

Karta przedmiotu

Nazwa i kod przedmiotu	DevOps and Microservices, PG_00177820						
Kierunek studiów	Informatyka i ekonometria (O)						
Data rozpoczęcia studiów	październik 2026 r.	Rok akademicki realizacji przedmiotu			2027/2028		
Poziom kształcenia	II stopnia	Grupa zajęć			Grupa zajęć obowiązkowych z zakresu kierunku studiów Grupa zajęć fakultatywnych Grupa zajęć powiązanych z prowadzonymi badaniami naukowymi w dziedzinie nauki związanej z kierunkiem - profil ogólnoakademicki		
Forma studiów	stacjonarne	Sposób realizacji			na uczelni		
Rok studiów	2	Język wykładowy			angielski		
Semestr studiów	3	Liczba punktów ECTS			7.0		
Profil kształcenia	ogólnoakademicki	Forma zaliczenia			zaliczenie		
Jednostka prowadząca	Rektor -> Wydział Zarządzania -> Katedra Informatyki Ekonomicznej						
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot	mgr Piotr Porzuczek					
	Prowadzący zajęcia z przedmiotu						
Formy zajęć	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	15.0	0.0	60.0	0.0	0.0	75
	W tym liczba godzin zajęć na odległość: 0.0						
Aktywność studenta i liczba godzin pracy	Aktywność studenta	Udział w zajęciach dydaktycznych, objętych planem studiów		Udział w konsultacjach		Praca własna studenta	RAZEM
	Liczba godzin pracy studenta	75		4.0		96.0	175
Cel przedmiotu	<ul style="list-style-type: none"> • Zapoznanie studentów z filozofią DevOps oraz jej wpływem na procesy tworzenia i wdrażania aplikacji. • Nabycie umiejętności automatyzacji procesów za pomocą narzędzi takich jak Jenkins. • Wdrożenie metodologii Continuous Integration, Continuous Deployment oraz Continuous Testing. • Poznanie i praktyczne zastosowanie koncepcji takich jak "Infrastructure as Code," "chaos engineering," "immutable infrastructure" oraz "shift left security". • Zastosowanie metodyk Agile w procesach DevOps. 						

Efekty uczenia się przedmiotu	Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu
	[liEMU2_U11] Student potrafi współdziałać i pracować w zespołach, a także kierować ich pracami lub podejmować wiodącą w nich rolę.	liE2_U11: Potrafi konteneryzować aplikacje używając Docker, zarządzać kontenerami w środowiskach produkcyjnych oraz konfigurować i zarządzać klastrami Kubernetes dla orkiestracji kontenerów.	[SU5] realizacja zadania problemowego
	[liEMU2_U12] Student potrafi przystosowywać, projektować lub tworzyć oraz eksploatować systemy informatyczne, wspierające funkcjonowanie podmiotów gospodarczych.	liE2_U12: Potrafi implementować monitoring i obserwabliwość systemów mikrousługowych, wykorzystując narzędzia oraz analizować metryki i logi w celu diagnozowania problemów i optymalizacji wydajności.	[SU5] realizacja zadania problemowego
	[liEMU2_U02] Student potrafi przystosować konwencjonalne lub opracować innowacyjne narzędzia statystyczne, ekonometryczne lub informatyczne oraz stosować je do analizy zjawisk ekonomicznych i społecznych.	liE2_U02: Potrafi zaprojektować i wdrożyć potoki CI/CD wykorzystując narzędzia takie jak Jenkins, GitLab CI lub GitHub Actions, automatyzując procesy budowania, testowania i wdrażania aplikacji.	[SU5] realizacja zadania problemowego [SU8] obserwacja samodzielnej lub zespołowej pracy studenta
	[liEMU2_W06] Student ma uporządkowaną i podbudowaną teoretycznie wiedzę z zakresu procesów, metod i narzędzi projektowania, tworzenia, rozwoju i zapewnienia odpowiednich warunków użytkowania narzędzi informatycznych, ekonometrycznych lub statystycznych.	liE2_W06: Zna zaawansowane narzędzia i technologie DevOps, w tym systemy kontroli wersji, narzędzia CI/CD, systemy konteneryzacji i orkiestracji oraz rozumie ich rolę w automatyzacji procesów wytwarzania oprogramowania.	[SW5] realizacja zadania problemowego
[liEMU2_W08] Student w pogłębionym stopniu zna i rozumie metody, uwarunkowania, kierunki rozwoju oraz dylematy związane z zastosowaniem zaawansowanych narzędzi ekonometrycznych, informatycznych lub statystycznych, w kontekście dynamicznych zmian otoczenia.	liE2_W08: Rozumie architekturę mikrousług, wzorce projektowe stosowane w systemach rozproszonych oraz metody komunikacji między serwisami, a także zna zasady projektowania skalowalnych i odpornych na błędy systemów.	[SW4] test/egzamin - ustny lub pisemny	
Treści przedmiotu	<p>Wykłady:</p> <ul style="list-style-type: none"> Wprowadzenie do DevOps i jego filozofii Automatyzacja procesów tworzenia i wdrażania aplikacji Continuous Integration (CI), Continuous Deployment (CD), Continuous Testing (CT) Bezpieczeństwo w Jenkins Infrastructure as Code, chaos engineering, immutable infrastructure Praktyczne zastosowanie DevOps w różnych branżach (case studies) <p>Laboratoria:</p> <ul style="list-style-type: none"> Architektura i konfiguracja Jenkins Rozszerzanie Jenkins za pomocą wtyczek i skryptów Integracja Jenkins z Docker i narzędziami zarządzania wersjami Automatyzacja testów z Selenium Infrastruktura jako kod: Ansible Chaos engineering i immutable infrastructure 		
Wymagania wstępne i dodatkowe			
Sposoby i kryteria oceniania osiągniętych efektów uczenia się	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
	Wykład: test końcowy	51.0%	50.0%
	Laboratorium: ocena projektu	51.0%	50.0%
Zalecana lista lektur	Podstawowa lista lektur	<ul style="list-style-type: none"> Nicole Forsgren i in., <i>Accelerate</i>, 2019 Jose Manuel Ortega Candel, <i>Bezpieczeństwo kontenerów w DevOps</i>, 2021 	
	Uzupełniająca lista lektur	<ul style="list-style-type: none"> Jez Humble i in., <i>Ciągłe dostarczanie oprogramowania</i>, 2015 	
	Adresy eZasobów		

Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania	<ul style="list-style-type: none">• Automatyizacja tworzenia i wdrażania aplikacji przy użyciu Jenkins• Zastosowanie Infrastructure as Code w projektach zespołowych• Wdrożenie metodyk Agile w procesach DevOps
Praktyki zawodowe w ramach przedmiotu	Nie dotyczy

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.