Processing Network Logs in Elasticsearch vs. Oracle Database

Contents

1. Introduction 3

2. Architecture and data model 4

3. Performance and scalability 6

4. Security and access management 7

5. Multi-tenant scenarios 9

6. Practical use and recommendations 10

7. Conclusion and recommendations 12

1. Introduction

1.1 Importance of network logs

Network logs are an integral part of modern information systems. They record all events and interactions within the network infrastructure, which enables:

- **Detection and prevention of security incidents**: Network logs provide detailed information about access, unauthorized access attempts and other anomalies, which is key to identifying and responding to security threats.
- **Monitoring and optimization of network performance**: Analysis of network logs makes it possible to identify congestion, outages or inefficient configurations, leading to better performance and availability of services.
- **Compliance with regulatory requirements and auditing**: Retention and analysis of network logs is often essential for meeting legislative and security standards such as GDPR, HIPAA or ISO standards.
- **Support for incident handling and forensic analysis**: In the event of incidents, network logs provide valuable information for retrospective analysis and identification of the causes of problems.

1.2 Objectives of the study

The aim of this study is to compare two technologies, **Elasticsearch** and **Oracle Database**, in terms of their capabilities for processing network logs. We will focus on the following aspects:

- **Architecture and data model**: How each technology stores and processes network logs.
- **Performance and scalability**: How the systems behave under different volumes of network logs and load.
- **Security and access management**: What security and access control mechanisms they offer when working with network logs.
- **Multi-tenant scenarios**: How they support multiple tenants in a single system when processing network logs.
- **Practical use and recommendations**: In which cases it is appropriate to use one or the other technology for processing network logs.

1.3 Overview of technologies

Elasticsearch:

- It is a distributed search and analytics engine based on s
- Optimized for fast searching and analysis of large volumes of data in real time, including network logs.
- It uses a document-oriented model in which data are stored as JSON documents.
- It is often used in combination with tools such as Logstash and Kibana for processing and visualizing network logs.

Oracle Database:

- It is a fully fledged relational database system with an emphasis on transactionality and data integrity.
- It supports ACID properties, ensuring the consistency and reliability of operations.
- It provides robust security features and access management capabilities, which is important when processing sensitive network logs.
- It enables integration with various data analysis and visualization tools.

2. Architecture and data model

2.1 Elasticsearch

Elasticsearch is a distributed search and analytics engine optimized for processing large volumes of data in real time. Its architecture is designed to enable horizontal scaling and high availability.

Key components:

- **Indices and documents**: Data are organized into indices containing documents in JSON format. Each document represents a unit of data, for example a single network log record.
- **Shards and replicas**: Each index is divided into primary and replica shards, which enables parallel processing and increases the system's resilience to outages.
- **Ingest pipeline**: Before data are stored in an index, they can be processed using an ingest pipeline, which enables transformations, data enrichment and other operations.

- **Integration with Logstash and Beats**: Tools such as Logstash and Beats are often used for the collection and pre-processing of network logs, ensuring efficient processing and transfer of data to Elasticsearch.

This architecture enables Elasticsearch to process and analyse network logs efficiently in real time, which is key for monitoring and security purposes.

2.2 Oracle Database

Oracle Database is a robust relational database system that provides strong support for transaction processing and data integrity. Its architecture is designed to ensure high availability, scalability and security.

Key components:

- **Relational data model**: Data are organized into tables with defined schemas, which enables a precise structure and data integrity.
- **Process architecture**: Oracle uses a combination of client and server processes, including background processes such as DBWR (Database Writer), LGWR (Log Writer) and others, which ensure efficient request processing and resource management.
- **Redo logs and archives**: Changes in the database are recorded in redo logs, which enables data recovery in the event of failure and supports replication through tools such as Oracle Data Guard.
- **Support for unstructured data**: Although Oracle is primarily a relational database system, it provides tools such as Oracle Text and XML DB that enable work with unstructured data, including network logs.

This architecture enables Oracle Database to process and analyse network logs efficiently, especially in environments where Oracle is already deployed as the primary database and where high demands are placed on security and data integrity.

2.3 Comparison

Criterion	Elasticsearch	Oracle Database
Data model	Document-oriented (JSON)	Relational (tables with a schema)
Schema flexibility	High – the schema can be dynamic	Low – requires a firmly defined schema
Indexing	Inverted indices for full-text search	Various types of indices, including full-text search via Oracle Text

Scalability	Horizontal via shards and replicas	Primarily vertical, horizontal via Oracle RAC
Tool integration	Logstash, Beats for log collection and processing	Oracle LogMiner, Audit Vault for log collection and analysis
Suitability for network logs	High – optimized for fast processing of large volumes of logs	Suitable – strong integrity and security, but less flexible for unstructured data

3. Performance and scalability

3.1 Elasticsearch

Elasticsearch is designed for high performance when processing large volumes of data in real time, which is ideal for network logs generated in real time. Its horizontal scaling capability allows it to adapt to growing log processing requirements.

Key features:

- **Fast indexing and searching**: It uses inverted indices and parallel data processing, which enables fast searching across huge amounts of logs.
- **Horizontal scaling**: Support for distributing data across multiple nodes ensures high availability and performance.
- **Performance optimization**: The possibility of optimizing performance through shard management, replication and the use of SSD drives to improve I/O operations.
- **Index Lifecycle Management (ILM)**: Enables automated management of the index lifecycle, which helps to optimize performance and costs.

3.2 Oracle Database

Oracle Database provides robust performance for transaction processing; however, when processing large volumes of network logs, it may be necessary to optimize the configuration and hardware resources. Vertical scaling and the use of features such as partitioning can help to improve performance.

Key features:

- **ACID properties**: They ensure the consistency and reliability of stored logs.

- **Advanced security features**: They enable detailed auditing and control of access to logs.
- **Partitioning**: Enables large tables to be divided into smaller parts, which improves query performance and the processing of large volumes of data.
- **Oracle Real Application Clusters (RAC)**: Provides horizontal scaling and high availability through multiple nodes.

3.3 Comparison of performance and scalability

Criterion	Elasticsearch	Oracle Database
Data indexing	Fast indexing of large volumes of data in real time	Efficient in transaction processing, may require optimization for large volumes
Search	Fast full-text search and analysis	Strong in structured queries, less efficient in full-text search
Scalability	Horizontal scaling by adding nodes	Primarily vertical, horizontal via Oracle RAC
Management of large volumes of data	Efficient processing and analysis of large volumes of data	Requires partitioning and optimization for efficient processing of large volumes of data
Performance optimization	Automated management of indices and shards, use of ILM	Requires manual optimization and partitioning management

4. Security and access management

4.1 Elasticsearch

Elasticsearch provides a comprehensive set of security features that ensure data protection and access control across the entire Elastic Stack.

Key security mechanisms:

- **Authentication**: Support for multiple authentication methods, including integration with LDAP, Active Directory, SAML, Kerberos and the option of using custom authentication mechanisms.
- **Authorization and RBAC**: The implementation of Role-Based Access Control (RBAC) makes it possible to define detailed access rights at the level of indices, documents and fields. In this way, it is possible to determine precisely who has which permissions within the system.
- **Data encryption**: Support for encryption of data in transit using TLS/SSL and encryption of data at rest, thereby ensuring the protection of sensitive information against unauthorized access.
- **Auditing**: The ability to monitor and record user actions and system events, which is important for ensuring regulatory compliance and for forensic analysis.
- **IP filtering**: Makes it possible to restrict access to Elasticsearch and Kibana based on IP addresses or ranges, thereby increasing system security.

4.2 Oracle Database

Oracle Database offers extensive security features focused on data protection, access management and ensuring compliance with regulatory requirements.

Key security mechanisms:

- **Authentication**: Support for multiple authentication methods, including passwords, certificates, Kerberos, RADIUS and integration with LDAP directories.
- **Authorization and rights management**: The use of roles and privileges to control access to database objects. The possibility of implementing fine-grained access control using Oracle Virtual Private Database (VPD) and Oracle Label Security (OLS).
- **Data encryption and masking**: Oracle Advanced Security provides encryption of data at rest and in transit, as well as features for masking sensitive data.
- **Auditing and monitoring**: Oracle Database Vault and Oracle Audit Vault enable the monitoring and control of access, thereby ensuring protection against unauthorized access, including by privileged users.
- **Separation of duties**: The possibility of implementing a separation of duties policy at the operating system and database level, thereby reducing the risk of abuse of authority.

4.3 Comparison of security mechanisms

Criterion	Elasticsearch	Oracle Database
Authentication	LDAP, Active Directory, SAML, Kerberos, custom mechanisms	Passwords, certificates, Kerberos, RADIUS, LDAP
Authorization and RBAC	Role-Based Access Control (RBAC) at the level of indices, documents and fields	Roles and privileges, VPD, OLS for fine-grained access control
Data encryption	TLS/SSL for transit, encryption of data at rest	Encryption of data at rest and in transit via Oracle Advanced Security
Auditing	Monitoring of user actions and system events	Oracle Database Vault, Audit Vault for monitoring and controlling access
IP filtering	Restriction of access based on IP addresses or ranges	Possibility of restricting access via network policies and a firewall
Separation of duties	Possibility of defining different levels of access for different roles	Implementation of a separation of duties policy at the OS and database level

The above comparison shows that Elasticsearch provides flexible and modern security mechanisms suitable for dynamic environments and fast processing of network logs. On the other hand, Oracle Database offers robust and comprehensive security features that are ideal for environments with high requirements for security, data integrity and compliance with regulatory standards.

5. Multi-tenant scenarios

5.1 Elasticsearch

Elasticsearch does not provide native support for a multi-tenant architecture in the traditional sense, but it enables the implementation of multi-tenant solutions through various approaches:

- **Separate indices for each tenant**: Each tenant has its own index, which ensures data isolation. However, this approach can lead to increased complexity in managing a large number of indices and to potential performance problems.
- **Shared indices with a tenant identifier**: The data of multiple tenants are stored in a single index, with each document containing a tenant identifier. This approach reduces the number of indices but requires rigorous security to prevent unauthorized access to other tenants' data.
- **Security and access control**: Elasticsearch provides data security mechanisms such as authentication, authorization and auditing, which are key when implementing multi-tenant solutions.

When designing a multi-tenant solution in Elasticsearch, it is important to consider the trade-offs between data isolation, performance and administrative complexity.

5.2 Oracle Database

Oracle Database provides native support for a multi-tenant architecture through the concept of the Container Database (CDB) and Pluggable Databases (PDBs)

- **Container Database (CDB)**: The main database, which can contain multiple PDBs.
- **Pluggable Databases (PDBs)**: Isolated databases that can be plugged into a CDB. Each PDB has its own schemas, objects and data, which ensures strong isolation between tenants.
- **Applications and application containers**: Oracle makes it possible to create applications and application containers that can be shared among multiple PDBs, which facilitates the management and deployment of applications in a multi-tenant environment.
- **Security and access management**: Oracle provides advanced mechanisms for data security and access management, including separation of duties, auditing and encryption, which is key in a multi-tenant environment.

The multi-tenant architecture of Oracle Database makes it possible to manage multiple databases efficiently within a single environment, reduces infrastructure costs and simplifies the management and deployment of applications.

6. Practical use and recommendations

6.1 Scenarios suitable for Elasticsearch

Elasticsearch is ideal for cases where fast searching and analysis of large volumes of data in real time are required. Thanks to its horizontal scaling capability and support for full-text search, it is often used in the following scenarios:

- **Log monitoring and analysis**: Elasticsearch is frequently used for the centralized collection and analysis of logging data from various systems, which enables rapid identification of problems and anomalies.
- **Search applications**: Thanks to its support for full-text search, Elasticsearch is suitable for implementing search functions in applications such as e-shops, portals or documentation systems.
- **Analytical dashboards**: In combination with tools such as Kibana, Elasticsearch makes it possible to create interactive dashboards for real-time data visualization and analysis.

6.2 Scenarios suitable for Oracle Database

Oracle Database is a robust relational database system that provides a high level of reliability, security and transactional integrity. It is suitable for the following scenarios:

- **Transactional systems**: Applications that require strict data consistency and support for ACID properties, such as banking systems, ERP or CRM, benefit from the robustness of Oracle Database.
- **Complex queries and reporting**: Oracle provides advanced features for building complex queries, reporting and analytics, which is advantageous for enterprise applications.
- **Multi-tenant environments**: With native support for a multi-tenant architecture through CDB and PDB, Oracle is suitable for SaaS solution providers and organizations managing multiple database environments.

6.3 Combined use of Elasticsearch and Oracle Database

In many cases, it is advantageous to combine both systems, with each of them fulfilling a specific role:

- **Elasticsearch as an analytical tool**: Data from Oracle Database can be replicated to Elasticsearch for the purposes of fast analysis and searching, while Oracle serves as the primary storage with an emphasis on transactional integrity.
- **Search offloading**: Search operations can be moved to Elasticsearch, thereby reducing the load on Oracle Database and improving the overall performance of the system.

6.4 Recommendations for system selection

When deciding between Elasticsearch and Oracle Database, it is important to consider the following factors:

- **Nature of the data**: For structured data with complex relationships, Oracle is more suitable, whereas for unstructured or semi-structured data requiring fast searching, Elasticsearch is more suitable.
- **Consistency requirements**: If strong consistency and transactional integrity are critical, Oracle is the preferred choice.
- **Performance and scalability**: For applications with high performance demands and a need for horizontal scaling, Elasticsearch is more advantageous.
- **Security and access management**: Oracle provides advanced security features and access management tools, which is important for applications with high security requirements.

7. Conclusion and recommendations

The comparison of **Elasticsearch** and **Oracle Database** in network log processing revealed that each of these technologies has its own strengths and is suitable for different use scenarios.

7.1 Main findings

- **Elasticsearch**:
    - It excels at fast indexing and searching of large volumes of data in real time.
    - It is ideal for log monitoring and analysis, the creation of interactive dashboards and anomaly detection.
    - It supports horizontal scaling, which enables efficient processing of growing volumes of data.
    - In combination with tools such as Kibana and Logstash, it forms a strong ecosystem for observability and log analysis.
- **Oracle Database**:
    - It provides robust support for transaction processing with an emphasis on ACID properties.
    - It is suitable for applications requiring a high level of data consistency and security.

- It offers advanced features for auditing, access management and multi-tenant architecture.
- It integrates tools such as Oracle Audit Vault and Database Firewall for advanced database monitoring and security.

7.2 Recommendations

- **For organizations focused on fast analysis and visualization of network logs**:
  - **Elasticsearch** is a suitable choice thanks to its ability to process and analyse large volumes of logs in real time.
  - It is ideal for DevOps teams and security analysts who need fast access to information and the ability to create customized dashboards.
- **For organizations with high requirements for security, consistency and data management**:
  - **Oracle Database** is suitable for applications where transactional integrity and compliance with regulatory requirements are critical.
  - It is ideal for financial institutions, healthcare organizations and SaaS solution providers that require robust security and access management.
- **For hybrid scenarios**:
  - A combination of both technologies can bring the best of both worlds.
  - **Oracle Database** can serve as the primary storage for transactional data, while **Elasticsearch** can be used for fast analysis and visualization of these data.
  - This approach enables efficient data processing with an emphasis on security, while at the same time providing flexibility in the analysis and searching of information.