XXXVIII Jornadas de Automática
https://doi.org/10.17979/spudc.9788497497749.626

FRAMEWORK PARA EL MODELADO DE UN LAGO DE DATOS

J.M. Torres[1], R.M. Aguilar[2], C.A. Martín[3], S. Diaz
Department of Computer and Systems Engineering, University of La Laguna
38200 La Laguna (Tenerife), España.
[1]jmtorres@ull.edu.es,[2]raguilar@ull.edu.es, [3]carlos.martin.galan@iac.es

Resumen

La necesidad de trabajar con grandes cantidades de datos para realizar clasificaciones, detectar patrones, identificar sistemas y predecir el comportamiento futuro de los mismos exige un entorno escalable que permita obtener el valor oculto en los datos. En este trabajo se presenta una pila formada por un almacén de datos, un visualizador y un gestor de flujos de trabajo para la ejecución de tareas, como marco general para el uso masivo de los datos. Siendo esta estructura general para cualquier dominio y formada por herramientas de software libre.

Palabras Clave: Lago de Datos, Elasticsearch, Kibana, AirFlow

1 INTRODUCCIÓN

No es ninguna novedad, que en la actualidad la captura y el almacenamiento de datos tiene un coste muy bajo. Este hecho nos permite disponer de una gran cantidad de datos almacenados. Además del gran volumen de información, existe una gran variedad de datos que pueden ser representados de diversas maneras: Por ejemplo, a través de los dispositivos móviles se puede capturar audio, video, localizaciones de GPS y otros valores proporcionados por los distintos sensores del dispositivo. Además existen incontables sensores digitales en equipos industriales, automóviles, medidores eléctricos, veletas, anemómetros, etc., los cuales pueden medir y comunicar posición, movimiento, vibración, temperatura, humedad y hasta los cambios químicos que sufre el aire. Pero no sólo nos referimos a datos procedentes de tecnologías que permiten conectarse a otros dispositivos Machine-to-Machine (M2M); también se deben incluir datos alojados en las redes sociales como Facebook, Twitter, LinkedIn, blogs, etc; que nos permiten conocer los gustos y preferencias de los usuarios y, yendo más allá, incluso sus estados de ánimo. O datos procedentes de las transacciones realizadas entre cliente-empresa, que incluyen registros de facturación y, en telecomunicaciones, registros detallados de las llamadas, etc. Además, de la información que generamos las personas en un call center al establecer una llamada telefónica: notas de voz, correos electrónicos, documentos electrónicos, estudios médicos, etc. Todo ello nos lleva a pensar que no sólo disponemos de una cantidad ingente de datos almacenados sino que además, estos pueden ser estructurados, no estructurados o semi estructurados.

Los datos pueden esconder información de gran valor para saber no sólo lo que sucede a nuestro alrededor, sino también lo que va a pasar en un futuro, obteniendo niveles de precisión muy altos. Para poder tratar estas cantidades de datos y darle valor añadido a los datos extrayendo el conocimiento existen en ellos, las técnicas tradicionales de estadística y las herramientas de gestión clásicas no sirven debido a que no están preparadas para trabajar con tanta información ni aún menos con datos tan variados. Por lo que se hace evidente la necesidad de nuevas herramientas de análisis de este Big-Data.

Big Data es el proceso de recolección de grandes cantidades de datos y su inmediato análisis para encontrar información oculta, patrones recurrentes, nuevas correlaciones, etc. El conjunto de datos es tan grande y complejo que los medios tradicionales de procesamiento son ineficaces. Y es que estamos hablando de desafíos como analizar, capturar, recolectar, buscar, compartir, almacenar, transferir, visualizar, etc., ingentes cantidades de información, obtener conocimiento en tiempo real y poner todos los sentidos en la protección de datos personales. Las características que exhiben este proceso son [1]:

- Volumen: captar y organizar absolutamente toda la información que nos llega es esencial para tener registros completos e insesgados, y que las conclusiones que obtengamos sirvan eficientemente a la hora de la toma de decisiones.
- Velocidad: es importante el tiempo si afrontamos tanto la necesidad de generar información como de analizarla, pero lo es más si necesitamos reaccionar inmediatamente. Todo el proceso pide agilidad para extraer valor de negocio a la

información que se estudia y que no se pierda la oportunidad.

- Variedad: Hay que dar uniformidad a toda la información, que tendrá su origen en datos de lo más heterogéneos. Una de las fortalezas del Big Data reside en poder conjugar y combinar cada tipo de información y su tratamiento específico para alcanzar un todo homogéneo.
- Veracidad: la calidad del dato y su disponibilidad hay que encontrar herramientas para comprobar la bondad de la información recibida.
- Valor: Trabajar con Big Data tiene que servir para aportar valor a la sociedad, las empresas, los gobiernos, en definitiva, a las personas; todo el proceso tiene que ayudar a impulsar el desarrollo, la innovación y la competitividad, pero también mejorar la calidad de vida de las personas.

Debido a estas características, se hace fundamental el diseño de un marco de monitorización, almacenamiento, análisis y visualización de esta gran cantidad de datos con el objetivo de maximizar la escalabilidad. En este trabajo se presenta un modelo para la explotación de grandes volúmenes de datos de manera que permita la monitorización en tiempo real de datos procedentes de distintas fuentes, su almacenamiento, análisis y visualización para la ayuda a la toma de decisiones.
El artículo describe en la sección 2 las características de un lago de datos y el marco de trabajo en el que organizarlo. En la sección 3 se describe la implementación de dicho framework. Finalmente se terminan con unas conclusiones.

2 MODELO DE GESTIÓN DE UN LAGO DE DATOS

Un lago de datos alberga una gran cantidad de datos primarios en su formato nativo hasta que resultan necesarios [2]. Mientras que un depósito de datos jerárquico (datawarehouse) almacena datos en archivos o carpetas, un lago de datos utiliza una arquitectura plana para almacenar datos. Cada elemento de datos en un lago tiene asignado un identificador único y está marcado con un conjunto de etiquetas de metadatos extendidos. Cuando surge el análisis de un elemento, resulta posible efectuar una consulta al lago de datos en busca de datos relevantes y, al mismo tiempo, cabe la posibilidad de analizar dicho conjunto de datos más pequeño para ayudar a responder a la consulta.
Esta estructura plana de lago de datos se adapta bien a los datos de los cuales decidimos mantener la historia sin necesidad de saber de antemano qué análisis les serán aplicados. Manteniendo los datos en bruto y sin estructura, ninguna elección previa restringe las posibilidades posteriores de análisis. Los datos son almacenados en una forma de multitud de archivos distribuidos. Y es en el momento de la fase de análisis que los datos son reagrupados y que una eventual estructura es creada. Conservar, por ejemplo, los logs de un sitio web durante varios años, los tuits mencionando unos temas, los estados sociales, los comentarios de los blogs, las fotos etiquetadas, etc. todo esto sin saber previamente cómo estos datos serán cruzados en el futuro, son ejemplos de lago de datos
Con un lago de datos, simplemente basta volcar todos los datos, tanto los estructurados como los no estructurados, en el lago y luego permitir que las personas “destilen” sus propias visualizaciones particulares utilizando aquella tecnología que mejor se adapte a la tarea (por ej., SQL o NoSQL, bases de datos basadas en disco o en memoria, MPP o SMP.) Y el usuario crea sus visualizaciones de empresa mediante la compilación y agregación de datos desde múltiples vistas locales.
Pero a pesar de los muchos beneficios, los lagos de datos también vienen con riesgos, incluyendo la pérdida de contexto significativo, si estos datos no están debidamente gestionados. La gobernanza de los datos no incluye sólo la catalogación e indexación, así como la gestión de metadatos. Sino que es importante la estrategia, como la localización de quién se encarga de decidir qué datos se almacenan y cómo se realiza la definición adecuada de los datos.
En este trabajo se define un marco de desarrollo que establece los mecanismos para la monitorización, almacenamiento, análisis y visualización del lago de datos. Teniendo como objetivos que este modelado cubra las especificaciones de tiempo real y escalabilidad en el manejo de datos.
Hay muchas soluciones en el mercado para cubrir esta necesidad, tanto de pago como libres, pero una de las más populares es la plataforma Elasticsearch, Logstash y Kibana, conocida como ELK.
Esto es debido, en parte, a la escalabilidad que proporcionan los clusters de Elasticsearch, que puede manejar terabytes de datos sin ningún problema. Kibana, por su parte, aporta gran funcionalidad en la visualización de datos, pues no está limitado a gráficas de series temporales y puede manejar cualquier conjunto de datos.
Actualmente muchos sistemas genera un fichero de logs de su actividad (sistemas operativos, programas, aplicaciones móviles, redes, etc.) y uno de los principales usos de la plataforma ELK es el almacenamiento y análisis de estos. En ese contexto el papel de Logstash es el de agente de recogida y preparación, para su almacenamiento en Elasticsearch, de dichos logs, por lo que muchas

veces es necesario cuando se pone en marcha una plataforma de este tipo.
En ocasiones los datos deben ser procesados y analizados antes de visualizarlos. Además de que en muchos casos los datos no proceden directamente de un fichero de log, donde se han almacenado las transacciones que se realizan en un sistema, sino que se requiere ejecutar un proceso más complejo para extraer los datos a analizar. Entonces es muy útil disponer de una herramientas de gestión de flujos de trabajo que se encargue de ejecutar las tareas necesarias. Por lo tanto, la pila de gestión de un lago de datos estaría formado por un gestor de datos tipo ElasticSearch, un visualizador de datos como Kibana y un gestor de tareas para la recogida y análisis de datos (Figura 1) que en este caso será AirFlow. en lugar del componente Logstash de la pila ELK estándar.
La propuesta que se plantea en este trabajo es utilizar la herramienta Airflow para la monitorización y análisis de datos. Elasticsearch como contenedor del lago de datos. Y la herramienta visualizadora Kibana para la realización de los cuadros de mandos pertinentes.

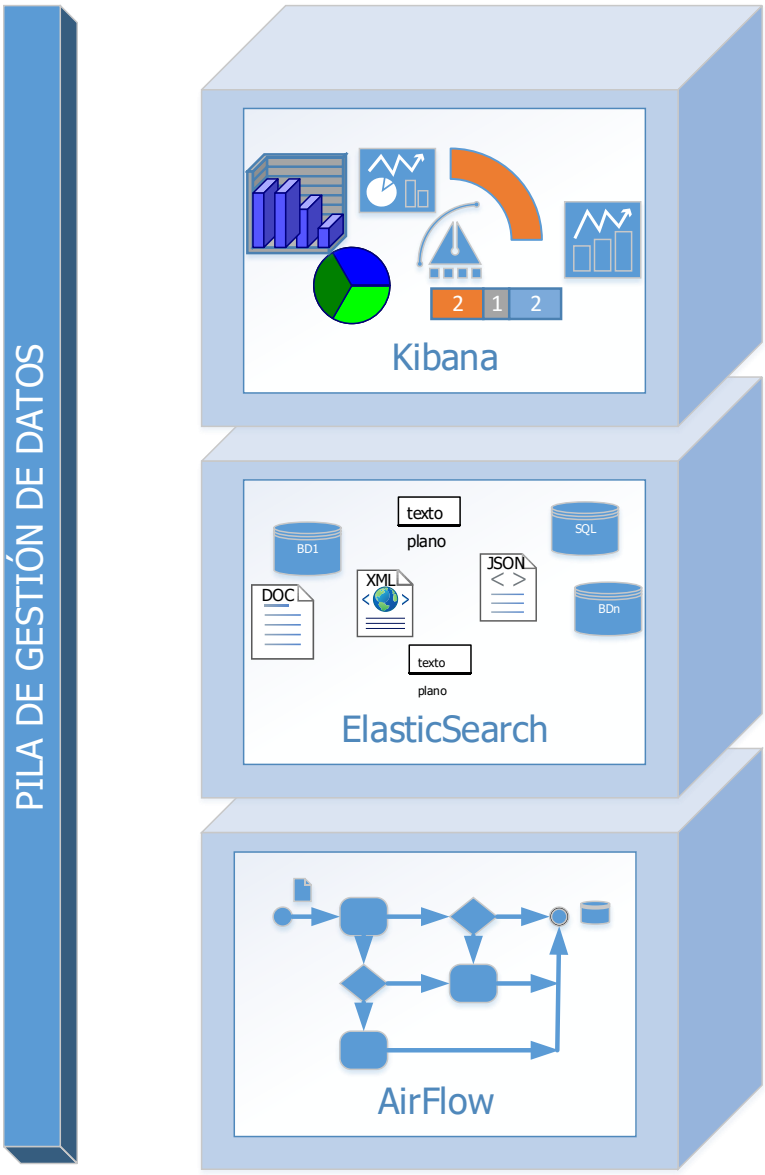

Figura 1.- Pila AEK (Airflow-ElasticSearch-Kibana) para la gestión de un lago de datos.

3 IMPLEMENTACIÓN DE LA PILA AEK EN REDES SOCIALES

3.1 AIRFLOW

Airflow es una herramienta que facilita la gestión de flujos de trabajo, permitiendo su programación, planificación y control. Tiene funcionalidades avanzadas que permiten a través de interfaz web explorar el conjunto de tareas programadas, pudiendo hacer un seguimiento a las que se han realizado con éxito o las que están pendientes de ejecución.

Así, para programar las tareas a realizar se define el grafo dirigido acíclico (DAG – Directed Acyclic Graph) donde se especifica el orden de las tareas a realizar. El encargado de ejecutar este grafo es el planificador que se asegura del cumplimiento de las dependencias entre tareas para respetar la causalidad.

Mientras que los DAG describen cómo ejecutar un flujo de trabajo (Figura 2), los operadores son los que determinan lo que realmente se hace. Un operador describe una única tarea en un flujo de trabajo. Los operadores son por lo general atómicos, lo que significa que no necesitan compartir recursos con otros operadores. El DAG se asegura de que los operadores se ejecuten en el orden correcto; aparte de esas dependencias los operadores generalmente se ejecutan de forma independiente.

El conjunto de tareas a realizar son la monitorización de los entornos donde extraer los datos (redes sociales, sistemas empresariales, dispositivos móviles, …). Tareas relacionadas con el análisis de estos datos, que pueden ser desde estadísticas hasta predicciones basadas en redes neuronales.

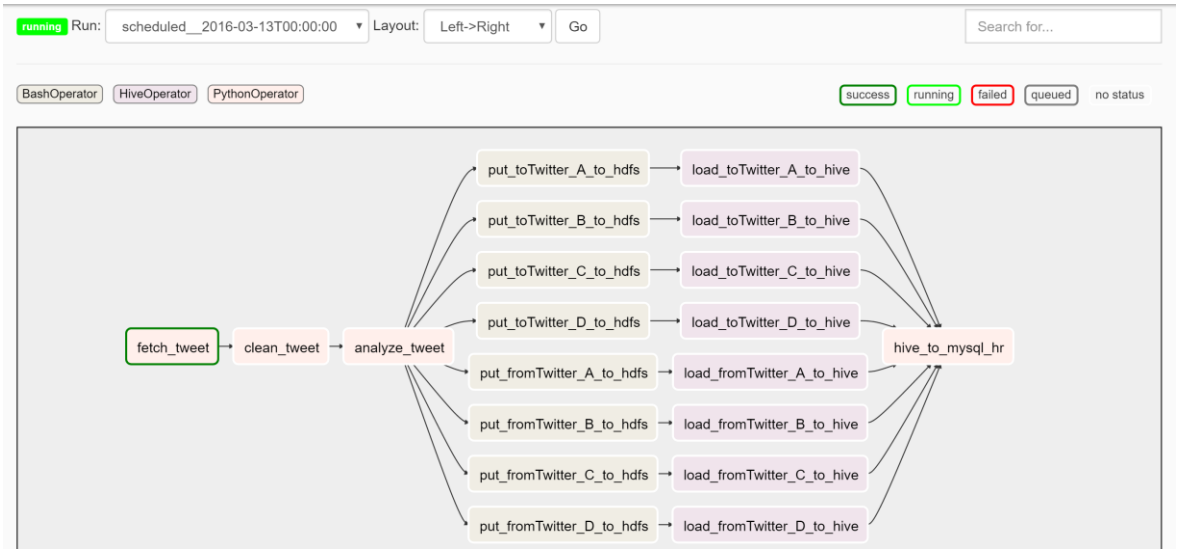

Figura 2.- DAG de Airflow con un flujo de trabajo para la extracción, transformación y almacenamiento (ETL) de mensajes en la red social Twitter.

3.2 ELASTICSEARCH

Es una herramienta, de software libre, que permite indexar y analizar en tiempo real grandes cantidades de datos de manera distribuida. Elasticsearch permite almacenar documentos (estructurados o no) e indexar todos los campos de estos documentos en tiempo casi real. Elasticsearch se basa en Lucene, pero expone su funcionalidad a través de una interfaz REST recibiendo y enviando datos en formato JSON y oculta mediante esta interfaz los detalles internos de Lucene. Esta interfaz permite que pueda ser utilizada por cualquier plataforma y no solo desde Java. En concreto puede usarse desde Python, .NET, PHP o incluso desde un navegador con Javascript. Es persistente, es decir, que lo que indexemos en ella sobrevivirá a un reinicio del servidor.

3.3 KIBANA

La representación visual del dato es una tarea fundamental para obtener el valor oculto que tienen los datos. Un gran esfuerzo en un proyecto de tratamiento de datos, integración y depuración, etc., puede no servir para la toma de decisiones si finalmente los datos no se visualizan apropiadamente y el usuario debe poder sacar conclusiones con ellos.

Las buenas representaciones gráficas, deben cumplir una serie de características [3]:

- Señalar relaciones, tendencias o patrones

- Explorar datos para inferir nuevo conocimiento
- Facilitar el entendimiento de un concepto, idea o hecho
- Permitir la observación de una realidad desde diferentes puntos de vista
- Y permitir recordar una idea.

Se propone para la visualización de los datos Kibana. Es una herramienta de software libre, perteneciente a Elastic, que nos permite hacer análisis, búsquedas, visualizar y explorar datos que se encuentran indexados en Elasticsearch. Kibana ofrece un interfaz muy potente para crear cuadros de mando a medida, con características como personalización, selección de rangos, drill down, además de poder compartirse y guardarse (Figura 3).

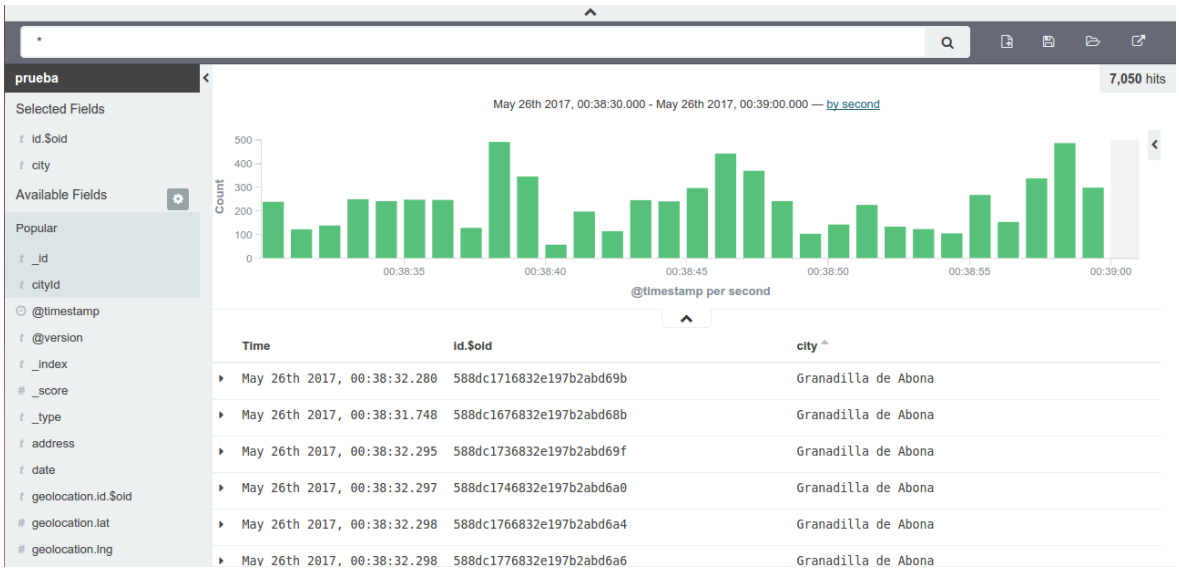

Figura 3.- Interface de Kibana

3.4 DOCKERS

Una de las cuestiones a resolver cuando nuestra aplicación está lista para desplegar en producción es cómo monitorizar su funcionamiento para poder actuar de manera rápida y efectiva ante cualquier incidencia. Hay que trabajar con elementos estructurados que nos permitan volver a estados estables cuando detectamos errores, además que nos facilite su despliegue. Para ello proponemos el despliegue de cada uno de los elementos de la pila AEK en contenedores.

Para el desarrollo de estos contenedores utilizamos la herramienta Docker, que empaqueta una pieza de software, su entorno de ejecución, sistema de ficheros, sus dependencias, etc. para ser ejecutadas por esta misma herramienta de forma aislada del resto del sistema. Este empaquetado es conocido como imagen, mientras que a una imagen en ejecución se la conoce como container. Se trata de un sistema de ejecución muy ligero ya que el container comparte el kernel del sistema operativo que hay por debajo, evitando necesitar un sistema operativo completo, como ocurre con las máquinas virtuales.

Para su funcionamiento Docker internamente está basado en LXC (LinuX Containers) que utiliza capacidades de Linux como son los *cgroups* y *namespaces* para que los procesos se ejecuten en entornos seguros. Está compuesto de 3 piezas principales que se conocen como Docker Engine:

- Docker Daemon: Entorno de ejecución de containers que corre sobre el anfitrión.
- Docker Client: Herramienta de linea de comando para comunicarse con Docker Daemon y que ejecuta los comandos de compilación, ejecución, etc.
- Rest API: Similar al Docker Client, permite controlar el Docker Daemon de manera remota.

Además del Docker Engine, existen otras herramientas que completan el ecosistema de Docker.

- Dockerfile: es un fichero que define el contenido de una imagen, así como qué procesos han de ejecutarse dentro de ella, si expone algo al exterior (volúmenes, puertos, etc.).
- Dockerhub Public Registry: Las imágenes, una vez construidas, se almacenan en el disco duro local. Para que otros usuarios las puedan usar, Docker ofrece un repositorio público de imágenes llamado Dockerhub. Creando una cuenta en este servicio y con un comando del

Docker Client, es posible subir y compartir las imágenes que hayamos creado con otros desarrolladores, así como descargar las de otros para tu uso personal.

- Docker-compose: es a la vez un fichero y una herramienta. Como fichero, define un conjunto de imágenes que van a trabajar de manera conjunta. Define cómo se comunican entre ellas, variables de entorno, volúmenes y puertos a exponer. Como herramienta, lee la definición del fichero y la ejecuta sobre el Docker Engine, atendiendo a las interdependencias definidas.

Se debe implementar Docker para cada una de las aplicaciones que forman la pila. En la forja Github podemos encontrar la distribución del Docker para el gestor de flujos de trabajos Airflow (Figura 4).

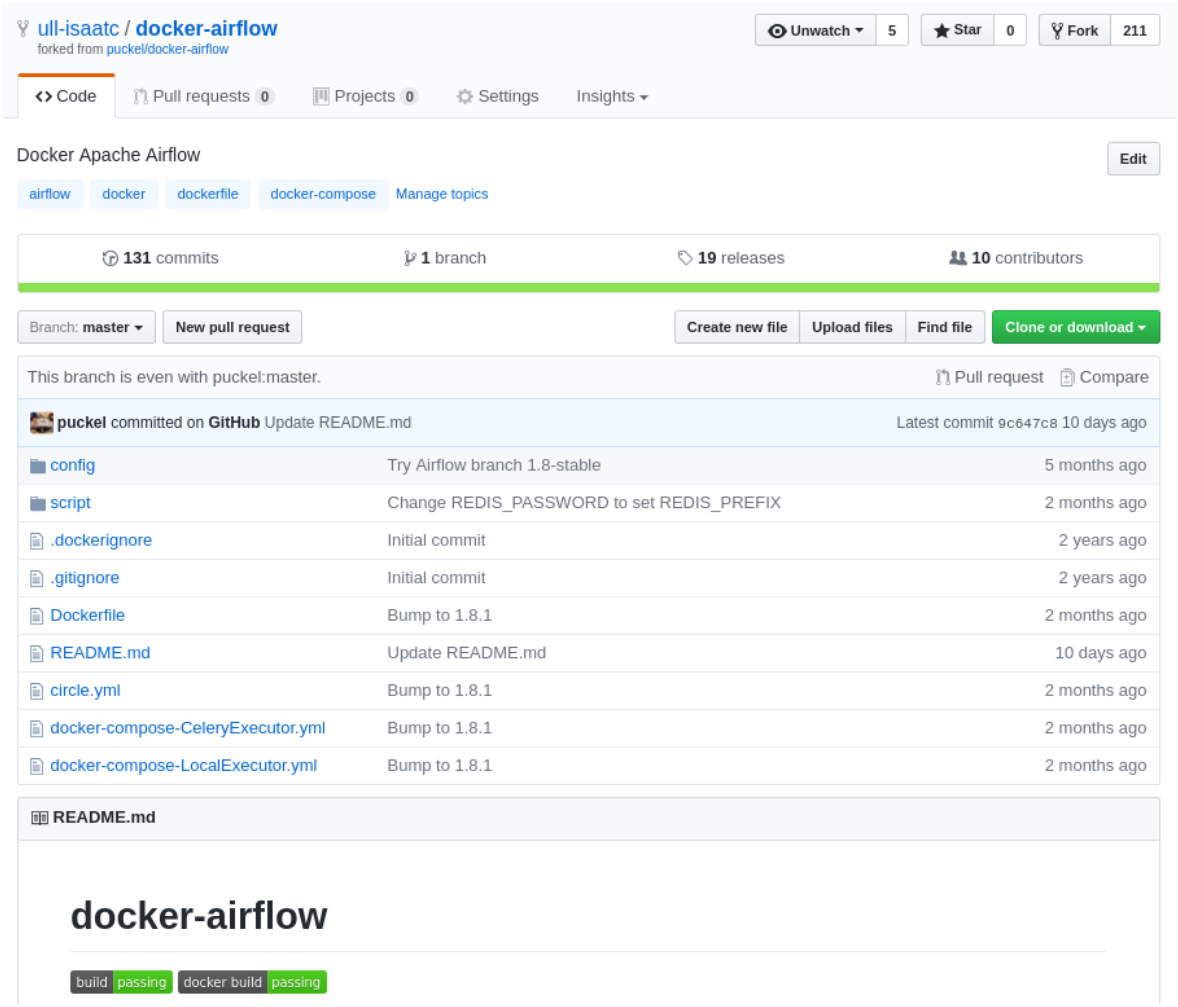

Figura 4.- https://github.com/ull-isaatc/docker-airflow

4 CONCLUSIONES

Actualmente hay una explosión de almacenamiento, análisis y visualización de big data en grandes repositorios de datos estructurados y sin estructura, comúnmente llamados lago de datos. Estos grandes volúmenes de datos requieren de nuevas técnicas de integración de datos y alineación de esquemas para hacer que los datos sean utilizables por sus consumidores y para descubrir las relaciones que vinculan su contenido. Sin embargo, actualmente no existe un enfoque sistemático para este tipo de procesamiento y gestión del lago de datos. En este trabajo, proponemos un marco para la monitorización, almacenamiento, análisis y visualización analítica de los datos existentes en un lago de datos. Datos procedentes de diferentes fuentes que se integran para ayudar a la toma de decisiones, encuentran el valor que encierra tal cantidad de datos.

Se define para la gestión de este big data una pila de servidores cuyas características deseadas es su gran escalabilidad y que sean de software libre. Se propone, como primer elemento, una herramienta de gestión de flujos de trabajo que periódicamente lance tareas para realizar tanto labores de monitorización de datos, para incluirlos en la lago de datos, como tareas de análisis de estos datos. Para dicho trabajo se utiliza Airflow, una herramienta que ejecuta tareas periódicamente, planificando el orden en el que se ejecutan las mismas. En medio de la pila nos encontramos con el repositorio de los datos, un servidor que provee un sistema de búsqueda de datos estructurados y no estructurados. Se propone el uso de Elasticsearch, que tiene la capacidad de multitenencia, con una interfaz web RESTful y con documentos JSON. Finalmente se hace uso de un visualizador que nos permita representar los datos de una forma amigable al usuario, para lo que se utiliza el servidor de Kibana.

Como propuesta para el mantenimiento de dicha estructura o pila AEK (Airflow-ElasticSearch-Kibana) se empaqueta cada servidor en un contenedor (concretamente utilizamos Docker) que nos permite el despliegue ágil del marco propuesto.

Agradecimientos

Este trabajo ha sido financiado por la Fundación CajaCanarias a través del proyecto titulado "VITUIN: Vigilancia Turística Inteligente de Tenerife en Redes Sociales" nº2016TUR15.

Referencias

[1] Natalia Miloslavskaya, Alexander Tolstoy, Big Data, Fast Data and Data Lake Concepts, Procedia Computer Science, Volume 88, 2016, Pages 300-305, ISSN 1877-0509

[2] Laskowski, N. (2016). Data lake governance: A big data do or die. URL: http://searchcio.techtarget.com/feature/Data-lake-governance-A-big-data-do-or-die (access date 28/05/2016)

[3] Edward Tufte, *The Visual Display of Quantitative Information*, Graphics Press USA;2nd edition, 2001