

## Program szkolenia:

# Kubernetes - deployment microservisów

### Informacje:

<b>Nazwa:</b>	<b>Kubernetes - deployment microservisów</b>
<b>Kod:</b>	<b>Arch-Tools-Kuber</b>
<b>Kategoria:</b>	Narzędzia
<b>Grupa docelowa:</b>	DevOps architekci
<b>Czas trwania:</b>	2 dni
<b>Forma:</b>	50% wykłady / 50% warsztaty

Nowoczesne środowisko developerskie nie istnieje obecnie bez popularnych kontenerów. Są one powszechnie używane szczególnie przy tworzeniu aplikacji w architekturze mikroserwisów.

Podczas szkolenia skupimy się na efektywnym wykorzystaniu kontenerów Docker, omawianiu najlepszych praktyk procesu budowania własnych obrazów i osadzania w nich aplikacji.

Uczestnicy posiadają praktyczne umiejętności orkiestracji kontenerów wykorzystując jedno z najpopularniejszych rozwiązań tego typu dedykowanych do środowisk testowych oraz produkcyjnych - produktem Kubernetes powstającym przy intensywnym zaangażowaniu inżynierów z Google. W trakcie praktycznych zajęć każdy uczestnik w ramach własnego klastra uczy się jak deployować kontenery z prywatnego rejestru, skalować, zarządzać konfiguracją oraz przechowywać dane na zewnętrznych wolumenach. Pokazane też będą najlepsze metody diagnozowania problemów, gromadzenia logów i badania wydajności środowiska.

### Uczestnik po szkoleniu będzie:

- Znał zasadę działania i zalety wykorzystania kontenerów Docker
- Potrafił uruchomić kontener z gotowego obrazu
- Potrafił udostępnić usługi z kontenera
- Potrafił zbudować obraz z aplikacją zgodnie z najlepszymi praktykami
- Potrafił połączyć kilka kontenerów i wykorzystywać zewnętrzne wolumeny
- Potrafił udostępnić obrazy w prywatnym rejestrze
- Znał podstawową architekturę Kubernetes oraz najważniejsze obiekty
- Potrafił uruchomić kontener z własnego obrazu w wielu instancjach na klastrze Kubernetes
- Potrafił wykorzystać funkcję rolling update podczas aktualizacji aplikacji
- Potrafił przeglądać logi w centralnym miejscu oraz badać i diagnozować wydajność całego środowiska z wykorzystaniem centralnej konsoli z monitoringiem

### Zalety szkolenia:

- Praktyczne podejście do infrastruktury microservices
- Ukierunkowanie na tworzenie kultury pracy Devops
- Sprawdzone rozwiązania gotowe do zastosowania w środowisku produkcyjnym

## Szczegółowy program:

### 1. Kontenery

#### 1.1. Kontenery Docker

1.1.1. Zasada działania kontenerów

1.1.2. Budowa kontenera dockerowego

1.1.3. Cykl życia kontenera

1.1.4. Łączenie ze sobą kontenerów

1.1.5. Ograniczanie zasobów (pamięć, cpu)

1.1.6. Dostęp do usług kontenera z zewnątrz

1.1.7. Obsługa wolumenów

#### 1.2. Obrazy

1.2.1. Warstwy, repozytoria i rejestry

1.2.2. Budowanie obrazów ad-hoc

1.2.3. Budowanie obrazów z użyciem Dockerfile

1.2.4. Dobre praktyki przy budowaniu obrazów

#### 1.3. docker-compose

1.3.1. Składnia i użycie

1.3.2. Budowanie złożonych konfiguracji

#### 1.4. Registry

1.4.1. Konfiguracja prywatnego rejestru

1.4.2. Użycie prywatnych rejestrów w pracy z własnymi obrazami

#### 1.5. Inne silniki kontenerów

1.5.1. Rkt jako alternatywa

1.5.2. Kompatybilność rkt z Docker

**2. Kubernetes**

## 2.1. Budowa i zasada działania

## 2.1.1. Instalacja

## 2.1.2. Architektura i używane komponenty

## 2.1.3. Budowa klastra

## 2.1.4. Komunikacja sieciowa

## 2.1.5. Zarządzanie przez command line i GUI

## 2.1.6. Efemeryczność i bezstanowość kontenerów

## 2.2. Obiekty

## 2.2.1. Omówienie i przykłady użycia

## 2.2.2. Wspólne atrybuty

## 2.2.3. Kontenery i Pody

## 2.2.4. ReplicaSet i ReplicationController

## 2.2.5. Service

## 2.2.6. Deployment

## 2.2.7. Jobs

## 2.2.8. DaemonSet

## 2.3. Cechy i funkcje

## 2.3.1. Replikacja kontenerów

## 2.3.2. Load balancing

## 2.3.3. Centralnie przechowywanie konfiguracji oraz danych poufnych

## 2.3.4. Współdzielenie klastra dla wielu izolowanych środowisk/projektów

## 2.3.5. Ograniczanie zasobów na poziomie projektu/środowisk, poda i kontenera.

## 2.3.6. Rolling update i rollback

## 2.3.7. Horyzontalny autoscaling

2.3.8. Health checks

2.4. Deployment aplikacji

2.4.1. Obrazy z prywatnych rejestrów

2.4.2. Scenariusze deploymentu

2.4.3. Health checks

2.5. Zewnętrzne wolumeny

2.5.1. PersistentVolume a PersistentVolumeClaim

2.6. Logowanie z kontenerów

2.6.1. Logowanie na poziomie kontenerów

2.6.2. Centralne składowanie logów i dostęp do nich

2.7. Monitoring zasobów

2.7.1. cAdvisor

2.7.2. Badanie wydajności aplikacji