

Installation of Domino on Docker using the HCL Image

Written by Roberto Boccadoro

November 2020

Introduction

To get started, here are some Docker terms that are useful to know.

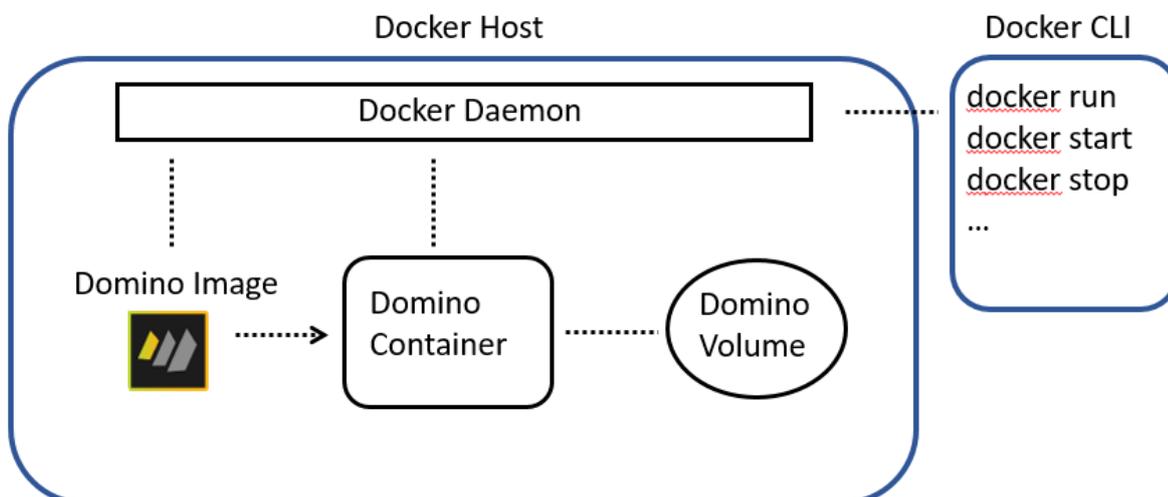
image A file that contains a built, executable version of an application. Images are uploaded to Docker and used to create and run containers.

container A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings. Usually this is a Linux runtime

volume A location on Docker that stores data to be read or written by the application. While containers are easily deleted and recreated, volumes persist. The Domino data directory is stored in a volume and you can act on the corresponding Domino container without affecting the contents of the volume.

Docker daemon The Docker service that you interact with through the docker commands to create and manage containers. Also referred to as the Docker Engine.

Docker host The physical computer that runs the Docker daemon. This can be a Linux server, physical or virtual, a Linux server running on Windows WSL or Docker desktop on a Mac



Prerequisites

The prerequisite is to have Docker or Podman installed.

If you use CentOS 8, it is preferred to use Podman rather than Docker. Make sure you are updated to the latest version; you can check, issuing this command in a terminal:

```
podman --version
```

You should be at version 1.6.4.

For reference I will use Docker in this guide. If you use podman just replace “docker” with “podman” when running the commands, e.g. if you see here a command like `docker image ls` replace it with `podman image ls`

Load the domino-docker tar archive file to Docker

Download the Domino Docker image from the Flexnet HCL site and copy it in a directory in your Linux machine. In this guide I will use the Domino V12 beta image as example.

From a terminal issue this command

```
docker load --input Domino_12.0_DockerImage_Nov2020EAP.tgz
```

To check the list of images you have, issue this command

```
docker image ls
```

You should see something like this

```
[root@centos-2gb-nbgl-1 ~]# podman image ls
REPOSITORY          TAG          IMAGE ID        CREATED         SIZE
localhost/domino-docker  V1200_11042020prod  268c36ad9df7   2 weeks ago    1.46 GB
[root@centos-2gb-nbgl-1 ~]#
```

Create the volume to store the data directory

Create a volume once for a given Domino server. This is a separate Docker container. It's easy to remove and recreate a Domino container, for example to upgrade to a new Domino version. When you do, the volume container persists, preserving the data directory.

Issue the following command

```
docker volume create notesdata
```

Create a temporary container for remote server setup

Use the docker run command with the arguments described in this procedure to create a container that runs Domino that enables you to run remote server setup. When the setup is complete we will delete this container.

Only remote server setup is supported; the Docker-compatible Red Hat Universal Base Image (UBI) doesn't contain the required X11 libraries to support Graphical User Interface (GUI) and display.

Issue the following command

```
docker run --rm --name dominosetup -v notesdata:/local/notesdata -  
-hostname v12.eldeng.it -p 8585:8585 -p 1352:1352 domino-  
docker:V1200_11042020prod --setup
```

“--rm” removes the container after the server setup is finished, since this container is not needed anymore.

“--name dominosetup” give the name dominosetup to the container

“-v notesdata:/local/notesdata” maps the Domino data directory of the server in the container, which is usually /local/notesdata to the volume we have created above. Since the volume is persistent, if you destroy the container, you can create a new one that maps the Domino data directory to the same volume and you will have all the previous data available.

“--hostname” sets the hostname of the Domino server

“-p 1352:1352” maps the internal port 1352 to the external port 1352. This means that you can change the port that the domino uses to something else very easily, just map 1352 to, say, 1353. This is useful when you want to run two domino servers on the same machine, just run two container and use different ports.

“domino-docker: V1200_11042020prod” tell to create the container based on the image whose tag is V1200_11042020prod (see above)

“--setup” runs the Domino server in setup mode

If you are setting up an additional server in the domain and the server ID is not stored in the Domino directory, have the server ID available; use the docker cp command to copy the server ID file to the Domino volume issuing the following command

```
docker cp c:\yourdir\server.id dominosetup:/local/notesdata
```

Now run the remote server setup

Create a container for Domino

Before you complete this procedure, decide what execution mode you want to use to run the container, interactive or detached. Interactive mode allows you to interact with the Domino server directly through the Docker terminal window. Detached mode runs the server without console and you interact with it using a remote console or the Domino Administrator.

To run the container in interactive mode run the following command

```
docker run -it --name domino12 -v notesdata:/local/notesdata --hostname v12.eldeng.it --cap-add=SYS_PTRACE -p 8585:8585 -p 1352:1352 -p 443:443 -p 80:80 domino-docker:V1200_11042020prod
```

The parameters are the same as above, the “--cap-add=SYS_PTRACE” enables settings to allow NSD to capture callstacks via the ptrace tool should a problem occur that requires a support ticket.

In the above command I exposed only the NRPC, HTTP and HTTPS ports, if you want to use other services (LDAP, POP2, IMAP.....) you should add the relevant port numbers in the command.

To run the container in detached mode run the following command

```
docker run -d --name domino12 -v notesdata:/local/notesdata --hostname v12.eldeng.it --cap-add=SYS_PTRACE -p 8585:8585 -p 1352:1352 -p 443:443 -p 80:80 domino-docker:V1200_11042020prod
```

To update a Domino server follow the procedure described here

https://help.hcltechsw.com/domino/11.0.1/admin/inst_dock_upgrading.html

Useful docker commands

`docker image ls` - list the docker images on the system.

```
[root@centos-2gb-nbg1-1 ~]# podman image ls
REPOSITORY          TAG          IMAGE ID        CREATED         SIZE
localhost/domino-docker  V1200_11042020prod  268c36ad9df7   2 weeks ago    1.46 GB
[root@centos-2gb-nbg1-1 ~]#
```

`docker image remove IMAGE ID` – removes the image

`docker container ls` – list the containers on the system

```
CONTAINER ID   IMAGE                                COMMAND          CREATED        STATUS        PORTS                    NAMES
afde0fa020de  localhost/domino-docker:V1200_11042020prod  2 hours ago    Up 2 hours ago  0.0.0.0:80->80/tcp      dominol2
[root@centos-2gb-nbg1-1 install]#
```

`docker container stop CONTAINER ID` - stops the container

`docker container start CONTAINER ID` – starts the container

`docker container remove CONTAINER ID` – removes the container

`docker volume create` – creates a volume to be used by a container.

`docker cp <container>:<full path to container file> <destination directory on host>` - Copies files from a container to the host system. For example: `docker cp domino:/local/notesdata/cert.id /tmp/cert.id`

`docker cp <full path to host file> <container>:<destination directory in container>` - Copies files from the host system to a container. For example: `docker cp /tmp/server.id domino:/local/notesdata`