

MBatch Stand Alone Docker Image Updating

Tod Casasent

2019-05-01-1000

1 General Notes

Blue highlighting indicates an important change in machine (DQS server versus HPC) or directory (rsrch1 versus scratch). This is for software developers or maybe DQS-IT when updating the MBatch Stand Alone Docker Image for new versions of the OS or installed applications.

2 Short Version

To get the newest internal updates, build the MBatch and MBatchUtils packages, get the new NGCHM and related files, and the new RStudio Server install and place them in the installations folder (removing old versions) before doing the build.

To get the newest external updates, update the OpenJDK version, remove the local debian:9 image, and rebuild the image.

3 Images Built On

The BEV SA Image is built on the debian:9 Docker image found on Docker Hub <https://hub.docker.com/debian/> and is the official Debian image. The "9" tag points to the newest release of Debian 9.

For updates to Debian, any existing debian:9 tagged images should be removed and the local images rebuilt, resulting in a download of the new Debian Docker image.

This is found in the Dockerfile_template:

```
FROM debian:9
```

4 External Applications and Libraries

4.1 Java OpenSDK and apt-related Utility Packages

The BEV SA Image uses Debian's OpenJDK 8 package for Java and four other "extra" packages: apt-utils, readline-common, procps, and curl.

The apt-get install commands are found in the Dockerfile_template around line 17:

```
RUN apt-get update && \
    apt-get upgrade -f -y && \
    apt-get install -f -y apt-utils readline-common procps curl && \
    apt-get install -f -y openjdk-8-jdk && \
    apt-get clean
```

For updates to Java or these other packages, rebuild the local image to get updates. To change to OpenJDK above 8, update the package name to the new package.

4.2 R 3.4 and related Debian Packages

In order to setup the R-Project repo for R, Debian requires the gnupg package. Note, the gnupg install does not work as part of another install-chain, for unknown reasons.

R also requires other packages such as texlive and various ssh/ssl/ and related packages. R is installed from a 3.4-specific repository.

The install commands are found in the Dockerfile_template starting around line 16:

```
# not sure why, but this does not work as part of the other apt-get chain
RUN apt-get update && \
    apt-get install -f -y gnupg

# Installing R 3.4 on Debian https://cran.r-project.org/bin/linux/debian/
# do apt-get update and install, if this fails with "temporary error", firewall may be blocking
# DNS, check your docker DNS entries
# https://stackoverflow.com/questions/24991136/docker-build-could-not-resolve-archive-ubuntu-com-
apt-get-fails-to-install-a?rq=1
# apt-key adv --keyserver hkps://keyserver.ubuntu.com:80 --recv-key
'E19F5F87128899B192B1A2C2AD5F960A256A04AF'
# apt-key adv --keyserver keys.gnupg.net --recv-key 'E19F5F87128899B192B1A2C2AD5F960A256A04AF'
#### apt-key adv --keyserver hkps://keyserver.ubuntu.com:80 --recv-key
'E19F5F87128899B192B1A2C2AD5F960A256A04AF' && \
# The above is constantly failing for one reason or another. Use saved key in file
####
# gdebi-core is used for installing RStudio later
COPY installations/jranke.key /jranke.key
RUN echo 'deb http://cran.r-project.org/bin/linux/debian stretch-cran34/' >>
/etc/apt/sources.list && \
    apt-key add /jranke.key && \
    rm /jranke.key && \
    apt-get update && \
    apt-get install -f -y texlive && \
    apt-get install -f -y wget libssh2-1 libcairo2-dev libxt-dev curl libcurl4-openssl-dev
libssl1.0-dev libxml2-dev && \
    apt-get install -f -y git openssh-client && \
    apt-get install -f -y -t stretch-cran34 littler r-cran-littler r-base r-base-dev r-
recommended && \
    apt-get install -f -y gdebi-core && \
    apt-get clean

# register Java 8 with R
RUN R CMD javareconf
```

For updates to Debian packages, rebuild the local image to get updates. To change R versions, update the package name to the new package. The current version of R is the 3.5 chain, whose deb line appears as shown below:

```
'deb http://cran.r-project.org/bin/linux/debian stretch-cran35/'
```

4.3 RStudio Server Install

The Image uses RStudio Server, downloaded beforehand from the RStudio homepage: <https://www.rstudio.com/products/rstudio/download-server/> and placed in the installations sub-directory.

The install commands are found in the Dockerfile_template around line 64:

```
# create and setup docker_rstudio
# also, set permissions and ownerships on internal docker directories
RUN mkdir /home/docker_rstudio && \
    useradd -l -s /bin/bash -d /home/docker_rstudio -u <USERID> docker_rstudio && \
    echo "docker_rstudio:docker_rstudio" | chpasswd && \
    chown -R docker_rstudio:docker_rstudio /home/docker_rstudio
```

5 Internal (MDA) Applications and Libraries

5.1 MBatch, MBatchUtils, and NGCHM and tsvio R packages

MBatch and MBatchUtils packages are provided as tar.gz files. They are build via R CMD build and copied to the installations directory. The tsvio R package is installed via tar.gz and is provided by Bradley for building NGCHMs.

To update the pages, for MBatch and MBatchUtils, rebuild and copy the tar.gz files to the installations directory. This is done as part of the HPC scripts.

For NGCHM and TSVIO packages, rebuild the images, the newest GitHub releases are used.

In both cases, rebuild the images.