Demo Script: Containers for Windows Devs

Michael S. Collier

[michael.collier@microsoft.com](mailto:michael.collier@microsoft.com) | <http://www.michaelscollier.com> | @MichaelCollier

# Deck

[insert path]

# Demo environment setup & prereqs

* Launch Hyper-V Manager
* Get to Windows Container support in Docker.

**& 'C:\Program Files\Docker\Docker\DockerCli.exe' -SwitchDaemon**

* Load VS2017

**C:\Projects\CodeMash2017\Containers for Windows Devs\HelloWorldConsole\HelloWorldConsole.sln**

* Launch SQL Server Management Studio

# Demo Reset

**docker rm -f $(docker ps -a -q)**

## Cache Images

**docker pull microsoft/windowsservercore**

**docker pull microsoft/nanoserver:latest**

**docker pull microsoft/dotnet:nanoserver**

**docker pull microsoft/dotnet-samples:dotnetapp-nanoserver**

**docker pull microsoft/iis**

**docker pull microsoft/mssql-server-windows**

# Demo Projects

## #1 - Show Docker running on Windows 10

In this first demo, we’ll show Docker running on Windows 10 (installed via Docker for Windows). We’ll also show the ability to switch the Docker daemon to flip between running Windows containers and Linux containers. Finally, we’ll show the MobyLinuxVM that is installed to enable us to run Linux containers.

|  |  |
| --- | --- |
| **docker version** | Show the Client and Server version for Docker.  Show the Server OS/Arch setting – either ‘windows/amd64’ or ‘linux/amd64’. |
| **& 'C:\Program Files\Docker\Docker\DockerCli.exe' -SwitchDaemon** |  |
| **docker version** | Show the Server OS/Arch setting has changed. |
|  |  |
| **Switch to Hyper-V Manager** | Show the ‘MobyLinuxVM’ running. |
|  |  |

## #2 – Manage Images

Let’s take a quick look at a few commands to view, pull, and remove images.

|  |  |
| --- | --- |
| **docker images** | List images in the local repository. |
|  |  |
| **docker pull microsoft/dotnet-samples:dotnetapp-nanoserver** | Get an image from Docker Hub. |
|  |  |
| **docker rmi microsoft/iis** | Remove a local image |
|  |  |

## #3 – Manage Containers

Let’s take a quick look at a few commands to run, start, and stop Windows containers.

|  |  |
| --- | --- |
| **docker run microsoft/dotnet-samples:dotnetapp-nanoserver** | Run a container |
|  |  |
| **docker ps** | List running containers |
|  |  |
| **docker ps -a** | List all containers, including stopped containers |
|  |  |
| **docker stop <container\_id>** | Stop the container |
|  |  |
| **docker start -ia <container\_id>** | Run an existing, stopped |
|  |  |

## #4 – Dockerize a .NET Console Application

In this demo, we’ll show how to take a simplistic .NET 4.5.2 console application and ‘Dockerize’ it; create a docker image from the application. We’ll then run that newly created docker image as a container.

|  |  |
| --- | --- |
| Show the code in HelloWorldConsole/Program.cs | Simple .NET 4.5.2. console application.  Readline() just to keep the console application open. |
|  |  |
| Create the Dockerfile / Add the existing Dockerfile  FROM microsoft/windowsservercore  ADD publish/ /  ENTRYPOINT HelloWorldConsole.exe | The base image used for a console .NET Framework application is microsoft/windowsservercore, publicly available on Docker Hub.  The base image contains a minimal installation of Windows Server 2016, .NET Framework 4.6.2 and serves as the base OS image for Windows Containers.  Add the content in the ‘publish/’ directory.  There is a post-build event added to the project to copy the necessary files.  The Entrypoint for the container is the project executable. |
|  |  |
| **cd 'C:\projects\CodeMash2017\Containers for Windows Devs\HelloWorldConsole\HelloWorldConsole\'** | Move to the project directory. |
|  |  |
| **docker build -t helloworldconsole:codemash2017 .** | Create the image  *(don’t forget the “.”)* |
|  |  |
| **docker images** | Show the image just created. |
|  |  |
| **docker run --rm helloworldconsole:codemash2017 “mike”** | Run the image, specifying the input parameter.  Using “--rm” will remove the container after it exits. |
|  |  |
| **docker ps -a** | Show there are no containers |

## #5 – SQL Server vNext in a Container

In this demo, we’ll look at running the next version of SQL Server in a Windows container (via Docker). As we do this, we’ll make a small modification to the container, create an image, and push the image to Azure Container Registry.

|  |  |
| --- | --- |
| Run through the script located at "C:\Projects\CodeMash2017\Containers for Windows Devs\SQL vNext Containers Demo.ps1" |  |
|  |  |

## #6 – Visual Studio 2017 (Docker Tools)

Todo

|  |  |
| --- | --- |
|  |  |
|  |  |