



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Projektowanie systemów rozproszonych

Przedmiot

Kierunek studiów

Informatyka

Studia w zakresie (specjalność)

Systemy rozproszone

Poziom studiów

drugiego stopnia

Forma studiów

stacjonarne

Rok/semestr

2/3

Profil studiów

ogólnoakademicki

Język oferowanego przedmiotu

polski

Wymagalność

obligatoryjny

Liczba godzin

Wykład

20

Laboratoria

Inne (np. online)

Ćwiczenia

Projekty/seminaria

30

Liczba punktów ECTS

4

Wykładowcy

Odpowiedzialny za przedmiot/wykładowca:

dr inż. Cezary Sobaniec

Odpowiedzialny za przedmiot/wykładowca:

Wymagania wstępne

Student posiada podstawową wiedzę z zakresu systemów operacyjnych, sieci komputerowych, baz danych i systemów rozproszonych.

Student potrafi: pozyskiwać informacje z literatury (w języku polskim i angielskim), dokonywać ich interpretacji i krytycznej oceny, wyciągać wnioski oraz formułować i wyczerpująco uzasadniać opinie, planować i przeprowadzać eksperymenty, w tym pomiary i symulacje komputerowe, interpretować uzyskane wyniki i wyciągać wnioski oraz formułować i weryfikować hipotezy związane ze złożonymi problemami inżynierskimi i prostymi problemami badawczymi, integrować wiedzę z różnych obszarów informatyki (a w razie potrzeby także wiedzę z innych dyscyplin naukowych).

Student rozumie znaczenie wykorzystywania najnowszej wiedzy z zakresu informatyki w rozwiązywaniu problemów projektowych.

Cel przedmiotu

Celem przedmiotu jest zapoznanie studentów z metodyką projektowania i implementacji złożonych systemów rozproszonych pracujących w środowisku chmur obliczeniowych.



Przedmiotowe efekty uczenia się

Wiedza

1. Ma zaawansowaną i pogłębioną wiedzę z zakresu szeroko rozumianych rozproszonych systemów przetwarzania w chmurze, podstaw teoretycznych ich funkcjonowania oraz usług, narzędzi i środowisk programistycznych wykorzystywanych implementacji rozproszonych aplikacji.
2. Ma uporządkowaną i podbudowaną teoretycznie wiedzę ogólną związaną z systemami rozproszonymi i przetwarzaniem w chmurze .
3. Ma zaawansowaną wiedzę szczegółową dotyczącą projektowania i wdrażania rozproszonych aplikacji w systemach przetwarzania w chmurze.
4. Ma zaawansowaną i szczegółową wiedzę o procesach zachodzących w cyklu życia rozproszonych systemów informatycznych.

Umiejętności

1. Potrafi planować i przeprowadzać eksperymenty, w tym pomiary i symulacje komputerowe, interpretować uzyskane wyniki i wyciągać wnioski oraz formułować i weryfikować hipotezy związane ze złożonymi problemami inżynierskimi i prostymi problemami badawczymi.
2. Potrafi wykorzystać do formułowania i rozwiązywania zadań inżynierskich i prostych problemów badawczych metody analityczne i eksperymentalne.
3. Potrafi ocenić przydatność i możliwość wykorzystania nowych osiągnięć (metod i narzędzi) oraz nowych produktów informatycznych.
4. Potrafi ocenić przydatność metod i narzędzi służących do rozwiązania zadania inżynierskiego, polegającego na ocenie systemu informatycznego lub jego składowych, w tym dostrzec ograniczenia tych metod i narzędzi.
5. Potrafi stosując m.in. koncepcyjnie nowe metody rozwiązywać złożone zadania informatyczne, w tym zadania nietypowe oraz zadania zawierające komponent badawczy.
6. Potrafi zgodnie z zadaną specyfikacją, uwzględniającą aspekty pozatechniczne zaprojektować złożone urządzenie, system informatyczny lub proces oraz zrealizować ten projekt — co najmniej w części — używając właściwych metod, technik i narzędzi, w tym przystosowując do tego celu istniejące lub opracowując nowe narzędzia.

Kompetencje społeczne

1. Rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe.
2. Rozumie znaczenie wykorzystywania najnowszej wiedzy z zakresu informatyki w rozwiązywaniu problemów badawczych i praktycznych.

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Przedmiot rozliczany jest na podstawie projektu realizowanego przez cały semestr. Zadanie obejmuje zaprojektowanie aplikacji działającej w chmurze obliczeniowej wraz z jej prototypową implementacją. Ocena jest indywidualna i obejmuje weryfikację znajomości użytych technologii.

Treści programowe



W ramach wykładu podejmowane będą następujące zagadnienia:

1. Wprowadzenie do architektury chmur obliczeniowych.
2. Porównanie architektury monolitycznej i mikroserwisowej.
3. Architektury mikroserwisowe w chmurze obliczeniowej.
4. Zapewnianie wysokiej dostępności aplikacji opartych o architekturę mikroserwisową.
5. Projektowanie mechanizmów odtwarzania danych w chmurze obliczeniowej.
6. Projektowanie aplikacji rozproszonych geograficznie w chmurach publicznych.
7. Zapewnienie obserwowalności w aplikacjach rozproszonych.
8. Architektury bezserwerowe w chmurze publicznej.
9. Mechanizmy ciągłej integracji i wdrażania w chmurze obliczeniowej.

Ćwiczenia projektowe mają na celu opracowanie przykładowej aplikacji w chmurze publicznej na wybranej platformie. Projekt aplikacji powinien:

1. Zapewniać powtarzalność wdrożenia aplikacji za pomocą immutable infrastructure.
2. Zapewniać mechanizmy odtwarzania aplikacji w poziomach: warm-standby, pilot light i DR.
3. Zapewniać bezpieczeństwo oraz audyt zmian w infrastrukturze.
4. Wykorzystywać wzorce fan-out i fan-in do obsługi zdarzeń w infrastrukturze i aplikacji.
5. Uwzględniać mechanizmy monitoringu.
6. Być zgodny z Well Architected Framework firmy Amazon.
7. Zapewniać bezpieczeństwo wdrożeń przez mechanizmy ciągłej integracji.

Metody dydaktyczne

Wykład: prezentacja multimedialna, ilustrowana przykładami podawanymi na tablicy.

Projekt: dyskusje problemowe, konsultacje kolejnych etapów projektu.

Literatura

Podstawowa

1. Site reliability Engineering, Edited by Betsy Beyer, Chris Jones, Jennifer Petoff and Niall Richard Murphy, O'Reilly, 2016.
2. Site reliability workbook, Edited by Betsy Beyer, Niall Richard Murphy, David K. Rensin, Kent Kawahara and Stephen Thorne. O'Reilly, 2018.
3. Building Microservices, Sam Newman, 2015, O'Reilly.
4. Serverless Architectures on AWS, Second Edition , Peter Sbarski, Ajay Nair, 2018.

Uzupełniająca



Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	100	4
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	50	2,0
Praca własna studenta (studia literaturowe, przygotowanie do wykładu/zajęć projektowych, wykonanie projektu) ¹	50	2,0

¹ niepotrzebne skreślić lub dopisać inne czynności