

An abstract graphic on the left side of the slide, consisting of white lines and circles on a blue background, resembling a circuit board or a network diagram.

DOCKER

BERNAT COSTA



DOCKER COMO HERRAMIENTA DE TRABAJO

- Docker nos permitirá "virtualizar" varias bbdd en un mismo equipo
- Un contenedor por servicio
- Podremos tener varios mysql, sqlservers, apaches... en una misma máquina funcionando en puertos distintos

¿PORQUE DOCKER Y NO VIRTUALBOX?

Más ágil de levantar máquinas

No replica todo el SO, el kernel se aprovecha el del anfitrión

Contenedores más livianos que Maquinas virtuales

Contenedores preconfigurados.

¿QUE DEBO SABER DE DOCKER?

- Imagenes y contenedores
- Puertos
- Variables de entorno
- Volumenes



IMAGENES Y CONTENEDORES

Con docker vamos a construir contenedores a partir de una imagen.

La imagen es la base de nuestro contenedor.

Docker nos garantiza que una imagen, se comportará igual en cualquier host.

Los contenedores son volátiles. Cuando se paran y destruyen, los datos se pierden.

Un contenedor corre un servicio (una bbdd, un servidor web, un servidor de correo....)

Parámetros de configuración

¿DE DONDE SACAMOS LAS IMAGENES?

- Las podemos crear con un fichero Dockerfile y un comando de docker (docker build).
- No nos interesa crear imagenes, sino usarlas
- [Hub.docker.com](https://hub.docker.com)



¿COMO CORREMOS UN CONTENEDOR?



Arrancar un apache:

```
docker run httpd
```

En background

```
docker run -d httpd
```

Ponerle un nombre al contenedor

```
docker run -d --name apachealpine httpd
```

Tags

```
docker run -d --name apachealpine httpd:alpine
```



QUE PODEMOS HACER CON UN CONTENEDOR

- Ver la lista de contenedores corriendo
 - Docker ps
- Ver el log de un contenedor
 - Docker logs nombredelcontenedor (o el hash)
- Parar/arrancar/borrar un contenedor:
 - Docker stop/start/rm

PUERTOS

- Para que el contenedor sea útil, debemos exponer puertos del contenedor en la máquina anfitrión.
- ¿De que nos sirve un apache si no nos da una web?
- Con el parámetro -p "enlazamos" el puerto de nuestra máquina con el del contenedor.
 - `docker run -d --name apachealpine -p 81:80 httpd:alpine`
- **IMPORTANTE:** cada contenedor, tendrá sus puertos abiertos. Mirar la documentación en hub.docker.com



VARIABLES DE ENTORNO DEL CONTENEDOR

- Hay imagenes que te permiten pasarle variables de entorno, por ejemplo, el pass de root de un sgbbdd.
 - `docker run -d --name mysql -e MYSQL_ROOT_PASSWORD=root -p 3306:3306 mysql:5.7.32`
- Ahora, con un cliente de mysql, podriamos conectarnos al localhost:3306 con password root.

VOLÚMENES

- Los contenedores son volátiles. No son persistentes
- Las bbdd son persistentes.
- ¿Como podemos montar una bbdd en un contenedor volatil?
- Parámetro v
 - `docker run -d --name mysql -e MYSQL_ROOT_PASSWORD=root -p 3306:3306 -v mysql_data:/var/lib/mysql mysql:5.7.32`

DOCKER-COMPOSE

- Es una herramienta para levantar y parar contenedores de forma amigable
- Son scripts, donde pondremos toda la configuración, y así nos ahorramos de poner todos los comandos.
- Se basa en un fichero `docker-compose.yml`
- Se arranca/para con
 - `Docker-compose -d up`
 - `Docker-compose down`

EJEMPLOS DOCKER COMPOSE

Un wordpress
con su bbdd

Una bbdd
Microsoft
SQLServer
persistente.

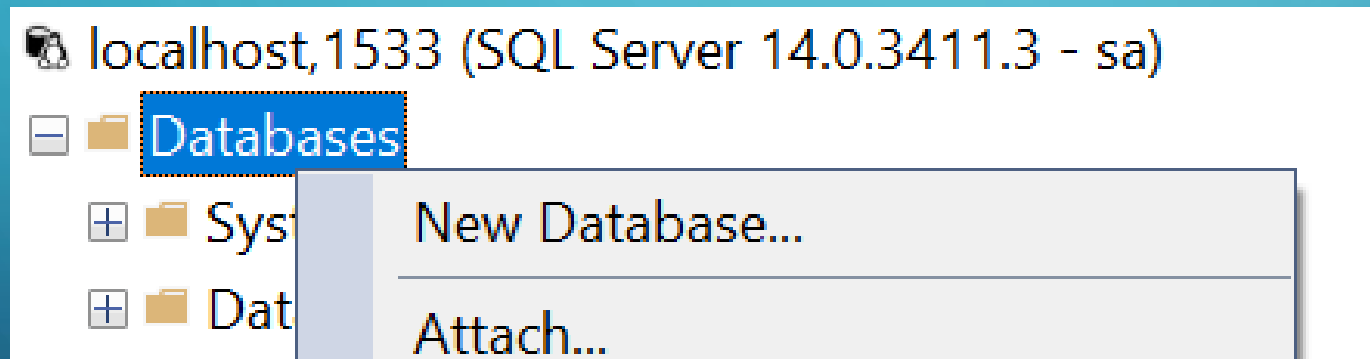


EJERCICIO

1. Vamos a levantar un SQL Server
2. Instalaremos en nuestro host el cliente SQL Server para conectarnos
3. Restauraremos un backup de AdventureWorks
4. Analizaremos la bbdd
5. Realizaremos consultas varias sobre esa bbdd.

¿COMO RESTAURAMOS UN BACKUP?

Una vez conectados con el SQL Management Studio, creamos una BBDD donde restaurar nuestro backup:



INDICAMOS NOMBRE...

New Database

Select a page

- General
- Options
- Filegroups

Script Help

Database name: AdventureWorks2017

Owner: <default>

☒ Use full-text indexing

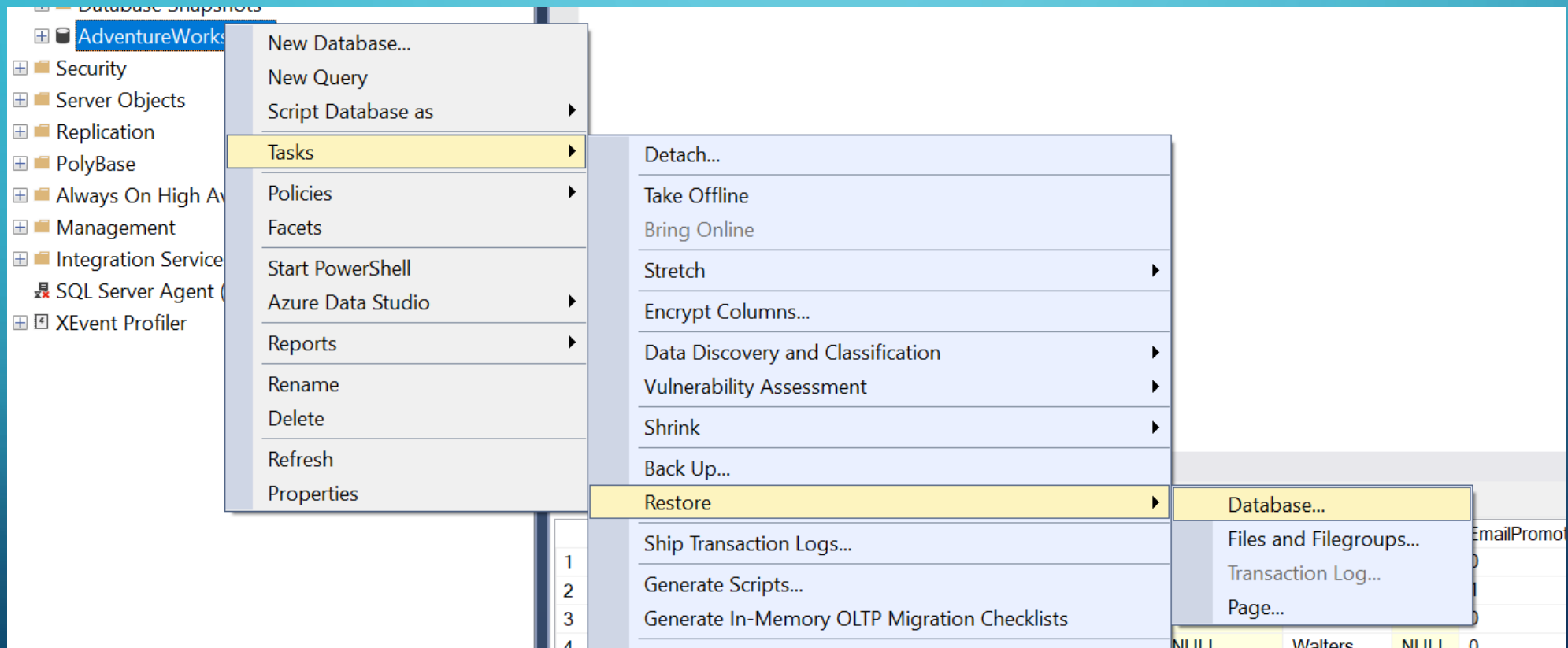
Database files:

Logical Name	File Type	Filegroup	Initial Size (M...	Autogrowth / Maxsize	Path
AdventureW...	ROWS ...	PRIMARY	8	By 64 MB, Unlimited	/var/opt/r
AdventureW...	LOG	Not Applicable	8	By 64 MB, Unlimited	/var/opt/r

Connection

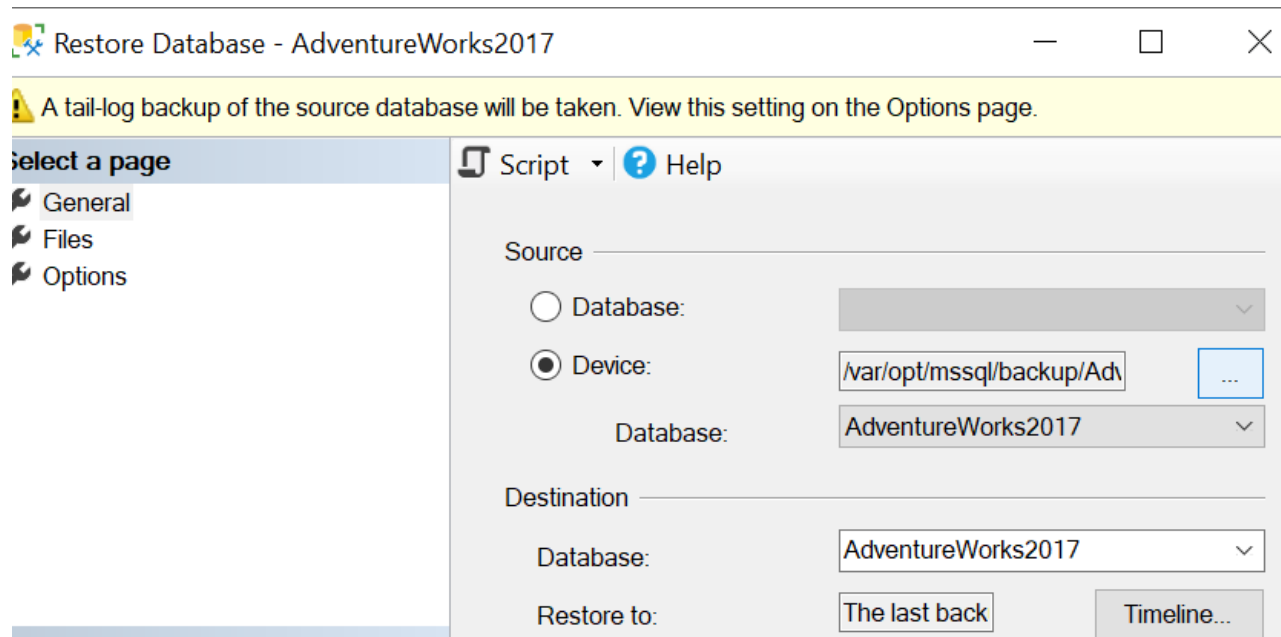
Server:

RESTAURANDO BACKUP...



BUSCAMOS EL BACKUP

- Seleccionamos Devide, y en los ... seleccionamos el fichero de backup.
- Importante haber copiado antes el backup a nuestro contenedor. Lo podemos hacer con un nuevo volumen para la carpeta backups.



DOCKER COMPOSE CON LINEA DE BACKUP

```
🐙 docker-compose.yml
1  version: "3.2"
2  services:
3
4    sql-server-db:
5      container_name: sql-server-db
6      image: mcr.microsoft.com/mssql/server:2017-latest
7      ports:
8        - "1533:1433"
9      environment:
10        SA_PASSWORD: "12345Ab##"
11        ACCEPT_EULA: "Y"
12      volumes:
13        - ./backup:/var/opt/mssql/backup
14        - ./data:/var/opt/mssql/data
15
```

MOVER LOS FICHEROS....

- Como nuestro backup está hecho sobre un servidor windows, las rutas son c:, hay que cambiarlas por rutas Unix.

⚠ A tail-log backup of the source database will be taken. View this setting on the Options page.

Select a page

- General
- Files
- Options

Script | ? Help

Restore database files as

☒ Relocate all files to folder

Data file folder :

Log file folder :

Logical File Name	File Type	Original File Name
AdventureWorks...	Rows Data	C:\Program Files\Microsoft S
AdventureWorks...	Log	C:\Program Files\Microsoft S

Connection

REESCRIBIR LA BBDD EN BLANCO EXISTENTE

Select a page

- General
- Files
- Options

Connection

localhost, 1533 [sa]

[View connection properties](#)

Progress

Done

Script Help

Restore options

- ☒ Overwrite the existing database (WITH REPLACE)
- ☐ Preserve the replication settings (WITH KEEP_REPLICATION)
- ☐ Restrict access to the restored database (WITH RESTRICTED_USER)

Recovery state: RESTORE WITH RECOVERY

Standby file: /var/opt/mssql/data/AdventureWor

Leave the database ready to use by rolling back uncommitted transaction

Tail-Log backup

- ☐ Take tail-log backup before restore
 - ☐ Leave source database in the restoring state (WITH NORECOVERY)

Backup file: /var/opt/mssql/data/AdventureWor

Server connections

- ☐ Close existing connections to destination database

Prompt

OK Cancel Help

DOCKER NATIVO SOBRE GNU/LINUX

- Para docker nativo sobre un sistema Linux, no es necesario crear la BBDD antes, se puede restaurar directamente desde el backup. Solo hará falta cambiar las rutas.
- Desde Linux, se puede hacer desde comandos SQL con las siguientes instrucciones:
- Esto pasa por que Docker Desktop no gestiona los permisos en las carpetas de los volúmenes de la misma manera y da problemas si restauras la BBDD sin antes crear una misma en blanco con el mismo nombre. Sin Docker Desktop, con una máquina linux, no es necesario.

```
-- 1 descargar el backup dentro del contenedor:
--1.1 entrar en la consola del contenedor:
    -- docker exec -it sql-server-db /bin/bash
--1.2 crear carpeta backups
    -- mkdir /var/opt/mssql/BACKUP
--1,3 descargar backup:
    -- wget https://github.com/Microsoft/sql-server-samples/releases/download/adventureworks/AdventureWorks2017.bak

--una vez copiado el fichero .bak al directorio de backups del SQLServer de nuestro contenedor, ejecutar estas consultas:

--Para ver los ficheros de la bbdd a restaurar:
RESTORE FILELISTONLY FROM DISK = '/var/opt/mssql/backup/AdventureWorks2017.bak'

--Para restaurar la bbdd cambiando la ruta de los ficheros de c:/...
--a una ruta linux compatible con nuestro contenedor
RESTORE DATABASE AdventureWorks2017
FROM DISK = '/var/opt/mssql/backup/AdventureWorks2017.bak' WITH
MOVE 'AdventureWorks2017' TO '/var/opt/mssql/data/AdventureWorks2017.mdf',
MOVE 'AdventureWorks2017_log' TO '/var/opt/mssql/data/AdventureWorks2017_log.ldf'
```