

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

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ
до виконання лабораторних робіт з навчальної дисципліни
«DevOps та Cloud-обчислення»

*для студентів денної та заочної форми навчання
другого (магістерського) рівня вищої освіти
галузі «Інформаційні технології».*

ЗАТВЕРДЖЕНО

на засіданні кафедри кібербезпеки та
програмного забезпечення, протокол
№ 1 від 26.08.2025 року

Кропивницький
2025

УДК 004.5

DevOps та Cloud-обчислення: методичні рекомендації до виконання лабораторних робіт для студентів денної та заочної форми навчання, другого (магістерського) рівня вищої освіти, галузі «Інформаційні технології» / М-во освіти і науки України, Центральноукр. нац. техн. ун-т; [уклад. О.В. Коваленко, Р.О. Голубець, К.Д. Авраменко, Р.О. Ткачук, Л.І. Поліщук] – Кропивницький: ЦНТУ, 2025. – 48 с.

Укладачі:

Коваленко О.В., докт. техн. наук, доц;

Голубець Р.О. Lead .Net Engineer Україна м. Львів;

Авраменко К.Д. Lead Back-End Developer\Architect, Швейцарія м. Цюрих;

Ткачук Р.О. Senior DevOps Engineer EPAM Systems / асистент кафедри кібербезпеки та програмного забезпечення Центральноукраїнського національного технічного університету;

Поліщук Л.І. старший викладач кафедри кібербезпеки та програмного забезпечення Центральноукраїнського національного технічного університету.

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

© Центральноукраїнський
національний технічний
університет, 2025

ЗМІСТ

ВСТУП.....	4
Лабораторна робота №1.....	10
Лабораторна робота №2.....	14
Лабораторна робота №3.....	18
Лабораторна робота №4.....	23
Лабораторна робота №5.....	29
Лабораторна робота №6.....	32
Лабораторна робота №7.....	35
Лабораторна робота №8.....	38
Система оцінювання та вимоги.....	43
Список використаної літератури.....	44

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 поєднання 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 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.

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, 505, 507, 508, 517) з доступом до центру зберігання та обробки даних кафедри кібербезпеки та програмного забезпечення Центральноукраїнського національного технічного університету ЦЗОД КБПЗ, з локальною мережею та 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 роздатних 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, кожний 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. Перелік тем лабораторних занять наведено у табл. 1.

Таблиця 1 – Перелік тем лабораторних занять

№ з/п	Назва теми	Кількість Годин	
		денна форма навчання	заочна форма навчання (повна, магістр)
1	ЛР 1. Автоматизація типових операцій супроводу серверів ЦЗОД засобами ANSIBLE	2	0,25
2	ЛР 2. Контейнеризація та розгортання сервісу на сервері дата центру засобами DOCKER	2	0,25
3	ЛР 3. Розгортання та адміністрування вузла розподілених обчислень	2	0,25
4	ЛР 4. Моніторинг та віддалене управління програмно-технічною інфраструктурою центру зберігання та обробки даних	2	0,25

5	ЛР 5. Платформи безперервної інтеграції і доставки програмного забезпечення	2	0,25
6	ЛР 6. Інструменти інфраструктури як коду і засоби керування конфігураціями	2	0,25
7	ЛР 7. Використання та порівняння сервісів Microsoft Azure, Amazon WEB Services, Google Cloud	2	0,25
8	ЛР 8. Високошвидкісні мережеві конфігурації маршрутизаторів високої продуктивності у ЦЗОД, оцінка енергоефективності та продуктивності	2	0,25
Усього годин		16	2

При оцінці лабораторної роботи викладач ураховує правильність та розуміння роботи розроблених програмних продуктів, уміння працювати у програмному середовищі.

Оцінки за кожен 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.

ЛАБОРАТОРНА РОБОТА №1

ТЕМА: АВТОМАТИЗАЦІЯ ТИПОВИХ ОПЕРАЦІЙ СУПРОВОДУ СЕРВЕРІВ ЦЗОД ЗАСОБАМИ ANSIBLE

МЕТА: Сформувати практичні навички автоматизації типових адміністративних операцій супроводу серверної інфраструктури центру обробки даних засобами Ansible шляхом розроблення та застосування playbook і ролей для віддаленого керування конфігураціями, користувачами та мережевими політиками доступу.

ЗНАТИ: Механізми віддаленого виконання операцій через SSH, типові підходи до автоматизації керування користувачами, пакетами, конфігураціями та мережевими правилами доступу на сервері.

ТЕОРЕТИЧНІ ВІДОМОСТІ

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

Основна документація:

– Використання Ansible та робота з системою автоматизації:
https://docs.ansible.com/projects/ansible/latest/getting_started/index.html

– Робота з inventory засобами Ansible
https://docs.ansible.com/projects/ansible/latest/inventory_guide/intro_inventory.html

– Робота зі сховищем valut засобами Ansible
https://docs.ansible.com/projects/ansible/latest/vault_guide/index.html

Додаткова документація:

– <https://roadmap.sh/devops>

– <https://www.gnu.org/software/bash/manual/>

– <https://docs.docker.com/>

УМОВИ ВИКОНАННЯ Л/Р

Лабораторна робота виконується на сервері центру зберігання та обробки даних ЦНТУ, що передбачено ресурсним забезпеченням кафедри КБПЗ ЦНТУ. При переході ЦНТУ у он-лайн навчання, формат взаємодії питати у викладача що проводить л/р.

ЗАВДАННЯ

Необхідно за допомогою Ansible автоматизувати базові адміністративні дії на сервері(ах) дата центру ЦНТУ та підтвердити повторюваність і ідемпотентність виконання.

Підготовка до виконання

1. На своєму локальному комп'ютері встановити Ansible. Переконайтеся, що з вашого комп'ютера можна зайти на сервер(и) ЦЗОД по SSH. За відсутності можливості використання власного комп'ютера використовуйте тонкі клієнти з завчасно налаштованим середовищем для виконання лабораторних робіт.

Підготувати список серверів для Ansible

2. Створити файл inventory, де зазначено, якими серверами керувати (1–2 сервери) і як до них підключатися. Створити файл зі змінними, які необхідно використовувати для роботи скрипту. Завчасно запропонувати значення змінних за замовчуванням.

Зробити автоматичні налаштування на сервері (погоджувати з викладачем)

3. Підготувати роль для встановлення важливих системних утиліт: mc, htop, unzip, 7z, git, curl. Додатково окремо написати роль для встановлення docker з обов'язковою перевіркою його наявності на сервері. Якщо docker вже існує та доступний встановлення має ігноруватися. Обов'язково

вконувати усі дії від імені окремого користувача та видаляти тимчасові файли створені після завершення роботи скрипта.

Перевірка ідемпотентності та керованості

4. Запустити Ansible-налаштування двічі і показати, що другий запуск майже нічого не змінює та ігнорує встановлення docker. Змінити список обов'язкових системних утилит, додавши туди ncdu, та показати, що після оновлення скрипту необхідна утиліта буде додатково встановлена.

ОФОРМЛЕННЯ ЗВІТУ

У звіт лабораторної роботи помістити:

- Завдання л/роботи.
- Покрокові матеріали що повністю підтверджують виконання л/роботи (текст, таблиці, принтскріни, та інші матеріали для підтвердження виконання л/роботи): Показати структуру Ansible-проєкту (які файли ви створили); Додати короткі докази виконання л/р (фрагменти файлів, список серверів, налаштування) та результати запусків (перший і повторний).

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

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Що таке inventory в Ansible і навіщо потрібні групи хостів?
2. У чому різниця між playbook і роллю та коли доцільно використовувати ролі?
3. Ідемпотентність і як перевірити її на практиці в Ansible?
4. Для чого використовують handlers і коли вони спрацьовують?
5. Які способи передачі змінних існують (group_vars, host_vars, extra_vars) і чим вони відрізняються?
6. Які ризики автоматизації налаштувань SSH і як уникнути втрати доступу до сервера?

7. Чому важливо обмежувати відкриті порти на сервері ЦЗОД і як це реалізується в playbook?
8. Як організувати безпечне зберігання секретів в Ansible ?

ЛАБОРАТОРНА РОБОТА №2

ТЕМА: КОНТЕЙНЕРИЗАЦІЯ ТА РОЗГОРТАННЯ СЕРВІСУ НА СЕРВЕРІ ДАТА ЦЕНТРУ ЗАСОБАМИ DOCKER

МЕТА: Сформувати практичні навички контейнеризації та розгортання прикладного сервісу на сервері дата центру засобами Docker і Docker Compose шляхом підготовки Dockerfile, побудови образу, налаштування мережевої взаємодії та збереження даних, а також виконання операцій супроводу контейнеризованого застосунку.

ЗНАТИ: принципи побудови образів, основи роботи з мережами і портами контейнерів, відмінності томів (volumes) і bind mount

ТЕОРЕТИЧНІ ВІДОМОСТІ

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

Основна документація:

- <https://docs.docker.com/get-started/>
- <https://docs.docker.com/engine/install/>
- <https://docs.docker.com/reference/dockerfile/>
- <https://docs.docker.com/build/building/best-practices/>
- <https://docs.docker.com/reference/compose-file/>

Додаткова документація:

- <https://roadmap.sh/devops>
- <https://www.gnu.org/software/bash/manual/>
- <https://docs.docker.com/>

УМОВИ ВИКОНАННЯ Л/Р

Лабораторна робота виконується на сервері центру зберігання та обробки даних ЦНТУ, що передбачено ресурсним забезпеченням кафедри КБПЗ ЦНТУ. При переході ЦНТУ у он-лайн навчання, формат взаємодії питати у викладача що проводить л/р.

ЗАВДАННЯ

Необхідно зібрати власний Docker образ сервісу, розгорнути його на сервері ЦЗОД та організувати керування запуском через Docker Compose.

Підготовка до виконання

3. Забезпечити віддалений доступ до серверу ЦЗОД по SSH, підготувати каталог проєкту та перевірити працездатність інструментарію Docker Engine і Docker Compose.

Підготовка контейнеризованого сервісу

4. Визначити з викладачем параметри контейнеризованого сервісу і його функціональне призначення та мінімальні вимоги до виконання (порти, змінні середовища, файлові ресурси).

5. Розробити Dockerfile відповідно до обраного сервісу, забезпечивши коректне формування контейнерного образу та мінімізацію надлишкового контенту у контексті збірки.

Розгортання та експлуатаційні дії

6. Здійснити побудову (build) контейнерного образу та виконати верифікацію збірки (успішність, відсутність критичних помилок, відповідність очікуваній конфігурації).

7. Виконати розгортання сервісу у вигляді контейнера, налаштувавши параметри запуску (відображення портів, змінні середовища,

політику перезапуску, а також механізм збереження даних за допомогою томів або bind mount).

8. Описати конфігурацію розгортання у форматі Docker Compose (compose.yaml), включивши необхідні компоненти (сервіс, мережі, томи), та забезпечити відтворюваність запуску в середовищі ЦЗОД.

9. Провести функціональну перевірку розгорнутого сервісу, підтвердивши доступність, коректність обробки запитів (або надання контенту), а також узгодженість результатів із заданими параметрами конфігурації.

10. Виконати базові операції супроводу контейнеризованого сервісу, а саме перегляд журналів, контроль стану контейнерів, керований перезапуск, інспекція параметрів, оцінювання використання ресурсів.

11. Здійснити оновлення розгортання (зміна версії образу або параметрів конфігурації) та підтвердити збереження даних у томах і стабільність працездатності сервісу після оновлення.

ОФОРМЛЕННЯ ЗВІТУ

У звіт лабораторної роботи помістити:

- Завдання л/роботи.
- Покрокові матеріали що повністю підтверджують виконання л/роботи (текст, таблиці, принтскріни, та інші матеріали для підтвердження виконання л/роботи): Dockerfile, compose.yaml, короткий опис архітектури розгортання, а також докази перевірки працездатності і відтворюваності (вихідні повідомлення або скріншоти стану та логів).

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

КОНТРОЛЬНІ ЗАПИТАННЯ

1. У чому різниця між Docker образом і контейнером?
2. Які ключові інструкції Dockerfile використовуються для формування образу і що вони означають?
3. Для чого потрібен .dockerignore і як він впливає на збірку.
4. Чим відрізняються іменовані томи від bind mount і коли який підхід доцільний?
5. Як працює мережа контейнерів у Docker Compose і навіщо потрібні окремі мережі?
6. Що таке port mapping і які ризики виникають при відкритті портів у ЦЗОД.
7. Для чого використовується restart policy і як вона впливає на доступність сервісу?
8. Як організувати оновлення сервісу через Docker Compose так, щоб зменшити час недоступності?

ЛАБОРАТОРНА РОБОТА №3

ТЕМА: РОЗГОРТАННЯ ТА АДМІНІСТРУВАННЯ ВУЗЛА РОЗПОДІЛЕНИХ ОБЧИСЛЕНЬ ІЗ ЗАСТОСУВАННЯМ ПІДХОДІВ БЕЗПЕЧНИХ ОБЧИСЛЕНЬ

МЕТА: Сформувати практичні навички розгортання та адміністрування програмно технічної інфраструктури у центрі зберігання та обробки даних із застосуванням засобів автоматизації, моніторингу та віддаленого управління на прикладі сервісу Berkeley Open Infrastructure for Network Computing (BOINC) і підсистеми віртуалізації VirtualBox

ЗНАТИ: Використання SSH (Secure Shell)

ТЕОРЕТИЧНІ ВІДОМОСТІ

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

Основна документація:

- <https://github.com/BOINC/boinc/wiki/Installing-BOINC-on-Debian-or-Ubuntu>
- <https://wiki.debian.org/BOINC>
- <https://wiki.archlinux.org/title/BOINC>
- <https://docs.oracle.com/en/virtualization/virtualbox/7.2/user/EN-VBOX-7-2-USER.pdf>
- <https://github.com/BOINC/boinc/wiki/VboxApps>
- https://github.com/BOINC/boinc/wiki/Controlling_BOINC_remotely
- <https://prometheus.io/docs/guides/node-exporter/>
- <https://www2.cs.uh.edu/~jaspal/cosc7397/06papers/boinc.pdf>

Додаткова документація:

- <https://roadmap.sh/devops>
- <https://www.gnu.org/software/bash/manual/>
- <https://docs.docker.com/>
- <https://docs.gitlab.com/ci/>
- <https://www.freedesktop.org/software/systemd/man/latest/>

УМОВИ ВИКОНАННЯ Л/Р

Лабораторна робота виконується на сервері центру зберігання та обробки даних ЦНТУ, що передбачено ресурсним забезпеченням кафедри КБПЗ ЦНТУ. При переході ЦНТУ у он-лайн навчання, формат взаємодії питати у викладача що проводить л/р.

ЗАВДАННЯ

У межах лабораторної роботи здобувач вищої освіти розгортає VOINC як сервіс, перевіряє готовність підсистеми віртуалізації та виконує підключення вузла до обраного VOINC проекту з контролем стану, журналів і параметрів ресурсів.

Підготовка

1. Організувати віддалене управління сервером через SSH, зафіксувати обліковий запис, IP або DNS, версію ОС, ядро, CPU, RAM, доступність Інтернету. Перевірити стан служби синхронізації часу та коректність системного часу, як передумову коректної роботи сервісів і TLS з'єднань.

Розгортання VOINC як сервісу

2. Встановити VOINC client та додаткові пакети керування, VOINC Manager або boinctui.

3. Налаштувати запуск BOINC як системної служби, перевірити автозапуск після перезавантаження, перевірити статус служби та журнали подій.

4. Підготувати мінімальний сценарій автоматизації розгортання, наприклад bash скрипт або Ansible playbook, який відтворює встановлення, запуск та первинну перевірку сервісу.

Розгортання та валідація підсистеми віртуалізації

5. Встановити VirtualBox, перевірити, що апаратна віртуалізація увімкнена на рівні BIOS або UEFI і доступна в ОС. Перевірити коректність встановлення VirtualBox, версію, доступність модулів, базову працездатність, зокрема можливість ініціалізації гіпервізора, або запуск тестової віртуальної машини за політикою ЦЗОД.

Додатково встановити VirtualBox Extension Pack сумісний з основним дистрибутивом VirtualBox, для забезпечення повної функціональності та перевірити його коректну інтеграцію з системою

Підключення вузла до BOINC проєкту та реєстрація

6. Підключитися до локального сервісу BOINC через BOINC Manager або boinctui. Зареєструвати обліковий запис на сайті обраного BOINC проєкту. Додати проєкт у BOINC, виконати прив'язку вузла до облікового запису. На сайті проєкту перевірити, що сервер коректно відображається як host, а також перевірити, що підтягнулися параметри хоста, зокрема ознаки віртуалізації та наявність VirtualBox, якщо проєкт використовує віртуалізовані задачі.

Налаштування виконання задач, ресурси, надійність

7. У параметрах проєкту та клієнта BOINC налаштувати політики ресурсів, частка CPU, обмеження RAM, часові вікна роботи, мережеві обмеження, відповідно до правил експлуатації серверів. Вказати, які підпроєкти або типи задач дозволено отримувати, якщо це підтримується проєктом. Дочекатися отримання задач, зафіксувати перехід у стани download і running, проконтролювати журнали BOINC та системні журнали,

підтвердити завершення хоча б однієї задачі або стабільне виконання протягом заданого інтервалу.

Моніторинг та спостережуваність

8. Налаштувати збір і перегляд журналів BOINC та VirtualBox у системних журналах, забезпечити відтворюваний спосіб діагностики типових збоїв. Реалізувати мінімальний контроль стану, наприклад health check через systemd, ідентифікацію процесів, контроль використання ресурсів, CPU, RAM, диск, мережа.

ОФОРМЛЕННЯ ЗВІТУ

У звіт лабораторної роботи помістити:

- Завдання л/роботи.
- Покрокові матеріали що повністю підтверджують виконання л/роботи (текст, таблиці, принтскрини, та інші матеріали для підтвердження виконання л/роботи).
- Короткі змістовні відповіді на л/роботи.

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

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Які компоненти формують програмно технічну інфраструктуру сервера у ЦЗОД у межах цієї роботи, і які ролі виконують BOINC та VirtualBox?
2. Які ризики для надійності виникають при виконанні розподілених обчислень на сервері ЦЗОД, і як їх знижувати?
3. Які системні журнали та метрики є мінімально достатніми для моніторингу працездатності BOINC сервісу?
4. Як перевіряється доступність апаратної віртуалізації, і чому це критично для частини BOINC проєктів?
5. Які параметри обмеження ресурсів BOINC є пріоритетними у серверному середовищі, і як вони впливають на масштабованість?

6. Яким чином автоматизація розгортання підвищує відтворюваність і керованість інфраструктури, що саме автоматизується в цій роботі?
7. Як організовується віддалене управління та контроль змін у конфігурації сервісу на сервері?
8. Які симптоми свідчать про некоректну інтеграцію BOINC з VirtualBox, і які кроки діагностики є першочерговими?

ЛАБОРАТОРНА РОБОТА №4

ТЕМА: МОНІТОРИНГ ТА ВІДДАЛЕНЕ УПРАВЛІННЯ ПРОГРАМНО-ТЕХНІЧНОЮ ІНФРАСТРУКТУРОЮ ЦЕНТРУ ЗБЕРІГАННЯ ТА ОБРОБКИ ДАНИХ

МЕТА: Сформувати практичні навички розгортання та адміністрування програмно технічної інфраструктури центру зберігання та обробки даних шляхом застосування засобів автоматизації, контейнеризації, моніторингу та віддаленого управління, забезпечуючи надійність і масштабованість комп'ютерних систем на прикладі побудови системи збору метрик сервера і параметрів живлення з візуалізацією у Grafana.

ЗНАТИ: Принципи розгортання і адміністрування із застосуванням автоматизації і віддаленого управління

ТЕОРЕТИЧНІ ВІДОМОСТІ

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

Основна документація:

– Prometheus Getting started: пояснює базову архітектуру Prometheus, конфігурацію scrape та перевірку збору метрик.

https://prometheus.io/docs/prometheus/latest/getting_started/

– Prometheus Node Exporter guide: показує, як отримувати метрики Linux сервера (CPU, RAM, диски, мережа) у форматі Prometheus.

<https://prometheus.io/docs/guides/node-exporter/>

– Prometheus Exporters and integrations. Офіційний перелік експортерів, зокрема SNMP exporter і NUT exporter <https://prometheus.io/docs/instrumenting/exporters/>

– SNMP Exporter (офіційний репозиторій) описує підхід до збору SNMP даних (UPS або PDU), конфігурацію модулів і генератор конфігів. https://github.com/prometheus/snmp_exporter

– Prometheus NUT Exporter описує збір метрик UPS через Network UPS Tools і експортування їх у Prometheus <https://github.com/HON95/prometheus-nut-exporter>

– Get started with Grafana and Prometheus показує побудову перших дашбордів у Grafana на даних Prometheus <https://grafana.com/docs/grafana/latest/fundamentals/getting-started/first-dashboards/get-started-grafana-prometheus/>

– Grafana Prometheus data source: пояснює налаштування джерела даних Prometheus у Grafana (параметри, підключення, перевірка) <https://grafana.com/docs/grafana/latest/datasources/prometheus/configure/>

– Docker Compose file reference дає специфікацію compose файлу для відтворюваного контейнеризованого розгортання Prometheus і Grafana <https://docs.docker.com/reference/compose-file>

– IPMI Exporter використовується для збору апаратних метрик сервера, особливо параметри електроживлення https://github.com/prometheus-community/ipmi_exporter

– Ansible як основний засіб розгортання та створення автоматичних конфігурацій запуску програмних продуктів на серверній частині

Додаткова документація:

– <https://roadmap.sh/devops>

– <https://www.gnu.org/software/bash/manual/>

– <https://docs.docker.com/i>

– https://docs.ansible.com/projects/ansible/latest/getting_started/index.html

УМОВИ ВИКОНАННЯ Л/Р

Лабораторна робота виконується на сервері центру зберігання та обробки даних ЦНТУ, що передбачено ресурсним забезпеченням кафедри КБПЗ ЦНТУ. При переході ЦНТУ у он-лайн навчання, формат взаємодії питати у викладача що проводить л/р.

ЗАВДАННЯ

У межах лабораторної роботи здобувач вищої освіти налаштовує моніторинг сервера ЦЗОД ЦНТУ та параметрів живлення через Raspberry Pi з візуалізацією метрик у Grafana.

Підготовка

1. Отримати доступ (облікові дані, адреси, політики доступу) до сервера центру зберігання та обробки даних ЦНТУ та до Raspberry Pi як вузла збору даних. Узгодити перелік об'єктів моніторингу (у викладача що проводить л/р.): сервер (обчислювальні ресурси та мережеві інтерфейси) і підсистема живлення (UPS або PDU або інше джерело телеметрії).

Підготовка програмно-технічного середовища

2. Забезпечити мережеву досяжність між Raspberry Pi, сервером і джерелом даних живлення, з урахуванням правил доступу в ЦЗОД.

Розгортання системи моніторингу з контейнеризацією

3. Розгорнути у контейнерах стек моніторингу (Prometheus як сховище даних для метрик, Grafana як візуалізація). Це реалізує додаткових прошарок паравіртуалізації та безпеки в адмініструванні інфраструктури. Для реалізації запропоновано використання Docker Compose. Для автоматизації запуску та відтворення поточного статусу рекомендується використання Ansible.

Підключення метрик сервера

4. Організувати збір базових метрик сервера (навантаження на процесори, використання оперативної пам'яті, операції та використання

накопичувачів, кількість та швидкість мережевих операцій, температурні показники, тощо) через стандартний підхід експорту метрик і підключення до Prometheus. Перевірити, що метрики регулярно оновлюються та коректно ідентифікуються мітками (ім'я вузла, роль, середовище). Особливо необхідно звернути увагу на відображення назви серверів, замість їх ір-адреси у системі Grafana.

Підключення метрик живлення через Raspberry Pi

5. Визначити інтерфейс отримання даних живлення (наприклад SNMP для UPS або PDU, або інший доступний канал телеметрії). Налаштувати Raspberry Pi як вузол збору телеметрії живлення і як точку експорту метрик у форматі, придатному для Prometheus. Додати відповідні цілі збору до конфігурації Prometheus і перевірити повноту показників (напруга, навантаження, стан батареї та інші доступні параметри). Також, за погодження з лектором, необхідно збирати метрики використання електроенергії, статус системи охолодження та поточний стан окремо з кожного сервера за допомогою IPMI. Для цього також необхідно зробити окремий дашборд.

Візуалізація у Grafana

6. Враховуючи, що Prometheus як джерело даних Grafana вже працює та відображає дані перераховані в завданні 4, створити окремий дашборд з мінімально необхідними панелями: навантаження на процесор; використання оперативної пам'яті; дисковий простір та операції накопичувачів; мережевий трафік; показники живлення (ключові параметри UPS або PDU). Додати часові фільтри та короткі підписи панелей, щоб дашборд був придатний для експлуатаційного використання.

Перевірка працездатності та надійності

7. Провести контрольну перевірку, створити короткочасну зміну навантаження на сервері допустимими засобами та переконатися, що графіки відображають зміну. Налаштувати автозапуск контейнерів і відновлення роботи після перезавантаження, як елемент забезпечення надійності.

Автоматизація та віддалене управління

8. Описати у конфігураціях повторюваний процес розгортання (одна команда або один сценарій запуску стеку, єдине місце налаштувань). Зафіксувати регламент віддалених дій: перегляд стану, перезапуск сервісів моніторингу, перевірка доступності кінцевих точок. Для забезпечення роботи сценарію рекомендується використовувати ansible

ОФОРМЛЕННЯ ЗВІТУ

У звіт лабораторної роботи помістити:

- Завдання л/роботи.
- Покрокові матеріали що повністю підтверджують виконання л/роботи (текст, таблиці, принтскрини, та інші матеріали для підтвердження виконання л/роботи): схему взаємодії компонентів (сервер, Raspberry Pi, Prometheus, Grafana, джерело телеметрії живлення); перелік зібраних метрик (серверні та енергетичні); скріншоти дашборда Grafana; фрагменти конфігурацій, що підтверджують контейнеризацію, збір метрик і автозапуск; надати короткий опис, як виконана л/робота реалізує супровід інфраструктури через автоматизацію, розгортання та моніторинг у ЦЗОД і як забезпечується адміністрування з контейнеризацією та віддаленим управлінням

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

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Яку роль виконує Raspberry Pi у запропонованій системі моніторингу ЦЗОД та які переваги це дає для супроводу інфраструктури, зокрема у інтернет сервіс провайдерах?
2. Які метрики сервера є базовими для оцінювання надійності та потенціалу масштабування і як їх інтерпретувати на графіках?

3. У чому різниця між Pull та Push моделями збору метрик і яку модель доцільно застосувати для Prometheus у цій л/р?
4. Які типові ризики виникають під час контейнеризованого розгортання Prometheus і Grafana у ЦЗОД та як їх мінімізувати?
5. Для чого у метриках використовують мітки (labels) і як мітки допомагають розділяти дані за вузлами, ролями та середовищами?
6. Які варіанти отримання телеметрії живлення (UPS або PDU) можливі у ЦЗОД і які ключові параметри живлення варто контролювати?
7. Як забезпечити відтворюваність і автоматизацію розгортання стеку моніторингу та які артефакти конфігурацій мають це підтверджувати.
8. Які дії віддаленого адміністрування потрібні для підтримання працездатності системи моніторингу та як перевірити її готовність після перезавантаження?

ЛАБОРАТОРНА РОБОТА №5

ТЕМА: ПЛАТФОРМИ БЕЗПЕРЕРВНОЇ ІНТЕГРАЦІЇ І ДОСТАВКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

МЕТА: Сформувати здатність обґрунтовано обирати та застосовувати інструменти безперервної інтеграції і доставки шляхом розгортання та налаштування типового CI пайплайна проєкту, виконання контрольних сценаріїв і порівняння функціональних, експлуатаційних та організаційних характеристик GitHub Actions, GitLab CI, TeamCity і Jenkins в умовах інфраструктури дата центру.

ЗНАТИ: Базові поняття CI/CD та структури пайплайна

ТЕОРЕТИЧНІ ВІДОМОСТІ

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

Основна документація:

- <https://docs.github.com/actions>
- <https://docs.gitlab.com/ci/>
- <https://www.jenkins.io/doc/book/pipeline/>
- <https://www.jenkins.io/doc/>
- <https://www.jetbrains.com/help/teamcity/teamcity-documentation.html>

Додаткова документація:

- <https://roadmap.sh/devops>
- <https://www.gnu.org/software/bash/manual/>
- <https://docs.docker.com/>
- <https://docs.gitlab.com/ci/>
- <https://www.freedesktop.org/software/systemd/man/latest/>

ЗАВДАННЯ

Ознайомитися з платформами безперервної інтеграції та доставки програмного забезпечення (GitHub Actions, GitLab CI, TeamCity, Jenkins). У реферативному стилі оформити звіт лабораторної роботи.

Звіт повинен включати опис та скріншоти наступних пунктів:

- Ліцензійні вимоги та умови використання платформ;
- Попередні налаштування для початку роботи (створення або підключення проєкту, налаштування агентів або runner, базові параметри доступу);
- Загальний опис архітектури та можливостей платформ (пайплайни, тригери, артефакти, інтеграції, журналювання);
- Переваги та недоліки платформ з позиції навчального та практичного використання;
- Ціни, моделі оплати та особливості експлуатації у власній інфраструктурі дата центру.

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

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Які спільні принципи побудови CI/CD пайплайнів характерні для різних платформ і які елементи є обов'язковими в типовому пайплайні?
2. Які відмінності між підходами до виконання задач у CI/CD платформах (runner/agent/executor) і як це впливає на розгортання у власній інфраструктурі дата центру?
3. Як організуються тригери запуску пайплайна (push, merge request/pull request, розклад) і які сценарії доцільні для навчального проєкту?
4. Які можливості керування доступом і ролями надають CI/CD платформи та які ризики виникають при неправильному налаштуванні прав?

5. Як реалізується керування секретами (tokens, keys, credentials) у CI/CD платформах і які практики є базовими для безпечного використання?
6. Як платформи працюють з артефактами і журналами виконання (збереження, доступ, термін зберігання) і як це впливає на налагодження та аудит?
7. Які критерії доцільно застосовувати для порівняння CI/CD платформ у звіті (функціональність, інтеграції, зручність, масштабованість, вартість володіння, підтримка)?
8. Які моделі ліцензування і ціноутворення притаманні CI/CD платформам та як вони впливають на вибір рішення для використання в організації або у дата центрі.

ЛАБОРАТОРНА РОБОТА №6

ТЕМА: ІНСТРУМЕНТИ ІНФРАСТРУКТУРИ ЯК КОДУ І ЗАСОБИ КЕРУВАННЯ КОНФІГУРАЦІЯМИ

МЕТА: Сформувати практичні навички застосування та обґрунтованого вибору інструментів автоматизації інфраструктури і конфігурацій шляхом реалізації однакового сценарію розгортання та супроводу сервісу на тестових ресурсах дата центру, виконання контрольних перевірок і підготовки порівняльного звіту за визначеними критеріями.

ЗНАТИ: Принципи декларативного і імперативного опису інфраструктури і конфігурацій

ТЕОРЕТИЧНІ ВІДОМОСТІ

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

Основна документація:

- <https://help.puppet.com/core/>
- <https://www.puppet.com/docs/index.html>
- <https://docs.chef.io/>
- https://docs.chef.io/chef_overview/
- <https://developer.hashicorp.com/terraform/docs>
- <https://developer.hashicorp.com/terraform/language>
- <https://docs.ansible.com/>
- <https://docs.aws.amazon.com/cloudformation/>
- <https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/Welcome.html>

Додаткова документація:

- <https://roadmap.sh/devops>
- <https://www.gnu.org/software/bash/manual/>

- <https://docs.docker.com/>
- <https://docs.gitlab.com/ci/>
- <https://www.freedesktop.org/software/systemd/man/latest/>

ЗАВДАННЯ

Ознайомитися з інструментами автоматизації інфраструктури та керування конфігураціями (Puppet, Chef, Terraform, Ansible, AWS CloudFormation). У реферативному стилі оформити звіт лабораторної роботи.

Звіт повинен включати опис та скріншоти наступних пунктів:

- Ліцензійні вимоги та умови використання інструментів;
- Загальний опис підходу та можливостей інструментів (керування конфігураціями або інфраструктурою як код, модульність, шаблони, стан, ідемпотентність, механізми застосування);
- Переваги та недоліки інструментів з позиції використання у дата центрі та прикладних сценаріях;
- Ціни, моделі ліцензування та особливості експлуатації (вимоги до інфраструктури, потреба в агентах, складність супроводу, масштабування, аудит).

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

КОНТРОЛЬНІ ЗАПИТАННЯ

1. У чому полягає відмінність між підходами керування конфігураціями і інфраструктурою як кодом, та які задачі кожен з них вирішує у дата центрі?
2. Які принципові відмінності між агентною і безагентною моделлю керування вузлами, і як це впливає на безпеку та експлуатаційні витрати?

3. Що таке ідемпотентність у контексті автоматизації інфраструктури, і як її перевірити під час повторного запуску сценаріїв?

4. Що таке стан (state) в IaC інструментах, для чого він потрібний і які ризики виникають при його некоректному зберіганні або паралельному доступі?

5. Як інструменти виявляють і усувають drift (відхилення фактичного стану від заданого), і які практичні дії доцільні для контролю drift у ЦЗОД?

6. Які механізми модульності і повторного використання (roles/modules/cookbooks/manifests/stacks) застосовуються в різних інструментах і як вони впливають на масштабування проєкту?

7. Як організовується керування секретами і доступами під час автоматизації (ключі, токени, паролі), і які базові практики забезпечують мінімізацію ризиків?

8. За якими критеріями доцільно порівнювати інструменти для вибору у реальному середовищі (складність впровадження, підтримка, інтеграції, аудит, масштабованість, ліцензування, вартість володіння) та який висновок можна зробити для сценарію дата центру?

ЛАБОРАТОРНА РОБОТА №7

ТЕМА: ВИКОРИСТАННЯ ТА ПОРІВНЯННЯ СЕРВІСІВ MICROSOFT AZURE, AMAZON WEB SERVICES, GOOGLE CLOUD

МЕТА: Отримати практичні навички роботи з програмними продуктами Microsoft Azure, Amazon Web Services, Google cloud

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/>
- Microsoft Azure: <https://azure.microsoft.com/>
- Google cloud <https://cloud.google.com/>

Додаткова документація:

- <https://roadmap.sh/devops>
- <https://www.gnu.org/software/bash/manual/>
- <https://docs.docker.com/>
- <https://docs.gitlab.com/ci/>
- <https://www.freedesktop.org/software/systemd/man/latest/>

ЗАВДАННЯ

Ознайомитися з сервісами Amazon Web Services Microsoft Azure, Google cloud. 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;
- Ціни та особливості використання.

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

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Які ліцензійні та правові умови використання AWS, Microsoft Azure та Google Cloud визначають права користувача і обмеження відповідальності провайдера?
 2. Які типи облікових записів і планів доступні на кожній платформі та які вимоги діють для реєстрації і верифікації?
 3. Які дії входять до початкового налаштування профілю в кожній платформі, включно з налаштуванням безпеки доступу та платіжних параметрів?
 4. Які базові категорії сервісів надають AWS, Azure та Google Cloud і які типові задачі вони покривають?
 5. Які механізми керування доступом користувачів і ролей застосовуються на кожній платформі та чому це критично для безпечного використання?
 6. Які сильні сторони кожної платформи є ключовими для вибору під конкретні сценарії використання?
 7. Які типові обмеження або ризики використання кожної платформи впливають на прийняття рішення?
 8. Які моделі ціноутворення застосовуються в хмарних платформах і які фактори найбільше впливають на підсумкову вартість?

9. Які безкоштовні рівні або пробні кредити існують у AWS, Azure та Google Cloud і які обмеження діють щодо термінів і ресурсів?

10. Які практичні кроки допомагають контролювати витрати під час роботи з платформами та запобігати неочікуваним нарахуванням?

ЛАБОРАТОРНА РОБОТА №8

**ТЕМА: ВИСОКОШВИДКІСНІ МЕРЕЖЕВІ КОНФІГУРАЦІЇ
МАРШРУТИЗАТОРІВ ВИСОКОЇ ПРОДУКТИВНОСТІ У ЦЗОД,
ОЦІНКА ЕНЕРГОЕФЕКТИВНОСТІ ТА ПРОДУКТИВНОСТІ**

МЕТА: Сформувати практичні навички розроблення та реалізації високошвидкісних мережевих рішень L2 і L3 на маршрутизаторах у середовищі ЦЗОБ із застосуванням раціональної обробки трафіку, порівняльне оцінювання енергоспоживання в різних конфігураціях із розрахунком показників енергоефективності та продуктивності, оформлюючи конфігурації та результати як відтворювані артефакти DevOps практики з контролем змін.

ЗНАТИ: Використання SSH (Secure Shell), інтерпретатор командного рядка Bash

ТЕОРЕТИЧНІ ВІДОМОСТІ

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

Основна документація:

- <https://help.mikrotik.com/docs/spaces/ROS/pages/25690117/Health>
- <https://help.mikrotik.com/docs/spaces/ROS/pages/328068/Bridging%2Band%2BSwitching>
- <https://help.mikrotik.com/docs/spaces/ROS/pages/15302988/Switch%2BChip%2BFeatures>
- <https://help.mikrotik.com/docs/spaces/ROS/pages/62390319/L3%2BHardware%2BOffloading>
- <https://help.mikrotik.com/docs/spaces/ROS/pages/7962644/Bandwidth%2BTest>
- <https://help.mikrotik.com/docs/spaces/ROS/pages/128221376/Traffic%2BGenerator>
- <https://help.mikrotik.com/docs/spaces/ROS/pages/328227/Packet%2BFlow%2Bin%2BRouterOS>

Додаткова документація:

- <https://roadmap.sh/devops>
- <https://www.gnu.org/software/bash/manual/>
- <https://docs.docker.com/>

УМОВИ ВИКОНАННЯ Л/Р

Лабораторна робота виконується маршрутизаторі центру зберігання та обробки даних ЦНТУ в узгодженому навчальному сегменті, без впливу на продуктивні сервіси, що передбачено ресурсним забезпеченням кафедри КБПЗ ЦНТУ. Перед внесенням змін студент зберігає поточну конфігурацію та забезпечує можливість відкату (backup). При переході ЦНТУ у он-лайн навчання, формат взаємодії питати у викладача що проводить л/р.

ЗАВДАННЯ

У межах лабораторної роботи студент налаштовує серверний роутер MikroTik ЦЗОД КБПЗ ЦНТУ на високошвидкісній мережеві конфігурації та демонструє різницю енергоспоживання та продуктивність в різних конфігураціях.

Порівнюються 3 конфігурації:

- **Конфігурація 1** високошвидкісний L2 switching з апаратним offload (максимальна пропускна здатність при мінімальному навантаженні CPU).
- **Конфігурація 2** L2 bridging без апаратного offload (контрольний сценарій з перенесенням обробки на CPU для виявлення зміни ват і продуктивності).
- **Конфігурація 3** L3 маршрутизація між VLAN (маршрутизація через CPU).

КРОКИ РЕАЛІЗАЦІЇ Л/Р

Фіксація базового стану

1. Фіксація версії RouterOS, моделі роутера MikroTik, активні інтерфейси та базову топологію тестового стенду.

2. Збереження поточної конфігурації як бекап, з можливістю пожальшого відтворення.

3. Зчитування System Health та фіксування наявності параметра power-consumption, температур і швидкості вентиляторів.

Вимірювання енергоспоживання, стан простою

1. При відсутності тестового трафіку провести вимірювання енергоспоживання протягом 3 хвилин. Фіксація середнього значення ват і діапазон коливань з врахуванням, що надто часте опитування health може впливати на CPU та результати.

Конфігурація 1, L2 switching з апаратним offload

1.1. Реалізація L2 сегментації на базі bridge і VLAN відповідно до схеми стенду (питати у викладача що проводить л/р).

1.2. Формування тестового трафіку (наприклад через Bandwidth Test, Traffic Generator, iPerf3) – **вимір пропускної здатності (throughput), параметрів якості каналу, фіксація енергоспоживання та завантаження CPU.**

Конфігурація 2, L2 bridging без апаратного offload

2.1. Переключення обробки L2 трафіку в програмний режим, змінюючи параметр hw у портах мосту

(<https://help.mikrotik.com/docs/spaces/ROS/pages/328068/Bridging%2Band%2BSwitching>)

2.2. Формування тестового трафіку (наприклад через Bandwidth Test, Traffic Generator, iPerf3) – **вимір пропускної здатності (throughput),**

параметрів якості каналу, фіксація енергоспоживання та завантаження CPU.

2.3 Порівняння throughput, CPU load, power-consumption з конфігурацією 1.

Конфігурація 3, L3 маршрутизація між VLAN

3.1 Налаштування L3 інтерфейсів VLAN та маршрутизацію між ними.

3.2 Вимір маршрутизації через CPU.

3.3 Порівняння throughput, CPU load і power-consumption для конфігурації 1 та 2.

Методика оцінювання енергоефективності

Розрахунок та подача в звіті показники для кожної конфігурації (опціонально подача у вигляді часових графіків перетину):

1. Середнє енергоспоживання, W.
2. Середня пропускна здатність, Gbit/s.
3. Енергоефективність як відношення W на Gbit/s.

ОФОРМЛЕННЯ ЗВІТУ

У звіт лабораторної роботи помістити:

- Завдання л/роботи.
- Покрокові матеріали що повністю підтверджують виконання л/роботи (текст, таблиці, принтскріни, та інші матеріали для підтвердження виконання л/роботи): Опис стенду, схема VLAN і портів, умови тестування; Таблиця чи графіки результатів конфігурації 1,2,

- Короткі змістовні відповіді на л/роботи.

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

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Яким чином апаратний offload у bridge впливає на CPU load і throughput, та як це пов'язано з енергоефективністю?
2. Чому protocol-mode rstp на частині пристроїв впливає на апаратний offload і як це враховувати при проектуванні високошвидкісного L2 сегмента?
3. Який принцип L3 hardware offloading та які переваги він дає для високошвидкісної маршрутизації на CRS?
4. Які обмеження FastTrack потрібно враховувати у складних сценаріях маршрутизації?
5. Чому надмірно часті опитування System Health можуть спотворювати експеримент з вимірювання енергоспоживання?

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. Carter, Thompson. CI/CD for Developers: How to Automate Deployment and Testing in Software Engineering. A Hands-On Guide to Jenkins, GitHub Actions, and Docker Pipelines. Independently published, 2025. ISBN 9798310909687.
2. Gold, Alex. DevOps for Data Science. 1st ed. Chapman & Hall, 2024. 274 p. ISBN 9781032100340.
3. Sharma, Hitesh Kumar; Kumar, Anuj; Pant, Sangeeta; Ram, Mangey. DevOps: A Journey from Microservice to Cloud Based Containerization. 1st ed. River Publishers, 2023. 194 p. ISBN 9788770228466.
4. FreeWheel Biz-UI Team. Cloud-Native Application Architecture: Microservice Development Best Practice. 1st ed. Springer Singapore, 2024. XXI, 382 p. DOI 10.1007/9789811997822. ISBN 9789811997815.
5. Cline, Karen; Mazzara, Manuel; Levina, Daria; Brandic, Ivona (eds.). CyberSecurity in a DevOps Environment: From Requirements to Monitoring. Springer Cham, 2024. 324 p. DOI 10.1007/9783031422126. ISBN 9783031422119
6. 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=resultlist> (**Scopus**).
7. 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». Вимірвальна

та обчислювальна техніка в технологічних процесах, № 1 (2022). С. 15-26.
2022. Режим доступу:

<http://vottp.khmnmu.edu.ua/index.php/vottp/article/view/30/36> (Фахове видання.
Категорія «Б»)

8. Коваленко О.В. Масштабування імітаційної моделі технології тестування безпеки. Системи 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 (Фахове видання. Категорія «Б»)

9. 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

(Фахове видання. Категорія «Б»)

10. 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»)

11. 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 (Фахове видання. Категорія «Б»)

Допоміжна

12. Коваленко О.В., Смирнов О.А., Комплекс математичних моделей технології тестування WEB-додатків. Інформаційні технології: сучасний стан та перспективи: монографія / За загальною редакцією В.С. Пономаренка. – Х.: ТОВ «ДІСА ПЛЮС», 2018. – 461 с.

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

14. Daniel J. Barrett. Efficient Linux at the Command Line. O'Reilly Media, Inc. 2022. 248 c.
15. Shaun R Smith, Peter Membrey. Beginning Ansible Concepts and Application. Apress Berkeley. 2022. 296 c.
16. Dan Sullivan. Google Cloud Certified Professional Cloud Architect Study Guide Second Edition. John Wiley & Sons, Inc. 2022. 371 c.
17. Michael Elder, Jake Kitchener, and Dr. Brad Topol. Hybrid Cloud Apps with OpenShift and Kubernetes. O'Reilly Media, Inc. 2021. 270 c.
18. J.D. Linux The Greatest Guide for Beginners to the Linux Commands and Linux Operating System. Independently published. 2021. 108 c.
19. John Culkin, Mike Zazon AWS. Cookbook Recipes for Success on AWS. O'Reilly Media. 2022. 568c.
20. Hiroko Nishimura. AWS for Non-Engineers. Manning Publications. 2022. 132 c.
21. Simon Bisson, Mary Branscombe, Chris Hoder & Anand Raman. Azure AI Services at Scale for Cloud, Mobile, and Edge. O'Reilly Media. 2022. 227 c.
22. Carl Albing, JP Vossen. bash Idioms. O'Reilly Media. 2022. 134 c.
23. Ashley Davis. Bootstrapping Microservices with Docker, Kubernetes, and Terraform. Manning Publications. 2021. 442 c.
24. Josh Armitage. Cloud Native Security Cookbook. O'Reilly Media. 2022. 516 c.
25. Mauricio Salatino Continuous Delivery for Kubernetes. Manning Publications. 2022. 202 c.
26. Jay Vyas, Chris Love. Core Kubernetes. Manning Publications. 2022. 336 c.
27. Bradley Smith. Devops for the Desperate a Hands-on Survival Guide. No Starch Press. 2022. 188 c.
28. Stephen Chin, Melissa McKay, Ixchel Ruiz, Baruch Sadogursky. DevOps Tools for Java Developers. O'Reilly Media. 2022. 344 c.
29. Jeroen Mulder. Enterprise DevOps for Architects. Packt Publishing. 2021. 289 c.

30. Nigel Poulton. Docker Deep Dive. Packt Publishing. 2020. 350 с.
31. William Denniss. Kubernetes for Developers. Manning Publications. 2022. 258 с.
32. John Clingan, Ken Finnigan. Kubernetes Native Microservices with Quarkus and MicroProfile. Manning Publications. 2022. 330 с.
33. James Petty, Travis Plunk, Tyler Leonhardt, Don Jones, Jeffery Hicks. Learn PowerShell in a Month of Lunches. Manning Publications. 2022. 359 с.
34. Michael Hausenblas. Learning Modern Linux. O'Reilly Media. 2022. 261 с.
35. Vedran Dakic, Jasmin Redzepagic. Linux Command Line and Shell Scripting Techniques. Packt Publishing. 2022. 552 с.
36. Jack-Benny Persson. Linux System Programming Techniques. Packt Publishing. 2021. 432 с.
37. Scott Alan Miller. Linux Administration Best Practices. Packt Publishing. 2022. 404 с.
38. James Freeman, Jesse Keating. Mastering Ansible. Packt Publishing. 2021. 540 с.
39. Russ McKendrick. Mastering Docker. Packt Publishing. 2020. 568 с.
40. Mark S. Merkow. Practical Security for Agile and DevOps. CRC Press. 2022. 236 с.
41. Richard Bullington-McGuire, Andrew K. Dennis, Michael Schwartz. Docker for Developers. Packt Publishing. 2020. 468 с.

Інформаційні ресурси

42. Курс «DevOps та Cloud-обчислення» на сервері дистанційної освіта ЦНТУ. – URL: <https://moodle.kntu.kr.ua/course/view.php?id=1086>
43. Основи створення мобільних додатків [Електронний ресурс]. – Режим доступу: <https://developer.android.com/guide>
44. Онлайн-курси UDEMY. – URL: <https://www.udemy.com/> –платформа онлайн-курсів різних ІТ тематик.

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