

Devops

Docker

2023/2024



- Présentation disponible à l'adresse: <https://gounthar.github.io/gounthar/cours-devops-docker/updated-dependencies>
- Version PDF de la présentation : [Cliquez ici](#)
- Contenu sous licence [Creative Commons Attribution 4.0 International License](#)
- Code source de la présentation: <https://github.com/gounthar/gounthar/cours-devops-docker>

Comment utiliser cette présentation ?

- Pour naviguer, utilisez les flèches en bas à droite (ou celles de votre clavier)
 - Gauche/Droite: changer de chapitre
 - Haut/Bas: naviguer dans un chapitre
- Pour avoir une vue globale : utiliser la touche "o" (pour "**Overview**")

Bonjour !

La suite: vers le bas ↓





Bruno VERACHTEN

- Sr Developer Relations chez CloudBees pour le projet Jenkins 
- Me contacter :
 -  gounthar@gmail.com
 -  gounthar
 - <https://bruno.verachten.fr/>
 -  Bruno Verachten
 -  @poddigue

Et vous ?



A propos du cours

- Première itération d'une découverte Docker dans le cadre du DevOps
- Contenu entièrement libre et open-source
- Méchamment basé sur le travail de Damien Duportal et Amaury Willemant
 - N'hésitez pas ouvrir des Pull Request si vous voyez des améliorations ou problèmes: sur cette page (😉 wink wink)

Plan

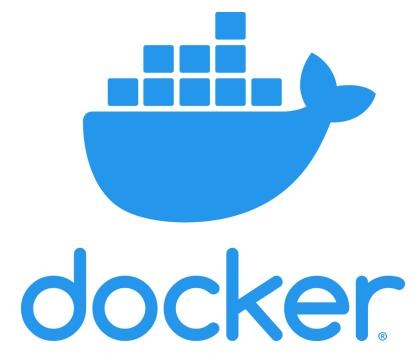
- Intro
- Containers
- Images
- Fichiers, nommage, inspect
- Volumes
- Réseaux
- Docker Compose
- Bonus

La suite: vers la droite ➔

Docker

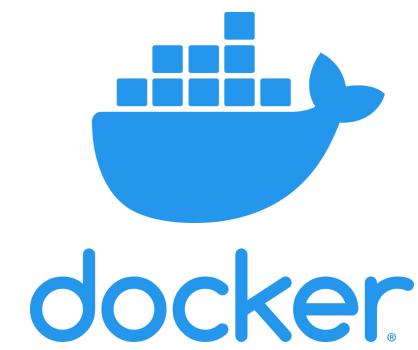
"La Base"

Pourquoi ?



🤔 Quel est le problème ?

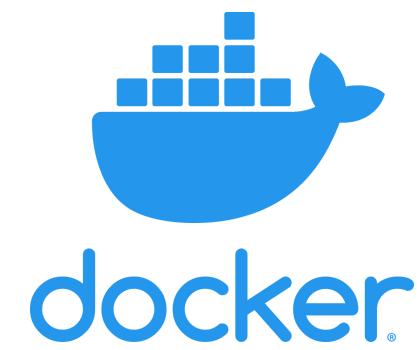
Pourquoi commencer par un problème ?



🤔 Commençons plutôt par une définition:

Docker c'est ...

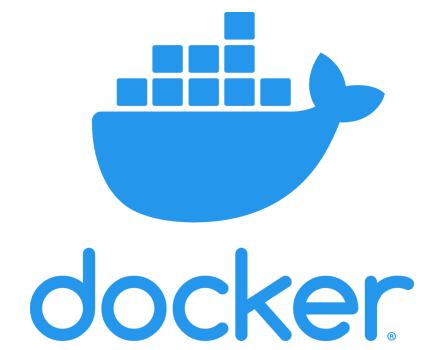
Pourquoi commencer par un problème ?



🤔 Définition quelque peu datée (2014):

Docker is ...

Pourquoi commencer par un problème ?

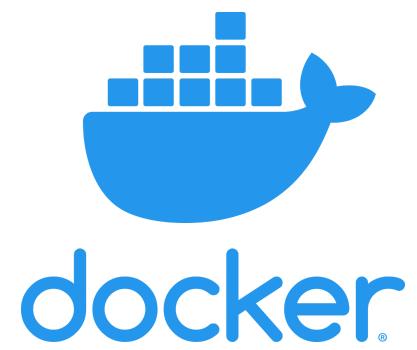


🤔 Définition quelque peu datée (2014):

Docker is a toolset for Linux containers designed to 'build, ship and run' distributed applications.

<https://www.infoq.com/articles/docker-future/>

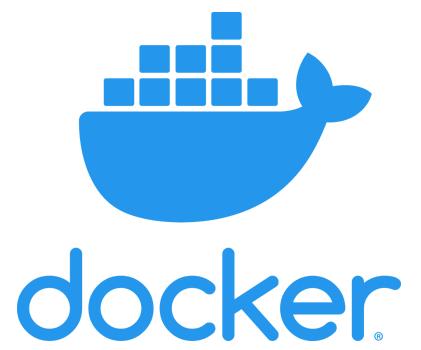
Linux containers ?



🤔 Nous voilà bien...

C'est quoi un container?

Linux containers ?

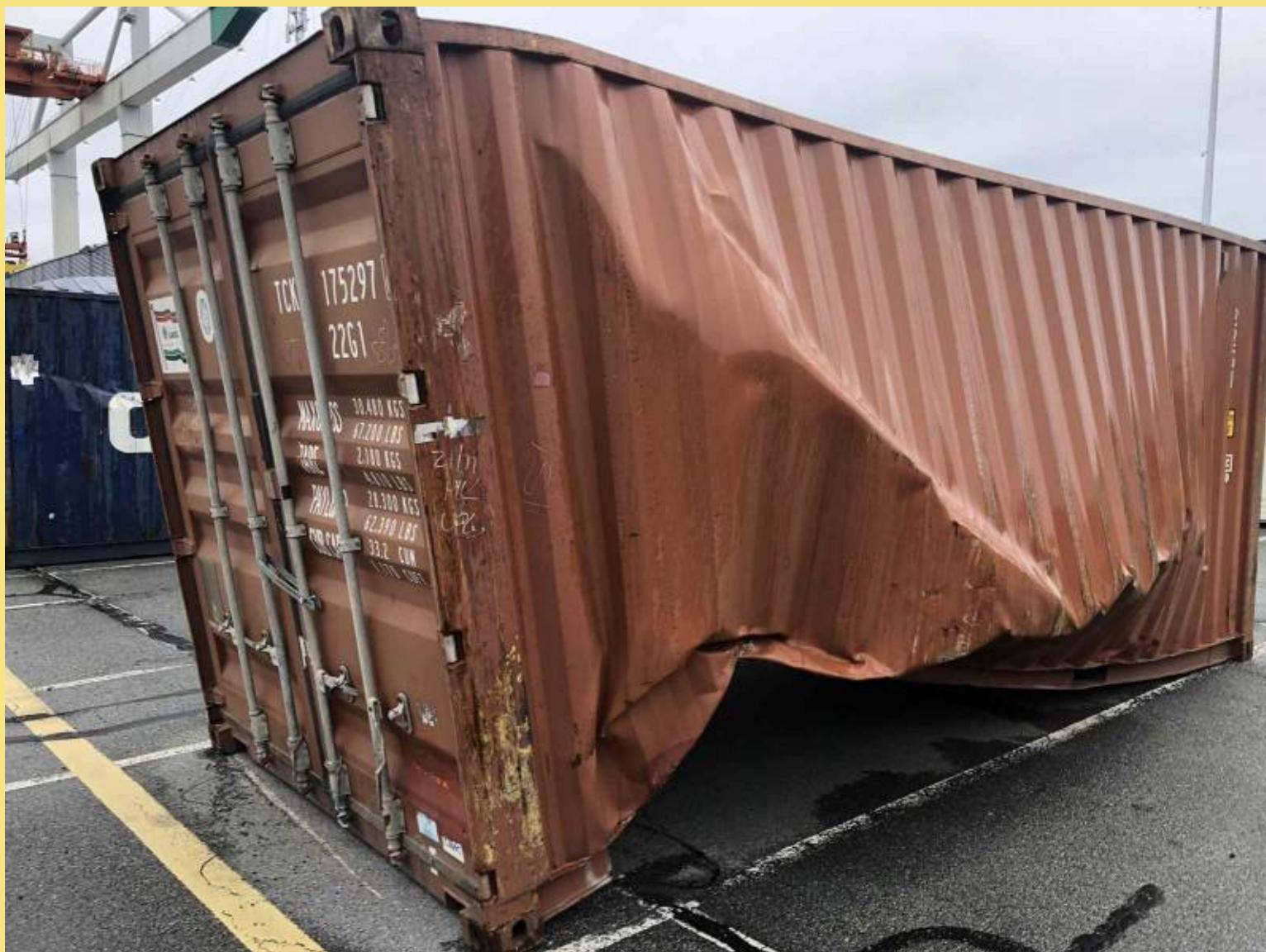


🤔 Nous voilà bien...

C'est quoi un container?

Vous voulez la version enfant de 5 ans? Je ne crois pas...

Docker est vieux



10 ans déjà...

The screenshot shows the homepage of the Docker Pirates website. The header features a skull and crossbones icon. The main title is "Docker Pirates ARMed with explosive stuff". Below the title, there is a subtext: "Roaming the seven seas in search for golden container plunder." A sidebar on the left contains links to "HOME", "GETTING STARTED", "DOWNLOADS", "FAQ", "COMMUNITY", "ABOUT US", "CREW", and "DONATE". At the bottom, there are links for "© 2020 Hypriot", "Legal Notice", and "Edit this blog on GitHub".

Docker on Raspberry Pi Workshop in Brussels

Thu, Mar 17, 2016

A couple of weeks ago we have been invited by the fine folks over at the Docker User Group in Brussels to help conduct a workshop. And of course this workshop was about Docker. Still it was not your ordinary Docker workshop.

It was special because instead of being a workshop about Docker on big servers it was about Docker on really small ARM devices. The very same devices that power the upcoming IoT revolution.

Turns out it is really amazing what you can do with Docker on those tiny machines.



- parler de mes débuts en 2015/2016
- parler de balena

On n'avait pas parlé d'un problème?

	?	?	?	?	?	?	?
Static Website	?	?	?	?	?	?	?
Web Frontend	?	?	?	?	?	?	?
Background Workers	?	?	?	?	?	?	?
User DB	?	?	?	?	?	?	?
Analytics DB	?	?	?	?	?	?	?
Queue	?	?	?	?	?	?	?
	Development VM	QA Server	Single Prod Server	Production Cluster	Public cloud	Developer's Laptop	Customer Servers

Source: <https://blog.docker.com/2013/08/paas-present-and-future/>

Problème de temps **exponentiel**

Déjà vu ?

L'IT n'est pas la seule industrie à résoudre des problèmes...

	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?

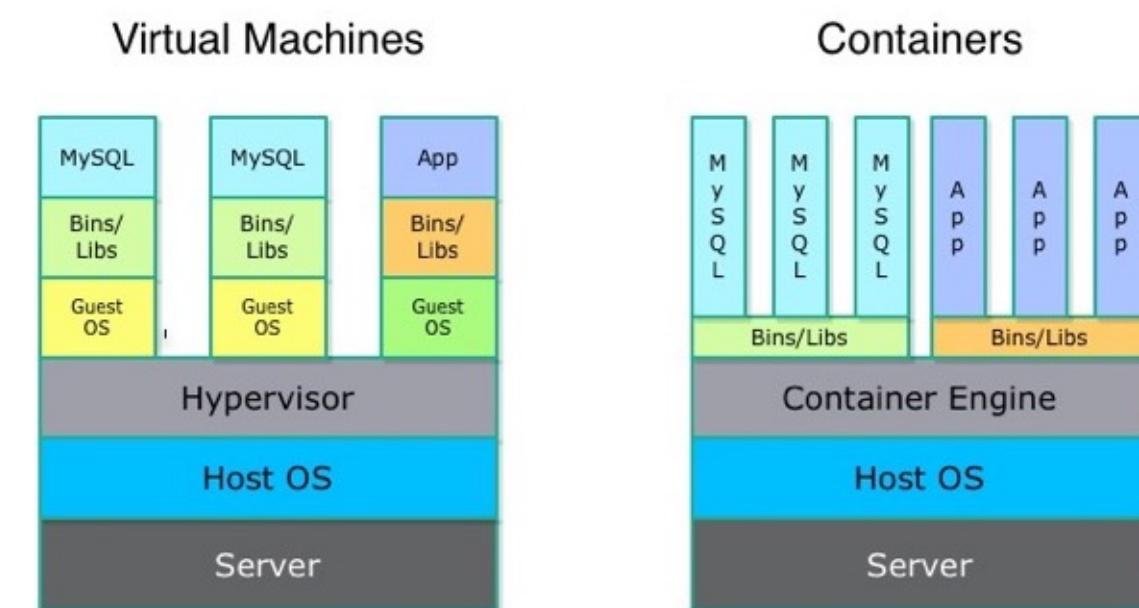
Solution: Le conteneur intermodal

"Separation of Concerns"



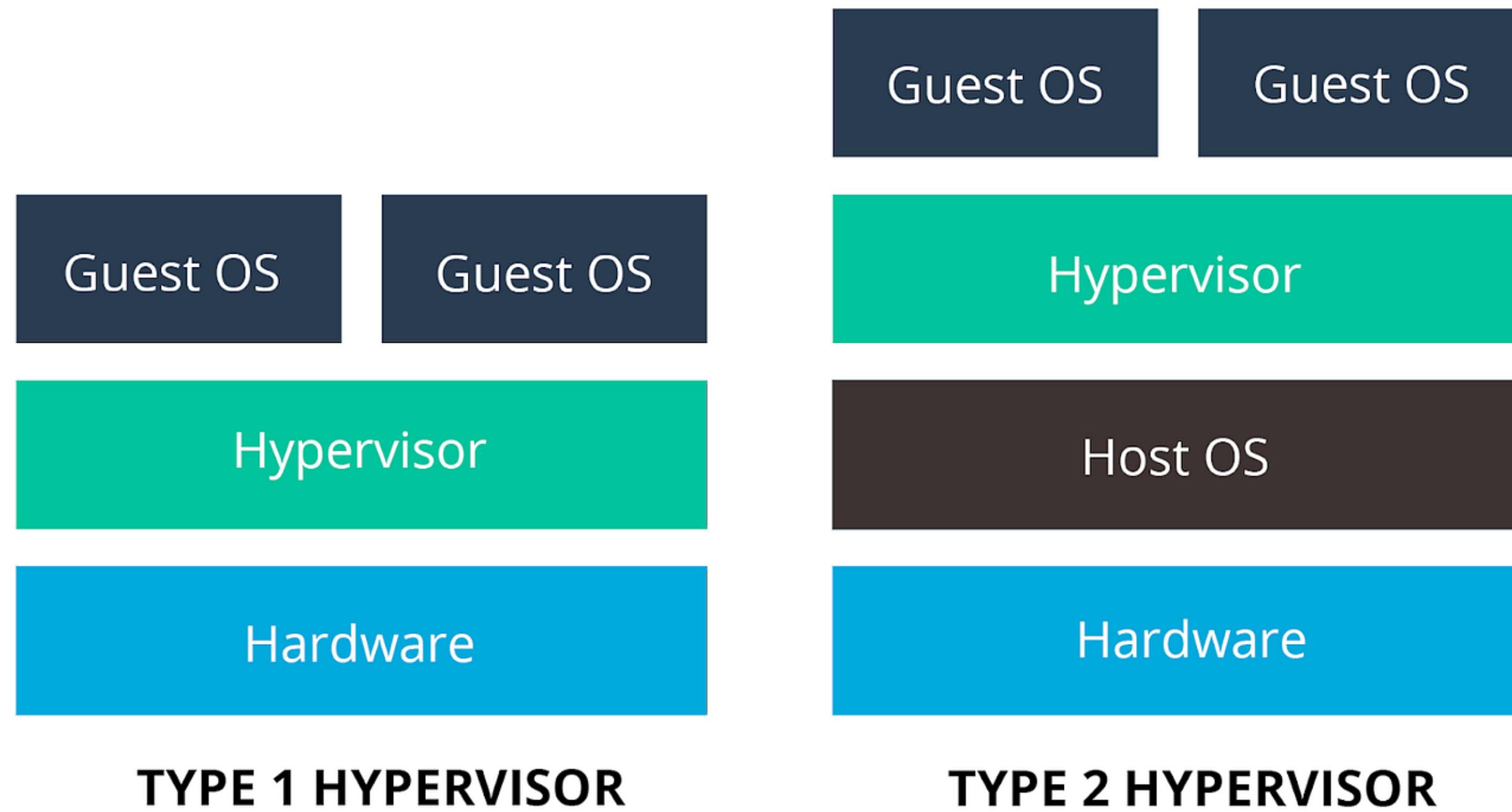
Comment ça marche ?

"Virtualisation Légère"



Comment ça marche ?

Virtualisation



Comment ça marche ?

Virtualisation

Type 1 Hypervisors

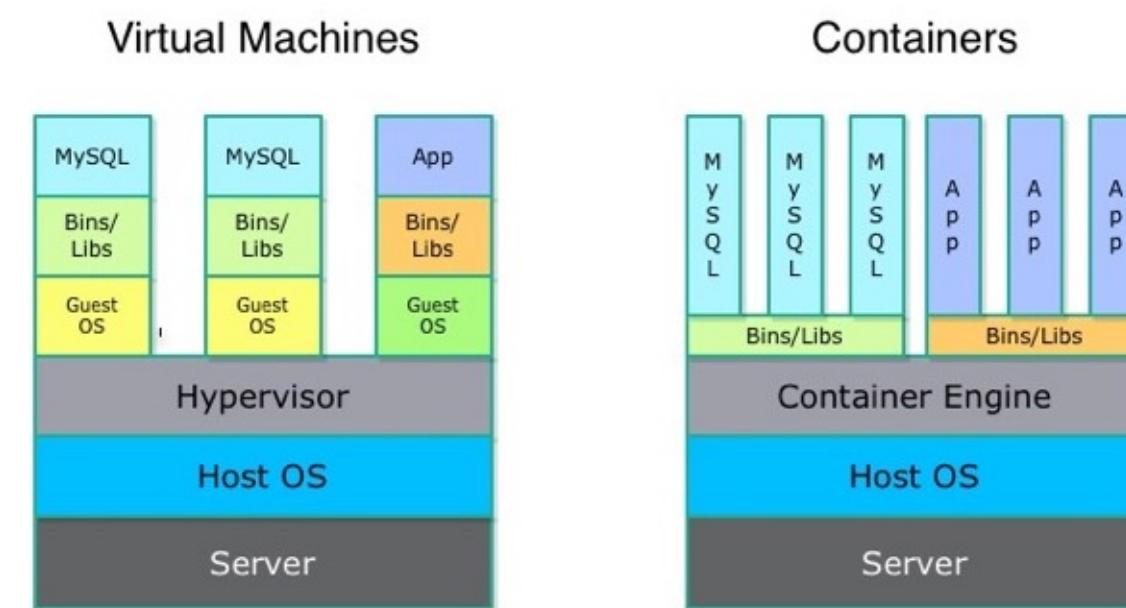


Type 2 Hypervisors



Comment ça marche ?

"Virtualisation Légère"



- Légère, vraiment?

Challenge: We Have a Winner!



VICTOR COISNE

Oct 20 2015

At the [end of DockerCon 2015](#), [Dieter Reuter from Hypriot](#) presented a demo running 500 Docker containers on a [Raspberry Pi 2](#) device but he knew that number could be at least doubled.

TL;DR Dieter was right.

Légère, vraiment!

<http://web.archive.org/web/20200810061020/https://www.docker.com/blog/raspberry-pi-dockercon-challenge-winner/>

A big congratulations to [Damien Duportal](#), [Nicolas de Loof](#) and [Yoann Dubreuil](#) on running 2500 web servers in Docker containers on a single Raspberry Pi 2!

Damien, Nicolas and Yoann each win a complimentary pass to [DockerCon EU 2015](#) and speaking slot during the conference to demo how they accomplished this.

#RpiDocker 2740 web servers running on a #Rpi, could have more. But using a patched docker daemon with a hack that isn't a valuable fix. —

Nicolas De loof (@ndeloof) October 13, 2015

Post Tags

- dockercon
- DockerCon Europe
- raspberry pi

Categories

- All
- Products
- Community
- Engineering
- Company



Aside from the above issues (and there are more caveats in the documentation), my installation is a pretty faithful reproduction of ESXi on a server or home lab. However, ESXi itself uses about 1.3GB of RAM, leaving only around 6.5GB usable. This could support perhaps 4-5 small VMs, but would be much more useful for running containerised applications and eliminating the individual VM overhead.

Légère, vraiment!

Proof of Concept

Remember that Flings are just proofs of concept and not always aimed at final production. VMware suggests a range of hardware options for ARM-based workloads and this is more likely where we will see the initial development of ESXi on ARM. The challenge for end users here is to determine whether the price/performance/power/cooling ratio works better on ARM than x86. Of course applications also need to be available, but it's not hard to find ARM-compiled versions of most open source software and operating systems.

localhost.brk1.brookendlab.com

Host

Manage

Virtual Machines

Storage

Networking

Version: ESXi on Arm Fling (Build 16966451)

State: Normal (not connected to any vCenter Server)

Uptime: 0 days

CPU: FREE: 8 GHz USED: 17 MHz CAPACITY: 6 GHz

MEMORY: FREE: 6.58 GB USED: 1.27 GB CAPACITY: 7.86 GB

STORAGE: FREE: 0 B USED: 0 B CAPACITY: NaN%

You are currently using ESXi in evaluation mode. This license will expire in 180 days.

Hardware: Manufacturer: Raspberry Pi Foundation

Configuration: Image profile: ESXi-7.0.0-16966451-standard (VMware, Inc.)

Conteneur != VM

"Separation of concerns": 1 "tâche" par conteneur

VM

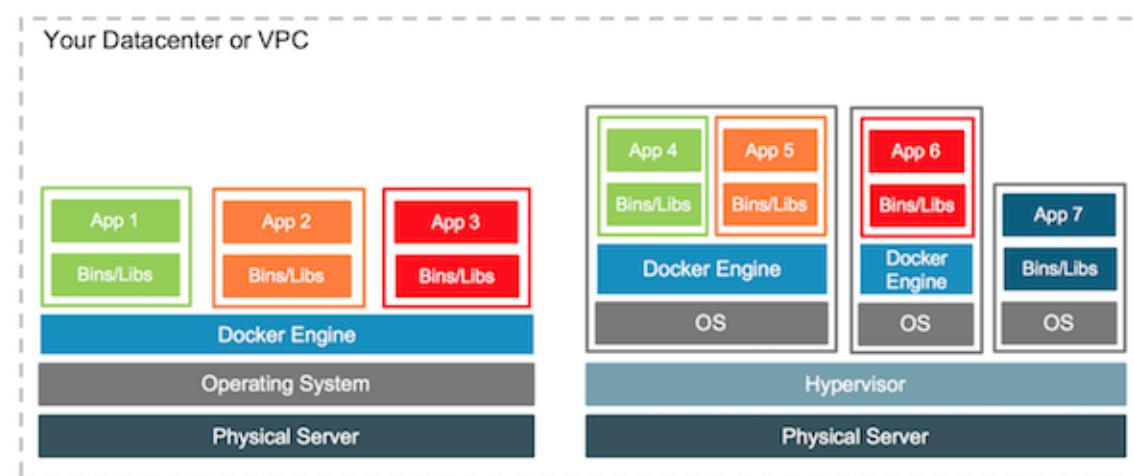


Containers



VMs & Conteneurs

Non exclusifs mutuellement





Exercice : Votre premier conteneur

C'est à vous (ouf) !

- Retournez dans Gitpod
- Dans un terminal, tapez la commande suivante :

```
docker container run hello-world  
# Équivalent de l'ancienne commande 'docker run'
```

Copy



Anatomie

- Un service "Docker Engine" tourne en tâche de fond et publie une API REST
- La commande `docker run ...` a envoyé une requête POST au service
- Le service a téléchargé une **Image** Docker depuis le registre **DockerHub**,
- Puis a exécuté un **conteneur** basé sur cette image



Exercice : Où est mon conteneur ?

C'est à vous !

```
docker container ls --help
# ...
docker container ls
# ...
docker container ls --all
```

Copy

⇒ 🤔 comment comprenez vous les résultats des 2 dernières commandes ?

✓ Solution : Où est mon conteneur ?

Le conteneur est toujours présent dans le "Docker Engine" même en étant arrêté

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES	Copy
109a9cdd3ec8	hello-world	"/hello"	33 seconds ago	Exited (0) 17 seconds ago		festive_faraday	

- Un conteneur == une commande "conteneurisée"
 - cf. colonne "**COMMAND**"
- Quand la commande s'arrête : le conteneur s'arrête
 - cf. code de sortie dans la colonne "**STATUS**"



Exercice : Cycle de vie d'un conteneur simple

- Lancez un nouveau conteneur nommé `bonjour`
 - 💡 `docker container run --help` ou Documentation en ligne
- Affichez les "logs" du conteneur (==traces d'exécution écrites sur le stdout + stderr de la commande conteneurisée)
 - 💡 `docker container logs --help` ou Documentation en ligne
- Lancez le conteneur avec la commande `docker container start`
 - Regardez le résultat dans les logs
- Supprimez le container avec la commande `docker container rm`

✓ Solution : Cycle de vie d'un conteneur simple

```
docker container run --name=bonjour hello-world
# Affiche le texte habituel

docker container logs bonjour
# Affiche le même texte : pratique si on a fermé le terminal

docker container start bonjour
# N'affiche pas le texte mais l'identifiant unique du conteneur 'bonjour'

docker container logs bonjour
# Le texte est affiché 2 fois !

docker container ls --all
# Le conteneur est présent
docker container rm bonjour
docker container ls --all
# Le conteneur n'est plus là : il a été supprimé ainsi que ses logs

docker container logs bonjour
# Error: No such container: bonjour
```

Copy



Que contient "hello-world" ?

- C'est une "image" de conteneur, c'est à dire un modèle (template) représentant une application auto-suffisante.
 - On peut voir ça comme un "paquetage" autonome
- C'est un système de fichier complet:
 - Il y a au moins une racine /
 - Ne contient que ce qui est censé être nécessaire (dépendances, librairies, binaires, etc.)

Docker Hub

- <https://hub.docker.com/> : C'est le registre d'images "par défaut"
 - Exemple : Image officielle de conteneur "Ubuntu"
- 🎓 Cherchez l'image hello-world pour en voir la page de documentation
 - 💡 pas besoin de créer de compte pour ça
- Il existe d'autre "registres" en fonction des besoins (GitHub GHCR, Google GCR, etc.)



Exercice : conteneur interactif

- Quel distribution Linux est utilisée dans le terminal Gitpod ?
 - 💡 Regardez le fichier `/etc/os-release`
- Exécutez un conteneur interactif basé sur `alpine:3.17` (une distribution Linux ultra-légère) et regardez le contenu du fichier au même emplacement
 - 💡 `docker container run --help`
 - 💡 Demandez un `tty` à Docker
 - 💡 Activez le mode interactif
- Exécutez la même commande dans un conteneur basé sur la même image mais en **NON** interactif
 - 💡 Comment surcharger la commande par défaut ?

✓ Solution : conteneur interactif

```
$ cat /etc/os-release
# ... Ubuntu ...

$ docker container run --tty --interactive alpine:3.17
/ # cat /etc/os-release
# ... Alpine ...
# Notez que le "prompt" du terminal est différent DANS le conteneur
/ # exit
$ docker container ls --all

$ docker container run alpine:3.17 cat /etc/os-release
# ... Alpine ...
```

Copy



Exercice : conteneur en tâche de fond

- Exécutez un conteneur, basé sur l'image nginx en tâche de fond ("Background"), nommé webserver-1
 - 💡 On parle de processus "détaché" (ou bien "démonisé")
 - ⚠ Pensez bien à docker container ls
- Regardez le contenu du fichier /etc/os-release dans ce conteneur
 - 💡 docker container exec
- Essayez d'arrêter, démarrer puis redémarrer le conteneur
 - ⚠ Pensez bien à docker container ls à chaque fois
 - 💡 stop, start, restart

✓ Solution : conteneur en tâche de fond

```
docker container run --detach --name=webserver-1 nginx
# <ID du conteneur>

docker container ls
docker container ls --all

docker container exec webserver-1 cat /etc/os-release
# ... Debian ...

docker container stop webserver-1
docker container ls
docker container ls --all

docker container start webserver-1
docker container ls
docker container ls --all

docker container start webserver-1
docker container ls
```

Copy

Checkpoint

- Docker essaye de résoudre le problème de l'empaquetage le plus "portable" possible
 - On n'en a pas encore vu les effets, ça arrive !
 - Vous avez vu qu'un conteneur permet d'exécuter une commande dans un environnement "préparé"
 - Catalogue d'images Docker par défaut : Le Docker Hub
 - Vous avez vu qu'on peut exécuter des conteneurs selon 3 modes :
 - "One shot"
 - Interactif
 - En tâche de fond
- ⇒  Mais comment ces images sont-elles fabriquées ? Quelle confiance leur accorder ?



Diapositive12



Diapositive13



Diapositive 14



Diapositive 15



Diapositive 16



Diapositive 16



Diapositive 17



Diapositive 18



Diapositive 19



Diapositive20



Diapositive21



Diapositive22



Diapositive23



Diapositive24



Diapositive25



Diapositive26



Diapositive27



Diapositive28



Diapositive29



Diapositive30



Diapositive31



Diapositive32



Diapositive33



Diapositive34



Diapositive35



Diapositive36



Diapositive37



Diapositive38



Diapositive39



Diapositive40



Diapositive41



Diapositive42



Diapositive43



Diapositive44



Diapositive45



Diapositive46



Diapositive47



Diapositive48



Diapositive49



Diapositive50



Diapositive51



Diapositive52



Diapositive53



Diapositive54



Diapositive55



Diapositive56





Diapositive58



Diapositive59



Diapositive60



Diapositive61



Diapositive62

Bibliographie

Ligne de commande

- <https://tldp.org>
- <https://en.wikipedia.org/wiki/POSIX>
- https://en.wikipedia.org/wiki/Read%E2%80%93eval%E2%80%93print_loop
- <https://linuxhandbook.com/linux-directory-structure/>

Git / VCS

- <https://docs.github.com>
- <https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control>
- <http://martinfowler.com/bliki/VersionControlTools.html>
- <http://martinfowler.com/bliki/FeatureBranch.html>
- <https://about.gitlab.com/2014/09/29/gitlab-flow/>
- <https://www.atlassian.com/git/tutorials/comparing-workflows>
- <http://nvie.com/posts/a-successful-git-branching-model/>

Intégration Continue

- <http://martinfowler.com/articles/continuousIntegration.html>
- <http://martinfowler.com/bliki/ContinuousDelivery.html>
- <https://jaxenter.com/implementing-continuous-delivery-117916.html>
- <https://technologyconversations.com/2014/04/29/continuous-delivery-introduction-to-concepts-and-tools/>
- <http://blog.arungupta.me/continuous-integration-delivery-deployment-maturity-model>
- <http://blog.crisp.se/2013/02/05/yassalsundman/continuous-delivery-vs-continuous-deployment>

Docker

- <https://gounthar.github.io/cours/cnam-docker-2018>
- <https://kodekloud.com/blog/docker-for-beginners/>
- <https://www.slideshare.net/dotCloud/why-docker>
- <https://docs.docker.com/engine/reference/builder/>
- <https://www.r-bloggers.com/2021/05/best-practices-for-r-with-docker/>
- <https://github.com/wagoodman/dive>
- <https://docs.docker.com/engine/tutorials/networkingcontainers/>
- <https://towardsdatascience.com/docker-networking-919461b7f498>

Merci !

✉ gounthar@gmail.com

Slides: <https://gounthar.github.io/gounthar/cours-devops-docker/updated-dependencies>



Source on : <https://github.com/gounthar/gounthar/cours-devops-docker>