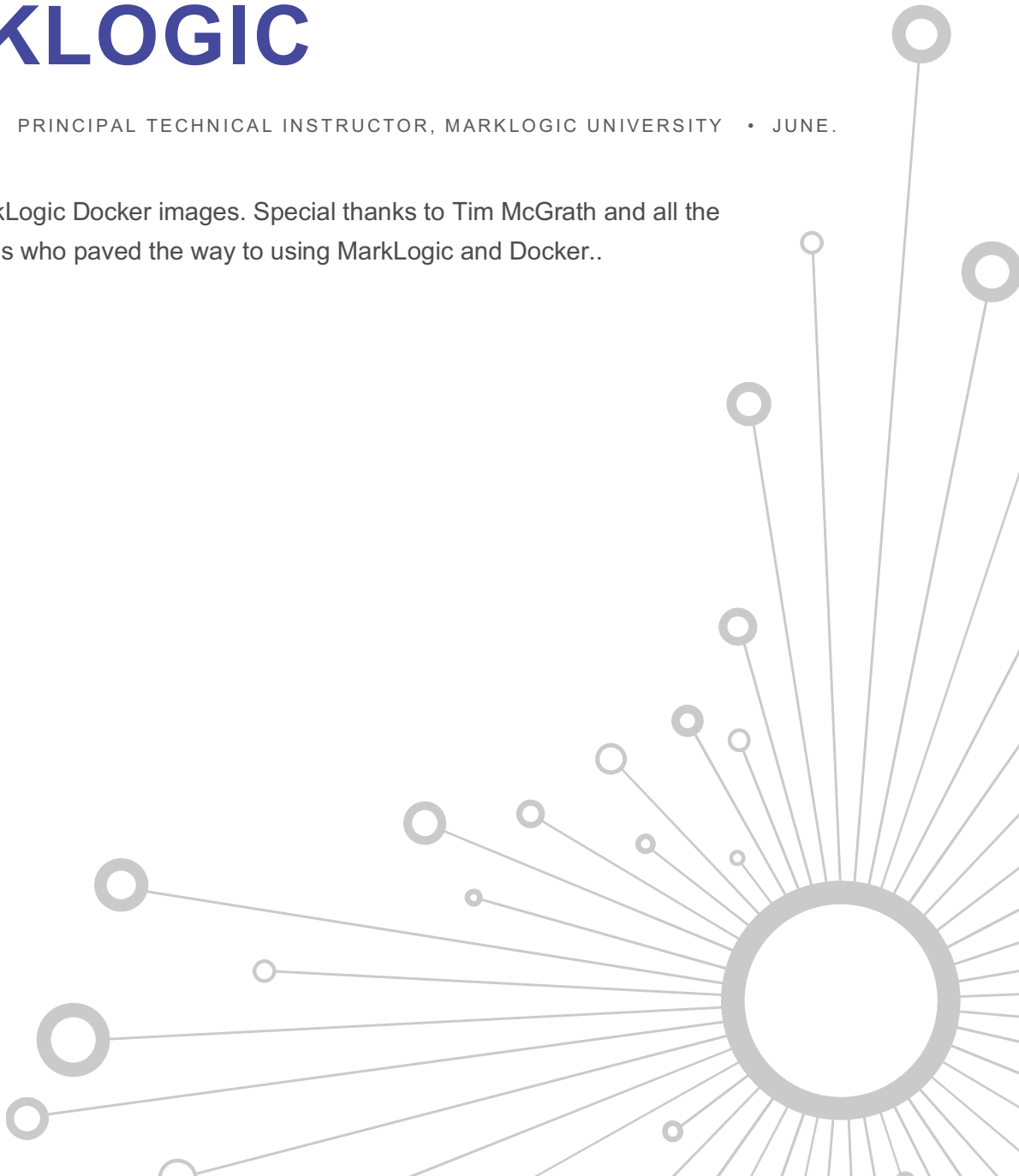
MarkLogic®

CREATING DOCKER IMAGES WITH MARKLOGIC

ALAN JOHNSON • PRINCIPAL TECHNICAL INSTRUCTOR, MARKLOGIC UNIVERSITY • JUNE. 2018

How to create MarkLogic Docker images. Special thanks to Tim McGrath and all the other MarkLogicians who paved the way to using MarkLogic and Docker..



Contents

WHAT IS DOCKER

From the Docker website (https://www.docker.com/what-docker)

> "Docker containers wrap a piece of software in a complete filesystem that contains everything needed to run: code, runtime, system tools, system libraries – anything that can be installed on a server. This guarantees that the software will always run the same, regardless of its environment."

CAN I RUN DOCKER ON WINDOWS?

Yes! The latest version of Docker no longer requires Virtual Box installed on Windows in order to use Docker. Windows 8 and Windows 10 have a hypervisor included as an optional service feature. A hypervisor is computer software, firmware or hardware that creates and runs virtual machines. Examples include VMWare, Parallels and Virtual Box.

CAN I RUN DOCKER AND A VM AT THE SAME TIME ON WINDOWS?

No. The hypervisors would conflict with each other. To run Docker on Windows, the Hyper-V feature must be enabled. The Docker installation will do so for you. If you then want to run VMWare or Virtual Box, you would have to disable the Hyper-V feature in Windows and reboot. You could then start your VM normally. Docker would be disabled until Hyper-V was enabled again. Instructions on how to do this in Windows 10 are at the end of this document.

HOW DO I CREATE DOCKER IMAGES AND USE THEM?

Below are instructions for getting a Docker image created with MarkLogic installed. Then anytime you need to run MarkLogic, just create a new container or start an existing one. Since MarkLogic is installed within containers, start the desired version easily and quickly. Containers start faster than VMs and take up less space than VMs.

I need to thank Tom McGrath, the original author of this document. I've changed some steps to be consistent with Docker best practices. I also need to acknowledge the dedication and work by Patrick McElwee and Ricard Louapre on the Dockerfile build scripts. Thanks for sharing your efforts, gentlemen!

DOCKER IMAGE BUILDING STEPS

1. Install Docker for your platform. Verify install using any Docker commands you wish from a command prompt. The following gives the current version of Docker.

```
$ docker version
```

2. Extract the respective MarkLogic Dockerfile into a directory path that is easy to get to. A sub-directory will be created; "**Docker-ML**".

Docker-ML
Contains a Dockerfile specifically for versions of MarkLogic less than MarkLogic 9 Early Access.

3. Download the MarkLogic RPM for CentOS 7 from developer.marklogic.com and move the .RMP file into the Docker-ML8 or Docker-ML9EA directory, depending upon the version of MarkLogic you wish to install.

You should now have a Dockerfile and the MarkLogic install .rpm file in the same **Docker-ML** directory.

4. Open the Dockerfile in a decent text editor. Search for the following comment:

```
# Copy the MarkLogic installer to a temp directory in the Docker image being built
```

Ensure the part after COPY points to the MarkLogic RPM you downloaded and copied to the <Current Working Directory>.

```
COPY MarkLogic-9.0-5.1.x86_64.rpm /tmp/MarkLogic.rpm
```

5. Build your machine. Choose one of the following commands depending on MarkLogic version.

In the **Docker-ML** directory (the trailing dot is required) -

```
$ docker build -t marklogic-initial-install:9.0-5.1 .
```

Tags (TL/DR)

Docker image names have 2 parts separated by a colon (":"). The first part is the image name. The part after the colon is called the "tag". Often, the tag is used to differentiate versions or specific variations among similar images.

The previous build commands created an image called "marklogic-initial-install". The image name was tagged with the MarkLogic version. This is optional but might be a good practice if you switch between multiple containers with differing MarkLogic versions on your desktop.

6. Run the machine. Choose one of the following depending on MarkLogic version. This will create a Docker container with MarkLogic after it's been installed and ready for the Initializing the Instance phase.

```
$ docker run -d --name=initial_install -p 8001:8001 marklogic-initial-install:9.0-5.1
```

The Docker run command starts a new container named "**initial_install**" with the CMD line from the Dockerfile. The following options are passed to the Docker run command: You can have as many Docker containers started from the same image as you desire. It's a best practice to give each container a unique name.

Parameter	Does what
`-d`	Run in detached mode. Run in the background rather than the default foreground mode.
`--name`	The name of the container. A Docker container can be stopped, started and deleted by its name or its container ID number.
`-p`	The port to contact the container on followed by the port exposed in the Dockerfile. Example – 18001:8001 To contact the container listening internally on port 8001, use the local computer port of 18001.

To get to the MarkLogic Admin Interface on port 8001 in the contain, simply point your browser to http://localhost:8001, exactly as if MarkLogic was installed on your local computer.

Important, if you have MarkLogic installed locally, either stop MarkLogic so the Docker container can communicate on port 8001 or assign a different port number in the Docker run command.

Example: `$ docker run -d -name=initial_install -p 18001:8001 marklogic-initial-install:9.0-5.1` which creates a Docker container named "initial_install". Use port 18001 locally to contact MarkLogic in the container listening in the container internal port of 8001. The internal port was referred to in the EXPOSE command in the Dockerfile.

The CMD command in the Dockerfile (TL/DR)

In the Dockerfile, we are running MarkLogic. Normally, this would start MarkLogic then the Docker container would then quit after successfully running the MarkLogic server. We want the Docker container to not quit after starting MarkLogic so we have access to it. Therefore, the CMD in the Dockerfile also starts the Linux "tail" command with the "-f" option which tells the "tail" command that if the end of the file is reached, wait for new data. As we are reading something that never has data (dev/null), the command will always just wait thus our Docker container doesn't immediately exit.

7. Open the MarkLogic Admin interface (http://localhost:8001/) and do an initial quick setup of MarkLogic. List all created Docker containers, running or stopped, with the following command.

```
$ docker ps -a
```

8. Commit your Docker container into an image to save its state (you can then spawn up multiple containers of this image at any time!). Choose one of the following depending on MarkLogic version.

```
$ docker commit initial_install ml-installed:9.0-5.1
```

9. Stop the Docker initial_install container and permanently remove the container. It is no longer necessary.

```
$ docker stop initial_install
```

```
$ docker rm initial_install
```

You must stop a container before permanently removing it.

10. Remove the initial_install image to save disk space. Choose one of the following depending on MarkLogic version.

```
$ docker rmi marklogic-initial-install:9.0-5.1
```

11. Create a new Docker container of MarkLogic as desired. This will act as your installed, clean MarkLogic instance. If you make changes to the container, you can stop the container but don't remove the container. This will stop the container from listening on your local ports. When you need the container again, simply start the container.

```
$ docker run -d --name=<desired container name> -p 8000-8002:8000-8002 <image name:tag>
```

Where...

Parameter	Does what
`--name`	Name of the new Docker container you would like to spin up.
`-p`	The ports you would like open for connecting to it. The port listed in the command above exposes all. Ports are listed as <host port>:<container port>. For example: -p 8000:8000 means port 8000 on the host computer maps to port 8000 in the Docker container. Ports can be listed as a range. See example in step 11 above.
`<image name:tag>`	Specifies from which image to create this Docker container.

12. Enjoy using MarkLogic in your Docker container.

HELPFUL COMMANDS TO KNOW ALONG THE WAY

LIST DOCKER CONTAINERS

Containers are the runtime objects of Docker.

To see those running:

```
$ docker ps
```

To see those that exist but are not running:

```
$ docker ps -a
```

LIST DOCKER IMAGES

Docker images save the states of the Docker containers.

```
$ docker images
```

MAPPING A HOST DIRECTORY TO A DOCKER CONTAINER'S DIRECTORY

When creating a container with the docker run command, you can specify a directory on the host computer to map to a directory in the container. If the directory doesn't exist in the container, it will be created. If there are files and sub-directories in the host directory, they will be copied into the container directory. Any change in either the host or container directory will be reflected in the other. To delete the container, you must specify the –v parameter. The host directory is not deleted. This is beneficial for persistent storage for the container such as MarkLogic's data directory where you want to keep your content and configurations but may delete the container when upgrading MarkLogic.

```
$ docker run –d –name=<container-name> –p 8000-8002:8000-8002 –v
/my/host/directory:/var/opt/MarkLogic marklogic-install:9.0-5.1
```

DELETING DOCKER CONTAINERS

To delete a Docker container (use –v if the container was created with a volume):

```
$ docker rm [-v] <container name of id>
```

To delete all Docker containers no matter if they are running or not (careful, this works very well):

```
$ docker rm [-v] $(docker ps -a -q)
```

DELETING DOCKER IMAGES

To delete a Docker image:

```
$ docker rmi <image name:imagetag>
```

Example:

```
$ docker rmi marklogic-initial-install:9.0-5.1
```

To delete all Docker images (careful, this works very well):

```
$ docker rmi $(docker images -q)
```

OPEN AN SSH TERMINAL INTO A RUNNING DOCKER CONTAINER

```
docker exec -it <contain name of id> bash
```

For example...

```
docker exec -it initial_install bash
```

COPY A FILE FROM DOCKER CONTAINER TO THE HOST MACHINE

When you want to re-use them in the future for a new Docker (for example).

```
docker cp <container name of id>:/file/path/within/container /host/path/target
```

COMMIT CHANGES TO A DOCKER CONTAINER TO A NEW IMAGE

```
$ docker commit <container name of id> <image name:image tag>
```

Committing a Docker container to a new image saves its state. You can then create a new container based upon that saved state.

Common reasons why you'd commit a container to a new image include:

1. Adding content to a MarkLogic database that you'd want to keep even if you deleted the container

2. Creating new MarkLogic App Servers so you have additional ports to EXPOSE in your Docker file and additional ports you'd want to contact in the Docker run command.

3. Adding other software to the container and you'd like to persist those changes so new containers also have that software. However, application such as Node.js or Java, you can call from your local computer and contact MarkLogic in your container as if the container was another computer.

ENABLING AND DISABLING HYPER-V ON WINDOWS 10

Below is an excellent step by step for enabling Hyper-V on Windows including requirements.

https://blogs.technet.microsoft.com/canitpro/2015/09/08/step-by-step-enabling-hyper-v-for-use-on-windows-10/

To disable, uncheck the feature and restart the computer.

I have read anecdotes regarding people who have run into difficulties after repeatedly enabling and disabling the Hyper-V feature in Windows. The consensus is to either enable and leave on or don't enable the feature.

Below are suggestions from others on enabling and disabling the Hyper-V features without registry changes. This hopes to avoid any issue of registry corruption.

FROM A WINDOWS COMMAND PROMPT, TYPE THE FOLLOWING TO DISABLE HYPER-V.

```
bcdedit /set hypervisorlaunchtype off
```

Reboot the computer

FROM A WINDOWS COMMAND PROMPT, TYPE THE FOLLOWING TO RE-ENABLE HYPER-V.

```
bcdedit /set hypervisorlaunchtype auto
```

Reboot the computer

About MarkLogic

For over a decade, organizations around the world have come to rely on MarkLogic to power their innovative information applications. As the world's experts at integrating data from silos, MarkLogic's operational and transactional Enterprise NoSQL database platform empowers our customers to build next generation applications on a unified, 360-degree view of their data. Headquartered in Silicon Valley, MarkLogic has offices throughout the U.S., Europe, Asia, and Australia.

MarkLogic®

999 SKYWAY ROAD, SUITE 200, SAN CARLOS, CA 94070

+1 650 655 2300 · +1 877 992 8885 · www.marklogic.com · sales@marklogic.com