

Containers

(les conteneurs)



Pr. Nabil Benamar



Server

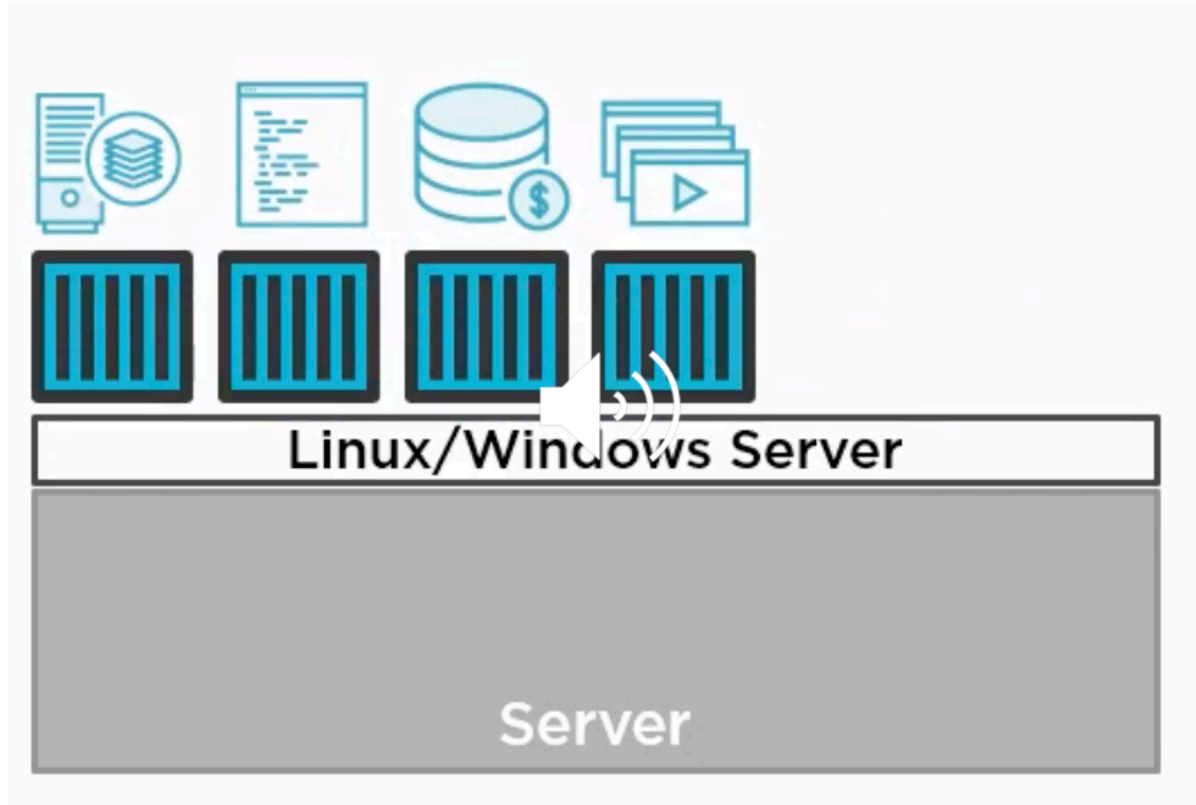


Linux/Windows Server

Server

Container

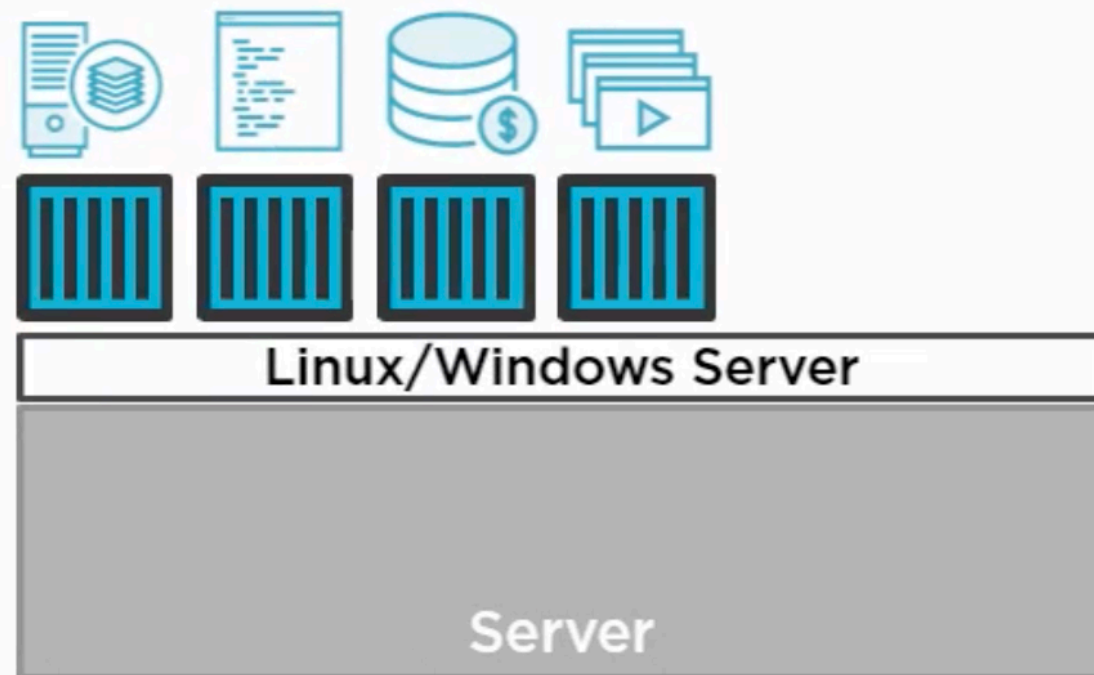
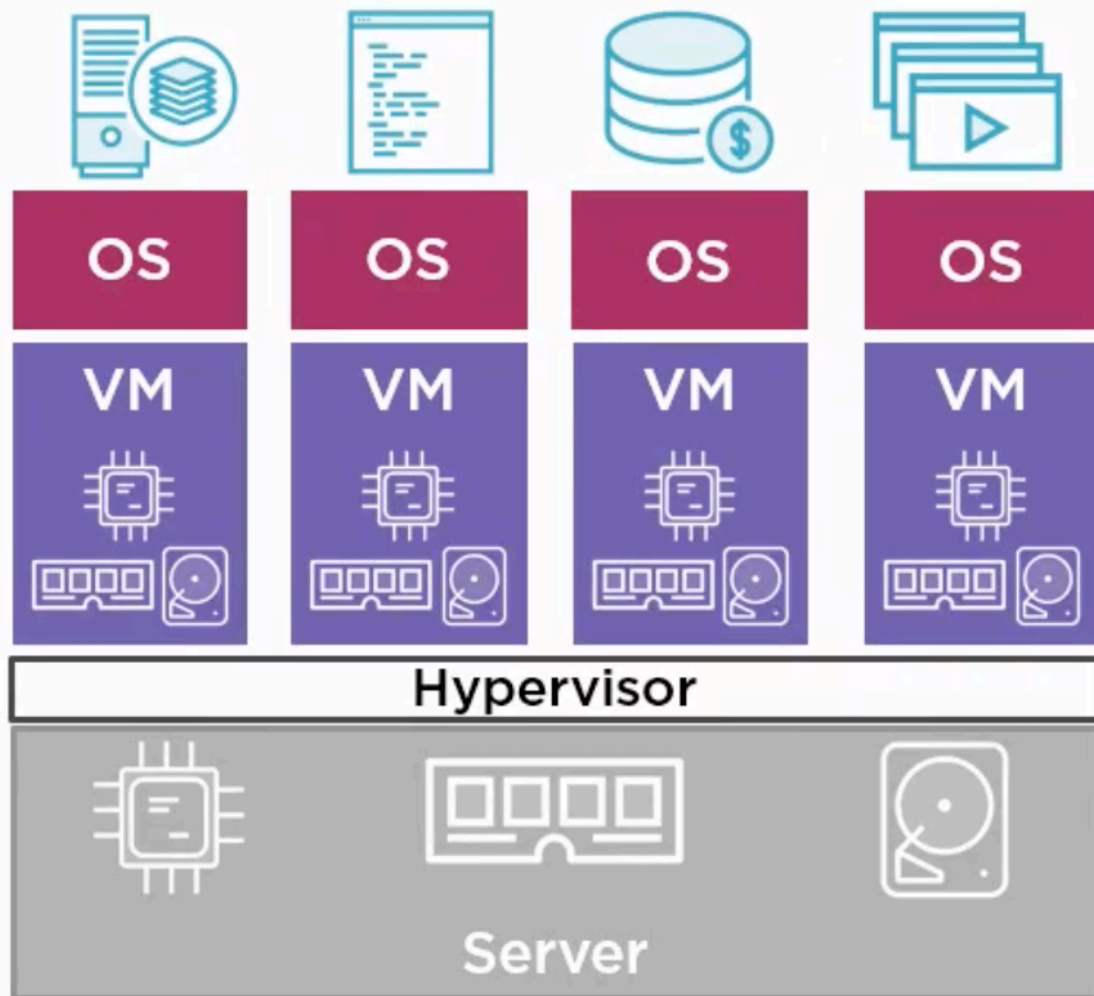
- Un conteneur est une unité logicielle standard qui regroupe le code et toutes ses dépendances.
- Afin que l'application s'exécute rapidement et de manière fiable d'un environnement informatique à l'autre.
- Une image de conteneur Docker est un package logiciel exécutable autonome et léger qui comprend tout le nécessaire pour exécuter une application:
- code, runtime, outils système, bibliothèques système et paramètres.

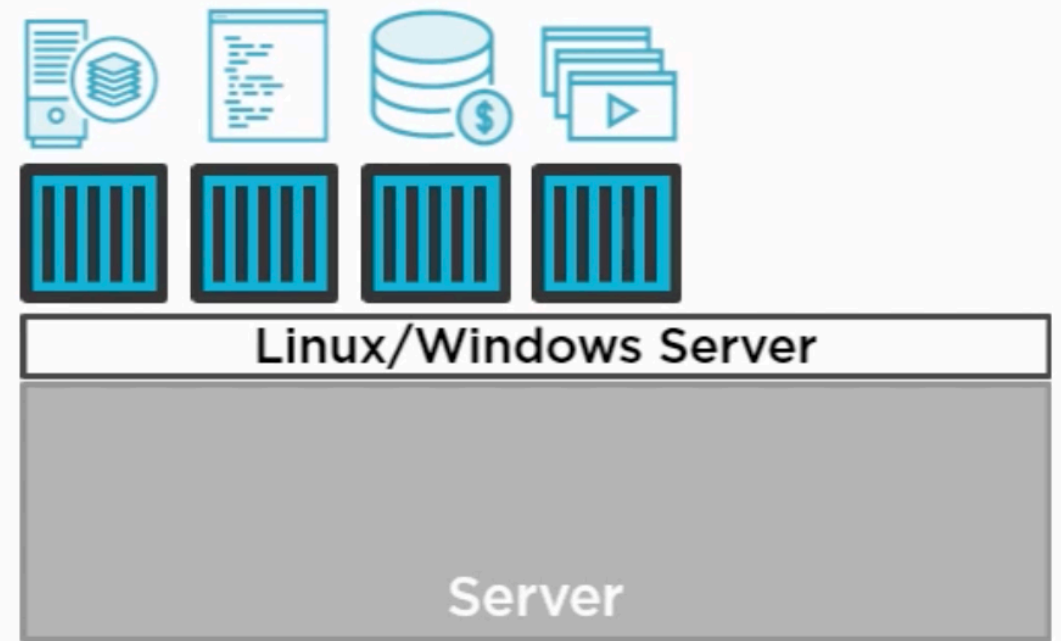
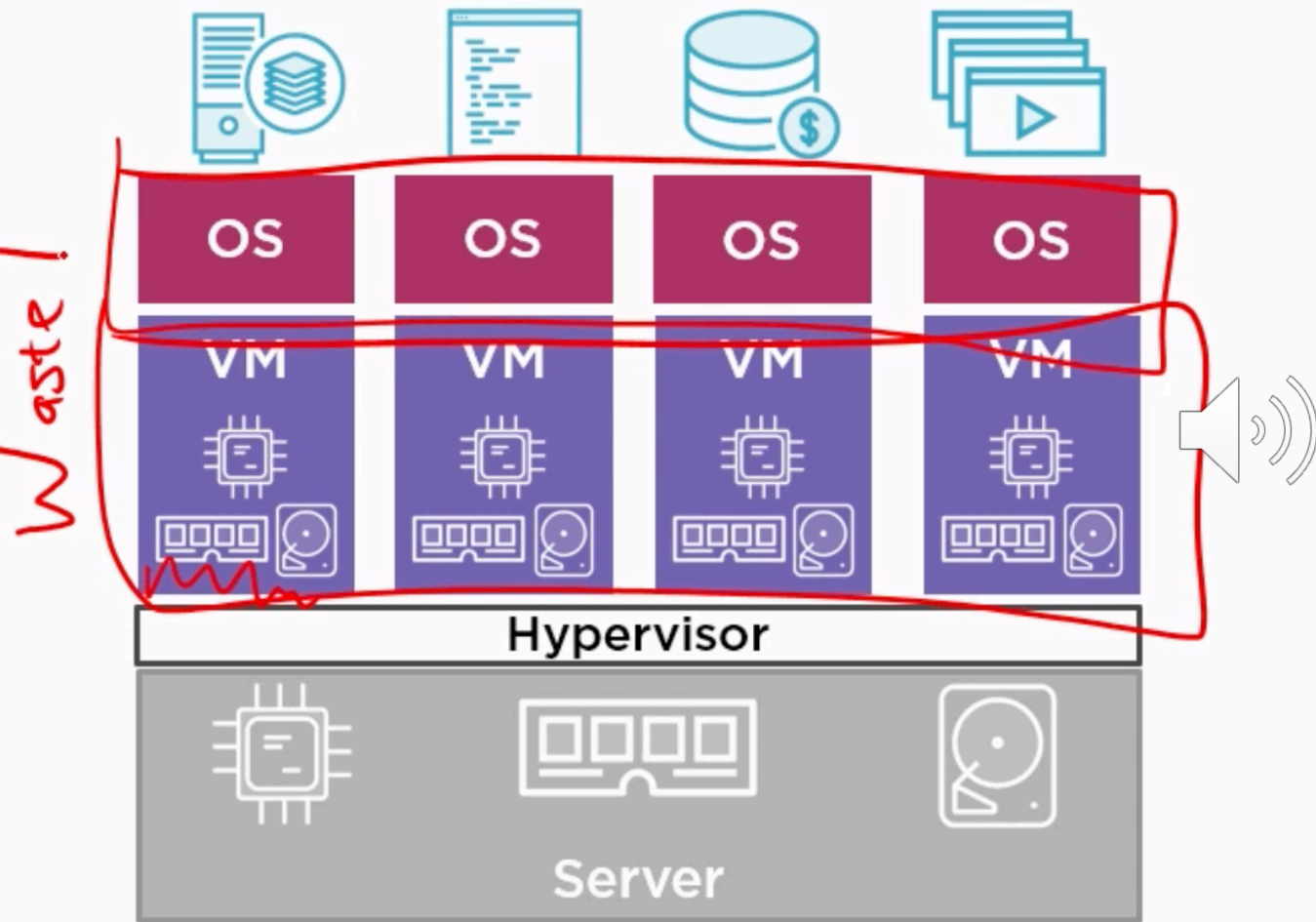


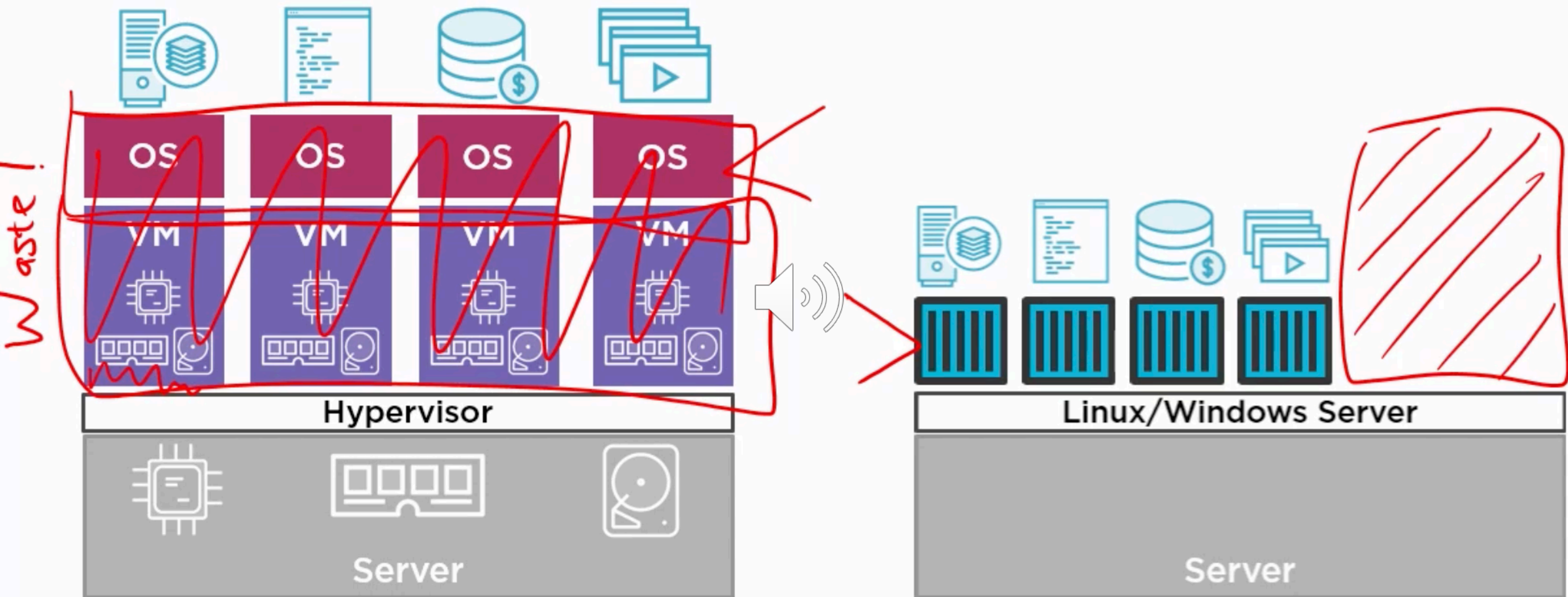
Docker's initial marketing
related them to VMs

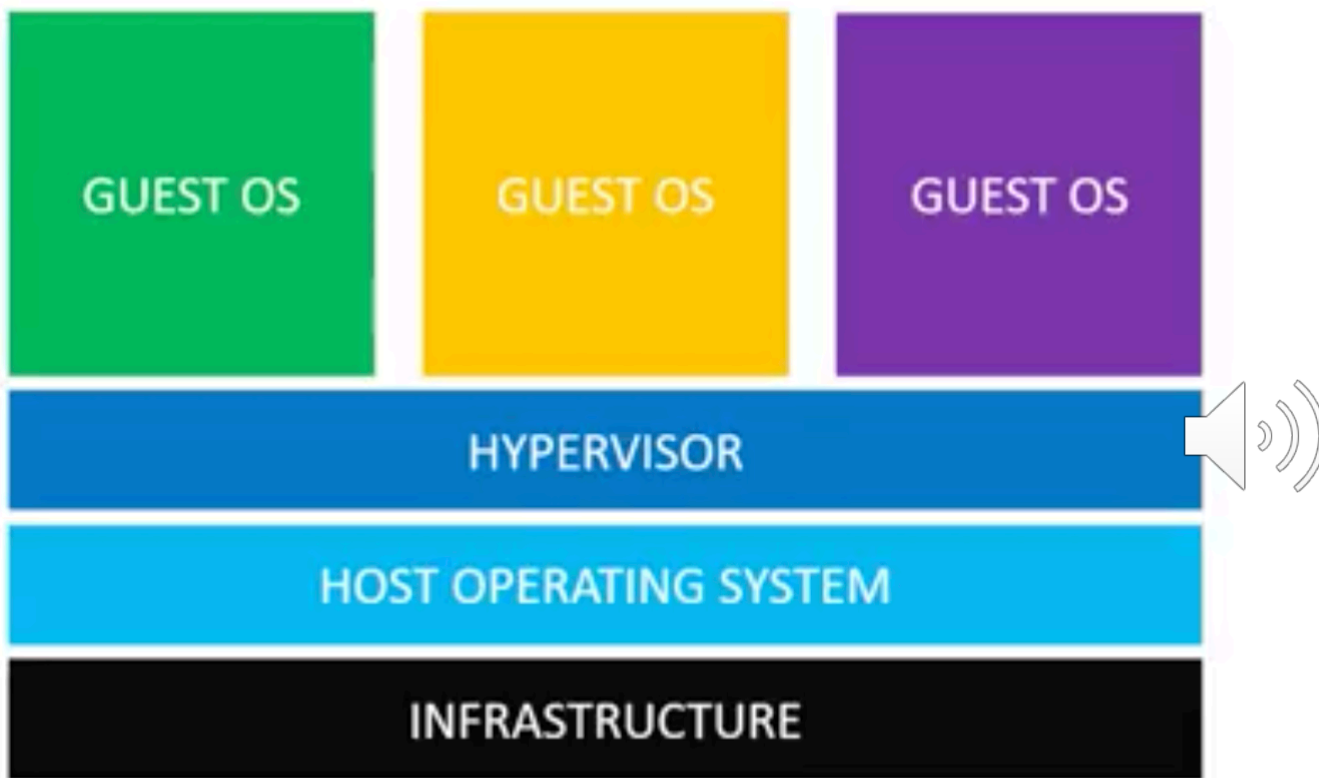
Think of containers as isolated processes

Docker containers are not VMs





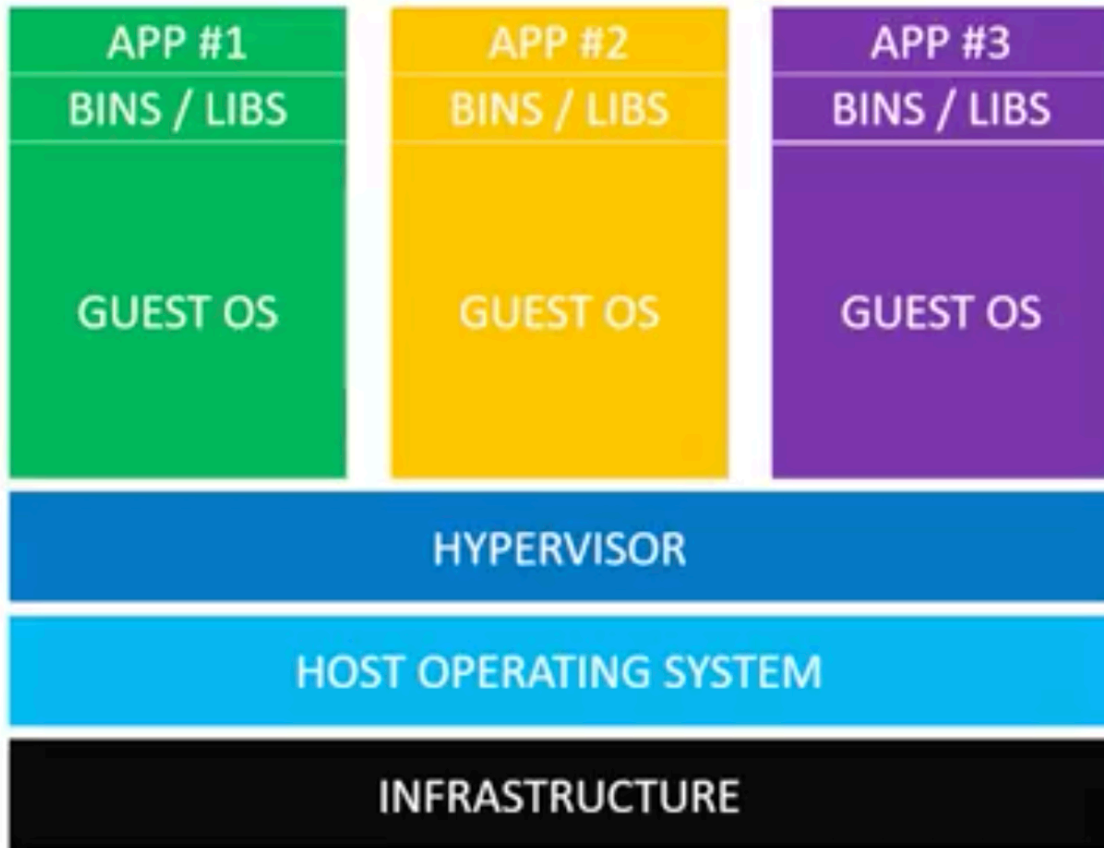




$700\text{MB} * 3 = 2.1\text{GB}$
+ CPU and Memory resources

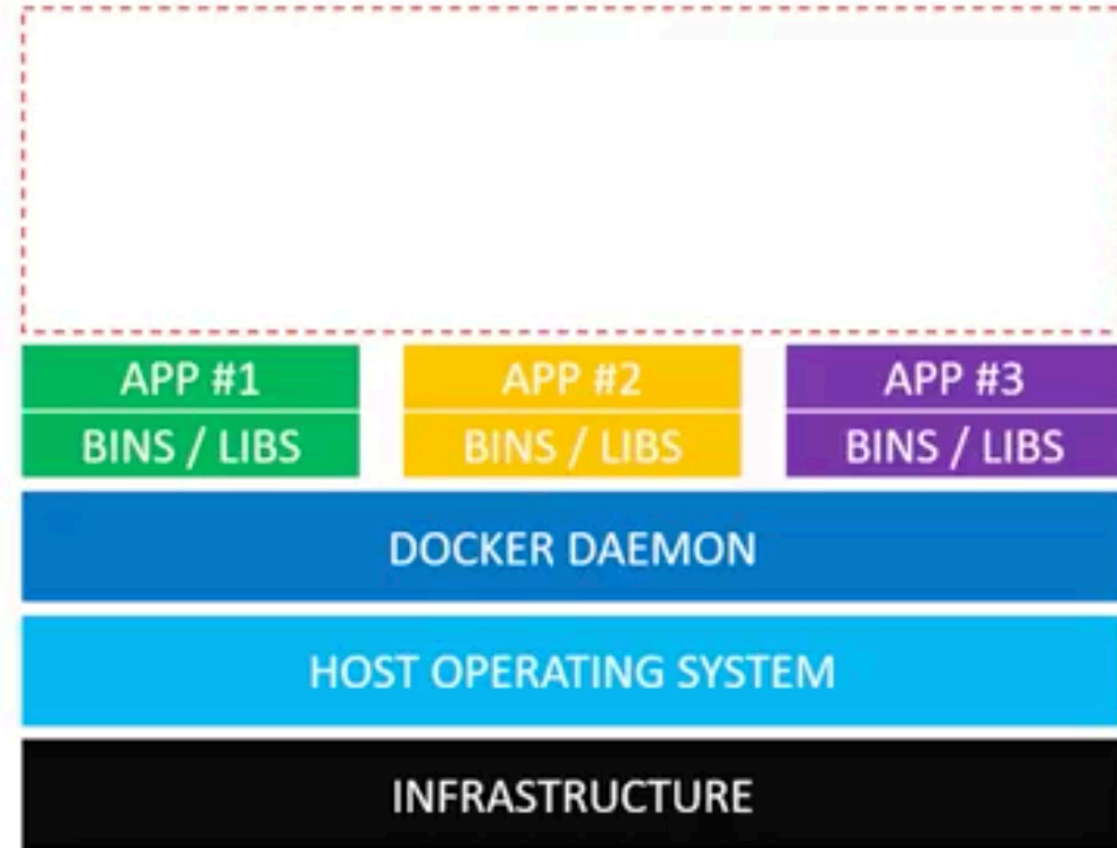
Virtual Machines

Starts in minutes



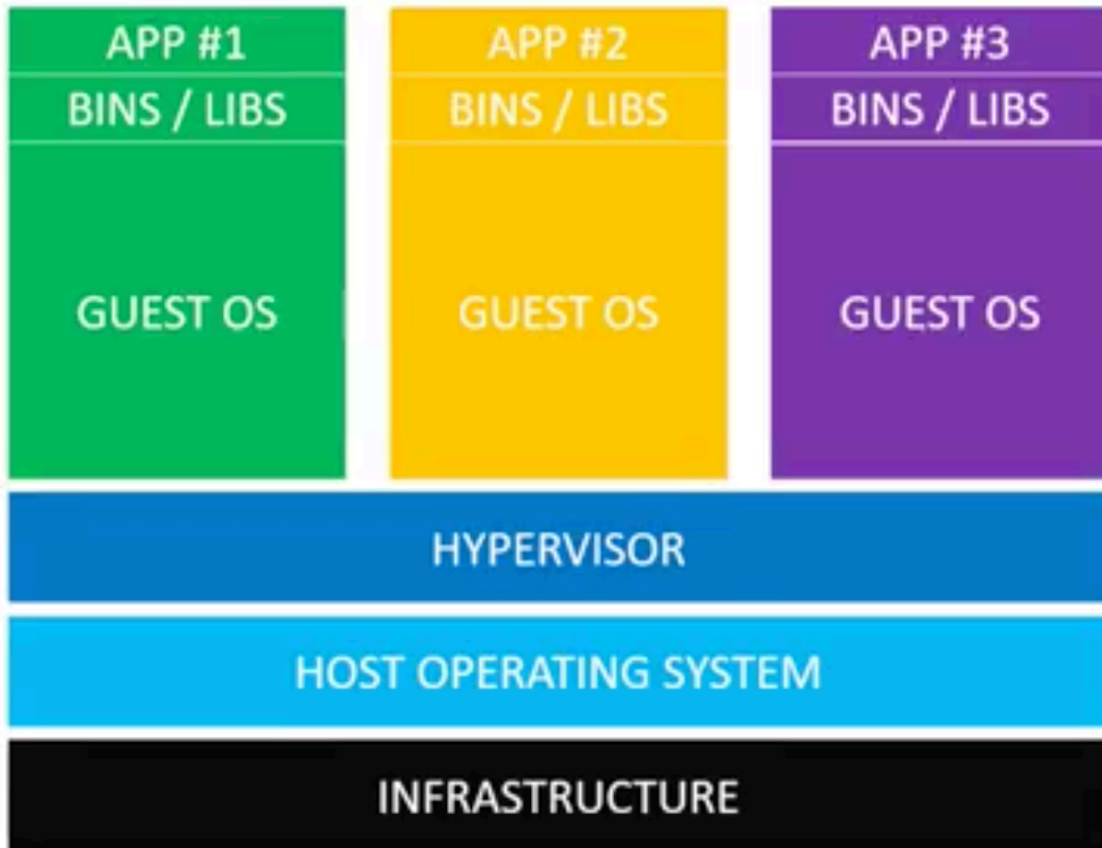
Virtual Machines

Starts in milliseconds



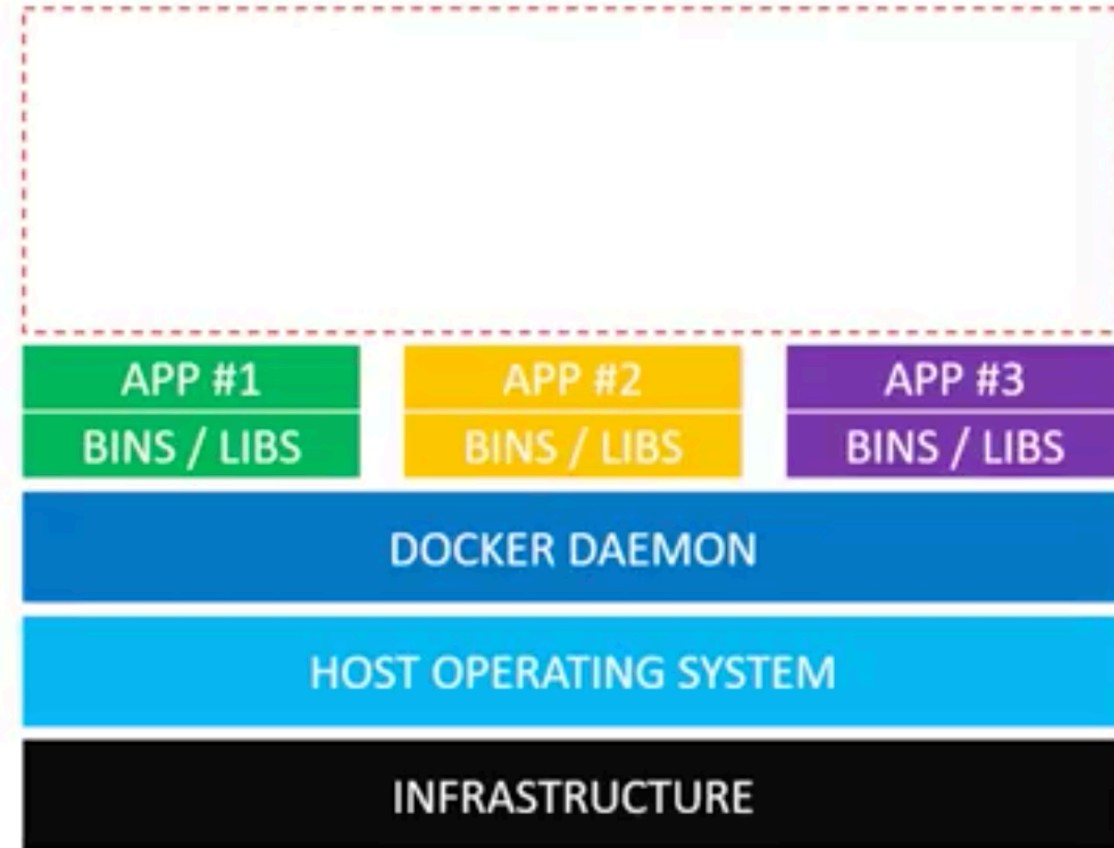
Docker Containers

Isolate systems



Virtual Machines

Isolate applications



Docker Containers

Installation de Docker sur Ubuntu

- <https://docs.docker.com/engine/install/ubuntu/#installation-methods>
- Install using the repository
- Before you install Docker Engine for the first time on a new host machine, you need to set up the Docker repository. Afterward, you can install and update Docker from the repository.
- SET UP THE REPOSITORY



1. Update the `apt` package index and install packages to allow `apt` to use a repository over HTTPS:

```
$ sudo apt-get update

$ sudo apt-get install \
  apt-transport-https \
  ca-certificates \
  curl \
  gnupg-agent \
  software-properties-common
```



2. Add Docker's official GPG key:

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

Verify that you now have the key with the fingerprint

9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88 , by searching for the last 8 characters of the fingerprint.



```
$ sudo apt-key fingerprint 0EBFCD88
```

```
pub    rsa4096 2017-02-22 [SCEA]  
       9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88  
uid          [ unknown] Docker Release (CE deb) <docker@docker.com>  
sub    rsa4096 2017-02-22 [S]
```

3. Use the following command to set up the **stable** repository. To add the **nightly** or **test** repository, add the word **nightly** or **test** (or both) after the word **stable** in the commands below. [Learn about nightly and test channels.](#)

Note: The `lsb_release -cs` sub-command below returns the name of your Ubuntu distribution, such as `xenial`. Sometimes, in a distribution like Linux Mint, you might need to change `$(lsb_release -cs)` to your parent Ubuntu distribution. For example, if you are using `Linux Mint Tessa`, you could use `bionic`. Docker does not offer any guarantees on untested and unsupported Ubuntu distributions.



x86_64 / amd64

armhf

arm64

ppc64le (IBM Power)

s390x (IBM Z)

```
$ sudo add-apt-repository \  
    "deb [arch=amd64] https://download.docker.com/linux/ubuntu \  
    $(lsb_release -cs) \  
    stable"
```

INSTALL DOCKER ENGINE

1. Update the `apt` package index, and install the *latest version* of Docker Engine and containerd, or go to the next step to install a specific version:

```
$ sudo apt-get update  
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
```



✔ Got multiple Docker repositories?

If you have multiple Docker repositories enabled, installing or updating without specifying a version in the `apt-get install` or `apt-get update` command always installs the highest possible version, which may not be appropriate for your stability needs.

2. To install a *specific version* of Docker Engine, list the available versions in the repo, then select and install:

a. List the versions available in your repo:

```
$ apt-cache madison docker-ce
```

```
docker-ce | 5:18.09.1~3-0~ubuntu-xenial | https://download.docker.com/linux/ubuntu xenial
docker-ce | 5:18.09.0~3-0~ubuntu-xenial | https://download.docker.com/linux/ubuntu xenial
docker-ce | 18.06.1~ce~3-0~ubuntu       | https://download.docker.com/linux/ubuntu xenial
docker-ce | 18.06.0~ce~3-0~ubuntu                 | https://download.docker.com/linux/ubuntu xenial
...
```

b. Install a specific version using the version string from the second column, for example,

```
5:18.09.1~3-0~ubuntu-xenial .
```



```
$ sudo apt-get install docker-ce=<VERSION_STRING> docker-ce-cli=<VERSION_STRING> containerd.io
```

3. Verify that Docker Engine is installed correctly by running the `hello-world` image.

```
$ sudo docker run hello-world
```

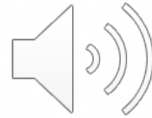
This command downloads a test image and runs it in a container. When the container runs, it prints an informational message and exits.

Example sur MacOS

```
(base) nabil@nabils-MacBook-Pro:~$ mkdir docker
mkdir: docker: File exists
(base) nabil@nabils-MacBook-Pro:~$ ls docker/
dockerfile      src
(base) nabil@nabils-MacBook-Pro:~$ cd docker/
(base) nabil@nabils-MacBook-Pro:docker$ ls
dockerfile      src
(base) nabil@nabils-MacBook-Pro:docker$ cd src/
(base) nabil@nabils-MacBook-Pro:src$ ls
index.php
(base) nabil@nabils-MacBook-Pro:src$ vim index.php
(base) nabil@nabils-MacBook-Pro:src$ pwd
/Users/nabil/docker/src
(base) nabil@nabils-MacBook-Pro:src$ cd ..
(base) nabil@nabils-MacBook-Pro:docker$ vim dockerfile
(base) nabil@nabils-MacBook-Pro:docker$
```


Le fichier dockerfile

```
FROM php:7.2-apache  
COPY src/ /var/www/html/
```



```
EXPOSE 80
```

Example sur MacOS

```
<html>
<body>

<h1>My first PHP page</h1>

<?php
echo "<h2>Salam Everybody, Hope you are all doing well during this containment period! </h2>";

?>

</html>
```

Docker build

```
[(base) nabil@nabils-MacBook-Pro:docker nabil$ docker build -t hello-world .  
Sending build context to Docker daemon 3.584kB  
Step 1/3 : FROM php:7.2-apache  
----> fc63b5c72cff  
Step 2/3 : COPY src/ /var/www/html/  
----> 0a6c55fab6a0  
Step 3/3 : EXPOSE 80  
----> Running in bcc46824b9d4  
Removing intermediate container bcc46824b9d4  
----> 8242c92ef327  
Successfully built 8242c92ef327  
Successfully tagged hello-world:latest  
(base) nabil@nabils-MacBook-Pro:docker nabil$
```



Lancer l'image de l'application

```
(base) nabils-MacBook-Pro:docker nabil$ docker: Error response from daemon: driver failed programming external connectivity on endpoint vigorous_dubinsky (dbc131cb932fe8f89e854b5129590518d78ad12dd0f0ee4c3eade7771e033a4): Bind for 0.0.0.0:80 failed: port is already allocated.
```

```
[1]+  Exit 125          docker run -p 80:80 -v /Users/nabil/docker/src/:/var/www/html hello-world
```

```
(base) nabils-MacBook-Pro:docker nabil$ docker run -p 80:80 -v /Users/nabil/docker/src/:/var/www/html hello-world  
docker: Error response from daemon: dial unix docker.raw.sock: connect: connection refused.  
See 'docker run --help'.
```

```
(base) nabils-MacBook-Pro:docker nabil$ docker run -p 80:80 -v /Users/nabil/docker/src/:/var/www/html hello-world  
docker: Error response from daemon: dial unix docker.raw.sock: connect: connection refused.  
See 'docker run --help'.
```

```
(base) nabils-MacBook-Pro:docker nabil$ docker run -p 80:80 -v /Users/nabil/docker/src/:/var/www/html hello-world  
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message  
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
```

```
[Sat Apr 11 07:28:00.324479 2020] [mpm_prefork:notice] [pid 1] AH00163: Apache/2.4.38 (Debian) PHP/7.2.29 configured -- resuming normal operations  
[Sat Apr 11 07:28:00.325119 2020] [core:notice] [pid 1] AH00094: Command line: 'apache2 -D FOREGROUND'
```

Verifier l'application

My first PHP page

Salam Everybody, Hope you are all doing well during this containment period!

