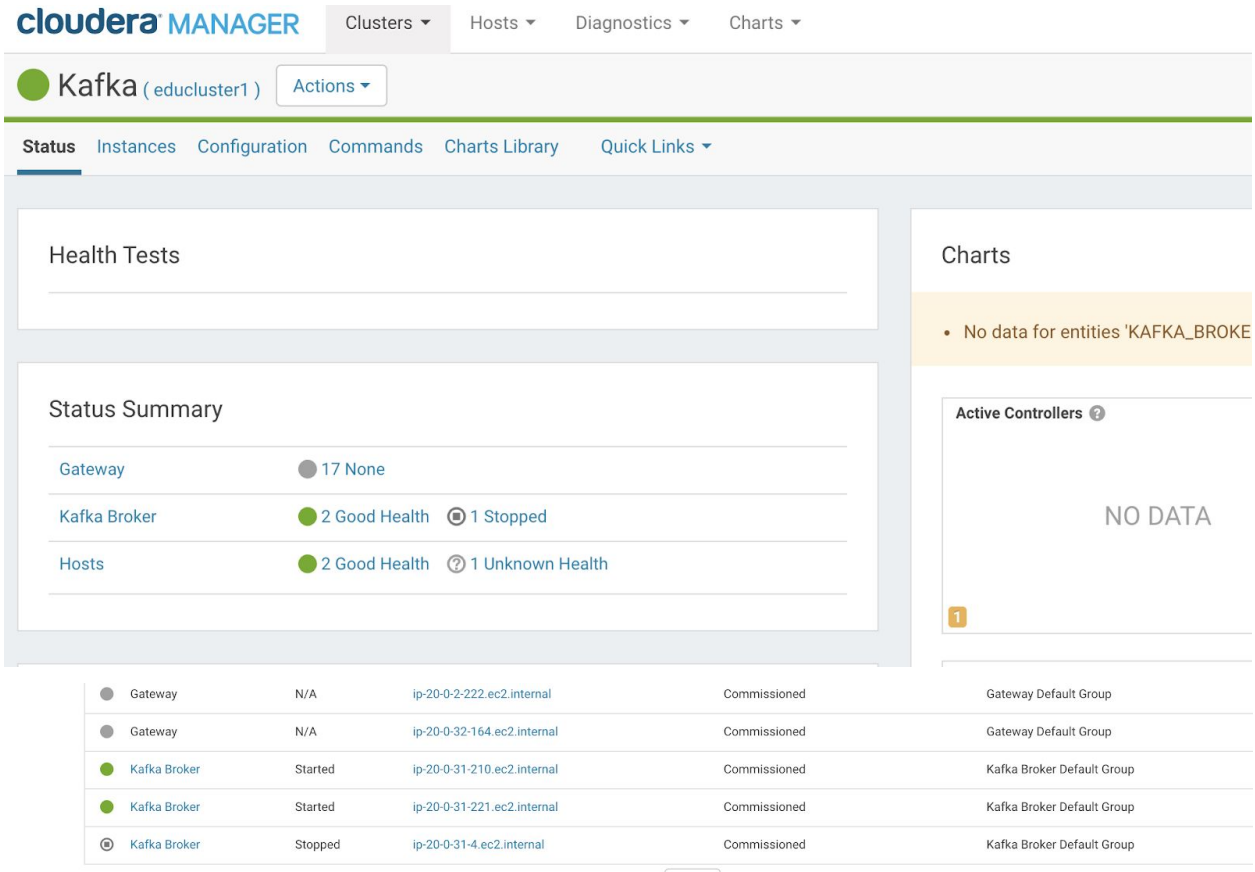
Case Study:Spam Detection
Domain:Telecom
A telecom software provider is building an application to monitor different telecom components in the production environment. For monitoring purpose, the application relies on log files by parsing the log files and looking for potential warning or exceptions in the logs and reporting them.
The PIC we had been working on, for SPAM Detection on the data of telecom operator forum, has been accepted and the sateke holders have asked us to work on the real-time example for predicting SPAM messages.

Tasks:

1. Verify the cluster



2. Create a topic in Kafka so that consumers and producers can enqueue/dequeue data respectively from the topic

```
kafka-topics --create --zookeeper
ip-20-0-31-210.ec2.internal:2181 --replication-factor 3
--partitions 3 --topic ip_Mod9CS1
```

```
19/07/24 11:02:31 INFO zookeeper.ZooKeeper: Client environment:user.dir=/mnt/home/edureka_524533
19/07/24 11:02:31 INFO zookeeper.ZooKeeper: Initiating client connection, connectString=ip-20-0-31-210
8
19/07/24 11:02:31 INFO zkclient.ZkClient: Waiting for keeper state SyncConnected
19/07/24 11:02:31 INFO zookeeper.ClientCnxn: Opening socket connection to server ip-20-0-31-210.ec2.in
19/07/24 11:02:31 INFO zookeeper.ClientCnxn: Socket connection established to ip-20-0-31-210.ec2.inter
19/07/24 11:02:31 INFO zookeeper.ClientCnxn: Session establishment complete on server ip-20-0-31-210.e
000
19/07/24 11:02:31 INFO zkclient.ZkClient: zookeeper state changed (SyncConnected)
WARNING: Due to limitations in metric names, topics with a period ('.') or underscore ('_') could coll
19/07/24 11:02:31 INFO admin.AdminUtils$: Topic creation {"version":1,"partitions":{"2":[296,297,298],"
Created topic "ip_Mod9CS1".
19/07/24 11:02:31 INFO zkclient.ZkEventThread: Terminate ZkClient event thread.
19/07/24 11:02:31 INFO zookeeper.ZooKeeper: Session: 0x16c0879e843334e closed
19/07/24 11:02:31 INFO zookeeper.ClientCnxn: EventThread shut down for session: 0x16c0879e843334e
```

3. Write the test Kafka Consumer and verify data is sent successfully
   kafka-console-consumer --zookeeper `ip-20-0-31-210.ec2.internal:2181`
   --topic `ip_Mod9CS1`

```
nal:9092
19/07/24 11:07:43 INFO utils.VerifiableProperties: Property request.timeout.ms is overridden to 30000
19/07/24 11:07:43 INFO client.ClientUtils$: Fetching metadata from broker BrokerEndPoint(298,ip-20-0-31-4.ec2.internal,9092) with correlation id 0 for 1 topic(s) Set(ip_Mod9
19/07/24 11:07:43 INFO producer.SyncProducer: Connected to ip-20-0-31-4.ec2.internal:9092 for producing
19/07/24 11:07:43 INFO producer.SyncProducer: Disconnecting from ip-20-0-31-4.ec2.internal:9092
19/07/24 11:07:43 INFO consumer.ConsumerFetcherThread: [ConsumerFetcherThread-console-consumer-77200_ip-20-0-32-164.ec2.internal-1563966462776-a5725e4c-0-296]: Starting
19/07/24 11:07:43 INFO consumer.ConsumerFetcherThread: [ConsumerFetcherThread-console-consumer-77200_ip-20-0-32-164.ec2.internal-1563966462776-a5725e4c-0-298]: Starting
19/07/24 11:07:43 INFO consumer.ConsumerFetcherThread: [ConsumerFetcherThread-console-consumer-77200_ip-20-0-32-164.ec2.internal-1563966462776-a5725e4c-0-297]: Starting
19/07/24 11:07:43 INFO consumer.ConsumerFetcherManager: [ConsumerFetcherManager-1563966462792] Added fetcher for partitions ArrayBuffer([ip_Mod9CS1-2, initOffset -1 to broke
int(296,ip-20-0-31-210.ec2.internal,9092)] , [ip_Mod9CS1-1, initOffset -1 to broker BrokerEndPoint(298,ip-20-0-31-4.ec2.internal,9092)] , [ip_Mod9CS1-0, initOffset -1 to bro
Point(297,ip-20-0-31-221.ec2.internal,9092)] )
```

   `Created Kafa Producer to check on the topic`
   kafka-console-producer --broker-list ip-20-0-31-221.ec2.internal:9092 --topic `ip_Mod9CS1`

```
        ssl.keystore.password = null
        ssl.keystore.type = JKS
        ssl.protocol = TLS
        ssl.provider = null
        ssl.secure.random.implementation = null
        ssl.trustmanager.algorithm = PKIX
        ssl.truststore.location = null
        ssl.truststore.password = null
        ssl.truststore.type = JKS
        transaction.timeout.ms = 60000
        transactional.id = null
        value.serializer = class org.apache.kafka.common.serialization.ByteArraySerializer

19/07/24 11:11:34 INFO utils.AppInfoParser: Kafka version : 0.11.0-kafka-3.0.0
19/07/24 11:11:34 INFO utils.AppInfoParser: Kafka commitId : unknown
>My First Kafka Message
>
```

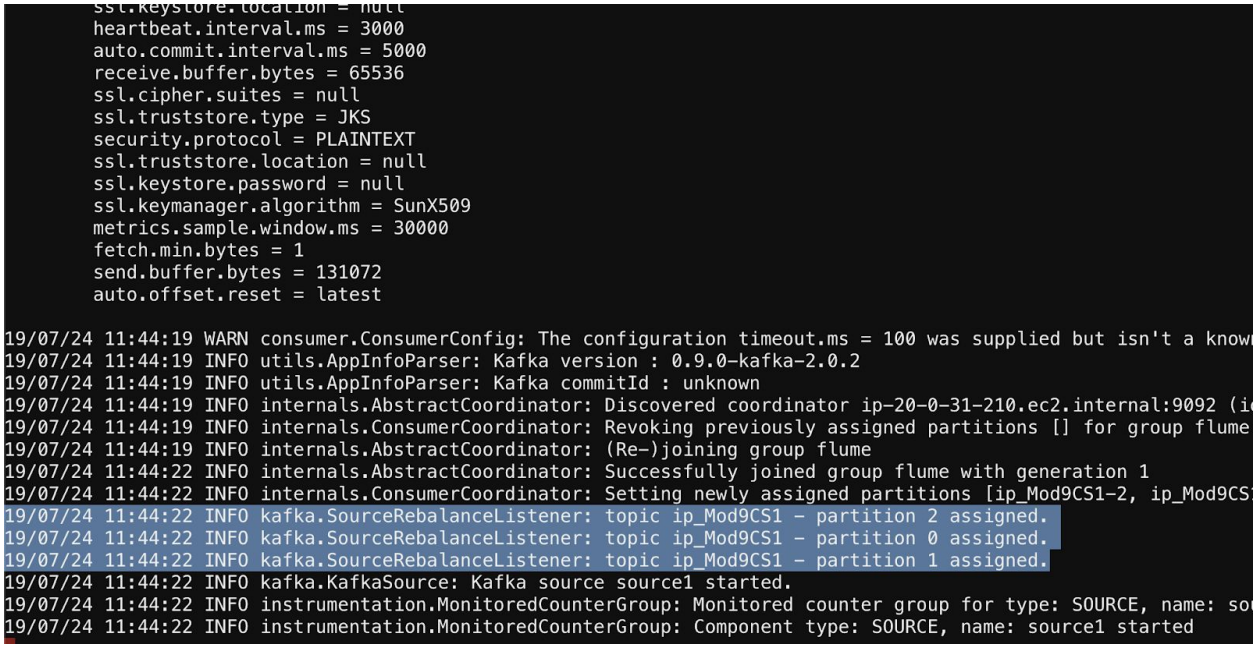
   `Received by the consumer`

```
19/07/24 11:07:43 INFO consumer.ConsumerFetcherThread: [ConsumerFetcherThread-console-consumer-77200_ip-20-0-32-164.ec2
19/07/24 11:07:43 INFO consumer.ConsumerFetcherManager: [ConsumerFetcherManager-1563966462792] Added fetcher for partit
int(296,ip-20-0-31-210.ec2.internal,9092)] , [ip_Mod9CS1-1, initOffset -1 to broker BrokerEndPoint(298,ip-20-0-31-4.ec2
Point(297,ip-20-0-31-221.ec2.internal,9092)] )
My First Kafka Message
```

4. Configure a flume agent to configure Kafka as the channel and HDFS as Sink
   Create Config file : `mod9cs1.conf`

5. Start Flume agent and test the output to HDFS
   Start the Flume Agent:

flume-ng agent --conf conf --conf-file mod9cs1.conf --name agent1 -Dflume.root.logger=INFO,console



6. Test the complete pipeline
   Start the Kafka Producer

```
kafka-console-producer --broker-list
ip-20-0-31-221.ec2.internal:9092 --topic ip_Mod9CS1
```

```
Type in the messages to be sent
```

```
19/07/24 11:11:34 INFO utils.AppInfoParser: Kafka version : 0.11.0-kafka-3.0.0
19/07/24 11:11:34 INFO utils.AppInfoParser: Kafka commitId : unknown
>My First Kafka Message
>My Second message
>Third
>Sjkjkjk
>
>
>Send another
>Send More
>Here
>Producer
>Kafka to Flume Message 1
>Kafka Flume Message 2
>KfkaFlume Message 3
>Message 4
>Message 5
>Message 6
>Message 7
>There
>There are many messages to be written to HDFS
>Here are a few ones added today
>
```

```
19/07/24 11:55:26 INFO instrumentation.MonitoredCounterGroup: Component type: SOURCE, name: source1 started
19/07/24 11:55:27 INFO hdfs.HDFSDataStream: Serializer = TEXT, UseRawLocalFileSystem = false
19/07/24 11:55:27 INFO hdfs.BucketWriter: Creating hdfs://nameservice1/user/edureka_524533/Flume_Kafka/FlumeData.15639
19/07/24 11:55:38 INFO hdfs.BucketWriter: Closing hdfs://nameservice1/user/edureka_524533/Flume_Kafka/FlumeData.156396
19/07/24 11:55:38 INFO hdfs.BucketWriter: Renaming hdfs://nameservice1/user/edureka_524533/Flume_Kafka/FlumeData.15639
FlumeData.1563969327089
19/07/24 11:55:38 INFO hdfs.HDFSEventSink: Writer callback called.
19/07/24 11:55:48 INFO hdfs.HDFSDataStream: Serializer = TEXT, UseRawLocalFileSystem = false
19/07/24 11:55:48 INFO hdfs.BucketWriter: Creating hdfs://nameservice1/user/edureka_524533/Flume_Kafka/FlumeData.15639
19/07/24 11:56:00 INFO hdfs.BucketWriter: Closing hdfs://nameservice1/user/edureka_524533/Flume_Kafka/FlumeData.156396
19/07/24 11:56:00 INFO hdfs.BucketWriter: Renaming hdfs://nameservice1/user/edureka_524533/Flume_Kafka/FlumeData.15639
FlumeData.1563969348094
19/07/24 11:56:00 INFO hdfs.HDFSEventSink: Writer callback called.
19/07/24 11:57:30 INFO hdfs.HDFSDataStream: Serializer = TEXT, UseRawLocalFileSystem = false
19/07/24 11:57:30 INFO hdfs.BucketWriter: Creating hdfs://nameservice1/user/edureka_524533/Flume_Kafka/FlumeData.15639
19/07/24 11:57:40 INFO hdfs.BucketWriter: Closing hdfs://nameservice1/user/edureka_524533/Flume_Kafka/FlumeData.156396
19/07/24 11:57:40 INFO hdfs.BucketWriter: Renaming hdfs://nameservice1/user/edureka_524533/Flume_Kafka/FlumeData.15639
FlumeData.1563969450115
19/07/24 11:57:40 INFO hdfs.HDFSEventSink: Writer callback called.
```

Home / user / edureka_524533 / Flume_Kafka / **FlumeData.1563969348094**

```
There are many messages to be written to HDFS
Here are a few ones added today
```