

# AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

## KARTA OPISU PRZEDMIOTU

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

Nazwa przedmiotu	Zaawansowana konteneryzacja aplikacji i orkiestracja kontenerów			
Kod przedmiotu	ZKAIOK			
Łączna liczba godzin	18	Tryb	stacjonarny	niestacjonarny
Profil kształcenia	Ogólnoakademicki (A)		Praktyczny (P)	
Forma zajęć	laboratorium			
Język przedmiotu	polski			
Liczba punktów ECTS	2			

<b>Prowadzący zajęcia</b>	
<b>Forma prowadzonych zajęć</b>	<b>Laboratorium</b>
<b>Wymiar zajęć</b>	<b>18 h</b>
<b>Stopień (tytuł) naukowy</b>	
<b>Imię</b>	
<b>Nazwisko</b>	

<b>Wymagania wstępne</b>	Ukończenie kursu Konteneryzacja i orkiestracja danych.
<b>Założenia i cele przedmiotu</b>	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.
<b>Metody dydaktyczne</b>	<ol style="list-style-type: none"> <li>1. Prezentacje multimedialne.</li> <li>2. Pokazy przykładowych rozwiązań problemów.</li> <li>3. Rozwiązywanie zadań praktycznych.</li> </ol>

<b>Efekty uczenia się (odniesienie do charakterystyk poziomów Polskiej Ramy Kwalifikacji)</b>		<b>Odniesienie do efektów dla kierunku</b>	<b>Odniesienie do efektów uczenia się wg Polskiej Ramy Kwalifikacji</b>
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
<b>Forma zajęć – laboratorium</b>		
1	Przypomnienie podstaw Docker i Kubernetes. Architektura, obiekty.	1
2	Zaawansowane funkcje Kubernetes. Operatorzy, Custom Resource Definitions (CRD), Helm Charts.	4
3	Sieci w Kubernetes. CNI, Service Mesh (Istio), konfiguracja Ingress.	4
4	Storage w Kubernetes. Persistent Volumes, Storage Classes, dynamiczne przydzielanie.	2
5	Bezpieczeństwo w Kubernetes. Role-Based Access Control (RBAC), Network Policies, zabezpieczanie klastrów.	2

## AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

6	Monitoring i logowanie w Kubernetes. Implementacja Prometheus, Grafana, EFK Stack	2
7	Automatyzacja zarządzania klastrami. GitOps w Kubernetes, narzędzia Argo CD, Flux. Zaliczenie.	3

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

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

Nakład pracy studenta	
	Liczba godzin
Zajęcia dydaktyczne	18
Przygotowanie się do zajęć	9
Studiowanie literatury	9
Udział w konsultacjach	2
Przygotowanie projektu / eseju / prezentacji itp.	22
Przygotowanie się do egzaminu / zaliczenia	-
Inne	-
<b>ŁĄCZNY nakład pracy studenta w godz.</b>	<b>60</b>
<b>Liczba punktów ECTS</b>	<b>2</b>