

Past DSpace Docker Implementation Conversations

OUTDATED / OBSOLETE: This is not being actively maintained. Please see [DSpace](#) and [Docker](#) instead.

Proposal: Support Multiple JDK Versions

- <https://github.com/DSpace-Labs/DSpace-Docker-Images/issues/57>
- <https://github.com/DSpace/DSpace/pull/2214>

Prior Dockerfile Names

branch	JDK	tag	Dockerfile
master	8	master	./Dockerfile
dspace-6_x	8	dspace-6_x	./Dockerfile
dspace-5_x	8	dspace-5_x	./Dockerfile
dspace-4_x	8	dspace-4_x	./Dockerfile

New Dockerfile Names

branch	JDK	tag	old tag name	Dockerfile	Maven	ANT	Tomcat	Note
master	8	dspace-7_x-jdk8	master	/Dockerfile.jdk8	maven:latest	1.10.5	tomcat:8	
master	8	dspace-7_x-jdk8-test		/Dockerfile.jdk8-test	maven:latest	1.10.5	tomcat:8	testing optimizations*
dspace-6_x	8	dspace-6_x-jdk8	dspace-6_x	/Dockerfile.jdk8	maven:latest	1.10.5	tomcat:8	
dspace-6_x	8	dspace-6_x-jdk8-test		/Dockerfile.jdk8-test	maven:latest	1.10.5	tomcat:8	testing optimizations*
dspace-5_x	8	dspace-5_x-jdk8	dspace-5_x	/Dockerfile.jdk8	maven:latest	1.10.5	tomcat:8	DSpace 5x/Tomcat8 not recommended for usage
dspace-5_x	8	dspace-5_x-jdk8-test		/Dockerfile.jdk8	maven:latest	1.10.5	tomcat:8	testing optimizations*
dspace-5_x	7	dspace-5_x-jdk7		/Dockerfile.jdk7	maven:3.5.4-jdk-7 or maven:latest	1.9.13	tomcat:7.0.90-jre7	
dspace-5_x	7	dspace-5_x-jdk7-test		/Dockerfile.jdk7-test	maven:3.5.4-jdk-7 or maven:latest	1.9.13	tomcat:7.0.90-jre7	testing optimizations*
dspace-4_x	7	dspace-4_x-jdk7		/Dockerfile.jdk7	maven:3.5.4-jdk-7 or maven:latest	1.9.13	tomcat:7.0.90-jre7	(The old tag dspace-4_x had been built with jdk8)
dspace-4_x	7	dspace-4_x-jdk7-test		/Dockerfile.jdk7-test	maven:3.5.4-jdk-7 or maven:latest	1.9.13	tomcat:7.0.90-jre7	(The old tag dspace-4_x had been built with jdk8)

Image Variants for Testing

Note: the following ***testing optimizations** will be added to image variants

- Localhost restriction for SOLR will be removed
- SSL requirement for the rest service will be removed

Implementation Notes

- Can we place multiple Dockerfiles into separate directories?
 - When building from the command line, it is possible to build with a Dockerfile in a subdirectory.
 - Unfortunately, the automated build process on Docker Hub changes the context directory for the build and cannot find necessary resources if the Dockerfile does not live in the root directory.
 - Solution: create multiple Dockerfiles in the root directory with meaningful names.
- We can create contextual README files for each release.
 - Unless we move the Dockerfiles to a subdirectory, we cannot change the README that displays in Docker Hub.
 - Solution: I have added a note to the main DSpace README file (see the PR above)

Image Verification Script

[Sample script output](#)

Docker Image Verification Script

```
#!/bin/sh
# CD to docker-compose dir
export DOCKER_OWNER=dspace
export DOCKER_OWNER=terrywbrady
export DPROJ=test-image

function checkImage {

    removeVols
    export DSPACE_VER=$1
    docker-compose -p $DPROJ up -d
    echo " ===== ${DSPACE_VER} ===== "
    docker exec ${DPROJ}_dspace_1 //dspace/bin/dspace version | egrep "DSpace version:|JRE:"
    checkUrl foo
    checkUrl xmlui
    checkUrl oai/request
    checkUrl solr
    checkUrl rest
    echo " ===== ===== "
    docker-compose -p $DPROJ down
}

function checkUrl {
    URL="http://localhost:8080/$1"
    echo " ---> ${URL} "
    curl -s -I -L $URL | grep "^HTTP"
}

function removeVols {
    vols=`docker volume ls --format "{{.Labels}};{{.Name}}" | grep "com.docker.compose.project=${DPROJ}" | cut -d';' -f2`
    for vol in $vols
    do
        docker volume rm $vol
    done
}

checkImage dspace-7_x-jdk8
checkImage dspace-7_x-jdk8-test
checkImage dspace-6_x-jdk8
checkImage dspace-6_x-jdk8-test
checkImage dspace-5_x-jdk8
checkImage dspace-5_x-jdk8-test
checkImage dspace-5_x-jdk7
checkImage dspace-5_x-jdk7-test
checkImage dspace-4_x-jdk7
checkImage dspace-4_x-jdk7-test
```