

DOCKER 101



docker

Johannes Sim & Renzo Veldkamp
Versie: 1.0

DOCKER 101

- Goal:
 - Practical 101 Docker
 - Tell / Show / Workshop
- Key take away:
 - What Docker is and can do for you
 - Doing Docker 101 with confidence

Audience: DEV colleagues

DISCLAIMER

- Our view and way of working
- Docker is changing constantly
 - Version 17.06.2-ce (17.09.0-ce)



CONTENT DOCKER 101

- **What & Why Docker?**
- **Examples**
 - static website in 1 container on 1 DockerHost
 - Linux host / Docker UI
- **(our) Docker workflow**
- **Docker commands explained**
- **Some Topics**

WHY DOCKER?

- No more: works on my laptop
- Time to Market
- CI/CD (Continuous Integration / Continuous Delivery)
 - Time from commit to in production (cycle time)
- DevOps
- *Try something new*
- *Testing*
-

OUR DOCKER DEFINITION

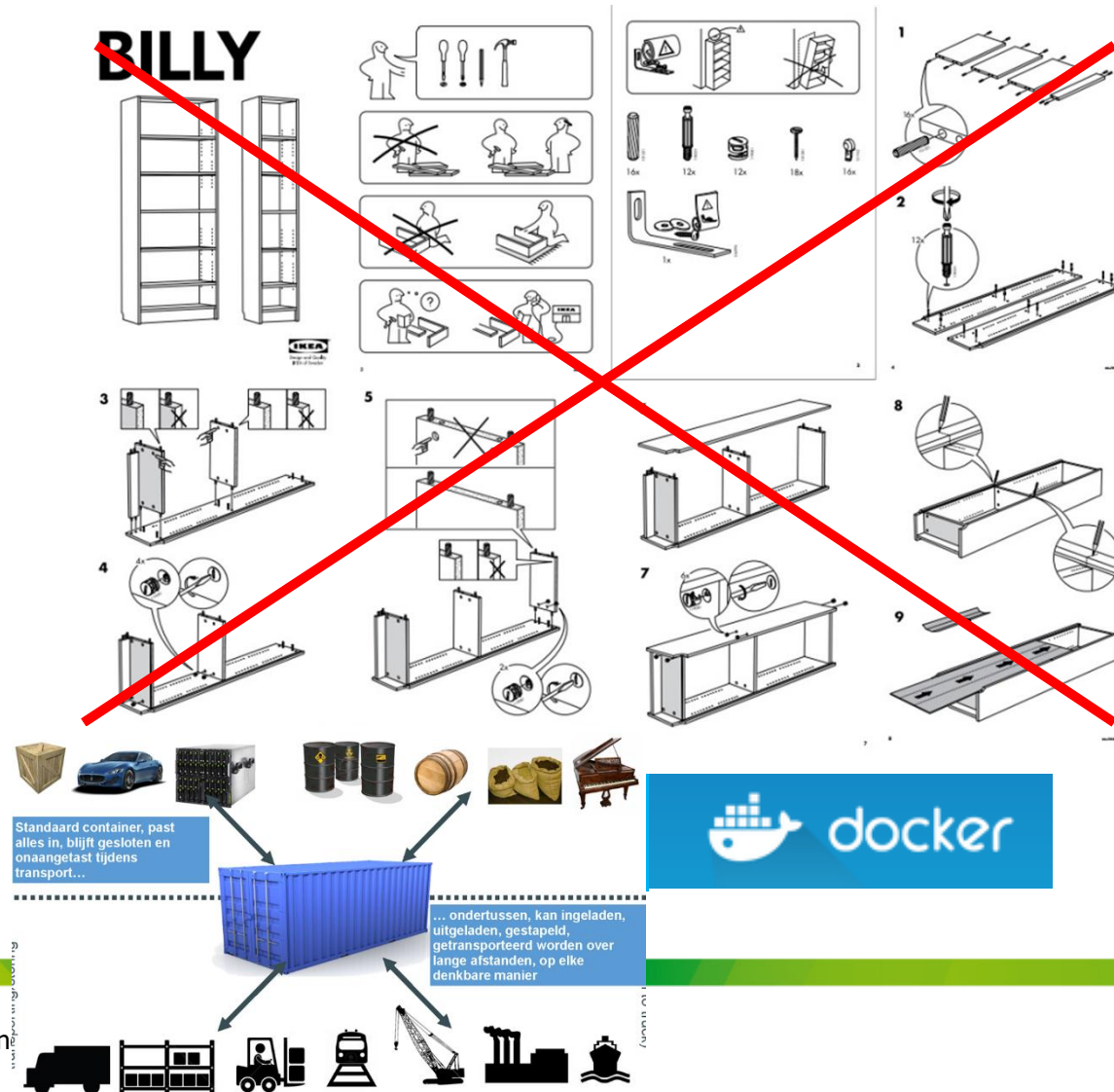
FOUND ON INTERNET

Is the world's leading software container platform available for developers, ops and businesses to **build, ship and run** any app on any infrastructure.

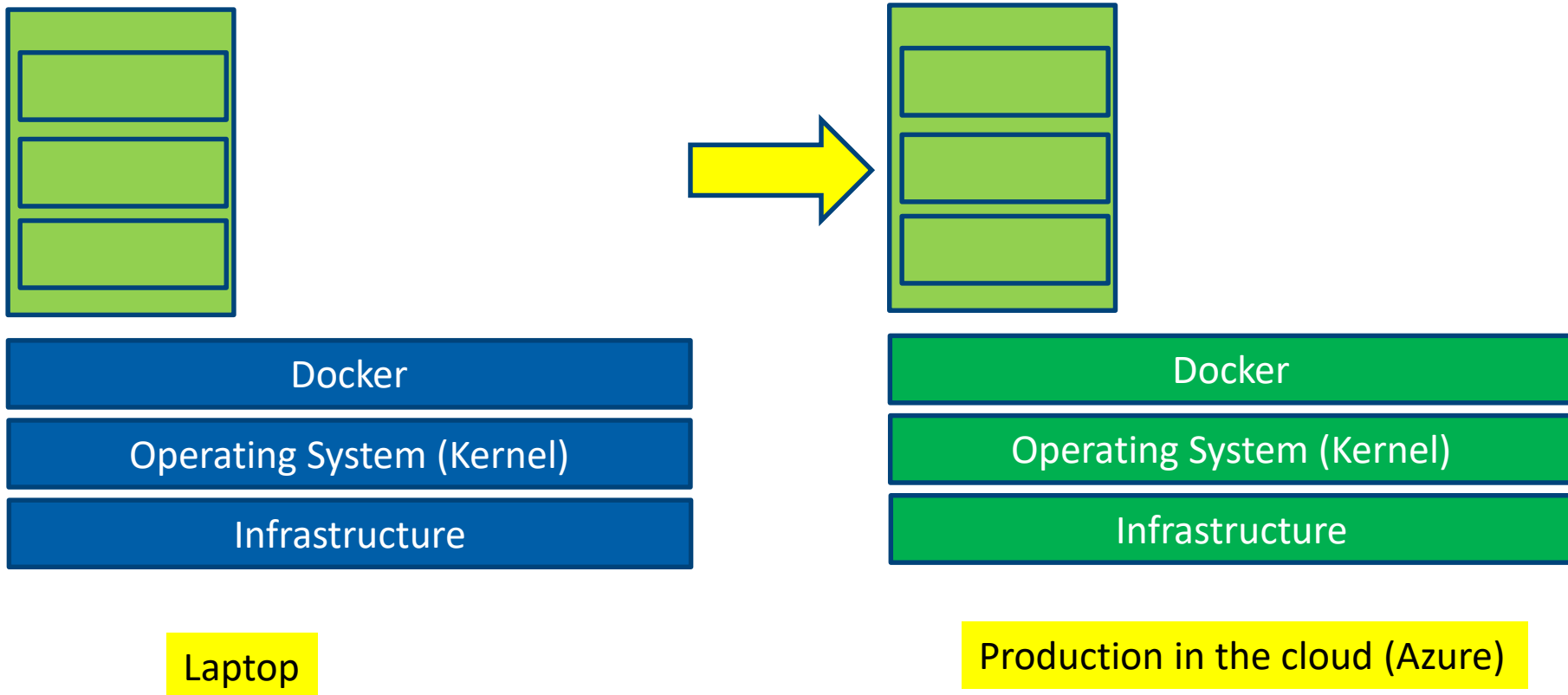
EXAMPLE: DOCKER USER STORIES

- As Product Owner I want features and an easy process to change and maintenance processes
- As DEV I want to build application only once and be sure that it is the same software
- As OPS I want the same software in each environment
- As Bus I want a short time to market for application feature

WORKS ON MY MACHINE APP IN A CONTAINER

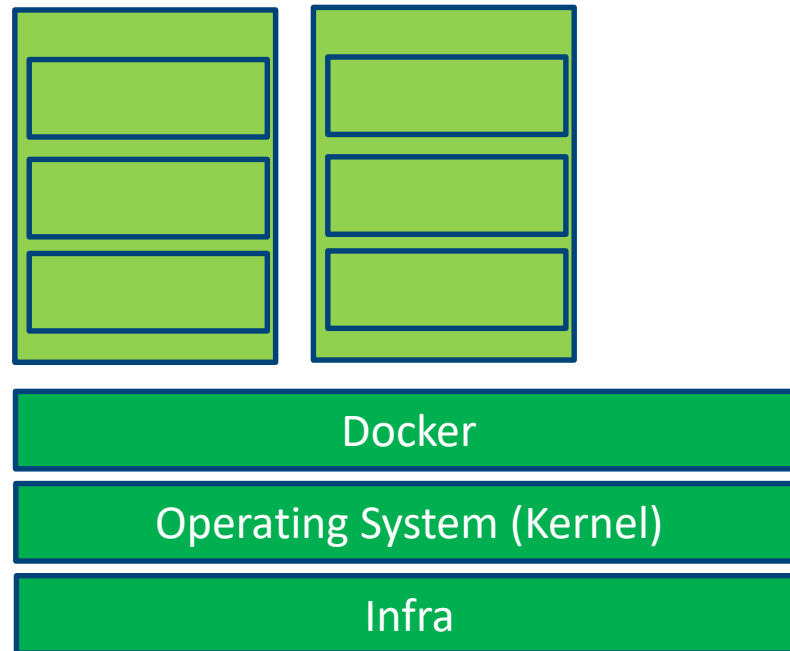


WORKS ON MY MACHINE RUN – CONTAINER

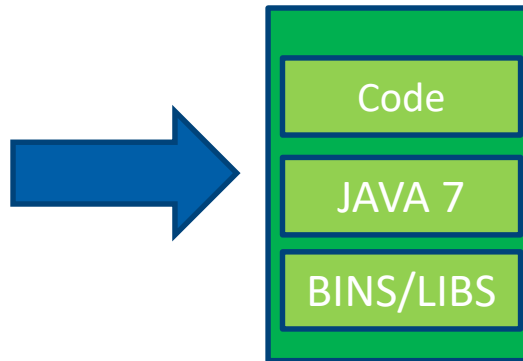


WORKS ON MY MACHINE

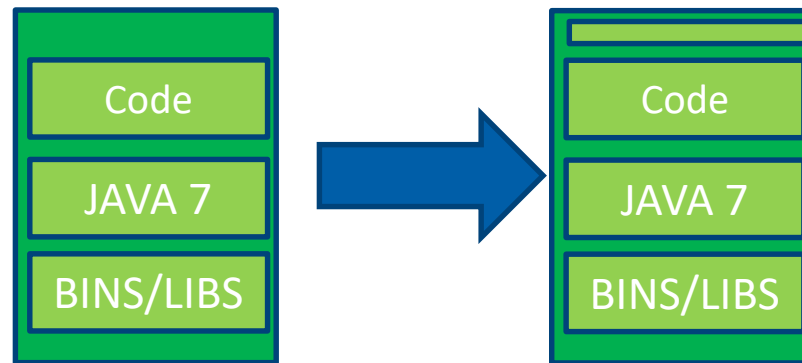
RUN - ISOLATION



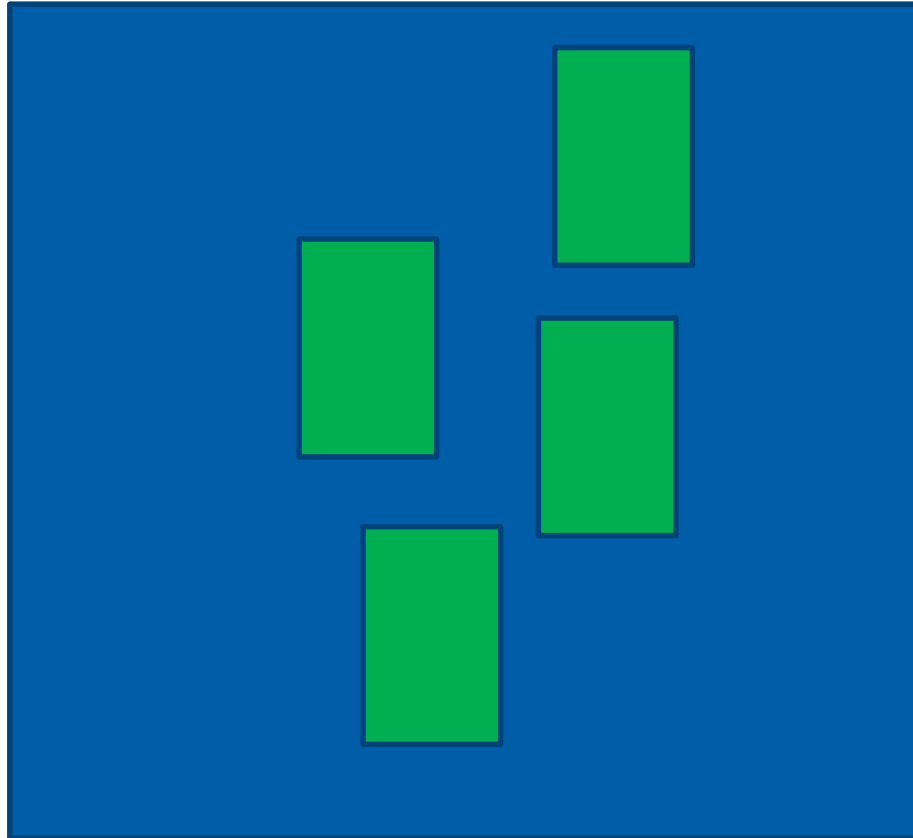
WORKS ON MY MACHINE BUILD IMAGE



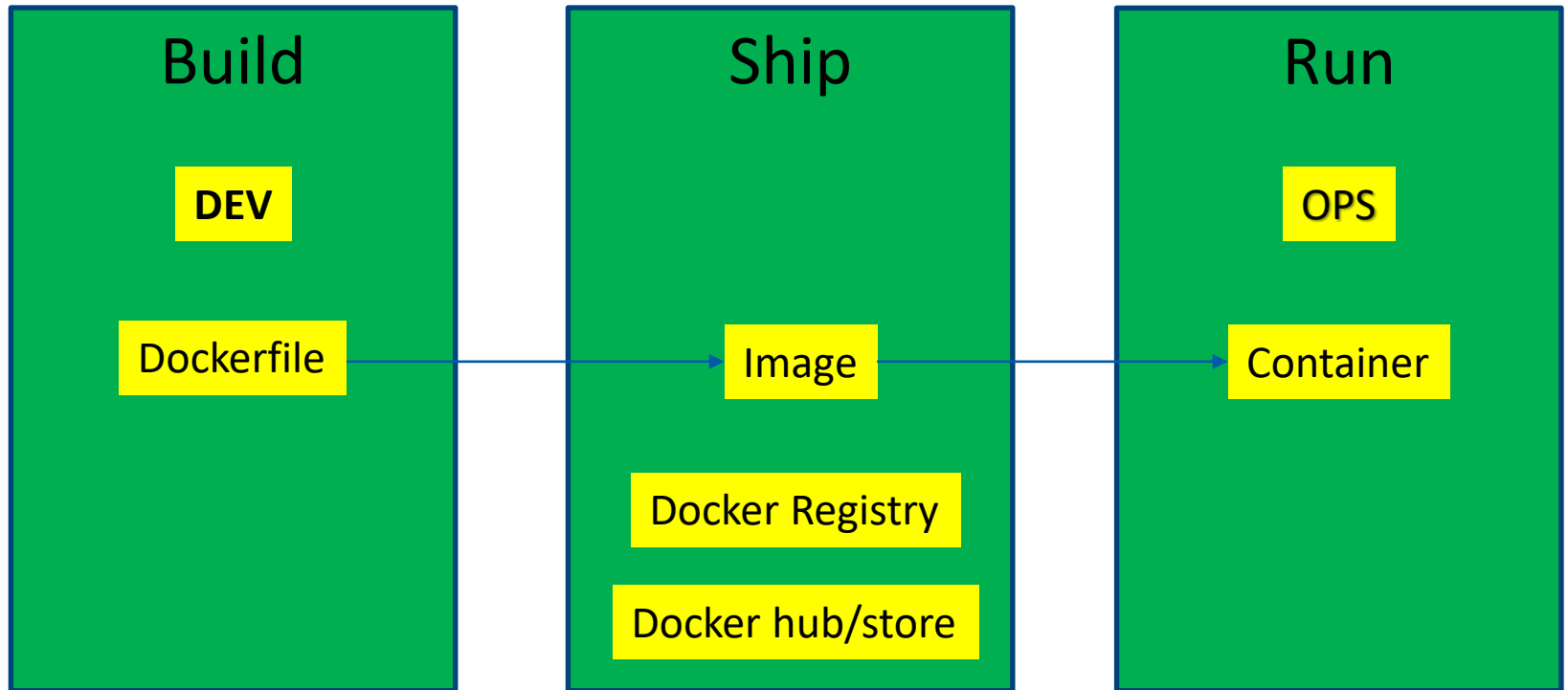
WORKS ON MY MACHINE IMAGE → CONTAINER



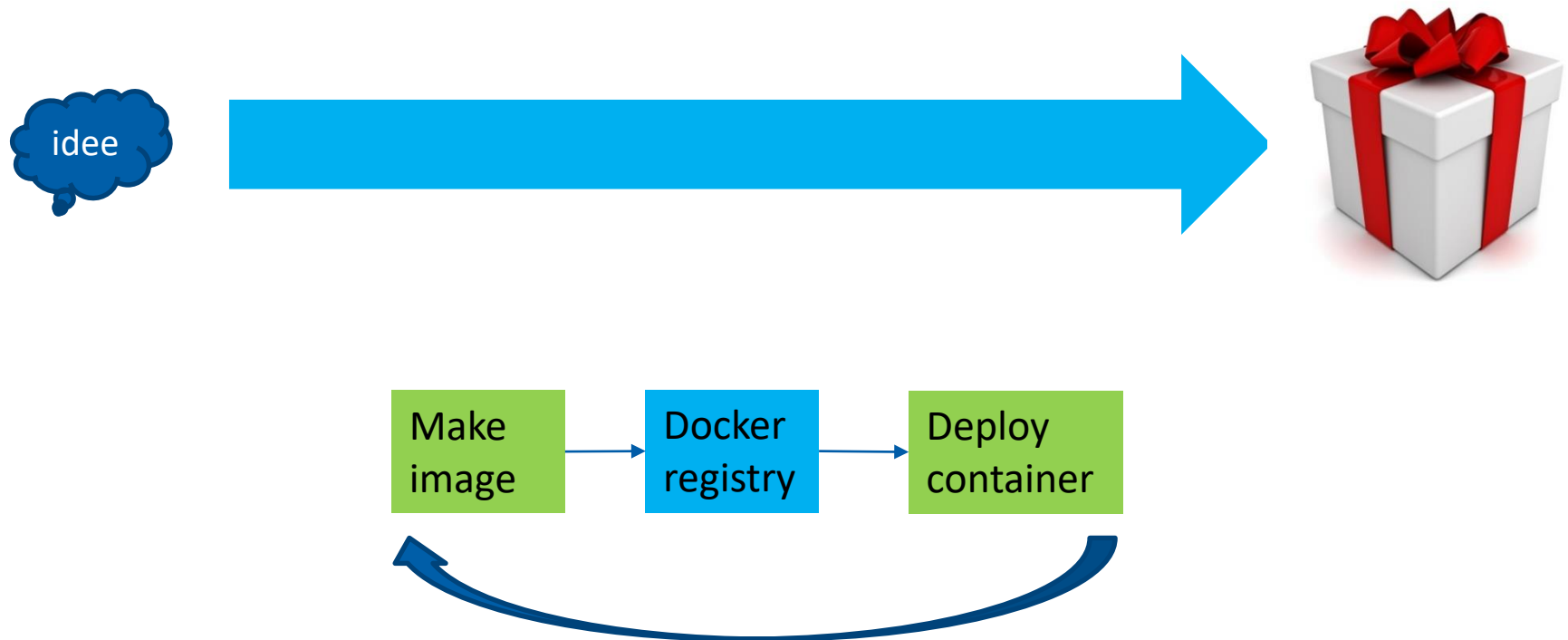
WORKS ON MY MACHINE SHARE – REGISTRY DOCKER HUB/STORE



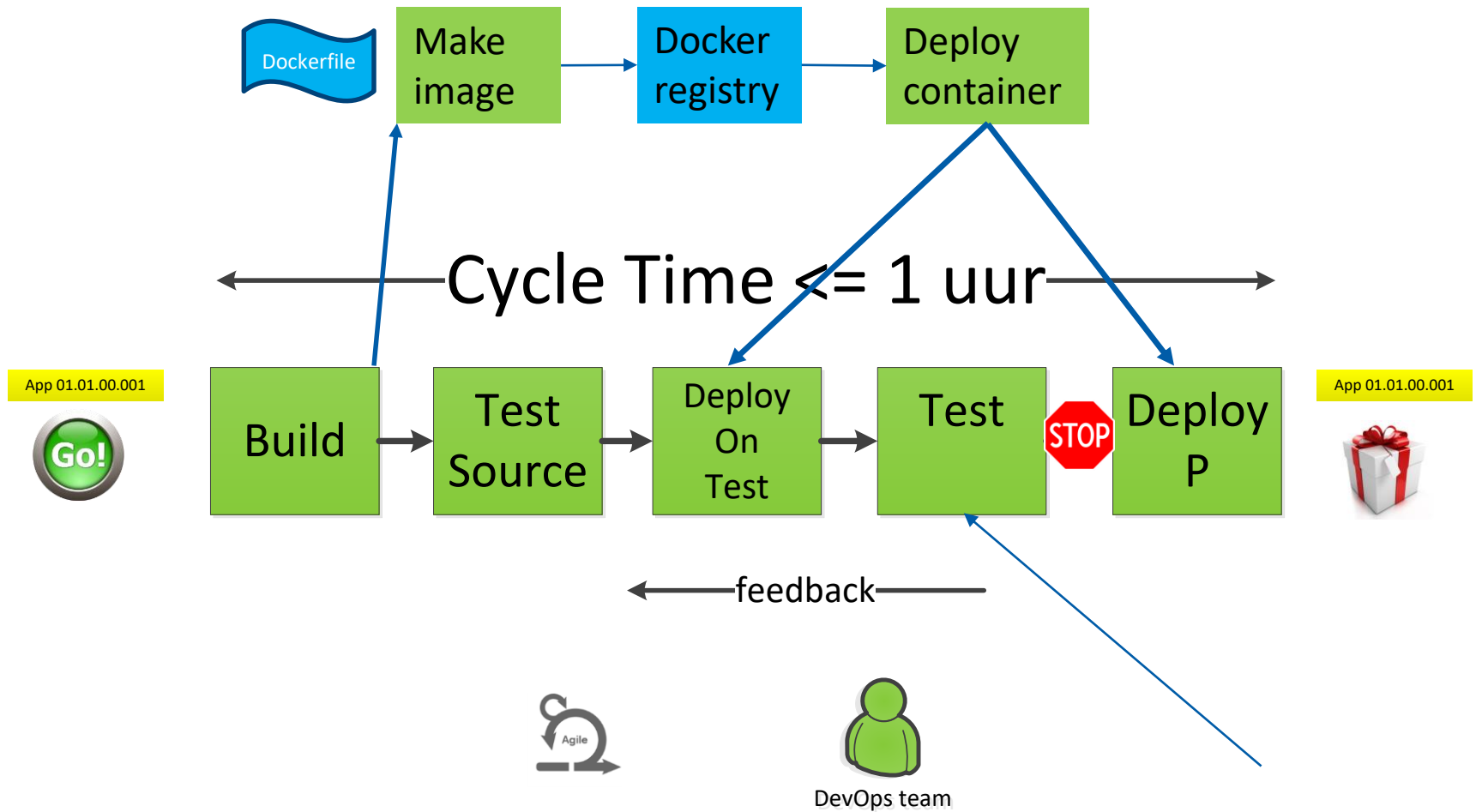
BUILD – SHIP - RUN



TIME TO MARKET

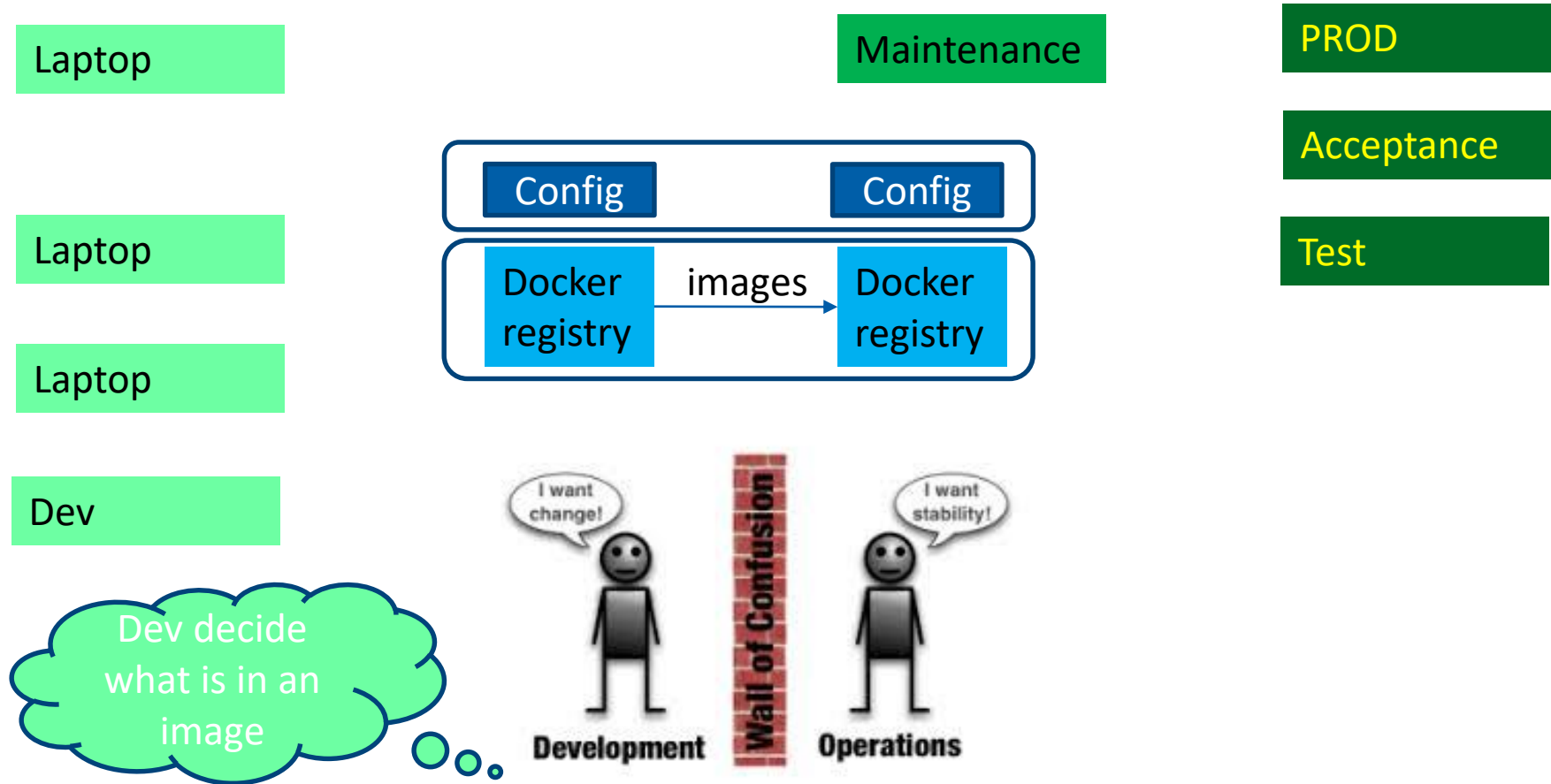


CI/CD



DEVOPS

IMAGE = INTERFACE DEV-OPS



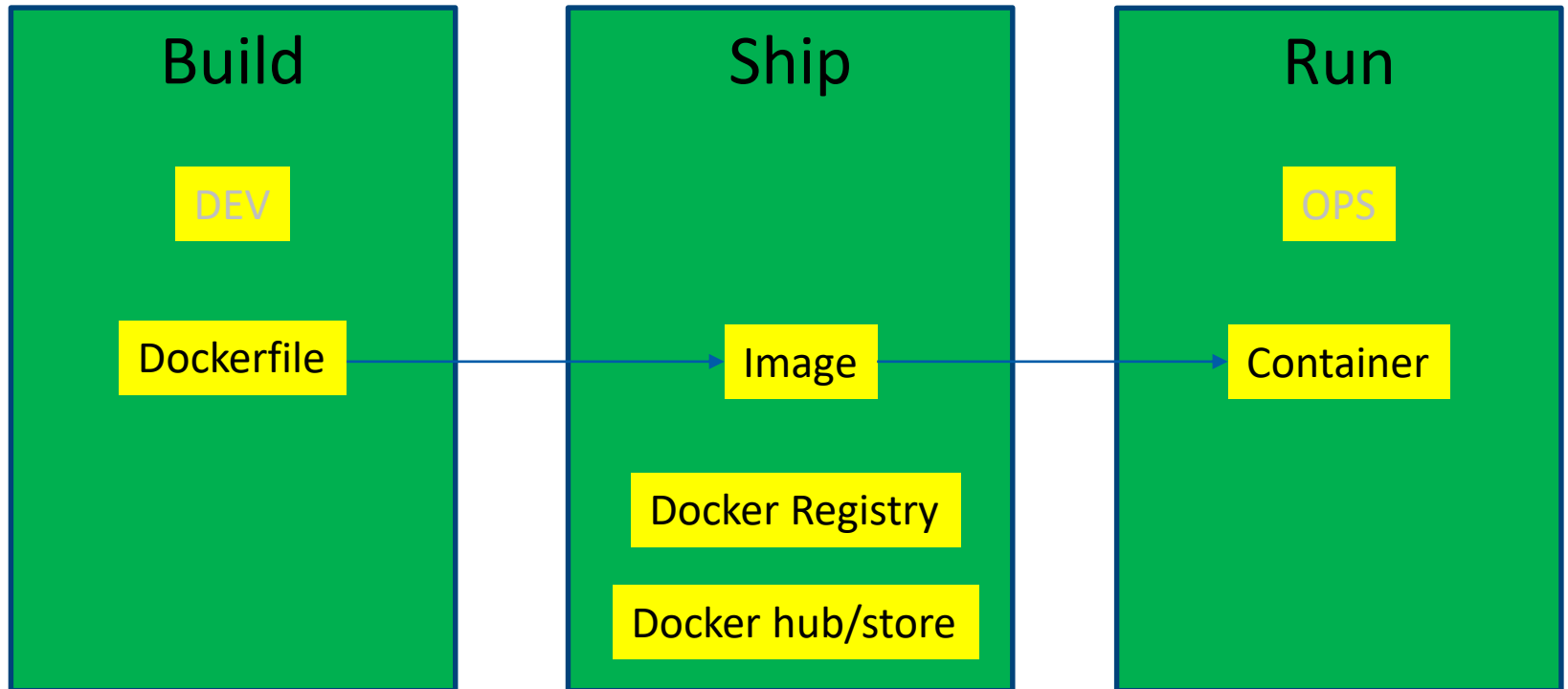
DOCKER “CONCEPTS”

- Container “standard”
- Build once – run anywhere
- Container
 - in “production”
 - Everything what is needed and nothing more
 - 1 process – 1 task
 - Immutable

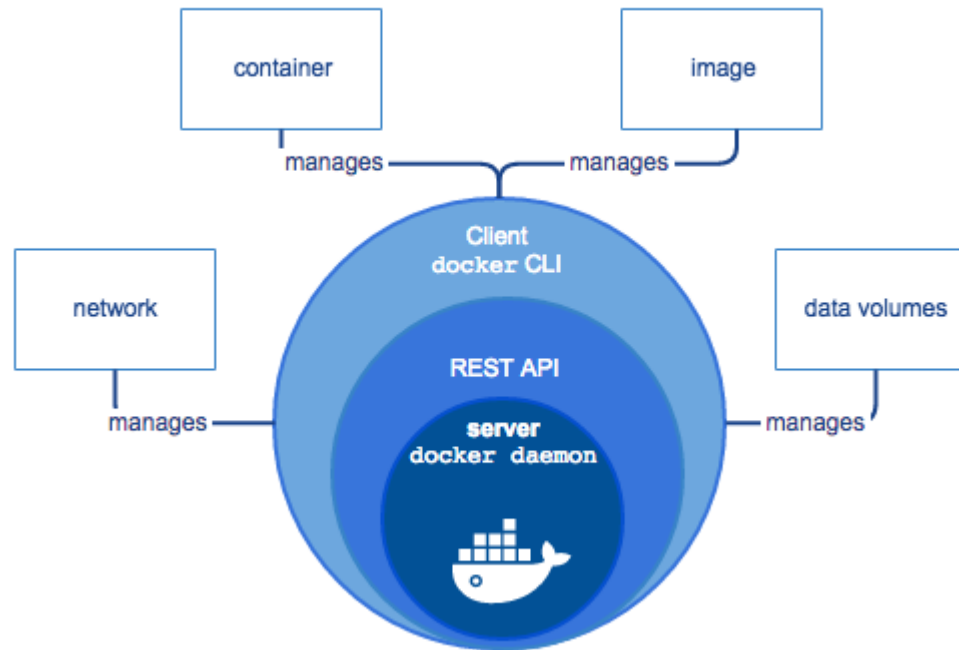
SUMMARY DOCKER

- DockerFile
 - blueprint
 - recipe
- Image
 - class
 - layers
- Registry (private) – Docker Hub – Docker store
 - “app store”
- Container
 - instance
 - running image

BUILD – SHIP – RUN SUMMARY

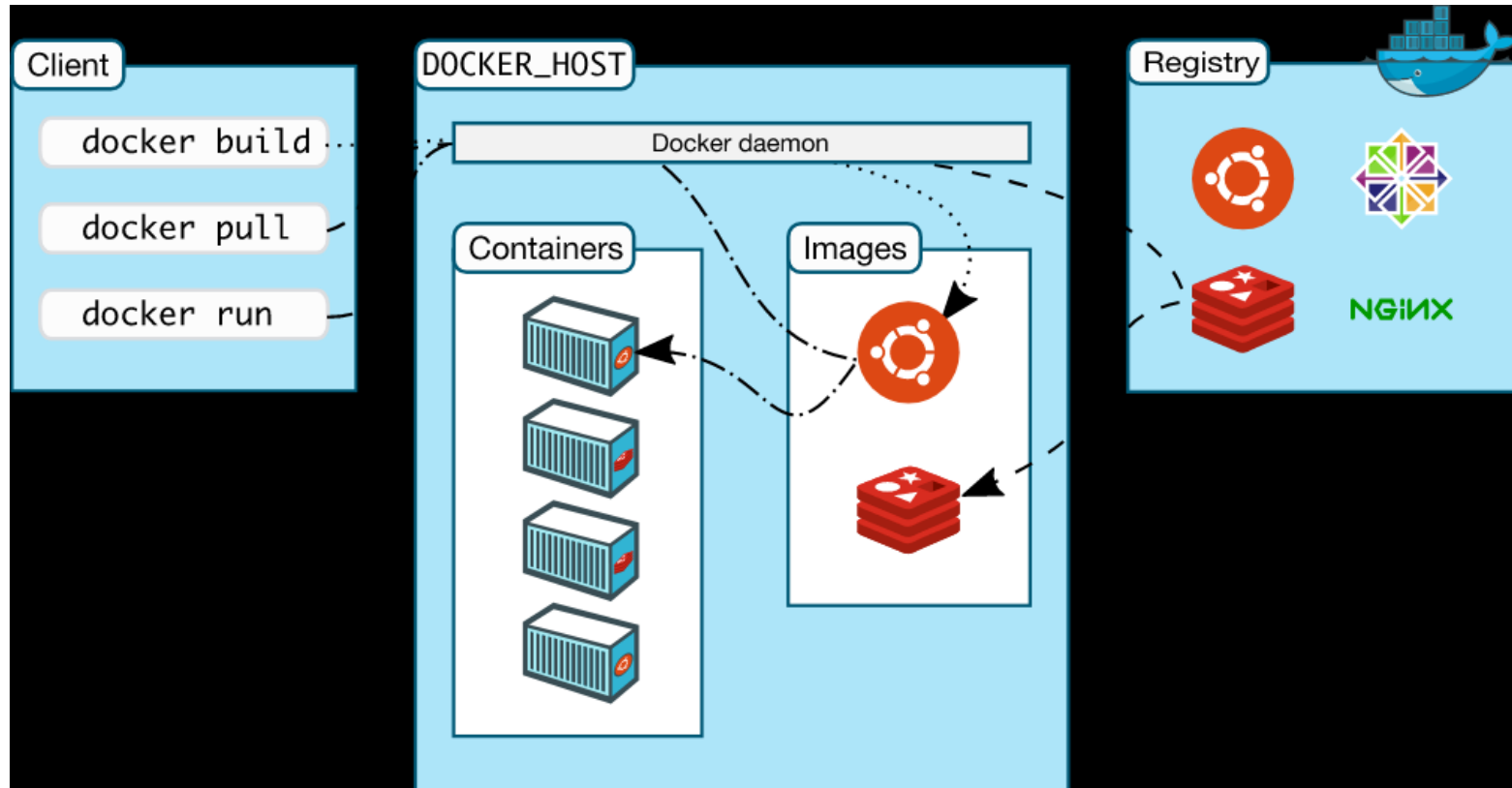


TOPIC: DOCKER ARCHITECTURE HIGH LEVEL



<https://docs.docker.com/engine/docker-overview/>

TOPIC: DOCKER ARCHITECTURE HIGH LEVEL



<https://docs.docker.com/engine/docker-overview/>

TOPIC:

VIRTUAL MACHINE VERSUS CONTAINER

- Docker containers are not VM's
- Analogy
 - Apartment building = VM
 - Apartment = Docker Container
 - Share of kernel
- Real world
 - DockerHost are VM's
 - Docker can also run directly on server with OS

TOPIC: DOCKER VERSIONS

- Containers are not new (Unix: 199x)
- Docker to open source: march 2013
- Docker 1.13 = Docker 17.03
 - Year.month
- CE (Community Edition)
 - 2 release channels
 - Edge each month
 - Stable each 3 month
- EE (Enterprise Edition)
- Current version: 17:09

EXAMPLE: OUR CONTAINER WORKFLOW

Case: Static website

- HTML pagina on webserver nginx
- Code: HTML
- Dockerfile
- Build an image
- Run container
- Show result
- Push to registry
- Pull from registry
- Run Azure DockerHost

EXAMPLE: HTML

```
1  <!DOCTYPE html>
2  <html>
3  <body>
4
5  <h1>Hello Static Website</h1>
6
7  <p>Continuously Delivered by:</p>
8
9  
10
11
12 <p>Versie: 01.00.00 - september 2017</p>
13 </body>
14 </html>
```

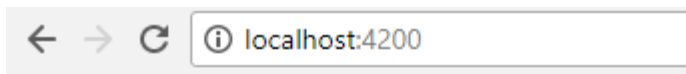
EXAMPLE DOCKERFILE

```
7 FROM nginx:stable-alpine
8 COPY index.html *.png /usr/share/nginx/html/
```

EXAMPLE DOCKER RUN

```
docker container run --name staticws -d -p 4200:80 centricms/staticws:latest
```

```
df16bf953be84b16fe3f8b67cd90d50a056b17b5a8a3fc4075a92634c3720b65
```



Hello Static Website

Continuously Delivered by:



Versie: 01.04.00 - september 2017

EXAMPLE PUSH

docker push centricms/staticws

The push refers to a repository [docker.io/centricms/staticws]

645034a2dbe2: Pushed

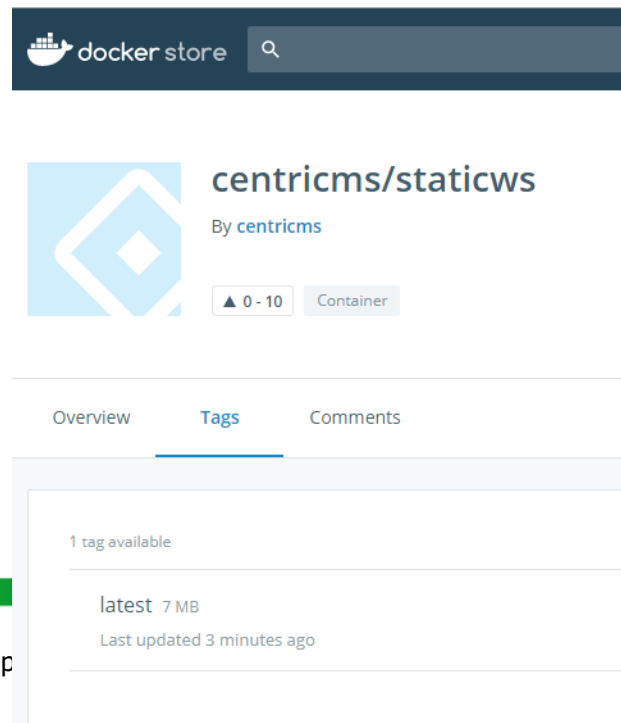
1690bc77acd5: Mounted from library/nginx

cbeb94c1f91a: Mounted from library/nginx

4bbcb364e643: Pushed

040fd7841192: Pushed

latest: digest: sha256:3b473ed6b592d06ea0f191972cf49ad13c62912733d8e0a7bb5c2a5319dbe5dd
size: 1362



EXAMPLE

PULL FROM AZURE HOST

docker pull centricms/staticws:latest

latest: Pulling from centricms/staticws

ab14e39f58e6: Pull complete

206c8e58fb89: Pull complete

aeb57280dce6: Pull complete

9a661d863527: Pull complete

0459f9b530b7: Pull complete

Digest: sha256:3b473ed6b592d06ea0f191972cf49ad13c62912733d8e0a7bb5c2a5319dbe5dd

Status: Downloaded newer image for centricms/staticws:latest

docker container run --name staticws -d -p 8901:80 centricms/staticws:latest

← → ↻ ⓘ dockerhost01.westeurope.cloudapp.azure.com:8901

Hello Static Website

Continuously Delivered by:



Versie: 01.04.00 - september 2017



OUR WAY OF WORKING

On laptop

- GIT
- VS CODE
- DockerForWindows

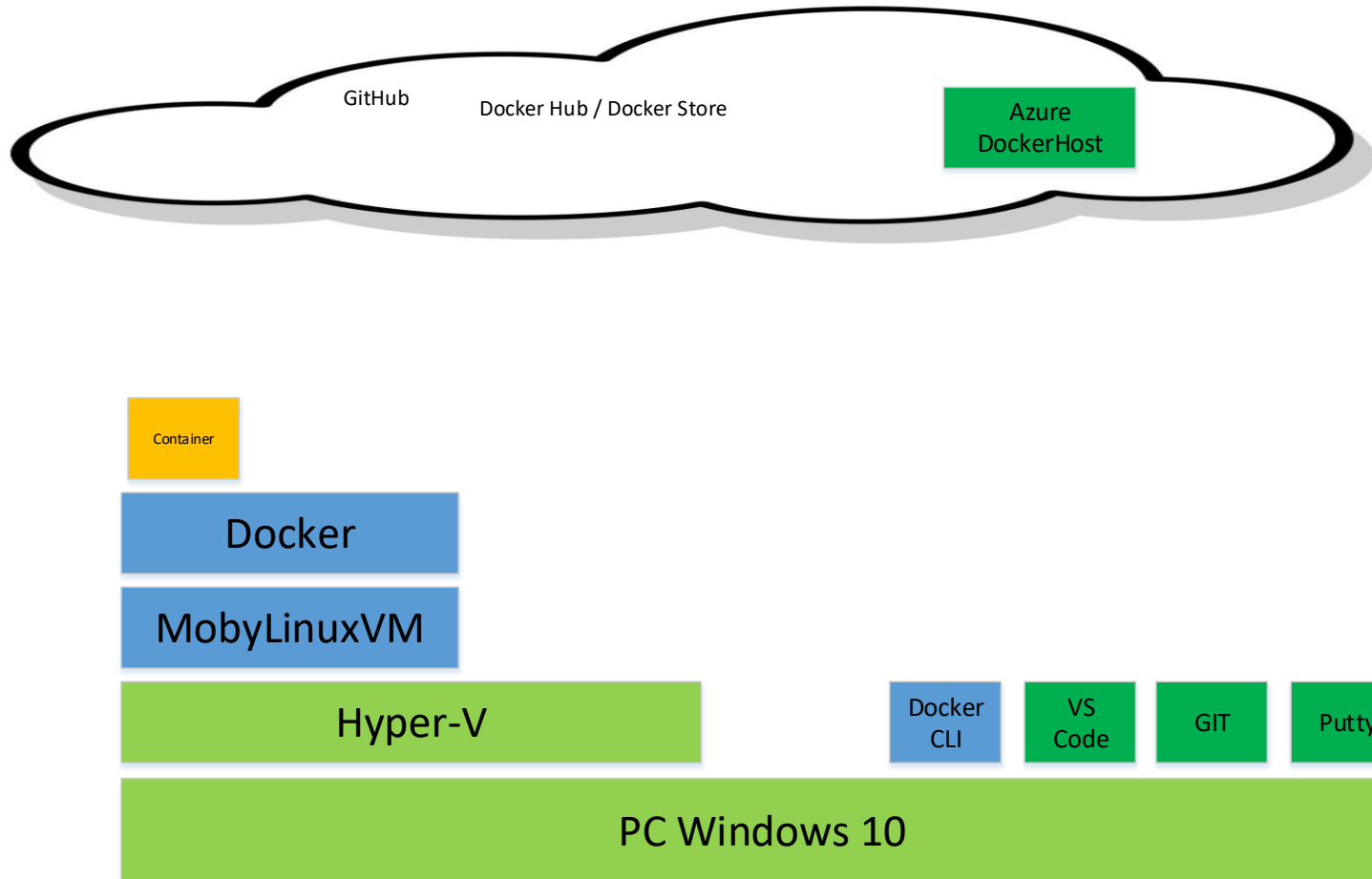
Internet

- GitHub
- Docker Hub / Docker Store

Cloud

- *DockerHost on Azure*

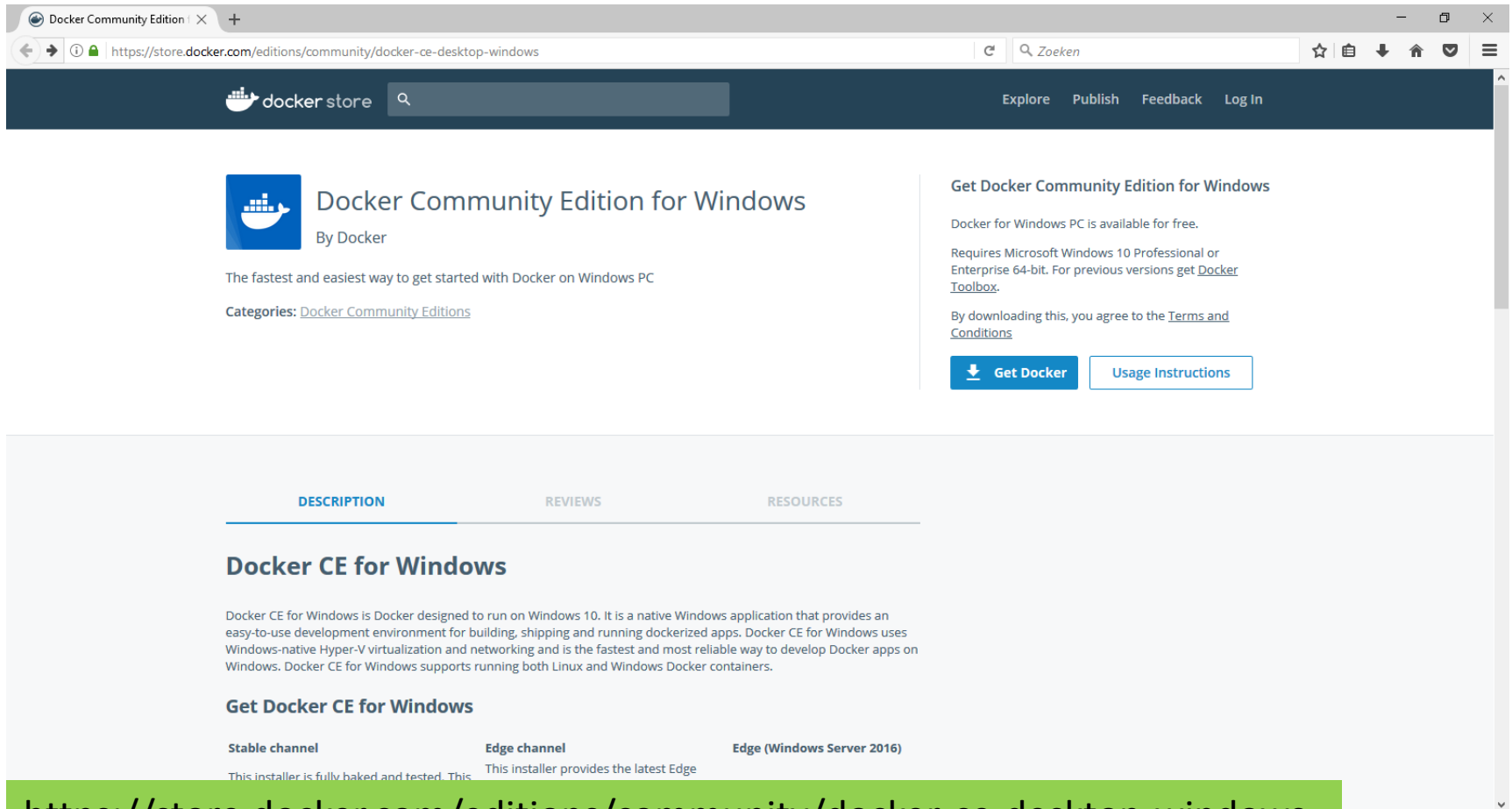
LAPTOP, INTERNET, CLOUD



INSTALL

- GIT
 - <https://git-scm.com/download/win>
- VS CODE
 - <https://code.visualstudio.com/docs/?dv=win32>
- Download GitHub repro
 - New directory
 - Clone repro
 - <https://github.com/RenzoVeldkamp/MicroContainer>
 - in directory
 - Git clone <https://github.com/RenzoVeldkamp/MicroContainer.git>

DOCKER DOWNLOADEN



The screenshot shows a web browser window with the address bar displaying <https://store.docker.com/editions/community/docker-ce-desktop-windows>. The page features the Docker logo and the title "Docker Community Edition for Windows" by Docker. It includes a description: "The fastest and easiest way to get started with Docker on Windows PC" and categories: "Docker Community Editions". On the right, there's a section titled "Get Docker Community Edition for Windows" with text stating "Docker for Windows PC is available for free." and "Requires Microsoft Windows 10 Professional or Enterprise 64-bit. For previous versions get [Docker Toolbox](#)." Below this, it says "By downloading this, you agree to the [Terms and Conditions](#)". There are two buttons: "Get Docker" and "Usage Instructions". At the bottom, there's a tabbed interface with "DESCRIPTION", "REVIEWS", and "RESOURCES". The "DESCRIPTION" tab is active, showing the title "Docker CE for Windows" and a paragraph: "Docker CE for Windows is Docker designed to run on Windows 10. It is a native Windows application that provides an easy-to-use development environment for building, shipping and running dockerized apps. Docker CE for Windows uses Windows-native Hyper-V virtualization and networking and is the fastest and most reliable way to develop Docker apps on Windows. Docker CE for Windows supports running both Linux and Windows Docker containers." Below this, there's a section "Get Docker CE for Windows" with three options: "Stable channel" (with a note "This installer is fully baked and tested. This"), "Edge channel" (with a note "This installer provides the latest Edge"), and "Edge (Windows Server 2016)".

Docker Community Edition for Windows
By Docker

The fastest and easiest way to get started with Docker on Windows PC

Categories: [Docker Community Editions](#)

Get Docker Community Edition for Windows

Docker for Windows PC is available for free.

Requires Microsoft Windows 10 Professional or Enterprise 64-bit. For previous versions get [Docker Toolbox](#).

By downloading this, you agree to the [Terms and Conditions](#)

[Get Docker](#) [Usage Instructions](#)

DESCRIPTION REVIEWS RESOURCES

Docker CE for Windows

Docker CE for Windows is Docker designed to run on Windows 10. It is a native Windows application that provides an easy-to-use development environment for building, shipping and running dockerized apps. Docker CE for Windows uses Windows-native Hyper-V virtualization and networking and is the fastest and most reliable way to develop Docker apps on Windows. Docker CE for Windows supports running both Linux and Windows Docker containers.

Get Docker CE for Windows

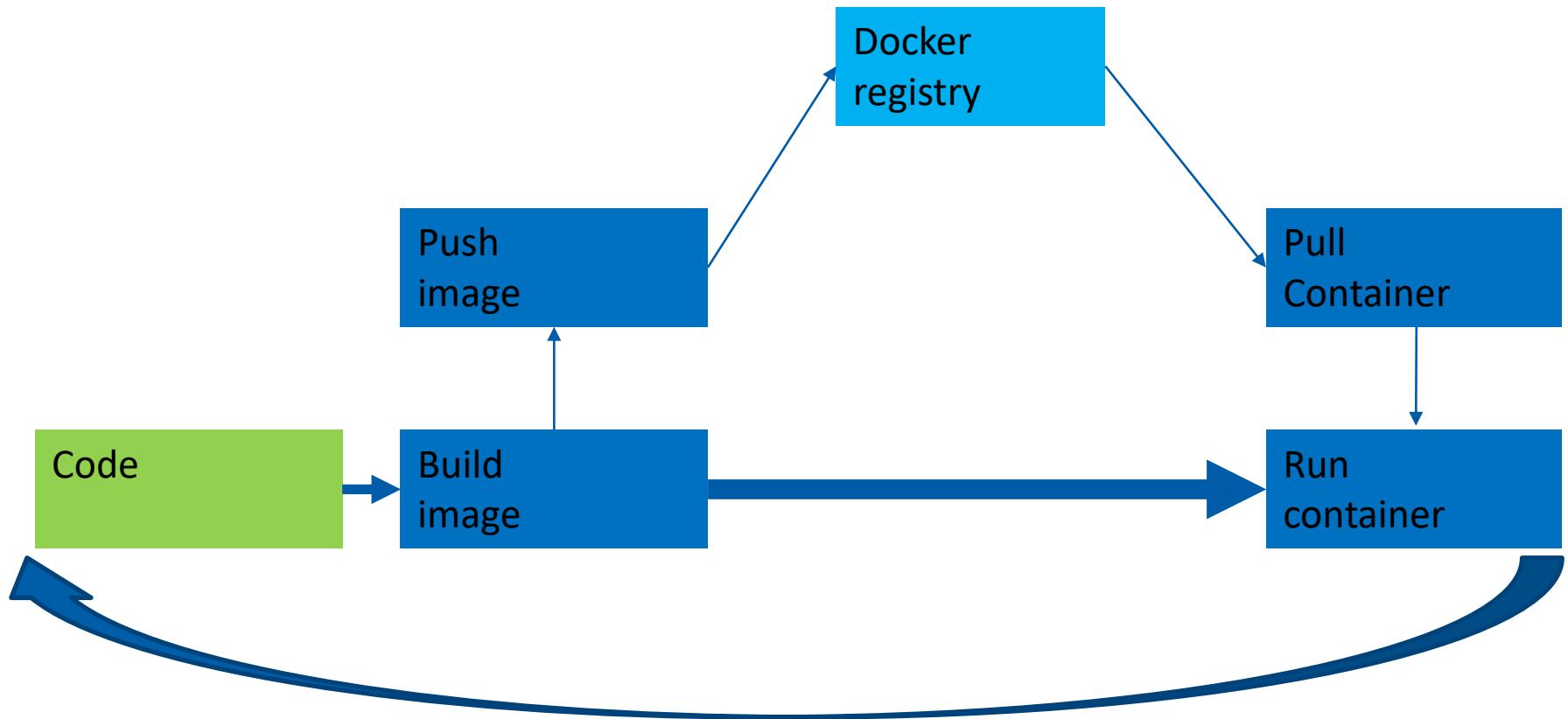
Stable channel
This installer is fully baked and tested. This

Edge channel
This installer provides the latest Edge

Edge (Windows Server 2016)

<https://store.docker.com/editions/community/docker-ce-desktop-windows>

WORKFLOW



COMMAND LINE INTERFACE (CLI)

- How:
 - **Powershell window in VS Code**
 - Powershell
 - CMD
- Docker <command>
- Docker management commands
 - docker container <command>
 - docker image <command>
- Docker --help
- <https://docs.docker.com/engine/reference>
- Google search

DOCKER COMMAND LINE INTERFACE

Build

Docker image ls

Docker image rm

Docker image build

Ship

Docker pull

Docker push

Run

Docker container run

Docker container ls

Docker container rm

Docker container exec

Docker container start

Docker container stop

Docker system info

Docker version

Docker system prune

WHICH DOCKER VERSION

- Docker version
- *Docker-compose version*
- *Docker-machine version*

```
Client:
Version:      17.06.1-ce
API version:  1.30
Go version:   go1.8.3
Git commit:   874a737
Built:        Thu Aug 17 22:48:20 2017
OS/Arch:      windows/amd64

Server:
Version:      17.06.1-ce
API version:  1.30 (minimum version 1.12)
Go version:   go1.8.3
Git commit:   874a737
Built:        Thu Aug 17 22:54:55 2017
OS/Arch:      linux/amd64
Experimental: true
```

- Docker info
 - Lots of info

DOCKER RUNNING CORRECTLY?

- docker container run hello-world

```
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
b04784fba78d: Pull complete
Digest: sha256:f3b3b28a45160805bb16542c9531888519430e9e6d6fffc09d72261b0d26ff74f
Status: Downloaded newer image for hello-world:latest

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.
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://cloud.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/engine/userguide/
```

DOCKER CONTAINER RUN

- docker container run hello-world
- Run container with the content of image called hello-world
- hello-world cannot be found on the DockerHost (Laptop)
- hello-world will be pulled from the Docker Hub / Store
- Hello-world will execute and end
- docker container run hello-world (once again)
 - Image on the Docker Host

WHICH CONTAINERS ARE RUNNING?

- **docker container ls**
- Will only show the running containers
- Old command: `docker ps`
 - backwards compatible

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

- **docker container ls -a**

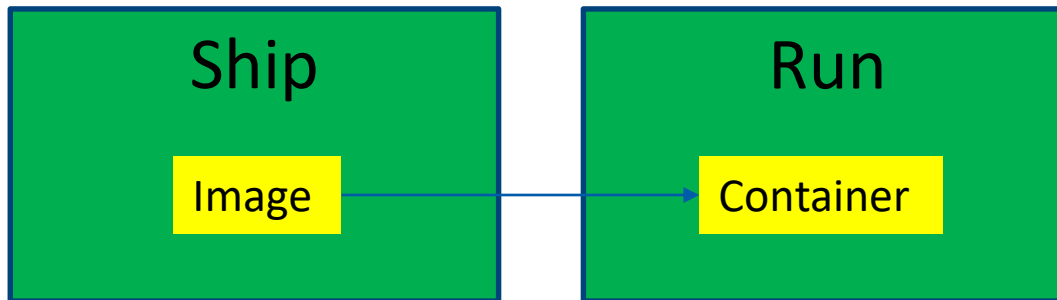
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
b5022e576368	hello-world	"/hello"	18 hours ago	Exited (0) 18 hours ago		jovial_booth

EASE OF DOCKER READY TO RUN - CONTAINERS

- Docker registry
- Docker Hub
 - <https://hub.docker.com/>
- Docker Store

EXAMPLE – EASE OF DOCKER READY TO RUN - CONTAINER

- Run a Linux machine on my laptop - Alpine
- Run a Docker UI - Portainer



- We show:
 - Docker store (docker hub)
 - Pull image
 - Run container
 - Show result

ALPINE REPO

The screenshot shows the Docker Store interface for the 'alpine' image. The browser address bar shows 'https://store.docker.com/images/alpine'. The page header includes the Docker Store logo, a search bar with 'alpine', and navigation links for 'Explore', 'Publish', and 'Feedback'. The user profile 'sim007' is visible in the top right.

alpine
By Docker

A minimal Docker image based on Alpine Linux with a complete package index and only 5 MB in size!

10M+ Pulls
Categories: [Featured Images](#), [Base Images](#), [Operating Systems](#)

Free Product Tier
\$0.00

[Terms of Service](#)

Copy and paste to pull this image

`docker pull alpine`

[View Available Tags](#)

DESCRIPTION | REVIEWS | RESOURCES

Supported tags and respective Dockerfile links

- [3.1 \(versions/library-3.1/Dockerfile\)](#)
- [3.2 \(versions/library-3.2/Dockerfile\)](#)
- [3.3 \(versions/library-3.3/Dockerfile\)](#)
- [3.4 \(versions/library-3.4/Dockerfile\)](#)
- [3.5, latest \(versions/library-3.5/Dockerfile\)](#)
- [edge \(versions/library-edge/Dockerfile\)](#)

Quick reference

Average Rating: ★★★★★ 13 Ratings

Select a product tier

★★★★★
Click to rate the selected product tier

[Add Review](#)

Install Docker (2).msi Geïnstalleerd

Install Docker (1).msi Geïnstalleerd

Alles weergeven

LINUX ALPINE

Nr	Step	Commando
1	Go to Docker store	https://store.docker.com/images/alpine
2	Copy docker command	copy
3	Go to Docker	docker pull alpine
4	Run the container	docker container run -it alpine /bin/sh

GIVE SOME COMMANDS ALPINE

	Command	
Linux version	cat /etc/*-release	
	Pwd	
Show directory	ls	
User	Whoami	
Close Alpine container	Ctrl-D	

Check if container is still running:

- Docker container ls
- Docker container ls -a

ALPINE RESULT

```
N:\>docker pull alpine
Using default tag: latest
latest: Pulling from library/alpine
88286f41530e: Pull complete
Digest: sha256:1072e499f3f655a032e88542330cf75b02e7bdf673278f701d7ba61629ee3ebe
Status: Downloaded newer image for alpine:latest

N:\>docker run -it alpine /bin/sh
/ # cat /etc/*-release
3.6.2
NAME="Alpine Linux"
ID=alpine
VERSION_ID=3.6.2
PRETTY_NAME="Alpine Linux v3.6"
HOME_URL="http://alpinelinux.org"
BUG_REPORT_URL="http://bugs.alpinelinux.org"
/ # pwd
/
/ # ls
bin    dev    etc    home   lib    media  mnt    proc   root   run    sbin   srv    sys    tmp    usr    var
/ # whoami
root
/ #
```

EXAMPLE – EASE OF DOCKER PORTAINER

- Command:
 - `docker container run -d -p 9000:9000 -v /var/run/docker.sock:/var/run/docker.sock portainer/portainer`
- Docker Store:
 - <https://store.docker.com/community/images/portainer/portainer>
- Website:
 - <https://portainer.io/>
- Documentation

PORTAINER: A DOCKER UI

DOCKER UI OR CLI?

The screenshot displays the Portainer web interface. On the left is a dark blue sidebar with the 'portainer.io' logo and a navigation menu. The main content area is titled 'Home Dashboard' and shows system information for a Docker Swarm node named 'moby'. It includes sections for Node info, Swarm info, and summary cards for Containers, Images, Volumes, and Networks. The user 'admin' is logged in.

portainer.io

ACTIVE ENDPOINT: local

ENDPOINT ACTIONS

- Dashboard
- App Templates
- Services
- Containers
- Images
- Networks
- Volumes
- Secrets
- Swarm

PORTAINER SETTINGS

- User management
- Endpoints
- Registries
- Settings

Portainer 1.13.4

Home Dashboard

admin
[my account](#) [log out](#)

Node info

Name	moby
Docker version	17.06.0-ce
CPU	2
Memory	2.1 GB

Swarm info

This node is part of a Swarm cluster

Node role	Manager
Nodes in the cluster	1

Summary Cards:

- Containers:** 2 (2 running, 0 stopped)
- Images:** 28 (2.6 GB)
- Volumes:** 1 (overlay2 driver)
- Networks:** 6

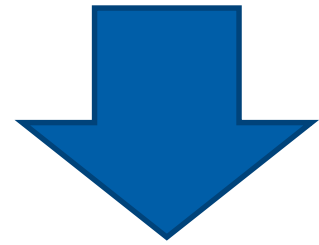
```
docker container run -d -p 9000:9000 -v /var/run/docker.sock:/var/run/docker.sock  
portainer/portainer
```

EXPLAIN CONFIGURATION ON RUNTIME

- `docker container run -d -p 9000:9000 -v /var/run/docker.sock:/var/run/docker.sock portainer/portainer`

	Explain
<code>docker container run</code>	depricated command: <code>docker run</code>
<code>-d</code>	detached – run on background in the cli
<code>-p <hostport> <container port></code>	publish
<code>-v</code>	use volume
Read more	https://docs.docker.com/engine/reference/commandline/container_run/

DOCKER COMMANDS



Build

Docker image ls

Docker image rm

Docker image build

Ship

Docker pull

Docker push

Run

Docker container run

Docker container ls

Docker container rm

Docker container exec

Docker container start

Docker container stop

Docker system info

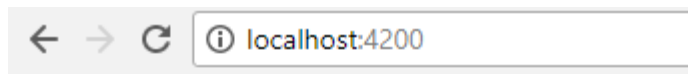
Docker version

Docker system prune

EXAMPLE: OUR STATIC WEB SITE

docker container run --name staticws -d -p 4200:80 centricms/staticws:latest

df16bf953be84b16fe3f8b67cd90d50a056b17b5a8a3fc4075a92634c3720b65



Hello Static Website

Continuously Delivered by:



Versie: 01.04.00 - september 2017

EXPLAIN DOCKER CONTAINER RUN

- `docker container run --name staticws -d -p 4200:80 centricms/staticws:latest`

	Explain
<code>docker container run</code>	Old command: <code>docker run</code>
<code>--name</code>	Container name Name must be unique!
<code>-d</code>	detached – run on background in the cli
<code>-p <hostport> <container port></code>	publish 4200:80
<code><name image></code>	<code><user>/<name>:<tag></code>
Read more	https://docs.docker.com/engine/reference/commandline/container_run/

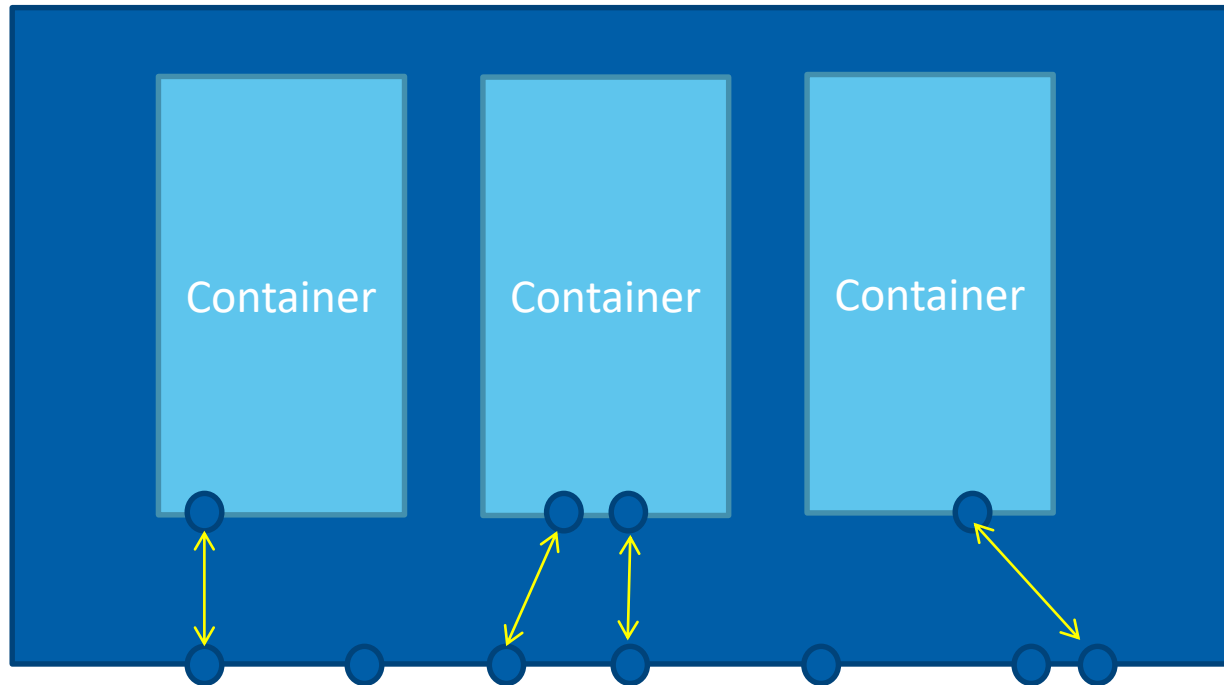
MULTIPLE CONTAINERS FROM 1 IMAGE

- Try:
 - `docker container run --name staticws -d -p 8000:80 centricms/staticws:latest`
 - `docker container run --name staticws -d -p 8000:80 centricms/staticws:latest`
 - → error same name

- `docker container run -d -p 8000:80 centricms/staticws:latest`
- `docker container run -d -p 8000:80 centricms/staticws:latest`
- → error same hostport

- `docker container run -d -p 8001:80 centricms/staticws:latest`
- `docker container run -d -p 8002:80 centricms/staticws:latest`
- → 3 static website

PUBLISH PORTS



EXPLAIN DOCKER LS

```
PS W:\ShowCase\microcontainer> docker container ls
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                               NAMES
a800b0b15a7f   centricms/staticws:latest          "nginx -g 'daemon ..." 36 seconds ago Up 35 seconds 0.0.0.0:8002->80/tcp   boring_shaw
f99f2346bdde   centricms/staticws:latest          "nginx -g 'daemon ..." 45 seconds ago Up 44 seconds 0.0.0.0:8001->80/tcp   priceless_kirch
bfa5f9587908   centricms/staticws:latest          "nginx -g 'daemon ..." About a minute ago Up About a minute 0.0.0.0:8000->80/tcp   staticws
```

	Explain
Container ID	Unique number
Image	3 Containers from the same image
Start command	detached – run on background in the cli
Created	
Status	
Ports	Host port → container port
Names	No –name, Docker provide unique name

EXPLAIN

- P, DOCKER WILL ASSIGN PORT

```
PS W:\ShowCase\microcontainer> docker container run -d -P centricms/staticws:latest  
f9eec56eb68b1111d931c07fef8ae953dcbf6dd34d195341ba0b4e263db5256f
```

```
PS W:\ShowCase\microcontainer> docker container ls  
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES  
f9eec56eb68b        centricms/staticws:latest    "nginx -g 'daemon ...'"   12 seconds ago     Up 10 seconds      0.0.0.0:32768->80/tcp    priceless_goldberg  
74c05a37bd26        centricms/staticws:latest    "nginx -g 'daemon ...'"   21 minutes ago     Up 21 minutes      80/tcp                 noport  
a800b0b15a7f        centricms/staticws:latest    "nginx -g 'daemon ...'"   30 minutes ago     Up 30 minutes      0.0.0.0:8002->80/tcp    boring_shaw  
f99f2346bdde        centricms/staticws:latest    "nginx -g 'daemon ...'"   30 minutes ago     Up 30 minutes      0.0.0.0:8001->80/tcp    priceless_kirch  
bfa5f9587908        centricms/staticws:latest    "nginx -g 'daemon ...'"   31 minutes ago     Up 31 minutes      0.0.0.0:8000->80/tcp    staticws
```

<input type="checkbox"/>	running	priceless_goldberg	centricms/staticws:latest	172.17.0.6	↗ 32768:80	👁 public
<input type="checkbox"/>	running	noport	centricms/staticws:latest	172.17.0.5	-	👁 public
<input type="checkbox"/>	running	boring_shaw	centricms/staticws:latest	172.17.0.4	↗ 8002:80	👁 public
<input type="checkbox"/>	running	priceless_kirch	centricms/staticws:latest	172.17.0.3	↗ 8001:80	👁 public
<input type="checkbox"/>	running	staticws	centricms/staticws:latest	172.17.0.2	↗ 8000:80	👁 public

EXPLAIN -D -DETACHED

```
PS W:\ShowCase\microcontainer> docker container run -p 8003:80 centricms/staticws:latest
172.17.0.1 - - [09/Sep/2017:11:12:37 +0000] "GET / HTTP/1.1" 200 217 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36" "-"
172.17.0.1 - - [09/Sep/2017:11:12:37 +0000] "GET /centric_logo_x2.png HTTP/1.1" 200 9853 "http://localhost:8003/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36" "-"
2017/09/09 11:12:37 [error] 7#7: *1 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 172.17.0.1, server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "localhost:8003", referer: "http://localhost:8003/"
172.17.0.1 - - [09/Sep/2017:11:12:37 +0000] "GET /favicon.ico HTTP/1.1" 404 571 "http://localhost:8003/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36" "-"
172.17.0.1 - - [09/Sep/2017:11:12:40 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36" "-"
172.17.0.1 - - [09/Sep/2017:11:12:41 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36" "-"
172.17.0.1 - - [09/Sep/2017:11:12:42 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36" "-"
172.17.0.1 - - [09/Sep/2017:11:12:43 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36" "-"
172.17.0.1 - - [09/Sep/2017:11:12:43 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36" "-"
172.17.0.1 - - [09/Sep/2017:11:12:44 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36" "-"
```

DOCKER CONTAINER COMMAND

- Docker container run
 - new container from image
 - Docker container run `-rm`
 - after stop remove
- Docker container stop
- Docker container start – container must exist
 - id or containername

DOCKER EXEC

- `docker container exec -it <containername> /bin/sh`
- Enter the container
 - Container must be running!
- Depends on base image which shell can be started

DOCKER LOGS

- docker container logs -f

```
PS W:\ShowCase\microcontainer> docker container logs -f staticws
172.17.0.1 - - [10/Sep/2017:08:07:04 +0000] "GET / HTTP/1.1" 200 217 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36" "-"
172.17.0.1 - - [10/Sep/2017:08:07:05 +0000] "GET /centric_logo_x2.png HTTP/1.1" 200 9853 "http://localhost:8000/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36" "-"
2017/09/10 08:07:05 [error] 6#6: *1 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 172.17.0.1, server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "localhost:8000", referer: "http://localhost:8000/"
172.17.0.1 - - [10/Sep/2017:08:07:05 +0000] "GET /favicon.ico HTTP/1.1" 404 571 "http://localhost:8000/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36" "-"
```

REMOVE DOCKER CONTAINER

- Running & Stopped containers
- Remove container
 - Docker container rm <container>
- Remove running container
 - Docker container rm -f <container>
- Remove all stopped containers
 - Docker container prune

DEMO: RUN ON AZURE

```
centric@DockerHost01:~$ sudo docker version
Client:
 Version:      17.06.0-ce
 API version:  1.30
 Go version:   go1.8.3
 Git commit:   02c1d87
 Built:        Fri Jun 23 21:23:31 2017
 OS/Arch:      linux/amd64

Server:
 Version:      17.06.0-ce
 API version:  1.30 (minimum version 1.12)
 Go version:   go1.8.3
 Git commit:   02c1d87
 Built:        Fri Jun 23 21:19:04 2017
 OS/Arch:      linux/amd64
 Experimental: false
```

```
centric@DockerHost01:~$ sudo docker container ls
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS             PORTS              NAMES
centric@DockerHost01:~$
```

```
centric@DockerHost01:~$ sudo docker container run -d -p:8901:80 centricms/staticws
bb20b7f4bd8a273cb951b46ff88130012d95d5caf7d7c51b9500f532d767dc35
centric@DockerHost01:~$ sudo docker container ls
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS             PORTS              NAMES
bb20b7f4bd8a        centricms/staticws "nginx -g 'daemon ..." About a minute
ago Up About a minute 0.0.0.0:8901->80/tcp  condescending_ritchie
centric@DockerHost01:~$
```

← → ↻ ⓘ dockerhost01.westeurope.cloudapp.azure.com:8901

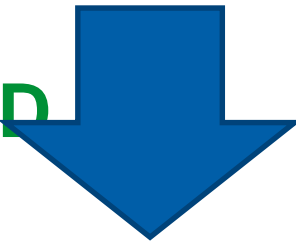
Hello Static Website

Continuously Delivered by:



Versie: 01.04.00 - september 2017

DOCKER COMMAND



Build

Docker image ls

Docker image rm

Docker image build

Ship

Docker pull

Docker push

Run

Docker container run

Docker container ls

Docker container rm

Docker container exec

Docker container start

Docker container stop

Docker system info

Docker version

Docker system prune

DOCKER REGISTRY

- Ready to use:
 - DOCKER HUB / DOCKER STORE
- Private registry
 - For companies & enterprises

DOCKER PUSH

- You need a user-id on Docker Hub
 - Free of charge for public repo (images)
- Docker login
- Every repo (image) name must start with your user
- Version – tag
- Example
 - `docker push centricms/staticws:latest`

DOCKER IMAGE PULL

- For Docker Hub / Store
 - No user-id needed for public repo
- Official & open source
 - Official by Docker or supplier
- Tips:
 - Use safe images: official, number of pulls, rating
 - Efficient download (only changed layers)
 - Image is first search on local DockerHost

DOCKER COMMAND

Build

Docker image ls

Docker image rm

Docker image build

Ship

Docker pull

Docker push

Run

Docker container run

Docker container ls

Docker container rm

Docker container exec

Docker container start

Docker container stop

Docker system info

Docker version

Docker system prune

DOCKERFILE

- Recipe or blue print
- Each line will be layer
- Each layer will be cached

WHAT IS IMAGE? - REVISITED

- Image = stack of layers
- Made by:
 - Dockerfile = recipe or blue print
 - Docker commit – seldomly used – not in this Docker 101
- Container made from image must run on any environment

DOCKER BUILD - DOCKERIZE

- DOCKERIZE = PACKAGE application in a Docker image
- Image / container
 - contains every dependency
 - Small
 - Secure
 -

Best practice:

- Use the same image for very environment

TOPIC: DOCKER BUILD DOCKERFILE

- Dockerfile: Recipe or blue print
- Base image
 - FROM SCRATCH
 - KNOWN TRUSTED IMAGE (OFFICIAL)


TOPIC: VERSION


- TAG
- Latest
 - Just a tag!
 - use with care
- No tag: build - snapshot
- Base OS
 - Alpine – Docker default - small
 - Debian
 - Windows

DOCKERFILE FOR STATICWS

- Goal & requirement:
 - Static website that can run on laptop and Azure
 - Easy webserver – use nginx
 - Fits in our way of working (CD flow)

SEARCH FOR NGINX

 docker store

Explore Publish Feedback  sim007

**nginx**
By Docker

Official build of Nginx.

10M+ Pulls

Categories: [Application Infrastructure](#)Official Image
\$0.00[Terms of Service](#)

Copy and paste to pull this image

`docker pull nginx` [View Available Tags](#)

DESCRIPTION REVIEWS RESOURCES

Supported tags and respective Dockerfile links

- [1.13.3, mainline, 1, 1.13, latest \(mainline/stretch/Dockerfile\)](#)
- [1.13.3-perl, mainline-perl, 1-perl, 1.13-perl, perl \(mainline/stretch-perl/Dockerfile\)](#)
- [1.13.3-alpine, mainline-alpine, 1-alpine, 1.13-alpine, alpine \(mainline/alpine/Dockerfile\)](#)
- [1.13.3-alpine-perl, mainline-alpine-perl, 1-alpine-perl, 1.13-alpine-perl, alpine-perl \(mainline/alpine-perl/Dockerfile\)](#)
- [1.12.1, stable, 1.12 \(stable/stretch/Dockerfile\)](#)
- [1.12.1-perl, stable-perl, 1.12-perl \(stable/stretch-perl/Dockerfile\)](#)
- [1.12.1-alpine, stable-alpine, 1.12-alpine \(stable/alpine/Dockerfile\)](#)
- [1.12.1-alpine-perl, stable-alpine-perl, 1.12-alpine-perl \(stable/alpine-perl/Dockerfile\)](#)

Average Rating: ★★★★★ 3 Ratings

Add Product Review

Select a product tier ▼

Start Your Review

DOCKER BUILD

- Make Dockerfile
 - Base on base image
 - From docker Store
 - From scratch
 - Copy static files to the correct directory
- Build the image with:
 - Docker image build -t <name> .
- Tips
 - Default filename is Dockerfile
 - CLI must be in the same directory as Dockerfile
 - Do not use networkshare on W10

EXPLAIN DOCKER BUILD

- Docker image build -t <name> .

	Explain
docker image build	deprecated command: docker build
-t <name>	Tag a unique name
.	directory
Read more	https://docs.docker.com/engine/reference/commandline/image_build/

EXAMPLE OUTPUT (FIRST TIME AND NEXT TIME)

```
PS C:\SimProjects\ShowCase\microcontainer\StaticWS> docker image build -t centricms/staticws:latest .
Sending build context to Docker daemon  14.34kB
Step 1/3 : FROM nginx:stable-alpine
stable-alpine: Pulling from library/nginx
019300c8a437: Pull complete
b9b9e74a248d: Pull complete
5efb344a3d05: Pull complete
1b297156e526: Pull complete
Digest: sha256:0fa9c1773647d8a60fb1df954b35c93e745de7b80e3b7fe261c61a059ee75b62
Status: Downloaded newer image for nginx:stable-alpine
--> 7c588c2e29dc
Step 2/3 : COPY index.html *.png /usr/share/nginx/html/
--> 03c9908810e8
Removing intermediate container 68018b79d2c6
Step 3/3 : LABEL MAINTAINER "Johannes Sim / Renzo Veldkamp" REFRESHED_AT "2017-09-17"
--> Running in 9370013c0bfc
--> b83189dccfa7
Removing intermediate container 9370013c0bfc
Successfully built b83189dccfa7
Successfully tagged centricms/staticws:latest
SECURITY WARNING: You are building a Docker image from Windows against a non-Windows Docker host. All files and directories added to build context will have '-rwxr-xr-x' permissions. It is recommended to double check and reset permissions for sensitive files and directories.
```

```
PS C:\SimProjects\ShowCase\microcontainer\StaticWS> docker image build -t centricms/staticws:latest .
Sending build context to Docker daemon  14.34kB
Step 1/3 : FROM nginx:stable-alpine
--> 7c588c2e29dc
Step 2/3 : COPY index.html *.png /usr/share/nginx/html/
--> Using cache
--> 03c9908810e8
Step 3/3 : LABEL MAINTAINER "Johannes Sim / Renzo Veldkamp" REFRESHED_AT "2017-09-17"
--> Using cache
--> b83189dccfa7
Successfully built b83189dccfa7
Successfully tagged centricms/staticws:latest
SECURITY WARNING: You are building a Docker image from Windows against a non-Windows Docker host. All files and directories added to build context will have '-rwxr-xr-x' permissions. It is recommended to double check and reset permissions for sensitive files and directories.
```

COMMAND DOCKER IMAGE

Docker image --help	List options for docker image																
Docker image ls	List docker images	Listed deprecated: Docker images Image has id and repo name															
<table><tr><th>REPOSITORY</th><th>TAG</th><th>IMAGE ID</th><th>CREATED</th><th>SIZE</th></tr><tr><td>centricms/staticws</td><td>latest</td><td>b83189dccfa7</td><td>7 minutes ago</td><td>15.5MB</td></tr><tr><td>nginx</td><td>stable-alpine</td><td>7c588c2e29dc</td><td>2 days ago</td><td>15.5MB</td></tr></table>			REPOSITORY	TAG	IMAGE ID	CREATED	SIZE	centricms/staticws	latest	b83189dccfa7	7 minutes ago	15.5MB	nginx	stable-alpine	7c588c2e29dc	2 days ago	15.5MB
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE													
centricms/staticws	latest	b83189dccfa7	7 minutes ago	15.5MB													
nginx	stable-alpine	7c588c2e29dc	2 days ago	15.5MB													
Docker history <image>	Returns Dockerfile commands																
Docker inspect <image>	Lots of information																

DOCKER COMMAND LINE INTERFACE

Build

Docker image ls

Docker image rm

Docker image build

Ship

Docker pull

Docker push

Run

Docker container run

Docker container ls

Docker container rm

Docker container exec

Docker container start

Docker container stop



Docker system info

Docker system version

Docker system prune

COMMAND DOCKER IMAGE

Docker system prune	Delete all stopped containers and used images	
Docker system prune -a	Delete all containers and images	Clean docker environment
Docker system df	Show docker disk usage	
Docker system events	Get real time events from the server	
Docker system info	Display system-wide information	

SUMMARY: OUR WORKFLOW

- What is on the laptop?
 - Static website
- Build
 - Build the image
- Ship –Flow
 - Push to Docker Hub / Docker Store
 - Pull from Docker Hub / Docker Store
- Run
 - Run on Laptop
 - Run on Azure host

OTHER LABS

LAB

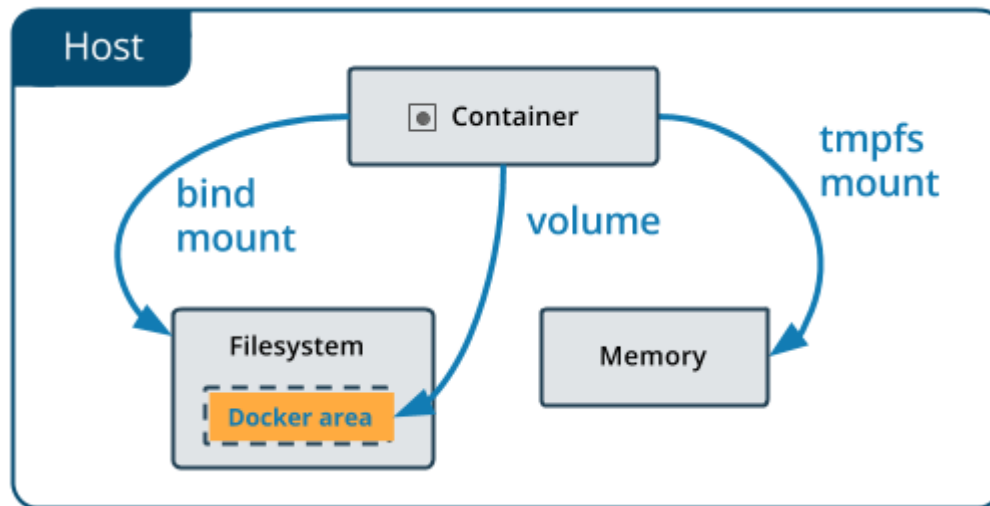
IMMUTABLE CONTAINER

- Start an alpine container
- Add a file Centric in home directory
 - `echo "deze file heet centric" >> centric`
- Stop the container with `exit`
- Case 1: start new container
 - `docker container -it alpine sh`
 - Result: no file Centric
- Case 2: start old container
 - `docker container ls -a` (find the container-id)
 - `docker container start <id>`
 - `docker exec -it <id>`
 - Result: Centric file

LAB

DOCKER VOLUME

- DEV flow – dev mode
- Changes in code change without building



LAB

EXPORT & IMPORT

- Ship without docker registry
- Docker export
- Docker import