

KARTA PRZEDMIOTU DLA NABORU 2023/2024 FORMA STUDIÓW: STACJONARNA/NIESTACJONARNA					
<b>INFORMACJE OGÓLNE</b>					
1. Nazwa przedmiotu: Technologie DevOps					
2. Nazwa kierunku Informatyka					
3. Poziom kształcenia studia drugiego stopnia					
4. Liczba punktów ECTS 4					
5. Liczba godzin w semestrze					
semestr	W S/NS	ćw	lab/lek S/NS	prj/zp	prk
II	15/9		30/18		
6. Język wykładowy polski					
7. Wykładowca dr inż. Jakub Smółka, mgr inż. Jarosław Wetoszka					
<b>INFORMACJE SZCZEGÓŁOWE</b>					
<b>8. Wymagania wstępne</b>					
1. Podstawowa znajomość procesu wytwarzania oprogramowania					
2. Umiejętność programowania w dowolnym języku skryptowym					
3. Wiedza z zakresu zarządzania projektem					
<b>9. Cele przedmiotu</b>					
C1 Zapoznanie z wybranymi technologiami z dziedziny DevOps w tym z technikami konteneryzacji (Docker) i orkiestracji (Kubernetes)					
C2 Zdobycie umiejętności w zakresie projektowania, wytwarzania, testowania, wdrażania i utrzymania aplikacji (szczególnie webowych) z wykorzystaniem technologii DevOps					
C3 Zdobycie wiedzy i umiejętności z zakresu tworzenia i zarządzania środowiskami CI/CD					
<b>10. Efekty uczenia się w zakresie wiedzy, umiejętności i kompetencji społecznych</b>					
Student, który zaliczył przedmiot:				odniesienie do kierunkowych efektów uczenia się	
<b>WIEDZA</b>					
EU01	zna aktualne trendy w zakresie wytwarzania, testowania, wdrażania i utrzymania aplikacji			K_W03, K_W04	
EU02	zna wybrane nowoczesne technologie DevOps i wie jakie korzyści niesie ich wykorzystanie			K_W03, K_W04	
<b>UMIEJĘTNOŚCI</b>					
EU03	umie stosować nowoczesne technologie DevOps w celu automatyzacji procesu wytwarzania oprogramowania			K_U03, K_U04	
EU04	umie tworzyć nowoczesne i skalowalne architektury aplikacji w oparciu o popularne narzędzia			K_U03, K_U04	
EU05	umie tworzyć i zarządzać środowiskami do ciągłej integracji i ciągłego wdrażania oprogramowania			K_U02, K_U03, K_U04, K_U09	
<b>KOMPETENCJE SPOŁECZNE</b>					
EU06	rozumie potrzebę dalszego kształcenia, zarówno w zakresie doskonalenia warsztatu programistycznego i zmieniających się			K_K01	

technologii, trendów i zasad wytwarzania oprogramowania	
<b>11. Treści programowe</b>	
<b>Forma zajęć</b> – wykłady/ ćwiczenia/laboratoria/zajęcia praktyczne itp.	
<b>Wykłady:</b> <ol style="list-style-type: none"> <li>1. Wprowadzenie do Devops, Linux, wirtualizacja</li> <li>2. Konteneryzacja aplikacji w środowisku Docker, zarządzania systemem skonteneryzowanym z wykorzystaniem docker-compose</li> <li>3. Architektura i narzędzia w systemach mikroserwisowych</li> <li>4. Poznanie oraz nabycie umiejętności tworzenia nowoczesnych i skalowalnych architektur aplikacji w oparciu o popularne narzędzia, tj Redis, RabbitMQ, Kafka</li> <li>5. Automatyzacja procesu konfiguracji hostów na przykładzie Ansible. Przygotowywanie playbooków i ról, umożliwiających grupowanie zadań, zmiennych oraz zarządzanych hostów.</li> <li>6. Systemy Continuous Integration, Continuous Delivery i Continuous Deployment (CI/CD). Dobre praktyki związane CI/CD z w oparciu o systemy tj. Jenkins oraz Gitlab.</li> <li>7. IaaS na przykładzie Terraform Tworzenie infrastruktury na przykładzie Terraform. Praca ze skryptami Terraform, walidacja, uruchamianie i dostosowywanie do własnych potrzeb.</li> <li>8. Orkiestracja aplikacji w środowisku Kubernetes. Podstawowe obiekty Kubernetesa, obsługa narzędzia kubectl „, zarządzanie konfiguracją.</li> <li>9. Chmura AWS. Uruchomienie produkcyjnej aplikacji w chmurze AWS. Przygotowanie systemu CI/CD wdrażającego usługę w chmurze.</li> </ol> <b>Laboratoria:</b> <ol style="list-style-type: none"> <li>1. Konfiguracja i zarządzanie środowiskiem wytwarzania oprogramowania.</li> <li>2. Praktyczne wykorzystanie Git. Branching, Gitflow.</li> <li>3. Tworzenia architektur aplikacji w oparciu o Redis, RabbitMQ, Kafka.</li> <li>4. Tworzenie skryptów shellowych na przykładzie basha.</li> <li>5. Automatyzacja procesu konfiguracji hostów na przykładzie Ansible, tworzenie prostego playbooka do instalacji i uruchomienia aplikacji.</li> <li>6. Konteneryzacja aplikacji w środowisku Docker, instalacja i konfiguracja, obrazy (tworzenie i obsługa) użycie docker-compose</li> <li>7. Orkiestracja aplikacji w środowisku Kubernetes Eksponowanie usług. Konfiguracja i współdzielenie informacji. Obsługi narzędzia kubectl.</li> <li>8. Tworzenie i zarządzanie środowiskami do ciągłej integracji i ciągłego wdrażania w oparciu o najbardziej popularne systemy, jak Jenkins oraz Gitlab. Automatyzacja deploymentu. Konfiguracje, umożliwiające budowanie, dostarczanie i wdrażanie różnych typów aplikacji. Metodyki wdrażania usług na serwery docelowe. Wydzielanie środowiska produkcyjnego od testowego.</li> </ol>	
<b>12. Narzędzia/metody dydaktyczne</b>	
<ol style="list-style-type: none"> <li>1. Wykład z wykorzystaniem prezentacji multimedialnej i „na żywo” kodowanie i pokazywanie możliwości poszczególnych narzędzi i technologii DevOps, pokazy praktycznych rozwiązań odnoszących się do prezentowanych zagadnień.</li> <li>2. Laboratoria: wykonywanie różnych konfiguracji środowisk deweloperskich z automatyzacją procesów, przygotowywanie skryptów automatyzacji itp</li> </ol>	
<b>13. Sposoby oceny (częstkowe, końcowe )</b>	
<ol style="list-style-type: none"> <li>1. Kolokwium przy komputerze.</li> <li>2. Kartkówki na laboratoriach.</li> <li>3. Ocena realizacji projektu zaliczeniowego.</li> <li>4. Ocena ciągła pracy indywidualnej podczas laboratoriów.</li> </ol>	
<b>14. Obciążenie pracą studenta</b>	
Forma aktywności	liczba godzin
1. Zajęcia z bezpośrednim udziałem nauczyciela oraz konsultacje	55/27
2. Nakład pracy studenta	45/73
suma	100
liczba punktów ECTS	4

<b>15. Literatura</b>
Literatura podstawowa:
1. Gene Kim, Jez Humble, Patrick Debois, John Willis, Nicole Forsgre, DevOps. Światowej klasy zwinność, niezawodność i bezpieczeństwo w Twojej organizacji. Helion, 2016.
2. Dokumentacja techniczna omawianych technologii, tutoriale, kursy na platformach e-learningowych
3. Przyspieszenie. Lean i DevOps w rozwoju firm technologicznych, Nicole Forsgren PhD, Jez Humble, Gene Kim, Helion, 2018.
4. Kubernetes i Docker w środowisku produkcyjnym przedsiębiorstwa. Konteneryzacja i skalowanie aplikacji oraz jej integracja z systemami korporacyjnymi, Scott Surovich, Marc Boorshtein, Helion, 2020.
Literatura uzupełniająca:
1. Docker dla programistów. Rozwijanie aplikacji i narzędzia ciągłego dostarczania DevOps. Richard Bullington-McGuire, Michael Schwartz, Andrew K. Dennis, Helion, 2020.
2. Release It!: Design and Deploy Production-Ready Software 2nd Edition, Michael T. Nygard, Pragmatic Bookshelf, 2018.
3. Kubernetes - rozwiązania chmurowe w świecie DevOps. Tworzenie, wdrażanie i skalowanie nowoczesnych aplikacji chmurowych, John Arundel, Justin Domingus, Helion, 2020.
<b>16. Formy oceny – szczegóły</b>
Warunki uzyskania zaliczenia przedmiotu: zajęcia kończą się zaliczeniem z oceną. Sposób weryfikacji efektów uczenia się: Ocena stopnia osiągniętych przez studenta efektów uczenia się następuje wg poniższych kryteriów: 5.0 – zakładany efekt uczenia się został osiągnięty bez zastrzeżeń 4.5 – zakładany efekt uczenia się został osiągnięty z pojedynczymi brakami/błędami 4.0 – zakładany efekt uczenia się został osiągnięty z nielicznymi brakami/błędami 3.5 – zakładany efekt uczenia się został osiągnięty z wieloma brakami/błędami 3.0 – zakładany efekt kształcenia został osiągnięty z licznymi i istotnymi brakami/błędami (minimalnie wymagany poziom osiągnięcia efektu) 2.0 – zakładany efekt uczenia się nie został osiągnięty
<b>17. Inne przydatne informacje o przedmiocie</b>
1. Bezpośrednich informacji o problematyce zajęć i treściach programowych udziela Prowadzący w trakcie zajęć i podczas konsultacji
2. Zajęcia odbywać się będą w Akademii Białskiej Nauk Stosowanych im. Papieża Jana Pawła II lub na platformie e-learningowej
3. Zajęcia odbywać się będą zgodnie z aktualnym planem zajęć
4. Konsultacje odbywać się będą zgodnie z obowiązującym terminarzem