

Laboratorio 4.- NFS y MQTT

Contenido:

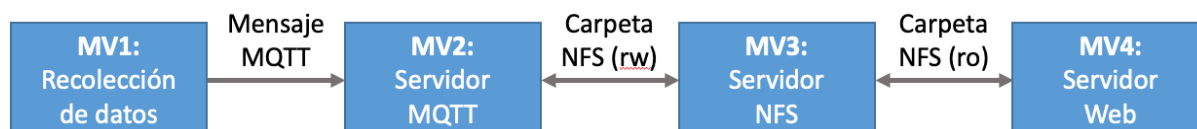
1	INTRODUCCIÓN	2
2	API PARA OBTENCIÓN DE DATOS	2
3	CONFIGURACIÓN DEL SISTEMA	4
3.1	MV1: RECOLECCIÓN DE DATOS	4
3.2	MV2: BROKER MQTT	4
3.3	MV3: SERVIDOR NFS	4
3.4	MV 4: SERVIDOR WEB.....	5
4	EXTENSIÓN DEL SISTEMA	5

Objetivos: Poner en práctica la configuración de servicios en red.

1 Introducción

El objetivo de este laboratorio es crear un sistema distribuido de recogida y monitorización de datos relacionados con los tipos de cambios entre divisas. Para ello, se propone desplegar varios servicios en diferentes máquinas virtuales, cada una con un objetivo concreto.

La arquitectura de la solución propuesta es la siguiente:



Cada uno de los bloques MV en el diagrama es una Máquina Virtual que debéis crear y que tiene un cometido concreto:

- **MV1:** Será la encargada de recopilar datos de cambios entre divisas y de enviarlos a MV2 a través de MQTT.
- **MV2:** Alojará un Broker Mosquitto para recibir datos por MQTT. Los datos recibidos serán escritos en un fichero. Este fichero residirá en un servidor externo, y se accederá a él mediante una carpeta montada por NFS.
- **MV3:** Será el servidor NFS de la arquitectura y guardará un fichero con los datos de conversión de divisa en una carpeta que será accesible por MV2 y MV4.
- **MV4:** Leerá los datos de conversión de divisa a través de la carpeta compartida por NFS y los usará para crear una interfaz Web simple. Esta MV contendrá un servidor Web que expondrá una Web públicamente accesible con el siguiente diseño:



La información necesaria para crear el sistema se encuentra en las siguientes secciones de este documento.

Este laboratorio está pensado para ser desarrollado por parejas:

- Un miembro de la pareja preparará MV1 y MV2.
- El otro miembro de la pareja preparará MV3 y MV4.

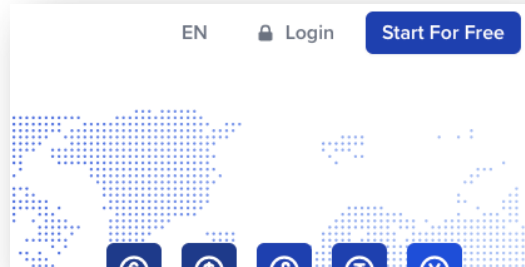
Cada miembro de la pareja debe crear las máquinas virtuales asignadas en su proyecto de Google Cloud Platform. En caso de realizar este laboratorio de forma individual o en grupos más grandes, podéis repartir el trabajo de forma diferente, a vuestro criterio.

2 API para obtención de datos

Los datos sobre tipos de cambio se obtendrán de CurrencyAPI (<https://currencyapi.com/>), una web dedicada a recopilar, almacenar y exponer datos financieros. Este portal provee datos históricos y en tiempo real sobre conversión entre divisas, además de una API para su consumo de forma programática. Los servicios de CurrencyAPI son comerciales, pero dispone de una capa gratuita (con límites) que podremos utilizar (<https://currencyapi.com/pricing/>).

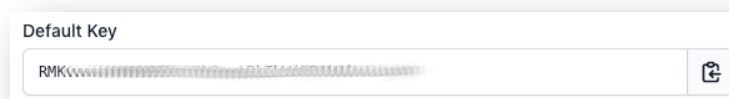
Administración de Sistemas - Curso 2023 / 2024

Para poder utilizar los servicios de CurrencyAPI es necesario darnos de alta y crear una nueva cuenta. Utilizar el botón "Start for free" de su portal para crear una nueva cuenta:



Como parte del proceso, será necesario proveer una dirección de e-mail, a la cual CurrencyAPI enviará un e-mail de confirmación (puede tardar unos minutos en llegar). Este e-mail contiene un enlace que es necesario utilizar para activar la cuenta.

Una vez que la cuenta esté activada, se puede acceder al Dashboard de control de CurrencyAPI. Lo más relevante para este laboratorio es obtener la clave de API, un String alfanumérico necesario para usar la API y obtener datos. Se encuentra en la sección "Default Key" del Dashboard:



Esta clave se puede utilizar para hacer consultas a los servicios de CurrencyAPI. Utilizando la herramienta de línea de comando *curl*, se puede consultar el valor de intercambio más reciente del Euro frente a la Libra Esterlina (es necesario reemplazar la etiqueta <CLAVE> con vuestra clave):

```
curl "https://api.currencyapi.com/v3/latest?base_currency="EUR"&currencies="GBP"&apikey=<CLAVE>" |  
python3 -m json.tool
```

Este comando devuelve el dato en el siguiente formato JSON:

```
{  
  "meta": {  
    "last_updated_at": "2023-09-28T23:59:59Z"  
  },  
  "data": {  
    "GBP": {  
      "code": "GBP",  
      "value": 0.8654591555  
    }  
  }  
}
```

Administración de Sistemas - Curso 2023 / 2024

Esta información será el punto de partida para la recolección de datos en la MV1.

Como nota final, la capa gratuita de CurrencyAPI permite un total de 300 consultas a sus servicios. Al superar ese límite, el portal pedirá introducir un método de pago para poder continuar utilizando su API.

3 Configuración del sistema

Esta sección describe la configuración y desarrollos a realizar en cada máquina virtual para crear el sistema completo. Si en algún punto no se proveen nombres, rutas o parámetros concretos para la tarea, significa que la elección queda a vuestro criterio.

En la primera versión del sistema, la información que se quiere monitorizar es la ratio de conversión del Euro frente a la Libra Esterlina.

3.1 MV1: Recolección de datos

Esta MV debe contener un script que:

- Recoja el último dato de conversión de EUR a GBP. Se recomienda utilizar el comando que consulta CurrencyAPI descrito en la sección anterior.
- Envíe este dato a un *topic* MQTT llamado "EUR/GBP", cuyo Broker reside en la MV2.

Este script se debe ejecutar 1 vez cada minuto.

3.2 MV2: Broker MQTT

Esta MV debe cumplir 3 funciones:

- Ejecutar un Broker MQTT con Mosquitto.
- Tener una montada una carpeta NFS en /tmp/carpetaRemota. Esta carpeta se exporta desde MV3 y allí está ubicada en /datosNfs. La carpeta está montada de forma que MV2 puede modificar sus ficheros.
- Escribir cada mensaje que se envíe al *topic* EUR/GBP en un fichero llamado HistoricoEurGbp.txt que se ubica en /tmp/carpetaRemota.

3.3 MV3: Servidor NFS

Esta MV contiene un servidor NFS y exporta su carpeta /datosNfs de la siguiente forma:

- A MV2, con permisos de lectura y escritura.
- A MV4, con permisos de sólo lectura.

Administración de Sistemas - Curso 2023 / 2024

3.4 MV 4: Servidor Web

La funcionalidad principal de esta MV es recoger el último dato de conversión de Euro a Libra Esterlina e incluirlo en una web con la forma mostrada en la 1ª sección del documento. Para ello, se propone utilizar la funcionalidad "http.server" de Python:

```
sudo python3 -m http.server 80
```

Este comando sirve el directorio de un sistema de ficheros por HTTP en el puerto especificado (en el ejemplo, el 80). Si se ejecuta desde el directorio /home/unai/miWeb, ese será el directorio servido. Si además existe un fichero index.html en el directorio, este será el que se muestre por defecto. Este fichero puede contener texto plano o código HTML (p.e., se puede usar
 para crear un salto de línea).

Basado en esta funcionalidad, esta MV debe cumplir estas funcionalidades:

- Tener la carpeta /datosNfs de MV3 montada en /tmp/carpetaRemota, para poder leer los datos de tipo de cambio de divisa escritos por MV2.
- Tener un fichero index.html en una carpeta, que sirva como web a mostrar.
- Tener el servidor web en marcha con la funcionalidad "http.server" de Python3.
- Tener un script que, cada minuto, recoja el último dato de conversión disponible en el fichero de datos de la carpeta NFS y actualice el fichero index.html.

Si todas las configuraciones son correctas, la web debe mostrarse en el navegador accediendo a <http://<IP-pública-de-la-máquina>>. El tipo de cambio entre Euro y Libra Esterlina debería actualizarse a medida que se reciban nuevos datos de CurrencyAPI (requiere refrescar la web de forma manual).

4 Extensión del sistema

Una vez que el sistema esté en marcha, se propone extenderlo para que monitorice la ratio de conversión del Euro a otras divisas, p.e. el dólar norteamericano. Esto implicará, entre otros:

- Un nuevo *topic* de MQTT, p.e. EUR/USD.
- Un nuevo fichero de datos.
- Modificación del fichero index.html para mostrar ambos datos actualizados.