PS D:\> docker --version

Docker version 26.1.1, build 4cf5afa

PS D:\> docker search mq

NAME DESCRIPTION STARS OFFICIAL

eclipse-mosquitto Eclipse Mosquitto is an open source message … 1249 [OK]

emqx The most scalable open-source MQTT broker fo… 82 [OK]

greenbone/mqtt-broker A mosquitto.org MQTT broker for the Greenbon… 2

sensebox/mqtt-osem-integration MQTT Integration for openSenseMap API 0

alpinelinux/mqtt-exec 0

zwavejs/zwavejs2mqtt Fully configurable Zwave to MQTT gateway and… 96

zwavejs/zwave-js-ui Fully configurable Zwave Control Panel and M… 72

solace/solace-pubsub-connector-ibmmq Solace PubSub+ Connector for IBM MQ 0

fraunhoferiosb/frost-server-mqtt The MQTT-Only package of the FROST-Server 0

lintoai/linto-platform-overwatch MQTT global subscriber for a fleet of LinTO … 0

edgexfoundry/device-mqtt Device service ingesting device data via MQT… 0

nasqueron/rabbitmq RabbitMQ wth management, MQTT and STOMP plug… 0

edgexfoundry/device-mqtt-arm64 ARM64 device service ingesting device data v… 0

edgexfoundry/docker-device-mqtt ARCHIVED! The legacy device service ingestin… 0

openquantumsafe/mosquitto This provides a quantum-safe crypto enabled … 1

edgexfoundry/docker-device-mqtt-arm64 ARCHIVED! ARM64 legacy device service ingest… 1

mquandalle/wekan An open-source kanban 109

mqttgateway/mqttgateway Prometheus MQTT Gateway 2

mq60/giftr-api-w21 0

mq83/noisy 0

mq83/trojan 0

mq83/polipo 0

mq83/myss shadowsocks-libev + v2ray-plugin(Support tls… 1

mqkeda/sample-app 0

mqarty/docker-py3-boto3 0

PS D:\> docker pull icr.io/ibm-messaging/mq:latest

latest: Pulling from ibm-messaging/mq

570d5d7d3a6a: Pull complete

df89b03e7846: Pull complete

e1aacfe1da0a: Pull complete

599968acb1ec: Pull complete

e03737b7efc7: Pull complete

0102033d6b75: Pull complete

97431af1612e: Pull complete 4e2c979ad44c: Pull complete

e01e4d2baccf: Pull complete

fb3282fd7422: Pull complete

31975ebe2841: Pull complete

a0d70aa6766d: Pull complete 50ed1b373a33: Pull complete

931aa9665562: Pull complete

fba745507543: Pull complete

ed3cffb3b6f4: Pull complete

9d1013bd815a: Pull complete

7fee3da613d3: Pull complete

bdbb47614638: Pull complete

Digest: sha256:cdebb5b8ae79b4b6d62b97c532a2907745e63510ea190a00e223f9701b52108c

Status: Downloaded newer image for icr.io/ibm-messaging/mq:latest

icr.io/ibm-messaging/mq:latest

PS D:\> docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

icr.io/ibm-messaging/mq latest 74a7d60fd2bf 9 days ago 1.66GB

now if we create a container directly from this image and then if we stop that later, then all the MQ data will be gone so we need to create volume on local disc.

So the /var/mqm where the data is kept on your container so that container data will be replicated on my host operating system which is currently Windows11. Then that data will be available irrespective whether the container is live or not.

So I can stop and create a new container and I can remove this container with same exact parameters and that container will start appending the MQ data which was previously written by the previous container.

This feature is enabled by Volume.

PS D:\> docker volume ls

DRIVER VOLUME NAME

PS D:\> docker volume create qm1data

qm1data

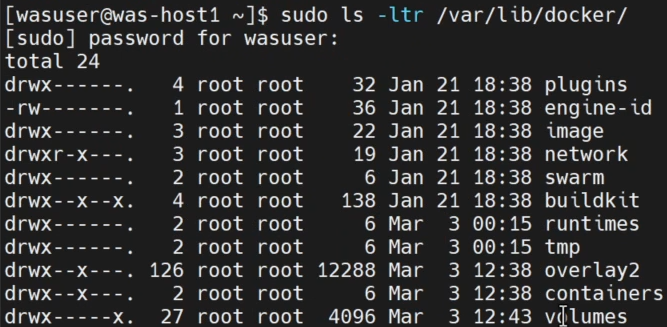
PS D:\> docker volume ls

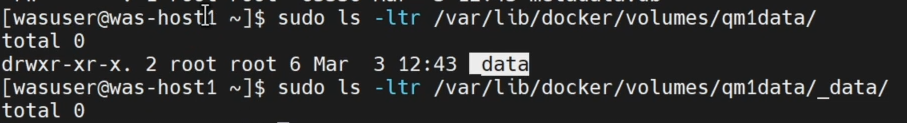
DRIVER VOLUME NAME

local qm1data

where this volume created ?

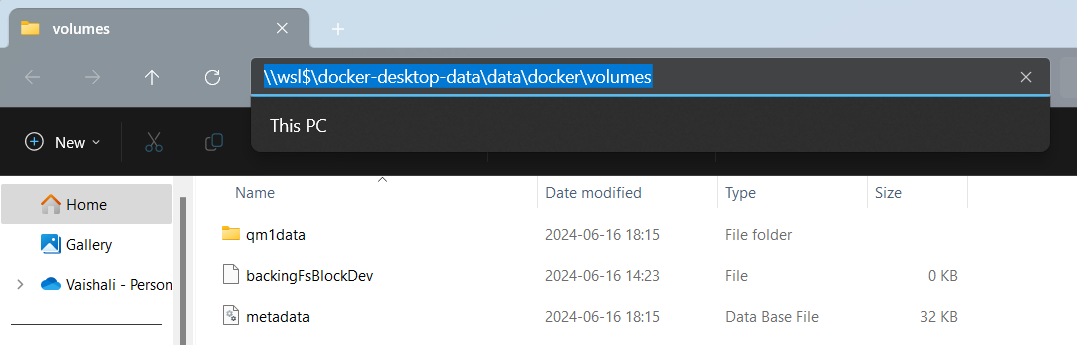
if you are on linux ->





On Windows - >

\\wsl$\docker-desktop-data\data\docker\volumes



PS D:\> docker run --env LICENSE=accept --env MQ\_QMGR\_NAME=QM1 --volume qm1data:/mnt/mqm --publish 1414:1414 --publish 9443:9443 --detach --env MQ\_APP\_PASSWORD=passw0rd --env MQ\_ADMIN\_PASSWORD=passw0rd --name QM1 icr.io/ibm-messaging/mq:latest

0dd51cb324a883fd9ea3125f52f9cad9832340a178e34f66d3209ab53944a001

--volume qm1data:/mnt/mqm --- volumename:workpath .. in this image IBM has replaced /var/ibm/mq with /mnt/mqm

1414:1414 - first port is local OS port like for Windows 11 and 2nd one for container.

9443:9443 - same but this is used for Web interface

MQ\_ADMIN\_PASSWORD=passw0rd -- creds for Web GUI ( userID : admin )

2nd port could be same if you run multiple container as their IP’s can be different but not the first port.



Lets run the same command again to run another container ----

PS D:\> docker run --env LICENSE=accept --env MQ\_QMGR\_NAME=QM1 --volume qm1data:/mnt/mqm --publish 1414:1414 --publish 9443:9443 --detach --env MQ\_APP\_PASSWORD=passw0rd --env MQ\_ADMIN\_PASSWORD=passw0rd --name QM1 icr.io/ibm-messaging/mq:latest

docker: Error response from daemon: Conflict. The container name "/QM1" is already in use by container "0dd51cb324a883fd9ea3125f52f9cad9832340a178e34f66d3209ab53944a001". You have to remove (or rename) that container to be able to reuse that name.

lets change the queue mananger name : -

PS D:\> docker run --env LICENSE=accept --env MQ\_QMGR\_NAME=QM2 --volume qm1data:/mnt/mqm --publish 1414:1414 --publish 9443:9443 --detach --env MQ\_APP\_PASSWORD=passw0rd --env MQ\_ADMIN\_PASSWORD=passw0rd --name QM1 icr.io/ibm-messaging/mq:latest

docker: Error response from daemon: Conflict. The container name "/QM1" is already in use by container "0dd51cb324a883fd9ea3125f52f9cad9832340a178e34f66d3209ab53944a001". You have to remove (or rename) that container to be able to reuse that name.

this is due to same local ports.

PS D:\> docker exec -it 0dd51cb324a8 /bin/sh

sh-5.1$

sh-5.1$ exit

exit

PS D:\>

PS D:\> docker exec -it 0dd51cb324a8 /bin/sh

sh-5.1$ ls -ltr

total 68

drwxr-xr-x 2 root root 4096 Aug 9 2021 srv

lrwxrwxrwx 1 root root 8 Aug 9 2021 sbin -> usr/sbin

drwxr-xr-x 2 root root 4096 Aug 9 2021 media

lrwxrwxrwx 1 root root 9 Aug 9 2021 lib64 -> usr/lib64

lrwxrwxrwx 1 root root 7 Aug 9 2021 lib -> usr/lib

drwxr-xr-x 2 root root 4096 Aug 9 2021 home

dr-xr-xr-x 2 root root 4096 Aug 9 2021 boot

lrwxrwxrwx 1 root root 7 Aug 9 2021 bin -> usr/bin

dr-xr-xr-x 2 root root 4096 Aug 9 2021 afs

drwx------ 2 root root 4096 May 23 13:50 lost+found

drwxr-xr-x 1 root root 4096 May 23 13:50 usr

drwxr-xr-x 1 root root 4096 Jun 7 05:35 opt

drwxr-xr-x 1 root root 4096 Jun 7 05:35 var

dr-xr-x--- 1 root root 4096 Jun 7 05:35 root

drwxrwsr-x 1 1001 root 4096 Jun 7 05:35 mnt

drwxr-xr-x 2 root root 4096 Jun 7 05:35 licenses

drwxr-xr-x 1 root root 4096 Jun 16 16:36 etc

dr-xr-xr-x 11 root root 0 Jun 16 16:36 sys

dr-xr-xr-x 233 root root 0 Jun 16 16:36 proc

drwxr-xr-x 5 root root 340 Jun 16 16:36 dev

drwxrwsr-x 1 1001 root 4096 Jun 16 16:36 run

drwxrwxrwt 1 root root 4096 Jun 16 16:36 tmp

sh-5.1$ cd /mnt

sh-5.1$ ls

mqm mqm-data mqm-log

sh-5.1$ cd mqm

sh-5.1$ ls

data

sh-5.1$ cd data

sh-5.1$ ls

config conv errors exits exits64 log mqclient.ini mqft mqs.ini qmgrs service.env sockets trace web

sh-5.1$ cd qmgrs

sh-5.1$ ls

@SYSTEM QM1

sh-5.1$ cd QM1

sh-5.1$ ls

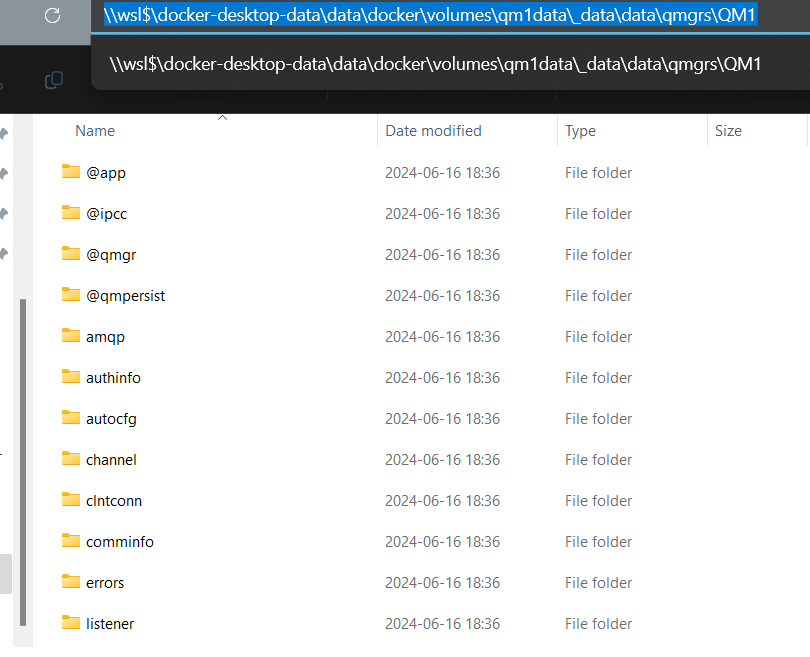
@app @qmpersist amqp blockaddr.ini comminfo master namelist qm.ini queues services startprm

@ipcc active authinfo channel errors mqat.ini plugcomp qmanager reserve.dat ssl topic

@qmgr amqalchk.fil autocfg clntconn listener mqxr procdef qmstatus.ini scratch standby userdata

sh-5.1$ exit

exit

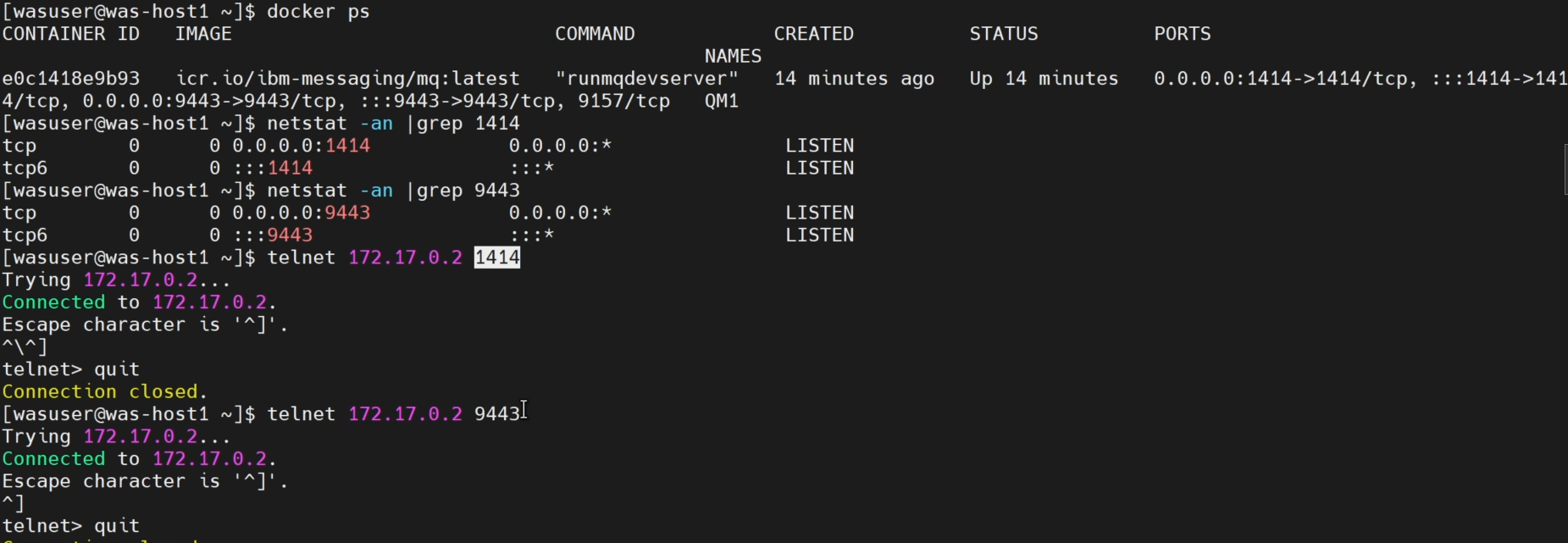


You can see its identical. This helps you to removing the existing container and recreate container with the same commands and continue the work which you have been doing. So no disturbance of the work which you are doing, coz binaries will get uninstalled, new binaries will get installed but your data will remain as it is coz its coming from volume.

PS D:\> docker inspect 0dd51cb324a8



"IPAddress": "172.17.0.2"



PS D:\> docker exec -it 0dd51cb324a8 bash

bash-5.1$ runmqsc QM1

5724-H72 (C) Copyright IBM Corp. 1994, 2024.

Starting MQSC for queue manager QM1.

DISPLAY CHANNEL(\*)

1 : DISPLAY CHANNEL(\*)

AMQ8414I: Display Channel details.

CHANNEL(DEV.ADMIN.SVRCONN) CHLTYPE(SVRCONN)

AMQ8414I: Display Channel details.

CHANNEL(DEV.APP.SVRCONN) CHLTYPE(SVRCONN)

AMQ8414I: Display Channel details.

CHANNEL(SYSTEM.AUTO.RECEIVER) CHLTYPE(RCVR)

AMQ8414I: Display Channel details.

CHANNEL(SYSTEM.AUTO.SVRCONN) CHLTYPE(SVRCONN)

AMQ8414I: Display Channel details.

CHANNEL(SYSTEM.DEF.AMQP) CHLTYPE(AMQP)

AMQ8414I: Display Channel details.

CHANNEL(SYSTEM.DEF.CLUSRCVR) CHLTYPE(CLUSRCVR)

AMQ8414I: Display Channel details.

CHANNEL(SYSTEM.DEF.CLUSSDR) CHLTYPE(CLUSSDR)

AMQ8414I: Display Channel details.

CHANNEL(SYSTEM.DEF.RECEIVER) CHLTYPE(RCVR)

AMQ8414I: Display Channel details.

CHANNEL(SYSTEM.DEF.REQUESTER) CHLTYPE(RQSTR)

AMQ8414I: Display Channel details.

CHANNEL(SYSTEM.DEF.SENDER) CHLTYPE(SDR)

AMQ8414I: Display Channel details.

CHANNEL(SYSTEM.DEF.SERVER) CHLTYPE(SVR)

AMQ8414I: Display Channel details.

CHANNEL(SYSTEM.DEF.SVRCONN) CHLTYPE(SVRCONN)

AMQ8414I: Display Channel details.

CHANNEL(SYSTEM.DEF.CLNTCONN) CHLTYPE(CLNTCONN)