**PG Certificate Program in**

**Software Engineering for Data Science, IIIT Hyderabad**

**Capstone Project: News Articles Classifier**

**Report**

**Git Repo: https://github.com/anudeepkalitkar/CapstoneProject**

**Anudeep Kalitkar**

**Naveen Reddy Chatlaparthi**

**Architecture**

Source1

Source2

**RSS Feed / REST APIs**

**Producer**

**Consumer**



**KAFKA Server**

**Front End News Classifier**

**Architecture Components**

1. Data Source:
   1. Rapid API
   2. Various RSS feed from TOI, NDTV, CNN, etc.
2. Data Selection:
   1. Data from the source is preprocessed and only a selective data attributes will be pushed the kafka server
3. Kafka Server:
   1. Kafka Producer, a python script which produces the processed data as a Kafka topic into Kafka cluster
   2. A Kafka cluster with single broker and a Zookeeper instance that manages Kafka cluster
   3. Kafka Consumer, a python script that consumes the data as a Json object from Kafka.
4. Mongo DB:
   1. MongoDB, a NoSQL database that stores the data
5. ML Modeling:
   1. Model selection
   2. Model Training
   3. Model prediction
6. Front-End End Point:
   1. Either a rest-api or complete front end web application for prediction

**Report 1**

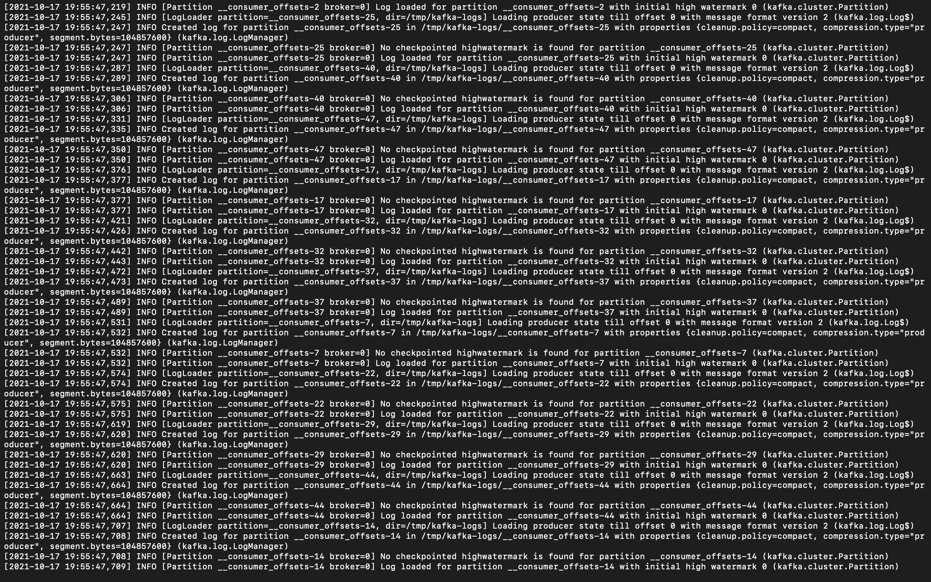
**Data Injestor:**

1. Rapid API:
   1. Rapid API has a huge collection of categorized NEWS data/information.
   2. It provides data in Json format with many attributes.
   3. Call to Rapid API is made once every 5 mins.
   4. Attributes consider for this project:
   5. Title, Summary, Link, Published date, Topic, Media link and CleanURL
2. RSS Feed:
   1. RSS Feed is a podcast, which has information about the published articles.
   2. RSS Feed is either HTML or XML based data
   3. Every other NEWS Publisher has their own RSS Feed page for various topics.
   4. We have considered RSS feeds from CNN, NDTV, and TOI.
   5. Calls to these RSS feeds are made once every 5 mins.
   6. Attributes consider for this project:
   7. Title, Summary, Link, Published date, Topic, Media link and CleanURL
3. Database:
   1. We have chosen MongoDB for its advantage of NOSQL Databases.
   2. MongoDB is a Document based database, which helps us to narrow down the errors for file upload.
   3. We are using json objects/ python dictionaries as data to insert into the database.
   4. As MongoDb provides a free cloud database, we are using it for ease of use.
4. Data handling:
   1. As we have data from various sources, we are preprocessing the data we acquire from the source to get the data we require.
   2. We have exception and error handlers in place to prevent the flow of the work stopping.
   3. We have used None, if we don't have the required data.
5. Kafka:
   1. We are using Apache Kafka for real-time data streaming with Producer and Consumer programs
   2. Faced a difficulty while testing our pipeline on windows, majorly starting a Kafka cluster in windows and hence chosen Ubuntu 20.04 Linux kernel on WSL 2 architecture for testing.

**Kafka Streaming**

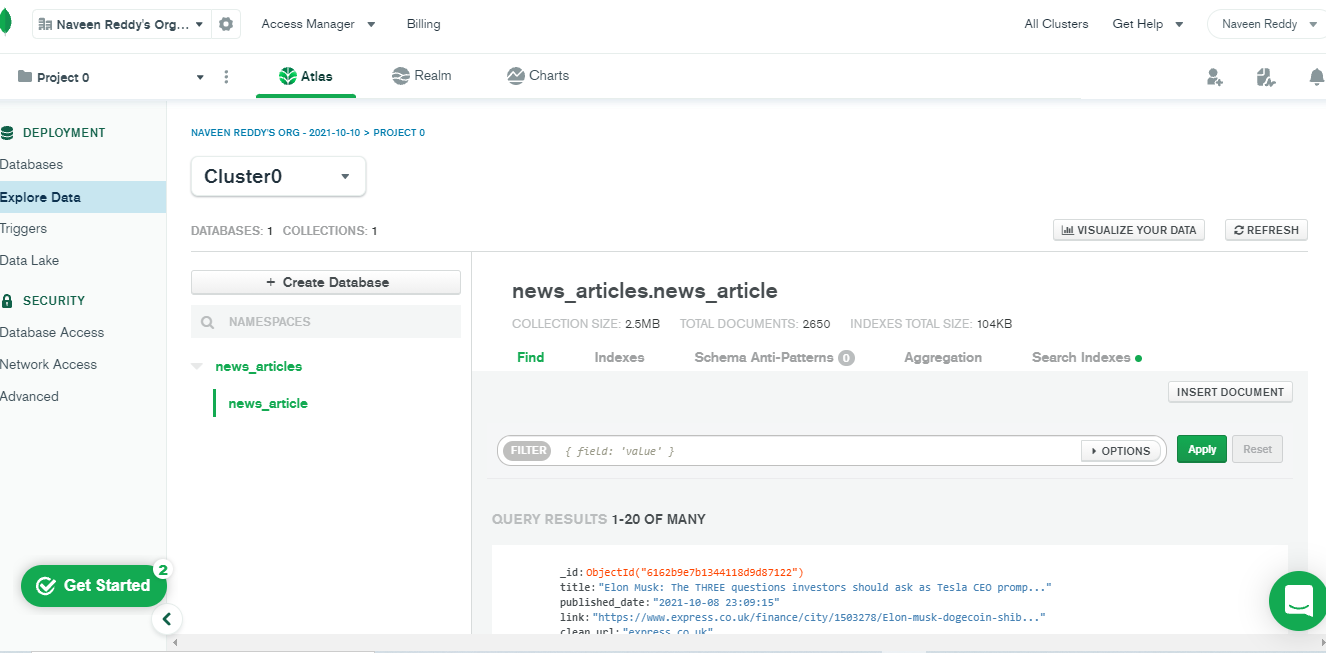
We have Zookeeper instance and Kafka Server running on localhost at ports 2181 and 9092 as shown in the images below.

****

****

.

**Online Mongo DB connection**



**Report 2**

1. Project Architecture Design Completed
2. ML Approach Documentation Completed
3. Developed Python script for Data preparation Completed
   1. Install and import packages mention in requirement.txt
4. Set up Kafka Completed
5. Setup MongoDB Completed
6. Initial Data Ingestion Python code development Completed
7. Data Preprocessing pipeline Completed
8. Build a ML Model Completed

**Challenges:**

1. Kafka setup
2. Kafka consumer and producer delayed response
3. Mongo DB Setup
4. ML model selection and prediction values.
5. RSS feed data segregation
6. Rapid API Hit count limit.