Docker Deel 2

Les 9 20181129

M. DIMA

Docker

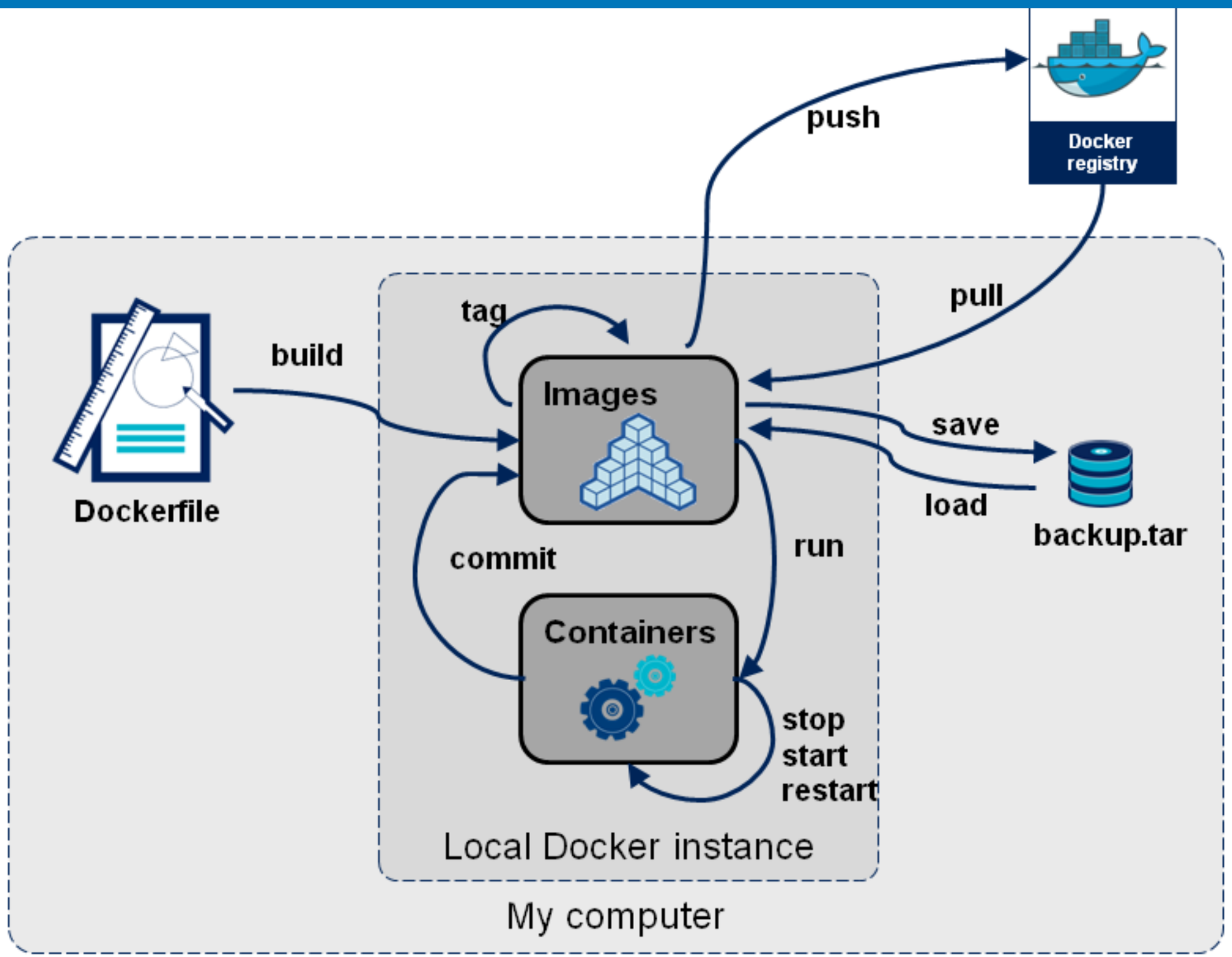
What is Docker?

Linux OS isolation tools made easy

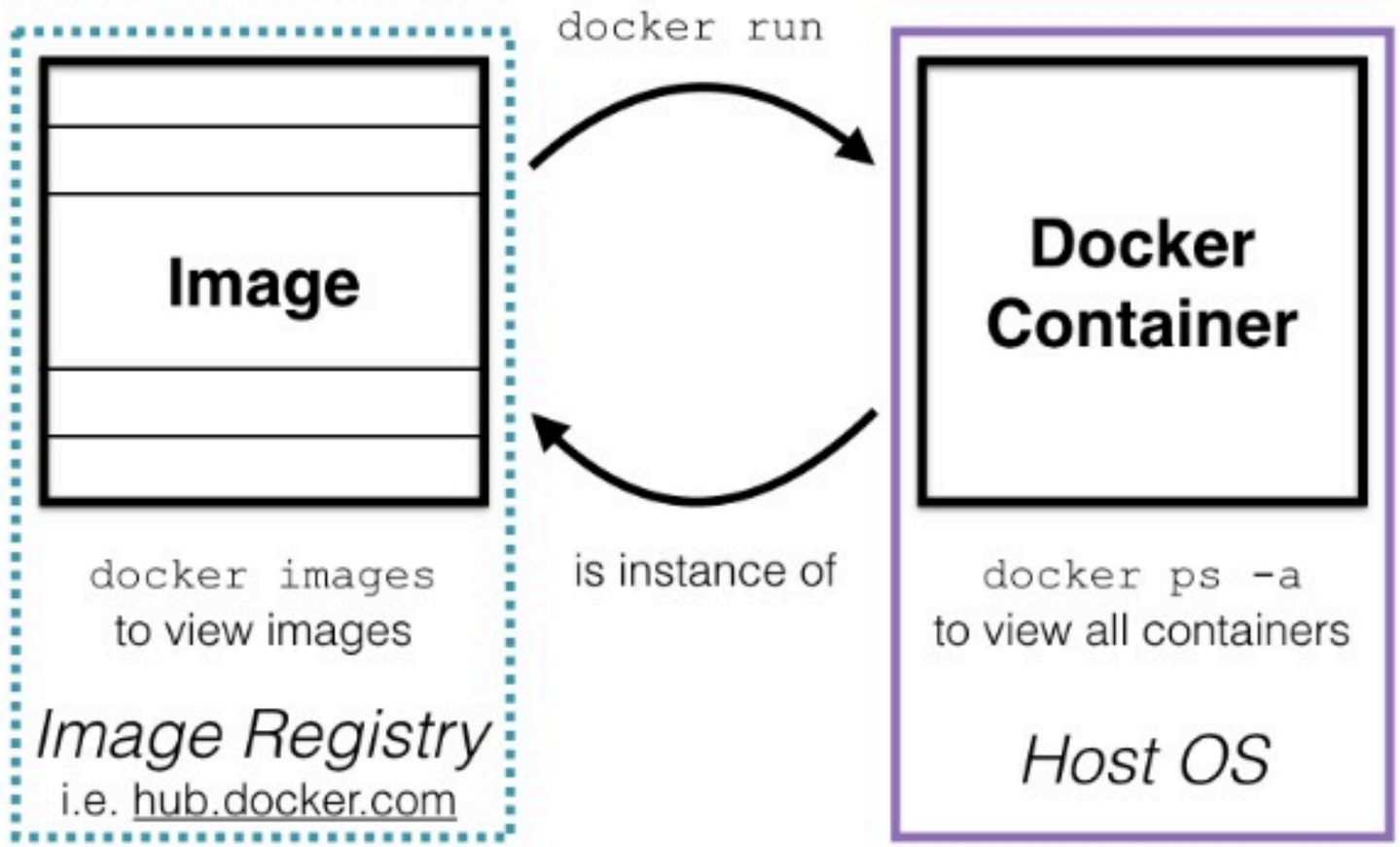
A Docker container looks like a virtual machine

Provide additional 'goodies' for app development

Images vs Containers - Herhaling



Images vs Containers - Herhaling



Listing Images

- `docker images`
- Lists downloaded images

```
$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
my-image-2          latest              2df1353ae753        5 hours ago         122 MB
my-image-3          latest              18f8666d2326        5 hours ago         122 MB
my-image-4          latest              18f8666d2326        5 hours ago         122 MB
my-image-5          latest              18f8666d2326        5 hours ago         122 MB
my-image            latest              18f8666d2326        5 hours ago         122 MB
<none>              <none>              bd717eee7137        22 hours ago        122 MB
ubuntu              latest              2fa927b5cdd3        12 days ago         122 MB
ubuntu              14.04               8f1bd21bd25c        12 days ago         188 MB
$
```

Tagging Images

- Tagging gives images names
- `docker commit` tags images for you

```
$ docker ps -l --format=$FORMAT

ID      bc8b849ef747
IMAGE   ubuntu:14.04
COMMAND "bash"
CREATED 38 minutes
STATUS  Up 37 minutes
PORTS
NAMES   server

$ docker commit bc8b849ef747 my-image-14
sha256:e40551abfd3f9a706b61bc7ca7f04c3c6b4d0184712da781fba3bf45d2de6baf
$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
my-image-14         latest              e40551abfd3f        3 seconds ago       188 MB
my-image-2          latest              2df1353ae753        5 hours ago         122 MB
my-image-3          latest              18f8666d2326        5 hours ago         122 MB
my-image-4          latest              18f8666d2326        5 hours ago         122 MB
my-image-5          latest              18f8666d2326        5 hours ago         122 MB
my-image            latest              18f8666d2326        5 hours ago         122 MB
<none>              <none>              bd717eee7137        22 hours ago        122 MB
ubuntu              latest              2fa927b5cdd3        12 days ago         122 MB
ubuntu              14.04               8f1bd21bd25c        12 days ago         188 MB
$
```

```
sha256:e40551abfd3f9a706b61bc7ca7f04c3c6b4d0184712da781fba3bf45d2de6baf
$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
my-image-14         latest              e40551abfd3f        3 seconds ago       188 MB
my-image-2          latest              2df1353ae753        5 hours ago         122 MB
my-image-3          latest              18f8666d2326        5 hours ago         122 MB
my-image-4          latest              18f8666d2326        5 hours ago         122 MB
my-image-5          latest              18f8666d2326        5 hours ago         122 MB
my-image            latest              18f8666d2326        5 hours ago         122 MB
<none>              <none>              bd717eee7137        22 hours ago        122 MB
ubuntu              latest              2fa927b5cdd3        12 days ago         122 MB
ubuntu              14.04               8f1bd21bd25c        12 days ago         188 MB
$ docker commit bc8b849ef747 my-image-14:v2.1
sha256:19afef609cc9e33853193bac35d751cc6cec694988d34a43297bdca1c5e22447
$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
my-image-14         v2.1                19afef609cc9        3 seconds ago       188 MB
my-image-14         latest              e40551abfd3f        27 seconds ago      188 MB
my-image-2          latest              2df1353ae753        5 hours ago         122 MB
my-image-3          latest              18f8666d2326        5 hours ago         122 MB
my-image-4          latest              18f8666d2326        5 hours ago         122 MB
my-image-5          latest              18f8666d2326        5 hours ago         122 MB
my-image            latest              18f8666d2326        5 hours ago         122 MB
<none>              <none>              bd717eee7137        22 hours ago        122 MB
ubuntu              latest              2fa927b5cdd3        12 days ago         122 MB
ubuntu              14.04               8f1bd21bd25c        12 days ago         188 MB
```

Tagging Images

- Tagging gives images names
- `docker commit` tags images for you
- This is an example of the name structure:

```
registry.example.com:port/organization/image-name:version-tag
```

Tagging Images

- You can leave out the parts you don't need
- `Organization/image-name` is often enough

Getting Images

- `docker pull`
- Run automatically by `docker run`
- Useful for offline work
- Opposite: `docker push`

Cleaning Up

```
# Delete all containers
docker rm $(docker ps -a -q)
# Delete all images
docker rmi $(docker images -q)
```

- Images can accumulate quickly

  `docker rmi image-name:tag`

  `docker rmi image-id`

```
$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED                  SIZE
my-image-14         v2.1                19afef609cc9        About a minute ago       188 MB
my-image-14         latest              e40551abfd3f        2 minutes ago            188 MB
my-image-2          latest              2df1353ae753        5 hours ago              122 MB
my-image-3          latest              18f8666d2326        5 hours ago              122 MB
my-image-4          latest              18f8666d2326        5 hours ago              122 MB
my-image-5          latest              18f8666d2326        5 hours ago              122 MB
my-image            latest              18f8666d2326        5 hours ago              122 MB
<none>              <none>              bd717eee7137        22 hours ago             122 MB
ubuntu              latest              2fa927b5cdd3        12 days ago              122 MB
ubuntu              14.04               8f1bd21bd25c        12 days ago              188 MB
$ docker rmi bd717eee7137
Deleted: sha256:bd717eee7137bb98a3ed138d50a917fc3a510631536868e1e745fea46835098e
$ docker rmi my-image
Untagged: my-image:latest
$
```

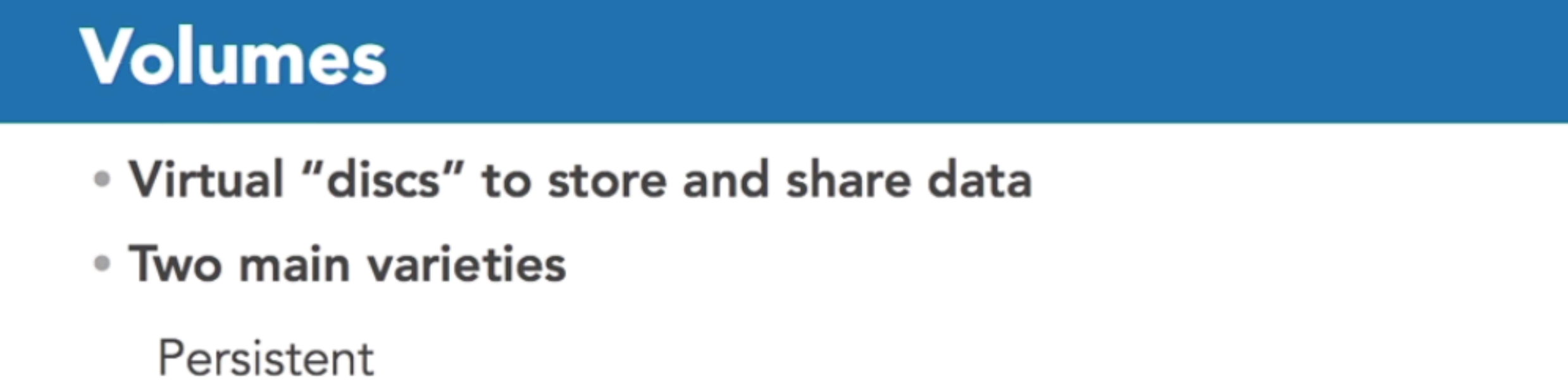
Volumes

- **Virtual "discs" to store and share data**
- **Two main varieties**

  Persistent

  Ephemeral

- **Not part of images**



Sharing Data with the Host

- "Shared folders" with the host
- Sharing a "single file" into a container

example — root@e9cba4b63d7a: / — -bash — 85×24

```
[milans-MBP:~ doc$ mkdir example
[milans-MBP:~ doc$ pwd
 /Users/doc
[milans-MBP:~ doc$ docker run -ti -v /Users/doc/example:/shared-folder ubuntu bash
[root@e9cba4b63d7a:/# ls
 bin   dev  home  lib64  mnt  proc  run   shared-folder  sys  usr
 boot  etc  lib   media  opt  root  sbin  srv            tmp  var
[root@e9cba4b63d7a:/# ls shared-folder
[root@e9cba4b63d7a:/# touch /shared-folder/my-data
[root@e9cba4b63d7a:/# ls shared-folder
 my-data
[root@e9cba4b63d7a:/# exit
 exit
[milans-MBP:~ doc$ cd example
[milans-MBP:example doc$ ls
 my-data
 milans-MBP:example doc$
```

Sharing Data with the Host

- "Shared folders" with the host
- Sharing a "single file" into a container
- Note that a file must exist *before* you start the container, or it will be assumed to be a directory

Sharing Data between Containers

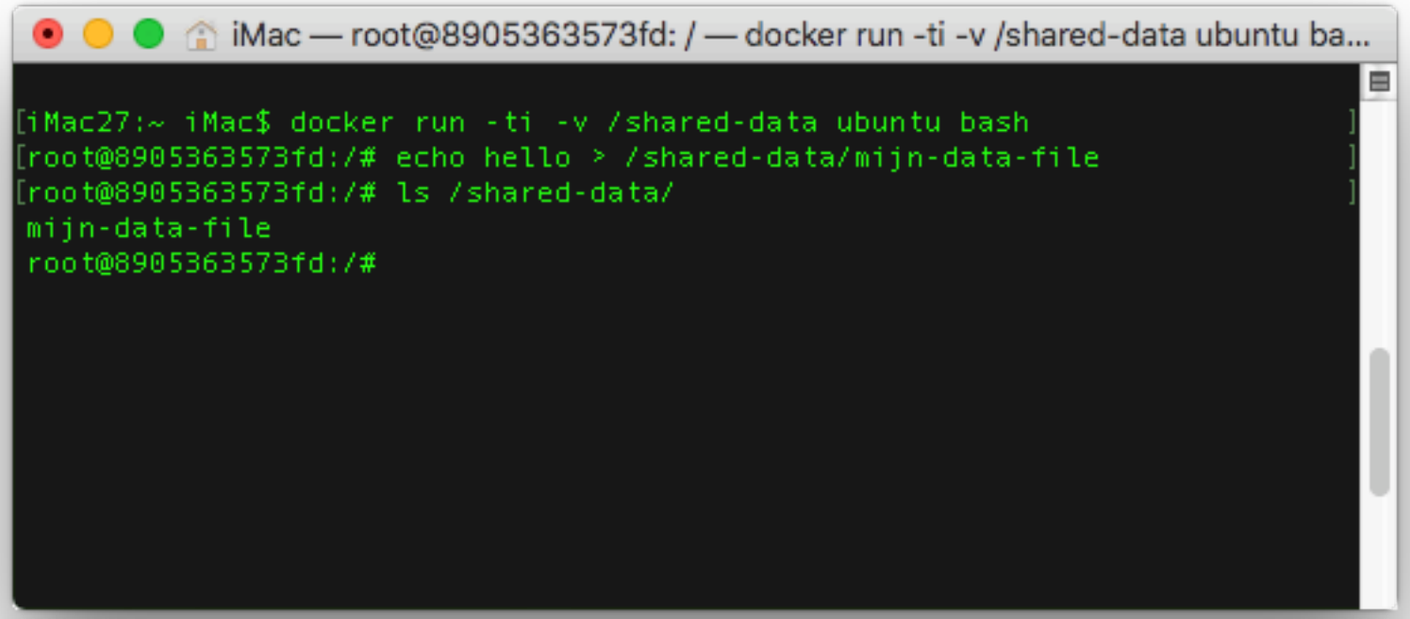
- `volumes-from`
- Shared “discs” that exist only as long as they are being used
- Can be shared between containers

```
iMac27:~ iMac$ docker run -ti -v /shared-data ubuntu bash
```

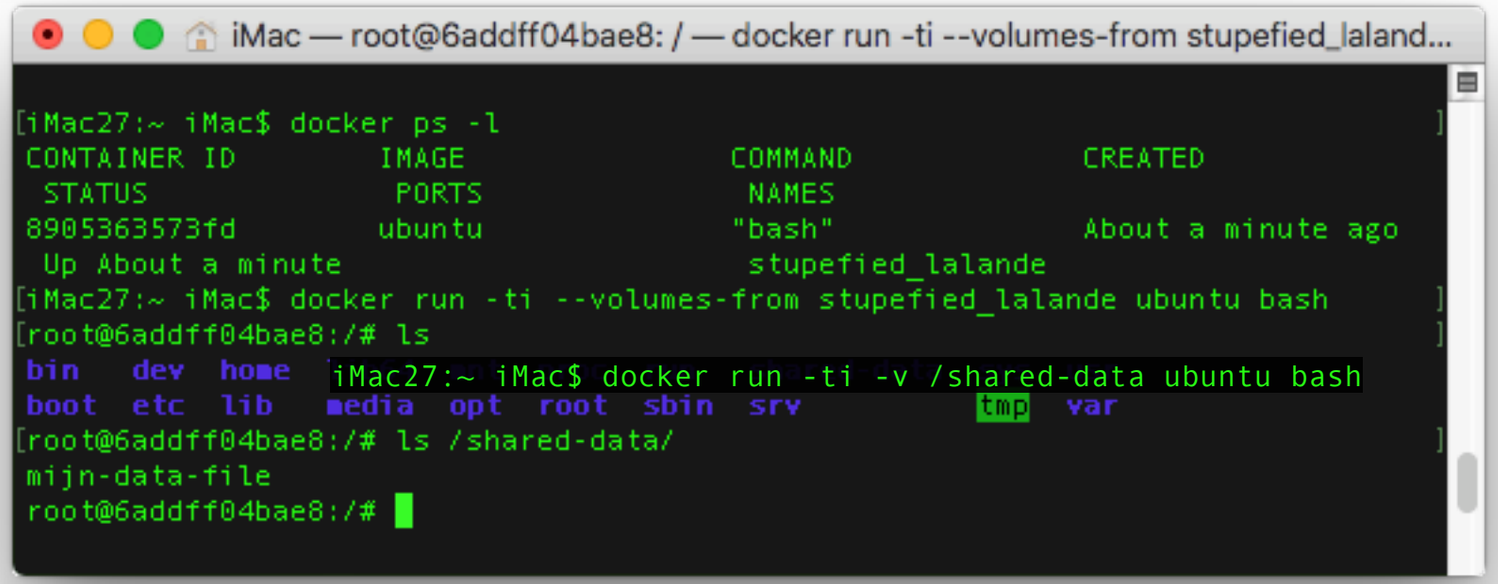
create volume

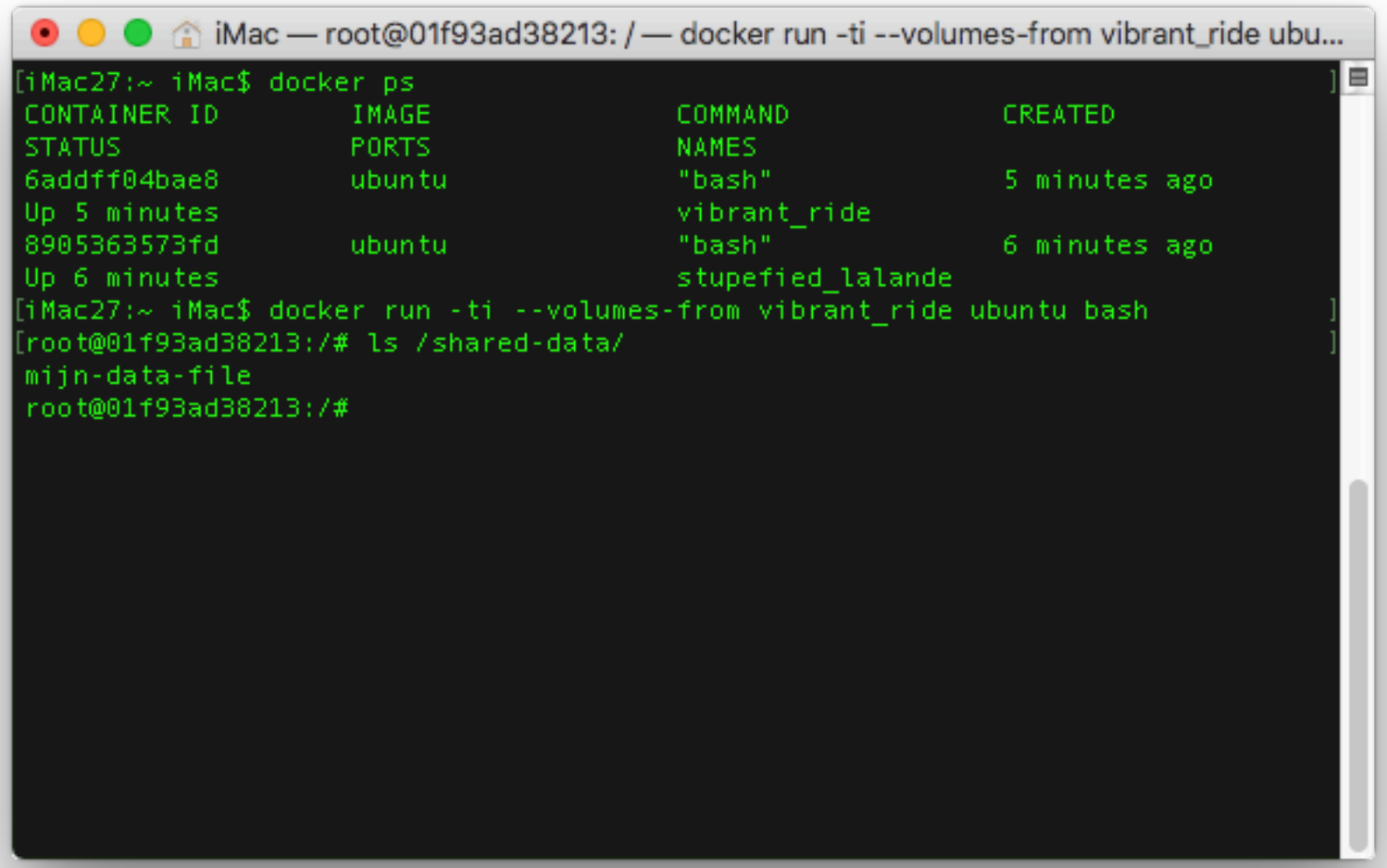
en map die hier

```
iMac27:~ iMac$ docker run -ti -v /shared-data ubuntu bash
```




```
iMac27:~ iMac$ docker run -ti --volumes-from stupefied_lalande ubuntu bash
```





T3

volume blijft bestaan zolang minsten één van de containers running is

Docker Registries

- Registries manage and distribute images
- Docker (the company) offers these for free
- You can run your own, as well

https://docs.docker.com/registry/deploying/

Finding Images

- Docker search *command*

```
$ docker search ubuntu
NAME                                DESCRIPTION                                     STARS     OFFICIAL   AUTOMATE
D
ubuntu                              Ubuntu is a Debian-based Linux operating s...   4066      [OK]
ubuntu-upstart                      Upstart is an event-based replacement for ...   63        [OK]
rastasheep/ubuntu-sshd              Dockerized SSH service, built on top of of...   28                   [OK]
torusware/speedus-ubuntu            Always updated official Ubuntu docker imag...   26                   [OK]
ubuntu-debootstrap                  debootstrap --variant=minbase --components...   25        [OK]
ioft/armhf-ubuntu                   [ABR] Ubuntu Docker images for the ARMv7(a...   12                   [OK]
nickistre/ubuntu-lamp               LAMP server on Ubuntu                           7                    [OK]
nickistre/ubuntu-lamp-wordpress     LAMP on Ubuntu with wp-cli installed            5                    [OK]
nuagebec/ubuntu                     Simple always updated Ubuntu docker images...   5                    [OK]
nimmis/ubuntu                       This is a docker images different LTS vers...   4                    [OK]
maxexcloo/ubuntu                    Docker base image built on Ubuntu with Sup...   2                    [OK]
darksheer/ubuntu                    Base Ubuntu Image -- Updated hourly             1                    [OK]
admiringworm/ubuntu                 Base ubuntu images based on the official u...   1                    [OK]
jordi/ubuntu                        Ubuntu Base Image                               1                    [OK]
esycat/ubuntu                       Ubuntu LTS                                      0                    [OK]
konstruktoid/ubuntu                 Ubuntu base image                               0                    [OK]
webhippie/ubuntu                    Docker images for ubuntu                        0                    [OK]
lynxtp/ubuntu                       https://github.com/lynxtp/docker-ubuntu         0                    [OK]
life360/ubuntu                      Ubuntu is a Debian-based Linux operating s...   0                    [OK]
widerplan/ubuntu                    Our basic Ubuntu images.                        0                    [OK]
teamrock/ubuntu                     TeamRock's Ubuntu image configured with AW...   0                    [OK]
datenbetrieb/ubuntu                 custom flavor of the official ubuntu base ...   0                    [OK]
ustclug/ubuntu                      ubuntu image for docker with USTC mirror        0                    [OK]
uvatbc/ubuntu                       Ubuntu images with unprivileged user            0                    [OK]
dorapro/ubuntu                      ubuntu image                                    0                    [OK]
$
```

Chrome File Edit View History Bookmarks People Window Help

https://hub.docker.com/search/?isAutomated=0&isOfficial=0&page=1&pullCount=0&q=node&starCount=0

Explore Help

node

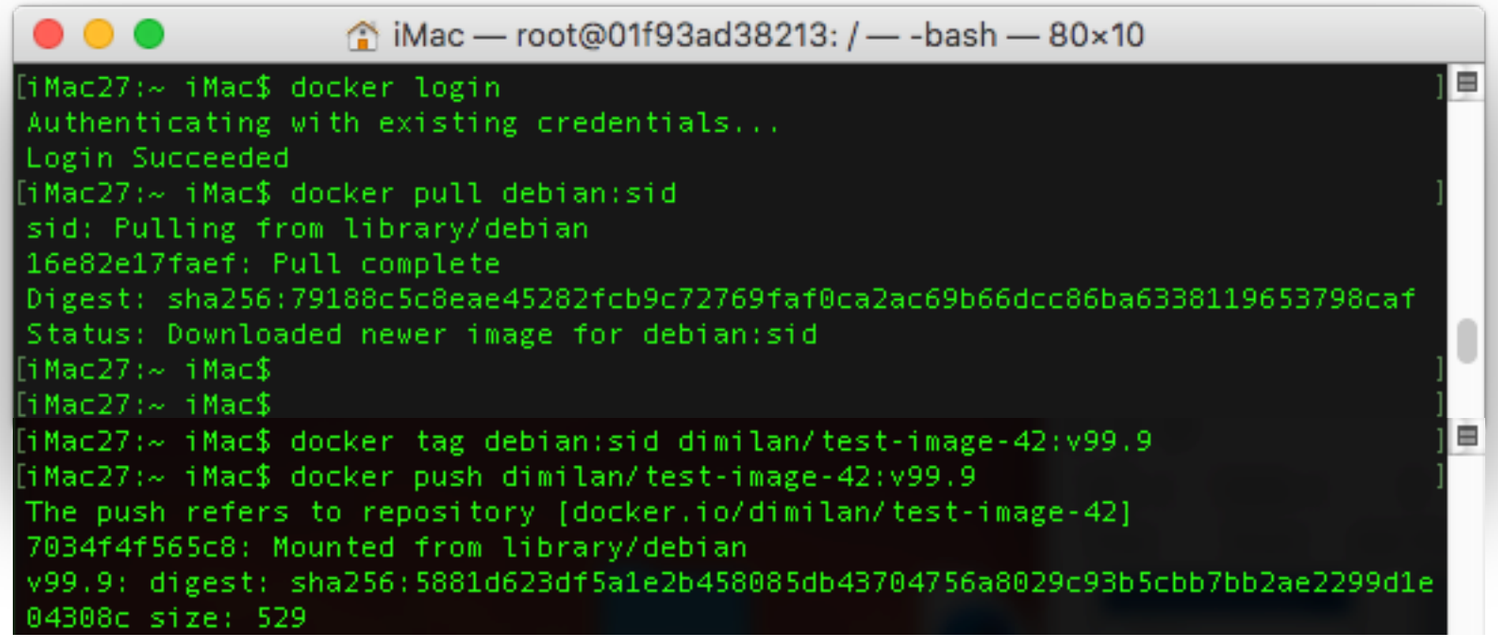
Sign up Log In

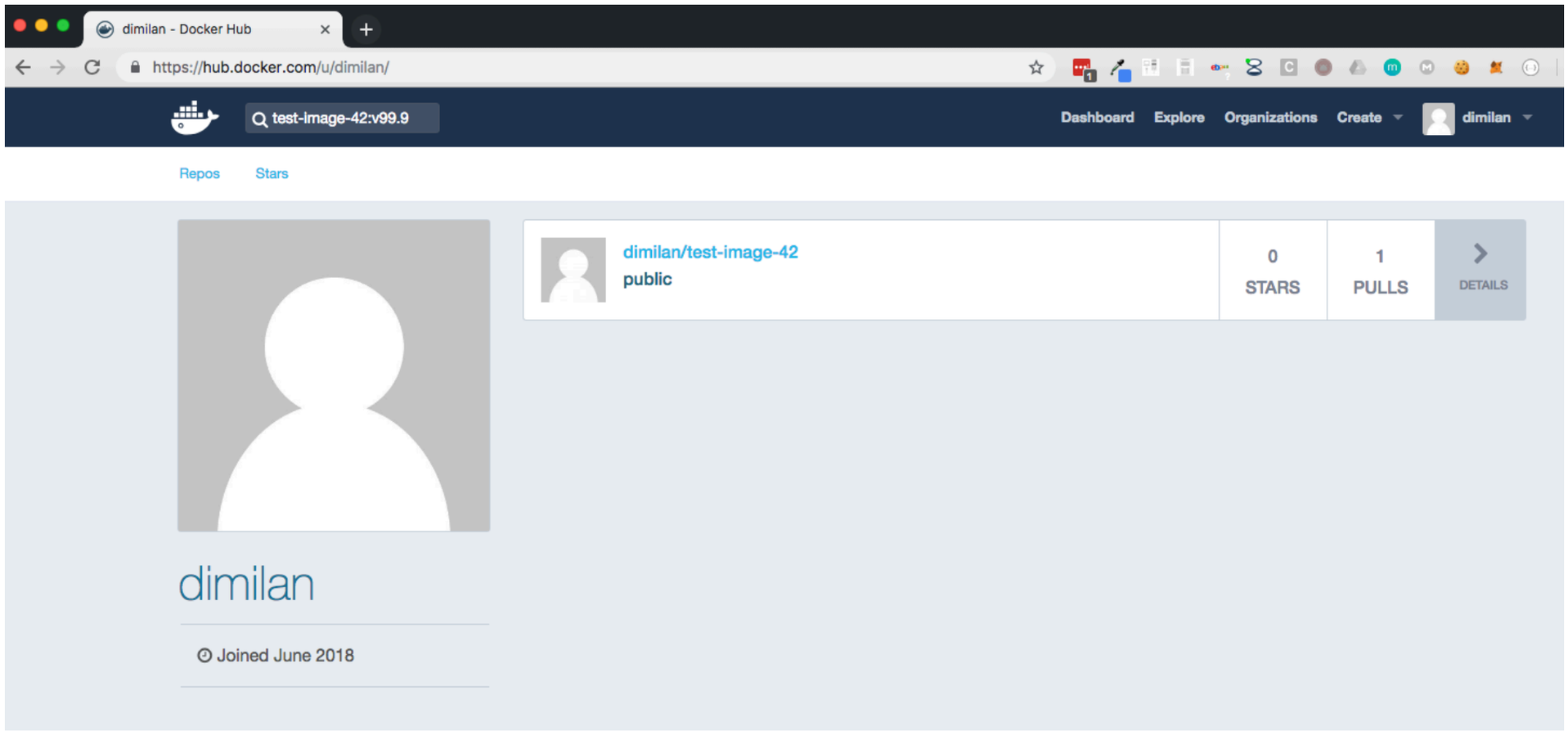
Repositories (8704)

All

node official	2.3K STARS	10M+ PULLS	DETAILS
strongloop/node public | automated build	28 STARS	9.9K PULLS	DETAILS
bitnami/node public | automated build	15 STARS	10K+ PULLS	DETAILS
cusspvz/node public | automated build	3 STARS	8.1K PULLS	DETAILS

https://hub.docker.com/_/node/


iMac — root@01f93ad38213: / — -bash — 80×10
iMac27:~ iMac$ docker login
Authenticating with existing credentials...
Login Succeeded
iMac27:~ iMac$ docker pull debian:sid
sid: Pulling from library/debian
16e82e17faef: Pull complete
Digest: sha256:79188c5c8eae45282fcb9c72769faf0ca2ac69b66dcc86ba6338119653798caf
Status: Downloaded newer image for debian:sid
iMac27:~ iMac$
iMac27:~ iMac$
iMac27:~ iMac$ docker tag debian:sid dimilan/test-image-42:v99.9
iMac27:~ iMac$ docker push dimilan/test-image-42:v99.9
The push refers to repository [docker.io/dimilan/test-image-42]
7034f4f565c8: Mounted from library/debian
v99.9: digest: sha256:5881d623df5a1e2b458085db43704756a8029c93b5cbb7bb2ae2299d1e
04308c size: 529


dimilan - Docker Hub
https://hub.docker.com/u/dimilan/
test-image-42:v99.9
Dashboard
Explore
Organizations
Create
dimilan
Repos
Stars
dimilan/test-image-42
public
0
STARS
1
PULLS
DETAILS
dimilan
Joined June 2018

Story Time

- Clean up your images regularly
- You will discover the images you really need to keep
- Be aware of how much you are trusting the containers you fetch

Geen wachtwoorden pushen!

What is a Dockerfile?

- This is a small "program" to create an image
- You run this program with

  `docker build -t` *`name-of-result`* `.` ← Note the `.` here

- When it finishes, the result will be in your local docker registry

https://docs.docker.com/engine/reference/builder/#usage

Producing the Next Image with Each Step

- Each line takes the image from the previous line and makes another image
- The previous image is unchanged
- It does not edit the state from the previous line
- You don’t want large files to span lines or your image will be huge

Dockerfile voorbeeld

https://docs.docker.com/engine/reference/builder/#dockerfile-examples

```
# Firefox over VNC
#
# VERSION                 0.3

FROM ubuntu

# Install vnc, xvfb in order to create a 'fake' display and firefox
RUN apt-get update && apt-get install -y x11vnc xvfb firefox
RUN mkdir ~/.vnc
# Setup a password
RUN x11vnc -storepasswd 1234 ~/.vnc/passwd
# Autostart firefox (might not be the best way, but it does the trick)
RUN bash -c 'echo "firefox" >> /.bashrc'

EXPOSE 5900
CMD    ["x11vnc", "-forever", "-usepw", "-create"]
```

Caching with Each Step

- This is important; watch the build output for “using cache ”
- Docker skips lines that have not changed since the last build
- If your first line is “download latest file,” it may not always run

Caching with Each Step

- This caching saves huge amounts of time
- The parts that change the most belong at the end of the Dockerfile

Not Shell Scripts

- Dockerfiles look like shell scripts
- Dockerfiles are *not* shell scripts
- Processes you start on one line will *not* be running on the next line
- Environment variables you set *will* be set on the next line

If you use the **ENV** command, remember that each line is its own call to docker run

The Most Basic Dockerfile

- **Put this in a file named `Dockerfile`:**

```
FROM busybox
RUN echo “building simple docker image.”
CMD echo “Hello Container”
```

example — root@6addff04bae8: / — -bash — 94×29

```
[iMac27:~ iMac$ mkdir example
[iMac27:~ iMac$ cd example
[iMac27:example iMac$ ls
 iMac27:example iMac$ nano Dockerfile
```

example — root@6addff04bae8: / — nano Dockerfile — 94×29

```
  GNU nano 2.0.6                File: Dockerfile                      Modified

FROM busybox
RUN echo "building simple docker image."
CMD echo "hello container"
```

example — root@6addff04bae8: / — -bash — 94×29

```
[iMac27:~ iMac$ mkdir example
[iMac27:~ iMac$ cd example
[iMac27:example iMac$ ls
[iMac27:example iMac$ nano Dockerfile
[iMac27:example iMac$ docker build -t hello .
 Sending build context to Docker daemon  2.048kB
 Step 1/3 : FROM busybox
 latest: Pulling from library/busybox
 90e01955edcd: Pull complete
 Digest: sha256:2a03a6059f21e150ae84b0973863609494aad70f0a80eaeb64bddd8d92465812
 Status: Downloaded newer image for busybox:latest
  ---> 59788edf1f3e
 Step 2/3 : RUN echo "building simple docker image."
  ---> Running in d3238fcf06ac
 building simple docker image.
 Removing intermediate container d3238fcf06ac
  ---> 5dddb945f091
 Step 3/3 : CMD echo "hello container"
  ---> Running in 081a16fb7bf5
 Removing intermediate container 081a16fb7bf5
  ---> 65688107ddc7
 Successfully built 65688107ddc7
 Successfully tagged hello:latest
 iMac27:example iMac$
```

^G Get
^X Exit

nieuwe image

build command

naam

```
iMac27:example iMac$ docker build -t hello .
```

example — root@6addff04bae8: / — -bash — 94×29

```
[iMac27:example iMac$ docker images
REPOSITORY                    TAG                 IMAGE ID            CREATED             SIZE
hello                         latest              65688107ddc7        3 minutes ago       1.15M
B
ubuntu                        latest              93fd78260bd1        9 days ago          86.2M
B
dimilan/test-image-42         v99.9               ff12142764fa        13 days ago         112MB
debian                        sid                 ff12142764fa        13 days ago         112MB
busybox                       latest              59788edf1f3e        8 weeks ago         1.15M
B
hyperledger/fabric-ca         x86_64-1.1.0        72617b4fa9b4        8 months ago        299MB
hyperledger/fabric-orderer    x86_64-1.1.0        ce0c810df36a        8 months ago        180MB
hyperledger/fabric-peer       x86_64-1.1.0        b023f9be0771        8 months ago        187MB
hyperledger/fabric-ccenv      x86_64-1.1.0        c8b4909d8d46        8 months ago        1.39G
B
hyperledger/fabric-couchdb    x86_64-0.4.6        7e73c828fc5b        9 months ago        1.56G
B
iMac27:example iMac$
```

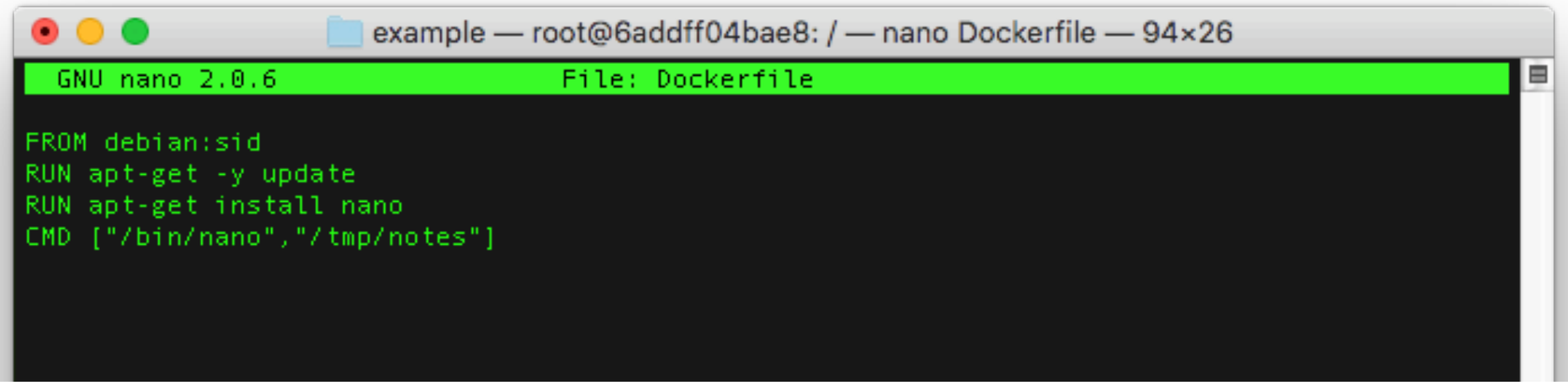
example — root@6addff04bae8: / — -bash — 94×6

```
[iMac27:example iMac$ docker run --rm hello
 hello container
 iMac27:example iMac$
```

Installing a Program with Docker Build

- **Put this in a Dockerfile:**

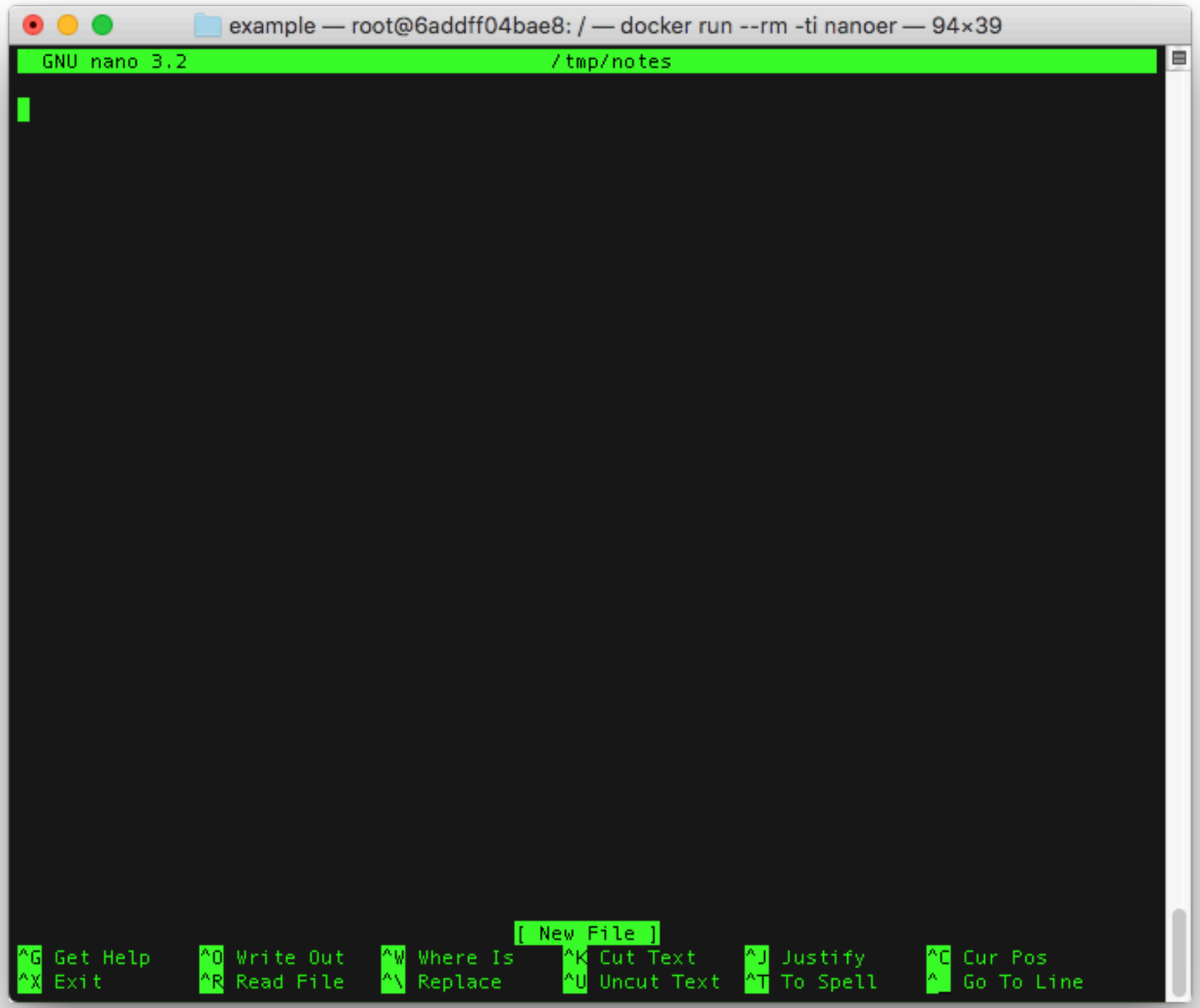
```
FROM debian:sid
RUN apt-get -y update
RUN apt-get install nano
CMD "nano" "/tmp/notes"
```



example — root@6addff04bae8: / — -bash — 94×39

```
[iMac27:example iMac$ nano Dockerfile
[iMac27:example iMac$ docker build -t nanoer .
Sending build context to Docker daemon  2.048kB
Step 1/4 : FROM debian:sid
 ---> ff12142764fa
Step 2/4 : RUN apt-get -y update
 ---> Using cache
 ---> 04af5df5e8dd
Step 3/4 : RUN apt-get install nano
 ---> Running in 4cd0723065c8
Reading package lists...
Building dependency tree...
Reading state information...
Suggested packages:
  spell
The following NEW packages will be installed:
  nano
0 upgraded, 1 newly installed, 0 to remove and 16 not upgraded.
Need to get 542 kB of archives.
After this operation, 2264 kB of additional disk space will be used.
Get:1 http://cdn-fastly.deb.debian.org/debian sid/main amd64 nano amd64 3.2-1 [542 kB]
debconf: delaying package configuration, since apt-utils is not installed
Fetched 542 kB in 1s (1030 kB/s)
Selecting previously unselected package nano.
(Reading database ... 6567 files and directories currently installed.)
Preparing to unpack .../archives/nano_3.2-1_amd64.deb ...
Unpacking nano (3.2-1) ...
Setting up nano (3.2-1) ...
update-alternatives: using /bin/nano to provide /usr/bin/editor (editor) in auto mode
update-alternatives: using /bin/nano to provide /usr/bin/pico (pico) in auto mode
Removing intermediate container 4cd0723065c8
 ---> 5e7b31ceebf4
Step 4/4 : CMD ["/bin/nano","/tmp/notes"]
 ---> Running in 5c8a9a582d71
Removing intermediate container 5c8a9a582d71
 ---> ec28d10986e8
Successfully built ec28d10986e8
Successfully tagged nanoer:latest
iMac27:example iMac$
```

```
iMac27:example iMac$ docker run --rm -ti nanoer
```



Adding a File through Docker Build

- **Put this in a Dockerfile:**

```
FROM example/nanoer
ADD notes.txt /notes.txt
CMD "nano" "/notex.txt"
```

```
 GNU nano 2.0.6                File: Dockerfile                      Modified

FROM example/nanoer
ADD notes.txt /notes.txt
CMD ["/bin/nano", "/notes.txt"]
```

local file

```
Save modified buffer (ANSWERING "No" WILL DESTROY CHANGES) ?
 Y Yes
 N No                ^C Cancel
```

```
$ ls
Dockerfile
$ nano Dockerfile
$ nano notes.txt
```

```
$ ls
Dockerfile      notes.txt
$ nano Dockerfile
$ nano notes.txt
$ docker build -t example/notes .
Sending build context to Docker daemon 3.072 kB
Step 1 : FROM example/nanoer
 ---> 1d27f07cc694
Step 2 : ADD notes.txt /notes.txt
 ---> 229b66d39146
Removing intermediate container 5a4bc53fc69a
Step 3 : CMD /bin/nano /notes.txt
 ---> Running in 4bfc208684ac
 ---> 330c5444fc3c
Removing intermediate container 4bfc208684ac
Successfully built 330c5444fc3c
$
$ docker run -ti --rm example/notes
```

```
  GNU nano 2.5.3                File: /notes.txt

TODO: learn more about dockerfiles

                          [ Read 1 line ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

The FROM Statement

- Which image to download and start from
- Must be the first command in your Dockerfile

The MAINTAINER Statement

- Defines the author of this Dockerfile

```
MAINTAINER Firstname Lastname <email@example.com>
```

The RUN Statement

- Runs the command line, waits for it to finish, and saves the result

```
RUN unzip install.zip /opt/install/
RUN echo hello docker
```

The ADD Statement

- Adds local files

```
ADD run.sh /run.sh
```

- Adds the contents of tar archives

```
ADD project.tar.gz /install/
```

- Works with URLs as well

```
ADD https://project.example.com/download/
1.0/project.rpm /project/
```

The ENV Statement

- Sets environment variables
- Both during the build and when running the result

```
ENV DB_HOST=db.production.example.com
ENV DB_PORT=5432
```

The ENTRYPOINT and CMD Statements

- `ENTRYPOINT` specifies the start of the command to run
- `CMD` specifies the whole command to run
- If you have both `ENTRYPOINT` and `CMD`, they are combined together

The ENTRYPOINT and CMD Statements

- If your container acts like a command-line program, you can use `ENTRYPOINT`

- If you are unsure, you can use `CMD`

Shell Form vs. Exec Form

- ENTRYPOINT RUN and CMD can use either form.
- Shell form looks like this:

```
nano notes.txt
```

- Exec form looks like this:

```
["/bin/nano", "notes.txt"]
```

The EXPOSE Statement

- Maps a port into the container

```
EXPOSE 8080
```

The VOLUME Statement

- Defines shared or ephemeral volumes

```
VOLUME [“/host/path/” “/container/path/”]
VOLUME [“/shared-data”]
```

- Avoid defining shared folders in Dockerfiles

The WORKDIR Statement

- Sets the directory the container starts in

```
WORKDIR /install/
```

The USER Statement

- Sets which user the container will run as

```
USER arthur
```

```
USER 1000
```

Chrome File Edit View History Bookmarks People Window Help

Dockerfile reference

https://docs.docker.com/engine/reference/builder/

docker

Support Training Docs Blog Docker Hub Get Started

Products Customers Community Partners Company Open Source

Docker Engine

Quickstart

Understand the architecture

Install

User guide

Administrate

Secure Engine

Extend Engine

Engine reference

Dockerfile reference

Docker can build images automatically by reading the instructions from a **`Dockerfile`**. A **`Dockerfile`** is a text document that contains all the commands a user could call on the command line to assemble an image. Using **`docker build`** users can create an automated build that executes several command-line instructions in succession.

This page describes the commands you can use in a **`Dockerfile`**. When you are done reading this page, refer to the **`Dockerfile`** Best Practices for a tip-oriented guide.

Usage

The **`docker build`** command builds an image from a **`Dockerfile`** and a *context*. The build's context is the files at a specified location **`PATH`** or **`URL`**. The **`PATH`** is a location of a Git repository.

A context is processed recursively. So, a **`PATH`** includes any subdirectories and the **`URL`** includes the repository and its submodules. A simple build command that uses the

https://docs.docker.com/engine/reference/builder

On this page:

vives

Oefeningen

1. Install Docker
2. Run 2 Ubuntu images (server client) waarbij je berichten kan verzenden van de server naar de client vie netcat
3. Run 2 ubuntu images en zorg dat je van de ene naar de andere kan pingen
4. Run een Ubuntu image, maak een file aan en sla de image op onder een andere naam
5. Run Ubuntu met arguments (-c) waarbij dat je de Distributie van de image naar een file op een shared volume op je host schrijft.
6. Creëer een network waar 3 ubuntu images in draaien en zorg ervoor dat ze berichten ontvangen via netcat
7. Run een ubuntu image en voeg in een andere terminal venster een nieuw proces toe (vrij te kiezen)
8. Zoek de 3 leukste docker containers op Docker Hub waarvan minstens een officieel en zorg dat je ze kan runnen. (vergeet niet de ports te exposen voor websites bv) (**1 indienen!**)