IT/Dev CONNECTIONS

Building Your First SQL Server Container Lab in Docker

Chris Bell
Founder
WaterOx Consulting, Inc

Delivered by KNect365 TMT
an Informa business

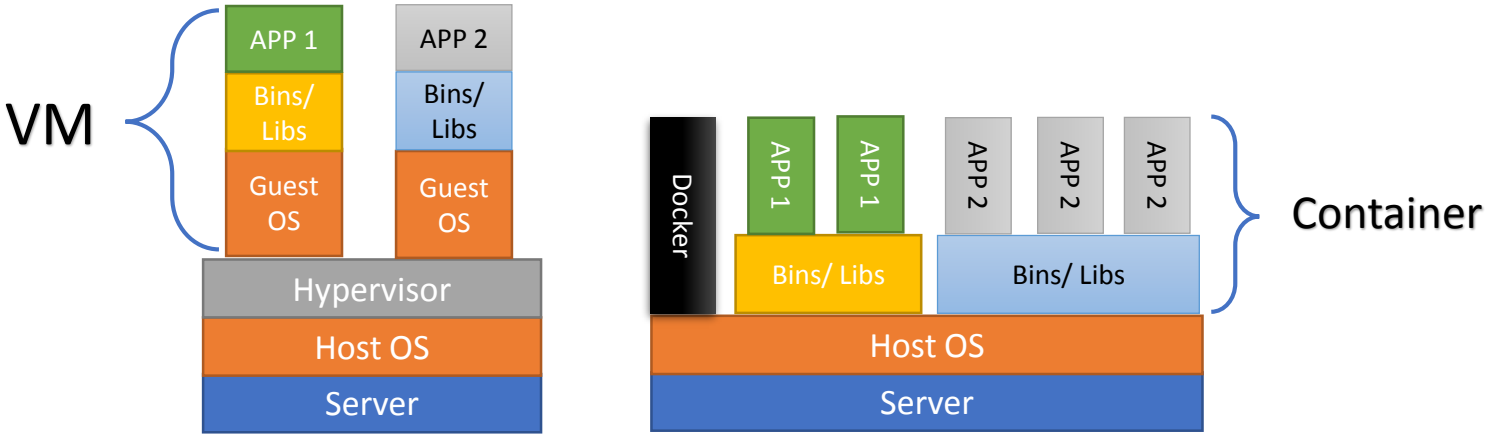
What is Docker?

- Opensource Technology
- Allows the packaging of all parts an application needs into one package
- Like a virtual machine without a virtual OS
- “Cross-platform”
  - Since contains all components needed in the package

What are Containers?

- Method of OS virtualization
- Pack code and dependencies into a container that can run ‘anywhere’
- Like a VM
  - Without the OS

Ok… So how are they different from a VM?



What We Need to Do

1. Get Docker installed & working
2. Get images to install
3. Create a SQL Server container
4. Interact with SQL Server
5. Advanced Settings - Static IP
6. Advanced Settings - Dockerfile
7. Advanced Settings - Data Volumes
8. Advanced Settings – SQL Backup & Restore

Get Docker Installed

- Powershell
  - Run as an admin
- Install-Module DockerMsftProvider – Force
- Install-Package Docker –ProviderName DockerMsftProvider – Force
  - Downloads and installs latest Docker

- Start Docker
  - Mine is configured for Windows containers

- Check version
  - Docker version

Make sure Docker Works

- Hello World!
  - Docker container run hello-world:nanoserver

Get Docker SQL Server Image

- Find SQL Server image on Dockerhub
  - Login first
- Docker search Microsoft/MSSQL
  - Looking for developer edition
  - Microsoft/mssql-server-windows-developer (SQL 2017)
- Docker pull Microsoft/mssql-server-windows-developer
  - These are not tiny files (~15 GB)
- When done check local repository with “Docker images”

Creating a SQL Container

- Docker run --name WOXDOCDEMO -d -p 14331:1433 -e sa_password=s@12345 -e ACCEPT_EULA=Y microsoft/mssql-server-windows-developer
- --name
  - Name of container
- -d
  - Run detached (like a service)
- -p
  - Port mapping (local : Container internal)
- -e
  - Environment Variables
  - Configuration values
- Image
  - Name of image to use to create the container

Checking Container Status

- Docker ps
  - List all running containers
- Docker ps –a
  - List all containers available on host
- Docker Inspect <container name>
  - Generate JSON of container details

Get Container IP Address

- IP address changes on each restart
- Get current IP with
  - $Container_1 = docker inspect WOXDOCDEMO | ConvertFrom-Json | select -ExpandProperty SyncRoot | select -expand netWorkSettings | select -ExpandProperty Networks | Select -ExpandProperty nat | Select IPAddress $Container_1

Interact with SQL in the Container

- Connect SSMS via IP address
- Connect SSMS via local & port number

- Works like normal SQL Server
  - Mostly
  - Can’t start/stop/restart the service
  - Must use powershell to stop the container

- Docker stop <container name>

Advanced Settings – Static IP

- Need to make a docker network first
  - Declare subnet range and gateways
- Docker run as before
  - Add ip address specification
  - Docker run --name WOXDOCDEMO --network="WoxDemo" --ip=192.168.10.2 -d -e sa_password=s@12345 -e ACCEPT_EULA=Y microsoft/mssql-server-windows-developer

Advanced Settings – Dockerfile

- Dockerfile
  - Text document
  - contains commands to assemble an image
  - Command-line instructions
- Can be used to attach DB files on startup

Advanced Settings – Data Volumes

- Data persists through Docker start & stop
  - Docker rm deletes everything
- Data volumes persists even after a container deletion
- Option 1: Mount host directory
  - Docker run and add –v <host directory>:/var/opt/mssql
- Option 2: Mount data volume container
  - Docker run and add –v <volume name>:/var/opt/mssql
- Docker volume ls
  - Docker volume rm to delete (permanently)

Advanced settings – SQL Backup & Restore

- Mount a volume for backups
  - Mounted volumes are accessible by both host and container
- Map with the Docker run –v <volume name> when starting container
- Restore like any other SQL Server instance
  - GUI or script


@CBELLDBA
Chris@WaterOxConsulting.com
WaterOxConsulting.com