DATABASE.GUIDE

How to Install SQL Server on a Mac

JULY 30, 2017 / IAN

Here I'll show you how to get SQL Server 2017 up and running on your Mac in less than half an hour. And the best part is, you'll have SQL Server running locally without needing any virtualization software.

Prior to SQL Server 2017, if you wanted to run SQL Server on your Mac, you first had to create a virtual machine (using VirtualBox, Parallels Desktop, VMware Fusion, or Bootcamp), then install Windows onto that VM, then finally SQL Server. This is still a valid option depending on your requirements (here's how to install SQL Server on a Mac with VirtualBox if you'd like to try that method).

Starting with SQL Server 2017, you can now install SQL Server directly on to a Linux machine. And because macOS is Unix based (and Linux is Unix based), you can run SQL Server for Linux on your Mac. The way to do this is to run SQL Server on Docker.

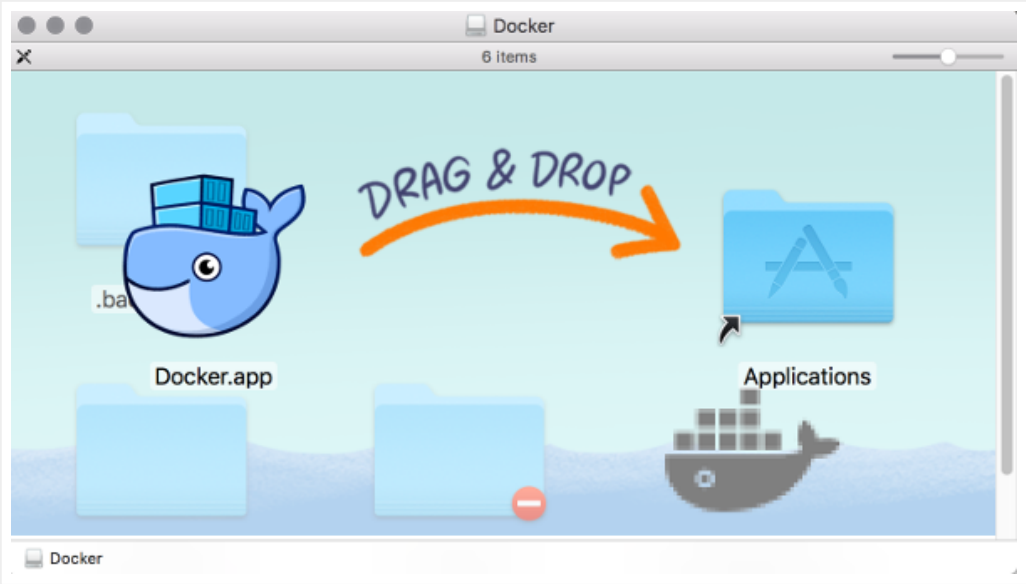
So let's go ahead and install Docker. Then we'll download and install SQL Server.

1. Install Docker

Download the (free) Docker Community Edition for Mac (unless you've already got it installed on your system). This will enable you to run SQL Server from within a Docker container.

To download, visit the Docker CE for Mac download page and click `Get Docker`.

To install, double-click on the `.dmg` file and then drag the `Docker.app` icon to your `Application` folder.



Docker installation on a Mac.

What is Docker?

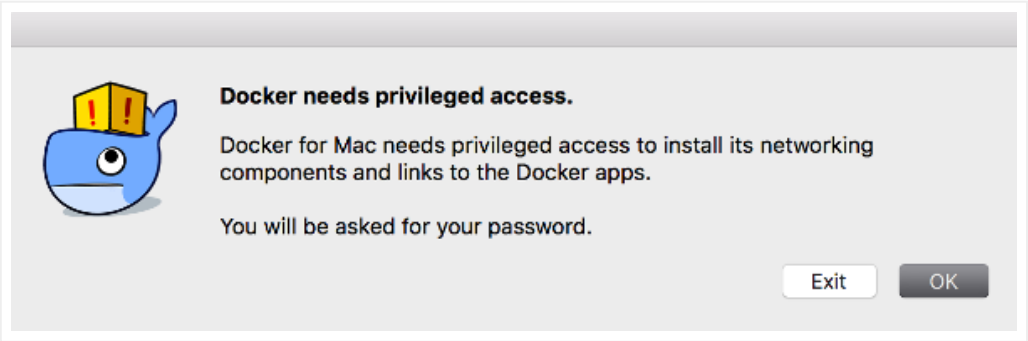
Docker is a platform that enables software to run in its own isolated environment. SQL Server 2017 can be run on Docker in its own isolated container. Once Docker is installed, you simply download — or "pull" — the SQL Server on Linux Docker Image to your Mac, then run it as a Docker container. This container is an isolated envionment that contains everything SQL Server needs to run.

2. Launch Docker

Launch Docker the same way you'd launch any other application (eg, via the Applications folder, the Launchpad, etc).

When you open Docker, you might be prompted for your password so that Docker can install its networking components and links to the Docker apps. Go

ahead and provide your password, as Docker needs this to run.



The password request dialog

3. Increase the Memory

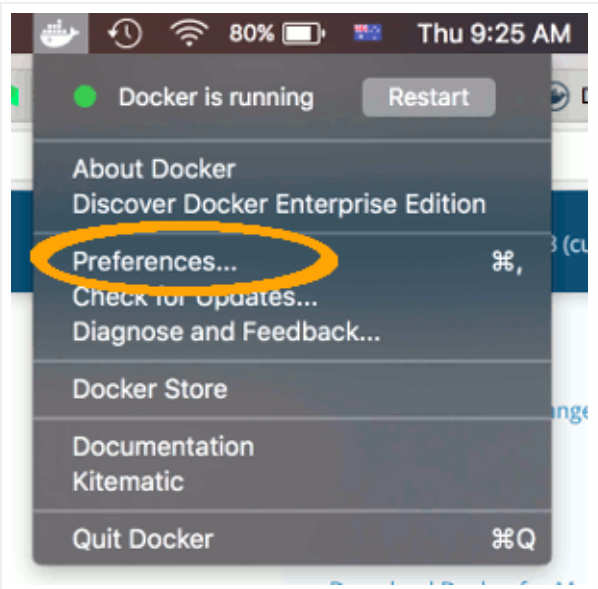
By default, Docker will have 2GB of memory allocated to it. SQL Server needs at least 3.25GB. To be safe, increase it to 4GB if you can.

To do this:

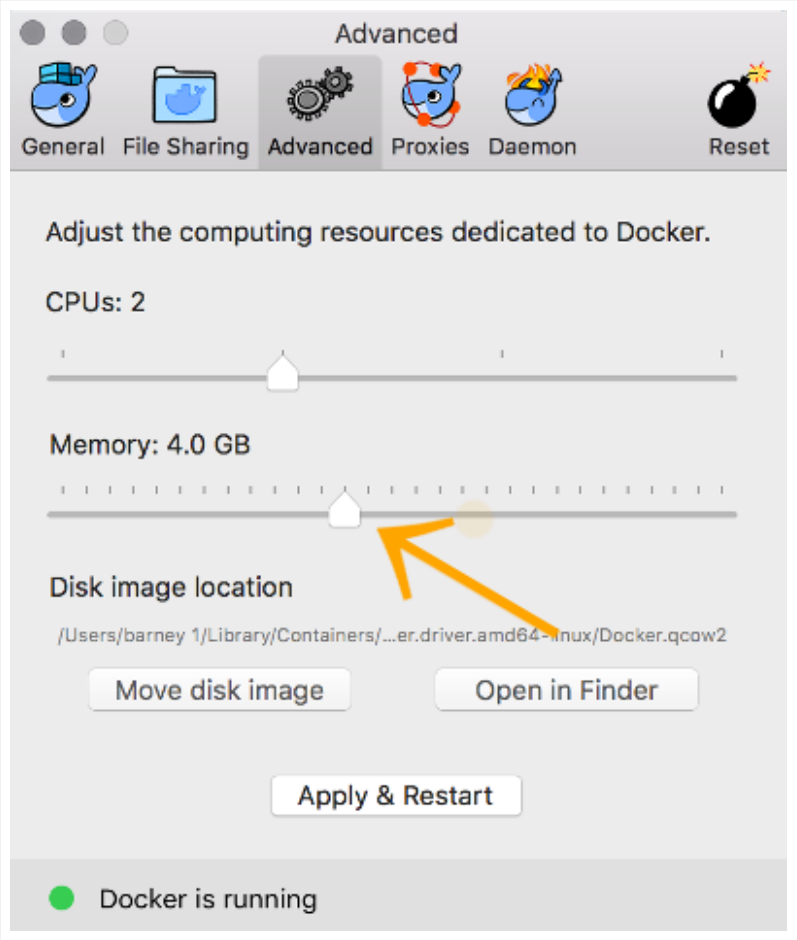
a. Select `Preferences` from the little Docker icon in the top menu

b. Slide the memory slider up to at least 4GB

c. Click `Apply & Restart`



Selecting the preferences.



Increasing the memory.

4. Download SQL Server

Now that Docker is installed and its memory has been increased, we can download and install SQL Server for Linux.

Open a Terminal window and run the following command.

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

This downloads the latest SQL Server for Linux Docker image to your computer.

5. Launch the Docker Image

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

```
docker run -d --name sql_server_demo -e 'ACCEPT_EULA=Y' -e '
```

But of course, use your own name and password.

Here's an explanation of the parameters:

`-d`
: This optional parameter launches the Docker container in daemon mode. This means that it runs in the background and doesn't need its own Terminal window open. You can omit this parameter to have the container run in its own Terminal window.

`--name sql_server_demo`
: Another optional parameter. This parameter allows you to name the container. This can be handy when stopping and starting your container from the Terminal.

`-e 'ACCEPT_EULA=Y'`
: The `Y` shows that you agree with the EULA (End User Licence Agreement). This is required in order to have SQL Server for Linux run on your Mac.

`-e 'SA_PASSWORD=reallyStrongPwd123'`
: 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.

Password Strength

If you get the following error at this step, try again, but with a stronger password.

```
Microsoft(R) SQL Server(R) setup failed with error cod
```

I received this error when using `reallyStrongPwd` as the password (but of course, it's *not* a really strong password!). I was able to overcome this by adding some numbers to the end. However, if it wasn't just a demo I'd definitely make it stronger than a few dictionary words and numbers.

6. Check the Docker container (optional)

You can type the following command to check that the Docker container is running.

```
docker ps
```

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

```
STATUS              PORTS                     NAMES
Up 21 seconds       0.0.0.0:1433->1433/tcp    sql_server_demo
```

7. Install sql-cli (unless already installed)

Run the following command to install the sql-cli command line tool. This tool allows you to run queries and other commands against your SQL Server instance.

```
npm install -g sql-cli
```

This assumes you have NodeJs installed. If you don't, download it from Nodejs.org first. Installing NodeJs will automatically install *npm* which is what we use in this command to install sql-cli.

Permissions Error?

If you get an error, and part of it reads something like *"Please try running this command again as root/Administrator"*, try again, but this time prepend `sudo` to your command:

```
sudo npm install -g sql-cli
```

8. Connect to SQL Server

Now that sql-cli is installed, we can start working with SQL Server via the Terminal window on our Mac.

Connect to SQL Server using the `mssql` command, followed by the username and password parameters.

```
mssql -u sa -p reallyStrongPwd123
```

You should see something like this:

```
Connecting to localhost...done

sql-cli version 0.6.0
Enter ".help" for usage hints.
mssql>
```

This means you've successfully connected to your instance of SQL Server.

9. ## Run a Quick Test

Run a quick test to check that SQL Server is up and running and you can query it.

For example, you can run the following command to see which version of SQL Server your running:

```
 select @@version
```

If it's running, you should see something like this (but of course, this will depend on which version you're running):

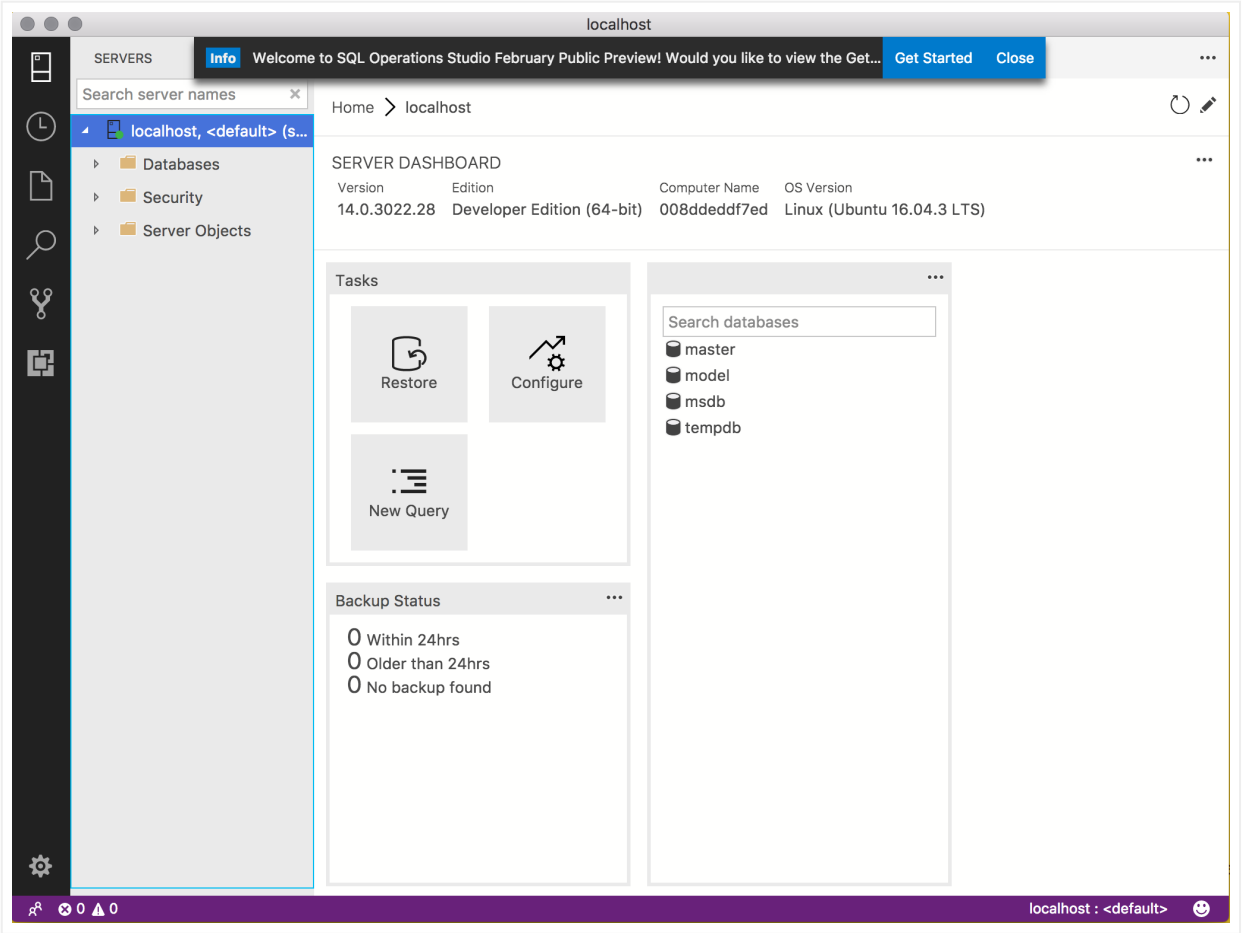
```
-------------------------------------------------------------
Microsoft SQL Server vNext (CTP2.0) - 14.0.500.272 (X64)
  Apr 13 2017 11:44:40
  Copyright (C) 2017 Microsoft Corporation. All rights reser
  Developer Edition (64-bit) on Linux (Ubuntu 16.04.2 LTS)

1 row(s) returned

Executed in 1 ms
mssql>
```

If you see a message like this, congratulations — SQL Server is now up and running on your Mac!

A SQL Server GUI for your Mac – SQL Operations Studio (SQLOPS)



SQL Operations Studio (SQLOPS) is a free GUI management tool that you can use to manage SQL Server on your Mac. You can use it to create and manage databases, write queries, backup and restore databases, and more.

SQLOPS is currently in "preview" mode and it's available on Windows, Mac and Linux.

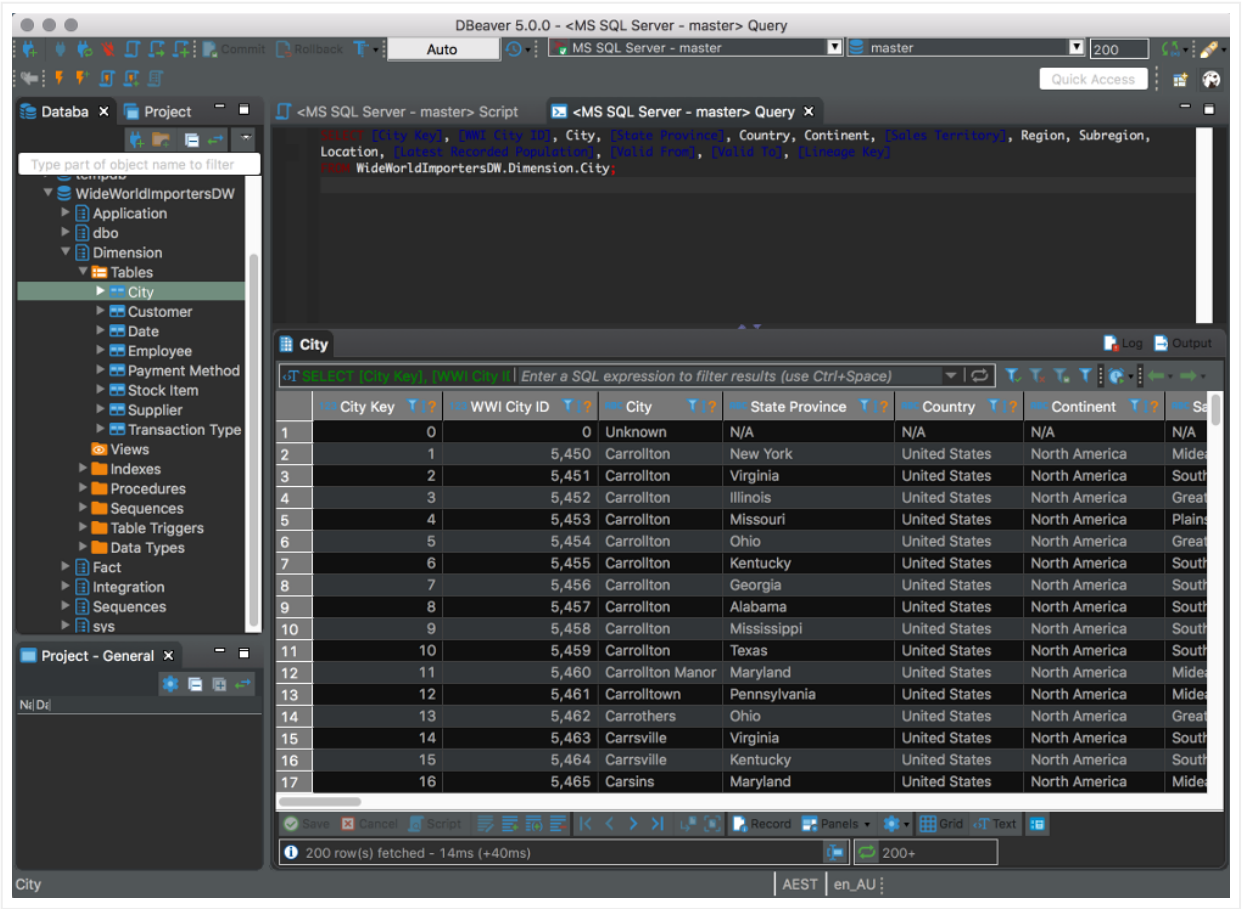
Here are some articles/tutorials I've written for SQLOPS:

- What is SQL Operations Studio (SQLOPS)
- How to install SQL Operations Studio on your Mac
- How to Create a Database with SQLOPS
- How to Restore a Database with SQLOPS on a Mac

Another Free SQL Server GUI – DBeaver

Another SQL Server GUI tool that you can use on your Mac (and Windows/Linux/Solaris) is DBeaver.

DBeaver is a free, open source database management tool that can be used on most database management systems (such as MySQL, PostgreSQL, MariaDB, SQLite, Oracle, DB2, SQL Server, Sybase, Microsoft Access, Teradata, Firebird, Derby, and more).



DBeaver using the "Dark" theme.

I wrote a little introduction to DBeaver, or you can go straight to the DBeaver download page and try it out with your new SQL Server installation.

Limitations of SQL Server for Linux/Mac

SQL Server 2017 for Linux does have some limitations (at least, in its initial release). The Linux release doesn't include many of the extra services that are available in the Windows release, such as Analysis Services, Reporting Services, etc. Here's a list of what's available and what's not on SQL Server 2017 for Linux.

Another limitation is that SQL Server Management Studio is not available on Mac or Linux. SSMS a full-blown GUI management for SQL Server, and it provides many more features than SQLOPS or DBeaver (at least at the time of writing). You can

still use SSMS on a Windows machine to connect to SQL Server on a Linux or Mac machine, but you just can't install it locally on the Linux or Mac machine.

If you need any of the features not supported in SQL Server for Linux, you'll need SQL Server for Windows. However, you can still run SQL Server for Windows on your Mac by using virtualization software. Here's how to install SQL Server for Windows on a Mac using VirtualBox.

SQL Server

MAC SQL SERVER

PREVIOUS POST
How to Export the results of a Query using MySQL Workbench

NEXT POST
RDBMS vs DBMS – What's the Difference?

ABOUT

About Database.Guide

TAGS

aggregate functions big data character set codd collation comparison functions conversion functions create database create form create query create relationship create table database diagram database schema database terms date functions dbms export how to json list mac macOS macros mathematical functions ms access

mssql mysql workbench nosql oracle rdbms relationships security sql SQL Operations Studio SQLOPS sql reference sql server stored procedures string functions t-sql tables tutorial views what is

RECENT POSTS