

Informe de Laboratorio Nro 07

Universidad Privada de Tacna

Escuela Profesional de Ingeniería de Sistemas

Curso : Base de Datos II

Aponte Roldan, Sigfredo (2016054468)

Tacna, Perú

1. INFORMACIÓN GENERAL

1.1. *Objetivos*

- Despliegue de contenedores.

1.2. *Equipos, materiales, programas y recursos utilizados*

- Computadora con sistema operativo Windows XP, Vista, Windows 7, Windows 8 y/o Windows 8.1.
- Docker Desktop (Para lo cual se debe primero crear una cuenta en Docker Hub)

2. MARCO TEÓRICO

- Contenedor: Docker trabaja con algo que se llama “contenedores de Linux” estos son un conjunto de tecnologías que juntas forman un contenedor (de Docker). Las herramientas del contenedor, como Docker, ofrecen un modelo de implementación basado en imágenes. Esto permite compartir una aplicación, o un conjunto de servicios, con todas sus dependencias en varios entornos.
- Docker: Es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos.

3. PROCEDIMIENTO

Paso 1 : Iniciando Docker

- a) Hacemos doble clic en el acceso directo "Docker Desktop" para iniciar Docker. Luego presionamos clic derecho sobre el icono y seleccionamos la opción "Sig In".



- b) Ingresamos nuestras credenciales usuario y contraseña.
c) Ejecutamos la aplicación PowerShell, ejecutamos como Administrador. En la ventana de comandos digitamos lo siguiente **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

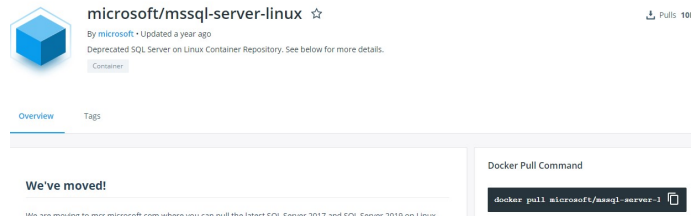
Server: Docker Engine - Community
 Engine:
  Version:          19.03.1
  API version:       1.40 (minimum version 1.12)
  Go version:        go1.12.5
  Git commit:        74b1e89
  Built:             Thu Jul 25 21:17:52 2019
  OS/Arch:           linux/amd64
  Experimental:      false
 containerd:
  Version:          v1.2.6
  GitCommit:        894b81a4b802e4eb2a91d1ce216b8817763c29fb
```

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

- d) Ejecutamos el siguiente comando **docker search mssql** en la ventana de PowerShell.
e) Ingresamos a nuestra cuenta en la página web Desktop Hub y buscamos el repositorio "**microsoft/mssql-server-linux**".

```
PS C:\Windows\system32> docker search mssql
```

NAME	OFFICIAL	AUTOMATED	DESCRIPTION	STARS
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



f) Copiamos el comando en la aplicación PowerShell.

```
docker pull microsoft/mssql-server-linux
```

El comando descargará la imagen del contenedor de Microsoft SQL Server en un servidor Linux y mostrará la siguiente resultado.

```
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
```

- g) Verificamos la imagen con el siguiente comando **docker images**.
- h) Luego digitamos el siguiente comando para iniciar un nuevo contenedor.
- i) La ejecución del comando anterior nos devolverá el ID del contenedor.
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:

```
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

```
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
```

Nombre del servidor : 127.0.0.1,16111

Inicio de sesión :sa

Contraseña: Tacna.2019

l) Iniciamos una nueva consulta, escribir y ejecutar la siguiente sentencia:

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

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.

```
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	SQLLN01

```
PS C:\Windows\system32>
```

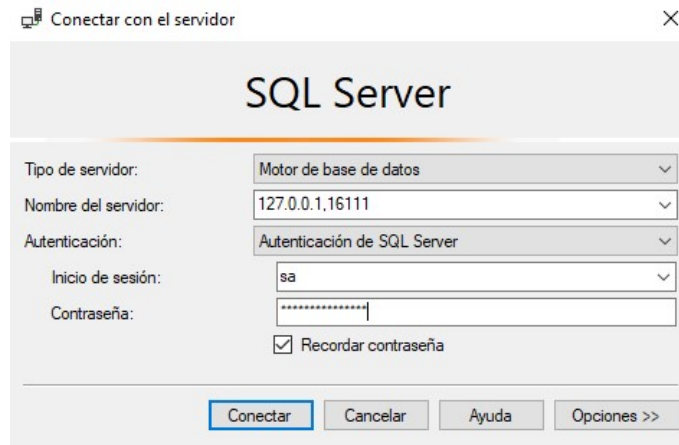
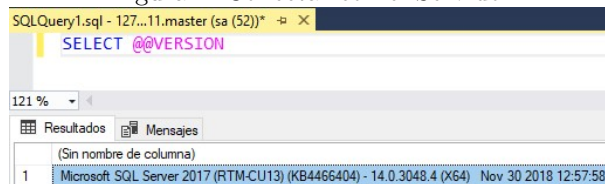


Figura 1: Conectar con el Servidor



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?

```
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

6. CONCLUSIONES

La perspectiva de Docker de empaquetado liviano e implementación de aplicaciones y dependencias es emocionante, y la comunidad la está adoptando rápidamente. y se está abriendo camino en entornos de producción. Por ejemplo, Red Hat anunció en diciembre soporte para Docker en el próximo Red Hat Enterprise Linux 7. Sin embargo, Docker sigue siendo un proyecto joven y está creciendo a una velocidad vertiginosa. Será emocionante ver cómo el proyecto se acerca a su versión 1.0, que se supone será la primera versión oficialmente sancionada para entornos de producción. Docker se basa en tecnologías establecidas, algunas de las cuales han existido durante más de un década, pero eso no lo hace menos revolucionario. Esperemos que este artículo proporcione suficiente información e inspiración para descargar Docker y experimentar con la herramienta.

7. WEBGRAFIA

[https://es.wikipedia.org/wiki/Docker\(software\)](https://es.wikipedia.org/wiki/Docker(software))
<https://docs.docker.com/get-started/>
<https://www.redhat.com/es/topics/containers/what-is-docker>