

Лабораторна робота № 4. Пошукові СУБД: Elasticsearch

МЕТА РОБОТИ: отримати навички роботи з пошуковими базами даних.

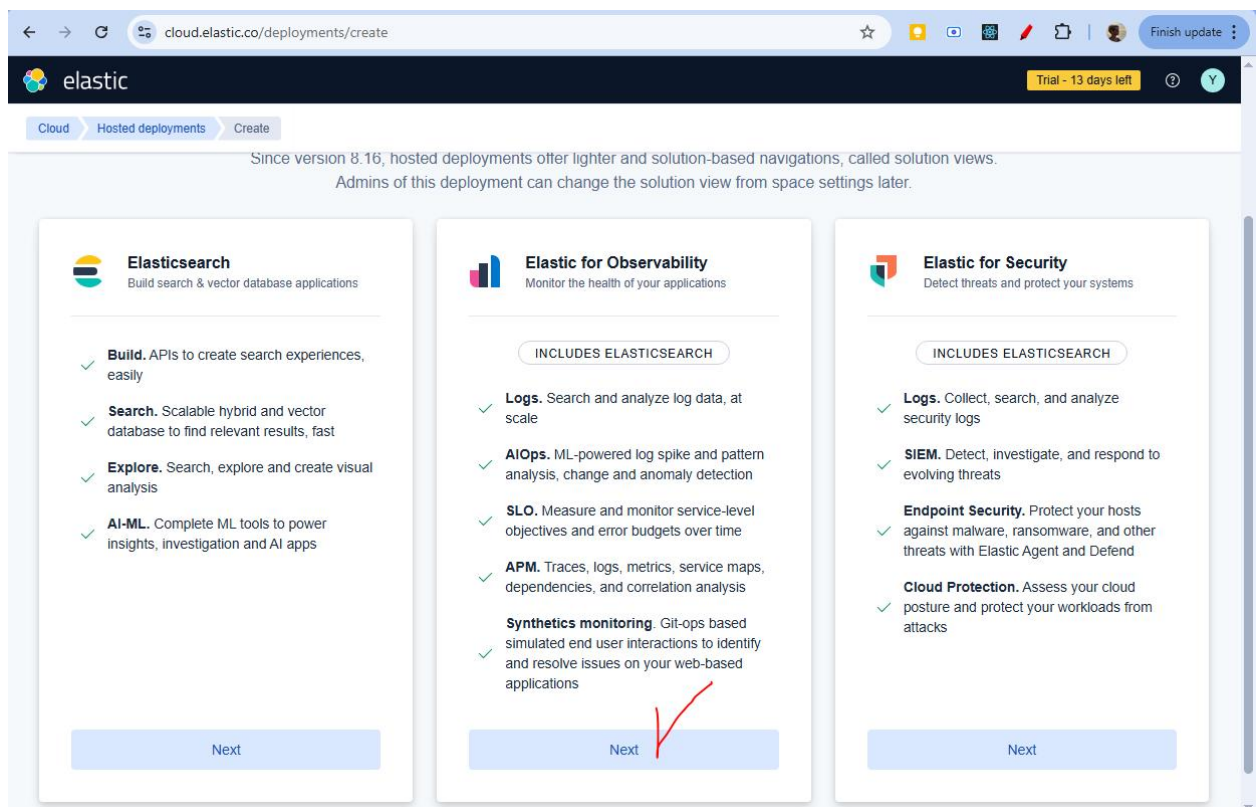
ПОСТАНОВКА ЗАДАЧІ

Використовуючи онлайн інстанси Elasticsearch виконати базові запити зі створення індексу, додавання та пошуку та видалення даних.

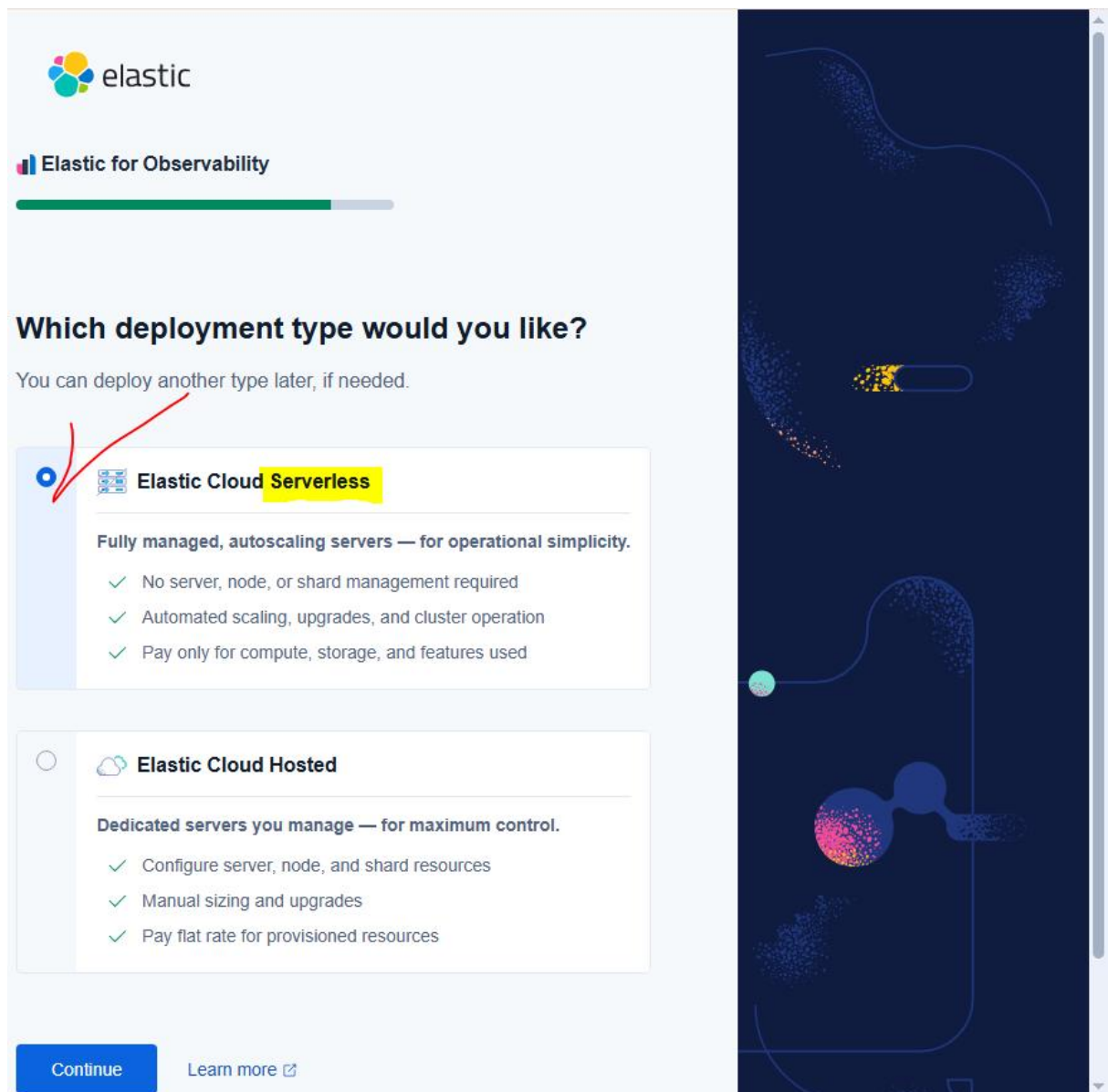
ПОРЯДОК ВИКОНАННЯ

1. Зареєструватись на сайті Elasticsearch за посиланням <https://cloud.elastic.co/> для отримання онлайн-доступу до екземпляра Elasticsearch, який розгорнуто в хмарі (реєстрація та використання безкоштовні в межах Trial періоду, реквізитів картки не вимагається).

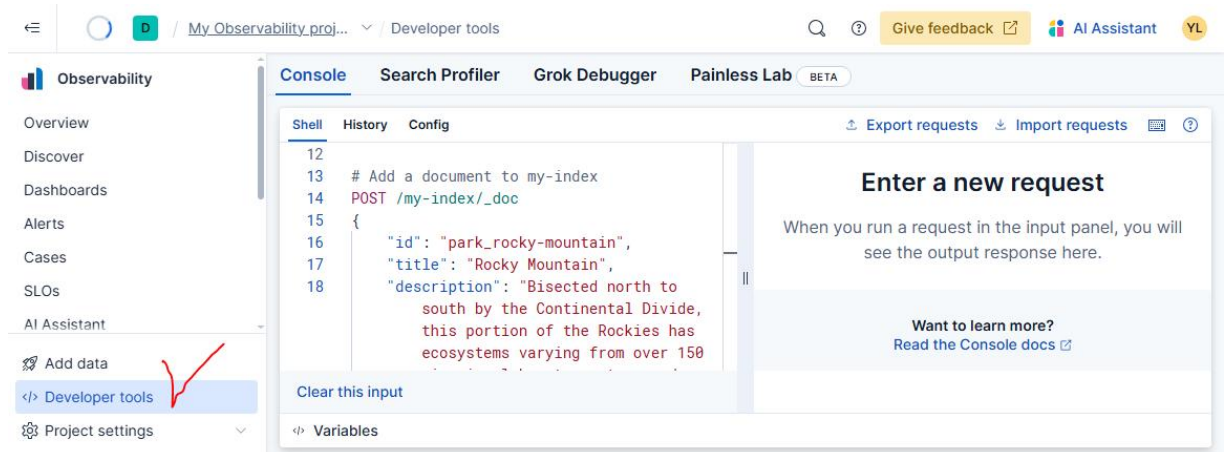
2. Після реєстрації обрати пункт Elastic for Observability:



3. На панелі, що згодом відкриється, обрати Serverless варіант:



4. Після цього відкриється навігаційна панель, в якій треба обрати Developer Tools. Перед вами з'являться 2 панелі: ліва призначена для вводу команд, в правій ви будете бачити результат виконання запитів до бази даних.

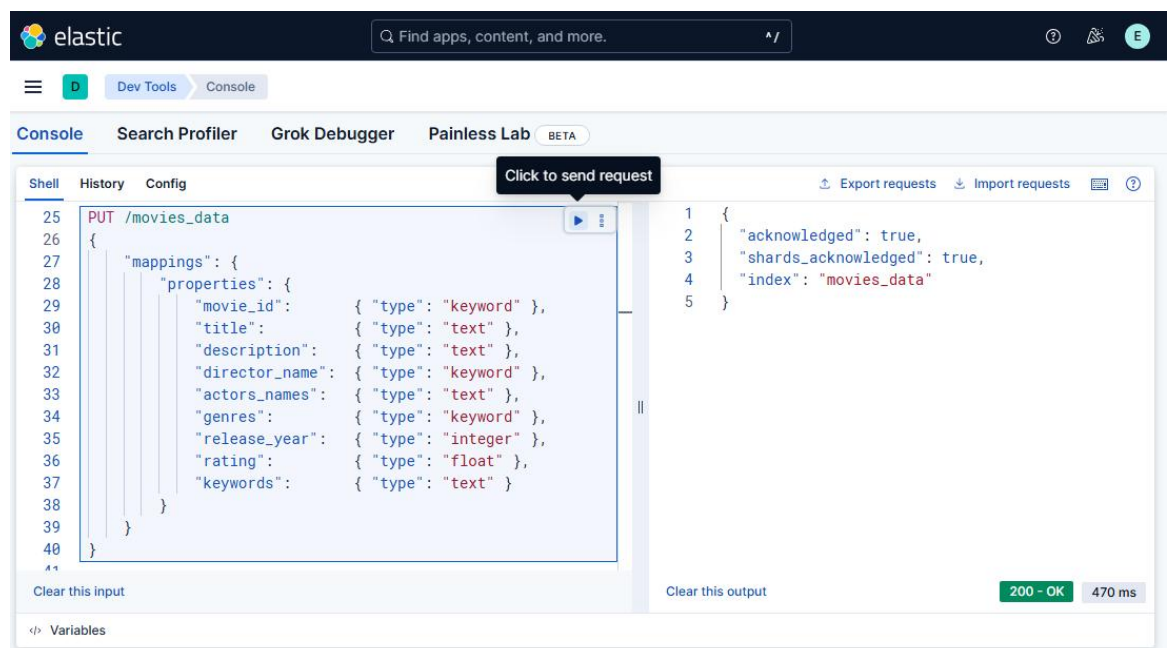


5. Використовуйте ліву панель для наступних запитів.

5.1. Визначення мапінгу (структури індексу/колекції) для бази даних з фільмами. Кількість полів за бажанням можна зменшити (в цьому випадку також треба буде підкорегувати запити на додавання даних).

PUT /movies_data

```
{
  "mappings": {
    "properties": {
      "movie_id":      { "type": "keyword" },
      "title":         { "type": "text" },
      "description":   { "type": "text" },
      "director_name": { "type": "keyword" },
      "actors_names":  { "type": "text" },
      "genres":        { "type": "keyword" },
      "release_year":  { "type": "integer" },
      "rating":         { "type": "float" },
      "keywords":       { "type": "text" }
    }
  }
}
```



5.2. Додавання документу до індексу:

POST /movies_data/_doc

```
{
  "movie_id": "1",
  "title": "The Matrix",
  "description": "Young hacker Neo learns that the world he lives in is a simulation created by machines...",
  "director_name": "Lana Wachowski",
  "actors_names": ["Keanu Reeves", "Laurence Fishburne", "Carrie-Anne Moss"],
  "genres": ["Science Fiction", "Action"],
  "release_year": 1999,
}
```

```
    "rating": 8.7,  
    "keywords": ["hacker", "simulation", "virtual reality"]  
}
```

Спробуйте додати ще 2-3 фільми, дотримуючись такої ж структури. Значення для полів можна задавати і кирилицею (в цьому випадку у пошукових запитах також треба буде використовувати кирилицю).

5.3. Виконати декілька пошукових запитів:

5.3.1. Знайти фільми, в описі яких є слова "computer" або "hacker":

```
GET /movies_data/_search  
{  
  "query": {  
    "match": {  
      "description": "computer hacker"  
    }  
  }  
}
```

5.3.2. Знайти фільми, в назві яких є точна фраза "The Matrix"

```
GET /movies_data/_search  
{  
  "query": {  
    "match_phrase": {  
      "title": "The Matrix"  
    }  
  }  
}
```

5.3.3. Знайти фільми за описом, який містить синтаксичні помилки:

```
GET /movies_data/_search  
{  
  "query": {  
    "match": {  
      "description": {  
        "query": "komputer haker", // замість "computer hacker"  
        "fuzziness": "AUTO"  
      }  
    }  
  }  
}
```

5.3.4. Знайти фільми, жанр яких в точності "Science Fiction"

```
GET /movies_data/_search  
{  
  "query": {  
    "term": {  
      "genres": "Science Fiction"  
    }  
  }  
}
```

Спробувати самостійно написати ще 2-3 пошукових запити. Наприклад, знайти фільми, в яких знімався певний актор.

5.4. Видалити індекс:

```
DELETE /movies_data
```

6. Оформити звіт, в якому навести скріншоти, що демонструють виконання завдань. Додати до звіту відповіді на контрольні запитання.

КОНТРОЛЬНІ ЗАПИТАННЯ

1. До якого типу NoSQL СУБД відноситься Elasticsearch? Які ще СУБД з цього класу є популярними на поточний момент? Аргументуйте свою відповідь, прикріпивши посилання на відповідні рейтинги.
2. Які сценарії використання пошукових СУБД ви знаєте?
3. Під який тип запитів в першу чергу оптимізовані пошукові СУБД?
4. У якому вигляді зберігає дані СУБД Elasticsearch?
5. Чи підтримує Elasticsearch механізм реплікацій та шардингу?