

“Wprowadzenie do DevOps dla programistów (i nie tylko) - automatyzacja CI/CD, monitorowanie i kultura współpracy”

Informacje o szkoleniu:

Czas trwania: 2 dni

Liczba godzin zegarowych: 16h

Liczba godzin szkoleniowych: 18h

Godziny realizacji szkolenia: 09:00-17:00

Forma: stacjonarnie lub online

Trener - Paweł Ufnalewski

Infrastructure Architect w Devtek i współzałożyciel grupy SysOps/DevOps Polska na Facebooku. Jest architektem infrastruktury z kilkunastoletnim doświadczeniem. Zajmuje się projektowaniem i wdrażaniem środowisk wysokodostępnych opartych na systemach Linux i Windows. Po godzinach wielki fan LEGO i foodingu.

Wymagania:

- komputer
- przeglądarka internetowa
- klient SSH (cały pakiet putty albo podobny - tak, aby był klient + agent SSH)
- możliwość podłączenia się do portów (minimum) 22,80,443 do dowolnego IP w Internecie, najlepiej nieograniczony dostęp do Internetu.
- podstawowa znajomość konteneryzacji (Docker) i orkiestracji (Kubernetes)

Program szkolenia:

1. GitOps

- Nowoczesne wdrażanie (GitOps) – automatyczne wdrażanie aplikacji przez ArgoCD zgodnie z przepływem GitOps (commit → sync → deploy).

2. Ciągła integracja (CI) - automatyzacja buildów i testów

- Co to jest CI – idea częstego integrowania kodu i rola automatyzacji.
- Przegląd narzędzi – jak działają i czym się różnią GitHub Actions, GitLab CI, Jenkins.
- Anatomia pipeline'u – budowanie prostych potoków: checkout → instalacja zależności → testy jednostkowe/lintery → build artefaktu.
- Tworzenie własnego workloadu – konfiguracja i uruchomienie pierwszego automatycznego zadania (joba) w wybranym narzędziu CI.

2. Ciągłe wdrażanie (CD) - zaawansowane strategie deploymentu

- Metody automatycznego wdrażania – jak bezpiecznie przenosić kod z repozytorium na różne środowiska.
- Strategie deploymentu na produkcji – różnice, wady i zalety podejść:
 - Rolling update (stopniowa wymiana podów),
 - Blue/Green deployment (dwa identyczne środowiska, przełączenie ruchu),
 - Canary deployment (kierowanie małej części ruchu do nowej wersji).
- Weryfikacja wdrożenia – uruchomienie kontenerowej aplikacji na środowisku testowym i automatyczne sprawdzanie jej dostępności (Health Checks).

3. Jak rozmawiać z DevOpsami – most komunikacyjny

- Dev vs DevOps – zderzenie perspektyw:
 - Programista: szybkość developmentu, dostarczanie nowych feature'ów, brak awarii lokalnie.

Szkolenie "Wprowadzenie do DevOps dla programistów - automatyzacja CI/CD, monitorowanie i kultura współpracy"

- DevOps: stabilność produkcji, bezpieczeństwo, monitorowanie kosztów, utrzymanie skalowalności.
- Mity i konflikty – jak eliminować syndrom „u mnie działa, czyli problem jest po stronie operacji”.
- Kultura komunikacji i język techniczny:
 - Sformułowania nieefektywne: „Wrzuć to na produkcję ASAP”, „Coś nie działa na testowym”.
 - Sformułowania pożądane: „Mój build wymaga X zasobów CPU/RAM, zależy od bazy w wersji Y, potrzebuję zmiennej środowiskowej Z”.
- Procesy i artefakty, które programista musi znać – pull requesty, testy jednostkowe, wersjonowanie semantyczne (SemVer), artefakty buildów.
- Efektywna współpraca (Jira / Ticketing) – jak poprawnie zgłosić wymagania infrastrukturalne. Co musi zawierać zgłoszenie, aby DevOps mógł je zrealizować bez zbędnych pytań.

4. Monitoring, obserwowalność i alerting

- Co i po co monitorujemy – telemetria aplikacji (logi, metryki biznesowe) vs monitoring infrastruktury (zużycie CPU, RAM, dysków, sieci).
- Stos technologiczny – zbieranie i wizualizacja danych przy użyciu zestawu Prometheus (podstawy języka PromQL) + Grafana.
- Reagowanie na awarie – konfiguracja powiadomień przez Prometheus Alertmanager (slack, mail, pager).
- Praca z danymi – jak poprawnie czytać dashboardy i wyciągać wnioski o stanie zdrowia systemu.

5. DevSecOps i kultura ciągłej poprawy

Szkolenie “Wprowadzenie do DevOps dla programistów - automatyzacja CI/CD, monitorowanie i kultura współpracy”

- Shift Left Security – automatyczne skany bezpieczeństwa (pod kątem podatności w kodzie SAST/DAST oraz w obrazach Dockerowych) bezpośrednio w pipeline CI/CD.
- Blameless Post-mortem – kultura analizy awarii bez szukania winnych. Jak prowadzić retrospektywy po incydentach, by wyciągać wnioski techniczne i procesowe na przyszłość.
- Podsumowanie – jak skutecznie i krok po kroku wdrażać kulturę DevOps w zespole projektowym.

UWAGA:

Szkolenie **“Wprowadzenie do DevOps dla programistów (i nie tylko) - automatyzacja CI/CD, monitorowanie i kultura współpracy”** może zostać połączone ze szkoleniem **“Wprowadzenie do DevOps dla programistów (i nie tylko) - Docker i Kubernetes”**. Oba szkolenia można łączyć w całości lub wybrane sekcje. Organizator zastrzega sobie zachowanie niektórych punktów programu ze względu na dobro procesu edukacyjnego.

Informacje o Organizatorze

Nazwa: Fundacja SysOps/DevOps Polska

Adres: ul. Zbigniewa Wasiutyńskiego 16, 00-707 Warszawa

NIP: 113-29-70-799

Regon: 369993392

KRS: 0000727192

RIS: 2.14/00224/2021