

UNIVERSIDAD PRIVADA DE TACNA



INGENIERIA DE SISTEMAS

TITULO:

**Informe de Laboratorio 7**  
**Instalación de una Instancia de Microsoft SQL Server**

CURSO:

Base De Datos II

DOCENTE:

Ing. Patrick Cuadros Quiroga

Integrantes:

Huillca Umpiri, Willian Arturo

(2015053793)

# Índice

<b>1. Información General</b>	<b>1</b>
1.1. Objetivos: . . . . .	1
1.2. Recursos Utilizados: . . . . .	1
1.3. Conocimientos: . . . . .	1
<b>2. Procedimientos.</b>	<b>2</b>
2.1. Descarga del Software. . . . .	2
2.2. Instalación del Docker Desktop . . . . .	4
2.3. Creación de Carpetas a Usar . . . . .	6
2.4. Configuración por PowerShell . . . . .	7
2.4.1. Parte 1. Aplicación en Ejecución . . . . .	7
2.4.2. Parte 2. Buscar PoweShell . . . . .	7
2.4.3. Parte 3. Verificar la Version de Docker Desktop . . . . .	8
2.4.4. Parte 4. Creación de un Contenedor con Microsoft SQL Server para Linux. . . . .	9
2.5. Consulta en SQL Server . . . . .	11
2.6. De nuevo en PowerShell - Eliminar contenedor creado . . . . .	13
2.7. Adicionando Persistencia. . . . .	14
2.7.1. Parte 1. Primeramente en PowerShell . . . . .	14
2.7.2. Parte 2. Seguidamente en Microsoft SQL Server Management . . . . .	16
2.7.3. Parte 3. Nuevamente en PowerShell . . . . .	18
2.7.4. Parte 4. Creando un contenedor con Microsoft SQL Server para Windows. . . . .	18
<b>3. Actividades Encargadas.</b>	<b>22</b>
3.1. ¿Con qué comando(s) exportaría la imagen de Docker de Microsoft SQL Server a otra PC o Servidor? . . . . .	22
3.2. ¿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)? . . . . .	22
3.3. Genere un nuevo contenedor y cree la Base de Datos con las siguientes características. . . . .	23

# **1. Información General**

## **1.1. Objetivos:**

- Descarga e Instalación del Docker Desktop
- Poder Configurar Correctamente el Docker Desktop en PowerShell
- Poder Instalar Correctamente las Consultas Requeridas

## **1.2. Recursos Utilizados:**

- Virtualizacion activada en el BIOS.
- Al menos 4 GB de RAM.
- Windows 10 64-bit: Pro, Enterprise o Education
- Docker Desktop
- Microsoft SQL Server Management Studio (Última Versión)
- Necesariamente Internet para la Descarga

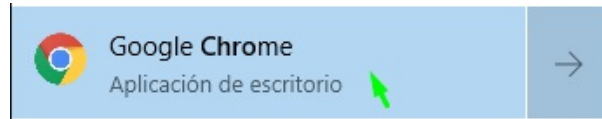
## **1.3. Conocimientos:**

- Para el desarrollo de esta practica se requerira de los siguientes conocimientos basicos:
- Conocimientos Básicos de Administracion de Base de Datos Microsoft SQL Server.
- Conocimientos Basicos de SQL.

## 2. Procedimientos.

### 2.1. Descarga del Software.

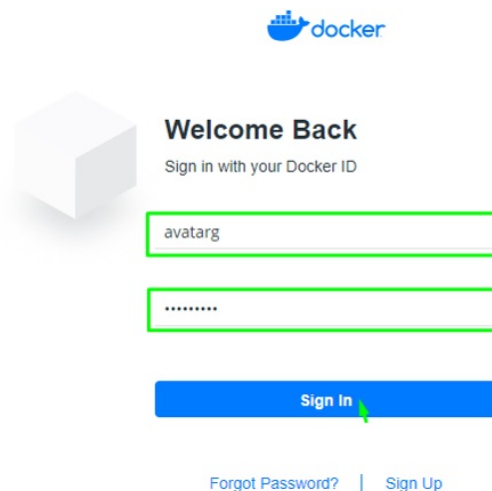
- Ingresamos a Nuestro Buscador de Internet Google Chrome o cualquier otro.



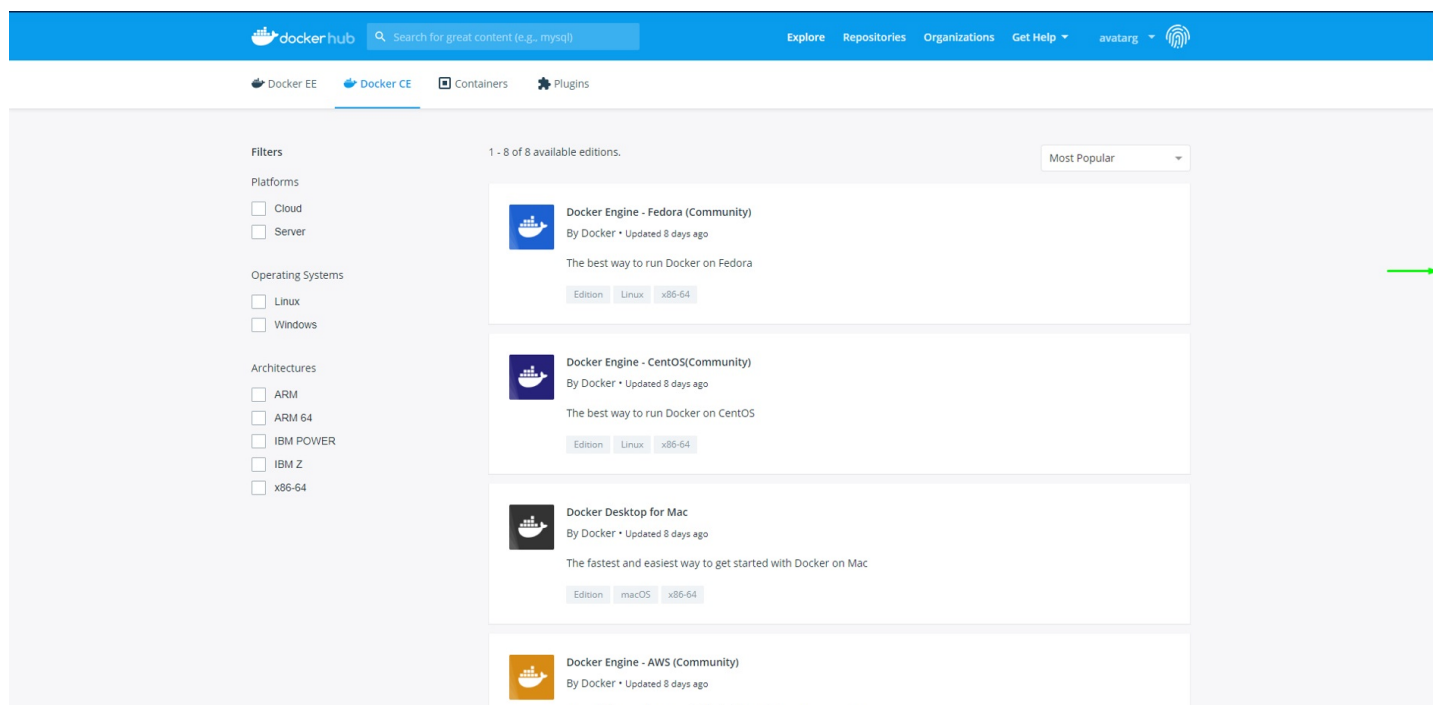
- Luego Copiamos y Pegamos el Siguiete Link.

`https://hub.docker.com/search/?type=edition&offering=community`

- Luego aparecerá esta pestaña, por lo que debemos primeramente tener una cuenta en Docker.



- Luego aparecerán varias aplicaciones para descargar del Docker, nos movemos hacia abajo con el mouse.



- Una vez hecho esto, nos llevará a una ventana para obtener el ejecutable del Docker.

## Get Docker Desktop for Windows

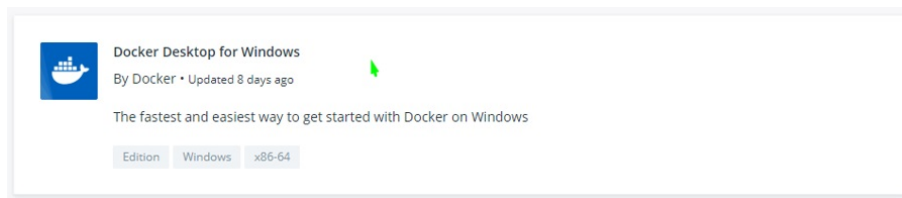
Docker Desktop for Windows is available for free.

Requires Microsoft Windows 10 Professional or Enterprise 64-bit. For previous versions get [Docker Toolbox](#).

By downloading this, you agree to the terms of the [Docker Software End User License Agreement](#) and the [Docker Data Processing Agreement \(DPA\)](#).



- Seleccionamos esta pestaña.

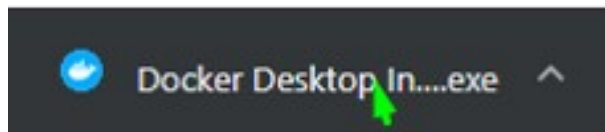


- Y empezará la descarga del ejecutable de Docker Desktop. Solo esperar a que termine la descarga, esto puede tomar varios minutos dependiendo de la velocidad del internet.

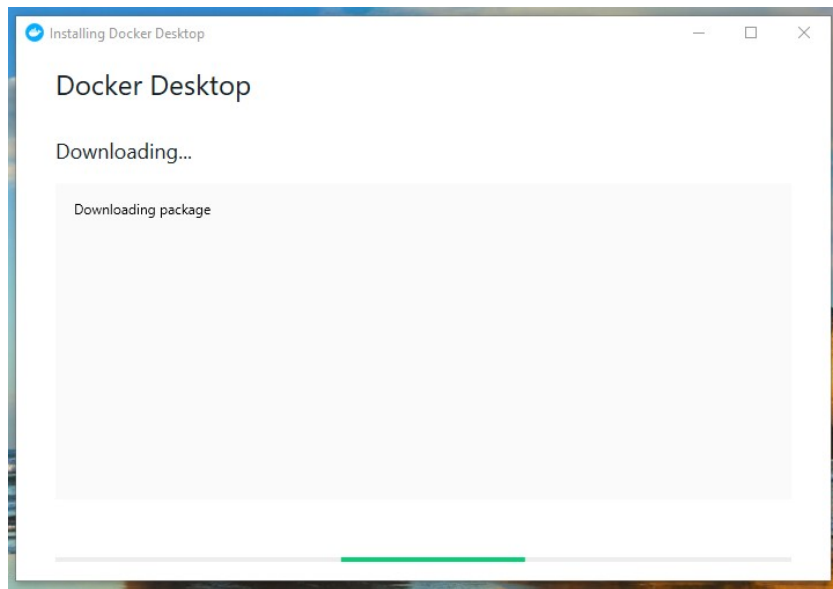


## 2.2. Instalación del Docker Desktop

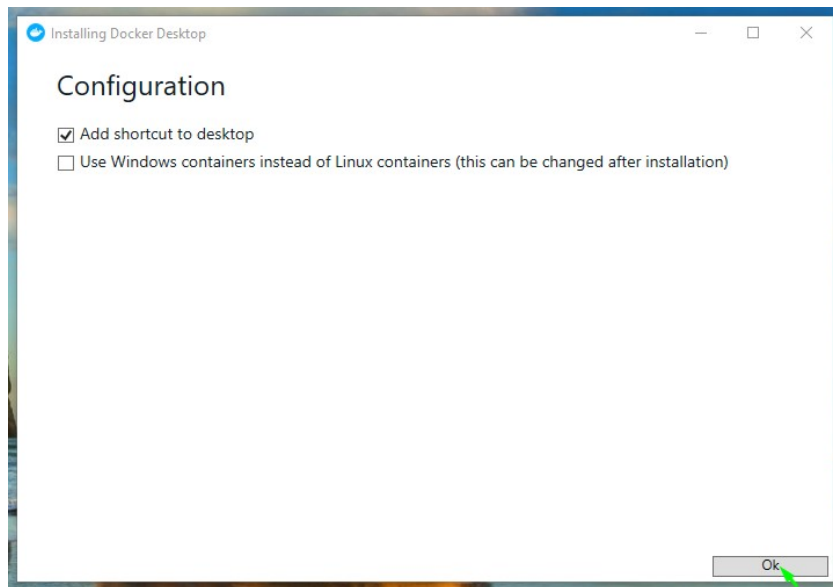
- Una vez ya terminada la descarga del ejecutable, la seleccionamos para empezar la instalación.



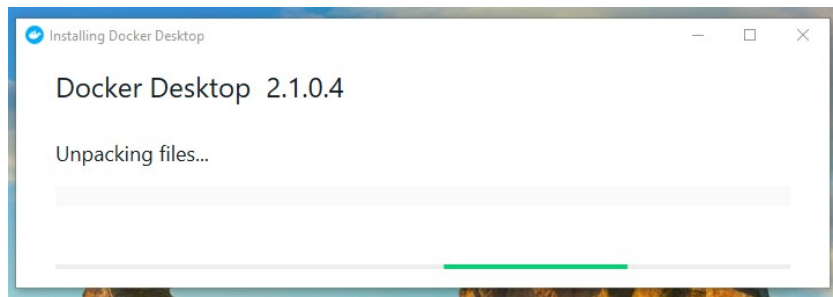
- Y aparecerá esta ventana indicando la inicialización de la instalación. Esperamos unos segundos hasta que aparezca otra ventana.



- Luego nos pedirá confirmación de si Aceptamos Añadir un Acceso Directo al Escritorio de nuestra Maquina.



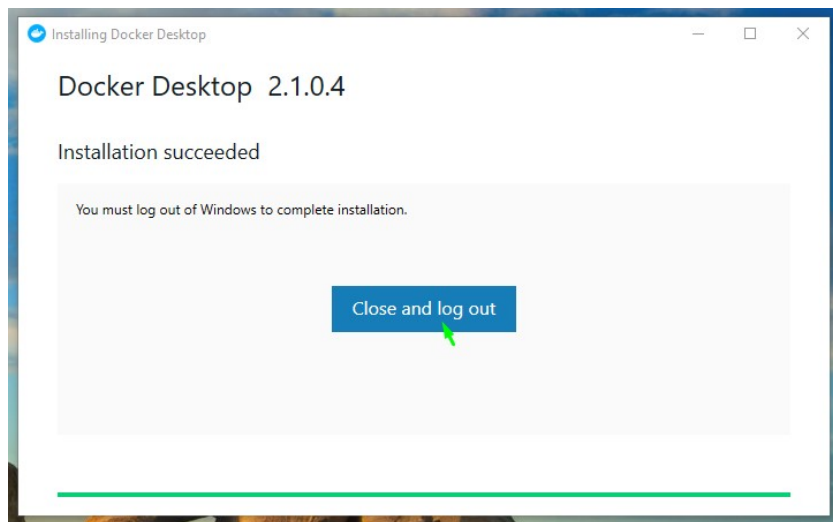
- Comenzará la instalación de cada paquete. Solo esperar hasta que termine.



- Durante la instalación de Docker, ya aparecerá el icono del Software en el escritorio, tal como se muestra en la siguiente imagen.



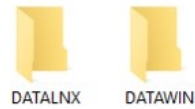
- Una vez terminada la instalación de Docker, seleccionamos el botón como aparece en la imagen y se reiniciará el equipo para actualizar y guardar las configuraciones hechas.



### 2.3. Creación de Carpetas a Usar

- Después de reiniciar la máquina, creamos dos carpetas en la unidad que desee, en mi caso lo crearé en el disco D.





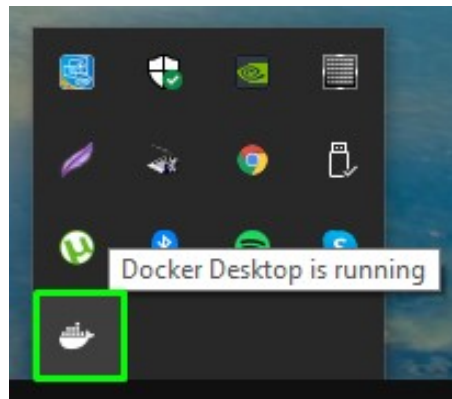
## 2.4. Configuración por PowerShell

### 2.4.1. Parte 1. Aplicación en Ejecución

- Luego aparecerá por defecto una notificación del Sistema Operativo.



- Y se agregará un icono en la barra de tareas en la parte inferior derecho.

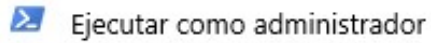


### 2.4.2. Parte 2. Buscar PoweShell

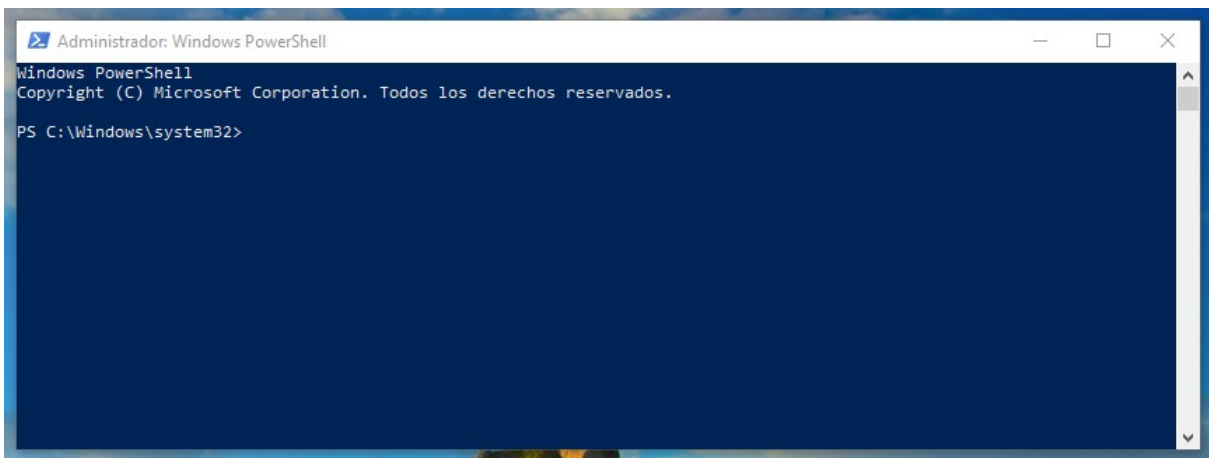
- Ahora buscamos el programa Windows PowerShell.



- Luego lo ejecutamos como administrador para que no genere problemas luego.



- Se mostrará una ventana como la que se ve en la imagen.



### 2.4.3. Parte 3. Verificar la Version de Docker Desktop

- Luego, digitamos lo siguiente para Verificar la Versión del Docker que hemos instalado.

**docker version**

- El resultado sería como se muestra en la siguiente imagen.

```
Administrador: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

PS C:\Windows\system32> docker version
Client: Docker Engine - Community
Version: 19.03.4
API version: 1.40
Go version: go1.12.10
Git commit: 9013bf5
Built: Thu Oct 17 23:44:48 2019
OS/Arch: windows/amd64
Experimental: false

Server: Docker Engine - Community
Engine:
Version: 19.03.4
API version: 1.40 (minimum version 1.12)
Go version: go1.12.10
Git commit: 9013bf5
Built: Thu Oct 17 23:50:38 2019
OS/Arch: linux/amd64
Experimental: false
containerd:
Version: v1.2.10
GitCommit: b34a5c8af56e510852c35414db4c1f4fa6172339
runc:
Version: 1.0.0-rc8+dev
GitCommit: 3e425f80a8c931f88e6d94a8c831b9d5aa481657
docker-init:
Version: 0.18.0
GitCommit: fec3683
PS C:\Windows\system32>
```

#### 2.4.4. Parte 4. Creación de un Contenedor con Microsoft SQL Server para Linux.

- Digitaremos el siguiente comando.

`docker search mssql`

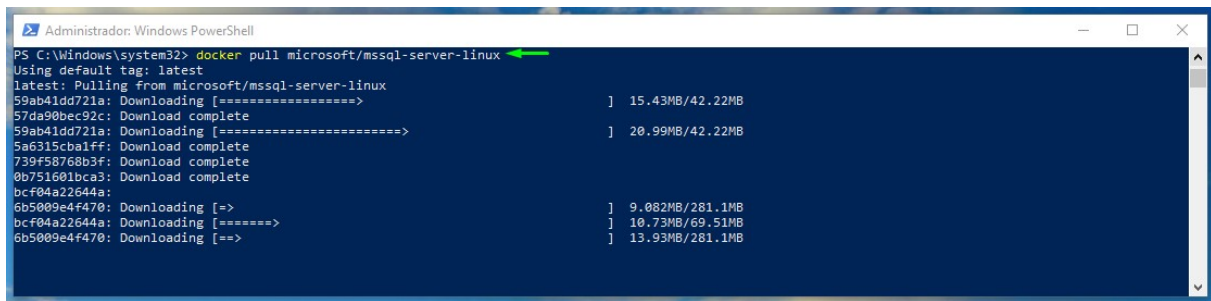
- El resultado sería similar a como se ve en la imagen.

```
Administrador: Windows PowerShell
PS C:\Windows\system32> docker search mssql
NAME                                DESCRIPTION                                STARS    OFFICIAL    AUTOMATED
microsoft/mssql-server-linux        Deprecated SQL Server on Linux Container Re... 1157
microsoft/mssql-server-windows-developer Official Microsoft SQL Server Developer Edit... 365
microsoft/mssql-server-windows-express Official Microsoft SQL Server Express Editio... 330
microsoft/mssql-tools                Official images for Microsoft SQL Server Com... 51
rsmoorthy/mssql                     MSSQL Database (version SQL2000)              12
datagrip/mssql-server-linux          SQL Server and SQL Server tools on Linux(201... 9
gantrior/mssql-server-2014-express-windows-with-iis mssql 2014 + IIS                             4
microsoft/mssql-monitoring-influxdb Sample Image for Influxdb, This image is des... 3
mcmoe/mssqldocker                   Builds on microsoft/mssql-server-linux and a... 3
tsgkadot/mssql-tools                SQL Server tools on Linux (sqlcmd)             3
ansibleplaybookbundle/mssql-apb     MS SQL Server on linux (APB)                   2
jboesl/mssql-server-linux           mssql-server-linux with mssql-tools installe... 2
bitwarden/mssql                     The Bitwarden database.                        2
mlitellovinx/mssql-server-linux      microsoft/mssql-server-linux with mssql-tool... 2
ansibleplaybookbundle/mssql-remote-apb An APB that deploys Microsoft SQL Server      1
microsoft/mssql-monitoring-collectd This Sample image is designed to work with t... 1
metaskills/mssql-server-linux-rails Microsoft SQL Server for Linux - Rails Adapt... 1
dhwanilraval/mssql-docker-rhel-official SQL Server in Docker                           1
liaisonintl/mssql-server-linux       mssql-server-linux                             0
softwareplant/mssql                 SQL Server test database                       0
uwthalesians/mssql                  Mssql Database Server for Perceptia           0
plever/mssql-server-linux-plus       Adding additional tooling to the base sql se... 0
christianacco/mssql-server-windows-express Fork of official microsoft/mssql-server-wind... 0
tchughesiv/mssql-server-linux        CentOS build                                    0
astronomerio/mssql-source            MSSQL source.                                  0
```

- Procedemos a digitar el siguiente comando.

`docker pull microsoft/mssql-server-linux`

- Y empezará la descarga de imagen de contenedor de Microsoft SQL Server en un servidor Linux.

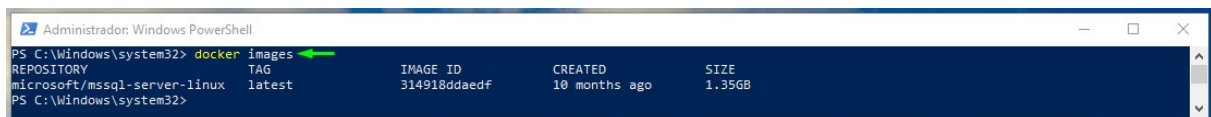


```
Administrador: Windows PowerShell
PS C:\Windows\system32> docker pull microsoft/mssql-server-linux
Using default tag: latest
latest: Pulling from microsoft/mssql-server-linux
59ab41dd721a: Downloading [=====] 15.43MB/42.22MB
57da90bec92c: Download complete
59ab41dd721a: Downloading [=====] 20.99MB/42.22MB
5a6315c8a1ff: Download complete
739f58768b3f: Download complete
0b751601bca3: Download complete
bcf04a22644a:
6b5009e4f470: Downloading [==>] 9.082MB/281.1MB
bcf04a22644a: Downloading [=====] 10.73MB/69.51MB
6b5009e4f470: Downloading [==>] 13.93MB/281.1MB
```

- Luego digitamos el siguiente comando.

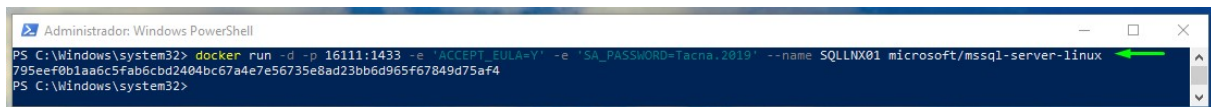
**docker images**

- Lo cual se verificará la imagen.



```
Administrador: Windows PowerShell
PS C:\Windows\system32> docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
microsoft/mssql-server-linux  latest         314918ddaedf    10 months ago  1.35GB
PS C:\Windows\system32>
```

- Seguidamente ejecutamos el comando, como se muestra en la imagen y como respuesta se visualizará un ID que corresponde al contenedor, como se muestra en la imagen.



```
Administrador: Windows PowerShell
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
795eef0b1aa6c5fab6cbd2404bc67a4e7e56735e8ad23bb6d965f67849d75af4
PS C:\Windows\system32>
```

- Ejecutamos el siguiente comando.

**docker ps**

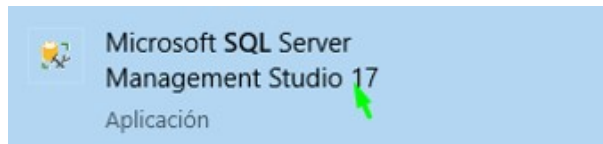
- Para verificar que el contenedor se este ejecutando correctamente.

```
Administrador: Windows PowerShell
PS C:\Windows\system32> docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                    NAMES
795eef0b1aa6   microsoft/mssql-server-linux       "/opt/mssql/bin/sqls...  12 minutes ago Up 12 minutes   0.0.0.0:16111->1433/tcp  SQLLNx01
PS C:\Windows\system32>
```

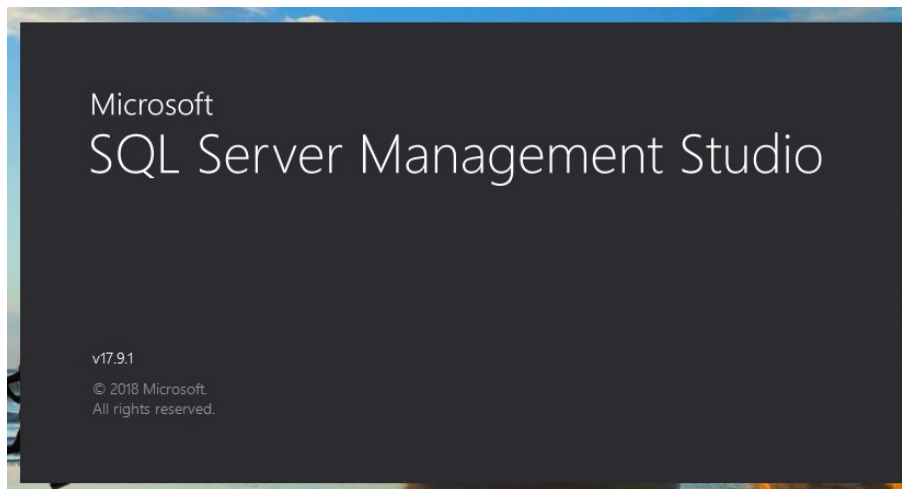
- Esperamos unos segundos.

## 2.5. Consulta en SQL Server

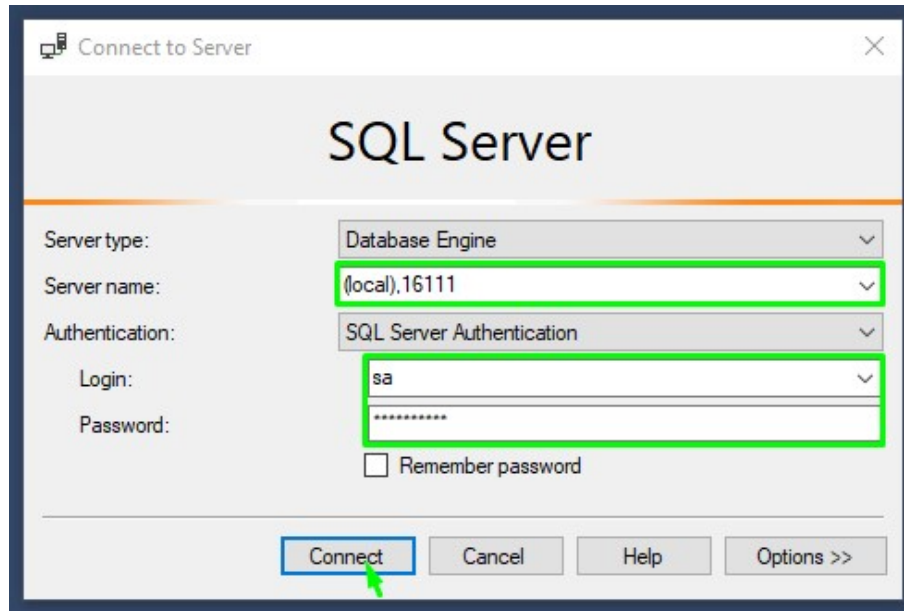
- Buscamos la Aplicación Microsoft SQL Server.



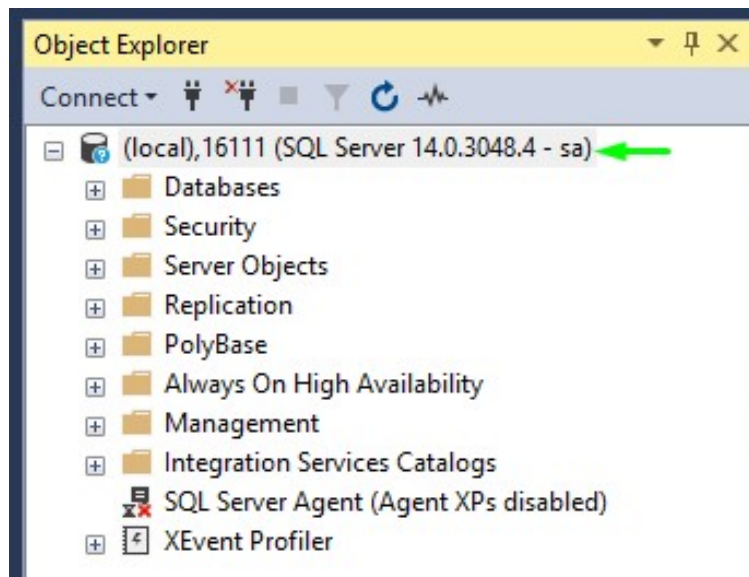
- Primeramente aparecerá esta pestaña y comenzara a cargar el programa.



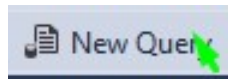
- Ingresamos estos datos por los que se muestra en la imagen para conectarnos a Microsoft SQL Server Management Studio.



- Y ya hemos ingresado a la base de datos por medio del servidor, como se muestra en la imagen.



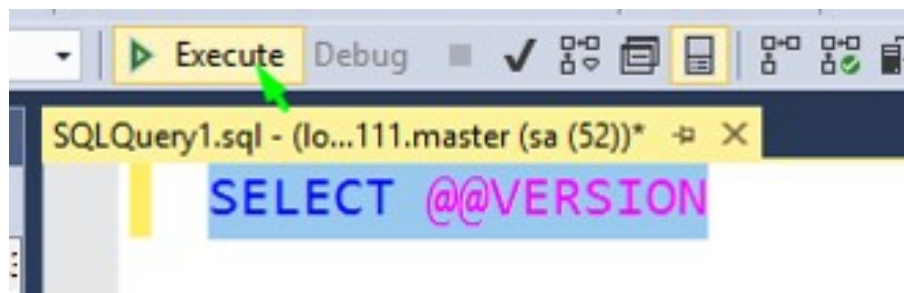
- Seleccionamos en el boton que se muestra en la imagen para agregar una consulta.



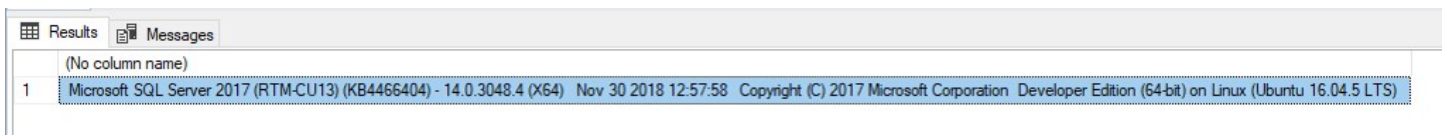
- Digitaremos lo siguiente.

**SELECT @@VERSION**

- Seleccionamos y ejecutamos la consulta.



- Como resultado aparecerá esto.

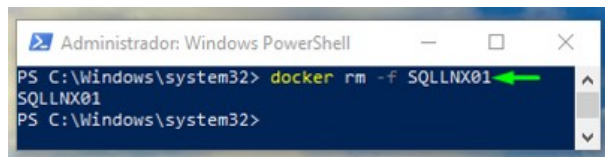
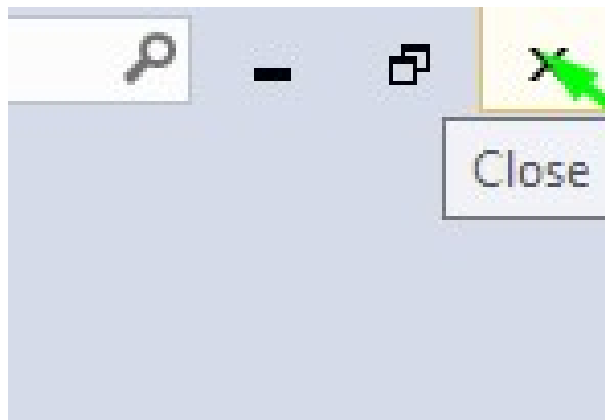


## 2.6. De nuevo en PowerShell - Eliminar contenedor creado

- Ahora cerramos la aplicación Microsoft SQL Server Management Studio.
- De nuevo en PowerShell, digitamos el siguiente código.

**docker rm -f SQLLN01**

- Lo ejecutamos y tendremos como resultado.



- Luego digitamos este comando nuevamente.

**docker ps**

- Lo ejecutamos y tendremos como resultado la eliminacion del contenedor creado.



## 2.7. Adicionando Persistencia.

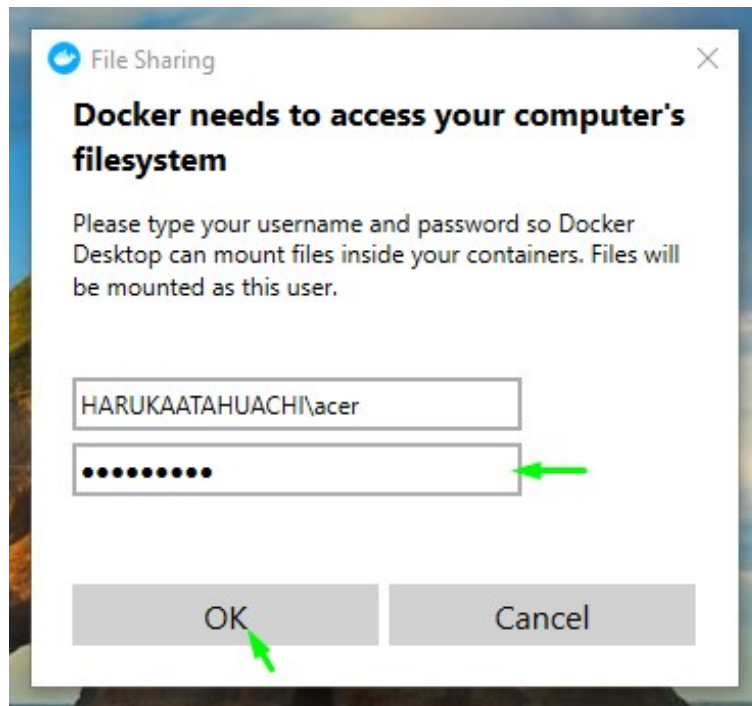
### 2.7.1. Parte 1. Primeramente en PowerShell

- Digitaremos el siguiente comando, como se muestra en la imagen y como respuesta se visualizará un ID que corresponde al contenedor.



```
Administrador: Windows PowerShell
PS C:\Windows\system32> docker run -d -p 16111:1433 -e 'ACCEPT_EULA=Y' -e 'SA_PASSWORD=Tacna.2019' -v D:\DATA\LNK:\var/opt/mssql --name SQLLNK02 microsoft/mssql-server-linux
```

- Luego aparecerá esta pestaña para verificar los datos de la máquina.



- Y finalmente aparecerá la ID que corresponde al contenedor.

```
Administrador: Windows PowerShell
PS C:\Windows\system32> docker run -d -p 16111:1433 -e 'ACCEPT_EULA=Y' -e 'SA_PASSWORD=Tacna.2019' -v F:\DATA\LNK:\var/opt/mssql --name SQLLNK02 microsoft/mssql-server-linux
cf8739259c665a0d497432457ec32fb2cf024023c2e5cb7f6a9b15df76b4987f
PS C:\Windows\system32>
```

- Ejecutamos de nuevo este comando

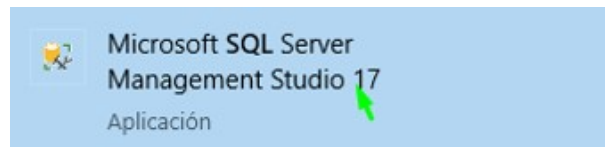
**docker ps**

- Y obtendremos como resultado lo que se muestra en la imagen.

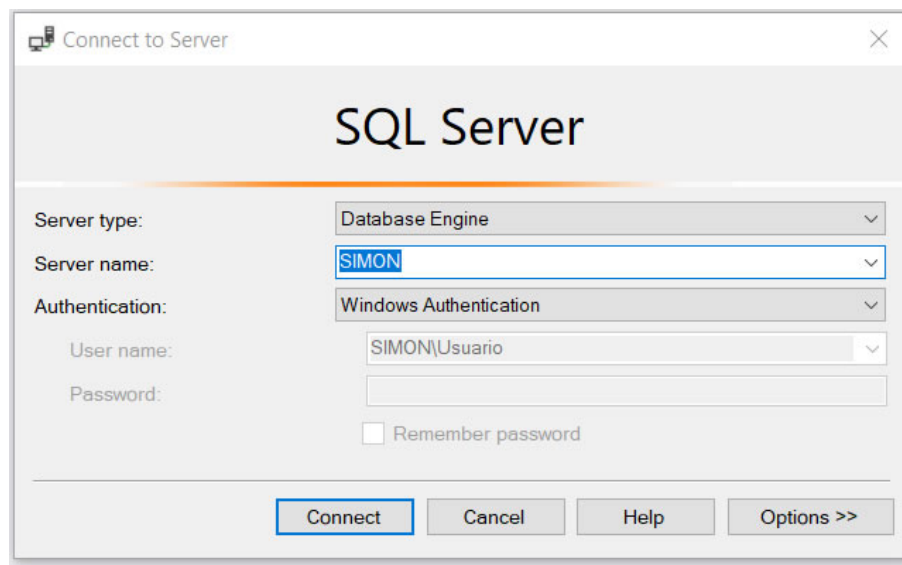
```
Administrador: Windows PowerShell
PS C:\Windows\system32> docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                    NAMES
c10793259c66   microsoft/mssql-server-linux       "/opt/mssql/bin/sqlsà"  About a minute Up About a minute  0.0.0.0:16111->1433/tcp  SQLLNK02
PS C:\Windows\system32>
```

## 2.7.2. Parte 2. Seguidamente en Microsoft SQL Server Management

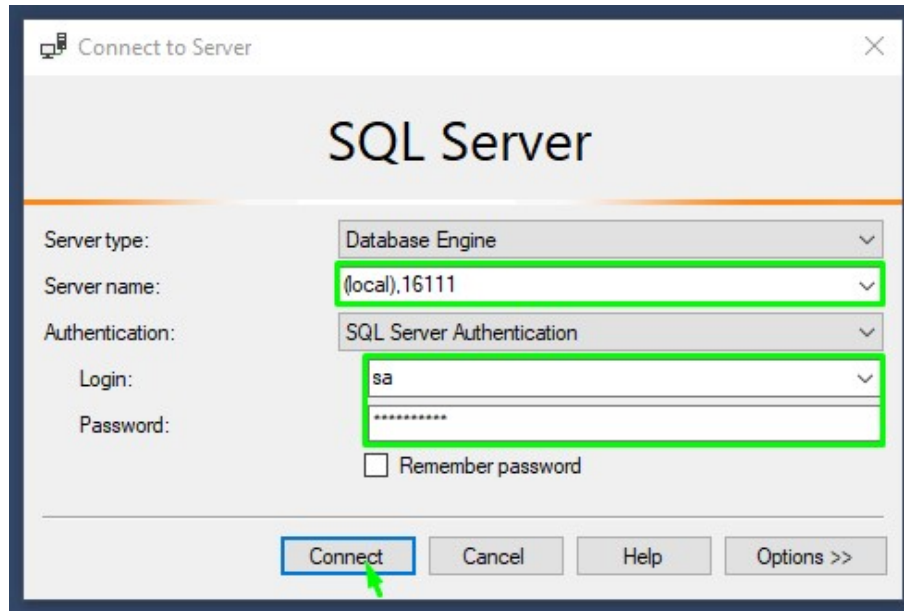
- Buscamos de nuevo el programa Microsoft SQL Server.



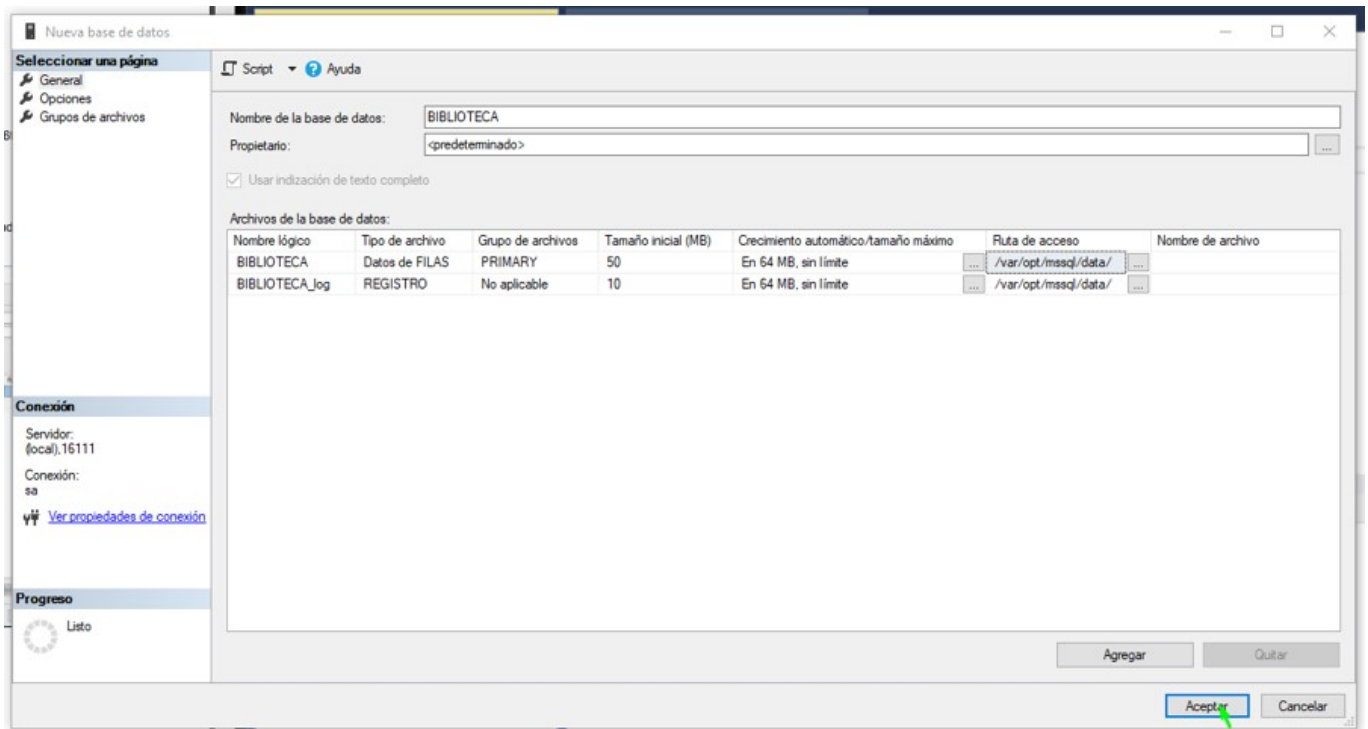
- Se puede observar que por defecto se autentica con el nombre de tu maquina, como en mi caso, HARUKAATAHUACHI



- Volvemos a ingresar estos datos que se muestra en la imagen para conectarnos a Microsoft SQL Server Management Studio.



- Generamos una base de datos de prueba en la Aplicación Microsoft SQL Server Management Studio, según la siguiente imagen.
- O sino mediante el siguiente script a continuación.



- Luego verificamos el contenido de la carpeta DATALNX.

### 2.7.3. Parte 3. Nuevamente en PowerShell

- Ejecutamos el siguiente comando.

```
docker rm -f SQLLNK02
```

- Y el resultado será como se muestra en la siguiente imagen.
- Finalmente ejecutamos el siguiente código para verificar la eliminación del contenedor.

```
dpcker ps
```

- Se mostrara algo como esto.

### 2.7.4. Parte 4. Creando un contenedor con Microsoft SQL Server para Windows.

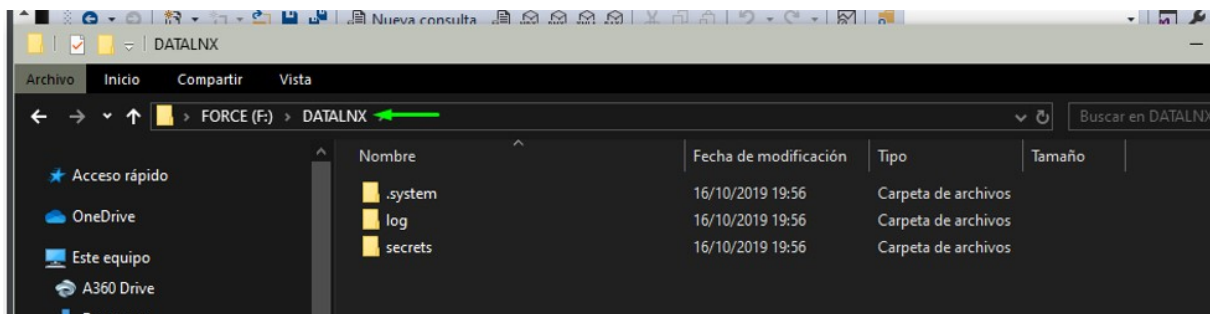
- En el icono de Docker ubicado en el área de notificaciones, hacer click con el botón derecho y utilizar la siguiente opción.
- Ejecutamos el siguiente comando.

```
docker search mssql
```

```

CREATE DATABASE BIBLIOTECA ON
PRIMARY (
    NAME = N'BIBLIOTECA',
    FILENAME = N'/var/opt/mssql/data/BIBLIOTECA.mdf',
    SIZE = 50MB ,
    FILEGROWTH = 10240KB
) LOG ON (
    NAME = N'BIBLIOTECA_log',
    FILENAME = N'/var/opt/mssql/data/BIBLIOTECA_log.ldf',
    SIZE = 10MB ,
    FILEGROWTH = 5MB
)
GO

```



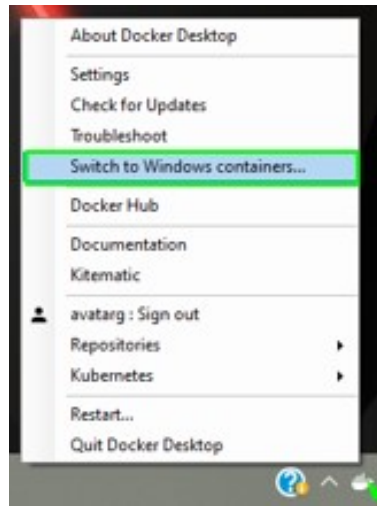
· Y el resultado será como se muestra en la siguiente imagen.



```

Administrator: Windows PowerShell
PS C:\Windows\system32> docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                               NAMES
efb30e7a0c21   microsoft/mssql-server-linux        "/opt/mssql/bin/sqlsà"  5 minutes ago Up 5 minutes  0.0.0.0:16111->1433/tcp             SQLLNK01
PS C:\Windows\system32>

```



- Seguimos con la ejecutamos del siguiente comando.

**docker pull microsoft/mssql-windows-developer**

- Lo cual descargará la imagen del contenedor de Microsoft SQL Server en un servidor Linux, como se muestra en la siguiente imagen.
- Procedemos a ejecutamos el siguiente comando.

**docker images**

- Lo cual deberá visualizar lo siguiente.

```

Administrator: Windows PowerShell
PS C:\Windows\system32> docker search mssql
NAME                                DESCRIPTION                                     STARS     OFFICIAL   AUTOMATED
microsoft/mssql-server-linux        Deprecated SQL Server on Linux Container Repa 1152
microsoft/mssql-server-windows-developer Official Microsoft SQL Server Developer Edità 361
microsoft/mssql-server-windows-express Official Microsoft SQL Server Express Editioà 328
microsoft/mssql-tools                Official images for Microsoft SQL Server Conà 51
rsmoorthy/mssql                     MSSQL Database (version SQL2008)              12
datagrip/mssql-server-linux         SQL Server and SQL Server tools on Linux(201à 9
gantrion/mssql-server-2014-express-windows-with-iis mssql 2014 + IIS                               4
microsoft/mssql-monitoring-influxdb Sample Image for Influxdb. This image is desà 3
mcroe/mssqldocker                   Builds on microsoft/mssql-server-linux and aà 3
tsgkadot/mssql-tools                SQL Server tools on Linux (sqlcmd)             3
ansibleplaybookbundle/mssql-apb     MS SQL Server on Linux (APB)                   2
bitwarden/mssql                     Microsoft/mssql-server-linux with mssql-toolà 2
jhoesl/mssql-server-linux           The Bitwarden database.                       2
metaskills/mssql-server-linux-rails mssql-server-linux with mssql-tools installeà 2
ansibleplaybookbundle/mssql-docker-rhel-official Microsoft SQL Server for Linux - Rails Adaptà 1
dhwanilraval/mssql-docker-rhel-official An APB that deploys Microsoft SQL Server      1
microsoft/mssql-monitoring-collectd SQL Server in Docker                           1
christianacca/mssql-server-windows-express This Sample image is designed to work with tà 1
softwareplant/mssql                 Fork of official microsoft/mssql-server-windà 0
uuthalesians/mssql                  SQL Server test database                      0
plever/mssql-server-linux-plus       Mssql Database Server for Perceptia           0
liaisonintl/mssql-server-linux       Adding additional tooling to the base sql seà 0
tchughesio/mssql-server-linux       mssql-server-linux                           0
astronomerio/mssql-source            CentOS build                                  0
MSSQL source.                        0
PS C:\Windows\system32>

```

```
Administrator: Windows PowerShell
PS C:\Windows\system32> docker pull microsoft/mssql-windows-developer
Using default tag: latest
Error response from daemon: Get https://registry-1.docker.io/v2/microsoft/mssql-windows-developer/manifests/latest: net/http: TLS handshake timeout
PS C:\Windows\system32>
```

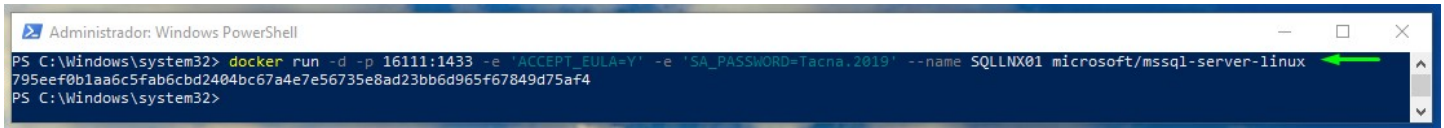
```
Administrator: Windows PowerShell
PS C:\Windows\system32> docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
PS C:\Windows\system32>
```



### 3. Actividades Encargadas.

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

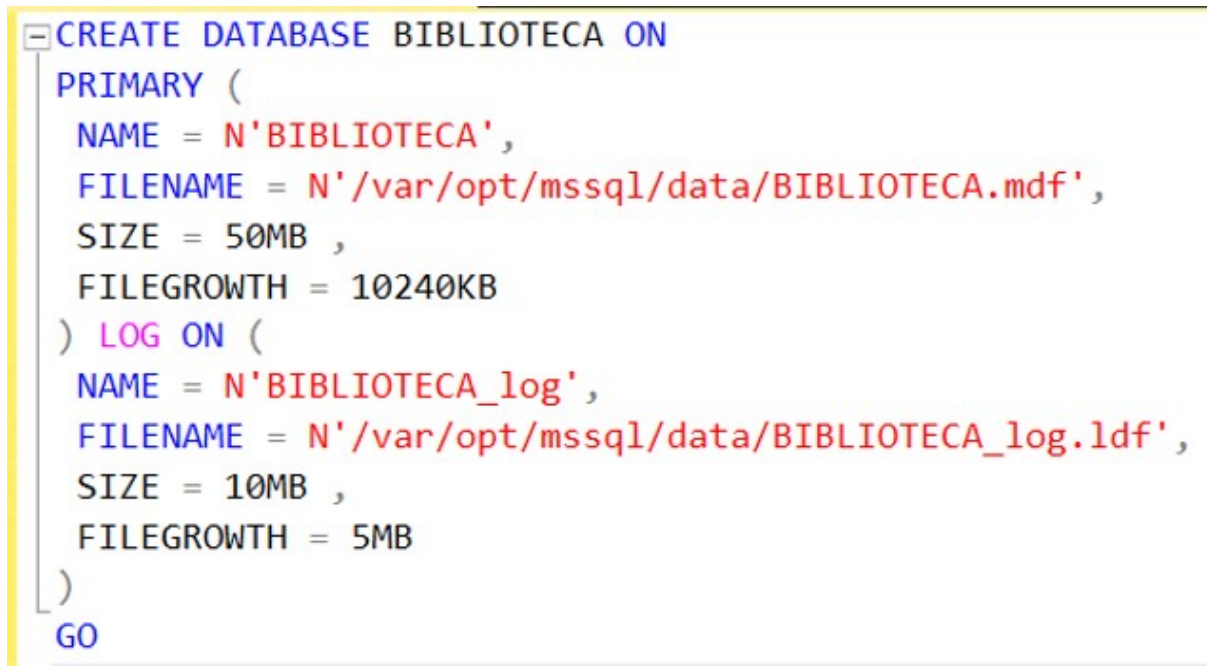
· Respuesta:



```
Administrador: Windows PowerShell
PS C:\Windows\system32> docker run -d -p 16111:1433 -e 'ACCEPT_EULA=Y' -e 'SA_PASSWORD=Tacna.2019' --name SQLLNK01 microsoft/mssql-server-linux
795eef0b1aa6c5fab6cbd2404bc67a4e7e56735e8ad23bb6d965f67849d75af4
PS C:\Windows\system32>
```

3.2. ¿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)?

· Respuesta:



```
CREATE DATABASE BIBLIOTECA ON
PRIMARY (
    NAME = N'BIBLIOTECA',
    FILENAME = N'/var/opt/mssql/data/BIBLIOTECA.mdf',
    SIZE = 50MB ,
    FILEGROWTH = 10240KB
) LOG ON (
    NAME = N'BIBLIOTECA_log',
    FILENAME = N'/var/opt/mssql/data/BIBLIOTECA_log.ldf',
    SIZE = 10MB ,
    FILEGROWTH = 5MB
)
GO
```



**3.3. Genere un nuevo contenedor y cree la Base de Datos con las siguientes características.**

· Respuesta:

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