MedFinder   
Installation Procedure

Document Date: 6/29/2015

Product Version 1.x

C:\work\documents\Graphics\Harmonia Logo (green and purple)\HarmoniaLogo2014-logo-only-transparent-small.png

Harmonia Holdings Group LLC • 2020 Kraft Drive • Suite 2020 • Blacksburg VA 24060

Table of Contents

[1. About this Document 1](#_Toc423516918)

[2. Prerequisites 1](#_Toc423516919)

[3. Install Docker 1](#_Toc423516920)

[4. Database Container Setup 1](#_Toc423516921)

[5. Application Container Setup 1](#_Toc423516922)

[6. Viewing Logs 2](#_Toc423516923)

# About this Document

This installation document provides guidance on the installation of the MedFinder application using Docker containers to separate the database from the application.

# Prerequisites

Ensure that you have a Docker-compatible host machine on which to build and install the containers. We recommend that the host machine be given 4GB of RAM and 10 GB of hard drive space. We used Red Hat Enterprise Linux 7 (RHEL7) on Amazon’s EC2 cloud infrastructure for our host environment.

In order to build MedFinder from the GitHub repository, your build machine will need to have the following build tools with at least the given versions: Java JDK 1.7, Apache Maven 3.2.5, NodeJS 0.12.5, and Grunt 0.4.5. These can be installed from their respective websites and should be made available on the path in your environment.

# Install Docker

Follow the instructions on the Docker page to perform the Docker Engine installation. There are instructions for many different platforms. We followed the instructions for Red Hat Enterprise Linux.

Docker RHEL Installation Instructions: <https://docs.docker.com/installation/rhel/>

# Database Container Setup

To install, configure, and run the database server, use a command similar to the following. The command below creates and runs a new Docker container named *medfinder-db* based on the *mysql* container provided by the Docker container repository. Each of the *-e* options specifies an environment variable specific to this Docker container to configure the application. The container uses these environment variables to initialize the database. The *-v* option maps a location on the host file system to the file system of the container. This ensures that all data written to the database is actually stored on the host file system. The *-d* option specifies that the container should run in daemon mode (as a service).

sudo docker run \  
 --name medfinder-db \  
 -e MYSQL\_ROOT\_PASSWORD=changeme \  
 -e MYSQL\_DATABASE=ads \  
 -e MYSQL\_USER=ads \  
 -e MYSQL\_PASSWORD=changeme \  
 -v /data/mysql:/var/lib/mysql \  
 -d \  
 mysql

# Application Container Setup

The application container depends on JBoss Wildfly and requires some customization to the default container provided by JBoss before it can be started.

1. Check out the repository.

git clone https://github.com/HarmoniaHoldings/medfinder.git

1. Build the Application using Maven.

cd medfinder

mvn install

1. Copy the *ads.war* file into the *docker/medfinder-app* directory.

cp medfinder-webapp/target/ads.war docker/medfinder-app

1. Run the automated build which will build the Docker application container image.

cd docker/medfinder-app

sudo docker build --tag wildfly-medfinder .

1. Once the Docker application container is built, we can create the run Docker application container with the following command.

The command below creates and runs a new Docker container named *medfinder-app* based on the wildfly-medfinder container we just created. The *--link* option tells Docker to allow the new container being created to communicate with the container named *medfinder-db* which hosts our database server. The *mysql* portion of this command forms the prefix for environment variables exposed by the *medfinder-db* container making them identifiable within the *medfinder-app* container. The *-p* option maps port *80* on the host to port *8080* on the container, allowing traffic on the host to access the container and changing the port in the process. The *-d* option specifies that the container should run in daemon mode (as a service).

sudo docker run \  
 --name medfinder-app \  
 --link medfinder-db:mysql \  
 -p 80:8080 \  
 -d \  
 wildfly-medfinder

# Viewing Logs

Problems with startup or execution of these containers can be debugged by viewing the logs files from their startup and execution processes. The following commands make this easy to do by combining the ability to ask Docker for the ID of the container with the specified name and then asking Docker to display its logs.

sudo docker logs $(sudo docker ps -a -q --filter=name=medfinder-app)

sudo docker logs $(sudo docker ps -a -q --filter=name=medfinder-db)