Youtube Trending Data Analysis

Using AWS ECS, ECR, ElasticSearch and Kibana

Submitted to:
Dr.Essam Mansour

COMP 6231
Distributed Systems Design

Ashu Kumar(40221569)
Vishnu Rameshbabu(40233562)
Benjamin Douglas Joseph David(40264251)
Ayyanar Kadalkani(40231399)
Hani Saravanan(40233005)
Jothi Basu LKV(40230416)

Introduction

Project Overview: Analyzing YouTube's trending videos to uncover viewer engagement and content popularity trends from 2020 to 2023.

Dataset Insights: Comprehensive collection of trending video data from YouTube. Structured in CSV format for ease of processing. Data spans across 11 countries, providing a global perspective.

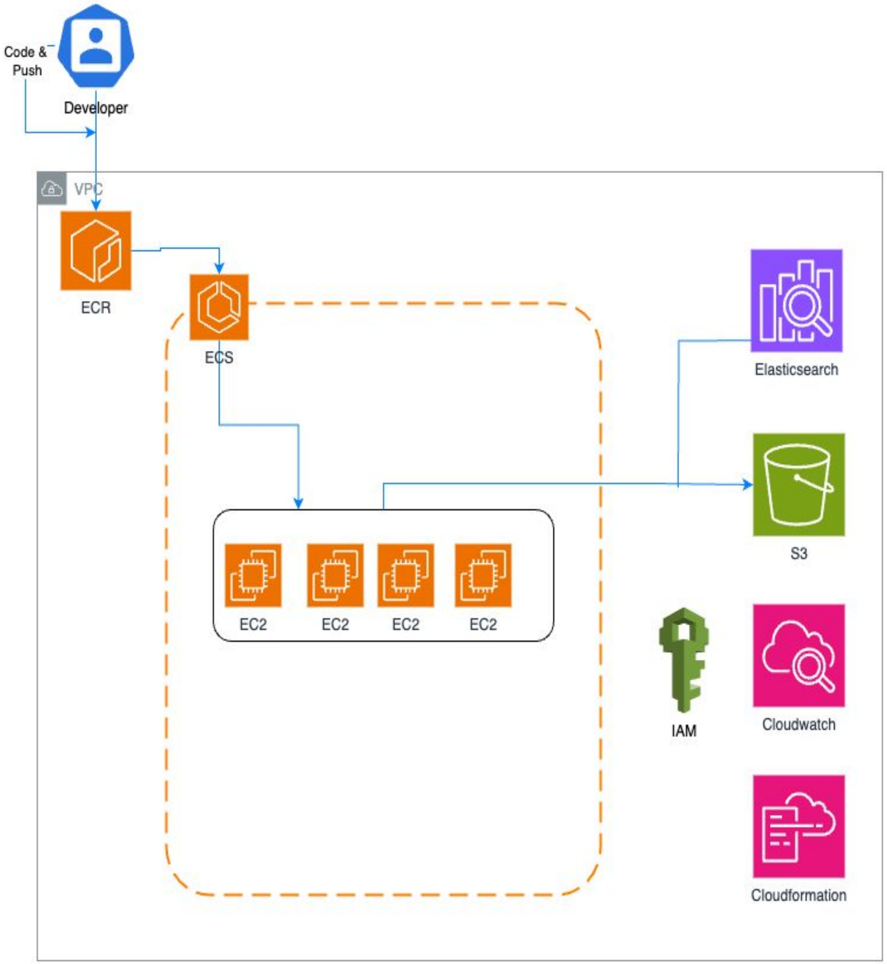
Key Analysis Conducted:

- **Trending Channel Analysis:** Monthly examination of the most viewed channels per country and year.
- **Category Preference Analysis:** Determination of the most and least favored video categories on a global scale.
- **Universal Trending Analysis:** Cross-country evaluation to spot videos trending in multiple regions.

Objective: To provide a deep-dive into the factors driving video popularity on YouTube. To assist content creators and marketers in strategizing based on data-driven insights.

Cloud Architecture

- **Developer Interaction:** Developers write and push code to the cloud infrastructure directly from their development environments.
- **Elastic Container Registry (ECR):** Stores Docker container images securely. Integrates with IAM for controlled access to images.
- **Elastic Container Service (ECS):** Manages the deployment of containers. Auto-scales the container instances as per the workload.
- **Amazon EC2 Instances:** Hosts and runs the containerized applications.
- **Elasticsearch Service:** Manages, searches, and analyzes large volumes of data quickly. Connects to S3 for data storage, ensuring durability and accessibility.
- **Amazon S3:** Provides object storage for various types of data. Works with Elasticsearch for storing and analyzing data.
- **IAM (Identity and Access Management):** Manages access to AWS services and resources securely. Controls who is authenticated (signed in) and authorized (has permissions).
- **CloudWatch:** Offers monitoring services and tracks application performance and operational health.
- **CloudFormation:** Enables the definition of infrastructure as code. Automates and orchestrates the setup of AWS resources.



System Architecture

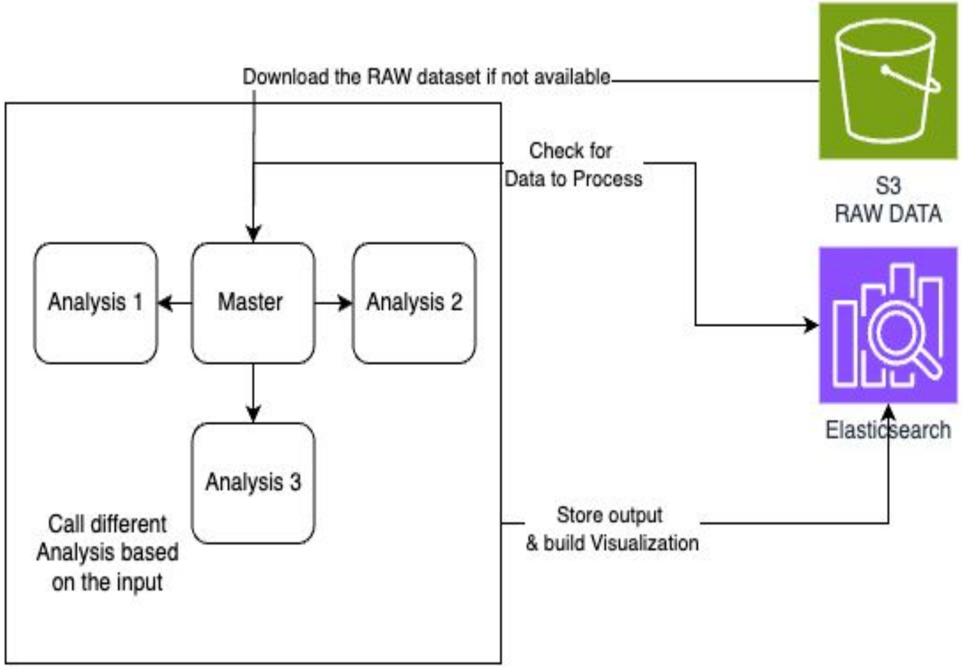
- An input script extracts the inputs from user and sends it as a JSON document to Elasticsearch.

- Master code synchronously checks elasticsearch for documents with the status as "**new**".

- Based on the data, it will call the respective analysis along with the year to analyze. And it will update status from "**new**" to "**processing**".

- Master code also synchronously checks documents every 5 mins if it's still in "**processing**". If it's present, then it will re-process the document again.

- Once the processing is done the individual analysis will push the resultant output and visualization will be done in Kibana.



Distributed Systems

- In elastic container service, we will be creating 4 EC2 instances to distribute the load.

- Since each analysis has complex data analysis and with the preprocessed dataset being over 1.2GB, for every analysis input one server takes control.

- This makes the data processing quicker than when it was with just one instance.

EC2
EC2
EC2
EC2

Fault Tolerance

- Implementation of advanced error-catching mechanisms in the processing code to detect and handle exceptions in real-time, ensuring minimal impact on the overall workflow.

- Deployment of a self-healing infrastructure that automatically detects server failures and swiftly replaces unhealthy instances, leveraging AWS's Auto Scaling and EC2 capabilities.

- Integration of AWS CloudWatch for continuous monitoring, coupled with AWS CloudFormation for quick recovery and redeployment, maintaining uninterrupted service and data integrity.

AutoScaling

- Auto-Scaling is one of the distributed system principles implemented in the project.

- Auto - Scaling is implemented when the EC2 instances have a CPU utilization of over 50%.

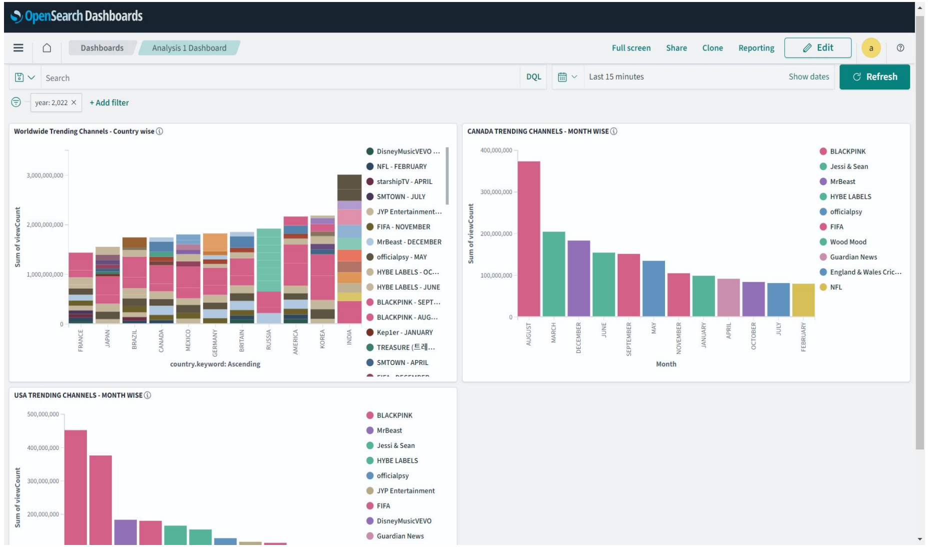
- When 50% CPU utilization is achieved, newer EC2 instances will be added to reduce the load with the current EC2 instances. This demonstrates the scalability principle.

Visualization

- Visualization for each analysis is implemented in **Kibana**, which works on Elasticsearch.

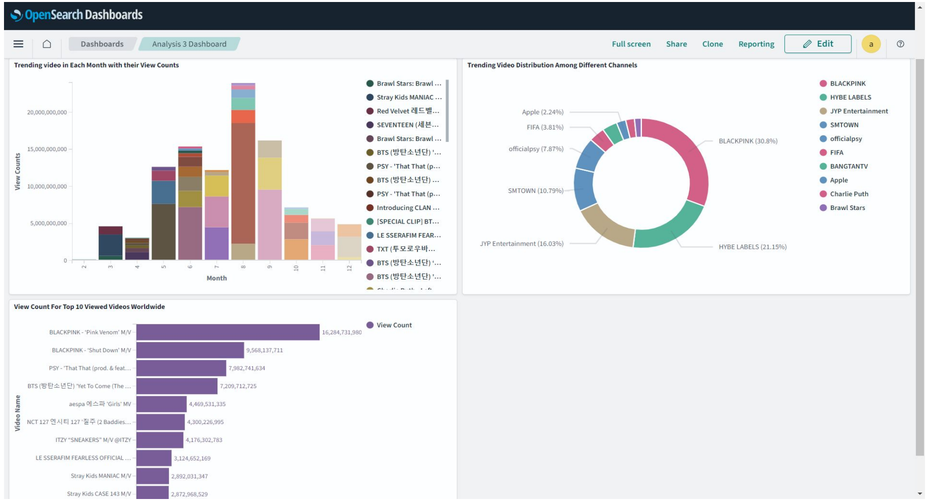
- Each analysis has an index name and using that, the indexes will be imported along with the data to Kibana.

- A dashboard is created for each analysis, which further has various visualizations within each one of them.



year: 2,022

Last 15 minutes



Conclusion

- **Successful Achievement of System Goals:** The system architecture robustly meets our design principles of efficiency, scalability, and fault tolerance.

- **Principles into Practice:** Demonstrated ability to handle large datasets with automated resilience against failures and errors.

- **Validation of Project Objectives:** Confirmed through rigorous testing and real-world application, our project objectives have been fully realized.