What you Need to Know about SQL Server and Docker Containers

Michael Otey, President, TECA, Inc
Moderated By: Donna Johnson

Immerse yourself

in the data community

Access deep-dive technical sessions, learn best practices, and discover new tips and tricks

Gain the technical skills and connections to advance your data career

Attend

PASS Summit

PASS

SUMMIT2018

NOV 6-9 | SEATTLE WA

PASS Summit is the largest conference for technical professionals who leverage the Microsoft Data Platform.

See everything PASS Summit has to offer at PASSsummit.com

More info:

Michael Otey,

President, TECA, Inc

https://www.linkedin.com/in/michaelotey

@michael_otey

IT industry for 25+ years

Helped start Windows IT Pro and SQL Server Pro

Former SQL Server MVP

Dozens of technical books & eLearning courses

Windows IT Pro and SQL Server Pro articles, labs and product reviews

IT Pro Today Senior Contributing Editor

What you Need to Know about SQL Server and Docker Containers

Michael Otey, President, TECA, Inc
Moderated By: Donna Johnson

WHAT THIS SESSION COVERS

- Container and Docker basics
  - Container platforms
  - Containers vs. VMs
  - Docker types
    - Docker for Windows Containers vs Docker for Windows
  - Running your first container
- SQL Server in Containers
  - Running SQL Server in Docker
  - Using volumes for data persistence
  - Building custom images
  - Creating container DB startup scripts

WHY CONTAINERS?

- Lighter weight than VMs
- Fast deployment
  - No need for SQL Server installation
- Portability
- Stateless
- New microservice style applications

CONTAINER PLATFORMS FOR SQL SERVER

- Windows Server 2016
- Windows 10
- Red Hat Enterprise Linux 7.2
- SUSE Enterprise Linux v12 SP2
- Ubuntu 16.04
- Other Linux distributions?

Windows Server

Windows 10

redhat

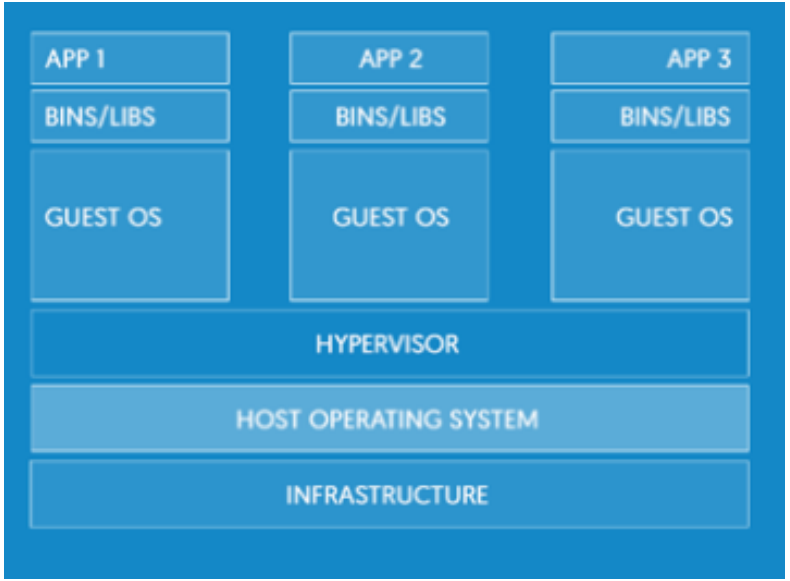
SUSE

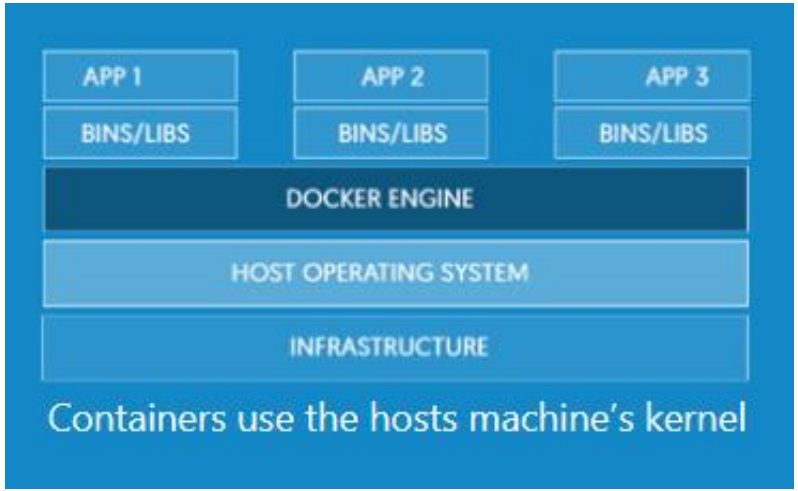
ubuntu

DOCKER CONCEPTS

- What is a Docker?
  - Multi-platform container engine that runs on Linux, Windows and Mac
- Container
  - A executing environment that encapsulates code, binaries, files and OS dependencies
  - Used to run applications
- Image
  - Template used for creating containers

VMS AND CONTAINERS



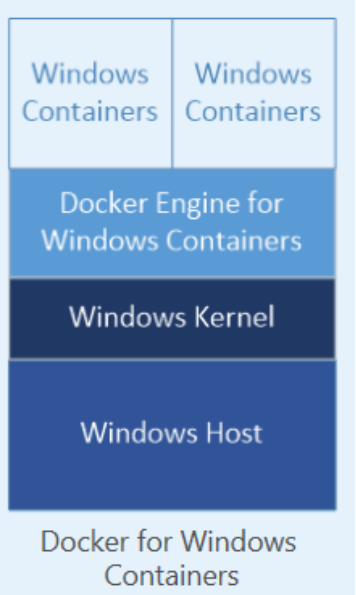


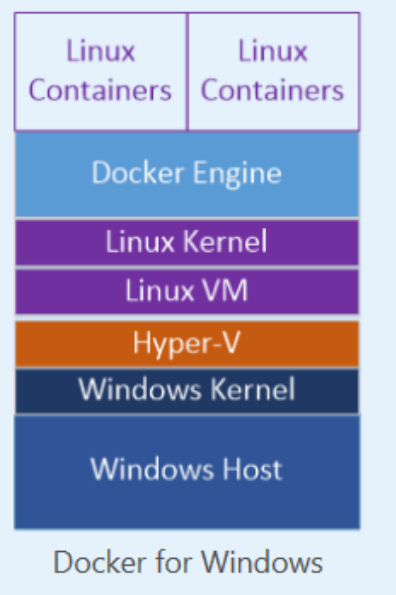
DOCKER TYPES

- Docker Engine
  - Installs on Linux and run Linux containers
  - Uses Linux OS
- Docker for Windows Containers
  - Installs on Windows and runs Windows containers
  - Uses Windows OS
- Docker for Mac and Windows
  - Installs a Linux VM on Mac or Windows
  - Uses Linux VM OS
  - Runs Linux containers

DOCKER FOR WINDOWS AND MAC

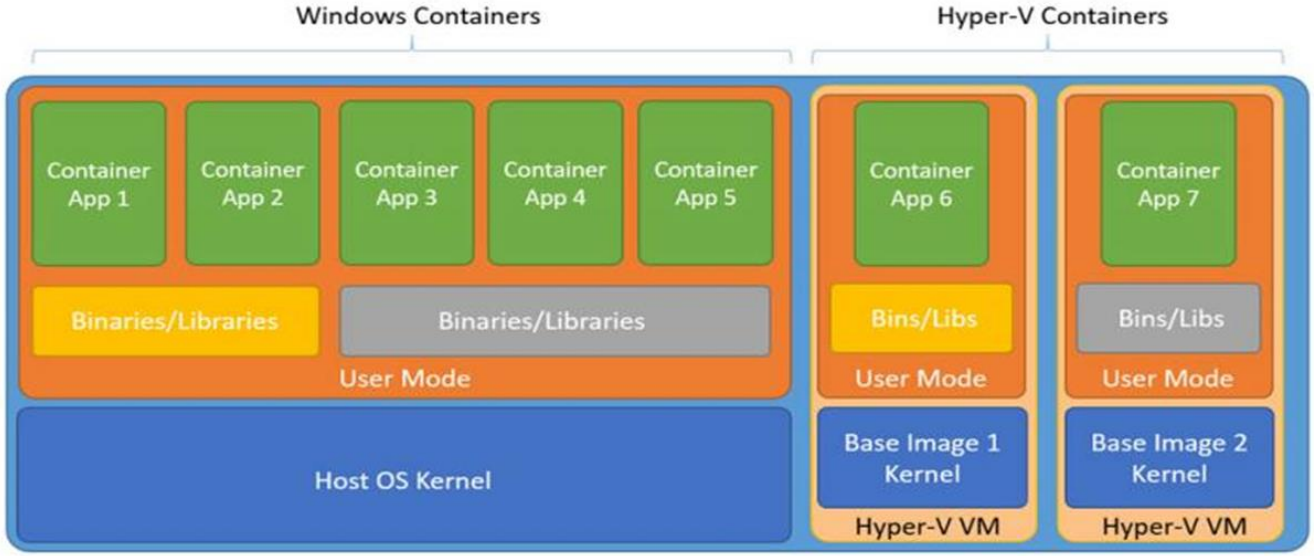
- Designed to run Linux VMs





WINDOWS CONTAINERS & HYPER-V CONTAINERS

- Each Hyper-V container has its own lightweight VM



WINDOWS CONTAINERS & HYPER-V CONTAINERS

WS Containers

- Shares host OS
- Does not require virtualization
- Less overhead
- Less secure
- Can run on Nano
- Can not run Linux

Hyper-V Containers

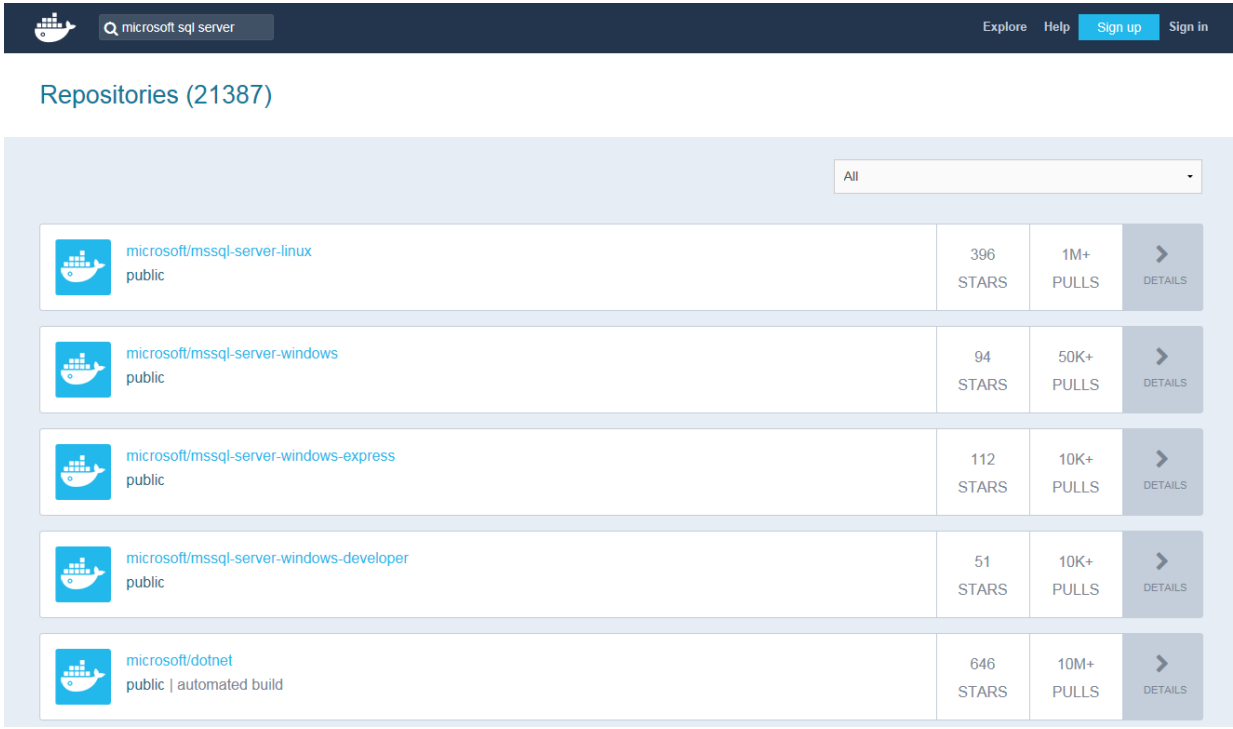
- Isolated from host OS
- Requires virtualization
- More overhead
- More secure
- Can run on Nano
- Can run Linux (Win10 WS1709)

WINDOWS CONTAINERS & HYPER-V CONTAINERS

- Running SQL Server Express in a Windows container
  - docker run -d -p 1433:1433 —env sa_password=<YOUR_PWD> microsoft/mssql-server-express-windows
- Running SQL Server Express in a Hyper-V container
  - docker run -d -p 1433:1433 —env sa_password=<YOUR_PWD> —isolation=hyperv microsoft/mssql-server-express-windows

DOCKER HUB

- Collection of images
  - Free repository
  - Most can be readily used
  - Posted by
    - Organizations
      - Microsoft
    - Users
  - You can keep your own local repositories



PASS MARATHON

MICROSOFT SQL SERVER DOCKER IMAGES

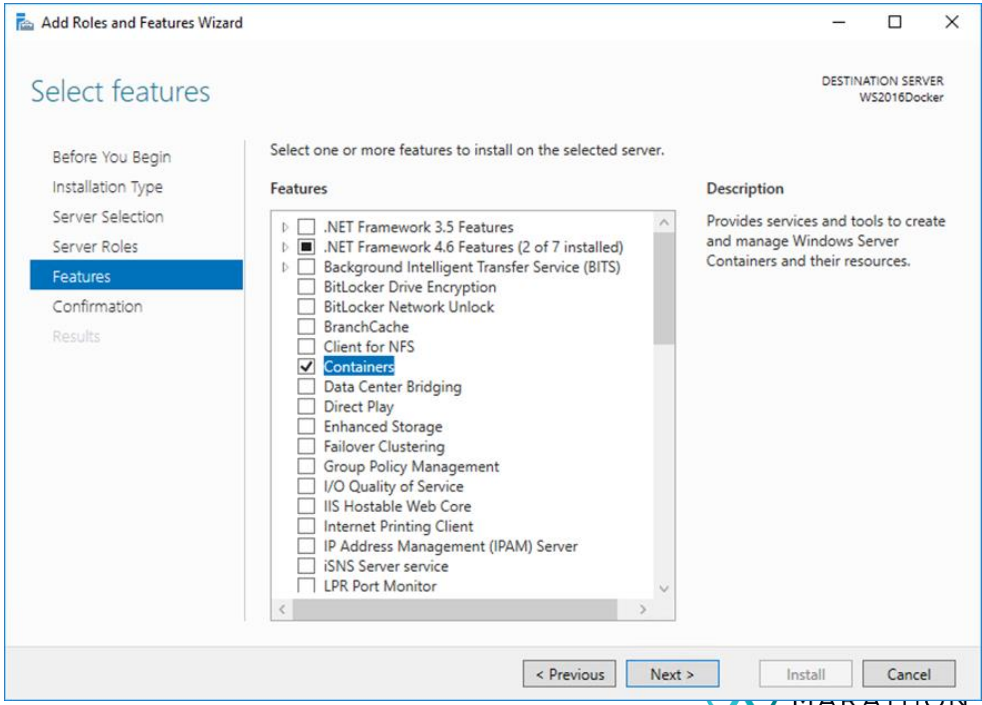
- microsoft/mssql-server-linux
  - docker pull microsoft/mssql-server-linux
  - Tags: latest, 2017-latest,2017-GA
- microsoft/mssql-server-windows-express
  - docker pull microsoft/mssql-server-windows-express
  - Tags: 2017, latest
- microsoft/mssql-server-windows-developer
  - docker pull microsoft/mssql-server-windows-developer
  - Tags: 2017, latest

PASS MARATHON

DOCKER BASICS

- Installing Container Support on Windows
  - Server Manager
  - Add Features – Containers
- Installing Docker
  - Elevated PowerShell Prompt

```
Install-Module -Name DockerMsftProvider
 -Repository PSGallery -Force
Install-Package -Name docker
 -ProviderName DockerMsftProvider
```



INSTALLING DOCKER SUPPORT ON LINUX - UBUNTU

- Add the GPG key for the Docker
  - curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add –
- Add the Docker repository to APT sources
  - sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
- Update the package database with the Docker packages
  - sudo apt-get update
- Install Docker
  - sudo apt-get install -y docker-ce
- Check the Docker daemon status
  - sudo systemctl status docker

```
administrator@UbuntuDocker:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: e
   Active: active (running) since Wed 2017-10-18 19:55:08 PDT; 56s ago
     Docs: https://docs.docker.com
 Main PID: 7918 (dockerd)
   CGroup: /system.slice/docker.service
           ├─7918 /usr/bin/dockerd -H fd://
           └─7925 docker-containerd -l unix:///var/run/docker/libcontainerd/dock

Oct 18 19:55:07 UbuntuDocker dockerd[7918]: time="2017-10-18T19:55:07.954386622-
Oct 18 19:55:07 UbuntuDocker dockerd[7918]: time="2017-10-18T19:55:07.954506922-
Oct 18 19:55:07 UbuntuDocker dockerd[7918]: time="2017-10-18T19:55:07.954919321-
Oct 18 19:55:08 UbuntuDocker dockerd[7918]: time="2017-10-18T19:55:08.222622393-
Oct 18 19:55:08 UbuntuDocker dockerd[7918]: time="2017-10-18T19:55:08.374140552-
Oct 18 19:55:08 UbuntuDocker dockerd[7918]: time="2017-10-18T19:55:08.397605915-
Oct 18 19:55:08 UbuntuDocker dockerd[7918]: time="2017-10-18T19:55:08.402435107-
Oct 18 19:55:08 UbuntuDocker dockerd[7918]: time="2017-10-18T19:55:08.402744307-
Oct 18 19:55:08 UbuntuDocker systemd[1]: Started Docker Application Container En
Oct 18 19:55:08 UbuntuDocker dockerd[7918]: time="2017-10-18T19:55:08.456353822-
lines 1-19/19 (END)
```

INSTALLING DOCKER SUPPORT ON LINUX - UBUNTU

- Executing Docker Without sudo (optional)
- Add your username to the docker group
  - sudo usermod -aG docker ${USER}
- Apply the new group membership
  - su - ${USER}
- Verify installation
  - Docker
  - Docker version

BASIC DOCKER TASKS

- Pulling an image from the Docker Hub
  - docker pull hello-world
- Create a container
  - docker run hello-world
- Show images
  - docker images
- Show running containers
  - docker ps

```
administrator@UbuntuDocker: ~
.41GB
administrator@UbuntuDocker:~$ docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://cloud.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/engine/userguide/

administrator@UbuntuDocker:~$
```

DEMO: DOCKER BASIS

PASS MARATHON

SQL SERVER CONTAINERS

- Your first SQL Server on Windows container

  > docker pull microsoft/mssql-server-windows-express

  > docker images

  > docker run –e ACCEPT_EULA=Y –e SA_PASSWORD=SQLpwd2017 –p 1433:1433 –d microsoft/mssql-server-windows-express

- Your first SQL Server on Linux container

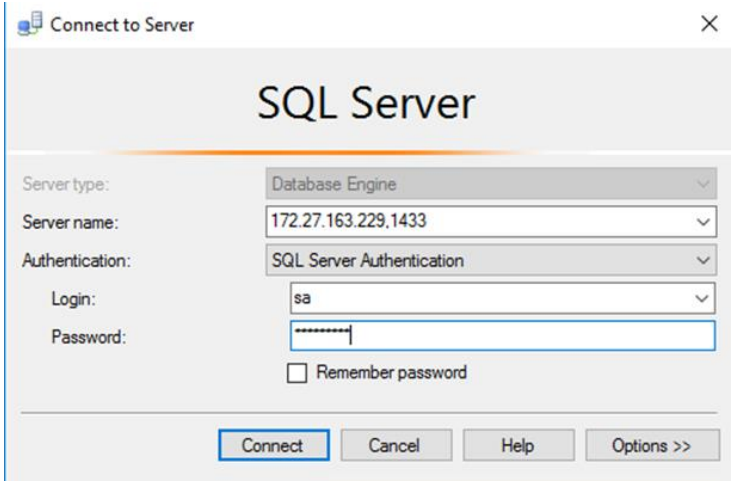
  $ docker pull microsoft/mssql-server-linux

  $ docker images

  $ docker run –e ACCEPT_EULA=Y –e SA_PASSWORD=SQLpwd2017 –p 1433:1433 –d microsoft/mssql-server-linux

CONNECTING TO A SQL SERVER CONTAINER

- SQLCMD
  - Server or IP, username sa, password mySQLPWD
- SSMS
  - Server or IP,port
- Finding the container IP address
  - Docker ps
  - docker inspect -format='{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' <containerid>



SQLCMD CONNECTED TO DOCKER INSTANCE

- docker exec -it 5e9 sqlcmd -S.

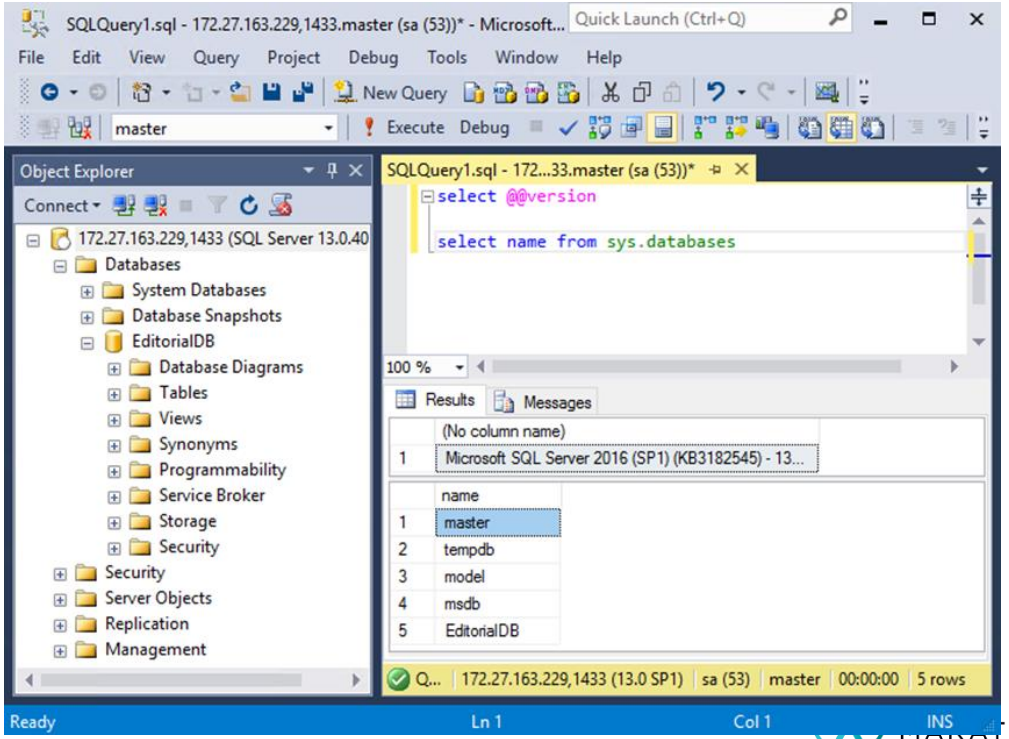
Administrator: Windows PowerShell

```
1> select @@version
2> go


-----------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------
Microsoft SQL Server 2016 (RTM) - 13.0.1601.5 (X64)
        Apr 29 2016 23:23:58
        Copyright (c) Microsoft Corporation
        Express Edition (64-bit) on Windows Server 2016 Datacenter 6.3 <X64> (Build 14393: ) (Hypervisor)


(1 rows affected)
1>
```

PASS MARATHON

SSMS CONNECTED TO A SQL SERVER CONTAINER

- Just like Windows or Linux
- DBs can be created in the container
- DBs can be attached with volumes



MANAGING A SQL SERVER CONTAINER

- Show running containers and get IDs
  - **> docker ps**
- Stop a container
  - **> docker stop <containerid>**
- Start container
  - **> docker start <containerid>**
- Remove a container
  - **> docker rm <containerid>**
- Show container logs
  - **> docker logs <containerid>**
- Kill/delete
  - **>docker kill <containerid>**

Administrator: Windows PowerShell

```
PS C:\Users\Administrator> docker ps
CONTAINER ID        IMAGE                                       COMMAND
              NAMES
6da7b665d780        microsoft/mssql-server-windows-express   "powershell -Comma..."
1->1433/tcp   infallible_wing
PS C:\Users\Administrator> docker logs 6d
VERBOSE: Starting SQL Server
VERBOSE: Changing SA login credentials
VERBOSE: Attaching 1 database(s)
VERBOSE: Invoke-Sqlcmd -Query IF EXISTS (SELECT 1 FROM SYS.DATABASES WHERE NAME
 = 'EditorialDb') BEGIN EXEC sp_detach_db [EditorialDb] END;CREATE DATABASE
[EditorialDb] ON (FILENAME = N'C:\temp\EditorialDB.mdf'),(FILENAME =
N'C:\temp\EditorialDB_log.ldf') FOR ATTACH;
VERBOSE: Started SQL Server.

TimeGenerated           EntryType Message
-------------           --------- -------
10/23/2017 1:30:45 PM Information Parallel redo is shutdown for database 'Ed...
10/23/2017 1:30:45 PM Information Parallel redo is started for database 'Edi...
10/23/2017 1:30:45 PM Information Starting up database 'EditorialDb'.
10/23/2017 1:35:57 PM Information Parallel redo is shutdown for database 'Ed...
10/23/2017 1:35:57 PM Information Parallel redo is started for database 'Edi...
10/23/2017 1:35:57 PM Information Starting up database 'EditorialDb'.
10/23/2017 1:35:57 PM Information Using 'xplog70.dll' version '2017.140.1000...
10/23/2017 1:35:57 PM Information Attempting to load library 'xplog70.dll' i...
PS C:\Users\Administrator>
```

PASS MARATHON

MANAGING IMAGES

- Creating a new image from a running container
  - >docker commit [containerID] [respository name]
- List images
  - > docker images
- Remove/delete an image
  - > docker rmi <imageid>

Administrator: Windows PowerShell

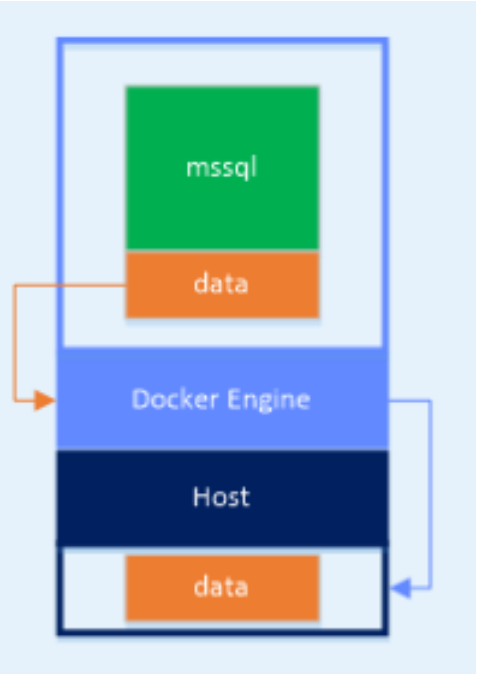
```
PS C:\Users\Administrator> docker images
REPOSITORY                                TAG                 IMAGE ID            CREATED
microsoft/mssql-server-windows-express    latest              60a5c1c6ba00        9 days ago
teca/sqlbuild                             latest              902ac3cb5035        5 months ago
sqlexpress                                latest              887012a3e2b5        5 months ago
sqlstandard                               latest              f258a7ffe182        5 months ago
teca/sqlstd1                              latest              c9488b1fd95f        7 months ago
microsoft/mssql-server-windows-developer  latest              2707b3c81de2        8 months ago
microsoft/dotnet-samples                  dotnetapp-nanoserver a331d851e765       8 months ago
microsoft/mssql-server-windows            latest              6fb5a1dbd3c8        9 months ago
microsoft/aspnet                          latest              e761eca2f8df        9 months ago
microsoft/windowsservercore               latest              4d83c32ad497        9 months ago
microsoft/nanoserver                      latest              d9bccb9d4cac        9 months ago
microsoft/mssql-server-windows-express    <none>              d5cc84b603b4        11 months ago
PS C:\Users\Administrator>
```

DEMO:
Getting started with Docker and SQL Server

PASS MARATHON

CONTAINERS AND DATA

- Containers are transient
  - When they are deleted all data in the container is also deleted
    - All data changes are lost
  - When they are restarted the image is fresh from the container
    - Test images can container ready-to-go DBs
- Map volumes from the host to the container
  - Use the –v option (No Mac support)
  - docker run [ContainerInfo] –v <host dir>:<container dir> [Image]



USING VOLUMES WITH SQL SERVER

- docker run -d -p 1433:1433 -e sa_password=SQLpwd2017 -e ACCEPT_EULA=Y
  -v C:/temp/:C:/temp/
  -e attach_dbs="[{'dbName':'EditorialDb','dbFiles':['C:\\temp\\editorialdb.mdf’, 'C:\\temp\\ editorialdb_log. ldf']}]"
  microsoft/mssql-server-windows-express
- Not available for Linux

DEMO:
Managing SQL Server Containers and Volumes

PASS MARATHON

BUILDING A CUSTOM IMAGE

- Start with Windows Server Core image
  - docker run -d -ti --name SQLStd –hostname SQLStd -v C:/SQLinstall/:C:/SQLinstall/ microsoft/windowsservercore
- Perform a command line SQL Server installation
  - docker exec -it SQLStd cmd
  - `Setup.exe /Q /ACTION=Install /FEATURES=SQL,IS /INSTANCENAME=MSSQLSERVER /SQLSVCACCOUNT="NT Authority\System" /SQLSYSADMINACCOUNTS="SQLStd\Administrator" /AGTSVCACCOUNT="NT AUTHORITY\Network Service" /SECURITYMODE=SQL /SAPWD="MyPa55w0rd" /TCPENABLED=1 /IACCEPTSQLSERVERLICENSETERMS=1`
- Commit the image
  - Docker stop SQLStd
  - Docker commit SQLStd teca/sqlstd1:latest

AUTOMATING CONTAINERS USING DOCKERFILES

- What are dockerfiles?
  - Batch commands to create Docker containers
  - Text files
- Why use them?
  - Automate and standardize container creation
- How do you run them?
  - >docker build
  - > docker build –t .sqlexpress
  - Takes files from the current directory

AUTOMATING SQL SERVER IMAGES

- The dockerfile runs the SQL Server command line installation
  - Builds a new image with the features you want
  - Does not use the prebuilt SQL Server images
- Starts with the Windows Server Core image
- Runs SQL Server setup.exe
- Can optionally setup databases
- Can connect to external volumes

AUTOMATING CONTAINERS USING DOCKERFILES

```
# SQL Server Express image
#  Build example
#     docker build -t sqlexpress .
FROM microsoft/mssql-server-windows-express

# create directory within SQL container for database files
RUN powershell -Command (mkdir C:\\SQLData)

#copy the database files from host to container
COPY AdventureWorks2014_Data.mdf C:\\SQLData
COPY AdventureWorks2014_Log.ldf C:\\SQLData

# Use Host port 1433
EXPOSE 1433
```

AUTOMATING CONTAINERS USING DOCKERFILES

- docker build -t sqlexpress .

Administrator: C:\Windows\system32\cmd.exe

```
C:\SQLExpressBuild>docker images
REPOSITORY                                  TAG                 IMAGE ID            CREATED             SIZE
<none>                                      <none>              97f6acbea55f        44 minutes ago      9.84 GB
sqlexpress                                  latest              887012a3e2b5        2 hours ago         12.4 GB
sqlstandard                                 latest              f258a7ffe182        16 hours ago        12.1 GB
<none>                                      <none>              9d6c6622dfa5        18 hours ago        9.84 GB
teca/sqlstd1                                latest              c9488b1fd95f        6 weeks ago         11.7 GB
microsoft/mssql-server-windows-developer    latest              2707b3c81de2        9 weeks ago         14.3 GB
microsoft/dotnet-samples                    dotnetapp-nanoserver a331d851e765       10 weeks ago        1.03 GB
microsoft/mssql-server-windows              latest              6fb5a1dbd3c8        3 months ago        14.6 GB
microsoft/aspnet                            latest              e761eca2f8df        3 months ago        10.1 GB
microsoft/windowsservercore                 latest              4d83c32ad497        3 months ago        9.56 GB
microsoft/nanoserver                        latest              d9bccb9d4cac        3 months ago        925 MB
microsoft/mssql-server-windows-express      latest              d5cc84b603b4        5 months ago        12 GB

C:\SQLExpressBuild>_
```

- docker run -d -e sa_password=SQLpwd2017 -e ACCEPT_EULA=Y -e attach_dbs="[{'dbName':'AdventureWorks2014','dbFiles':['C:\\SQLData\\AdventureWorks2014_Data.mdf','C:\\SQLData\\AdventureWorks2014_Log.ldf']}]" sqlexpress

PASS MARATHON

AUTOMATING SQL SERVER IMAGE

```
# Build command example
#                 docker build -t teca/sqlbuild:latest .
# Run command example
#                 docker run -d -ti --name SQLBuildStdContainer teca/sqlbuild
# Get IP Address
#                 docker inspect -format='{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}'
SQLBuildStdContainer

# Use the Windows Server Core image
FROM microsoft/windowsservercore

# Create directory for database files
RUN powershell -Command (New-Item c:\\SQLData -type directory)

# Copy the database files and setup files to the image
COPY *.mdf C:\\SQLData/
COPY *.ldf C:\\SQLData/
COPY c:\\sqlinstall/ c:\\sqlinstall/
```

AUTOMATING SQL SERVER IMAGES

```
# Run the unattended SQL Server setup
RUN c:\sqlinstall\Setup.exe /Q /ACTION=Install /FEATURES=SQL /INSTANCENAME=MSSQLSERVER
/SQLSVCACCOUNT="NT AUTHORITY\System" /SQLSYSADMINACCOUNTS="BUILTIN\ADMINISTRATORS"
/AGTSVCACCOUNT="NT AUTHORITY\Network Service" /SECURITYMODE=SQL /SAPWD="mySQLPWD"
/TCPENABLED=1 /IACCEPTSQLSERVERLICENSETERMS=1

# Setup ENV variables to attach DBs and expose SQL Server port 1433
ENV attach_dbs="[{'dbName':'AdventureWorks2014','dbFiles':['C:\\SQLData\\AdventureWorks2014_Data.mdf',
'C:\\SQLData\\AdventureWorks2014_Log.ldf']}]"
EXPOSE 1433

# Delete the setup files from the end container
RUN PowerShell -Command (Remove-Item c:\sqlinstall -Force -Recurse)

# End of dockerfile
```

SETTING THE STARTUP PROCESS

-e ATTACH_DBS

```
docker run -d -p 1433:1433 -e SA_PASSWORD=myNewpa55w0rd -e ACCEPT_EULA=Y -v C:/temp/:C:/temp/ -e ATTACH_DBS="[{'dbName':AdventureWorks,'dbFiles':['C:\\temp\\AdventureWorks2014.mdf','C:\\temp\\ AdventureWorks2014_log. ldf']}]" microsoft/mssql-server-windows-express
```

```
PS C:\> docker inspect -f "{{ .Config.Cmd }}" eb4
[powershell -Command $ErrorActionPreference = 'Stop'; $ProgressPreference = 'SilentlyContinue'; .\start -sa_password $env:sa_password -ACCEPT_EULA $env:ACCEPT_EULA -attach_dbs \"$env:attach_dbs\" -Verbose]
```

PASS MARATHON

START.PS1

```
# The script sets the sa password and start the SQL Service
# Also it attaches additional database from the disk
# The format for attach_dbs

param(
[Parameter(Mandatory=$false)]
[string]$sa_password,

[Parameter(Mandatory=$false)]
[string]$ACCEPT_EULA,

[Parameter(Mandatory=$false)]
[string]$attach_dbs
)


if($ACCEPT_EULA -ne "Y" -And $ACCEPT_EULA -ne "y")
{
        Write-Verbose "ERROR: You must accept the End User License Agreement before this container can start."
        Write-Verbose "Set the environment variable ACCEPT_EULA to 'Y' if you accept the agreement."

    exit 1
}
```

START.PS1 (CONTINUED)

```
# start the service
Write-Verbose "Starting SQL Server"
start-service MSSQL`$SQLEXPRESS

if($sa_password -ne "_")
{
    Write-Verbose "Changing SA login credentials"
    $sqlcmd = "ALTER LOGIN sa with password=" +"'" + $sa_password + "'" + ";ALTER LOGIN sa ENABLE;"
    & sqlcmd -Q $sqlcmd
}

$attach_dbs_cleaned = $attach_dbs.TrimStart('\\').TrimEnd('\\')

$dbs = $attach_dbs_cleaned | ConvertFrom-Json
```

START.PS1 (CONTINUED)

```
if ($null -ne $dbs -And $dbs.Length -gt 0)
{
    Write-Verbose "Attaching $($dbs.Length) database(s)"

    Foreach($db in $dbs)
    {
        $files = @();
        Foreach($file in $db.dbFiles)
        {
            $files += "(FILENAME = N'$($file)')";
        }

        $files = $files -join ","
        $sqlcmd = "IF EXISTS (SELECT 1 FROM SYS.DATABASES WHERE NAME = '" + $($db.dbName) + "') BEGIN EXEC sp_detach_db
[$($db.dbName)] END;CREATE DATABASE [$($db.dbName)] ON $($files) FOR ATTACH;"

        Write-Verbose "Invoke-Sqlcmd -Query $($sqlcmd)"
        & sqlcmd -Q $sqlcmd
        }
}
```

START.PS1 (END)

```
Write-Verbose "Started SQL Server."

$lastCheck = (Get-Date).AddSeconds(-2)
while ($true)
{
    Get-EventLog -LogName Application -Source "MSSQL*" -After $lastCheck | Select-Object
TimeGenerated, EntryType, Message
    $lastCheck = Get-Date
    Start-Sleep -Seconds 2
}
```

RUNNING START.PS1 FROM CM

- Add to dockerfile

```
#CMD without shell to run the start.ps1 script
CMD ["powershell -Command $ErrorActionPreference = 'Stop'; $ProgressPreference = 'SilentlyContinue'; .\\start -sa_password $env:sa_password -attach_dbs $env:attach_dbs -Verbose"]
```

SQL SERVER ON DOCKER BEST PRACTICE'S

- Change the SA password

```
> docker exec –it <containerID> /opt/mssql-tools/bin/sqlcmd
-S localhost –U sa –P SQLpwd2017
-Q 'ALTER LOGIN SA WITH PASSWORD="NewPassword";'
```

- Use different ports for multiple container instances

```
> docker run –e ACCEPT_EULA=Y –e SA_PASSWORD=SQLpwd2017
–p 1401:1433 –d microsoft/mssql-server-windows

> docker run –e ACCEPT_EULA=Y –e SA_PASSWORD=SQLpwd2017
–p 1402:1433 –d microsoft/mssql-server-windows
```

- Use volumes for persistent data
- Minimize what's running in the container
  - Don't use multiple instances

PASS MARATHON

CONTROLLING RESOURCE USAGE

- By default each container's access to the host machine's CPU cycles is unlimited
- Might not be the best for SQL Server licensing
- Docker Run
  - --cpus=<value>

```
docker run -d -p 1433:1433  --cpus=2  -e ACCEPT_EULA=Y -e SA_PASSWORD=SQLpwd2017
  -d microsoft/mssql-server-linux:latest

docker stats
```

PUSHING IMAGES TO THE DOCKER HUB

- Used to share Docker images
- Create an account on Docker Hub
  - docker login -u docker-registry-username
- Push your image
  - docker push docker-registry-username/docker-image-name

DEBUGGING CONTAINERS

- Docker logs [container ID]
  - STDOUT
  - STDERR
  - Container must be running
- Docker Inspect [container ID]
- Docker events (container ID)
- Docker run –i (interactive)
- Docker run -a (attach)
  - Shows STDOUT/STDERR

FAQS

Q: Can I run Linux containers on Windows Server?

A: You can with Windows 10 & Windows Server 1709

Q: How will Microsoft license SQL Server in a container

A: Identical to VM licensing – per Server or per Core

Q: Can multiple containers share the same external volume?

A: No. The host OS locks it

Q: My SQL Server container won't start how can I see what's happening?

A: Try running it interactively with the –i switch

Additional Resources

Windows Containers on Windows Server

https://docs.microsoft.com/en-us/virtualization/windowscontainers/quick-start/quick-start-windows-server

Run the SQL Server 2017 container image with Docker

https://docs.microsoft.com/en-us/sql/linux/quickstart-install-connect-docker

Tips on how-to Build Your First SQL Server container

http://searchsqlserver.techtarget.com/tip/Tips-on-how-to-build-your-first-SQL-Server-container

Dockerfile Reference

https://docs.docker.com/engine/reference/builder/

Docker images – Docker Hub

SQL Server 2017 – Windows and Linux

TechTarget – SQL Server Containers

PASS MARATHON

PASS MARATHON

Thank you for attending

Learn more from Michael Otey

@Michael_Otey

mikeo@teca.com

@sqlpass #sqlpass

@PASScommunity


PASS
MARATHON