

Creating a Docker image

To build a Docker image, you need to create a Dockerfile. A Dockerfile is a file that contains instructions on how to build the Docker image. By default, the file is named *Dockerfile* and this naming is used in the instructions below. To learn more about Docker, see <https://docs.docker.com/get-started/>.



Prerequisites

In addition to a Dockerfile, the following files are required to build the Docker image:

- *myapp.war* - a deployable archive of the desired web application, e.g., *webapp.war*, *collaborator.war*, or *admin.war*.
- `<install_root>/WebAppPlatform/shared/conf/logback.xml` - the logging configuration file. To learn how to configure data logging, see [Configuring data logging](#).
- `<install_root>/WebAppPlatform/shared/conf/webappplatform.properties` - the file with Web Application Platform configuration properties.
- You may also need additional files, like email or document export templates that need to be copied to the web container.

Here is a sample a Dockerfile that can be used to build the Docker image:

```
FROM tomcat:9-jdk17-openjdk

ARG TOMCAT_HOME=/usr/local/tomcat

COPY logback.xml ${TOMCAT_HOME}/shared/conf/
COPY webappplatform.properties ${TOMCAT_HOME}/shared/conf/
COPY myapp.war ${TOMCAT_HOME}/webapps/

RUN sed -i "s|shared\.loader=*$|shared\.loader=\"\${catalina.base}/shared/conf\"|" "${TOMCAT_HOME}/conf/catalina.properties"

EXPOSE 8080
CMD ["catalina.sh", "run"]
```

Instructions in the Dockerfile may be written in several ways depending on your particular needs and preferences. The table below lists script statements you can use with descriptions and the indication whether the script part is required or optional.

Script statement	Required or optional	Description
FROM tomcat:9-jdk17-openjdk	Required	In the FROM part where the base image needs to be specified, Tomcat should be selected as the base image and its version has to be compatible with the version of the web application code.
ARG TOMCAT_HOME=/usr/local/tomcat	Optional	An argument that is used as an intermediary variable to store the Tomcat home location. The location may be different depending on a particular base image used in the FROM part. This argument is introduced for convenience since further instructions reference the Tomcat home location multiple times.
COPY logback.xml \${TOMCAT_HOME}/shared/conf/	Optional (recommended)	Since Web Application Platform uses logback as the logging framework, there should be its configuration file available in the classpath. This command copies the logging configuration into the folder that will be included in the classpath. Note that it is possible to use other names for the configuration file. In this case, the script should be updated to reference another file. It is possible to omit the configuration file but it is not recommended.
COPY webappplatform.properties \${TOMCAT_HOME}/shared/conf/	Optional	This command copies the properties file to the folder that is present in the classpath. If it is not copied, then an alternative solution must be used to reference the properties file. To learn about alternative options, see Passing properties .
COPY myapp.war \${TOMCAT_HOME}/webapps/	Required	This command copies the web application to the Tomcat deployment folder. Note that you can put multiple web applications in same docker image. To do that, add a separate command for each .war file.
RUN sed -i "s shared\.loader=*\$ shared\.loader=\"\\${catalina.base}/shared/conf\" " "\${TOMCAT_HOME}/conf/catalina.properties"	Required if any additional files need to be in the classpath	This command updates the Tomcat configuration to include <code>\${TOMCAT_HOME}/shared/conf</code> in the classpath.

EXPOSE 8080	Required	The value should correspond to the one which is used by Tomcat to deploy the web application. You can use port 8080 or 8443 if SSL is needed.
CMD ["catalina.sh", "run"]	Required	A command used to run the Tomcat server.

Here is a sample command to build the Docker image:

```
docker build -t <image_name:version> .
```

The provided build command will work only if the Dockerfile script and all the files referenced in the script (in this case *myapp.war*, *logback.xml*, and *webappplatform.properties*) are stores in the current directory. To build Docker images using non-default file locations, consult the Docker documentation.

When a Docker image is successfully built, you can run the Docker container on port 8080 by executing the following command:

```
docker run -p 8080:8080 <image_name:version>
```