

Міністерство освіти і науки України  
Центральноукраїнський національний технічний університет  
Механіко-технологічний факультет  
Кафедра кібербезпеки та програмного забезпечення

## **DevOps та Cloud-обчислення**

*Методичні рекомендації до виконання лабораторних робіт для здобувачів  
денної форми навчання галузі 12 «Інформаційні технології»*

**ЗАТВЕРДЖЕНО**

на засіданні кафедри кібербезпеки та  
програмного забезпечення, протокол  
№ 2 від 30.08.2022

**Кропивницький**  
2022

**DevOps та Cloud-обчислення:** метод. рекомендації до викон. лаб. робіт для здобув. денної форми навчання гал. 12 «Інформаційні технології» // [уклад. О.В. Коваленко, А.С. Коваленко, Л.І. Поліщук]; М-во освіти і науки України, Центральноукр. нац. техн. ун-т; – Кропивницький: ЦНТУ, 2022. – 29 с.

**Укладачі:** Коваленко О.В., докт. техн. наук, доц;  
Коваленко А.С. канд. техн. наук, доц;  
Поліщук Л.І., ст. викладач.

**Рецензенти:** Смірнов О. А., докт. техн. наук, професор, завідувач кафедри;  
Якименко Н.М., к. ф.-м. наук, доцент.

© Центральноукраїнський  
національний технічний  
університет, 2022

## **ЗМІСТ**

<b>ВСТУП.....</b>	<b>4</b>
Лабораторна робота №1 .....	10
Лабораторна робота №2.....	12
Лабораторна робота №3.....	14
Лабораторна робота №4.....	16
Лабораторна робота №5.....	18
Лабораторна робота №6.....	20
Лабораторна робота №7.....	22
<b>Система оцінювання та вимоги.....</b>	<b>24</b>
<b>Список використаної літератури.....</b>	<b>24</b>

## ВСТУП

Навчальний курс **«DevOps та Cloud-обчислення»** призначений для набуття теоретичних знань та комплексу професійних навичок, які відповідають за успішне використання та впровадження низки сучасних практик, призначених для поглиблення взаємодії розробників із фахівцями інформаційно-технологічного обслуговування та зближення їхніх робочих процесів одне з одним. Що дає можливість забезпечити тісну взаємозалежність між розробкою та використанням програмного забезпечення і має на меті допомогти фірмам розробникам програмного забезпечення швидше створювати та оновлювати програмні продукти та послуги.

Метою викладання навчальної дисципліни **«DevOps та Cloud-обчислення»** є забезпечення здобувачів вищої освіти комплексом знань, умінь та навичок, необхідних для застосування в професійній діяльності низки сучасних практик у сервісних та продуктових ІТ компаніях.

Враховуючи послідовність накопичення знань і набуття вмінь, для опанування навчальної дисципліни необхідні знання й вміння, здобуті під час вивчення навчальних дисциплін «Проектування та моделювання складних систем», «Проектування комп'ютерних систем та мереж», «Big Data».

Лекційні заняття проводяться в аудиторіях обладнаних мультимедійним проектором. Лабораторні роботи виконуються у аудиторіях кафедри кібербезпеки та програмного забезпечення, обладнаних відповідним апаратним та програмним забезпеченням (ауд 501, 507, 508, 517), з відкритою бездротовою мережею Wi-Fi, вільним доступом до Інтернету. Оскільки при вивченні дисципліни використовуються інформаційні технології навчання, система дистанційної освіти Moodle, студенту необхідно мати комп'ютерну техніку (з виходом у Internet) та оргтехніку для комунікації з викладачами, виконання тестових завдань в системі дистанційної освіти.

Лабораторне заняття – форма навчального заняття, спрямована на закріплення та вдосконалення студентом теоретичних знань, отриманих як на лекційних і практичних заняттях, так і в процесі самостійного вивчення матеріалу.

Провідна форма навчання – лекція. Лекція дозволяє дуже економно, з мінімальними затратами часу і викладача, і студентів, надати великий обсяг інформації по темі, що розглядається. За характером логіки пізнання впроваджуються аналітичний, індуктивний та дедуктивний методи.

Супровідні методи – лабораторні роботи.

Основна дидактична мета практичного заняття – закріплення й деталізація знань, а головне – формування навичок і вмінь. Для проведення практичного заняття викладач готує відповідні методичні матеріали: тести для виявлення рівня оволодіння необхідними теоретичними положеннями ; набір практичних завдань різної складності для розв'язування їх на занятті та дидактичні засоби.

Під час лабораторного заняття студенти під керівництвом викладача набувають практичних навичок у роботі з обчислювальною технікою, оволодівають методикою створення програмних продуктів у програмному середовищі. При цьому у студентів формуються вміння й практичні навички використання різних програмних засобів ПК для розв'язання конкретних економічних задач відповідно до індивідуального завдання.

Проведення лабораторних занять ґрунтується на попередньо підготовлених методичних матеріалах: визначення підготовленості студентів до виконання завдань лабораторного заняття на основі тестового контролю знань основних положень теорії досліджуваної теми, усного контролю виконання домашнього завдання, пов'язаного з розробкою макетів документів, які необхідно розробити програмно під час заняття.

Індивідуальні завдання до кожної лабораторної роботи мають чітко виражену прикладну спрямованість, що враховує профіль підготовки

студентів, тобто охоплюють питання автоматизації рішення різних завдань економіки і підприємництва.

На лабораторному занятті студенти під керівництвом викладача проводять розробку ПЗ в навчальних лабораторіях з використанням комп'ютерної техніки. Основною метою лабораторного заняття є практичне підтвердження окремих теоретичних положень та набуття практичних вмінь з виконання обчислювальних експериментів.

Головна особливість цих занять полягає у тому, що вони об'єднують теорію з практикою, забезпечують їх єдність. Сукупність лабораторних занять з дисципліни є лабораторним практикумом, що сплановане за єдиним задумом. Лабораторні заняття плануються після проведення лекцій. А при необхідності розробки програм, проектування баз даних або підготовки складних розрахунків і початкових даних перед лабораторними заняттями проводяться консультації.

Лабораторні роботи виконуються у такій послідовності:

- вивчення навчального матеріалу з теми лабораторної роботи з використанням конспекту лекцій, рекомендованих підручників і навчальних посібників;
- самостійна підготовка студентами макетів інтерфейсів програм, які мають бути практично створені на занятті;
- виконання завдання на ПК відповідно до виданого варіанта й подання результатів викладачеві.

По завершенню кожної роботи студенти готують і оформлюють звіт й захищають отримані результати.

Звіт повинен містити:

- тему й мету роботи;
- зміст завдання й короткий опис порядку його виконання;
- аналіз отриманих результатів та висновки, роздруківку основних результатів виконання індивідуального завдання.

Напередодні проведення кожного лабораторного заняття (після відповідної лекції) студентам видається завдання, що містить: тему і мету заняття; скорочені теоретичні відомості щодо змісту лабораторного заняття; список питань для підготовки (це можуть бути контрольні питання по темі, що вивчається, заповнення розданих матеріалів індивідуальними даними, розробка програм, таблиць і т.д.); послідовність підлягаючих виконанню на занятті дій (завдання на лабораторну роботу); вимоги до змісту звіту. Студент повинен вивчити навчальний матеріал, завдання, підготувати необхідні для роботи на занятті матеріали і знати відповіді на контрольні питання.

У ході підготовки може бути створена заготовка звіту, що дозволить заощадити час на занятті. Лабораторні заняття проводяться в аудиторіях, академічна група ділиться на підгрупи.

Усі лабораторні заняття з дисципліни проводяться фронтально, кожен студент працює за окремим комп'ютером. На початку заняття, після оголошення теми, цільової установки і коротких указівок щодо особливостей роботи викладачем проводиться контроль підготовленості студентів, звичайно, шляхом перевірки відповідей на контрольні питання (тестів), рідше, у формі усної бесіди по темі заняття.

Для контролю може використовуватися і тестування. Обов'язково перевіряється наявність матеріалів для виконання роботи (програм, роздаткового матеріалу з відпрацьованими індивідуальними питаннями, початкових даних для вирішення задач, заготовок звіту і т. п.).

За відсутності матеріалів, необхідних для виконання роботи, і знань, які не дозволяють виконати роботу, студент до роботи не допускається, і йому пропонується виконати необхідну підготовку. Сама робота повинна виконуватися у додатковий час. У ході заняття студенти самостійно виконують передбачені завданням дії, заносючи результати в звіт. На це відводиться до 85–90% часу заняття.

Викладач здійснює контроль за роботою і надає допомогу при виникненні ускладнень, звертає увагу на складні ключові моменти. Причому основну увагу приділяється не вказівці на конкретну помилку, а методиці пошуку причин виникнення цих помилок.

Складання звіту – це відповідальний етап лабораторного заняття. При його складанні студенти розвивають навички аналізу, узагальнення і творчого осмислення результатів роботи, а також навички розробки документації до програмного продукту. Необхідно прагнути до того, щоб студенти оформляли звіт про виконану роботу і представили його викладачу до кінця лабораторної роботи. Цьому сприяє наявність наперед підготовленої заготовки, в яку послідовно заносяться всі необхідні дані і зроблені висновки.

Звіт повинен бути представлений у вигляді електронного документа. За наслідками контролю готовності студентів до роботи, об'єму і правильності її виконання, повноти і якості оформлення звіту і його захисту, терміну захисту викладач виставляє оцінку. Звіти, які не представлені під час заняття, захищаються в додатковий час. В окремих випадках оцінка може виставлятися за групу взаємопов'язаних робіт.

При оцінці лабораторної роботи викладач ураховує правильність та розуміння роботи розроблених програмних продуктів, уміння працювати у програмному середовищі. Оцінки за кожну лабораторну роботу вносяться у відповідний журнал. Студент, що пропустив лабораторне заняття або не допущений до нього, зобов'язаний виконати відповідну роботу під час самостійної підготовки і відзвітувати. Повторна здача робіт, які не були прийняті, проводиться під час консультацій або під час наступних лабораторних занять. Оцінки, отримані студентом за окремі лабораторні заняття враховуються при виставленні поточної модульної оцінки з навчальної дисципліни. У процесі лабораторного заняття викладач організує такі види методичної роботи зі студентами: вирішення поточних запропонованих індивідуальних завдань на лабораторну роботу; перевірку



завдань щодо розробки програм та алгоритмів; захист лабораторних робіт окремих студентів. Перелік тем лабораторних занять наведено у табл. 1.

Таблиця 1 – Перелік тем лабораторних занять

№ з/п	Назва теми	Кількість годин (денна форма навчання)
1	<b>ЛР 1.</b> Основи інсталяції та налаштування Teamcity.	2
2	<b>ЛР 2.</b> Основи інсталяції та налаштування Jenkins.	2
3	<b>ЛР 3.</b> Основи інсталяції та налаштування Capistrano.	2
4	<b>ЛР 4.</b> Основи зміни налаштувань сервера Puppet.	2
5	<b>ЛР 5.</b> Основи зміни налаштувань сервера Ansible.	2
6	<b>ЛР 6_1.</b> Основи зміни налаштувань сервера Chef. Теоретична частина.	2
7	<b>ЛР 6_2.</b> Основи зміни налаштувань сервера Chef. Практична частина.	2
8	<b>ЛР 7_1.</b> Основи використання та порівняння сервісів Microsoft Azure та Amazon Web Services. Теоретична частина.	2
9	<b>ЛР 7_2.</b> Основи використання та порівняння сервісів Microsoft Azure та Amazon Web Services. Практична частина.	2
<b>Усього годин</b>		<b>18</b>

## ЛАБОРАТОРНА РОБОТА №1

### ТЕМА: ОСНОВИ ІНСТАЛЯЦІЇ ТА НАЛАШТУВАННЯ TEAMCITY

**МЕТА:** Отримати практичні навички роботи з програмним продуктом Teamcity

**ЗНАТИ:** Основи використання консолі обраної ОС

### ТЕОРЕТИЧНІ ВІДОМОСТІ

У зв'язку з великим обсягом інформації використовувати електронну документацію (погоджувати з лектором).

Основна документація:

- Офіційна сторінка **Teamcity**: <https://www.jetbrains.com/teamcity/>
- **TeamCity** налаштовуємо CI/CD: <https://goo.su/tKS>
- **TeamCity** Plugins: <https://plugins.jetbrains.com/teamcity>

Додаткова документація:

- DevOps Roadmap: <https://roadmap.sh/devops>.
- Основи BASH 1: <https://habr.com/en/post/47163/>
- Основи BASH 2: <https://habr.com/en/post/52871/>
- Основи Systemd: <https://goo.su/oWM>
- Робота з сервісами: <https://goo.su/naD>
- Створення Init скриптів: <https://linux-notes.org/pishem-init-skript/>
- Створення власних образів: <https://goo.su/RQW>
- Використання Docker Compose: <https://goo.su/rLi>

## **ЗАВДАННЯ**

Ознайомитися з програмним продуктом **TEAMCITY**. У реферативному стилі оформити звіт лабораторної роботи.

Звіт повинен включати опис та скріншоти наступних пунктів:

- Пошук дистрибутиву;
- Ліцензійні вимоги використання;
- Попередні налаштування дистрибутиву;
- Інсталяція програмного продукту. ОС обирається за бажанням студента (погоджувати з лектором);
- Загальний опис та можливості програмного продукту;
- Переваги та недоліки програмного продукту;
- Порівняння з існуючими аналогами.

Лабораторну роботу оформляти по загальноприйнятій формі (ДСТУ) та аналогічно до інших предметів кафедри КБПЗ.

## **КОНТРОЛЬНІ ЗАПИТАННЯ**

1. Навіщо потрібна спеціалізація DevOps?
2. Для чого потрібен CI і CD-сервер?
3. Які ключові поняття CI / CD ?
4. Які ключові особливості для забезпечення неперервної інтеграції, тестування програмного забезпечення та організації ефективної колективної роботи над вихідним кодом?
5. В яких випадках використовується програмний продукт TeamCity?
6. Першочергово для роботи над якими проектами розроблявся TeamCity?
7. Які основні можливості TeamCity?

## ЛАБОРАТОРНА РОБОТА №2

### ТЕМА: ОСНОВИ ІНСТАЛЯЦІЇ ТА НАЛАШТУВАННЯ JENKINS

**МЕТА:** Отримати практичні навички роботи з програмним продуктом Jenkins

**ЗНАТИ:** Основи використання консолі обраної ОС

### ТЕОРЕТИЧНІ ВІДОМОСТІ

У зв'язку з великим обсягом інформації використовувати електронну документацію (погоджувати з лектором).

Основна документація:

- Офіційна сторінка **Jenkins**: <https://www.jenkins.io/>
- Мікросервіси **Jenkins**: <https://habr.com/en/post/493580/>
- Як налаштувати сервер збирання Jenkins - Amazon AWS: <https://goo.su/sDN>

Додаткова документація:

- DevOps Roadmap: <https://roadmap.sh/devops>.
- Основи BASH 1: <https://habr.com/en/post/47163/>
- Основи BASH 2: <https://habr.com/en/post/52871/>
- Основи Systemd: <https://goo.su/oWM>
- Робота з сервісами: <https://goo.su/naD>
- Створення Init скриптів: <https://linux-notes.org/pishem-init-skript/>
- Створення власних образів: <https://goo.su/RQW>
- Використання Docker Compose: <https://goo.su/rLi>

## ЗАВДАННЯ

Ознайомитися з програмним продуктом **Jenkins**. У реферативному стилі оформити звіт лабораторної роботи.

Звіт повинен включати опис та скріншоти наступних пунктів:

- Пошук дистрибутиву;
- Ліцензійні вимоги використання;
- Попередні налаштування дистрибутиву;
- Інсталяція програмного продукту. ОС обирається за бажанням студента (погоджувати з лектором);
- Загальний опис та можливості програмного продукту;
- Переваги та недоліки програмного продукту;
- Порівняння з існуючими аналогами.

Лабораторну роботу оформляти по загальноприйнятій формі (ДСТУ) та аналогічно до інших предметів кафедри КБПЗ.

## КОНТРОЛЬНІ ЗАПИТАННЯ

1. Для чого потрібні інструменти для безперервної інтеграції?
2. В яких випадках використовується безперервна доставка (Continuous delivery)?
3. Які ключові особливості використання безперервної доставки (Continuous delivery)?
4. В яких випадках використовується програмний продукт Jenkins?
5. Які основні можливості Jenkins?

## ЛАБОРАТОРНА РОБОТА №3

**ТЕМА: ОСНОВИ ІНСТАЛЯЦІЇ ТА НАЛАШТУВАННЯ CAPISTRANO.**

**МЕТА: Отримати практичні навички роботи з програмним продуктом Capistrano**

**ЗНАТИ: Основи використання консолі обраної ОС**

### ТЕОРЕТИЧНІ ВІДОМОСТІ

У зв'язку з великим обсягом інформації використовувати електронну документацію (погоджувати з лектором).

Основна документація:

- Офіційна сторінка **Capistrano**: <https://capistranorb.com/>
- Що таке Capistrano, для чого його використовувати та які у нього переваги: <https://blog.vistro.ru/capistrano/>
- Capistrano: віддалене розгортання додатків: <https://habr.com/en/post/49127/>
- Capistrano та php: <https://habr.com/en/post/110021/>
- Автоматизація розгортання за допомогою **Capistrano**: <https://goo.su/dKV>

Додаткова документація:

- DevOps Roadmap: <https://roadmap.sh/devops>.
- Основи BASH 1: <https://habr.com/en/post/47163/>
- Основи BASH 2: <https://habr.com/en/post/52871/>
- Основи Systemd: <https://goo.su/oWM>
- Робота з сервісами: <https://goo.su/naD>
- Створення Init скриптів: <https://linux-notes.org/pishem-init-skript/>

- Створення власних образів: <https://goo.su/RQW>
- Використання Docker Compose: <https://goo.su/rLi>

## ЗАВДАННЯ

Ознайомитися з програмним продуктом **Capistrano**. У реферативному стилі оформити звіт лабораторної роботи.

Звіт повинен включати опис та скріншоти наступних пунктів:

- Пошук дистрибутиву;
- Ліцензійні вимоги використання;
- Попередні налаштування дистрибутиву;
- Інсталяція програмного продукту. ОС обирається за бажанням студента (погоджувати з лектором);
- Загальний опис та можливості програмного продукту;
- Переваги та недоліки програмного продукту;
- Порівняння з існуючими аналогами.

Лабораторну роботу оформляти по загальноприйнятій формі (ДСТУ) та аналогічно до інших предметів кафедри КБПЗ.

## КОНТРОЛЬНІ ЗАПИТАННЯ

1. В яких випадках використовується програмний продукт Capistrano?
2. Які основні можливості Capistrano?
3. Які переваги використання Capistrano?
4. Який типовий порядок публікації вихідного коду за допомогою Capistrano?
5. На якій мові програмування створено Capistrano?

## ЛАБОРАТОРНА РОБОТА №4

### ТЕМА: ОСНОВИ ЗМІНИ НАЛАШТУВАНЬ СЕРВЕРА PUPPET

**МЕТА:** Отримати практичні навички роботи з програмним продуктом Puppet

**ЗНАТИ:** Основи використання консолі обраної ОС

### ТЕОРЕТИЧНІ ВІДОМОСТІ

У зв'язку з великим обсягом інформації використовувати електронну документацію (погоджувати з лектором).

Основна документація:

- Офіційна сторінка **Puppet**: <https://puppet.com/open-source/#osp>
- Вступ до Puppet: <https://habr.com/en/company/avito/blog/507346/>
- Puppet система керування конфігураціями: <https://habr.com/en/post/67471/>

Додаткова документація:

- DevOps Roadmap: <https://roadmap.sh/devops>.
- Основи BASH 1: <https://habr.com/en/post/47163/>
- Основи BASH 2: <https://habr.com/en/post/52871/>
- Основи Systemd: <https://goo.su/oWM>
- Робота з сервісами: <https://goo.su/naD>
- Створення Init скриптів: <https://linux-notes.org/pishem-init-skript/>
- Створення власних образів: <https://goo.su/RQW>
- Використання Docker Compose: <https://goo.su/rLi>



## ЗАВДАННЯ

Ознайомитися з програмним продуктом **Puppet**. У реферативному стилі оформити звіт лабораторної роботи.

Звіт повинен включати опис та скріншоти наступних пунктів:

- Пошук дистрибутиву;
- Ліцензійні вимоги використання;
- Попередні налаштування дистрибутиву;
- Інсталяція програмного продукту. ОС обирається за бажанням студента (погоджувати з лектором);
- Загальний опис та можливості програмного продукту;
- Переваги та недоліки програмного продукту;
- Порівняння з існуючими аналогами.

Лабораторну роботу оформляти по загальноприйнятій формі (ДСТУ) та аналогічно до інших предметів кафедри КБПЗ.

## КОНТРОЛЬНІ ЗАПИТАННЯ

1. В яких випадках використовується програмний продукт Puppet?
2. Які основні можливості Puppet?
3. Які переваги використання Puppet?
4. На якій мові програмування створено Puppet?
5. У яких випадках необхідно централізовано керувати конфігурацією операційних систем та програмного забезпечення?

## ЛАБОРАТОРНА РОБОТА №5

**ТЕМА: ОСНОВИ ЗМІНИ НАЛАШТУВАНЬ СЕРВЕРА ANSIBLE.**

**МЕТА:** Отримати практичні навички роботи з програмним продуктом **Ansible**

**ЗНАТИ:** Основи використання консолі обраної ОС

### ТЕОРЕТИЧНІ ВІДОМОСТІ

У зв'язку з великим обсягом інформації використовувати електронну документацію (погоджувати з лектором).

Основна документація:

- Офіційна сторінка **Ansible**: <https://www.ansible.com/>
- Що таке **Ansible** та як його використовувати: <https://goo.su/bpc>
- Посібник з **Ansible**: <https://habr.com/en/post/305400/>

Додаткова документація:

- DevOps Roadmap: <https://roadmap.sh/devops>.
- Основи BASH 1: <https://habr.com/en/post/47163/>
- Основи BASH 2: <https://habr.com/en/post/52871/>
- Основи Systemd: <https://goo.su/oWM>
- Робота з сервісами: <https://goo.su/naD>
- Створення Init скриптів: <https://linux-notes.org/pishem-init-skript/>
- Створення власних образів: <https://goo.su/RQW>
- Використання Docker Compose: <https://goo.su/rli>

## ЗАВДАННЯ

Ознайомитися з програмним продуктом **Ansible**. У реферативному стилі оформити звіт лабораторної роботи.

Звіт повинен включати опис та скріншоти наступних пунктів:

- Пошук дистрибутиву;
- Ліцензійні вимоги використання;
- Попередні налаштування дистрибутиву;
- Інсталяція програмного продукту. ОС обирається за бажанням студента (погоджувати з лектором);
- Загальний опис та можливості програмного продукту;
- Переваги та недоліки програмного продукту;
- Порівняння з існуючими аналогами.

Лабораторну роботу оформляти по загальноприйнятій формі (ДСТУ) та аналогічно до інших предметів кафедри КБПЗ.

## КОНТРОЛЬНІ ЗАПИТАННЯ

1. В яких випадках використовується програмний продукт Ansible?
2. Які основні можливості Ansible?
3. Які переваги використання Ansible?
4. На якій мові програмування створено Ansible?
5. Яка архітектура використовується у Ansible?

## ЛАБОРАТОРНА РОБОТА №6

**ТЕМА: ОСНОВИ ЗМІНИ НАЛАШТУВАНЬ СЕРВЕРА CHEF**

**МЕТА:** Отримати практичні навички роботи з програмним продуктом Chef

**ЗНАТИ:** Основи використання консолі обраної ОС

### ТЕОРЕТИЧНІ ВІДОМОСТІ

У зв'язку з великим обсягом інформації використовувати електронну документацію (погоджувати з лектором).

Основна документація:

– Офіційна сторінка **Chef**: <https://www.chef.io/products>

– **Chef** для вирішення повсякденних завдань:  
<https://habr.com/en/post/263341/>

– **Chef** для новачків:  
<https://habr.com/en/company/southbridge/blog/208858/>

Додаткова документація:

– DevOps Roadmap: <https://roadmap.sh/devops>.

– Основи BASH 1: <https://habr.com/en/post/47163/>

– Основи BASH 2: <https://habr.com/en/post/52871/>

– Основи Systemd: <https://goo.su/oWM>

– Робота з сервісами: <https://goo.su/naD>

– Створення Init скриптів: <https://linux-notes.org/pishem-init-skript/>

– Створення власних образів: <https://goo.su/RQW>

– Використання Docker Compose: <https://goo.su/rLi>

## ЗАВДАННЯ

Ознайомитися з програмним продуктом **Chef**. У реферативному стилі оформити звіт лабораторної роботи.

Звіт повинен включати опис та скріншоти наступних пунктів:

- Пошук дистрибутиву;
- Ліцензійні вимоги використання;
- Попередні налаштування дистрибутиву;
- Інсталяція програмного продукту. ОС обирається за бажанням студента (погоджувати з лектором);
- Загальний опис та можливості програмного продукту;
- Переваги та недоліки програмного продукту;
- Порівняння з існуючими аналогами.

Лабораторну роботу оформляти по загальноприйнятій формі (ДСТУ) та аналогічно до інших предметів кафедри КБПЗ.

## КОНТРОЛЬНІ ЗАПИТАННЯ

1. В яких випадках використовується програмний продукт Chef?
2. Які основні можливості Chef?
3. Які переваги використання Chef?
4. На якій мові програмування створено Chef?
5. Яка архітектура використовується у Chef?

## ЛАБОРАТОРНА РОБОТА №7

### ТЕМА: ОСНОВИ ВИКОРИСТАННЯ ТА ПОРІВНЯННЯ СЕРВІСІВ MICROSOFT AZURE ТА AMAZON WEB SERVICES

**МЕТА:** Отримати практичні навички роботи з програмними продуктами Microsoft Azure та Amazon Web Services

**ЗНАТИ:** Основи використання консолі обраної ОС

### ТЕОРЕТИЧНІ ВІДОМОСТІ

У зв'язку з великим обсягом інформації використовувати електронну документацію (погоджувати з лектором).

Основна документація:

- Cloud Services - Amazon Web Services (AWS): <https://aws.amazon.com/ru/>
- Microsoft Azure: Служби хмарних обчислень: <https://goo.su/cmf>
- Переваги Azure в порівнянні з AWS: <https://goo.su/ry8>
- Windows на платформі AWS Amazon Web Services: <https://goo.su/dSk>
- Інструментарій AWS для Azure DevOps: <https://aws.amazon.com/ru/vsts/>

Додаткова документація:

- DevOps Roadmap: <https://roadmap.sh/devops>
- Основи BASH 1: <https://habr.com/en/post/47163/>
- Основи BASH 2: <https://habr.com/en/post/52871/>
- Основи Systemd: <https://goo.su/oWM>
- Робота з сервісами: <https://goo.su/naD>

- Створення Init скриптів: <https://linux-notes.org/pishem-init-skript/>
- Створення власних образів: <https://goo.su/RQW>
- Використання Docker Compose: <https://goo.su/rLi>

## ЗАВДАННЯ

Ознайомитися з сервісами **Amazon Web Services** та ознайомитись з платформою **Microsoft Azure**. У реферативному стилі оформити звіт лабораторної роботи.

Звіт повинен включати опис та скріншоти наступних пунктів:

- Ліцензійні вимоги використання платформ;
- Попередні налаштування платформ використання профілю;
- Загальний опис та можливості платформ;
- Переваги та недоліки платформ;

Лабораторну роботу оформляти по загальноприйнятій формі (ДСТУ) та аналогічно до інших предметів кафедри КБПЗ.

## КОНТРОЛЬНІ ЗАПИТАННЯ

1. Які основні можливості Microsoft Azure та Amazon Web Services?
2. Які переваги використання Microsoft Azure?
3. Які переваги використання Amazon Web Services?
4. В яких випадках використовується платформа Microsoft Azure?
5. В яких випадках використовуються веб сервіси Amazon?

## Система оцінювання та вимоги

Контроль знань і умінь (поточний і підсумковий) з дисципліни «**DevOps та Cloud-обчислення**» здійснюється згідно з кредитною трансферно-накопичувальною системою організації навчального процесу. Рейтинг студента із засвоєння дисципліни визначається за 100-бальною шкалою. Він складається з рейтингу навчальної роботи (засвоєння теоретичного матеріалу під час аудиторних занять та самостійної роботи, виконання лабораторних завдань).

### Шкала оцінювання: національна та ЄКТС

Сума балів за всі види навчальної діяльності	Оцінка ЄКТС	Оцінка за національною шкалою
		для екзамену
90-100	A	Відмінно
82-89	B	Добре
74-81	C	
64-73	D	Задовільно
60-63	E	
35-59	FX	незадовільно з можливістю повторного складання
1-34	F	незадовільно з обов'язковим повторним вивченням дисципліни

*Критерії оцінювання знань і вмінь здобувачів визначені Положенням про організацію освітнього процесу в ЦНТУ (стор. 32-33).*



## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

### *Базова*

1. Kovalenko O., Khudov H., Mynko P., Ikhsanov S., Diakonov O., Solomonenko Y., Drob Y., Kharun O., Cherkashyn S., Serdiuk O. «Development A Method For Determining The Coordinates Of Air Objects By Radars With The Additional Use Of Multilateration Technology» Eastern-European Journal of Enterprise Technologies Volume 5, 2021, Pages 6-16. Режим доступу: <https://www.scopus.com/record/display.uri?eid=2-s2.0-85119667497&origin=resultslist> (**Scopus**).

2. Коваленко О.В., Смірнова Т.В., Моторін Ю.Ю., Буравченко К.О., Бочуля Т.В. «Вибір оптимальної технології побудови хмарної інформаційно-комунікаційної системи автоматизації виробничих процесів». Вимірювальна та обчислювальна техніка в технологічних процесах, № 1 (2022). С. 15-26. 2022. Режим доступу: <http://vottp.khmnu.edu.ua/index.php/vottp/article/view/30/36> (Фахове видання. Категорія «Б»)

3. Коваленко А.В. Масштабирование имитационной модели технологии тестирования безопасности. Системы управління, навігації та зв'язку. – Випуск 6 (46). – Полтава: ПолтНТУ. – 2017. – С. 181-184. Режим доступу: [http://nbuv.gov.ua/UJRN/suntz\\_2017\\_6\\_42](http://nbuv.gov.ua/UJRN/suntz_2017_6_42) (Фахове видання. Категорія «Б»)

4. Коваленко О.В. GERT-мережевий синтез технології тестування на вразливість WEB-додатків. Захист інформації. – Випуск 20(2) – К.: НАУ. – 2018. – С. 89-94. Режим доступу: [http://nbuv.gov.ua/UJRN/Zi\\_2018\\_20\\_2\\_6](http://nbuv.gov.ua/UJRN/Zi_2018_20_2_6) (Фахове видання. Категорія «Б»)

5. Коваленко О.В. Імітаційна модель технології тестування безпеки на основі положень теорії масштабування. Безпека інформації. – Випуск 24 (2). – К.: НАУ. – 2018. – С. 110-117. Режим доступу: <http://jrn1.nau.edu.ua/index.php/Infosecurity/article/view/13045>

**(Фахове видання. Категорія «Б»)**

6. Коваленко О.В. Розробка інформаційної технології передтестової компіляції та розподілу доступу. Системи управління, навігації та зв'язку. – Випуск 4 (50). – Полтава: ПолтНТУ. – 2018. – С. 115-119. Режим доступу: [http://nbuv.gov.ua/UJRN/suntz\\_2018\\_4\\_24](http://nbuv.gov.ua/UJRN/suntz_2018_4_24) **(Фахове видання. Категорія «Б»)**

7. Коваленко А.В., Смирнов А.А., Комплекс математических моделей технологии тестирования WEB-приложений. Інформаційні технології: сучасний стан та перспективи: **монографія** / За загальною редакцією В.С. Пономаренка. – Х.: ТОВ «ДІСА ПЛЮС», 2018. – 461 с.

8. Engy Fouda. A Complete Guide to Docker for Operations and Development. Apress. 2022. 201 с.

9. Daniel J. Barrett. Efficient Linux at the Command Line. O'Reilly Media, Inc. 2022. 248 с.

10. Shaun R Smith, Peter Membrey. Beginning Ansible Concepts and Application. Apress Berkeley. 2022. 296 с.

11. Dan Sullivan. Google Cloud Certified Professional Cloud Architect Study Guide Second Edition. John Wiley & Sons, Inc. 2022. 371 с.

12. Michael Elder, Jake Kitchener, and Dr. Brad Topol. Hybrid Cloud Apps with OpenShift and Kubernetes. O'Reilly Media, Inc. 2021. 270 с.

### ***Допоміжна***

13. J.D. Linux The Greatest Guide for Beginners to the Linux Commands and Linux Operating System. Independently published. 2021. 108 с.

14. Gerard Maas, François Garillot. Stream Processing with Apache Spark. O'Reilly Media, Inc. 2019. 453 с.

15. John Culkin, Mike Zazon AWS. Cookbook Recipes for Success on AWS. O'Reilly Media. 2022. 568c.
16. Hiroko Nishimura. AWS for Non-Engineers. Manning Publications. 2022. 132 c.
17. Simon Bisson, Mary Branscombe, Chris Hoder & Anand Raman. Azure AI Services at Scale for Cloud, Mobile, and Edge. O'Reilly Media. 2022. 227 c.
18. Carl Albing, JP Vossen. bash Idioms. O'Reilly Media. 2022. 134 c.
19. Ashley Davis. Bootstrapping Microservices with Docker, Kubernetes, and Terraform. Manning Publications. 2021. 442 c.
20. Josh Armitage. Cloud Native Security Cookbook. O'Reilly Media. 2022. 516 c.
21. Mauricio Salatino Continuous Delivery for Kubernetes. Manning Publications. 2022. 202 c.
22. Jay Vyas, Chris Love. Core Kubernetes. Manning Publications. 2022. 336 c.
23. Bradley Smith. Devops for the Desperate a Hands-on Survival Guide. No Starch Press. 2022. 188 c.
24. Stephen Chin, Melissa McKay, Ixchel Ruiz, Baruch Sadogursky. DevOps Tools for Java Developers. O'Reilly Media. 2022. 344 c.
25. Jeroen Mulder. Enterprise DevOps for Architects. Packt Publishing. 2021. 289 c.
26. Nigel Poulton. Docker Deep Dive. Packt Publishing. 2020. 350 c.
27. William Denniss. Kubernetes for Developers. Manning Publications. 2022. 258 c.
28. John Clingan, Ken Finnigan. Kubernetes Native Microservices with Quarkus and MicroProfile. Manning Publications. 2022. 330 c.
29. James Petty, Travis Plunk, Tyler Leonhardt, Don Jones, Jeffery Hicks. Learn PowerShell in a Month of Lunches. Manning Publications. 2022. 359 c.
30. Michael Hausenblas. Learning Modern Linux. O'Reilly Media. 2022. 261 c.

31. Vedran Dakic, Jasmin Redzepagic. Linux Command Line and Shell Scripting Techniques. Packt Publishing. 2022. 552 с.
32. Jack-Benny Persson. Linux System Programming Techniques. Packt Publishing. 2021. 432 с.
33. Scott Alan Miller. Linux Administration Best Practices. Packt Publishing. 2022. 404 с.
34. James Freeman, Jesse Keating. Mastering Ansible. Packt Publishing. 2021. 540 с.
35. Russ McKendrick. Mastering Docker. Packt Publishing. 2020. 568 с.
36. Michael Herman. Microservices with Docker, Flask, and React. Independently published. 2018. 577 с.
37. Mark S. Merkow. Practical Security for Agile and DevOps. CRC Press. 2022. 236 с.
38. Richard Bullington-McGuire, Andrew K. Dennis, Michael Schwartz. Docker for Developers. Packt Publishing. 2020. 468 с.

### ***Інформаційні ресурси***

39. Курс «DevOps та Cloud-обчислення» на сервері дистанційної освіти ЦНТУ. – URL: <http://moodle.kntu.kr.ua/course/view.php?id=684>
40. Основи створення мобільних додатків [Електронний ресурс]. – Режим доступу: <https://developer.android.com/guide>
41. Онлайн-курси UDEMY. – URL: <https://www.udemy.com/> – платформа онлайн-курсів різних ІТ тематик.
42. Онлайн-курси Prometheus. – URL: <https://prometheus.org.ua/> – українська платформа безкоштовних онлайн-курсів
43. Онлайн-курси Coursera. – URL: <https://www.coursera.org> – платформа онлайн-курсів різних ІТ тематик.
44. <https://habr.com> – колективний блог з новинами та аналітичними статтями про інформаційні технології та програмування.

45. <http://stackoverflow.com/> – система питань і відповідей для професійних програмістів та новачків у програмуванні.
46. <https://dou.ua/> – український веб-сайт з елементами колективного блогу, створений для розповсюдження новин, аналітичних статей та свіжої інформації пов'язаної із інформаційними технологіями.
47. <https://www.google.com/> – основна пошукова платформа.
48. <https://www.youtube.com> – Відеохостинг, що надає користувачам послуги зберігання, доставки та показу відео. На платформі розміщено багато курсів ІТ спрямованості.
49. <https://biblprog.org.ua/ua/programming/> – каталог безкоштовних середовищ розроблення ПЗ.
50. Національна бібліотека України імені В. І. Вернадського: Електронні ресурси НБУВ [Електронний ресурс]. – Режим доступу: <http://www.nbuv.gov.ua/>.