

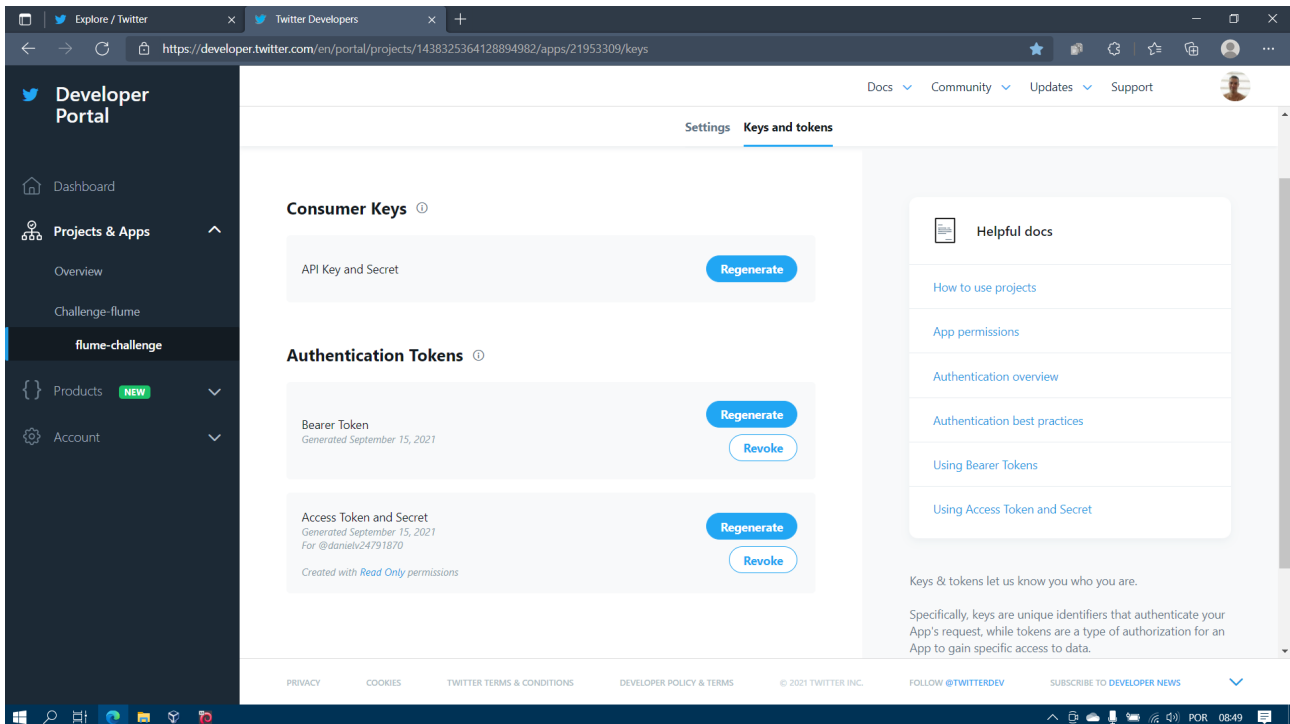
Carrefour Data Challenge

O desafio teve como objetivo aquisição de dados do Twitter em Streaming e criação de insights dos mesmos. Na etapa final foi criada uma CloudWord, que pode ser interpretada através das palavras mais mencionadas dos Trendings Topics no momento da coleta de dados. Foi criado um Data Lake com 3 camadas (Extração, Mensageiria e Armazenamento) e depois foi utilizado ferramentas para análise dos dados. Na camada de Extração foi utilizado o Apache NIFI, que consiste em uma solução ETL que permite automatizar o fluxo de dados entre sistemas.

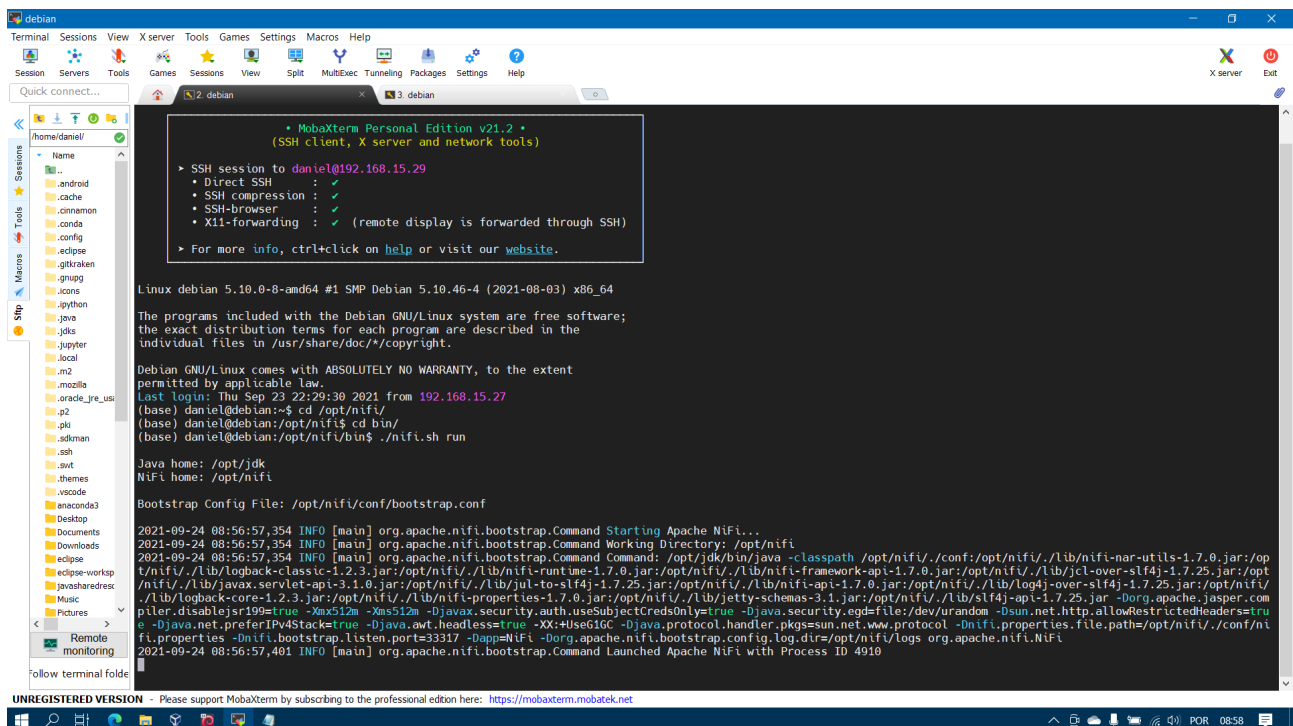
Na camada de mensagens cujo objetivo é tratar fluxos de dados gerados em tempo real das mais variadas fontes. Uma das principais funções dessa camada é a capacidade de dissociar a origem (produtor) e o destino (consumidor), capacidade de manipular mensagens de alta velocidade na ordem de centenas de megabytes por segundo de cada nó do servidor de aplicativos, lidar com grandes volumes de dados da ordem de terabytes a petabytes, fornecer a mesma mensagem a vários consumidores. Por exemplo: fornecer mensagens para a camada Speed Layer (que vai processar os dados em tempo real) e Data Storage Layer (que vai armazenar os dados) ao mesmo tempo, análise de dados para derivar estatísticas operacionais. Neste caso foi utilizado o Apache Kafka que é uma plataforma open-source de processamento de streaming de dados, mantida pela Apache Software Foundation, escrita em Scala e Java. Sua camada de armazenamento é essencialmente uma "fila de mensagens massivamente escalável arquitetada como um log de transações distribuídas", tornando altamente valioso para infraestruturas corporativas de processamento de streaming de dados. Sobre o Apache Kafka foi utilizado o Apache Zookeeper para gerenciar o sistema. Na camada de armazenamento foi utilizado o banco de dados noSQL MongoDB, capaz de armazenar grandes volumes de dados não estruturados. O ambiente utilizado foi o Sistema operacional Debian (Linux)

11.

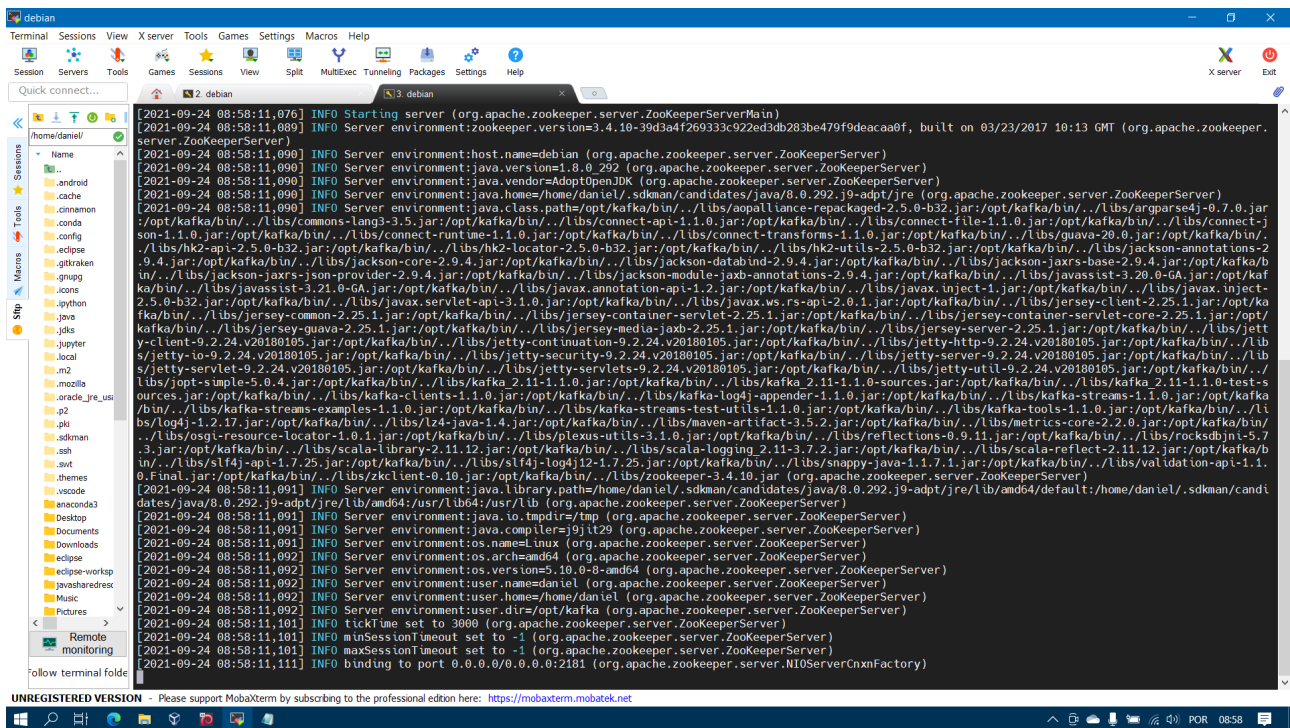
As camadas do Data Lake estão ilustradas nas figuras abaixo:



- Para extrair os dados foi criado uma API do Twitter, liberando chaves de acesso para o consumo de dados.

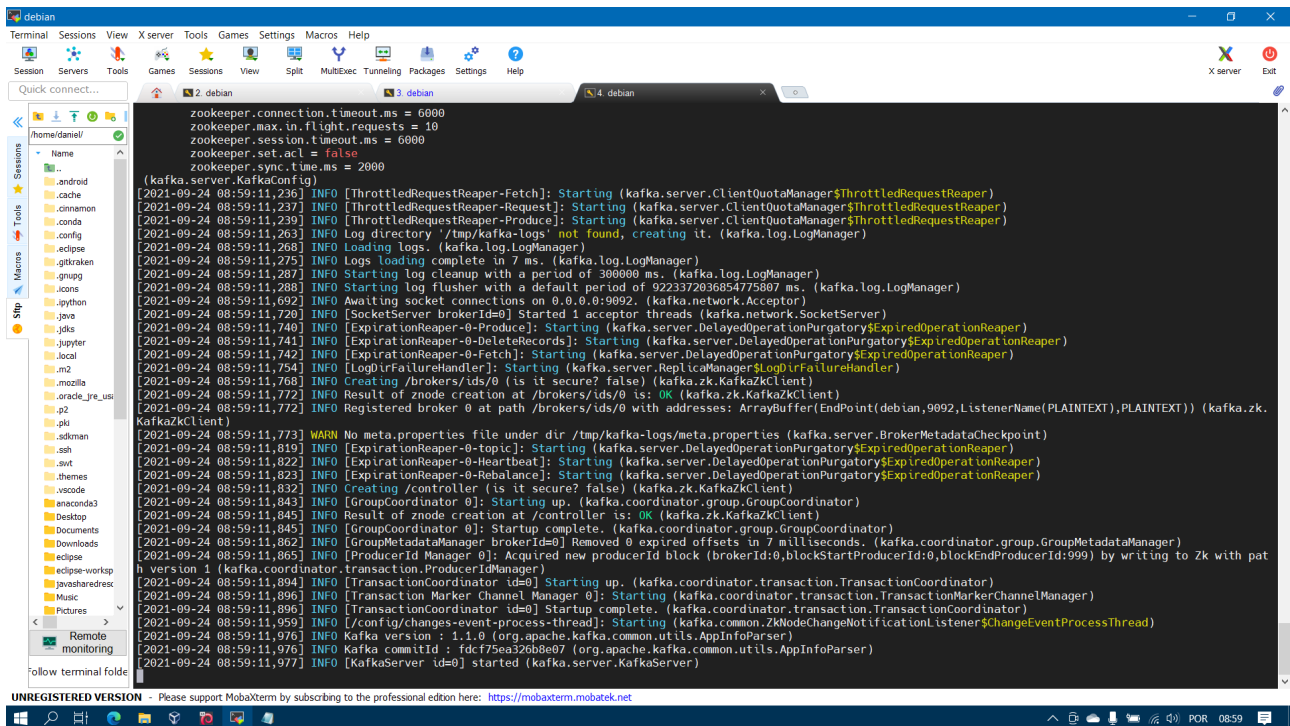


- A figura acima ilustra o início do frameWork Apache NIFI.



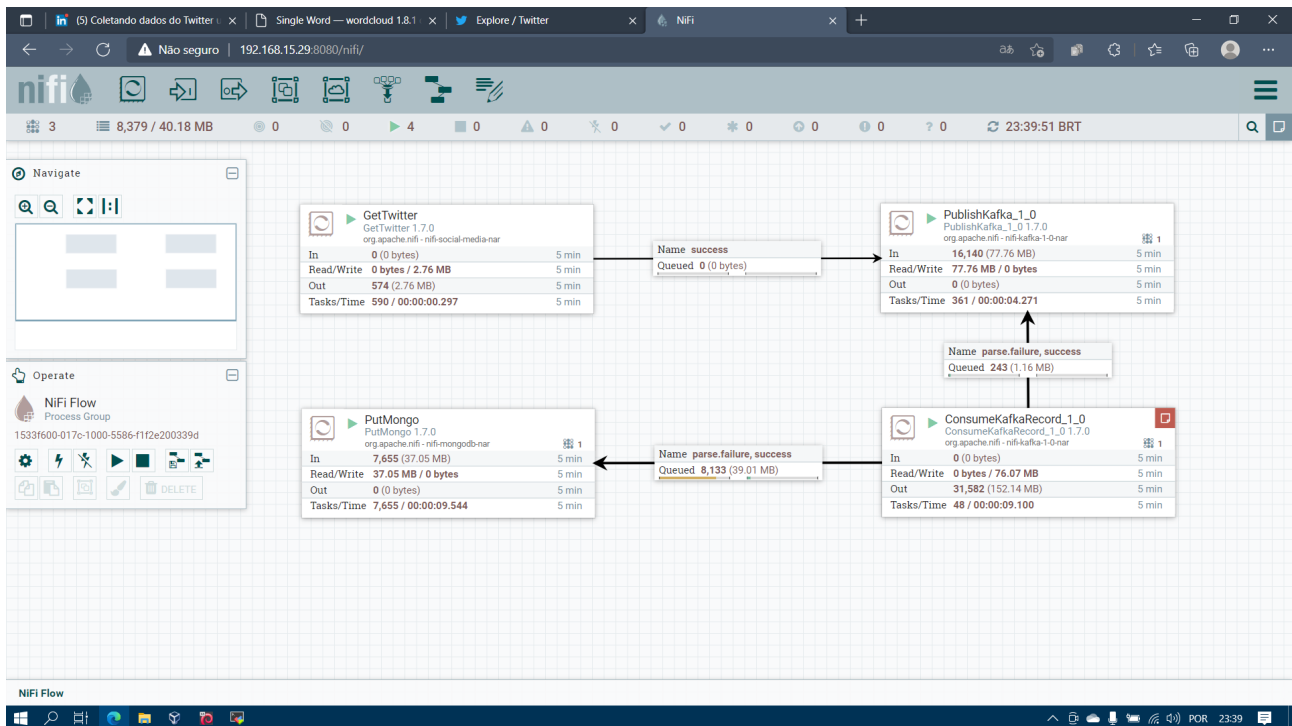
```
[2021-09-24 08:58:11,076] INFO Starting server (org.apache.zookeeper.server.ZooKeeperServerMain)
[2021-09-24 08:58:11,089] INFO Server environment:zookeeper.version=3.4.10-30d3a4f26933c922ed3db283be479f9deacaa0f, built on 03/23/2017 10:13 GMT (org.apache.zookeeper.server.ZooKeeperServer)
[2021-09-24 08:58:11,090] INFO Server environment:host.name=debian (org.apache.zookeeper.server.ZooKeeperServer)
[2021-09-24 08:58:11,090] INFO Server environment:java.version=1.8.0_292 (org.apache.zookeeper.server.ZooKeeperServer)
[2021-09-24 08:58:11,090] INFO Server environment:java.vendor=AdoptOpenJDK (org.apache.zookeeper.server.ZooKeeperServer)
[2021-09-24 08:58:11,090] INFO Server environment:java.home=/home/daniel/.sdkman/candidates/java/8.0.292-j9-adpt/jre (org.apache.zookeeper.server.ZooKeeperServer)
[2021-09-24 08:58:11,090] INFO Server environment:java.class.path=/opt/kafka/bin/./libs/aopalliance-repackaged-2.5.0-b32.jar:/opt/kafka/bin/./libs/argparse4j-0.7.0.jar:/opt/kafka/bin/./libs/commons-lang3-3.5.jar:/opt/kafka/bin/./libs/connect-api-1.1.0.jar:/opt/kafka/bin/./libs/connect-file-1.1.0.jar:/opt/kafka/bin/./libs/connect-json-1.1.0.jar:/opt/kafka/bin/./libs/connect-runtime-1.1.0.jar:/opt/kafka/bin/./libs/connect-transforms-1.1.0.jar:/opt/kafka/bin/./libs/guava-20.0.jar:/opt/kafka/bin/./libs/hk2-api-2.5.0-b32.jar:/opt/kafka/bin/./libs/hk2-locator-2.5.0-b32.jar:/opt/kafka/bin/./libs/hk2-utils-2.5.0-b32.jar:/opt/kafka/bin/./libs/jackson-annotations-2.9.4.jar:/opt/kafka/bin/./libs/jackson-core-2.9.4.jar:/opt/kafka/bin/./libs/jackson-databind-2.9.4.jar:/opt/kafka/bin/./libs/jackson-jaxrs-base-2.9.4.jar:/opt/kafka/bin/./libs/jackson-jaxrs-json-provider-2.9.4.jar:/opt/kafka/bin/./libs/jackson-module-jaxb-annotations-2.9.4.jar:/opt/kafka/bin/./libs/javassist-3.20.0-GA.jar:/opt/kafka/bin/./libs/javax-2.31.0-GA.jar:/opt/kafka/bin/./libs/javax.annotation-api-1.2.jar:/opt/kafka/bin/./libs/javax.inject-1.jar:/opt/kafka/bin/./libs/javax.inject-2.5.0-b32.jar:/opt/kafka/bin/./libs/javax.servlet-api-3.1.0.jar:/opt/kafka/bin/./libs/javax.ws.rs-api-2.0.1.jar:/opt/kafka/bin/./libs/jersey-client-2.25.1.jar:/opt/kafka/bin/./libs/jersey-common-2.25.1.jar:/opt/kafka/bin/./libs/jersey-container-servlet-2.25.1.jar:/opt/kafka/bin/./libs/jersey-container-servlet-core-2.25.1.jar:/opt/kafka/bin/./libs/jersey-guava-2.25.1.jar:/opt/kafka/bin/./libs/jersey-media-jaxb-2.25.1.jar:/opt/kafka/bin/./libs/jersey-server-2.25.1.jar:/opt/kafka/bin/./libs/jetty-client-9.2.24.v20180105.jar:/opt/kafka/bin/./libs/jetty-continuation-9.2.24.v20180105.jar:/opt/kafka/bin/./libs/jetty-http-9.2.24.v20180105.jar:/opt/kafka/bin/./libs/jetty-io-9.2.24.v20180105.jar:/opt/kafka/bin/./libs/jetty-security-9.2.24.v20180105.jar:/opt/kafka/bin/./libs/jetty-server-9.2.24.v20180105.jar:/opt/kafka/bin/./libs/jetty-servlet-9.2.24.v20180105.jar:/opt/kafka/bin/./libs/jetty-servlets-9.2.24.v20180105.jar:/opt/kafka/bin/./libs/jetty-util-9.2.24.v20180105.jar:/opt/kafka/bin/./libs/opt-simple-5.0.4.jar:/opt/kafka/bin/./libs/kafka-2.11-1.1.0.jar:/opt/kafka/bin/./libs/kafka-2.11-1.1.0-sources.jar:/opt/kafka/bin/./libs/kafka-2.11-1.1.0-test-sources.jar:/opt/kafka/bin/./libs/kafka-clients-1.1.0.jar:/opt/kafka/bin/./libs/kafka-log4j-appender-1.1.0.jar:/opt/kafka/bin/./libs/kafka-streams-1.1.0.jar:/opt/kafka/bin/./libs/kafka-streams-examples-1.1.0.jar:/opt/kafka/bin/./libs/kafka-streams-test-utils-1.1.0.jar:/opt/kafka/bin/./libs/kafka-tools-1.1.0.jar:/opt/kafka/bin/./libs/log4j-1.2.17.jar:/opt/kafka/bin/./libs/lz4-java-1.4.jar:/opt/kafka/bin/./libs/maven-artifact-3.5.2.jar:/opt/kafka/bin/./libs/metrics-core-2.2.0.jar:/opt/kafka/bin/./libs/osgi-resource-locator-1.0.1.jar:/opt/kafka/bin/./libs/plexus-utils-3.1.0.jar:/opt/kafka/bin/./libs/reflections-0.9.11.jar:/opt/kafka/bin/./libs/rocksdbjni-5.7.3.jar:/opt/kafka/bin/./libs/scala-library-2.11.12.jar:/opt/kafka/bin/./libs/scala-logging-2.11-3.7.2.jar:/opt/kafka/bin/./libs/scala-reflect-2.11.12.jar:/opt/kafka/bin/./libs/slf4j-api-1.7.25.jar:/opt/kafka/bin/./libs/slf4j-log4j12-1.7.25.jar:/opt/kafka/bin/./libs/snappy-java-1.1.7.1.jar:/opt/kafka/bin/./libs/validation-api-1.1.0-Final.jar:/opt/kafka/bin/./libs/zkclient-0.10.jar:/opt/kafka/bin/./libs/zookeeper-3.4.10.jar (org.apache.zookeeper.server.ZooKeeperServer)
[2021-09-24 08:58:11,091] INFO Server environment:java.library.path=/home/daniel/.sdkman/candidates/java/8.0.292-j9-adpt/jre/lib/amd64/default:/home/daniel/.sdkman/candidates/java/8.0.292-j9-adpt/jre/lib/amd64 (org.apache.zookeeper.server.ZooKeeperServer)
[2021-09-24 08:58:11,091] INFO Server environment:java.io.tmpdir=/tmp (org.apache.zookeeper.server.ZooKeeperServer)
[2021-09-24 08:58:11,091] INFO Server environment:java.compiler=j9jit29 (org.apache.zookeeper.server.ZooKeeperServer)
[2021-09-24 08:58:11,091] INFO Server environment:os.name=Linux (org.apache.zookeeper.server.ZooKeeperServer)
[2021-09-24 08:58:11,092] INFO Server environment:os.arch=amd64 (org.apache.zookeeper.server.ZooKeeperServer)
[2021-09-24 08:58:11,092] INFO Server environment:os.version=5.10.0-8-amd64 (org.apache.zookeeper.server.ZooKeeperServer)
[2021-09-24 08:58:11,092] INFO Server environment:user.name=daniel (org.apache.zookeeper.server.ZooKeeperServer)
[2021-09-24 08:58:11,092] INFO Server environment:user.home=/home/daniel (org.apache.zookeeper.server.ZooKeeperServer)
[2021-09-24 08:58:11,092] INFO Server environment:user.dir=/opt/kafka (org.apache.zookeeper.server.ZooKeeperServer)
[2021-09-24 08:58:11,101] INFO tickTime set to 3000 (org.apache.zookeeper.server.ZooKeeperServer)
[2021-09-24 08:58:11,101] INFO minSessionTimeout set to -1 (org.apache.zookeeper.server.ZooKeeperServer)
[2021-09-24 08:58:11,101] INFO maxSessionTimeout set to -1 (org.apache.zookeeper.server.ZooKeeperServer)
[2021-09-24 08:58:11,111] INFO binding to port 0.0.0.0:0.0.0:2181 (org.apache.zookeeper.server.NIOServerCnxnFactory)
```

- Esta imagem mostra a subida do Apache Zookeeper.

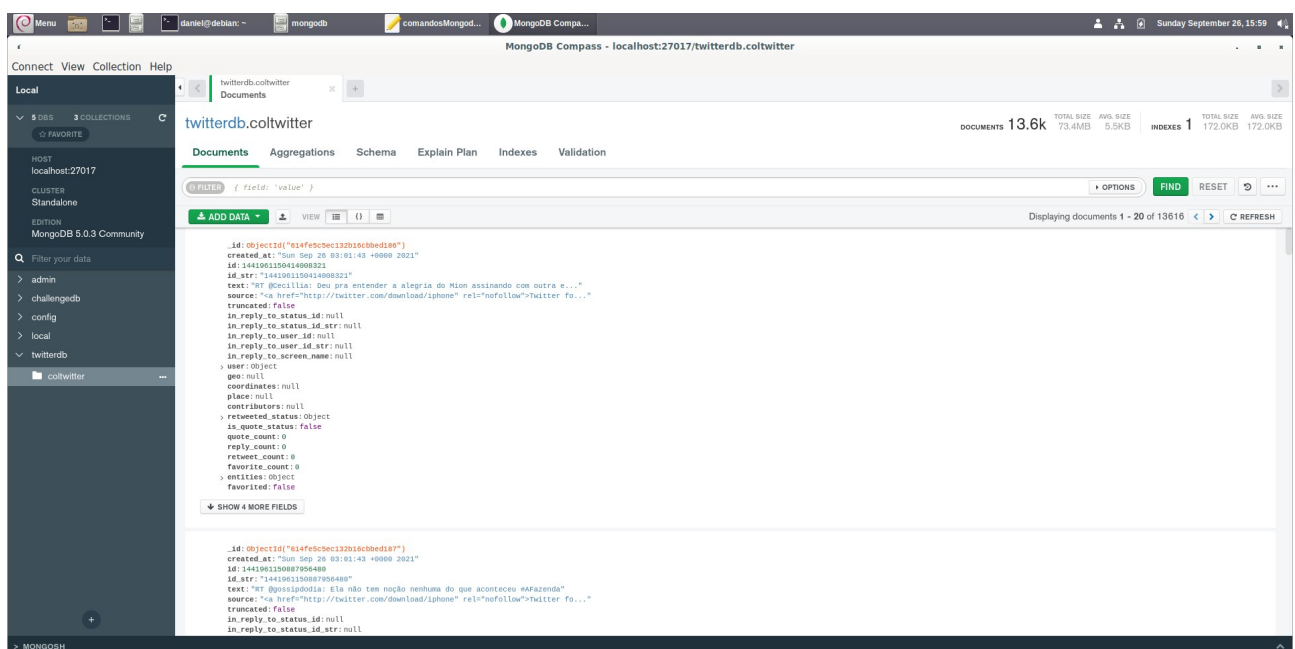


```
zookeeper.connection.timeout.ms = 6000
zookeeper.max.in.flight.requests = 10
zookeeper.session.timeout.ms = 6000
zookeeper.set.acl = false
zookeeper.sync.time.ms = 2000
(kafka.server.KafkaConfig)
[2021-09-24 08:59:11,236] INFO [ThrottledRequestReaper-Fetch]: Starting (kafka.server.ClientQuotaManager$ThrottledRequestReaper)
[2021-09-24 08:59:11,237] INFO [ThrottledRequestReaper-Request]: Starting (kafka.server.ClientQuotaManager$ThrottledRequestReaper)
[2021-09-24 08:59:11,239] INFO [ThrottledRequestReaper-Producer]: Starting (kafka.server.ClientQuotaManager$ThrottledRequestReaper)
[2021-09-24 08:59:11,263] INFO Log directory '/tmp/kafka-logs' not found, creating it. (kafka.log.LogManager)
[2021-09-24 08:59:11,268] INFO Loading logs. (kafka.log.LogManager)
[2021-09-24 08:59:11,275] INFO Logs loading complete in 7 ms. (kafka.log.LogManager)
[2021-09-24 08:59:11,287] INFO Starting log flusher with a period of 360000 ms. (kafka.log.LogManager)
[2021-09-24 08:59:11,288] INFO Starting log flusher with a default period of 9223372036854775807 ms. (kafka.log.LogManager)
[2021-09-24 08:59:11,692] INFO Awaiting socket connections on 0.0.0.0:9092. (kafka.network.Acceptor)
[2021-09-24 08:59:11,720] INFO [SocketServer brokerId=0] started 1 acceptor threads (kafka.network.SocketServer)
[2021-09-24 08:59:11,740] INFO [ExpirationReaper-0-Producer]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
[2021-09-24 08:59:11,741] INFO [ExpirationReaper-0-DeleteRecords]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
[2021-09-24 08:59:11,742] INFO [ExpirationReaper-0-Fetch]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
[2021-09-24 08:59:11,754] INFO [LogDirFailureHandler]: Starting (kafka.server.ReplicaManager$LogDirFailureHandler)
[2021-09-24 08:59:11,768] INFO Creating /brokers/ids/0 (is it secure? false) (kafka.zk.KafkaZkClient)
[2021-09-24 08:59:11,772] INFO Result of znode creation at /brokers/ids/0 is: OK (kafka.zk.KafkaZkClient)
[2021-09-24 08:59:11,772] INFO Registered broker 0 at path /brokers/ids/0 with addresses: ArrayBuffer(EndPoint(debian,9092,ListenerName(PLAINTEXT)),PLAINTEXT)) (kafka.zk.KafkaZkClient)
[2021-09-24 08:59:11,773] WARN No meta.properties file under dir /tmp/kafka-logs/meta.properties (kafka.server.BrokerMetadataCheckpoint)
[2021-09-24 08:59:11,819] INFO [ExpirationReaper-0-topic]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
[2021-09-24 08:59:11,822] INFO [ExpirationReaper-0-Heartbeat]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
[2021-09-24 08:59:11,823] INFO [ExpirationReaper-0-Rebalance]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
[2021-09-24 08:59:11,832] INFO Creating /controller (is it secure? false) (kafka.zk.KafkaZkClient)
[2021-09-24 08:59:11,843] INFO [GroupCoordinator 0]: Starting up. (kafka.coordinator.group.GroupCoordinator)
[2021-09-24 08:59:11,845] INFO Result of znode creation at /controller is: OK (kafka.zk.KafkaZkClient)
[2021-09-24 08:59:11,845] INFO [GroupCoordinator 0]: Startup complete. (kafka.coordinator.group.GroupCoordinator)
[2021-09-24 08:59:11,862] INFO [GroupMetadataManager brokerId=0] Removed 0 expired offsets in 7 milliseconds. (kafka.coordinator.group.GroupMetadataManager)
[2021-09-24 08:59:11,865] INFO [ProducerId Manager 0]: Acquired new producerId block (brokerId=0,blockStartProducerId=0,blockEndProducerId=999) by writing to Zk with path version 1 (kafka.coordinator.transaction.ProducerIdManager)
[2021-09-24 08:59:11,894] INFO [TransactionCoordinator id=0] Starting up. (kafka.coordinator.transaction.TransactionCoordinator)
[2021-09-24 08:59:11,896] INFO [TransactionMarker Channel Manager 0]: Starting (kafka.coordinator.transaction.TransactionMarkerChannelManager)
[2021-09-24 08:59:11,896] INFO [TransactionCoordinator id=0] Startup complete. (kafka.coordinator.transaction.TransactionCoordinator)
[2021-09-24 08:59:11,959] INFO [/config/changes-event-process-thread]: Starting (kafka.common.ZkNodeChangeNotificationListener$ChangeEventProcessThread)
[2021-09-24 08:59:11,976] INFO Kafka version : 1.1.0 (org.apache.kafka.common.utils.AppInfoParser)
[2021-09-24 08:59:11,976] INFO Kafka commitId : fdcf75ea326b8e07 (org.apache.kafka.common.utils.AppInfoParser)
[2021-09-24 08:59:11,977] INFO [KafkaServer id=0] started (kafka.server.KafkaServer)
```

- Nesta imagem em questão mostra a subida do Broker que consiste em um servidor Kafka que recebe mensagens dos produtores e as grava no disco. Cada broker gerencia uma lista de tópicos e cada tópico é dividido em diversas partições.



- Nesta imagem mostra o processo sendo executado. Nifi coletando os dados do Twitter, o Kafka como intermediário(Middleware) produzindo as mensagens e armazenando o MongoDB, que por sua vez vai ser consumido para transformação dos dados em Python.



- Dados armazenados no MongoDB.

