

Program szkolenia:

Kubernetes - deployment microservisów

Informacje:

Nazwa:	Kubernetes - deployment microservisów
Kod:	Arch-Tools-Kuber
Kategoria:	Narzędzia
Grupa docelowa:	DevOps architekci
Czas trwania:	2 dni
Forma:	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. Wprowadzenie

1.1. 12-factor apps

1.2. Microservices - design pattern

1.3. Jak łatwiej wdrażać i utrzymywać współczesne aplikacje

2. Kontenery

2.1. Kontenery Docker

2.1.1. Zasada działania kontenerów

2.1.2. Budowa kontenera dockerowego

2.1.3. Cykl życia kontenera

2.1.4. Łączenie ze sobą kontenerów

2.1.5. Ograniczanie zasobów (pamięć, cpu)

2.1.6. Dostęp do usług kontenera z zewnątrz

2.1.7. Obsługa wolumenów

2.2. Obrazy

2.2.1. Warstwy, repozytoria i rejestry

2.2.2. Budowanie obrazów ad-hoc

2.2.3. Budowanie obrazów z użyciem Dockerfile

2.2.4. Dobre praktyki przy budowaniu obrazów

2.3. docker-compose

2.3.1. Składnia i użycie

2.3.2. Budowanie złożonych konfiguracji

2.4. Registry

2.4.1. Konfiguracja prywatnego rejestru

2.4.2. Użycie prywatnych rejestrów w pracy z własnymi obrazami

2.5. Inne silniki kontenerów

2.5.1. Rkt jako alternatywa

2.5.2. Kompatybilność rkt z Docker

3. Kubernetes

3.1. Budowa i zasada działania

3.1.1. Instalacja

3.1.2. Architektura i używane komponenty

3.1.3. Budowa klastra

3.1.4. Komunikacja sieciowa

3.1.5. Zarządzanie przez command line i GUI

3.1.6. Efemeryczność i bezstanowość kontenerów

3.2. Obiekty

3.2.1. Omówienie i przykłady użycia

3.2.2. Wspólne atrybuty

3.2.3. Kontenery i Pody

3.2.4. ReplicaSet i ReplicationController

3.2.5. Service

3.2.6. Deployment

3.2.7. Jobs

3.2.8. DaemonSet

3.3. Cechy i funkcje

3.3.1. Replikacja kontenerów

3.3.2. Load balancing

3.3.3. Centralnie przechowywanie konfiguracji oraz danych poufnych

3.3.4. Współdzielenie klastra dla wielu izolowanych środowisk/projektów

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

3.3.6. Rolling update i rollback

3.3.7. Horyzontalny autoscaling

3.3.8. Health checks

3.4. Deployment aplikacji

3.4.1. Obrazy z prywatnych rejestrów

3.4.2. Scenariusze deploymentu

3.4.3. Health checks

3.5. Zewnętrzne wolumeny

3.5.1. PersistentVolume a PersistentVolumeClaim

3.6. Logowanie z kontenerów

3.6.1. Logowanie na poziomie kontenerów

3.6.2. Centralne składowanie logów i dostęp do nich

3.7. Monitoring zasobów

3.7.1. cAdvisor

3.7.2. Badanie wydajności aplikacji

4. Zagadnienia zaawansowane

4.1. Stateless Applications i Stateful Applications