# Week 11 Lab 2 Databases 3 Practical:

# Containerised SQL Server

Goal 1: Get local container working, visible to SQL Server Management Studio.

Goal 2: Get hosted container working, visible to SQL Server Management Studio.

Goal 3: Sync container with Docker Hub

(Optional deeper learning:)

Docker basics: <https://docs.docker.com/get-started/>

(Quickstart for useful references and jumping off point:)

<https://docs.microsoft.com/en-us/sql/linux/quickstart-install-connect-docker?view=sql-server-2017>

Docker ToolBox

<https://docs.docker.com/toolbox/toolbox_install_windows/>

<https://docs.docker.com/docker-for-mac/install/>

> docker --help

> docker container --help

> docker container ls --help

> docker run --help

ONE:

## Local Install – can be replicated at home:

* Fire up Docker Quickstart Terminal
* Remove the default virtual machine and make one with more resources:
  + docker-machine rm default
  + docker-machine create -d virtualbox --virtualbox-cpu-count=2 \  
    --virtualbox-memory=4096 --virtualbox-disk-size=50000 default

Restart the terminal

* Hello World working?
  + docker run hello-world
* Make sure you have noted the IP address of the virtual machine running the docker container
* Get the SQL Server image from the Microsoft Container Registry
  + docker pull mcr.microsoft.com/mssql/server:2017-latest
* Start SQL Server
  + docker run -e 'ACCEPT\_EULA=Y' -e 'SA\_PASSWORD=P@ssw0rd' \

-p 1433:1433 --name sql1 \

-d mcr.microsoft.com/mssql/server:2017-latest

* CHECK: Is the image running?
* docker ps
* Is SQL server in good health?
  + docker logs sql1
* Can you connect via SSMS (using the IP address noted before)
  + Run SSMS on the same host that’s running the virtual machine
  + Use SQL Server authentication, username SA, password as set on the command line when you started the server

TWO:

Remote Install – It is worthwhile reading some additional documentation:

<https://docs.microsoft.com/en-us/sql/linux/sql-server-linux-configure-docker?view=sql-server-2017>

* SSH to db3docker.ict.op.ac.nz
* Authenticate using YourUserName / StudentID
* Hello World working?

docker run hello-world

* Get the SQL Server image from the Microsoft Container Registry

docker pull mcr.microsoft.com/mssql/server:2017-latest

* Start SQL Server

We are running multiple containers on the same server (see documentation above), Docker provides a way to run multiple SQL Server containers on the same host machine. Therefor each container must expose itself on a different port.

You will need to use your allocated portNumber (see Teams)

Do not forget your password or you will have to run setup from the beginning

docker run -e 'ACCEPT\_EULA=Y' -e 'SA\_PASSWORD=***YourSecurePassword***'

-p ***portNumber***:*1433* --name *username*

-d mcr.microsoft.com/mssql/server:2017-latest

* CHECK: Is the image running?

docker ps

* Is SQL server in good health?

docker logs *username*

* If You need to remove the container, first find its container ID

docker ps –a

docker rm <containerid>

If it is already running you will need to stop it first

docker stop <containerid>

* Can you connect via SSMS (using your allocated port number , SSMS uses the format <*ServerName,port#>* or <*ip-addy,port#*>

Use SQL Server authentication, username SA, password as set on the commandline when you started the server

THREE

Docker Hub

* Create an account on Docker Hub – cloud.docker.com Read the documentation: [https://docs.docker.com](https://docs.docker.com/)
* Back in your DB3Docker container – find the id of your image

Docker image ls

* Tag your image with an appropriate label

docker tag imageid dockerHubUserName/dockerHubRepositoryName:tag

*ex:* docker tag cfe5615bf6a8 krissiw/in705db3test:servertest

* Push your image to Docker Hub

Docker push imageid dockerHubUserName/dockerHubRepositoryName:tag

* You may need to supply Docker Hub credentials

Docker login docker.io