



IV. Opis programu studiów

3. KARTA PRZEDMIOTU

Kod przedmiotu	E-TD-01-s6
Nazwa przedmiotu	Wirtualizacja i konteneryzacja
Nazwa przedmiotu w języku angielskim	Virtualisation and containerisation
Obowiązuje od roku akademickiego	2020/21

USYTUOWANIE MODUŁU W SYSTEMIE STUDIÓW

Kierunek studiów	Teleinformatyka
Poziom kształcenia	I stopień
Profil studiów	Praktyczny
Forma i tryb prowadzenia studiów	Studia stacjonarne
Zakres	
Jednostka prowadząca przedmiot	Katedra Systemów Informatycznych
Koordynator przedmiotu	Dr inż. Mirosław Płaza
Zatwierdził	Dziekan Wydziału Elektrotechniki Automatyki i Informatyki Dr hab. inż. Antoni Różowicz, prof. PŚk

OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

Przynależność do grupy/bloku przedmiotów*	Przedmiot kierunkowy
Status przedmiotu*	Obowiązkowy
Język prowadzenia zajęć	Polski
Usytuowanie modułu w planie studiów - semestr	Semestr VI
Wymagania wstępne	Systemy Data Center
Egzamin (TAK/NIE)	NIE
Liczba punktów ECTS	3

*pozostawić właściwe

Forma prowadzenia zajęć	wykład	ćwiczenia	laboratorium	projekt	Inne
Liczba godzin w semestrze	15	0	30		0

EFEKTY UCZENIA SIĘ

Kategoria	Symbol efektu	Efekty kształcenia	Odniesienie do efektów kierunkowych
Wiedza	W01	Posiada uporządkowaną wiedzę w zakresie wirtualizacji zasobów obliczeniowych oraz rozproszonych systemów pamięci masowej .	TI1_W13
	W02	Posiada wiedzę na temat rozwiązań w zakresie konteneryzacji i zarządzania kontenerami.	TI1_W13
	W03	Posiada wiedzę na temat automatyzacji oraz bezpieczeństwa systemów wirtualizacji i konteneryzacji.	TI1_W13
Umiejętności	U01	Potrafi tworzyć maszyny wirtualne, zarządzać nimi oraz automatyzować i zabezpieczać ich działanie.	TI1_U17 TI1_U18
	U02	Potrafi tworzyć oraz konfigurować rozproszone systemy pamięci masowej.	TI1_U17
	U03	Potrafi pracować w środowiskach Kubernetes oraz Docker.	TI1_U11
Kompetencje społeczne	K01	Ma świadomość znaczenia metod wirtualizacji i konteneryzacji oraz ich wpływu na aspekty społeczne. .	TI1_K01
	K02	Potrafi pracować w grupie w zakresie obejmującym zagadnienia wirtualizacji i konteneryzacji	TI1_K01

TREŚCI PROGRAMOWE

Forma zajęć*	Treści programowe
wykład	1. Wprowadzenie do wirtualizacji – przegląd rozwiązań wirtualizacji zasobów obliczeniowych w oparciu o VMWare, Linux KVM, Microsoft HyperV. Wprowadzenie do wirtualizacji zasobów pamięci masowej, wprowadzenie do rozwiązań sieciowych definiowanych programowo (SDN) w oparciu o OpenVSwitch. Omówienie budowy systemów chmury prywatnej w oparciu o OpenStack.
	2. Wirtualizacja zasobów obliczeniowych z wykorzystaniem rozwiązań VMWare i KVM/QEMU, wirtualne maszyny, konfiguracja sieci na hostach, zarządzanie farmą serwerów (vCenter, virsh/libvirt)
	3. Warstwa sieciowa w systemach Linux i Sieci Programowalne – Linux Bridge, OpenVSwitch, OpenFlow
	4. Rozproszone systemy pamięci masowej – VMWare vSAN, CEPH, hadoop
	5. Wprowadzenie do konteneryzacji – Linux resources isolation, cgroups, budowa kontenera od podstaw, obraz kontenera (image), środowisko uruchomieniowe – Docker, dostęp do zasobów pamięci masowej. Architektura aplikacji zbudowana w oparciu o mikroserwisy (stateless)

	6. Systemy zarządzania kontenerami – Kubernetes, omówienie komponentów, podstawowych obiektów (namespaces, pod, deployment, daemon-set, service, secrets, configmaps, persistent volume, roles and rolebindings), omówienie dostępu do danych, komunikacja pomiędzy kontenerami oraz interfejsy sieciowe (CNI). Interfejs pamięci masowej (CSI).
	7. Automatyzacja systemów wirtualizacji i konteneryzacji za pomocą Ansible lub Terraform
laboratorium	1. Wirtualizacja w oparciu o Vmware cz. I – stworzenie wirtualnej maszyny, zarządzanie siecią wirtualną, konfiguracja DVS, port-groups, zarządzanie fizycznymi łączami.
	2. Wirtualizacja w oparciu o Vmware cz. II – zarządzanie mechanizmami wysokiej dostępności, vMotion, DRS, klonowanie, budowa szablonów, eksport OVA.
	3. Wirtualizacja w oparciu o KVM / QEMU cz. I – stworzenie wirtualnej maszyny, zarządzanie siecią wirtualną, dostęp do zasobów pamięci masowej, virtual-disk
	4. Wirtualizacja w oparciu o KVM/QEMU cz. II – zarządzanie wirtualizacją za pomocą libvirt-API (tworzenie wirtualnych maszyn, zasobów pamięci masowej, klonowanie), virt-manager (libvirt GUI),
	5. Zarządzanie siecią na pojedynczym serwerze – Interfejsy, subinterfejsy i tagowanie, linux bonding, Linux Bridges, OpenVSwitch.
	6. Budowa sieci nakładkowej (overlay) bezpośrednio na serwerach z wykorzystaniem OpenVSwitch programowalnym przez protokół OpenFlow, sterowanym z kontrolera (np. OpenDayLight)
	7. Rozproszone systemy pamięci masowej cz. I – instalacja klastra CEPH
	8. Rozproszone systemy pamięci masowej cz. II – instalacja klastra HADOOP
	9. Docker cz.I – Instalacja środowiska docker na pojedynczej maszynie, stworzenie obrazu kontenera, umieszczenie w repozytorium, uruchomienie kontenera
	10. Docker cz. II – przegląd logów kontenera, logowanie się do kontenera, kopiowanie plików do i z kontenera, dostęp do stałego wolumenu (persistent volume), przekierowanie portu hosta do kontenera (service expose)

	11. Kubernetes cz. I – Instalacja Kłustra Kubernetes z wykorzystaniem narzędzi kubeadm, przegląd konfiguracji kłustra, uruchomienie przykładowej aplikacji, przegląd konfiguracji POD, Deployment, Service
	12. Kubernetes cz. II – Metody aktualizacji aplikacji (rolling upgrade), dostęp do persistent volume, nadanie praw dostępu użytkownikom, tworzenie namespace, przekazywanie parametrów do kontenera za pomocą zmiennych środowiskowych, configmap oraz Secrets
	13. Automatyzacja uruchomienia wirtualnych maszyn za pomocą Ansible lub Terraform w środowisku KVM lub VMware
	14. Automatyzacja uruchomienia aplikacji w środowisku Kubernetes z wykorzystaniem Terraform
	15. Konteneryzacja w chmurze publicznej – Amazon ECS, EKS, Google GKE – przegląd, uruchomienie przykładowej aplikacji.

*) zostawić tylko realizowane formy zajęć

METODY WERYFIKACJI EFEKTÓW UCZENIA SIĘ

Symbol efektu	Metody sprawdzania efektów uczenia się					
	Egzamin ustny	Egzamin pisemny	Kolokwium	Projekt	Sprawozdanie	Inne
W01			x			
W02			x			
W02			x			
U01			x		x	
U02			x		x	
U03			x		x	
K01			x		x	
K02			x			

FORMA I WARUNKI ZALICZENIA

Forma zajęć*	Forma zaliczenia*	Warunki zaliczenia
Wykład		Uzyskanie co najmniej 50% punktów w trakcie zajęć
Laboratorium		Uzyskanie co najmniej 50% punktów w trakcie zajęć

*) zostawić tylko realizowane formy zajęć oraz wybrać formę zaliczenia

NAKŁAD PRACY STUDENTA

Bilans punktów ECTS							
Lp.	Rodzaj aktywności	Obciążenie studenta					Jednostka
		W	C	L	P	S	
1.	Udział w zajęciach zgodnie z planem studiów	15	-	30	0	-	h
3.	Inne (konsultacje, egzamin)*	2	-	2	-	-	h
4.	Razem przy bezpośrednim udziale nauczyciela akademickiego	54					h
5.	Liczba punktów ECTS, którą student uzyskuje przy bezpośrednim udziale nauczyciela akademickiego	2,16					ECTS
6.	Liczba godzin samodzielnej pracy studenta	28					h
7.	Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy	0,84					ECTS
8.	Nakład pracy związany z zajęciami o charakterze praktycznym	30					h
9.	Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym	1,20					ECTS
10.	Sumaryczne godzinowe obciążenie pracą studenta	75					h
11.	Punkty ECTS za moduł 1 punkt ECTS=25 godzin obciążenia studenta	3					

* wszelkie formy weryfikacji efektów, w tym egzaminy oraz nie więcej niż 2 godziny konsultacji dla każdej formy zajęć

LITERATURA

1. Marek Serafin, "Wirtualizacja w praktyce", 2011
2. Karl Matthias, Sean P. Kane, "Docker. Praktyczne zastosowania", 2017
3. Kelsey Hightower, Brendan Burns, Joe Beda, „Kubernetes. Tworzenie niezawodnych systemów rozproszonych”, 2019