

AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

KARTA OPISU PRZEDMIOTU

Wydział		Informatyki	
Kierunek		Informatyka	
Specjalność		AI Platform Engineering	
Semestr	II	Program studiów, dla którego obowiązuje sylabus	2025/2026
Stopień studiów	II		

Nazwa przedmiotu	Konteneryzacja aplikacji			
Kod przedmiotu	KA			
Łączna liczba godzin	60	Tryb	stacjonarny	niestacjonarny
Profil kształcenia	Ogólnoakademicki (A)		Praktyczny (P)	
Forma zajęć	wykład + laboratorium			
Język przedmiotu	polski			
Liczba punktów ECTS	6 (3+3)			

Wymagania wstępne	Znajomość podstaw przetwarzania w chmurze i modeli usług chmurowych, umiejętność programowania aplikacji webowych, podstawowa znajomość systemów operacyjnych Linux i Windows, podstawowa wiedza z zakresu sieci komputerowych i protokołów internetowych.
Założenia i cele przedmiotu	Celem przedmiotu jest zapoznanie studentów z koncepcją konteneryzacji aplikacji oraz narzędziami takimi jak Docker i Kubernetes. Studenci poznają zasady tworzenia i zarządzania obrazami Docker, konfiguracji sieci i woluminów oraz definiowania usług wielokontenerowych. Zdobędą praktyczne umiejętności w tworzeniu własnych obrazów, pracy z Docker Compose oraz uruchamianiu klastrów Kubernetes.
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. Koncepcję konteneryzacji oraz jej zastosowania w inżynierii oprogramowania. W02. Architekturę Dockera oraz zasady tworzenia i zarządzania obrazami i kontenerami. W03. Sposób konfiguracji sieci i przechowywania danych w środowisku Docker. W04. Zasady definiowania usług wielokontenerowych z użyciem Docker Compose. W05. Podstawy orkiestracji kontenerów przy użyciu	K_W06 K_W09	P7S_WG P7S_WG_INŻ

AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

	Kubernetes.		
UMIEJĘTNOŚCI – absolwent potrafi:	U01. Instalować i konfigurować środowisko Docker oraz uruchamiać kontenery. U02. Tworzyć własne obrazy Docker, korzystając z Dockerfile. U03. Definiować i zarządzać aplikacjami wielokontenerowymi za pomocą Docker Compose. U04. Konfigurować sieci i woluminy w Dockerze dla potrzeb aplikacji. U05. Uruchamiać i zarządzać klastrami Kubernetes lokalnie.	K_U11	P7S_UW P7S_UW_INŻ
KOMPETENCJE SPOŁECZNE – absolwent jest gotów do	K01. Pracy w zespole, przyjmując w nim różne role. K02. Krytycznej oceny możliwości oprogramowania i systemów dostępnych na rynku IT. K03. Ciągłego samokształcenia się w celu dostosowywania się do dynamicznie zmieniających się technologii.	K_K04 K_K05 K_K06	P6S_UO P6S_KR P6S_KK

Treści programowe		
Lp.	Tematyka zajęć	Liczba godzin
Forma zajęć – wykład		
1	Wprowadzenie do konteneryzacji.	4
2	Podstawy Dockera. Architektura, obrazy, kontenery, rejestry.	6
3	Tworzenie i zarządzanie obrazami Docker. Dockerfile, najlepsze praktyki.	6
4	Sieci i woluminy w Dockerze. Konfiguracja sieci, persystencja danych.	6
5	Docker Compose. Definiowanie usług wielokontenerowych.	4
6	Wprowadzenie do Kubernetes. Podstawy orkiestracji kontenerów. Zaliczenie.	4
Forma zajęć – laboratorium		
1	Instalacja i konfiguracja Dockera. Uruchamianie pierwszych kontenerów.	6
2	Tworzenie własnych obrazów. Ćwiczenia z Dockerfile.	6
3	Praca z Docker Compose. Tworzenie aplikacji wielokontenerowych.	6
4	Sieci i woluminy. Ćwiczenia z konfiguracji sieci i przechowywania danych.	6

AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

5	Kubernetes: uruchamianie klastrów lokalnie.	6
Systemy monitoringu i zgłoszeń incydentów		
Forma zajęć – wykład		
1	Podstawy monitoringu systemów IT. Metryki, logi, alert. Narzędzia do monitoringu: Prometheus, Grafana.	4
2	Systemy logowania i analizy logów. ELK Stack, Splunk	6
3	Zarządzanie incydentami: procesy ITIL, narzędzia do zgłaszania incydentów	5
Forma zajęć – laboratorium		
1	Instalacja i konfiguracja Prometheus i Grafana. Tworzenie dashboardów	4
2	Implementacja ELK Stack. Przetwarzanie i analiza logów.	6
3	Zarządzanie incydentami: symulacja incydentów, obsługa zgłoszeń. Zaliczenie.	5
Architektura aplikacji AI		
Forma zajęć – wykład		
1	Wprowadzenie do architektury aplikacji AI. Komponenty, wzorce projektowe	4
2	Przetwarzanie danych dla AI. ETL, pre-processing danych	4
3	Projektowanie skalowalnych aplikacji AI. Mikroserwisy, chmura	4
4	Integracja modeli AI z aplikacjami. API, frameworki.	3
Forma zajęć – laboratorium		
1	Przygotowanie danych dla modeli AI.	6
2	Implementacja prostych modeli AI. Scikit-learn, TensorFlow .	8
3	Budowanie API dla modeli AI. Flask, FastAPI.	8
4	Deploy aplikacji AI w chmurze. Zaliczenie.	8
Systemy DevOps/GitOps dla AI		
Forma zajęć – wykład		
1	Wprowadzenie do DevOps i GitOps: filozofia, narzędzia.	4
2	CI/CD w kontekście AI. Automatyzacja treningu i wdrażania modeli.	4
3	Narzędzia GitOps: Argo CD, Flux.	4
4	Bezpieczeństwo w procesach DevOps. DevSecOps, dobre praktyki.	3
Forma zajęć – laboratorium		
1	Konfiguracja pipeline'ów CI/CD. Jenkins, GitLab CI.	8
2	Automatyzacja wdrażania aplikacji AI.	8

AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

3	Implementacja GitOps. Wdrożenie z użyciem Argo CD lub Flux.	8
4	Zabezpieczanie procesów DevOps. Skanowanie kodu, zarządzanie dostępem. Zaliczenie.	6
Zaawansowane systemy danych		
Forma zajęć – wykład		
1	Bazy danych NoSQL: MongoDB, Cassandra – podstawy i zastosowania.	6
2	Przetwarzanie strumieniowe danych. Apache Kafka, podstawy Stream Processing.	6
3	Systemy rozproszone. HDFS, MapReduce, podstawy Hadoop.	6
4	Data Warehousing i Data Lakes. Koncepcje, architektury.	6
5	Bezpieczeństwo danych: szyfrowanie, kontrola dostępu, zgodność z regulacjami.	6
Forma zajęć – laboratorium		
1	Praca z bazami NoSQL. Ćwiczenia z MongoDB i Cassandra	6
2	Implementacja przetwarzania strumieniowego. Projekty z Apache Kafka.	6
3	Analiza danych w systemach rozproszonych. Praktyka z Hadoop i Spark.	6
4	Wykorzystanie data warehouses i data lakes w praktyce.	6
5	Bezpieczeństwo danych w praktyce. Konfiguracja zabezpieczeń, zgodność z RODO. Zaliczenie.	6
Projekt zespołowy AI Platform Engineering		
1	Planowanie projektu: wybór tematu, podział ról w zespole.	4
2	Projektowanie architektury rozwiązania. Dokumentacja techniczna, wybór technologii.	6
3	Implementacja projektu: praca zespołowa nad realizacją.	12
4	Testowanie i optymalizacja. Walidacja funkcjonalności, poprawa wydajności.	4
5	Prezentacja projektu i dokumentacji końcowej. Zaliczenie.	4

Forma i warunki zaliczenia przedmiotu	Egzamin pisemny z wykładu. Wykonanie projektu w ramach laboratorium.	
Metody weryfikacji efektów uczenia się		Nr efektu uczenia się z sylabusu
	Egzamin pisemny	W01-W05
	Projekt	U01-U05, K01-03

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.
Literatura uzupełniająca	<ol style="list-style-type: none"> 1. M. Krief, <i>DevOps w praktyce. Wdrażanie narzędzi Terraform, Azure DevOps, Kubernetes i Jenkins</i>,

AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

Helion, Gliwice 2023.

Nakład pracy studenta	
	Liczba godzin
Zajęcia dydaktyczne	60
Przygotowanie się do zajęć	20
Studiowanie literatury	20
Udział w konsultacjach	2
Przygotowanie projektu / eseju / prezentacji itp.	25
Przygotowanie się do egzaminu / zaliczenia	25
Inne	-
ŁĄCZNY nakład pracy studenta w godz.	152
Liczba punktów ECTS	6