



## Kubernetes: orkiestracja kontenerów - szkolenie kompleksowe

Numer usługi 2026/02/09/202247/3316410

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

📅 12.10.2026 do 16.10.2026

## Informacje podstawowe

<b>Kategoria</b>	Informatyka i telekomunikacja / Administracja IT i systemy komputerowe
<b>Grupa docelowa usługi</b>	Skierowane do deweloperów i administratorów, a także wszystkich zainteresowanych tworzeniem, uruchamianiem i zarządzaniem aplikacjami kontenerowymi.
<b>Minimalna liczba uczestników</b>	6
<b>Maksymalna liczba uczestników</b>	15
<b>Data zakończenia rekrutacji</b>	08-10-2026
<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

Nabycie przez uczestników kompleksowych umiejętności praktycznych w zakresie orkiestracji kontenerów przy użyciu Kubernetes, umożliwiających samodzielne wdrażanie, skalowanie i zarządzanie aplikacjami kontenerowymi w klastrze Kubernetes w środowiskach produkcyjnych.

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

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
<p>Wdraża aplikacje kontenerowe w klastrze Kubernetes.</p> <p>Zarządza konfiguracją i sekretami w Kubernetes (ConfigMap, Secret).</p>	<p>Uczestnik tworzy manifesty Deployment i Service dla wskazanej aplikacji, wdraża je w klastrze i weryfikuje dostępność aplikacji przez Service.</p> <p>Uczestnik tworzy ConfigMap i Secret, montuje je w podzie jako zmienne środowiskowe i woluminy, weryfikując poprawność konfiguracji aplikacji.</p>	<p>Test teoretyczny z wynikiem generowanym automatycznie</p> <p>Test teoretyczny z wynikiem generowanym automatycznie</p>
<p>Konfiguruje skalowanie automatyczne (HPA) i zarządza zasobami (requests/limits).</p> <p>Diagnostuje problemy z podami i usługami w klastrze Kubernetes.</p>	<p>Uczestnik konfiguruje HPA dla deployment'u z metryką CPU, symuluje obciążenie i obserwuje automatyczne skalowanie liczby replik w klastrze.</p> <p>Uczestnik identyfikuje i rozwiązuje co najmniej 2 typowe problemy (CrashLoopBackOff, ImagePullBackOff, problem z Service) przy użyciu kubectl describe, logs i exec.</p>	<p>Test teoretyczny z wynikiem generowanym automatycznie</p> <p>Test teoretyczny z wynikiem generowanym automatycznie</p>

## Kwalifikacje

### Kompetencje

Usługa prowadzi do nabycia kompetencji.

#### Warunki uznania kompetencji

Pytanie 1. Czy dokument potwierdzający uzyskanie kompetencji lub wyraźnie z nim powiązane inne dokumenty związane ze wsparciem zawierają opis efektów uczenia się?

TAK

Pytanie 2. Czy dokument lub wyraźnie z nim powiązane inne dokumenty związane ze wsparciem potwierdzają, że walidacja została przeprowadzona w oparciu o zdefiniowane w efektach uczenia się kryteria ich weryfikacji i zgodnie z zaplanowanymi metodami walidacji?

TAK

Pytanie 3. Czy dokument lub wyraźnie z nim powiązane inne dokumenty związane ze wsparciem potwierdzają zastosowanie rozwiązań zapewniających rozdzielenie procesów kształcenia i szkolenia od walidacji?

TAK

## Program

### 1. Wprowadzenie

o Architektura aplikacji kontenerowych - Docker vs Portainer vs ContainerD - czy to ma znaczenie?

- o Optymalizacja Dockerfile, jak to robić dobrze
- o Bezpieczeństwo kontenerów jak o nie zadbać i jak je skanować?
- o Hardening obrazów - czyli jak dodatkowo zabezpieczyć obraz
- o Jak dobrze zarządzać konfiguracjami w dockerze na przykładzie docker compose
- o Administracja kontenerami - na co zwrócić uwagę, skonfigurowanie logowania, metryki, tracing, itp.
- o Narzędzia wspierające codzienną pracę z dockerem.

## 2. Kubernetes

- o Podstawy architektury

- ☒ API

- ☒ Opis obiektów

- o Zarządzanie obiektami Kubernetes (imperatywne oraz deklaratywne)

- o Manifesty obiektów - struktura oraz format (YAML)

- o Omówienie kubectl i podobieństwa z docker

- ☒ Wejście w POD i kontener

- ☒ Zarządzanie kontekstem

- o Pod

- ☒ Lifecycle kontenera - co się dzieje jak POD nie działa?

- ☒ Uruchamianie kontenerów/zadań przed startem głównej aplikacji

- ☒ Zarządzanie zasobami - jak ograniczyć zasoby naszego kontenera

- ☒ Healthcheck - co to jest i jak go stosować (Liveness, Readiness oraz Startup probes)

- ☒ Wiele kontenerów w Pod - czy można?

- ☒ Przeglądanie logów i eventów - co jak idzie coś nie tak...

- ☒ Label, Selector i Adnotacje - jakie label stosować, aby się nie pogubić

- o Service discovery w Kubernetes

- ☒ Service discovery w Kubernetes

- ☒ DNS

- o Repliki, skalowanie a deployment wersji

- ☒ Recreate i rolling update - jak wdrażać nasze aplikacje na produkcję

- ☒ Omówienie polityk aktualizacji aplikacji (recreate, ramped, blue/green, canary, a/b testing, shadow)

- ☒ Canary Release - jak to dobrze zrobić na Kubernetesie (oraz jak zrobić to źle)

- ☒ Manualne i automatyczne skalowanie aplikacji - co zrobić jak nasza aplikacja potrzebuje więcej zasobów czy może się sama zeskalować - HPA

- ☒ Startowanie aplikacji na określonych maszynach oraz konfiguracja zależności pomiędzy uruchomionymi kontenerami (nodeSelector, affinity/antiAffinity, taints/tolerations)

- ☒ Zarządzanie zasobami klastra: minimalnymi oraz maksymalnymi limitami przypisanymi do kontenerów

- ☒ Priorytety aplikacji w klastrze oraz wyłączenie kontenerów z niskim priorytetem

- ☒ Utrzymanie maszyn klastra wraz z przygotowaniem okien obsługi

o Namespaces - wirtualne przestrzenie

o Eksponowanie usług

☒ NodePort

☒ ClusterIP

☒ LoadBalancer

☒ Ingress

☒ NGINX ingress controller i alternatywy

☒ Zarządzanie routowaniem aplikacji

☒ Używanie certyfikatów w ingress

☒ Automatyczne generowanie certyfikatów na przykładzie Let's encrypt i Cert Managera

o Konfiguracja i współdzielenie informacji

☒ Sekrety

☒ ConfigMaps

☒ Dobre praktyki na produkcji - wykorzystanie Hashicorp Vault

o Job i CronJob - zadania wsadowe i cykliczne

o Storage

☒ Lokalne montowanie danych

☒ EmptyDir

☒ Zarządzanie zmiennymi środowiskowymi i plikami konfiguracyjnymi aplikacji z użyciem ConfigMaps

☒ Przechowywanie danych wrażliwych takich jak: hasła, klucze czy tokeny za pomocą Secrets (generic, docker-registry, tls)

☒ Utrwalanie danych z użyciem wolumenów różnego typu

☒ Dynamiczne oraz statyczne zarządzanie wolumenami w klastrze

o DemonSets

o StatefulSets - czyli bazy danych w docker - stosować czy nie? Jak to robić? I kiedy nie robić.

o QoS

### **3. Uwierzytelnianie oraz autoryzacja**

o Organizacja informacji o klastrach i użytkownikach w pliku kubeconfig

o Przedstawienie typów użytkowników w Kubernetes: używanych przez administratorów (użytkownicy) oraz aplikacje (konta serwisowe)

o Omówienie strategii uwierzytelniania w klastrze, od użytkownika z hasłem, przez certyfikaty x509, po tokeny OpenID

o Kontrola dostępu oparta na rolach jako sposób na przypisywanie uprawnień do użytkowników

o Walidacja lub modyfikacja żądań za pomocą Admission Controllers

### **4. Sieci**

o CNI jako interfejs służący do konfiguracji kart sieciowych kontenerów

o Który CNI wybrać?

o Blokada komunikacji sieciowej w klastrze za pomocą Network Policies

o Mesh - co to jest

## 5. Monitorowanie aplikacji

- o Jakie są sposoby monitorowania Kubernetesa?
- o Prometheus czy jest to najlepsze narzędzie?
- o Dashboard kubernetesowy i alternatywy

## 6. Centralne systemy logowania

- o Omówienie różnych architektur zbierania logów w klastrze Kubernetes wraz z przeglądem najpopularniejszych narzędzi
- o Centralny monitoring zasobów w klastrze na przykładzie: Prometheus, AlertManager, Grafana

## 7. Jak postawić klaster od zera - jakie są możliwości?

## 8. Kilka przydatnych rzeczy

- o Helm
- o Kustomize

## 9. Security w Kubernetes

- o Testy penetracyjne klastra Kubernetes
- o Ograniczenie uprawnień oraz kontrola dostępu aplikacji do komponentów systemu operacyjnego z użyciem SecurityContext
- o Wymuszanie globalnych standardów specyfikacji aplikacji wykorzystując Pod Security Policy

## 10. Dystrybucje oraz użyteczne narzędzia

- o Kubernetes jako usługa na przykładzie najpopularniejszych dostawców chmur publicznych (GKE, EKS, AKS)
- o Zautomatyzowana instalacja klastra przy użyciu Kubernetes Operations (KOPS)
- o Instalacja klastra Kubernetes "on premise" (Kubespray)
- o kubectl, kubens, telepresence, lens i inne narzędzia użyteczne w codziennej pracy z klastrem Kubernetes

# Harmonogram

Liczba pozycji harmonogramu: 16

Przedmiot / temat	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
<b>1 z 16</b> Wprowadzenie	Maciej Krajewski	12-10-2026	09:00	12:30	03:30
<b>2 z 16</b> Przerwa Obiadowa	Maciej Krajewski	12-10-2026	12:30	13:00	00:30
<b>3 z 16</b> Kubernetes	Maciej Krajewski	12-10-2026	13:00	16:00	03:00
<b>4 z 16</b> Uwierzytelnianie oraz autoryzacja	Maciej Krajewski	13-10-2026	09:00	12:30	03:30

Przedmiot / temat	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
5 z 16 Przerwa Obiadowa	Maciej Krajewski	13-10-2026	12:30	13:00	00:30
6 z 16 Sieci	Maciej Krajewski	13-10-2026	13:00	16:00	03:00
7 z 16 Monitorowanie aplikacji	Maciej Krajewski	14-10-2026	09:00	12:30	03:30
8 z 16 Przerwa Obiadowa	Maciej Krajewski	14-10-2026	12:30	13:00	00:30
9 z 16 Centralne systemy logowania	Maciej Krajewski	14-10-2026	13:00	16:00	03:00
10 z 16 Jak postawić klaster od zera - jakie są możliwości?	Maciej Krajewski	15-10-2026	09:00	12:30	03:30
11 z 16 Przerwa Obiadowa	Maciej Krajewski	15-10-2026	12:30	13:00	00:30
12 z 16 Kilka przydatnych rzeczy	Maciej Krajewski	15-10-2026	13:00	16:00	03:00
13 z 16 Security w Kubernetes	Maciej Krajewski	16-10-2026	09:00	12:30	03:30
14 z 16 Przerwa Obiadowa	Maciej Krajewski	16-10-2026	12:30	13:00	00:30
15 z 16 Dystrybucje oraz użyteczne narzędzia	Maciej Krajewski	16-10-2026	13:00	15:30	02:30
16 z 16 Walidacja	Maciej Krajewski	16-10-2026	15:30	16:00	00:30

## 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

Trener posiada wieloletnie doświadczenie w obszarze architektury i administracji środowisk kontenerowych oraz infrastruktury chmurowej, przy czym kluczowe kwalifikacje w zakresie Docker, Kubernetes, DevOps i automatyzacji procesów CI/CD zostały zdobyte i są czynnie wykorzystywane w okresie ostatnich 5 lat (od 2021 roku do chwili obecnej). Potwierdzają to zrealizowane projekty wdrożeniowe dla klientów komercyjnych oraz szkolenia z konteneryzacji i DevOps w latach 2022–2026.

## Informacje dodatkowe

### Informacje o materiałach dla uczestników usługi

Informacje o materiałach dla uczestników usługi - Uczestnicy otrzymają komplet materiałów PDF. Każdy uczestnik otrzymuje kod dostępu i

dane logowania do platformy ZOOM na 7 dni przed datą rozpoczęcia szkolenia. Dane

przesyłane są na adres e-mail podany podczas rejestracji.

### Warunki uczestnictwa

Praktyczna znajomość technologii kontenerowej np. Docker Wiedza z zakresu programowania, sieci komputerowych oraz systemu Linux, umiejętność korzystania z komputera

### Informacje dodatkowe

Warunkiem ukończenia szkolenia i otrzymania zaświadczenia jest uzyskanie minimalnej

frekwencji na poziomie 80% całkowitego czasu trwania usługi. Obecność uczestnika będzie

potwierdzana na podstawie codziennych list obecności lub logów z platformy online.

# Warunki techniczne

Uczestnik musi dysponować sprzętem i łączem o parametrach:

- Procesor: min. 4-rdzeniowy (np. Intel i5/i7 lub odpowiednik AMD/M1/M2)
- Pamięć RAM: min. 16 GB
- Dysk: min. 20 GB wolnej przestrzeni
- System operacyjny: Windows 10/11 Pro, Linux lub macOS
- Multimedia: Sprawna kamera internetowa oraz mikrofon (wymagane do komunikacji i weryfikacji obecności)
- Łącze internetowe: Stabilne połączenie o minimalnej prędkości 10 Mbps (download) / 5 Mbps (upload)
- Oprogramowanie: Uprawnienia administratora pozwalające na instalację narzędzi

## Kontakt



### Biuro Obsługi Klienta

**E-mail** [biuro@jssystems.pl](mailto:biuro@jssystems.pl)

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