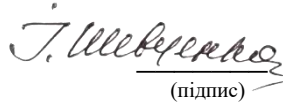


Міністерство освіти і науки України
Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Кафедра інженерії програмного забезпечення (№ 603)

ЗАТВЕРДЖУЮ

Гарант освітньо-наукової
програми

 I. В. Шевченко
(підпис) (ініціали та прізвище)

« 30 » серпня 2024 р.

**СИЛАБУС *ОБОВ'ЯЗКОВОЇ*
НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

Переддипломний курс. Основи DevOps
(назва навчальної дисципліни)

Галузь знань: 12 «Інформаційні технології»

(шифр і найменування галузі знань)

Спеціальність: 121 «Інженерія програмного забезпечення»

(код та найменування спеціальності)

Освітньо-професійна програма: «Інженерія програмного забезпечення»

(найменування освітньої програми)

Форма навчання: денна

Рівень вищої освіти: *перший (бакалаврський)*

Вводиться в дію з «01» вересня 2024 р.

Харків – 2024

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

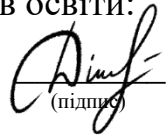
Силабус навчальної дисципліни розглянуто на засіданні кафедри інженерії
програмного забезпечення (№ 603)

(назва кафедри)

Протокол № 1 від « 30 » серпня 2024 р.

Завідувач кафедри, д.т.н., професор  І. Б. Туркін
(науковий ступінь і вчене звання) (підпис) (ініціали та прізвище)

Погоджено з представником здобувачів освіти:

 Д.В. Дикун
(підпис) (ініціали та прізвище)

1. Загальна інформація про викладача



Кузнецова Юлія Анатоліївна доцент каф. 603, к.т.н., доцент.

Основні навчальні курси, що веде викладач: «UI/UX дизайн» (24 години), «Git-технологія командної розробки проєктів» (48 годин), «Функціональне програмування мовою F#» (32 години), «Переддипломний курс. Основи DevOps» (32 години) «Технології DevOps» (32 години) «DevOps та Continuous Monitoring» (32 години) щорічно. Розробила методичне забезпечення лекційного матеріалу, лабораторних і практичних робіт.

Має **79** публікацій, з них **74** наукових та **11** навчально-методичного характеру, у тому числі **15** наукових праць у фахових наукових виданнях України та **7** публікацій, які включено до наукометричних баз **Scopus** і **WebOfScience**.

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.

2. Опис навчальної дисципліни

Семестр, в якому викладається дисципліна – 7 семестр.

Обсяг дисципліни: 5 кредитів ЄКТС/150 годин, у тому числі аудиторних – 64 год, самостійної роботи здобувачів – 86 год.

Форма здобуття освіти – денна, дистанційна.

Дисципліна – обов’язкова.

Види навчальної діяльності – лекції, лабораторні роботи, самостійна робота здобувача.

Види контролю – модульний контроль, іспит.

Мова викладання – українська.

Пререквізити – «Комп’ютерні мережі», «Операційні системи», «Архітектура та проєктування програмного забезпечення .Net».

Кореквізити – «Кваліфікаційна робота бакалавра».

3. Мета та завдання навчальної дисципліни

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:

Загальні компетентності:

ЗК01. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК02. Здатність спілкуватися іноземною мовою як усно, так і письмово.

ЗК04. Здатність спілкуватися з представниками інших професійних груп різного рівня (з експертами інших галузей знань/видів економічної діяльності).

ЗК05. Здатність генерувати нові ідеї (креативність).

Фахові компетентності:

ФК01. Здатність аналізувати предметні області, формувати, класифікувати вимоги до програмного забезпечення.

ФК02. Здатність розробляти і реалізовувати наукові та/або прикладні проєкти у сфері інженерії програмного забезпечення.

ФК06. Здатність ефективно керувати фінансовими, людськими, технічними та іншими проєктними ресурсами у сфері інженерії програмного забезпечення.

ФК07. Здатність критично осмислювати проблеми у галузі інформаційних технологій та на межі галузей знань, інтегрувати відповідні знання та розв'язувати складні задачі у широких або мультидисциплінарних контекстах.

ФК08. Здатність розробляти і координувати процеси, етапи та ітерації життєвого циклу програмного забезпечення на основі застосування сучасних моделей, методів та технологій розроблення програмного забезпечення.

ФК10. Здатність визначати мету та стратегію розвитку програмного продукту відповідно до політики організації в частині сталого розвитку.

ФК11. Здатність забезпечувати використання інновацій та поліпшень, які підвищують конкурентоспроможність або ефективність.

ФК12. Здатність демонструвати вищому керівництву бізнес-переваги потенційних змін.

Очікувані результати навчання:

ПРН01. Знати і застосовувати сучасні професійні стандарти і інші нормативно-правові документи з інженерії програмного забезпечення

ПРН03. Будувати і досліджувати моделі інформаційних процесів у прикладній області.

ПРН04. Виявляти інформаційні потреби і класифікувати дані для проектування програмного забезпечення.

ПРН06. Розробляти і оцінювати стратегії проектування програмних засобів; обґрунтовувати, аналізувати і оцінювати варіанти проектних рішень з точки зору якості кінцевого програмного продукту, ресурсних обмежень та інших факторів.

ПРН07. Аналізувати, оцінювати і застосовувати на системному рівні сучасні програмні та апаратні платформи для розв'язання складних задач інженерії програмного забезпечення.

ПРН08. Розробляти і модифікувати архітектуру програмного забезпечення для реалізації вимог замовника.

ПРН09. Обґрунтовано вибирати парадигми і мови програмування для розроблення програмного забезпечення; застосовувати на практиці сучасні засоби розроблення програмного забезпечення.

ПРН11. 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.

ПРН12. Приймати ефективні організаційно-управлінські рішення в умовах невизначеності та зміни вимог, порівнювати альтернативи, оцінювати ризики.

ПРН13. Конфігурувати програмне забезпечення, керувати його змінами та розробленням програмної документації на всіх етапах життєвого циклу.

ПРН14. Прогнозувати розвиток програмних систем та інформаційних технологій.

ПРН15. Здійснювати реінжиніринг програмного забезпечення відповідно до вимог замовника.

ПРН18. Вміти документувати результати розробок та аналізу для представлення їх колегам, керівництву, замовнику.

4. Зміст навчальної дисципліни

Модуль 1.

Змістовий модуль 1. Вступ до DevOps та автоматизації процесів розробки програмного забезпечення.

Тема 1. DevOps як методологія розроблення програмного забезпечення.

Обсяг аудиторного навантаження – 6 годин.

Анотація: Студенти дізнаються, як DevOps трансформує процес розробки, об'єднуючи команди розробників і операторів, а також інтегруючи автоматизацію у всі етапи життєвого циклу ПЗ.

Тема лекції: «DevOps як методологія розроблення програмного забезпечення».

Тема практичного заняття: «Основи впровадження DevOps у команді розробки».

Обсяг самостійної роботи здобувачів – 8 годин.

Теми, види робіт, що належать до самостійної роботи здобувача: «Вивчення основ автоматизації в DevOps», а також повторення лекційного матеріалу, підготовка до виконання практичної роботи та ознайомлення з інструментами DevOps.

Змістовий модуль 2. Системи контролю версій та їх роль у DevOps.

Тема 2. Системи контролю версій.

Обсяг аудиторного навантаження – 6 годин.

Анотація: Розгляд ключових систем контролю версій (Git, SVN), їх значення для командної роботи та інтеграції у процеси DevOps.

Тема лекції: «Системи контролю версій: Git та SVN».

Тема практичного заняття: «Практична робота з Git: основи використання та командна робота».

Обсяг самостійної роботи здобувачів – 8 годин.

Теми, види робіт, що належать до самостійної роботи здобувача: «Засвоєння основ роботи з Git», а також підготовка звіту по виконаних завданнях.

Змістовий модуль 3. Безперервна інтеграція та автоматизація збірки проекту.

Тема 3. Безперервна інтеграція проєкту. Сервера збірки.

Обсяг аудиторного навантаження – 6 годин.

Анотація: Тема пояснює концепцію CI, використання серверів збірки та автоматизації для забезпечення якості та надійності ПЗ.

Тема лекції: «Безперервна інтеграція: концепції та інструменти».

Тема практичного заняття: «Налаштування сервера збірки для безперервної інтеграції».

Обсяг самостійної роботи здобувачів – 8 годин.

Теми, види робіт, що належать до самостійної роботи здобувача: «Вивчення інструментів CI/CD», аналіз вибору серверів збірки.

Змістовий модуль 4. Розвиток кар'єри у DevOps.

Тема 4. Хто такий DevOps і як ним стати: план навчання.

Обсяг аудиторного навантаження – 6 годин.

Анотація: Лекція присвячена кар'єрним можливостям у DevOps, необхідним знанням і навичкам, а також плану навчання для оволодіння професією.

Тема лекції: «Шлях до кар'єри DevOps: необхідні навички і кроки».

Тема практичного заняття: «Розробка індивідуального плану розвитку для кар'єри DevOps».

Обсяг самостійної роботи здобувачів – 8 годин.

Теми, види робіт, що належать до самостійної роботи здобувача: «Розробка свого кар'єрного плану у сфері DevOps».

Змістовий модуль 5. Роль DevOps у продуктивній розробці.

Тема 5. DevOps як сервіс для продуктової розробки.

Обсяг аудиторного навантаження – 6 годин.

Анотація: Студенти дізнаються, як DevOps підтримує продуктивні команди шляхом створення швидших, стабільніших і адаптованих процесів розробки.

Тема лекції: «DevOps у контексті продуктової розробки».

Тема практичного заняття: «Імплементация DevOps у продуктивних командах».

Обсяг самостійної роботи здобувачів – 8 годин.

Теми, види робіт, що належать до самостійної роботи здобувача: «Аналіз кейсів застосування DevOps у продуктивних командах».

Змістовий модуль 6. Безперервна інтеграція: етапи та інструменти.

Тема 6. Етапи Continuous Integration (CI).

Обсяг аудиторного навантаження – 6 годин.

Анотація: Розглядаються основні етапи побудови CI, включаючи тестування, збірку і доставку ПЗ.

Тема лекції: «Етапи безперервної інтеграції».

Тема практичного заняття: «Налаштування етапів CI в реальному проекті».

Обсяг самостійної роботи здобувачів – 8 годин.

Теми, види робіт, що належать до самостійної роботи здобувача: «Оптимізація етапів CI для покращення процесу розробки».

Модульний контроль

Модуль 2.

Змістовий модуль 7. Використання API у DevOps.

Тема 7. API: загальна характеристика.

Обсяг аудиторного навантаження – 6 годин.

Анотація: Лекція охоплює основи створення та використання API у DevOps для інтеграції систем і автоматизації процесів.

Тема лекції: «Розробка та інтеграція API у DevOps».

Тема практичного заняття: «Створення API для автоматизації DevOps процесів».

Обсяг самостійної роботи здобувачів – 8 годин.

Теми, види робіт, що належать до самостійної роботи здобувача: «Розробка власного API для інтеграції з інструментами DevOps».

Змістовий модуль 8. Використання контейнерів та DockerFile у DevOps.

Тема 8. Повний перелік інструкцій DockerFile.

Обсяг аудиторного навантаження – 6 годин.

Анотація: Детальний розгляд DockerFile для створення оптимальних контейнерів і керування ними в DevOps середовищах.

Тема лекції: «Інструкції DockerFile для ефективної роботи в DevOps».

Тема практичного заняття: «Розробка DockerFile для контейнеризації проектів».

Обсяг самостійної роботи здобувачів – 8 годин.

Теми, види робіт, що належать до самостійної роботи здобувача: «Оптимізація DockerFile для зменшення часу зборки контейнерів».

Змістовий модуль 9. Планування та розгортання DevOps процесів.

Тема 9. Діаграма розгортання.

Обсяг аудиторного навантаження – 8 годин.

Анотація: Студенти ознайомляться з побудовою діаграм розгортання для візуалізації і планування DevOps-процесів.

Тема лекції: «Діаграми розгортання у DevOps».

Тема практичного заняття: «Розробка діаграм розгортання для проекту».

Обсяг самостійної роботи здобувачів – 10 годин.

Теми, види робіт, що належать до самостійної роботи здобувача: «Побудова діаграм для розгортання та моніторингу DevOps процесів».

Змістовий модуль 10. Вплив DevOps на архітектуру програмного забезпечення.

Тема 10. Зв'язок цього курсу з архітектурою програмного забезпечення.

Обсяг аудиторного навантаження – 8 годин.

Анотація: Як DevOps впливає на вибір архітектури ПЗ та полегшує управління складними системами.

Тема лекції: «Вплив DevOps на архітектуру ПЗ».

Тема практичного заняття: «Проектування архітектури ПЗ з урахуванням принципів DevOps».

Обсяг самостійної роботи здобувачів – 10 годин.

Теми, види робіт, що належать до самостійної роботи здобувача: «Аналіз архітектурних рішень у контексті DevOps».

Модульний контроль

5. Індивідуальні завдання

Розрахункова робота (РГР). Індивідуальне завдання полягає в розробці діаграми розгортання для проекту, що використовує методологію DevOps. Студент повинен описати проект, визначити інфраструктуру DevOps, що включає етапи безперервної інтеграції, тестування, збірки та доставки ПЗ, а також інструменти для кожного етапу (наприклад, Git, Jenkins, Docker, Kubernetes). На основі цього необхідно побудувати діаграму розгортання, що відображає взаємодію між серверами, інструментами та етапами процесу. Студент має пояснити, як вибрані інструменти та етапи автоматизують процеси і забезпечують якість ПЗ. Робота повинна містити графічне зображення діаграми та текстові пояснення кожного етапу.

6. Методи навчання

Методи навчання студентів дисципліни, що охоплює DevOps, автоматизацію та системи розробки програмного забезпечення, повинні враховувати як теоретичний, так і практичний аспекти дисципліни.

1. Лекції

Теоретичні лекції. Викладачі представляють основи DevOps, ключові інструменти та методології, що використовуються в розробці ПЗ. Під час лекцій студенти знайомляться з основними концепціями, такими як CI/CD, автоматизація, системи контролю версій.

Інтерактивні лекції включають обговорення кейсів з реальних проектів, відповіді на запитання, обговорення проблемних ситуацій, що виникають при впровадженні DevOps-практик.

2. Практичні заняття

Робота з інструментами DevOps. Студенти безпосередньо працюють з популярними інструментами, такими як Git, Jenkins, Docker. Це дозволяє студентам отримати практичні навички та глибше розуміння теоретичних основ.

Проектна діяльність. Робота над реальними або змодельованими проектами, де студенти можуть впроваджувати процеси CI/CD, автоматизацію тестування, управління конфігураціями та розгортання.

Рішення задач та кейсів. Аналіз реальних сценаріїв з використанням DevOps-підходів. Це дозволяє студентам оцінити та покращити навички вирішення практичних проблем.

3. Самостійна робота

Онлайн-курси та платформи. Використання онлайн-курсів (наприклад, на платформах Coursera, Udeemy) для освоєння нових інструментів і технік DevOps. Студенти можуть отримати додаткові знання та практичний досвід.

Індивідуальні завдання. Завдання, що передбачають створення документів або сценаріїв для автоматизації певних етапів розробки (створення DockerFile, налаштування серверів CI/CD тощо).

4. Інтерактивні заняття

Обговорення реальних кейсів з галузі. Залучення представників компаній або експертів для обговорення кейсів, практичних проблем і тенденцій в області DevOps.

5. Семінари та майстер-класи

Семінари за участю фахівців. Проводяться для обговорення нових інструментів і методик, що використовуються в DevOps.

Майстер-класи. Практичні сесії, на яких студенти отримують конкретні навички, такі як налаштування серверів CI/CD, створення інфраструктури за допомогою Docker та Kubernetes.

Ці методи дозволяють створити інтерактивне та продуктивне середовище для навчання студентів та забезпечують необхідні практичні навички для роботи в сфері DevOps.

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

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

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

Критерії оцінювання роботи здобувача протягом семестру

Задовільно (60 – 74). Студент повинен здати індивідуальне завдання та основні лабораторні роботи, пройти тестування. Знати основи DevOps як методології, системи контролю версій (Git, SVN), концепції безперервної інтеграції (CI), основи використання API в DevOps, базові принципи DockerFile та контейнеризації. Мати розуміння ролі DevOps у продуктивній розробці, плануванні розгортання процесів. Знати вплив DevOps на архітектуру ПЗ та принципи автоматизації розробки. Вміти застосовувати основні інструменти DevOps для командної роботи, налаштування серверів збірки, створення API та DockerFile. Знати основи створення діаграм розгортання для планування DevOps процесів. Вміти класифікувати основні типи архітектури ПЗ (монолітна, мікросервісна, серверлесс).

Добре (75 – 89). Студент впевнено володіє знаннями з основ DevOps, інтеграції версій (Git, SVN), безперервної інтеграції та автоматизації процесів розробки. Вміє ефективно використовувати інструменти для CI/CD, налаштовувати сервери збірки, створювати API для автоматизації DevOps процесів. Має досвід роботи з DockerFile та оптимізацією контейнеризації для проектів. Добре розуміється на впливі DevOps на продуктивні команди та архітектуру програмного забезпечення, зокрема на класифікацію архітектури ПЗ, розрізняючи монолітні, мікросервісні, та серверлесс архітектури. Вміє планувати та впроваджувати розгортання DevOps процесів, створювати діаграми розгортання для візуалізації та організації процесів. Вміє аналізувати, яку архітектуру обрати для певного проекту в контексті DevOps.

Відмінно (90 – 100). Студент демонструє глибоке розуміння всіх аспектів DevOps, від методології та інструментів до впливу на архітектуру ПЗ. Має досконалі навички у використанні Git, налаштуванні CI/CD процесів, розробці API для інтеграції з DevOps, а також оптимізації DockerFile. Уміє впроваджувати та налаштовувати всі етапи DevOps у реальних проектах. Має глибоке розуміння ролі DevOps у продуктивній розробці, може ефективно працювати з командою для створення швидших та стабільніших процесів розробки. Вміє проектувати архітектуру ПЗ з урахуванням принципів DevOps, ефективно класифікує архітектури програмного забезпечення, визначаючи переваги та недоліки монолітних, мікросервісних, серверлесс архітектур в контексті DevOps. Також створює детальні діаграми розгортання для планування та моніторингу DevOps процесів у реальному середовищі.

Шкала оцінювання: бальна і традиційна

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

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&xt=no&theme_path=0%2C1665%2C14849%2C13426%2C10001&themes_basket=&ttp_themes_basket=&discipline_search=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:

<http://mentor.khai.edu/course/view.php?id=8433>

11. Рекомендована література

Базова література

1. Michael Hüttermann. *DevOps for Developers*. Apress, 2012. – 184 p.
2. Gene Kim, Jez Humble, Patrick Debois, John Willis. *The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win*. IT Revolution Press, 2013. – 432 p.
3. Jez Humble, David Farley. *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Addison-Wesley, 2010. – 456 p.

4. Nicole Forsgren, Jez Humble, Gene Kim. *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations*. IT Revolution Press, 2018. – 318 p.
5. Len Bass, Ingo Weber, Liming Zhu. *DevOps: A Software Architect's Perspective*. Addison-Wesley, 2015. – 338 p.
6. Version Control Systems [Електронний ресурс]. – Доступ: <https://www.geeksforgeeks.org/version-control-systems/>
7. What is Continuous Integration, Continuous Delivery, and Continuous Deployment? [Електронний ресурс]. – Доступ: <https://www.cloudbees.com/continuous-delivery/continuous-integration>

Додаткова література

1. Victor Farcic. *The DevOps 2.0 Toolkit: Automating the Continuous Deployment Pipeline with Containerized Microservices*. 2016. – 404 p.
2. John Willis, Damon Edwards. *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. IT Revolution Press, 2016. – 400 p.
3. Mandi Walls. *Building a DevOps Culture*. O'Reilly Media, 2013.
4. Mary Poppendieck, Tom Poppendieck. *Implementing Lean Software Development: From Concept to Cash*. Addison-Wesley, 2007.
5. Docker Documentation. *DockerFile Reference*. – Доступ: <https://docs.docker.com/engine/reference/builder/>
6. CI/CD with Jenkins [Електронний ресурс]. – Доступ: <https://www.jenkins.io/doc/>
7. Infrastructure as Code (IaC). *Learn about Infrastructure as Code*. – Доступ: <https://www.terraform.io/intro>
8. GitHub Guides. *GitHub Flow* – Доступ: <https://guides.github.com/introduction/flow/>

Інформаційні ресурси

1. DevOps.com. *DevOps Tutorials and Resources*. – Доступ: <https://www.devops.com/>
2. CI/CD Full Course | CI/CD Tutorial | Continuous Integration and Continuous Delivery | Simplilearn. – Доступ: <https://www.youtube.com/watch?v=h9K1NnqwUvE>
3. Arrested DevOps Podcast. – Доступ: <https://www.arresteddevops.com/>