FACULTAD DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN



Internet de las Cosas

ALUMNOS:

Quiñonez Lopez Efrain Saravia Velasquez Paul Diego Torres Mamani Fernando Miguel

> AREQUIPA, PERÚ 2021

1. Docker-compose

Para ejecutar los ejercicios del capítulo 7 del libro de "Build Your Own IoT Platform" era necesario levantar los servicios de:

- Node-red
- Mosquitto
- MySQL
- phpMyAdmin

En la ruta ./ raíz del repositorio podrá encontrar el archivo docker-compose.yml, este archivo contendrá los 4 servicios ya antes mencionados, podrá encontrar etiquetas por cada servicio como:

- Image: Nombre de la imagen a descargar desde dockerhub.
- Container name: Nombre para identificar al contenedor con el servicio
- Networks: Esta configuración permitirá tener a todos los contenedores que lo posean conexión y visibilidad entre ellos.
- Ports: Esta etiqueta mapea tanto el puerto del contenedor como el de nuestra máquina local, para poder acceder sin problemas.
- Configuración de base de datos: Aquí existen varias variables de configuración como las contraseñas y usuarios para la base de datos.

Como primer paso, luego de analizar nuestro yml procedemos a ejecutar con el siguiente comando este archivo.

docker-compose up -d

Captura de resultado:

```
efra@efra-X540UA:~/Documentos/unsa-codex/2021-B/iot/iot_continua$ docker-compose up -d Creating network "iot_continua_von" with the default driver
Recreating db ... done
Recreating mosquitto ... done
Recreating pma ... done
Recreating node-red ... done
```

Luego podremos observar los 4 servicios funcionando si todo va bien, se usa el siguiente comando:

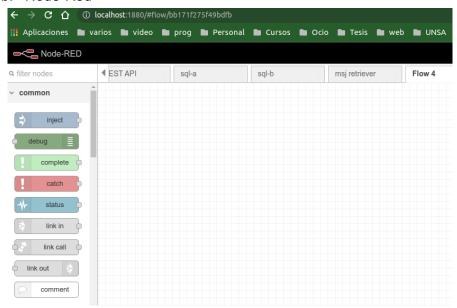
docker ps

efra@efra-X540	efra@efra-X540UA:~/Documentos/unsa-codex/2021-B/iot/iot_continua\$ docker ps													
CONTAINER ID	IMAGE NAMES	COMMAND	CREATED	STATUS	PORTS									
27a3ddf9bc49	phpmyadmin/phpmyadmin pma	"/docker-entrypoint"	About an hour ago	Up About an hour	0.0.0.0:80->80/tcp, :::80->80/tcp									
f3a5568dd527	mysql:5.7 db	"docker-entrypoint.s"	About an hour ago	Up About an hour	0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp									
95503f717b48	nodered/node-red:latest node-red	"npmno-update-not"	About an hour ago	Up About an hour (healthy)	0.0.0.0:1880->1880/tcp, :::1880->1880/tcp									
c0248e68c372 001->9001/tcp	eclipse-mosquitto:1.6 mosquitto	"/docker-entrypoint"	About an hour ago	Up About an hour	0.0.0.0:1883->1883/tcp, :::1883->1883/tcp, 0.0.0.0:9001->9001/tcp, :::9									

- 2. Verificación de los diferentes servicios
 - a. MySQL y phpMyAdmin

(i) localh	ost/index.ph								
■ varios	video 🖿	prog	Personal	Cursos	Ocio	■ Tesis	web	unsa	paper
				F		MyAd		in	
						uo u piip	my Adm		
				Idioma -	Language				
				Español	- Spanish		~		
				Iniciar se	esión 🕡				
				Servidor:					
				Usuario:	r	oot			
				Contraseña	a:				
							(Continuar	

b. Node-Red



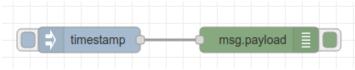
3. Desarrollo de actividades del capítulo 7

A continuación se presentan las actividades con su respectiva ruta en nuestro repositorio donde encontrará los archivos JSON donde contienen la mayoría de configuraciones para poder importarlo en el servicio de Node-Red para replicarlo con mayor facilidad.

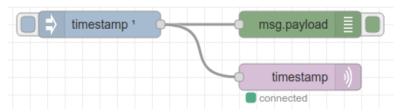
a. Creación de base de datos para próximos ejercicios Ruta: ./database/tSeriesDB.sql

b. Creando el primer flujo para nuestra plataforma

Ruta: ./json/0_intro.json



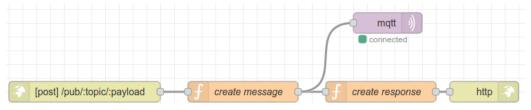
c. Adición de la capacidad de publicación de MQTT Ruta: ./json/1_TimeStamp.json



d. Publicador de mensajes de API REST

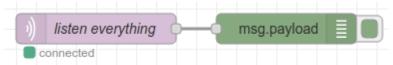
En este ejercicio se logró replicar la arquitectura del servicio en node-red, lamentablemente el servicio en la nube que se usaba para obtener respuestas o mensajes, dejó de funcionar por lo cual solo se llegó hasta este punto.

Ruta: ./json/2_rest_api.json



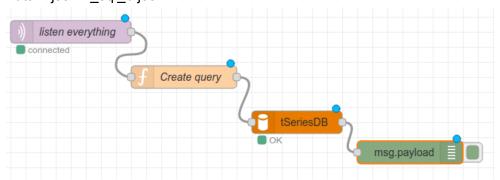
e. Configurar el nodo de entrada MQTT e implementar

Ruta: ./json/3_sql_a.json

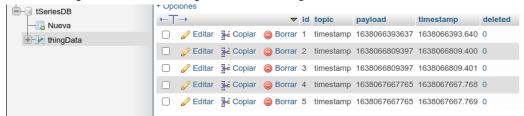


 f. Agregue y configure el nodo MySQL con nuestras credenciales de base de datos de series de tiempo

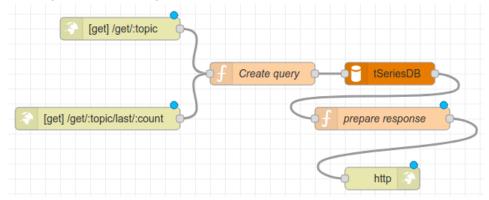
Ruta: ./json/4_sql_b.json



Luego de ejecutar este servicio, se detecto el almacenamiento en la base de datos configurada, donde se guardaron los siguientes datos:



g. Recuperador de mensajes de la API REST Lamentablemente este ejercicio usa una dirección donde el servicio que escucha ya no está en línea, por lo cual se logró replicar la estructura pero no se pudo realizar las peticiones. Ruta: ./json/5_retriever.json



h. Ejecución de Node-RED en segundo plano continuamente Para este ejercicio debíamos ingresar al contenedor para poder instalar una instancia de una libreria "forever", se logro listar el servicio de mosquito:

- 4. Link de Github
- https://github.com/PaulSV123/Internet-de-las-Cosas