

	Кам'янець-Подільський національний університет імені Івана Огієнка Фізико-математичний факультет Кафедра комп'ютерних наук СИЛАБУС навчальної дисципліни «КОНТЕЙНЕРИЗАЦІЯ ТА DEVOPS»
---	--

1. Загальна інформація про курс

Назва курсу, мова викладання	КОНТЕЙНЕРИЗАЦІЯ ТА DEVOPS мова викладання – українська
Викладачі	Моцик Ростислав Васильович, доцент кафедри комп'ютерних наук
Профайл викладача	https://cs.kpnu.edu.ua/2019/11/04/motsyk-rostyslav-vasylovych/
Е-mail:	motsyk@kpnu.edu.ua
Сторінка курсу в MOODLE	
Консультації	Розклад проведення консультацій: щочетверга з 16-00 до 17-00 в ауд. №11 корпусу №4; формат консультацій – групові та індивідуальні у вигляді співбесіди

2. Анотація до курсу

Навчальна дисципліна «Контейнеризація та DevOps» призначена для здобувачів вищої освіти, які прагнуть ознайомитись із філософією та культурою DevOps, навчаючи їх ключовим практикам для прискорення та підвищення надійності циклу постачання програмного забезпечення.

3. Мета і завдання курсу

Основний акцент робиться на опануванні контейнеризації з використанням Docker, здобуття практичних навичок зі створення Docker-образів, керування контейнерами та багатокомпонентними застосунками за допомогою Docker Compose. Навчатися створювати конвеєри безперервної інтеграції (CI) для автоматичної збірки та тестування коду, використовуючи сучасні інструменти. Це включає налаштування автоматичного розгортання застосунків у хмарному середовищі, базовий моніторинг та основи оркестрації контейнерів. У результаті здобувачі вищої освіти набудуть критично важливих інженерних компетенцій для роботи на сучасному IT-ринку.

4. Результати навчання

Після завершення курсу здобувачі ВО зможуть:

- вміти проєктувати та створювати Docker-образи (Docker Images) для складних застосунків та конфігурувати багатоконтейнерні застосунки за допомогою Docker Compose.
- вміти розробляти, налаштовувати та підтримувати конвеєри CI/CD (наприклад, використовуючи GitHub Actions або GitLab CI) для автоматичної збірки, тестування та доставки коду.
- вміти впроваджувати прості стратегії розгортання (In-Place, Rolling Update) та налаштовувати базові інструменти для моніторингу та централізованого логування.

5. Формат курсу

Стандартний курс (очний).

6. Обсяг і ознаки курсу

Інформація з робочої програми навчальної дисципліни:

Найменування показників	Характеристика навчального курсу
	денна форма навчання
Освітня програма, спеціальність	Освітньо-професійна програма: <i>Комп'ютерні науки та інформаційні технології</i> спеціальність: <i>122 Комп'ютерні науки</i>
Рік навчання/ рік викладання	третій, або четвертий
Семестр вивчення	5-ий (6-ий)
нормативна/вибіркова	вибіркова
Кількість кредитів ЄКТС	4

Загальний обсяг годин	120
Кількість годин навчальних занять	40
Лекційні заняття	12 год.
Практичні заняття	-
Семінарські заняття	-
Лабораторні заняття	28 год
Самостійна та індивідуальна робота	80 год.
Форма підсумкового контролю	залік

7. Пререквізити курсу

Освітні компоненти, які мають бути вивчені раніше: програмування, інформаційні технології, основи роботи з електронними таблицями.

8. Технічне й програмне забезпечення /обладнання

Для проведення лекцій необхідним є мультимедійне забезпечення. Для проведення лабораторних робіт – навчальна лабораторія обчислювальної техніки з доступом до мережі Інтернет.

9. Політика курсу (правила та вимоги)

Увесь навчальний контент розміщено в модульному середовищі навчання К-ПНУ імені Івана Огієнка – moodle. Підготовка до лабораторних занять, виконання завдань лабораторних робіт є обов’язковими для кожного здобувача вищої освіти.

Академічна доброчесність. Очікується, що роботи здобувачів ВО будуть їх оригінальними дослідженнями чи міркуваннями. Відсутність покликань на використані джерела, фабрикування джерел, списування, втручання в роботу інших здобувачів ВО становлять приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в роботі здобувача ВО є підставою для її незарахування викладачем, незалежно від масштабів плагіату чи обману.

Відвідання занять. Очікується, що всі студенти відвідають усі лекції та лабораторні заняття курсу. Здобувачі ВО мають інформувати викладача про неможливість відвідати заняття. Для того, щоб опрацювати питання пропущеної лекції, достатньо скласти і продемонструвати викладачу конспект. Студенти зобов’язані дотримуватися термінів виконання усіх видів робіт, передбачених курсом.

Форми поточного та підсумкового контролю. Поточний контроль реалізується на лабораторних заняттях.

Підсумковий контроль з навчальної дисципліни – залік, який виставляється за результатами поточного контролю.

10. Схема курсу

Денна форма здобуття вищої освіти

Тема 1: Основи Контейнеризації (Docker).

Контейнери та віртуальні машини. Архітектура та ключові концепції Docker. Робота з Docker Images та Containers. Створення оптимізованих Dockerfile. Мережева взаємодія в Docker.

Тема 2: Оркестрація та Мікросервіси

Багатоконтейнерні додатки з Docker Compose. Вступ до мікросервісної архітектури. Основи Kubernetes.

Тема 3: Автоматизація та CI/CD

Філософія DevOps та принципи CALMS. Концепції безперервної інтеграції (CI). Концепції безперервного розгортання (CD). Автоматичне складання Docker-образів у CI/CD. Вступ до інфраструктури як Коду (IaC).

Тема 4: Експлуатація та Безпека

Централізоване логування (Logging) контейнерів. Моніторинг Prometheus та Grafana. Алертинг та трасування. Основи безпеки контейнерів та сканування образів. Управління секретами (Secrets Management).

11. Система оцінювання та вимоги

Оцінювання здобувачів вищої освіти здійснюється відповідно до Положення про рейтингову систему оцінювання навчальних досягнень здобувачів вищої освіти. Поточний контроль передбачає оцінювання роботи (знань і вмінь) здобувачів освіти впродовж практичних занять за 12-бальною шкалою.

Таблиця 1. Розподіл балів за поточний і модульний контроль відповідно до робочої програми навчальної дисципліни

Поточний і модульний контроль (100 балів)	Сума
Поточний контроль	100
100 балів	

Підсумковий семестровий контроль з навчальної дисципліни передбачений у формі заліку.

Відповідно до Положення про організацію освітнього процесу в Кам'янець-Подільському національному університеті імені Івана Огієнка (від 03.07.2024 р. за № 79-ОД) здобувач вищої освіти, який не має академічної заборгованості за результатами поточного контролю з освітнього компонента,

підсумковою формою контролю за яким встановлено залік, отримує підсумкову рейтингову оцінку з цього освітнього компонента відповідно до Таблиці 3:

Здобувач вищої освіти, знання, уміння і навички якого на навчальних заняттях оцінено від 1 до 3 балів, вважається таким, що недостатньо підготувався до цих занять і має академічну заборгованість за результатами поточного контролю. Поточну заборгованість, пов'язану з непідготовленістю або недостатньою підготовленістю до навчальних занять, здобувач вищої освіти повинен ліквідувати. За відпрацьовані лекційні заняття оцінки не ставляться, за практичні та лабораторні заняття нараховуються бали середнього (4, 5, 6), достатнього (7, 8, 9) та високого рівня (10, 11, 12).

Таблиця 2. Критерії оцінювання знань, умінь, навичок здобувачів вищої освіти

Рівень навчальних досягнень	Оцінка в балах	Критерії оцінювання
Початковий	1	Студент не демонструє жодних знань з матеріалу курсу, не володіє базовою термінологією (Docker, CI/CD, Git), не може пояснити різницю між CI та CD, а також повністю ігнорує практичні завдання та не надає жодного обґрунтування своїм відповідям, що свідчить про повну відсутність роботи з дисципліною.
	2	Студент демонструє лише фрагментарне, поверхове знання окремих термінів, не розуміючи їхньої суті та зв'язку, плутає команди Git або Dockerfile, не здатний створити навіть найпростіший Docker-образ і не може застосувати отримані знання для найпростіших практичних завдань із контейнеризації, що вимагає значної допомоги викладача.
	3	Студент має несистематичні знання, здатний частково відтворити лише близько 30% навчального матеріалу, відповідає на питання з помилками, невпевнено володіє термінологією DevOps та не може логічно і послідовно викласти матеріал (наприклад, описуючи етапи CI/CD), а також не виконує або виконує з критичними помилками практичні завдання, що стосуються роботи з Docker-образами чи написання базового Docker Compose файлу.
Середній	4	Студент володіє знаннями на рівні відтворення основного матеріалу, але допускає значні неточності та має труднощі з обґрунтуванням, не розуміє принципів роботи шарів Docker-образів, плутає команди <code>docker run</code> і <code>docker exec</code> , виконує лабораторні роботи лише за чіткими покроковими інструкціями та не здатний до самостійного створення конвеєра CI/CD.
	5	Студент має загальні знання, може відповісти на більшість питань, але демонструє поверховість розуміння (наприклад, знає про CI/CD, але не про його практичну

Рівень навчальних досягнень	Оцінка в балах	Критерії оцінювання
		реалізацію), допускає помилки в застосуванні термінології та не здатний самостійно впоратися з комплексною частиною практичних робіт (наприклад, налаштуванням Volumes чи мереж у Docker Compose), що вимагає додаткових пояснень та наведення прикладів.
	6	Студент має досить повні знання основних розділів курсу, вміє пояснити ключові поняття DevOps, контейнеризації та автоматизації, виконує більшість практичних завдань (зокрема, створення багатокомпонентного застосунку через Docker Compose), але демонструє брак систематичності та логічної послідовності у відповідях, має труднощі із синтезом знань із різних розділів (наприклад, інтеграція автоматизованого тестування у CI) та допускає невеликі, але суттєві помилки у теоретичних та практичних аспектах.
Достатній	7	Студент демонструє впевнене, систематичне розуміння більшості навчального матеріалу, самостійно виконує більшість практичних завдань (включаючи створення оптимізованих Docker-образів та написання базового CI-Pipeline), здатен пояснювати архітектурні рішення, але допускає несуттєві помилки у складних теоретичних питаннях (наприклад, у деталях роботи оркестрації) та не завжди може швидко знайти оптимальне рішення у нетипових практичних ситуаціях.
	8	Студент має гарні, глибокі знання, вільно оперує термінологією, здатен аргументовано пояснювати переваги та недоліки різних стратегій розгортання, успішно виконує всі лабораторні роботи, включаючи розробку повноцінного CI/CD конвеєра для простого застосунку, але його відповіді можуть бути недостатньо вичерпними або мати невеликі логічні порушення при викладі складного матеріалу, а також є потреба у додатковому стимулюванні для більш ґрунтовного обґрунтування прийнятих рішень.
	9	Студент демонструє повні та міцні знання, логічно та послідовно викладає матеріал, здатен якісно проектувати та реалізовувати CI/CD конвеєри, має глибоке розуміння оптимізації Docker-образів та принципів роботи IaC, самостійно виконує всі завдання курсу, а його відповіді є майже вичерпними, хоча можливі незначні недоліки у деталізації або обґрунтуванні вибору певних інструментів чи технологій.
Високий	10	Студент демонструє вичерпні знання, вміло застосовує теоретичний матеріал для розв'язання практичних завдань, здатен до критичного аналізу та порівняння різних

Рівень навчальних досягнень	Оцінка в балах	Критерії оцінювання
		інструментів CI/CD, успішно захищає проєкт, що демонструє комплексне застосування знань, а також самостійно знаходить додаткову інформацію, але його рішення можуть бути не завжди оптимальними або йому бракує глибини в обґрунтуванні вибору архітектурних патернів.
	11	Студент має глибокі, міцні та творчі знання, вільно оперує всіма інструментами та методологіями, здатен розробляти та обґрунтовувати нестандартні рішення для складних проблем автоматизації, демонструє відмінні навички проєктування відмовостійких CI/CD систем, а його проєкт є високоякісним, інноваційним та містить елементи оптимізації продуктивності або безпеки, що відповідає високим бакалаврським вимогам.
	12	Студент демонструє винятковий, оригінальний та творчий підхід до розв'язання усіх завдань курсу, має повні, глибокі, системні знання, здатний до професійної оцінки та проєктування комплексних конвеєрів розгортання, вільно використовує спеціалізовані інструменти (Docker, IaC, CI/CD) на експертному рівні, а його фінальний проєкт не лише ідеально виконаний, але й має високу практичну цінність, виходячи за межі стандартних вимог, наприклад, через впровадження просунутих стратегій розгортання (Blue/Green або Canary).

Таблиця 3. Підсумковий рейтинг з кредитного модуля (дисципліни)

Рейтингова оцінка з кредитного модуля	Оцінка за шкалою ECTS	Оцінка за національною шкалою
90-100	A (відмінно)	зараховано
82-89	B (добре)	
75-81	C (добре)	
67-74	D (задовільно)	
60-66	E (достатньо)	
35-59	FX (незадовільно з можливістю повторного складання)	не зараховано
34 і менше	F (незадовільно з обов'язковим проведенням додаткової роботи щодо вивчення навчального матеріалу кредитного модуля)	

Здобувач вищої освіти, який має академічну заборгованість за результатами поточного контролю з навчальної дисципліни, підсумковою формою контролю за яким встановлено залік, отримує оцінку F за шкалою ECTS та «не зараховано» за національною шкалою.

Здобувач вищої освіти, який має академічну заборгованість за результатами підсумкового контролю з освітнього компонента у формі заліку,

зобов'язаний ліквідувати її в терміни, визначені графіком ліквідації академічної заборгованості.

Частина кредитів може бути перезарахована за умови успішного проходження масового відкритого онлайн курсу з підтвердженням персоніфікованим сертифікатом освітньої платформи. Перелік масових відкритих онлайн курсів та обсяг кредитів для перезарахування визначає викладач. Наприклад, <https://prometheus.org.ua/>, <https://www.coursera.org/>, <https://www.udemy.com> та ін.

12. Рекомендована література.

Основна

1. Kim, G., Humble, J., Debois, P., & Willis, J. The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations. 2016/2021. P. 528.
2. Humble, J., & Farley, D. (). Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. 2010. P. 512.
3. Turnbull, J. Docker: Up & Running: Shipping Reliable Containers in Production. (Kane, S. P., Matthias, K. - 3rd Ed.) 2023. P. 416.
4. Chacon, S., & Straub, B. Pro Git. Обов'язковий ресурс для опанування професійних аспектів контролю версій та командної роботи, необхідних для CI. 2024. P. 456.
5. Pethuru Raj & M. Ganesh. Cloud DevOps: A Practical Guide to Implement DevOps in Cloud Environments. 2020. P. 352.

Додаткова

1. Офіційна документація Docker та Docker Compose. Електронний ресурс. <https://docs.docker.com/manuals/>
2. Офіційна документація обраного CI/CD інструменту. Електронний ресурс. <https://docs.github.com/ru/actions>
3. Klabnik, S., Beda, J., Hightower, K. Kubernetes Up and Running. Evenson, L. - 3rd Ed. 2022. P. 328.
4. Hassine, Y. Terraform: Up & Running: Writing Infrastructure as Code. 2022. P. 760.
5. Turnbull, J. Monitoring with Prometheus. 2018. P. 350.