



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 "**O**verview")

Bruno VERACHTEN

- Sr Developer Relations chez CloudBees pour le projet Jenkins 🧑🔧
- Me contacter :
 - ✉️ gounthar@gmail.com
 - 🐙 [gounthar](#)
 - <https://bruno.verachten.fr/>
 - [in](#) Bruno Verachten
 - 🐦 [@poddingue](#)

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 ?

 docker logo monochromatic

🤔 Quel est le problème ?

Pourquoi commencer par un problème ?

 docker logo monochromatic

🤔 Commençons plutôt par une définition:

Docker c'est ...

Pourquoi commencer par un problème ?

 docker logo monochromatic

🤔 Définition quelque peu datée (2014):

Docker is ...

Pourquoi commencer par un problème ?


 docker logo monochromatic

🤔 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 ?

 docker logo monochromatic

🤔 Nous voilà bien...

C'est quoi un container?

Linux containers ?

 docker logo monochromatic

🤔 Nous voilà bien...

C'est quoi un container?

[Vous voulez la version enfant de 5 ans? Je ne crois pas...](#)

Docker est vieux

 containerconteneuraccidenteavariejpg 5e8b87ccc8c27

10 ans déjà...

On n'avait pas parlé d'un problème?

matrixfromhell

Problème de temps **exponentiel**

Déjà vu ?

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

 also a matrix from hell

Solution: Le conteneur intermodal

"Separation of Concerns"

 blue shipping container

Comment ça marche ?

"Virtualisation **Légère**"

 container vs vm

Conteneur != VM

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

 vm and container

VMs & Conteneurs

Non exclusifs mutuellement

 cont vm not excl

Comment ça marche ?

Exercice : Votre premier conteneur

C'est à vous (ouf) !

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

```
docker container run hello-world  
# Equivalent 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
109a9cdd3ec8	hello-world	"/hello"	33 seconds ago	Exited (0) 17 seconds ago		festive_faraday

Copy

- 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'emballage 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

 Diapositive14

 Diapositive15

 Diapositive16

 Diapositive16

 Diapositive17

 Diapositive18

 Diapositive19

 Diapositive20

 Diapositive21

 Diapositive23

 Diapositive24

 Diapositive25

 Diapositive26

 Diapositive27

 Diapositive28

 Diapositive29

 Diapositive30

 Diapositive31

 Diapositive32

 Diapositive34

 Diapositive35

 Diapositive36

 Diapositive37

 Diapositive38

 Diapositive39

 Diapositive40

 Diapositive41

 Diapositive42

 Diapositive44

 Diapositive45

 Diapositive46

 Diapositive47

 Diapositive48

 Diapositive49

 Diapositive50

 Diapositive51

 Diapositive52

 Diapositive54

 Diapositive55

 Diapositive56

 Diapositive57

 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>