

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КРЕМЕНЧУЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ МИХАЙЛА ОСТРОГРАДСЬКОГО
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЕЛЕКТРИЧНОЇ ІНЖЕНЕРІЇ
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ



МЕТОДИЧНІ ВКАЗІВКИ
ЩОДО САМОСТІЙНОЇ РОБОТИ
З НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
«ТЕХНОЛОГІЇ DEVOPS»
ДЛЯ СТУДЕНТІВ ДЕННОЇ ФОРМИ НАВЧАННЯ
ЗІ СПЕЦІАЛЬНОСТЕЙ:
122 – «КОМП'ЮТЕРНІ НАУКИ», 123 – «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ»
ОСВІТНЬО-ПРОФЕСІЙНИХ ПРОГРАМ:
«КОМП'ЮТЕРНІ НАУКИ», «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ»
ОСВІТНЬОГО СТУПЕНЯ «БАКАЛАВР»


Методичні вказівки щодо самостійної роботи з навчальної дисципліни «Технології DevOps» для студентів денної форми навчання зі спеціальностей: 122 – «Комп'ютерні науки», 123 – «Комп'ютерна інженерія» освітньо-професійних програм: «Комп'ютерні науки», «Комп'ютерна інженерія» освітнього ступеня «Бакалавр»

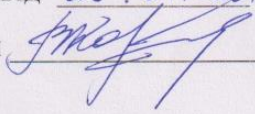
Укладачі: д. т. н., проф. А. Л. Перекрест;
асист. К. О. Вадурін

Рецензент к. т. н., доц. В. М. Сидоренко

Кафедра комп'ютерної інженерії та електроніки

Затверджено методичною радою Кременчуцького національного університету імені Михайла Остроградського

Протокол № 7 від 25.04.2024 року 

Голова методичної ради  проф. В. В. Костін

ЗМІСТ

Вступ.....	4
1 Теми та погодинний розклад лекцій і самостійної роботи з навчальної дисципліни	8
2 Перелік тем і питань, для самостійного опрацювання.....	9
3 Питання до модульного контролю	15
4 Критерії оцінювання знань студентів	21
Список літератури	22

ВСТУП

Зміст навчальної дисципліни полягає у розумінні головних ідей, на яких лежать ґрунтується DevOps, урахуваючи спільну відповідальність за розробку та експлуатацію, автоматизацію процесів; ознайомленні з популярними інструментами, які використовуються у DevOps, такими як системи керування конфігурацією, інструменти для автоматизації складання і розгортання, контейнеризація, системи моніторингу і журналювання; розробці та налагоджуванні пайплайнів для автоматизованого складання, тестування та розгортання програмного коду; створенні і керуванні контейнерами, та розгортання їх на платформах.

Метою навчальної дисципліни «Технології DevOps» є оволодіння знаннями та навичками, необхідними для впровадження та застосування DevOps-практик у процесі розробки та експлуатації програмного забезпечення, що дозволить їм ефективно співпрацювати з іншими фахівцями в рамках DevOps-команди для забезпечення швидкого та якісного випуску програмних продуктів.

У результаті вивчення навчальної дисципліни студент повинен

знати:

- принципи та концепції руху DevOps, у тому числі розуміння життєвого циклу DevOps;
- інструменти та технології, пов'язані з DevOps, такі як Jenkins, Docker, та Chef;
- процеси інтеграції з Jenkins, у тому числі створення та налагодження веб-додатків JEE з використанням Maven;
- створення коду та налагодження конвеєра складання, використовуючи плагін Building Pipeline та розгортання файлу WAR;
- процес огляду та встановлення контейнерів Docker;
- відмінності між віртуальними машинами та контейнерами;
- принципи хмарних обчислень та використання інструменту керування конфігурацією Chef;

уміти:

- розгортати та обслуговувати веб-додатки JEE з використанням Jenkins;
- створювати та налагоджувати конвеєри складання, використовуючи плагіни та інструменти DevOps;
- встановлювати та налагоджувати контейнери Docker для ефективного розгортання додатків;
- виконувати автоматизоване тестування, включаючи функціональне тестування з використанням Selenium та навантажувальне тестування з використанням Apache JMeter;
- налаштовувати наскрізну автоматизацію за допомогою Jenkins, Chef і AWS EC2;
- забезпечувати безпеку в Jenkins та VSTS, а також виконувати моніторинг Jenkins і Microsoft Azure.

Згідно з вимогами Стандарту вищої освіти України, першого (бакалаврського) рівня, галузі знань 12 Інформаційні технології, спеціальності 122 – «Комп’ютерні науки», навчальна дисципліна «Технології DevOps» забезпечує набуття здобувачами вищої освіти таких **компетентностей**.

ЗК 3. Знання та розуміння предметної області та розуміння професійної діяльності.

ЗК 9. Здатність працювати в команді.

ЗК 11. Здатність приймати обґрунтовані рішення.

ЗК 12. Здатність оцінювати та забезпечувати якість виконуваних робіт.

СК 8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об’єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

СК 9. Здатність реалізувати багаторівневу обчислювальну модель на основі архітектури клієнт-сервер, включаючи бази даних, знань і сховища даних, виконувати розподілену обробку великих наборів даних на кластерах стандартних серверів для забезпечення обчислювальних потреб користувачів, у тому числі на хмарних сервісах.

СК 10. Здатність застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника.

Програмні результати (ПР), які здобуває студент спеціальності 122 – «Комп'ютерні науки» під час вивчення цієї навчальної дисципліни:

ПР 11. Володіти навичками управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти розробляти проєктну документацію (техніко-економічне обґрунтування, технічне завдання, бізнес-план, угоду, договір, контракт).

ПР 13. Володіти мовами системного програмування та методами розробки програм, що взаємодіють з компонентами комп'ютерних систем, знати мережні технології, архітектури комп'ютерних мереж, мати практичні навички технології адміністрування комп'ютерних мереж та їх програмного забезпечення.

ПР 15. Застосовувати знання методології та CASE-засобів проєктування складних систем, методів структурного аналізу систем, об'єктно-орієнтованої методології проєктування при розробці і дослідженні функціональних моделей організаційно-економічних і виробничо-технічних систем.

Згідно з вимогами Стандарту вищої освіти України, першого (бакалаврського) рівня, галузі знань 12 Інформаційні технології, спеціальності 123 – «Комп'ютерна інженерія», навчальна дисципліна «Технології DevOps» забезпечує набуття здобувачами вищої освіти таких **компетентностей**.

ЗК 3. Здатність проводити дослідження на відповідному рівні.

ЗК 4. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ЗК 6. Здатність виявляти, ставити та вирішувати проблеми.

ЗК 7. Здатність приймати обґрунтовані рішення.

СК 2. Здатність розробляти алгоритмічне та програмне забезпечення, компоненти комп'ютерних систем та мереж, Інтернет додатків, кіберфізичних систем з використанням сучасних методів і мов програмування, а також засобів і систем автоматизації проектування.

СК 5. Здатність будувати архітектуру та створювати системне і прикладне програмне забезпечення комп'ютерних систем та мереж.

СК 6. Здатність використовувати та впроваджувати нові технології, включаючи технології розумних, мобільних, зелених і безпечних обчислень, брати участь в модернізації та реконструкції комп'ютерних систем та мереж, різноманітних вбудованих і розподілених додатків, зокрема з метою підвищення їх ефективності.

СК 8. Здатність забезпечувати якість продуктів і сервісів інформаційних технологій на протязі їх життєвого циклу.

СК 11. Здатність обирати ефективні методи розв'язування складних задач комп'ютерної інженерії, критично оцінювати отримані результати та аргументувати прийняті рішення.

Програмні результати (РН), які здобуває студент спеціальності 123 – «Комп'ютерна інженерія» під час вивчення цієї навчальної дисципліни:

РН 5. Розробляти і реалізовувати проекти у сфері комп'ютерної інженерії та дотичні до неї міждисциплінарні проекти з урахуванням інженерних, соціальних, економічних, правових та інших аспектів.

РН 8. Застосовувати знання технічних характеристик, конструктивних особливостей, призначення і правил експлуатації програмно-технічних засобів комп'ютерних систем та мереж для вирішення складних задач комп'ютерної інженерії та дотичних проблем.

РН 11. Приймати ефективні рішення з питань розроблення, впровадження та експлуатації комп'ютерних систем і мереж, аналізувати альтернативи, оцінювати ризики та імовірні наслідки рішень.

РН 13. Зрозуміло і недвозначно доносити власні знання, висновки та аргументацію з питань інформаційних технологій і дотичних міжгалузевих питань до фахівців і нефахівців, зокрема до осіб, які навчаються.

1 ТЕМИ І ПОГОДИННИЙ РОЗКЛАД ЛЕКЦІЙ ТА САМОСТІЙНОЇ РОБОТИ З НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Назви змістових модулів і тем	Кількість годин	
	денна форма навчання	
	у тому числі	
	лк	с. р.
Модуль 1 (семестр б)		
Змістовий модуль 1 Концепції DevOps, Jenkins, Docker, Chef		
Тема 1 Концепції, інструменти та технології DevOps	2	8
Тема 2 Інтеграція з Jenkins	2	8
Тема 3 Створення коду та налагоджування конвеєра складання	2	8
Тема 4 Контейнери	4	10
Тема 5 Хмарні обчислення та керування конфігурацією	4	10
Індивідуальне завдання 1	–	4
Контрольна робота 1 (тест)	–	2
Усього годин за змістовим модулем 1	14	50
Змістовий модуль 2 Автоматизація та забезпечення безпеки		
Тема 6 Автоматизоване тестування	2	10
Тема 7 Наскрізна автоматизація	4	16
Тема 8 Безпека та моніторинг	4	18
Індивідуальне завдання 2	–	4
Контрольна робота 2 (тест)	–	2
Усього годин за змістовим модулем 2	10	50
Забезпечення семестрового контролю: диф. залік	–	–
Усього годин за модулем 1	24	100

2 ПЕРЕЛІК ТЕМ І ПИТАНЬ З НАВЧАЛЬНОЇ ДИСЦИПЛІНИ ДЛЯ САМОСТІЙНОГО ОПРАЦЮВАННЯ

Модуль 1 (семестр 6)

Змістовий модуль 1 Концепції DevOps, Jenkins, Docker, Chef

Тема 1 Концепції, інструменти та технології DevOps

1. Розуміння руху DevOps.
2. Життєвий цикл DevOps.
3. Інструменти та технології DevOps.

Питання для самоперевірки

1. Що таке DevOps і яке його призначення?
2. Які головні принципи руху DevOps?
3. Які етапи містить життєвий цикл DevOps?
4. Які головні завдання на кожному етапі життєвого циклу DevOps?
5. Які інструменти використовуються для автоматизації процесів у DevOps?
6. Наведіть приклади інструментів для кожної фази життєвого циклу DevOps.
7. Які технології підтримують DevOps процеси?
8. Які переваги має впровадження DevOps для розробки програмного забезпечення?
9. Які можуть бути виклики під час упровадження DevOps і як їх подолати?
10. Яким чином DevOps сприяє поліпшенню співпраці між розробниками і операторами?
11. Як ви оцінюєте успішність і результативність упровадження DevOps у вашому проєкті або організації?

Література: [1–13].

Тема 2 Інтеграція з Jenkins

1. Установлення Jenkins.
2. Створення та налагоджування веб-додатків JEE на основі Maven.
3. Інтеграція Jenkins і SonarQube.

Питання для самоперевірки

1. Як установити Jenkins на сервері?
2. Які кроки потрібно виконати для налагодження Jenkins після встановлення?
3. Як створити новий проєкт (job) у Jenkins?
4. Які головні параметри конфігурації проєкту у Jenkins?
5. Як підключити Maven до проєкту в Jenkins?
6. Які головні кроки для налаштування веб-додатків JEE у Maven?
7. Як створити автоматичний процес складання, тестування та розгортання веб-додатка в Jenkins?
8. Як налагодити інтеграцію Jenkins і SonarQube?
9. Як перевірити результати аналізу коду, який виконує SonarQube, у Jenkins?
10. Які інструменти аналізу коду можна використовувати разом з Jenkins для покращення якості коду?

Література: [1–13].

Тема 3 Створення коду та налагоджування конвеєра складання

1. Створення вбудованих конвеєрів доставки.
2. Плагін Building Pipeline.
3. Розгортання файлу WAR.

Питання для самоперевірки

1. Що таке конвеєр складання та яке призначення він має?
2. Які інструменти можна використовувати для створення вбудованих конвеєрів доставки?

3. Які переваги та недоліки вбудованих конвеєрів доставки порівняно з іншими методами доставки?
4. Що таке плагін Building Pipeline та яке його призначення у процесі створення конвеєра складання?
5. Які головні кроки для налагодження конвеєра складання з використанням плагіна Building Pipeline?
6. Які інструменти або сервіси можна використовувати для розгортання файлу WAR?
7. Які кроки потрібно виконати для розгортання файлу WAR за допомогою вибраного інструменту чи сервісу?
8. Які можливі проблеми або помилки можуть виникнути під час створення конвеєра складання чи розгортання файлу WAR, і як їх розв'язати?
9. Які підходи можна використовувати для автоматизації тестування в процесі конвеєра складання?
10. Як можна організувати процес складання та розгортання, щоб забезпечити безпеку та стабільність продукту?

Література: [1–13].

Тема 4 Контейнери

1. Огляд контейнерів Docker.
2. Різниця між віртуальними машинами та контейнерами.
3. Установлення та налагодження Docker.

Питання для самоперевірки

1. Що таке контейнери Docker і як вони працюють?
2. Яка головна мета використання контейнерів у сучасних розробках?
3. Які переваги надає використання контейнерів порівняно з віртуальними машинами?
4. Яка різниця між віртуальною машиною і контейнером?
5. Що таке Docker та які головні можливості цього інструменту?
6. Як установити Docker на операційну систему (наприклад, Linux або

Windows)?

7. Які головні кроки для налагодження Docker після встановлення?

8. Які команди можна використовувати для керування контейнерами в Docker?

9. Як виконати розгортання та запуск контейнера з використанням Docker?

10. Які є найбільш поширені використання контейнерів у сфері розробки програмного забезпечення?

Література: [1–13].

Тема 5 Хмарні обчислення та керування конфігурацією

1. Огляд інструменту керування конфігурацією Chef.

2. Установлення та налагодження робочої станції Chef.

Питання для самоперевірки

1. Що таке керування конфігурацією і яке його значення в інфраструктурі хмарних обчислень?

2. Які головні переваги використання інструментів керування конфігурацією, таких як Chef, порівняно з ручними методами?

3. Які є ключові концепції та терміни, пов'язані з Chef?

4. Як ви можете описати архітектуру Chef та її компоненти?

5. Які головні функції Chef та як вони співпрацюють у процесі керування конфігурацією?

6. Як установити Chef на робочу станцію та що для цього потрібно?

7. Які головні кроки налагодження робочої станції Chef після її встановлення?

8. Які є головні інструменти та ресурси Chef, які можна використовувати для опису конфігурації системи?

9. Які є загальні практики та поради щодо написання рецептів Chef для керування конфігурацією?

10. Як відбувається тестування конфігурації за допомогою Chef і які інструменти використовуються для автоматизованого тестування?

Література: [1–13].

Змістовий модуль 2 Автоматизація та забезпечення безпеки

Тема 6 Автоматизоване тестування

1. Функціональне тестування з використанням Selenium.

2. Навантажувальне тестування за допомогою тесту на основі URL-адреси та Apache JMeter для Microsoft.

Питання для самоперевірки

1. Що таке автоматизоване тестування і чому воно важливе для розробки програмного забезпечення?

2. Які переваги має функціональне тестування порівняно з ручним тестуванням?

3. Що таке Selenium і як він використовується для функціонального тестування?

4. Які головні компоненти Selenium і як вони взаємодіють між собою?

5. Як створити тестовий сценарій з використанням Selenium WebDriver?

6. Які інструменти доступні для навантажувального тестування?

7. Які головні принципи навантажувального тестування?

8. Що таке тест на основі URL-адреси і як він використовується для навантажувального тестування?

9. Як Apache JMeter допомагає в проведенні навантажувальних тестів?

10. Які метрики використовуються для оцінювання результатів навантажувального тестування?

11. Які можуть бути проблеми під час навантажувального тестування і як їх розв'язувати?

12. Які стратегії використовуються для оптимізації навантажувальних тестів?

Література: [1–13].

Тема 7 Наскрізна автоматизація

1. Наскрізна автоматизація за допомогою Jenkins, Chef і AWS EC2.

Питання для самоперевірки

1. Що таке наскрізна автоматизація і чому вона важлива для процесу розробки та впровадження програмного забезпечення?

2. Які головні переваги використання Jenkins у процесі наскрізної автоматизації?

3. Які інструменти та можливості Chef надає для автоматизації інфраструктури?

4. Які є переваги використання AWS EC2 для наскрізної автоматизації порівняно з іншими хмарними сервісами?

5. Які кроки необхідно виконати для створення та налагодження конфігурації Jenkins для автоматизованого розгортання програмного забезпечення на AWS EC2?

6. Як Chef і Jenkins можуть працювати разом у процесі наскрізної автоматизації?

7. Які стратегії резервного копіювання та відновлення даних можна реалізувати з використанням Jenkins, Chef і AWS EC2?

8. Які є головні кроки для створення та налагодження автоматизованого масштабування інфраструктури за допомогою Jenkins, Chef і AWS EC2?

9. Як можна забезпечити безпеку процесу наскрізної автоматизації з використанням засобів, що надаються Jenkins, Chef і AWS EC2?

10. Як можна виміряти ефективність та результативність наскрізної автоматизації проєкту та виявити можливі способи її оптимізації?

Література: [1–13].

Тема 8 Безпека та моніторинг

1. Безпека в Jenkins і VSTS.

2. Керування користувачами в Jenkins.

3. Моніторинг Jenkins і Microsoft Azure.

Питання для самоперевірки

1. Які головні методи безпеки в Jenkins і VSTS?
2. Як можна захистити Jenkins від несанкціонованого доступу?
3. Як вибрати найбільш безпечний метод аутентифікації для Jenkins і VSTS?
4. Які кроки потрібно вжити для захисту конфіденційних даних в Jenkins і VSTS?
5. Як настроїти ролі та дозволи користувачів у Jenkins?
6. Що таке механізми авторизації в Jenkins, і як вони працюють?
7. Як ви можете забезпечити безпеку плагінів в Jenkins?
8. Як видалити або заблокувати користувача в Jenkins?
9. Як установити моніторинг Jenkins для виявлення аномалій?
10. Які інструменти або плагіни можна використовувати для моніторингу Jenkins і Microsoft Azure?
11. Які головні метрики важливо відслідковувати в процесі моніторингу Jenkins і Microsoft Azure?
12. Як автоматизувати процес моніторингу в Jenkins і Microsoft Azure?
13. Як забезпечити надійність моніторингу в умовах великої кількості даних Jenkins і Microsoft Azure?
14. Як виявити та усунути проблеми з безпекою в Jenkins та VSTS за допомогою моніторингу?
15. Як забезпечити взаємодію між моніторингом Jenkins і Microsoft Azure для виявлення загроз?

Література: [1–13].

3 ПИТАННЯ ДО МОДУЛЬНОГО КОНТРОЛЮ

Модуль 1 (семестр 1)

1. Що таке DevOps і яке його призначення?
2. Які головні принципи руху DevOps?

3. Які етапи містить життєвий цикл DevOps?
4. Які головні завдання на кожному етапі життєвого циклу DevOps?
5. Які інструменти використовуються для автоматизації процесів у DevOps?
6. Наведіть приклади інструментів для кожної фази життєвого циклу DevOps.
7. Які технології підтримують DevOps процеси?
8. Які переваги має впровадження DevOps для розробки програмного забезпечення?
9. Які можуть бути виклики під час упровадження DevOps і як їх подолати?
10. Як DevOps сприяє поліпшенню співпраці між розробниками і операторами?
11. Як ви оцінюєте успішність і результативність упровадження DevOps у вашому проєкті або організації?
12. Як установити Jenkins на сервері?
13. Які кроки потрібно виконати для налагодження Jenkins після встановлення?
14. Як створити новий проєкт (job) у Jenkins?
15. Які головні параметри конфігурації проєкту у Jenkins?
16. Як підключити Maven до проєкту в Jenkins?
17. Які головні кроки для налагодження веб-додатків JEE у Maven?
18. Як створити автоматичний процес складання, тестування та розгортання веб-додатка в Jenkins?
19. Як настроїти інтеграцію Jenkins і SonarQube?
20. Як перевірити результати аналізу коду, який виконує SonarQube, у Jenkins?
21. Які інструменти аналізу коду можна використовувати разом з Jenkins для покращення якості коду?
22. Що таке конвеєр складання та яке призначення він має?
23. Які інструменти можна використовувати для створення вбудованих конвеєрів доставки?

24. Які переваги та недоліки вбудованих конвеєрів доставки порівняно з іншими методами доставки?
25. Що таке плагін Building Pipeline та яке його призначення у процесі створення конвеєра складання?
26. Які головні кроки для налагодження конвеєра складання з використанням плагіна Building Pipeline?
27. Які інструменти або сервіси можна використовувати для розгортання файлу WAR?
28. Які кроки потрібно виконати для розгортання файлу WAR за допомогою вибраного інструменту чи сервісу?
29. Які можливі проблеми або помилки можуть виникнути під час створення конвеєра складання чи розгортання файлу WAR, і як їх розв'язати?
30. Які підходи можна використовувати для автоматизації тестування в процесі конвеєра складання?
31. Як можна організувати процес складання та розгортання, щоб забезпечити безпеку та стабільність продукту?
32. Що таке контейнери Docker і як вони працюють?
33. Яка головна мета використання контейнерів у сучасних розробках?
34. Які переваги надає використання контейнерів порівняно з віртуальними машинами?
35. Яка різниця між віртуальною машиною і контейнером?
36. Що таке Docker та які головні можливості цього інструменту?
37. Як установити Docker на операційну систему (наприклад, Linux або Windows)?
38. Які головні кроки для налагодження Docker після встановлення?
39. Які команди можна використовувати для керування контейнерами в Docker?
40. Як виконати розгортання та запуск контейнера з використанням Docker?
41. Які є найбільш поширені використання контейнерів у сфері розробки програмного забезпечення?

42. Що таке керування конфігурацією і яке його значення в інфраструктурі хмарних обчислень?
43. Які головні переваги використання інструментів керування конфігурацією, таких як Chef, порівняно з ручними методами?
44. Які є ключові концепції та терміни, пов'язані з Chef?
45. Як ви можете описати архітектуру Chef та її компоненти?
46. Які головні функції Chef та як вони співпрацюють у процесі керування конфігурацією?
47. Як установити Chef на робочу станцію та що для цього потрібно?
48. Які головні кроки налагоджування робочої станції Chef після її встановлення?
49. Які є головні інструменти та ресурси Chef, які можна використовувати для опису конфігурації системи?
50. Які є загальні практики та поради щодо написання рецептів Chef для керування конфігурацією?
51. Як відбувається тестування конфігурації за допомогою Chef і які інструменти використовують для автоматизованого тестування?
52. Що таке автоматизоване тестування і чому воно важливе для розробки програмного забезпечення?
53. Які переваги має функціональне тестування порівняно з ручним тестуванням?
54. Що таке Selenium і як він використовується для функціонального тестування?
55. Які головні компоненти Selenium і як вони взаємодіють між собою?
56. Як створити тестовий сценарій з використанням Selenium WebDriver?
57. Які інструменти доступні для навантажувального тестування?
58. Які головні принципи навантажувального тестування?
59. Що таке тест на основі URL-адреси і як він використовується для навантажувального тестування?
60. Як Apache JMeter допомагає в проведенні навантажувальних тестів?
61. Які метрики використовуються для оцінювання результатів навантажувального тестування?

62. Які можуть бути проблеми під час навантажувального тестування і як їх розв'язувати?
63. Які стратегії використовуються для оптимізації навантажувальних тестів?
64. Що таке наскрізна автоматизація і чому вона важлива для процесу розробки та впровадження програмного забезпечення?
65. Які головні переваги використання Jenkins у процесі наскрізної автоматизації?
66. Які інструменти та можливості Chef надає для автоматизації інфраструктури?
67. Які є переваги використання AWS EC2 для наскрізної автоматизації порівняно з іншими хмарними сервісами?
68. Які кроки необхідно виконати для створення та налагодження конфігурації Jenkins для автоматизованого розгортання програмного забезпечення на AWS EC2?
69. Як Chef і Jenkins можуть працювати разом у процесі наскрізної автоматизації?
70. Які стратегії резервного копіювання та відновлення даних можна реалізувати з використанням Jenkins, Chef і AWS EC2?
71. Які є головні кроки для створення та налагодження автоматизованого масштабування інфраструктури за допомогою Jenkins, Chef і AWS EC2?
72. Як можна забезпечити безпеку процесу наскрізної автоматизації з використанням засобів, що надаються Jenkins, Chef і AWS EC2?
73. Як можна виміряти ефективність та результативність наскрізної автоматизації існуючого проєкту та виявити можливі способи її оптимізації?
74. Які головні методи безпеки в Jenkins і VSTS?
75. Як можна захистити Jenkins від несанкціонованого доступу?
76. Як вибрати найбільш безпечний метод аутентифікації для Jenkins і VSTS?
77. Яких заходів потрібно вжити для захисту конфіденційних даних в Jenkins і VSTS?
78. Як настроїти ролі та дозволи користувачів у Jenkins?

79. Що таке механізми авторизації в Jenkins, і як вони працюють?
80. Як можна забезпечити безпеку плагінів у Jenkins?
81. Як видалити або заблокувати користувача в Jenkins?
82. Як установити моніторинг Jenkins для виявлення аномалій?
83. Які інструменти або плагіни можна використовувати для моніторингу Jenkins і Microsoft Azure?
84. Які головні метрики важливо відслідковувати в процесі моніторингу Jenkins і Microsoft Azure?
85. Як автоматизувати процес моніторингу в Jenkins і Microsoft Azure?
86. Як забезпечити надійність моніторингу в умовах великої кількості даних Jenkins і Microsoft Azure?
87. Як виявити та усунути проблеми щодо безпеки в Jenkins та VSTS за допомогою моніторингу?
88. Як забезпечити взаємодію між моніторингом Jenkins і Microsoft Azure для виявлення загроз?

4 КРИТЕРІЇ ОЦІНЮВАННЯ ЗНАТЬ СТУДЕНТІВ

Вид роботи	Зміст	Бали
Робота на лекціях	Активна участь у дискусіях, розв'язання практичних прикладів тощо	15
Завдання з ЛР (відвідування, активність, опитування, виконання індивідуальних завдань, перевірка самостійної роботи)	Взаємодія та специфіка роботи з GitHub. Створення першого репозиторію	5
	AWS, розгортання типової серверної архітектури	5
	Огляд контейнерів Docker. Установлення та налагоджування Docker	5
	Створення коду та налагоджування конвеєра складання	5
	Установлення Chef Workstation і сервера на Ubuntu. Налagodжування сервера Chef за замовчанням	5
	Безпека та моніторинг. Безпека в Jenkins і VSTS. Керування користувачами в Jenkins. Моніторинг Jenkins і Microsoft Azure	5
	Наскрізна автоматизація	5
Поточний контроль	Індивідуальні завдання 1, 2	30
Семестровий контроль	Контрольні роботи 1, 2 (тести)	20
	Усього балів	100

СПИСОК ЛІТЕРАТУРИ

Основна

1. Armstrong S. DevOps for Networking. Birmingham, England: Packt Publishing, 2023. P. 185. ISBN: 9781786464859.
2. Duffy M. DevOps Automation Cookbook. Birmingham, England: Packt Publishing, 2015. P. 334. ISBN: 9781784392826.
3. Giridhar C. Automate it! – Recipes to upskill your business. Birmingham, England: Packt Publishing, 2017. P. 383. ISBN: 9781786464859.
4. Soni M. Jenkins Essentials. Birmingham, England: Packt Publishing, 2015. P. 186. ISBN: 9781783553471.
5. Soni M. DevOps Bootcamp. Birmingham, England: Packt Publishing, 2023. P. 309. ISBN: 9781787285965.
6. Soni M. DevOps for Web Development. Birmingham, England: Packt Publishing, 2023. P. 301. ISBN: 9781786465702.

Додаткова

7. Sydorenko V., Perekrest A., Shendryk V., Shendryk S. Machine Learning Optimization of Air Heating Time in the Heating Control System of a Smart House. *New Technologies, Development and Application. Lecture Notes in Networks and Systems*. 2023, vol. 707, P. 36–44. (scopus, Q4).
8. Коростельов А. С., Гученко, М. І., Перекрест А. Л., Самойлов А. М., Вадурін К. О. Аналітичні розрахунки корпоративної мережі базованої на технологіях інтернету речей підприємства з екологічних досліджень. *Вчені записки ТНУ імені В. І. Вернадського. Серія: Технічні науки*. Том 34 (73) № 5, 2023, С. 140–148. (фахове видання).
9. Korostelov A., Guchenko M., Perekrest A., Nikitina A., Vadurin, K. Модель корпоративної мережі базованої на технологіях інтернету речей підприємства з екологічних досліджень. *Системи управління, навігації та зв'язку: збірник наукових праць*. Полтава: ПНТУ, Т. 3 (73), 2023. С. 111–114. (фахове видання).

10. Перекрест А. Л., Бахарєв В. С., Вадурін К. О., Дерієнко А. І., Іващенко А. В., Шкарупа С. А. Розробка бази даних для зберігання показників стану атмосферного повітря з дослідних станцій комунального підприємства. *Проблеми інформатизації та управління*. Київ: НАУ, 2023. Випуск 3, № 75, 2023. С. 68–86. (фахове видання)

11. Перекрест А. Л., Вадурін К. О., Юдіна А. Л. Цифровий автомат як інструмент моделювання стану суспільства. *Вісник Кременчуцького національного університету імені Михайла Остроградського*. Кременчук: КрНУ, 2023. Випуск 3 (140). С. 52–61. (фахове видання).

12. Перекрест А. Л., Дружиніна В. В., Морозов Ю. О., Ноженко В. Ю. Використання технологій доповненої та віртуальної реальності під час створення інноваційної екосистеми громад. *Вісник Кременчуцького національного університету імені Михайла Остроградського*. Кременчук: КрНУ, 2023. Випуск 3 (140). С. 62–73. (фахове видання).

13. Chencheva O., Lashko Y., Rieznik D., Perekrst A., Vozhyk M. Development and research of the functional possibilities of the automated fuzzy indoor air quality management system of production premises. *Labour Protection Problems in Ukraine*, 2023, 39 (3–4), P. 36–42.

Інформаційні ресурси

14. DevOps – The Web's Largest Collection of DevOps Content. DevOps.com. URL: <https://devops.com/> (дата звернення: 26.03.2023).

Методичні вказівки щодо самостійної роботи з навчальної дисципліни «Технології DevOps» для студентів денної форми навчання зі спеціальностей: 122 – «Комп'ютерні науки», 123 – «Комп'ютерна інженерія» освітньо-професійних програм: «Комп'ютерні науки», «Комп'ютерна інженерія» освітнього ступеня «Бакалавр»

Укладачі: д. т. н., проф. А. Л. Перекрест;
асист. К. О. Вадурін

Відповідальний за випуск д. т. н., проф. М. І. Гученко

Підп. до др. _____. Формат 60×84 1/16. Папір тип. Друк ризографія.
Ум. друк. арк. _____. Наклад _____ прим. Зам. № _____. Безкоштовно.

Редакційно-видавничий відділ
Кременчуцького національного університету
імені Михайла Остроградського
вул. Університетська 20, м. Кременчук, 39600