



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez
Unię Europejską w ramach
Europejskiego Funduszu
Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Nazwa przedmiotu		Kod ECTS	
Wybrane technologie DevOps NS		11.3.1328	
Nazwa jednostki prowadzącej przedmiot			
Instytut Informatyki			
Studia			
wydział	kierunek	poziom	drugiego stopnia
Wydział Matematyki, Fizyki i Informatyki	Informatyka	forma	niestacjonarne (zaoczne)
		moduł	wszystkie
		specjalnościowy	wszystkie
		specjalizacja	wszystkie
Nazwisko osoby prowadzącej (osób prowadzących)			
dr Jakub Neumann			
Formy zajęć, sposób ich realizacji i przypisana im liczba godzin		Liczba punktów ECTS	
Formy zajęć		7 Przedmiot w wymiarze 20h wykładu i 20h laboratorium + praca własna studenta.	
Wykład, Ćw. laboratoryjne			
Sposób realizacji zajęć			
zajęcia w sali dydaktycznej			
Liczba godzin			
Wykład: 20 godz., Ćw. laboratoryjne: 20 godz.			
Termin realizacji przedmiotu			
2020/2021 letni			
Status przedmiotu		Język wykładowy	
fakultatywny (do wyboru)		polski	
Metody dydaktyczne		Forma i sposób zaliczenia oraz podstawowe kryteria oceny lub wymagania egzaminacyjne	
<ul style="list-style-type: none"> - Projektowanie doświadczeń - Wykonywanie doświadczeń - Wykład z prezentacją multimedialną - w trakcie wykładu liczne przykłady kodowane (implementowane) na żywo - ćwiczenia polegają na implementacji kilku, na ogół prostych, aplikacji w omawianej aktualnie na wykładzie technologii 		Sposób zaliczenia	
		<ul style="list-style-type: none"> - Zaliczenie na ocenę - Egzamin 	
		Formy zaliczenia	
		<ul style="list-style-type: none"> - egzamin ustny - Egzamin w formie obrony projektów częściowych lub jednego większego projektu - ustalenie oceny zaliczeniowej na podstawie ocen częściowych otrzymywanych w trakcie trwania semestru 	
		Podstawowe kryteria oceny	
		Ocenie podlegają zrozumienie i umiejętności praktyczne posługiwania się omawianymi technikami i technologiami. Ocenie podlegają projekty, które student wykona podczas zajęć.	
Sposób weryfikacji założonych efektów kształcenia			

zakładany efekt kształcenia	egzamin	kolokwium	projekt	sprawdzian	referat	raport	aktywność w dyskusji	obserwacja postawy
	Wiedza							
K_W03	X	X						
K_W06								X
	Umiejętności							
K_U03			X					X
	Kompetencje							
K_K01								X

Określenie przedmiotów wprowadzających wraz z wymogami wstępnymi

A. Wymagania formalne

Dobra znajomość systemu Linux, protokołu HTTP, umiejętność programowania w języku JavaScript

B. Wymagania wstępne

Dobra znajomość systemu Linux, protokołu HTTP, umiejętność programowania w języku JavaScript

Cele kształcenia

Celem przedmiotu jest zapoznanie słuchaczy z wybranymi technologiami z dziedziny DevOps w szczególności z technikami konteneryzacji (na przykładzie systemu Docker) i orkiestracji (na przykładzie systemu Kubernetes) oraz zastosowaniami tych rozwiązań w zakresie projektowania, wytwarzania, testowania, wdrażania i utrzymania aplikacji (szczególnie webowych)

Treści programowe

Dziedzina DevOps (z ang. *software development* (Dev) i *information-technology operations* (Ops)) należy dziś do jednych z najbardziej rozwijających się dziedzin informatyki. Wśród bardzo wielu technologii tego działu w ostatnich latach szczególnie rozwijały się i nabierały znaczenia technologie związane z wirtualizacją, konteneryzacją oraz orkiestracją usług. Przedmiot poświęcony będzie w szczególności (eko)systemowi konteneryzacji Docker oraz orkiestracji Kubernetes. Słuchacze nauczą się w jaki sposób wykorzystywać te rozwiązania w zakresie projektowania, wytwarzania, testowania, wdrażania i utrzymania aplikacji (szczególnie webowych) oraz ich bezpieczeństwa

Wykaz literatury

- Dokumentacja techniczna omawianych technologii, liczne tutoriale, artykuły w internecie, kursy na platformach e-learningowych

Kierunkowe efekty kształcenia

K_W03: ma pogłębioną wiedzę na temat paradygmatów programowania oraz zaawansowanych konstrukcji programistycznych; zna aktualne trendy w językach programowania
K_W06: zna dobrze zasady bezpieczeństwa i higieny pracy w zawodzie informatyka
K_U03: projektuje, analizuje pod kątem poprawności i złożoności obliczeniowej oraz buduje algorytmy z wykorzystaniem zaawansowanych technik programistycznych i struktur danych
K_K01: zna ograniczenia własnej wiedzy i rozumie potrzebę dalszego uczenia się

Wiedza

Student:
ma pogłębioną wiedzę na temat podstawowych paradygmatów programowania; zna również aktualne trendy w językach programowania
Zna najważniejsze konstrukcje programistyczne oraz struktury danych

Umiejętności

Student:
ma umiejętność projektowania abstrakcyjnych struktur danych i ich wydajnych implementacji

Kompetencje społeczne (postawy)

rozumie potrzebę dalszego kształcenia, zarówno w doskonaleniu warsztatu programistycznego jak i w odniesieniu do zmieniających się trendów i nowych koncepcji wytwarzania aplikacji webowych
potrafi precyzyjnie formułować pytania, służące pogłębieniu własnego rozumowania danego tematu lub odnalezieniu brakujących elementów rozumowania szczególnie w odniesieniu do często zmieniających się technologii i frameworków środowiska wytwórczego aplikacji webowych. Dyskusja i liczne przykłady kodowania na żywo wspierają i zachęcają Studenta do aktywnego uczestnictwa w wykładzie

Kontakt

jakub.neumann@ug.edu.pl