

Kubernetes – poziom zaawansowany

Monitoring, “distributed tracing”, service mesh

Cele szkolenia

Celem szkolenia jest zaznajomienie uczestników z technikami monitoringu, rozproszonego śledzenia (“distributed tracing”), routingu w aplikacjach uruchomionych na Kubernetesie, poznanie tematyki związanej z service mesh, zaawansowanych mechanizmów schedulingu oraz nabycie praktycznych umiejętności we wspomnianych tematach.

Umiejętności

Dzięki szkoleniu uczestnik będzie:

- Posługiwał się zaawansowanymi mechanizmami “schedulingu”: affinity, anti-affinity, taints, tolerations
- Korzystał z priorytetów podów oraz PodDisruptionBudget
- Znał podstawy Prometheusa
- Rozumiał, do czego służy NodeExporter
- Konfigurował alerty na podstawie metryk z Prometheusa
- Stosował Service Mesh jako narzędzie wspomagające routing, monitoring, tracing
- Używał wzorca “rate limiting” przy wykorzystaniu Service Mesh
- Rozumiał, czym jest distributed tracing
- Śledził zapytanie “składające się” z wielu elementów

Profil uczestników

Szkolenie przeznaczone jest dla programistów, inżynierów systemowych, architektów mających za sobą pierwsze doświadczenia z Kubernetesem, którzy chcą poznać w praktyce bardziej samego Kubernetesa oraz narzędzia wspomagające monitoring.

Przygotowanie uczestników

- Elementarna umiejętność poruszania się po wierszu poleceń.
- Podstawowa znajomość Kubernetesa.
- W przypadku przyniesienia swojego laptopa zainstalowany klient ssh.

Szczegółowy program szkolenia

1. Wstęp
 - 1.1. Środowisko szkoleniowe – wprowadzenie
 - 1.2. Kubernetes, monitoring, tracing

- 1.3. Service mesh
2. Scheduling w Kubernetesie
 - 2.1. Affinity, anti-affinity
 - 2.2. Taints, tolerations
 - 2.3. Priorytety podów, PodDisruptionBudget
3. Monitoring
 - 3.1. Prometheus – wprowadzenie
 - 3.2. Monitoring workerów przy pomocy NodeExportera
 - 3.3. Rules
 - 3.4. Alerty
4. Service mesh
 - 4.1. Zaawansowany routing
 - 4.2. „Rate limiting”
 - 4.3. „Fault injection”
 - 4.4. Monitoring
5. Rozproszone śledzenie
 - 5.1. Wprowadzenie
 - 5.2. Śledzenie zapytań przechodzących przez wiele aplikacji
6. Autoskalowanie aplikacji
 - 6.1. Omówienie HorizontalPodAutoscaler
 - 6.2. Autoskalowanie na podstawie własnych metryk

Metoda realizacji szkolenia

Szkolenie realizowane jest w formie naprzemiennie następujących po sobie mini wykładów oraz ćwiczeń praktycznych. Szkolenie łączy w sobie fachową wiedzę merytoryczną z praktycznymi przykładami jej wykorzystania w środowisku pracy.

Liczba dni, liczba godzin szkoleniowych

2 dni, 16 godzin szkoleniowych

Ścieżka rozwoju po szkoleniu

Po ukończeniu kursu rekomendowane jest dalsze używanie Kubernetesa oraz Prometheusa / Jaegera.

W razie zainteresowania monitoringiem dostępny jest kurs poświęcony wyłącznie Prometheusowi - [Prometheus - monitoring. Jak zautomatyzować wykrywanie nieprawidłowości](#)