

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Львівський національний університет імені Івана Франка**  
**Факультет електроніки та комп'ютерних технологій**  
**Кафедра радіоелектронних і комп'ютерних систем**

**Затверджено**

На засіданні  
кафедри радіоелектронних і  
комп'ютерних систем  
факультету електроніки та комп'ютерних  
технологій  
Львівського національного університету  
імені Івана Франка  
(протокол № 1/2 від 28.08. 2023 р.)

Завідувач кафедри:

  
Ігор ОЛЕНИЧ

**Силабус з навчальної дисципліни**  
**“DevOps”,**  
**що викладається в межах ОПП**  
**“ Високопродуктивний комп'ютинг ”**  
**першого (бакалаврського) рівня вищої освіти для здобувачів з**  
**спеціальності 121 – Інженерія програмного забезпечення**

Львів 2023 р.

<b>Назва дисципліни</b>	DevOps
<b>Адреса викладання дисципліни</b>	Корпус факультету електроніки та комп'ютерних технологій, Львівський національний університет імені Івана Франка, вул. Драгоманова 50, м. Львів, 79005, вул. Ген. Тарнавського 107, м. Львів, 79011
<b>Факультет та кафедра, за якою закріплена дисципліна</b>	Факультет електроніки та комп'ютерних технологій Кафедра радіоелектронних і комп'ютерних систем
<b>Галузь знань, шифр та назва спеціальності</b>	12 – інформаційні технології 121 – Інженерія програмного забезпечення
<b>Викладачі дисципліни</b>	Оленич Ігор Богданович, д-р фіз.-мат. наук, професор Флюнт О. Є., канд. ф.-м. н., доцент
<b>Контактна інформація викладачів</b>	<a href="mailto:igor.olenych@lnu.edu.ua">igor.olenych@lnu.edu.ua</a> <a href="https://electronics.lnu.edu.ua/employee/olenych-i-b/">https://electronics.lnu.edu.ua/employee/olenych-i-b/</a> <a href="mailto:orest.fluint@lnu.edu.ua">orest.fluint@lnu.edu.ua</a> <a href="http://electronics.lnu.edu.ua/employee/fluint-orest-yevhenovych">http://electronics.lnu.edu.ua/employee/fluint-orest-yevhenovych</a>
<b>Консультації з питань навчання по дисципліні відбуваються</b>	Консультації в день проведення лекцій/лабораторних занять (за попередньою домовленістю): ауд. 101, корпус факультету електроніки та комп'ютерних технологій, вул. Драгоманова 50, м. Львів
<b>Сторінка курсу</b>	<a href="http://e-learning.lnu.edu.ua/course/view.php?id=3136">http://e-learning.lnu.edu.ua/course/view.php?id=3136</a>
<b>Коротка анотація дисципліни</b>	Дисципліна „DevOps” є нормативною дисципліною з спеціальності 121 Інженерія програмного забезпечення для освітньої програми Високопродуктивний комп'ютерин, яка викладається в 7 семестрі в обсязі 3.5 кредита (за Європейською Кредитно-Трансферною Системою ECTS).
<b>Інформація про дисципліну</b>	Курс розроблено таким чином, щоб надати учасникам необхідні знання, обов'язкові для того, щоб бути обізнаними з сучасними підходами та засобами неперервного розгортання обчислювальних систем. Тому у курсі представлено як огляд концепцій неперервних розробки, постачання і розгортання, так і процесів та інструментів, які потрібні для організації контролю версій програмного забезпечення, створення образів і розміщення на репозиторіях, безперервного розгортання обчислювальних систем та програмного забезпечення.
<b>Мета та цілі дисципліни</b>	<b>Метою</b> вивчення нормативної дисципліни „DevOps” є ознайомлення студентів із сучасними практиками контролю версій, методами перевірки якості програмних додатків, неперервного постачання та розгортання програмних додатків для оволодіння сучасними підходами та інструментами безперервної розробки та безперервного розгортання. <b>Цілі:</b> Для досягнення поставленої у навчальному курсі мети студентам потрібно ознайомитись з системою контролю версій Git, системою автоматизованої перевірки якості Selenium, системою віртуалізації та контейнеризації Docker, системою неперервного розгортання Kubernetes, системою контролю процесів Jenkins і ін. Після проходження навчання за цією програмою

	студенти повинні вміти реалізувати цикл розгортання простого програмного забезпечення від розробки та тестування до автоматизованого розгортання.
<b>Література для вивчення дисципліни</b>	<p>Основна література</p> <ol style="list-style-type: none"> <li>1. McQuaid M., Chacon S. Git in practice. – Manning, 2015. <a href="https://www.manning.com/books/git-in-practice_a_bid=5688bbf4&amp;a_aid=MikeMcQuaid#toc">https://www.manning.com/books/git-in-practice_a_bid=5688bbf4&amp;a_aid=MikeMcQuaid#toc</a></li> <li>2. Chacon S. Git internals // Pro Git. – 2009. – P. 223-250. <a href="file:///home/orest/Downloads/peepcode-git.pdf">file:///home/orest/Downloads/peepcode-git.pdf</a></li> <li>3. Farhan Hasin Chowdhury The Docker Handbook <a href="https://www.freecodecamp.org/news/thedocker-handbook/">https://www.freecodecamp.org/news/thedocker-handbook/</a></li> <li>4. Jenkins Handbook <a href="http://www.jenkins.io/doc/book/">www.jenkins.io/doc/book/</a></li> <li>5. Kubernetes Handbook <a href="https://tanzu.vmware.com/kubernetes-handbook">https://tanzu.vmware.com/kubernetes-handbook</a></li> <li>6. Kubernetes Documentation <a href="https://kubernetes.io/docs/home/">https://kubernetes.io/docs/home/</a></li> </ol> <p>Додаткова література</p> <ol style="list-style-type: none"> <li>1. Chacon S., Straub B. Pro git. – Springer Nature, 2014. – P. 456.</li> <li>2. Kubernetes Handbook Non-Programmer's Guide to Deploy Applications with Kubernetes Stephen Fleming</li> <li>3. Docker Engine overview <a href="https://docs.docker.com/engine/">https://docs.docker.com/engine/</a></li> <li>4. Jenkins User Handbook <a href="mailto:jenkinsci-docs@googlegroups.com">jenkinsci-docs@googlegroups.com</a></li> </ol>
<b>Обсяг курсу</b>	Загальне навантаження: 105 годин. 64 годин аудиторних занять. З них 32 годин лекцій, 32 години лабораторних робіт та 118 годин самостійної роботи.
<b>Очікувані результати навчання</b>	<p>Після завершення цього курсу студент буде:</p> <ul style="list-style-type: none"> <li>- Знати основні поняття, визначення, процеси та підходи до тестування програмного забезпечення, сучасні системи та підходи до контролю версій програмних додатків, методи та практики автоматизації запуску програмних процесів, основи віртуалізації за допомогою контейнерів, основи роботи обчислювальних кластерів.</li> <li>- Вміти працювати з системами контролю версій програмного забезпечення, проводити тестування програмного забезпечення різних видів та рівнів, автоматизувати запуск різних програм та писати власні скрипти, створювати образи власних програмних додатків та завантажувати їх на репозиторії, віртуалізувати процеси за допомогою контейнерів, створювати та налаштовувати кластери, керувати їхньою роботою.</li> </ul> <p>Після вивчення даного курсу «DevOps» здобувачі набудуть таких Загальних(ЗК)/Фахових(ФК) компетентностей та Програмних результатів навчання (ПРН):</p> <p>ЗК01. Здатність до абстрактного мислення, аналізу та синтезу.</p> <p>ЗК02. Здатність застосовувати знання у практичних ситуаціях.</p> <p>ЗК04. Здатність спілкуватися іноземною мовою як усно, так і письмово.</p> <p>ЗК05. Здатність вчитися і оволодівати сучасними знаннями.</p> <p>ЗК06. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.</p> <p>ФК14. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.</p> <p>ФК17. Здатність дотримуватися специфікацій, стандартів, правил і</p>

	<p>рекомендацій в професійній галузі при реалізації процесів життєвого циклу.</p> <p>ФК18. Здатність аналізувати, вибирати і застосовувати методи і засоби для забезпечення інформаційної безпеки (в тому числі кібербезпеки).</p> <p>ФК23. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.</p> <p>ФК24. Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення.</p> <p>ФК25. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.</p> <p>ФК27. Здатність розробляти високопродуктивні програмні комплекси для вирішення задач наук про дані, систем штучного інтелекту, вбудованих та інших інноваційних систем.</p> <p>ФК28. Володіння методами розроблення і впровадження систем підвищеної продуктивності, серверних, мікросервісних, хмаркових, розподілених та інших новітніх технологій.</p> <p>ФК29. Здатність здійснювати розробку програмних продуктів використовуючи різні методології і технології із забезпеченням їх інноваційності та високої продуктивності.</p> <p>ПРН02. Знати кодекс професійної етики, розуміти соціальну значимість та культурні аспекти інженерії програмного забезпечення і дотримуватись їх в професійній діяльності.</p> <p>ПРН03. Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.</p> <p>ПРН04. Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення.</p> <p>ПРН12. Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення.</p> <p>ПРН14. Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення.</p> <p>ПРН15. Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.</p> <p>ПРН16. Мати навички командної розробки, погодження, оформлення і випуску всіх видів програмної документації.</p> <p>ПРН21. Знати, аналізувати, вибирати, кваліфіковано застосовувати засоби забезпечення інформаційної безпеки (в тому числі кібербезпеки) і цілісності даних відповідно до розв'язуваних прикладних завдань та створюваних програмних систем.</p> <p>ПРН22. Знати та вміти застосовувати методи та засоби управління проектами.</p> <p>ПРН25. Вміти застосовувати інноваційні технологічні рішення при розробці високопродуктивних систем.</p> <p>ПРН26. Знати засоби інтеграції, розгортання та підтримки спеціалізованих програмних компонентів, розроблених на основі інноваційних технологій для вирішення завдань високопродуктивних технологій.</p>
<b>Ключові слова</b>	<p>Безперервне розгортання, система контролю версій, репозиторій, докер, контейнер, том, под, віртуалізація, Git, GitLab, GitHub, Docker, Jenkins,</p>

	Kubernetes
<b>Формат курсу</b>	Очний
<b>Теми</b>	Див. <b>Схема курсу</b>
<b>Підсумковий контроль, форма</b>	Екзамен в кінці семестру
<b>Пререквізити</b>	Для вивчення курсу студенти потребують базових знань з „Операційні системи”, „Основи програмування”, „Бази даних та знань дисциплін,” „Основи Web програмування”, достатніх для сприйняття категоріального апарату віртуалізації, системи контролю версій, основ функціонування операційних систем, зокрема Linux, bash оболонки команд.
<b>Навчальні методи та техніки, які будуть використовуватися під час викладання курсу</b>	Інформаційні методи (лекції, презентації, лабораторні роботи, написання рефератів, виконання індивідуальних завдань, робота у групі, командна робота, обговорення, консультації для поглибленого розуміння тем, бесіда, ілюстрація, демонстрація), дедуктивні методи на основі узагальнень, евристичні методи (проблемна лекція), інтерактивні методи (дискусія).
<b>Необхідне обладнання</b>	<p>Для проведення лекційних занять: комп'ютер (мінімальні характеристики: процесор Intel Core i3(4 ядра/8 потоків), 8ГБ оперативної пам'яті, 50ГБ вільного місця на диску, доступ до мережі Internet, засоби мультимедіа (в т.ч. проектор).</p> <p>Для проведення лабораторних занять: Комп'ютер (мінімальні характеристики: процесор Intel Core i3(4 ядра/8 потоків), 8ГБ оперативної пам'яті, 50ГБ вільного місця на диску. Необхідне програмне забезпечення включає в себе ОС Windows 10, або Ubuntu 22.04 LTS, середовище розробки MS Visual Studio/PyCharm/Vim/VS Code, компілятор мови програмування C++/Python.</p> <p>Вільне спеціальне програмне забезпечення Git, GitLab, Jenkins, Docker, Kubernetes, бази даних SQL, операційна система Linux.</p>
<b>Критерії оцінювання (окремо для кожного виду навчальної діяльності)</b>	<p>Оцінювання проводиться за 100-бальною шкалою. Бали нараховуються за наступним співвідношенням: • лабораторні роботи: 30% семестрової оцінки; максимальна кількість балів 30 • контрольні заміри (змістові модулі): 20% семестрової оцінки; максимальна кількість балів 20 • екзамен: 50% семестрової оцінки. Максимальна кількість балів 50. Підсумкова максимальна кількість балів 100.</p> <p><b>Змістовий модуль</b> – самостійна робота студента оформлена у вигляді робочої програми та звіту – робота друкованим текстом, рекомендованим обсягом до 5 сторінок (шрифт Times New Roman, 14). Звіт включає в себе детальний розгляд завдання до модуля, приведення прикладів та лістингів коду програм, огляду технологій. Код програм повинен бути обов'язково прокоментований та пояснений, необхідно також продемонструвати його роботу у разі, якщо в якості прикладу наводяться не окремі елементи технології, а суцільна програма. Фінальна версія звіту у .pdf форматі разом з кодом завантажується на віддалений Git-репозиторій.</p> <p><b>Академічна доброчесність:</b> Очікується, що роботи студентів будуть оригінальними дослідженнями чи міркуваннями. Списування та втручання</p>

в роботу інших студентів становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в написанні завдань є підставою для її незарахування викладачем, незалежно від масштабів плагіату чи обману. Жодні форми порушення академічної доброчесності не толеруються.

**Відвідання занять** є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції та лабораторні заняття курсу. Студенти повинні інформувати викладача про неможливість відвідати заняття. У будь-якому випадку студенти зобов'язані дотримуватися термінів визначених для виконання всіх видів робіт, передбачених курсом.

**Література.** Уся література, яку студенти не зможуть знайти самостійно, буде надана викладачем виключно в освітніх цілях без права її передачі третім особам. Студенти заохочуються до використання також й іншої літератури та джерел, яких немає серед рекомендованих.

**Політика виставлення балів.** Враховуються бали, набрані при поточному контролі та бали за виконання лабораторних робіт. При цьому обов'язково враховуються присутність на заняттях та активність студента під час практичного заняття; недопустимість пропусків та запізнь на заняття; користування мобільним телефоном, планшетом чи іншими мобільними пристроями під час заняття в цілях не пов'язаних з навчанням; списування та плагіат; несвоєчасне виконання поставленого завдання і т. ін.

**Оцінювання лабораторних робіт** (15 лабораторних робіт, максимальна кількість балів: 30) відбувається шляхом оцінки роботи студента під час проведення лабораторної роботи в аудиторії та захисту звіту по виконаній лабораторній роботі (0-2 балів за одну роботу).

Бали оцінювання лабораторних робіт нараховуються за наступним співвідношенням:

2 – студент в повному обсязі володіє навчальним матеріалом, має повне розуміння розглянутої теми, надає правильні відповіді на запитання по темі, код програми функціонує відповідно до завдання;

1 - студент достатньо розуміє розглянутий матеріал та принципи написаного ним коду програми, присутні неточності та незначні помилки у відповідях на запитання по темі, код програми функціонує відповідно до завдання, проте з помірними недоліками;

0 - студент зовсім не засвоїв розглянутий матеріал, написаний ним код програми не відповідає темі/не функціонує взагалі.

**Оцінювання змістових модулів** (2 змістових модулів, 10 балів за кожний) — за результатами написаних студентом звіту та коду програми.

Бали оцінювання змістових модулів нараховуються за наступним співвідношенням:

10-8 - розглянута тема відтворюється в повному обсязі, правильно, обґрунтовано, логічно, містить аналіз і систематизацію, аргументовані висновки. Засвідчено глибоке володіння матеріалом. Наведений код повністю робочий та відповідає темі. Можуть бути присутні несуттєві помилки та невідповідності;

8-6 - відтворюється значна частина розглянутої теми. Виявлено знання і розуміння основних положень навчальної дисципліни, проте присутні

	<p>неточності та/або невідповідності основній темі. Наведений код частково робочий, проте в загальному відповідають темі;</p> <p>6-4 - відстежується загальне розуміння розглянутої теми. Виявлені множинні неточності та невідповідності, пояснення наведеного коду відсутні, код функціонує із значними неточностями (або відсутні приклади запуску коду на виконання взагалі);</p> <p>4-2 – студент погано розуміє розглянуту тему. Виявлені суттєві неточності та невідповідності. Наведені приклади коду з суттєвими недоліками, або не відповідають темі;</p> <p>2 – 0 – студент взагалі не розуміє розглянуту тему. Тему не розкрито, кількість викладеного матеріалу не відповідає загальним нормам обраного виду роботи. Наведений код не робочий, або відсутній як такий.</p> <p><b>Критерії оцінювання результатів неформальної освіти:</b>  Нарахування балів відбувається за написання студентом тез доповідей на конференціях, наукових статей, участь у діяльності наукових гуртків, участь у наукових семінарах та круглих столах, конкурсах, участь у заходах неформальної освіти за отримання сертифікатів про проходження навчання на різних освітніх платформах (Coursera, Prometheus тощо), курсах на провідних ІТ компаніях за тематикою навчальної дисципліни. Кількість балів визначається відсотком покриття результатів відповідної активності до вимог результатів навчання з навчальної дисципліни.</p>
<p><b>Питання до заліку чи екзамену.</b></p>	<p>Типовий перелік питань та завдань для проведення поточних оцінювань знань. Веб-сторінка Moodle: <a href="http://e-learning.lnu.edu.ua/course/view.php?id=3136">http://e-learning.lnu.edu.ua/course/view.php?id=3136</a></p>
<p><b>Опитування</b></p>	<p>Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.</p>

**Схема курсу “DevOps”  
для студентів спеціальності 121 – Високопродуктивний комп’ютинг**

**Схема курсу (лекції)**

<b>Тиж.</b>	<b>Тема, план, короткі тези</b>	<b>Форма діяльності (заняття)* *лекція, самостійна, дискусія, групова робота)</b>	<b>Література.** * Ресурси в інтернеті</b>	<b>Завдання, год</b>	<b>Термін виконання</b>
1	<b>Вступ до безперервного розгортання обчислювальних систем.</b> Сучасні практики повного циклу програмних додатків. Таблиця інструментів безперервної інтеграції/ безперервного розгортання.	Лекція	Сайт курсу	2	кінець поточного тижня
2	<b>Вступ до системи контролю версій Git.</b> Коротка історія Git. Основи Git. Командний рядок. Установка Git. Початкові налаштування Git. Отримання допомоги. Основи Git. Створення Git-сховища. Запис змін в репозиторій. Перегляд історії комітів. Операції скасування. Робота з віддаленими репозиторіями. Робота з мітками. Псевдоніми в Git.	Лекція	Сайт курсу	2	кінець поточного тижня
3	<b>Розгалуження в Git.</b> Про розгалуження в двох словах. Основи розгалуження і злиття. Управління гілками. Робота з гілками. Віддалені гілки. Перебазування. Git на сервері. Протоколи. Установка Git на сервер. Генерація відкритого SSH ключа. Налаштуємо сервер. Git-демон. Розумний HTTP. GitWeb. GitLab. Git-хостинг.	Лекція	Сайт курсу	2	кінець поточного тижня
4	<b>Розподілений Git.</b> Розподілений робочий процес. Участь в проекті. Супровід проекту. GitHub. Налаштування і конфігурація облікового запису. Внесення власного вкладу в проекти. Супровід проекту. Управління організацією. Scripting GitHub.	Лекція	Сайт курсу	2	кінець поточного тижня
5	<b>Інструменти Git.</b> Вибір ревізії. Інтерактивне індексування. Збереження і очищення. Підпис результатів вашої роботи. Пошук. виправлення історії. Розкриття таємниць reset. Просунуте злиття. Regere. Виявлення помилок за допомогою Git. Підмодулі. Створення пакетів. Заміна. Сховище облікових даних.	Лекція	Сайт курсу	2	кінець поточного тижня
6	<b>Налаштування Git.</b> Конфігурація Git. Атрибути Git. Хукі в Git. Приклад примусової політики Git. Git і інші	Лекція	Сайт курсу	2	кінець поточного тижня



	системи контролю версій. Git як клієнт. Міграція на Git. Git зсередини. Об'єкти Git. Посилання в Git. Pack-файли. Специфікації посилань. Протоколи передачі даних. Обслуговування сховища та відновлення даних. Змінні оточення.				
7	<b>Вступ до контейнеризації та Docker.</b> Порівняння віртуальних машин і контейнерів. Встановлення Docker. Hello, World в Docker. Архітектура Docker. Зображення та контейнери. Реєстри. Повна картина. Маніпулювання контейнерами. Запуск контейнерів. Перерахування контейнерів. Перезапуск контейнерів. Очищення звисаючих контейнерів. Запуск контейнерів в інтерактивному режимі. Створення контейнерів за допомогою виконуваних зображень. Запуск контейнерів у відокремленому режимі.	Лекція	Сайт курсу	2	кінець поточного тижня
8	<b>Виконання команд всередині запущеного контейнера.</b> Запуск оболонки всередині запущеного контейнера. Доступ до журналів із запущеного контейнера. Зупинка або ліквідація запущеного контейнера. Картування портів. Демонстрація ізоляції контейнера. Створення власних зображень. Основи створення зображень. Створення виконуваного зображення.	Лекція	Сайт курсу	2	кінець поточного тижня
9	<b>Вміщення експрес-програми.</b> Робота з томами. Багатоетапні збірки. Завантаження вбудованих зображень у Docker Hub. Робота з багатоконтейнерними програмами за допомогою Docker Compose. Основи складання. Послуги лістингу. Виконання команд всередині запущеної служби. Запуск оболонки всередині запущеної служби. Доступ до журналів із запущеної служби. Припинення запуску служб. Складання програми з повним стеком.	Лекція	Сайт курсу	2	кінець поточного тижня
10	<b>Автоматизація CI/CD.</b> Встановлення Jenkins на Linux Ubuntu. Адм і н істр ування Jenkins . Керування плагінами Jenkins. Найпрост і ш і Jobs Jenkins включаючи Deployment. До давання Slave/Node в Jenkins. Віддалене и локальне керування через Jenkins CLI Client. Deployment з GitHub. Автоматизація запуск у Build Job. Jenkins Build Triggers.	Лекція	Сайт курсу	2	кінець поточного тижня
11	<b>Автоматизація запуску Build з GitHub.</b> Jenkins trigger from GitHub, Jenkins webhook. Build з параметрами. Deploy в AWS Elastic Beanstalk. Запуск Groovy Script. Обнулення лічильника	Лекція	Сайт курсу	2	кінець поточного тижня

	Jenkins Build. Основи Jenkins Pipeline і Jenkinsfile.				
12	<b>Призначення та задачі Kubernetes.</b> Призначення інструмента для організації контейнерів. Функції інструментів для оркестрації контейнерів. Базова архітектура: Вузли ведучого-підлеглого, процеси Kubernetes. Основні поняття: поди , контейнери, сервіси. Їхнє призначення. Приклад файлу конфігурації.	Лекція	Сайт курсу	2	кінець поточного тижня
13	<b>Вузол і Pod. Служба та вхід. ConfigMap і Secret.</b> Томи. Розгортання та StatefulSet. Основні узагальнені компоненти Kubernetes Робочі вузли. 3 процеси вузлів: виконання контейнера, Kubelet, Kube Proxy. майстервузли. Майстер-процеси. Сервер API. Планувальник. Менеджер контролера. etcd. Мозок кластера. Приклад налаштування кластера.	Лекція	Сайт курсу	2	кінець поточного тижня
14	<b>Доступність та масштабованість; аварійне відновлення.</b> Порівняння Kubernetes і AWS. Мінікубе. Kubectl – Встановлення minikube та kubectl. Створення та запуск кластера minikube. Запуск кластера у режимі налагодження Статус різних компонентів. Створення під розгортання. Шари абстракції CRUD, застосування файлу конфігурації. Особливості команд	Лекція	Сайт курсу	2	кінець поточного тижня
15	<b>Файл конфігурації Kubernetes.</b> Три частини файлу конфігурації Kubernetes. метадані та специфікація. статус. формат конфігураційного файлу. план для стручків (шаблон). підключення служб до розгортань та стручків (мітка, селектор та порт). підключення розгортання до подів . Підключення послуг до розгортання портів. Оглядова діаграма створених компонентів Kubernetes. Потік запитів браузера. Розгортання MongoDB. Секрет. Внутрішня служба для MongoDB. Розгортання MongoExpress. ConfigMap. Зовнішня служба MongoExpress.	Лекція	Сайт курсу	2	кінець поточного тижня
16	<b>Простір імен.</b> Простори імен за замовчуванням. Призначення просторів імен. Випадки використання. Організація компонентів. Уникайте конфліктів з іншими командами. Спільний доступ до ресурсів. Обмеження доступу та ресурсів. Характеристики просторів імен. Створення компонентів у просторах імен. Зміна активного простору імен Ingress. Порівняння Зовнішньої служб и з Ingress. Приклади файлів конфігурації YAML для зовнішньої служби та Ingress. Внутрішня конфігурація служби для Ingress. Налаштування	Лекція	Сайт курсу	2	кінець поточного тижня

	Ingress у кластері. Демонстрація: Налаштування Ingress в Minikube. Встановлення контролера Ingress в Minikube. Створення правила введення. Вхідна інформація за замовчуванням. Введення випадків маршрутизації. Кілька шляхів для одного хоста. Кілька шляхів субдомени або домени. Налаштування сертифіката TLS				
--	--	--	--	--	--

### Схема курсу (лабораторні роботи)

Тиж.	Тема, план, короткі тези	Форма діяльності (заняття)* *лекція, самостійна, дискусія, групова робота)	Література.** * Ресурси в інтернеті	Завдання, год	Термін виконання
1	Вступне заняття та інструктаж з техніки безпеки. Ознайомлення з організацією навчання в системі Moodle	Лабораторна	Сайт курсу	2	кінець поточного тижня
2	Встановлення та налаштування системи контролю версій Git. Створення Git-сховища.	Лабораторна	Сайт курсу	2	кінець поточного тижня
3	Створення простого Web ресурсу з бекендом для подальшого автоматичного розгортання	Лабораторна	Сайт курсу	2	кінець поточного тижня
4	Встановлення та налаштування Docker. Виконання Hello, World в Docker.	Лабораторна	Сайт курсу	2	кінець поточного тижня
5	Вивчення команд Docker CLI	Лабораторна	Сайт курсу	2	кінець поточного тижня
6	Створення image образу в Docker	Лабораторна	Сайт курсу	2	кінець поточного тижня
7	Написання Dockerfile. Команди роботи з контейнерами і образами	Лабораторна	Сайт курсу	2	кінець поточного тижня
8	З'єднання Docker образів між собою	Лабораторна	Сайт курсу	2	кінець поточного тижня
9	Розгортання образу Docker з GitHub (GitLab)	Лабораторна	Сайт курсу	2	кінець поточного тижня
10	Встановлення та адміністрування Jenkins. Керування плагінами Jenkins. Керування найпростішими Jenkins Jobs	Лабораторна	Сайт курсу	2	кінець поточного тижня
11	Встановлення та робота з Minikube та Kubectl	Лабораторна	Сайт курсу	2	кінець поточного тижня
12	Створення файлу конфігурацій Kubernetes. Автоматичне розгортання проекту.	Лабораторна	Сайт курсу	2	кінець поточного тижня
13	Написання власного yaml файлу	Лабораторна	Сайт курсу	2	кінець

	Kubernetes. Автоматичне розгортання				поточного тижня
14	Автоматичне розгортання бази даних за допомогою Kubernetes	Лабораторна	Сайт курсу	2	кінець поточного тижня
15	Створення Kubernetes кластера. Робота з простором імен в Kubernetes	Лабораторна	Сайт курсу	2	кінець поточного тижня