



Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота №3
Тенденції розвитку інформаційних систем та технологій
Централізовані системи логування. EFK.

Виконав
студент групи IT-41ф

Новиков Д. М.

Перевірив:

ас. Цимбал С. І.

Мета роботи: ознайомлення із централізованими системами логування на прикладі EFK.

Хід роботи:

1. Запустити тестовий EFK стек

Оскільки я буду писати Web API на ASP.NET Core, провайдер Serilog взаємодіє з Elasticsearch напряму. Таким чином, зі стеку EFK (Elasticsearch, Fluentd, Kibana) мені не потрібен Fluentd. Його роль збирача та форматувача логів виконує спеціальний компонент Serilog - Sink. Підготуємо Docker-compose.yml для створення ELK:

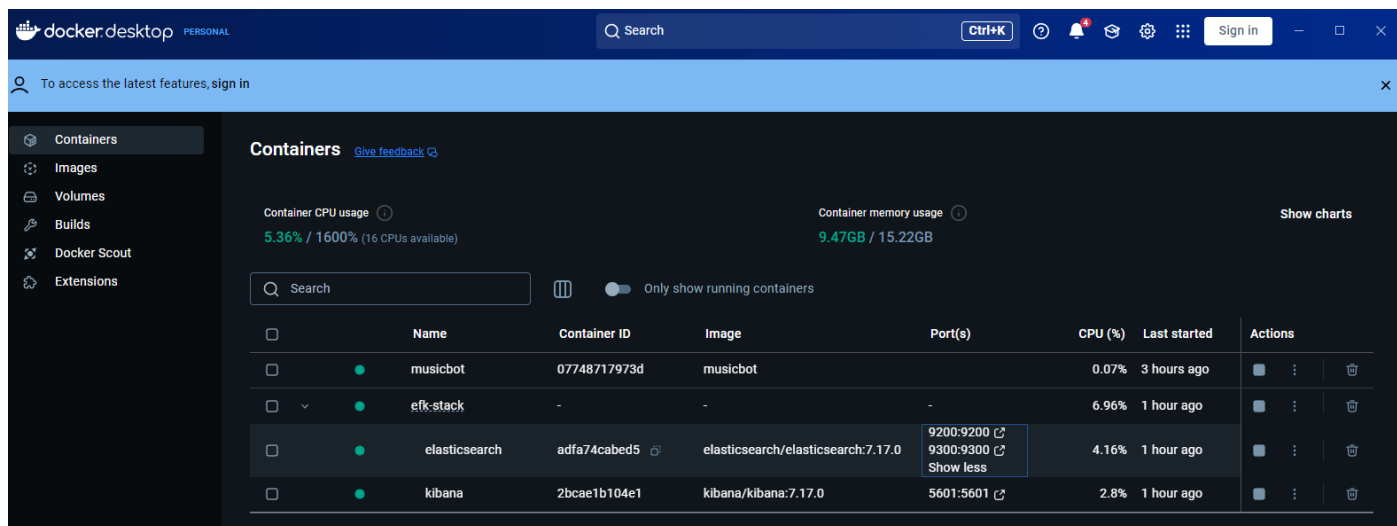
```
services:
  elasticsearch:
    image: docker.elastic.co/elasticsearch/elasticsearch:7.17.0
    container_name: elasticsearch
    environment:
      discovery.type: single-node
      xpack.monitoring.enabled: true
      xpack.watcher.enabled: false
    ports:
      - 9200:9200
      - 9300:9300
    volumes: # Stores elasticsearch data locally on the es_data Docker volume
      - es_data:/usr/share/elasticsearch/data

  kibana:
    image: docker.elastic.co/kibana/kibana:7.17.0
    container_name: kibana
    environment:
      ELASTICSEARCH_URL: http://elasticsearch:9200
    ports:
      - 5601:5601
    depends_on:
      - elasticsearch

volumes:
  es_data:
```

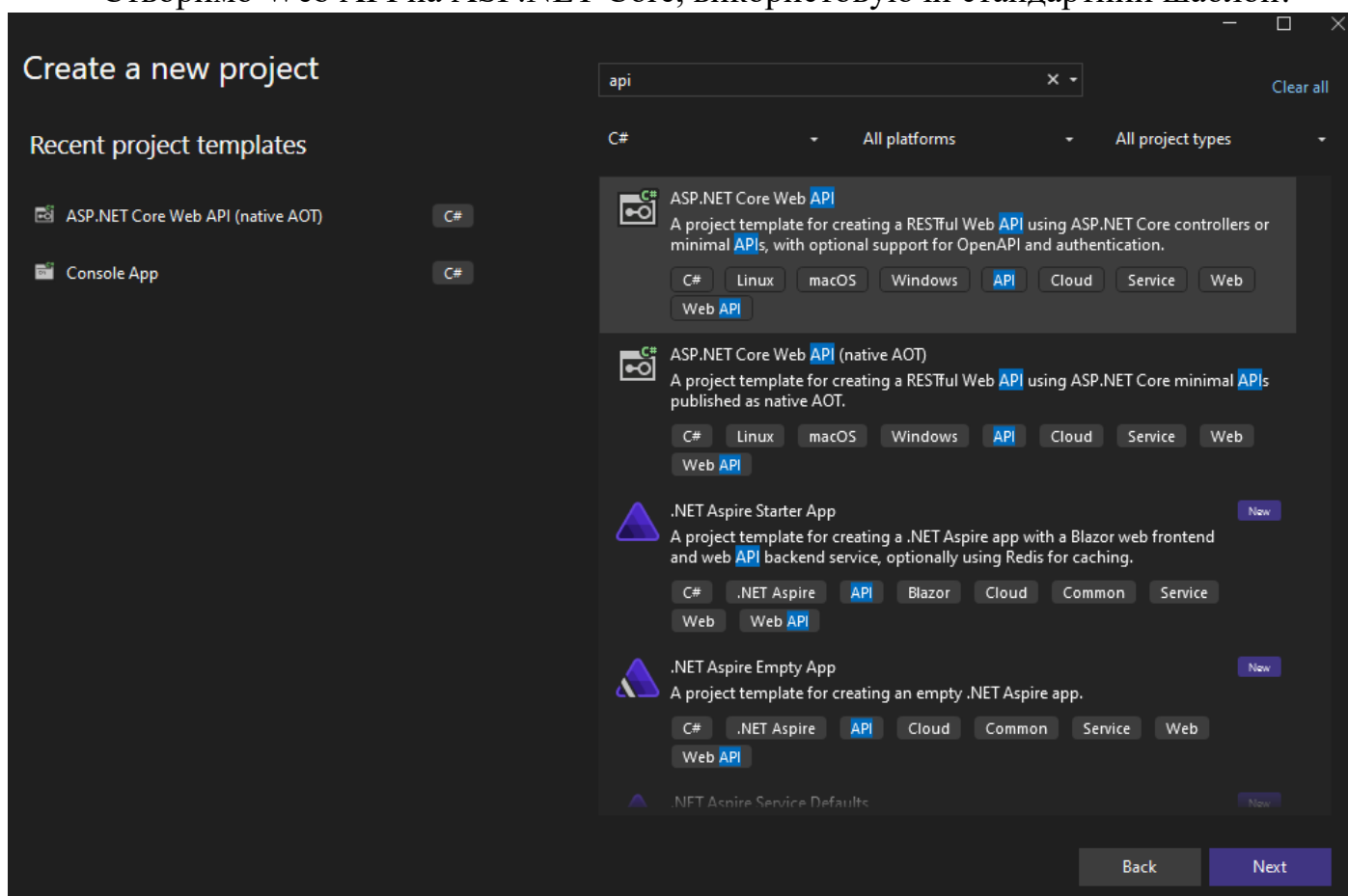
Запустимо стек за допомогою команди docker-compose up:

```
PS E:\Univer\Sharaga 13 Term\TRIST\LR3\EFK-stack> docker-compose up -d
[+] Running 3/3
✓ Network efk-stack_default Created
✓ Container elasticsearch Started
✓ Container kibana Started
```



2. Розробити найпростіший застосунок, який буде записувати логи, щоб їх можна було побачити, фільтрувати у Kibana.

Створимо Web API на ASP.NET Core, використовуючи стандартний шаблон:



Additional information

ASP.NET Core Web API C# Linux macOS Windows API Cloud Service Web Web API

Framework ⓘ
.NET 8.0 (Long Term Support)

Authentication type ⓘ
None

☐ Configure for HTTPS ⓘ
☒ Enable container support ⓘ

Container OS ⓘ
Linux

Container build type ⓘ
Dockerfile

☒ Enable OpenAPI support ⓘ
☒ Do not use top-level statements ⓘ
☒ Use controllers ⓘ
☐ Enlist in .NET Aspire orchestration ⓘ

Aspire version ⓘ
9.0

Microsoft Visual Studio
Creating project...

Cancel

Back Create

а) Додамо необхідні залежності:

- Serilog;
- Serilog.AspNetCore;
- Serilog.Sinks.Console;
- Serilog.Sinks.Elasticsearch.

б) У файлі Program.cs налаштуємо конфігурацію для Serilog, а також додамо кілька логів:

```
public class Program
{
    public static void Main(string[] args)
    {
        // Configure Serilog using appsettings.json.
        Log.Logger = new LoggerConfiguration()
            .ReadFrom.Configuration(new ConfigurationBuilder()
                .AddJsonFile("appsettings.json", optional: false, reloadOnChange: true)
                .Build())
            .CreateLogger();

        try
        {
            Log.Information("Starting the application.");

            var builder = WebApplication.CreateBuilder(args);

            ...

            app.Run();
        }
        catch (Exception ex)
        {
            Log.Fatal(ex, "Application startup failed.");
        }
        finally
        {
            Log.CloseAndFlush();
        }
    }
}
```

- в) Модифікуємо існуючий контролер WeatherForecastController, додавши логування на початку виклику методу та перед поверненням результату:

```
[HttpGet(Name = "GetWeatherForecast")]
0 references
public IEnumerable<WeatherForecast> Get()
{
    _logger.LogInformation("Weather forecast requested.");

    var forecasts = Enumerable.Range(1, 5).Select(index => new WeatherForecast
    {
        Date = DateOnly.FromDateTime(DateTime.Now.AddDays(index)),
        TemperatureC = Random.Shared.Next(-20, 55),
        Summary = Summaries[Random.Shared.Next(Summaries.Length)]
    })
    .ToArray();

    _logger.LogInformation("Returning {Count} forecasts.", forecasts.Length);
    return forecasts;
}
```

- г) Створимо додатковий контролер для тестування:

```
namespace EFKLoggingApi.Controllers
{
    [ApiController]
    [Route("[controller]")]
    3 references
    public class MaaaaahController : ControllerBase
    {
        private readonly ILogger<MaaaaahController> _logger;
        0 references
        public MaaaaahController(ILogger<MaaaaahController> logger)
        {
            _logger = logger;
        }

        [HttpGet("GetRandomMaaaaahValue", Name = "GetRandomMaaaaahValue")]
        0 references
        public int GetRandomMaaaaahValue()
        {
            _logger.LogInformation("GetRandomMaaaaahValue requested.");

            var randomValue = new Random().Next(0, 100);
            _logger.LogInformation("Maaaaaaah value is [{RandomMaaaaahValue}].", randomValue);

            return randomValue;
        }

        [HttpGet("ThrowErrorMaaaaaahMessage/{id}", Name = "ThrowErrorMaaaaaahMessage")]
        0 references
        public string ThrowErrorMaaaaaahMessage(int id)
        {
            _logger.LogInformation("ThrowErrorMaaaaaahMessage requested.");

            try
            {
                if (id <= 0)
                {
                    throw new Exception($"id cannot be less than or equal to 0. Value passed is [{id}].");
                }

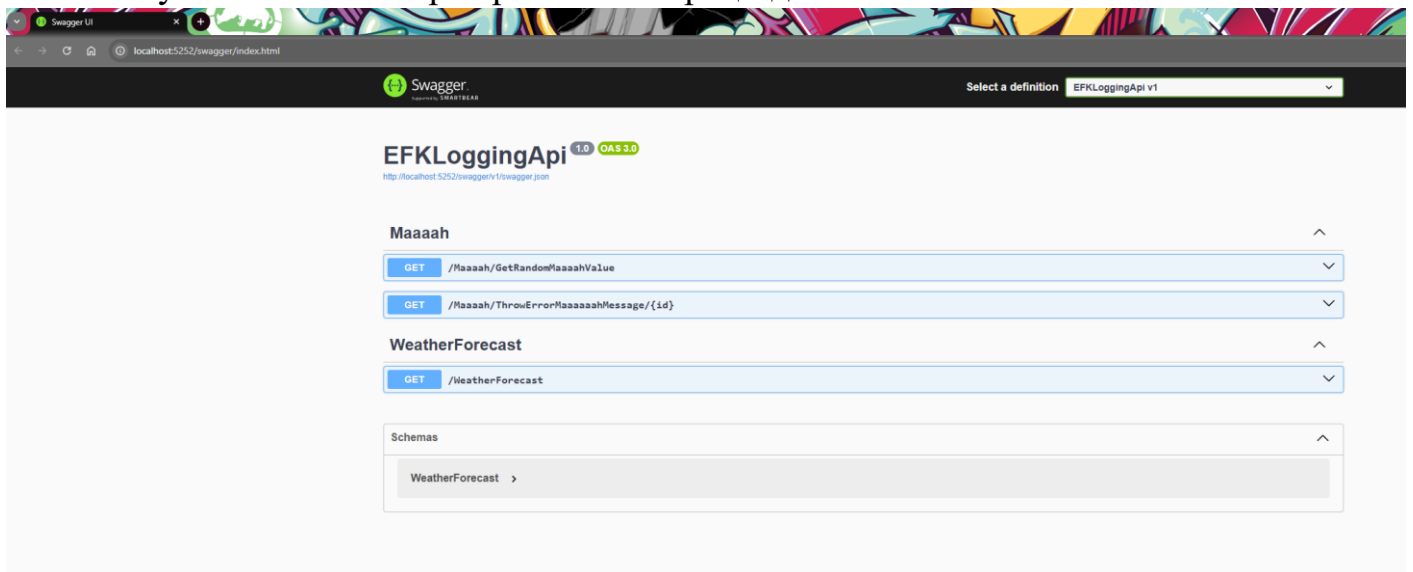
                return id.ToString();
            }
            catch (Exception ex)
            {
                _logger.LogError(ex, ex.Message);
            }

            return string.Empty;
        }
    }
}
```

- д) Модифікуємо конфігураційний файл appsettings.json, щоб налаштувати Serilog. Вкажемо формат індексу, адресу (endpoint) Elasticsearch і додаткові поля:

```
libman.json  appsettings.json  NuGet: EFKLoggingApi  WeatherForecastController.cs  Program.cs
schema: https://json.schemastore.org/appsettings.json
1  {
2    --"AllowedHosts": "*",
3    --"Serilog": {
4      --"Using": [ "Serilog.Sinks.Elasticsearch", "Serilog.Sinks.Console" ],
5      --"MinimumLevel": "Information",
6      --"WriteTo": [
7        {
8          --"Name": "Console"
9        },
10       {
11         --"Name": "Elasticsearch",
12         --"Args": {
13           --"nodeUri": "http://localhost:9200", //Elasticsearch endpoint
14           --"indexFormat": "EFKLoggingApi-{0:yyyy.MM.dd}", //Log index format
15           --"templateName": "ecs-template", //Template name for Elasticsearch
16           --"autoRegisterTemplate": true //Automatically register ECS-compatible index template
17         }
18       },
19     ],
20     --"Enrich": [ "FromLogContext", "WithMachineName", "WithThreadId" ],
21     --"Properties": {
22       --"Application": "EFKLoggingApi"
23     }
24   }
25 }
26 }
```

Запустимо API та перевіримо його працездатність:



Maaaah

GET

/Maaaah/GetRandomMaaaahValue

Parameters

No parameters

ExecuteClear

Responses

Curl

curl -X 'GET' \n'http://localhost:5252/Maaaah/GetRandomMaaaahValue' \n-H 'accept: text/plain'

Request URL

http://localhost:5252/Maaaah/GetRandomMaaaahValue

Server response

CodeDetails

200

Response body

13

Response headers

content-type: application/json; charset=utf-8\ndate: Sun,01 Dec 2024 19:52:09 GMT\nserver: Kestrel\ntransfer-encoding: chunked

Responses

Code	Description	Links
200	OK	No links

Media type

text/plain

Controls Accept header

Example Value | Schema

0

WeatherForecast

GET

/WeatherForecast

Parameters

No parameters

ExecuteClear

Responses

Curl

curl -X 'GET' \n'http://localhost:5252/WeatherForecast' \n-H 'accept: text/plain'

Request URL

http://localhost:5252/WeatherForecast

Server response

CodeDetails

200

Response body

[{\n \"date\": \"2024-12-02\", \n \"temperatureC\": -7, \n \"temperatureF\": 19, \n \"summary\": \"Chilly\" \n}, {\n \"date\": \"2024-12-03\", \n \"temperatureC\": 20, \n \"temperatureF\": 68, \n \"summary\": \"Scorching\" \n}, {\n \"date\": \"2024-12-04\", \n \"temperatureC\": 10, \n \"temperatureF\": 50, \n \"summary\": \"Chilly\" \n}, {\n \"date\": \"2024-12-05\", \n \"temperatureC\": -8, \n \"temperatureF\": 18, \n \"summary\": \"Cool\" \n}, {\n \"date\": \"2024-12-06\", \n \"temperatureC\": -1, \n \"temperatureF\": 30, \n \"summary\": \"Chilly\" \n}]\n

Response headers

content-type: application/json; charset=utf-8\ndate: Sun,01 Dec 2024 19:52:25 GMT\nserver: Kestrel\ntransfer-encoding: chunked

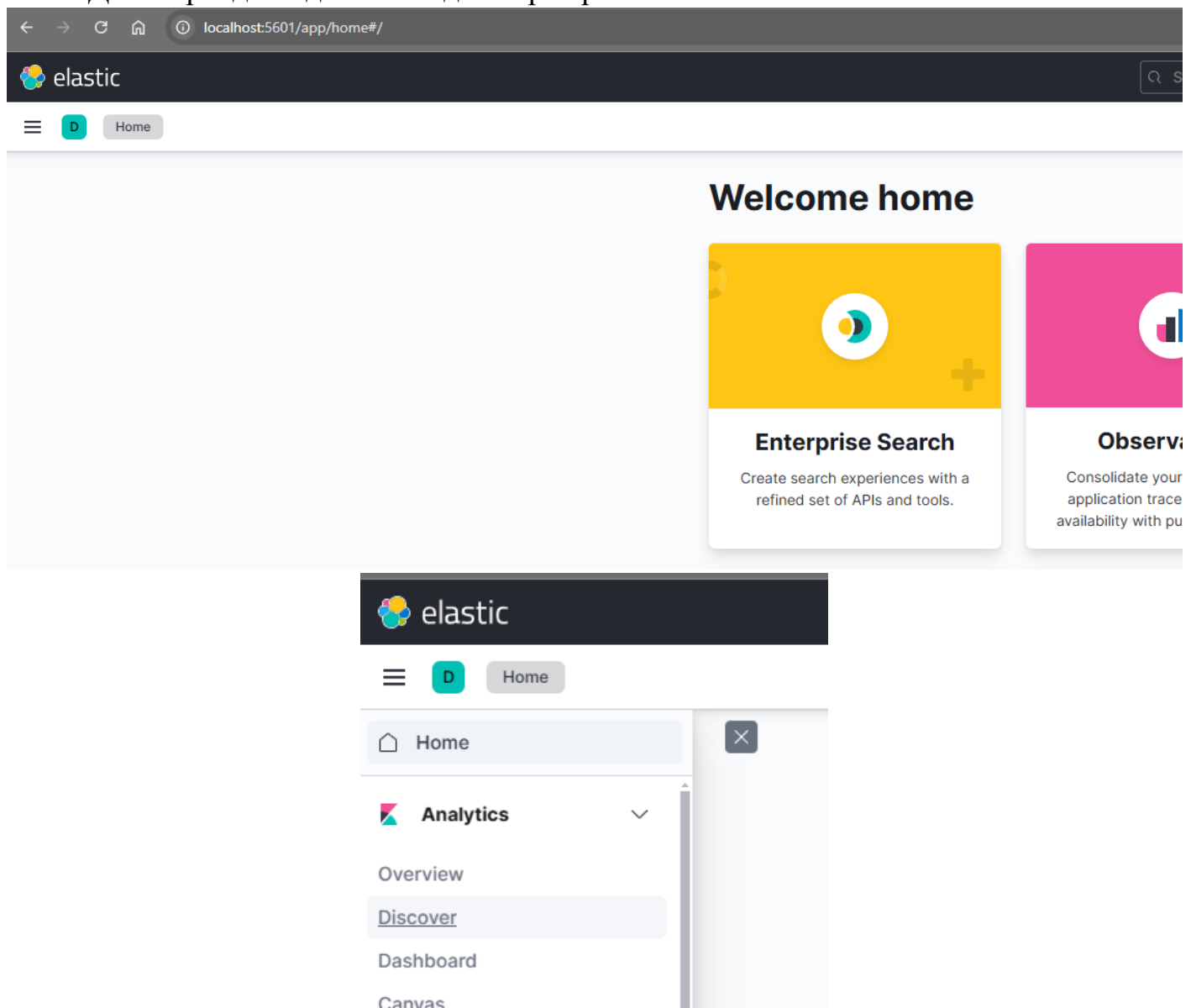
Responses

```

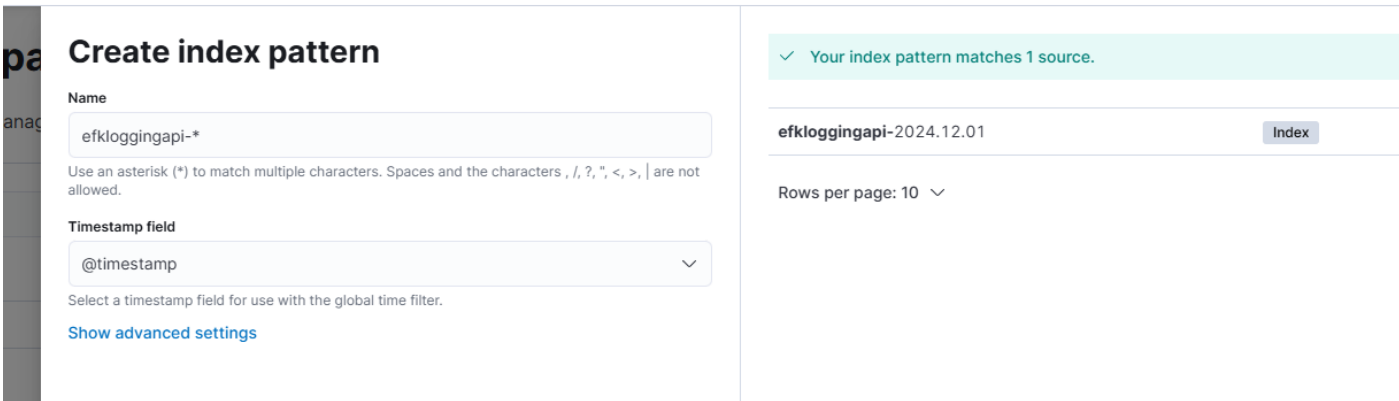
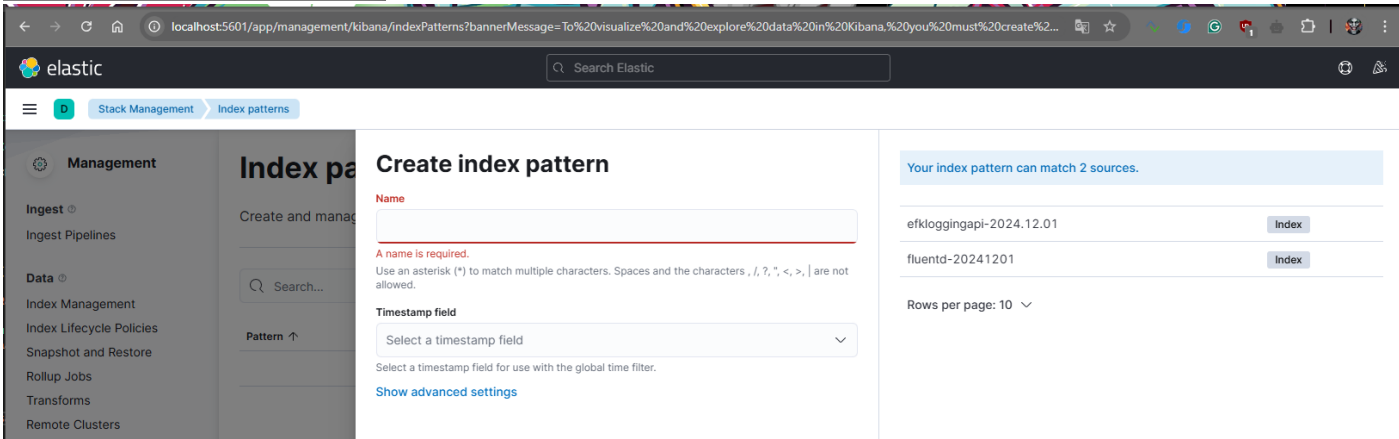
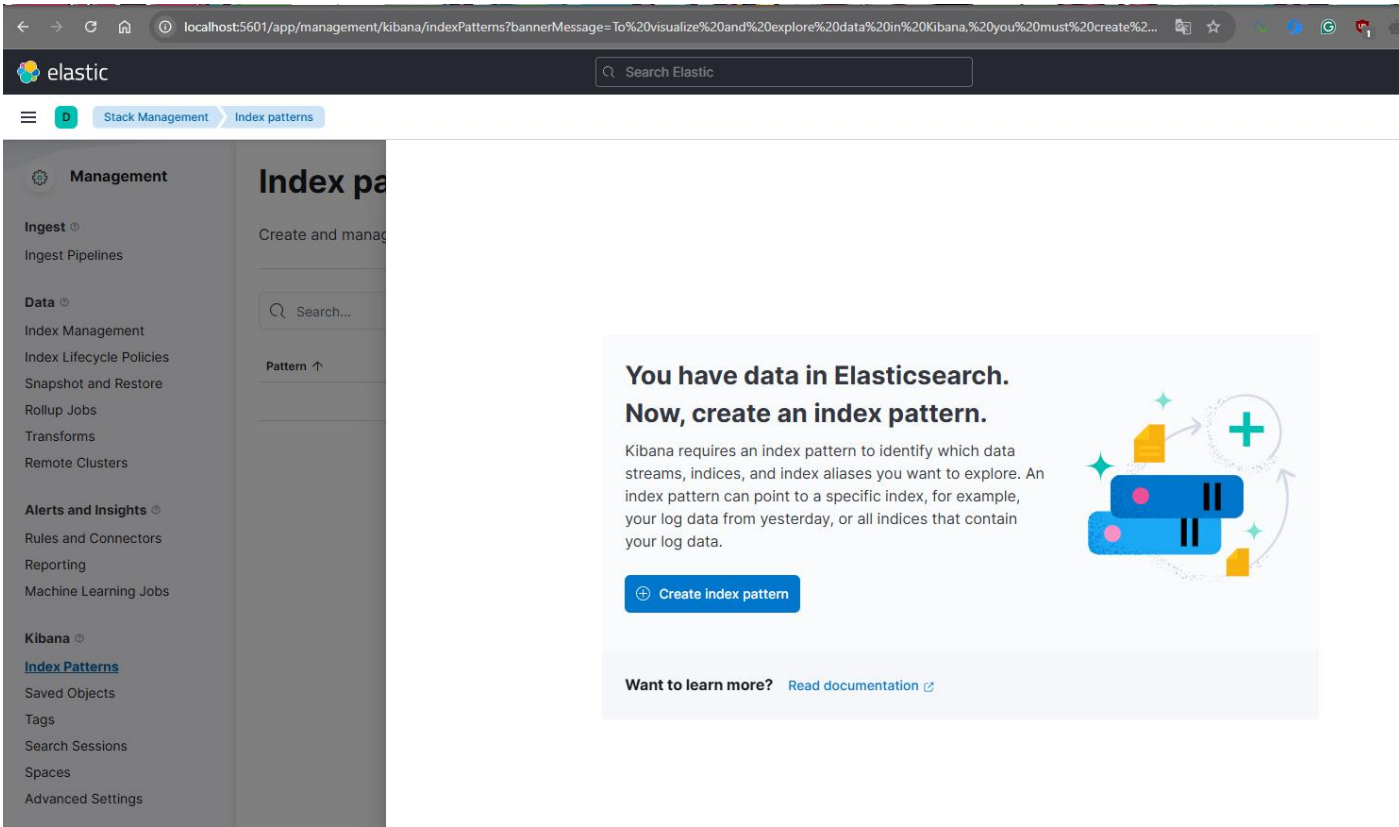
21:52:17 INF] Executed endpoint 'EFKLoggingApi.Controllers.MaaaaahController.ThrowErrorMaaaaaaahMessage (EFKLoggingApi)'
21:52:17 INF] Request finished HTTP/1.1 GET http://localhost:5252/Maaaaah/ThrowErrorMaaaaaaahMessage/5 - 200 null text/pl
ain; charset=utf-8 21.5716ms
21:52:20 INF] Request starting HTTP/1.1 GET http://localhost:5252/Maaaaah/ThrowErrorMaaaaaaahMessage/-2 - null null
21:52:20 INF] Executing endpoint 'EFKLoggingApi.Controllers.MaaaaahController.ThrowErrorMaaaaaaahMessage (EFKLoggingApi)'
21:52:20 INF] Route matched with {action = "ThrowErrorMaaaaaaahMessage", controller = "Maaaaah"}. Executing controller ac
tion with signature System.String ThrowErrorMaaaaaaahMessage(Int32) on controller EFKLoggingApi.Controllers.MaaaaahControl
ler (EFKLoggingApi).
21:52:20 INF] ThrowErrorMaaaaaaahMessage requested.
21:52:20 ERR] id cannot be less than or equal to 0. Value passed is [-2].
System.Exception: id cannot be less than or equal to 0. Value passed is [-2].
   at EFKLoggingApi.Controllers.MaaaaahController.ThrowErrorMaaaaaaahMessage(Int32 id) in E:\Univer\Sharaga 13 Term\TRIST\
LR3\EFKLoggingApi\Controllers\MaaaaahController.cs:line 34
21:52:20 INF] Executing ObjectResult, writing value of type 'System.String'.
21:52:20 INF] Executed action EFKLoggingApi.Controllers.MaaaaahController.ThrowErrorMaaaaaaahMessage (EFKLoggingApi) in 9
1.9511ms
21:52:20 INF] Executed endpoint 'EFKLoggingApi.Controllers.MaaaaahController.ThrowErrorMaaaaaaahMessage (EFKLoggingApi)'
21:52:20 INF] Request finished HTTP/1.1 GET http://localhost:5252/Maaaaah/ThrowErrorMaaaaaaahMessage/-2 - 200 0 text/plai
n; charset=utf-8 101.1145ms
21:52:23 INF] Request starting HTTP/1.1 GET http://localhost:5252/WeatherForecast - null null
21:52:23 INF] Executing endpoint 'EFKLoggingApi.Controllers.WeatherForecastController.Get (EFKLoggingApi)'
21:52:23 INF] Route matched with {action = "Get", controller = "WeatherForecast"}. Executing controller action with sig
nature System.Collections.Generic.IEnumerable`1[EFKLoggingApi.WeatherForecast] Get() on controller EFKLoggingApi.Control
lers.WeatherForecastController (EFKLoggingApi).
21:52:25 INF] Weather forecast requested.
21:52:25 INF] Returning 5 forecasts.
21:52:25 INF] Executing ObjectResult, writing value of type 'EFKLoggingApi.WeatherForecast[]'.
21:52:25 INF] Executed action EFKLoggingApi.Controllers.WeatherForecastController.Get (EFKLoggingApi) in 1522.0483ms
21:52:25 INF] Executed endpoint 'EFKLoggingApi.Controllers.WeatherForecastController.Get (EFKLoggingApi)'
21:52:25 INF] Request finished HTTP/1.1 GET http://localhost:5252/WeatherForecast - 200 null application/json; charset=
utf-8 1530.5107ms

```

Далі перейдемо до Kibana для перевірки логів:



а) Створимо шаблон індексу для даних з нашого API:



elastic

Stack Management Index patterns efkloggingapi-*

Management

Ingest Ingest Pipelines

Data

- Index Management
- Index Lifecycle Policies
- Snapshot and Restore
- Rollup Jobs
- Transforms
- Remote Clusters

Alerts and Insights

- Rules and Connectors
- Reporting
- Machine Learning Jobs

Kibana

Index Patterns

- Saved Objects
- Tags
- Search Sessions
- Spaces
- Advanced Settings

Stack

- License Management
- Upgrade Assistant

efkloggingapi-*

Time field: @timestamp

View and edit fields in efkloggingapi-*. Field attributes, such as type and searchability, are based on [field mappings](#) in Elasticsearch.

Fields (83) Scripted fields (0) Field filters (0)

Search

All field types Add field

Name	Type	Format	Searchable	Aggregatable	Excluded
@timestamp	date				
_id	_id				
_index	_index				
_score					
_source	_source				
_type	_type				
exceptions.ClassName	text				
exceptions.ClassName.keyword	keyword				
exceptions.Depth	long				
exceptions.HResult	long				

Rows per page: 10

б) Перейдемо до розділу Discovery і перевіримо наші логи:

elastic

Discover

Search

Filter by type

Selected fields

Available fields

Popular

283 hits

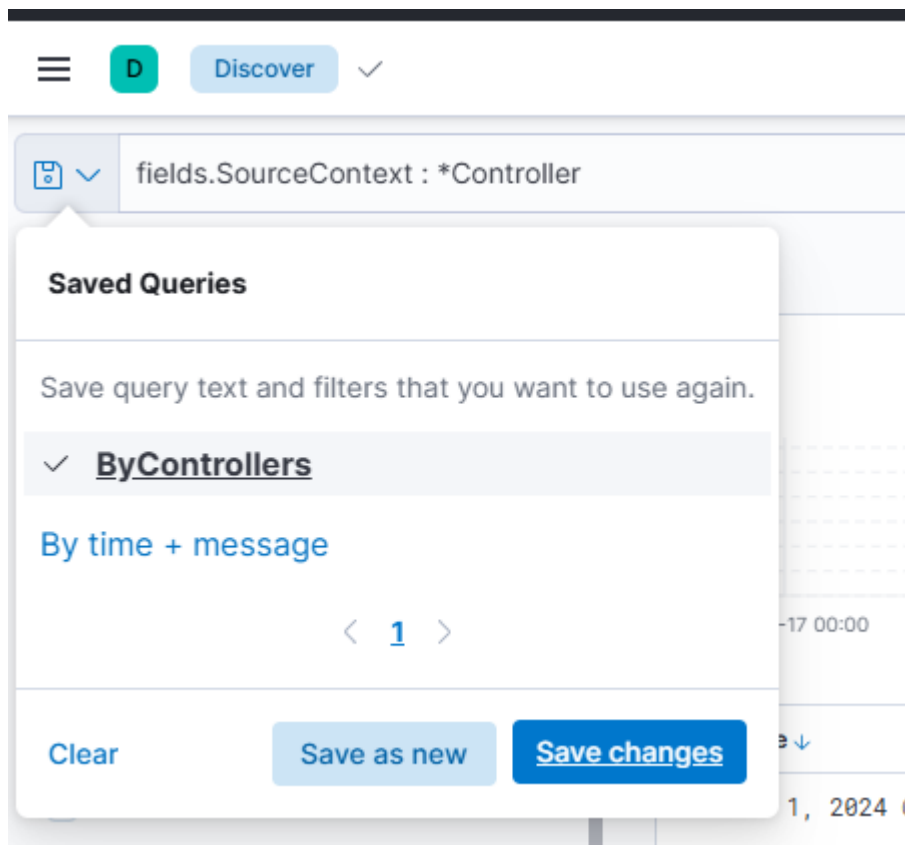
Time	message	fields.Application	fields.MachineName	fields.RequestId	fields.SourceContext	fields.RequestPath	exceptions.Message	fields.ConnectionId	fields.ActionId	level	@timestamp
Dec 1, 2024 @ 21:52:25.287	Request finished "HTTP/1.1" "GET" "ht...	EFKLoggingApi	JKK	00000000-0000-0000-0000-000000000000	Microsoft.AspNetCore.Hosting.Diagnostics	/WeatherForecast	-	00000000-0000-0000-0000-000000000000	-	Information	Dec 1, 2024 @ 21:52:25.287
Dec 1, 2024 @ 21:52:25.286	Executing endpoint "EFKLoggingApi.Controllers.WeatherForecastController.get (EFKLoggingApi)"	EFKLoggingApi	JKK	00000000-0000-0000-0000-000000000000	Microsoft.AspNetCore.Mvc.Infrastructure.ControllerActionInvoker	/WeatherForecast	-	00000000-0000-0000-0000-000000000000	-	Information	Dec 1, 2024 @ 21:52:25.286
Dec 1, 2024 @ 21:52:25.285	Executing action "EFKLoggingApi.Controllers.WeatherForecastController.get (EFKLoggingApi)" in 1022.8488ms	EFKLoggingApi	JKK	00000000-0000-0000-0000-000000000000	Microsoft.AspNetCore.Mvc.Infrastructure.ControllerActionInvoker	/WeatherForecast	-	00000000-0000-0000-0000-000000000000	-	Information	Dec 1, 2024 @ 21:52:25.285
Dec 1, 2024 @ 21:52:25.278	Executing "ObjectResult", writing value of type "EFKLoggingApi.WeatherForecast[]"	EFKLoggingApi	JKK	00000000-0000-0000-0000-000000000000	Microsoft.AspNetCore.Mvc.Infrastructure.ObjectResultExecutor	/WeatherForecast	-	00000000-0000-0000-0000-000000000000	dbd13ac-8111-487a-99ac-15decfab939	Information	Dec 1, 2024 @ 21:52:25.278
Dec 1, 2024 @ 21:52:25.277	Returning 5 forecasts.	EFKLoggingApi	JKK	00000000-0000-0000-0000-000000000000	EFKLoggingApi.Controllers.WeatherForecastController	/WeatherForecast	-	00000000-0000-0000-0000-000000000000	dbd13ac-8111-487a-99ac-15decfab939	Information	Dec 1, 2024 @ 21:52:25.277
Dec 1, 2024 @ 21:52:25.276	Weather forecast requested.	EFKLoggingApi	JKK	00000000-0000-0000-0000-000000000000	EFKLoggingApi.Controllers.WeatherForecastController	/WeatherForecast	-	00000000-0000-0000-0000-000000000000	dbd13ac-8111-487a-99ac-15decfab939	Information	Dec 1, 2024 @ 21:52:25.276
Dec 1, 2024 @ 21:52:23.761	Route matched with "action = 'get', controller = 'WeatherForecast'". Executing controller action with signature "System.Collections.Generic.IEnumerable<EFKLoggingApi.WeatherForecast> get()" on controller "EFKLoggingApi.Controllers.WeatherForecastController".	EFKLoggingApi	JKK	00000000-0000-0000-0000-000000000000	Microsoft.AspNetCore.Mvc.Infrastructure.ControllerActionInvoker	/WeatherForecast	-	00000000-0000-0000-0000-000000000000	dbd13ac-8111-487a-99ac-15decfab939	Information	Dec 1, 2024 @ 21:52:23.761
Dec 1, 2024 @ 21:52:23.759	Executing endpoint "EFKLoggingApi.Controllers.WeatherForecastController.get (EFKLoggingApi)"	EFKLoggingApi	JKK	00000000-0000-0000-0000-000000000000	Microsoft.AspNetCore.Mvc.Infrastructure.EndpointMiddleware	/WeatherForecast	-	00000000-0000-0000-0000-000000000000	-	Information	Dec 1, 2024 @ 21:52:23.759
Dec 1, 2024 @ 21:52:23.757	Request starting "HTTP/1.1" "GET" "http://localhost:5032/" "WeatherForecast" - null null	EFKLoggingApi	JKK	00000000-0000-0000-0000-000000000000	Microsoft.AspNetCore.Hosting.Diagnostics	/WeatherForecast	-	00000000-0000-0000-0000-000000000000	-	Information	Dec 1, 2024 @ 21:52:23.757
Dec 1, 2024 @ 21:52:26.284	Request finished "HTTP/1.1" "GET" "http://localhost:5032/" "Maasha/ThrowErrorMaashaMessage/2" - 200 0 "text/plain; charset=utf-8" 191.1148ms	EFKLoggingApi	JKK	00000000-0000-0000-0000-000000000000	Microsoft.AspNetCore.Hosting.Diagnostics	/Maasha/ThrowErrorMaashaMessage/2	-	00000000-0000-0000-0000-000000000000	-	Information	Dec 1, 2024 @ 21:52:26.284
Dec 1, 2024 @ 21:52:26.282	Executing endpoint "EFKLoggingApi.Controllers.MaashaController.ThrowErrorMaasha"	EFKLoggingApi	JKK	00000000-0000-0000-0000-000000000000	Microsoft.AspNetCore.Mvc.Infrastructure.EndpointMiddleware	/Maasha/ThrowErrorMaashaMessage/2	-	00000000-0000-0000-0000-000000000000	-	Information	Dec 1, 2024 @ 21:52:26.282

в) Додамо фільтр за SourceContext, щоб відобразити лише ті логи, які були згенеровані методами контролерів:

Discover

fields.SourceContext : *Controller

fields.SourceContext : *Controller



Time	message	fields.Application	fields.MachineName	fields.RequestId	fields.SourceContext	fields.RequestPath	exceptions.Message	fields.ConnectionId	fields.ActionId	level	@timestamp
Dec 1, 2024 @ 21:52:25.277	Returning 5 forecasts.	EFKLoggingApi	JK	00000000000000000000	EFKLoggingApi.Controllers.WeatherForecastController	/WeatherForecast	-	00000000000000000000	db0c13ac-8111-487a-99a-c-15dec3fab039	Information	Dec 1, 2024 @ 21:52:25.277
Dec 1, 2024 @ 21:52:25.276	Weather forecast requested.	EFKLoggingApi	JK	00000000000000000000	EFKLoggingApi.Controllers.WeatherForecastController	/WeatherForecast	-	00000000000000000000	db0c13ac-8111-487a-99a-c-15dec3fab039	Information	Dec 1, 2024 @ 21:52:25.276
Dec 1, 2024 @ 21:52:26.190	id cannot be less than or equal to 0. Value passed is [-2].	EFKLoggingApi	JK	00000000000000000000	EFKLoggingApi.Controllers.WeatherForecastController	/WeatherForecast	id cannot be less than or equal to 0. Value passed is [-2].	00000000000000000000	4a54a95b-b0b0-4b12-493b-f852899fada	Error	Dec 1, 2024 @ 21:52:26.190
Dec 1, 2024 @ 21:52:26.189	Weather forecast requested.	EFKLoggingApi	JK	00000000000000000000	EFKLoggingApi.Controllers.WeatherForecastController	/WeatherForecast	-	00000000000000000000	4a54a95b-b0b0-4b12-493b-f852899fada	Information	Dec 1, 2024 @ 21:52:26.189
Dec 1, 2024 @ 21:52:17.153	Weather forecast requested.	EFKLoggingApi	JK	00000000000000000000	EFKLoggingApi.Controllers.WeatherForecastController	/WeatherForecast	-	00000000000000000000	4a54a95b-b0b0-4b12-493b-f852899fada	Information	Dec 1, 2024 @ 21:52:17.153
Dec 1, 2024 @ 21:52:09.638	Weather forecast requested.	EFKLoggingApi	JK	00000000000000000000	EFKLoggingApi.Controllers.WeatherForecastController	/WeatherForecast	-	00000000000000000000	382c148b-999e-43b0-b49c-3e8962fe7fec	Information	Dec 1, 2024 @ 21:52:09.638
Dec 1, 2024 @ 21:52:09.637	Weather forecast requested.	EFKLoggingApi	JK	00000000000000000000	EFKLoggingApi.Controllers.WeatherForecastController	/WeatherForecast	-	00000000000000000000	382c148b-999e-43b0-b49c-3e8962fe7fec	Information	Dec 1, 2024 @ 21:52:09.637
Dec 1, 2024 @ 21:51:52.720	Weather forecast requested.	EFKLoggingApi	JK	00000000000000000000	EFKLoggingApi.Controllers.WeatherForecastController	/WeatherForecast	-	00000000000000000000	382c148b-999e-43b0-b49c-3e8962fe7fec	Information	Dec 1, 2024 @ 21:51:52.720
Dec 1, 2024 @ 21:51:52.719	Weather forecast requested.	EFKLoggingApi	JK	00000000000000000000	EFKLoggingApi.Controllers.WeatherForecastController	/WeatherForecast	-	00000000000000000000	382c148b-999e-43b0-b49c-3e8962fe7fec	Information	Dec 1, 2024 @ 21:51:52.719
Dec 1, 2024 @ 21:51:13.622	Weather forecast requested.	EFKLoggingApi	JK	00000000000000000000	EFKLoggingApi.Controllers.WeatherForecastController	/WeatherForecast	-	00000000000000000000	6d2f118f-645d-4acd-bac0-7781b0162c79	Information	Dec 1, 2024 @ 21:51:13.622
Dec 1, 2024 @ 21:51:13.621	Weather forecast requested.	EFKLoggingApi	JK	00000000000000000000	EFKLoggingApi.Controllers.WeatherForecastController	/WeatherForecast	-	00000000000000000000	6d2f118f-645d-4acd-bac0-7781b0162c79	Information	Dec 1, 2024 @ 21:51:13.621
Dec 1, 2024 @ 21:51:12.517	Weather forecast requested.	EFKLoggingApi	JK	00000000000000000000	EFKLoggingApi.Controllers.WeatherForecastController	/WeatherForecast	-	00000000000000000000	6d2f118f-645d-4acd-bac0-7781b0162c79	Information	Dec 1, 2024 @ 21:51:12.517

Висновки: в результаті виконання цієї лабораторної роботи було ознайомлено з базовими концепціями технологій централізованих систем логуювання на прикладі EFK. Однак через вибір мови програмування Fluentd не використовувався.

На основі отриманих знань було реалізовано практичну частину, яка полягала у створенні застосунку, що передає логи до централізованої системи для їх збереження, подальшої обробки та аналізу.

Вихідний код застосунку можна знайти за наступним посиланням на [GitHub](#).