

Docker SQL Server Set Up

17th Oct 2018

Contents

[1. Introduction 3](#_Toc527544618)

[2. Dependencies 3](#_Toc527544619)

[3. Quick Start Guide 3](#_Toc527544620)

[3.1 – Opening your Command Prompt 3](#_Toc527544621)

[3.2 Checking docker is running 3](#_Toc527544622)

[3.3 Checking The Right Docker Image is Installed 5](#_Toc527544623)

[3.4 Starting The Container 5](#_Toc527544624)

[3.5 Connecting to the Container 6](#_Toc527544625)

# 1. Introduction

This document is to provide a step by step guide on getting started with using docker on a local machine and is focused on using an SQL image to speed up the rebuilding of environments for business courses.

There are two key sections one will be the quick start guide if docker is already installed on the machine and the full build guide if the machine does not have docker installed.

# 2. Dependencies

This document will not cover the installation of SQL platforms to engage with the database however the below links are areas for download regarding two of SQL applications that could be utilised:

* SQL Server Management Studio – this is the core Microsoft product for engaging with SQL Servers and the link is [here](https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-2017)
* SQL Operations Studio – this is a lighter weight install and focuses more on the pure engagement with running SQL commands and queries and the link can be found [here](https://docs.microsoft.com/en-us/sql/azure-data-studio/download?view=sql-server-2017)
  + This could be an option in the future to reduce the storage imprint on laptops

# 3. Quick Start Guide

This section of the guide considers that docker is already installed and accessible via the command prompt. If step 3.2 of this guide fails it means docker is not installed step straight into the full guide.

## 3.1 – Opening your Command Prompt

You will need to access the command line application on your windows machine. There are a couple of steps to open your command line application:

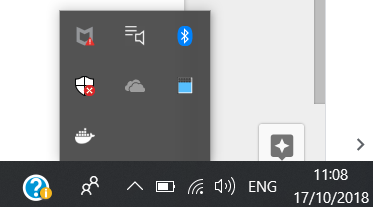
1. Easiest -> in the search bar in the bottom left click and type ‘*command’* and it should be one of the first applications to appear. **OR;**
2. Search for the command line app within your applications list from the windows app menu which you will find under the folder ‘*Windows System*’

## 3.2 Checking docker is running

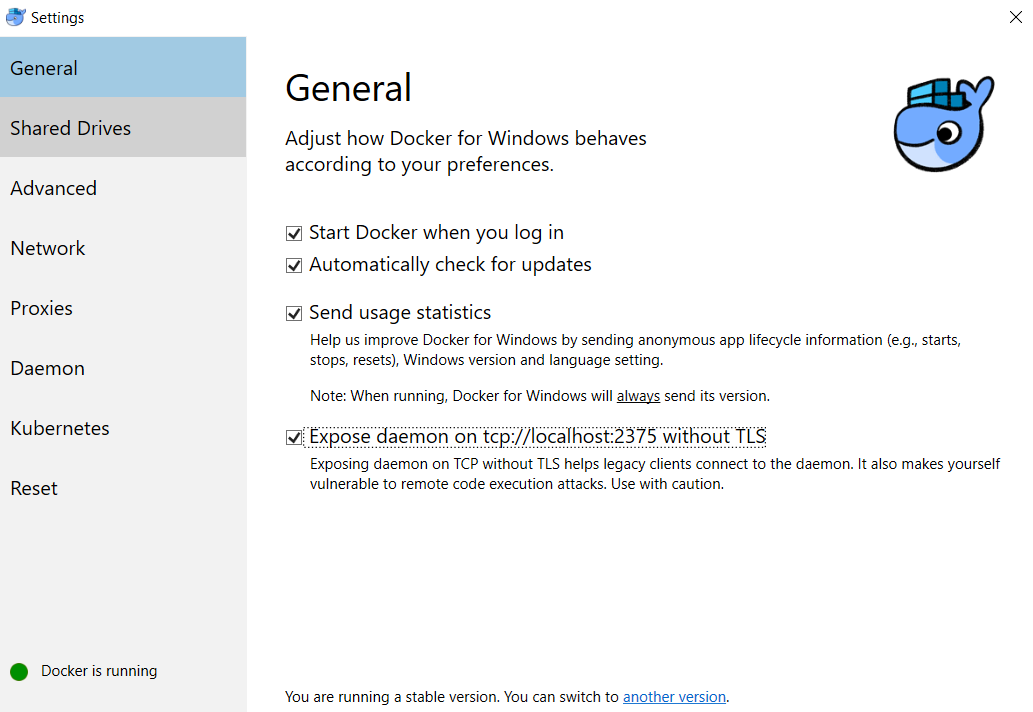
Docker will start up on starting the laptop and to see whether it is running visually you can check your task bar in the bottom right and click on the chevron which looks like below:



Once open you should see the Docker symbol  in the menu:



If the square blocks are ‘pulsing’ that means Docker is still starting up and you should wait for this to complete. To check further you can right click on the icon and select settings which will bring up the below screen:



Ensure that the Expose Daemon box is checked and you should see that docker is running in the bottom left corner.

If none of the above areas are running look for the Docker icon on your Desktop or within the applications area of the windows menu. If an application exists i.e. a blue whale, double click it and it should initiate Docker starting.

Finally, and most importantly if none of the above can be found or confirmed it is likely Docker is not installed and please move to the full installation menu.

## 3.3 Checking The Right Docker Image is Installed

You should have your command prompt open and to check things are working type the below command:

*docker -v*

you should see a docker version appear if not please review step 3.2.

next type:

*docker images*

You should see a list of images appear and the repository to look for is ‘*kcornwall/sqlnorthwind*’ If this is not there you will need to run the below command

*docker pull kcornwall/sqlnorthwind*

It will begin to download and wait for it to complete. Once completed run ‘*docker images*’ again and you should see ‘*kcornwall/sqlnorthwind*’

## 3.4 Starting The Container

Now we have the base image in place we can create and start our northwind container:

The command is:

*docker run -d --name sql -p 1433:1433 kcornwall/sqlnorthwind*

Below is a breakdown of what the command does:

* docker run -> When docker run is executed the container process that runs is isolated in that it has its own file system, its own networking, and its own isolated process tree separate from the host. In short we're asking to create the container.
* -d -> This runs the container in detached mode. Essentially it means the container will run in the background allowing you to use exposed resources locally (as an example) or attache to the container itself.
* --name sql1 -> Naming your container is good practice and the --name flag allows you to do this. This is not mandatory however as Docker will name containers automatically.
* -p 1401:1433 -> the p flag maps ports from your container to your localhost. In this instance we are mapping port 1401 from the container to port 1433 on our localhost. The standard SQL port is 1433 but in the container it has been set to 1401.
* kcornwall/sqlnorthwind -> This is the base image name to create the local container from.

Once you hit enter you should see a serial number appear.

If you now type the command ‘docker ps’ you should see the below information:

* Container ID -> This will be specific to your machine
* Image -> kcornwall/sqlnorthwind
* Command -> "/opt/mssql/bin/sqls…"
* Created -> 45 seconds ago
* Status -> Up 44 seconds
* Ports -> 0.0.0.0:1433->1433/tcp
* Names -> sql

This shows that our SQL Northwind contain is up and running.

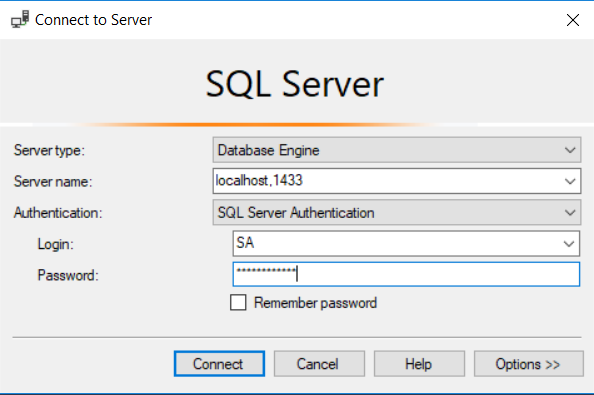
## 3.5 Connecting to the Container

Open your SSMS or SQL operations studio (they should be similar but the images used below are from SSMS)

When met with the connect screen input the below details (Please note the details will always be the same!)

* **Server name** -> localhost,1433 NOTE -> You will need to overwrite this manually and type it in and it is a comma in between localhost and 1433
* **Authentication** -> Ensure this is set to SQL Server Authentication
* **Login** -> SA (Caps is necessary)
* **Password** -> Passw0rd2018

NOTE – please check the image below



You should then be connected, if you run any SQL query against the Northwind DB you should see a response.

# 4. Managing containers

## 4.1 Stopping and Starting Containers

You would have already been using `*docker ps*` which will show the state of the container. If you run `*docker ps*` and see nothing it could be due to the fact the container may have stopped. In this case you can run:

*docker ps -al*

This will bring up all containers historically and those that may be shut down. Therefore, if you find your ‘sql’ container you will be able to start is again by using:

*docker start sql*

To stop the container will be…. You guessed it:

*docker stop sql*

## 4.2 destroying your Container and Starting Again

There will be an instance where you will need to destroy your container because you’ve deleted something you need or added to much you don’t have to work out where it went wrong.

Ensure that your container is stopped by running

*docker stop sql*

You should then check that the container has been stopped and can be seen using the below command.

*docker ps -al*

Once stopped we then need to remove the container, in this instance the container is ‘sql’ so we will use this container in the example:

*docker rm sql*

And that’s it your container is deleted! **NOTE!!!! -> anything you had on this container is now gone!**

**To rebuild the container**, it would be a case of running the below command from earlier in this document:

*docker run -d --name sql -p 1433:1433 kcornwall/sqlnorthwind*

This will start up a new container with the Northwind DB ready to go.