

DevOps Engineer (kod: K-DEVOPS)

Opis i cel kursu

Kurs obejmuje najważniejsze narzędzia i technologie DevOps, takie jak: Git, Jenkins, Docker, Kubernetes, czy Ansible. Zdobędziesz doświadczenie w projektowaniu i implementacji nowoczesnych aplikacji oraz procesów CI/CD, które są kluczem do efektywnej infrastruktury IT. Nauczysz się automatyzować procesy uruchamiania, aktualizacji i skalowania złożonych systemów w środowiskach wirtualnych i cloudowych.

Zapytaj o szczegóły

tel. 22 63 64 164

akademia@alx.pl

Program

- Wprowadzenie do Continuous Integration/Delivery (CI/CD)**
 - Cykl rozwijania oprogramowania
 - Porównanie alternatywnych podejść: chmura kontra rozwiązania własne (on-premises)
 - Ciągłe budowanie / integracja (continuous building / integration)
 - Ciągła inspekcja (continuous inspection)
 - Ciągłe wdrażanie (continuous deployment)
- Git – rozproszony system kontroli wersji**
 - Konfiguracja
 - Podstawowe komendy
 - Praca na gałęziach
 - Zdalne repozytoria
 - Praca ze zdalnymi repozytoriami
 - Zaawansowane komendy (reset, merge, revert)
- Dostawcy repozytoriów zdalnych Git**
 - GitHub, Bitbucket oraz Gitlab
 - Podstawowe funkcjonalności
 - Pipeline na przykładzie Github Actions (wprowadzanie oraz podstawy, przykłady konfiguracji oraz działania)
 - Runnery - na przykładzie Gitlab.com
- Testowanie oprogramowania i jego rola w procesach CI/CD**
 - Ręczne (manualne) testowanie wersji aplikacji
 - Cel i specyfika testowania automatycznego
 - Testy jednostkowe, a testy integracyjne
 - Testowanie funkcjonalne/akceptacyjne
 - Testowanie wydajnościowe
 - Testy w Selenium/Puppeter – wprowadzenie
 - Przykładowe testy jednostkowe w Pythonie
 - LAB: Przygotowanie scenariusza testowego
- Serwer automatyzacyjny Jenkins**
 - Wprowadzenie (cel, architektura, zasada działania)
 - Budowa typowego projektu
 - jenkinsfile – konfiguracja przy użyciu kodu
 - Instalowanie i używanie pluginów
 - Dodawanie i używanie kluczy oraz haseł
 - Praca z agentami
 - Integracja narzędzi do testowania z serwerem Jenkins
 - LAB: uruchamianie testów oraz deploy aplikacji w zależności od wyniku testów
- Ansible**
 - Termin IAAC – Infrastructure as a code oraz jego znaczenie i rola
 - Podstawowe pojęcia oraz elementy

Najbliższe terminy

2026-06-20 (Zdalnie)
2026-06-20 (Warszawa)
2026-06-23 (Warszawa)
2026-06-23 (Zdalnie)
2026-10-10 (Zdalnie)
2026-10-10 (Warszawa)
2026-10-15 (Zdalnie)
2026-10-15 (Warszawa)

- Instalacja oraz konfiguracja
- Inventory
- Wykonywanie komend na wielu serwerach jednocześnie
- Prosty przykładowy playbook
- Wprowadzenie do ról
- Instalowanie oraz używanie pluginów
- Tworzenie bardziej zaawansowanych ról
- Bazowanie na faktach

7. Docker

- Podstawowe komendy oraz zarządzanie kontenerami
- Sieci
- Volumeny
- Dockerfile (tworzenie, budowanie obrazu, podstawowe instrukcje, ENTRYPOINT i CMD)
- LAB: budowa kontenera na bazie prostej aplikacji
- Docker registry
- Docker compose

8. Kubernetes

- Wprowadzenie do Kubernetes
 - Historia i podstawowe koncepcje Kubernetes
 - Architektura Kubernetes
 - Podstawowe komponenty: Pod, Node, Cluster
- Instalacja Kubernetes
 - Różne metody instalacji Kubernetes
 - Wprowadzenie do minidystrybucji k8s
 - Praktyczne ćwiczenie: Instalacja Kubernetes na lokalnej maszynie
 - Porównanie minidystrybucji z pełnym Kubernetes
 - Praktyczne ćwiczenie: tworzenie przykładowego klastra
- Podstawowe operacje w Kubernetes
 - Tworzenie i zarządzanie Podami
 - Skalowanie aplikacji
 - Aktualizacje i roll-backi
- Networking w Kubernetes
 - Sieciowe modele w Kubernetes
 - Service, Ingress i Network Policies
 - Praktyczne ćwiczenie: Konfiguracja sieci w Kubernetes
- Storage w Kubernetes
 - Persistent Volumes i Persistent Volume Claims
 - Storage Classes
 - Praktyczne ćwiczenie: Zarządzanie danymi w Kubernetes
- Zaawansowane tematy
 - Helm: zarządzanie aplikacjami w Kubernetes
 - Zarządzanie Kubernetesem prosto z UI
 - Praktyczne ćwiczenie: Tworzenie własnych zasobów

9. Monitoring i logowanie

- Narzędzia do monitorowania Kubernetes (Prometheus, Grafana)
- Zarządzanie logami (ELK stack, Fluentd)
- Praktyczne ćwiczenie: Implementacja monitoringu

10. Najlepsze praktyki i case study

- Najlepsze praktyki w zarządzaniu klastrem
- Przykłady różnych zastosowań
- Chmurowe rozwiązania AKS, EKS, GKS

11. Monitoring

Zapytaj o szczegóły

tel. 22 63 64 164

akademia@alx.pl

Najbliższe terminy

2026-06-20 (Zdalnie)

2026-06-20 (Warszawa)

2026-06-23 (Warszawa)

2026-06-23 (Zdalnie)

2026-10-10 (Zdalnie)

2026-10-10 (Warszawa)

2026-10-15 (Zdalnie)

2026-10-15 (Warszawa)

- Prometheus, Grafana oraz Alertmanager (wprowadzenie, podstawy działania, przykładowa konfiguracja, Service Discovery)
 - Agregator logów (logowanie w formacie JSON, ELK, Grafana Loki)
12. **AWS**
- Wprowadzenie (rodzaje i najpopularniejsi dostawcy)
 - Regiony oraz Availability Zones – ich cel i ważne aspekty
 - Zarządzanie budżetem
 - Opis podstawowych usług chmurowych (serwery wirtualne, managed services, storage, VPC)
 - IAM – Zarządzanie użytkownikami, politykami oraz uprawnieniami
 - awscli
 - Szerszy opis usług EC2, S3, ALB oraz ECR
13. **Terraform**
- HCL – Hashicorp Configuration Language
 - Instalacja oraz konfiguracja narzędzia Terraform
 - Podstawowe komendy
 - Dostawcy oraz zasoby
 - tfstate – znaczenie i rola
 - Provisioner – zasada działania, przykłady zastosowania
 - Zmienne
 - Output
 - LAB: w postaci uruchomienia kilku usług a następnie skonfigurowanie niektórych z nich (np. EC2) za pomocą Ansible
 - LAB: terraform i Jenkins

Zapytaj o szczegóły

tel. 22 63 64 164

akademia@alx.pl

Najbliższe terminy

2026-06-20 (Zdalnie)

2026-06-20 (Warszawa)

2026-06-23 (Warszawa)

2026-06-23 (Zdalnie)

2026-10-10 (Zdalnie)

2026-10-10 (Warszawa)

2026-10-15 (Zdalnie)

2026-10-15 (Warszawa)

Przeznaczenie i wymagania

Od zapisujących się wymagamy co najmniej podstawowej, praktycznej znajomości tematyki związanej z Linuksem i sieciami TCP/IP, oraz dowolnego języka skryptowego (może być linuxowy bash, może być Python).

Certyfikaty

Uczestnicy szkolenia otrzymują imienne certyfikaty sygnowane przez ALX.