



TRABAJO FIN DE MÁSTER ARQUITECTURA BIG DATA

MÁSTER UNIVERSITARIO EN BUSINESS ANALYTICS Y BIG DATA

Docente. Juan Manuel Moreno Lamparero

Curso académico 2018-2020

MÁSTER UNIVERSITARIO EN BUSINESS ANALYTICS Y BIG DATA

Contenido

1. INTRODUCCIÓN
2. OBJETIVO
3. FUENTES DE DATOS
4. PLANTEAMIENTOS INICIALES
2. ANÁLISIS DE SOLUCIONES CLOUD
3. RESULTADOS ESPERADOS
4. CREACIÓN Y CONFIGURACIÓN DE MV
4.1 CREACIÓN DE CLÚSTER KAFKA + ZOOKIPER9
4.1.1 Crear Instancias GCP
4.1.2 Instalación Apache Kafka
4.1.3 Configuración Zookeeper 11
4.1.4 Configuración Kafka Brokers
4.1.5 Creation de Topic
4.1.6 Clientes Apache Kafka 15
4.2 CREACIÓN DE CLÚSTER ELASTIC + KIBANA
4.2.1 Creación Instancias GCP 15
4.2.2 Instalación ElasticSearch
4.2.3 Configuración ElasticSearch
4.2.4 Instalación Kibana
4.2.5 Configuración Kibana
5. DESARROLLO Y EJECUCIÓN DE SCRIPTS
5.1 DATOS
5.2 CAMPOS DEL CONJUNTO DE DATOS
5.3 SCRIPT DESCARGA DE DATOS Y PUBLICACIÓN EN KAFKA
5.4 SCRIPT CONSUMER KAFKA E INSERT EN ELASTICSEARCH
6. RESULTADOS
7. CONCLUSIONES
8. ASIGNATURAS/MÓDULOS RELACIONADOS
9. MÉTODOS, MATERIALES Y TECNOLOGÍAS

1. INTRODUCCIÓN

Hoy en día muchas empresas tecnológicas trabajan para optimizar los recursos y costes de las empresas y de la misma forma buscan una manera para mejorar la ecología global. Si solo nos fijamos en los camiones de logística optimizando sus rutas no solo podemos mejorar las horas trabajadas, consumo de combustible, ahorro en recursos, sino también la contaminación de la atmósfera.

Muchos analistas de datos trabajan con los conocidos algoritmos de los problemas de rutas de vehículos (Vehicle Routing Problem - VRP) utilizando las librerías de **OpenStreetMap OSRM** y **OR-TOOLS** de **Google**, pero para planificar las rutas optimizadas tenemos que predecir el tráfico para calcular mejor dichas rutas. ¿Podemos predecir el tráfico que vamos a encontrar dentro de un mes? Nos es una tarea fácil ni rápida, necesitamos el histórico de datos de trafico de los años anteriores meses, semanas horas e incluso minutos.

2. OBJETIVO

Mi objetivo principal con este proyecto no es la adquisición, procesado y analítica de datos, es crear la arquitectura Big data distribuida y tolerante a fallos, decido obtener los datos en tiempo real cueles puedo compartir con mis compañeros de trabajo una vez terminado mi proyecto.

3. FUENTES DE DATOS

En el portal OpenData oficial de DGT encuentro los datos históricos desde 2013 cuales puedo descargar para las predicciones, también encuentro datos en el tiempo real (según la DGT: la información se actualiza casi en tiempo real, con una periodicidad de unos 5 minutos, que es el tiempo mínimo de varios ciclos de semáforo, necesarios para dar una medición real, y que no se vea afectada la medición por si el semáforo está abierto o cerrado.), con 0estos datos podemos mantener nuestros datos siempre actualizados.

4. PLANTEAMIENTOS INICIALES

Antes de la solución definitiva, tenía varias ideas y diseños que finalmente fueron desestimados por diferentes razones. Se detallan brevemente los más relevantes.

4.1 ARQUITECTURA LAMBDA VS ARQUITECTURA KAPPA

Existen dos tipos de arquitecturas que son las más comunes en proyectos con tecnologías Big Data. Por un lado, están las arquitecturas Lambda, en las que se plantea una parte para el tratamiento de la información en batch, y otra para la parte online. En la primera tendríamos la información en crudo, sin modificar, e iríamos acumulando los datos, pudiendo cargarlos con "snapshots" o situaciones a final del día desde el sistema actual, o directamente con streaming, pero sin procesar, mientras que en la segunda nos centraríamos en la información online, los datos nuevos, y que en esta parte procesaríamos. Por otro, tenemos las arquitecturas Kappa, en las que se prescinde de la parte batch, y procesamos únicamente datos en streaming continuo.

En un principio he pensado en el diseño de una arquitectura Lambda, que permitiera el almacenamiento de la información sin tratar, en un sistema de ficheros distribuido, como HDFS, y que se utilizará para procesos distintos (analítica y predicción) a los que explotarían la capa de streaming (métricas e informes necesarios para el negocio).

Finalmente, esta solución se descartó, ya que implicaba varios problemas:

- Sincronización de la información en todos los puntos: Al tener el sistema original, la capa online, y la capa batch, obligaba a establecer mecanismos y controles para evitar inconsistencias y pérdida de calidad de los datos.
- La complejidad de la arquitectura se incrementaba: Esta solución obligaba al uso de más herramientas, más esfuerzo y más recursos.

Este planteamiento podría realizarse en un equipo con más recursos.

4.2 BBDD ONLINE & BBDD HISTÓRICA

En la mayoría de los sistemas suelen existir dos niveles de explotación de la información, una primera que requiere de los datos más actuales, y que aglutina un alto porcentaje de las consultas, y otra que puede hacer uso de datos históricos, y que generalmente es para accesos más residuales. Por ello, inicialmente se trató de llevar esos dos niveles a la arquitectura diseñada, estableciendo una BBDD para la información reciente, y otra para el histórico.

Al igual que pasó con la arquitectura Lambda, se descartó, por las siguientes razones:

- Aumento de costes y el uso de herramientas con posibilidad de soporte, la opción de dos BBDD, con más licencias necesarias, aumenta el coste de la solución.
- Al igual que el punto anterior, más herramientas implicaba más esfuerzo en la implantación.

Sin embargo, este punto si es interesante, no para este proyecto, pero si para evaluarlo sobre una solución más amplia a largo plazo.

4.3 HERRAMIENTAS REAL TIME

Para ello he decido utilizar dos componentes principales: un gestor de colas que permita contener y no saturar el soporte final dónde se almacenarán los datos, y un framework de procesamiento distribuido que ofrezca la posibilidad de tratar eso datos previamente a su almacenamiento sin retardo en la inserción final. Una vez decidida la naturaleza de las herramientas, analizo diversas alternativas, decidiendo finalmente el uso de Apache Kafka como gestor de colas (por ser una de las más potentes y cuyo uso es bastante extendido) y Apache Spark como motor de procesamiento (permite programación en Python).

4.4 ALMACENAMIENTO

Para ello decido utilizar Elasticsearch es almacenamiento de documentos distribuidos. En lugar de almacenar información como filas de datos en columnas, Elasticsearch almacena estructuras de datos complejas que se serializa como documentos JSON.

Elasticsearch es un motor de analítica y análisis distribuido y open source para todos los tipos de datos, incluidos textuales, numéricos, **geoespaciales**, estructurados y **desestructurados**. La velocidad y escalabilidad de Elasticsearch y su capacidad de indexar muchos tipos de contenido significan que puede usarse para una variedad de casos de uso.

- Está muy bien integrada con otras dos aplicaciones de la misma compañía es rápido
- Distribuido por naturaleza
- Puede dar respuesta a múltiples clientes.
- Los documentos y configuración se pueden modificar y manipular mediante consultas HTTP. y, además, trabaja con ficheros con formato JSON.

4.5 VISUALIZACIÓN

Como Elasticsearch está muy bien integrada con la aplicación de la misma compañía, en este caso no hoy duda la decisión está clara: Kibana.

Kibana es una herramienta de visualización y gestión de datos para Elasticsearch que brinda histogramas en tiempo real, gráficos circulares y mapas. Kibana también incluye aplicaciones avanzadas, como Canvas, que permite a los usuarios crear infografías dinámicas personalizadas con base en sus datos, y Elastic Maps para visualizar los datos geoespaciales.

4.6 RESUMEN DE LA DECISIÓN TOMADA SOBRE LA ARQUITECTURA Y LA LAS HERRAMIENTAS A UTILIZAR EN EL PROYECTO.



Img 1. Arquitectura Kappa. Elaboración propia.



2. ANÁLISIS DE SOLUCIONES CLOUD

Una vez se decide la definición de la arquitectura en un entorno Cloud, comienzo un periodo de análisis y pruebas sobre los recursos ofrecidos por algunos de los proveedores.

Partiendo de un presupuesto de cero euros, lo que implica aprovechar algunas de las opciones de cuentas gratuitas que me ofrecen, y ver si con dichas cuentas es suficiente para dar soporte a las necesidades del proyecto.

Se evalúan: AWS, Azure y Google Cloud Platform, ya que son las que actualmente tienen más clientes, al constar del más amplias funcionalidades ofrecidas.

2.1 MICROSOFT AZURE

Esta plataforma dispone de un amplio abanico de servicios y plantillas de arquitecturas para proyectos Big Data. En esta plataforma hay la posibilidad de obtener una cuenta gratuita con un crédito de 170 € los 30 primeros días, para montar un clúster en Azure no requiere ningún esfuerzo, solo hay que seguir los pasos, y el Azure todo hace por ti (configuración de los nodos). **Mi objetivo es configurar un clúster real desde 0**, configurar manualmente para entender mejor su funcionamiento probar que ocurre cuando uno de los nodos se cae, por ejemplo.

2.2 AWS

Esta plataforma se ha utilizado en clase para la realización de algunas prácticas ofrece todas las funcionalidades necesitadas. Se descarta solo por el motivo de haberlo visto a lo largo del curso.

2.3 GOOGLE CLOUD PLATFORM

GCP ofrece multitud de servicios y recursos para el desarrollo de aplicaciones y sistemas. Con las cuentas gratuitas con un crédito de 300\$ por año, puedo, a través de Google Compute Engine, definir las máquinas virtuales con los recursos que necesito.

Al realizar las pruebas de concepto el único punto a tener en cuenta es que todas las maquinas debo de pararlas en el momento en el que dejo de trabajar con ellas, para evitar consumir el saldo de la cuenta. Esto implica la renovación de las IPs externas, por lo que algunas configuraciones existentes se han intentado establecer con IPs internas, aunque no siempre ha sido posible.

Por otro lado, existe alguna restricción en cuanto a los recursos que se pueden utilizar, con las cuentas gratuitas, en las diversas zonas geográficas, por lo que mis máquinas están repartidas por todo el mundo. Esto también ha incrementado el coste de la solución, ya que el precio no es el mismo en todas ellas.





2.3 Resumen de la decisión tomada sobre infraestructura.

Img 2. Infraestructura. Elaboración propia.

Finalmente decido montar dos Clústeres en la plataforma Google Cloud Platform.

Para ello, y tras numerosas pruebas y reajustes, la infraestructura es la siguiente:

- Tres nodos para Apache Kafka, uno de los cuales compartirá recursos con Apache Zookeeper.
- Tres nodos para Elastic uno de los cuales compartirá recursos con Kibana
- Para los scripts de Producer y Consumer de Kafka utilizare los clústeres de Spark en DataBricks para ahorrar el tiempo, utilizare dos cuentas diferentes para ver la ejecución de los scripts en paralelo.

3. RESULTADOS ESPERADOS

- Montar los clústeres desde cero en la nube, configurar correctamente, en caso de Clúster de Kafka con Zookeeper configurar correctamente los Brokers, Productores y Consumidores. Tener visibilidad entre los nodos del mismo Clúster y tener acceso a los servicios desde un cliente externo.
- 2. Con PySpark, hacer una pequeña transformación de datos obtenidos desde el portal de la DGT, los datos solo se pueden descargar en el formato XML con las coordinadas en el formato UTM del centroide que representa al punto de medida en el fichero georreferenciado, por lo cual es necesario pasar el fichero XML a JSON (en mi caso geoJson) y convertir los datos geográficos a EPSG (European Petroleum Survey Group) con los Sistemas de Referencia de Coordenadas habitualmente usados en los Servicios Web de Mapas (WMS).
- 3. Una vez esta convertido al objeto geoJson publicar en un Topic.
- 4. En paralelo orto script **Spark Streaming** tiene que ser capaz de **recibir los mensajes** de mismo Topic e insertar en Elastic.
- 5. Finalmente crear un informe en Kibana para visualización y análisis de los datos obtenidos.

4. CREACIÓN Y CONFIGURACIÓN DE MV

Creamos una cuenta gratuita en Google Cloud Platform <u>https://cloud.google.com/</u>. Desde la consola Menu \rightarrow Compute Engine \rightarrow Instancias de VM

4.1 CREACION DE C	LUSTER KAFKA + ZOOKIPER
4.1.1 Crear Instar	ncias GCP
Creamos nueva instancia com	o se muestra en la siguiente imagen
	Configuración de la máquina
	Familia de máquinas
	Uso general Con memoria optimizada
	Tipos de máquinas para cargas de trabajo habituales, optimizadas en cuanto al coste y a la flexibilidad
	Serie
	N1 -
	Con la tecnología de la plataforma de CPU Intel Skylake o de uno de sus predecesores
	Tipo de máquina
	n1-standard-4 (4 vCPU, 15 GB de memoria) 🔹
	VCPU Memoria 4 15 GB
	V Plataforma de CPU y GPU
	Contenedor Contenedor Contenedor en esta instancia de VM. Más información
	Disco de arranque 🔞
SS -	Nuevo disco persistente estándar de 20 GB Imagen CentOS 7 Cambiar
	Identidad y acceso de API 🛞
	Cuenta de servicio 🔞
	Compute Engine default service account
	Alcance del acceso Permitir el acceso predeterminado Permitir el acceso completo a todas las API de Cloud Definir acceso para cada API

Img 3 Creación de la nueva instancia.

Creamos 3 instancias como se muestra en la Img 3, una vez terminado INICIAMOS las 3 instancias.



Instancias de VM	CREAR INSTANCIA	🗄 IMPORTAR VM	C ACTUALIZAR	► INICIAR	DETEN	ER () RESTABLE
= Estado : En ejecución (Filtrar las instancias de VM				×	Colu	imnas 🔻
Nombre ^	Zona Recom	endación Usada por	IP interna	IP externa		Conecta	r
🔮 kafka-node2	europe-west1-b		10.132.0.14 (nic0)	35.195.167	.204 🖸	SSH	- :
🔮 kafka-node3	europe-west1-b		10.132.0.15 (nic0)	104.155.58	.76 🖓	SSH	- :
🗌 🔮 zookeeper-kafka-node1	europe-west1-b		10.132.0.13 (nic0)	35.195.134	.228 🖸	SSH	- :

Img 4 Instancias Iniciadas.

Clúster	Nodo GCP	Herramientas	CPUs	RAM	Zona	Disco	S.O.
Apache	zookeeper-kafka-nodo1	Zookeeper + Kafka	4	15	europe-west1-b	10 GB	Centos 7
Kafka							
Apache	kafka-nodo2	Kafka	4	15	europe-west1-b	10 GB	Centos 7
Kafka							
Apache	kafka-nodo3	Kafka	4	15	europe-west1-b	10 GB	Centos 7
Kafka							

4.1.2 Instalación Apache Kafka

Entramos a la consola de las 3 máquinas pulsando el botón SSH, y primero modificamos el fichero hosts en las 3 MV con el comando vim /etc/hosts

Y añadimos en las 3 máquinas

10.132.0.13 zookeeper-kafka-node1.europe-west1-b.c.proyectotfm-

268719.internal zookeeper-kafka-node1

10.132.0.14 kafka-node2.europe-west1-b.c.proyectotfm-268719.internal kafkanode2

10.132.0.15 kafka-node3.europe-west1-b.c.proyectotfm-268719.internal kafkanode3

Instalamos Apache Kafka en las 3 Instancias

```
$ sudo yum -y install java-1.8.0-openjdk
$ sudo yum -y install wget
$ wget http://apache.uvigo.es/kafka/2.4.1/kafka_2.11-2.4.1.tgz
tar -xzf kafka 2.11-2.4.1.tgz
```

Modificamos el bash_profile en las 3 MV

\$ vim .bash_profile :\$HOME/kafka_2.11-2.4.1/bin \$ source .bash profile

4.1.3 Configuración Zookeeper

Configuramos **Zookeeper**, para ello modificamos fichero de configuración de **Zookepeer**, solo en la maquina donde vamos a arrancar el **Zookeeper**

\$ vi kafka_2.11-2.4.1/config/zookeeper.properties

Cambiamos la propiedad

dataDir= /home/a_l_es/zookeeper_data

\$ mkdir /home/a_l_es/zookeeper_data

\$ mkdir /home/anastasia_lukina_es/zookeeper_data

Arrancamos el **Zookeeper**

\$ zookeeper-server-start.sh kafka_2.11-2.4.1/config/zookeeper.properties

Como se comentado anteriormente para ahora los costes de consumo de crédito gratuito tengo que apagar más instancias una vez que dejo de trabajar con ellas. Para no volver a arrancar a mano cada vez que arranco la instancias, añado al fichero rc.local

\$ sudo vi /etc/rc.d/rc.local

Añadimos siguiente línea:

```
/home/a_l_es/kafka_2.11-2.4.1/bin/zookeeper-server-start.sh
/home/a_l_es/kafka_2.11-2.4.1/config/zookeeper.properties > /dev/null 2>&1 &
```

\$ sudo chmod +x /etc/rc.d/rc.local
\$ sudo systemctl enable rc-local

\$ sudo systemctl start rc-local



4.1.4 Configuración Kafka Brokers

Configuramos Kafka Brokers en las 3 MV

A diferencia de **Zookeeper**, cambiaremos más propiedades de configuración para los **Kafka Brokers**.

Configuración	Kafka-node1	Kafka-node2	Kafka-node3				
broker.id	0	1	2				
broker.rack	RACK1	RACK1	RACK2				
listeners	PLAINTEXT://zooke eper-kafka- node1:9092	PLAINTEXT://kafka- node2:9092	PLAINTEXT://kafka- node3:9092				
advertised.listeners	PLAINTEXT://35.195 .134.228:9092	PLAINTEXT://35.195.167.20 4:9092	PLAINTEXT://35.195.167.20 4:9092				
log.dirs		/home/a_l_es/kafka_dat	a				
offsets.topic.num.partitions		3					
offsets.topic.replication.fac		2					
tor							
default.replication.factor	2						
zookeeper.connect		10.132.0.13:2181					

* advertised.listeners IPs Externas (una de las primeras problema encontradas)

Creamos un directorio para guardar los logs

\$ mkdir /home/anastasia_lukina_es/kafka_data

Modificamos el fichero de configuración de Kafka server.properties

\$ vi kafka_2.11-2.4.1/config/server.properties

🕲 anastasia_lukina_es@zookeeper-kafka-node1:~ - Google Chrome — 🗆 🗙	🕲 anastasia_lukina_es@kafka-node2:~ - Google Chrome — 🗆 🗙	🐵 anastasia_lukina_es@kafka-node3:~ - Google Chrome — 🗆 🗙
ssh.cloud.google.com/projects/proyectotfm-268719/zones/europe-west1-b/instances/	ssh.cloud.google.com/projects/proyectotfm-268719/zones/europe-west1-b/instances/ka	ssh.cloud.google.com/projects/proyectotfm-268719/zones/europe-west1-b/instances/kafka-node3
<pre>incket_rack=reference. intervention intervention int</pre>	1 The maximum size of a log segment file. When this size . A size and the consequent will be created. log.segment.bytes=_0777701001 The interval at which log segments are checked to see if they can be deleted according to the retention policies log.retention policies log.retention policies log.retention policies 2000keeper connection string (see zookeeper docs for details). This is a comma separated host-port pairs, each corresponding to a zk source, e.g. "127,0.0.1:3000,127.0.0.1:3002". Froot Girectory for all kafks modes. zookeeper.connection.timeout.ms=_000	<pre>the brokers. num.partitions=:</pre>
etails 40,1 17%		86,1 54%
	136,0-1 94%	

Img 5. config/server.properties de las 3 MV

Arrancamos Kafka, manualmente y añadimos al rc.local para no arrancar manualmente cada vez que apagamos y arrancamos las MV

```
$ kafka-server-start.sh kafka_2.11-2.4.1/config/server.properties
$ kafka-server-stop.sh kafka_2.11-2.4.1/config/server.properties
$ sudo vi /etc/rc.d/rc.local
```

Añadimos al rc.local en las 3 MV

```
/home/a_l_es/kafka_2.11-2.4.1/bin/kafka-server-start.sh
/home/a_l_es/kafka_2.11-2.4.1/config/server.properties > /dev/null 2>&1 &
```

- \$ sudo chmod +x /etc/rc.d/rc.local \$ sudo systemctl enable rc-local
- \$ sudo systemctl start rc-local

Antes de comprobar la instalación tenemos que establecer la conexión SSH entra las máquinas, para ello hay que generar la clave pública y privada

```
$ cd .ssh
$ ssh-keygen
$ cat id_rsa.pub
```

Copiamos la clave y añadimos a metadatos Claves SSH







Antes Ahora ya podemos comprobar la instalación de Apache Kafka

Ahora ya podemos comprobar la instalación de Apache Kafka

\$ zookeeper-shell.sh zookeeper-katka-node1:2181 Is /brokers/ids
[anastasia_lukina_es@kafka-node3 ~]\$ vi kafka_2.11-2.4.1/config/server.properties
[anastasia_lukina_es@kafka-node3 ~]\$ sudo systemctl restart rc-local
[anastasia_lukina_es@kafka-node3 ~]\$ zookeeper-shell.sh zookeeper-kafka-node1:2181 ls /broker s/ids
Connecting to zookeeper-kafka-node1:2181
WATCHER::
WatchedEvent state:SyncConnected type:None path:null
[1]
[anastasia lukina es@karka-nodes ~]\$ vi karka 2.11-2.4.1/coning/server.properties
[anastasia lukina es@karka-node3 ~]\$ sudo systemcti restart rc-local
[anastasia_lukina_es@karka-node3 ~]\$ zookeeper-snell.sn zookeeper-karka-nodel:2181 18 /broker s/ids
Connecting to zookeeper-kafka-node1:2181
WATCHER::
WatchedEvent state:SyncConnected type:None path:null
[anastasia_lukina_es@kafka-node3 ~]\$

Img 7. Brokers Kafka

4.1.5 Creation de Topic

Creamos un Topic

\$ sudo /home/anastasia_lukina_es/kafka_2.11-2.4.1/bin/kafka-consoleproducer.sh --broker-list localhost:9092 --topic dgt

Hasta aquí todo va bien, vamos a probar a conectar con un cliente para comprobar si puedo conectar a Kafka con un cliente externo.

Para poder de conectar con un cliente externo tenemos que configurar las reglas de cortafuegos. Para ello desde la consola de Google Cloud Platform (GCP) desde menu \rightarrow Red VPC \rightarrow Reglas de cortafuegos, Creamos dos nuevas reglas.

tcp2181 y tcp 9092

kafka	Entrada	http-server	Intervalos de IPs: 0.0.0.0/0	tcp:2181	Permitir	1000	default
kafkaproducer	Entrada	http-server	Intervalos de IPs: 0.0.0/0, 10.132.0.0/24	tcp:9092	Permitir	1000	default

Img 8. Reglas de cortafuegos



4.1.6 Clientes Apache Kafka

Vamos a descargar Kafka Tool una aplicación GUI para administrar y usar clústeres de Apache Kafka

Wafka Tool 2.0.7		
<u>File Edit T</u> ools <u>H</u> elp		
Sa Browser X		
	Properties Data Partitions Config	
Clusters Arrow kafka		
Brokers Subscriptions Detribution 2 Detribution 2		0

Img 9. Kafka Tool.

Como podemos comprobar tenemos nuestro Clúster con **3 Brokers**, nuestro **Topic dgt** con **3 particiones**.

Vamos a descargar Conduktor Cliente de Apache Kafka de Escritorio https://www.conduktor.io/

CONDUKTOR		O CONSUMER	O PRODUCER	. •	Read-only OFF	Options	•
🗞 kafka	BROKERS				Roll	ing Restart	
A Overview	3 Yes +2.3.x Yes Brokers Controller Version Similar Config				Rebai	ance Gluster.	
😅 Brokers	3 ID Rack Listener	Partitions	Skew 0	Leaders	Skew 0	Size	
📑 Topics	2 0 RACK1 35.195.134.228:9092	3	+13%	1	-25%	616 B	•
t 0	1 RACK1 ★ 35.195.167.204;9092	1	-62%	0	-100%	5,7 MB	0
E Consumers	0 2 RACK2 104.155.58.76:9092	4	+50%	3	+125%	5,7 MB	•

Img 10. App Conduktor

Ya tenemos nuestro Clúster bien configurado, arrancado y listo para poder utilizar desde cualquier cliente.

¡NOTA! La seguridad no ha sido un punto en el que se ha hecho especial hincapié, en cuanto a configuraciones establecidas en este proyecto, pero sí lo son para futuras implementaciones en entornos corporativos.

4.2 CREACIÓN DE CLÚSTER ELASTIC + KIBANA

4.2.1 Creación Instancias GCP

Creamos 3 nuevas instancias con las mismas características como se muestra en la **Img 3. Creación** de la nueva instancia.

MÁSTER UNIVERSITARIO EN BUSINESS ANALYTICS Y BIG DATA

Instancias de VM	CREAR IN	STANCIA 🛃	IMPORTAR VM	C ACTUALIZAR	► INICIAR	DET	ENER	🖑 RESTABLECER	ELIMINAR
🌻 Se podría cambiar el tamañ	o de 3 instancias para	a ahorrar aproxima	damente hasta 63	\$ al mes. Más información					Cerrar todas
= Estado : En ejecución (Filtrar las instanc	ias de VM				×	Colu	imnas 💌	
Nombre A	Zona	Recomendación	Usada por	IP interna	IP externa		Conecta	r	
🗌 🔮 elasticsearch-node1	europe-west2-c			10.154.0.5 (nic0)	35.242.150.103	Ľ	SSH	- :	
elasticsearch-node2	europe-west1-b			10.132.0.10 (nic0)	35.205.76.179	2	SSH	- :	
🗌 🔮 kibana-node3	europe-west1-b			10.132.0.12 (nic0)	34.76.35.96 🖾		SSH	- :	

Clúster	Nodo GCP	Herramientas	CPUs	RAM	Zona	Disco	S.O.
Elastic	elasticsearch-node1	elasticsearch	4	15	europe-west2-c	10 GB	Centos 7
Elastic	elasticsearch-node1	elasticsearch	4	15	europe-west1-b	10 GB	Centos 7
Elastic +	kibana-node3	Elasticsearch + Kibana	4	15	europe-west1-b	10 GB	Centos 7
Kibana							

Cambiamos la password de usuario root (en las 3 instancias creadas)

\$ sudo passwd

Y cambiamos a usuario root

\$ su -

Modificamos el fichero hosts en las 3 MV con el comando

\$ vim /etc/hosts

Y añadimos en las 3 maquinas

10.154.0.5 elasticsearch-node1.europe-west2-c.c.proyectotfm-268719.internal

elasticsearch-node1

10.132.0.10 elasticsearch-node2.europe-west1 b.c.proyectotfm-268719.internal
elasticsearch-node2

10.132.0.12 kibana-node3.europe-west1-b.c.proyectotfm-268719.internal kibananode3



4.2.2 Instalación ElasticSearch

Seguimos las instrucciones de instalación da Elastic de la página https://www.elastic.co/guide/en/elasticsearch/reference/current/rpm.html

Descarga e instalación la clave de la clave pública

\$ rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch

Creacion de archivo elasticsearch.repo en el repositorio /etc/yum.repos.d/

\$ cd /etc/yum.repos.d/

\$ touch elasticsearch.repo

\$ vim elasticsearch.repo



Instalamos Elastic

\$ sudo yum install --enablerepo=elasticsearch elasticsearch

Para configurar que Elasticserche se inicie automáticamente cuando se inicie la MV ejecutamos los siguientes comandos

\$ sudo /bin/systemctl daemon-reload
\$ sudo /bin/systemctl enable elasticsearch.service

4.2.3 Configuración ElasticSearch

Vamos a modificar el fichero de configuración

\$ vim /etc/elasticsearch/elasticsearch.yml

Configuración elasticsearch-node1 /elasticsearch.yml

MÁSTER UNIVERSITARIO EN BUSINESS ANALYTICS Y BIG DATA

cluster.name: elastic-cluster	
node.name: elasticsearch-node1	
node.master: true	
network.host: elasticsearch-node1	
http.port: 9200	
discovery.seed_hosts: ["10.132.0.10", "10.132.0.12"]	
<pre>cluster.initial_master_nodes: ["elasticsearch-node1"]</pre>	

Configuración elasticsearch-node2 /elasticsearch.yml

cluster.name: elastic-cluster	
node.data: true	
<pre>xpack.security.audit.enabled: true</pre>	
node.name: elasticsearch-node2	
network.host: elasticsearch-node2	
http.port: 9200	
discovery.seed_hosts: ["10.154.0.5", "10.132.0.12"]	
<pre>cluster.initial_master_nodes: ["elasticsearch-node1"]</pre>	

kibana-node3 /elasticsearch.yml



Arrancamos ElasticSearch

\$ systemctl	start elasticsearch.service
\$ systemctl	status elasticsearch.service



<pre>[root@elasticsearch-node1 ~]# systemctl status elasticsearch.service</pre>
elasticsearch.service - Elasticsearch
Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; enabled; vendor preset: disab
led)
Active: active (running) since Fri 2020-04-10 17:45:47 UTC; 1h 29min ago
Docs: http://www.elastic.co
Main PID: 1385 (java)
CGroup: /system.slice/elasticsearch.service
-1385 /usr/share/elasticsearch/jdk/bin/java -Des.networkaddress.cache.ttl=60 -De
└─1826 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86 64/bin/contr
Apr 10 17:45:20 elasticsearch-node1 systemd[1]: Starting Elasticsearch
Apr 10 17:45:26 elasticsearch-node1 elasticsearch[1385]: OpenJDK 64-Bit Server VM warning:
Apr 10 17:45:47 elasticsearch-node1 systemd[1]: Started Elasticsearch.
Hint: Some lines were ellipsized, use -1 to show in full.

Img 11. Elasticsearch Running

4.2.4 Instalación Kibana

Instalamos Kibana en el tercer nodo kibana-node3, igualmente siguiendo las instrucciones de instalación https://www.elastic.co/guide/en/kibana/current/rpm.html

Descarga e instalación la clave de la clave pública

\$ rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch

Creación de archivo kibana.repo en el repositorio /etc/yum.repos.d/



Instalamos Kibana

\$ sudo yum install kibana

4.2.5 Configuración Kibana

Modificamos fechero de configuración de Kibana

server.port: 5601

MÁSTER UNIVERSITARIO EN BUSINESS ANALYTICS Y BIG DATA

server.host: "10.132.0.12" elasticsearch.hosts: ["http://10.154.0.5:9200"]

Para poder de conectar con un cliente externo tenemos que configurar las reglas de cortafuegos. Para ello desde la consola de Google Cloud Platform (GCP) desde menu \rightarrow Red VPC \rightarrow Reglas de cortafuegos, Creamos dos nuevas reglas.

Tcp5601 y tcp 9092

kibana	Entrada	http-server	Intervalos de IPs: 0.0.0.0/0		tcp:5601	Permitir	1000	default
			Img 12. Cort	a fuegos Ki	bana	.9		
elasticsearch	Entrada	http-server	Intervalos de IPs: 0.0.0.0/0		tcp:9200	Permitir	1000	default
			lmg 13. Corta fu	uegos Elast	icsearch	5		
homoo onf	tror o Kihana							
barnos en	irar a Nibana	1						
Clusters								٥
Enter setup mode P					• ~	Last 1 hour	Show dates	ී Refre
elastic-cluster								
Elasticsearch	• Health is yellow Basic lice	nse						
Elasticsearch	Health is yellow Basic lice	Nodes: 3		Indices: 12		Logs		
Elasticsearch	• Health is yellow Basic lice	Nodes: 3 Disk Available	67.96%	Indices: 12 Documents	215,812	Logs	aund	
Elasticsearch	Health is yellow Basic lice 7.6.1 11 hours	Nodes: 3 Disk Available JVM Heap	67.96% 20.4 GB / 30.0 GB 22.49%	Indices: 12 Documents Disk Usage	215,812 130.1 MB	No log data fo Set up Filebast, the to your monitoring	bund an configure your Elastic: cluster.	search output
Elasticsearch	Health is yellow Basic lice 7.6.1 11 hours	Nodes: 3 Disk Available JVM Heap	67.96% 20.4 GB / 30.0 GB 22.49% 668.4 MB / 2.9 GB	Indices: 12 Documents Disk Usage Primary Shards	215,812 130.1 MB 12	Do log data fo Set up Filebeat, the to your monitoring	ound en configure your Elastic: cluster.	search outpu
Elasticsearch Overview Version Uptime	• Health is yellow Basic lice 7.6.1 11 hours	Nodes: 3 Disk Available JVM Heap	67.96% 20.4 GB / 30.0 GB 22.49% 668.4 MB / 2.9 GB	Indices: 12 Documents Disk Usage Primary Shards Replica Shards	215,812 130.1 MB 12 6	Logs No log data fo Set up Filebeat, the to your monitoring	ound en configure your Elastic: cluster.	search output
Elasticsearch Overview Version Uptime Kibana • Healt	• Health is yellow Basic lice 7.6.1 11 hours th is green	Nodes: 3 Disk Available JVM Heap	67.96% 20.4 GB / 30.0 GB 22.49% 668.4 MB / 2.9 GB	Indices: 12 Documents Disk Usage Primary Shards Replica Shards	215,812 130.1 MB 12 6	Dogs No log data fo Set up Filebeat, the to your monitoring	ound en configure your Elastic: cluster.	search output
Elasticsearch Overview Version Uptime Kibana • Healt Overview	• Health is yellow Basic lice 7.6.1 11 hours th is green	nse Nodes: 3 Disk Available JVM Heap	67.96% 20.4 GB / 30.0 GB 22.49% 668.4 MB / 2.9 GB	Indices: 12 Documents Disk Usage Primary Shards Replica Shards	215,812 130.1 MB 12 6	Logs No log data fo Set up Filebeat, the to your monitoring	bund en configure your Elastic: cluster.	search output
Elasticsearch Overview Version Uptime Kibana • Healt Overview Requests	Health is yellow Basic lice 7.6.1 11 hours th is green 12	nse Nodes: 3 Disk Available JVM Heap Instances: 1 Connections	67.96% 20.4 GB / 30.0 GB 22.49% 668.4 MB / 2.9 GB	Indices: 12 Documents Disk Usage Primary Shards Replica Shards	215,812 130.1 MB 12 6	Logs No log data fo Set up Fiebeat, the to your manitoring	ound an configure your Elastics cluster.	search output

Probamos con un plugin de Google Chrome, añadimos el plugin de Elasticserch



Elasticsearch	http://34.76.35.96:920	0/	Connect elastic-	cluster cluster he	alth: yellow (21 of	28)					
Overview Indices Browser	Structured Query [+]	Any Request [+]	Index Filter								
	ilm- history- 1-00001 size: 25.4ki (25.4ki) docs: 18 (18) (info • Actions •	filebeat- 7.6.1- 2020.03.29- 000001 size: 2338 (5668) docs: 0(0) (info ~ Actions ~ filebeat-7.6.1 X	dgt size: 3.28Mi (6.49Mi) docs: 15.620 (31.240) Info Actions	.monitoring- kibana-7- 2020.04.10 size: 307ki (596ki) docs: 795 (1637) Info • Actions •	.monitoring- kibana-7- 2020.04.08 size: 1.23Mi (2.64Mi) docs: 4487 (8974) (info - Actions ~	.monitoring- kibana-7- 2020.04.06 size: 745ki (1.45Mi) docs: 2713 (5426) (info - Actions ~	.monitoring- kibana-7- 2020.04.05 size: 660ki (660ki) docs: 2312 (2312) (info ~) Actions ~)	.monitoring- cs-7- 2020.04.10 size: 11.8Mi (11.8Mi) docs: 19.558 (43.074) info ~ Actions ~	.monitoring- es-7- 2020.04.08 size: 58.5Mi (58.5Mi) docs: 96.167 (204.465) info • Actions •	.monitoring- es-7- 2020.04.06 size: 30.3Mi (30.3Mi) docs: 52.637 (96.433) fnto v Actions v	.monitorin es-7- 2020.04.0 size: 33.7Mi (33.7Mi) doss: 55.678 (112.222) inio Actions
Unassigned	ilm-history-1 X						0	0	0	0	0
		0	0	0	0	0					
elasticsearch- node2		0	0	0	0	0					
kibana-node3	0						0	0	0	0	0

Img 15. ElasticSearch pluginChrome

Hemos terminado con las configuraciones de los dos Clúster, comprobamos que los dos Cluasters están configurados correctamente, tenemos visibilidad desde fuera.

En el caso del Clúster Kafka para poder publicar y consumir el Topic con un cliente externo hemos instalado dos aplicaciones de escritorio Constructor y Kafka Tools.

En el caso de Elasticserch y Kibana hemos comprobado que nuestro Clúster tiene 3 nodos y el nodo 1 es el nodo maestro, tal y como hemos configurado.

La seguridad no ha sido un punto en el que se ha hecho especial hincapié, en cuanto a configuraciones establecidas en este proyecto, pero sí lo son para futuras implementaciones en entornos corporativos.

ANASTASIA LUKINA CHIBUSOVA CURSO 2018 - 2020

5. DESARROLLO Y EJECUCIÓN DE SCRIPTS

5.1 DATOS.

Tal y como se ha mencionado anteriormente vamos a descargar los datos de tráfico casi en tempo real, la información se actualiza casi en tiempo real, con una periodicidad de unos 5 minutos, que es el tiempo mínimo de varios ciclos de semáforo, necesarios para dar una medición real, y que no se vea afectada la medición por si el semáforo está abierto o cerrado.

La sensorización del tráfico se efectúa por medio de diversos equipamientos que permiten la realización del conteo de vehículos junto con la obtención del grado de ocupación de la vía. De esta manera se puede conocer el nivel de servicio, que se corresponde con los cuatro escalones típicos: tráfico fluido (nivel 0), tráfico lento (nivel 1), retenciones (nivel 2) y congestión (nivel 3).

Estos sistemas de detección son, en su mayoría, lazos electromagnéticos que se colocan debajo del pavimento y detectan la masa metálica de los vehículos que pasan sobre ellos, siendo sistemas de gran calidad y precisión. Tienen las limitaciones que se circunscriben a la toma de datos en un único punto, además de no disponer de visión de la zona para verificar o conocer la causa de los datos que suministran.

El objetivo principal de este conjunto de datos es proporcionar información del tráfico, en tiempo real, de la ciudad de Madrid.

En la ciudad de Madrid existen alrededor de 7.800 detectores de vehículos, de los que alrededor de unos 200 son sistemas ópticos de visión artificial con control desde el Centro de Gestión de la Movilidad, unos 1.400 son específicos de vías rápidas y acceso a la ciudad y, el resto, unos 6.200, son sistemas básicos de control de semáforos.

Estos 7.800 detectores conforman en la ciudad, aproximadamente, unos 4.100 puntos de medida, de los cuales 300 puntos están ubicados en M - 30 y se diferencian de los del resto de la ciudad porque disponen de un sistema para el control de la velocidad, caracterización de los vehículos y doble lazo de lectura.

Respecto a la información en tiempo real, ésta se obtiene por medio de un fichero en formato .xml que presenta, en tiempo real, los valores del tráfico medidos por los sistemas de control. El formato de presentación de la información es el estándar XML. El acceso a estos datos es libre desde el Portal de Datos Abiertos del Ayuntamiento de Madrid.

5.2 CAMPOS DEL CONJUNTO DE DATOS

Cabecera con la fecha y hora en que se obtuvieron los datos:

САМРО	DESCRIPCIÓN
fecha_hora	fecha y hora en que se obtuvieron los datos

Para cada uno de los puntos de medida de tráfico urbano, los campos que se proporcionan son los siguientes:

CAMPO	DESCRIPCIÓN
idolom	Identificador del punto de medida. Se
	acreanando con al compo "idolom" procento en
	corresponde con el campo idelem presente en
	el lichero geoffelerenciado y que permite su
	posicionamiento sobre plano e identificación del
	Vial y sentido de la circulación
descripcion	Denominación del punto de medida
accesoAsociado	Codigo de control relacionado con el control
	sematórico para la modificación de los tiempos
intensidad	Intensidad de número de vehículos por hora. Un
	valor negativo implica la ausencia de datos.
ocupación	Porcentaje de ocupación del punto de control por
	los vehículos. Un valor negativo implica la
	ausencia de datos.
carga	Parámetro de carga del vial en función de la
	intensidad, ocupación y características de la
	infraestructura. Un valor negativo implica la
	ausencia de datos.
nivelServicio	Nivel de servicio
intensidadSat	Intensidad de saturación de la vía en veh/hora y
	que se corresponde con el máximo número de
	vehículos que pueden pasar en el acceso a la
	intersección manteniéndose la fase verde del
	semáforo
error	Código de control de la validez de los datos del
	punto de medida



subarea	Identificador de la subarea de explotación de
	tráfico a la que pertenece el punto de medida
st_x	Coordenada X UTM del centroide que representa
	al punto de medida en el fichero georreferenciado
st_y	Coordenada Y UTM del centroide que representa
	al punto de medida en el fichero georreferenciado

Para poder almacenar los datos en **Elacticsearch** tenemos que transformar el XML a GeoJson y también transformar las coordenadas de tipo UTM del centroide a WKT Point para esto vamos a utilizar la **librería pyproj de Python.**

Well Known Text

La representación Well Known Text o de texto conocido (también llamado WKT en su acrónimo inglés) es una codificación o sintaxis en formato ASCII estandarizada diseñada para describir objetos espaciales expresados de forma vectorial. Los objetos que es capaz de describir el formato WKT son los siguientes:

- Puntos.
- Multipuntos.
- Líneas.
- Multilíneas.
- Polígonos.
- Multipolígonos.
- Colecciones de geometría.
- Puntos en 3 y 4 dimensiones.

Su especificación ha sido promovida por un organismo internacional Open Geospatial Consortium (OGC), siendo su sintaxis muy fácil de utilizar, de forma que su uso se encuentra muy generalizado en la industria geoinformática. De hecho, WKT es la base de otros formatos más conocidos como el KML utilizado en Google Maps y Google Earth.

La mayoría de las bases de datos espaciales, y en especial PostgreSQL a través de su extensión para el almacenamiento y análisis espacial PostGIS, utiliza esta codificación.

Existe una variante de este lenguaje, pero expresada de forma binaria, denominada Well Know Binary o binario conocido (WKB en su acrónimo inglés), la cual es también utilizada por estos gestores espaciales, pero con la ventaja de que al ser compilada en forma binaria la velocidad de proceso es muy elevada.

A efectos prácticos la sintaxis WKT consta de una descripción de los vértices que componen la geometría. Para que esta forma de especificar las geometrías tenga sentido deben de acompañarse de una indicación de la referencia espacial o proyección cartográfica utilizada en dicho vector.

Fuente https://es.wikipedia.org/wiki/Well_Known_Text

Para ahorrar el tiempo y coste del proyecte vamos a utilizar el Cluster de DataBriks. Para descargar los datos de la DGT vamos a ejecutar nuestro código cada minuto, en este caso lo hare con un Timer. Los ficheros descargados en el formato XML voy a guardar en **DBFS** en un sistema de archivos distribuido montado en un área de trabajo de Azure Databricks.



1 %fs ls dbfs:/DGT_MADRID/		
nath	2200	cizo
dbfs:/DGT_MADRID/05_04_2020_23_36_intensidad.xml	05_04_2020_23_36_intensidad.xml	1822822
dbfs:/DGT_MADRID/06_04_2020_00_o0_intensidad.xml	06_04_2020_00_00_intensidad.xml	1822765
dbfs:/DGT_MADRID/06_04_2020_00_01_intensidad.xml	06_04_2020_00_01_intensidad.xml	1822718
dbfs:/DGT_MADRID/06_04_2020_00_07_intensidad.xml	06_04_2020_00_07_intensidad.xml	1822747
dbfs:/DGT_MADRID/06_04_2020_00_14_intensidad.xml	06_04_2020_00_14_intensidad.xml	1822724
dbfs:/DGT_MADRID/06_04_2020_00_31_intensidad.xml	06_04_2020_00_31_intensidad.xml	1822674
dbfs:/DGT_MADRID/06_04_2020_00_35_intensidad.xml	06_04_2020_00_35_intensidad.xml	1822674
dbfs:/DGT MADRID/06 04 2020 00 36 intensidad.xml	06 04 2020 00 36 intensidad.xml	1822691
± - In. ■		

mand took 5.65 seconds -- by imf.anastasia@gmail.com at 8/4/2020 23:27:49 on cluster_tfm

Img 16. Ficheros XML guardados en DBFS

5.3 SCRIPT DESCARGA DE DATOS Y PUBLICACIÓN EN KAFKA

- 1. Descargamos los datos
- 2. Comprobamos la fecha y hora de obtención de los datos
- 3. Transformamos los campos de localización
- 4. Convertimos XML a GeoJson
- 5. Publicamos el GeoJson en Kafka

		= x - x
1	import requests	
2	import time	
3	from datetime import datetime	
4	import os	
5	from pyproj import Proj	
6	import pandas as pd	
7	import numpy as np	
8	from pyspark.sql.functions import UserDefinedFunction	
9	<pre>from pyspark.sql.functions import *</pre>	
10	from pyspark.sql.functions import udf	
11	<pre>from pyspark.sql.types import StructType, StructField, DoubleType, StringType</pre>	
12		
13	from kafka import KafkaProducer	
14	import json,time	
15	from kafka.errors import KafkaError	
Com	tack to 4 seconds by inf apartarial can at 8/4/1929 20:24:50 as cluster the	

Img 16. Script Import de las Librerías

1	#1.Dascargamos el fichero xml de la DGT
2	<pre>def download_file(url):</pre>
3	<pre>local_filename="/tmp/trafico-intensidad.xml"</pre>
4	# NOTE the stream=True parameter
5	r = requests.get(url, stream= True)
6	<pre>with open(local_filename, 'wb') as f:</pre>
7	<pre>for chunk in r.iter_content(chunk_size=1024):</pre>
8	<pre>if chunk: # filter out keep-alive new chunks</pre>
9	f.write(chunk)
10	f.flush()
11	return local_filename





<pre>1 def convert_toDouble(data): 2 udf = UserDefinedFunction(lambda x: x.replace(",","."), StringType()) 3 data_out = data.withColum("st_x", udf(col("st_x")).cast(DoubleType())) 4,vithColum("st_y", udf(col("st_y")).cast(DoubleType())) 5 return data_out</pre>	
<pre>F = UserUeTinedFunction(Lamoda x: X:replace(",",",",", Stringtype()) Let a cout = data.withColum("st_y", udf(col("st_y",).cast(DoubleType())) .withColum("st_y", udf(col("st_y")).cast(DoubleType())) Let a cout = data.out ok 0.02 seconds by inf.anastaia@gmsil.com at 8/4/2020 20:25:00 on cluster_tfm b) = Proj("+proj=utm +zone=30 +ellps=intl +towgs84=0.0000,0.00000,0.000000,0.0000000,0.000000</pre>	
Cnd 4	
<pre>1 myProj = Proj("*proj=utm *zone=30 *ellps=intl *towgs84=0.0000,0.00000,0.000000,0.0000000,0.000000</pre>	
Command took 0.02 seconds by imf.anastasia@gmail.com at 8/4/2020 20:25:00 on cluster_tfm	
Cnd 5	
<pre>1 schema = StructType([2 StructField("lon", DoubleType(), False), 3 StructField("lat", DoubleType(), False) 4])</pre>	
Command took 0.02 seconds by imf.anastasia@gmail.com at 8/4/2020 20:25:13 on cluster_tfm	
Cnd 6	
<pre>1 lon_lat_udf = udf(lambda row: convert_lon_lat(row[0], row[1]), schema)</pre>	

Img 18. Script Funciones para convertir las coordenadas de tipo UTM del centroide a WKT Point



Img 19. Script función para convertir XML a DataFrame



Img 20. Script funciones Kafka Producer

1 def df_to_geojson(df, properties,fecha, lat='lat', lon='lon'): 2 geo_obj = {} 3 for _, row in df.iterrows(): geo_obj['location'] = "POINT(" + str(row[lon]) + " " + str(row[lat]) + ")" 4 5 geo_obj['fecha']= datetime.strptime(fecha, "%d/%m/%Y %H:%M:%S").strftime("%Y-%m-%d %H:%M:%S") 6 for prop in properties: geo_obj[prop] = row[prop] #Enviar a Kafka 8 9 #print(geo_obj) 10 send_kafka(geo_obj)



MÁSTER UNIVERSITARIO **EN BUSINESS ANALYTICS Y BIG DATA**

1	def sleeper():
2	fecha_ant = None
з	while True:
4	try:
5	#1.Descargar el fichero xml desde la web DGT
6	url = "http://informo.munimadrid.es/informo/tmadrid/pm.xml"
7	res = download_file(url)
8	print(res)
9	#2.Subir los ficheros a dbfs
10	now = datetime.now()
11	dbfs_name= "dbfs:/DGT_MADRID/" + now.strftime("%d_%m_%Y_%H_%M") + "_intensidad.xml"
12	print(dbfs_name)
13	dbutils.fs.cp("file://" + res, dbfs_name)
14	#3.Leer fecha del fichero
15	df_fecha = spark.read\
16	.format("com.databricks.spark.xml")\
17	.option("rowTag", "pms")\
18	.load(dbfs_name)
19	<pre>if fecha_ant != df_fecha.select("fecha_hora").collect()[0]:</pre>
20	# actualizamos la var fecha ant
21	print('Fecha anterior: %s' % fecha_ant)
22	<pre>fecha_ant = df_fecha.select("fecha_hora").collect()[0]</pre>
23	print('Fecha actual: %s' % fecha_ant)
24	# Leemos el xml, convertire a json y mandar a kafka
25	df_data=read_data(dbfs_name)
26	df_data= df_data.replace(np.nan, -1)
27	df_data["color"] = [nivel_to_color(x) for x in df_data['nivelServicio']]
28	cols = ['accesoAsociado','carga','descripcion','error','idelem','intensidad','intensidadSat','nivelServicio','ocupacion','subarea','velocidad','color']
29	geojson = df_to_geojson(df_data, cols, fecha_ant.fecha_hora)
30	#else:
31	#Eliminas el fichero del dbfs
32	dbutils.fs.rm(dbts_name)
33	print('Se ha elinado: %s' % dbfs_name)
34	os.remove(res)
35	print('se na elinado: %s' % res)
36	except Exception as ex:
37	print(ex)
38	continue
29	the sum time already arready
40	a kun dur cline.steep() command,
41	a and show the before and after time
42	print(before, as a cime.ctime())
43	clime.s.tetp(log)
44	princ(writer, ws/m w cime.come())

Img 22. Script función principal Timer.

Script completo formato html



Para siguiente script voy a tener que crear una nueva cuenta en **DataBricks**, para poder ejecutar los dos scripts al mismo tiempo y a de más hay que abrir en deferentes navegadores para que no se haga el autologin.

Este script creamos con pysparkStreaming

Agregamos las librerías necesarias

databricks	Create L Library Source Upload	ibrary ce DBFS/S3	PyPI	Maven	CRAN									
Home	Coordinates						Search Packages							
Ъ	Maven Coordinates (com.databricks:spark-csv_2.10:1.0.0) Se Coordinate is a required field					Search	Maven Central 😫 🕻	Q elasticsearch						
Workspace	Repository O					Group Id		Artifact Id		Releases		Options		
0	Optional						software.amazon.awssdk		elasticsearch		2.11.12		Select	
Recents	Exclusions						software.amazon.awscdk		elasticsearch		1.32.0		Select	
	Dependenc	cies to exclude	e (log4j:log	4j,junit:junit	t)		com.weicoder		elasticsearch		3.3.4	\sim	Select	
Data							org.elasticsearch.distribution	n.integ-test-zip	elasticsearch		7.6.2	\sim	Select	
Clusters	Create	Cancel					org.elasticsearch		elasticsearch		7.6.2		Select	





- elasticsearch
- elasticsearch-spark-20_2.10-7.6.2
- kafka-spark-consumer-2.1.0
- spark-kafka_2.10-0.6.0
- spark-streaming-kafka_2.11-1.6.3
- spark-streaming-kafka-0-10_2.11-2.4.5
- spark-streaming-kafka-0-10-assembly_2.11-2.4.5
- spark-streaming-kafka-0-8_2.11-2.4.5
- spark-streaming-kafka-0-8-assembly_2.11-2.4.5

Img 23. Librerías necesarias para Elastic y Kafka Comsumer

5.4 SCRIPT CONSUMER KAFKA E INSERT EN ELASTICSEARCH

- 1. Suscribirse al Topic
- 2. Insertar en Elastic

Script Kafka Consumer

Е

1 import sys 2 from pyspark import SparkContext 3 from pyspark.streaming import StreamingContext 4 from pyspark.streaming.kafka import KafkaUtils 5 from elasticsearch import Elasticsearch	▶•
6 import json	
Command took 0.25 seconds by anastasia.lukina.es@gmail.com at 8/4/2020 22:49:54 on cluster_tfm (clone)	
Img 24. Script Import Librerias	
<pre>1 #Las IpsExternas para conectar a Elastic 2 es = Elasticsearch(["34.77.57.255:9200", "35.241.242.33:9200", "35.242.150.103:9200"])</pre>	
Command took 0.07 seconds by anastasia.lukina.es@gmail.com at 8/4/2020 22:49:56 on cluster_tfm (clone)	
Cmd 3	
<pre>1 #Inser en Elastic 2 def send_elastic(data): 3 es.index(index="dgt", doc_type="_doc", body=data) 4</pre>	

Img 25. Script función Insert en Elastic



1	ifname == "main":
2	try:
3	<pre>ssc = StreamingContext(sc, 2)</pre>
4	#Conector a Kafka
5	<pre>kvs = KafkaUtils.createDirectStream(ssc, ["dgt"], {"metadata.broker.list": '35.195.134.228:9092'})</pre>
6	lines = kvs.map(lambda x: x[1])
7	#Mapeamos a Json
8	<pre>parsed = kvs.map(lambda x: json.loads(x[1]))</pre>
9	<pre>def function(x):</pre>
10	y = x.collect()
11	for d in y:
12	data= json.dumps(d)
13	print(data)
14	send_elastic(data)
15	#Recorremos RDD
16	<pre>parsed.foreachRDD(lambda x :function(x))</pre>
17	ssc.start()
18	<pre>ssc.awaitTermination()</pre>
19	except Exception as ex:
20	print(ex)
21	pass



Script completo en el formato html



6. RESULTADOS

Ya están todos los pasos completados, ahora podemos proceder a realizar las pruebas del flujo completo.

1. Arrancamos los Clústeres en Google Cloud Platform

	ProyectoTFM			٩		
м						
	Instancias de VM	CREAR INSTANCIA	🛃 IMPORTAR VM	C ACTUALIZAR		IER 🖑 RESTABLE
	Ξ Filtrar las instancias de VM					Columnas •
	Nombre ^	Zona Recomend	ación Usada por	IP interna	IP externa	Conectar
	elasticsearch-node1	europe-west2-c		10.154.0.5 (nic0)	35.242.150.103 🗳	SSH 🖌 🚦
	elasticsearch-node2	europe-west1-b		10.132.0.10 (nic0)	104.155.121.61 🖾	SSH 🖌 🚦
	🛛 🕑 kafka-node2	europe-west1-b		10.132.0.14 (nic0)	35.205.42.76 🖾	SSH 🗸
	🕑 kafka-node3	europe-west1-b		10.132.0.15 (nic0)	34.76.27.239 🖸	SSH 🗸
	🔮 kibana-node3	europe-west1-b		10.132.0.12 (nic0)	34.77.188.80 🖾	SSH 🗸 🚦
	Zookeeper-kafka-node1	europe-west1-b		10.132.0.13 (nic0)	34.77.115.245 🖾	SSH 🗸





2. Creación de Index con sus tipos en Elasticsearch



Img 28. Creación de Index en Elastic

3. Ejecutamos los dos scripts al mismo tiempo para comprobar que justo al publicar recibimos los mensajes en tiempo real.

8	KafkaConsumer (Python)		0	?		GetData_DGT (Python) © ?
databricks	🚠 • cluster_tfm 🛛 🗸 🖼 🖛 🚔 🔳 🍠 🖛	C.	•	0 3	databricks	📥 🖷 cluster_tfm 🛛 🗤 📾 🖛 🔺 🔳 🍠 🖛 🖼 🛤
databricks Home Workspace Records Data Clusters	• cluster_thm • · · · · · · · · · · · · ·	88", "ac 3", "ac 38", "a 38", "a 38", "a 38", "a	© ccesoA ccesoA ccesoA ccesoA ccesoA	Asocli Asocli Asocli Asocli Ssocli	databricks databricks Home Come Recents Data Clusters	▲ ckustritm Image: Second state Concil 12 Image: Second state I steeper() Image: Second state Cancel Running command • (5) Spark Jobs View (1 stages) • Job 45 View (1 stages) • Job 46 View (1 stages) • Job 47 View (1 stages) • Job 48 View (1 stages) • Job 49 View (1 stages) • Job 48 View (1 stages) • Job 49 View (1 stages) • Sange 49: 1/1 (0 running) ● Send Kafka 0k Send Kafka 0k Send Kafka 0k Send Kafka 0k Send Kafka 0k Send Kafka 0k Send Kafka 0k
Jobs Q Search	<pre>['location: POIII('-3.68213155139644,40.3852144394439), "Fecha: "60/44/200121:9: ['location: POIII(''-3.6831318564,40.39512397469649)," Fecha: "60/44/200121:9: ['location: POIII(''-3.66185145647,4254,40.38512474514574,47547,17547</pre>	108", ". 108", ". 108", "ac. 108", "ac.	acceso acceso cesoAs acceso ccesoA cesoAs ccesoA cesoAs ccesoAs ccesoAs ccesoAs ccesoAs ccesoAs ccesoAs	DASOC: DASOC: DASOC: Sociad Sociac Sociac Asoci Asoci Sociac Soci	Jobs Q Search	Send Kafka 0k Send Kafka 0k

Img 29. Ejecución de los scripts

Como podemos observar en la Img 28, enviamos y recibimos los mensajes de Kafka en Tiempo Real.



4. Comprobar la inserción de los datos en Elastic.

clasticsearch	mp.//54.11.51.255.9200			ionnect elastic-cluster cluster health: yellow (18 of 24)								
Overview Indices Browser	Structured Query [+]	Any Reques	st [+]									
Browser										Re	efresh	
II Indices		Searched	1 of 1 share	ds. 10000 hits. 0.329 seconds								
ADICES		_index	_type	_id	_score	location	fecha 🖌	accesoAsociado	carga	descripcion	erro	
apm-agent-configuration		dgt	_doc	enkLW3EBnNTNauchPDig		POINT(-3.7446364660980054 40.39751317003505)	2020-03-22 01:10:02	7 241102	4	(TACTICO)CUART POBLET Ø44 O-E CUART POBLET-ALHAMBRA)	Ν ^	
kibana 1		dgt	_doc	Pm0LW3EBwnqyRnIuPR5Y		POINT(-3.7385098743847904 40.393009042968615)	2020-03-22 01:10:03	7 241603	4	(TACTICO) LOPEZ MEZQUÍA, 5 E-O (CASTELLFLORITE-V.CARPETANA)	N	
kibana task manager 1		dgt	_doc	Gm8LW3EBgIU6_gIRPVH8		POINT(-3.7373070487790754 40.394693623075504)	2020-03-22 01:10:0	7 241703	10	(TACTICO) MOCHUELO,1 S-N(PEDRO MTNEZ-V.CARPETANA)	N	
monitoring-es-7-2020.04.05		dgt	_doc	HG8LW3EBgIU6_gIRVFHB		POINT(-3.742173463107262 40.39014101213036)	2020-03-22 01:10:03	7 242101	13	(TACTICO) LAGUNA, 85 S-N (SABANERO-V.CARPETANA)	N	
monitoring-es-7-2020.04.06		dgt	_doc	fnkLW3EBnNTNauchVThk		POINT(-3.7476889015835235 40.403054474901325)	2020-03-22 01:10:02	7 242901	3	(TACTICO)GIRO CEBREROS FRENTE IGLESIA E-S (HIGUERAS-SEPULVEI	DA) N	
monitoring-es-7-2020.04.08		dgt	_doc	gHkLW3EBnNTNauchYzgq		POINT(-3.6978980379408335 40.38165687979307)	2020-03-22 01:10:02	7 252501	6	AV. CORDOBA Ø32 N-S(METRO ALMENDRALES-TOMELLOSO)	S	
monitoring-kibana-7-2020.04.05		dgt	_doc	VW0LW3EBwnqyRnIuYx7Y		POINT(-3.697809724443837 40.38077352648286)	2020-03-22 01:10:02	7 252504	0	(TACTICO)TOMELLOSO Ø3 O-E(SOCUELLAMOS-AV. CORDOBA)	N	
monitoring-kibana-7-2020.04.06		dgt	_doc	H28LW3EBgIU6_gIRalHP		POINT(-3.693205462694703 40.35776856440021)	2020-03-22 01:10:0	7 260203	2	AV. ANDALUCIA Ø21 S-N(FELICIDAD-UNANIMIDAD)	N	
monitoring-kibana-7-2020.04.08		dgt	_doc	gXkLW3EBnNTNauchazhy		POINT(-3.6933034485303136 40.35703598657618)	2020-03-22 01:10:03	7 260201	2	AV. ANDALUCIA Ø24 N-S(TERTULIA-BOHEMIOS)	N	
lgt		dgt	_doc	G28LW3EBgIU6_gIRS1GX		POINT(-3.7529599599303096 40.403593114798284)	2020-03-22 01:10:03	70	0	(TACTICO) VILLAMANIN Ø45 S-N GIRO IZDA (A-5-VILLASANDINO)	N	
ilebeat-7.6.1-2020.03.29-000001		dgt	_doc	fHkLW3EBnNTNauchTDg5		POINT(-3.7533014409216388 40.40368522361857)	2020-03-22 01:10:02	70	0	(TACTICO) VILLAGARCIA Ø61 O-E (VILLAGARCIA-VILLAMMANIN)	N	
m-history-1-000001		dgt	_doc	VG0LW3EBwngyRnIuWx72		POINT(-3.6985286687302157 40.38441067283433)	2020-03-22 01:10:0	7 481506	2	AV. CORDOBA Ø9 S-N(SEBASTIAN GOMEZ-GTA. CADIZ)	N	
PES		dgt	_doc	QG0LW3EBwnqyRnIuVB4c		POINT(-3.7531121487195023 40.404162780312866)	2020-03-22 01:10:03	7 241902	0	(TACTICO) VILLASANDINO Ø41 E-O (PARODIA-VILLAMMANIN)	N	
doc.		dgt	_doc	P20LW3EBwngyRnIuRR4A		POINT(-3.753172881404354 40.403975596065166)	2020-03-22 01:10:03	7 0	3	(TACTICO) VILLAMANIN Ø42 S-N (A-5-VILLASANDINO)	N	
IELDS		dgt	_doc	Hm8LW3EBgIU6_gIRXFGb		POINT(-3.6970024854493233 40.37952642734947)	2020-03-22 01:10:03	7 252503	6	AV. CORDOBA Ø47 S-N(SOCUELLAMOS-TOMELLOSO)	N	
©timestamp	,	dgt	_doc	e3kLW3EBnNTNauchRDhU		POINT(-3.7534208400571125 40.404230758375355)	2020-03-22 01:10:03	70	0	(TACTICO)VILLAMANIN Ø40 N-S (VILLAMANIN-VILLASANDINO)	N	
accesoAsociado		dgt	_doc	bW0LW3EBwngyRnIumR6p		POINT(-3.6946468938677626 40.36745320037686)	2020-03-22 01:10:03	7 391202	2	AV.ROSALES Ø1001 E-O(AV. ANDALUCIA-EDUARDO BARREIROS)	S	
action.actionTypeId		dgt	_doc	JW8LW3EBgIU6_gIRmlFO		POINT(-3.6946159833584247 40.35936758076343)	2020-03-22 01:10:02	7 261504	5	AV. VERBENA PALOMA Ø13 O-E (PAN Y TOROS-AV. ANDALUCIA)	N	
					Im	a 20 Browsor Dlugi	h Chrom					

 Elasticsearch
 http://35.246.118.45.9200/
 Connect
 elastic-cluster
 cluster health: yellow (15 of 20)

 Overview
 Indices
 Browser
 Structured Query (=)
 Any Request (=)

Browser											Refresh
All Indices .	Sea	arched 10	of 10 sha	rds. 209 hits. 1.058 seconds							
INDICES	_in	dex .	_type	_id	_score	▲ location	fecha	accesoAsociado	carga	descripcion	en
.apm-agent-configuration	tfm	1 1	dgt	UwART3EBzWFISL97sZ_0	1	POINT(-3.743585662623674,40.399339634597034)	06/04/2020 12:30:08	240602	2	CJAL. FCO. J. JIMENEZ Ø76 E-O (HURTUMPASCUAL-ALHAMBRA)	÷
.kibana_1	tfm	1 1	dgt	de4RT3EBcTP6NZzasoy1	1	POINT(-3.7445964495846003,40.39839509771498)	06/04/2020 12:30:08	241101	2	ALHAMBRA Ø23 N-S (CJAL. FCO. J. JIMENEZ -CUART POBLET)	
.kibana_task_manager_1	tfm	1 1	dgt	p7ERT3EBTJD-I-fOs_Rb	1	POINT(-3.7445128237560072,40.39845967578998)	06/04/2020 12:30:08	240603	3	ALHAMBRA Ø23 S-N (CUART POBLET-CJAL. FCO. J. JIMENEZ)	
.monitoring-es-7-2020.04.05	tfm		dgt	vLERT3EBTJD-I-f04PSk		Politic-strategications/2007 Result Source		-241602	8	V.CARPETANA Ø123 E-O (SALVADOR ATARAZONA)	
.monitoring-es-7-2020.04.06	tfm		dgt	WQART3EBzWFISL974Z9X		POINT(-3.7429208456168204,40.38967312644194)		242104	7	VIA CARPETANA, 167 O-E (LEON V ARMENIA-LAGUNA)	
.monitoring-kibana-7-2020.04.05	tfm		dgt	VwART3EBzWFISL9703_M		POINT(-3.739841120078756,40.392283611262705)	05/04/2020 12:30:08	241502	7	(AFOROS) V.CARPETANA Ø138 E-O (ATECA-N.S.VALVANERA)	
.monitoring-kibana-7-2020.04.06	tfm		dgt	ee4RT3EBcTP6NZza0Yx-	ind	ex p(1111(-3.739730539491257,40.39231893844186)	06/04/2020 12:30:08	241604	4	(AFOROS) V.CARPETANA Ø138 O-E (N.S.VALVANERA-ALBENTOSA)	
filebeat-7.6.1-2020.03.29-000001	tfm		dgt	u7ERT3EBTJD-I-f02fTi	- Cid	"UWARTBEBzWEISL97625070335,40.39116124602727)	06/04/2020 12:30:08	241503	6	N.S.VALVANERA, 114 S-N (A.PAVAS-V.CARPETANA)	
ilm-history-1-000001	tfm		dgt	WAART3EBzWFISL972p-X	_ve	sion c hrt(-3.741570071900254,40.39096073268709)	06/04/2020 12:30:08	242102	19	V.CARPETANA, 143 E-O (N.S.VALVANERA-LAGUNA)	
tfm	tfm		dgt	eO4RT3EBcTP6NZzayYwo	1 7	source (-3.7429050751751864,40.39313578121623)	05/04/2020 12:30:08	241303	5	N.S.VALVANERA Ø56 S-N (LAGUNA-GTA.CARMENES)	
Types	tfm		dgt	qbERT3EBTJD-I-fOyfTR	1	location 1 POINT(-3.743585662623674,40.39933963459	7034) /2020 12:30:08	241501	8	N.S.VALVANERA Ø43 N-S (LAGUNA-V.CARPETANA)	
_doc	tfm		dgt	UgART3EBzWFISL97q5-Z	1	techa": "06/04/2020 12:30:08" accesoAsociado": 9406093740705,40.399940418672784)	06/04/2020 12:30:08	240102	7	SEPULVEDA Ø118 N-S (CEBREROS-CJAL. FCO. J. JIMENEZ)	
dgt	tfm		dgt	c-4RT3EBcTP6NZzarIzb		carga: 27(-3.7536747191061552,40.39822324815227)		240101	3	CJAL. FCO. J. JIMENEZ Ø126 E-O (BERLANAS-SEPULVEDA)	
FIELDS	tfm		dgt	VAART3EBzWFISL97uZ9V	1	descripcion"; "CJAL, FCO, J. JIMENEZ Ø76 E-O. (HURTUMP/	ASCUAL-ALHAMBRA)"	241103	50	ALHAMBRA Ø31 S-N (CERRO MICA-ÁNGEL SANZ BRIZ)	
@timestamp	tfm		dgt	prERT3EBTJD-I-fOrfSM		idelem : 4741745584124333698,40.39901894165384)	06/04/2020 12:30:08	240504	7	CJAL. FCO. J. JIMENEZ Ø86 O-E (F. CALVO-ALHAMBRA)	
accesoAsociado	tfm		dgt	e-4RT3EBcTP6NZza24xI	1	intensidad : 48, 1445850881009, 40.391007696401836)	06/04/2020 12:30:08	241504	0	V.CARPETANA, 143 O-E (LAGUNA-N.S.VALVANERA)	
action.actionTypeId	tfm		dgt	d-4RT3EBcTP6NZzawYw		nivelServicio": 0,3853036126266,40.395178937759745)	06/04/2020 12:30:08	241301	3	ALHAMBRA, 94 N-S (CULLERA-GTA. LOS CARMENES)	
action.name	tfm		dgt	qLERT3EBTJD-I-fOwfTJ	1	ocupacion : 0,7448363117084,40.39354007231361)	05/04/2020 12:30:08	241304	0	DUQ. PARCENT Ø3 O-E (M. CASTILLO-GTA.LOS CARMENES)	
action.secrets	tfm	i i	dgt	VQART3EBzWFISL97wp98		velocidad : "None", 5975057813434,40.394187218456594)	05/04/2020 12:30:08	241302	0	GALLUR Ø46 E-O (ESCALONILLA-GTA.CARMENES)	
action task params.actionId	tfm		dgt	fe4RT3EBcTP6NZza6Iwj		color Cligreen 7447237359612515,40.400056284022796)	06/04/2020 12:30:08	240601	13	ALHAMBRA,28 N-S (A.CALVO-GREG.VACAS)	
action task params.apiKey	tfm		dgt	vbERT3EBTJD-I-fO6PTK		POINT(-3.7452543724001357,40.402576426294424)	06/04/2020 12:30:08	242503	0	HIGUERAS Ø41 S-N(PALMIPEDO-SEPULVEDA)	
activemo caller	tfm		dgt	WgART3EBzWFISL978J9	1	POINT(-3.7451877170820276,40.40405479412473)	06/04/2020 12:30:08	242501	3	HIGUERAS Ø25 N-S(ALMAZAN-CEBREROS)	
- over entities	tfm		dat	fu4RT3EBrTP6N7za8Vw	1	POINT(-3 746388701356062 40 402829289926)	05/04/2020 12:30:08	242802	10	SEPHIVEDA 119 E-O (HIGHERAS-A MORÁN)	

Img 31. Visualización de los datos con el Plugin Chrome

Info 🔹 🔒

5. Visualización de datos en Kibana

En la siguiente imagen podemos ver los datos en tiempo real según están insertando los datos se van pintando en la pantalla



Img 33. Vista2

MÁSTER UNIVERSITARIO EN BUSINESS ANALYTICS Y BIG DATA



📕 🖸 Dash	hboard / Editing New Dashboard	d (unsaved)										٥	
Save Cance	el Add Options Share												
 ⊘ (a) √ Sea (b) √ Sea (c) + Add 	arch filter						,	QL © ~	Apr 8, 2	020 @ 23:00:00.0	$00 \rightarrow \sim in 2 hours$	ି Refre	<u>sh</u>
© Mar 20, 2 © Mar 20, 2 © 9 ©	e .484 Total	3 Fechas 10 2.5 2 5 1.5 0 1 0 0.5 0	23:15	23-30 23	-45 00:00 fer	00:15 cha per minute			00:45	01:00	8		
		Madri											

Img 34. Vista3

7. CONCLUSIONES

1. Problemas encontrados:

- 1.1 A la hora de montar los **Clústeres** he encontrado varios problemas:
- 1.2 Poca información de como montar y configurar un Clúster real (con las plataformas AWS, Azure y GCP existen los clústeres preconfigurados lo único que hay que seguir los pasos donde nos quedamos sin saber cómo se configura por debajo)
- 1.3 En caso de Clúster de Kafka la configuración de Publicador y Consumidor tenía que poner los IPs externos, y cada vez que arrancaba de nuevo las MV las IPs externas cambiaban, por lo cual siempre tenía que cambiar los ficheros de configuración en los 3 nodos. La configuración del Clúster Kafka en mi caso me ha resultado más costoso del todo el trabajo.
- 1.4 Al terminar todo proceso por primera vez no podría crear las capas geográficas en **Kibana**, porque no encontraba documentos de tipo geoespaciales.
- 1.5 Al utiliza el clúster de **DataBrikcs** solo se podría ejecutar un script y para realizar la prueba completa de publicar y recibir los mensajes **de Kafka en tiempo real** y que el ejemplo se acerca más al caso real, he tenido que crear otra cuenta de prueba con otro correo electrónico y abrir en otro navegador.

2. Resultados:

Finalmente, después de miles de pruebas he conseguido realizar el trabajo tal y como se planteaba desde el principio.

8. ASIGNATURAS/MÓDULOS RELACIONADOS

Parra desarrollar el proyecto propuesto me he basado en los siguientes módulos/asignaturas:

1. Mod 6. Infraestructura Big Data

- 1.1 Procesamiento de datos con Spark
- 1.2 Arquitecturas En Streaming
- 1.3 Componentes De Arquitectura En Streaming
- 1.4 Plataformas Y APIS En La Nube
- 2. Mod 7. Almacenamiento e Integración de Datos 1.1 Adquisición De Datos
- 3. Mod 8. Valor y Contexto de la Analítica Big Data 1.1 Proyectos De Big Data
- 4. Mod 9. Aplicaciones Analíticas
 1.1 Caso De Estudio De Técnicas De Recuperación De Información

9. MÉTODOS, MATERIALES Y TECNOLOGÍAS

1. Google Cloud Platform (GCP) – Networking

- 1.1. https://youtu.be/3ioh96OWoll
- 1.2. <u>https://youtu.be/BPBUAn9EGh0</u>
- 1.3. <u>https://cloud.google.com/compute/docs/instances/connecting-to-instance</u>
- 2. Apache Kafka
 - 2.1. <u>https://kafka.apache.org/</u>
 - 2.2. https://docs.bitnami.com/oci/infrastructure/kafka/administration/create-cluster/
 - 2.3. Kafka => Sparkstreaming => Elastic <u>https://hadoopist.wordpress.com/2016/06/17/kafka-sparkstreaming-elastic/</u>
 - 2.4. <u>https://www.datasciencecentral.com/profiles/blogs/setting-up-your-first-kafka-development-environment-in-google</u>
 - 2.5. <u>https://www.digitalocean.com/community/tutorials/how-to-install-apache-kafka-on-centos-7</u>
- 3. Elasticsearch + kibana
 - 3.1. https://www.elastic.co/guide/en/elasticsearch/reference/current/rpm.html
 - 3.2. https://www.elastic.co/guide/en/kibana/current/rpm.html
 - 3.3. https://www.elastic.co/guide/en/elasticsearch/reference/current/cluster-health.html
- 4. Datos del tráfico en tiempo real
 - 4.1. https://datos.madrid.es/
- 5. Obtención y tratamiento de datos
 - 5.1. geo-point elastic <u>https://www.elastic.co/guide/en/elasticsearch/reference/7.6/geo-point.html#geo-point</u>
 - 5.2. Códigos EPSG (European Petroleum Survey Group) <u>https://www.mapa.gob.es/es/cartografia-y-sig/ide/directorio_datos_servicios/caracteristicas_wms.aspx</u>
- 6. Otros recursos Cursos Udemy
 - 6.1. Hadoop Big Data desde cero <u>https://www.udemy.com/course/monta-un-cluster-hadoop-big-data-desde-cero/learn/lecture/4858040#overview</u>

- 6.2. Procesando el Big Data con Apache Spark (en español) <u>https://www.udemy.com/course/programacion-</u> <u>con-apache-spark/learn/</u>
- 6.3. Domina Apache Spark 2.0 con Scala, curso intensivo <u>https://www.udemy.com/course/apache-spark-y-scala-curso-intensivo-apache-spark/learn/</u>
- 6.4. Apache Kafka Series Learn Apache Kafka for Beginners v2 <u>https://indra.udemy.com/course/apache-kafka/learn/</u>
- 6.5. Apache Kafka Real-time Stream Processing (Master Class)
- 6.6. https://indra.udemy.com/course/kafka-streams-real-time-stream-processing-master-class/learn/
- 7. Herramienta para crear diagramas gratis 7.1 <u>https://app.diagrams.net/</u>