Recover Data about Detected Defects of Underground Metal Elements of Constructions in Amazon Elasticsearch Service

Roman Mysiuk ¹, Volodymyr Yuzevych ²

¹ Ivan Franko National University of Lviv
1 Universytetska Street, Lviv, 79000, Ukraine

² Karpenko Physico-mechanical Institute of the NAS of Ukraine
5 Naukova Street, Lviv, 79060, Ukraine

Abstract. This paper examines data manipulation in terms of data recovery using cloud computing and search engine. Accidental deletion or problems with the remote service cause information loss. This case has unpredictable consequences, as data must be re-collected. In some cases, this is not possible due to system features. The primary purpose of this work is to offer solutions for received data on detected defects of underground metal structural elements using modern information technologies.

The main factors that affect underground metal structural elements' durability are the soil environment’s external action and constant maintenance-free use. Defects can usually occur in several places, so control must be carried out along the entire length of the underground network. To avoid the loss of essential data, approaches for recovery using Amazon Web Service and a developed web service based on the REST architecture are considered. The general algorithm of the system is proposed in work to collect and monitor data on defects of underground metal structural elements. The result of the study for the possibility of data recovery using automatic snapshots or backup data duplication for the developed system.

Keywords: recovering data; search; cloud computing; snapshots; corrosion; defects; underground metal elements of construction.

INTRODUCTION

Underground metal elements of constructions provide transportation and delivery of oil and gas products to different countries of Eastern and Central Europe, which is 2–3 times faster than in any other way. With constant operation in underground metal elements of constructions, there are various defects due to active mechanical stress and function in specific conditions of the soil environment. Severe defects usually cause accidents on such lines and lead to environmental and even man-made disasters. Based on the possible risks, all underground metal elements of constructions are carefully monitored and maintained in good working order so that no cracks can affect the functioning of the structure [1, 2]. To monitor such systems, a large amount of data needs to be processed because the length of underground metal elements of constructions is kilometres [3]. Big Data is actively developing in Ukraine and is used in various sectors of the economy due to the rapid scalability of such systems. The most popular way to transfer data is to use network data transfer protocols. Elasticsearch is a non-relational database (NoSQL) in which data is transmitted via network requests. It can be used as a portable microcomputer with the possibility of using additional modules to collect input information for the system. One of the most popular cloud environments is Amazon Web Service (AWS). This allows constant access to the database and data processing using cloud technology. The work aims to create a secure system that can analyse input information and show the possibility of data recovery in case of loss in such scenarios.
MATERIALS AND METHODS

Materials and Methods

There are various types of databases used for storing different varieties of data:

- A centralised database differs from others in that it is accessed through a computer network, which can give them access to a central processing unit (CPU) that supports the database. Such a means of using databases are commonly used in local networks.

- A distributed database is a database that consists of two or more files located on different sites or the same network, or completely different networks. Parts of the database are stored in several physical locations, and processing is distributed among many database nodes.

- A NoSQL database, which provides a mechanism for storing and retrieving data, is different from the approach of relationship tables in relational databases. NoSQL databases are increasingly used in big data tasks and real-time web applications.

- A relational database is a collection of data elements organised as a set of formally described tables, from which data can be accessed or recollected in many ways without reorganising the database tables.

- A cloud database is a type of database that performs queries on a remote machine.

- In computing, a graph database uses various nodes, edges, and properties to represent and store information for semantic queries. This type is used to visualise complex structural connections better.

Cloud technologies and computing are becoming increasingly popular. This is because cloud environments have many useful features and security. Amazon Web Service is chosen for the developed system due to its user-friendly interface and easy integration.

This system uses a non-relational database because NoSQL databases are document, key-value, graph, or wide-column stores and are well suited for working with many JavaScript Object Notation (JSON) documents.

Elasticsearch (ES) is a search engine that uses the Lucene library. This allows you to create a distributed full-text search with the ability to use multiple units using the Hypertext Transfer Protocol (HTTP) web interface and schema-free JSON documents [4]. Elasticsearch provides a full Query Domain-Specific Language DSL (DSL) based on JSON to define queries. Elastic search queries can include many conditions simultaneously, and the size of such a query can be significant. Queries can be divided into geo, compound, range, term level, string, multi-match, full text, and match all [5].

In ES, all data is contained in separate indexes, and searches can join using a query with multiple fields at once [6]. Indexes can be divided into shards, and each guard can have any number of replicas. Routing and rebalancing operations are performed automatically when new documents are added. This database has a separate bulk Application Programming Interface (API) function for data ingestion, which allows multiple operations to index and deletes in a single API request. It can significantly increase the indexing speed. To execute bulk API, we need to have a preformed data file and run it from the command line using a simple curl command. There are some restrictions on file size and the number of entries. It has been tested that the file should not exceed about 50 MB or the number of records of 100,000 for the presented case [7].

Amazon Elasticsearch Service (Amazon ES) lets you store up to 3 PB of data in a single cluster, enabling you to run large log analytics workloads via a single Kibana interface. Using the AWS cloud environment allows us to have a global, highly accessible, secure domain to access ES [8]. Another advantage is the quick-to-search data [9].

Data recovery is one of the main issues [10]. There is a unique mechanism for recovering data lost or deleted, called snapshots. Snapshots are backups of the Amazon ES cluster indexes and status. The state includes cluster parameters, node information, index parameters, and shard distribution. Data unavailability may be due to index status. Possible positions include green, yellow, and red. The last two signal some problems; in this case, data recovery is also essential. Amazon Elasticsearch is ready to recover data [11]:

- Automated snapshots are for cluster recovery only. Amazon ES stores automated shots in a preconfigured bucket of AWS Simple Cloud Storage (S3) at no extra charge.

- Manual snapshots are used to restore a cluster or to move data from one cluster to another. This
requires manual shooting. These snapshots usually use your Amazon S3 bucket, and S3 payments apply accordingly.

**Experimental procedures**

This section describes the system's main configuration, data preparation, and design. The processes of setting up the cloud environment, filling it with data, and the structure of the web service are described.

**System Configuration.** A restful web service is created that works with ES and displays information about the status of underground metal elements of constructions in real time. This web service is developed using Java programming language and Spring framework [12].

Before starting with our approach, you should follow a few preliminary steps. The following tools are used in the development:

- **Java Development Kit** – a set of development tools and libraries containing documentation, standard libraries, and code compiler for Java classes. Today, the approach of building a client-server architecture using the Java programming language through cross-platform and object-oriented language is prevalent.

- **Maven** is an open-source programming tool for building our projects. There are a few phases of the lifecycle: validate, compile, test, package, verify, install, and deploy. Each of these phases helps to perform specific actions with the code. This application construction allows using ready-made functions to collect code into one assembly. Moreover, all additional libraries are downloaded from a remote repository, which replaces jar files.

To efficiently run multiple instances of Elasticsearch locally, you need to download the tar or zip archive from the official site and run the bat file in the bin folder.

To test ES in the cloud, follow these steps using the Amazon ES, AWS Command Line Interface (AWS CLI), or AWS Software Development Kit (AWS SDK):

1. Create an Amazon ES domain. To do this, you must select the appropriate general settings, such as deployment and instance types, and provide a name for the domain, etc. All other settings can be ignored for now. In my case, one of the smallest instance types was selected **t3.small.elasticsearch.** According to on-demand instance pricing for this type, vCPU, and Memory (GiB) are 2. Which is enough for initial testing [8].

2. Wait until the settings are activated. It typically takes 15–20 minutes to initialise.

3. Upload the data to the Amazon ES domain for indexing.

**Data Preparation.** For the current testing of this system, the generated test data using Bash were used randomly. The Restful service mocking feature allows us to simulate accurate data using generated test data. In the future, there is a plan to integrate actual test data from some external devices.

Bash is a scripting language where any command can be grouped and executed in a single file. Functions can be created in any programming language, simplifying code reuse. The bash script is a temporary solution. Instead of fundamental values, random values of characters are generated using functions in Bash. The main steps are following in this script:

- Create variables for index names for devices and ES domains.
- Create a mapping of records in ES for each index using a loop [13]. The default mapping is created at the first record, but when you want to make a custom. This is filled in once and is editable. Mapping indicates in what format and type of data will be stored. The curl tool sends requests via the command line, which allows you to work with URL syntax. The following command can be executed:

  ```bash
curl -X PUT "$domain/$indexName" -H "Content-Type: application/json" -H "Accept: application/json" -d @$deviceIndexMapping
  ```

- Generate a temporary file with test data filled with random values using Bash. Figure 1 displays a model of test data. The JSON format, which is presented in a key-value format, was used to describe the data. This presentation is easy to read and acceptable for Elasticsearch search engines.

- Push data from file to ES using Bulk API:

  ```bash
curl -H 'Content-Type: application/x-ndjson' -XPOST $domain/_bulk --data-binary @"generated_data.json"
  ```
– Remove the temporary file.

Generated files should be divided and pushed in a loop due to the limitation of bulk API operations. For each index to have approximately the same number of records, the number of completed cycles must be multiplied by the number of records per push and divided by the number of indices. As a result, it allows the approximate number of iterations of loops to reach a certain number in each index. There is also a collection of information on various parameters, such as time and relativity to the device, allowing you to display it on web applications.

```json
{
   "id": "\$\{randomId\}"
   "device_name": "\$\{($RANDOM % 1000 + 1)\}"
   "date": "\$\{date \-\%Y\%m\%d\}"
   "time": "\$\{date \-\%H\%M\%S\}"
   "external_factors": [
   {
      "soil_moisture": "\$\{($RANDOM % 1000 + 1)\}"
      "temperature": "\$\{($RANDOM % 1000 + 1)\}"
   }
   },
   "potential_current": [
   {
      "value": "\$\{($RANDOM % 1000 + 1)\}"
   }
   },
   "geographic_coordinate_system": [
   {
      "latitude": "\$\{($RANDOM % 1000 + 1)\}"
      "longitude": "\$\{($RANDOM % 1000 + 1)\}"
   }
   ]
}
```

Figure 1 – Example of test data filled with random values using Bash script

**System design.** The Internet is a universal way to access data anywhere in the world. The development of web services as a backend part of the browser has become very popular due to the increasing use of web browsers [14].

The structure of the proposed system is shown in Figure 2. The central part of the system is a web service. The most used architectural style REST is used for creation. This web service can easily obtain and display the desired data. The service is developed based on Spring Framework.

To avoid external data interception and make the system more closed, it is recommended to use the secure Hypertext Transfer Protocol Secure. With Transport Layer Security (TLS) or Secure Sockets Layer (SSL), data is encrypted with this approach. Therefore, you need to configure environment variables in the settings or the environment of the program or computer with which the web service works.

![Figure 2 – The system for collecting and analysis for underground metal elements of constructions architecture](image)

Although personnel data is not used in this system, we must take care of our product’s security by reducing the system’s vulnerability to external attacks. The approach of using Open Authorization (OAuth) for greater security is also popular, which is added to the existing web service too. With the help of generated tokens with a request, communication with the system can be sent.

Several endpoints are developed in web service for restoring data using some from the list of standard HTTPS methods such as GET, HEAD, POST, PUT, PATCH, DELETE, CONNECT, OPTIONS, and TRACE. For instance: GET method with request template:

```plaintext
{domain}/snapshots/{snapshot_id}, POST {domain}/snapshots with the following request body
{"from":"from\_snapshot\_id",
"to":"to\_snapshot\_id"}
```

This group of operations is based on the Reindex API in ES. This feature is an additional option for data recovery, although it can cause redundancy of duplicate data.

In addition, the system allows getting information on time, id, device, and all the information, in general, using the web service. The following endpoints have been created for getting information:
This system can be considered a cyber-physical system. A new method for assessing the degree of the information security risk of the system for the control of underground metal constructions is proposed. In addition, taking into account the index of the probability of a successful attack on the system, the index of the impact of the attack, the adjustment index, which provides feedback, and the quality criterion \[15\].

Web service works like an additional layer on the system. In this case, that helps us to communicate with data using a particular query. From on security perspective, that allows preventing different external attached to the system in further testing.

### RESULTS AND DISCUSSION

Before starting the web service, the environment variables must be configured for setting OAuth token and using HTTPS, then:

1. Run Bash script to generate and push test data in Elasticsearch.

2. Check created indices in Figure 3. The created indexes are presented in the form of a table with the size of the existing data, the current state of the cluster, and other information. To view this information, the \/_cat/indices\ endpoint of Elasticsearch must be reached.

3. Run the project using the command line `mvn clean install`.

![Figure 3 - List of created indices with generated data](image-url)

**Index duplication.** The first way to prevent data loss is to create a duplicate index. This method should be considered one of the most possible. This does not provide complete security, but in case of problems with the current index allows you to switch to snapshot quickly. In Figure 4, the execution of the query for duplicating indexes with data is demonstrated. For web service testing purposes, the Postman desktop application is used. This tool allows querying web resources. It is beneficial when you need to make requests to create, update, and delete documents. A regular browser can be used to view GET requests, as shown in Figure 3.
After duplication, the existing Elasticsearch indexes can be checked. Figure 5 shows the duplicated device-1-snapshot index.

In general, duplicating data is not the best way to avoid data loss, as an increase in records and memory accompanies it.

The number of indices is not a significant factor in productivity. An important parameter is the number of shards that must be selected correctly.

The cluster should be optimised to avoid impact on performance. The best ratio of the number of bits and replicas can be chosen using performance testing [16]. Amazon CloudWatch can be used for monitoring the results of cluster metrics [8].
Automated and manual snapshots

The second way to recover data is to use automated and manual snapshots on Amazon ES if we do not have a duplicated data set.

Elasticsearch requires repository-s3 plugin to work with an S3 bucket. Therefore, the following operations on the command line must be performed:

- Installing repository-s3: `sudo bin/elasticsearch-plugin install repository-s3`
- Restarting Elasticsearch: `sudo systemctl restart elasticsearch.service`

The next step is to log in to the AWS account and create an identity-based user policy (IAM) with the necessary S3 permissions [8].

The next necessary step is to specify a bucket name to register the S3 repository using the following command:

```
curl --location --request PUT
'http://{domain}/_snapshot/backup_repository_s3'
--header 'Content-Type: application/json'
data-raw '{ "type": "s3", "settings": {"bucket": "elastic-slm"} }'
```

The last setting is defining our Snapshot lifecycle management (SLM) policy. This means that we will always be able to have a copy of the indexes available, which are generated and deleted at the specified time. In the JSON body, the following fields should be defined: schedule (when the snapshot must be taken into S3 bucket using Cron syntax), name (how to name the snapshot with the current date using date math), repository (where to store the snapshot), config (which indices to include in the snapshot), retention (define the expiration time and the minimum and maximum snapshot count) [17].

To apply these changes, the command should be performed:

```
curl --location --request PUT
'http://{domain}/_slm/policy/backup_policy_daily_s3'
--header 'Content-Type: application/json'
data-raw '{    "schedule": "0 01 2 * * ?", "name": "backup-device-{now/d}" },
"repository": "backup_repository_s3", "config": {    "indices": ["*"],
"retention": { "expire_after": "1d", "min_count": 5, "max_count": 20 } }
```

The web interface for lifecycle management snapshots is shown in Figure 6.

```
aws s3 ls s3://elastic-slm --human-readable --recursive --summarize
```

In the same way, indices can be restored using a web interface on Amazon ES.

The restore command must be executed to restore the snapshot after the last step in the AWS command line:
curl -X POST 'elasticsearch-domain-end-point/_snapshot/repository/snapshot/_restore'

According to Amazon documentation, most automated images are stored in cs-automated storage. If your domain encrypts data at rest, it is stored in the cs-automated-enc repository [11].

One index can be changed to another duplicate index with the same working indices. In ES, bulk index alias API functionality is created to create and remove multiple index aliases in a single API request. Alternatively, an alias can be used as a second name to refer to one or more existing indexes. However, several indexes may contain the same alias, but it is impossible to have the same name as the index. This HTTP request is as follows:

```json
POST {ES domain}/_aliases {"actions": [{"remove": {"index": "device-1", "alias": "device-1-alias-1" } }, { "add": { "index": "device-1", "alias": "device-1-alias-2" } }]}
```

**CONCLUSIONS**

Monitoring data on defects in underground metal structural elements is helpful for the timely prevention of hazards that often occur today during the transportation of various products and substances. For the smooth operation of the proposed system, you need to save the data and, in case of loss, quickly restore it. This article describes how to configure these tools for use. Methods of automatic recovery through snapshots in Amazon Web Service or through duplicate indexes in Elasticsearch are considered. Therefore, given the implemented system, we can make the following results:

- Checked the data recovery application from the non-relational database Elasticsearch and the created web service.
- Tested database completion based on a set of test data using Bash.
- Considered the system for possible detection of defects in underground metal elements of structures.

**REFERENCES**


