Databases 3 Practical:

Containerised SQL Server

Goal 1: Get SQL Server launched in Docker, visible to SQL Server Management Studio.

Goal 2: Import user data and write T-SQL scripts to set up Logins, Users, and Permissions.

Goal 3: Successfully run through a backup and restore cycle.

Stretch Goal: Demonstrate a High Availability Configuration using containerised SQL Server

ONE:

(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>

Walkthrough:

* 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 issue command-line SQL from inside the container?
  + docker exec –it sql1 bash
  + (Explore the container as you would any Unix machine; cd, ls etc)
  + /opt/mssql-tools/bin/sqlcmd –S localhost –U SA
  + …Issue SQL commands followed by GO – e.g.
    - CREATE DATABASE mytestdb
    - GO
    - SELECT \* FROM INFORMATION\_SCHEMA.TABLES
    - GO
* 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 commandline when you started the server