

Informe de Laboratorio 07 - Contenedor de Microsoft SQL Server

Robles Flores, Anthony Richard (2016056192)

Universidad Privada de Tacna

Escuela Profesional de Ingeniería de Sistemas

Tacna, Perú

1. INFORMACIÓN GENERAL

Objetivos:

- Crear un contenedor de la imagen de Microsoft SQL Server
- Desplegar una BD usando un contenedor

Equipos y programas utilizados: Para el siguiente laboratorio requerimos de:

- Computadora con sistema operativo Windows 10
- Microsoft SQL Server Management Studio
- Docker Desktop

2. MARCO TEÓRICO

- Docker empaqueta software en “contenedores” que incluyen en ellos todo lo necesario para que dicho software se ejecute, incluidas librerías. Con Docker se puede implementar y ajustar la escala de aplicaciones de una forma rápida en cualquier entorno con la garantía de que el código se ejecutará.

A primera vista se piensa en Docker como una especie de máquina virtual liviana, pero la verdad no lo es. En Docker no existe un hypervisor

que virtualice hardware sobre el cual corra un sistema operativo completo. En Docker lo que se hace es usar las funcionalidades del Kernel para encapsular un sistema, de esta forma el proyecto que corre dentro de el no tendrá conocimiento que está en un contenedor. Los contenedores se encuentran aislados entre sí y se comportaran como máquinas independientes.

Iniciar un contenedor no tiene un gran impacto a diferencia de iniciar una máquina virtual ya que no tiene que iniciar un sistema operativo completo (desde cero). Gracias al uso de contenedores la demanda de recursos baja limitándose sólo al consumo de la aplicación que contenga. Un contenedor inicia en milisegundos.

- Anteriormente hablamos de manera de introducción sobre los contenedores pero ahora definamos en sí este concepto, Docker trabaja con algo que se llama contenedores de Linux estos son un conjunto de tecnologías que juntas forman un contenedor (de Docker), este conjunto de tecnologías se llaman: Namespace , Cgroups y Chroot.

3. PROCEDIMIENTO

Paso 1 : Iniciar Docker

- a) Iniciaremos con nuestra cuenta docker en Docker Desktop.



- b) Debemos estar registrados para ingresar.
- c) Ejecutamos PowerShell y agregamos el siguiente comando **docker version**.

```

PS C:\Windows\system32> docker version
Client: Docker Engine - Community
Version: 19.03.1
API version: 1.40
Go version: go1.12.5
Git commit: 74b1e89
Built: Thu Jul 25 21:17:08 2019
OS/Arch: windows/amd64
Experimental: false

Server: Docker Engine - Community
Engine:
Version: 19.03.1
API version: 1.40 (minimum version 1.12)
Go version: go1.12.5
Git commit: 74b1e89
Built: Thu Jul 25 21:17:52 2019
OS/Arch: linux/amd64
Experimental: false
containerd:
Version: v1.2.6
GitCommit: 894b81a4b802e4eb2a91dice216b8817763c29fb

```

Paso 2 : Crearemos un contenedor con Microsoft SQL Server para Linux

- d) Ejecutaremos el siguiente comando **docker search mssql** en PowerShell.

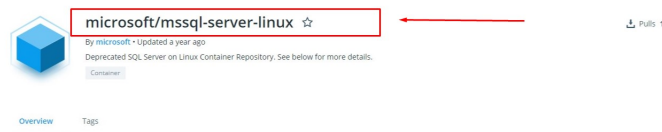
```

PS C:\Windows\system32> docker search mssql

```

NAME	DESCRIPTION	STARS
OFFICIAL		
microsoft/mssql-server-linux	Deprecated SQL Server on Linux Container Rep...	1157
microsoft/mssql-server-windows-developer	Official Microsoft SQL Server Developer Edit...	366
microsoft/mssql-server-windows-express	Official Microsoft SQL Server Express Editio...	330
microsoft/mssql-tools	Official images for Microsoft SQL Server Com...	51

- e) Ingresaremos nuestra cuenta en la página web Desktop Hub y buscaremos el repositorio **"microsoft/mssql-server-linux"**.



- f) Copiando el comando en la aplicación PowerShell (docker pull microsoft/mssql-server-linux) Se descargará la imagen del contenedor de Microsoft SQL Server en un servidor Linux como podemos ver en la imagen.
- g) Verificaremos la imagen descargada con el siguiente comando **docker images**.
- h) Iniciaremos el contenedor con el siguiente comando
- i) Nos devolverá el ID del contenedor.

```

PS C:\Windows\system32> docker pull microsoft/mssql-server-linux
Using default tag: latest
latest: Pulling from microsoft/mssql-server-linux
59ab41dd721a: Pull complete
57da90bec92c: Pull complete
06fe57530625: Pull complete
5a6315cba1ff: Pull complete
739f58768b3f: Pull complete
0b751601bca3: Pull complete
bcf04a22644a: Pull complete
6b5009e4f470: Pull complete
a9dca2f6722a: Pull complete
Digest: sha256:9b700672670bb3db4b212e8aef841ca79eb2fce7d5975a5ce35b7129a9b90ec0
Status: Downloaded newer image for microsoft/mssql-server-linux:latest
docker.io/microsoft/mssql-server-linux:latest

```

```

PS C:\Windows\system32> docker images

```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
microsoft/mssql-server-linux	latest	314918ddaedf	10 months ago	1.35GB
mcr.microsoft.com/mssql/server	latest	885d07287041	13 months ago	1.45GB

9d7552bed067814df9ba0541e7f216667d361f56add1832bb38ed83fb9fdf9db

- j) Verificamos que el contenedor se está ejecutando correctamente con el siguiente comando docker ps. El resultado será similar al siguiente:
- k) Abriremos el programa Microsoft SQL Server Management Studio 17 y conectamos con los siguientes datos:

Nombre del servidor : 127.0.0.1,16111

Inicio de sesión :sa

Contraseña:Epis2019

- l) Iniciaremos una consulta ingresando lo siguiente :

SELECT @@VERSION

- m) En la aplicación PowerShell ejecutamos el siguiente comando:

docker rm -f SQLLN01

- n) Verificamos la eliminación del contenedor con el siguiente comando:

docker ps

```

PS C:\Windows\system32> docker run -d -p 16111:1433 -e 'ACCEPT_EULA=Y' -e 'SA_PASSWORD=Tacna.2019' --name SQLLN01 microsoft/mssql-server-linux
9d7552bed067814df9ba0541e7f216667d361f56add1832bb38ed83fb9fdf9db

```

```

PS C:\Windows\system32> docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                               NAMES
9d7552bed067   microsoft/mssql-server-linux       "/opt/mssql/bin/sqls..." 2 minutes ago  Up 2 minutes  0.0.0.0:16111->1433/tcp             SQLNXX01
PS C:\Windows\system32>

```

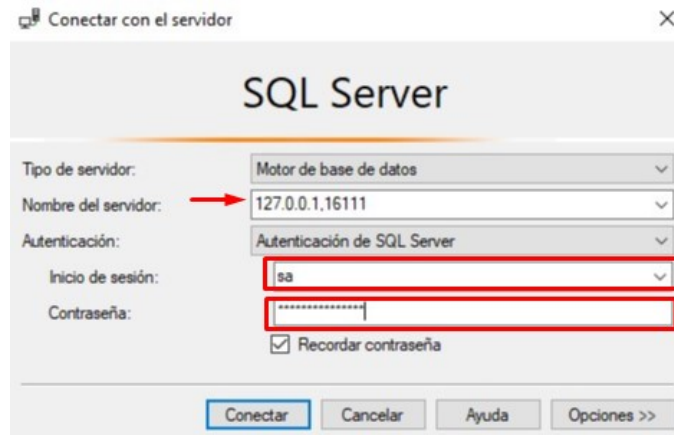
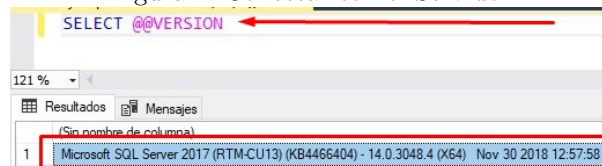


Figura 1: Conectar con el Servidor



4. ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

- a) Con el comando para iniciar con un contenedor podemos asignar los siguientes parámetros:

-p : Asignar el puerto.

'sapassword' : Contraseña del Inicio de Sesión SQL, usuario sa.

-name : Nombre del contenedor.

5. CUESTIONARIO

- a) ¿Con qué comando(s) exportaría la imagen de Docker de Microsoft SQL Server a otra PC o servidor?

Exportar la Imagen de Docker de Microsoft SQL Server "docker export (ID contenedor) ¿Nombreimagen.tar"

docker
9d7552bed067814df9ba0541e7f216667d361f56add1832bb38ed83fb9fdf9db
¿SQL.tar

```
docker export 9d7552bed067814df9ba0541e7f216667d361f56add1832bb38ed83fb9fdf9db > SQL.tar
```

- b) ¿Con qué comando(s) podría generar dos volúmenes para un contenedor para distribuir en un volumen el Archivo de Datos (.mdf) y en otro el Archivo Log (.ldf)?

```
CREATE DATABASE NAMEDATABASE ON  
( FILENAME = N'/var/opt/mssql/data2/NDATABASE.mdf' ),  
( FILENAME = N'/var/opt/mssql/data2/NDATABASElog.ldf' )  
FOR ATTACH  
GO
```

- c) Genere un nuevo contenedor y cree la base de datos con las siguientes características.

Nombre : FINANCIERA

Archivos:

- DATOS (mdf) : Tamaño Inicial : 50MB, Incremento: 10MB, Ilimitado
- INDICES (ndf) Tamaño Inicial : 100MB, Incremento: 20MB, Maximo: 1GB
- HISTORICO (ndf) Tamaño Inicial : 100MB, Incremento: 50MB, Ilimitado
- LOG (ldf) Tamaño Inicial : 10MB, Incremento: 10MB, Ilimitado

- d) ¿Cuál sería el script SQL que generaría esta base de datos?

6. CONCLUSIONES

Los contenedores de Docker nacen a partir de una imagen y en estos contenedores podemos solo ejecutar e instalar servicios, viene siendo como crear una maquina virtual a partir de una imagen (snapshot) pero muchísimo más ligera.

```

CREATE DATABASE FINANCIERA ON
PRIMARY
(
    NAME = 'FINANCIERA_DATOS',
    FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\FINANCIERA_DATOS.mdf',
    SIZE = 50MB,
    FILEGROWTH = 10MB,
    MAXSIZE= UNLIMITED
),
(
    NAME = 'FINANCIERA_INDICES',
    FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\FINANCIERA_INDICES.ndf',
    SIZE = 100MB,
    FILEGROWTH = 20MB,
    MAXSIZE= 1024MB
),
(
    NAME = 'FINANCIERA_HISTORICO',
    FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\FINANCIERA_HISTORICO.ndf',
    SIZE = 100MB,
    FILEGROWTH = 50MB,
    MAXSIZE= UNLIMITED
)
LOG ON (
    NAME = 'FINANCIERA_log',
    FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\FINANCIERA_LOG.ldf',
    SIZE = 10MB,
    FILEGROWTH = 10MB,
    MAXSIZE= UNLIMITED
)
GO

```

FINANCIERA_DATOS	05/11/2019 13:07	SQL Server Databa...	51.200 KB	←
FINANCIERA_HISTORICO	05/11/2019 13:07	SQL Server Databa...	102.400 KB	←
FINANCIERA_INDICES	05/11/2019 13:07	SQL Server Databa...	102.400 KB	←
FINANCIERA_LOG	05/11/2019 13:07	SQL Server Databa...	10.240 KB	←