





le wagon

July 3 - August 25

Summer **coding** bootcamp

Build something meaningful this summer





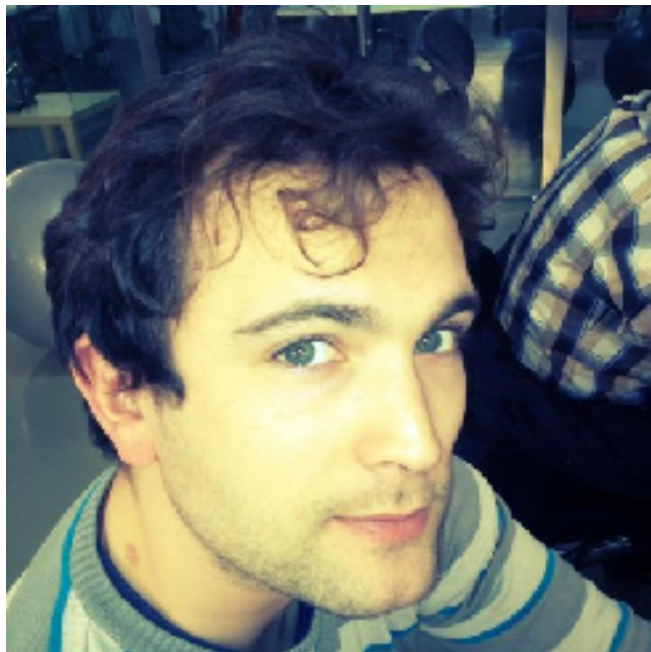
Le Wagon brings **technical skills**
to **creative people**

Introduction to Docker



le wagon

Vincent Viallet



- **DevOps Director** and Partner at Wiredcraft
- Handle systems at scale (**Starbucks, Apple, WorldBank**, etc.)
- Automation / Architecture / Performance



Agenda: Introduction to Docker

- Key concepts and usage
- Docker setup
- Key Commands
- Docker compose
- Dockerfile

Agenda: Introduction to **Docker**

→ Key concepts and usage

- Docker setup
- Key Commands
- Docker compose
- Dockerfile

Overview

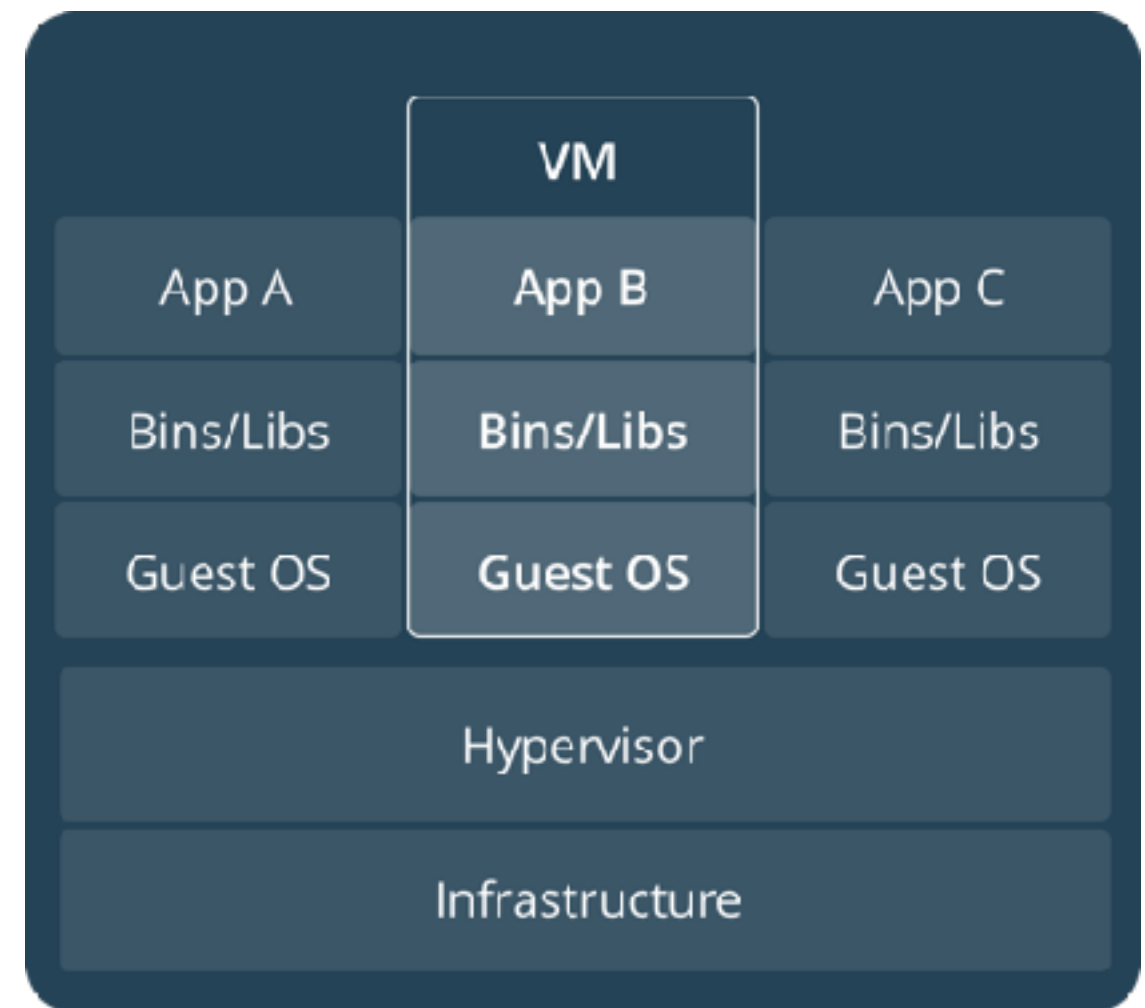
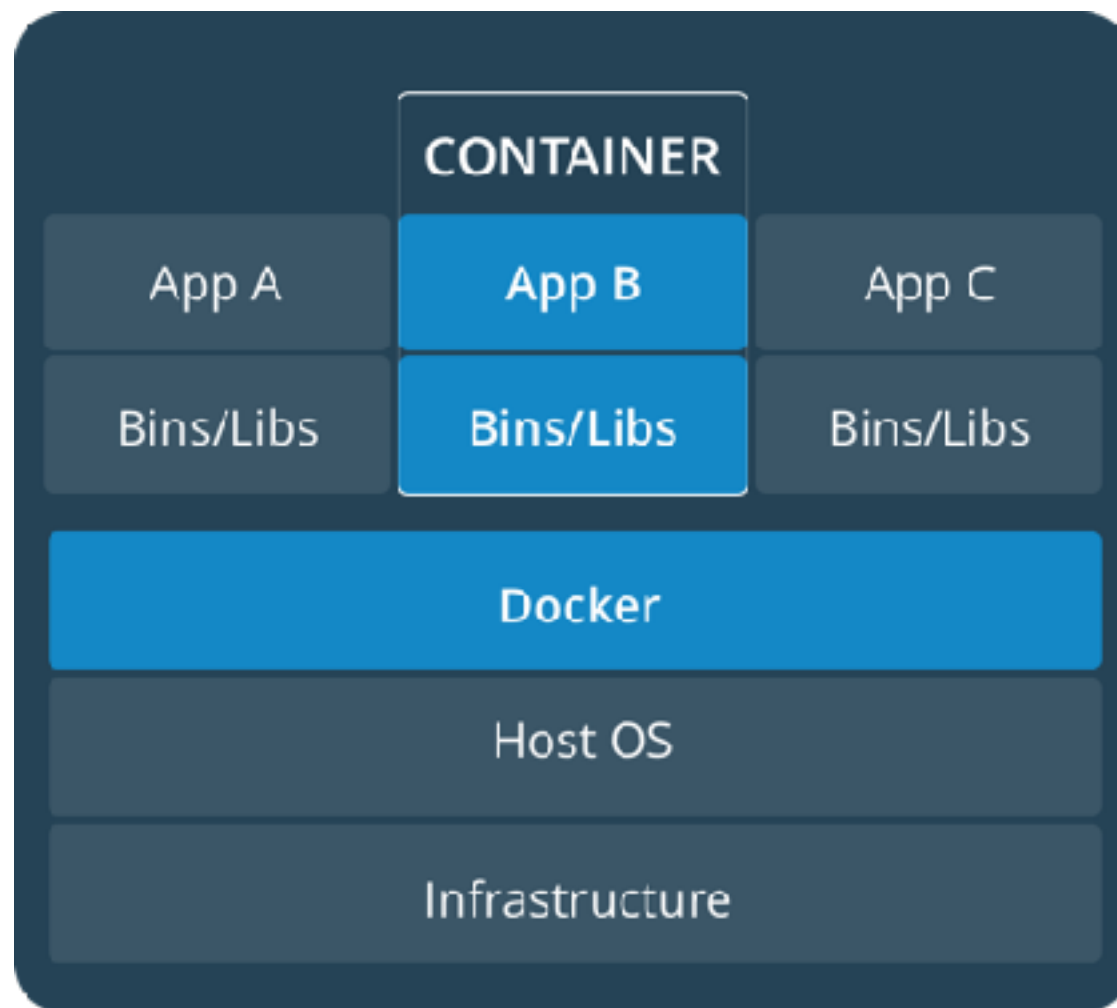
Docker is the world's leading software container platform. **Developers** use Docker to **eliminate “works on my machine” problems** when collaborating on code with co-workers. **Operators** use Docker to run and manage apps side-by-side in **isolated containers to get better compute density**. **Enterprises** use Docker to build agile software delivery pipelines to **ship new features faster**, more securely and with confidence for both Linux and Windows Server apps.

- docker.com

Containers

- Lightweight
- Standard
- Secure

Containers vs. VM



Do

- 1 process / container
- plan ahead (ephemeral vs. persistent, reusable images)
 - use dockerfile caching
 - log

Don't

- consider as light VM (no init, no ...)
 - careful when considering using containers for DB...

More (aging): <http://devo.ps/blog/docker-dos-and-donts/>

Usage

Development

- mock services
 - fast db
- spawn local architecture

Staging/Prod

- build release as images
 - fast release
- replicable setup (QA!)

Agenda: Introduction to Docker

- Key concepts and usage
- ➔ **Docker setup**
- Key Commands
- Docker compose
- Dockerfile

Setup

Workstation

Windows & MacOS

Servers

Linux (all flavours) &
Windows

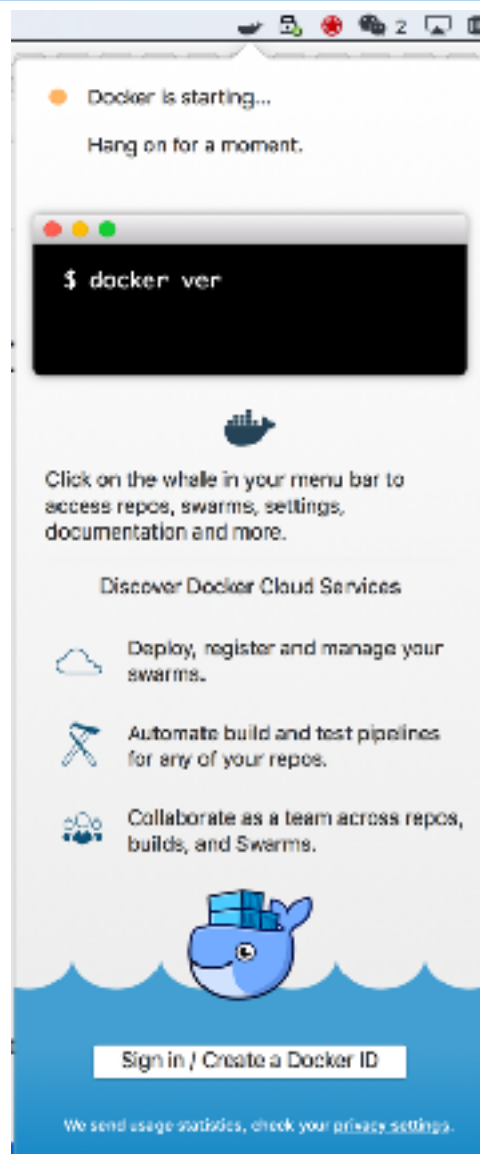
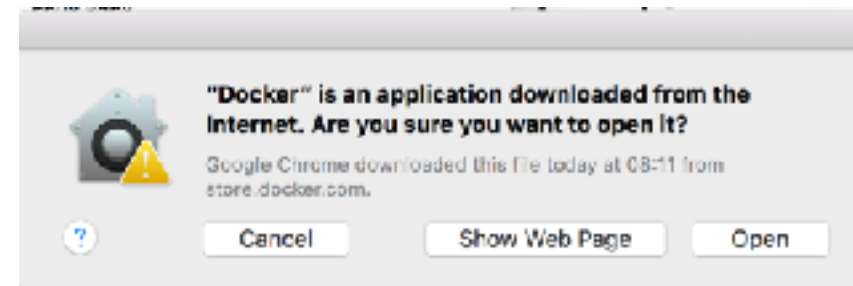
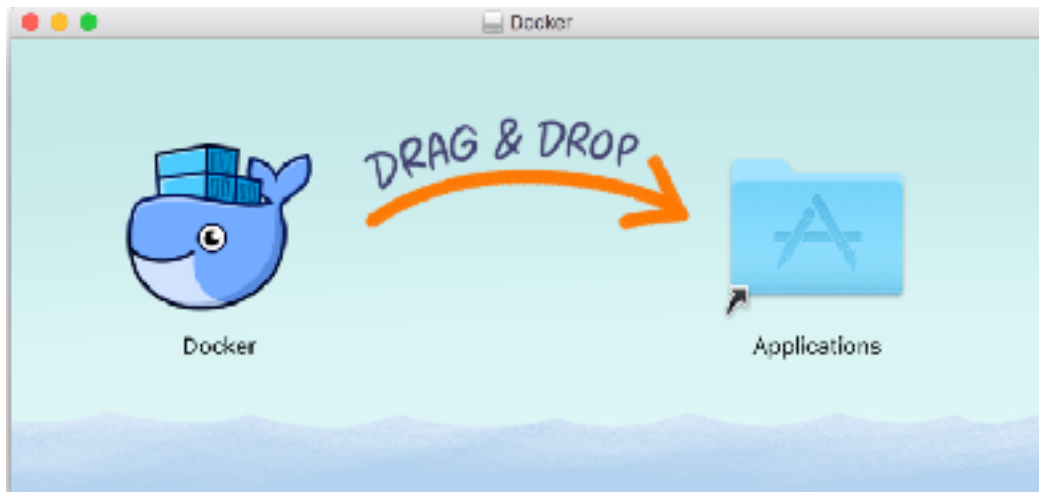
Setup MacOS

Go on; <https://www.docker.com/docker-mac>

Follow the download link and fetch the DMG from the store

Local mirror (LeWagon intro): <http://139.198.190.11/Docker.dmg>

Setup MacOS



```
[vincent:~ vincent]$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
b04784fba78d: Pull complete
Digest: sha256:f3b3b28a45160805bb16542c9531888519438e9e6d5ffc09d72261b0d26f1
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 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/getting-started/>

Setup Windows

Go on: <https://www.docker.com/docker-windows>

Follow the download link and fetch the MSI from the store

Local mirror (LeWagon intro):
139.198.190.11/InstallDocker.msi

Setup Windows

The image is a collage of five screenshots illustrating the Docker setup process on Windows:

- Top Left:** Docker Desktop welcome screen. It states "Docker is now up and running!" and instructs the user to "Open your favorite terminal and start typing Docker commands." It shows a terminal window with the command `> docker run hello-world`. Below, it says "Click on the whale in your task bar to access your repos, repositories, documentation, settings and more." It also promotes "Discover Docker Cloud Services" with links to Docker Cloud, Docker Hub, and Docker Swarm.
- Top Middle:** Windows PowerShell terminal window. It shows the output of the `docker run hello-world` command. The output indicates that the Docker client contacted the Docker daemon, the daemon pulled the "hello-world" image from the Docker Hub, created a new container from that image, and the container executed the command that produced the output.
- Top Right:** Windows taskbar showing the Docker icon (a blue whale) circled in red. A red arrow points from the icon to the Docker Desktop menu.
- Bottom Left:** Docker Desktop settings window. It shows the "About Docker" section with options like "Automatically start Docker when you log in" and "Automatically check for updates". It also has an "Experimental" section with "VPN Compatibility mode" and links to "Change VM settings...", "Manage shared drives...", "Import Toolbox default machine context...", and "Reset to factory defaults...".
- Bottom Middle:** Docker Desktop network configuration window. It shows the "Network" settings, including "Internal Virtual Switch" (Subnet Address: 10.0.75.0, Subnet Mask: 255.255.255.0) and "DNS Server" (Automatic, 8.8.8.8). There is an "Apply" button at the bottom right.
- Bottom Right:** Docker Desktop menu. It lists various options: "About Docker", "Discover Docker Enterprise Edition", "Settings...", "Check for Updates...", "Diagnose and Feedback...", "Switch to Windows containers...", "Docker Store", "Documentation", "Kitematic", "orangesnap", "Swarms", "Repositories", and "Quit Docker".

Setup Linux

No cute UI like Windows/Mac

Go on: <https://www.docker.com/docker-ubuntu>
or <https://www.docker.com/docker-centos-distribution> **or ...**

Follow the download link from the store and follow the indication (Yum repo, APT repo, etc.)

Additional resources

- hub.docker.org: free docker container registry (for OSS)

Search feature to use / reuse container images.

Auto build of containers



Explore Help [Sign up](#) [Sign in](#)

Repositories (22622)

All

 ubuntu official	6.2K STARS	10M+ PULLS	> DETAILS
 ubuntu-upstart official	74 STARS	100K+ PULLS	> DETAILS

Agenda: Introduction to Docker

- Key concepts and usage
- Docker setup
- **Key Commands**
- Docker compose
- Dockerfile

RTFM

Use `--help` on any docker command to get help!

docker --help
docker pull --help
docker run --help
etc.

Lots of really good doc at:

<https://docs.docker.com/>
<https://docs.docker.com/reference/>

Docker concepts

- **image:** used as “template” to spawn containers
- **container:** running code with its own filesystem
- **volume:** external storage that can be mounted on a container at run time
- **port:** listening ports can be exposed on the underlying host
- **link:** containers can be linked to one another allowing direct access to their network ports using the name

Pull images

docker pull <image>

Fetch container image

```
vincent:~ vincent$ 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
```

```
vincent:~ vincent$ docker pull alpine:3.5
3.5: Pulling from library/alpine
019300c8a437: Pull complete
Digest: sha256:af2a5bd2f8de8fc1ecabf1c76611cdc6a5f1ada1a2bdd7d3816e121b70300308
Status: Downloaded newer image for alpine:3.5
vincent:~ vincent$
```

<image> can be either of:

- **image (only)**; assume latest image
- **image:tag**; e.g. ubuntu:16.04
- **repo/image:tag**; fetching from another repo than hub.docker.io

Note: Check for **alpine** versions! much smaller

List images

docker images [image]

List all local images

```
[vincent:~ vincent$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
alpine	latest	7328f6f8b418	3 days ago	3.97MB
alpine	3.5	074d602a59d7	3 days ago	3.99MB
ubuntu	latest	d355ed3537e9	10 days ago	119MB
hello-world	latest	1815c82652c0	2 weeks ago	1.84kB

```
[vincent:~ vincent$ docker images redis
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
redis	alpine	159ac12c79e1	2 days ago	19.8MB
redis	latest	4e482b286430	7 days ago	99MB

[image] is optional but can be used to filter to only show the various versions of a same container

Run container

docker run [options] image [cmd]

run a container based on image

```
[vincent:~ vincent$ docker run -d redis
```

```
cd0c3759c675e831b17b6f025250c50aef5c70f04c4e210a795b0bae255ae852
```

```
[vincent:~ vincent$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
cd0c3759c675	redis	"docker-entrypoint..."	4 seconds ago	Up 3 seconds

[options]:

- d**; detach (run in background)
- rm**; remove the container when exited
- ti**; get a tty and interactive mode
- name**; give a name to the container

[cmd]:

cmd; the command you want to run instead of the default one (e.g. bash)

List containers

docker ps [-a]

List all running containers (list all with -a)

```
[vincent:~ vincent$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
8f265f4ae9de	redis	"docker-entrypoint..."	5 minutes ago	Up 5 minutes

```
[vincent:~ vincent$
```

```
[vincent:~ vincent$
```

```
[vincent:~ vincent$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
8f265f4ae9de	redis	"docker-entrypoint..."	5 minutes ago	Up 5 minutes
cd0c3759c675	redis	"docker-entrypoint..."	21 minutes ago	Exited (0) 6 m
9eb609dfe9af	redis	"docker-entrypoint..."	22 minutes ago	Exited (0) 22
1781c1172993	_hello-world	"/hello"	2 hours ago	Exited (0) 2 h

Stop / start / delete container

docker stop container_id/name

stop a running container

docker start container_id/name

start a previously stopped container (it need to have been created via run)

docker rm container_id/name

effectively removes the container and it files

```
-----  
[vincent:~ vincent$ docker run -d --name redis redis  
8f265f4ae9deab2074507b8ef6ea2e0a306d8b58cf61b41f171679f487f29d79  
[vincent:~ vincent$ docker stop redis  
redis  
[vincent:~ vincent$ docker start redis  
redis
```

Run a command in a running container

(also used to gain shell access)

`docker exec -ti container_id/name cmd`

```
[vincent:~ vincent$ docker exec -ti redis bash
[root@8f265f4ae9de:/data# ps aux
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
redis         1  0.1  0.1  33316   3896 ?        Ssl   03:12   0:00 redis-server *:6379
root          14  0.5  0.1  20252   3268 pts/0    Ss    03:14   0:00 bash
root          19  0.0  0.0  17504   2036 pts/0    R+    03:14   0:00 ps aux
[root@8f265f4ae9de:/data#
```

you can exit via ``exit`` or ``ctrl+d``

Check the details of a container / image

docker inspect [options] image/container

e.g.

docker inspect alpine

docker inspect --format='{{range .NetworkSettings.Networks}}
{{.IPAddress}}{{end}}' redis

```
[vincent:~ vincent$ docker inspect alpine
```

```
[
  {
    "Id": "sha256:7328f6f8b41890597575cbaadc884e7386ae0acc53b747401ebce5cf0d624560",
    "RepoTags": [
      "alpine:latest"
    ],
    "RepoDigests": [
      "alpine@sha256:1072e499f3f655a032e88542330cf75b02e7bdf673278f701d7ba61629ee3ebe"
    ],
    "Parent": "",

```

```
vincent:~ vincent$ docker inspect --format='{{range .NetworkSettings.Networks}}  
{{.IPAddress}}{{end}}' redis  
172.17.0.2
```

[https://docs.docker.com/engine/reference/
commandline/inspect/#examples](https://docs.docker.com/engine/reference/commandline/inspect/#examples)

Build container

docker build [-f Dockerfile]

Create a new image based on the Dockerfile (more later)

```
[vincent:examples vincent]$ docker build .  
Sending build context to Docker daemon 2.048kB  
Step 1/2 : FROM redis  
----> 4e482b286430  
Step 2/2 : RUN echo "toto" > /toto  
----> Running in eac5ca3e175f  
----> 63239e9219eb  
Removing intermediate container eac5ca3e175f  
Successfully built 63239e9219eb
```

```
[vincent:examples vincent]$ docker build .  
Sending build context to Docker daemon 2.048kB  
Step 1/2 : FROM redis  
----> 4e482b286430  
Step 2/2 : RUN echo "toto" > /toto  
----> Using cache  
----> 63239e9219eb  
Successfully built 63239e9219eb
```

Examples

See <https://github.com/Wiredcraft/docker-workshop>

git clone **<https://github.com/Wiredcraft/docker-workshop.git>**

cd docker-workshop/python

Check the README.md

Run container, publish port, use permanent storage connect your local code.

Agenda: Introduction to Docker

- Key concepts and usage
- Docker setup
- Key Commands
- **Docker compose**
- Dockerfile

What is docker-compose

The docker-compose CLI command manage full multi-container environment

Ever improving tool - Many different versions of **docker-compose.yml** files

<https://docs.docker.com/compose/compose-file/>

docker-compose

docker-compose up

starts all the containers - same options as for docker (-d, etc.)

docker-compose down

stop all the containers

docker-compose ps

list all the containers

Example

**in the previously
cloned repo - go in
the docker-compose
folder**

Agenda: Introduction to Docker

- Key concepts and usage
- Docker setup
- Key Commands
- Docker compose
- ➔ **Dockerfile**