Thursday, August 10, 2017

Extending MapR PACC

Extending MapR PACC

Although we cannot modify a MapR-provided Docker image directly we can build a custom image that is based on MapR Persistent Application Client Containe
(PACC). This blog shows a custom Dockerfile that is used to create a new Docker image.

Requirement I : In this example, I am creating a PACC image with CentOS 7.3, hadoop, Fuse and **maria**db (Open source Mysql) .

1) Create a directory where we will write a Docker file with set of instructions to built a custom image and download maps-setup.sh script.

```
mkdir abizer ; cd abizer/

wget http://package.mapr.com/releases/installer/mapr-setup.sh

chmod 777 mapr-setup.sh
```

2) Now vi to create Docker file with set of instructions .

```
[root@node112rhel72 abizer]# vi Dockerfile
FROM centos:centos7
ENV container docker
RUN yum -y upgrade && yum install -y curl file net-tools openssl sudo syslinux wget which mysql java-1.8.0-openjdk-devel && yum -q clean all
LABEL mapr.os=centos7 mapr.version=5.2.1 mapr.mep_version=3.0
COPY mapr-setup.sh /opt/mapr/installer/docker/
RUN /opt/mapr/installer/docker/mapr-setup.sh -r http://package.mapr.com/releases container client 5.2.1 3.0 mapr-client mapr-posix-client-container
ENTRYPOINT ["/opt/mapr/installer/docker/mapr-setup.sh", "container"]
```

3) Ensure both scripts are executable .

```
[root@node112rhel72 abizer]# ls -l mapr-*
-rwxrwxrwx. 1 root root   2476 Aug 10 17:12 mapr-docker-client.sh

-rwxrwxrwx. 1 root root 106152 Aug  9 11:24 mapr-setup.sh
```

4) Modify mapr-docker-client.sh script with cluster name , CLDB IP and Docker image TAG (abizer:5.2.1_3.0_centos7_mysql)

```
[root@node112rhel72 abizer]# vi mapr-docker-client.sh
-------------------------------------------------------------------------------------
#!/bin/sh

# The environment variables in this file are for example only. These variables
# must be altered to match your docker container deployment needs

MAPR_CLUSTER=ClusterNFS4
MAPR_CLDB_HOSTS=10.10.70.117

# MapR POSIX client mount path to enable direct MapR-FS access
MAPR_MOUNT_PATH=/mapr

# MapR secure cluster ticket file path
MAPR_TICKETFILE_LOCATION=

# MapR client user / group
MAPR_CONTAINER_USER=$(id -u -n)
MAPR_CONTAINER_UID=$(id -u)
MAPR_CONTAINER_GROUP=$(id -g -n)
MAPR_CONTAINER_GID=$(id -g)
MAPR_CONTAINER_PASSWORD=

# Container memory: specify host XX[kmg] or 0 for no limit. Ex: 8192m, 12g
MAPR_MEMORY=0

# Container timezone: filename from /usr/share/zoneinfo
MAPR_TZ=${TZ:-"America/New_York"}

# Container network mode: "host" causes the container's sshd service to conflict
# with the host's sshd port (22) and so it will not be enabled in that case
MAPR_DOCKER_NETWORK=bridge

# Container security: --privileged or --cap-add SYS_ADMIN /dev/<device>
MAPR_DOCKER_SECURITY="$([ -n $"MAPR_MOUNT_PATH" ] && echo "--cap-add SYS_ADMIN --cap-add SYS_RESOURCE --device /dev/fuse")"

# Other Docker run args:
MAPR_DOCKER_ARGS=""

### do not edit below this line ###
grep -q -s DISTRIB_ID=Ubuntu /etc/lsb-release && \
  MAPR_DOCKER_SECURITY="$MAPR_DOCKER_SECURITY --security-opt apparmor:unconfined"

MAPR_DOCKER_ARGS="$MAPR_DOCKER_SECURITY \
  --memory $MAPR_MEMORY \
  --network=$MAPR_DOCKER_NETWORK \
  -e MAPR_DISKS=$MAPR_DISKS \
  -e MAPR_CLUSTER=$MAPR_CLUSTER \
  -e MAPR_LICENSE_MODULES=$MAPR_LICENSE_MODULES \
  -e MAPR_MEMORY=$MAPR_MEMORY \
  -e MAPR_MOUNT_PATH=$MAPR_MOUNT_PATH \
  -e MAPR_SECURITY=$MAPR_SECURITY \
  -e MAPR_TZ=$MAPR_TZ \
  -e MAPR_USER=$MAPR_USER \
  -e MAPR_CONTAINER_USER=$MAPR_CONTAINER_USER \
  -e MAPR_CONTAINER_UID=$MAPR_CONTAINER_UID \
  -e MAPR_CONTAINER_GROUP=$MAPR_CONTAINER_GROUP \
  -e MAPR_CONTAINER_GID=$MAPR_CONTAINER_GID \
  -e MAPR_CONTAINER_PASSWORD=$MAPR_CONTAINER_PASSWORD \
  -e MAPR_CLDB_HOSTS=$MAPR_CLDB_HOSTS \
  -e MAPR_HS_HOST=$MAPR_HS_HOST \
```

```
  -e MAPR_OT_HOSTS=$MAPR_OT_HOSTS \
  -e MAPR_ZK_HOSTS=$MAPR_ZK_HOSTS \
  $MAPR_DOCKER_ARGS"

[ -f "$MAPR_TICKETFILE_LOCATION" ] && MAPR_DOCKER_ARGS="$MAPR_DOCKER_ARGS \
  -e MAPR_TICKETFILE_LOCATION=/tmp/mapr_ticket \
  -v $MAPR_TICKETFILE_LOCATION:/tmp/mapr_ticket:ro"
[ -d /sys/fs/cgroup ] && MAPR_DOCKER_ARGS="$MAPR_DOCKER_ARGS -v /sys/fs/cgroup:/sys/fs/cgroup:ro"

echo $MAPR_DOCKER_ARGS
docker run -it $MAPR_DOCKER_ARGS abizer:5.2.1_3.0_centos7_mysql "$@"
```

---

5) Now built the docker image , this has 7 steps which we defined in Docker file.

i) Pull CentOS 7 image
ii) Built the ENV which is docker container.
iii) Yum install required packages.
iv) Label the image or add metadata to an image
v) Copy mapr-setup.sh in docker container (Image to be built)
vi) Run mapr-setup.sh script in docker which will download all kinds PACC rpms needed.
vii) ENTRYPOINT will configure the container and built the image

Running below commands reads docker file from current location and builts and image .

```
[root@node112rhel72 abizer]# docker build -t abizer:5.2.1_3.0_centos7_mysql .
Sending build context to Docker daemon 112.6 kB
Step 1/7 : FROM centos:centos7
centos7: Pulling from library/centos
Digest: sha256:26f74cefad82967f97f3eeeef88c1b6262f9b42bc96f2ad61d6f3fdf544759b8
Status: Downloaded newer image for centos:centos7
 ---> 328edcd84f1b
Step 2/7 : ENV container docker
 ---> Using cache
 ---> 4f5d082615b7
Step 3/7 : RUN yum -y upgrade && yum install -y curl file net-tools openssl sudo syslinux wget which mysql java-1.8.0-openjdk-devel && yum -q clean all
 ---> Using cache
 ---> 6dd287d2a77f
Step 4/7 : LABEL mapr.os centos7 mapr.version 5.2.1 mapr.mep_version 3.0
 ---> Using cache
 ---> 5c1a0b1e55de
Step 5/7 : COPY mapr-setup.sh /opt/mapr/installer/docker/
 ---> Using cache
 ---> b1aaadae2bce
Step 6/7 : RUN /opt/mapr/installer/docker/mapr-setup.sh -r http://package.mapr.com/releases container client 5.2.1 3.0 mapr-client mapr-posix-client-container
 ---> Using cache
 ---> cec7a8030014
Step 7/7 : ENTRYPOINT /opt/mapr/installer/docker/mapr-setup.sh container
 ---> Using cache
 ---> 5065d2d71c57
Successfully built 5065d2d71c57
```

6) Image (**5.2.1_3.0_centos7_mysql**) is built as expected without errors.

```
[root@node112rhel72 abizer]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
abizer              5.2.1_3.0_centos7_mysql   5065d2d71c57        50 minutes ago      1.03 GB
centos              centos7             328edcd84f1b        7 days ago          193 MB
maprtech/installer   ubuntu16            c7cbc778e81e        2 weeks ago         480 MB
```

7) Now to connect to container run below script which will connect to container and start Fuse as well.

```
[root@node112rhel72 abizer]# sh mapr-docker-client.sh

Testing for cluster user account...

Enter MapR cluster user name: root
 ...Success

Configuring MapR client ( -c -C 10.10.70.117 -N ClusterNFS4)...

create /opt/mapr/conf/conf.old
Configuring Hadoop-2.7.0 at /opt/mapr/hadoop/hadoop-2.7.0
Done configuring Hadoop
CLDB node list: 10.10.70.117:7222
Zookeeper node list:

...Success

Starting services (mapr-posix-client-container)...

Started service mapr-posix-client-container

...Success
```

Validation : Fuse mount is mounted and mariaDB package is installed .

```
[root@373bb434b21c /]# df -hP
Filesystem              Size  Used Avail Use% Mounted on
overlay                  39G  7.1G   32G  19% /
tmpfs                    32G     0   32G   0% /dev
/dev/mapper/rhel-root    39G  7.1G   32G  19% /etc/hosts
shm                      64M     0   64M   0% /dev/shm
tmpfs                    32G     0   32G   0% /sys/fs/cgroup
tmpfs                    32G     0   32G   0% /sys/firmware
posix-client-container  119G  504M  118G   1% /mapr

[root@373bb434b21c /]# rpm -qa| grep -i maria
mariadb-5.5.52-1.el7.x86_64
mariadb-libs-5.5.52-1.el7.x86_64
```

Requirement II :

I had a similar requirement to built docker Image with OEL and have packages for Percona Server for MySQL 5.7 installed and pre-built docker Image .

Created below docker file and followed all the steps listed earlier in this blog to built and image.

```
[root@node112rhel72 abizeroraclemysql]# cat Dockerfile
# Download base image OEL 7.3
FROM oraclelinux:7.3
# Define Env
ENV container docker
# Install java and other packages needed by MapR Fuse
RUN yum -y upgrade && yum install -y curl file net-tools openssl sudo syslinux wget which java-1.8.0-openjdk-devel && yum -q clean all
# Import Keys for percona packages
RUN rpm --import https://www.percona.com/downloads/RPM-GPG-KEY-percona
# Install required percona packages
RUN yum install -y http://www.percona.com/downloads/percona-release/redhat/0.1-4/percona-release-0.1-4.noarch.rpm https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm && yum -y up
&& yum install -y which nc sysbench perl-Digest-MD5 percona-xtrabackup-24 Percona-Server-{client,server,shared,test}-57
# Label the image or add metadata to an image
LABEL mapr.os=OEL7 mapr.version=5.2.1 mapr.mep_version=3.0
# Copy mapr-setup.sh in docker container ( Image to be built )
COPY mapr-setup.sh /opt/mapr/installer/docker/
# Run mapr-setup.sh script in docker to download all PACC rpms
RUN /opt/mapr/installer/docker/mapr-setup.sh -r http://package.mapr.com/releases container client 5.2.1 3.0 mapr-client mapr-posix-client-container
# Built the container image

ENTRYPOINT ["/opt/mapr/installer/docker/mapr-setup.sh", "container"]
```

During the Built stage below was logged on the Screen .

```
[root@node112rhel72 abizeroraclemysql]# docker build -t abizer:5.2.1_3.0_OEL_PerconaServer .
Sending build context to Docker daemon 112.6 kB
Step 1/9 : FROM oraclelinux:7.3
 ---> 1046eb4afff7
Step 2/9 : ENV container docker
 ---> Using cache
 ---> 93c6b4c14d49
Step 3/9 : RUN yum -y upgrade && yum install -y curl file net-tools openssl sudo syslinux wget which java-1.8.0-openjdk-devel && yum -q clean all
 ---> Using cache
 ---> 70dc76cbcaf4
Step 4/9 : RUN rpm --import https://www.percona.com/downloads/RPM-GPG-KEY-percona
 ---> Running in 1eb5c8f53cd0
 ---> 130781585df6
Removing intermediate container 1eb5c8f53cd0
Step 5/9 : RUN yum install -y http://www.percona.com/downloads/percona-release/redhat/0.1-4/percona-release-0.1-4.noarch.rpm https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm &&
y update && yum install -y which nc sysbench perl-Digest-MD5 percona-xtrabackup-24 Percona-Server-{client,server,shared,test}-57

---  ---
---  ---
---  ---

Removing intermediate container 04c231aa66f2
Step 9/9 : ENTRYPOINT /opt/mapr/installer/docker/mapr-setup.sh container
 ---> Running in a7ef6b58fa66
 ---> 6e65cc2a18ba
Removing intermediate container a7ef6b58fa66
Successfully built 6e65cc2a18ba
```

Now I could see new image built .

```
[root@node112rhel72 abizeroraclemysql]# docker images
REPOSITORY          TAG                          IMAGE ID            CREATED             SIZE
abizer              5.2.1_3.0_OEL_PerconaServer  6e65cc2a18ba        51 seconds ago      2.73 GB
abizeroracle        5.2.1_3.0_centos7_mysql      9f7d491aa8fc        17 hours ago        1.39 GB
pacc                5.2.1_3.0_centos7_mysql      38b24a3d40cb        20 hours ago        1.03 GB
abizer              5.2.1_3.0_centos7_mysql      5065d2d71c57        20 hours ago        1.03 GB
centos              centos7                      328edcd84f1b        7 days ago          193 MB
maprtech/installer  ubuntu16                     c7cbc778e81e        2 weeks ago         480 MB
oraclelinux         7.3                          1046eb4afff7        4 weeks ago         225 MB
```

Modified Image tag in the script.

```
[root@node112rhel72 abizeroraclemysql]# tail -1 mapr-docker-client.sh

docker run -it $MAPR_DOCKER_ARGS abizer:5.2.1_3.0_OEL_PerconaServer "$@"
```

Verified : I was able to spin up Docker container and it had access to cluster via FUSE and had packages needed .

```
[root@node112rhel72 abizeroraclemysql]# sh mapr-docker-client.sh

Testing for cluster user account...

Enter MapR cluster user name: root
 ...Success

Configuring MapR client ( -c -C 10.10.70.117 -N ClusterNFS4)...

create /opt/mapr/conf/conf.old
Configuring Hadoop-2.7.0 at /opt/mapr/hadoop/hadoop-2.7.0
Done configuring Hadoop
CLDB node list: 10.10.70.117:7222
Zookeeper node list:

...Success

Starting services (mapr-posix-client-container)...

Started service mapr-posix-client-container

...Success

[root@6042662bc958 /]# df -hP /mapr
```

Filesystem Size Used Avail Use% Mounted on
posix-client-container 119G 504M 118G 1% /mapr
[root@6042662bc958 /]# ls /mapr/ClusterNFS4/
a **abizer** **apps** **hbase** **opt** [redacted] test [redacted] **user** **var**
[root@6042662bc958 /]# rpm -qa | grep percona
percona-release-0.1-4.noarch
percona-xtrabackup-24-2.4.8-1.el7.x86_64
[root@6042662bc958 /]# rpm -qa | grep -i percona
percona-release-0.1-4.noarch
Percona-Server-shared-compat-57-5.7.18-16.1.el7.x86_64
percona-xtrabackup-24-2.4.8-1.el7.x86_64
Percona-Server-test-57-5.7.18-16.1.el7.x86_64
Percona-Server-shared-57-5.7.18-16.1.el7.x86_64
Percona-Server-client-57-5.7.18-16.1.el7.x86_64
Percona-Server-server-57-5.7.18-16.1.el7.x86_64

Posted by Abizer Adenwalla at 6:08 PM

No comments:

Post a Comment

Enter your comment...

Comment as: Select profile...

Publish Preview

Newer Post Home Older P

Subscribe to: Post Comments (Atom)