

AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

KARTA OPISU PRZEDMIOTU

Wydział		Informatyki	
Kierunek		Informatyka	
Specjalność		AI DevOps Engineering	
Semestr	VI	Program studiów, dla którego obowiązuje sylabus	2025/2026
Stopień studiów	I		

Nazwa przedmiotu	Zaawansowana konteneryzacja aplikacji i orkiestracja kontenerów			
Kod przedmiotu	ZKAIOK			
Łączna liczba godzin	30	Tryb	stacjonarny	niestacjonarny
Profil kształcenia	Ogólnoakademicki (A)		Praktyczny (P)	
Forma zajęć	laboratorium			
Język przedmiotu	polski			
Liczba punktów ECTS	2			

Prowadzący zajęcia	
Forma prowadzonych zajęć	Laboratorium
Wymiar zajęć	30 h
Stopień (tytuł) naukowy	
Imię	
Nazwisko	

Wymagania wstępne	Ukończenie kursu Konteneryzacja i orkiestracja danych.
Założenia i cele przedmiotu	Przedmiot skupia się na pogłębieniu wiedzy i umiejętności związanych z konteneryzacją oraz orkiestracją kontenerów w kontekście DevOps i GitOps. Studenci poznają zaawansowane praktyki kontroli wersji (GitFlow), narzędzia GitOps (Flux, Argo CD), zasady CI/CD (Jenkins, GitLab CI), koncepcję infrastruktury jako kodu (Terraform, Ansible), a także monitoring, logowanie i bezpieczeństwo w środowiskach kontenerowych.
Metody dydaktyczne	<ol style="list-style-type: none"> 1. Prezentacje multimedialne. 2. Pokazy przykładowych rozwiązań problemów. 3. Rozwiązywanie zadań praktycznych.

Efekty uczenia się (odniesienie do charakterystyk poziomów Polskiej Ramy Kwalifikacji)		Odniesienie do efektów dla kierunku	Odniesienie do efektów uczenia się wg Polskiej Ramy Kwalifikacji
WIEDZA – absolwent zna i rozumie:	W01. Filozofię DevOps i GitOps oraz podobieństwa i różnice między tymi podejściami.	K_W06 K_W19 K_W20 K_W21 K_W23	P6S_WG P6S_WG_INŻ
	W02. Zaawansowane metody kontroli wersji w Git (np. GitFlow) oraz zastosowanie narzędzi GitOps (Flux, Argo CD).		
	W03. Zasady Continuous Integration/Continuous		

AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

	<p>Deployment oraz narzędzia do budowy pipeline'ów (Jenkins, GitLab CI).</p> <p>W04. Koncepcję infrastruktury jako kodu (Terraform, Ansible) i jej znaczenie w zarządzaniu środowiskami IT.</p> <p>W05. Metody monitoringu, logowania i praktyki DevSecOps w środowiskach kontenerowych.</p>		
UMIEJĘTNOŚCI – absolwent potrafi:	<p>U01. Zastosować zaawansowane techniki kontroli wersji w Git, takie jak GitFlow, w projektach kontenerowych.</p> <p>U02. Wdrożyć GitOps przy użyciu Flux lub Argo CD, automatyzując proces wdrożenia aplikacji.</p> <p>U03. Skonfigurować pipeline CI/CD z wykorzystaniem narzędzi takich jak Jenkins lub GitLab CI.</p> <p>U04. Zaimplementować infrastrukturę jako kod z użyciem Terraform i Ansible, automatyzując zarządzanie zasobami.</p> <p>U05. Wdrożyć monitoring (Prometheus, Grafana) i logowanie (ELK Stack) oraz uwzględnić praktyki DevSecOps.</p>	<p>K_U01 K_U02 K_U03 K_U04 K_U09 K_U24 K_U25</p>	<p>P6S_UW P6S_UW_INŻ P6S_UO P6S_KK P6S_UK</p>
KOMPETENCJE SPOŁECZNE – absolwent jest gotów do	<p>K01. Pracy w zespole, przyjmując w nim różne role.</p> <p>K02. Krytycznej oceny możliwości urządzeń, oprogramowania i systemów dostępnych na rynku IT.</p> <p>K03. Ciągłego samokształcenia się w celu dostosowywania się do dynamicznie zmieniających się technologii.</p>	<p>K_K04 K_K05 K_K06</p>	<p>P6S_UO P6S_KR P6S_KK</p>

Lp.	Tematyka zajęć	Liczba godzin
Forma zajęć – laboratorium		
1	Przypomnienie podstaw Docker i Kubernetes. Architektura, obiekty.	2
2	Zaawansowane funkcje Kubernetes. Operatorzy, Custom Resource Definitions (CRD), Helm Charts.	6
3	Sieci w Kubernetes. CNI, Service Mesh (Istio), konfiguracja Ingress.	6
4	Storage w Kubernetes. Persistent Volumes, Storage Classes, dynamiczne przydzielanie.	4
5	Bezpieczeństwo w Kubernetes. Role-Based Access Control (RBAC), Network Policies, zabezpieczanie klastrów.	4

AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

6	Monitoring i logowanie w Kubernetes. Implementacja Prometheus, Grafana, EFK Stack	4
7	Automatyzacja zarządzania klastrami. GitOps w Kubernetes, narzędzia Argo CD, Flux. Zaliczenie.	4

Forma i warunki zaliczenia przedmiotu	Wykonanie projektów. Częstkowe prezentacje, zdawanie raportów, obrona projektów.	
Metody weryfikacji efektów uczenia się		Nr efektu uczenia się z sylabusu
	Ocena projektów i częściowych prezentacji.	W01-W05, U01-U05, K01-K03

Literatura podstawowa	<ol style="list-style-type: none"> 1. S. Kane, K. Matthias, <i>Docker. Praktyczne zastosowania</i>, Helion, Gliwice 2019. 2. B. Burns, J. Beda, K. Hightower, <i>Kubernetes. Tworzenie niezawodnych systemów rozproszonych</i>, Helion, Gliwice 2020. 3. M. Kief, <i>Infrastruktura jako kod. Dynamiczne systemy w epoce chmury</i>, APN Promise 2021.
Literatura uzupełniająca	<ol style="list-style-type: none"> 1. M. Krief, <i>DevOps w praktyce. Wdrażanie narzędzi Terraform, Azure DevOps, Kubernetes i Jenkins</i>, Helion, Gliwice 2023.

Nakład pracy studenta	
	Liczba godzin
Zajęcia dydaktyczne	30
Przygotowanie się do zajęć	5
Studiowanie literatury	5
Udział w konsultacjach	2
Przygotowanie projektu / eseju / prezentacji itp.	18
Przygotowanie się do egzaminu / zaliczenia	-
Inne	-
ŁĄCZNY nakład pracy studenta w godz.	60
Liczba punktów ECTS	2