



Using Docker to Restore MSSQL Databases on M1 Macs

by Ryan Malani

📅 February 23, 2023

🕒 4 min read

Like many businesses, we decided to make a change in our internal software tools for attracting talent, and while making the change we were faced with the challenge of ensuring that our data was accessible, intact, and could be used in creative ways going forward regardless of the new platform we chose. With that being said, we were provided a .bak file¹, .cer file², and a .pvk file³ from the provider we were moving from.

getting started

Much of the getting started will be inspired by a Medium article from Thenusan Santhirakumar⁴. There are a couple of things that you'll need to get started on this project. Some of these may be interchangeable with other tools available on the M1/M2 Mac platforms, but for simplicity I'll dive deep into the methodology we used. Below are a couple tools you'll need access to [with download links, **be sure to download the Apple Silicon versions if using an M1/M2 Mac**]:

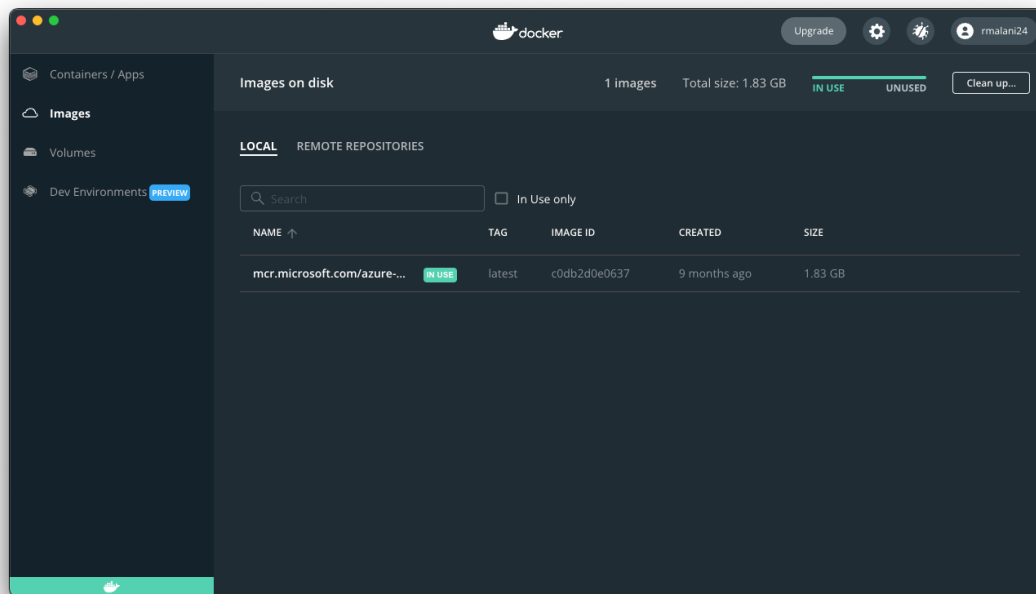
1. [Azure Data Studio](https://learn.microsoft.com/en-us/sql/azure-data-studio/download-azure-data-studio?view=sql-server-ver16&culture=en-us&country=us&tabs=redhat-install%2Credhat-uninstall) (<https://learn.microsoft.com/en-us/sql/azure-data-studio/download-azure-data-studio?view=sql-server-ver16&culture=en-us&country=us&tabs=redhat-install%2Credhat-uninstall>)
2. [Docker](https://www.docker.com/products/docker-desktop/) (<https://www.docker.com/products/docker-desktop/>)
3. Certificate credentials

setting up docker

First will be creating a login for docker (<https://hub.docker.com>) if you don't already have one. The following commands will be inspired by the official Microsoft documentation^[^5]. You'll need to open up the terminal and run the following command first:

```
docker pull mcr.microsoft.com/azure-sql-edge:latest
```

Once the Azure SQL Edge image has been successfully downloaded, you should see it in your desktop application:



Next, you'll need to execute the image in a container on a localhost port using this command (**please note that here is where you will name your container using the `--name` option and set your password with `MSSQL_SA_PASSWORD`**):

```
docker run --cap-add SYS_PTRACE -e 'ACCEPT_EULA=1' -e  
'MSSQL_SA_PASSWORD=yourStrong(!)Password' -p 1433:1433 --name  
azuresqledge -d mcr.microsoft.com/azure-sql-edge
```

setting up azure data studio

When you first open Azure Data Studio, you'll be greeted by the main get started menu. From there, you'll want to create a new connection. Make sure your container is running, and then a connection menu will open up and you'll input the following information:

- Connection type: **Microsoft SQL Server**
- Server: **localhost**
- Authentication type: **SQL Login**
- User name: **sa**
- Password: **The password you just made**
- Database: **Default**
- Encrypt: **True**
- Trust server certificate: **True**
- Server group: **Default**

moving files to the container

In order to successfully restore the database from the backup file, you'll have to move the .bak, .cer, and .pvk files from your local machine to the docker container and ensure that the files have the correct permissions. To do so, use the following commands in your terminal, replacing the paths appropriately and replacing container-name with the name of your container [e.g. azuresqlledge:var/opt/mssql/backup/file.cer]:

```
docker cp local/path/to/file container-  
name:/path/to/file/on/container
```

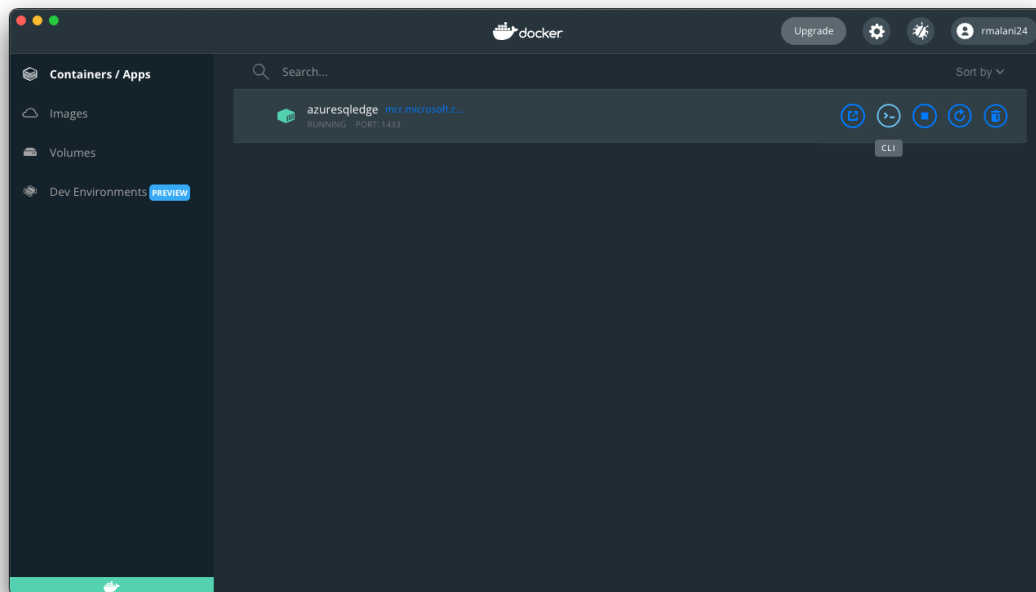
Run the command above to copy each file to the container. Make sure to do this for each file, including .bak, .cer, and .pvk files. Once that is complete, you'll need to change the permissions of the files in the

container. Traditionally, this has been a pain point of Docker, but I've found the following command to be successful so that the server connection in Azure Data Studio can access them properly:

```
sudo docker exec -u root container-name chmod 644  
path/to/file/on/container
```

This command will give the file's owner read and write access, group read access, and public read access. If you prefer to use other values, reference the [chmod calculator](https://chmod-calculator.com) (<https://chmod-calculator.com>).

Once these have executed, you can check your work by opening Docker Desktop, and clicking the CLI option of your container to open the terminal of your container:



With the CLI open, use the `cd` command followed by the path to navigate to the files and the `ls -l` command to list out the files in that directory and their details (including permissions). If you used 644, it should read **rw-r--r--** like the following:

```
ryanmalani — com.docker.cli • docker exec -it  
~ — com.docker.cli • docker exec -it  
$ cd var/opt/mssql/backup  
$ ls -l  
[total 8  
drwxr-xr-x 2 mssql root    4096 Feb 21 20:43  
drwxr-xr-x 2 501 dialout 4096 Feb  9 16:26  
$ cd  
$ ls -l  
[total 7006964  
-rw-r--r-- 1 501 dialout    742 Feb 10 16:31 .cer  
-rw-r--r-- 1 501 dialout 7175116288 Feb 10 16:32 .bak  
-rw-r--r-- 1 501 dialout   1212 Feb 10 16:31 .pvk  
$
```

restoring the database

Now that your localhost server is connected and your container is up and running, it's time to restore the database from the files you were provided. There are a couple of things you'll need to do. Open a New Query from the master database and do the following:

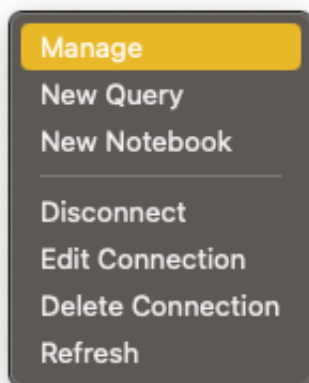
1. Create a master key. Make sure to replace "yourPassword" with a strong, personalized password.

```
USE master  
GO  
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'yourPassword'  
GO
```

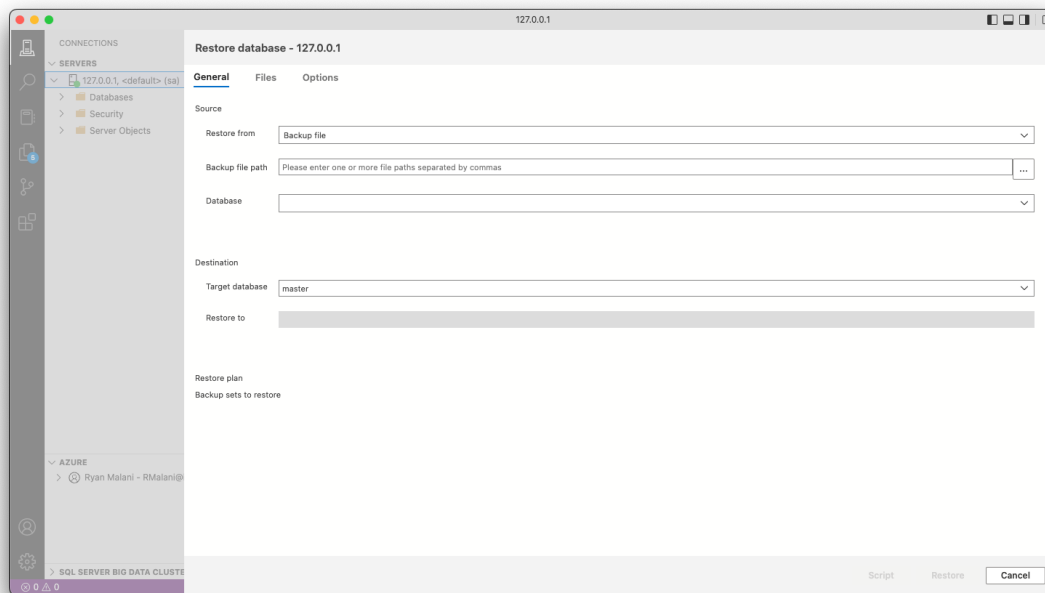
1. Create a certificate in Azure Data Studio using the master key, certificate file, and private key file. Replace the master key decryption password with the one you just created in step 1 and the certificate/private key decryption password with the one provided by the creator of the backup file. Replace `certificate_name` with the name of the certificate preceeding the `.cer` extension, and replace the file paths with the path you chose and the respective names of your files.

```
USE master
GO
OPEN MASTER KEY DECRYPTION BY PASSWORD='yourPassword'
CREATE CERTIFICATE certificate_name
FROM FILE = '/var/opt/mssql/backup/your_file.cer'
WITH PRIVATE KEY(
FILE = '/var/opt/mssql/backup/your_file.pvk',
DECRYPTION BY PASSWORD = 'providerPassword'
)
```

Now, it's time to restore from the backup file. With the server connected, you want to right click on it and click manage.



That will open up the server's dashboard, and of the options "New Query", "New Notebook", and "Restore" you'll select **Restore**. Under Source, select Restore from->**Backup file**. The menu you see should look like this:



Now, you'll want to click the three dots next to the Backup file path and it should open the directory structure of your container. If you chose the more traditional path, your file will be under **var/opt/mssql/backup/your_file.bak**. Select that file and click **OK**. The rest of the fields should autofill and you can click **Restore**. Give it a few minutes, and your database should be successfully restored locally!

usage

With your database now restored locally, you can begin to query the data using SQL commands! For reference on how to use SQL, I recommend W3Schools (https://www.w3schools.com/sql/sql_syntax.asp). Any questions? Feel free to contact us and ask away.

-
1. <https://file.org/extension/bak> (<https://file.org/extension/bak>) ↩
 2. <https://file.org/extension/cer> (<https://file.org/extension/cer>) ↩
 3. <https://file.org/extension/pvk> (<https://file.org/extension/pvk>) ↩
 4. <https://medium.com/geekculture/how-to-install-sql-server-in-mac-m1-41121e110214> (<https://medium.com/geekculture/how-to-install-sql-server-in-mac-m1-41121e110214>) ↩