# Building and Running DSpace in Docker

**Content has moved**

See https://github.com/DSpace-Labs/DSpace-Docker-Images/blob/master/README.md



The following notes describe my experimentation building and running DSpace 6 within Docker on my Mac.

As it is configured, it takes 15 minutes for tomcat to start, so this is not a recommended development approach.  It was a good exercise in learning more about Docker.

## Pre-requisites

- Install Docker on your desktop
- Create a DSpace deployment directory.  In this example ~/dspace will be used.

## Building DSpace

If you have a development desktop configured, it is likely that you would also have a JDK, Maven, and Ant on that machine; so this process is probably not needed but it provides an introduction to running common DSpace development tasks in Docker.

### Retrieve the latest DSpace 6 code

- git clone https://github.com/DSpace/DSpace.git
- cd DSpace
- git checkout dspace-6_x

### Define a local config file (local.cfg)

Because Docker allows a client application to map ports and paths within a Docker container, these instructions will rely on the default settings in dspace.cfg including the default root path of /dspace.

In the root directory, create a local.cfg file containing the following setting

**[dspace-src]/local.cfg**

```
db.url = jdbc:postgresql://dspacedbhost:5432/dspace
```

### Run Maven Build within a Docker Container

The following code will run the maven container defined on Dockerhub.

- Within that container "mvn clean install" will be run.
- The working directory will be /opt/maven
- The -v $(pwd):/opt/maven maps the user's current directory to /opt/maven
- The results will persist on the user's desktop in the current directory

**Run maven in Docker**

```
docker run -it --rm -v "$(pwd)":/opt/maven -w /opt/maven maven mvn clean install
```

## Preparing to run ant

The final steps of the DSpace ant build verify that a postgres database is available.  Before running the ant step, we will start postgres in a Docker container.

## Starting Postgres in a Docker container

The following code will run the terrywbrady/dspacedb container defined on Dockerhub.

- This container was built from the source code defined in https://github.com/DSpace-Labs/dspace-dev-docker/tree/master/postgres
- This container extends the postgres Docker image and ensures that the pgcrypto extension is installed.
    - This is ensured by adding an init script to the Docker file.

**terrywbrady/dspacedb – Dockerfile**

```
# From https://github.com/DSpace-Labs/dspace-dev-docker/tree/master/postgres
FROM postgres


ENV POSTGRES_DB dspace
ENV POSTGRES_USER dspace
ENV POSTGRES_PASSWORD dspace


COPY install-pgcrypto.sh /docker-entrypoint-initdb.d/
```

**terrywbrady/dspacedb – pgcrypto.sh**

```
#!/bin/bash
set -e


psql -v ON_ERROR_STOP=1 --username "$POSTGRES_USER" <<-EOSQL
  -- Create a new schema in this database named "extensions" (or whatever you want to name it)
  CREATE SCHEMA extensions;
  -- Enable this extension in this new schema
  CREATE EXTENSION pgcrypto SCHEMA extensions;
  -- Update your database's "search_path" to also search the new "extensions" schema.
  -- You are just appending it on the end of the existing comma-separated list.
  ALTER DATABASE dspace SET search_path TO "\$user",public,extensions;
  -- Grant rights to call functions in the extensions schema to your dspace user
  GRANT USAGE ON SCHEMA extensions TO $POSTGRES_USER;
EOSQL
```

Notable parameters

- The -d flag runs the service in a detached mode allowing it to run outside of the terminal.
- The -p 5432:5432 exposes the postgres port 5432 as 5432
- The --name dspacedbhost runs this container in a Dockerhost name "dspacedbhost"
    - **Question: I had hoped to be able to treat this container as a service on localhost:5432, but I was not successful.**

**Start postgres in Docker**

```
docker run -it --rm -d -p 5432:5432 --name dspacedbhost terrywbrady/dspacedb
```

## Verifying that postgres is running

The following code will use a postgres Docker image to run the "psql" client program to connect to a postgres as user "dspace".

```
The "--link dspacedbhost:dspacedbhost" links the hostname of the postgres server container to the container running psql.
```

**Run psql within a Docker container**

```
docker run -it --rm --link dspacedbhost:dspacedbhost postgres psql -h dspacedbhost -U dspace
```

### Sample output

```
Mac-mini:DSpace terry$ docker run -it --rm --link dspacedbhost:localhost postgres psql -h localhost -U dspace
Password for user dspace:
psql (10.2 (Debian 10.2-1.pgdg90+1))
Type "help" for help.


dspace=# select gen_random_uuid();
           gen_random_uuid
--------------------------------------
 f0035e0a-d60a-461e-a379-f840d14abbdc
(1 row)
```

## Running ant

I attempted to run ant using an existing image such as webratio/ant, but I had difficulty completing the database initialization steps.

Therefore, I created the terrywbrady/dspace-docker-ant image using the following docker file.

**terrywbrady/dspace-docker-ant – Dockerfile**

```
FROM openjdk
EXPOSE 4403 8000 8080 9876 22


ENV ANT_VERSION 1.10.2
ENV ANT_HOME /opt/ant-$ANT_VERSION
ENV PATH $ANT_HOME/bin:$PATH


RUN mkdir $ANT_HOME && \
    wget -qO- "https://www.apache.org/dist/ant/binaries/apache-ant-$ANT_VERSION-bin.tar.gz" | tar -zx --strip-
components=1 -C $ANT_HOME


RUN mkdir /dspace
RUN mkdir /dspace/upload
```

Using this Docker image, the DSpace build can be completed.

- The '-v "$(pwd)":/opt/maven' parameter makes the current directory available again as /opt/maven
- The '-v ~/dspace:/dspace' makes the local ~/dspace directory available as the default DSpace install directory /dspace
- -w /opt/maven/dspace/target/dspace-installer sets the ant working directory
- --link dspacedbhost:dspacedbhost makes the postgres image available as hostname dspacedbhost
- "ant update" is run on the command line

**Running ant within Docker**

```
docker run -it --rm -v "$(pwd)":/opt/maven -v ~/dspace:/dspace -w /opt/maven/dspace/target/dspace-installer --
link dspacedbhost:dspacedbhost terrywbrady/dspace-docker-ant ant update
```

### Launching tomcat

While it is likely impractical to run the build steps in Docker containers, there is definite value in running tomcat inside of a Docker container.

As configured, it took 15+ minutes to launch tomcat, so some additional tuning will be needed.

This process uses the official tomcat image

- The '-v ~/dspace:/dspace' makes the local ~/dspace directory available as the default DSpace install directory /dspace
- -v ~/dspace/webapps:/usr/local/tomcat/webapps makes the dspace webapps directory available as the tomcat webapps directory
- --link dspacedbhost:dspacedbhost makes the postgres image available as hostname dspacedbhost
- -p 8080:8080 exposes the container's port 8080 as localhost 8080

**Launching tomcat within Docker**

```
docker run -it --rm -d -v ~/dspace:/dspace -v ~/dspace/webapps:/usr/local/tomcat/webapps --link dspacedbhost:
dspacedbhost -p 8080:8080 tomcat:8.0
```

### Open a web browser

After the initialization is complete, open http://localhost:8080/xmlui to view DSpace.

## Simple Config - running maven and ant on desktop (MacOS)

### Setup - local.cfg

- Create /installs/dspace locally
- Do not change db.url

## Docker Commands

**Start postgres in Docker**

```
docker run -it --rm -d -p 5432:5432 --name dspacedbhost terrywbrady/dspacedb
docker run -it --rm -v ~/dspace:/dspace -v ~/dspace/webapps:/usr/local/tomcat/webapps -p 8080:8080 tomcat:8.0
```

# Simple Config - running maven and ant on desktop (Windows10)

The DSpace ant build needs to run on the machine where the install will take place.  Therefore, the ant build should run in a Docker image that runs on the same Docker network

- Before cloning the repo on Windows
    - git config --global core.autocrlf false
    - git config --global core.eol lf
- DSpace Branch: dspace-6_x
- local.cfg
    - dspace.dir=/dspace
    - db.url = jdbc:postgresql://dspacedbhost:5432/dspace
    - dspace.hostname = dspacetomcat
    - dspace.baseUrl = http://dspacetomcat:8080
- dspace-src:
    - desktop: c:\Users\[user]\Documents\GitHub\DSpace
- dspace-install:
    - desktop: c:\installs\dspace
    - docker image: /dspace

```
docker network create dspacenet

docker volume create pgdataD6

docker run -it -d --network dspacenet -p 5432:5432 --name dspacedbhost -v pgdataD6:/var/lib/postgresql/data
terrywbrady/dspacedb

docker volume create dspaceD6

docker run -it --rm --network dspacenet -v ${PWD}/dspace/target/dspace-installer:/installer -v dspaceD6:/dspace
-w /installer terrywbrady/dspace-docker-ant ant update clean_backups

docker run -it --rm --network dspacenet -v dspaceD6:/dspace -v dspaceD6/dspace/webapps/solr:/usr/local/tomcat
/webapps/solr:ro -v dspaceD6/dspace/webapps/xmlui:/usr/local/tomcat/webapps/xmlui:ro -p 8080:8080 -e JAVA_OPTS=-
Xmx2000m --name dspacetomcat tomcat:8

docker run -it --rm --network dspacenet -v dspaceD6:/dspace -p 8080:8080 -e JAVA_OPTS=-Xmx2000m --name
dspacetomcat tomcat:8


docker exec -it --detach-keys "ctrl-p" dspacetomcat /bin/bash

docker exec -it --detach-keys "ctrl-p" dspacedbhost psql -U dspace
```