



Maharishi International University

Big Data Technology

Final Project: Twitter Data Analysis

Presented By:

- Kidus Mamuye Tekeste
ID: 612361

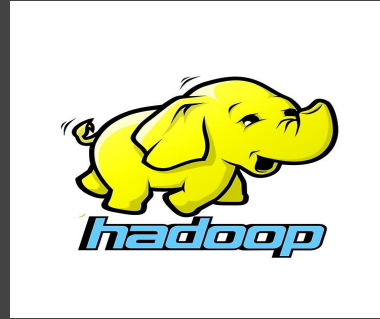
Tue Sep 27, 2022

What the project do:

- Twitter API Analysis
- Consumes Twitter API
- Publishes those tweet data to Kafka
- Zookeeper manages the broker and the cluster at which Kafka is running on
- Apache Spark Streaming consumes the data and using SQL module it integrate with Hive to store those tweet data on Hadoop Cluster
- Hive also exports those data as CSV data to our warehouse storages
- Using Apache Spark SQL and Hive module we read those data from hive database and analyse the data and displays them
- MySQL is used behind the scene as a local metastore for hive and spark integration for storing metadata information about the tweets

Technologies Used for This Project

- Zookeeper
- Kafka
- Apache Spark
- Apache Hive
- MySQL
- Hadoop



Spec of my Machine(Environment)

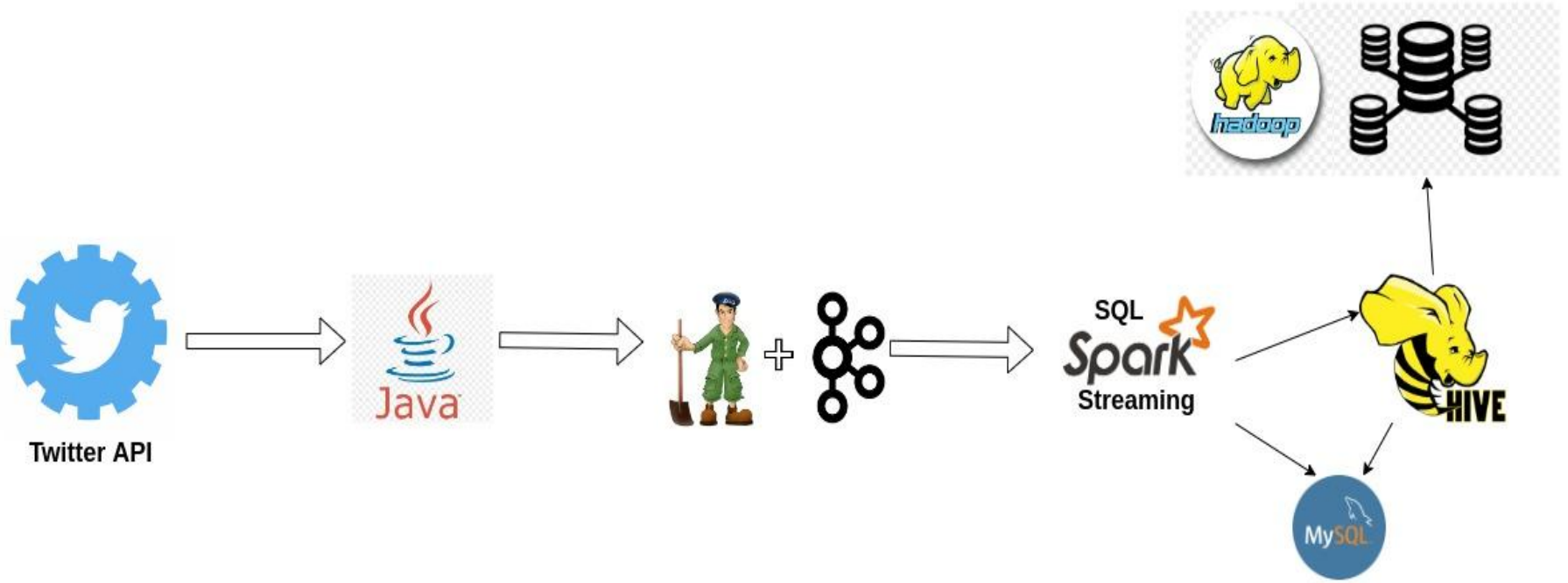
- OS: Linux Mint
- RAM: 24GB
- Storage: 512GB SSD



System info

Operating System	Linux Mint 20.2 Cinnamon
Cinnamon Version	5.0.7
Linux Kernel	5.15.0-48-generic
Processor	11th Gen Intel® Core™ i7-1165G7 @ 2.80GHz × 4
Memory	23.2 GiB
Hard Drive	512.1 GB
Graphics Card	02.0 VGA compatible controller
<input type="button" value="Upload system information"/>	

Architecture of Project



```
public class KafkaProducer {
```

```
public static void main(String[] args) throws InterruptedException {
```

```
ConfigurationBuilder configBuilder = new ConfigurationBuilder();  
configBuilder.setDebugEnabled(true)  
    .setOAuthConsumerKey(CONSUMER_KEY)  
    .setOAuthConsumerSecret(CONSUMER_SECRET)  
    .setOAuthAccessToken(ACCESS_TOKEN)  
    .setOAuthAccessTokenSecret(ACCESS_TOKEN_SECRET);
```

Twitter Cred. Config.

```
TwitterStream twitterStream = new TwitterStreamFactory(configBuilder.build()).getInstance();  
LinkedBlockingQueue<Status> tweetQueues = new LinkedBlockingQueue<>();  
twitterStream.addListener(new TwitterStatusListener(tweetQueues));  
twitterStream.filter(new FilterQuery().track(TWITTER_HASH_TAGS));
```

Twitter Topics

```
Properties kafkaProp = new Properties();  
kafkaProp.put("metadata.broker.list", KAFKA_BOOTSTRAP_SERVERS);  
kafkaProp.put("bootstrap.servers", KAFKA_BOOTSTRAP_SERVERS);  
kafkaProp.put("client.id", KAFKA_CLIENT_ID);
```

```
kafkaProp.put("key.serializer", "org.apache.kafka.common.serialization.StringSerializer");  
kafkaProp.put("value.serializer", "org.apache.kafka.common.serialization.StringSerializer");
```

```
int i = 0;  
for (;;) {  
    Status tweets = tweetQueues.poll();  
    if (tweets == null) {  
        Thread.sleep(500);  
    } else {  
        String msg = twitterToString(tweets);  
        org.apache.kafka.clients.producer.KafkaProducer kafkaProducer =  
            new org.apache.kafka.clients.producer.KafkaProducer(kafkaProp);  
        kafkaProducer.send(new ProducerRecord<>(TOPIC_NAME, String.valueOf(i++), msg),  
            new KafkaCallBack(System.currentTimeMillis(), String.valueOf(i++), msg));  
    }  
}
```

Kafka Producer

- Twitter Credential Configs
- Kafka Producer Configs

```
public class KafkaConsumer {  
    public static void main(String[] args) throws StreamingQueryException {
```

```
        SparkSession spark = SparkSession.builder()  
            .appName("Spark Kafka Integration Structured Streaming")  
            .config("hive.metastore.uris", THRIFT_URL)  
            .master("local[*]")  
            .enableHiveSupport()  
            .getOrCreate();
```

```
        Dataset<Row> ds = spark.readStream().format("kafka")  
            .option("kafka.bootstrap.servers", KAFKA_BOOTSTRAP_SERVERS)  
            .option("subscribe", TOPIC_NAME)  
            .load();
```

```
        Dataset<Row> lines = ds.selectExpr("CAST(value AS STRING)");  
        Dataset<Row> dataset = processTwitterDataSet(lines);
```

```
        dataset.writeStream()  
            .foreachBatch((rowDataset, aLong) -> rowDataset  
                .write()  
                .mode(SaveMode.Append)  
                .insertInto(TABLE_NAME))  
            .option("spark.sql.streaming.checkpointLocation", WAREHOUSE_DIR)  
            .start()  
            .awaitTermination();
```

Persists into
Hive Database

```
        StreamingQuery query = dataset.writeStream().outputMode("append").format("console").start();  
        query.awaitTermination();
```

Console Print

```
        dataset  
            .writeStream()  
            .format("csv")  
            .outputMode("append")  
            .option("path", WAREHOUSE_DIR)  
            .option("checkpointLocation", WAREHOUSE_DIR)  
            .start()  
            .awaitTermination();
```

Persists them as CSV file format

Kafka Consumer

- Spark Hive Integration
- Kafka Subscription Configs
- Spark Persist to Hive Warehouse
- Spark Exports to Hadoop HDFS as CSV

```

> public class SparkSQLHive {
>     public static void main(String[] args) {

        final SparkConf sparkConf = new SparkConf();
        sparkConf.setMaster("local[*]");
        sparkConf.set("hive.metastore.uris", THRIFT_URL);

        SparkSession sparkSession = SparkSession
            .builder()
            .config(sparkConf)
            .appName("Java Spark Hive Example")
            .config("spark.sql.warehouse.dir", WAREHOUSE_DIR)
            .enableHiveSupport()
            .getOrCreate();

        sparkSession.sql(USE_DEFAULT_SQL);

        Dataset<Row> rowDataset = sparkSession.sql(LOAD_ALL_TWEETS_SQL);
        rowDataset.show();

        rowDataset = sparkSession.sql(ADVANCED_SQL);
        rowDataset.show();

    }
}

```

Spark SQL Hive

- Spark Hive Integration
- Loading data from Hive Warehouse
 - Select All Query
 - Advanced Query


```

public class Utils {

    // SQL
    public static final String USE_DEFAULT_SQL = "USE default";
    public static final String LOAD_ALL_TWEETS_SQL = "SELECT * FROM tweets";
    public static final String ADVANCED_SQL = "SELECT ScreenName, SUM(default.tweets.favouritescount) AS TotalFavorites FROM";

    // Spark - Hive
    public static final String TABLE_NAME = "Tweets";
    public static final String WAREHOUSE_DIR = "/home/kidusmt/Desktop/BDT-FinalProject/apache-hive-3.1.2-bin/warehouse";
    public static final String THRIFT_URL = "thrift://localhost:9083";

    // Twitter
    protected static final String[] TWITTER_HASH_TAGS = {"#influencer", "#tbt", "#love", "#competition"};
    public static final String CONSUMER_KEY = " ";
    public static final String CONSUMER_SECRET = " ";
    public static final String ACCESS_TOKEN = " ";
    public static final String ACCESS_TOKEN_SECRET = " ";

    // Kafka
    public static final String TOPIC_NAME = "Tweets";
    public static final String KAFKA_BOOTSTRAP_SERVERS = "localhost:9092";
    public static final String KAFKA_CLIENT_ID = "TweetProducer";

    // Data variables
    public static final String createdAt = "createdAt";
    public static final String UserName = "UserName";
    public static final String ScreenName = "ScreenName";
    public static final String FollowersCount = "FollowersCount";
    public static final String FriendsCount = "FriendsCount";
    public static final String FavouritesCount = "FavouritesCount";
    public static final String Location = "Location";
    public static final String RetweetCount = "RetweetCount";
    public static final String FavoriteCount = "FavoriteCount";
    public static final String Lang = "Lang";
    public static final String Source = "Source";

    public static String getLocation(String loc) {
        if (loc == null)
            return "null";
        else return loc.split(";")[0];
    }
}

```

Twitter Hash tag Topics

Utils

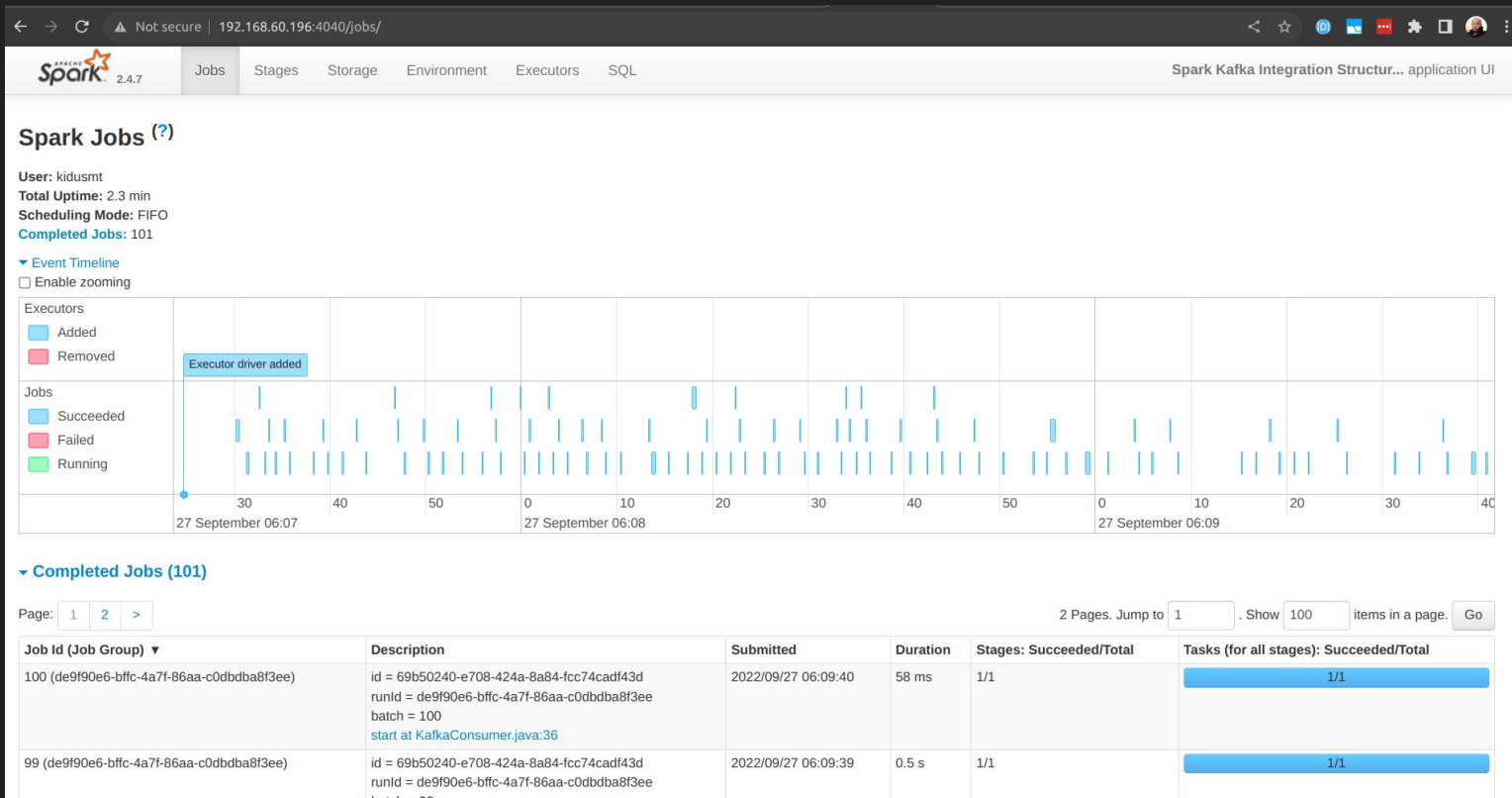
-
- SQL queries
 - Spark configs
 - Hive Configs
 - Kafka Configs
 - Data model variables
 - functions used throughout classes

Kafka Producer - Twitter API

Kafka Consumer - Spark, Hadoop, Hive


```
Run: KafkaConsumer x KafkaProducer x
22/09/27 06:05:18 INFO SessionState: Could not get hdfsEncryptionShim, it is only applicable to hdfs filesystem.
22/09/27 06:05:18 INFO Hive: Renaming src: hdfs://localhost:9000/twitterStore/.hive-staging_hive_2022-09-27_06-05-18_176_7104521593363159816-1/-ext-10000/part-00000-56077382-66a8-4e9e-97d4-e00ae630865c-c000, dest: hdfs://localhost:9000/twitterStore/part
22/09/27 06:05:18 INFO CheckpointFileManager: Writing atomically to file:/tmp/temporary-a07e2137-5217-4bb4-938d-ce624c1c6cfe/commits/20 using temp file file:/tmp/temporary-a07e2137-5217-4bb4-938d-ce624c1c6cfe/commits/.20.71293dc2-4000-4c42-91c9-574f5914
22/09/27 06:05:18 INFO CheckpointFileManager: Renamed temp file file:/tmp/temporary-a07e2137-5217-4bb4-938d-ce624c1c6cfe/commits/.20.71293dc2-4000-4c42-91c9-574f59148299.tmp to file:/tmp/temporary-a07e2137-5217-4bb4-938d-ce624c1c6cfe/commits/20
22/09/27 06:05:18 INFO MicroBatchExecution: Streaming query made progress: {
  "id" : "8911304a-0b53-4cd4-a0ef-140e2ff91f9d",
  "runId" : "d9ea2194-48db-4333-95c8-60ab87531169",
  "name" : null,
  "timestamp" : "2022-09-27T11:05:18.052Z",
  "batchId" : 20,
  "numInputRows" : 1,
  "inputRowsPerSecond" : 83.33333333333333,
  "processedRowsPerSecond" : 2.8818443804034586,
  "durationMs" : {
    "addBatch" : 255,
    "getBatch" : 1,
    "getEndOffset" : 0,
    "queryPlanning" : 59,
    "setOffsetRange" : 1,
    "triggerExecution" : 347,
    "walCommit" : 18
  },
  "stateOperators" : [ ],
  "sources" : [ {
    "description" : "KafkaV2[Subscribe[Tweets]]",
    "startOffset" : {
      "Tweets" : {
        "0" : 33483
      }
    },
    "endOffset" : {
      "Tweets" : {
        "0" : 33484
      }
    },
    "numInputRows" : 1,
    "inputRowsPerSecond" : 83.33333333333333,
    "processedRowsPerSecond" : 2.8818443804034586
  } ],
}
```

Spark: Jobs and Event Timeline



Spark Executors: RDD Blocks Detail and Drivers

← → ↺ ⚠ Not secure | 192.168.60.196:4040/executors/

 2.4.7

JobsStagesStorageEnvironmentExecutorsSQL

Spark Kafka Integration Structur... application UI

Executors

▼Show Additional Metrics

☐ Select All☐ On Heap Memory☐ Off Heap Memory

Summary

	▲ RDD Blocks ↕	Storage Memory ↕	Disk Used ↕	Cores ↕	Active Tasks ↕	Failed Tasks ↕	Complete Tasks ↕	Total Tasks ↕	Task Time (GC Time) ↕	Input ↕	Shuffle Read ↕	Shuffle Write ↕	Blacklisted ↕
Active(1)	0	609.6 KB / 3.1 GB	0.0 B	8	0	0	127	127	8 s (51 ms)	0.0 B	0.0 B	0.0 B	0
Dead(0)	0	0.0 B / 0.0 B	0.0 B	0	0	0	0	0	0 ms (0 ms)	0.0 B	0.0 B	0.0 B	0
Total(1)	0	609.6 KB / 3.1 GB	0.0 B	8	0	0	127	127	8 s (51 ms)	0.0 B	0.0 B	0.0 B	0

Executors

Show

20

 entries

Search:


Executor ID	▲ Address ↕	Status ↕	RDD Blocks ↕	Storage Memory ↕	Disk Used ↕	Cores ↕	Active Tasks ↕	Failed Tasks ↕	Complete Tasks ↕	Total Tasks ↕	Task Time (GC Time) ↕	Input ↕	Shuffle Read ↕	Shuffle Write ↕	Thread Dump ↕
driver	192.168.60.196:36097	Active	0	609.6 KB / 3.1 GB	0.0 B	8	0	0	127	127	8 s (51 ms)	0.0 B	0.0 B	0.0 B	Thread Dump

Showing 1 to 1 of 1 entries

[Previous](#) [1](#) [Next](#)

Spark SQL: Queries

← → ↺ ⚠ Not secure | 192.168.60.196:4040/SQL/ 🔍 ⚙ 🗖 🧑

 2.4.7

Jobs

Stages

Storage

Environment

Executors

SQL

Spark Kafka Integration Structur... application UI

SQL

Completed Queries: 280

Completed Queries (280)

ID	Description		Submitted	Duration	Job IDs
279	start at KafkaConsumer.java:36 org.apache.spark.sql.streaming.DataStreamWriter.start(DataStreamWriter.scala:297) cs523.KafkaConsumer.main(KafkaConsumer.java:36)	+details	2022/09/27 06:10:31	0.2 s	[139]
278	start at KafkaConsumer.java:36 org.apache.spark.sql.streaming.DataStreamWriter.start(DataStreamWriter.scala:297) cs523.KafkaConsumer.main(KafkaConsumer.java:36)	+details	2022/09/27 06:10:30	0.3 s	
277	start at KafkaConsumer.java:36	+details	2022/09/27 06:10:29	0.2 s	[138]
276	start at KafkaConsumer.java:36	+details	2022/09/27 06:10:29	0.2 s	
275	start at KafkaConsumer.java:36	+details	2022/09/27 06:10:28	0.2 s	[137]
274	start at KafkaConsumer.java:36	+details	2022/09/27 06:10:28	0.3 s	
273	start at KafkaConsumer.java:36	+details	2022/09/27 06:10:27	0.3 s	[136]
272	start at KafkaConsumer.java:36	+details	2022/09/27 06:10:27	0.3 s	
271	start at KafkaConsumer.java:36	+details	2022/09/27 06:10:25	0.2 s	[135]
270	start at KafkaConsumer.java:36	+details	2022/09/27 06:10:25	0.3 s	
269	start at KafkaConsumer.java:36	+details	2022/09/27 06:10:20	0.2 s	[134]
268	start at KafkaConsumer.java:36	+details	2022/09/27 06:10:20	0.3 s	
267	start at KafkaConsumer.java:36	+details	2022/09/27 06:10:18	0.2 s	[133]
266	start at KafkaConsumer.java:36	+details	2022/09/27 06:10:18	0.2 s	
265	start at KafkaConsumer.java:36	+details	2022/09/27 06:10:17	0.2 s	[132]
264	start at KafkaConsumer.java:36	+details	2022/09/27 06:10:17	0.2 s	

SPark Job: DAG Visualization

←

→

↺

⚠ Not secure

192.168.60.196:4040/jobs/job?id=173

🔍

🔗

☆

🌐

📧

🔴

⚙

🖨

👤

⋮

APACHE **spark** 2.4.7

Jobs

Stages

Storage

Environment

Executors

SQL

Spark Kafka Integration Structur... application UI

Details for Job 173

Status: SUCCEEDED

Job Group: de9f90e6-bffc-4a7f-86aa-c0dbdba8f3ee

Completed Stages: 1

▶ Event Timeline

▼ DAG Visualization

Stage 173

```
graph TD; A[WholeStageCodegen] --> B[map]; B --> C[Scan]; C --> D[WholeStageCodegen];
```

▼ Completed Stages (1)

Stage Id ▼	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
173	id = 69b50240-e708-424a-8a84-1cc74cad143d runId = de9f90e6-bffc-4a7f-86aa-c0dbdba8f3ee batch = 173 start at KafkaConsumer.java:36	2022/09/27 06:11:31	35 ms	1/1		224.0 B		

+details

Hive Server2 Info

localhost:10002

[Home](#)[Local logs](#)[Metrics Dump](#)[Hive Configuration](#)[Stack Trace](#)[Llap Daemons](#)

HiveServer2

Active Sessions

User Name	IP Address	Operation Count	Active Time (s)	Idle Time (s)
-----------	------------	-----------------	-----------------	---------------

Total number of sessions: 0

Open Queries

User Name	Query	Execution Engine	State	Opened Timestamp	Opened (s)	Latency (s)	Drilldown Link
-----------	-------	------------------	-------	------------------	------------	-------------	----------------

Total number of queries: 0

Last Max 25 Closed Queries

User Name	Query	Execution Engine	State	Opened (s)	Closed Timestamp	Latency (s)	Drilldown Link
-----------	-------	------------------	-------	------------	------------------	-------------	----------------

Total number of queries: 0

Software Attributes

Attribute Name	Value	Description
Hive Version	3.1.2, r8190d2be7b7165effa62bd21b7d60ef81fb0e4af	Hive version and revision
Hive Compiled	Thu Aug 22 15:01:18 PDT 2019, gates	When Hive was compiled and by whom
HiveServer2 Start Time	Tue Sep 27 06:08:41 CDT 2022	Date stamp of when this HiveServer2 was started

Hadoop: Data Node

←

→

↺

🔒

localhost:9864/datanode.html

🔗

☆

🗺

📧

📺

🔴

⚙

🖥

👤

⋮

Hadoop

Overview

Utilities ▾

DataNode on kmt:9866

Cluster ID:	CID-711c612b-13ca-4923-8e91-4b5588f435ca
Started:	Mon Sep 26 01:24:29 -0500 2022
Version:	3.2.4, r7e5d9983b388e372fe640f21f048f2f2ae6e9eba

Block Pools

Namenode Address	Block Pool ID	Actor State	Last Heartbeat	Last Block Report	Last Block Report Size (Max Size)
localhost:9000	BP-995926966-127.0.0.1-1664045558450	RUNNING	2s	3 hours	66.56 KB (64 MB)

Volume Information

Directory	StorageType	Capacity Used	Capacity Left	Capacity Reserved	Reserved Space for Replicas	Blocks
/home/hadoop/hadoopdata/hdfs/datanode	DISK	103.69 MB	52.41 GB	0 B	0 B	8316

Hadoop, 2022.

Hadoop: Cluster Info

←

→

↺

📍 localhost:8088/cluster/cluster

🔍

⌂

☆

🌐

📧


🔴

⚙️

🖥️

👤

⋮



Logged in as: dr.who

About the Cluster

Cluster

About

Nodes

Node Labels

Applications

NEW

NEW SAVING

SUBMITTED

ACCEPTED

RUNNING

FINISHED

FAILED

KILLED

Scheduler

Tools

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Used Resources	Total Resources	Reserved Resources	Physical Mem Used %	Physical VCores Used %
0	0	0	0	0	<memory:0 B, vCores:0>	<memory:0 B, vCores:0>	<memory:0 B, vCores:0>	0	0

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes	Shutdown Nodes
0	0	0	0	1	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Priority	Scheduler Busy %
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>	<memory:8192, vCores:4>	0	0

Cluster overview

Cluster ID:

1664173476903

ResourceManager state:

STARTED

ResourceManager HA state:

active

ResourceManager HA zookeeper connection state:

Could not find leader elector. Verify both HA and automatic failover are enabled.

ResourceManager RMStateStore:

org.apache.hadoop.yarn.server.resourcemanager.recovery.NullRMStateStore

ResourceManager started on:

Mon Sep 26 01:24:36 -0500 2022

ResourceManager version:

3.2.4 from 7e5d9983b388e372fe40f21f048f2f2ae6e9eba by ubuntu source checksum a8ec9b5946c3975e838a2b608e1278a on 2022-07-12T12:07Z

Hadoop version:

3.2.4 from 7e5d9983b388e372fe40f21f048f2f2ae6e9eba by ubuntu source checksum ee031c16fe785bb35252c749418712 on 2022-07-12T11:58Z