

Creating Your Own BRIDGE Docker Image



To install the BRIDGE you can use the Docker image delivered by us but you can also create a BRIDGE Docker image of your own to fulfill special needs. You can do this from scratch, or you can use the delivered BRIDGE Docker image and change it according to your needs.

The BRIDGE Docker image is built in two steps:

1. Creation of a Debian basic image containing an Oracle and an SQL Server instant client.
2. Creation of the actual BRIDGE image on base of the Debian basic image.

On this Page:

- [Required Files to Create a BRIDGE Docker Image](#)
- [The Debian Basic Docker File Explained](#)
- [The BRIDGE Docker File Explained](#)

Required Files to Create a BRIDGE Docker Image

If you want to create the BRIDGE Docker image you need the following files:

File	In ZIP	Usage
Dockerfile Debian Basic	✓	The Docker file that rules the build of a Debian Basic image.
Dockerfile BRIDGE	✓	The Docker file that rules the build of a BRIDGE image on base of the Debian Basic image.
BridgeInstaller-linux-64-x.x.x.jar	✗	A BRIDGE installer.
guilessinstaller.properties	✓	A properties file for GUI-less installation.
proxies.xml	✓	A file containing proxy definitions. You need it only if you use sudo.
server.cfg	✓	A file containing the configuration for MySQL and SQL Server client libraries.
sudo_httpd	✓	A file containing sudo and httpd configurations.

You can download a ZIP archive containing all files that are marked with **In ZIP** from our [download page](#).

Related Pages:

- [Installing the BRIDGE Image to a Docker Container](#)

Downloads:

[Docker files ZIP](#)

The Debian Basic Docker File Explained

In a first step, a Debian basic image is created that is used later on as a base for the actual BRIDGE image. Expand the code section below to see and copy & paste the Docker file.

Debian Basic Docker File
<pre> ARG INSTANTCLIENT_VERSION ARG SQLSERVER_VERSION FROM oraclelinux:7-slim AS oracleclient ARG INSTANTCLIENT_VERSION RUN curl -o /etc/yum.repos.d/public-yum-ol7.repo https://yum.oracle.com /public-yum-ol7.repo && \ yum-config-manager --enable ol7_oracle_instantclient && \ yum -y install oracle-instantclient\$INSTANTCLIENT_VERSION-basiclite && \ rm -rf /var/cache/yum FROM debian:buster-slim ARG INSTANTCLIENT_VERSION ARG SQLSERVER_VERSION ENV LANG=C.UTF-8 ENV TZ="Europe/Zurich" RUN apt-get update -q \ && apt-get install -qq sudo procps curl net-tools unzip vim mariadb- client gnupg apt-transport-https \ && curl https://packages.microsoft.com/keys/microsoft.asc sudo apt- key add - \ && echo "deb https://packages.microsoft.com/debian/10/prod/ buster main" > /etc/apt/sources.list.d/msprod.list \ && apt-get update -q \ && ACCEPT_EULA=y apt-get install -qq msodbcsql17: amd64=\$SQLSERVER_VERSION mssql-tools:amd64=\$SQLSERVER_VERSION \ && rm -rf /var/lib/apt/lists/* \ && groupadd -r bridge \ && useradd -r --create-home -g bridge bridge \ && (cd /etc; ln -f -s /usr/share/zoneinfo/\$TZ localtime) RUN mkdir /opt/oracle COPY --chown=root:root --from=oracleclient /usr/lib/oracle /\$INSTANTCLIENT_VERSION/client64/lib /opt/oracle ENV ORACLE_HOME=/opt/oracle </pre>

Code	Description
Preparations - Get an Oracle Instant Client	
<pre> ARG INSTANTCLIENT_VERSION ARG SQLSERVER_VERSION </pre>	Provide the actually used versions, e.g. INSTANTCLIENT_VERSION=19.6 and SQLSERVER_VERSION=17.5.2.1-1 with the call.
<pre> FROM oraclelinux:7-slim AS oracleclient </pre>	Download Oracle Linux image.
<pre> RUN curl -o /etc/yum.repos.d/public-yum-ol7. repo https://yum.oracle.com/public-yum-ol7.repo && \ yum-config-manager --enable ol7_oracle_instantclient && \ yum -y install oracle- instantclient\$INSTANTCLIENT_VERSION-basiclite && \ rm -rf /var/cache/yum </pre>	Install instant client to Oracle Linux image.
Creating the Debian Basic Image	

FROM debian:buster-slim	Download Debian buster slim image.
ARG INSTANTCLIENT_VERSION ARG SQLSERVER_VERSION	See further above.
ENV LANG=C.UTF-8 ENV TZ="Europe/Zurich"	Set language and timezone in the system environment.
ENV LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:\$ORACLE_HOME ENV PATH=\$PATH:\$JAVA_HOME/bin:\$ORACLE_HOME	Configure the Oracle Database client library.
RUN apt-get update -q \ && apt-get install -qq sudo procps curl net-tools unzip vim mariadb-client gnupg apt- transport-https \ 	Install necessary packages.
&& curl https://packages.microsoft.com/keys /microsoft.asc sudo apt-key add - && echo "deb https://packages.microsoft.com /debian/10/prod/ buster main" > /etc/apt /sources.list.d/msprod.list \ && apt-get update -q \ && ACCEPT_EULA=y apt-get install -qq msodbcsql17:amd64=\$SQLSERVER_VERSION mssql- tools:amd64=\$SQLSERVER_VERSION \ 	Install MS SQL Server.
&& rm -rf /var/lib/apt/lists/* \ 	
&& groupadd -r bridge \ && useradd -r --create-home -g bridge bridge \ 	As the BRIDGE should not be run as root, create a group bridge and a user bridge .
&& (cd /etc; ln -f -s /usr/share/zoneinfo /\$TZ localtime)	Set timezone.
RUN --mount=type=secret,id=nexuspass -- mount=type=secret,id=nexususer export NEXUS_PASSWORD=\$(cat /run/secrets/nexuspass) && export NEXUS_USER=\$(cat /run/secrets /nexususer) \ 	Get login information for nexus
&& mkdir /tmp/db2_odbc_cli \ 	
&& curl -o /tmp/db2_odbc_cli/db2_odbc_cli. tar.gz -u \$NEXUS_USER:\$NEXUS_PASSWORD https://gitlab.scheer-group.com:8080 /repository/maven2_lib-release-local/ch/e2e /runtime/db2_odbc_cli/\$DB2_ODBC_CLI_VERSION /db2_odbc_cli-\$DB2_ODBC_CLI_VERSION.tar.gz \ 	Download driver from nexus
&& (cd /tmp/db2_odbc_cli; tar -xvzf db2_odbc_cli.tar.gz) \ 	Unpack driver
&& (cp -r /tmp/db2_odbc_cli/odbc_cli /opt /odbc_cli; rm -rf /tmp/db2_odbc_cli)	Copy driver to BRIDGE location and delete temporary files.
RUN mkdir /opt/oracle COPY --chown=root:root --from=oracleclient /usr /lib/oracle/\$INSTANTCLIENT_VERSION/client64 /lib /opt/oracle	Copy instantclient from Oracle Linux image.
COPY --chown=bridge:bridge proxies.xml /opt /bridge_data/proxies/	Configure proxies.xml. Only needed if sudo package is installed.

COPY --chown=bridge:bridge server.cfg /opt /bridge_data/server.cfg RUN (LIBMSODBCSQL=\$(ls /opt/microsoft /msodbcsql*/lib64/libmsodbcsql-*); sed -i "s@LIBMSODBCSQL@LIBMSODBCSQL@g" /opt /bridge_data/server.cfg	Configure MySQL and MS SQL Server client libraries.
ENV ORACLE_HOME=/opt/oracle	Set environment for Oracle home directory.

The BRIDGE Docker File Explained

In a second step, the actual BRIDGE image is created on base of the previously created Debian basic image. Expand the code section below to see and copy & paste the Docker file.

Docker File
<pre> FROM adoptopenjdk/openjdk11:jre AS install ARG consoleProtocol COPY guilessinstaller.properties /tmp/guilessinstaller.properties COPY BridgeInstaller-linux-64-*.jar /tmp/BridgeInstaller.jar RUN groupadd -r bridge \ && useradd -r -g bridge bridge \ && java -Dconsole.protocol=\${consoleProtocol:-https} -jar /tmp /BridgeInstaller.jar -guiless -installproperties /tmp/guilessinstaller. properties -debuglevel info \ && sed s/host=\"localhost\" // -i /opt/bridge_data/domain/nodes.xml FROM debian_basic:latest EXPOSE 8080/tcp ENV JAVA_HOME=/opt/bridge_prog/j2re-11.0.7/linux-64 ENV LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:\$ORACLE_HOME ENV PATH=\$PATH:\$JAVA_HOME/bin:\$ORACLE_HOME COPY sudo_httpd /etc/sudoers.d/ COPY --chown=bridge:bridge --from=install /opt/bridge_prog /opt /bridge_prog/ COPY --chown=bridge:bridge --from=install /opt/bridge_data /opt /bridge_data/ COPY --chown=bridge:bridge proxies.xml /opt/bridge_data/proxies/ COPY --chown=bridge:bridge server.cfg /opt/bridge_data/server.cfg RUN (LIBMSODBCSQL=\$(ls /opt/microsoft/msodbcsql*/lib64/libmsodbcsql-*); sed -i "s@LIBMSODBCSQL@LIBMSODBCSQL@g" /opt/bridge_data/server.cfg) USER bridge CMD ["/opt/bridge_prog/bin/e2e_console.sh", "docker" </pre>

Code	Description
Preparations	
FROM adoptopenjdk/openjdk11:jre AS install	Create a Docker image in stall . You need Java 11 to install the BRIDGE.
ARG consoleProtocol	Provide the protocol of the BRIDGE UI with the call. Can be either http or https .

COPY guilessinstaller.properties /tmp /guilessinstaller.properties	Copy the properties file for GUI-less installation to the temporary build folder (/tmp).
COPY BridgeInstaller-linux-64-*.jar /tmp /BridgeInstaller.jar	Copy the BRIDGE installer file to the temporary build folder (/tmp).
RUN groupadd -r bridge \ && useradd -r -g bridge bridge \ && java -Dconsole.protocol=\${consoleProtocol:-https} -jar /tmp/BridgeInstaller.jar -guiless -installproperties /tmp/guilessinstaller.properties -debuglevel info \ && sed s/host=\"localhost\" // -i /opt/bridge_data/domain/nodes.xml	As the BRIDGE should not be run as root, create a group bridge and a user bridge . Start the installation.
Creating the BRIDGE Image	
FROM debian_basic:latest	Use Debian basic as base for the BRIDGE image.
EXPOSE 8080/tcp	Expose the BRIDGE port.
ENV JAVA_HOME=/opt/bridge_prog/j2re-<version>/linux-64	Set environment so that the system uses the Java coming with the BRIDGE.
ENV LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:\$ORACLE_HOME ENV PATH=\$PATH:\$JAVA_HOME/bin:\$ORACLE_HOME	Configure the Oracle Database client library.
COPY sudo_httpd /etc/sudoers.d/	Configure sudo_http. Only needed if sudo package is installed.
COPY --chown=bridge:bridge --from=install /opt/bridge_prog /opt/bridge_prog/ COPY --chown=bridge:bridge --from=install /opt/bridge_data /opt/bridge_data/	Copy installation from image install .
COPY --chown=bridge:bridge proxies.xml /opt/bridge_data/proxies/	Configure proxies.xml. Only needed if sudo package is installed.
COPY --chown=bridge:bridge server.cfg /opt/bridge_data/server.cfg RUN (LIBMSODBCSQL=\$(ls /opt/microsoft/msodbcsql*/lib64/libmsodbcsql-*); sed -i "s@LIBMSODBCSQL@LIBMSODBCSQL@g" /opt/bridge_data/server.cfg	Configure MySQL and MS SQL Server client libraries.
USER bridge	Specify the user the BRIDGE should use to run.
CMD ["/opt/bridge_prog/bin/e2e_console.sh", "docker"]	Call BRIDGE standard start script with parameter docker .