

Running MQ .NET Framework Application on Docker containers

Harsha Ketaraju

Published on 14/08/2019

Introduction:

This blog gives you an overview on how to run a MQ .NET Framework Application on docker containers.

Create Project Folder:

1. Create a new folder by name “SimplePut”.
2. Copy SimplePut.cs and SimplePut.csproj files to the SimplePut folder created in Step 1 from the following location:

Windows:

MQ_INST_PATH\tools\dotnet\samples\cs\base\SimplePut

3. Remove the following line from SimplePut.csproj file
<HintPath>..\..\..\..\..\Lib\amqmdnet.dll</HintPath>
4. Copy amqmdnetstd.dll file to the SimplePut folder created in Step 1 from the following location:

Windows:

MQ_INST_PATH\bin

Create a DockerFile for MQ .NET Framework Application:

1. Create a file by name “DockerFile” in SimplePut project folder which will be used as Dockerfile for sample application to build docker image.
2. Add the below mentioned text to your DockerFile for Windows Containers as follows:
3. FROM microsoft/windowsservercore
4. COPY /bin/debug/ /root/
5. ENTRYPOINT /root/SimplePut.exe

Build the Docker Image:

1. Open command prompt to build the docker image.
2. To verify whether docker is installed or not, issue the following command on command line as follows:
docker -v
Output:
Docker version 18.09.0, build 4d60db4
If not, please install Docker for windows containers from Docker website.
3. Navigate to your project folder and use the below command to build your Docker image.
docker build -t simpleput .
Note: Image_name should be lowercase alphabets.
4. Docker image with simpleput will be successfully created. Output as follows:
5. Sending build context to Docker daemon 2.381MB
6. Step 1/3 : FROM microsoft/windowsservercore

7. ---> 81d80198a492
8. Step 2/3 : COPY /bin/debug/ /root/
9. ---> Using cache
10. ---> 5d81394da6f4
11. Step 3/3 : ENTRYPOINT /root/SimplePut.exe
12. ---> Using cache
13. ---> f250db37f272
14. Successfully built c9e02535kz8e
15. Successfully tagged app:latest
16. By issuing the following command, you can view the latest docker image created.

`docker images`

17. Output for the above command as follows:

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
simpleput	latest	c9e02535kz8e	35 seconds ago	1.54 GB

Running the Docker Image:

1. Run the docker image by passing arguments as follows:
`docker run simpleput -qm Test1 -q Q1 -h 9.yyy.zzz.ww -p 1423`
2. Program will be putting messages successfully onto the queue which is mentioned as arguments.
3. Start of SimplePut Application
4. MQ Parameters
5. 1) queueName = Q1
6. 2) keyRepository =
7. 3) cipherSpec =
8. 4) host = 9.yyy.zzz.ww
9. 5) port = 1423
10. 6) channel = SYSTEM.DEF.SVRCONN
11. 7) numberOfMsgs = 1
12. 8) sslPeerName =
13. 9) keyResetCount = 0
14. 10) sslCertRevocationCheck = False
- 15.
16. Connecting to queue manager.. done
17. Accessing queue Q1.. done
18. Message 1 <test message>.. put
19. Closing queue.. done
20. Disconnecting queue manager.. done
- 21.
22. End of SimplePut Application

References:

[Installing IBM MQ classes for .NET Standard](#)