

Enunciado Práctica

Enunciado Práctica

Arquitecturas en la Nube

Objetivo

Que el alumno ponga en práctica los conocimientos adquiridos en la asignatura Arquitecturas en la nube del Master en Data Science de la URJC en el curso académico 2017 / 2018.

La práctica se divide en dos partes:

- **Parte A:** Dockerización de aplicaciones
- **Parte B:** Diseño de servicios de análisis de datos en AWS

Fecha y mecanismo de entrega

La práctica puede realizarse de forma individual o por parejas.

La entrega se podrá realizar hasta el **8 de Enero de 2017** por el campus virtual.

Se deberá entregar un fichero .zip con dos carpetas, una por cada parte de la práctica. El nombre del fichero .zip tiene que ser igual al identificador del alumno en la URJC (la parte antes de la @ del correo electrónico del alumno en la URJC). En caso de dos alumnos, se incluirán ambos nombres separados por "-".

Parte A: Dockerización de aplicaciones

El objetivo de esta parte consiste en que el alumno empaquete y distribuya una aplicación en contenedores docker. En vez de crear una nueva aplicación para esta asignatura, se partirá de la práctica realizada para la asignatura de "Sistemas Distribuidos de Procesamiento de Datos". Esta práctica consiste básicamente en la descarga de tweets y su análisis en base a diferentes criterios haciendo uso de Hadoop.

Una vez desarrollada esa aplicación para la otra asignatura, se realizará una pequeña ampliación y se empaquetará como contenedores docker. La ampliación consistirá en que los resultados del análisis se deberán guardar en una base de datos. El tipo de base de datos concreta será elegida por el alumno. Puede ser una base de datos relacional como MySQL o una base de datos NoSQL como MongoDB.

Una vez implementada la aplicación completa (con la escritura de los resultados en la base de datos) se deberá empaquetar en diferentes contenedores orquestados con un fichero docker-compose. En el docker compose se deberán definir al menos dos servicios:

Enunciado Práctica

- **tweetanalysis:** Que ejecutará la descarga y análisis de tweets.
- **db:** Que ejecutará la base de datos.

El fichero docker-compose deberá poder ejecutarse con el siguiente comando:

```
$ docker-compose run -e TWEETER_ACCOUNT=XXX -e DOWNLOAD_TIME=XXX  
tweetanalysis
```

Como se puede ver, la cuenta de twitter de la que obtener los tweets y el tiempo de descarga se podrán especificar como parámetros del comando. Estos parámetros podrán recuperarse dentro del contenedor como variables de entorno.

Más información sobre el comando run de docker-compose se puede encontrar aquí:

<https://docs.docker.com/compose/reference/run/>

Los contenedores necesarios deberán estar disponibles en DockerHub, de forma que sólo sea necesario usar el fichero docker-compose para poder ejecutar la aplicación.

En el zip deberán entregarse todos los ficheros necesarios para poder crear los contenedores y la documentación sobre cómo crear los mismos en un fichero README.txt en texto plano o README.md en formato Markdown.

Parte B: Diseño de servicios de análisis de datos en AWS

La conocida empresa de venta por Internet **Tajo** quiere modernizar su anticuada plataforma de venta online y migrarla a la nube con el proveedor Amazon Web Services. Además, una vez migrada la plataforma, quiere incorporar nuevas funcionalidades usando servicios de análisis de datos ofrecidos por AWS. No obstante, antes de comenzar con la migración y la implementación de las nuevas funcionalidades, quiere tener una descripción detallada de los servicios que se podrían usar así como un presupuesto de los mismos.

Actualmente la plataforma de venta online gestiona una gran cantidad de ventas al día. Está implementada con una base de datos relacional y está alojado en servidores propios de la organización. Los servicios de análisis de datos que se quieren añadir son los siguientes:

- **Informes:** Generar informes básicos de ventas dos veces al día y generar un informe completo una vez al mes:
 - Esos informes deberán generarse con un cluster Hadoop
 - Una vez generados, los informes deberán transmitirse a los servidores de la organización para que puedan ser accedidos desde su intranet. Además, se guardarán en los servicios de AWS como copia de seguridad.
- **Recomendación:** Incorporar un sistema de recomendación de ventas:
 - Se usará un sistema de aprendizaje automático.
 - Cada mes se enviará a los clientes que no hayan entrado en el último mes a la web una oferta de un producto que se estime que puedan comprar en base a su perfil y su historial de compras.

Enunciado Práctica

- Cada vez que un cliente entra en la web se le hace una recomendación en base a su perfil.

En base a esta descripción del sistema hay muchos aspectos que no se conocen con precisión. Por ejemplo, no tenemos información sobre el número de visitas que se realizan al día o el tamaño en Kilobytes del informe. En ese caso, a la hora de hacer el presupuesto se puede considerar que existen dos escenarios: por ejemplo, en el caso de las visitas, podemos considerar que hay 1000 o 5000 visitas al día. En el caso de los informes, podemos considerar que con 1000 usuarios, el informe ocupa 500Kb y con 1000 usuarios, 1Mb.

La descripción detallada del nuevo sistema y los dos presupuestos mensuales en función de dos estimaciones de carga diferentes se deberán presentar en forma de presentación PowerPoint, LibreOffice o similar.