Міністерство освіти і науки України
Національний аерокосмічний університет
«Харківський авіаційний інститут»

Кафедра інженерії програмного забезпечення (№ 603)

ЗАТВЕРДЖУЮ

Гарант освітньо-професійної програми

_____ Л. О. Волобуєва_____
(підпис) (ініціали та прізвище)

«_29_» ___серпня____ 2025 р.

СИЛАБУС *ОБОВ'ЯЗКОВОЇ* НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Технології DevOps
(назва навчальної дисципліни)

Галузь знань: 12 (F) «Інформаційні технології»
(шифр і найменування галузі знань)

Спеціальність: 121 (F2) «Інженерія програмного забезпечення»
(код та найменування спеціальності)

Освітньо-професійна програма: «Інженерія програмного забезпечення»
(найменування освітньої програми)

Рівень вищої освіти: *другий (магістерський)*

Вводиться в дію з «01» вересня 2025 р.

Харків – 2025 р.

Розробник: __Кузнецова Ю. А., доцент каф. 603, к.т.н., доцент_ _____
(прізвище та ініціали, посада, науковий ступінь і вчене звання) (підпис)

Силабус навчальної дисципліни розглянуто на засіданні кафедри інженерії програмного забезпечення (№ 603)_____
(назва кафедри)

Протокол № _1_ від « _29_» __серпня_ 2025 р.

Завідувач кафедри, _д.т.н., професор_____ _____ __І. Б. Туркін_____
(науковий ступінь і вчене звання) (підпис) (ініціали та прізвище)

Погоджено з представником здобувачів освіти:

Представник студентського самоврядування

_____ _Д. В. Дикун__
(підпис) (ініціали та прізвище)

1. Загальна інформація про викладача

ПІБ: Кузнецова Юлія Анатоліївна
Посада: доцент кафедри інженерії програмного забезпечення
Науковий ступінь: кандидат технічних наук
Вчене звання: доцент
Перелік дисциплін, які викладає:
«UI/UX дизайн», «Git-технологія командної розробки проєктів», «Парадигми функціонального програмування», «Переддипломний курс. Основи DevOps», «Технології DevOps», «DevOps та Continuous Monitoring»
Напрями наукових досліджень:
IoT & IoB, Trending "-Ops", високонавантажені системи, контейнеризація та безсерверні обчислення, UI/UX дизайн, інтерактивне мультимедійне навчальне тестування, дослідження стратегій тестування програмного забезпечення.
Контактна інформація:
y.kuznetsova@khai.edu
https://t.me/jk_25april

2. Опис навчальної дисципліни

Форма здобуття освіти	*Денна*
Семестр	1 семестр
Мова викладання	Українська
Тип дисципліни	*обов'язкова*
Обсяг дисципліни: кредити ЄКТС/ кількість годин	4 кредити ЄКТС / 120 годин (48 аудиторних, з яких: лекції – 24, практичні – 24; СРЗ – 72);.
Види навчальної діяльності	Лекції, практичні заняття, самостійна робота
Види контролю	Поточний контроль, модульний контроль, семестровий контроль – залік
Пререквізити	Немає

3. Мета та завдання навчальної дисципліни, переліки компетентностей та очікуваних результатів навчання

Мета – надання знань про основні поняття, концепції й інструменти мережевого обладнання, застосунків та сервісів. Застосування Agile-підходу задля усунення організаційних і тимчасових бар'єрів між командами розробників та іншими учасниками життєвого циклу програмного забезпечення (тестувальниками, адміністраторами, техпідтримкою).

Завдання – надання практичних навичок використання інструментів автоматизації для підвищення ефективності процесів розробки і експлуатації програмного забезпечення за рахунок їх безперервної інтеграції та активної взаємодії профільних фахівців.

Компетентності, які набуваються:

Інтегральна компетентність: здатність розв’язувати задачі дослідницького та/або інноваційного характеру інженерії програмного забезпечення у професійній діяльності та/або у процесі навчання

Фахові компетентності:

Після закінчення цієї програми здобувач освіти буде здатен:

СК01. Здатність аналізувати предметні області, формувати, класифікувати вимоги до програмного забезпечення.

СК02. Здатність розробляти і реалізовувати наукові та/або прикладні проєкти у сфері інженерії програмного забезпечення.

СК03. Здатність проєктувати архітектуру програмного забезпечення, моделювати процеси функціонування окремих підсистем і модулів.

СК04. Здатність розвивати і реалізовувати нові конкурентоспроможні ідеї в інженерії програмного забезпечення.

СК05. Здатність розробляти, аналізувати та застосовувати специфікації, стандарти, правила і рекомендації в сфері інженерії програмного

забезпечення.

СК06. Здатність ефективно керувати фінансовими, людськими, технічними та іншими проєктними ресурсами у сфері інженерії програмного забезпечення.

СК08. Здатність розробляти і координувати процеси, етапи та ітерації життєвого циклу програмного забезпечення на основі застосування сучасних моделей, методів та технологій розроблення програмного забезпечення.

СК09. Здатність забезпечувати якість програмного забезпечення.

СК10. Здатність використовувати широкий спектр спеціальних знань для створення процесу всього циклу інтеграції, включаючи розподіл ресурсів.

СК11. Здатність діяти відповідно до політики організації в частині сталого розвитку програмного продукту та процесів життєвого циклу.

Програмні результати навчання:

РН02. Оцінювати і вибирати ефективні методи і моделі розроблення, впровадження, супроводу програмного забезпечення та управління відповідними процесами на всіх етапах життєвого циклу.

РН05. Розробляти, аналізувати, обґрунтовувати та систематизувати вимоги до програмного забезпечення.

РН06. Розробляти і оцінювати стратегії проєктування програмних засобів; обґрунтовувати, аналізувати і оцінювати варіанти проектних рішень з точки зору якості кінцевого програмного продукту, ресурсних обмежень та інших факторів.

РН07. Аналізувати, оцінювати і застосовувати на системному рівні сучасні програмні та апаратні платформи для розв’язання складних задач інженерії програмного забезпечення.

РН09. Обґрунтовано вибирати парадигми і мови програмування для розроблення програмного забезпечення; застосовувати на практиці сучасні засоби розроблення програмного забезпечення.

РН10. Модифікувати існуючі та розробляти нові алгоритмічні рішення детального проєктування програмного забезпечення.

РН11. Забезпечувати якість на всіх стадіях життєвого циклу програмного забезпечення, у тому числі з використанням релевантних моделей та методів оцінювання, а також засобів автоматизованого тестування і верифікації програмного забезпечення.

РН12. Приймати ефективні організаційно-управлінські рішення в умовах невизначеності та зміни вимог, порівнювати альтернативи, оцінювати ризики.

РН13. Конфігурувати програмне забезпечення, керувати його змінами та розробленням програмної документації на всіх етапах життєвого циклу.

РН15. Здійснювати реінжиніринг програмного забезпечення відповідно до вимог замовника.

РН16. Планувати, організовувати та здійснювати тестування, верифікацію та валідацію програмного забезпечення.

РН18. Вміти використовувати гнучкі технології та інструменти для здійснення практики безперервної інтеграції та безперервної доставки/розгортання програмного продукту.

РН19. Розуміти завдання інженерії програмного забезпечення для реалізації концепції сталого розвитку суспільства.

4. Зміст навчальної дисципліни

Модуль 1.

Змістовий модуль № 1. Основи DevOps. Організація взаємодії між окремими співробітниками та командами.

Тема 1. Основи DevOps. Культура розгортання програмного забезпечення

Анотація.

Тема присвячена формуванню базових уявлень про концепцію DevOps, її виникнення та еволюцію. Розглядаються основні принципи культури DevOps, взаємозв'язок між процесами розробки, тестування та експлуатації програмного забезпечення. Аналізуються передумови переходу від традиційних моделей розробки до гнучких та хмарооріентованих підходів.

Лекційні заняття: Поняття та цілі DevOps. Еволюція культури розгортання програмного забезпечення. Життєвий цикл програмного забезпечення. Гнучка інфраструктура та cloud-native підхід. Концепції розробки, релізу та деплою. Основні засади DevOps та типові антишаблони.

Практичні заняття: Встановлення та налаштування віртуальної машини з Linux. Аналіз, оптимізація та аварійні роботи в середовищі Linux.

Самостійна робота здобувача освіти: Опрацювання матеріалу лекцій. Вивчення основ операційної системи Linux. Аналіз моделей життєвого циклу ПЗ. Підготовка до захисту лабораторної роботи. Підготовка до модульного контролю.

Тема 2. Співробітництво всередині однієї команди. Наставництво

Анотація.

У темі розглядаються соціальні та організаційні аспекти DevOps-підходу. Особлива увага приділяється питанням командної взаємодії, розвитку професійних навичок, ролі наставництва та формуванню культури зворотного зв'язку.

Лекційні заняття: Поняття співробітництва в IT-командах. Індивідуальні навички та професійні ролі. Наставництво та навчальні організації. Роль емпатії, довіри та комунікації. Стилі спілкування та вирішення конфліктів. Типові омани співробітництва.

Практичні заняття: Програмування та автоматизація. Система керування конфігураціями Ansible.

Самостійна робота здобувача освіти: Опрацювання лекційного матеріалу. Вивчення принципів конфігураційного управління. Самостійне налаштування Ansible. Підготовка до захисту лабораторної роботи. Підготовка до модульного контролю.

Тема 3. Близькість між командами. Організаційна структура

Анотація.

Тема присвячена аналізу взаємодії між різними командами в межах організації. Розглядаються організаційні структури, фактори створення ефективних команд, кооперація між підрозділами та переваги тісної взаємодії у процесі розроблення програмного забезпечення.

Лекційні заняття: Командна та організаційна структура. Мережі взаємодії між командами. Пошук точок дотику між підрозділами. Переваги посиленої близькості. Проблеми взаємодії та способи їх усунення.

Практичні заняття: Хмарні технології та мікросервісна архітектура. Деплой тестової програми в кластері Kubernetes. Контейнеризація додатків за допомогою Docker.

Самостійна робота здобувача освіти: Вивчення мікросервісної архітектури. Аналіз принципів контейнеризації. Самостійне розгортання тестових контейнерів. Підготовка до захисту лабораторної роботи. Підготовка до модульного контролю.

Модульний контроль

Модуль 2.

Змістовий модуль № 2. Впровадження DevOps. Інструменти, масштабування, існуючі практики.

Тема 4. Інструменти впровадження DevOps

Анотація.

Тема охоплює сучасну екосистему інструментів DevOps, їх класифікацію та принципи вибору. Аналізується роль автоматизації, систем моніторингу та інструментів безперервної інтеграції у сучасних програмних проєктах.

Лекційні заняття: Інструменти розробки програмного забезпечення. Інструменти автоматизації DevOps. Системи CI/CD. Моніторинг та логування. Стандартизація інструментів. Вплив інструментів на DevOps-культуру.

Практичні заняття: Налагодження процесів безперервної інтеграції та доставки. Налаштування системи моніторингу програмного застосунку.

Самостійна робота здобувача освіти: Аналіз сучасних CI/CD платформ. Вивчення підходів до моніторингу. Налагодження багтрекінгової системи. Підняття серверу збірки проєкту. Підготовка до модульного контролю.

Тема 5. Масштабування та корпоративні DevOps-практики

Анотація.

Розглядаються принципи масштабування DevOps-практик у великих організаціях, питання зростання команд, управління складністю систем та стратегій масштабування інфраструктури й процесів.

Лекційні заняття: Поняття масштабування в DevOps. Масштабування команд і процесів. Корпоративні DevOps-практики. Життєвий цикл організації. Стратегії зростання та оптимізації.

Практичні заняття: Управління за методологією Agile. Використання user stories, jobs to be done, lean canvas.

Самостійна робота здобувача освіти: Опрацювання Agile-фреймворків. Інтеграція проєкту до систем безперервної інтеграції. Налаштування TeamCity. Підготовка до захисту лабораторної роботи. Підготовка до модульного контролю.

Тема 6. Об’єднання культур DevOps та впровадження практик

Анотація.

Тема присвячена інтеграції технічних і організаційних аспектів DevOps, формуванню єдиної культури між командами та організаціями, а також практичному впровадженню DevOps-підходів.

Лекційні заняття: Теорія та практика DevOps. Взаємодія між організаціями. Заохочення міжкомандної співпраці. Практичні аспекти впровадження DevOps.

Самостійна робота здобувача освіти: Аналіз кейсів впровадження DevOps. Підготовка до підсумкового контролю. Узагальнення матеріалу дисципліни.

Модульний контроль

Обов’язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): відсутні.

До самостійної роботи здобувачів належить опрацьовування лекційного матеріалу, підготовка до практичних занять, а також самостійне вивчення тем та питань, які не викладаються на лекційних заняттях.

5. Індивідуальні завдання

Розрахункова робота (РР) спрямована на формування навичок проєктування, розгортання та супроводу сучасних програмних систем у хмарному середовищі. Завдання передбачає повний цикл створення інфраструктури для програмного застосунку з використанням підходу Infrastructure as Code (IaC) та інструментів автоматизації процесів розробки й доставки програмного забезпечення.

У ході виконання роботи студент повинен здійснити розгортання декількох ізольованих середовищ (наприклад, розробки, тестування та продуктивного середовища) у хмарі з використанням контейнерної оркестрації Kubernetes та керованих сервісів баз даних (DBaaS). Для автоматизації створення інфраструктури необхідно застосувати інструмент Terraform, забезпечивши відтворюваність, масштабованість та керованість інфраструктурних ресурсів.

Додатково студент має розробити та налаштувати CI/CD-pipeline з використанням GitLab CI/CD, який включає етапи автоматизованого тестування, збірки застосунку та його розгортання з підтримкою канаркового деплою (Canary Deployment). У межах завдання також необхідно реалізувати систему моніторингу, логування та трасування для контролю працездатності застосунку. Результати виконання роботи оформлюються у вигляді технічного звіту з описом архітектури рішення, використаних інструментів та отриманих результатів.

6. Методи навчання

1. За джерелами придбання знань – словесні: лекція (вступна, традиційна, проблемна), бесіда (евристична), диспут, дискусія, робота з друкованими та інтернет-джерелами; наочні: ілюстрація, спостереження; практичні: лабораторна робота, домашнє завдання.

2. За характером пізнавальної діяльності тих, хто навчається, – інформаційно-репродуктивний, репродуктивний, проблемне викладання, частково-пошуковий.

3. За логікою пізнання – індуктивний, дедуктивний, метод аналогій, метод вивідних знань.

4. Методи перевірки й оцінки знань, умінь, навичок: спостереження, усне опитування, контрольні питання до лабораторної роботи, тестування (традиційне та машинне).

Під час викладання дисципліни буде використовуватися презентаційна модель проведення лекційних занять з використанням проектору.

При викладанні навчальної дисципліни для активізації навчального процесу передбачено застосування сучасних навчальних технологій, таких, як: *проблемні лекції; робота в малих групах; презентації; кейс-метод.*

Проблемні лекції спрямовані на розвиток логічного мислення студентів. Коло питань теми лекції обмежується двома-трьома ключовими моментами, увага студентів концентрується на матеріалі, що не знайшов широкого відображення в підручниках, використовується досвід закордонних навчальних закладів з роздаванням студентам під час лекцій друкованого матеріалу та виділенням головних висновків з питань, що розглядаються.

При викладанні лекційного матеріалу студентам пропонуються питання для самостійного розмірковування. При цьому лектор задає запитання, які спонукають студента шукати розв'язання проблемної ситуації. Така система примушує студентів сконцентруватися і почати активно мислити в пошуках правильної відповіді.

На початку проведення проблемної лекції необхідно чітко сформулювати проблему, яку необхідно вирішити студентам. При викладанні лекційного матеріалу слід уникати прямої відповіді на поставлені запитання, а висвітлювати лекційний матеріал таким чином, щоб отриману інформацію студент міг використовувати при розв'язанні проблеми.

Міні-лекції передбачають викладення навчального матеріалу за короткий проміжок часу й характеризуються значною ємністю, складністю логічних побудов, образів, доказів та узагальнень. Міні-лекції проводяться, як правило, як частина заняття-дослідження. На початку проведення міні-лекції за вказаними темами лектор акцентує увагу студентів на необхідності представити викладений лекційний матеріал у так званому структурно-логічному вигляді. На розгляд виносяться питання, які зафіксовані у плані лекцій, але викладаються вони стисло.

Лекційне заняття, проведене у такий спосіб, пробуджує у студента активність та увагу при сприйнятті матеріалу, а також спрямовує його на використання системного підходу при відтворенні інформації, яку він одержав від викладача. Проблемні лекції та міні-лекції доцільно поєднувати з такою формою активізації навчального процесу, як робота в малих групах.

Робота в малих групах дає змогу структурувати лекційні або лабораторні заняття за формою і змістом, створює можливості для участі кожного студента в роботі за темою заняття, забезпечує формування особистісних якостей та досвіду соціального спілкування.

Після висвітлення проблеми (при використанні проблемних лекцій) або стислого викладання матеріалу (при використанні міні-лекцій) студентам

пропонується об'єднуватися у групи по 5-6 осіб та презентувати наприкінці заняття своє бачення та сприйняття матеріалу.

Презентації – виступи перед аудиторією, що використовуються для представлення певних досягнень, результатів роботи групи, звіту про виконання індивідуальних завдань.

Однією з позитивних рис презентації та її переваг при використанні в навчальному процесі є обмін досвідом, який здобули студенти при роботі у певній малій групі.

Кейс-метод – метод аналізу конкретних ситуацій, який дає змогу наблизити процес навчання до реальної практичної діяльності спеціалістів і передбачає розгляд виробничих, управлінських та інших ситуацій, складних конфліктних випадків, проблемних ситуацій, інцидентів у процесі вивчення навчального матеріалу.

7. Методи контролю

1. Опитування.
2. Лабораторні роботи.
3. Модульні контрольні роботи.
4. Форма підсумкового контролю успішності навчання: іспит (письмово).

8. Критерії оцінювання та розподіл балів, які отримують здобувачі освіти

Таблиця 8.1 – Розподіл балів, які отримують здобувачі освіти

Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
Змістовний модуль 1			
Робота на лекціях			
Робота на практичних заняттях			
Виконання і захист лабораторних (практичних) робіт	4...8	2	8...16
Модульний контроль	8…10	1	8…10
Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів

Змістовний модуль 2			
Робота на лекціях			
Робота на практичних заняттях			
Виконання і захист лабораторних (практичних) робіт	3...6	2	6...12
Модульний контроль	4…8	1	4…8
Усього за семестр			**60…100**

Семестровий контроль (залік) проводиться у разі відмови студента від балів поточного тестування й за наявності допуску до заліку. Під час складання семестрового заліку студент має можливість отримати максимум 100 балів.

Білет для заліку складається з двох теоретичних питань (кожне питання 30 балів) та одного практичного питання (питання оцінюється в 40 балів).

Таблиця 8.3 – Шкали оцінювання: бальна і традиційна

Сума балів	Оцінка за традиційною шкалою	
	Іспит, диференційний залік	Залік
90 – 100	Відмінно	Зараховано
75 – 89	Добре	
60 – 74	Задовільно	
0 – 59	Незадовільно	Не зараховано

Критерії оцінювання роботи здобувача протягом семестру

Задовільно (60-74). Показати мінімум знань та умінь. Здати індивідуальне завдання та основні лабораторні. здати тестування. Знати основи операційної системи Linux, принципи створення комп'ютерних мереж, основні мережеві протоколи, основні сервіси на Linux для підприємства. Мати базові знання про системи віртуалізації, контейнеризацію (віртуалізація на рівні ядра ОС), основи хмарних технологій, мікросервісної архітектури та контейнеризації.

Вміти обирати програмні засоби супроводження розробки програмного забезпечення; за допомогою програмних засобів планувати та спостерігати за ходом виконання робіт, їх контроль і коректування прийнятих рішень.

Добре (75-89). Твердо знати мінімум, здати всі лабораторні роботи, індивідуальне завдання та тестування. Досконало знати основні сервіси на Linux для підприємства: зберігання даних, роутер на Linux, забезпечення безпеки, мережеві сервіси, DNS, PKI та OpenVPN, Пошта, балансування навантаження, сервіси високої доступності. Добре орієнтуватися у системах віртуалізації

(технології ефективної віртуалізації, гіпервізори 2-го типу (інтегровані з хост-системою), гіпервізори 1-го типу (bare metal hypervisors), вкладена віртуалізація та віртуалізація введення-виводу. Добре розумітися на контейнеризації, хмарних технологіях (види хмарних сервісів (*-as-a-Service), Cloud-native архітектурі та Infrastructure as Code (IaC), а також мікросервісній архітектурі та контейнеризації (Docker, зберігання даних та ресурси, мережеві абстракції Kubernetes). Досконало вміти обирати програмні засоби супроводження розробки програмного забезпечення; за допомогою програмних засобів планувати та спостерігати за ходом виконання робіт, їх контроль і коректування прийнятих рішень.

Відмінно (90-100). Здати всі контрольні точки з оцінкою «відмінно». Досконально знати всі теми та уміти застосовувати їх.

9. Політика навчального курсу

Відпрацювання пропущених занять відбувається відповідно до розкладу консультацій, за попереднім погодженням з викладачем. Питання, що стосуються академічної доброчесності, розглядає викладач або за процедурою, визначеною у Положенні про академічну доброчесність.

10. Методичне забезпечення

Підручники, навчальні посібники, навчально-методичні посібники, конспекти лекцій, методичні рекомендації з проведення лабораторних робіт тощо, які видані в Університеті, знаходяться за посиланням:

http://library.khai.edu/catalog?clear_all_params=0&mode=BookList&lang=rus&ext=no&theme_path=0%2C1665%2C14849%2C13426%2C10001&themes_basket=&ttp_themes_basket=&disciplinesearch=no&top_list=1&fullsearch_fld=&author_fld=&docname_fld=&docname_cond=beginwith&theme_context=&theme_cond=all_theme&theme_id=10001&is_ttp=0&combiningAND=0&step=20&tpage=1

Сторінка дисципліни знаходиться за посиланням:

https://mentor.khai.edu/enrol/index.php?id=6754

11. Рекомендована література

Базова

1. Kim G., Humble J., Debois P., Willis J. The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations. IT Revolution Press, 2021. – 480 p.
2. Humble J., Farley D. Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. Addison-Wesley, 2011. – 512 p.

3. Bass L., Weber I., Zhu L. DevOps: A Software Architect's Perspective. Addison-Wesley, 2015. – 338 p.

4. Burns B. Designing Distributed Systems: Patterns and Paradigms for Scalable, Reliable Services. O'Reilly Media, 2018. – 165 p.

5. Hightower K., Burns B., Beda J. Kubernetes: Up and Running. O'Reilly Media, 2021. – 350 p.

6. Turnbull J. The Docker Book: Containerization Is the New Virtualization. James Turnbull Publishing, 2019. – 300 p.

7. Morris K. Infrastructure as Code: Managing Servers in the Cloud. O'Reilly Media, 2021. – 320 p.

8. Brikman Y. Terraform: Up & Running. O'Reilly Media, 2022. – 430 p.

9. Newman S. Building Microservices: Designing Fine-Grained Systems. O'Reilly Media, 2021. – 600 p.

10. Richards M., Ford N. Fundamentals of Software Architecture. O'Reilly Media, 2020. – 420 p.

Додаткова

1. Forsgren N., Humble J., Kim G. Accelerate: The Science of Lean Software and DevOps. IT Revolution Press, 2018. – 288 p.

2. Eder T. Continuous Architecture in Practice. Leanpub, 2021. – 260 p.

3. Nygard M. Release It! Design and Deploy Production-Ready Software. Pragmatic Bookshelf, 2018. – 368 p.

4. Hüttermann M. DevOps for Developers. Apress, 2012. – 184 p.

5. Walls M. Building a DevOps Culture. O'Reilly Media, 2013. – 110 p.

6. Loukides M. Cloud Native Patterns. O'Reilly Media, 2019. – 260 p.

7. Pahl C., Jamshidi P. Microservices: Principles and Technology. Springer, 2021. – 350 p.

8. Merkel D. Docker: Lightweight Linux Containers for Consistent Development and Deployment. Linux Journal, 2014.

9. Hoff T. Principles of High Availability and Scalability. IEEE Software, 2016.

10. Kleppmann M. Designing Data-Intensive Applications. O'Reilly Media, 2017. – 616 p.

12. Інформаційні ресурси

1. Kubernetes Documentation – https://kubernetes.io/docs/
2. Docker Documentation – https://docs.docker.com/
3. Terraform Documentation – https://developer.hashicorp.com/terraform
4. GitLab CI/CD Documentation – https://docs.gitlab.com/ee/ci/

5. CNCF Cloud Native Landscape – https://landscape.cncf.io/
6. Prometheus Documentation – https://prometheus.io/docs/
7. Grafana Documentation – https://grafana.com/docs/
8. OpenTelemetry Documentation – https://opentelemetry.io/docs/
9. Argo Rollouts – https://argo-rollouts.readthedocs.io/
10. Twelve-Factor App – https://12factor.net/