Running Oracle Database and Applications in Docker Containers on Windows

Christian Shay
Product Manager, Windows
Oracle
October 23, 2018
Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle’s products may change and remains at the sole discretion of Oracle Corporation.
Program Agenda

1. Introduction to Containers and Docker
2. Oracle Database on Windows Containers
3. Instant Client on Windows Containers
4. Oracle Data Provider for .NET Core on Nano Containers
Introduction to Containers and Docker
What are Containers?

• Unlike a VM which provides hardware virtualization, a container provides operating-system-level virtualization by abstracting the “user space”
  – Containers *share* the host system’s kernel with other containers.

• Containers are lightweight (when compared to VMs)
  – No need to install Guest OS
  – Less CPU, RAM, Storage required
  – Can start up and shut down very quickly

• Provide uniformity despite differences between development and deployment
What is Docker?

• A software container platform designed for developing, shipping and running apps leveraging container tech

• Originated from Linux / Linux Containers
  – Also available on Windows and Mac OS X

• Editions:
  – Commercial Edition (EE) – Sold by Docker Corp
  – Community Edition (CE) – Part of Open Source Linux
What is Docker?

**Terminology**

- **docker-engine**: The host software running the containers
- **Images**: Collection of software to be run as a container
- **Containers**: A container on the host OS
- **Registry**: Place to store and download images
- **Volumes**: Place to persist data outside the container
Docker Key Components

- Registry
- Images
- Containers
- Docker daemon/engine

Source: https://docs.docker.com/engine/docker-overview/
Docker Key Components - Volumes

- Registry
- Images
- Containers
- Docker daemon/engine

Source: https://docs.docker.com/engine/docker-overview/
What is Docker?

**Concepts**

- Containers are non-persistent
  - Once a container is deleted, all files inside that container are gone
What is Docker?

Concepts

• Containers are non-persistent
  – Once a container is deleted, all files inside that container are gone

• Images are immutable
  – Changes to an image require to build a new image
  – Data to be persisted has to be stored in volumes
What is Docker?

**Concepts**

- Containers are non-persistent
  - Once a container is deleted, all files inside that container are gone
- Images are immutable
  - Changes to an image require to build a new image
  - Data to be persisted has to be stored in volumes
- Containers are spun up from images
Docker on Windows Requirements

• Windows Server 2016
  – Build 14393
  – Hyper V or Native Windows Containers

• Windows 10:
  – Windows 10 64bit: Pro, Enterprise or Education (1607 Anniversary Update, Build 14393 or later).
  – Virtualization enabled in BIOS
  – Hyper-V enabled
Server Core vs Nano Server

- **Server Core**
  - Minimalistic but includes more functionality for legacy apps (e.g., IIS web server)
  - Many legacy windows apps can run on it (e.g., Oracle Client)

- **Nano Server**
  - Much smaller and faster
  - Mostly for .NET Core applications only
  - Other executables built using Nano API
  - Missing features – aka Powershell, unzip
Server Core vs Nano Server

.NET Support

What OS to target with .NET containers

- Legacy .NET applications
  - .NET Framework 3.5, 4.x
    - Windows Server Core
      - Compatible with legacy apps
        - IIS
        - Larger image
    - Windows Nano Server
      - Cloud optimized,
        - Container OS,
        - Kestrel
        - Smaller faster start time
  - .NET Core
    - Linux
Docker on Windows Base Images

• Server Core and Nano:
  – https://hub.docker.com/r/microsoft/windowsservercore/
  – https://hub.docker.com/r/microsoft/nanoserver/

• .NET Development
  – https://hub.docker.com/r/microsoft/dotnet/

• More:
  – https://hub.docker.com/u/microsoft/
Container version vs Host version compatibility

• Tight dependencies between version of Host and container versions:
  – Compatibility matrix: https://bit.ly/2JSTo5A

• Resolve by using tags when adding official container images
  – FROM microsoft/windowsservercore:1709
  – FROM microsoft/nanoserver:1709_KB4043961
  – FROM microsoft/windowsservercore:1803
  – FROM microsoft/dotnet:2.1-aspnetcore-runtime-nanoserver-1803
Oracle Database on Windows Containers
Oracle Database on Windows Containers

How to get it working today

• Do a silent install of Oracle Database using software only install
  – Provide a response file

• Workaround required to avoid “Null pointer exception” issue:
  – Set INVENTORY_LOCATION=C:\Program Files\Oracle\Inventory in the response file

• Create Database
  – Copy data files, or
  – Generate database create script SQL scripts with DBCA outside of container
Oracle Database on Windows Containers

Files used for building docker image demo

• Windows Server 2016 running in Oracle Cloud (Compute)
• Oracle 18.3 installation package for Windows as zip file
• Response file as db.rsp
  – With INVENTORY_LOCATION fix added
• vsredist_x64.exe from
  – http://download.microsoft.com/download/0/5/6/056DCDA9-D667-4E27-8001-8A0C6971D6B1/vcredist_x64.exe
• Database Creation Assistant (DBCA) scripts or database files
Oracle Database on Windows Containers

Demo Docker File

FROM microsoft/windowsservercore

COPY /WINDOWS.X64_180000_db_home.zip c:/data/db_home.zip

RUN powershell -command Expand-Archive c:\data\db_home.zip -DestinationPath c:\data\db_home

RUN setx path ";c:\data\db_home\Bin;%path%;"

ENV ORACLE_HOME c:\data\db_home

COPY /vcredist_x64.exe c:/vcredist_x64.exe

RUN powershell.exe -Command \
    $ErrorActionPreference = 'Stop'; \
    Start-Process c:\vcredist_x64.exe -ArgumentList '/install /passive /norestart' -Wait ;

    Remove-Item c:\vcredist_x64.exe -Force
Oracle Database on Windows Containers

Demo Docker File

COPY db.rsp c:\data\db.rsp

RUN c:\data\db_home\setup.bat -silent -responseFile c:\data\db.rsp

ENV ORACLE_HOME c:\\data\\db_home

ENV ORACLE_SID orclcdb

ENV CLASSPATH c:\data\db_home\jlib;c:\data\db_home\rdbms\jlib;

EXPOSE 5500 1521

COPY /scripts C:/data/app/myhome/admin/orclcdb

RUN c:/data/app/myhome/admin/orclcdb/scripts/orclcdb.bat

COPY post_install.bat c:/data/post_install.bat

CMD ["c:/data/post_install.bat"]
Oracle Database on Windows Containers

post_install.bat

• Additional post installation items can be scripted, for example:
  – orapwd file=PWDorclcdb.ora password=new_passwd_99 entries=100
  – lsnrctl start
  – alter pluggable database orclpdb open

• Post installation batch file should not terminate or container will stop
Oracle Database on Windows Containers

Demo

• Windows Server 2016 running in Oracle Cloud (Compute)
• Try this yourself!
  – Free cloud credits: https://cloud.oracle.com/tryit
Database in Windows Container
Oracle Database on Windows Containers

Demo Commands

• `docker build -t dboow c:\dockerdemo\dbdemo`

• `docker run -it dboow`
  – Interactive shell

• `docker run -d dboow`
  – In background
Oracle Database on Windows Containers

Connect to the database from outside Oracle clients (eg other containers)

• Find out IP address of database container
  – From interactive shell: `ipconfig`
  – From host OS: `Docker ps` and `Docker inspect container_name`

• Find out Service name
  – From interactive shell: `Lsnrctl services`

• Set up TNSNAMES.ORA and SQLNET.ORA in client

• Then connect as usual
Instant Client on Windows Containers
Oracle Instant Client on Windows Container

Dockerfile

• Use Server Core
  – Won’t run on Nano Server containers

• Copy instant client zips (from OTN) and extract them

• Set path to point to Instant Client

• That’s it!
Demo: Node.js and Oracle Instant Client

• Banana Farmer demo from Chris Jones’ blog -- converted to Windows:
  – *A node-oracledb Web Service in Docker*

• Creates a web service that responds to GET, POST, PUT
  – eg GET [http://172.17.0.3:3000/bananas/Gita](http://172.17.0.3:3000/bananas/Gita)
  – [{"SHIPMENT":"{ "farmer":"Gita", "ripeness":"All Green", "kilograms":100 }"}]
The Scenario

• Shipments of bananas from farmers are recorded
• Shipments have a farmer name, ripeness, and weight
• Shipments can be inserted, queried, updated or deleted
Banana Color Guide

ALL GREEN
As received at your warehouse from Central and South America

1

First color change during warehouse processing, usually seen on the shoulder.

2

50% GREEN, 50% YELLOW
Recommended color for warehouse outturn. Adjust back or ahead of delivery time, temperature, distance and retail color preferences. Consumer purchase now to enjoy later.

3

MORE YELLOW THAN GREEN
Firm fruit with great eating flavor. Easily bruised – handle with extra care.

4

YELLOW FLECKED WITH BROWN
Sweet eating flavor. Perfect texture and consistency for blender drinks and baking.

5

6

7

©2004 Dole Fresh Fruit Company
DOLE is a registered trademark of Dole Food Company, Inc.
The Schema

CREATE TABLE bananas (shipment VARCHAR2(4000) CHECK (shipment IS JSON));

INSERT INTO bananas VALUES ('{ "farmer": "Gita", "ripeness": "All Green", "kilograms": 100 }');

INSERT INTO bananas VALUES ('{ "farmer": "Ravi", "ripeness": "Full Yellow", "kilograms": 90 }');

INSERT INTO bananas VALUES ('{ "farmer": "Mindy", "ripeness": "More Yellow than Green", "kilograms": 92 }');
Demo: Node.js and Oracle Instant Client

Banana Farmer demo

• Demo dockerfile does the following:
  – Copies Instant Client zip and required MSVCRT120.DLL dependency
  – Automatically downloads node installation zip file
  – Expands zip files
  – Copies package.json and Server.js
  – Npm install
    • installs node-oracledb dependency listed in packages.json
  – Set path to include instantclient and node.js
  – Start node
Demo: Node.js and Oracle Instant Client

Demo Docker File

FROM microsoft/windowsservercore:1709 as builder

SHELL ["powershell", "-Command", "$ErrorActionPreference = 'Stop'; $ProgressPreference = 'SilentlyContinue';"]

ENV NODE_VERSION 8.12.0

ENV IC_FILENAME instantclient-basic-windows.x64-18.3.0.0.0dbru.zip

ENV IC_FILENAME2 instantclient-sqlplus-windows.x64-18.3.0.0.0dbru.zip

ENV IC_FOLDER instantclient_18_3

RUN Invoke-WebRequest $('https://nodejs.org/dist/v{0}/node-v{0}-win-x64.zip' -f $env:NODE_VERSION) -OutFile 'node.zip' -UseBasicParsing ;

Expand-Archive node.zip -DestinationPath C:\ ;

Rename-Item -Path $('C:\node-v{0}-win-x64' -f $env:NODE_VERSION) -NewName 'C:\nodejs';
Demo: Node.js and Oracle Instant Client

Demo Docker File

COPY $IC_FILENAME instantclient.zip
COPY $IC_FILENAME2 instantclient2.zip
COPY msvcr120.dll c:/windows/system32/msvcr120.dll

RUN Expand-Archive instantclient.zip -DestinationPath C:\; \\
Expand-Archive instantclient2.zip -DestinationPath C:\; \\

Rename-Item -Path $(env:IC_FOLDER -f $env:NODE_VERSION) -NewName 'C:\instantclient';

RUN $env:PATH = 'C:\instantclient;C:\nodejs;{0}' -f $env:PATH ; \\
    [Environment]::SetEnvironmentVariable('PATH', $env:PATH, [EnvironmentVariableTarget]::Machine)

RUN mkdir c:/demo
Demo: Node.js and Oracle Instant Client

Demo Docker File

WORKDIR c:/demo
COPY package.json package.json
COPY server.js server.js
RUN npm install
FROM microsoft/windowsservercore:1709
SHELL ["powershell", "-Command", "+$ErrorActionPreference = 'Stop'; $ProgressPreference = 'SilentlyContinue';"]
COPY --from=builder /nodejs /nodejs
COPY --from=builder /instantclient /instantclient
COPY --from=builder /demo /demo
COPY --from=builder /windows/system32/msvcr120.dll /nodejs/msvcr120.dll
Demo: Node.js and Oracle Instant Client

Demo Docker File

WORKDIR c:/demo

ARG SETX=/M

RUN setx /M PATH $('C:\instantclient;C:\nodejs;'+$Env:PATH)

CMD ["c:/nodejs/npm.cmd", "start"]
Instant Client in Windows Container
Oracle Data Provider for .NET Core on Nano Containers
ODP.NET Core

RTM

• Available on nuget.org

• Supports Oracle Linux and Red Hat Linux

• Connectivity to Oracle Database 11.2 and higher
Oracle Data Provider for .NET Core on Nano Server

Demo

• Build .NET Core Web application in Visual Studio or command line
• “Publish” to folder
• Zip up folder
• Move zip into deployment directory
Oracle Data Provider for .NET Core on Nano Server

Demo

• In dockerfile:
  – FROM microsoft/dotnet:2.1-aspnetcore-runtime-nanoserver-1709
  – (Copy expanded zip file into container)
  – ENTRYPOINT ["dotnet", "/publish/WebApplication10.dll"]

• docker run -p 45000:80 -P --env-file d:\dockertest\dotnetdemo\env.list dotnetdemo

• From browser: http://127.0.0.1:45000/
ODP.NET Core in Nano Server
Upcoming .NET Sessions - Today

• Accelerate Application Performance: Tips for Faster Oracle Database .NET Programs
  • Wednesday – 4:45 PM – 5:30 PM Moscone West – 3010
Upcoming .NET Sessions - Thursday

• Eliminate Application Downtime with Oracle Database and .NET
  – Thursday – 9:00 AM – 9:45 AM Moscone West – 3010

• Hands on Lab: Building .NET Applications with Oracle
  – Thursday – 12:00 PM – 1:00 PM Marriott Marquis (Yerba Buena Level) – Salon 3
.NET Demo at the Exchange – Booth DBA-WU2
Questions and Answers