



Приватний заклад вищої освіти
Одеський технологічний університет «ШАГ»
Кафедра інформаційних технологій та фундаментальної підготовки

Випускна кваліфікаційна робота бакалавра

«СИСТЕМА УПРАВЛІННЯ КОНТЕНТОМ ЕЛЕКТРОННОЇ КОМЕРЦІЇ (за прикладом сервісу Comfy) (DevOps)»

на здобуття бакалаврського рівня вищої освіти
зі спеціальності «122 Комп'ютерні науки»

Виконавці проекту

№	П.І.Б.	Група
1	Єлізар'єв В. А.	КН-П-191
2	Бондарчук М. Є.	КН-П-191
3	Ісаєнко Д. О.	КН-П-191
4	Сідху Т.	КН-Д-192
5	Лавров А. В.	КН-А-191

Автор звіту

_____ Лавров Андрій Вікторович

АНОТАЦІЯ

Лавров А. В. Автореферат випускної кваліфікаційної роботи на здобуття бакалаврського рівня вищої освіти зі спеціальності «Комп'ютерні науки». - Одеса: ОТУШ, 2023 - 13 с.

Дипломний проєкт присвячений налаштування DevOps частині онлайн-гіпермаркету на прикладі Comfy.ua. Робота включає аналіз, налаштування та розгортання онлайн-платформи для електронної торгівлі з великим асортиментом товарів.

ПЗ мають електронний каталог товарів, де покупці можуть вибрати необхідні товари. Кожен товар супроводжується описом, ціною та іноді відгуками від інших покупців.

Після вибору товарів покупці можуть додати в кошик і перейти до оформлення замовлення. Онлайн-гіпермаркету може пропонувати різні варіанти доставки, включаючи доставку додому або самовивіз з магазину. Після оплати замовлення він обробляється та готується до доставки або самовивезення.

Переваги ПЗ включають зручність та гнучкість для покупців, які можуть робити покупки у будь-який зручний для них час та з будь-якого місця з доступом до інтернету. Також онлайн-гіпермаркету можуть пропонувати акції та знижки щоб залучити та утримати клієнтів.

Важливим аспектом ПЗ є забезпечення безпеки платежів та захисту особистої інформації покупців. Тому ПЗ зазвичай використовують шифрування даних та інші заходи безпеки захисту інформації клієнтів.

Для розробки на стороні сервера використовується технологія ASP .NET Core 6, мова програмування C# і база даних PostgreSQL і Redis. Для розгортання та хостування додатку використовуються технології AWS, Docker, Jenkins. Розробка на клієнтській стороні використовує мову програмування Node JS, React і Redux.

ЗМІСТ

ВСТУП_____	4
Технічне завдання до проекту_____	5
РОЗДІЛ 1. Загальна характеристика системи_____	7
РОЗДІЛ 2. Реалізація DevOPS частини_____	9
РОЗДІЛ 3. Оцінка_____	14
ВИСНОВКИ_____	15
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ_____	16

ВСТУП

У сучасному динамічному світі розробка програмного забезпечення та операційна підтримка стикаються з постійно мінливими вимогами та викликами. Для успішного розвитку та надання високоякісних продуктів необхідна ефективна взаємодія та співпраця між командами розробників та операційних фахівців. У цьому контексті DevOps стає ключовим підходом, що поєднує розробку.

Роль та значення DevOps у сучасній розробці програмного забезпечення неухильно зростають. Використання DevOps-підходу дозволяє знизити час доставки нових функцій та покращень, підвищити надійність та стабільність додатків, а також забезпечити більш гнучку та прозору взаємодію між командами розробки та операцій. У світлі цих переваг, розуміння та застосування DevOps-практик стає невід'ємною частиною сучасної програмістської освіти.

У даному промі розглянемо проєкт зроблений на основі Comfy. Система управління контентом електронної комерції, її реалізацію зі сторони DevOPS.

ТЕХНІЧНЕ ЗАВДАННЯ ДО ПРОЕКТУ

Загальний опис

Програмне забезпечення для створення інформаційної системи інтернет крамниці для організації роздрібної інтернет торгівлі.

Призначення

Надання в Інтернет просторі інформації про доступні товари (оферти продавця) та організації оформлення замовлень (акцепту покупця) через мережу Інтернет. Оперативне управління веб-контентом Інтернет крамниці. Збереження даних та регулювання доступу до них. Організація взаємодії з іншими інформаційними системами.

Вимоги до функціоналу

- Інтернет крамниця
 - Навігація по веб вітрині крамниці
 - Пошук товарів за критеріями
 - Реєстрація покупців
 - Формування кошика покупця
 - Оформлення замовлень
 - Зворотній зв'язок та відгуки
- Управління контентом інтернет крамниці
 - Навігація, перегляд, створення, редагування структури та контенту
 - Керування доступом та розподіл дозволів користувачів

- Створення вітрини товарів
- Управління даними користувачів
- Оформлення замовлень
- Модерація коментарів
- Реєстрація та авторизація користувачів
 - Користувач ПЗ має змогу зареєструватись чи увійти у систему як за допомогою авторизаційних даних (email та пароль), так і за допомогою облікових записів популярних соціальних мереж (Google, тощо)
 - Користувач ПЗ має змогу увімкнути та вимкнути функцію двофакторної автентифікації
- Особистий кабінет покупця
 - Перегляд історії замовлень
 - Кошик покупця
 - Реквізити для платежів та доставки
- Взаємодія із інтернет додатками
 - Web-API моделі та запити для управління аккаунтом користувача
 - Web-API моделі та запити для одержання, створення та редагування інформації, що формує профіль покупця
 - Web-API моделі та запити для вибору товару в корзину та оформлення замовлення користувачем

- Web-API моделі та запити для створення питань та відгуків, а також для їх оцінок (лайків/дізлайків)
- Web-API моделі та запити для пошуку товарів з необхідними фільтрацією та сортуванням
- Web-API моделі та запити для отримання категорій товарів, банерів, тощо

Архітектура:

- Сервер бази даних - серверне програмне забезпечення, що забезпечує збереження та обробку поточних даних.
- Сервіс Web-API – серверне програмне забезпечення, що надає програмний інтерфейс для взаємодії (REST-API) з іншим програмним забезпеченням.
- Веб додаток керування контентом - доступний з веб-простору сайт, що забезпечує візуальний інтерфейс для менеджерів системи.
- Веб крамниця - доступний з веб-простору сайт, що дозволяє взаємодіяти з інтернет-магазином покупцям.

Технології:

- Для кожного з модулів рекомендується (з можливістю часткової зміни) наступні технології та методології створення:
 - Бази даних – PostgreSQL, Redis
 - Сервіс Web-API – REST-API, ASP.NET Core, C#
 - Веб додаток керування контентом – MVC, ASP.NET Core, Razor Views, C#
 - Веб додаток для покупця – Node JS, React, Redux

- При розгортанні програмного забезпечення рекомендується задіяти хмарні технології
- При розробці програмного забезпечення необхідно задіяти зберігання вихідних кодів ПЗ повинно здійснюватись за допомогою системи контролю версій (вибір конкретної системи узгоджується групою проекту).
- Вихідний код повинен оформлятися у відповідності до загальноприйнятих стандартів, на зразок стандартів GNU <http://www.gnu.org/prep/standards/>, у тому числі з дотриманням вимог до коментування та документування коду.

РОЗДІЛ 1

ЗАГАЛЬНА ХАРАКТЕРИСТИКА СИСТЕМИ

З вимогами для реалізації проекту треба:

- «Frontend розробник»
- «Backend розробник»
- «Дизайнер»
- «DevOps інженер»

«FRONTEND»

«Frontend» містить розробку та реалізацію веб-сторінок, взаємодію з користувачем, навігацію, відображення товарів, розташування категорій, оформлення замовлень та багато іншого. Вона забезпечує зручний та багатофункціональний інтерфейс, де користувачі можуть шукати та придбати товари, додавати їх до кошика, переглядати інформацію про товари, редагувати персональні дані тощо.

«BACKEND»

«Backend» розроблена мовою програмування C#, за допомогою фреймворку ASP.NET Core . Для надійного зберігання персональних було використано СУБД PostgreSQL. Також ця СУБД підтримується популярним ORM (Object Relational Mapping) Entity Framework Core від Microsoft, який спрощує розробку та взаємодію з базою даних за допомогою мови запитів LINQ. Сервер реалізує RESTful API, який обробляє мережеві запити від клієнтської частини за допомогою протоколу HTTP. Щоб забезпечити безпеку у додатку, була реалізована авторизація та автентифікація за допомогою JWT та Refresh токенів.

«Дизайн»

«Дизайн»- частина яка відповідає за створення та розробки вигляду інтерфейсу. Типовий інтерфейс користувача використовує методи UI/UX.

«Дизайн» включає розробку та організацію вигляду сторінок та компонентів, вибір кольорової схеми, типографіки та графічних елементів, створення інтуїтивно зрозумілої навігації, анімації та взаємодії з користувачем.

«DEVOPS»

«DevOps» займає роль у розробці нашого проекту реалізацію постійної автоматизації безперервного розгортання веб-сайту на хмарному сервері AWS EC2. Ми використовуємо технологію Docker для розгортання, а за виконання автоматизація відповідає Jenkins.

Методологія «DevOps» також має на увазі проксування, який забезпечує доступ клієнта до сервера. Для просування ми використовуємо nginx, який гарантує безпеку та ефективність зв'язку з сервером.

Системи моніторингу яку ми використовуємо Prometheus, він дозволяє нам стежити за споживанням ресурсів, аналізувати дані та вчасно виявляти будь-які проблеми.

РОЗДІЛ 2

РЕАЛІЗАЦІЯ DEVOPS ЧАСТИНИ

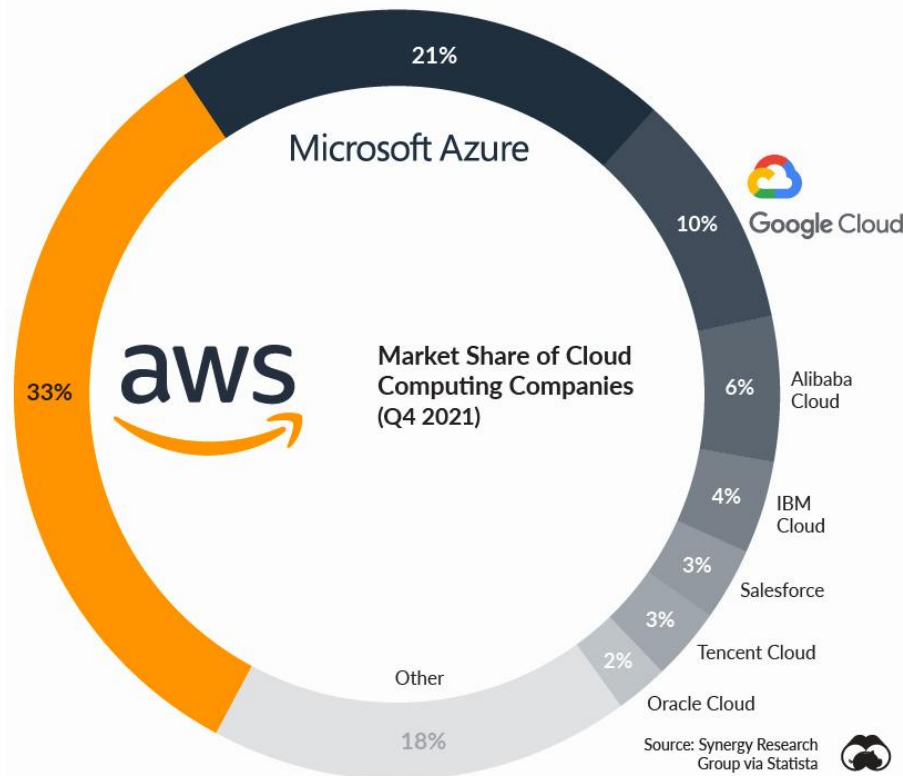
Для визначення потреб проєкту для застосування практики DevOps. Ми з командою визначили, які компоненти і процеси повинні бути:

- Хмарні технології
- Постійна інтеграція та розгортання (CI/CD)
- Моніторинг системи
- Безпека та захист даних

Спочатку ми вибрали де саме буде розгортатися ПЗ. Ми вибрали хмарним сервіс AWS EC2. Чому саме хмара? І чому AWS?

Хмарні послуги можуть надати низку переваг організаціям, які хочуть підтримувати свої програми. Ці послуги можуть допомогти знизити загальну вартість володіння додатком та пропонують безліч інших переваг. Можливо, найбільш важливим є те, що хмарні сервіси можуть допомогти підвищити продуктивність і доступність додатків.

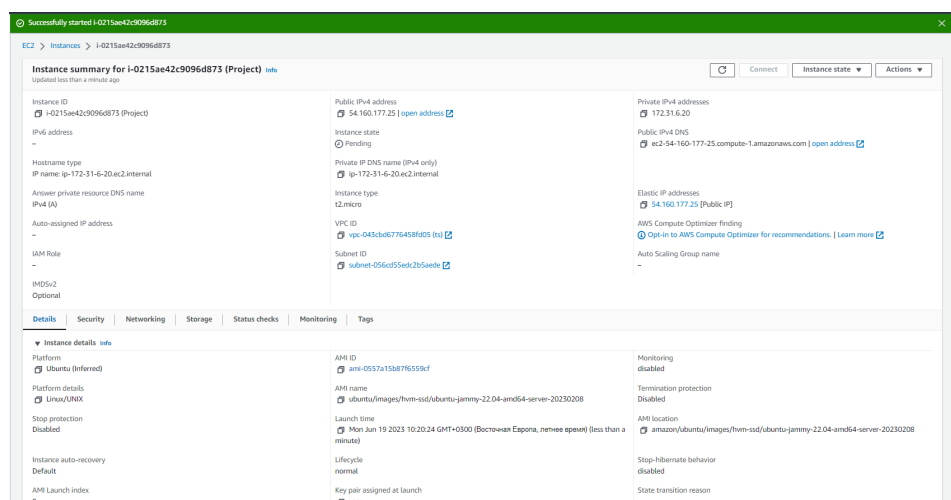
Серед всіх хмарними сервісами присутні такі як(Google Cloud Platform, Microsoft Azure, Amazon Web Services і так далі) ми вибрали з них AWS. AWS Cloud - це найпоширеніша у світі хмара з найширшими можливостями, що надає понад 200 повнофункціональних сервісів для центрів обробки даних у всьому світі. AWS має глобальну інфраструктуру даних та центрів обробки даних (регіони), що дозволяє клієнтам розгортати свої програми та забезпечувати високу доступність та швидкодію для своїх користувачів по всьому світу.



Зобр. Статистика між усіма хмарними сервісами

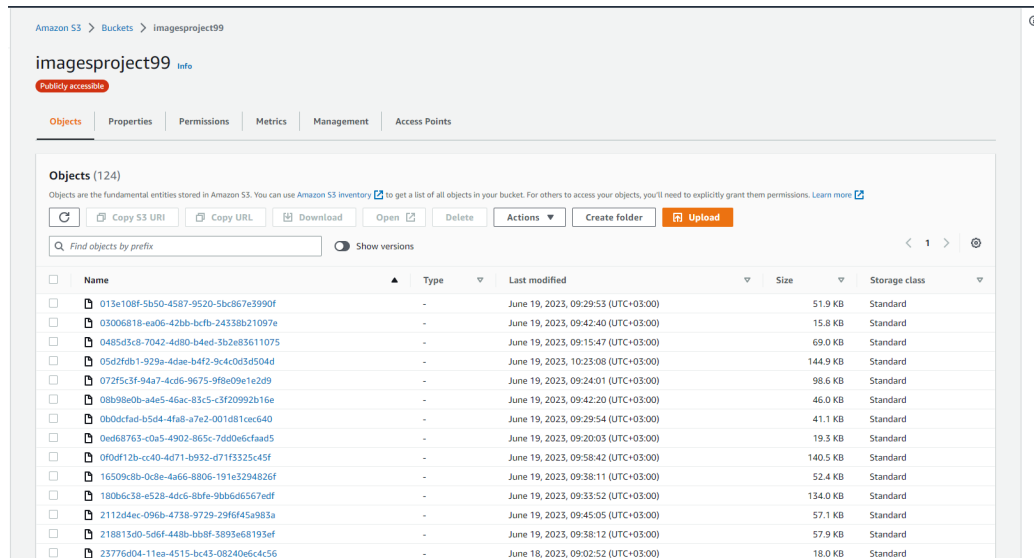
Сервіси які будуть використовуватися:

- EC2 - надає віртуальні сервери у хмарі, на яких можна запускати різні програми та робочі навантаження.



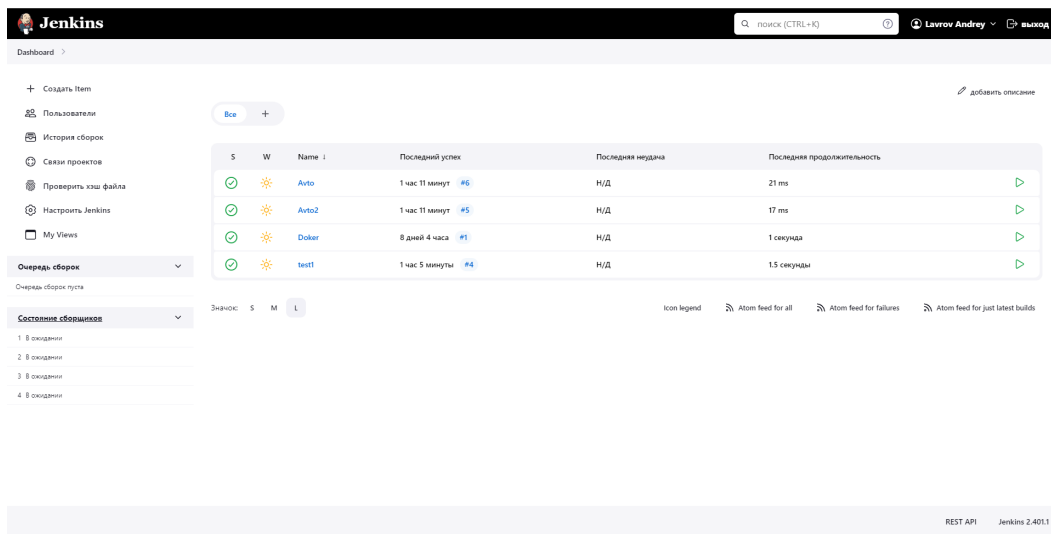
Зобр. VM Linux зі статичним IP

- S3 - це сховище об'єктів, яке дозволяє зберігати та виймати будь-яку кількість даних з будь-якого місця в Інтернеті.



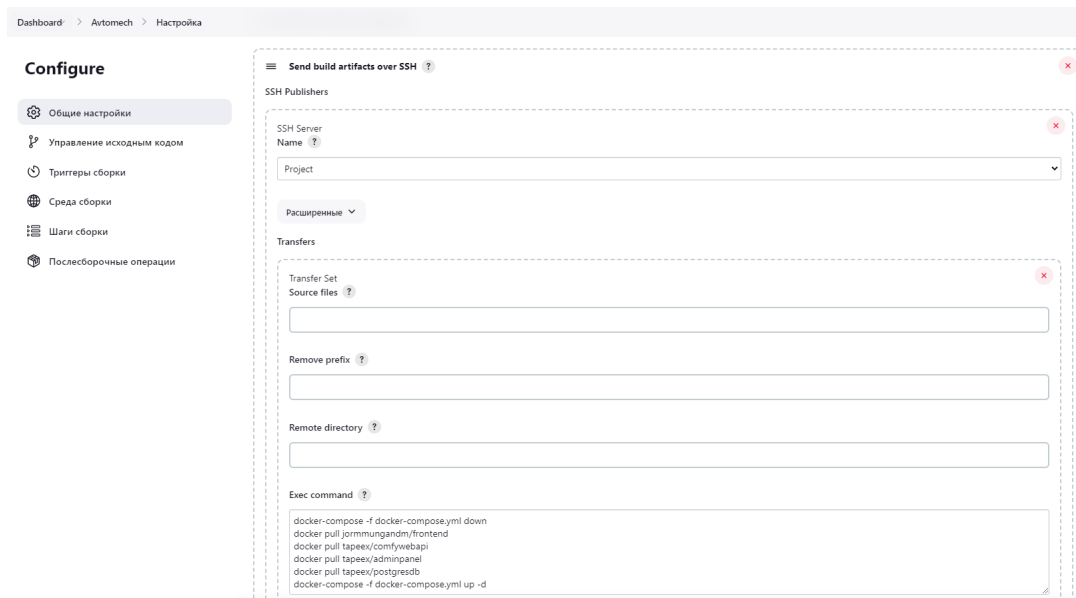
Зобр. Bucket з фотографіями

Для реалізації етапу постійної інтеграція та розгортання(CI/CD), у якому обов'язково участь людини, було обрано сервіс Jenkins. Jenkins має широке співтовариство користувачів і розробників, що робить його однією з найпопулярніших і підтримуваних платформ для CI/CD. Це означає, що є безліч ресурсів, документації, плагінів та підтримки, доступних для користувачів. Jenkins надає потужний набір інструментів та можливостей для налаштування та розширення своєї функціональності. Завдяки величезній кількості плагінів, користувачі можуть інтегрувати Jenkins з різними інструментами та технологіями, щоб задовольнити свої унікальні потреби.



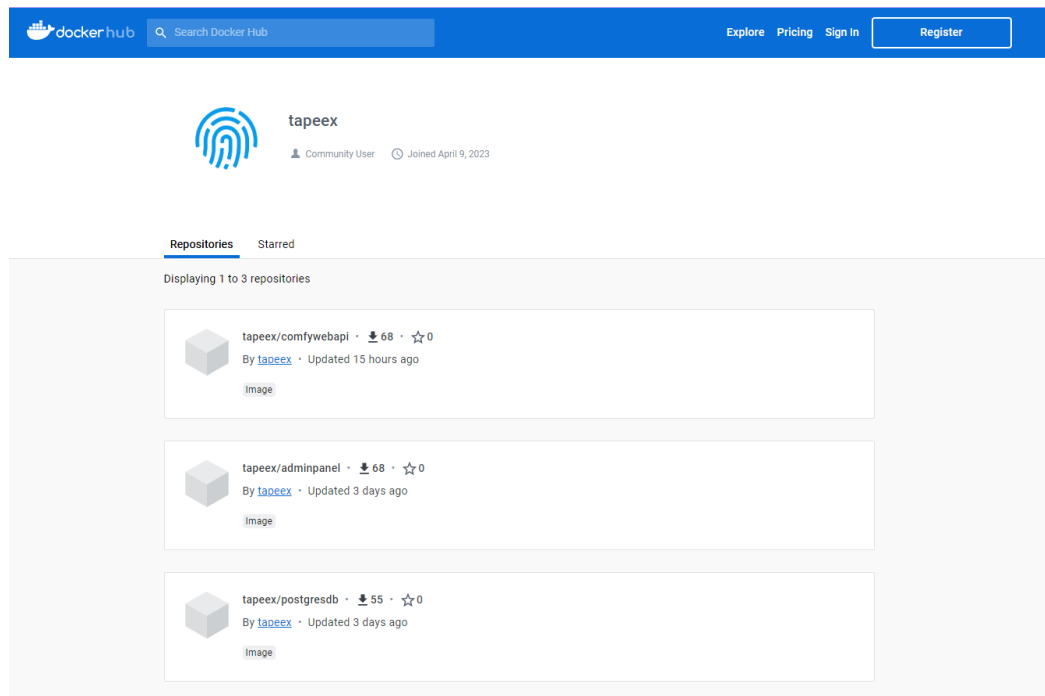
Зобр. Зовнішній вигляд Jenkins

Створено Job для розгортання але потрібно було встановити плагін Publish Over SSH для підключення до AWS EC2.



Зобр. Додатка плагину

Створено було Docker - це популярним інструментом для розгортання, керування та масштабування додатків у середовищах контейнеризації він дозволяє упаковувати додатки та всі їхні залежності в однакові та ізольовані контейнери. Кожен контейнер є незалежним та легковажним оточенням, що містить все необхідне для запуску програми, включаючи код, бібліотеки, залежності та конфігурацію. Docker дозволяє швидко та легко розгорнути програми, запускаючи контейнери з однаковим оточенням на різних серверах або хмарних платформах.



Зобр. Контейнери на DockerHub

ПЗ який розташований на сервері захищений за допомогою проксі Nginx, та налаштованого Certbot з підтвердженим сертифікатом від Let's Encrypt. Проксі Nginx через HTTPS, він перенаправляє запит на контейнер ПЗ, який в повертає відповідь на сервер.

ubuntu@ip-172-31-6-20: ~

```
GNU nano 6.2
server {

    server_name apanel.loffy.pp.ua;

    location / {
        proxy_pass http://127.0.0.1:30080/;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }
    location /images/ {
        root var/www/images/;
    }

    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/apanel.loffy.pp.ua/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/apanel.loffy.pp.ua/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

}

server {
    server_name loffy.pp.ua www.loffy.pp.ua;

    location / {
        proxy_pass http://172.31.93.211/;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }

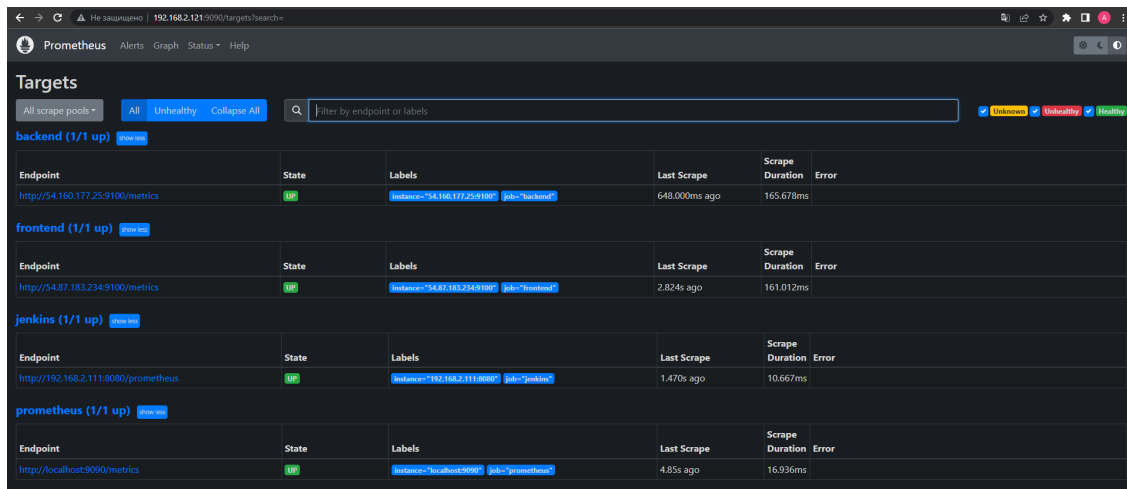
    location /api/ {
        proxy_pass http://127.0.0.1:20080/api/;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Server $host;
        proxy_set_header X-Forwarded-Path $request_uri;
    }

    location /swagger/ {
        proxy_pass http://127.0.0.1:20080/swagger/;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Server $host;
        proxy_set_header X-Forwarded-Path $request_uri;
    }

    location /images/ {
        root var/www/images/;
    }
}
```

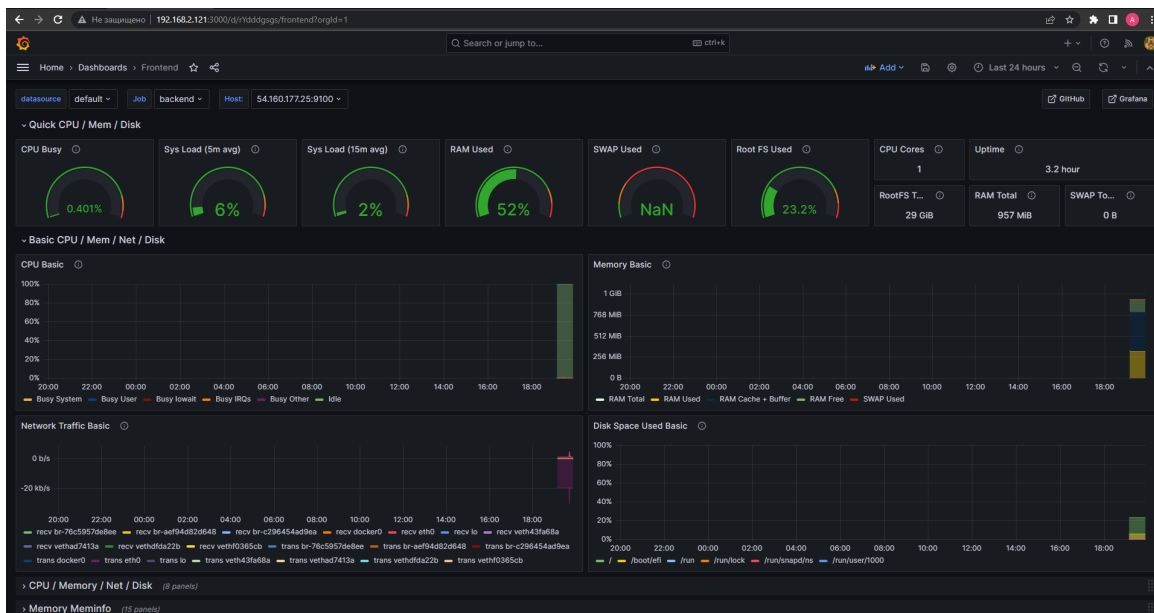
Зобр. Налаштований конф.файл Nginx

Для надання більш детальної про напруження на всі сервів, треба система моніторингу. Використання Prometheus і Grafana разом дозволяє створювати повноцінну систему моніторингу та візуалізації метрик. Prometheus відповідає за збір та збереження метрик, а Grafana надає зручний інтерфейс для їх візуалізації та аналізу. Разом вони дозволяють операторам та розробникам моніторити стан систем, аналізувати динаміку метрик, створювати графіки та панелі для зручного спостереження та аналізу даних.



Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
backend (1/1 up)					
http://54.160.177.25:9100/metrics	UP	instance="54.160.177.25:9100" job="backend"	648.000ms ago	165.678ms	
frontend (1/1 up)					
http://54.87.183.234:9100/metrics	UP	instance="54.87.183.234:9100" job="frontend"	2.824s ago	161.012ms	
jenkins (1/1 up)					
http://192.168.2.111:8080/prometheus	UP	instance="192.168.2.111:8080" job="jenkins"	1.470s ago	10.667ms	
prometheus (1/1 up)					
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus"	4.85s ago	16.936ms	

Зобр. Prometheus метрики



Зобр. Grafana дашбоард системної інформації

ВИСНОВКИ

У рамках даної дипломної роботи було проведено дослідження та аналіз принципів, методологій та інструментів DevOps. Дослідження показало, що DevOps є ефективним та цінним підходом до розробки та операційної підтримки програмного забезпечення, що сприяє покращенню процесів та результатів розробки.

У ході дослідження було розглянуто основні компоненти DevOps, такі як інтеграція та безперервне постачання, інфраструктура як код, контейнеризація, моніторинг та безпека. Були вивчені інструменти, такі як Jenkins, Docker, а також практики, пов'язані з використанням DevOps-підходу.

Результати дослідження показують, що використання DevOps може значно підвищити ефективність та продуктивність розробки програмного забезпечення. Застосування автоматизації процесів за допомогою інструментів, таких як Jenkins дозволяє скоротити час на складання, тестування та розгортання додатків, покращити якість коду та забезпечити більш надійну доставку змін у виробниче середовище.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

№	Ресурс
1.	Docker Розробник: Docker, Inc. Документація
2.	Amazon Web Services Розробник: AWS, Inc. Документація
3.	Nginx Розробник: F5, Inc. Документація
4.	Let's Encrypt, CertBot Розробник: F5, Inc. Документація
5.	Prometheus Розробник: SoundCloud Документація
6.	Grafana Розробник: Grafana Labs Документація