DOCKER IMAGE BUILDING FOR CAN DEPLOYMENT

Version 5.0

AUGUST 10, 2020

AVANSEUS TECHNOLOGIES PVT. LTD.

REVISION HISTORY

Version	Date	Change description	Created by	Updated by	Reviewed by
V 5.0	Aug, 2020	-	Naveen	Sandeep	Chiranjib

TABLE OF CONTENTS

1. Focus

This document focuses on building an image of CAN application & all other supporting applications. This document is mainly for the Avanseus team who builds the Docker image before distributing. The Docker images created for CAN applications to run are as follows.

1. MongoDB image
2. LDAP image (With OpenDS and one default user)
3. VBI python module image
4. CAN application image (Master node)
5. CAN application image (Slave node)

Please download and use the quickstart package for building these images from this link. Un-tar and you'll find directories with important artifacts needed for building images.

2. Image Creation

1. MongoDB Image

DockerHub repository by default provides a MongoDB image. We will use that to create the setup.

Command to download MongoDB-3.4.6 image:

```
sudo docker pull mongo:3.4.6
```

2. LDAP Image

Go to the "LDAP" directory of the quickstart package. The LDAP package already consists of a LDAP user whose details will be shared separately.

DockerHub repository does not provide OpenDS LDAP image. Hence, we need to create one from scratch. We will reuse the Ubuntu image provided by DockerHub repository to run LDAP in it.

The steps to create an OpenDS LDAP image are as follows:

```
FROM ubuntu:18.04

RUN groupadd --gid 5000 appuser \
   && useradd --home-dir /home/appuser --create-home --uid 5000 \
      --gid 5000 --shell /bin/sh --skel /dev/null appuser

COPY ./jdk-6u45-linux-x64.bin /home/appuser/jdk-6u45-linux-x64.bin
COPY ./OpenDS-2.2.1 /home/appuser/OpenDS-2.2.1/

WORKDIR /home/appuser/
RUN ./jdk-6u45-linux-x64.bin

RUN chown -R appuser:appuser /home

USER appuser

RUN echo "export PATH=$PATH:/home/appuser/jdk1.6.0_45/bin/" >> ~/.bashrc
RUN rm /home/appuser/OpenDS-2.2.1/lib/set-java-home
ENV PATH="${PATH}:/home/appuser/jdk1.6.0_45/bin/"


CMD ["/home/appuser/OpenDS-2.2.1/bin/start-ds","--nodetach"]
```

Run the below command to build the image.

```
docker build . -t ldapimage:1
```

3. VBI Python Module Image

Go into the "VBI" directory of the quickstart package. Update the latest python script file in the directory with name "docker_socket_server_json.py".
DockerHub repository by default provides a Python image. We will use that to create the setup.

Create a file "Dockerfile" and add the following instructions or reuse the existing one.

```
FROM python:3.7.3

RUN pip3.7 install nltk numpy scipy pytorch-pretrained-bert --no-cache-dir

RUN groupadd --gid 5000 appuser \
   && useradd --home-dir /home/appuser --create-home --uid 5000 \
      --gid 5000 --shell /bin/sh --skel /dev/null appuser


COPY ./PreTrainedBert/uncased_L-12_H-768_A-12/ /home/appuser/bert-baseuncased/
COPY ./PreTrainedBert/Model/ /home/appuser/bert-base-uncased/
COPY ./docker_socket_server_json.py /home/appuser/docker_socket_server_json.py

RUN chown -R appuser:appuser /home/appuser

USER appuser

WORKDIR /home/appuser/

CMD ["python3.7","-u","docker_socket_server_json.py"]
```

Run the below command to build the image.

```
docker build . -t pyvbi:v1
```

4. CAN Application Image (MASTER)

Go into the directory named "tomcats" in the quickstart package. Please update the directory with the latest CAN & CAS war. Update the "catalina.properties" file with the encryption key. Also update the "config.properties" based on the configurations. We shall use the Ubuntu 18.04 docker image to build CAN application image.

Create a file with name "Dockerfile" and add the following instructions or reuse the existing one.

```
FROM ubuntu:18.04

RUN apt-get update && apt-get install -y  software-properties-common && apt-get install -y
openjdk-8-jdk

#install gosu
RUN apt-get -y install gosu

RUN groupadd --gid 5000 appuser \
   && useradd --home-dir /home/appuser --create-home --uid 5000 \
      --gid 5000 --shell /bin/sh --skel /dev/null appuser


RUN mkdir -p /data/workspace
ADD entrypoint.sh /data/workspace/entrypoint.sh
ADD apache-tomcat-8.5.6 /data/workspace/apache-tomcat-8.5.6
RUN mv /data/workspace/apache-tomcat-8.5.6 /data/workspace/tomcatCAS
RUN cp -rf /data/workspace/tomcatCAS /data/workspace/tomcatCAN
ADD serverCAN.xml /data/workspace/tomcatCAN/conf/server.xml
ADD serverCAS.xml /data/workspace/tomcatCAS/conf/server.xml

ADD CAN.war /data/workspace/tomcatCAN/webapps/CAN.war
ADD CAS.war /data/workspace/tomcatCAS/webapps/CAS.war

ADD security.policy /data/workspace/tomcatCAN/conf/security.policy

ADD nasmount /data/workspace/nasmount

ADD tomcat_startup.sh /data/workspace/

RUN mkdir -p /data/workspace/logs
RUN chown -R appuser:appuser /data
RUN chmod -R 755 /data

RUN chmod +x /data/workspace/tomcatCAN/bin/*.sh
RUN chmod +x /data/workspace/tomcatCAS/bin/*.sh
RUN chmod +x /data/workspace/tomcat_startup.sh

WORKDIR /data/workspace/
ENTRYPOINT ["/data/workspace/entrypoint.sh"]
CMD ["/bin/bash"]
```

Please create a file with name “entrypoint.sh” in the same location as Dockerfile and add the following:

```
#!/bin/bash
set -e
chown -R appuser /data/workspace/logs
gosu appuser sh tomcat_startup.sh
exec /bin/bash
```

Run the below command to build the image.

```
sudo docker build . -t canapp:1
```

5. CAN Application Image (SLAVE)

Go into the directory named “slaves” in the quickstart package. Please update the directory with the latest CAN & CAS war. Update the “catalina.properties” file with the encryption key. Also update the “config.properties” based on the configurations. We shall use the Ubuntu 18.04 docker image to build CAN application image.

Create a file with name “Dockerfile” and add the following instructions or reuse the existing one.

```
FROM ubuntu:18.04

RUN apt-get update && apt-get install -y  software-properties-common && apt-get install -y
openjdk-8-jdk

RUN apt-get -y install gosu

RUN groupadd --gid 5000 appuser \
   && useradd --home-dir /home/appuser --create-home --uid 5000 \
      --gid 5000 --shell /bin/sh --skel /dev/null appuser

RUN mkdir -p /data/workspace
ADD apache-tomcat-8.5.6 /data/workspace/apache-tomcat-8.5.6
RUN mv /data/workspace/apache-tomcat-8.5.6 /data/workspace/tomcatCAN

ADD entrypoint.sh /data/workspace/entrypoint.sh

ADD CAN.war /data/workspace/tomcatCAN/webapps/CAN.war

ADD catalina.properties /data/workspace/tomcatCAN/conf/catalina.properties

ADD setenv.sh /data/workspace/tomcatCAN/bin/setenv.sh
ADD security.policy /data/workspace/tomcatCAN/conf/security.policy

ADD nasmount /data/workspace/nasmount

ADD tomcat_startup.sh /data/workspace/

RUN chmod -R 755 /data/
RUN mkdir -p /data/workspace/logs
RUN chown -R appuser:appuser /data
CMD chmod +x /data/workspace/tomcatCAN/bin/catalina.sh
WORKDIR /data/workspace/
ENTRYPOINT ["/data/workspace/entrypoint.sh"]
CMD ["/bin/bash"]
```

Please create a file with name “entrypoint.sh” in the same location as Dockerfile and add the following:

```
#!/bin/bash
set -e
chown -R appuser /data/workspace/logs
gosu appuser sh tomcat_startup.sh
exec /bin/bash
```

Run the below command to build the image.

```
sudo docker build . -t canslaveapp:1
```