Integrating IntelMQ with MongoDB: A Detailed Report

1. Introduction

IntelMQ is a robust framework for collecting and processing threat intelligence data. For efficient storage and management of this data, integrating IntelMQ with a suitable database system is essential. This report details the process of integrating IntelMQ with MongoDB, discusses the suitability of MongoDB for data-intensive applications, and compares MongoDB with PostgreSQL and SQLite.

2. MongoDB Overview

MongoDB is a NoSQL database known for its scalability, flexibility, and performance. It stores data in JSON-like documents, making it a good fit for handling unstructured data. MongoDB is widely used in applications requiring real-time analytics, content management, and IoT data management.

3. IntelMQ Overview

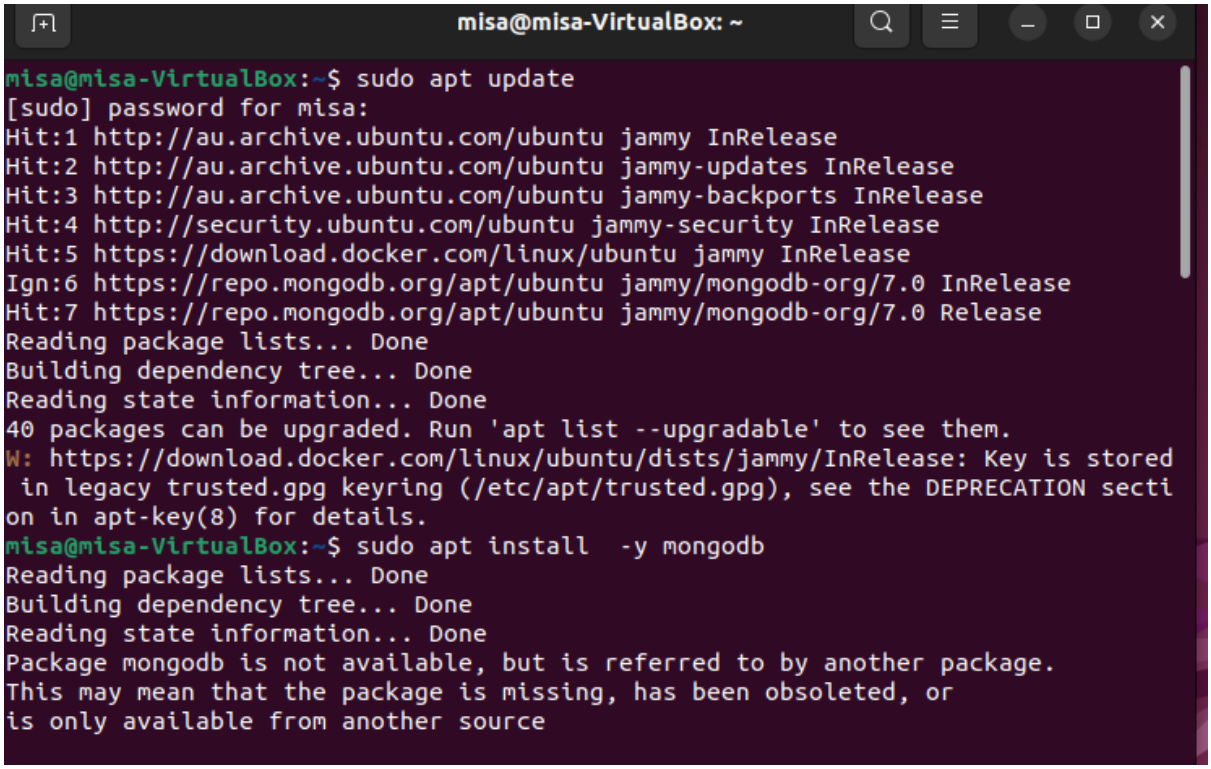
IntelMQ is an open-source solution for handling threat intelligence data. It consists of bots that collect, process, and store threat intelligence data. Each bot performs specific tasks, such as data collection, parsing, and storage. The output bots are responsible for sending the processed data to various storage systems, including databases.

4. Process of Integration

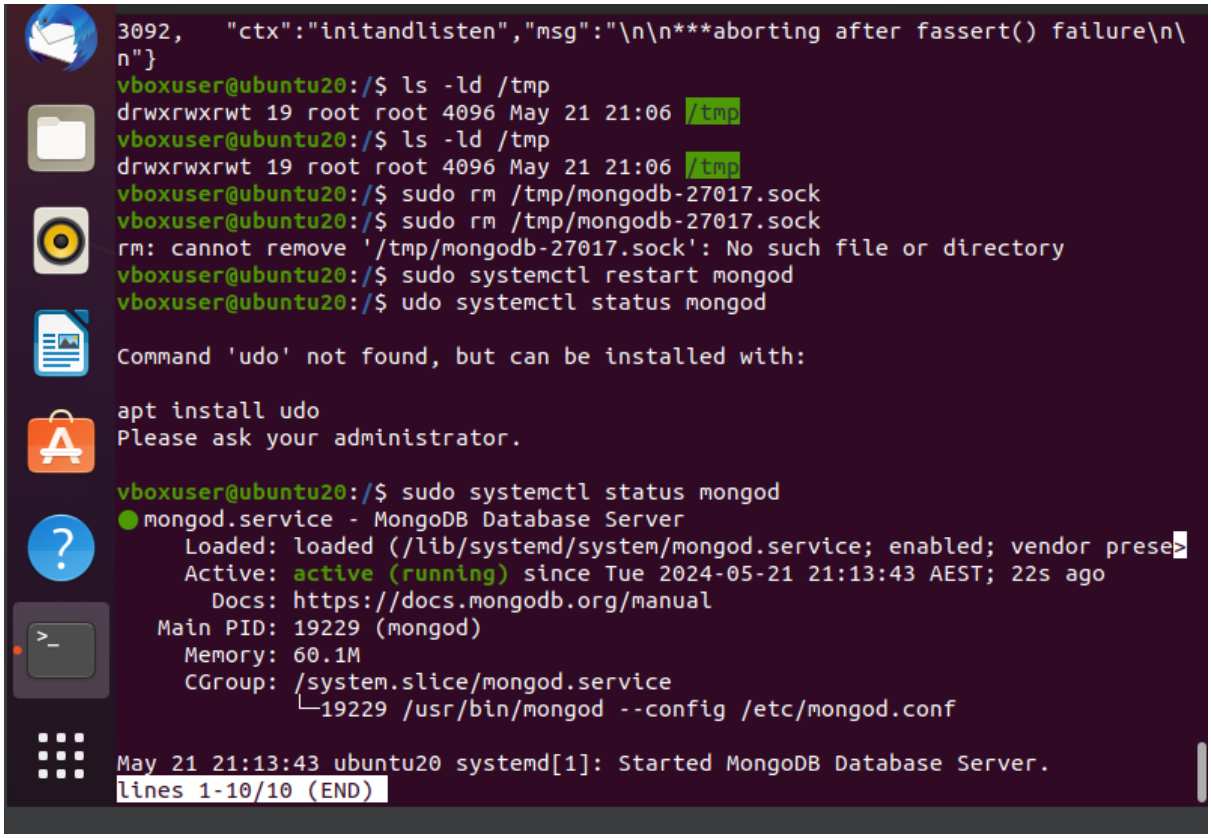
Step 1: Setting Up MongoDB on Ubuntu 20.04

Note :the GPG Key will be different for different ubuntu versions and can be obtained from simple google search.

- Import the MongoDB GPG Key:
  ```
  wget -qO - https://www.mongodb.org/static/pgp/server-4.4.asc | sudo apt-key add –
  ```
- Create the MongoDB List File:
  ```
  echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/4.4 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-4.4.list
  ```
- Update the Package Database:
  ```
  sudo apt update
  ```

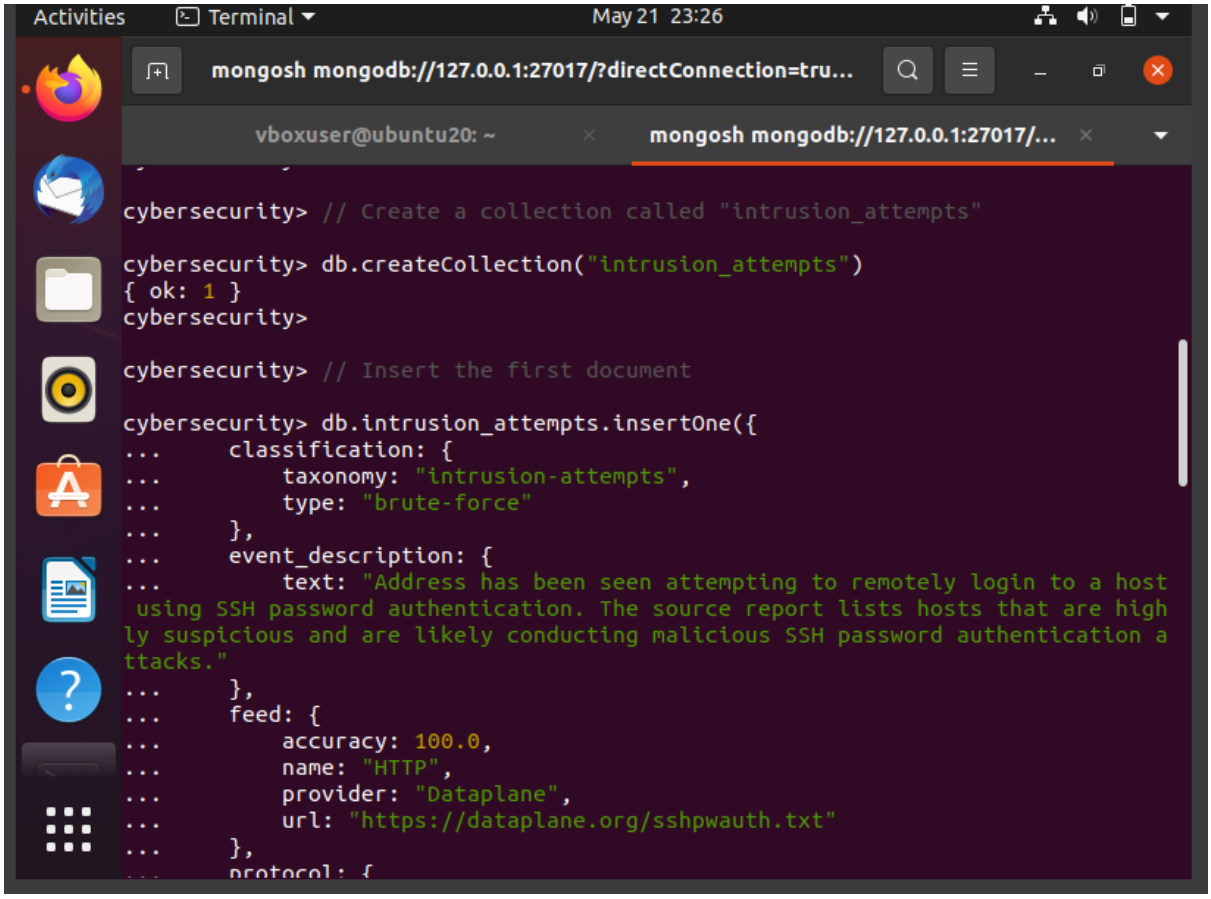


- Install MongoDB Packages:
  `sudo apt install -y mongodb-org`
- Start and Enable MongoDB Service:
  `sudo systemctl start mongod`
  `sudo systemctl enable mongod`
- Verify MongoDB Installation:
  `mongod –version`
  `sudo systemctl status mongod`
- you should be able to see monod Active and running as shown below.

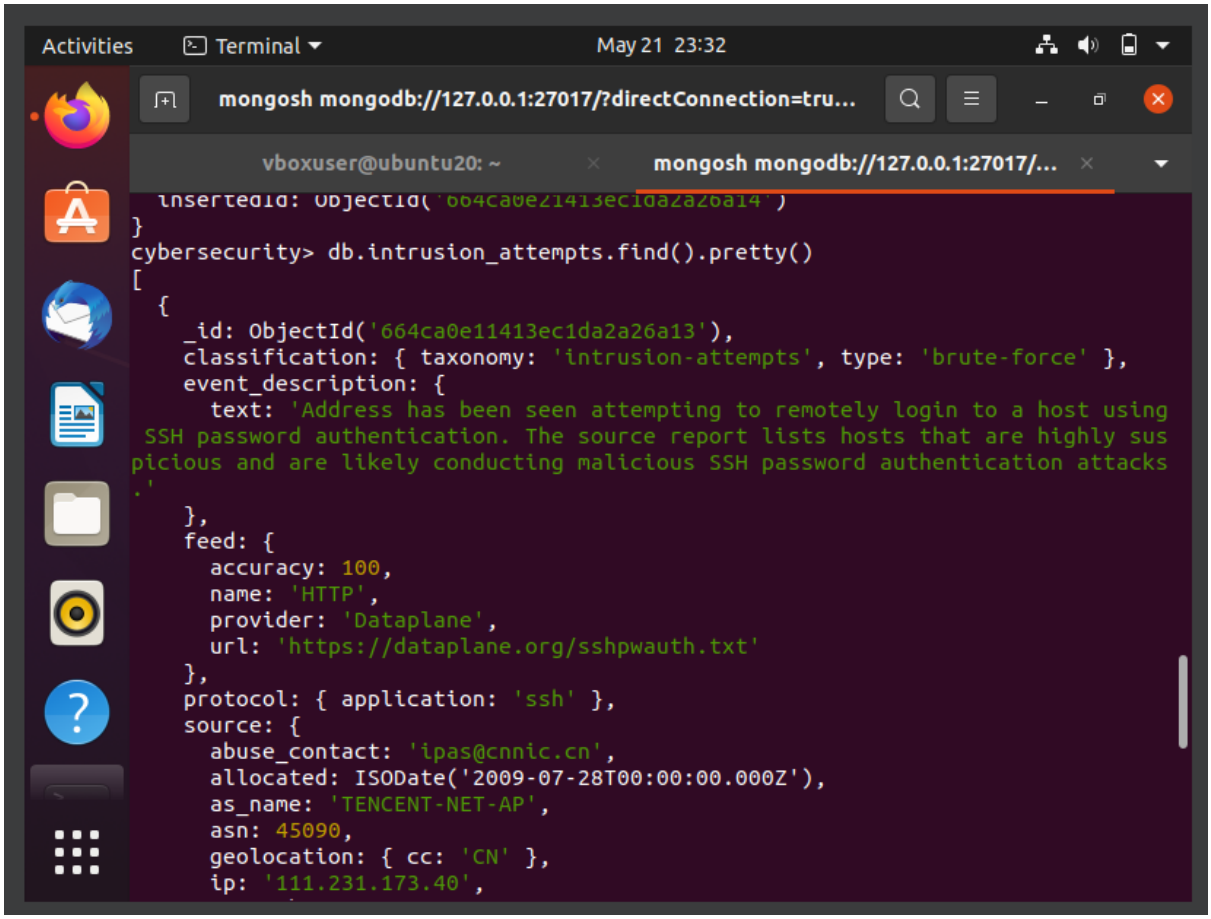


- Create a collection and insert a document:

```
db.intrusion_attempts.insertOne({
classification_taxonomy: "intrusion-attempts",
classification_type: "brute-force",
event_description_text: "Address has been seen attempting to remotely login to a host using SSH password authentication. The source report lists hosts that are highly suspicious and are likely conducting malicious SSH password authentication attacks.",
feed_accuracy: 100.0,
feed_name: "HTTP",
feed_provider: "Dataplane",
feed_url: "https://dataplane.org/sshpwauth.txt",
protocol_application: "ssh",
source_abuse_contact: "qcloud_net_duty@tencent.com",
source_allocated: "1994-03-25T00:00:00+00:00",
source_as_name: "TENCENT-NET-AP-CN",
source_asn: 132203,
source_geolocation_cc: "SG",
source_ip: "170.106.114.187",
source_network: "170.106.114.0/23",
source_registry: "APNIC",
time_observation: "2024-04-13T18:41:46+00:00",
time_source: "2024-04-13T17:59:54+00:00"
})
```



- Verify the inserted document:
  `db.intrusion_attempts.find().pretty()`

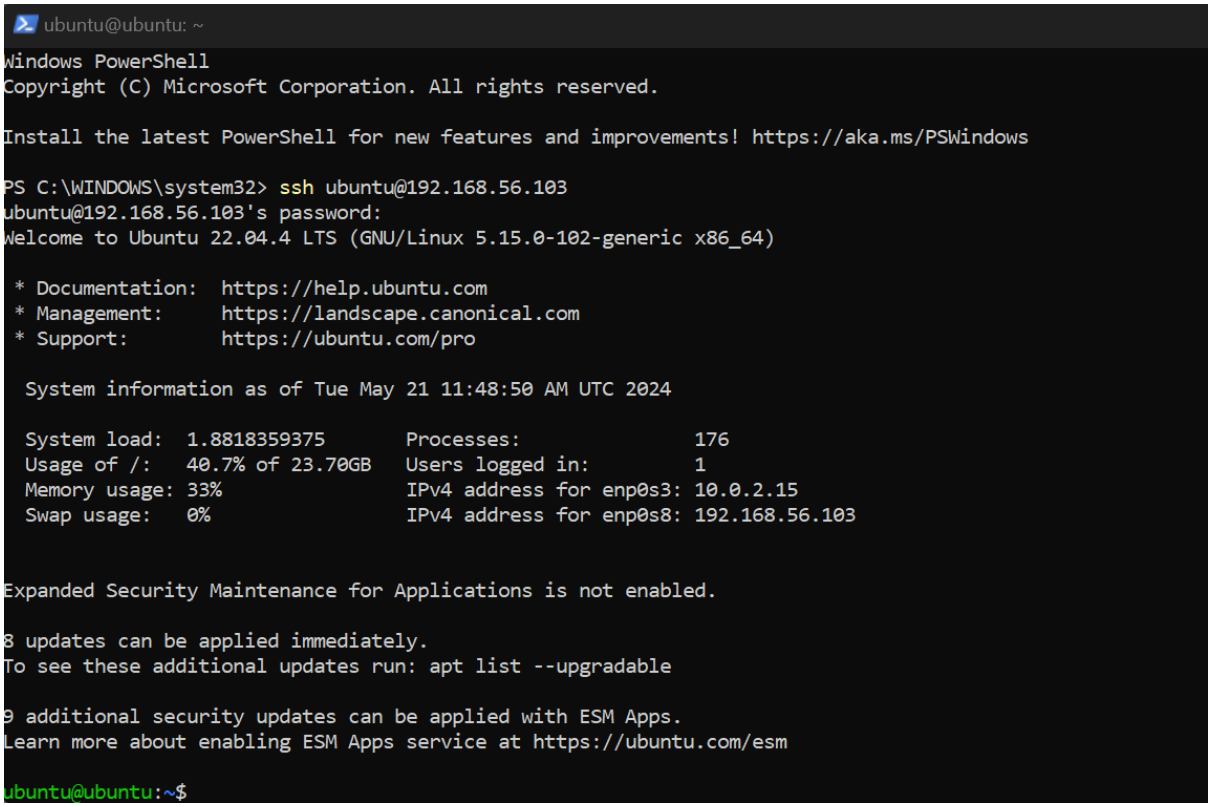


Step 2: Setting Up IntelMQ

- Install IntelMQ:
  Follow the IntelMQ installation guide from the official documentation:
  https://intelmq.readthedocs.io/en/latest/user/installation.html

- For our use case, we have already got intelmq virtual machine configurated for our project.

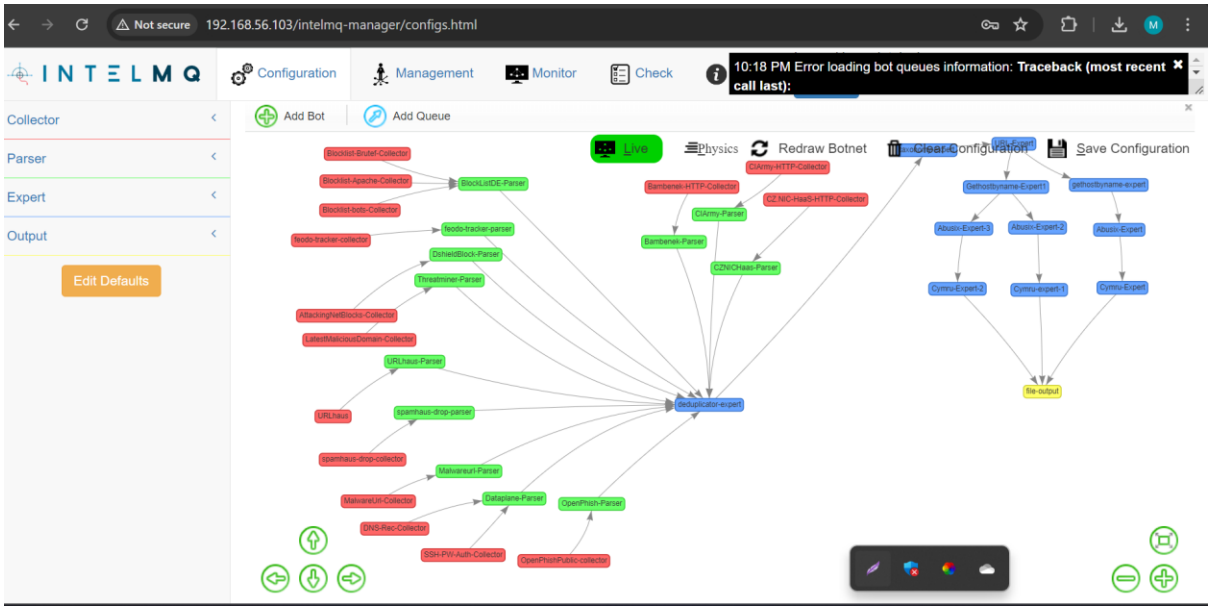
- Connect to IntelMQ VM:
  Use the ssh command to connect to your IntelMQ VM. Replace username with your actual username on the IntelMQ VM.
  powershell
  ```
  ssh username@192.168.56.103
  ```
- Enter the password when prompted.



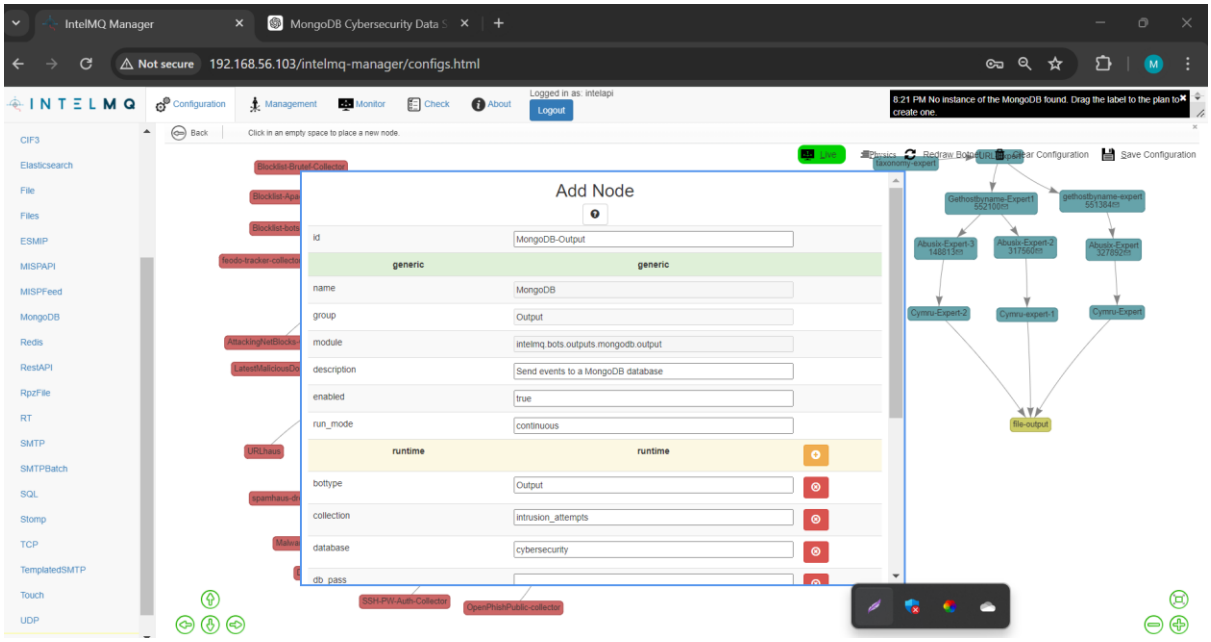
- To open intelmq manager, type this command

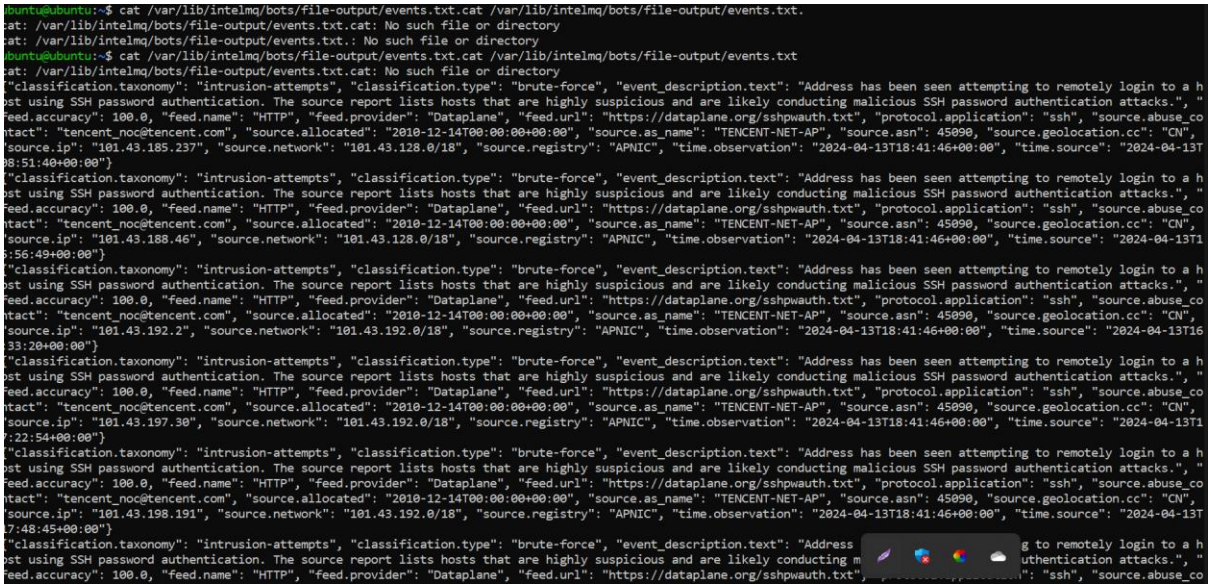
  http://<ip of intelmq>/intelmq-manager/configs.html

- IntelMQ Manager provides a graphical user interface (GUI) to easily configure, manage, and monitor the data flow within IntelMQ. This makes the process more intuitive and accessible compared to manually editing configuration files.
  Login credential:
  Username : intelapi
  Password : intelapi



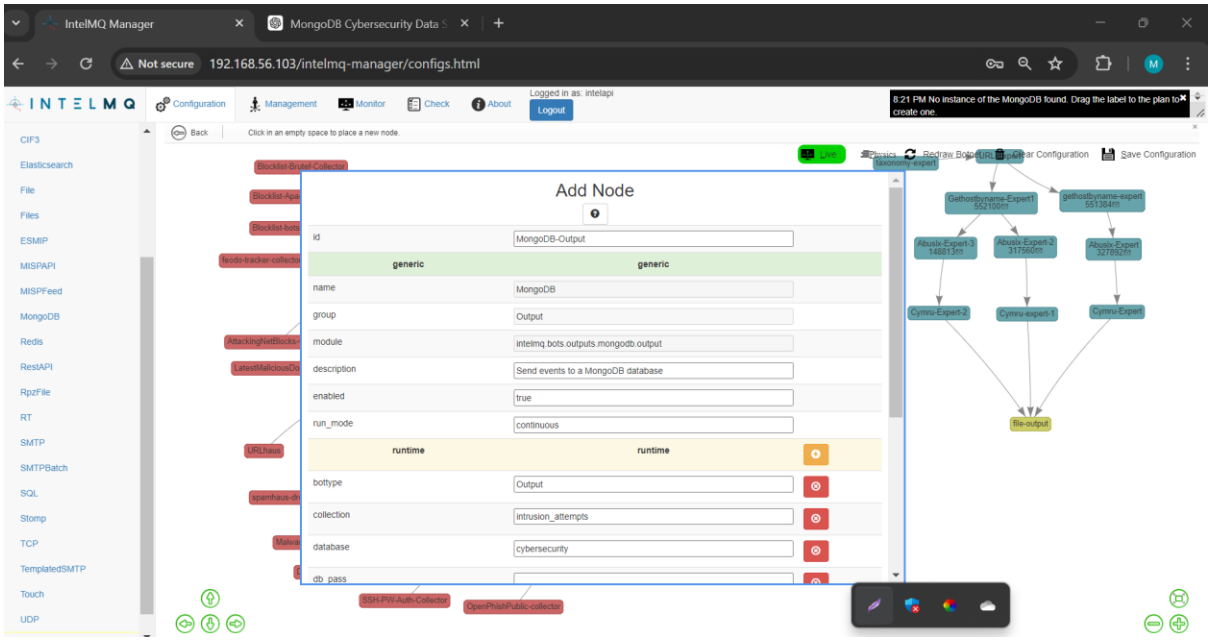
- Current Setup: File Output Configuration
  Currently, the output data from IntelMQ is being logged into a file. Here is an example of the raw data logged inside the file output:



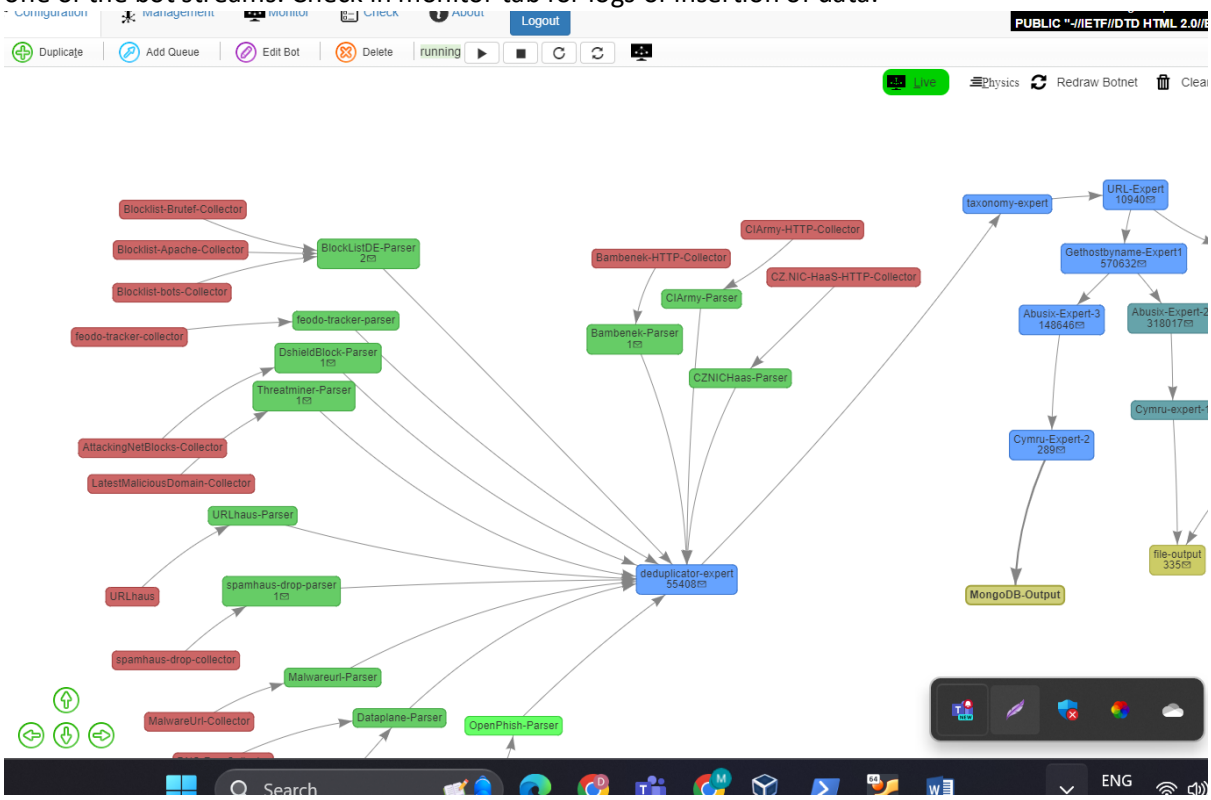


Drag and drop the mongodb output bot and add the configurations as mentioned below. Make sure to add the host ip address in host column to connect it with databse server. Alternatively you can add the belwo given confiuration in YAML file.

```
MongoDBOutput:
  bot_id: mongodb-output
  description: Store events in MongoDB database
  enabled: true
  group: Output
  groupname: outputs
  module: intelmq.bots.outputs.mongodb.output
  parameters:
    database: intelmqdb
    collection: events
    host: mongodb-vm-ip-address
    port: 27017
  run_mode: continuous
```



Add queue to connect Mongodb output bot with Cymru-Expert-2 bot. Wait for data to pass into the one of the bot streams. Check in monitor tab for logs of insertion of data.



5. MongoDB Suitability for IntelMQ Data Storage

Advantages:

Flexible Schema: Easily handles varied data formats without requiring schema changes.

High Performance: Efficient for read/write operations with large datasets.

Scalability: Supports horizontal scaling to accommodate growing data volumes.

Disadvantages:

Operational Complexity: Requires careful indexing and query optimization for performance.

Resource Intensive: Higher memory and storage usage compared to some other databases.

6. Successes:

Installation and Configuration: Successfully installed MongoDB on the dedicated Ubuntu virtual machine and configured it to run as a service.

Database Creation: Created a MongoDB database named intelmq and initialized a collection named intrusion_attempts for storing threat intelligence data.

Bot Integration: Configured the MongoDB Output Bot through the IntelMQ Manager interface, specifying the MongoDB URI, database name (intelmq), collection name (intrusion_attempts), and other necessary parameters.

Bot Activation: Enabled and started the MongoDB Output Bot within the IntelMQ environment to facilitate data transmission to the MongoDB collection.

7. Challenges:

Network Issues: Encountered challenges with network connectivity between the IntelMQ server and the MongoDB server, potentially leading to communication errors or data transmission issues.

Firewall Configuration: Suspected firewall settings or network isolation issues on either the IntelMQ or MongoDB server side, hindering successful data transmission between the two systems.

Debugging Limitations: Lack of detailed logs or error messages from the MongoDB Output Bot within the IntelMQ environment makes troubleshooting and identifying the root cause of integration issues challenging.

Data Verification: Inability to verify whether threat intelligence data is successfully stored in the MongoDB collection due to integration issues and lack of comprehensive logging.

8. Next Steps:

Network Troubleshooting: Conduct thorough network troubleshooting to identify and resolve any connectivity issues between the IntelMQ and MongoDB servers, ensuring seamless communication.

Detailed Logging: Implement comprehensive logging mechanisms within the MongoDB Output Bot to capture detailed error messages and facilitate effective troubleshooting.

Data Validation: Develop a validation process to verify the successful storage of threat intelligence data in the MongoDB collection, ensuring data integrity and completeness.

9. Conclusion:

The integration of IntelMQ with MongoDB presents significant potential for enhancing the storage and management of threat intelligence data. Despite encountering challenges related to network connectivity and firewall configuration, successful installation, configuration, and initial setup of MongoDB were achieved. Additionally, the configuration of the MongoDB Output Bot within the IntelMQ environment demonstrated progress towards facilitating data transmission to the MongoDB collection.

Moving forward, resolving the identified network issues, and implementing comprehensive logging mechanisms are essential steps to ensure seamless integration and effective data transmission between IntelMQ and MongoDB. Additionally, establishing a robust validation process to verify the successful storage of threat intelligence data in the MongoDB collection will be crucial for ensuring data integrity and completeness.

Through collaborative efforts and diligent troubleshooting, we aim to overcome the existing challenges and fully leverage the capabilities of MongoDB for storing and managing threat intelligence data within the IntelMQ environment. Ultimately, the successful integration of IntelMQ with MongoDB will contribute to enhancing cybersecurity operations, enabling organizations to better detect, analyze, and respond to emerging threats.

References:

[1] [MongoDB, "Install MongoDB Community Edition on Ubuntu — MongoDB Manual," MongoDB, 2024. [Online]. Available: https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-ubuntu/. [Accessed: 17-May-2024].
[2] IntelMQ Team, "EventDB — IntelMQ 3.0.0 documentation," IntelMQ, 2024. [Online]. Available: https://intelmq.readthedocs.io/en/develop/user/eventdb.html. [Accessed: 17-May-2024].
[3] IntelMQ Team, "IntelMQ Tutorial: Lesson 3," GitHub, 2024. [Online]. Available: https://github.com/certtools/intelmq-tutorial/blob/master/lesson-3.md. [Accessed: 17-May-2024].
[4] IntelMQ Team, "intelmq/bots/outputs at develop," GitHub, 2024. [Online]. Available: https://github.com/certtools/intelmq/tree/develop/intelmq/bots/outputs. [Accessed: 17-May-2024].
[5] A. Kaplan, "IntelMQ: A Framework for CERTs for Collecting and Processing Security Feeds," FIRST, 2020. [Online]. Available: https://www.first.org/resources/papers/malaga20/PUBLIC-Aaron-Kaplan-IntelMQ-malaga-20200131.pdf. [Accessed: 17-May-2024].