Installing Domino on Docker
Step by Step Guide

Version 1.0.0

July 2019

This guide is offered as a free resource to the Domino community. It was authored by the following individuals:

Daniel Nashed – Nash!Com

Thomas Hampel- HCL

Compiled by Roberto Boccadoro – ELD Engineering

Introduction

Community guides are offered as free resources to the Domino community. They are authored and maintained by members of the community and may be found on the OpenNTF web site (wiki.openntf.org). OpenNTF would like to graciously acknowledge the contribution of the authors and editors of this document.

We welcome contributions to Community Guides by all members of the community. Log in to wiki.openntf.org to find out more information. If you have suggested edits to this document please post your feedback in the wiki.

Document Conventions

This guide provides a description of each step in the process.

Commands are presented in **bold red text**. These are to be entered using a console.

Operating System

In this guide we will use Centos 7.4 as base OS. We are starting with a default install with no configuration beyond a functional network. Complete all commands as ***root*** user.

Install required packages

yum install -y net-tools wget yum-utils device-mapper-persistent-data lvm2

```
[root@linux ~]# yum install -y net-tools wget yum-utils device-mapper-persistent-data lvm2
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: pkg.adfinis-sygroup.ch
 * extras: pkg.adfinis-sygroup.ch
 * updates: pkg.adfinis-sygroup.ch
Package yum-utils-1.1.31-50.el7.noarch already installed and latest version
Resolving Dependencies
--> Running transaction check
---> Package device-mapper-persistent-data.x86_64 0:0.7.0-0.1.rc6.el7 will be updated
---> Package device-mapper-persistent-data.x86_64 0:0.7.3-3.el7 will be an update
---> Package lvm2.x86_64 7:2.02.180-10.el7_6.2 will be updated
---> Package lvm2.x86_64 7:2.02.180-10.el7_6.7 will be an update
--> Processing Dependency: lvm2-libs = 7:2.02.180-10.el7_6.7 for package: 7:lvm2-2.02.180-10.el7_6.7.x86_64
---> Package net-tools.x86_64 0:2.0-0.22.20131004git.el7 will be updated
---> Package net-tools.x86_64 0:2.0-0.24.20131004git.el7 will be an update
---> Package wget.x86_64 0:1.14-15.el7 will be updated
---> Package wget.x86_64 0:1.14-18.el7_6.1 will be an update
--> Running transaction check
---> Package lvm2-libs.x86_64 7:2.02.180-10.el7_6.2 will be updated
---> Package lvm2-libs.x86_64 7:2.02.180-10.el7_6.7 will be an update
--> Processing Dependency: device-mapper-event = 7:1.02.149-10.el7_6.7 for package: 7:lvm2-libs-2.02.180-10.el7_6
.7.x86_64
--> Running transaction check
```

When finished you will see this

```
  Verifying  : wget-1.14-18.el7_6.1.x86_64                                                 8/18
  Verifying  : 7:device-mapper-1.02.149-10.el7_6.7.x86_64                                  9/18
  Verifying  : 7:device-mapper-event-1.02.149-10.el7_6.2.x86_64                           10/18
  Verifying  : wget-1.14-15.el7.x86_64                                                    11/18
  Verifying  : net-tools-2.0-0.22.20131004git.el7.x86_64                                  12/18
  Verifying  : 7:device-mapper-1.02.149-10.el7_6.2.x86_64                                 13/18
  Verifying  : 7:lvm2-libs-2.02.180-10.el7_6.2.x86_64                                     14/18
  Verifying  : 7:lvm2-2.02.180-10.el7_6.2.x86_64                                          15/18
  Verifying  : 7:device-mapper-event-libs-1.02.149-10.el7_6.2.x86_64                      16/18
  Verifying  : device-mapper-persistent-data-0.7.0-0.1.rc6.el7.x86_64                     17/18
  Verifying  : 7:device-mapper-libs-1.02.149-10.el7_6.2.x86_64                            18/18

Updated:
  device-mapper-persistent-data.x86_64 0:0.7.3-3.el7             lvm2.x86_64 7:2.02.180-10.el7_6.7
  net-tools.x86_64 0:2.0-0.24.20131004git.el7                    wget.x86_64 0:1.14-18.el7_6.1

Dependency Updated:
  device-mapper.x86_64 7:1.02.149-10.el7_6.7               device-mapper-event.x86_64 7:1.02.149-10.el7_6.7
  device-mapper-event-libs.x86_64 7:1.02.149-10.el7_6.7    device-mapper-libs.x86_64 7:1.02.149-10.el7_6.7
  lvm2-libs.x86_64 7:2.02.180-10.el7_6.7

Complete!
[root@linux ~]#
```

yum update

This will update existing packages and kernel to the current version

Install Docker CE

Installation Docker documentation for Cent OS, RHEL.

Docker CE for CentOS - https://docs.docker.com/install/linux/docker-ce/centos/

Docker EE for CentOS - https://docs.docker.com/install/linux/docker-ee/centos/

Docker EE for Red Hat Enterprise Linux - https://docs.docker.com/install/linux/docker-ee/rhel/

Add the docker repository for software downloads

yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo

Install Docker CE, the command-line and containerd

yum install -y docker-ce docker-ce-cli containerd.io

Allow this host to forward/route IP traffic and restart the network

echo net.ipv4.ip_forward=1 >> /etc/sysctl.conf

systemctl restart network

Required because Docker uses it's own network interfaces → else no outside communication

Enable (auto start) and start the Docker Service

systemctl start docker

systemctl enable docker

Test the installation

Run the Docker Hello World image to verify the Docker installations is OK

docker run hello-world

```
[root@linux ~]# docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

Work with Docker

Important commands

docker images → lists all locally available images
The ***docker ps*** command only shows running containers by default. To see all containers, use the -a (or --all) flag

docker ps → shows all running containers

docker ps -a → also shows stopped containers

docker ps -a -q to list down all containers with only numeric IDs

To stop all container one liner command comes in handy

docker stop $(docker ps -a -q)

To Remove all containers one liner command

docker rm $(docker ps -a -q)

use -f to remove/stop container forcefully

docker run → creates container from image and starts it

docker start/stop → starts/stops existing containers

docker inspect → shows detailed information for a container

docker volume ls → lists existing volumes

docker volume rm → removes a volume

docker exec → executes a command inside a container

Docker Volumes

By default all data is stored in the container

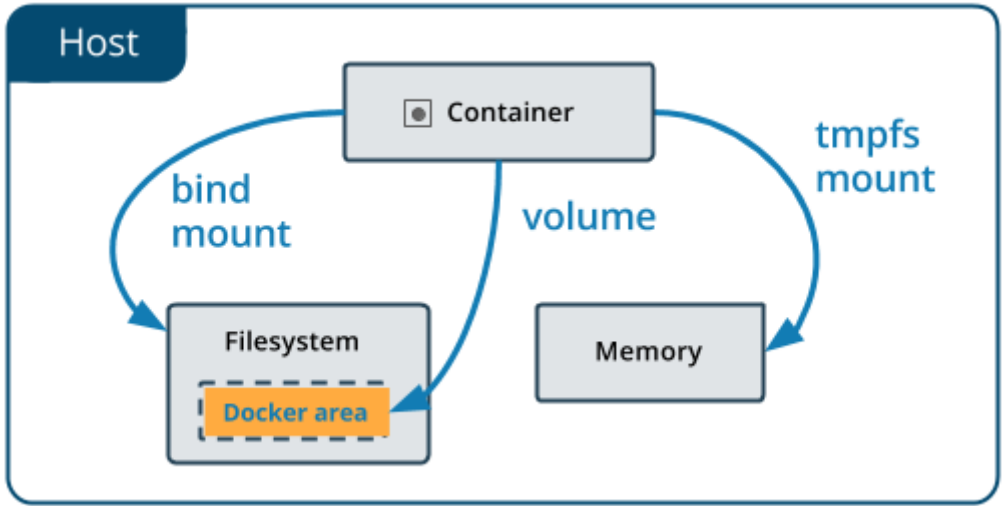
For applications with local storage requirements, this does not work well

Therefore Docker supports “**volumes**” which are mapped in to the container

The data from the local directory will be copied to the volume at first run when the volume is empty

The default implementation is a local disk

You can either create a volume manually, mount existing directories or let Docker create it



Docker Volume Commands

docker volume ls

Lists all volumes

docker volume inspect my-vol

Shows details about one volume

docker volume create my-vol

Creates local volume

docker volume rm my-vol

Removes volume!

docker system df

Checks for used/free space

docker system prune

WARNING! This will remove:

- all stopped containers

- all networks not used by at least one container

- all dangling images

- all dangling build cache

Domino V10 on Docker

First of all we need to install the git software

yum install -y git

Create a new directory for your git projects and switch to it

mkdir -p /local/github

cd /local/github

Clone ("download") the official repository locally

git clone https://github.com/IBM/domino-docker

The IBM Docker Script requires software either locally or on a download location

Domino CE 10.0.1 → Domino **C**ommunity **E**dition

Domino CE 10.0.1 FP1 → the Community Edition has it's own Fixpack!

There are two different modes

a.) Store the files locally and have a local NGINX Server provide the software locally

b.) Specify central / remote download location

In both cases the install process downloads files into the running install image, after installation files are removed to keep the image "smaller"

In this guide we will use the first mode so you need to download the packages and put them in /local/github/domino-docker/software

Important: do not change the names of the packages.

Build the Domino image

cd /local/github/domino-docker

./build.sh domino

```
 ---> be64627dfc1f
Step 20/21 : LABEL DominoDocker.description=IBM Domino Enterprise Server
 ---> Running in c8123f89d7f1
Removing intermediate container c8123f89d7f1
 ---> df3f7481c281
Step 21/21 : LABEL DominoDocker.version=10.0.1FP1
 ---> Running in 30d3e85d1c89
Removing intermediate container 30d3e85d1c89
 ---> 3bbaefb7756e
Successfully built 3bbaefb7756e
Successfully tagged ibmcom/domino:10.0.1FP1
Successfully tagged ibmcom/domino:latest
/local/github/domino-docker



Completed in 4 minutes and 48 seconds

ibmsoftware
ibmsoftware
Stopped & Removed Software Repository Container

Completed in 4 minutes and 59 seconds
[root@linux domino-docker]#
```

Create the Volume

First create a new/empty persistent volume that will be used as the Domino Data directory later on. In this example we are calling it "notesdata".

docker volume create notesdata

Create a container from the image and rint the Domino server

Go in the directory /local/github/domino-docker/management and change the file env_domino according to your needs

The default provided file is this:

Domino Server Configuration

AdminFirstName=John

AdminLastName=Doe

AdminPassword=domino4ever

DominoDomainName=Acme

OrganizationName=Acme

ServerName=domino-acme-01

cd /local/github/domino-docker

docker run -it -p 80:80 -p 1352:1352 --hostname=myhost --name mycontainer --cap-add=SYS_PTRACE --env-file management/env_domino -v notesdata:/local/notesdata ibmcom/domino:latest

open another terminal

docker exec -it mycontainer /bin/bash

domino console

```
[root@linux ~]# docker exec -it domino1001fp1 /bin/bash
[root@linux /]# domino console


--- Live Console for notes ---

To close console, always type 'close' or 'stop'.


[000708:000002-00007F149B86E740] 05/20/2019 17:10:48   LDAP Server: Started
[001170:000002-00007F2178DF0740] 05/20/2019 17:10:48   Started verifying directo
ry tree on 'names.nsf'...
[001170:000002-00007F2178DF0740] 05/20/2019 17:10:48   Finished verifying direct
ory tree on 'names.nsf'
[000703:000005-00007F40B0A1F740] 05/20/2019 17:10:48   Informational, rebuilding
 view - no container or index (reading /local/notesdata/names.nsf view note Titl
e:'($Clusters)')
[000703:000002-00007F40B0A1F740] 05/20/2019 17:10:48   Router: Mail Router start
ed for domain ACME
[000703:000002-00007F40B0A1F740] 05/20/2019 17:10:48   Router: Internet SMTP hos
t linux in domain eld.it; SMTP display host name linux.eld.it
[000421:000002-00007FA6384B0740] 05/20/2019 17:10:50   Event: Setting up default
 monitors in Monitoring Configuration database.
[000421:000002-00007FA6384B0740] 05/20/2019 17:10:51   Event: Upgrading the desi
```

Stop the Domino server

Docker by default sends a **SIGTERM** signal to the main process of the container

If the main process does not stop within **10 seconds**, a **SIGKILL** signal is send to the container

This would not allow Domino to shutdown cleanly in most of the cases

Solution

Specify a longer shutdown grace period and ensure that the main process will catch the shutdown

request to start the Domino shutdown

Open a terminal

docker stop --time=120 mycontainer

Start again the Domino server

docker container start mycontainer