

ЗВІТ

підготувала команда Squad:

Захарін Едуард (ПМО-51)

Лисишин Володимир (ПМІ-51)

Катарина Тарас (ПМО-51)

Полевий Назар (ПМІ-53)

Суть роботи полягає в розробці системи веб-реклами, дослідженні мікросервісної архітектури та хмари сервісів AWS компанії Amazon.

У ході роботи розроблено систему, яка б дозволила одним користувачам розміщувати оголошення про те, якого виду рекламу вони готові надавати, а іншим переглядати вже наявні пропозиції про надання рекламних послуг. Система розроблена з використанням концептів побудови мікросервісних систем, та використання вже готових мікросервісів, які надає хмара сервісів *AWS*.

Можливості користувачів при роботі з даною аплікацією наступні:

- а) користувачі зможуть зареєструватись в системі
- б) тільки зареєстровані користувачі матимуть змогу опублікувати оголошення про надання рекламних послуг
- в) усі користувачі зможуть переглядати оголошення
- г) усі користувачі зможуть здійснювати пошук по оголошеннях
- д) усі користувачі зможуть переглядати деталі оголошення
- е) оголошення міститиме наступні атрибути:
 - 1) заголовок
 - 2) опис
 - 3) ціна
 - 4) шлях до зображення

Технології:

- а) система побудована на основі мікросервісної архітектури;
- б) система побудована з використанням платформи .NET CORE та фреймворку ASP.NET CORE з використанням мови C#;
- в) система використовує засоби хмари AWS, де це можливо і потрібно;
- г) система розгорнута у хмарі AWS.

У ході розробки використано наступні сервіси хмари AWS:

а) Amazon Cognito – дозволяє швидко і просто додавати можливості реєстрації, авторизації і контролю доступу користувачів в мобільні та інтернет-додатки.

б) Amazon S3 - онлайнова веб-служба, пропонована Amazon Web Services, що надає можливість для зберігання і отримання будь-якого обсягу даних, в будь-який час з будь-якої точки мережі, так званий файловий хостинг;

в) Amazon CloudFront - веб-сервіс для доставки контенту. CloudFront інтегрується з іншими Amazon Web Services. Мета сервісу - дати розробникам і підприємствам простий спосіб поширювати контент для кінцевих користувачів з мінімальними затримками і високою швидкістю передачі даних;

г) Amazon DynamoDB - система управління базами даних класу NoSQL в форматі «ключ-значення», пропонована Amazon як частина AWS. Поставляється тільки як служба хмари Amazon;

д) Amazon CloudWatch - це сервіс моніторингу і спостереження, який надає дані і аналітичні відомості для моніторингу додатків, реагування на зміни продуктивності в масштабах системи, оптимізації використання ресурсів та отримання єдиного уявлення про працездатність системи;

е) Amazon Simple Notification Service - сервіс посилки повідомлень між компонентами AWS і зовнішніми компонентами, забезпечує масову доставку повідомлень в складній мережі;

є) Amazon Lambda — безсерверна обчислювальна платформа керована подіями. Вона надається Amazon як одна з веб-служб. Це обчислювальна служба, яка запускає код у відповідь на події і автоматично керує обчислювальними ресурсами, необхідними для виконання цього коду;

ж) Amazon Elasticsearch Service - це повністю керований сервіс, який дозволяє просто і безпечно розгортати і використовувати Elasticsearch в будь-якому масштабі.



У ході роботи побудовані наступні сервіси:

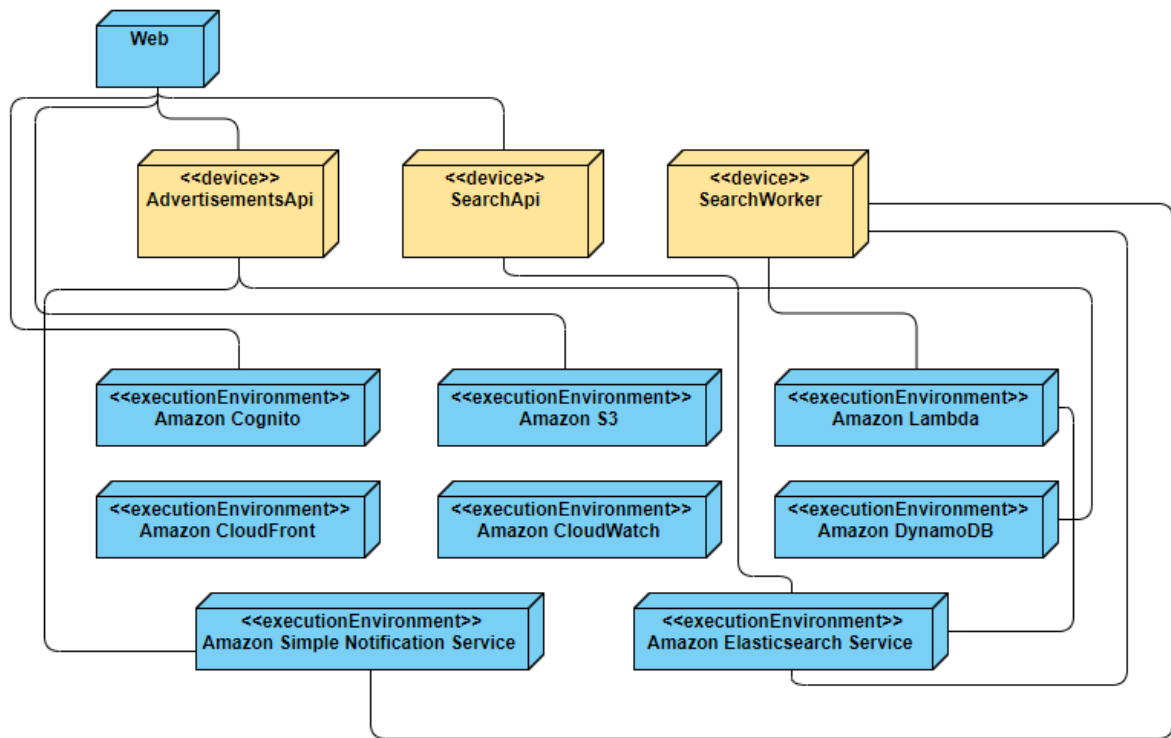
а) WebAdvertisements.AdvertisementsApi – мікросервіс на основі фреймворку Asp.net Core Web API, відповідає за здійснення операцій, що стосуються сутності ‘Advertisement’. Зберігає дані у NoSQL базу даних DynamoDb. З його допомогою можна створювати об’єкти, модифікувати та витягувати. При успішній операції створення об’єкту ‘Advertisement’ надсилає повідомлення в топик Amazon SNS;

б) WebAdvertisements.SearchWorker – проект на основі .Net Core class library, містить код, який Amazon Lambda запускає у відповідь на публікацію повідомлення в спеціальний топик SNS та записує дані в індекс пошукової системи Amazon Elasticsearch Service;

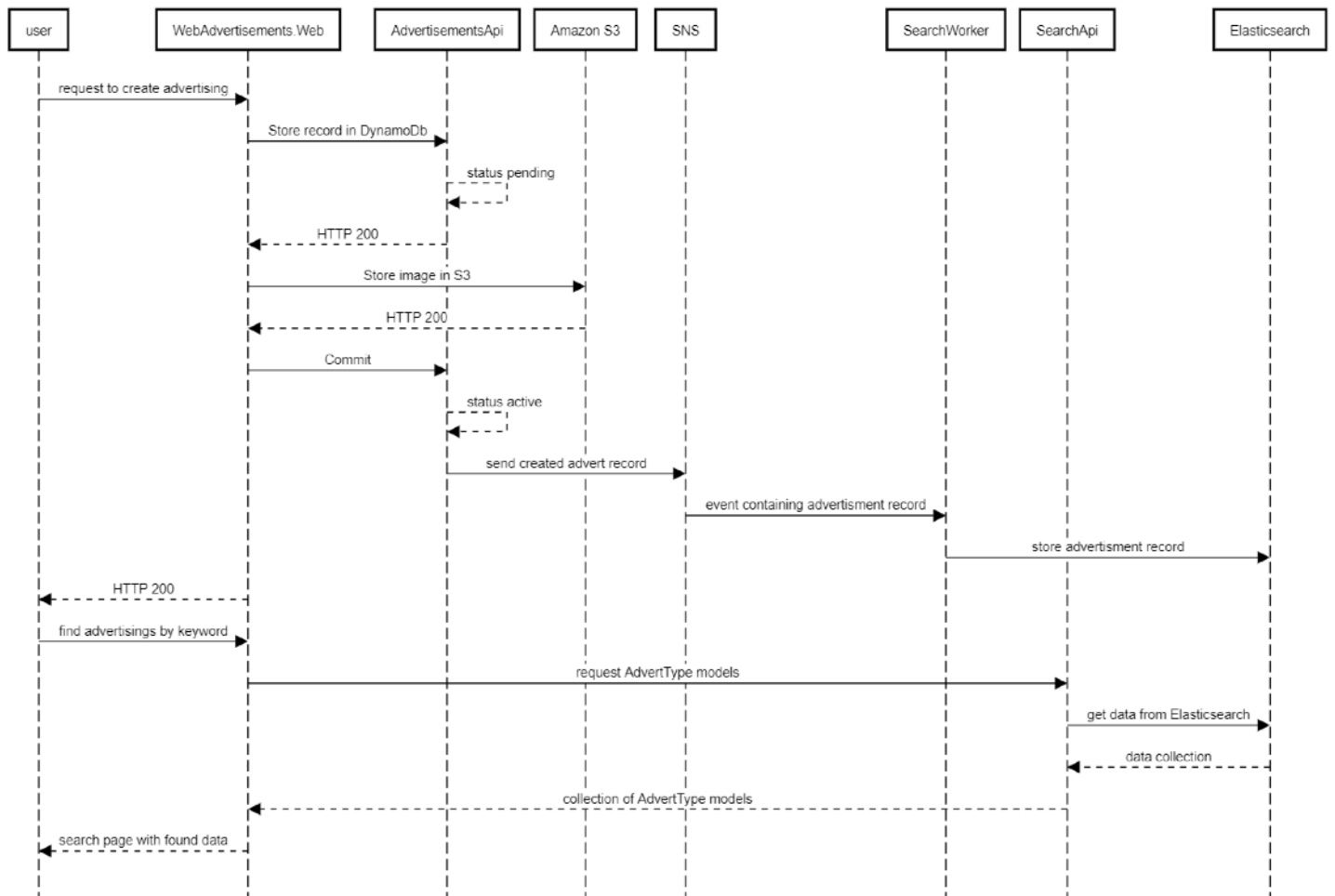
в) WebAdvertisements.SearchApi – мікросервіс на основі фреймворку Asp.net Core Web API, використовується для пошуку оголошень. Витягує дані з індексу Amazon Elasticsearch Service;

г) WebAdvertisements.Web – веб-аплікація на основі фреймворку Asp.net Core MVC. Використовується кінцевими користувачами для реєстрації, автентифікації, створенням оголошень та публікації зображень для них, перегляду оголошень та пошуку.

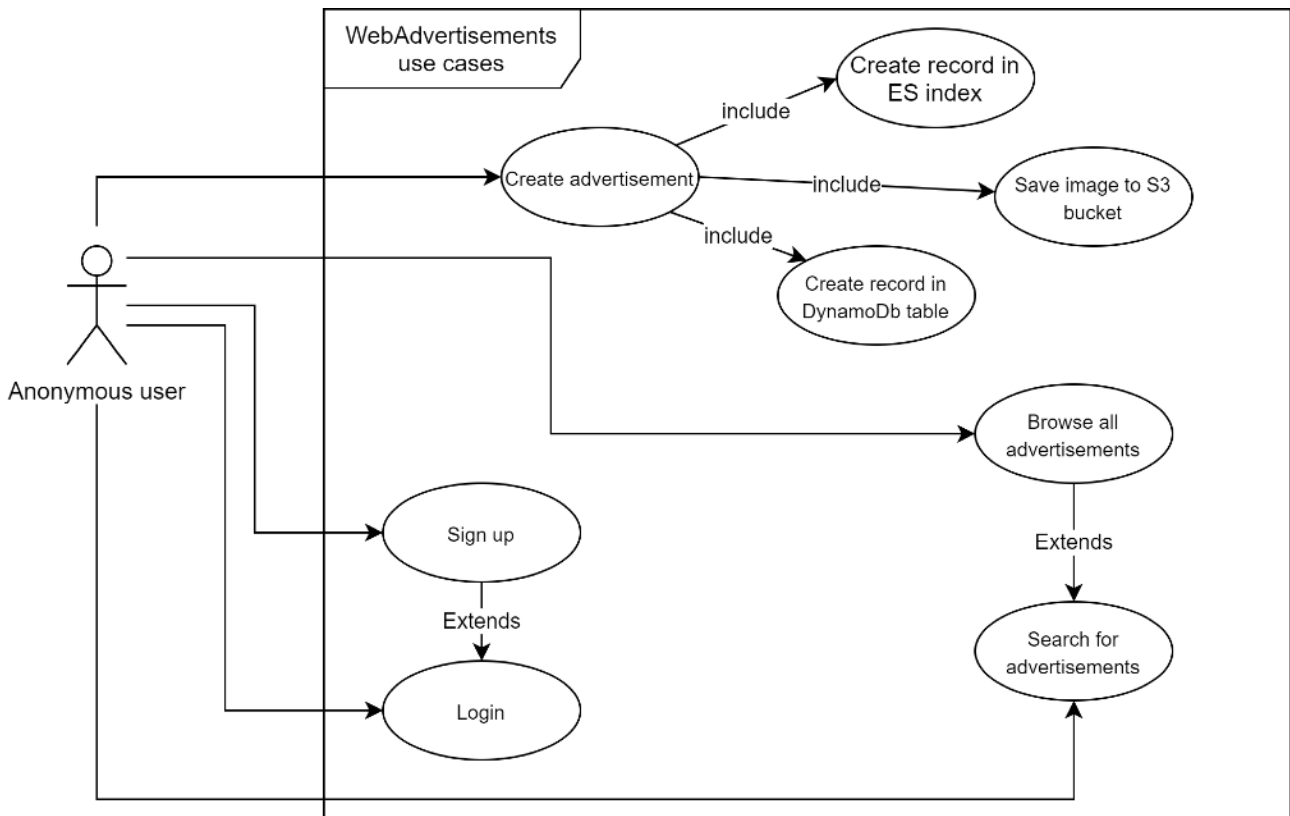
Deployment diagram



Sequence diagram



Use case diagram



Патерни конкурентного програмування:

а) Thread pool — підтримує кілька потоків, які очікують на призначення завдань для одночасного виконання програмою. Підтримуючи пул потоків, модель підвищує продуктивність і уникає затримок у виконанні через часте створення та знищення потоків для короточасних завдань. Цей механізм реалізовано в .NET і роботу з ним забезпечує `System.Threading.ThreadPool`.

Дана система веб-реклами реалізована з використанням об'єктів `Task`, які виконують в потоках пула асинхронні задачі.

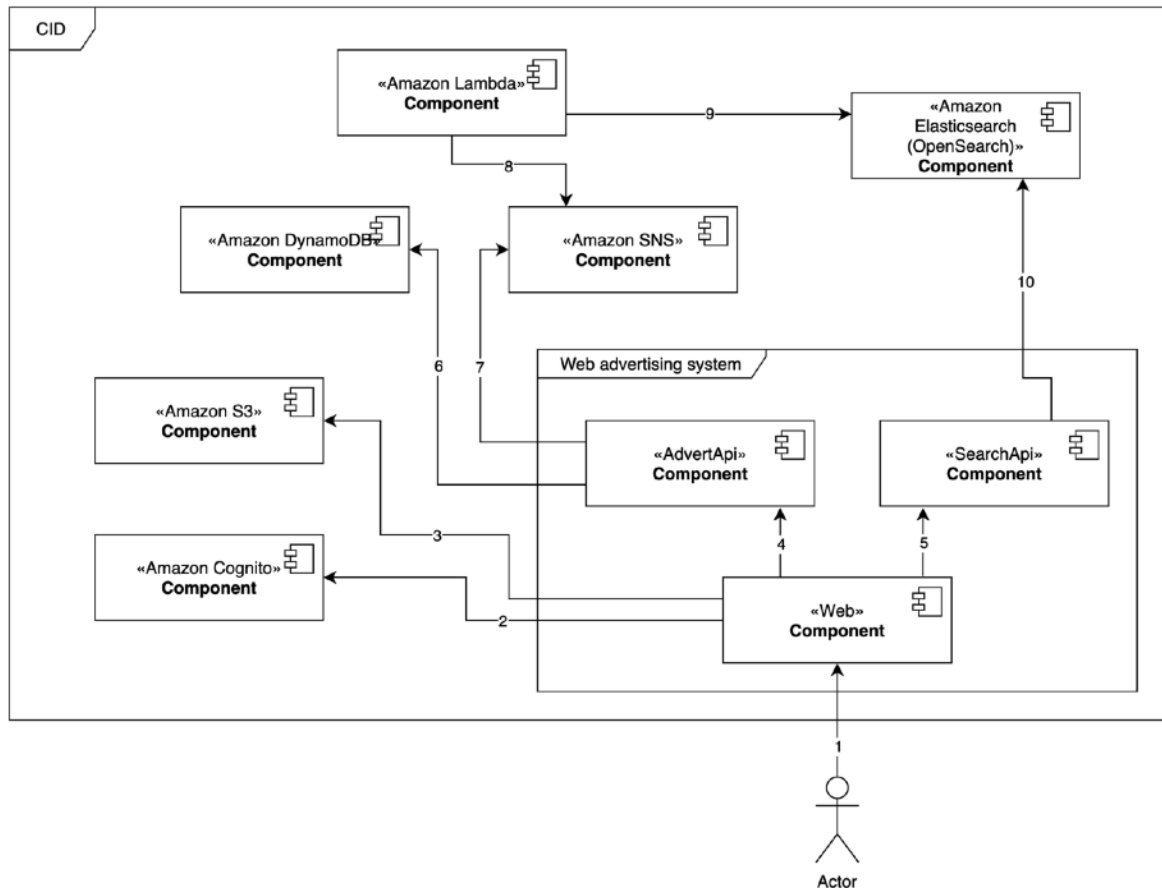
б) Double-checked locking — зазвичай використовується для зменшення витрат на блокування під час реалізації «лінивої ініціалізації» в багатопотоковому середовищі, особливо як частина патерну `Singleton`. Це дозволяє уникнути ініціалізації значення до першого доступу до нього.

`IElasticClient` було реалізовано використовуючи даний підхід.

Resiliency Model

1) Pre-work

Component Interaction Diagram



RMA workbook

ID	Interaction	Interaction description
1	User — Web application	All users interact with Web UI application.
2	Web — Amazon Cognito	Web app connects with Cognito to store and retrieve user data & auth
3	Web — Amazon S3	Web app connects with S3 for advertisement files store.
4	Web — AdvertApi	Web app connects with AdvertApi for advertisements creation.
5	Web — SearchApi	Web app connects with SearchApi to find ads.
6	AdvertApi — DynamoDB	AdvertApi component connects with DynamoDB to store ads info.
7	AdvertApi — SNS	AdvertApi publishes events in SNS as they are created

8	Lambda — SNS	Runs code execution for new events published into SNS
9	Lambda — Elasticsearch	Index new ad records into Elasticsearch
10	SeachApi — Elasticsearch	During a new ad search, searches are performed in Elasticsearch

2) Discover

ID	Interaction ID	Interaction	Failure Short Name	Failure Description	Response
1	1	User — Web application	Latency: too slow	Due to instance scaling container is shut down when not in use.	Wait...
2	2	Web — Amazon Cognito	Incorrectness: Invalid data	User entered to weak password, existing or invalid email.	User is notified and gets a new try.
3	2	Web — Amazon Cognito	Bad Auth: Authentication	Incorrect credentials are entered.	User is notified with an appropriate message and gets a new try.
4	2	Web — Amazon Cognito	Bad Auth: Authentication	Incorrect credentials are entered too many times.	User account should be blocked for security reasons, and user should be notified.
5	4	Web — AdvertApi	Incorrectness: Invalid data	User didn't populated required fields.	User is notified and forced to enter proper data

3) Rate

Discover Id	Effects	Portion affected	Detection	Resolution	Likelihood	Risk
1	Minor impairment	Less than 2%	Less than 5 min	Less than 5 min	More than once a month	3/5
2	Minor impairment	More that 50%	Less than 5 min	Less than 5 min	More than once a month	1/5
3	Minor impairment	More that 50%	Less than 5 min	Less than 5 min	More than once a month	1/5
4	Major impairment	100%	Less than 5 min	Between 5 min and 45 min	More than once a month	5/5
5	No noticeable effect	More that 50%	Less than 5 min	Less than 5 min	More than once a month	1/5