oraz kryteria weryfikacji ich osiągnięcia i Metody walidacji

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
<p>Po ukończeniu tego szkolenia uczestnik będzie potrafił:</p> <ul style="list-style-type: none"><li>• wykorzystywać skrypty shellowe na przykładzie basha,</li><li>• zagadnienia wstępne z zakresu GIT i Ansible, Dockera - tworzenia własnych obrazów,</li><li>• repozytorium obrazów, Docker Compose, Kubernetesa, logowania, monitoringu,</li><li>• koncepcje pracy z wieloma środowiskami i dobrych praktyk.</li></ul>	<p>Każde omawiane zagadnienie jest utrwalane rzędem ćwiczeń o wzrastającym poziomie trudności. Podczas szkolenia uczestnicy wykonują ponad 50 warsztatów. W szkoleniu duży nacisk jest kładziony na dobre zrozumienie zasad działania wykorzystywanych technik, a nie ich odtwórcze stosowanie.</p>	<p>Obserwacja w warunkach rzeczywistych</p>

## Kwalifikacje

### Kompetencje

Usługa prowadzi do nabycia kompetencji.

### Warunki uznania kompetencji

Pytanie 1. Czy dokument potwierdzający uzyskanie kompetencji zawiera opis efektów uczenia się?

TAK

Pytanie 2. Czy dokument potwierdza, że walidacja została przeprowadzona w oparciu o zdefiniowane w efektach uczenia się kryteria ich weryfikacji?

TAK

Pytanie 3. Czy dokument potwierdza zastosowanie rozwiązań zapewniających rozdzielenie procesów kształcenia i szkolenia od walidacji?

TAK

## Program

### 1. Wprowadzenie

### 2. Po co DEVOPS... Kto to DEVOPS?

o Kultura czy funkcja?

o Po co devops?

### 3. GIT

o Przypomnienie i podstawy

o Branching - dlaczego jest ważny

o Gitflow - bez tego ani rusz - po co z tego korzystamy

#### **4. Skrypty shellowe na przykładzie basha**

o Zmienne

o Pętle

o Instrukcje warunkowe - if then else

o Obsługa błędów

o Funkcje

o Operator case

o Interfejs graficzny w skryptach

o Przykładowy skrypt do robienia backupu - rollbacku

#### **5. Ansible - wstęp do Ansible**

o Czym jest Ansible

o Sposoby instalacji Ansible

o Praca z modułami uruchamianymi z linii komend

o Tworzenie pierwszych playbook-ów

o Role

o Przygotowanie od zera deploymentu aplikacji - tworzenie prostego playbook'a do instalacji i uruchomienia aplikacji

#### **6. REST API**

o Wprowadzenie do REST API

o Metody autoryzacji

o Jak wykorzystywać REST API praktycznie z konsoli

#### **7. Docker – co to jest, gdzie pływa i gdzie dokuje?**

o Instalacja i konfiguracja

o Docker images – czym są obrazy, tworzenie i obsługa

o UnionFs

o Uruchomienie Hello World!

o Kontenery – tworzenie, współdzielenie, uruchamianie i podział warstw

#### **8. Docker - tworzenie własnych obrazów**

o Budujemy własne obrazy i poznajemy Dockerfile

o ENTRYPOINT vs CMD - jaka jest różnica i po co się to stosuje

o .dockerignore - co jeżeli nie chcemy mieć niektórych rzeczy

o Multistage build - staramy się trzymać małe obrazy

o Optymalizacja warstw w Docker - jak budować dobre obrazy?

#### **9. Registry - czyli repozytorium obrazów**

o Uruchamiamy lokalne registry

o Zabezpieczamy dostęp do niego

o Docker Hub – ściąganie i wysyłanie obrazów

o Chmura czy lokalne registry (jakie są rozwiązania i które będzie dla mnie najlepsze)

## 10. Security i inne

o Bezpieczeństwo kontenerów i skanowanie pod kątem bezpieczeństwa.

## 11. Docker Compose

o Wprowadzenie i instalacja

o Konfiguracja i format YAML

## 12. Kubernetes

o Podstawy architektury

☒ API

☒ Opis obiektów

o Omówienie kubectl i podobieństwa z docker

☒ Wejście w POD i kontener

☒ Zarządzanie kontekstem

o Repliki, skalowanie a deployment wersji

o Eksponowanie usług

☒ NodePort

☒ ClusterIP

☒ Ingress

o Konfiguracja i współdzielenie informacji

☒ Sekrety

☒ ConfigMaps

## 13. Portainer – graficzny interfejs użytkownika - co to jest i czy warto

## 14. Logowanie

o Centralne logowanie na przykładzie rsyslog

o Centralne logowanie z wykorzystaniem Elastic Stack

## 15. Monitoring

o Dobre praktyki monitoringu serwerów jak i aplikacji

o Przegląd aplikacji do monitoringu

o Instalacja oraz konfiguracja Zabbixa

o Instalacja oraz konfiguracja Prometheusa

o Grafana - one dashboard to rule them all

## 16. Koncepcje pracy z wieloma środowiskami

o dev, test, QA, UAT, PROD - jak to wszystko połączyć i utrzymać!

## 17. Dobre praktyki - 12 factor app

o Przegląd dobrych praktyk przy tworzeniu aplikacji

o Przykłady złych aplikacji

## 18. CI/CD

o Jak robić Code Review mikroserwisów

o Ułożenie Continuous Integration pipeline - na przykładzie Jenkinsa

# Harmonogram

Liczba pozycji harmonogramu: 0

Przedmiot / temat	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
Brak wyników.					

# Cennik

## Cennik

Rodzaj ceny	Cena
Koszt przypadający na 1 uczestnika brutto	5 535,00 PLN
Koszt przypadający na 1 uczestnika netto	4 500,00 PLN
Koszt osobogodziny brutto	158,14 PLN
Koszt osobogodziny netto	128,57 PLN

# Prowadzący

Liczba prowadzących: 1



1 z 1

## Maciej Krajewski

Pasjonat kultury DEVOPS, zaprawiony w bojach programista JAVA i DEVOPS od 15 lat (jeszcze zanim to było modne). Ma wieloletnie doświadczenie w zarządzaniu wszelorakimi zespołami developerskimi. Zjadł żęby na realizacji projektów informatycznych w metodologiach klasycznych i zwinnych. Od systemów telekomunikacyjnych opartych na VoIP, rozwiązań video dla firm ubezpieczeniowych, banków, kilka fintechowych blockchainów, po centralny system do przetwarzania faktur medycznych dla rządu w Ghanie. Uwielbia proste rozwiązania i na codzień jest leniwy, dlatego jeżeli może automatyzuje co może - stąd wielka miłość do konteneryzacji i k8s'a, technologii CI/CD oraz CM! Maciej posiada wieloletnie doświadczenie trenerskie, Master of

Ceremony na scenie TECH konferencji Infoshare, na codzień zarządza holendersko-polską technologiczną

## Informacje dodatkowe

### Informacje o materiałach dla uczestników usługi

Uczestnicy otrzymają komplet materiałów PDF

### Warunki uczestnictwa

Podstawowa znajomość systemu Linux, umiejętność korzystania z komputera

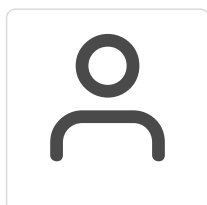
### Informacje dodatkowe

Szkolenie kończy się wydaniem certyfikatu imiennego

## Warunki techniczne

Dostęp do internetu, laptop/komputer, wideoaplikacja ZOOM lub dostęp do wersji w przeglądarce, możliwość podłączenia się do hostów przygotowanych na pulpicie zdalnym.

## Kontakt



### Biuro Obsługi Klienta

**E-mail** [biuro@jssystem.com](mailto:biuro@jssystem.com)

**Telefon** (+48) 534 506 503