

1. T MAJ-800 Healthtech : Data report

1.1. Summary

- 1. T MAJ-800 Healthtech : Data report
 - 1.1. Summary
 - 1.2. MongoDB
 - 1.2.1. Presentation
 - 1.2.1.1. Introduction
 - 1.2.1.2. Key Features : Document-oriented
 - 1.2.1.3. Advanced Functionality : Aggregation Framework
 - 1.2.1.4. Integration and Ecosystem : Programming Language Support
 - 1.2.1.5. Security and Compliance : Role-Based Access Control
 - 1.2.1.6. Conclusion
 - 1.2.2. Our case : IoT
 - 1.2.3. Installation
 - 1.2.3.1. Init MongoDB on Ubuntu
 - 1.2.3.2. Add security to access the MongoDB server
 - 1.3. Calculations
 - 1.3.1. Spot estimation
 - 1.4. Costs
 - 1.5. Conclusion

1.2. MongoDB

1.2.1. Presentation

1.2.1.1. Introduction

1. MongoDB, a document-oriented NoSQL database, revolutionizes data storage and management with its flexible and scalable architecture.
2. Developed by MongoDB Inc., it offers a dynamic platform for building modern applications that demand high performance, flexibility, and real-time data access.
3. MongoDB is an open source solution and can be installed on Linux, Windows and Mac.

1.2.1.2. Key Features : Document-oriented

MongoDB stores data in flexible, JSON-like documents, making it easy to represent complex structures and hierarchical relationships, which allows several advantages like:

- **Scalability:** Its horizontal scaling capabilities allow seamless distribution of data across multiple servers, ensuring optimal performance as the dataset grows.
- **High Performance:** MongoDB utilizes in-memory computing and indexes for efficient query processing, enabling lightning-fast data retrieval.
- **Flexible Schema:** With a dynamic schema, MongoDB allows on-the-fly updates to data models, making it adaptable to evolving application requirements.

- **Replication and High Availability:** MongoDB provides automatic replication, ensuring data redundancy and fault tolerance for continuous operation.
- **Geospatial Capabilities:** Built-in support for geospatial data enables location-based queries, making it ideal for applications involving maps or spatial analysis.

1.2.1.3. Advanced Functionality : Aggregation Framework

MongoDB's powerful aggregation pipeline allows complex data processing, including filtering, grouping, and transforming data, which allows several advantages like :

- **Full-text Search:** It offers efficient text indexing and querying capabilities, enabling users to perform complex searches across large volumes of textual data.
- **Change Streams:** MongoDB supports real-time data processing with change streams, allowing applications to react to database changes in real-time.
- **GridFS:** MongoDB's GridFS provides a way to store and retrieve large files efficiently, making it suitable for handling multimedia content.

1.2.1.4. Integration and Ecosystem : Programming Language Support

MongoDB supports a wide range of programming languages, including Java, Python, Node.js, and many others, which allows several advantages like :

- **Cloud Integration:** It seamlessly integrates with popular cloud platforms, such as AWS, Azure, and Google Cloud, allowing for easy deployment and management.
- **MongoDB Atlas:** MongoDB's fully managed database service offers automated provisioning, scaling, and monitoring, simplifying the deployment process.
- **Community and Support:** MongoDB boasts a vibrant community and provides extensive documentation, tutorials, and forums for developers.

1.2.1.5. Security and Compliance : Role-Based Access Control

MongoDB allows fine-grained access control, ensuring data security by defining user roles and permissions.

- **Encryption:** It provides encryption at rest and in transit, safeguarding sensitive data from unauthorized access.
- **Auditing and Compliance:** MongoDB offers auditing capabilities to track database activities, aiding compliance with industry regulations such as GDPR.

1.2.1.6. Conclusion

MongoDB's robust features, scalability, and flexibility make it an excellent choice for modern applications requiring high-performance data storage and retrieval. Whether you're building a web application, IoT system, or analytical platform, MongoDB empowers developers to handle complex data requirements efficiently and unleash the true potential of their applications.

1.2.2. Our case : IoT

MongoDB is a great choice for IoT, like on our case, due to its various features and capabilities :

- **Document-oriented:** MongoDB stores data in flexible, JSON-like documents. This allows for easy representation of complex structures and hierarchical relationships, which is beneficial for IoT systems that deal with diverse and interconnected data.
- **Scalability:** MongoDB offers horizontal scaling capabilities, enabling the distribution of data across multiple servers. This ensures optimal performance as the dataset grows, which is crucial for handling the large volumes of data generated by IoT devices.
- **High Performance:** MongoDB utilizes in-memory computing and indexes, resulting in efficient query processing. This translates to lightning-fast data retrieval, which is essential for real-time IoT applications that require quick access to data.
- **Flexible Schema:** MongoDB's dynamic schema allows for on-the-fly updates to data models. This adaptability is valuable in IoT systems where data formats and structures may evolve over time.
- **Replication and High Availability:** MongoDB provides automatic replication, ensuring data redundancy and fault tolerance. This feature guarantees continuous operation even in the event of failures, which is critical for the reliability of IoT systems.
- **Geospatial Capabilities:** MongoDB has built-in support for geospatial data, making it ideal for IoT applications involving location-based queries or spatial analysis. This is valuable for scenarios such as asset tracking or smart city systems.
- **Aggregation Framework:** MongoDB's powerful aggregation pipeline enables complex data processing, including filtering, grouping, and transforming data. This functionality is beneficial for IoT systems that require advanced data analysis and decision-making.
- **Integration and Ecosystem:** MongoDB supports a wide range of programming languages and seamlessly integrates with popular cloud platforms. This facilitates easy integration of IoT devices and enables efficient deployment, management, and scalability in the cloud.
- **Security and Compliance:** MongoDB offers role-based access control, encryption at rest and in transit, as well as auditing capabilities. These features ensure data security and compliance with industry regulations, making MongoDB a reliable choice for IoT systems dealing with sensitive information.

In conclusion, MongoDB's robust features, scalability, flexibility, and security make it an excellent database solution for IoT systems. It empowers developers to efficiently handle the complexities of IoT data, enabling them to build high-performance applications and unlock the full potential of their IoT deployments.

1.2.3. Installation

1.2.3.1. Init MongoDB on Ubuntu

Install MongoDB: Open a terminal on an Ubuntu virtual machine and execute the following commands to install MongoDB:

```
sudo apt update
sudo apt install mongodb
```

Start the MongoDB service: MongoDB should start automatically after installation. Confirm the status of the service by running:

```
sudo systemctl status mongodb
```

If the service is not running, start it with the following command:

```
sudo systemctl start mongod
```

Enable MongoDB to start on boot: MongoDB will start automatically whenever the virtual machine boots up, to automate the process, enable the service using:

```
sudo systemctl enable mongod
```

Rule to access the MongoDB port: Add a rule to allow incoming connections on port 27017 :

```
sudo ufw allow 27017
```

1.2.3.2. Add security to access the MongoDB server

Enable Authentication: MongoDB supports authentication to ensure that only authorized users can access the database. To enable authentication, you need to configure MongoDB to require authentication for all connections.

- Open the MongoDB configuration file in a text editor: `sudo nano /etc/mongodb.conf`
- Uncomment the `#security:` line and add the following lines below it:

```
security:  
  authorization: enabled
```

Create an Administrative User: Next, you'll create an administrative user who will have full privileges to manage the MongoDB server.

- Connect to MongoDB using the MongoDB shell: `mongo`
- Switch to the admin database: `use admin`
- Create the administrative user with a username and password:

```
db.createUser({  
  user: "User_Name",  
  pwd: "Password",
```

```
roles: [ { role: "root", db: "admin" } ]  
})
```

With 'User_Name' the admin user name and 'Password' his password.

Restart MongoDB: After enabling authentication and creating the administrative user, restart the MongoDB service for the changes to take effect.

```
sudo systemctl restart mongod
```

Configure Network Access: By default, MongoDB listens on all available network interfaces. To enhance security, you can bind MongoDB to specific IP addresses or network interfaces.

- Open the MongoDB configuration file:

```
sudo nano /etc/mongodb.conf
```

- Find the **net** section and add the following line:

```
bindIp: Server_IP
```

With 'Server_IP' the server IP adresse (ex : 127.0.0.1).

Restart MongoDB: Restart the MongoDB service to apply the network configuration changes.

```
sudo systemctl restart mongod
```

1.3. Calculations

Following the part on [MongoDB installation](#), we can define three main tasks:

- Creation of the VM
- Initialization of MongoDB
- Securing the server

1.3.1. Spot estimation

Creation of the VM :

- Developement : 4H
- Test : 0H

- Correctifs : 0H

Initialization of MongoDB :

- Developement : 2.5H
- Test : 0.5H

Securing the server :

- Developement : 1H
- Test : 0.5H

The deployment of the database should take a total of 8.5 hours, assuming the server is operational.

1.4. Costs

There is three points to predictit the cost for the deployment of the database :

- The server / stockage capacity : It depend on the [solution](#) that take the client.
- The solution : MongoDB is a free open source solution.
- The work made for the deployment : 8.5 hours of work

The total cost for the deployment of the database is the price of 8.5 hours of work.

1.5. Conclusion

MongoDB's rich feature set, scalability, cost-effectiveness, and security measures make it the best-suited database solution for an IoT project. By leveraging its capabilities, organizations can efficiently handle the vast amounts of data generated by IoT devices, adapt to changing data structures, and scale their systems while keeping costs under control.