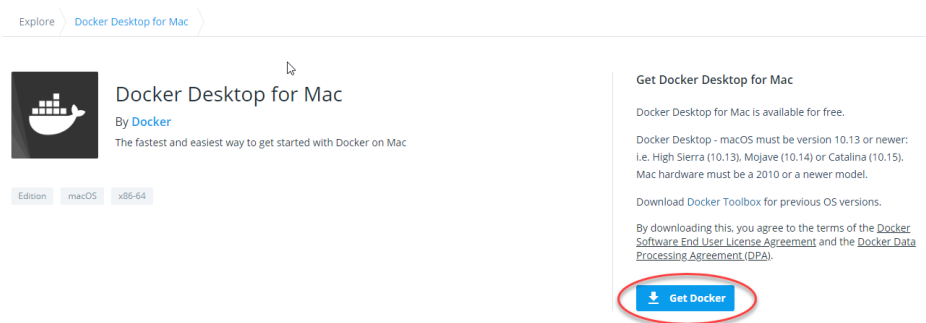# Software required for this class

For Mac Users:

A. Install and configure Docker for SQL Server

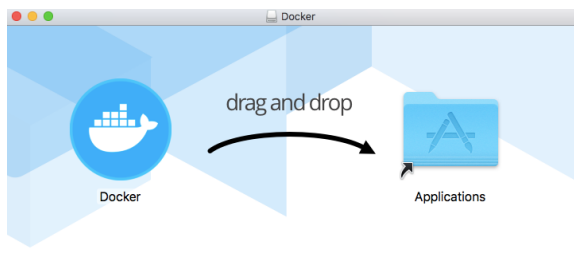Follow the steps below to download and install Docker including configuring it for SQL Server[1].

Step 1: Download Docker

Visit the Docker Community Edition for Mac download page and click on **Get Docker**

Step 2: Install Docker

Double-click on the downloaded .dmg file and then drag the Docker.app icon to your Application folder.


Step 3: Launch Docker

Launch Docker the same way you'd launch any other application through Launchpad or the application folder or by searching in the search bar etc.

When you first Launch Docker, you might be prompted for your **password**. You can go ahead and provide it as Docker will need that information in order to install its networking components and other dependencies.

---

[1] The configuration involves allocating enough memory to Docker so that SQL Server can run without hinderance.

Click **OK**
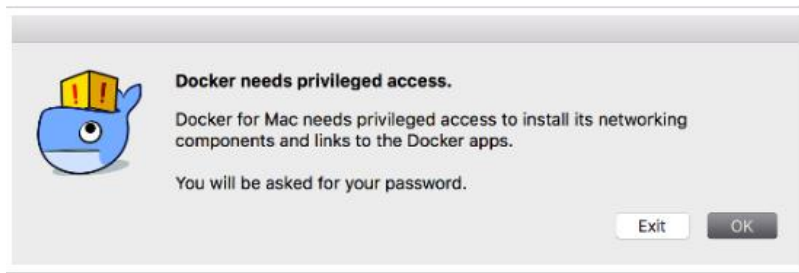
Step 4: Increase the Memory

SQL Server needs at least 3.25GB of memory to run, but Docker only allocates 2GB by default. Therefore, you should increase the memory allocation to allow SQL Server to run. It is recommended to allocate at least **4GB** (just to be on the safer side).

To do this, follow the steps below that are accompanied by screenshots:

1. Click on the little Docker icon in the top menu
2. Select **Preferences...** from the options
3. Select **Advanced** (if it's not already selected)
4. Slide the memory slider up to 4GB
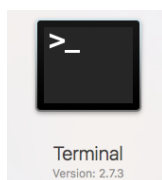5. Click **Apply & Restart**
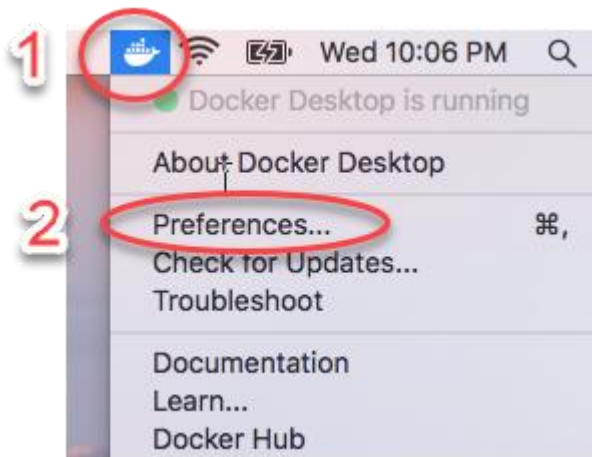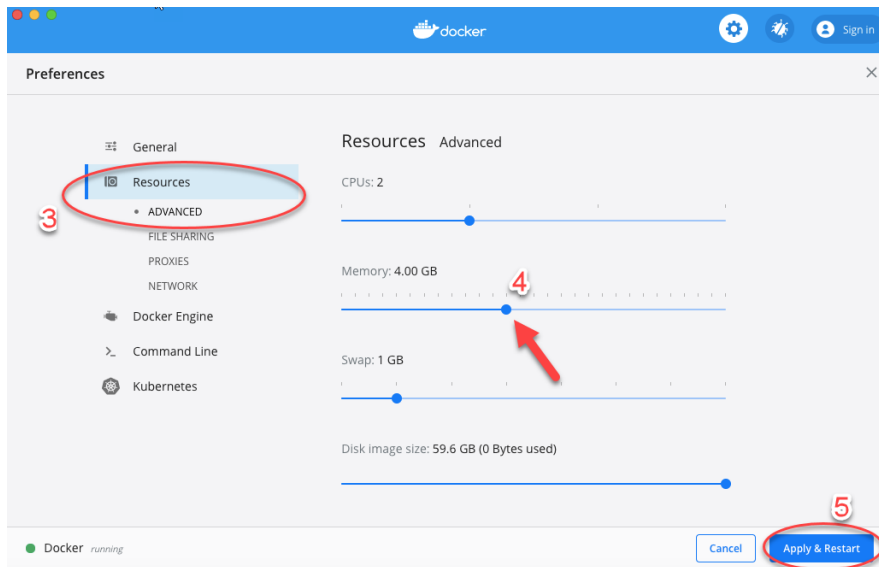
Now that you've installed Docker and have increased its memory allocation, you can proceed to install SQL Server.

B. How to Install SQL Server

Step 1: Pull the SQL Server Image

Open a Terminal window (the icon of terminal looks like the screenshot below):

Type the following command and run it (i.e. hit the Enter key on your keyboard):

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

This will pull the latest SQL Server for Linux docker image to your computer. If you run it correctly, the output should look like this screenshot

```
[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
```

Step 2: Launch the SQL Server Image

Run the following command to launch an instance of the Docker image you just downloaded.

**docker run -d --name sqlcourse -e 'ACCEPT_EULA=Y' -e
'SA_PASSWORD=1VERYStrongSecurePassw0rd' -p 1433:1433
microsoft/mssql-server-linux**

Note: You should type the command all in one line or you can copy and paste as deemed fit. You can feel free to replace the container name (**sqlcourse**) and password (**1VERYStrongSecurePassw0rd**) with your own chosen name and password before running the command. Be sure to use a strong password, otherwise you may get an error. Also, make note of your password as you will need it later to connect to SQL Server.

Here's an explanation of the above parameters:

-d
This is an optional parameter that launches the Docker container in daemon mode. This means that it runs in the background and doesn't need its own Terminal window open.

--name sqlcourse
This optional parameter provides a name for the container. This will come in handy when stopping and starting the container from the Terminal.

-e 'ACCEPT_EULA=Y'
The Y shows that you agree with the EULA (End User License Agreement). This is required in order to install SQL Server.

-e 'SA_PASSWORD=myPassw0rd'
This is a required parameter that sets the sa database password.

-p 1433:1433
This maps the local port 1433 to port 1433 on the container. This is the default TCP port that SQL Server uses to listen for connections.

microsoft/mssql-server-linux
This tells Docker which image to use.

Step 3: Check the Docker container to ensure that it is running (optional)

Type the following command to check that the Docker container is running.

```
docker ps
```

If it's up and running, then it should return something like this:

```
CONTAINER ID    IMAGE                         COMMAND                  CREATED          STATUS           PORTS                     NAMES
4d12f1e17c25    microsoft/mssql-server-linux   "/opt/mssql/bin/sqls…"   36 seconds ago   Up 35 seconds    0.0.0.0:1433->1433/tcp    sqlcourse
```

C. Install Azure Data Studio and use it to connect to SQL Server

Step 1: Download Azure Data Studio

1. Visit the Azure Data Studio download page

2. Scroll down to where it says **Get Azure Data Studio for macOS** and click on 'Azure Data Studio for macOS to download the zip file. See screenshot below

Step 2: Install Azure Data Studio

1. Once the .zip file is finished downloading as shown below, click to expand its contents
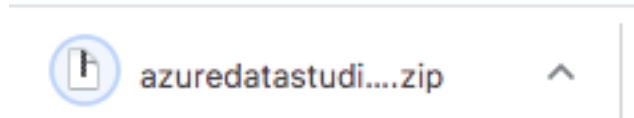
(Double) click to open the Azure Data Studio app. You can also drag the .app file to the Applications folder as you did previously with the Docker app.

Step 3: Launch Azure Data Studio and connect to SQL Server

Launch Azure Data Studio (by clicking on its icon in your Launchpad or the Applications folder).

Click 'New connection'
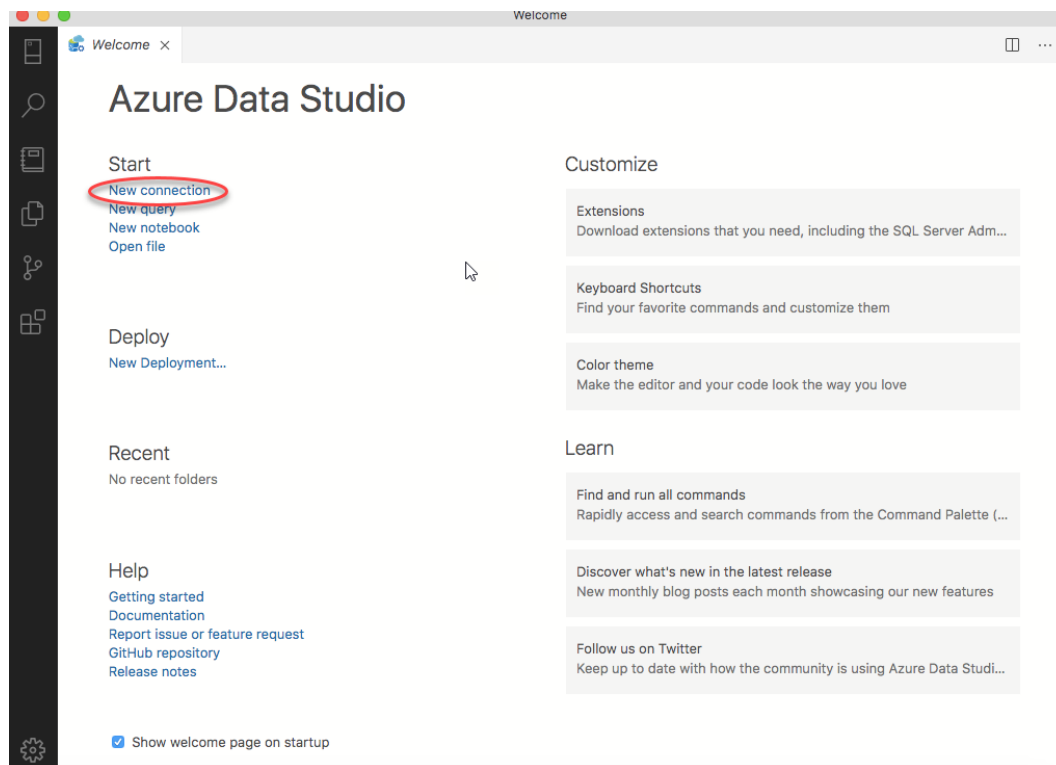
Enter the following SQL Server connection details

- Server: **127.0.0.1**
- Authentication Type: choose **SQL Login**
- User name: **sa**
- **Password**: \<enter the password you chose before, from page 4\>
- Check the box for **Remember Password**

Click **Connect**

Connection Details

Connection type	Microsoft SQL Server
Server	127.0.0.1
Authentication type	SQL Login
User name	sa
Password	....
	☑ Remember password
Database	\<Default\>
Server group	\<Default\>
Name (optional)	

Advanced...

Connect   Cancel