Tutoriel Docker Documentation

Release 0.1.0 (2018-03-09)

id3 Technologies

2020-9-14 12:6

Contents

Fig. 1: Logo Docker https://www.docker.com

Fig. 2: Un essaim de conteneurs géré par docker swarm

```
https://fr.wikipedia.org/wiki/Docker_(logiciel)
```

See also:

- https://gitlab.com/gdevops
- https://gdevops.gitlab.io/tuto_devops/
- https://gitlab.com/gdevops/tuto_docker
- https://gdevops.gitlab.io/tuto_docker/
- https://docs.docker.com/
- https://www.docker.com/
- https://twitter.com/Docker/lists/docker-captains/members
- https://twitter.com/docker
- https://www.youtube.com/user/dockerrun
- https://plus.google.com/communities/108146856671494713993
- http://www.slideshare.net/docker
- http://training.play-with-docker.com/
- https://hub.docker.com/u/id3pvergain/

See also:

- https://jpetazzo.github.io/2018/03/28/containers-par-ou-commencer/

Chapter 1

Introduction à Docker

Fig. 1: Le logo Docker

See also:

- https://fr.wikipedia.org/wiki/Docker_(logiciel)
- https://www.docker.com/
- https://twitter.com/docker
- https://en.wikipedia.org/wiki/Cloud_computing
- https://fr.wikipedia.org/wiki/Virtualisation
- https://en.wikipedia.org/wiki/Devops
- https://docs.docker.com/get-started/

Contents

1.1 Pourquoi utiliser docker ?

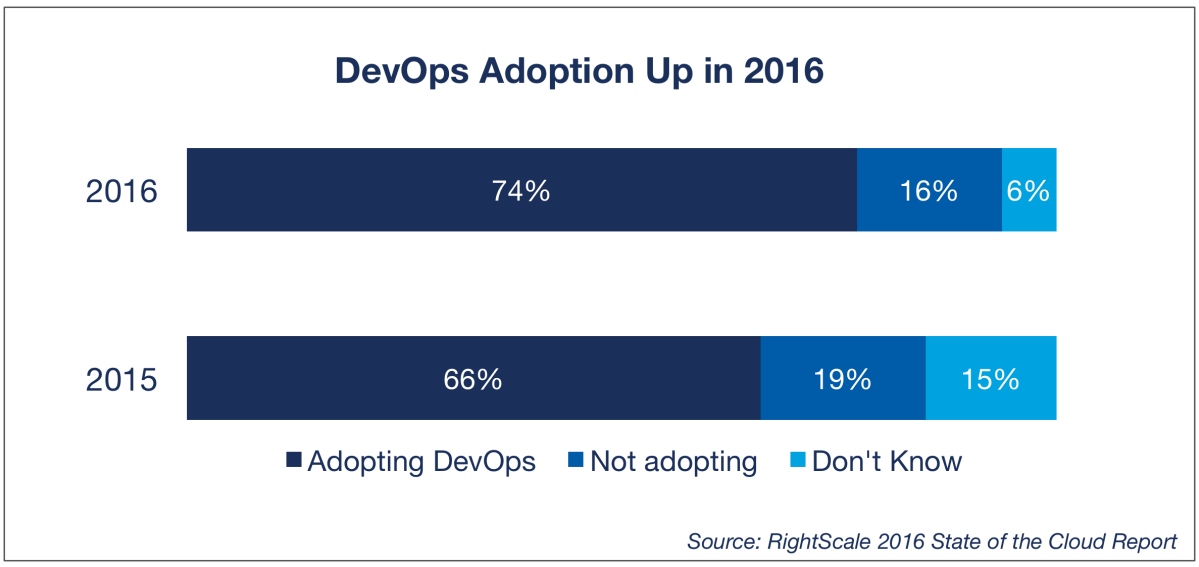
1.1.1 Transformation de la DSI des entreprises

See also:

- https://actu.alfa-safety.fr/service-aws/devops-et-transformation-dela-dsi/

Trois évolutions majeures convergent depuis peu et poussent à une transformation de la DSI des entreprises:

- la pression du time to market : l'accélération du rythme d'évolution des applications, en particulier web, pour sortir au plus vite de nouvelles fonctionnalités et répondre aux besoins du marché
- Le Devops : pour livrer plus vite, les équipes de Dev font évoluer leurs méthodes pour rendre les déploiements plus rapides, fréquents et fluides, et attendent que l'infrastructure, les « Ops » évoluent au même rythme.
- le cloud public : il a atteint un niveau de maturité et d'efficacité tel qu'une majorité des DSI travaille maintenant à l'intégrer, souvent sous la pression des équipes de développement,



1.1.2 Pour donner davantage d'autonomie aux développeurs

See also:

- https://actu.alfa-safety.fr/devops/docker-en-production/

Avec Docker, donnez davantage d'autonomie aux développeurs

L'un des atouts du conteneur est de donner davantage d'autonomie au développeur. Ce dernier doit pouvoir travailler sur son application sans se soucier de la configuration de la machine sur laquelle il travaille : il doit pouvoir développer sur son poste de travail et pousser son conteneur sur un serveur de test, puis pré-production, et jusqu'en production sans rencontrer de difficultés.

Le développeur doit aussi pouvoir modifier son docker et en gérer les versions sans se préoccuper des conséquences pour la production.

En résumé, un des bénéfices du conteneur c'est qu'il doit pouvoir se déployer n'importe où en toute sécurité.

1.1.3 Faire évoluer son système d'information

See also:

- https://linuxfr.org/forums/linux-general/posts/docker-en-prod

Bonjour à tous, après la virtualisation il y a docker (qui a le vent en poupe). Je me dis qu'il y a peut-être quelque chose à faire. Le concept est assez simple, l'utilisation a l'air souple.

Comme par hasard je dois migrer le serveur intranet de ma boite, actuellement il est en RHE 5.x et depuis la version 6.5 docker est intégré par RedHat. Il sert à plusieurs choses :

- dev pour les sites internet;
- PIM interne
- Cacti
- ...

J'aimerais bien avoir un environnement qui me permette d'ajouter Ruby par exemple sans tout péter sur les autres devs, ou installer la version de php 7 alors que le reste doit rester en php 5, la lib rrdtool 1.4 alors qu'un autre doit rester en 1.2... Enfin le genre de chose **bien prise de tête à gérer**.

Après avoir lu pas mal de doc autres que celles de RH je me rend compte qu'à chaque fois se sont des environnements de dev qui sont mis en place mais jamais de la prod, du vrai, du concret, avec du users bien bourrin.

Avez-vous des exemples ou des expériences (réussi ou pas) d'archi en prod ?

1.1.4 Pour que ça fonctionne aussi sur une autre machine

See also:

- http://putaindecode.io/fr/articles/docker/

Il était une fois un jeune développeur qui codait tranquillement sur son ordinateur. Il était pressé car comme tout étudiant qui se respecte il devait présenter son travail le lendemain matin. Après des heures de travail, l'application était là, et elle fonctionnait à merveille ! Le lendemain, notre codeur arriva tout fier pour sa présentation, avec son projet sur une clé usb. Il le transfère sur l'ordinateur de son pote et là, ça ne fonctionne pas !

Quel est le problème ?

L'application de notre jeune développeur ne fonctionne pas sur l'ordinateur de son ami à cause d'un problème d'environnement. Entre deux systèmes, il peut y avoir des différences de version sur les dépendances ou encore des bibliothèques manquantes.

1.1.5 Livre blanc Ubuntu

`ubuntu/WP_The_no-nonsense-way-to-accelerate-your-business-with_containers.pdf`

1.2 Définitions concernant l'agilité et le mouvement Devops

1.2.1 Définition de Devops p.34 Programmez! p.214 janvier 2018

See also:

- https://en.wikipedia.org/wiki/Devops
- http://david.monniaux.free.fr/dotclear/index.php/post/2018/01/05/Pourquoi-l-informatique-devient-incompr%C3%A9hensible%2C-et-l-impact-sur-la-s%C3%A9curit%C3%A9

Si le mouvement **Devops** fait bien référence à l'automatisation des tests unitaires ou fonctionnels avec la mise en place de l'intégration continue ou à l'automatisation, ce n'est pas l'aspect principal qu'évoque le mouvement **Devops**.

Le **Devops** est un mouvement qui privilégie la mise en place d'un alignement de l'ensemble de la DSI autour **d'objectifs communs**; le terme **Devops** est la concaténation de dev pour développeur et ops pour opérationnels, soit les ingénieurs responsables des infrastructures.

Avoir une équipe enfermée dans une pièce totalement isolée des équipes de développement pour mettre en place des solutions d'intégration continue ou de livraison continue ne correspond pas à ce concept **Devops**. C'est pourtant cette façon de faire que nous voyons de plus en plus aujourd'hui.

1.2.2 Définition 2, Le Devops pour répondre à l'appel de l'innovation 2018-01-04

See also:

- https://www.programmez.com/avis-experts/le-**Devops**-pour-repondre-lappel-de-linnovation-26954

Le **Devops** est axé sur la collaboration, nécessaire pour développer, tester et déployer des applications rapidement et régulièrement.

C'est un changement culturel, qui met l'accent sur le renforcement de la communication et de la collaboration entre différentes équipes, telles que celles chargées du développement, de l'exploitation et de l'assurance-qualité.

L'objectif est de décloisonner les services qui composent une organisation afin de créer un lieu de travail plus collaboratif et de créer ainsi une synergie qui, en bout de chaîne, profite à l'utilisateur final. Car c'est un fait avéré, la création et la conservation de relations solides avec les clients offrent des avantages exponentiels, dont une diminution de la perte de clientèle et des sources de revenus potentiellement plus nombreuses.

Car le **Devops** est avant tout un concept, il n'existe par UN outil de **Devops** à proprement parler, mais un faisceau d'outils permettant d'instaurer et d'entretenir une culture **Devops**. Il regroupe à la fois des outils open source et propriétaires dédiés à des tâches spécifiques dans les processus de développement et de déploiement.

D'ailleurs, en parlant de processus, évoquons un instant le déploiement continu.

Le déploiement continu repose entièrement sur des processus, et l'automatisation y joue un rôle clé. Les processus de déploiement continu sont l'un des éléments fondamentaux d'une transformation **Devops**. Le déploiement continu et le **Devops** permettent aux équipes de développement d'accélérer considérablement la livraison de logiciels. Grâce aux processus de déploiement continu et à la culture **Devops**, les équipes peuvent offrir en permanence du code sûr, testé et prêt à être utilisé en production. Cela inclut la publication de mises à jour logicielles, ce qui, dans une entreprise de télécommunication, peut parfois survenir trois fois par jour, voire plus.

1.2.3 Définition 3, extrait p.53 MISC N95, Janvier/février, 2018, "Ne pas prévoir, c'est déjà gémir"

L'ère des hyperviseurs est-elle révolue ? La bataille commerciale autour de la sécurité et de la performance persiste-t-elle ?

C'est à présent un conflit dépassé, car la sécurité est prise en compte désormais dans les conteneurs au niveau des prérequis.

L'importance du choix de la sécurité réside davantage dans l'édifice construit et son évolution.

Il devient évident que la **virtualisation légère** va gagner du terrain, les hyperviseurs vont alors devenir obsolètes et c'est dans ce contexte qu'il fait repenser l'action des équipes de sécurité.

En faisant avancer les vrais échanges entre Dev et Ops, le **Devops** a changé la donne et la **production bénéficie enfin de l'agilité prônée depuis quelques années**.

En intégrant la sécurité dans le **SecDevops**, et en s'assurant d'avoir des composants sécurisés au maximum, l'aspect sécuritaire devient alors une composante à valeur ajoutée pour la production.

Certains pensent qu'utiliser les systèmes qui ont fait leur preuve dans le temps serait le gage d'une sécurité beaucoup plus fiable et plus simple à mettre en œuvre.

Il semble aujourd'hui de plus en plus évident pour un responsable de systèmes d'information que manquer ce tournant de la technologie des conteneurs, serait une **assurance d'être rapidement mis à l'écart des évolutions en cours**.

1.2.3.1 Citations

1.2.3.1.1 Ne pas prévoir, c'est déjà gémir

"Ne pas prévoir, c'est déjà gémir" Léonard de vinci.

1.2.3.1.2 La vie, c'est comme une bicyclette, il faut avancer pour ne pas perdre l'équilibre

La vie, c'est comme une bicyclette, il faut avancer pour ne pas perdre l'équilibre Albert Einstein

1.2.4 Devops, intégration et déploiement continus, pourquoi est-ce capital et comment y aller ?

See also:

- https://actu.alfa-safety.fr/devops/devops-integration-et-deploiement-continus-pourquoi-est-ce-capital-et-comment-y-aller/

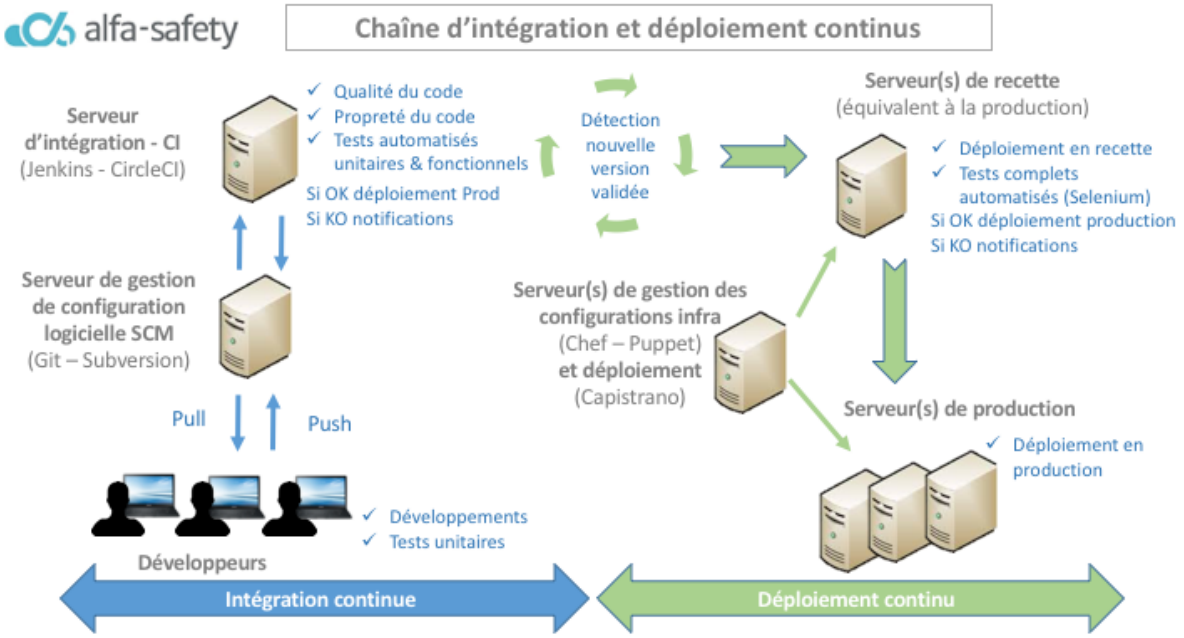


Fig. 2: Intégration continue

« Intégration continue » (CI), « déploiement continu » (CD), « Devops », autant de termes que l'on entend très fréquemment dès que l'on parle d'applications Web et de transformation numérique, et pourtant ce sont des concepts encore mal connus dans leur mise en œuvre.

De quoi s'agit-il ? Tout simplement d'assurer la sortie de nouvelles fonctionnalités d'une application sur un rythme beaucoup plus régulier et rapide.

Traditionnellement, un rythme de déploiement standard sur une application classique est d'une à deux versions majeures par an. Pour chaque version majeure, on regroupe un ensemble de nouvelles fonctionnalités, ce qui donne délai de 6 à 12 mois entre deux nouveautés.

Entretemps, on se contente de corriger les bugs, de sortir des versions mineures. C'est terriblement long, surtout à l'ère d'internet. L'objectif est d'assurer la cohérence des évolutions, regrouper les testss, sécuriser la production et limiter les migrations pour les clients, mais cela pénalise les délais.

Ce délai s'explique par le fait que c'est un processus séquentiel, impliquant différentes équipes et qu'à chaque étape, il faut synchroniser les acteurs, faire des demandes, les planifier, tout cela générant des délais.

Le déploiement continu prend le contrepied et permet d'permet d'accélérer ce rythme en :

- découpant les versions en un plus grand nombre de livraisons de moindre taille et moins complexes à tester,
- automatisant au maximum les étapes de tests et passages en production d'une nouvelle version afin de réduire les cycles,
- permettant un déploiement très régulier des nouveautés.

1.2.5 Agilité et Devops: Extrait p. 35 de [Programmez!] , N°214, janvier 2018

See also:

- https://www.programmez.com/magazine/article/agilite-developpeurs/**Devops**-une-bonne-idee

Les développeurs **doivent** évoluer pour suivre ces deux mouvements populaires (Agilité + **Devops**) qui se déploient très rapidement au sein de l'ensemble des DSI françaises. L'agilité et le **Devops** sont de très bonnes évolutions tant elles apportent aux DSI et au produit final.

1.2.6 What is a DevOps Engineer ?

See also:

http://blog.shippable.com/how-to-be-a-great-devops-engineer

A major part of adopting DevOps is to create a better working relationship between development and operations teams.

Some suggestions to do this include seating the teams together, involving them in each other's processes and workflows, and even creating one cross-functional team that does everything.

In all these methods, Dev is still Dev and Ops is still Ops.

The term DevOps Engineer tries to blur this divide between Dev and Ops altogether and suggests that the best approach is to hire engineers who can be excellent coders as well as handle all the Ops functions.

In short, a DevOps engineer can be a developer who can think with an Operations mindset and has the following skillset:

- Familiarity and experience with a variety of Ops and Automation tools
- Great at writing scripts
- Comfortable with dealing with frequent testing and incremental releases
- Understanding of Ops challenges and how they can be addressed during design and development
- Soft skills for better collaboration across the team

According to Amazon CTO Werner Vogels:

```
Giving developers operational responsibilities has greatly enhanced
the quality of the services, both from a customer and a technology
point of view.
The traditional model is that you take your software to the wall
that separates development and operations, and throw it over and
then forget about it. Not at Amazon. You build it, you run it.
This brings developers into contact with the day-to-day operation
of their software. It also brings them into day-to-day contact with
the customer. This customer feedback loop is essential for improving
the quality of the service.
```

It is easier than ever before for a developer to move to a DevOps role. Software delivery automation is getting better every day and DevOps platforms like Shippable are making it easy to implement automation while also giving you a Single Pane of Glass view across your entire CI/CD pipeline.

Can an Ops engineer move to a DevOps role? Definitely, but it can be a little more challenging since you will need to learn design and programming skills before making that transformation. However, with the upsurge in number of coding bootcamps, it is probably an easier transition to make than it was a few years ago. Ops engineers can bring much needed insights into how software design can cause Ops challenges, so once you get past the initial learning curve for design/coding, you're likely to become a valued DevOps engineer.

1.3 Définitions concernant Docker

See also:

- https://www.docker.com/what-docker

1.3.1 Définition de Docker sur Wikipedia en français

Docker est un logiciel libre qui automatise le déploiement d'applications dans des conteneurs logiciels. Selon la firme de recherche sur l'industrie 451 Research:

```
Docker est un outil qui peut empaqueter une application et ses
dépendances dans un conteneur isolé, qui pourra être exécuté sur
n'importe quel serveur.
```

Ceci permet d'étendre la flexibilité et la portabilité d'exécution d'une application, que ce soit sur la machine locale, un cloud privé ou public, une machine nue, etc

1.3.2 Docker est "agile"

Améliorations des temps de développement et de déplpoiement par 13.

Fig. 3: Source: https://www.docker.com/what-docker

1.3.3 Docker est portable

Docker est portable ce qui permet d'avoir des environnements de développement, test et production pratiquement identiques.

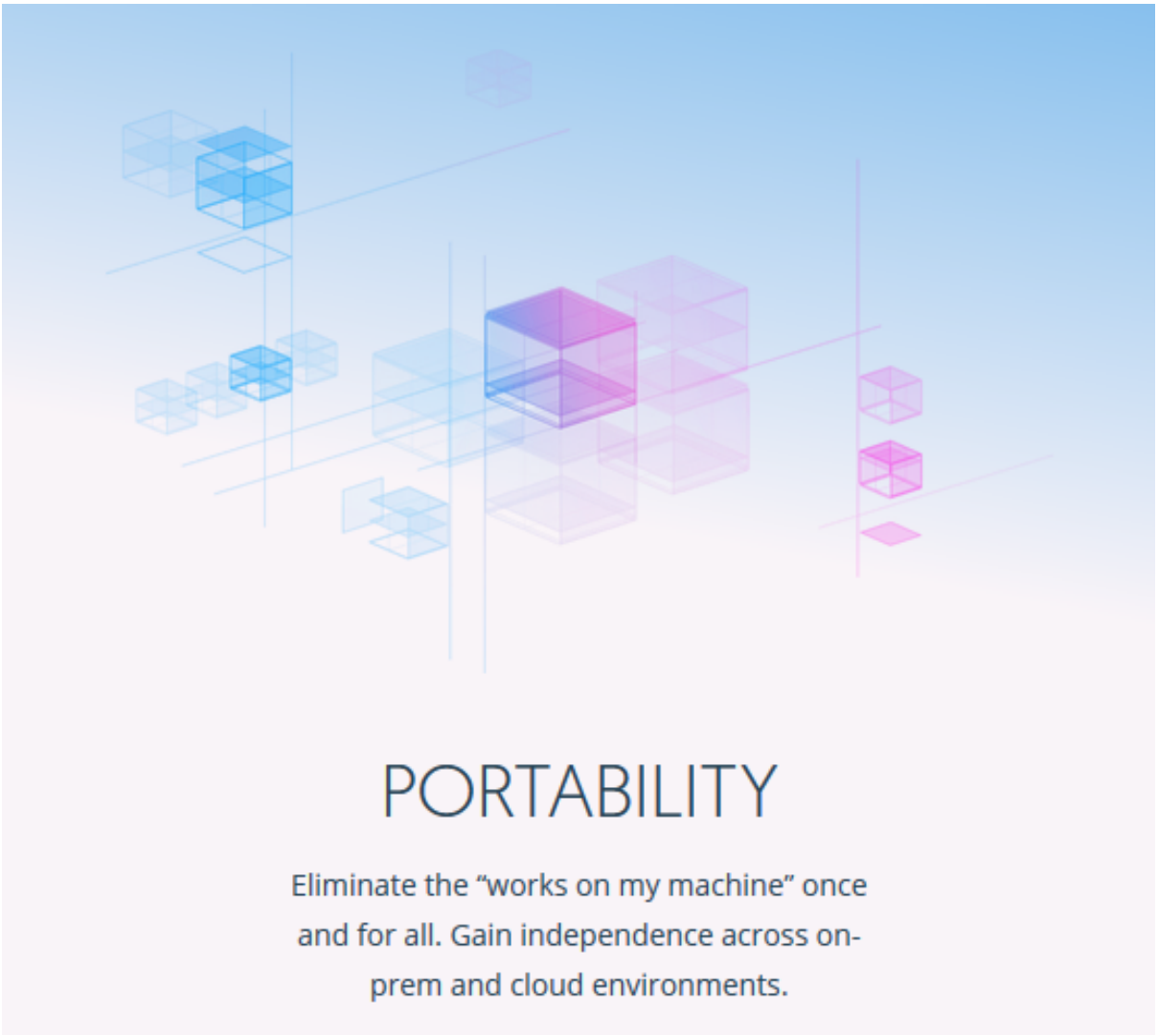


Fig. 4: Source: https://www.docker.com/what-docker

1.3.4 Docker est sécurisé

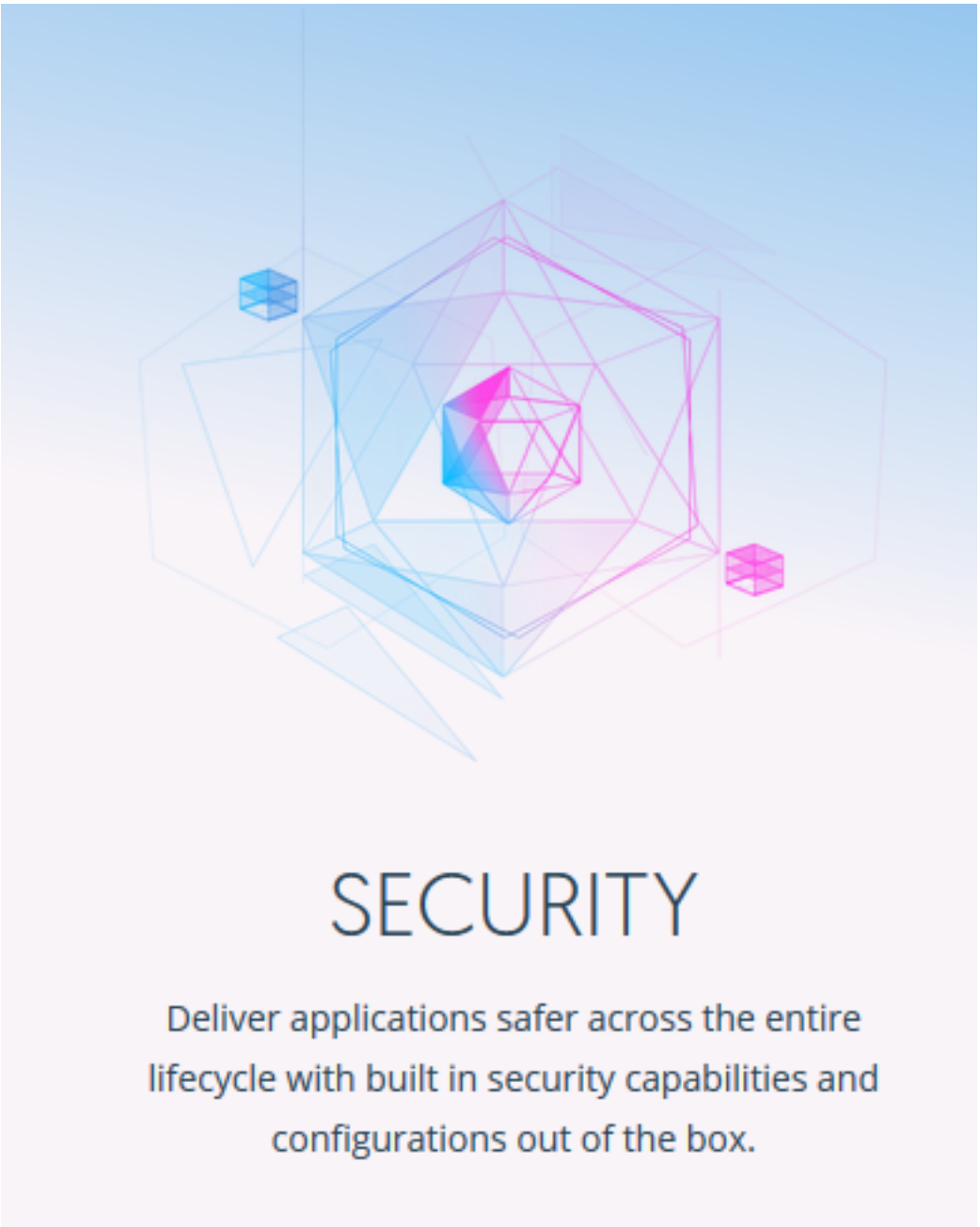


Fig. 5: Source: https://www.docker.com/what-docker

1.3.5 Les conteneurs Docker sont plus légers et rapides que les machines virtuelles

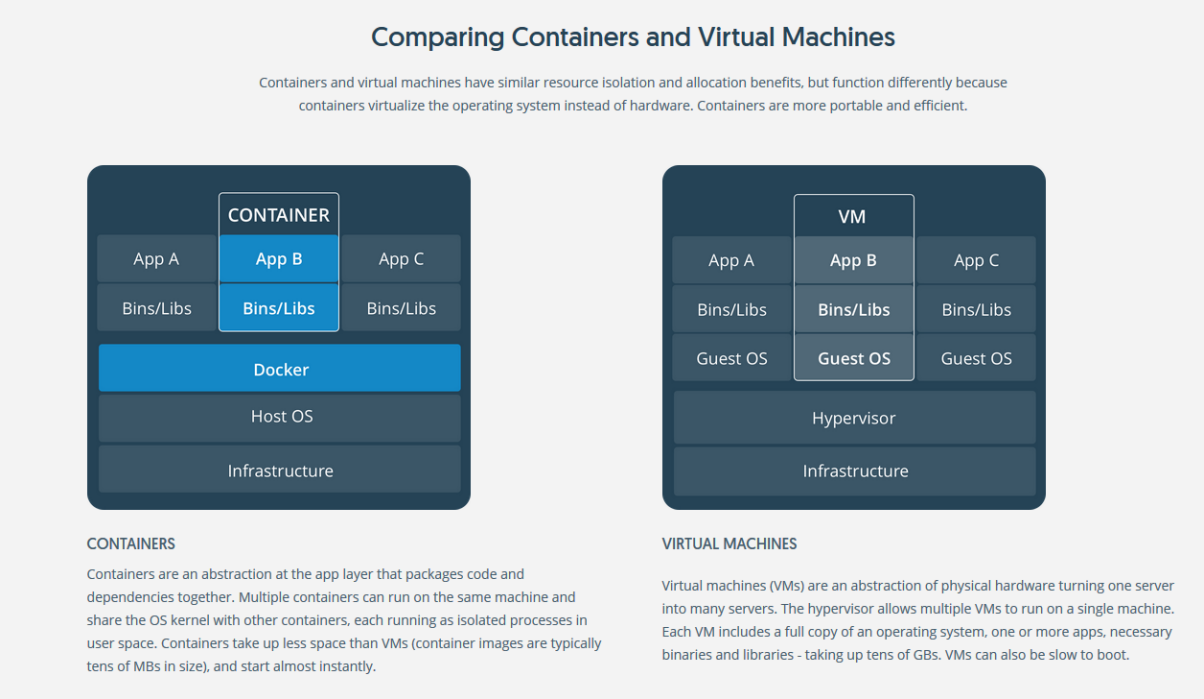


Fig. 6: Source: https://www.docker.com/what-container

1.3.5.1 Containers

Containers are an abstraction at the app layer that packages code and dependencies together. Multiple containers can run on the same machine and share the OS kernel with other containers, each running as isolated processes in user space.

Containers take up less space than VMs (container images are typically tens of MBs in size), and start almost instantly.

1.3.5.2 Virtual machines (VMs)

Virtual machines (VMs) are an abstraction of physical hardware turning one server into many servers.

The hypervisor allows multiple VMs to run on a single machine.

Each VM includes a full copy of an operating system, one or more apps, necessary binaries and libraries - taking up tens of GBs.

VMs can also be slow to boot.

1.3.5.3 Docker can run your applications in production at native speed

Source: p.255 du livre “Python Microservices Development” de Tarek Ziadé.

```
...
that is where VMs are a great solution to run your applications.
...
```

In the past ten years, many software projects that required an elaborate setup to run started to provide read-to-run VMs, using tools such as VMWare or VirtualBox. Those VMs included the whole stack, like prefilled databases. Demos became easyly runnable on most platforms with a single command. That was progress.

However, some of those tools were not fully open source virtualization tool and they were very slow to run, and greedy in memory and CPU and terrible with disk I/O. It was **unthinkable** to run them in production, and they were mostly used for demos.

The **big revolution** came with Docker, an open source virtualization tool, which wa first released in 2013, and became hugley popular. Moreover, unlike VMWare or VirtualBox, Docker can run your applications in production at **native speed**.

1.4 Dossier Docker dans le dossier MISC N°95 de janvier/février 2018

See also:

- https://www.miscmag.com/misc-n95-references-de-larticle-docker-les-bons-reflexes-a-adopter/

Chapter 2

Qui utilise Docker en production ?

See also:

- https://actu.alfa-safety.fr/devops/docker-en-production/

Contents

2.1 Historique

2.1.1 Janvier 2018

See also:

- https://blog.docker.com/2018/01/5-tips-learn-docker-2018/

As the holiday season ends, many of us are making New Year's resolutions for 2018. Now is a great time to think about the new skills or technologies you'd like to learn. So much can change each year as technology progresses and companies are looking to innovate or modernize their legacy applications or infrastructure.

At the same time the market for Docker jobs continues to grow as companies such as Visa, MetLife and Splunk adopt Docker Enterprise Edition (EE) in production.

2.2 Paypal

See also:

- https://www.docker.com/customers/paypal-uses-docker-containerize-existing-apps-save-money-and-boost-security

2.2.1 Challenges

Today PayPal is leveraging OpenStack for their private cloud and runs more than 100,000 VMs. This private cloud runs 100% of their web and mid-tier applications and services. One of the biggest desires of the PayPal business is to modernize their datacenter infrastructure, making it more on demand, improving its security, meeting compliance regulations and lastly, making everything cost efficient.

They wanted to refactor their existing Java and C++ legacy applications by dockerizing them and deploying them as containers.

This called for a technology that provides a distributed application deployment architecture and can manage workloads but must also be deployed in both private, and eventually public cloud environments. Being cost efficient was extremely important for the company. Since PayPal runs their own cloud, they pay close attention to how much money they are spending on actually running their datacenter infrastructure.

Functioning within the online payment industry, PayPal must ensure the security of their internal data (binaries and artifacts with the source code of their applications). This makes them a **very security-conscious company**.

Their sensitive data needs to be kept on-premises where their security teams can run ongoing scans and sign their code before deploying out to production. PayPal's massive popularity is a good thing, but it also means they must handle the deluge of demands from their users. At times they process more than 200 payments per second. When including Braintree and Venmo, the companies that PayPal acquired, that number continues to soar even higher. Recently, it was announced that Braintree is processing more than a billion a month when it comes to mobile payments!. That adds quite a bit of extra pressure on their infrastructure.

2.2.2 Solution

Today PayPal uses Docker's commercial solutions to enable them to not only provide gains for their developers, in terms of productivity and agility, but also for their infrastructure teams in the form of cost efficiency and enterprise-grade security.

The tools being used in production today include:

- Docker Commercially Supported engine (CS Engine),
- Docker Trusted Registry
- as well as Docker Compose.

The company believes that containers and VMs can coexist and combine the two technologies. Leveraging Docker containers and VMs together gives PayPal the ability to run more applications while reducing the number of total VMs, optimizing their infrastructure. This also allows PayPal to spin up a new application much more quickly, and on an "as needed" basis.

Since containers are more lightweight and instantiate in a fraction of a second, while VMs take minutes, they can roll out a new application instance quickly, patch an existing application, or even add capacity for holiday readiness to compensate for peak times within the year.

This helps drive innovation and help them outpace competition. Docker Trusted Registry gives their team enterprise security features like granular role based access controls, and image signing that ensures that all of PayPal's checks and balances are in place.

The tool provides them with the on-premises enterprise-grade registry service they need in order to provide secure collaboration for their image content. There security team can run ongoing scans and sign code before deploying to production.

With Docker, the company has gained the ability to scale quickly, deploy faster, and one day even provide local desktop-based development environments with Docker. For that, they are looking to Docker for Mac and Docker for Windows, which offer Docker as a local development environment to their 4,000+ developers located across the globe.

Chapter 3

Actions/news

3.1 Actions/news 2018

3.1.1 Actions/news mai 2018

3.1.1.1 DjangoCon 2018 - An Intro to Docker for Djangonauts by Lacey Williams

See also:

- https://www.youtube.com/watch?v=v5jfDDg55xs&feature=youtu.be&a=
- *A Brief Intro to Docker for Djangonauts par Lacey Williams*

3.1.1.2 hard-multi-tenancy-in-kubernetes

See also:

- https://blog.jessfraz.com/post/hard-multi-tenancy-in-kubernetes/

3.1.1.3 containers-security-and-echo-chambers

See also:

- https://blog.jessfraz.com/post/containers-security-and-echo-chambers/

3.1.1.4 Aly Sivji, Joe Jasinski, tathagata dasgupta (t) - Docker for Data Science - PyCon 2018

See also:

- https://github.com/docker-for-data-science/docker-for-data-science-tutorial
- https://www.youtube.com/watch?v=jbb1dbFaovg
- https://t.co/ZW7g1JY3va

3.1.1.4.1 Description

Jupyter notebooks simplify the process of developing and sharing Data Science projects across groups and organizations. However, when we want to deploy our work into production, we need to extract the model from the

notebook and package it up with the required artifacts (data, dependencies, configurations, etc) to ensure it works in other environments.

Containerization technologies such as Docker can be used to streamline this workflow.

This hands-on tutorial presents Docker in the context of Reproducible Data Science - from idea to application deployment.

You will get a thorough introduction to the world of containers; learn how to incorporate Docker into various Data Science projects; and walk through the process of building a Machine Learning model in Jupyter and deploying it as a containerized Flask REST API.

3.1.1.5 Créez un cluster hybride ARM/AMD64 (GNU/Linux N°215 mai 2018)

3.1.2 Actions/news avril 2018

Contents

3.1.2.1 Docker for the busy researcher (from Erik Matsen)

See also:

- http://erick.matsen.org/2018/04/19/docker.html

3.1.2.1.1 Why Docker ?

Have you ever been frustrated because a software package's installation instructions were incomplete ? Or have you wanted to try out software without going through a complex installation process? Or have you wanted to execute your software on some remote machine in a defined environment?

Docker can help.

In my group, we use Docker to make sure that our code compiles properly in a defined environment and analyses are reproducible. We automatically create Docker images through Dockerfiles. This provides a clear list of dependencies which are guaranteed to work starting from a defined starting point.

Once a Docker image is built, it can be run anywhere that runs the Docker engine.

3.1.3 Actions/news mars 2018

3.1.3.1 Jeudi 29 mars 2018 : Article de Jérôme Petazzoni : Containers par où commencer ?

See also:

- https://jpetazzo.github.io/2018/03/28/containers-par-ou-commencer/

3.1.4 Actions/news février 2018

3.1.4.1 Mardi 13 février 2018: import d'une nouvelle base de données données db_id3_intranet

Contents

3.1.4.1.1 Suppression du volume djangoid3_intranet_volume (docker volume rm djangoid3_intranet_volume)

```
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\django_id3> docker volume ls
```

```
DRIVER              VOLUME NAME
local               djangoid3_intranet_volume
local               postgresql_volume_intranet
```

```
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\django_id3> docker volume rm djangoid3_intranet_volume
```

```
djangoid3_intranet_volume
```

3.1.4.1.2 Import de la nouvelle base de données (docker-compose -f docker-compose_for_existing_database.yml up –build)

```
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\django_id3> docker-compose -f docker-compose_for_existing_
↪database.yml up --build
```

```
    WARNING: The Docker Engine you're using is running in swarm mode.

    Compose does not use swarm mode to deploy services to multiple nodes in a␣
↪swarm. All containers will be scheduled on the current node.

    To deploy your application across the swarm, use `docker stack deploy`.

    Creating network "djangoid3_default" with the default driver
    Creating volume "djangoid3_intranet_volume" with default driver
    Building db
    Step 1/3 : FROM postgres:10.2
     ---> 6e3b6a866c37
    Step 2/3 : RUN localedef -i fr_FR -c -f UTF-8 -A /usr/share/locale/locale.
↪alias fr_FR.UTF-8
     ---> Using cache
     ---> 65da73d90928
    Step 3/3 : ENV LANG fr_FR.utf8
     ---> Using cache
     ---> a932c8fcf807
    Successfully built a932c8fcf807
    Successfully tagged djangoid3_db:latest
```

(continues on next page)

(continued from previous page)

```
    Creating container_database ... done
    Attaching to container_database
    container_database | Les fichiers de ce cluster appartiendront ├á l
↪'utilisateur ½ postgres .
    container_database | Le processus serveur doit également lui appartenir.
    container_database |
    container_database | L'instance sera initialisée avec la locale ½ fr_FR.utf8 .
    container_database | L'encodage par défaut des bases de données a été␣
↪configuré en conséquence
    container_database | avec ½ UTF8 .
    container_database | La configuration de la recherche plein texte a été␣
↪initialisée ├á ½ french .
    container_database |
    container_database | Les sommes de contr├┤les des pages de données sont␣
↪désactivées.
    container_database |
    container_database | correction des droits sur le répertoire existant /var/lib/
↪postgresql/data... ok
    container_database | création des sous-répertoires... ok
    container_database | sélection de la valeur par défaut de max_connections...␣
↪100
    container_database | sélection de la valeur par défaut pour shared_buffers...␣
↪128MB
    container_database | sélection de l'implémentation de la mémoire partagée␣
↪dynamique...posix
    container_database | création des fichiers de configuration... ok
    container_database | lancement du script bootstrap...ok
    container_database | exécution de l'initialisation après bootstrap...ok
    container_database | synchronisation des données sur disqueok
    container_database |
    container_database | ATTENTION : active l'authentification ½ trust  pour les␣
↪connexions
    container_database | locales.
    container_database | Vous pouvez changer cette configuration en éditant le␣
↪fichier pg_hba.conf
    container_database | ou en utilisant l'option -A, ou --auth-local et --auth-
↪host au prochain
    container_database | lancement d'initdb.
    container_database |
    container_database | Succès. Vous pouvez maintenant lancer le serveur de bases␣
↪de données en utilisant :
    container_database |
    container_database |     pg_ctl -D /var/lib/postgresql/data -l fichier de␣
↪trace start
    container_database |
    container_database | ****************************************************
    container_database | WARNING: No password has been set for the database.
    container_database |          This will allow anyone with access to the
    container_database |          Postgres port to access your database. In
    container_database |          Docker's default configuration, this is
    container_database |          effectively any other container on the same
    container_database |          system.
    container_database |
    container_database |          Use "-e POSTGRES_PASSWORD=password" to set
    container_database |          it in "docker run".
    container_database | ****************************************************
    container_database | en attente du démarrage du serveur....2018-02-14 12:52:43.
↪323 UTC [38] LOG:  en écoute sur IPv4, adresse ½ 127.0.0.1 , port 5432
    container_database | 2018-02-14 12:52:43.342 UTC [38] LOG:  n'a pas pu lier␣
↪IPv6 ├á l'adresse ½ ::1  : Ne peut attribuer l'adresse demandée
    container_database | 2018-02-14 12:52:43.342 UTC [38] ASTUCE :  Un autre␣
↪postmaster fonctionne-t'il déj├á sur le port 5432 ?
```

(continues on next page)

(continued from previous page)

```
    container_database |    Sinon, attendez quelques secondes et réessayez.
    container_database | 2018-02-14 12:52:43.508 UTC [38] LOG:  écoute sur la␣
↪socket Unix ½ /var/run/postgresql/.s.PGSQL.5432
    container_database | 2018-02-14 12:52:43.693 UTC [39] LOG:  le système de␣
↪bases de données a été arr├¬té ├á 2018-02-14 12:52:40 UTC
    container_database | 2018-02-14 12:52:43.791 UTC [38] LOG:  le système de␣
↪bases de données est pr├¬t pour accepter les connexions
    container_database |  effectué
    container_database | serveur démarré
    container_database | ALTER ROLE
    container_database |
    container_database |
    container_database | /usr/local/bin/docker-entrypoint.sh: running /docker-
↪entrypoint-initdb.d/dump_id3_intranet.sql
    container_database | CREATE ROLE
    container_database | SET
    container_database | SET
    container_database | SET

...
    container_database | ALTER TABLE
    container_database | ALTER TABLE
    container_database | ALTER TABLE
    container_database | GRANT
    container_database |
    container_database |
    container_database | en attente de l'arr├¬t du serveur....2018-02-14 12:53:39.
↪199 UTC [38] LOG:  a re├°u une demande d'arr├¬t rapide
    container_database | 2018-02-14 12:53:39.297 UTC [38] LOG:  annulation des␣
↪transactions actives
    container_database | 2018-02-14 12:53:39.302 UTC [38] LOG:  processus de␣
↪travail: logical replication launcher (PID 45) quitte avec le code de sortie 1
    container_database | 2018-02-14 12:53:39.304 UTC [40] LOG:  arr├¬t en cours
    container_database | .......2018-02-14 12:53:46.826 UTC [38] LOG:  le syst├¿me␣
↪de base de donn├®es est arr├¬t├®
    container_database |  effectu├®
    container_database | serveur arr├¬t├®
    container_database |
    container_database | PostgreSQL init process complete; ready for start up.
    container_database |
    container_database | 2018-02-14 12:53:47.027 UTC [1] LOG:  en ├®coute sur IPv4,
↪ adresse ½ 0.0.0.0 , port 5432
    container_database | 2018-02-14 12:53:47.027 UTC [1] LOG:  en ├®coute sur IPv6,
↪ adresse ½ :: , port 5432
    container_database | 2018-02-14 12:53:47.252 UTC [1] LOG:  ├®coute sur la␣
↪socket Unix ½ /var/run/postgresql/.s.PGSQL.5432
    container_database | 2018-02-14 12:53:47.522 UTC [68] LOG:  le syst├¿me de␣
↪bases de donn├®es a ├®t├® arr├¬t├® ├á 2018-02-14 12:53:46 UTC
    container_database | 2018-02-14 12:53:47.648 UTC [1] LOG:  le syst├¿me de␣
↪bases de donn├®es est pr├¬t pour accepter les connexions
```

3.1.4.1.3 Accès à la nouvelle base de données (docker-compose exec db bash)

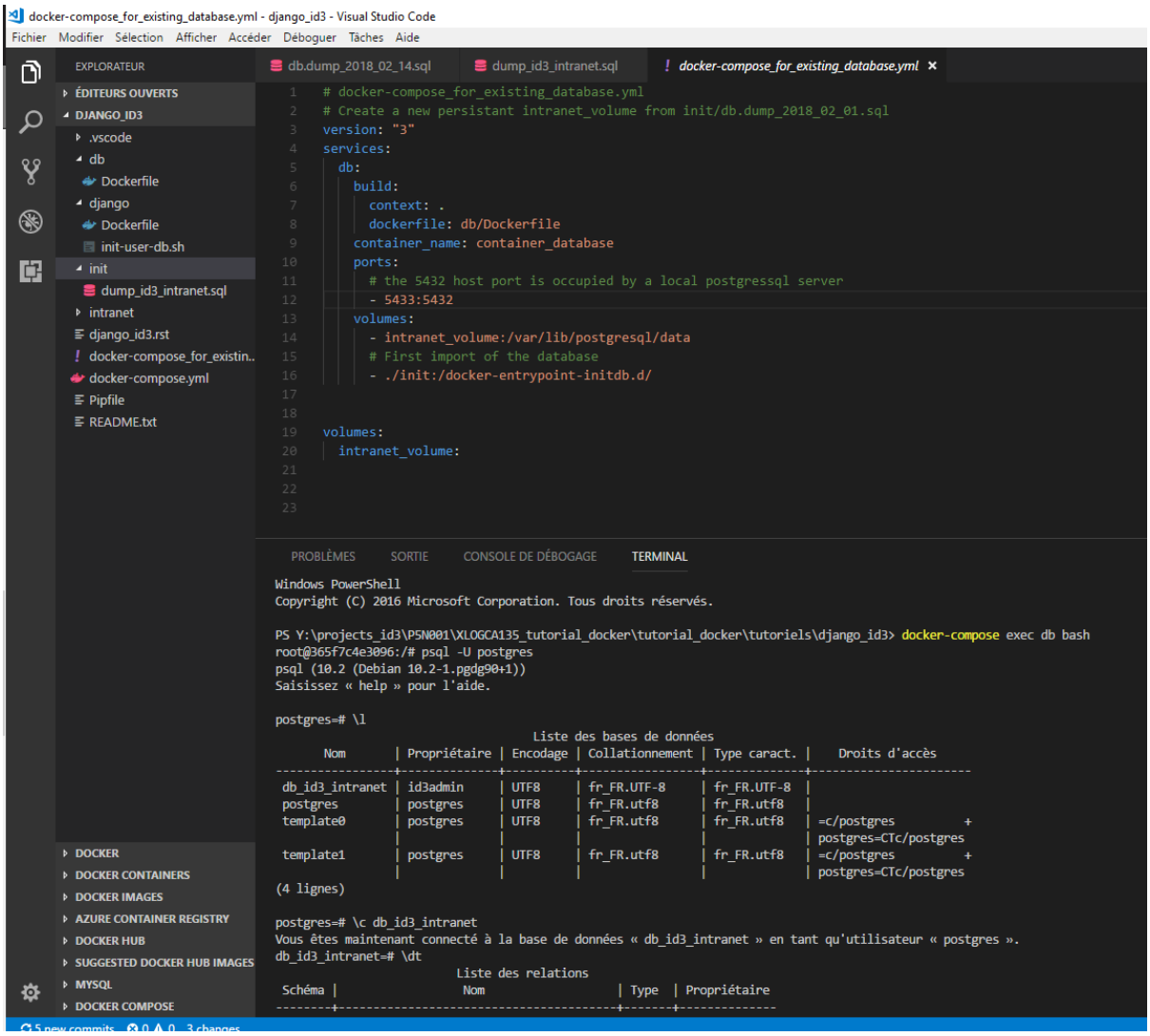


Fig. 1: Accès à la base de données mise à jour avec les données de sybase

```
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\django_id3> docker-compose exec db bash
```

```
root@365f7c4e3096:/# psql -U postgres
```

```
psql (10.2 (Debian 10.2-1.pgdg90+1))
Saisissez « help » pour l'aide.
```

```
postgres=# \l
```

```
                                                                  Liste des␣
↪bases de données
          Nom        | Propriétaire | Encodage | Collationnement | Type caract. | ␣
↪  Droits d'accès
-----------------+--------------+----------+-----------------+--------------+------
↪-----------------
 db_id3_intranet | id3admin      | UTF8     | fr_FR.UTF-8     | fr_FR.UTF-8  |
```

(continues on next page)

(continued from previous page)

```
 postgres         | postgres      | UTF8      | fr_FR.utf8       | fr_FR.utf8    |
 template0        | postgres      | UTF8      | fr_FR.utf8       | fr_FR.utf8    | =c/
↪postgres          +
                                  |               |           |                   |
↪          | postgres=CTc/postgres
 template1        | postgres      | UTF8      | fr_FR.utf8       | fr_FR.utf8    | =c/
↪postgres          +
                                  |               |           |                   |
↪          | postgres=CTc/postgres
(4 lignes)

postgres=# \c db_id3_intranet
```

```
Vous êtes maintenant connecté à la base de données « db_id3_intranet » en tant qu
↪'utilisateur « postgres ».
db_id3_intranet=# \dt
```

3.1.4.1.4 Arrêt du service (docker-compose -f .docker-compose_for_existing_database.yml down)

```
docker-compose -f .\docker-compose_for_existing_database.yml down
```

3.1.4.2 Mardi 13 février 2018: mise en place d'une base de données PostgreSQL 10.2 avec import de la base de données db_id3_intranet

Contents

3.1.4.2.1 docker-compose_for_existing_database.yml

La ligne très importante qu'il fallait trouver est la ligne:

```
- ./init:/docker-entrypoint-initdb.d/
```

```
# docker-compose_for_existing_database.yml
# Create a new persistant intranet_volume from init/db.dump_2018_02_01.sql
version: "3"
services:
  db:
        build:
          context: .
          dockerfile: db/Dockerfile
        container_name: container_database
        ports:
```

(continues on next page)

(continued from previous page)

```
          # the 5432 host port is occupied by a local postgressql server
          - 5433:5432
        volumes:
          - intranet_volume:/var/lib/postgresql/data
          # First import of the database
          - ./init:/docker-entrypoint-initdb.d/


volumes:
  intranet_volume:
```

3.1.4.2.2 Contenu du répertoire init

```
Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        13/02/2018     11:05       34177687 db.dump_2018_02_01.sql
```

L'entête du fichier SQL étant:

```
--
-- PostgreSQL database dump
--

-- Dumped from database version 10.1
-- Dumped by pg_dump version 10.1

SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SET check_function_bodies = false;
SET client_min_messages = warning;
SET row_security = off;

--
-- Name: db_id3_intranet; Type: DATABASE; Schema: -; Owner: id3admin
--

CREATE DATABASE db_id3_intranet WITH TEMPLATE = template0 ENCODING = 'UTF8' LC_
↪COLLATE = 'fr_FR.UTF-8' LC_CTYPE = 'fr_FR.UTF-8';

CREATE USER id3admin WITH
  LOGIN
  NOSUPERUSER
  INHERIT
  NOCREATEDB
  NOCREATEROLE
  NOREPLICATION
  password 'id338';

ALTER DATABASE db_id3_intranet OWNER TO id3admin;

\connect db_id3_intranet

SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET client_encoding = 'UTF8';
```

(continues on next page)

(continued from previous page)

```
SET standard_conforming_strings = on;
SET check_function_bodies = false;
SET client_min_messages = warning;
SET row_security = off;

--
-- Name: db_id3_intranet; Type: COMMENT; Schema: -; Owner: id3admin
--

COMMENT ON DATABASE db_id3_intranet IS 'La base db_id3_intranet';
```

3.1.4.2.2.1 Création de la base db_id3_intranet

```
CREATE DATABASE db_id3_intranet WITH TEMPLATE = template0 ENCODING = 'UTF8' LC_
↪COLLATE = 'fr_FR.UTF-8' LC_CTYPE = 'fr_FR.UTF-8';
```

3.1.4.2.2.2 Création de l'utilisateur id3admin

```
CREATE USER id3admin WITH
  LOGIN
  NOSUPERUSER
  INHERIT
  NOCREATEDB
  NOCREATEROLE
  NOREPLICATION
  password 'id338';
```

3.1.4.3 Lundi 12 février 2018: mise en place d'une base de données PostgreSQL 10.2

Contents

3.1.4.3.1 Dockerfile

```
# https://store.docker.com/images/postgres
FROM postgres:10.2
# avec cette image on peut mettre en place la locale fr_FR.utf8
RUN localedef -i fr_FR -c -f UTF-8 -A /usr/share/locale/locale.alias fr_FR.UTF-8
ENV LANG fr_FR.utf8
```

3.1.4.3.2 docker-compose.yml

```
version: "3"
services:
  db:
        build:
          context: .
          dockerfile: Dockerfile
        ports:
          # the 5432 host port is occupied by a local postgressql server
          - 5433:5432
        volumes:
          - volume_intranet:/var/lib/postgresql/data/

volumes:
  volume_intranet:
```

3.1.4.3.3 Accès HeidiSQL à partir de la machine hôte

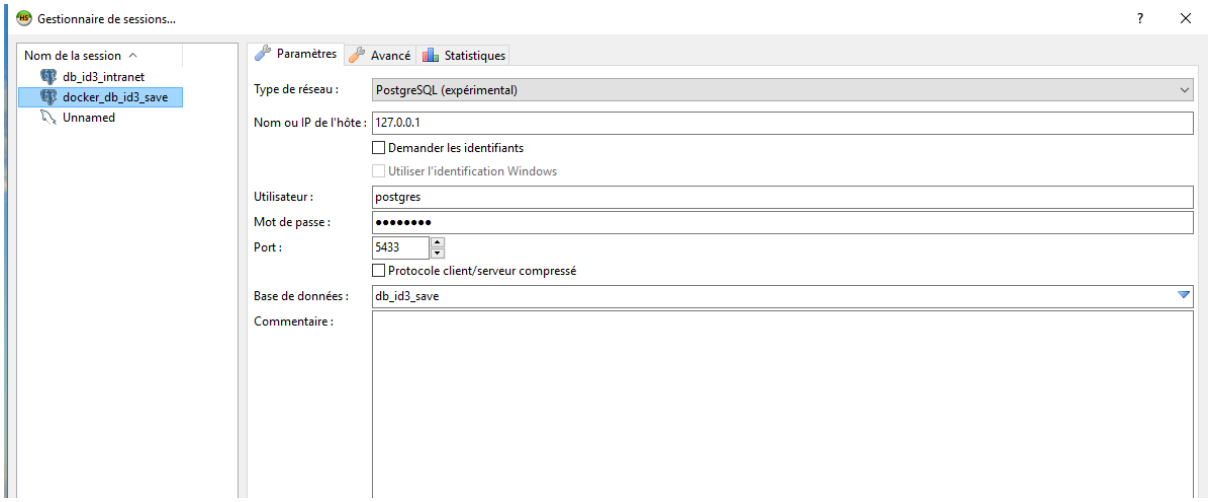


Fig. 2: Accès HeidiSQL à partir de la machine hôte sur le port 5433

3.1.5 Actions/news janvier 2018

3.1.5.1 Mercredi 31 janvier 2018 : export/import d'une base de données PostgreSQL (tutoriel PostgreSQL)

See also:

- *Mercredi 31 janvier 2018 : export/import d'une base de données PostgreSQL (tutoriel PostgreSQL)*

3.1.5.1.1 Dockerfile

```
FROM postgres:10.1
RUN localedef -i fr_FR -c -f UTF-8 -A /usr/share/locale/locale.alias fr_FR.UTF-8
ENV LANG fr_FR.utf8
```

3.1.5.1.2 docker-compose.yml

```
version: "3"
services:
  db:
        build:
          context: .
          dockerfile: Dockerfile
        image: postgres:10.1
        container_name: container_intranet
        volumes:
          - volume_intranet:/var/lib/postgresql/data/
          - .:/code

volumes:
  volume_intranet:
```

3.1.5.1.3 Export

- pg_dump -U postgres –clean –create -f db.dump.sql db_id3_intranet

3.1.5.1.4 Import

- psql -U postgres -f db.dump.sql

3.1.5.1.5 Commandes docker-compose

- docker-compose up
- docker-compose down
- docker-compose exec db bash

3.1.5.2 Mercredi 31 janvier 2018 : Bilan mardi 30 janvier 2018

See also:

- *Tutoriel Docker et Postgresql*
- *Mardi 30 janvier 2018 : écriture des fichiers Dockerfile et docker-compose.yml*
- *Images PostgreSQL*

Contents

- *Pour accéder au conteneur*
  - *docker ps*
  - *docker exec -ti caa4db30ee94 bash*
- *Livre* PostgreSQL : Administration et exploitation de vos bases de données

3.1.5.2.1 Suppression de la base db_id3_intranet

3.1.5.2.1.1 psql -U postgres

```
root@caa4db30ee94:/# psql -U postgres
```

```
psql (10.1)
Type "help" for help.
```

3.1.5.2.1.2 l

```
postgres=# \l
```

```
                                                                 List of␣
↪databases
          Name       |  Owner   | Encoding |  Collate   |   Ctype    |   Access␣
↪privileges
-----------------+----------+----------+------------+------------+-----------------
↪------
 db_id3_intranet | id3admin | UTF8     | en_US.utf8 | en_US.utf8 |
 postgres        | postgres | UTF8     | en_US.utf8 | en_US.utf8 |
 template0       | postgres | UTF8     | en_US.utf8 | en_US.utf8 | =c/postgres     ␣
↪     +
                                 |          |          |            |            |␣
↪postgres=CTc/postgres
 template1       | postgres | UTF8     | en_US.utf8 | en_US.utf8 | =c/postgres     ␣
↪     +
                                 |          |          |            |            |␣
↪postgres=CTc/postgres
(4 rows)
```

3.1.5.2.1.3 drop database db_id3_intranet;

```
postgres=# drop database db_id3_intranet;
```

```
DROP DATABASE
```

3.1.5.2.2 Bilan mardi 30 janvier 2018

Pour pouvoir importer une base données PostgreSQL, il faut utiliser cette suite de commandes dans le fichier docker-compose.yml.

```
version: "3"
```

(continues on next page)

(continued from previous page)

```
services:
  db:
        image: postgres:10.1
        container_name: container_intranet
        volumes:
          - volume_intranet:/var/lib/postgresql/data/
          - .:/code

volumes:
  volume_intranet:
```

La commande **.:/code** permet de voir ce qu'il y a dans le répertoire du coté *host*.

```
root@caa4db30ee94:/# ls -als code
```

```
total 33897
4 drwxr-xr-x 2 root root     4096 Jan 31 08:24 .
4 drwxr-xr-x 1 root root     4096 Jan 30 13:46 ..
33776 -rwxr-xr-x 1 root root 34586512 Jan 25 13:51 db_id3_intranet_2018_01_25.sql
1 -rwxr-xr-x 1 root root      214 Jan 30 13:46 docker-compose.yml
24 -rwxr-xr-x 1 root root    23949 Jan 30 14:04 postgresql.rst
8 -rwxr-xr-x 1 root root     6238 Jan 31 08:24 README.txt
80 -rwxr-xr-x 1 root root    80802 Jan 22 12:03 stack_overflow_postgres.png
```

On voit bien le fichier **db_id3_intranet_2018_01_25.sql**

3.1.5.2.3 Pour lancer PostgreSQL

```
docker-compose up
```

```
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_docker\tutoriels\postgresql> docker-compose up
WARNING: The Docker Engine you're using is running in swarm mode.

Compose does not use swarm mode to deploy services to multiple nodes in a swarm. All containers will be scheduled on the current node.

To deploy your application across the swarm, use `docker stack deploy`.

Starting container_intranet ... done
Attaching to container_intranet
container_intranet | 2018-01-31 07:46:47.402 UTC [1] LOG:  listening on IPv4 address "0.0.0.0", port 5432
container_intranet | 2018-01-31 07:46:47.402 UTC [1] LOG:  listening on IPv6 address "::", port 5432
container_intranet | 2018-01-31 07:46:47.556 UTC [1] LOG:  listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
container_intranet | 2018-01-31 07:46:47.786 UTC [20] LOG:  database system was interrupted; last known up at 2018-01-30 14:43:18 UTC
container_intranet | 2018-01-31 07:46:49.498 UTC [20] LOG:  database system was not properly shut down; automatic recovery in progress
container_intranet | 2018-01-31 07:46:49.688 UTC [20] LOG:  redo starts at 0/167C828
container_intranet | 2018-01-31 07:46:49.688 UTC [20] LOG:  invalid record length at 0/167C908: wanted 24, got 0
container_intranet | 2018-01-31 07:46:49.688 UTC [20] LOG:  redo done at 0/167C8D0
container_intranet | 2018-01-31 07:46:50.423 UTC [1] LOG:  database system is ready to accept connections
```

Fig. 3: docker-compose up

3.1.5.2.4 Pour accéder au conteneur

3.1.5.2.4.1 docker ps

```
docker ps
```

```
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\postgresql> docker ps
```

```
CONTAINER ID        IMAGE               COMMAND                  CREATED
↪   STATUS              PORTS               NAMES
caa4db30ee94        postgres:10.1       "docker-entrypoint.s..."   19 hours ago
↪     Up 34 minutes       5432/tcp            container_intranet
```

3.1.5.2.4.2 docker exec -ti caa4db30ee94 bash

```
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\postgresql> docker exec -ti caa4db30ee94 bash
```

```
root@caa4db30ee94:/# ls -als
total 80
4 drwxr-xr-x   1 root root 4096 Jan 30 13:46 .
4 drwxr-xr-x   1 root root 4096 Jan 30 13:46 ..
4 drwxr-xr-x   1 root root 4096 Dec 12 06:04 bin
4 drwxr-xr-x   2 root root 4096 Nov 19 15:25 boot
4 drwxr-xr-x   2 root root 4096 Jan 31 08:22 code
0 drwxr-xr-x   5 root root  340 Jan 31 07:46 dev
4 drwxr-xr-x   2 root root 4096 Dec 12 06:04 docker-entrypoint-initdb.d
0 lrwxrwxrwx   1 root root   34 Dec 12 06:05 docker-entrypoint.sh -> usr/local/bin/
↪docker-entrypoint.sh
0 -rwxr-xr-x   1 root root    0 Jan 30 13:46 .dockerenv
4 drwxr-xr-x   1 root root 4096 Jan 30 13:46 etc
4 drwxr-xr-x   2 root root 4096 Nov 19 15:25 home
4 drwxr-xr-x   1 root root 4096 Dec 10 00:00 lib
4 drwxr-xr-x   2 root root 4096 Dec 10 00:00 lib64
4 drwxr-xr-x   2 root root 4096 Dec 10 00:00 media
4 drwxr-xr-x   2 root root 4096 Dec 10 00:00 mnt
4 drwxr-xr-x   2 root root 4096 Dec 10 00:00 opt
0 dr-xr-xr-x 132 root root    0 Jan 31 07:46 proc
4 drwx------   1 root root 4096 Jan 30 14:32 root
4 drwxr-xr-x   1 root root 4096 Dec 12 06:05 run
4 drwxr-xr-x   1 root root 4096 Dec 12 06:04 sbin
4 drwxr-xr-x   2 root root 4096 Dec 10 00:00 srv
0 dr-xr-xr-x  13 root root    0 Jan 31 07:46 sys
4 drwxrwxrwt   1 root root 4096 Jan 30 13:46 tmp
4 drwxr-xr-x   1 root root 4096 Dec 10 00:00 usr
4 drwxr-xr-x   1 root root 4096 Dec 10 00:00 var
```

3.1.5.2.5 Livre *PostgreSQL : Administration et exploitation de vos bases de données*

De nombreuses informations **très intéressantes**.

- psql -f nom_fichier.sql
- explications sur les bases template0 et template1

3.1.5.3 Mardi 30 janvier 2018 : écriture des fichiers Dockerfile et docker-compose.yml

See also:

- *Tutoriel Docker et Postgresql*
- *Mercredi 31 janvier 2018 : Bilan mardi 30 janvier 2018*

Contents

3.1.5.3.1 Objectifs pour la journée

Mises et point et premières exécutions.

Dans un premier temps on ne prend pas en charge les secrets.

3.1.5.3.2 Avancement, découverte

- je repasse sur le tutoriel *postgresql* pour essayer de comprendre les volumes.

3.1.5.3.3 Historique

- ajout MISC95

```
CREATE DATABASE db_test WITH OWNER = id3admin ENCODING = 'UTF8' CONNECTION LIMIT =␣
↪-1;


C:\Tmp>psql -U postgres < create_database.sql
Mot de passe pour l'utilisateur postgres : id338

CREATE DATABASE
```

3.1.5.4 Lundi 29 janvier 2018 : encore un nouveau tutoriel : A Simple Recipe for Django Development In Docker (Bonus: Testing with Selenium) de Jacob Cook

See also:

- https://medium.com/@adamzeitcode/a-simple-recipe-for-django-development-in-docker-bonus-testing-with-selenium-6a03

3.1.5.4.1 Analyse et plan de travail pour la journée

S'inspirer des 4 tutoriels pour créer les fichiers Dockerfile et Docker-compose.yml

- https://medium.com/@adamzeitcode/a-simple-recipe-for-django-development-in-docker-bonus-testing-with-selenium-6a03
- https://peakwinter.net/blog/modern-devops-django/
- https://www.revsys.com/tidbits/brief-intro-docker-djangonauts/
- https://wsvincent.com/django-docker-postgresql/

3.1.5.4.2 Autre projet intéressant

3.1.5.4.2.1 dockerize-all-the-things

See also:

- https://github.com/DrewDahlman/dockerize-all-the-things

3.1.5.4.2.2 Kill it with fire

- docker rm $(docker ps -a -q) - Kills all containers
- docker rmi $(docker images -q) - will toast ALL of your images

Something to keep in mind is that sometimes docker containers and images can get bloated on your machine and you might have to toast everything.

The great thing about using docker like this is that you can quickly rebuild a project and get right back into working.

Also when you close a console you are not stopping the container, you always need to run docker-compose down when stopping a project, otherwise it will just keep running in the background.

docker rm $(docker ps -a -q)

```
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_docker\actions_
↪news\2018\2018_01\01__2018_01_29> docker rm $(docker ps -a -q)
```

```
367ce1d9818a
c467c2469b34
7fb912b6a3e2
1746a16a91eb
6ee9dc365c9d
8ae3930ee2d6
97592a1a70ea
8ffcde2f70f6
3d1169398f02
e629ebfc3981
ddbe7a8e2502
7c1afd485479
ebe371507dc2
2b8fff5f4068
cb62ace67ba4
685915373a4c
e150d0531321
7d6e93a39de5
807d38ada261
eebf7e801b96
```

```
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_docker\actions_
↪news\2018\2018_01\01__2018_01_29> docker rmi -f $(docker images -q)
```

```
Untagged: saleor_celery:latest
Untagged: saleor_web:latest
Deleted: sha256:fe40ed7484fe2f111dfdc7b8e79d3353534f29cc28c9019a0b0d6fcb2b624ac4
Deleted: sha256:b49f310a4175a6f56d4ad302c60307c989774a7c853a6dd068cbf68fc234926c
Deleted: sha256:5601669ae105acb6a632cd7d3dd473158b25ff6f1c7d65a95b04a2c12bad713d
Deleted: sha256:bf662c677b1ec758f37dac85c90d55c0c005be7f283723f0f85deaf1e0418c1c
Deleted: sha256:08889c646f293f56cf2a4bc2087a7fe3263f745536f9dd6c0d910264b2e10361
Deleted: sha256:64b9f0663d35de8d404374e8574484d60195e55507b3a87897821bc383c1b69d
Deleted: sha256:716475184da4626198a7da0d47d14072b4bb7c96384b1c6e67eb97daecd25b25
Deleted: sha256:9deb54f781dd986aab78aeaebeef6ed8c587837595b02f7fb8b9008eb80006d6
Deleted: sha256:bb6904496c708da82760b2ca6e3f737608180e377ba88129060318a7af311398
Deleted: sha256:bc59713a5080512001bf80eecce306b85096858601f07ce23d8e0a9233ad69d9
```

3.2 Actions/news 2017

3.2.1 Actions/news août 2017

3.2.1.1 4 août 2017 “Docker et Shorewall” par Guillaume Cheramy

See also:

- https://www.guillaume-cheramy.fr/docker-et-shorewall/

- https://twitter.com/cheramy_linux

3.2.1.1.1 Créer les règles shorewall pour Docker

Il faut créer dans shorewall les règles pour que les conteneurs puissent avoir accès à Internet :

Chapter 4

Tutoriels Docker

See also:

- https://docs.docker.com/
- https://github.com/docker/labs/
- https://twitter.com/Docker/lists/docker-captains/members
- https://github.com/jpetazzo/container.training
- https://hackr.io/tutorials/learn-docker

4.1 Les conseils et formations de Jérôme Petazzoni

See also:

- https://jpetazzo.github.io/2018/03/28/containers-par-ou-commencer/
- https://github.com/jpetazzo
- https://github.com/jpetazzo/container.training
- https://training.play-with-docker.com
- http://paris.container.training/intro.html
- http://paris.container.training/kube.html
- https://www.youtube.com/playlist?list=PLBAFXs0YjviLgqTum8MkspG_8VzGl6C07 (Docker)
- https://www.youtube.com/playlist?list=PLBAFXs0YjviLrsyydCzxWrIP_1-wkcSHS (Kubernetes)

4.1.1 Se former, seul ou accompagné

La communauté Docker est extrêmement riche en tutoriels divers pour démarrer et aller plus loin.

Je recommande particulièrement les **labs** disponibles sur training.play-with-docker.com[1]

Si vous préférez être formé en personne, c'est aussi possible !

Publicité bien ordonnée commence par soi-même : en avril, j'organise deux formations à Paris avec Jérémy Garrouste.

- Le 11 et 12 avril, Introduction aux containers : de la pratique aux bonnes pratiques[2].

[1] https://training.play-with-docker.com
[2] http://paris.container.training/intro.html

- Le 13 avril, Introduction à l'orchestration : Kubernetes par l'exemple[3]

La première formation vous permettra d'être à même d'accomplir les deux premières étapes décrites dans le plan exposé plus haut.

La seconde formation vous permettra d'aborder les étapes 3 et 4.

Si vous voulez vous faire une idée de la qualité du contenu de ces formations, vous pouvez consulter des vidéos et slides de formations précédentes, par exemple :

- journée d'introduction à Docker[4]
- demi-journée d'introduction à Kubernetes[5]

Ces vidéos sont en anglais, mais les formations que je vous propose à Paris en avril sont en français (le support de formation, lui, reste en anglais).

Vous pouvez trouver d'autres vidéos, ainsi qu'une collection de supports (slides etc.) sur http://container.training/.

Cela vous permettra de juger au mieux si ces formations sont adaptées à votre besoin !

4.1.2 Jérôme Petazzoni Container training

See also:

- https://github.com/jpetazzo/container.training
- http://container.training/

4.2 Tutoriels Docker pour Windows

See also:

- https://docs.docker.com/docker-for-windows/

Contents

4.2.1 Installation

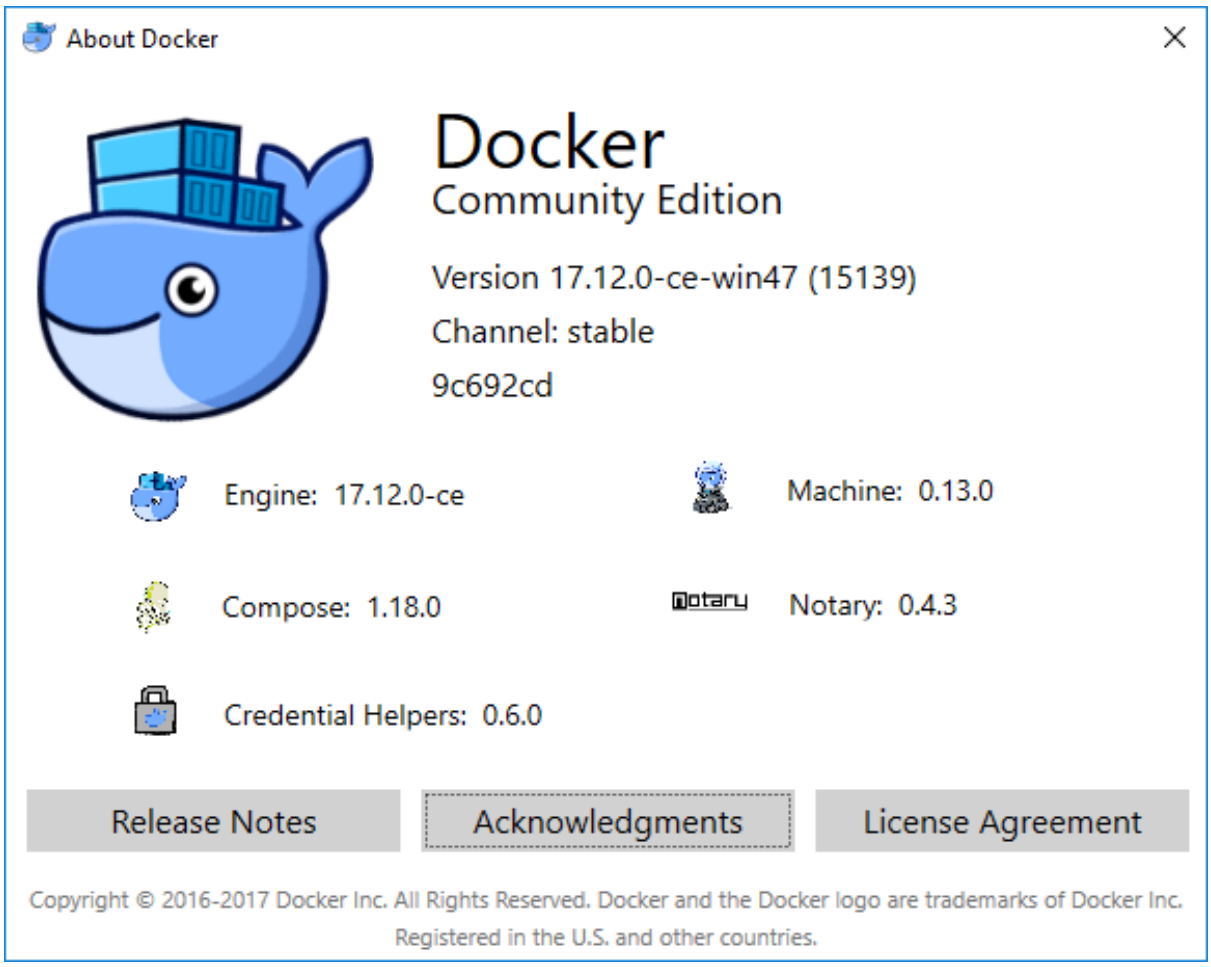
See also:

- https://docs.docker.com/docker-for-windows/install/
- https://download.docker.com/win/stable/Docker%20for%20Windows%20Installer.exe

[3] http://paris.container.training/kube.html
[4] https://www.youtube.com/playlist?list=PLBAFXs0YjviLgqTum8MkspG_8VzGl6C07
[5] https://www.youtube.com/playlist?list=PLBAFXs0YjviLrsyydCzxWrIP_1-wkcSHS

- https://nickjanetakis.com/blog/setting-up-docker-for-windows-and-wsl-to-work-flawlessly

Installation de “Docker for windows” quand on a une machine sous Windows 10.



4.2.2 docker –version

```
docker --version
```

```
Docker version 17.12.0-ce, build c97c6d6
```

4.2.3 docker-compose –version

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_docker>docker-compose --
↪version
```

```
docker-compose version 1.18.0, build 8dd22a96
```

4.2.4 docker-machine –version

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_docker>docker-machine --
↪version
```

```
docker-machine version 0.13.0, build 9ba6da9
```

4.2.5 notary version

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_docker>notary version
```

```
notary
 Version:    0.4.3
 Git commit: 9211198
```

4.2.6 Binaires docker sous Windows 10

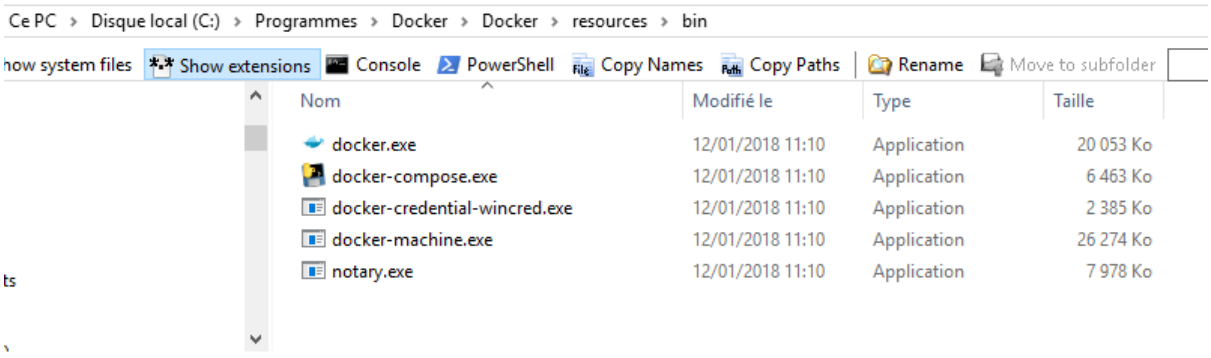


Fig. 1: Binaires docker sous Windows 10

4.2.7 Where to go next

See also:

- https://docs.docker.com/get-started/
- https://github.com/docker/labs/
- https://docs.docker.com/engine/reference/commandline/docker/
- https://blog.docker.com/2017/01/whats-new-in-docker-1-13/
- Try out the walkthrough at Get Started[6].
- Dig in deeper with Docker Labs[7] example walkthroughs and source code.
- For a summary of Docker command line interface (CLI) commands, see the Docker CLI Reference Guide[8].
- Check out the blog post Introducing Docker 1.13.0[9].

4.3 Get started (https://docs.docker.com/get-started/)

See also:

- https://docs.docker.com/get-started/

[6] https://docs.docker.com/get-started/
[7] https://github.com/docker/labs/
[8] https://docs.docker.com/engine/reference/commandline/docker/
[9] https://blog.docker.com/2017/01/whats-new-in-docker-1-13/

- *Tutoriels Docker pour Windows*

Contents

4.3.1 docker run hello-world

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_docker>docker run hello-
↪world
```

```
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.
        (amd64)
 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/
```

4.3.2 docker –version

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_docker>docker --version
```

```
Docker version 17.12.0-ce, build c97c6d6
```

4.3.3 Conclusion

The unit of scale being an individual, portable executable has vast implications.

It means CI/CD can push updates to any part of a distributed application, system dependencies are not an issue, and resource density is increased.

Orchestration of scaling behavior is a matter of spinning up new executables, not new VM hosts.

We'll be learning about all of these things, but first let's learn to walk.

4.3.4 Parts

4.3.4.1 Get started Part2 : Containers

See also:

- https://docs.docker.com/get-started/part2/
- https://hub.docker.com/
- https://hub.docker.com/u/id3pvergain/
- *Tutoriels Docker pour Windows*

Contents

4.3.4.1.1 Prérequis

Ne pas oublier de démarrer le serveur docker.

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_docker\tutoriels\get_
↪started\part2>docker build -t friendlyhello .
```

```
error during connect: Post http://%2F%2F.%2Fpipe%2Fdocker_engine/v1.35/build?
↪buildargs=%7B%7D&cachefrom=%5B%5D&cgroupparent=&cpuperiod=0&cpuquota=0&
↪cpusetcpus=&cpusetmems=&cpushares=0&dockerfile=Dockerfile&labels=%7B%7D&memory=0&
↪memswap=0&networkmode=default&rm=1&
↪session=503be270159342059d8cbfa34d94c9f1e312558a1dcef2ef4369cb0b440ad6a3&
↪shmsize=0&t=friendlyhello&target=&ulimits=null:
open //./pipe/docker_engine: Le fichier spécifié est introuvable.
In the default daemon configuration on Windows, the docker client
must be run elevated to connect.
This error may also indicate that the docker daemon is not running.
```

4.3.4.1.2 Build the app: docker build -t friendlyhello .

:: docker build -t friendlyhello .

```
Sending build context to Docker daemon  7.168kB
Step 1/7 : FROM python:2.7-slim
2.7-slim: Pulling from library/python
```

(continues on next page)

(continued from previous page)

```
c4bb02b17bb4: Pull complete
c5c896dce5ee: Pull complete
cf210b898cc6: Pull complete
5117cef49bdb: Pull complete
Digest: sha256:22112f2295fe9ea84b72e5344af73a2580a47b1014a1f4c58eccf6095b7ea18f
Status: Downloaded newer image for python:2.7-slim
 ---> 4fd30fc83117
Step 2/7 : WORKDIR /app
Removing intermediate container 8ed2ad0d0958
 ---> 7400c8709865
Step 3/7 : ADD . /app
 ---> 728e5124216a
Step 4/7 : RUN pip install --trusted-host pypi.python.org -r requirements.txt
 ---> Running in 847d00a0831e
Collecting Flask (from -r requirements.txt (line 1))
  Downloading Flask-0.12.2-py2.py3-none-any.whl (83kB)
Collecting Redis (from -r requirements.txt (line 2))
  Downloading redis-2.10.6-py2.py3-none-any.whl (64kB)
Collecting itsdangerous>=0.21 (from Flask->-r requirements.txt (line 1))
  Downloading itsdangerous-0.24.tar.gz (46kB)
Collecting Jinja2>=2.4 (from Flask->-r requirements.txt (line 1))
  Downloading Jinja2-2.10-py2.py3-none-any.whl (126kB)
Collecting Werkzeug>=0.7 (from Flask->-r requirements.txt (line 1))
  Downloading Werkzeug-0.14.1-py2.py3-none-any.whl (322kB)
Collecting click>=2.0 (from Flask->-r requirements.txt (line 1))
  Downloading click-6.7-py2.py3-none-any.whl (71kB)
Collecting MarkupSafe>=0.23 (from Jinja2>=2.4->Flask->-r requirements.txt (line 1))
  Downloading MarkupSafe-1.0.tar.gz
Building wheels for collected packages: itsdangerous, MarkupSafe
  Running setup.py bdist_wheel for itsdangerous: started
  Running setup.py bdist_wheel for itsdangerous: finished with status 'done'
  Stored in directory: /root/.cache/pip/wheels/fc/a8/66/
↪24d655233c757e178d45dea2de22a04c6d92766abfb741129a
  Running setup.py bdist_wheel for MarkupSafe: started
  Running setup.py bdist_wheel for MarkupSafe: finished with status 'done'
  Stored in directory: /root/.cache/pip/wheels/88/a7/30/
↪e39a54a87bcbe25308fa3ca64e8ddc75d9b3e5afa21ee32d57
Successfully built itsdangerous MarkupSafe
Installing collected packages: itsdangerous, MarkupSafe, Jinja2, Werkzeug, click,␣
↪Flask, Redis
Successfully installed Flask-0.12.2 Jinja2-2.10 MarkupSafe-1.0 Redis-2.10.6␣
↪Werkzeug-0.14.1 click-6.7 itsdangerous-0.24
Removing intermediate container 847d00a0831e
 ---> 3dc371ea405c
Step 5/7 : EXPOSE 80
 ---> Running in 0f4b33dbfcd0
Removing intermediate container 0f4b33dbfcd0
 ---> d1d59914b22b
Step 6/7 : ENV NAME World
 ---> Running in a742b8e9bddb
Removing intermediate container a742b8e9bddb
 ---> b79587f955c5
Step 7/7 : CMD ["python", "app.py"]
 ---> Running in f9c7ee2841c0
Removing intermediate container f9c7ee2841c0
 ---> ed5b70620e49
Successfully built ed5b70620e49
Successfully tagged friendlyhello: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.
```

(continues on next page)

(continued from previous page)

```
It is recommended to double check and reset permissions for sensitive
files and directories.
```

4.3.4.1.3 docker images

```
docker images
```

```
REPOSITORY                  TAG                 IMAGE ID            CREATED
↪     SIZE
friendlyhello               latest              ed5b70620e49        10 minutes ago
↪     148MB
wordpress                   latest              28084cde273b        6 days ago
↪     408MB
centos                      latest              ff426288ea90        6 days ago
↪     207MB
nginx                       latest              3f8a4339aadd        2 weeks ago
↪     108MB
python                      2.7-slim            4fd30fc83117        4 weeks ago
↪     138MB
hello-world                 latest              f2a91732366c        7 weeks ago
↪     1.85kB
docker4w/nsenter-dockerd    latest              cae870735e91        2 months ago
↪     187kB
```

4.3.4.1.4 Run the app: docker run -p 4000:80 friendlyhello

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_docker\tutoriels\part2>
↪docker run -p 4000:80 friendlyhello
```

```
* Running on http://0.0.0.0:80/ (Press CTRL+C to quit)
```

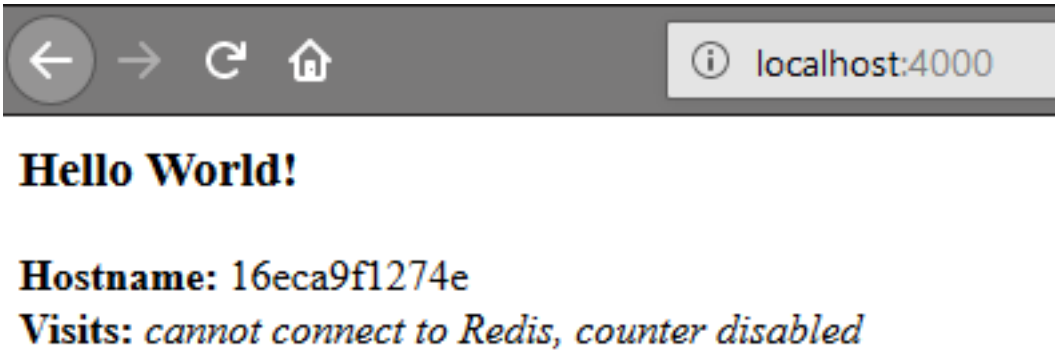


Fig. 2: http://localhost:4000/

4.3.4.1.5 docker container ls

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_docker>docker container
↪ls
```

```
CONTAINER ID        IMAGE               COMMAND             CREATED
↪STATUS              PORTS                NAMES
06193b763075        friendlyhello       "python app.py"     41 minutes ago      Up
↪41 minutes       0.0.0.0:4000->80/tcp   boring_goodall
```

4.3.4.1.6 docker container stop 06193b763075

```
docker container stop 06193b763075
```

```
06193b763075
```

4.3.4.1.7 Tag the image: docker tag friendlyhello id3pvergain/get-started:part2

```
docker tag friendlyhello id3pvergain/get-started:part2
```

4.3.4.1.8 Publish the image

```
docker push id3pvergain/get-started:part2
```

```
The push refers to repository [docker.io/id3pvergain/get-started]
af88fcfe37d7: Pushed
b13ed1abc5b3: Pushed
150ac820623b: Pushed
94b0b6f67798: Mounted from library/python
e0c374004259: Mounted from library/python
56ee7573ea0f: Mounted from library/python
cfce7a8ae632: Mounted from library/python
part2: digest:
↪sha256:1afb795959667db38cc58581d8d455ce10eff78be3cce18560ba887fb6f8c920 size:
↪1788
```

Once complete, the results of this upload are publicly available. If you log in to Docker Hub, you will see the new image there, with its pull command.

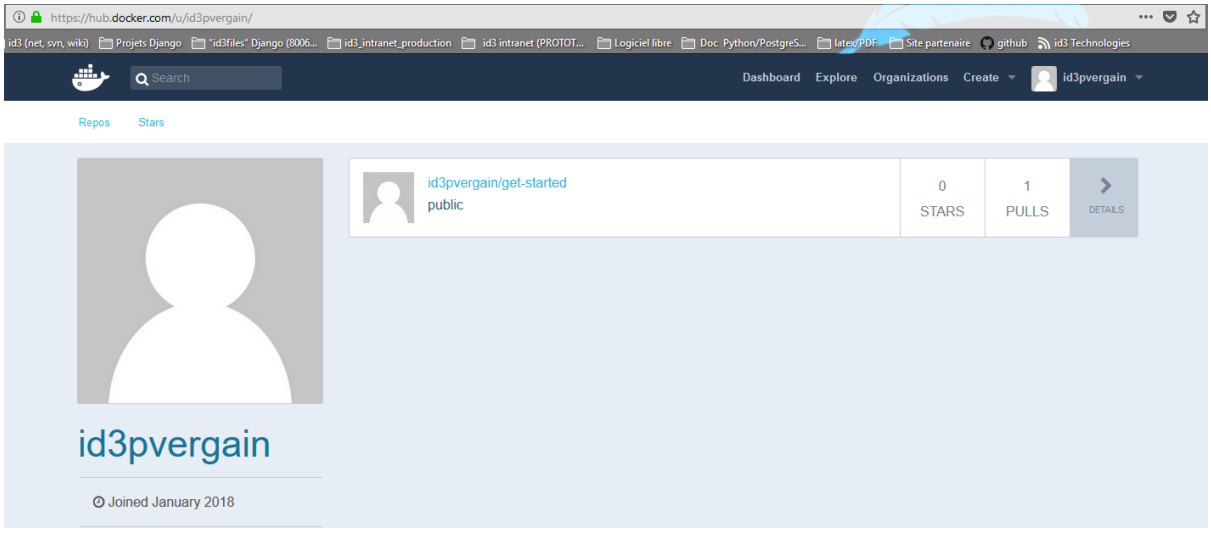


Fig. 3: https://hub.docker.com/u/id3pvergain/

4.3.4.1.9 Pull and run the image from the remote repository

See also:

- https://docs.docker.com/get-started/part2/#pull-and-run-the-image-from-the-remote-repository

From now on, you can use docker run and run your app on any machine with this command:

```
docker run -p 4000:80 id3pvergain/get-started:part2
```

If the image isn't available locally on the machine, Docker will pull it from the repository.

Here is a list of the basic Docker commands from this page, and some related ones if you'd like to explore a bit before moving on.

```
docker build -t friendlyhello .  # Create image using this directory's Dockerfile
docker run -p 4000:80 friendlyhello  # Run "friendlyname" mapping port 4000 to 80
docker run -d -p 4000:80 friendlyhello         # Same thing, but in detached mode
docker container ls                                # List all running containers
docker container ls -a             # List all containers, even those not running
docker container stop <hash>           # Gracefully stop the specified container
docker container kill <hash>         # Force shutdown of the specified container
docker container rm <hash>        # Remove specified container from this machine
docker container rm $(docker container ls -a -q)         # Remove all containers
docker image ls -a                             # List all images on this machine
docker image rm <image id>            # Remove specified image from this machine
docker image rm $(docker image ls -a -q)   # Remove all images from this machine
docker login             # Log in this CLI session using your Docker credentials
docker tag <image> username/repository:tag  # Tag <image> for upload to registry
docker push username/repository:tag            # Upload tagged image to registry
docker run username/repository:tag                   # Run image from a registry
```

4.3.4.2 Get started Part3 : services

See also:

- https://docs.docker.com/get-started/part3/

Contents

- *Take down the app (docker stack rm getstartedlab)*
- *Take down the swarm (docker swarm leave –force)*

4.3.4.2.1 Prerequisites

Be sure your image works as a deployed container. Run this command, slotting in your info for username, repo, and tag:

```
docker run -p 80:80 id3pvergain/get-started:part2
```

then visit http://localhost/.

4.3.4.2.2 Introduction

In part 3, we scale our application and enable load-balancing.

To do this, we must go one level up in the hierarchy of a distributed application: the service.

- Stack
- Services (you are here)
- Container (covered in part 2)

4.3.4.2.3 About services

In a distributed application, different pieces of the app are called "services." For example, if you imagine a video sharing site, it probably includes a service for storing application data in a database, a service for video transcoding in the background after a user uploads something, a service for the front-end, and so on.

Services are really just "containers in production." A service only runs one image, but it codifies the way that image runs—what ports it should use, how many replicas of the container should run so the service has the capacity it needs, and so on.

Scaling a service changes the number of container instances running that piece of software, assigning more computing resources to the service in the process.

Luckily it's very easy to define, run, and scale services with the Docker platform – just write a docker-compose.yml file.

4.3.4.2.4 Your first docker-compose.yml file

A docker-compose.yml file is a YAML file that defines how Docker containers should behave in production.

Save this file as docker-compose.yml wherever you want. Be sure you have pushed the image you created in Part 2 to a registry, and update this .yml by replacing username/repo:tag with your image details.

```
version: "3"
services:
  web:
    # replace username/repo:tag with your name and image details
    image: id3pvergain/get-started:part2
    deploy:
      replicas: 3
      resources:
        limits:
          cpus: "0.1"
```

(continues on next page)

(continued from previous page)

```
          memory: 50M
      restart_policy:
        condition: on-failure
    ports:
      - "80:80"
    networks:
      - webnet
networks:
  webnet:
```

This docker-compose.yml file tells Docker to do the following:

- Pull the image we uploaded in step 2 from the registry.
- Run 5 instances of that image as a service called web, limiting each one to use, at most, 10% of the CPU (across all cores), and 50MB of RAM.
- Immediately restart containers if one fails.
- Map port 80 on the host to web's port 80.
- Instruct web's containers to share port 80 via a load-balanced network called webnet. (Internally, the containers themselves will publish to web's port 80 at an ephemeral port.)
- Define the webnet network with the default settings (which is a load-balanced overlay network).

4.3.4.2.5 Run your new load-balanced app

4.3.4.2.6 docker swarm init

Before we can use the docker stack deploy command we'll first run:

```
docker swarm init
```

```
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_docker\tutoriels\get_
↪started\part3> docker swarm init
```

```
Swarm initialized: current node (pnbte8079jvn6eceltf17kysp) is now a manager.

To add a worker to this swarm, run the following command:

        docker swarm join --token SWMTKN-1-
↪24yfg27ko4ma40mgips1yn5syhcs6fmcc7jesi7rwq56a9volj-4152plyrb8p3l6fpnbmqaaa7x 192.
↪168.65.3:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow␣
↪the instructions.
```

4.3.4.2.7 docker stack deploy -c docker-compose.yml getstartedlab

Now let's run it. You have to give your app a name. Here, it is set to *getstartedlab*:

```
docker stack deploy -c docker-compose.yml getstartedlab
```

```
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_docker\tutoriels\get_
↪started\part3> docker stack deploy -c docker-compose.yml getstartedlab
```

```
Creating network getstartedlab_webnet
Creating service getstartedlab_web
```

4.3.4.2.8 docker service ls

Our single service stack is running 5 container instances of our deployed image on one host. Let's investigate.

Get the service ID for the one service in our application:

```
docker service ls
```

```
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_docker\tutoriels\get_
↪started\part3> docker service ls
```

```
ID                  NAME                MODE                REPLICAS            ␣
↪IMAGE                               PORTS
tzjfv6o4bpxb        getstartedlab_web   replicated          5/5                 ␣
↪id3pvergain/get-started:part2   *:80->80/tcp
```

You'll see output for the web service, prepended with your app name. If you named it the same as shown in this example, the name will be getstartedlab_web. The service ID is listed as well, along with the number of replicas, image name, and exposed ports.

A single container running in a service is called a task.

Tasks are given unique IDs that numerically increment, up to the number of replicas you defined in docker-compose.yml.

List the tasks for your service:

4.3.4.2.9 docker service ps getstartedlab_web

```
docker service ps getstartedlab_web
```

```
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_docker\tutoriels\get_
↪started\part3> docker service ps getstartedlab_web
```

```
ID                  NAME                  IMAGE                              NODE
↪                DESIRED STATE       CURRENT STATE            ERROR                ␣
↪PORTS
qx6cvv7knp0m        getstartedlab_web.1   id3pvergain/get-started:part2   linuxkit-
↪00155d280a10   Running             Running 31 minutes ago
z9m5tsjo75pz        getstartedlab_web.2   id3pvergain/get-started:part2   linuxkit-
↪00155d280a10   Running             Running 31 minutes ago
kv05oigiytuf        getstartedlab_web.3   id3pvergain/get-started:part2   linuxkit-
↪00155d280a10   Running             Running 31 minutes ago
as0f73cwv5l8        getstartedlab_web.4   id3pvergain/get-started:part2   linuxkit-
↪00155d280a10   Running             Running 31 minutes ago
w4qqxjhsqxw3        getstartedlab_web.5   id3pvergain/get-started:part2   linuxkit-
↪00155d280a10   Running             Running 31 minutes ago
```

4.3.4.2.10 docker container ls -q

Tasks also show up if you just list all the containers on your system, though that will not be filtered by service:

```
docker container ls -q
```

```
c31e71b41bdb
8780b68999cf
4ead2b07d319
473d75fd76f2
cae7ae5c659b
f45453da50cf
b47fd081642e
```

4.3.4.2.11 Sous WSL (Windows Subsystem Linux)

```
pvergain@uc026:/mnt/c/Users/pvergain/Documents$ which curl
```

```
/usr/bin/curl
```

```
pvergain@uc026:/etc/apt$ curl http://localhost
```

```
<h3>Hello World!</h3><b>Hostname:</b> f45453da50cf<br/><b>Visits:</b> <i>cannot␣
↪connect to Redis, counter disabled</i>
```

4.3.4.2.12 Scale the app

You can scale the app by changing the replicas value in docker-compose.yml, saving the change, and re-running the docker stack deploy command:

```
docker stack deploy -c docker-compose.yml getstartedlab
```

Docker will do an in-place update, no need to tear the stack down first or kill any containers.

Now, re-run docker container ls -q to see the deployed instances reconfigured. If you scaled up the replicas, more tasks, and hence, more containers, are started.

4.3.4.2.13 Take down the app (docker stack rm getstartedlab)

Take the app down with docker stack rm:

```
docker stack rm getstartedlab
```

```
Removing service getstartedlab_web
Removing network getstartedlab_webnet
```

4.3.4.2.14 Take down the swarm (docker swarm leave –force)

```
docker swarm leave --force
```

```
Node left the swarm.
```

It's as easy as that to stand up and scale your app with Docker. You've taken a huge step towards learning how to run containers in production. Up next, you will learn how to run this app as a bonafide swarm on a cluster of Docker machines.

To recap, while typing docker run is simple enough, the true implementation of a container in production is running it as a service.

Services codify a container's behavior in a Compose file, and this file can be used to scale, limit, and redeploy our app.

Changes to the service can be applied in place, as it runs, using the same command that launched the service: **docker stack deploy**.

Some commands to explore at this stage:

```
docker stack ls                                            # List stacks or apps
docker stack deploy -c <composefile> <appname>  # Run the specified Compose file
docker service ls                 # List running services associated with an app
docker service ps <service>                  # List tasks associated with an app
docker inspect <task or container>                   # Inspect task or container
docker container ls -q                                      # List container IDs
docker stack rm <appname>                             # Tear down an application
docker swarm leave --force      # Take down a single node swarm from the manager
```

4.3.4.3 Get started Part4 : swarms

See also:

- https://docs.docker.com/get-started/part4/

Contents

4.3.4.3.1 Introduction

In *part 3*, you took an app you wrote in *part 2*, and defined how it should run in production by turning it into a service, scaling it up 5x in the process.

Here in part 4, you deploy this application onto a cluster, running it on multiple machines.

Multi-container, multi-machine applications are made possible by joining multiple machines into a *Dockerized cluster* called a **swarm**.

4.3.4.3.2 Understanding Swarm clusters

A swarm is a group of machines that are running Docker and joined into a cluster. After that has happened, you continue to run the Docker commands you're used to, but now they are executed on a cluster by a **swarm manager**.

The machines in a swarm can be physical or virtual. After joining a swarm, they are referred to as **nodes**.

Swarm managers can use several strategies to run containers, such as *emptiest node* – which fills the least utilized machines with containers. Or *global*, which ensures that each machine gets exactly one instance of the specified

container. You instruct the swarm manager to use these strategies in the Compose file, just like the one you have already been using.

Swarm managers are the only machines in a swarm that can execute your commands, or authorize other machines to join the swarm as workers. Workers are just there to provide capacity and do not have the authority to tell any other machine what it can and cannot do.

Up until now, you have been using Docker in a single-host mode on your local machine. But Docker also can be switched into swarm mode, and that's what enables the use of swarms. Enabling **swarm mode** instantly makes the current machine a swarm manager. From then on, Docker will run the commands you execute on the swarm you're managing, rather than just on the current machine.

4.3.4.3.3 Set up your swarm

A swarm is made up of multiple nodes, which can be either physical or virtual machines. The basic concept is simple enough: run docker swarm init to enable swarm mode and make your current machine a swarm manager, then run docker swarm join on other machines to have them join the swarm as workers.

Choose a tab below to see how this plays out in various contexts. We'll use VMs to quickly create a two-machine cluster and turn it into a swarm.

4.3.4.3.4 Encore Bloqué

See also:

- https://github.com/boot2docker/boot2docker/releases/download/v18.01.0-ce/boot2docker.iso

```
PS C:/WINDOWS/system32> docker-machine create -d hyperv --hyperv-virtual-switch
 ↪"myswitch" myvm1
```

```
PS C:/WINDOWS/system32> docker-machine create -d hyperv --hyperv-virtual-switch
 ↪"myswitch" myvm1
Creating CA: C:/Users/compadm/.docker/machine/certs/ca.pem
Creating client certificate: C:/Users/compadm/.docker/machine/certs/cert.pem
Running pre-create checks...
(myvm1) Image cache directory does not exist, creating it at C:/Users/compadm/.
 ↪docker/machine/cache...
(myvm1) No default Boot2Docker ISO found locally, downloading the latest release...
(myvm1) Latest release for github.com/boot2docker/boot2docker is v18.01.0-ce
(myvm1) Downloading C:/Users/compadm/.docker/machine/cache/boot2docker.iso from␣
 ↪https://github.com/boot2docker/boot2dock
er/releases/download/v18.01.0-ce/boot2docker.iso...
Error with pre-create check: "Get https://github-production-release-asset-2e65be.
 ↪s3.amazonaws.com/14930729/634fb5b0-f6ac-11e7-8f12-e1c4544a979b?X-Amz-
 ↪Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=A
KIAIWNJYAX4CSVEH53A%2F20180115%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-
 ↪Date=20180115T134730Z&X-Amz-Expires=300&X-Amz-
 ↪Signature=5efdfe365c94b790f1a95579a7f424a0731be82a19a2d806340d18c5608577be&X-Amz
-SignedHeaders=host&actor_id=0&response-content-disposition=attachment%3B
 ↪%20filename%3Dboot2docker.iso&response-content-type=application%2Foctet-stream:␣
 ↪read tcp 10.0.40.41:55806->54.231.48.184:4
43: wsarecv: Une tentative de connexion a échoué car le parti connecté n'a pas␣
 ↪répondu convenablement au-delà d'une certaine durée ou une connexion établie a␣
 ↪échoué car l'hôte de connexion n'a pa
s répondu."
```

Warning: impossible d'accéder au stockage Amazon S3[10].

[10] https://fr.wikipedia.org/wiki/Amazon_S3

4.3.4.3.4.1 Solution

Téléchargement à la maison et copie manuelle sous `C:/Users/compadm/.docker/machine/cache`.

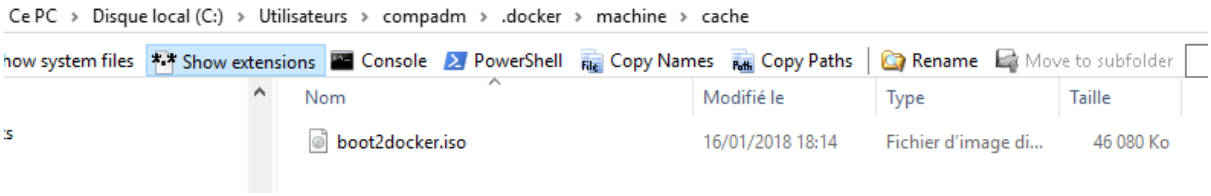


Fig. 4: https://github.com/boot2docker/boot2docker/releases/download/v18.01.0-ce/boot2docker.iso

4.4 A Simple Recipe for Django Development In Docker par Adam King (Advanced tutorial)

See also:

- https://medium.com/@adamzeitcode/a-simple-recipe-for-django-development-in-docker-bonus-testing-with-selenium-6a03

Contents

Fig. 5: https://medium.com/@adamzeitcode/a-simple-recipe-for-django-development-in-docker-bonus-testing-with-selenium-6a03

4.4.1 Dockerfile Adam King

```
# My Site
# Version: 1.0

FROM python:3

# Install Python and Package Libraries
RUN apt-get update && apt-get upgrade -y && apt-get autoremove && apt-get autoclean
RUN apt-get install -y \
    libffi-dev \
    libssl-dev \
    libmysqlclient-dev \
    libxml2-dev \
    libxslt-dev \
    libjpeg-dev \
    libfreetype6-dev \
    zlib1g-dev \
    net-tools \
    vim

# Project Files and Settings
ARG PROJECT=myproject
ARG PROJECT_DIR=/var/www/${PROJECT}

RUN mkdir -p $PROJECT_DIR
WORKDIR $PROJECT_DIR
COPY Pipfile Pipfile.lock ./
RUN pip install -U pipenv
RUN pipenv install --system

# Server
EXPOSE 8000
STOPSIGNAL SIGINT
ENTRYPOINT ["python", "manage.py"]
CMD ["runserver", "0.0.0.0:8000"]
```

Without getting too deep in the weeds about creating Dockerfiles, let's take a quick look at what's going on here. We specify some packages we want installed on our Django server (The Ubuntu image is pretty bare-bones, it doesn't even come with ping!).

4.4.1.1 WORKDIR

The WORKDIR variable is interesting in this case it's setting `/var/www/myproject/` on the server as the equivalent to your Django project's root directory. We also expose port 8000 and run the server.

Note that in this case, we're using pipenv to manage our package dependencies.

4.4.2 docker-compose.yml Adam King

```
version: "2"
services:
  django:
    container_name: django_server
    build:
      context: .
      dockerfile: Dockerfile
    image: docker_tutorial_django
    stdin_open: true
```

(continues on next page)

(continued from previous page)

```
    tty: true
    volumes:
      - .:/var/www/myproject
    ports:
      - "8000:8000"
```

Now we can run **docker-compose build** and it'll build our image which we named docker_tutorial_django that will run inside a container called django_server.

Spin it up by running **docker-compose up**.

Before we go any further, take a quick look at that docker-compose.yml file. The lines,

4.4.2.1 stdin_open: true, tty:true

```
stdin_open: true
tty: true
```

are important, because they let us run an interactive terminal.

Hit ctrl-c to kill the server running in your terminal, and then bring it up in the background with **docker-compose up -d**

docker ps tells us it's still running.

4.4.2.2 docker-compose up -d

We need to attach to that running container, in order to see its server output and pdb breakpoints. The command docker attach django_server will present you with a blank line, but if you refresh your web browser, you'll see the server output.

Drop:

```
import pdb; pdb.set_trace()
```

in your code and you'll get the interactive debugger, just like you're used to.

4.4.3 Explore your container (docker-compose exec django bash)

With your container running, you can run the command:

```
docker-compose exec django bash
```

which is a shorthand for the command:

```
docker exec -it django_server bash.
```

You'll be dropped into a bash terminal inside your running container, with a working directory of /var/www/myproject, just like you specified in your Docker configuration.

This console is where you'll want to run your manage.py tasks: execute tests, make and apply migrations, use the python shell, etc.

4.4.4 Take a break

Before we go further, let's stop and think about what we've accomplished so far.

We've now got our Django server running in a reproducible Docker container.

If you have collaborators on your project or just want to do development work on another computer, all you need to get up and running is a copy of your:

- **Dockerfile**
- **docker-compose.yml**
- **Pipfile**

You can rest easy knowing that the environments will be identical.

When it comes time to push your code to a staging or production environment, you can build on your existing Dockerfile maybe add some error logging, a production-quality web server, etc.

4.4.5 Next Steps: Add a MySQL Database

Now, we could stop here and we'd still be in a pretty good spot, but there's still a lot of Docker goodness left on the table.

Let's add a real database.

Open up your docker-compose.yml file and update it:

```
version: "2"
services:
  django:
    container_name: django_server
    build:
      context: .
      dockerfile: Dockerfile
    image: docker_tutorial_django
    stdin_open: true
    tty: true
    volumes:
      - .:/var/www/myproject
    ports:
      - "8000:8000"
    links:
      - db
    environment:
      - DATABASE_URL=mysql://root:itsasecret@db:3306/docker_tutorial_django_db

  db:
    container_name: mysql_database
    image: mysql/mysql-server
    ports:
      - "3306:3306"
    environment:
      - MYSQL_ROOT_PASSWORD=itsasecret
    volumes:
      - /Users/Adam/Development/data/mysql:/var/lib/mysql
```

4.4.5.1 db

We added a new service to our docker-compose.yml called **db**.

I named the container **mysql_database**, and we are basing it off the image mysql/mysql-server. Check out http://hub.docker.com for, like, a million Docker images.

4.4.5.1.1 MYSQL_ROOT_PASSWORD

We set the root password for the MySQL server, as well as expose a port (host-port:container-port) to the 'outer world.' We also need to specify the location of our MySQL files. I'm putting them in a directory called data in my Development directory.

In our django service, I added a link to the db service. docker-compose acts as a sort of 'internal DNS' for our Docker containers. If I run docker-compose up -d and then jump into my running Django container with docker-compose exec django bash, I can ping db and confirm the connection:

```
root@e94891041716:/var/www/myproject# ping db
```

```
PING db (172.23.0.3): 56 data bytes
64 bytes from 172.23.0.3: icmp_seq=0 ttl=64 time=0.232 ms
64 bytes from 172.23.0.3: icmp_seq=1 ttl=64 time=0.229 ms
64 bytes from 172.23.0.3: icmp_seq=2 ttl=64 time=0.247 ms
64 bytes from 172.23.0.3: icmp_seq=3 ttl=64 time=0.321 ms
64 bytes from 172.23.0.3: icmp_seq=4 ttl=64 time=0.310 ms
^C--- db ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.229/0.268/0.321/0.040 ms
root@e94891041716:/var/www/myproject#
```

4.4.5.2 DATABASE_URL

Adding the environment variable, DATABASE_URL=mysql://root:itsasecret@db:3306/docker_tutorial_django_db Will allow our Django database to use a real, production-ready version of MySQL instead of the default SQLite.

Note that you'll need to use a package like getenv in your settings.py to read environment variables:

```
DATABASE_URL=env('DATABASE_URL')
```

If it's your first time running a MySQL server, you might have a little bit of housekeeping: setting the root password, granting privileges, etc.

Check the corresponding documentation for the server you're running. You can jump into the running MySQL server the same way:

```
$ docker-compose exec db bash
```

```
$ mysql -p itsasecret
> CREATE DATABASE docker_tutorial_django_db;
etc, etc
```

4.5 Modern DevOps with Django par Jacob Cook (Advanced tutorial)

See also:

- https://peakwinter.net/
- https://twitter.com/peakwinter
- https://github.com/peakwinter/modern-devops-django-sample
- https://peakwinter.net/blog/modern-devops-django/

Contents

4.5.1 tree

```
pvergain@uc026:/mnt/y/projects_id3/P5N001/XLOGCA135_tutorial_docker/tutorial_
↪docker/tutoriels/modern_devops$ tree
```

```
├── modern-devops-django-sample
│   ├── docker-compose.ci.yml
│   ├── docker-compose.prod.yml
│   ├── docker-compose.staging.yml
│   ├── docker-compose.test.yml
│   ├── docker-compose.yml
│   ├── Dockerfile
│   ├── LICENSE
│   ├── manage.py
│   ├── modern_devops
│   │   ├── __init__.py
│   │   ├── settings.py
│   │   ├── urls.py
│   │   └── wsgi.py
│   ├── myapp
│   │   ├── admin.py
│   │   ├── apps.py
│   │   ├── __init__.py
│   │   ├── migrations
│   │   │   └── __init__.py
│   │   ├── models.py
│   │   ├── tests.py
│   │   └── views.py
│   ├── README.md
│   ├── requirements.txt
│   └── uwsgi.ini
└── modern_devops.rst
```

4.5.2 Dockerfile Jacob Cook

```
FROM python:3-alpine3.6

ENV PYTHONUNBUFFERED=1

RUN apk add --no-cache linux-headers bash gcc \
    musl-dev libjpeg-turbo-dev libpng libpq \
    postgresql-dev uwsgi uwsgi-python3 git \
```

(continues on next page)

(continued from previous page)

```
    zlib-dev libmagic

WORKDIR /site
COPY ./ /site
RUN pip install -U -r /site/requirements.txt
CMD python manage.py migrate && uwsgi --ini=/site/uwsgi.ini
```

First things first is our Dockerfile. This is the configuration that takes a base image (in our case Python 3.6 installed on a thin copy of Alpine Linux) and installs everything our application needs to run, including our Python dependencies.

It also sets a default command to use - this is the command that will be executed each time our container starts up in production.

We want it to check for any pending migrations, run them, then start up our uWSGI server to make our application available to the Internet. It's safe to do this because if any migrations failed after our automatic deployments to staging, we would be able to recover from that and make the necessary changes before we tag a release and deploy to production.

This Dockerfile example builds a container with necessary dependencies for things like image uploads as well as connections to a PostgreSQL database.

4.5.3 docker-compose.yml Jacob Cook

We can now build our application with docker build -t myapp . and run it with docker run -it myapp. But in the case of our development environment, **we are going to use Docker Compose in practice**.

The Docker Compose configuration below is sufficient for our development environment, and will serve as a base for our configurations in staging and production, which can include things like Celery workers and monitoring services.

```
version: '3'

services:
  app:
    build: ./
    command: bash -c "python3 manage.py migrate && python3 manage.py runserver 0.0.
↪0.0:8000"
    volumes:
      - ./:/site:rw
    depends_on:
      - postgresql
      - redis
    environment:
      DJANGO_SETTINGS_MODULE: myapp.settings.dev
    ports:
      - "8000:8000"

  postgresql:
    restart: always
    image: postgres:10-alpine
    volumes:
      - ./.dbdata:/var/lib/postgresql:rw
    environment:
      POSTGRES_USER: myapp
      POSTGRES_PASSWORD: myapp
      POSTGRES_DB: myapp

  redis:
    restart: always
    image: redis:latest
```

This is a pretty basic configuration - all we are doing is setting a startup command for our app (similar to the entrypoint in our Docker container, except this time we are going to run Django's internal dev server instead) and initializing PostgreSQL and Redis containers that will be linked with it.

It's important to note that volumes line in our app service — this is going to bind the current directory of source code on our host machine to the installation folder inside the container.

That way we can make changes to the code locally and still use the automatic reloading feature of the Django dev server.

At this point, all we need to do is **docker-compose up**, and our Django application will be listening on port 8000, just as if we were running it from a virtualenv locally. This configuration is perfectly suitable for developer environments — all anyone needs to do to get started using the exact same environment as you is to clone the Git repository and run docker-compose up !

4.5.4 Testing and Production

For testing your application, whether that's on your local machine or via Gitlab CI, I've found it's helpful to create a clone of this docker-compose.yml configuration and customize the command directive to instead run whatever starts your test suite. In my case, I use the Python coverage library, so I have a second file called docker-compose.test.yml which is exactly the same as the first, save for the command directive has been changed to:

```
command: bash -c "coverage run --source='.' manage.py test myapp && coverage report
↪"
```

4.5.4.1 docker-compose.test.yml

```
version: '3'

services:
  app:
        build: ./
        command: bash -c "coverage run --source='.' manage.py test kanban &&␣
↪coverage report"
        volumes:
          - ./:/site:rw
        depends_on:
          - postgresql
          - redis
        environment:
          DJANGO_SETTINGS_MODULE: modern_devops.settings.test

  postgresql:
        restart: always
        image: postgres:10-alpine
        environment:
          POSTGRES_USER: myapp_test
          POSTGRES_PASSWORD: myapp_test
          POSTGRES_DB: myapp_test

  redis:
        restart: always
        image: redis:latest
```

Then, I run my test suite locally with:

```
docker-compose -p test -f docker-compose.test.yml up.
```

4.5.4.2 docker-compose.staging.yml

```
version: '3'

services:
  app:
        image: registry.gitlab.com/pathto/myapp:staging
        environment:
          DJANGO_SETTINGS_MODULE: modern_devops.settings.staging
        volumes:
          - /var/data/myapp/staging/settings.py:/site/modern_devops/settings/
↪staging.py:ro
        depends_on:
          - postgresql
          - redis
        networks:
          - default
          - public

  postgresql:
        image: postgres:10-alpine
        volumes:
          - /var/data/realtime/myapp/staging/db:/var/lib/postgresql/data:rw
        environment:
          POSTGRES_USER: myapp_staging
          POSTGRES_PASSWORD: myapp_staging
          POSTGRES_DB: myapp_staging

  redis:
        image: redis:latest

networks:
  public:
        external: true
```

4.5.4.3 docker-compose.prod.yml

For production and staging environments, I do the same thing — duplicate the file with the few changes I need to make for the environment in particular. In this case, for production, I don't want to provide a build path — I want to tell Docker that it needs to take my application from the container registry each time it starts up.

To do so, remove the build directive and add an image one like so:

```
image: registry.gitlab.com/pathto/myapp:prod
```

```
version: '3'

services:
  app:
        image: registry.gitlab.com/pathto/myapp:prod
        environment:
          DJANGO_SETTINGS_MODULE: modern_devops.settings.prod
        volumes:
          - /var/data/myapp/prod/settings.py:/site/modern_devops/settings/prod.
↪py:ro
        depends_on:
          - postgresql
          - redis
        networks:
          - default
```

(continues on next page)

(continued from previous page)

```
      - public

  postgresql:
        image: postgres:10-alpine
        volumes:
          - /var/data/realtime/myapp/prod/db:/var/lib/postgresql/data:rw
        environment:
          POSTGRES_USER: myapp_staging
          POSTGRES_PASSWORD: myapp_staging
          POSTGRES_DB: myapp_staging

  redis:
        image: redis:latest

networks:
  public:
        external: true
```

4.6 Django for beginners par William Vincent

Contents

```
        * d posts_post
```

4.6.1 Thanks to William Vincent

See also:

- https://twitter.com/wsv3000
- https://wsvincent.com/django-docker-postgresql/
- https://gitlab.com/gdevops/wsdjangoforbeginners

Fig. 6: https://twitter.com/wsv3000

```
total 52
drwxrwxr-x. 6 pvergain pvergain 4096 28 mai   16:10 ch10-bootstrap
drwxrwxr-x. 6 pvergain pvergain 4096 28 mai   16:10 ch11-password-change-reset
drwxrwxr-x. 6 pvergain pvergain 4096 28 mai   16:10 ch12-email
drwxrwxr-x. 7 pvergain pvergain 4096 28 mai   16:10 ch13-newspaper-app
```

(continues on next page)

(continued from previous page)

```
drwxrwxr-x. 7 pvergain pvergain 4096 28 mai   16:10 ch14-permissions-and-
↪authorizations
drwxrwxr-x. 7 pvergain pvergain 4096 28 mai   16:10 ch15-comments
drwxrwxr-x. 4 pvergain pvergain   92 28 mai   16:10 ch2-hello-world-app
drwxrwxr-x. 5 pvergain pvergain  103 28 mai   16:10 ch3-pages-app
drwxrwxr-x. 5 pvergain pvergain 4096 28 mai   16:15 ch4-message-board-app
drwxrwxr-x. 7 pvergain pvergain 4096 28 mai   16:10 ch5-blog-app
drwxrwxr-x. 7 pvergain pvergain 4096 28 mai   16:10 ch6-blog-app-with-forms
drwxrwxr-x. 7 pvergain pvergain 4096 28 mai   16:10 ch7-blog-app-with-users
drwxrwxr-x. 4 pvergain pvergain 4096 28 mai   16:10 ch8-custom-user-model
drwxrwxr-x. 5 pvergain pvergain 4096 28 mai   16:10 ch9-user-authentication
-rw-rw-r--. 1 pvergain pvergain  689 28 mai   16:15 Readme.md
```

4.6.2 tree ch4-message-board-app

See also:

- https://gitlab.com/gdevops/wsdjangoforbeginners

```
tree ch4-message-board-app
```

```
ch4-message-board-app/
├── Dockerfile
├── manage.py
├── mb_project
│   ├── __init__.py
│   ├── settings.py
│   ├── urls.py
│   └── wsgi.py
├── Pipfile
├── Pipfile.lock
├── posts
│   ├── admin.py
│   ├── apps.py
│   ├── __init__.py
│   ├── migrations
│   │   ├── 0001_initial.py
│   │   └── __init__.py
│   ├── models.py
│   ├── tests.py
│   ├── urls.py
│   └── views.py
├── Procfile
└── templates
    └── home.html
```

4.6.3 Dockerfile from Will Vincent

See also:

- https://wsvincent.com/django-docker-postgresql/
- https://gitlab.com/gdevops/wsdjangoforbeginners/blob/master/ch4-message-board-app/Dockerfile

```
FROM python:3.6

ENV PYTHONUNBUFFERED 1

```

(continues on next page)

(continued from previous page)

```
COPY . /code/
WORKDIR /code/

RUN pip install pipenv
RUN pipenv install --system

EXPOSE 8000
```

4.6.4 docker build .

We can't run a Docker container until it has an image so let's do that by building it.

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\djangoforbeginners\ch4-message-board-app>docker build .
```

```
Sending build context to Docker daemon   47.1kB
Step 1/7 : FROM python:3.6
 ---> c1e459c00dc3
Step 2/7 : ENV PYTHONUNBUFFERED 1
 ---> Using cache
 ---> 221d2e9ab9e4
Step 3/7 : COPY . /code/
 ---> 874258521e07
Step 4/7 : WORKDIR /code/
Removing intermediate container 19c5a97f7968
 ---> b1a4a747aea7
Step 5/7 : RUN pip install pipenv
 ---> Running in 42f7073e751d
Collecting pipenv
  Downloading pipenv-9.0.3.tar.gz (3.9MB)
Collecting virtualenv (from pipenv)
  Downloading virtualenv-15.1.0-py2.py3-none-any.whl (1.8MB)
Collecting pew>=0.1.26 (from pipenv)
  Downloading pew-1.1.2-py2.py3-none-any.whl
Requirement already satisfied: pip>=9.0.1 in /usr/local/lib/python3.6/site-
↪packages (from pipenv)
Collecting requests>2.18.0 (from pipenv)
  Downloading requests-2.18.4-py2.py3-none-any.whl (88kB)
Collecting flake8>=3.0.0 (from pipenv)
  Downloading flake8-3.5.0-py2.py3-none-any.whl (69kB)
Collecting urllib3>=1.21.1 (from pipenv)
  Downloading urllib3-1.22-py2.py3-none-any.whl (132kB)
Requirement already satisfied: setuptools>=17.1 in /usr/local/lib/python3.6/site-
↪packages (from pew>=0.1.26->pipenv)
Collecting virtualenv-clone>=0.2.5 (from pew>=0.1.26->pipenv)
  Downloading virtualenv-clone-0.2.6.tar.gz
Collecting certifi>=2017.4.17 (from requests>2.18.0->pipenv)
  Downloading certifi-2018.1.18-py2.py3-none-any.whl (151kB)
Collecting idna<2.7,>=2.5 (from requests>2.18.0->pipenv)
  Downloading idna-2.6-py2.py3-none-any.whl (56kB)
Collecting chardet<3.1.0,>=3.0.2 (from requests>2.18.0->pipenv)
  Downloading chardet-3.0.4-py2.py3-none-any.whl (133kB)
Collecting pycodestyle<2.4.0,>=2.0.0 (from flake8>=3.0.0->pipenv)
  Downloading pycodestyle-2.3.1-py2.py3-none-any.whl (45kB)
Collecting mccabe<0.7.0,>=0.6.0 (from flake8>=3.0.0->pipenv)
  Downloading mccabe-0.6.1-py2.py3-none-any.whl
Collecting pyflakes<1.7.0,>=1.5.0 (from flake8>=3.0.0->pipenv)
  Downloading pyflakes-1.6.0-py2.py3-none-any.whl (227kB)
Building wheels for collected packages: pipenv, virtualenv-clone
```

(continues on next page)

(continued from previous page)

```
  Running setup.py bdist_wheel for pipenv: started
  Running setup.py bdist_wheel for pipenv: finished with status 'done'
  Stored in directory: /root/.cache/pip/wheels/78/cf/b7/
↪549d89ddbafb1cf3da825b97b730a7e1ac75602de9865d036e
  Running setup.py bdist_wheel for virtualenv-clone: started
  Running setup.py bdist_wheel for virtualenv-clone: finished with status 'done'
  Stored in directory: /root/.cache/pip/wheels/24/51/ef/
↪93120d304d240b4b6c2066454250a1626e04f73d34417b956d
Successfully built pipenv virtualenv-clone
Installing collected packages: virtualenv, virtualenv-clone, pew, certifi, idna,␣
↪chardet, urllib3, requests, pycodestyle, mccabe, pyflakes, flake8, pipenv
Successfully installed certifi-2018.1.18 chardet-3.0.4 flake8-3.5.0 idna-2.6␣
↪mccabe-0.6.1 pew-1.1.2 pipenv-9.0.3 pycodestyle-2.3.1 pyflakes-1.6.0 requests-2.
↪18.4 urllib3-1.22 virtualenv-15.1.0 virtualenv-clone-0.2.6
Removing intermediate container 42f7073e751d
 ---> 89cfca6a042a
Step 6/7 : RUN pipenv install --system
 ---> Running in 95effdc52999
Installing dependencies from Pipfile.lock (48d763)...
Removing intermediate container 95effdc52999
 ---> 60e848b90903
Step 7/7 : EXPOSE 8000
 ---> Running in 325a08f841b9
Removing intermediate container 325a08f841b9
 ---> 7bd32294cda7
Successfully built 7bd32294cda7
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.
```

4.6.5 mb_project/settings.py

```
"""
Django settings for mb_project project.

Generated by 'django-admin startproject' using Django 2.0.

For more information on this file, see
https://docs.djangoproject.com/en/2.0/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/2.0/ref/settings/
"""

import os

# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))


# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/2.0/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = '_%!$0#1!cxd(rrj%=5rmeu%qiccz)7vsorclhey9-w00xq7&t4'

# SECURITY WARNING: don't run with debug turned on in production!
```

(continues on next page)

(continued from previous page)

```
DEBUG = True

ALLOWED_HOSTS = ['*']


# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'posts',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'mb_project.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'templates')],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'mb_project.wsgi.application'


# Database
# https://docs.djangoproject.com/en/2.0/ref/settings/#databases

# https://djangoforbeginners.com/docker-postgresql/
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'postgres',
        'USER': 'postgres',
        'HOST': 'db', # set in docker-compose.yml
        'POST': 5432 # default postgres port
    }
}
```

(continues on next page)

(continued from previous page)

```
# Password validation
# https://docs.djangoproject.com/en/2.0/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.
↪UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

# Internationalization
# https://docs.djangoproject.com/en/2.0/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_L10N = True

USE_TZ = True


# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/2.0/howto/static-files/

STATIC_URL = '/static/'
```

4.6.6 pipenv install psycopg2

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\djangoforbeginners\ch4-message-board-app>pipenv install psycopg2
```

```
Installing psycopg2...
Collecting psycopg2
  Using cached psycopg2-2.7.3.2-cp36-cp36m-win_amd64.whl
Installing collected packages: psycopg2
Successfully installed psycopg2-2.7.3.2

Adding psycopg2 to Pipfile's [packages]...
Locking [dev-packages] dependencies...
Locking [packages] dependencies...
Updated Pipfile.lock (c2c6d4)!
```

4.6.7 docker-compose.yml William Vincent

```
version: '3'

services:
  db:
    image: postgres:10.1
    volumes:
      - postgres_data:/var/lib/postgresl/data/

  web:
    build: .
    command: python /code/manage.py migrate --noinput
    command: python /code/manage.py runserver 0.0.0.0:8000
    volumes:
      - .:/code
    ports:
      - "8000:8000"
    depends_on:
      - db

volumes:
  postgres_data:
```

On the top line we're using the most recent version of Compose which is **3**.

4.6.7.1 db

Under **db** for the database we want the Docker image for Postgres 10.1 and use volumes to tell Compose where the container should be located in our Docker container.

4.6.7.2 web

For **web** we're specifying how the web service will run. First Compose needs to build an image from the current directory, automatically run migrations and hide the output, then start up the server at 0.0.0.0:8000.

4.6.7.3 volumes

We use volumes to tell Compose to store the code in our Docker container at /code/.

> **Warning:** Cela nous permet d'avoir accès à notre code sur le host.

4.6.7.4 ports

The ports config lets us map our own port 8000 to the port 8000 in the Docker container.

And finally depends_on says that we should start the db first before running our web services.

4.6.7.5 volumes

The last section volumes is because Compose has a rule that you must list named volumes in a top-level volumes key.

Docker is all set!

4.6.8 docker-compose run web python /code/manage.py migrate –noinput

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\djangoforbeginners\ch4-message-board-app>docker-compose run web␣
↪python /code/manage.py migrate --noinput
```

```
WARNING: The Docker Engine you're using is running in swarm mode.

Compose does not use swarm mode to deploy services to multiple nodes in a swarm.␣
↪All containers will be scheduled on the current node.

To deploy your application across the swarm, use `docker stack deploy`.

Creating network "ch4messageboardapp_default" with the default driver
Creating volume "ch4messageboardapp_postgres_data" with default driver
Pulling db (postgres:10.1)...
10.1: Pulling from library/postgres
Digest: sha256:3f4441460029e12905a5d447a3549ae2ac13323d045391b0cb0cf8b48ea17463
Status: Downloaded newer image for postgres:10.1
Creating ch4messageboardapp_db_1 ... done
Building web
Step 1/7 : FROM python:3.6
 ---> c1e459c00dc3
Step 2/7 : ENV PYTHONUNBUFFERED 1
 ---> Using cache
 ---> 221d2e9ab9e4
Step 3/7 : COPY . /code/
 ---> e03ac813d986
Step 4/7 : WORKDIR /code/
Removing intermediate container 7d82a1620667
 ---> f810a068e5ab
Step 5/7 : RUN pip install pipenv
 ---> Running in 95827f363022
Collecting pipenv
  Downloading pipenv-9.0.3.tar.gz (3.9MB)
Collecting virtualenv (from pipenv)
  Downloading virtualenv-15.1.0-py2.py3-none-any.whl (1.8MB)
Collecting pew>=0.1.26 (from pipenv)
  Downloading pew-1.1.2-py2.py3-none-any.whl
Requirement already satisfied: pip>=9.0.1 in /usr/local/lib/python3.6/site-
↪packages (from pipenv)
Collecting requests>2.18.0 (from pipenv)
  Downloading requests-2.18.4-py2.py3-none-any.whl (88kB)
Collecting flake8>=3.0.0 (from pipenv)
  Downloading flake8-3.5.0-py2.py3-none-any.whl (69kB)
Collecting urllib3>=1.21.1 (from pipenv)
  Downloading urllib3-1.22-py2.py3-none-any.whl (132kB)
Collecting virtualenv-clone>=0.2.5 (from pew>=0.1.26->pipenv)
  Downloading virtualenv-clone-0.2.6.tar.gz
Requirement already satisfied: setuptools>=17.1 in /usr/local/lib/python3.6/site-
↪packages (from pew>=0.1.26->pipenv)
Collecting certifi>=2017.4.17 (from requests>2.18.0->pipenv)
  Downloading certifi-2018.1.18-py2.py3-none-any.whl (151kB)
Collecting idna<2.7,>=2.5 (from requests>2.18.0->pipenv)
  Downloading idna-2.6-py2.py3-none-any.whl (56kB)
Collecting chardet<3.1.0,>=3.0.2 (from requests>2.18.0->pipenv)
  Downloading chardet-3.0.4-py2.py3-none-any.whl (133kB)
Collecting pycodestyle<2.4.0,>=2.0.0 (from flake8>=3.0.0->pipenv)
  Downloading pycodestyle-2.3.1-py2.py3-none-any.whl (45kB)
Collecting mccabe<0.7.0,>=0.6.0 (from flake8>=3.0.0->pipenv)
  Downloading mccabe-0.6.1-py2.py3-none-any.whl
Collecting pyflakes<1.7.0,>=1.5.0 (from flake8>=3.0.0->pipenv)
```

(continues on next page)

(continued from previous page)

```
  Downloading pyflakes-1.6.0-py2.py3-none-any.whl (227kB)
Building wheels for collected packages: pipenv, virtualenv-clone
  Running setup.py bdist_wheel for pipenv: started
  Running setup.py bdist_wheel for pipenv: finished with status 'done'
  Stored in directory: /root/.cache/pip/wheels/78/cf/b7/
↪549d89ddbafb1cf3da825b97b730a7e1ac75602de9865d036e
  Running setup.py bdist_wheel for virtualenv-clone: started
  Running setup.py bdist_wheel for virtualenv-clone: finished with status 'done'
  Stored in directory: /root/.cache/pip/wheels/24/51/ef/
↪93120d304d240b4b6c2066454250a1626e04f73d34417b956d
Successfully built pipenv virtualenv-clone
Installing collected packages: virtualenv, virtualenv-clone, pew, certifi, idna,␣
↪chardet, urllib3, requests, pycodestyle, mccabe, pyflakes, flake8, pipenv
Successfully installed certifi-2018.1.18 chardet-3.0.4 flake8-3.5.0 idna-2.6␣
↪mccabe-0.6.1 pew-1.1.2 pipenv-9.0.3 pycodestyle-2.3.1 pyflakes-1.6.0 requests-2.
↪18.4 urllib3-1.22 virtualenv-15.1.0 virtualenv-clone-0.2.6
Removing intermediate container 95827f363022
 ---> 5c4805a82b1e
Step 6/7 : RUN pipenv install --system
 ---> Running in 083ee437bbd2
Installing dependencies from Pipfile.lock (c2c6d4)ÔÇª
Removing intermediate container 083ee437bbd2
 ---> 8750b71fcc3f
Step 7/7 : EXPOSE 8000
 ---> Running in 79daa2dc8134
Removing intermediate container 79daa2dc8134
 ---> c5e7e58a668c
Successfully built c5e7e58a668c
Successfully tagged ch4messageboardapp_web:latest
WARNING: Image for service web was built because it did not already exist. To␣
↪rebuild this image you must use `docker-compose build` or `docker-compose up --
↪build`.
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, posts, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying posts.0001_initial... OK
  Applying sessions.0001_initial... OK
```

4.6.9 docker-compose run web python /code/manage.py createsuperuser

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\djangoforbeginners\ch4-message-board-app>docker-compose run web␣
↪python /code/manage.py createsuperuser
```

```
WARNING: The Docker Engine you're using is running in swarm mode.
```

(continues on next page)

(continued from previous page)

```
Compose does not use swarm mode to deploy services to multiple nodes in a swarm.␣
↪All containers will be scheduled on the current node.

To deploy your application across the swarm, use `docker stack deploy`.

Starting ch4messageboardapp_db_1 ... done
Username (leave blank to use 'root'):
Email address: patrick.vergain@id3.eu
Password:
Password (again):
The password is too similar to the email address.
This password is too short. It must contain at least 8 characters.
Password:
Password (again):
Superuser created successfully.
```

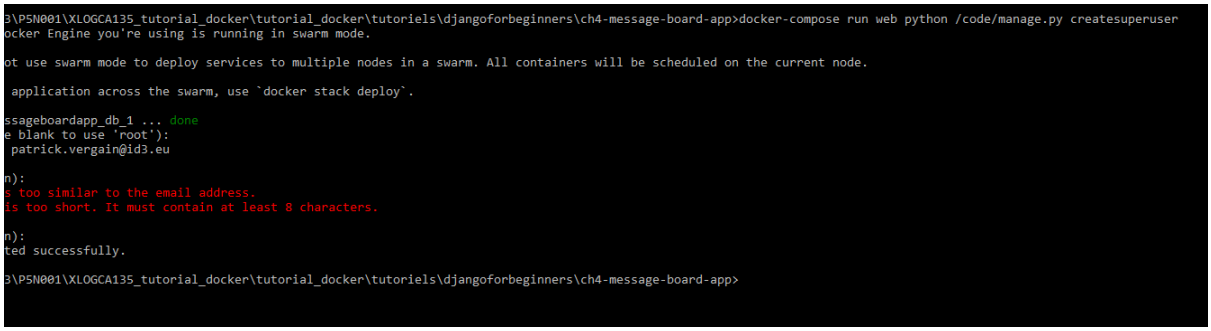


Fig. 7: docker-compose run web python /code/manage.py createsuperuser

4.6.10 docker-compose up

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\djangoforbeginners\ch4-message-board-app>docker-compose up
```

```
WARNING: The Docker Engine you're using is running in swarm mode.

Compose does not use swarm mode to deploy services to multiple nodes in a swarm.␣
↪All containers will be scheduled on the current node.

To deploy your application across the swarm, use `docker stack deploy`.

ch4messageboardapp_db_1 is up-to-date
Creating ch4messageboardapp_web_1 ... done
Attaching to ch4messageboardapp_db_1, ch4messageboardapp_web_1
db_1   | The files belonging to this database system will be owned by user
↪"postgres".
db_1   | This user must also own the server process.
db_1   |
db_1   | The database cluster will be initialized with locale "en_US.utf8".
db_1   | The default database encoding has accordingly been set to "UTF8".
db_1   | The default text search configuration will be set to "english".
db_1   |
db_1   | Data page checksums are disabled.
db_1   |
db_1   | fixing permissions on existing directory /var/lib/postgresql/data ... ok
db_1   | creating subdirectories ... ok
db_1   | selecting default max_connections ... 100
```

(continues on next page)

(continued from previous page)

```
db_1    | selecting default shared_buffers ... 128MB
db_1    | selecting dynamic shared memory implementation ... posix
db_1    | creating configuration files ... ok
db_1    | running bootstrap script ... ok
db_1    | performing post-bootstrap initialization ... ok
db_1    | syncing data to disk ... ok
db_1    |
db_1    | Success. You can now start the database server using:
db_1    |
db_1    |     pg_ctl -D /var/lib/postgresql/data -l logfile start
db_1    |
db_1    |
db_1    | WARNING: enabling "trust" authentication for local connections
db_1    | You can change this by editing pg_hba.conf or using the option -A, or
db_1    | --auth-local and --auth-host, the next time you run initdb.
db_1    | ****************************************************
db_1    | WARNING: No password has been set for the database.
db_1    |          This will allow anyone with access to the
db_1    |          Postgres port to access your database. In
db_1    |          Docker's default configuration, this is
db_1    |          effectively any other container on the same
db_1    |          system.
db_1    |
db_1    |          Use "-e POSTGRES_PASSWORD=password" to set
db_1    |          it in "docker run".
db_1    | ****************************************************
db_1    | waiting for server to start....2018-01-23 08:34:30.556 UTC [39] LOG:
↪listening on IPv4 address "127.0.0.1", port 5432
db_1    | 2018-01-23 08:34:30.557 UTC [39] LOG:  could not bind IPv6 address "::1":
↪Cannot assign requested address
db_1    | 2018-01-23 08:34:30.557 UTC [39] HINT:  Is another postmaster already
↪running on port 5432? If not, wait a few seconds and retry.
db_1    | 2018-01-23 08:34:30.682 UTC [39] LOG:  listening on Unix socket "/var/run/
↪postgresql/.s.PGSQL.5432"
db_1    | 2018-01-23 08:34:30.865 UTC [40] LOG:  database system was shut down at
↪2018-01-23 08:34:28 UTC
db_1    | 2018-01-23 08:34:30.928 UTC [39] LOG:  database system is ready to accept
↪connections
db_1    |  done
db_1    | server started
db_1    | ALTER ROLE
db_1    |
db_1    |
db_1    | /usr/local/bin/docker-entrypoint.sh: ignoring /docker-entrypoint-initdb.d/
↪*
db_1    |
db_1    | 2018-01-23 08:34:31.493 UTC [39] LOG:  received fast shutdown request
db_1    | waiting for server to shut down....2018-01-23 08:34:31.557 UTC [39] LOG:
↪aborting any active transactions
db_1    | 2018-01-23 08:34:31.559 UTC [39] LOG:  worker process: logical
↪replication launcher (PID 46) exited with exit code 1
db_1    | 2018-01-23 08:34:31.560 UTC [41] LOG:  shutting down
db_1    | 2018-01-23 08:34:32.052 UTC [39] LOG:  database system is shut down
db_1    |  done
db_1    | server stopped
db_1    |
db_1    | PostgreSQL init process complete; ready for start up.
db_1    |
db_1    | 2018-01-23 08:34:32.156 UTC [1] LOG:  listening on IPv4 address "0.0.0.0",
↪ port 5432
db_1    | 2018-01-23 08:34:32.156 UTC [1] LOG:  listening on IPv6 address "::",
↪port 5432
```

(continues on next page)

(continued from previous page)

```
db_1    | 2018-01-23 08:34:32.256 UTC [1] LOG:  listening on Unix socket "/var/run/
↪postgresql/.s.PGSQL.5432"
db_1    | 2018-01-23 08:34:32.429 UTC [57] LOG:  database system was shut down at␣
↪2018-01-23 08:34:31 UTC
db_1    | 2018-01-23 08:34:32.483 UTC [1] LOG:  database system is ready to accept␣
↪connections
web_1   | Performing system checks...
web_1   |
web_1   | System check identified no issues (0 silenced).
web_1   | January 23, 2018 - 08:46:09
web_1   | Django version 2.0.1, using settings 'mb_project.settings'
web_1   | Starting development server at http://0.0.0.0:8000/
web_1   | Quit the server with CONTROL-C.
```

We can confirm it works by navigating to http://127.0.0.1:8000/ where you'll see the same homepage as before.

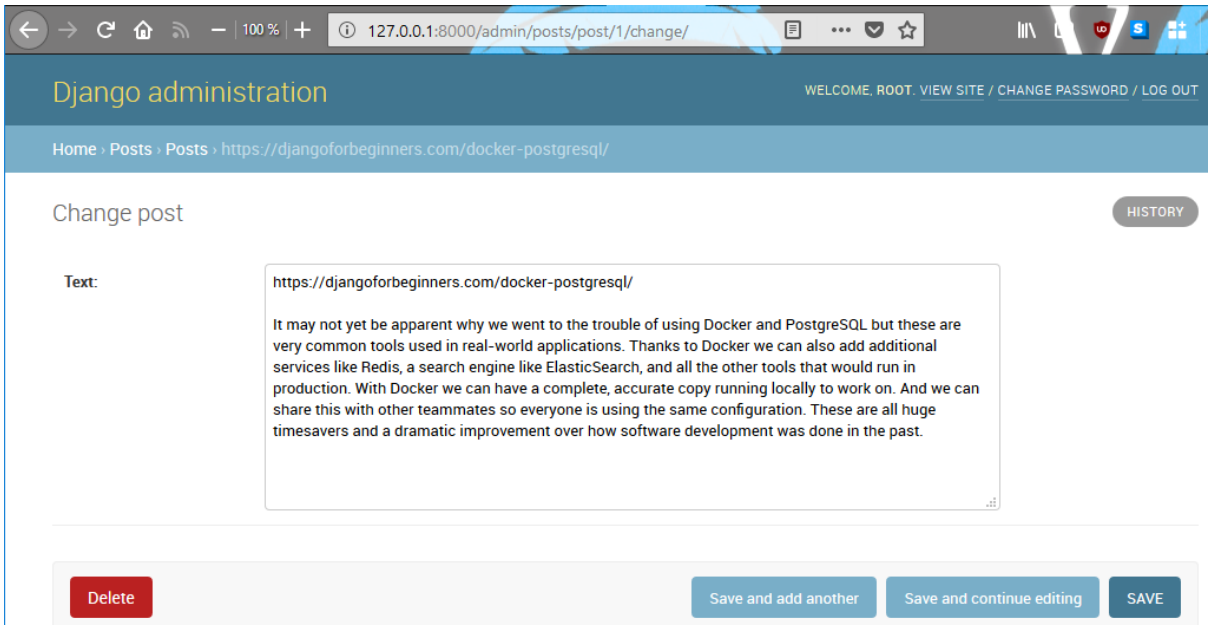


Fig. 8: http://127.0.0.1:8000/admin/posts/post/1/change/

4.6.11 docker-compose ps

```
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\william_vincent\ch4-message-board-app> docker-compose ps
```

```
          Name                        Command               State           Ports
-------------------------------------------------------------------------------------
↪-------
ch4messageboardapp_db_1    docker-entrypoint.sh postgres    Up      5432/tcp
ch4messageboardapp_web_1   python /code/manage.py run ...   Up      0.0.0.0:8000->
↪8000/tcp

::
```

4.6.12 docker-compose exec db bash

```
docker-compose exec db bash
```

4.6.13 psql -d db -U postgres

```
root@ee941cf5bc20:/# psql -U postgres
```

```
psql (10.1)
Type "help" for help.
```

4.6.13.1 dt

```
postgres=# \dt
```

```
                List of relations
 Schema |            Name             | Type  |  Owner
--------+-----------------------------+-------+----------
 public | auth_group                  | table | postgres
 public | auth_group_permissions      | table | postgres
 public | auth_permission             | table | postgres
 public | auth_user                   | table | postgres
 public | auth_user_groups            | table | postgres
 public | auth_user_user_permissions  | table | postgres
 public | django_admin_log            | table | postgres
 public | django_content_type         | table | postgres
 public | django_migrations           | table | postgres
 public | django_session              | table | postgres
 public | posts_post                  | table | postgres
(11 rows)
```

4.6.13.2 conninfo

```
postgres=# \conninfo
```

```
You are connected to database "postgres" as user "postgres" via socket in "/var/
↪run/postgresql" at port "5432".
```

```
postgres=# \l
```

```
                                 List of databases
   Name    |  Owner   | Encoding |  Collate   |   Ctype    |   Access privileges
-----------+----------+----------+------------+------------+-----------------------
 postgres  | postgres | UTF8     | en_US.utf8 | en_US.utf8 |
 template0 | postgres | UTF8     | en_US.utf8 | en_US.utf8 | =c/postgres          +
           |          |          |            |            | postgres=CTc/postgres
 template1 | postgres | UTF8     | en_US.utf8 | en_US.utf8 | =c/postgres          +
           |          |          |            |            | postgres=CTc/postgres
(3 rows)
```

4.6.13.3 dn

```
postgres=# \dn
  List of schemas
  Name  |  Owner
--------+----------
 public | postgres
(1 row)
```

4.6.13.4 d posts_post

```
postgres=# \d posts_post
```

```
                          Table "public.posts_post"
 Column |  Type   | Collation | Nullable |                 Default
--------+---------+-----------+----------+-----------------------------------------
 id     | integer |           | not null | nextval('posts_post_id_seq'::regclass)
 text   | text    |           | not null |
Indexes:
    "posts_post_pkey" PRIMARY KEY, btree (id)
```

4.7 A Brief Intro to Docker for Djangonauts par Lacey Williams

See also:

- https://twitter.com/laceynwilliams
- https://twitter.com/laceynwilliams/status/921421761039818754
- https://www.revsys.com/tidbits/brief-intro-docker-djangonauts/
- https://www.revsys.com/tidbits/brief-intro-docker-djangonauts/
- https://www.revsys.com/tidbits/docker-useful-command-line-stuff/
- https://www.youtube.com/watch?v=v5jfDDg55xs&feature=youtu.be&a=

Contents

* *db*
  - *volumes*
* *web*
  - *build .*
  - *command: python /code/manage.py migrate –noinput*
  - *command: python /code/manage.py runserver 0.0.0.0:8000*

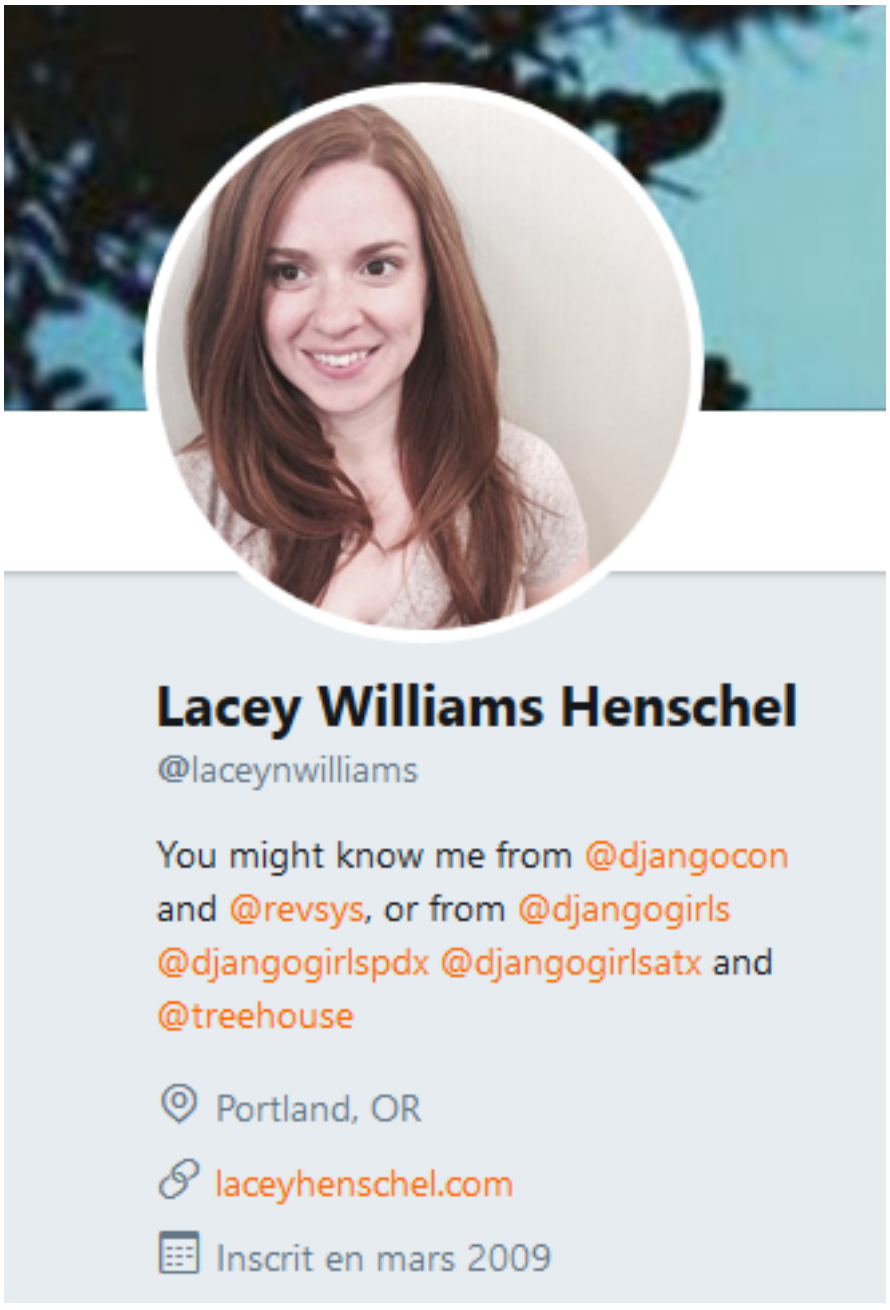


Fig. 9: https://twitter.com/laceynwilliams

4.7.1 Introduction

I'll be honest: I was pretty trepidatious about using Docker.

It wasn't something we used at my last job and most tutorials felt like this comic by Van Oktop.

Fig. 10: How to draw a horse

4.7.2 Annonce de l'écriture du tutoriel le 20 octobre 2017

Fig. 11: https://www.revsys.com/tidbits/brief-intro-docker-djangonauts/

4.7.3 Dockerfile Lacey Williams

```
FROM python:3.6

ENV PYTHONUNBUFFERED 1
ENV DJANGO_ENV dev
ENV DOCKER_CONTAINER 1

COPY ./requirements.txt /code/requirements.txt
RUN pip install -r /code/requirements.txt

COPY . /code/
WORKDIR /code/

EXPOSE 8000
```

4.7.3.1 FROM python:3.6

You don't need to create your Docker image from scratch. You can base your image off of code in another image in the Docker Hub, a repository of existing Docker images.

On this line, I've told Docker to base my image off of the Python 3.6 image, which (you guessed it) contains Python 3.6. Pointing to Python 3.6 versus 3.6.x ensures that we get the latest 3.6.x version, which will include bug fixes and security updates for that version of Python.

4.7.3.2 ENV PYTHONUNBUFFERED 1

ENV creates an environment variable called PYTHONUNBUFFERED and sets it to 1 (which, remember, is "truthy"). All together, this statement means that Docker won't buffer the output from your application; instead, you will get to see your output in your console the way you're used to.

4.7.3.3 ENV DJANGO_ENV dev

If you use multiple environment-based settings.py files, this creates an environment variable called DJANGO_ENV and sets it to the development environment.

You might call that "test" or "local" or something else.

4.7.3.4 ENV DOCKER_CONTAINER 1

This creates an environment variable called DOCKER_CONTAINER that you can use in settings.py to load different databases depending on whether you're running your application inside a Docker container.

4.7.3.5 EXPOSE 8000

In order to runserver like a champ, your Docker container will need access to port 8000. This bestows that access.

Huzzah! Your first Dockerfile is ready to go.

4.7.4 docker-compose.yml Lacey Williams

Docker Compose lets you run more than one container in a Docker application. It's especially useful if you want to have a database, like Postgres, running in a container alongside your web app. (Docker's overview of Compose is helpful.) Compose allows you to define several services that will make up your app and run them all together.

Examples of services you might define include:

- web: defines your web service
- db: your database
- redis or another caching service

Compose can also help you relate those services to each other. For example, you likely don't want your web service to start running until your db is ready, right?

Create a new file called **docker-compose.yml** in the same directory as your Dockerfile. While Dockerfile doesn't have an extension, the docker-compose file is written in YAML, so it has the extension .yml.

Mine defines two services, web and db, and looks like this:

```
version: '3'

services:
  db:
    image: postgres:9.6.5
    volumes:
      - postgres_data:/var/lib/postgresql/data/
  web:
    build: .
    command: python /code/manage.py migrate --noinput
    command: python /code/manage.py runserver 0.0.0.0:8000
    volumes:
      - .:/code
    ports:
```

(continues on next page)

(continued from previous page)

```
      - "8000:8000"
    depends_on:
      - db

volumes:
  postgres_data:
```

Just like we did with the Dockerfile, let's go through the parts of this docker-compose.yml file.

4.7.5 version: '3'

This line defines the version of Compose we want to use. We're using version 3, the most recent version.

4.7.6 services

Indented under this line, we will define the services we want our image to run in separate containers when we run our project.

4.7.6.1 db

```
db:
  image: postgres:9.6.5
  volumes:
    - postgres_data:/var/lib/postgresql/data/
```

This is where Compose gets exciting: this section sets up the db service as a Postgres database and instructs Compose to pull version 9.6.5 of Postgres from the image that already exists in Docker Hub. This means that I don't need to download Postgres on my computer at all in order to use it as my local database.

Upgrading Postgres from one minor version to another while keeping your data requires running some extra scripts, pgdump and pgrestore, and can get a little complicated. If you don't want to mess with this, set your Postgres image to a specific version (like 9.6.5). You will probably want to upgrade the Postgres version eventually, but this will save you from having to upgrade with every minor version release.

4.7.6.1.1 volumes

volumes tells Compose where in the container I would like it to store my data: in /var/lib/postgresql/data/.

Remember when I said that each container had its own set of subdirectories and that is why you needed to copy your application code into a directory named /code/? /var/ is one of those other subdirectories.

A **volume** also lets your data persist beyond the lifecycle of a specific container.

4.7.6.2 web

```
web:
        build: .
        command: python /code/manage.py migrate --noinput
        command: python /code/manage.py runserver 0.0.0.0:8000
        volumes:
          - .:/code
        ports:
          - "8000:8000"
        depends_on:
          - db
```

This section sets up the web service, the one that will run my application code.

4.7.6.2.1 build .

build: . tells Compose to build the image from the current directory.

4.7.6.2.2 command: python /code/manage.py migrate –noinput

command: python /code/manage.py migrate –noinput will automatically run migrations when I run the container and hide the output from me in the console.

4.7.6.2.3 command: python /code/manage.py runserver 0.0.0.0:8000

command: python /code/manage.py runserver 0.0.0.0:8000 will start the server when I run the container.

4.8 Docker: les bons réflexes à adopter par Paul MARS (MISC 95)

See also:

- https://www.miscmag.com/misc-n95-references-de-larticle-docker-les-bons-reflexes-a-adopter/
- https://www.miscmag.com/misc-n95-references-de-larticle-apercu-de-la-securite-de-docker/

Contents

4.8.1 Dockerfile MISC 95

```
FROM python:2.7-alpine  # usage d'une image de base du dépôt officiel
    LABEL description "Internal info on the challenge" version "0.1" # ajout
    d'informations à l'image pour pouvoir l'identifier plus facilement
    WORKDIR /opt/app/  # définition d'un dossier de travail pour l'exécution
    des instructions suivantes
    RUN addgroup -S ndh && adduser -S -g ndh ndh  # exécution d'une commande
    dans l'image
    USER ndh
    COPY requirements.txt /opt/app/  # copie de plusieurs ressources depuis
    l'hôte vers l'image
    COPY flag.txt /etc/x.b64
RUN pip install -r requirements.txt
    RUN rm requirements.txt
    COPY wsgi.py /opt/app/
    COPY cmd.sh /opt/app/
    COPY xml_challenge /opt/app/xml_challenge
    EXPOSE 8002  # définition de la liste des ports que les conteneurs
    instanciés sur l'image pourraient exposer
    CMD [ "/bin/sh", "cmd.sh" ] # définition de la commande qui sera
    lancée à l'instanciation d'un conteneur à partir de l'image
```

Vous aurez noté la présence d'une directive USER dans le Dockerfile précédent ainsi que de la création d'un utilisateur *ndh* quelques lignes plus haut. Par défaut, un processus lancé dans un conteneur s'exécute en tant que **root**. Vous vous doutez que ce comportement par défaut n'est pas une bonne pratique. Docker propose la directive USER permettant d'opérer le changement d'utilisateur.

Il faut simplement l'avoir créé avant dans le Dockerfile ou qu'il soit présent dans l'image sur laquelle se base la vôtre. Toutes les commandes exécutées au sein de l'image et du conteneur instancié sur cette image seront effectuées avec cet utilisateur après la directive. Pour chacun des services, il a été créé un utilisateur **ndh** dont les droits ont été modulés en fonction des besoins (besoin d'un shell ou non, droits sur certains fichiers). En pratique, cela a permis de donner un shell aux utilisateurs afin qu'ils récupèrent un drapeau sur le serveur sans qu'ils puissent le modifier ou changer l'environnement d'exécution du service.

La présence de secrets dans un Dockerfile ou un fichier docker-compose.yml.

4.8.2 Fichiers .env

Ces fichiers sont destinés à être versionnés et manipulés par plusieurs équipes. Docker dispose de fonctionnalités de gestion des secrets à travers la commande docker secrets (vous vous en doutiez, n'est-ce pas ?).

En parallèle de cette commande, une bonne pratique est de gérer les secrets par variable d'environnement et de passer ces variables à l'instanciation via la lecture d'un fichier de configuration.

4.9 Tutoriel Django step by step

See also:

- https://blog.devartis.com/django-development-with-docker-a-step-by-step-guide-525c0d08291

Contents

4.10 Tutoriel erroneousboat Docker Django

See also:

- https://github.com/erroneousboat/docker-django

Contents

4.10.1 tree

```
pvergain@uc026:/mnt/y/projects_id3/P5N001/XLOGCA135_tutorial_docker/tutorial_
↪docker/tutoriels/docker_django$ tree
```

```
.
├── circle.yml
├── config
│   └── environment
│       └── development.env
├── docker-compose.yml
├── docker_django.rst
├── LICENSE
├── README.md
├── services
│   └── webserver
│       ├── config
│       │   ├── localhost.crt
│       │   ├── localhost.key
│       │   ├── nginx.tmpl
│       │   └── start.sh
│       └── Dockerfile
└── webapp
        ├── config
        │   ├── database-check.py
        │   ├── django-uwsgi.ini
        │   ├── requirements.txt
        │   └── start.sh
        ├── Dockerfile
        └── starter
                ├── manage.py
                └── starter
                        ├── __init__.py
                        ├── settings.py
                        ├── urls.py
                        └── wsgi.py

9 directories, 21 files
```

4.10.2 docker-compose.yml

```
#####
# Docker compose YAML file
#
# For documentation see: https://docs.docker.com/compose/yml/
#####

version: "3"

volumes:
    static-files:


services:
    db:
        image: postgres:10.1
        volumes:
            - /opt/starter/psql:/var/lib/postgresql/data/pgdata
        env_file:
            - ./config/environment/development.env

    webserver:
        build:
            context: .
            dockerfile: services/webserver/Dockerfile
```

(continues on next page)

(continued from previous page)

```
        ports:
            - "80:80"
            - "443:443"
        depends_on:
            - webapp
        volumes:
            - static-files:/srv/static-files
        env_file:
            - ./config/environment/development.env

    webapp:
        build:
            context: webapp
        volumes:
            - ./webapp/starter:/srv/starter
            - static-files:/srv/static-files
        expose:
            - "8000"
        depends_on:
            - db
        env_file:
            - ./config/environment/development.env
```

4.11 Tutoriel Utilisation de pipenv avec Docker

See also:

- https://github.com/dfederschmidt/docker-pipenv-sample
- https://github.com/pypa/pipenv/blob/master/Dockerfile

Contents

4.11.1 Les fichiers

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_docker\tutoriels\pipenv>
↪dir
```

```
Le volume dans le lecteur Y n'a pas de nom.
Le numéro de série du volume est B2B7-2241

Répertoire de Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\pipenv

22/01/2018  10:39    <DIR>          .
22/01/2018  10:39    <DIR>          ..
22/01/2018  08:23               250 app.py
22/01/2018  10:11               438 Dockerfile
22/01/2018  10:39             8 130 pipenv.rst
22/01/2018  08:23               129 Pipfile
22/01/2018  08:23             2 580 Pipfile.lock
22/01/2018  08:23               415 Readme.md
6 fichier(s)           11 942 octets
2 Rép(s)  20 168 241 152 octets libres
```

4.11.2 Réécriture du fichier Dockerfile

See also:

- https://github.com/pypa/pipenv/blob/master/Dockerfile

On part de la recommendation officielle de Kenneth Reitz[11].

```
# https://github.com/pypa/pipenv/blob/master/Dockerfile
FROM python:3.6

# -- Install Pipenv:
RUN set -ex && pip install pipenv --upgrade

# -- Install Application into container:
RUN set -ex && mkdir /app

WORKDIR /app

# -- Adding Pipfiles
COPY Pipfile Pipfile
# COPY Pipfile.lock Pipfile.lock

# -- Install dependencies:
RUN set -ex && pipenv install --deploy --system

COPY app.py  /app

CMD ["python", "app.py"]

```

4.11.3 app.py

```
""" This is a very basic flask server"""
from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello():
```

(continues on next page)

[11] https://github.com/pypa/pipenv/blob/master/Dockerfile

(continued from previous page)

```
    """docstring"""
    return "Hello World!"


if __name__ == '__main__':
    app.run(host="0.0.0.0", debug = True)
```

4.11.4 docker build -t docker-pipenv-sample . : construction de l'image

```
C:/projects_id3/docker_projects/docker-pipenv-sample>docker build -t docker-pipenv-
↪sample .
```

```
Sending build context to Docker daemon  78.34kB
Step 1/8 : FROM python:3.6
3.6: Pulling from library/python
Digest: sha256:98149ed5f37f48ea3fad26ae6c0042dd2b08228d58edc95ef0fce35f1b3d9e9f
Status: Downloaded newer image for python:3.6
 ---> c1e459c00dc3
Step 2/8 : RUN set -ex && pip install pipenv --upgrade
 ---> Running in 21e4931d7ee4
+ pip install pipenv --upgrade
Collecting pipenv
  Downloading pipenv-9.0.3.tar.gz (3.9MB)
Collecting virtualenv (from pipenv)
  Downloading virtualenv-15.1.0-py2.py3-none-any.whl (1.8MB)
Collecting pew>=0.1.26 (from pipenv)
  Downloading pew-1.1.2-py2.py3-none-any.whl
Requirement already up-to-date: pip>=9.0.1 in /usr/local/lib/python3.6/site-
↪packages (from pipenv)
Collecting requests>2.18.0 (from pipenv)
  Downloading requests-2.18.4-py2.py3-none-any.whl (88kB)
Collecting flake8>=3.0.0 (from pipenv)
  Downloading flake8-3.5.0-py2.py3-none-any.whl (69kB)
Collecting urllib3>=1.21.1 (from pipenv)
  Downloading urllib3-1.22-py2.py3-none-any.whl (132kB)
Collecting virtualenv-clone>=0.2.5 (from pew>=0.1.26->pipenv)
  Downloading virtualenv-clone-0.2.6.tar.gz
Collecting setuptools>=17.1 (from pew>=0.1.26->pipenv)
  Downloading setuptools-38.4.0-py2.py3-none-any.whl (489kB)
Collecting certifi>=2017.4.17 (from requests>2.18.0->pipenv)
  Downloading certifi-2018.1.18-py2.py3-none-any.whl (151kB)
Collecting chardet<3.1.0,>=3.0.2 (from requests>2.18.0->pipenv)
  Downloading chardet-3.0.4-py2.py3-none-any.whl (133kB)
Collecting idna<2.7,>=2.5 (from requests>2.18.0->pipenv)
  Downloading idna-2.6-py2.py3-none-any.whl (56kB)
Collecting mccabe<0.7.0,>=0.6.0 (from flake8>=3.0.0->pipenv)
  Downloading mccabe-0.6.1-py2.py3-none-any.whl
Collecting pycodestyle<2.4.0,>=2.0.0 (from flake8>=3.0.0->pipenv)
  Downloading pycodestyle-2.3.1-py2.py3-none-any.whl (45kB)
Collecting pyflakes<1.7.0,>=1.5.0 (from flake8>=3.0.0->pipenv)
  Downloading pyflakes-1.6.0-py2.py3-none-any.whl (227kB)
Building wheels for collected packages: pipenv, virtualenv-clone
  Running setup.py bdist_wheel for pipenv: started
  Running setup.py bdist_wheel for pipenv: finished with status 'done'
  Stored in directory: /root/.cache/pip/wheels/78/cf/b7/
↪549d89ddbafb1cf3da825b97b730a7e1ac75602de9865d036e
  Running setup.py bdist_wheel for virtualenv-clone: started
  Running setup.py bdist_wheel for virtualenv-clone: finished with status 'done'
  Stored in directory: /root/.cache/pip/wheels/24/51/ef/
↪93120d304d240b4b6c2066454250a1626e04f73d34417b956d
```

(continues on next page)

(continued from previous page)

```
Successfully built pipenv virtualenv-clone
Installing collected packages: virtualenv, virtualenv-clone, setuptools, pew,␣
↪urllib3, certifi, chardet, idna, requests, mccabe, pycodestyle, pyflakes, flake8,
↪ pipenv
  Found existing installation: setuptools 38.2.4
        Uninstalling setuptools-38.2.4:
          Successfully uninstalled setuptools-38.2.4
Successfully installed certifi-2018.1.18 chardet-3.0.4 flake8-3.5.0 idna-2.6␣
↪mccabe-0.6.1 pew-1.1.2 pipenv-9.0.3 pycodestyle-2.3.1 pyflakes-1.6.0 requests-2.
↪18.4 setuptools-38.4.0 urllib3-1.22 virtualenv-15.1.0 virtualenv-clone-0.2.6
Removing intermediate container 21e4931d7ee4
 ---> 0b1272e6e1c6
Step 3/8 : RUN set -ex && mkdir /app
 ---> Running in 21153ac29a7f
+ mkdir /app
Removing intermediate container 21153ac29a7f
 ---> 1f95b3a89e78
Step 4/8 : WORKDIR /app
Removing intermediate container d235da053693
 ---> c40c0a57be56
Step 5/8 : COPY Pipfile Pipfile
 ---> 72c20255a55d
Step 6/8 : COPY Pipfile.lock Pipfile.lock
 ---> 7f022488626e
Step 7/8 : RUN set -ex && pipenv install --deploy --system
 ---> Running in 7535ac2a9610
+ pipenv install --deploy --system
Installing dependencies from Pipfile.lock (d3d473)...
Removing intermediate container 7535ac2a9610
 ---> 7366de78a2f1
Step 8/8 : COPY . /app
 ---> 5c977e084023
Successfully built 5c977e084023
Successfully tagged docker-pipenv-sample: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.
```

4.11.5 docker run -p 5000:5000 docker-pipenv-sample

```
C:/projects_id3/docker_projects/docker-pipenv-sample>docker run -p 5000:5000␣
↪docker-pipenv-sample
```

```
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 153-767-505
```

4.11.6 http://localhost:5000/

4.11.7 docker ps

```
Y:/projects_id3/P5N001/XLOGCA135_tutorial_docker/tutorial_docker/tutoriels/pipenv>
↪docker ps
```

```
CONTAINER ID        IMAGE                  COMMAND             CREATED
↪STATUS              PORTS                   NAMES
b9bf3fbbb859        docker-pipenv-sample   "python app.py"     4 minutes ago
↪Up 4 minutes        0.0.0.0:5000->5000/tcp   condescending_hypatia
```

4.11.8 docker exec -it 1a0a3dc7924d bash

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_docker\tutoriels\pipenv>
↪docker exec -it b9bf3fbbb859 bash
```

```
root@b9bf3fbbb859:/app# ls -als
```

```
4 drwxr-xr-x 1 root root 4096 Jan 22 09:44 .
4 drwxr-xr-x 1 root root 4096 Jan 22 09:45 ..
4 -rwxr-xr-x 1 root root  129 Jan 22 07:23 Pipfile
4 -rwxr-xr-x 1 root root 2580 Jan 22 07:23 Pipfile.lock
4 -rwxr-xr-x 1 root root  248 Jan 22 09:43 app.py
```

```
root@1a0a3dc7924d:/app# ps -ef | grep python
```

```
root          1      0  0 08:42 ?        00:00:00 python app.py
root          7      1  0 08:42 ?        00:00:10 /usr/local/bin/python app.py
```

4.11.9 docker rm 1a0a3dc7924d: suppression du conteneur à l'arrêt

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_docker\tutoriels\pipenv>
↪docker rm 1a0a3dc7924d
```

```
1a0a3dc7924d
```

4.11.10 docker rmi docker-pipenv-sample: suppression de l'image

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_docker\tutoriels\pipenv>
↪docker rmi docker-pipenv-sample
```

```
Untagged: docker-pipenv-sample:latest
Deleted: sha256:f7cb7fa32f377aa356791f7149f8f21b2b668e6ce5011dc338cb8ea7c58778b9
Deleted: sha256:91953983b1e474e3aff636101c4625d825c8a54044a7a44026d8a4a049efa5d7
Deleted: sha256:b08673d3c06b5d6c576e64d0c87f1d09d53355ae8f416d9e12b125bb78425721
```

4.12 Centos7

See also:

- http://www.codeghar.com/blog/install-latest-python-on-centos-7.html

Contents

4.12.1 Plan de travail

- récupérer une image centos:7
- yum update
- yum install -y https://centos7.iuscommunity.org/ius-release.rpm
- yum install -y python36u python36u-libs python36u-devel python36u-pip
- yum install which
- yum install openldap-devel

- pip3.6 install pipenv

4.12.2 yum update

```
[root@20c8bd8c86f4 intranet]# yum update
```

```
Loaded plugins: fastestmirror, ovl
Loading mirror speeds from cached hostfile
* base: ftp.pasteur.fr
* epel: pkg.adfinis-sygroup.ch
* extras: mirror.plusserver.com
* ius: mirror.slu.cz
* updates: ftp.ciril.fr
Resolving Dependencies
--> Running transaction check
---> Package bind-license.noarch 32:9.9.4-51.el7_4.1 will be updated
---> Package bind-license.noarch 32:9.9.4-51.el7_4.2 will be an update
---> Package binutils.x86_64 0:2.25.1-32.base.el7_4.1 will be updated
---> Package binutils.x86_64 0:2.25.1-32.base.el7_4.2 will be an update
---> Package epel-release.noarch 0:7-9 will be updated
---> Package epel-release.noarch 0:7-11 will be an update
---> Package kmod.x86_64 0:20-15.el7_4.6 will be updated
---> Package kmod.x86_64 0:20-15.el7_4.7 will be an update
---> Package kmod-libs.x86_64 0:20-15.el7_4.6 will be updated
---> Package kmod-libs.x86_64 0:20-15.el7_4.7 will be an update
---> Package kpartx.x86_64 0:0.4.9-111.el7 will be updated
---> Package kpartx.x86_64 0:0.4.9-111.el7_4.2 will be an update
---> Package libdb.x86_64 0:5.3.21-20.el7 will be updated
---> Package libdb.x86_64 0:5.3.21-21.el7_4 will be an update
---> Package libdb-utils.x86_64 0:5.3.21-20.el7 will be updated
---> Package libdb-utils.x86_64 0:5.3.21-21.el7_4 will be an update
---> Package systemd.x86_64 0:219-42.el7_4.4 will be updated
---> Package systemd.x86_64 0:219-42.el7_4.7 will be an update
---> Package systemd-libs.x86_64 0:219-42.el7_4.4 will be updated
---> Package systemd-libs.x86_64 0:219-42.el7_4.7 will be an update
---> Package tzdata.noarch 0:2017c-1.el7 will be updated
---> Package tzdata.noarch 0:2018c-1.el7 will be an update
---> Package yum.noarch 0:3.4.3-154.el7.centos will be updated
---> Package yum.noarch 0:3.4.3-154.el7.centos.1 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
Package                                Arch                           Version
 ↪                                     Repository                       Size
================================================================================
Updating:
bind-license                           noarch                         32:9.9.4-51.
 ↪el7_4.2                               updates                          84 k
binutils                               x86_64                         2.25.1-32.
 ↪base.el7_4.2                          updates                         5.4 M
epel-release                           noarch                         7-11
 ↪                                     epel                             15 k
kmod                                   x86_64                         20-15.el7_4.
 ↪7                                     updates                         121 k
kmod-libs                              x86_64                         20-15.el7_4.
 ↪7                                     updates                          50 k
kpartx                                 x86_64                         0.4.9-111.
 ↪el7_4.2                               updates                          73 k
libdb                                  x86_64                         5.3.21-21.
 ↪el7_4                                 updates                         719 k
```

(continues on next page)

(continued from previous page)

```
libdb-utils                                x86_64                                5.3.21-21.
↪el7_4                                        updates                                    132 k
systemd                                    x86_64                                219-42.el7_4.
↪7                                         updates                                 5.2 M
systemd-libs                               x86_64                                219-42.el7_4.
↪7                                         updates                                 376 k
tzdata                                     noarch                                2018c-1.el7
↪                                          updates                                 479 k
yum                                        noarch                                3.4.3-154.
↪el7.centos.1                                 updates                                 1.2 M

Transaction Summary
================================================================================================
Upgrade  12 Packages

Total download size: 14 M
Is this ok [y/d/N]: y
Downloading packages:
Delta RPMs disabled because /usr/bin/applydeltarpm not installed.
(1/12): bind-license-9.9.4-51.el7_4.2.noarch.rpm
↪                                                          |  84 kB  00:00:00
(2/12): kmod-libs-20-15.el7_4.7.x86_64.rpm
↪                                                          |  50 kB  00:00:00
(3/12): kmod-20-15.el7_4.7.x86_64.rpm
↪                                                          | 121 kB  00:00:00
warning: /var/cache/yum/x86_64/7/epel/packages/epel-release-7-11.noarch.rpm:
↪Header V3 RSA/SHA256 Signature, key ID 352c64e5: NOKEY
Public key for epel-release-7-11.noarch.rpm is not installed
(4/12): epel-release-7-11.noarch.rpm
↪                                                          |  15 kB  00:00:00
(5/12): libdb-utils-5.3.21-21.el7_4.x86_64.rpm
↪                                                          | 132 kB  00:00:00
(6/12): kpartx-0.4.9-111.el7_4.2.x86_64.rpm
↪                                                          |  73 kB  00:00:00
(7/12): libdb-5.3.21-21.el7_4.x86_64.rpm
↪                                                          | 719 kB  00:00:01
(8/12): tzdata-2018c-1.el7.noarch.rpm
↪                                                          | 479 kB  00:00:01
(9/12): systemd-libs-219-42.el7_4.7.x86_64.rpm
↪                                                          | 376 kB  00:00:02
(10/12): yum-3.4.3-154.el7.centos.1.noarch.rpm
↪                                                          | 1.2 MB  00:00:03
(11/12): binutils-2.25.1-32.base.el7_4.2.x86_64.rpm
↪                                                          | 5.4 MB  00:00:10
(12/12): systemd-219-42.el7_4.7.x86_64.rpm
↪                                                          | 5.2 MB  00:00:10
------------------------------------------------------------------------------
↪------------------------------------------------------------------------------
Total
↪                                              1.2 MB/s |  14 MB  00:00:11
Retrieving key from file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7
Importing GPG key 0x352C64E5:
Userid     : "Fedora EPEL (7) <epel@fedoraproject.org>"
Fingerprint: 91e9 7d7c 4a5e 96f1 7f3e 888f 6a2f aea2 352c 64e5
Package    : epel-release-7-9.noarch (@extras)
From       : /etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7
Is this ok [y/N]: y
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
```

(continues on next page)

(continued from previous page)

```
Updating   : libdb-5.3.21-21.el7_4.x86_64                                  ⌙
↪                                                                    1/24
Updating   : binutils-2.25.1-32.base.el7_4.2.x86_64                        ⌙
↪                                                                    2/24
Updating   : kmod-20-15.el7_4.7.x86_64                                     ⌙
↪                                                                    3/24
Updating   : systemd-libs-219-42.el7_4.7.x86_64                            ⌙
↪                                                                    4/24
Updating   : kmod-libs-20-15.el7_4.7.x86_64                                ⌙
↪                                                                    5/24
Updating   : systemd-219-42.el7_4.7.x86_64                                 ⌙
↪                                                                    6/24
Updating   : libdb-utils-5.3.21-21.el7_4.x86_64                            ⌙
↪                                                                    7/24
Updating   : yum-3.4.3-154.el7.centos.1.noarch                             ⌙
↪                                                                    8/24
Updating   : 32:bind-license-9.9.4-51.el7_4.2.noarch                       ⌙
↪                                                                    9/24
Updating   : tzdata-2018c-1.el7.noarch                                     ⌙
↪                                                                   10/24
Updating   : kpartx-0.4.9-111.el7_4.2.x86_64                               ⌙
↪                                                                   11/24
Updating   : epel-release-7-11.noarch                                      ⌙
↪                                                                   12/24
Cleanup    : systemd-219-42.el7_4.4.x86_64                                 ⌙
↪                                                                   13/24
Cleanup    : kmod-20-15.el7_4.6.x86_64                                     ⌙
↪                                                                   14/24
Cleanup    : libdb-utils-5.3.21-20.el7.x86_64                              ⌙
↪                                                                   15/24
Cleanup    : yum-3.4.3-154.el7.centos.noarch                               ⌙
↪                                                                   16/24
Cleanup    : 32:bind-license-9.9.4-51.el7_4.1.noarch                       ⌙
↪                                                                   17/24
Cleanup    : tzdata-2017c-1.el7.noarch                                     ⌙
↪                                                                   18/24
Cleanup    : epel-release-7-9.noarch                                       ⌙
↪                                                                   19/24
Cleanup    : libdb-5.3.21-20.el7.x86_64                                    ⌙
↪                                                                   20/24
Cleanup    : binutils-2.25.1-32.base.el7_4.1.x86_64                        ⌙
↪                                                                   21/24
Cleanup    : kmod-libs-20-15.el7_4.6.x86_64                                ⌙
↪                                                                   22/24
Cleanup    : systemd-libs-219-42.el7_4.4.x86_64                            ⌙
↪                                                                   23/24
Cleanup    : kpartx-0.4.9-111.el7.x86_64                                   ⌙
↪                                                                   24/24
Verifying  : kmod-20-15.el7_4.7.x86_64                                     ⌙
↪                                                                    1/24
Verifying  : kmod-libs-20-15.el7_4.7.x86_64                                ⌙
↪                                                                    2/24
Verifying  : libdb-utils-5.3.21-21.el7_4.x86_64                            ⌙
↪                                                                    3/24
Verifying  : systemd-219-42.el7_4.7.x86_64                                 ⌙
↪                                                                    4/24
Verifying  : epel-release-7-11.noarch                                      ⌙
↪                                                                    5/24
Verifying  : kpartx-0.4.9-111.el7_4.2.x86_64                               ⌙
↪                                                                    6/24
Verifying  : tzdata-2018c-1.el7.noarch                                     ⌙
↪                                                                    7/24
```

(continues on next page)

(continued from previous page)

```
Verifying  : 32:bind-license-9.9.4-51.el7_4.2.noarch                          ⌙
↪                                                                        8/24
Verifying  : systemd-libs-219-42.el7_4.7.x86_64                               ⌙
↪                                                                        9/24
Verifying  : binutils-2.25.1-32.base.el7_4.2.x86_64                           ⌙
↪                                                                       10/24
Verifying  : libdb-5.3.21-21.el7_4.x86_64                                     ⌙
↪                                                                       11/24
Verifying  : yum-3.4.3-154.el7.centos.1.noarch                                ⌙
↪                                                                       12/24
Verifying  : epel-release-7-9.noarch                                          ⌙
↪                                                                       13/24
Verifying  : binutils-2.25.1-32.base.el7_4.1.x86_64                           ⌙
↪                                                                       14/24
Verifying  : 32:bind-license-9.9.4-51.el7_4.1.noarch                          ⌙
↪                                                                       15/24
Verifying  : systemd-libs-219-42.el7_4.4.x86_64                               ⌙
↪                                                                       16/24
Verifying  : kmod-20-15.el7_4.6.x86_64                                        ⌙
↪                                                                       17/24
Verifying  : systemd-219-42.el7_4.4.x86_64                                    ⌙
↪                                                                       18/24
Verifying  : libdb-utils-5.3.21-20.el7.x86_64                                 ⌙
↪                                                                       19/24
Verifying  : kmod-libs-20-15.el7_4.6.x86_64                                   ⌙
↪                                                                       20/24
Verifying  : tzdata-2017c-1.el7.noarch                                        ⌙
↪                                                                       21/24
Verifying  : kpartx-0.4.9-111.el7.x86_64                                      ⌙
↪                                                                       22/24
Verifying  : yum-3.4.3-154.el7.centos.noarch                                  ⌙
↪                                                                       23/24
Verifying  : libdb-5.3.21-20.el7.x86_64                                       ⌙
↪                                                                       24/24

Updated:
bind-license.noarch 32:9.9.4-51.el7_4.2   binutils.x86_64 0:2.25.1-32.base.el7_4.2⌙
↪  epel-release.noarch 0:7-11       kmod.x86_64 0:20-15.el7_4.7
kmod-libs.x86_64 0:20-15.el7_4.7          kpartx.x86_64 0:0.4.9-111.el7_4.2       ⌙
↪  libdb.x86_64 0:5.3.21-21.el7_4   libdb-utils.x86_64 0:5.3.21-21.el7_4
systemd.x86_64 0:219-42.el7_4.7           systemd-libs.x86_64 0:219-42.el7_4.7    ⌙
↪  tzdata.noarch 0:2018c-1.el7      yum.noarch 0:3.4.3-154.el7.centos.1

Complete!
[root@20c8bd8c86f4 intranet]#
```

4.12.3 yum install -y https://centos7.iuscommunity.org/ius-release.rpm

```
[root@20c8bd8c86f4 /]# yum install -y https://centos7.iuscommunity.org/ius-release.
↪rpm
```

```
Loaded plugins: fastestmirror, ovl
ius-release.rpm                                                               ⌙
↪                                                  | 8.1 kB  00:00:00
Examining /var/tmp/yum-root-KswZN7/ius-release.rpm: ius-release-1.0-15.ius.centos7.
↪noarch
Marking /var/tmp/yum-root-KswZN7/ius-release.rpm to be installed
Resolving Dependencies
--> Running transaction check
```

(continues on next page)

(continued from previous page)

```
---> Package ius-release.noarch 0:1.0-15.ius.centos7 will be installed
--> Processing Dependency: epel-release = 7 for package: ius-release-1.0-15.ius.
↪centos7.noarch
base
↪                                                          | 3.6 kB  00:00:00
extras
↪                                                          | 3.4 kB  00:00:00
updates
↪                                                          | 3.4 kB  00:00:00
(1/4): extras/7/x86_64/primary_db
↪                                                          | 166 kB  00:00:00
(2/4): base/7/x86_64/group_gz
↪                                                          | 156 kB  00:00:01
(3/4): updates/7/x86_64/primary_db
↪                                                          | 6.0 MB  00:00:04
(4/4): base/7/x86_64/primary_db
↪                                                          | 5.7 MB  00:00:14
Determining fastest mirrors
* base: ftp.pasteur.fr
* extras: mirror.plusserver.com
* updates: ftp.ciril.fr
--> Running transaction check
---> Package epel-release.noarch 0:7-9 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
Package                          Arch                          Version
↪                              Repository                        Size
================================================================================
Installing:
ius-release                      noarch                        1.0-15.ius.
↪centos7                       /ius-release                      8.5 k
Installing for dependencies:
epel-release                     noarch                        7-9
↪                              extras                            14 k

Transaction Summary
================================================================================
Install  1 Package (+1 Dependent package)

Total size: 23 k
Total download size: 14 k
Installed size: 33 k
Downloading packages:
warning: /var/cache/yum/x86_64/7/extras/packages/epel-release-7-9.noarch.rpm:
↪Header V3 RSA/SHA256 Signature, key ID f4a80eb5: NOKEYB/s |    0 B  --:--:-- ETA
Public key for epel-release-7-9.noarch.rpm is not installed
epel-release-7-9.noarch
↪                                                          |  14 kB  00:00:00
Retrieving key from file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
Importing GPG key 0xF4A80EB5:
Userid     : "CentOS-7 Key (CentOS 7 Official Signing Key) <security@centos.org>"
Fingerprint: 6341 ab27 53d7 8a78 a7c2 7bb1 24c6 a8a7 f4a8 0eb5
Package    : centos-release-7-4.1708.el7.centos.x86_64 (@CentOS)
From       : /etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
```

(continues on next page)

(continued from previous page)

```
Installing : epel-release-7-9.noarch                                          1/2
Installing : ius-release-1.0-15.ius.centos7.noarch                            2/2
Verifying  : ius-release-1.0-15.ius.centos7.noarch                            1/2
Verifying  : epel-release-7-9.noarch                                          2/2

Installed:
ius-release.noarch 0:1.0-15.ius.centos7

Dependency Installed:
epel-release.noarch 0:7-9

Complete!
```

4.12.4 yum install -y python36u python36u-libs python36u-devel python36u-pip

```
[root@20c8bd8c86f4 /]# yum install -y python36u python36u-libs python36u-devel
↪python36u-pip
```

```
Loaded plugins: fastestmirror, ovl
epel/x86_64/metalink                                      |  26 kB  00:00:00
epel                                                      | 4.7 kB  00:00:00
ius                                                       | 2.3 kB  00:00:00
(1/4): epel/x86_64/group_gz                               | 266 kB  00:00:01
(2/4): ius/x86_64/primary_db                              | 212 kB  00:00:01
(3/4): epel/x86_64/primary_db                             | 6.2 MB  00:00:05
(4/4): epel/x86_64/updateinfo                             | 880 kB  00:00:06
Loading mirror speeds from cached hostfile
* base: ftp.pasteur.fr
* epel: ftp-stud.hs-esslingen.de
* extras: mirror.plusserver.com
* ius: mirror.team-cymru.org
* updates: ftp.ciril.fr
Resolving Dependencies
--> Running transaction check
---> Package python36u.x86_64 0:3.6.4-1.ius.centos7 will be installed
---> Package python36u-devel.x86_64 0:3.6.4-1.ius.centos7 will be installed
---> Package python36u-libs.x86_64 0:3.6.4-1.ius.centos7 will be installed
---> Package python36u-pip.noarch 0:9.0.1-1.ius.centos7 will be installed
--> Processing Dependency: python36u-setuptools for package: python36u-pip-9.0.1-1.
↪ius.centos7.noarch
--> Running transaction check
---> Package python36u-setuptools.noarch 0:36.6.0-1.ius.centos7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
Package                                Arch
↪Version                                    Repository                    Size
```

(continues on next page)

(continued from previous page)

```
================================================================================
Installing:
python36u                                 x86_64                        3.6.4-
 ↪1.ius.centos7                          ius                            56 k
python36u-devel                           x86_64                        3.6.4-
 ↪1.ius.centos7                          ius                           839 k
python36u-libs                            x86_64                        3.6.4-
 ↪1.ius.centos7                          ius                           8.7 M
python36u-pip                             noarch                        9.0.1-
 ↪1.ius.centos7                          ius                           1.8 M
Installing for dependencies:
python36u-setuptools                      noarch                        36.6.0-
 ↪1.ius.centos7                          ius                           587 k

Transaction Summary
================================================================================
Install  4 Packages (+1 Dependent package)

Total download size: 12 M
Installed size: 53 M
Downloading packages:
warning: /var/cache/yum/x86_64/7/ius/packages/python36u-3.6.4-1.ius.centos7.x86_64.
 ↪rpm: Header V4 DSA/SHA1 Signature, key ID 9cd4953f: NOKEY2 kB  --:--:-- ETA
Public key for python36u-3.6.4-1.ius.centos7.x86_64.rpm is not installed
(1/5): python36u-3.6.4-1.ius.centos7.x86_64.rpm
 ↪                                                    |  56 kB  00:00:00
(2/5): python36u-setuptools-36.6.0-1.ius.centos7.noarch.rpm
 ↪                                                    | 587 kB  00:00:03
(3/5): python36u-pip-9.0.1-1.ius.centos7.noarch.rpm
 ↪                                                    | 1.8 MB  00:00:03
(4/5): python36u-devel-3.6.4-1.ius.centos7.x86_64.rpm
 ↪                                                    | 839 kB  00:00:06
(5/5): python36u-libs-3.6.4-1.ius.centos7.x86_64.rpm
 ↪                                                    | 8.7 MB  00:00:28
--------------------------------------------------------------------------------
 ↪------------------------------------------------------------------------------
Total
 ↪                                             432 kB/s |  12 MB  00:00:28
Retrieving key from file:///etc/pki/rpm-gpg/IUS-COMMUNITY-GPG-KEY
Importing GPG key 0x9CD4953F:
Userid     : "IUS Community Project <coredev@iuscommunity.org>"
Fingerprint: 8b84 6e3a b3fe 6462 74e8 670f da22 1cdf 9cd4 953f
Package    : ius-release-1.0-15.ius.centos7.noarch (installed)
From       : /etc/pki/rpm-gpg/IUS-COMMUNITY-GPG-KEY
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : python36u-libs-3.6.4-1.ius.centos7.x86_64
 ↪                                                                      1/5
Installing : python36u-3.6.4-1.ius.centos7.x86_64
 ↪                                                                      2/5
Installing : python36u-setuptools-36.6.0-1.ius.centos7.noarch
 ↪                                                                      3/5
Installing : python36u-pip-9.0.1-1.ius.centos7.noarch
 ↪                                                                      4/5
Installing : python36u-devel-3.6.4-1.ius.centos7.x86_64
 ↪                                                                      5/5
Verifying  : python36u-setuptools-36.6.0-1.ius.centos7.noarch
 ↪                                                                      1/5
Verifying  : python36u-pip-9.0.1-1.ius.centos7.noarch
 ↪                                                                      2/5
```

(continues on next page)

(continued from previous page)

```
Verifying  : python36u-3.6.4-1.ius.centos7.x86_64                                 ⌴
↪                                                                          3/5
Verifying  : python36u-libs-3.6.4-1.ius.centos7.x86_64                            ⌴
↪                                                                          4/5
Verifying  : python36u-devel-3.6.4-1.ius.centos7.x86_64                           ⌴
↪                                                                          5/5

Installed:
python36u.x86_64 0:3.6.4-1.ius.centos7             python36u-devel.x86_64 0:3.6.4-
↪1.ius.centos7         python36u-libs.x86_64 0:3.6.4-1.ius.centos7
python36u-pip.noarch 0:9.0.1-1.ius.centos7

Dependency Installed:
python36u-setuptools.noarch 0:36.6.0-1.ius.centos7

Complete!
[root@20c8bd8c86f4 /]#
```

4.12.5 python3.6

```
[root@20c8bd8c86f4 /]# python3.6
Python 3.6.4 (default, Dec 19 2017, 14:48:12)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-16)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

4.12.6 yum install which

```
[root@20c8bd8c86f4 /]# yum install which
```

```
Loaded plugins: fastestmirror, ovl
Loading mirror speeds from cached hostfile
 * base: ftp.pasteur.fr
 * epel: repo.boun.edu.tr
 * extras: mirror.plusserver.com
 * ius: mirror.its.dal.ca
 * updates: ftp.ciril.fr
Resolving Dependencies
--> Running transaction check
---> Package which.x86_64 0:2.20-7.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package                          Arch                            Version      ⌴
↪                                 Repository                     Size
================================================================================
Installing:
 which                            x86_64                          2.20-7.
↪el7                                base                           41 k

Transaction Summary
================================================================================
Install  1 Package

Total download size: 41 k
```

(continues on next page)

(continued from previous page)

```
Installed size: 75 k
Is this ok [y/d/N]: y
Downloading packages:
which-2.20-7.el7.x86_64.rpm
↪                                                          |  41 kB  00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : which-2.20-7.el7.x86_64
↪                                                                        1/1
install-info: No such file or directory for /usr/share/info/which.info.gz
  Verifying  : which-2.20-7.el7.x86_64
↪                                                                        1/1

Installed:
  which.x86_64 0:2.20-7.el7

Complete!
[root@20c8bd8c86f4 /]# which python3.6
/usr/bin/python3.6

[root@20c8bd8c86f4 /]# which python3.6
/usr/bin/python3.6
```

4.12.7 which pip3.6

```
[root@20c8bd8c86f4 /]# which pip3.6
```

```
/usr/bin/pip3.6
```

```
[root@20c8bd8c86f4 /]# pip3.6 install pipenv
```

```
Collecting pipenv
Downloading pipenv-9.0.3.tar.gz (3.9MB)
100% |################################| 3.9MB 291kB/s
Collecting virtualenv (from pipenv)
Downloading virtualenv-15.1.0-py2.py3-none-any.whl (1.8MB)
100% |################################| 1.8MB 610kB/s
Collecting pew>=0.1.26 (from pipenv)
Downloading pew-1.1.2-py2.py3-none-any.whl
Requirement already satisfied: pip>=9.0.1 in /usr/lib/python3.6/site-packages
↪(from pipenv)
Collecting requests>2.18.0 (from pipenv)
Downloading requests-2.18.4-py2.py3-none-any.whl (88kB)
100% |################################| 92kB 1.1MB/s
Collecting flake8>=3.0.0 (from pipenv)
Downloading flake8-3.5.0-py2.py3-none-any.whl (69kB)
100% |################################| 71kB 2.8MB/s
Collecting urllib3>=1.21.1 (from pipenv)
Downloading urllib3-1.22-py2.py3-none-any.whl (132kB)
100% |################################| 133kB 2.0MB/s
Requirement already satisfied: setuptools>=17.1 in /usr/lib/python3.6/site-
↪packages (from pew>=0.1.26->pipenv)
Collecting virtualenv-clone>=0.2.5 (from pew>=0.1.26->pipenv)
Downloading virtualenv-clone-0.2.6.tar.gz
Collecting certifi>=2017.4.17 (from requests>2.18.0->pipenv)
Downloading certifi-2018.1.18-py2.py3-none-any.whl (151kB)
```

(continues on next page)

(continued from previous page)

```
100% |################################| 153kB 1.0MB/s
Collecting chardet<3.1.0,>=3.0.2 (from requests>2.18.0->pipenv)
Downloading chardet-3.0.4-py2.py3-none-any.whl (133kB)
100% |################################| 143kB 2.4MB/s
Collecting idna<2.7,>=2.5 (from requests>2.18.0->pipenv)
Downloading idna-2.6-py2.py3-none-any.whl (56kB)
100% |################################| 61kB 920kB/s
Collecting mccabe<0.7.0,>=0.6.0 (from flake8>=3.0.0->pipenv)
Downloading mccabe-0.6.1-py2.py3-none-any.whl
Collecting pycodestyle<2.4.0,>=2.0.0 (from flake8>=3.0.0->pipenv)
Downloading pycodestyle-2.3.1-py2.py3-none-any.whl (45kB)
100% |################################| 51kB 2.2MB/s
Collecting pyflakes<1.7.0,>=1.5.0 (from flake8>=3.0.0->pipenv)
Downloading pyflakes-1.6.0-py2.py3-none-any.whl (227kB)
100% |################################| 235kB 2.3MB/s
Installing collected packages: virtualenv, virtualenv-clone, pew, certifi, urllib3,
↪ chardet, idna, requests, mccabe, pycodestyle, pyflakes, flake8, pipenv
Running setup.py install for virtualenv-clone ... done
Running setup.py install for pipenv ... done
Successfully installed certifi-2018.1.18 chardet-3.0.4 flake8-3.5.0 idna-2.6␣
↪mccabe-0.6.1 pew-1.1.2 pipenv-9.0.3 pycodestyle-2.3.1 pyflakes-1.6.0 requests-2.
↪18.4 urllib3-1.22 virtualenv-15.1.0 virtualenv-clone-0.2.6
```

```
(activate) [root@20c8bd8c86f4 intranet]# pip install django
Collecting django
  Downloading Django-2.0.2-py3-none-any.whl (7.1MB)
        100% |################################| 7.1MB 205kB/s
Collecting pytz (from django)
  Downloading pytz-2017.3-py2.py3-none-any.whl (511kB)
        100% |################################| 512kB 1.5MB/s
Installing collected packages: pytz, django
Successfully installed django-2.0.2 pytz-2017.3
```

4.12.8 docker build -t id3centos7:1 .

```
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\centos7> docker build -t id3centos7:1 .
```

```
Sending build context to Docker daemon  37.38kB
Step 1/5 : FROM centos:7
---> ff426288ea90
Step 2/5 : RUN yum update -y
---> Running in bd9bc627aeeb
Loaded plugins: fastestmirror, ovl
Determining fastest mirrors
* base: centos.quelquesmots.fr
* extras: fr.mirror.babylon.network
* updates: fr.mirror.babylon.network
Resolving Dependencies
--> Running transaction check
---> Package bind-license.noarch 32:9.9.4-51.el7_4.1 will be updated
---> Package bind-license.noarch 32:9.9.4-51.el7_4.2 will be an update
---> Package binutils.x86_64 0:2.25.1-32.base.el7_4.1 will be updated
---> Package binutils.x86_64 0:2.25.1-32.base.el7_4.2 will be an update
---> Package kmod.x86_64 0:20-15.el7_4.6 will be updated
---> Package kmod.x86_64 0:20-15.el7_4.7 will be an update
---> Package kmod-libs.x86_64 0:20-15.el7_4.6 will be updated
---> Package kmod-libs.x86_64 0:20-15.el7_4.7 will be an update
```

(continues on next page)

(continued from previous page)

```
---> Package kpartx.x86_64 0:0.4.9-111.el7 will be updated
---> Package kpartx.x86_64 0:0.4.9-111.el7_4.2 will be an update
---> Package libdb.x86_64 0:5.3.21-20.el7 will be updated
---> Package libdb.x86_64 0:5.3.21-21.el7_4 will be an update
---> Package libdb-utils.x86_64 0:5.3.21-20.el7 will be updated
---> Package libdb-utils.x86_64 0:5.3.21-21.el7_4 will be an update
---> Package systemd.x86_64 0:219-42.el7_4.4 will be updated
---> Package systemd.x86_64 0:219-42.el7_4.7 will be an update
---> Package systemd-libs.x86_64 0:219-42.el7_4.4 will be updated
---> Package systemd-libs.x86_64 0:219-42.el7_4.7 will be an update
---> Package tzdata.noarch 0:2017c-1.el7 will be updated
---> Package tzdata.noarch 0:2018c-1.el7 will be an update
---> Package yum.noarch 0:3.4.3-154.el7.centos will be updated
---> Package yum.noarch 0:3.4.3-154.el7.centos.1 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
Package            Arch        Version                          Repository    Size
================================================================================
Updating:
bind-license       noarch      32:9.9.4-51.el7_4.2              updates       84 k
binutils           x86_64      2.25.1-32.base.el7_4.2           updates      5.4 M
kmod               x86_64      20-15.el7_4.7                    updates      121 k
kmod-libs          x86_64      20-15.el7_4.7                    updates       50 k
kpartx             x86_64      0.4.9-111.el7_4.2                updates       73 k
libdb              x86_64      5.3.21-21.el7_4                  updates      719 k
libdb-utils        x86_64      5.3.21-21.el7_4                  updates      132 k
systemd            x86_64      219-42.el7_4.7                   updates      5.2 M
systemd-libs       x86_64      219-42.el7_4.7                   updates      376 k
tzdata             noarch      2018c-1.el7                      updates      479 k
yum                noarch      3.4.3-154.el7.centos.1           updates      1.2 M

Transaction Summary
================================================================================
Upgrade  11 Packages

Total download size: 14 M
Downloading packages:
Delta RPMs disabled because /usr/bin/applydeltarpm not installed.
warning: /var/cache/yum/x86_64/7/updates/packages/kmod-libs-20-15.el7_4.7.x86_64.
↪rpm: Header V3 RSA/SHA256 Signature, key ID f4a80eb5: NOKEY
Public key for kmod-libs-20-15.el7_4.7.x86_64.rpm is not installed
--------------------------------------------------------------------------------
Total                                           1.6 MB/s |  14 MB  00:08
Retrieving key from file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
Importing GPG key 0xF4A80EB5:
Userid     : "CentOS-7 Key (CentOS 7 Official Signing Key) <security@centos.org>"
Fingerprint: 6341 ab27 53d7 8a78 a7c2 7bb1 24c6 a8a7 f4a8 0eb5
Package    : centos-release-7-4.1708.el7.centos.x86_64 (@CentOS)
From       : /etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Updating   : libdb-5.3.21-21.el7_4.x86_64                                 1/22
Updating   : binutils-2.25.1-32.base.el7_4.2.x86_64                       2/22
Updating   : kmod-20-15.el7_4.7.x86_64                                    3/22
Updating   : systemd-libs-219-42.el7_4.7.x86_64                           4/22
Updating   : kmod-libs-20-15.el7_4.7.x86_64                               5/22
```

(continues on next page)

(continued from previous page)

```
Updating   : systemd-219-42.el7_4.7.x86_64                              6/22
Updating   : libdb-utils-5.3.21-21.el7_4.x86_64                         7/22
Updating   : yum-3.4.3-154.el7.centos.1.noarch                          8/22
Updating   : 32:bind-license-9.9.4-51.el7_4.2.noarch                    9/22
Updating   : tzdata-2018c-1.el7.noarch                                 10/22
Updating   : kpartx-0.4.9-111.el7_4.2.x86_64                           11/22
Cleanup    : systemd-219-42.el7_4.4.x86_64                             12/22
Cleanup    : kmod-20-15.el7_4.6.x86_64                                 13/22
Cleanup    : libdb-utils-5.3.21-20.el7.x86_64                          14/22
Cleanup    : yum-3.4.3-154.el7.centos.noarch                           15/22
Cleanup    : 32:bind-license-9.9.4-51.el7_4.1.noarch                   16/22
Cleanup    : tzdata-2017c-1.el7.noarch                                 17/22
Cleanup    : libdb-5.3.21-20.el7.x86_64                                18/22
Cleanup    : binutils-2.25.1-32.base.el7_4.1.x86_64                    19/22
Cleanup    : kmod-libs-20-15.el7_4.6.x86_64                            20/22
Cleanup    : systemd-libs-219-42.el7_4.4.x86_64                        21/22
Cleanup    : kpartx-0.4.9-111.el7.x86_64                               22/22
Verifying  : kmod-20-15.el7_4.7.x86_64                                  1/22
Verifying  : kmod-libs-20-15.el7_4.7.x86_64                             2/22
Verifying  : libdb-utils-5.3.21-21.el7_4.x86_64                         3/22
Verifying  : systemd-219-42.el7_4.7.x86_64                              4/22
Verifying  : kpartx-0.4.9-111.el7_4.2.x86_64                            5/22
Verifying  : tzdata-2018c-1.el7.noarch                                  6/22
Verifying  : 32:bind-license-9.9.4-51.el7_4.2.noarch                    7/22
Verifying  : systemd-libs-219-42.el7_4.7.x86_64                         8/22
Verifying  : binutils-2.25.1-32.base.el7_4.2.x86_64                     9/22
Verifying  : libdb-5.3.21-21.el7_4.x86_64                              10/22
Verifying  : yum-3.4.3-154.el7.centos.1.noarch                         11/22
Verifying  : binutils-2.25.1-32.base.el7_4.1.x86_64                    12/22
Verifying  : 32:bind-license-9.9.4-51.el7_4.1.noarch                   13/22
Verifying  : systemd-libs-219-42.el7_4.4.x86_64                        14/22
Verifying  : kmod-20-15.el7_4.6.x86_64                                 15/22
Verifying  : systemd-219-42.el7_4.4.x86_64                             16/22
Verifying  : libdb-utils-5.3.21-20.el7.x86_64                          17/22
Verifying  : kmod-libs-20-15.el7_4.6.x86_64                            18/22
Verifying  : tzdata-2017c-1.el7.noarch                                 19/22
Verifying  : kpartx-0.4.9-111.el7.x86_64                               20/22
Verifying  : yum-3.4.3-154.el7.centos.noarch                           21/22
Verifying  : libdb-5.3.21-20.el7.x86_64                                22/22

Updated:
bind-license.noarch 32:9.9.4-51.el7_4.2
binutils.x86_64 0:2.25.1-32.base.el7_4.2
kmod.x86_64 0:20-15.el7_4.7
kmod-libs.x86_64 0:20-15.el7_4.7
kpartx.x86_64 0:0.4.9-111.el7_4.2
libdb.x86_64 0:5.3.21-21.el7_4
libdb-utils.x86_64 0:5.3.21-21.el7_4
systemd.x86_64 0:219-42.el7_4.7
systemd-libs.x86_64 0:219-42.el7_4.7
tzdata.noarch 0:2018c-1.el7
yum.noarch 0:3.4.3-154.el7.centos.1

Complete!
Removing intermediate container bd9bc627aeeb
---> 90814f4b95d5
Step 3/5 : RUN yum install -y https://centos7.iuscommunity.org/ius-release.rpm
---> Running in cea6a40470fa
Loaded plugins: fastestmirror, ovl
Examining /var/tmp/yum-root-Z3I8ac/ius-release.rpm: ius-release-1.0-15.ius.centos7.
 ↪noarch
```

(continues on next page)

(continued from previous page)

```
Marking /var/tmp/yum-root-Z3I8ac/ius-release.rpm to be installed
Resolving Dependencies
--> Running transaction check
---> Package ius-release.noarch 0:1.0-15.ius.centos7 will be installed
--> Processing Dependency: epel-release = 7 for package: ius-release-1.0-15.ius.
↪centos7.noarch
Loading mirror speeds from cached hostfile
* base: centos.quelquesmots.fr
* extras: fr.mirror.babylon.network
* updates: fr.mirror.babylon.network
--> Running transaction check
---> Package epel-release.noarch 0:7-9 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
Package           Arch        Version                   Repository         Size
================================================================================
Installing:
ius-release       noarch      1.0-15.ius.centos7        /ius-release      8.5 k
Installing for dependencies:
epel-release      noarch      7-9                       extras             14 k

Transaction Summary
================================================================================
Install  1 Package (+1 Dependent package)

Total size: 23 k
Total download size: 14 k
Installed size: 33 k
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : epel-release-7-9.noarch                                       1/2
Installing : ius-release-1.0-15.ius.centos7.noarch                         2/2
Verifying  : ius-release-1.0-15.ius.centos7.noarch                         1/2
Verifying  : epel-release-7-9.noarch                                       2/2

Installed:
ius-release.noarch 0:1.0-15.ius.centos7

Dependency Installed:
epel-release.noarch 0:7-9

Complete!
Removing intermediate container cea6a40470fa
---> b9963da64678
Step 4/5 : RUN yum install -y python36u python36u-libs python36u-devel python36u-
↪pip
---> Running in f9691783f72c
Loaded plugins: fastestmirror, ovl
Loading mirror speeds from cached hostfile
* base: centos.quelquesmots.fr
* epel: fr.mirror.babylon.network
* extras: fr.mirror.babylon.network
* ius: mirrors.tongji.edu.cn
* updates: fr.mirror.babylon.network
Resolving Dependencies
```

(continues on next page)

(continued from previous page)

```
--> Running transaction check
---> Package python36u.x86_64 0:3.6.4-1.ius.centos7 will be installed
---> Package python36u-devel.x86_64 0:3.6.4-1.ius.centos7 will be installed
---> Package python36u-libs.x86_64 0:3.6.4-1.ius.centos7 will be installed
---> Package python36u-pip.noarch 0:9.0.1-1.ius.centos7 will be installed
--> Processing Dependency: python36u-setuptools for package: python36u-pip-9.0.1-1.
↪ius.centos7.noarch
--> Running transaction check
---> Package python36u-setuptools.noarch 0:36.6.0-1.ius.centos7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
Package                   Arch        Version                   Repository

↪                                                                  Size
================================================================================
Installing:
python36u                 x86_64      3.6.4-1.ius.centos7       ius        56 k
python36u-devel           x86_64      3.6.4-1.ius.centos7       ius       839 k
python36u-libs            x86_64      3.6.4-1.ius.centos7       ius       8.7 M
python36u-pip             noarch      9.0.1-1.ius.centos7       ius       1.8 M
Installing for dependencies:
python36u-setuptools      noarch      36.6.0-1.ius.centos7      ius       587 k

Transaction Summary
================================================================================
Install  4 Packages (+1 Dependent package)

Total download size: 12 M
Installed size: 53 M
Downloading packages:
warning: /var/cache/yum/x86_64/7/ius/packages/python36u-devel-3.6.4-1.ius.centos7.
↪x86_64.rpm: Header V4 DSA/SHA1 Signature, key ID 9cd4953f: NOKEY
Public key for python36u-devel-3.6.4-1.ius.centos7.x86_64.rpm is not installed
--------------------------------------------------------------------------------
Total                                           1.0 MB/s |  12 MB  00:12
Retrieving key from file:///etc/pki/rpm-gpg/IUS-COMMUNITY-GPG-KEY
Importing GPG key 0x9CD4953F:
Userid     : "IUS Community Project <coredev@iuscommunity.org>"
Fingerprint: 8b84 6e3a b3fe 6462 74e8 670f da22 1cdf 9cd4 953f
Package    : ius-release-1.0-15.ius.centos7.noarch (installed)
From       : /etc/pki/rpm-gpg/IUS-COMMUNITY-GPG-KEY
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : python36u-libs-3.6.4-1.ius.centos7.x86_64                    1/5
Installing : python36u-3.6.4-1.ius.centos7.x86_64                         2/5
Installing : python36u-setuptools-36.6.0-1.ius.centos7.noarch             3/5
Installing : python36u-pip-9.0.1-1.ius.centos7.noarch                     4/5
Installing : python36u-devel-3.6.4-1.ius.centos7.x86_64                   5/5
Verifying  : python36u-setuptools-36.6.0-1.ius.centos7.noarch             1/5
Verifying  : python36u-pip-9.0.1-1.ius.centos7.noarch                     2/5
Verifying  : python36u-3.6.4-1.ius.centos7.x86_64                         3/5
Verifying  : python36u-libs-3.6.4-1.ius.centos7.x86_64                    4/5
Verifying  : python36u-devel-3.6.4-1.ius.centos7.x86_64                   5/5

Installed:
python36u.x86_64 0:3.6.4-1.ius.centos7
```

(continues on next page)

(continued from previous page)

```
python36u-devel.x86_64 0:3.6.4-1.ius.centos7
python36u-libs.x86_64 0:3.6.4-1.ius.centos7
python36u-pip.noarch 0:9.0.1-1.ius.centos7

Dependency Installed:
python36u-setuptools.noarch 0:36.6.0-1.ius.centos7

Complete!
Removing intermediate container f9691783f72c
---> 2edcf9418ddb
Step 5/5 : RUN yum install -y which
---> Running in b7bf8af2a677
Loaded plugins: fastestmirror, ovl
Loading mirror speeds from cached hostfile
* base: centos.quelquesmots.fr
* epel: mirror.airenetworks.es
* extras: fr.mirror.babylon.network
* ius: mirrors.ircam.fr
* updates: fr.mirror.babylon.network
Resolving Dependencies
--> Running transaction check
---> Package which.x86_64 0:2.20-7.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
Package          Arch              Version                Repository       Size
================================================================================
Installing:
which            x86_64            2.20-7.el7             base             41 k

Transaction Summary
================================================================================
Install  1 Package

Total download size: 41 k
Installed size: 75 k
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : which-2.20-7.el7.x86_64                                      1/1
install-info: No such file or directory for /usr/share/info/which.info.gz
Verifying  : which-2.20-7.el7.x86_64                                      1/1

Installed:
which.x86_64 0:2.20-7.el7

Complete!
Removing intermediate container b7bf8af2a677
---> c0efabb4e2cb
Successfully built c0efabb4e2cb
Successfully tagged id3centos7:1
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 Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\centos7> docker images
```

(continues on next page)

(continued from previous page)

```
REPOSITORY                  TAG                 IMAGE ID            CREATED
↪     SIZE
id3centos7                  1                   c0efabb4e2cb        54 seconds ago
↪     770MB
ch4messageboardapp_web      latest              a08febb741e4        17 hours ago
↪     782MB
postgres                    10.1                b820823c41bd        17 hours ago
↪     290MB
<none>                      <none>              62b12eb064b3        17 hours ago
↪     729MB
<none>                      <none>              46dc0ae69726        17 hours ago
↪     729MB
<none>                      <none>              b940cde74b73        17 hours ago
↪     920MB
<none>                      <none>              ad18d8d88ab0        18 hours ago
↪     920MB
<none>                      <none>              71e39ba2a7bb        18 hours ago
↪     729MB
<none>                      <none>              9fda17d01d46        18 hours ago
↪     729MB
<none>                      <none>              326079a0d350        18 hours ago
↪     772MB
<none>                      <none>              a617107b453b        18 hours ago
↪     772MB
<none>                      <none>              8fdb1af40b0f        19 hours ago
↪     729MB
centos                      7                   ff426288ea90        3 weeks ago
↪     207MB
nginx                       latest              3f8a4339aadd        5 weeks ago
↪     108MB
python                      3.6                 c1e459c00dc3        6 weeks ago
↪     692MB
postgres                    <none>              ec61d13c8566        7 weeks ago
↪     287MB
docker4w/nsenter-dockerd    latest              cae870735e91        3 months ago
↪     187kB
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\centos7> doc
```

4.12.9 docker images

```
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\centos7> docker images
```

```
REPOSITORY                  TAG                 IMAGE ID            CREATED
↪     SIZE
id3centos7                  1                   c0efabb4e2cb        54 seconds ago
↪     770MB
ch4messageboardapp_web      latest              a08febb741e4        17 hours ago
↪     782MB
postgres                    10.1                b820823c41bd        17 hours ago
↪     290MB
<none>                      <none>              62b12eb064b3        17 hours ago
↪     729MB
<none>                      <none>              46dc0ae69726        17 hours ago
↪     729MB
<none>                      <none>              b940cde74b73        17 hours ago
↪     920MB
<none>                      <none>              ad18d8d88ab0        18 hours ago
↪     920MB
```

(continues on next page)

(continued from previous page)

```
<none>                          <none>              71e39ba2a7bb        18 hours ago␣
↪    729MB
<none>                          <none>              9fda17d01d46        18 hours ago␣
↪    729MB
<none>                          <none>              326079a0d350        18 hours ago␣
↪    772MB
<none>                          <none>              a617107b453b        18 hours ago␣
↪    772MB
<none>                          <none>              8fdb1af40b0f        19 hours ago␣
↪    729MB
centos                          7                   ff426288ea90        3 weeks ago␣
↪    207MB
nginx                           latest              3f8a4339aadd        5 weeks ago␣
↪    108MB
python                          3.6                 c1e459c00dc3        6 weeks ago␣
↪    692MB
postgres                        <none>              ec61d13c8566        7 weeks ago␣
↪    287MB
docker4w/nsenter-dockerd        latest              cae870735e91        3 months ago␣
↪    187kB
```

4.12.10 docker run –name test -it id3centos7:1

4.12.11 Probleme avec regex

regex = “*”

```
----------------------------------------
Failed building wheel for regex
Running setup.py clean for regex
Failed to build regex
Installing collected packages: regex
Running setup.py install for regex ... error
Complete output from command /opt/intranet/intranet/bin/python3.6 -u -c "import␣
↪setuptools, tokenize;__file__='/tmp/pip-build-rrdh2091/regex/setup.py';
↪f=getattr(tokenize, 'open', open)(__file__);code=f.read().replace('\r\n', '\n');
↪f.close();exec(compile(code, __file__, 'exec'))" install --record /tmp/pip-
↪fjizm5wj-record/install-record.txt --single-version-externally-managed --compile␣
↪--install-headers /opt/intranet/intranet/include/site/python3.6/regex:
/opt/intranet/intranet/lib/python3.6/site-packages/setuptools/dist.py:355:␣
↪UserWarning: Normalizing '2018.01.10' to '2018.1.10'
normalized_version,
running install
running build
running build_py
creating build
creating build/lib.linux-x86_64-3.6
copying regex_3/regex.py -> build/lib.linux-x86_64-3.6
copying regex_3/_regex_core.py -> build/lib.linux-x86_64-3.6
copying regex_3/test_regex.py -> build/lib.linux-x86_64-3.6
running build_ext
building '_regex' extension
creating build/temp.linux-x86_64-3.6
creating build/temp.linux-x86_64-3.6/regex_3
gcc -pthread -Wno-unused-result -Wsign-compare -DDYNAMIC_ANNOTATIONS_ENABLED=1 -
↪DNDEBUG -O2 -g -pipe -Wall -Wp,-D_FORTIFY_SOURCE=2 -fexceptions -fstack-
↪protector-strong --param=ssp-buffer-size=4 -grecord-gcc-switches -m64 -
↪mtune=generic -D_GNU_SOURCE -fPIC -fwrapv -fPIC -I/usr/include/python3.6m -c␣
↪regex_3/_regex.c -o build/temp.linux-x86_64-3.6/regex_3/_regex.o
```

(continues on next page)

(continued from previous page)

```
unable to execute 'gcc': No such file or directory
error: command 'gcc' failed with exit status 1

        ----------------------------------------
Command "/opt/intranet/intranet/bin/python3.6 -u -c "import setuptools, tokenize;__
↪file__='/tmp/pip-build-rrdh2091/regex/setup.py';f=getattr(tokenize, 'open',␣
↪open)(__file__);code=f.read().replace('\r\n', '\n');f.close();exec(compile(code,␣
↪__file__, 'exec'))" install --record /tmp/pip-fjizm5wj-record/install-record.txt␣
↪--single-version-externally-managed --compile --install-headers /opt/intranet/
↪intranet/include/site/python3.6/regex" failed with error code 1 in /tmp/pip-
↪build-rrdh2091/regex/
(intranet) [root@35d914e8c996 intranet]# yum install gcc gcc-devel
```

4.12.12 yum install gcc

```
(intranet) [root@35d914e8c996 intranet]# yum install gcc gcc-devel
```

```
Loaded plugins: fastestmirror, ovl
Loading mirror speeds from cached hostfile
* base: centos.quelquesmots.fr
* epel: mirror.vutbr.cz
* extras: fr.mirror.babylon.network
* ius: mirror.team-cymru.org
* updates: fr.mirror.babylon.network
No package gcc-devel available.
Resolving Dependencies
--> Running transaction check
---> Package gcc.x86_64 0:4.8.5-16.el7_4.1 will be installed
--> Processing Dependency: libgomp = 4.8.5-16.el7_4.1 for package: gcc-4.8.5-16.
↪el7_4.1.x86_64
--> Processing Dependency: cpp = 4.8.5-16.el7_4.1 for package: gcc-4.8.5-16.el7_4.
↪1.x86_64
--> Processing Dependency: glibc-devel >= 2.2.90-12 for package: gcc-4.8.5-16.el7_
↪4.1.x86_64
--> Processing Dependency: libmpfr.so.4()(64bit) for package: gcc-4.8.5-16.el7_4.1.
↪x86_64
--> Processing Dependency: libmpc.so.3()(64bit) for package: gcc-4.8.5-16.el7_4.1.
↪x86_64
--> Processing Dependency: libgomp.so.1()(64bit) for package: gcc-4.8.5-16.el7_4.1.
↪x86_64
--> Running transaction check
---> Package cpp.x86_64 0:4.8.5-16.el7_4.1 will be installed
---> Package glibc-devel.x86_64 0:2.17-196.el7_4.2 will be installed
--> Processing Dependency: glibc-headers = 2.17-196.el7_4.2 for package: glibc-
↪devel-2.17-196.el7_4.2.x86_64
--> Processing Dependency: glibc-headers for package: glibc-devel-2.17-196.el7_4.2.
↪x86_64
---> Package libgomp.x86_64 0:4.8.5-16.el7_4.1 will be installed
---> Package libmpc.x86_64 0:1.0.1-3.el7 will be installed
---> Package mpfr.x86_64 0:3.1.1-4.el7 will be installed
--> Running transaction check
---> Package glibc-headers.x86_64 0:2.17-196.el7_4.2 will be installed
--> Processing Dependency: kernel-headers >= 2.2.1 for package: glibc-headers-2.17-
↪196.el7_4.2.x86_64
--> Processing Dependency: kernel-headers for package: glibc-headers-2.17-196.el7_
↪4.2.x86_64
--> Running transaction check
---> Package kernel-headers.x86_64 0:3.10.0-693.17.1.el7 will be installed
--> Finished Dependency Resolution
```

(continues on next page)

(continued from previous page)

```
Dependencies Resolved

==========================================================================================
Package                                              Arch                                  ␣
↪   Version                                                      Repository                ␣
↪                 Size
==========================================================================================
Installing:
gcc                                                  x86_64                                ␣
↪   4.8.5-16.el7_4.1                                             updates                   ␣
↪                 16 M
Installing for dependencies:
cpp                                                  x86_64                                ␣
↪   4.8.5-16.el7_4.1                                             updates                   ␣
↪                5.9 M
glibc-devel                                          x86_64                                ␣
↪   2.17-196.el7_4.2                                             updates                   ␣
↪                1.1 M
glibc-headers                                        x86_64                                ␣
↪   2.17-196.el7_4.2                                             updates                   ␣
↪                676 k
kernel-headers                                       x86_64                                ␣
↪   3.10.0-693.17.1.el7                                          updates                   ␣
↪                6.0 M
libgomp                                              x86_64                                ␣
↪   4.8.5-16.el7_4.1                                             updates                   ␣
↪                154 k
libmpc                                               x86_64                                ␣
↪   1.0.1-3.el7                                                  base                      ␣
↪                 51 k
mpfr                                                 x86_64                                ␣
↪   3.1.1-4.el7                                                  base                      ␣
↪                203 k

Transaction Summary
==========================================================================================
Install  1 Package (+7 Dependent packages)

Total download size: 30 M
Installed size: 60 M
Is this ok [y/d/N]: y
Downloading packages:
(1/8): glibc-headers-2.17-196.el7_4.2.x86_64.rpm                                           ␣
↪                                                                                          ␣
↪| 676 kB  00:00:01
(2/8): libgomp-4.8.5-16.el7_4.1.x86_64.rpm                                                 ␣
↪                                                                                          ␣
↪| 154 kB  00:00:00
(3/8): glibc-devel-2.17-196.el7_4.2.x86_64.rpm                                             ␣
↪                                                                                          ␣
↪| 1.1 MB  00:00:02
(4/8): libmpc-1.0.1-3.el7.x86_64.rpm                                                       ␣
↪                                                                                          ␣
↪|  51 kB  00:00:00
(5/8): mpfr-3.1.1-4.el7.x86_64.rpm                                                         ␣
↪                                                                                          ␣
↪| 203 kB  00:00:00
(6/8): cpp-4.8.5-16.el7_4.1.x86_64.rpm                                                     ␣
↪                                                                                          ␣
↪| 5.9 MB  00:00:05
```

(continues on next page)

(continued from previous page)

```
(7/8): kernel-headers-3.10.0-693.17.1.el7.x86_64.rpm
↪                                                                                ␣
↪                                                                                ␣
↪| 6.0 MB  00:00:12
(8/8): gcc-4.8.5-16.el7_4.1.x86_64.rpm
↪                                                                                ␣
↪                                                                                ␣
↪|  16 MB  00:01:13
--------------------------------------------------------------------------------
↪--------------------------------------------------------------------------------
↪------------------------
Total
↪                                                                               ␣
↪                                                                    421 kB/s␣
↪|  30 MB  00:01:13
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : mpfr-3.1.1-4.el7.x86_64
↪                                                                                ␣
↪                                                                                ␣
↪                  1/8
Installing : libmpc-1.0.1-3.el7.x86_64
↪                                                                                ␣
↪                                                                                ␣
↪                  2/8
Installing : cpp-4.8.5-16.el7_4.1.x86_64
↪                                                                                ␣
↪                                                                                ␣
↪                  3/8
Installing : kernel-headers-3.10.0-693.17.1.el7.x86_64
↪                                                                                ␣
↪                                                                                ␣
↪                  4/8
Installing : glibc-headers-2.17-196.el7_4.2.x86_64
↪                                                                                ␣
↪                                                                                ␣
↪                  5/8
Installing : glibc-devel-2.17-196.el7_4.2.x86_64
↪                                                                                ␣
↪                                                                                ␣
↪                  6/8
Installing : libgomp-4.8.5-16.el7_4.1.x86_64
↪                                                                                ␣
↪                                                                                ␣
↪                  7/8
Installing : gcc-4.8.5-16.el7_4.1.x86_64
↪                                                                                ␣
↪                                                                                ␣
↪                  8/8
Verifying  : cpp-4.8.5-16.el7_4.1.x86_64
↪                                                                                ␣
↪                                                                                ␣
↪                  1/8
Verifying  : glibc-devel-2.17-196.el7_4.2.x86_64
↪                                                                                ␣
↪                                                                                ␣
↪                  2/8
Verifying  : mpfr-3.1.1-4.el7.x86_64
↪                                                                                ␣
↪                                                                                ␣
↪                  3/8
Verifying  : libgomp-4.8.5-16.el7_4.1.x86_64
↪                                                                                ␣
↪                                                                                ␣
↪                  4/8
Verifying  : libmpc-1.0.1-3.el7.x86_64
↪                                                                                ␣
↪                                                                                ␣
↪                  5/8
Verifying  : kernel-headers-3.10.0-693.17.1.el7.x86_64
↪                                                                                ␣
↪                                                                                ␣
↪                  6/8
Verifying  : glibc-headers-2.17-196.el7_4.2.x86_64
↪                                                                                ␣
↪                                                                                ␣
↪                  7/8
```

(continues on next page)

(continued from previous page)

```
Verifying  : gcc-4.8.5-16.el7_4.1.x86_64
↪                                                                              ␣
↪␣
↪                 8/8

Installed:
gcc.x86_64 0:4.8.5-16.el7_4.1

Dependency Installed:
cpp.x86_64 0:4.8.5-16.el7_4.1            glibc-devel.x86_64 0:2.17-196.el7_4.2
↪␣
↪   glibc-headers.x86_64 0:2.17-196.el7_4.2        kernel-headers.x86_64 0:3.10.0-
↪693.17.1.el7
libgomp.x86_64 0:4.8.5-16.el7_4.1        libmpc.x86_64 0:1.0.1-3.el7
↪␣
↪   mpfr.x86_64 0:3.1.1-4.el7

Complete!
```

```
(intranet) [root@35d914e8c996 intranet]# pip install regex
```

```
Collecting regex
Using cached regex-2018.01.10.tar.gz
Building wheels for collected packages: regex
Running setup.py bdist_wheel for regex ... done
Stored in directory: /root/.cache/pip/wheels/6c/44/28/
↪d58762d1fbdf2e6f6fb00d4fec7d3384ad0ac565b895c044eb
Successfully built regex
Installing collected packages: regex
Successfully installed regex-2018.1.10
```

4.12.13 yum install openldap-devel

```
(intranet) [root@35d914e8c996 intranet]# yum install openldap-devel
```

```
Loaded plugins: fastestmirror, ovl
Loading mirror speeds from cached hostfile
* base: centos.quelquesmots.fr
* epel: fr.mirror.babylon.network
* extras: fr.mirror.babylon.network
* ius: mirrors.tongji.edu.cn
* updates: fr.mirror.babylon.network
Resolving Dependencies
--> Running transaction check
---> Package openldap-devel.x86_64 0:2.4.44-5.el7 will be installed
--> Processing Dependency: cyrus-sasl-devel(x86-64) for package: openldap-devel-2.
↪4.44-5.el7.x86_64
--> Running transaction check
---> Package cyrus-sasl-devel.x86_64 0:2.1.26-21.el7 will be installed
--> Processing Dependency: cyrus-sasl(x86-64) = 2.1.26-21.el7 for package: cyrus-
↪sasl-devel-2.1.26-21.el7.x86_64
--> Running transaction check
---> Package cyrus-sasl.x86_64 0:2.1.26-21.el7 will be installed
--> Processing Dependency: /sbin/service for package: cyrus-sasl-2.1.26-21.el7.x86_
↪64
--> Running transaction check
---> Package initscripts.x86_64 0:9.49.39-1.el7_4.1 will be installed
--> Processing Dependency: sysvinit-tools >= 2.87-5 for package: initscripts-9.49.
↪39-1.el7_4.1.x86_64
--> Processing Dependency: iproute for package: initscripts-9.49.39-1.el7_4.1.x86_
↪64
```

(continues on next page)

(continued from previous page)

```
--> Running transaction check
---> Package iproute.x86_64 0:3.10.0-87.el7 will be installed
--> Processing Dependency: libmnl.so.0(LIBMNL_1.0)(64bit) for package: iproute-3.
↪10.0-87.el7.x86_64
--> Processing Dependency: libxtables.so.10()(64bit) for package: iproute-3.10.0-
↪87.el7.x86_64
--> Processing Dependency: libmnl.so.0()(64bit) for package: iproute-3.10.0-87.el7.
↪x86_64
---> Package sysvinit-tools.x86_64 0:2.88-14.dsf.el7 will be installed
--> Running transaction check
---> Package iptables.x86_64 0:1.4.21-18.2.el7_4 will be installed
--> Processing Dependency: libnfnetlink.so.0()(64bit) for package: iptables-1.4.21-
↪18.2.el7_4.x86_64
--> Processing Dependency: libnetfilter_conntrack.so.3()(64bit) for package:␣
↪iptables-1.4.21-18.2.el7_4.x86_64
---> Package libmnl.x86_64 0:1.0.3-7.el7 will be installed
--> Running transaction check
---> Package libnetfilter_conntrack.x86_64 0:1.0.6-1.el7_3 will be installed
---> Package libnfnetlink.x86_64 0:1.0.1-4.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================================
Package                                                 Arch
↪          Version                                            Repository                ␣
↪                  Size
================================================================================================
Installing:
openldap-devel                                          x86_64
↪          2.4.44-5.el7                                       base                      ␣
↪                 801 k
Installing for dependencies:
cyrus-sasl                                              x86_64
↪          2.1.26-21.el7                                      base                      ␣
↪                  88 k
cyrus-sasl-devel                                        x86_64
↪          2.1.26-21.el7                                      base                      ␣
↪                 310 k
initscripts                                             x86_64
↪          9.49.39-1.el7_4.1                                  updates                   ␣
↪                 435 k
iproute                                                 x86_64
↪          3.10.0-87.el7                                      base                      ␣
↪                 651 k
iptables                                                x86_64
↪          1.4.21-18.2.el7_4                                  updates                   ␣
↪                 428 k
libmnl                                                  x86_64
↪          1.0.3-7.el7                                        base                      ␣
↪                  23 k
libnetfilter_conntrack                                  x86_64
↪          1.0.6-1.el7_3                                      base                      ␣
↪                  55 k
libnfnetlink                                            x86_64
↪          1.0.1-4.el7                                        base                      ␣
↪                  26 k
sysvinit-tools                                          x86_64
↪          2.88-14.dsf.el7                                    base                      ␣
↪                  63 k
```

(continues on next page)

(continued from previous page)

```
Transaction Summary
================================================================================
Install  1 Package (+9 Dependent packages)

Total download size: 2.8 M
Installed size: 9.5 M
Is this ok [y/d/N]: y
Downloading packages:
(1/10): cyrus-sasl-2.1.26-21.el7.x86_64.rpm
↪                                                                              ⌙
↪|  88 kB  00:00:00
(2/10): cyrus-sasl-devel-2.1.26-21.el7.x86_64.rpm
↪                                                                              ⌙
↪| 310 kB  00:00:00
(3/10): libmnl-1.0.3-7.el7.x86_64.rpm
↪                                                                              ⌙
↪|  23 kB  00:00:00
(4/10): initscripts-9.49.39-1.el7_4.1.x86_64.rpm
↪                                                                              ⌙
↪| 435 kB  00:00:00
(5/10): libnetfilter_conntrack-1.0.6-1.el7_3.x86_64.rpm
↪                                                                              ⌙
↪|  55 kB  00:00:00
(6/10): libnfnetlink-1.0.1-4.el7.x86_64.rpm
↪                                                                              ⌙
↪|  26 kB  00:00:00
(7/10): iptables-1.4.21-18.2.el7_4.x86_64.rpm
↪                                                                              ⌙
↪| 428 kB  00:00:01
(8/10): sysvinit-tools-2.88-14.dsf.el7.x86_64.rpm
↪                                                                              ⌙
↪|  63 kB  00:00:00
(9/10): openldap-devel-2.4.44-5.el7.x86_64.rpm
↪                                                                              ⌙
↪| 801 kB  00:00:00
(10/10): iproute-3.10.0-87.el7.x86_64.rpm
↪                                                                              ⌙
↪| 651 kB  00:00:01
--------------------------------------------------------------------------------
↪--------------------------------------------------------------------------------
↪-----------------------
Total
↪                                                                  1.2 MB/s ⌙
↪| 2.8 MB  00:00:02
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : libnfnetlink-1.0.1-4.el7.x86_64
↪                                                                              ⌙
↪                1/10
Installing : libmnl-1.0.3-7.el7.x86_64
↪                                                                              ⌙
↪                2/10
Installing : libnetfilter_conntrack-1.0.6-1.el7_3.x86_64
↪                                                                              ⌙
↪                3/10
Installing : iptables-1.4.21-18.2.el7_4.x86_64
↪                                                                              ⌙
↪                4/10
Installing : iproute-3.10.0-87.el7.x86_64
↪                                                                              ⌙
↪                5/10
```

(continues on next page)

(continued from previous page)

```
Installing : sysvinit-tools-2.88-14.dsf.el7.x86_64
↪                                                                              ␣
↪                6/10
Installing : initscripts-9.49.39-1.el7_4.1.x86_64
↪                                                                              ␣
↪                7/10
Installing : cyrus-sasl-2.1.26-21.el7.x86_64
↪                                                                              ␣
↪                8/10
Installing : cyrus-sasl-devel-2.1.26-21.el7.x86_64
↪                                                                              ␣
↪                9/10
Installing : openldap-devel-2.4.44-5.el7.x86_64
↪                                                                              ␣
↪               10/10
Verifying  : iptables-1.4.21-18.2.el7_4.x86_64
↪                                                                              ␣
↪                1/10
Verifying  : libmnl-1.0.3-7.el7.x86_64
↪                                                                              ␣
↪                2/10
Verifying  : iproute-3.10.0-87.el7.x86_64
↪                                                                              ␣
↪                3/10
Verifying  : initscripts-9.49.39-1.el7_4.1.x86_64
↪                                                                              ␣
↪                4/10
Verifying  : cyrus-sasl-devel-2.1.26-21.el7.x86_64
↪                                                                              ␣
↪                5/10
Verifying  : libnfnetlink-1.0.1-4.el7.x86_64
↪                                                                              ␣
↪                6/10
Verifying  : sysvinit-tools-2.88-14.dsf.el7.x86_64
↪                                                                              ␣
↪                7/10
Verifying  : libnetfilter_conntrack-1.0.6-1.el7_3.x86_64
↪                                                                              ␣
↪                8/10
Verifying  : openldap-devel-2.4.44-5.el7.x86_64
↪                                                                              ␣
↪                9/10
Verifying  : cyrus-sasl-2.1.26-21.el7.x86_64
↪                                                                              ␣
↪               10/10

Installed:
openldap-devel.x86_64 0:2.4.44-5.el7

Dependency Installed:
cyrus-sasl.x86_64 0:2.1.26-21.el7             cyrus-sasl-devel.x86_64 0:2.1.26-21.
↪el7       initscripts.x86_64 0:9.49.39-1.el7_4.1              iproute.x86_64 0:3.
↪10.0-87.el7
iptables.x86_64 0:1.4.21-18.2.el7_4           libmnl.x86_64 0:1.0.3-7.el7
↪         libnetfilter_conntrack.x86_64 0:1.0.6-1.el7_3       libnfnetlink.x86_64␣
↪0:1.0.1-4.el7
sysvinit-tools.x86_64 0:2.88-14.dsf.el7

Complete!
```

4.12.14 pip install pyldap

```
(intranet) [root@35d914e8c996 intranet]# pip install pyldap
```

```
Collecting pyldap
Using cached pyldap-2.4.45.tar.gz
Requirement already satisfied: setuptools in ./intranet/lib/python3.6/site-
↪packages (from pyldap)
Building wheels for collected packages: pyldap
Running setup.py bdist_wheel for pyldap ... done
Stored in directory: /root/.cache/pip/wheels/0c/a3/42/
↪e6127de64a53567a11c4e3ee5991547cb8f5a3241d2d67947e
Successfully built pyldap
Installing collected packages: pyldap
Successfully installed pyldap-2.4.45
```

4.12.15 Nouveau fichier Dockerfile

4.12.15.1 Dockerfile

```
# Use an official centos7 image
FROM centos:7

RUN yum update -y \
        && yum install -y https://centos7.iuscommunity.org/ius-release.rpm \
        && yum install -y python36u python36u-libs python36u-devel python36u-pip \
        && yum install -y which gcc \ # we need regex and pyldap
        && yum install -y openldap-devel  # we need pyldap
```

4.12.15.2 which python3.6

```
[root@5a070209b99d /]# which python3.6
```

```
/usr/bin/python3.6
```

4.12.15.3 python3.6 -m pip install pipenv

```
python3.6 -m pip install pipenv
```

```
Collecting pipenv
  Downloading pipenv-9.0.3.tar.gz (3.9MB)
        100% || 3.9MB 336kB/s
Collecting virtualenv (from pipenv)
  Downloading virtualenv-15.1.0-py2.py3-none-any.whl (1.8MB)
        100% || 1.8MB 602kB/s
Collecting pew>=0.1.26 (from pipenv)
  Downloading pew-1.1.2-py2.py3-none-any.whl
Requirement already satisfied: pip>=9.0.1 in /usr/lib/python3.6/site-packages␣
↪(from pipenv)
Collecting requests>2.18.0 (from pipenv)
  Downloading requests-2.18.4-py2.py3-none-any.whl (88kB)
        100% || 92kB 2.2MB/s
Collecting flake8>=3.0.0 (from pipenv)
  Downloading flake8-3.5.0-py2.py3-none-any.whl (69kB)
```

(continues on next page)

(continued from previous page)

```
        100% || 71kB 1.8MB/s
Collecting urllib3>=1.21.1 (from pipenv)
  Downloading urllib3-1.22-py2.py3-none-any.whl (132kB)
        100% || 133kB 1.8MB/s
Requirement already satisfied: setuptools>=17.1 in /usr/lib/python3.6/site-
↪packages (from pew>=0.1.26->pipenv)
Collecting virtualenv-clone>=0.2.5 (from pew>=0.1.26->pipenv)
  Downloading virtualenv-clone-0.2.6.tar.gz
Collecting certifi>=2017.4.17 (from requests>2.18.0->pipenv)
  Downloading certifi-2018.1.18-py2.py3-none-any.whl (151kB)
        100% || 153kB 982kB/s
Collecting chardet<3.1.0,>=3.0.2 (from requests>2.18.0->pipenv)
  Downloading chardet-3.0.4-py2.py3-none-any.whl (133kB)
        100% || 143kB 1.8MB/s
Collecting idna<2.7,>=2.5 (from requests>2.18.0->pipenv)
  Downloading idna-2.6-py2.py3-none-any.whl (56kB)
        100% || 61kB 900kB/s
Collecting mccabe<0.7.0,>=0.6.0 (from flake8>=3.0.0->pipenv)
  Downloading mccabe-0.6.1-py2.py3-none-any.whl
Collecting pycodestyle<2.4.0,>=2.0.0 (from flake8>=3.0.0->pipenv)
  Downloading pycodestyle-2.3.1-py2.py3-none-any.whl (45kB)
        100% || 51kB 2.3MB/s
Collecting pyflakes<1.7.0,>=1.5.0 (from flake8>=3.0.0->pipenv)
  Downloading pyflakes-1.6.0-py2.py3-none-any.whl (227kB)
        100% || 235kB 2.2MB/s
Installing collected packages: virtualenv, virtualenv-clone, pew, urllib3, certifi,
↪ chardet, idna, requests, mccabe, pycodestyle, pyflakes, flake8, pipenv
  Running setup.py install for virtualenv-clone ... done
  Running setup.py install for pipenv ... done
Successfully installed certifi-2018.1.18 chardet-3.0.4 flake8-3.5.0 idna-2.6␣
↪mccabe-0.6.1 pew-1.1.2 pipenv-9.0.3 pycodestyle-2.3.1 pyflakes-1.6.0 requests-2.
↪18.4 urllib3-1.22 virtualenv-15.1.0 virtualenv-clone-0.2.6
```

4.12.16 Nouveau Dockerfile

4.12.16.1 Dockerfile

```
# Use an official centos7 image
FROM centos:7

RUN localedef -i fr_FR -c -f UTF-8 -A /usr/share/locale/locale.alias fr_FR.UTF-8
ENV LANG fr_FR.utf8

# gcc because we need regex and pyldap
# openldap-devel because we need pyldap
RUN yum update -y \
        && yum install -y https://centos7.iuscommunity.org/ius-release.rpm \
        && yum install -y python36u python36u-libs python36u-devel python36u-pip \
        && yum install -y which gcc \
        && yum install -y openldap-devel

RUN python3.6 -m pip install pipenv
```

4.12.16.2 docker build -t id3centos7:0.1.1 .

```
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\centos7> docker build -t id3centos7:0.1.1 .
```

```
Sending build context to Docker daemon  90.11kB
Step 1/5 : FROM centos:7
---> ff426288ea90
Step 2/5 : RUN localedef -i fr_FR -c -f UTF-8 -A /usr/share/locale/locale.alias fr_
↪FR.UTF-8
---> Running in b90f824550e7
Removing intermediate container b90f824550e7
---> b7dac1f044e3
Step 3/5 : ENV LANG fr_FR.utf8
---> Running in 107f8edaf492
Removing intermediate container 107f8edaf492
---> e28a88050b8f
Step 4/5 : RUN yum update -y     && yum install -y https://centos7.iuscommunity.
↪org/ius-release.rpm     && yum install -y python36u python36u-libs python36u-
↪devel python36u-pip     && yum install -y which gcc     && yum install -y␣
↪openldap-devel
---> Running in 531a6dcb0ab1
Loaded plugins: fastestmirror, ovl
Determining fastest mirrors
* base: centos.quelquesmots.fr
* extras: ftp.ciril.fr
* updates: centos.quelquesmots.fr
Resolving Dependencies
--> Running transaction check
---> Package bind-license.noarch 32:9.9.4-51.el7_4.1 will be updated
---> Package bind-license.noarch 32:9.9.4-51.el7_4.2 will be an update
---> Package binutils.x86_64 0:2.25.1-32.base.el7_4.1 will be updated
---> Package binutils.x86_64 0:2.25.1-32.base.el7_4.2 will be an update
---> Package kmod.x86_64 0:20-15.el7_4.6 will be updated
---> Package kmod.x86_64 0:20-15.el7_4.7 will be an update
---> Package kmod-libs.x86_64 0:20-15.el7_4.6 will be updated
---> Package kmod-libs.x86_64 0:20-15.el7_4.7 will be an update
---> Package kpartx.x86_64 0:0.4.9-111.el7 will be updated
---> Package kpartx.x86_64 0:0.4.9-111.el7_4.2 will be an update
---> Package libdb.x86_64 0:5.3.21-20.el7 will be updated
---> Package libdb.x86_64 0:5.3.21-21.el7_4 will be an update
---> Package libdb-utils.x86_64 0:5.3.21-20.el7 will be updated
---> Package libdb-utils.x86_64 0:5.3.21-21.el7_4 will be an update
---> Package systemd.x86_64 0:219-42.el7_4.4 will be updated
---> Package systemd.x86_64 0:219-42.el7_4.7 will be an update
---> Package systemd-libs.x86_64 0:219-42.el7_4.4 will be updated
---> Package systemd-libs.x86_64 0:219-42.el7_4.7 will be an update
---> Package tzdata.noarch 0:2017c-1.el7 will be updated
---> Package tzdata.noarch 0:2018c-1.el7 will be an update
---> Package yum.noarch 0:3.4.3-154.el7.centos will be updated
---> Package yum.noarch 0:3.4.3-154.el7.centos.1 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
Package            Arch        Version                       Repository    Size
================================================================================
Updating:
bind-license       noarch      32:9.9.4-51.el7_4.2           updates       84 k
binutils           x86_64      2.25.1-32.base.el7_4.2        updates      5.4 M
kmod               x86_64      20-15.el7_4.7                 updates      121 k
kmod-libs          x86_64      20-15.el7_4.7                 updates       50 k
kpartx             x86_64      0.4.9-111.el7_4.2             updates       73 k
libdb              x86_64      5.3.21-21.el7_4               updates      719 k
libdb-utils        x86_64      5.3.21-21.el7_4               updates      132 k
systemd            x86_64      219-42.el7_4.7                updates      5.2 M
```

(continues on next page)

(continued from previous page)

```
systemd-libs      x86_64      219-42.el7_4.7               updates      376 k
tzdata            noarch      2018c-1.el7                  updates      479 k
yum               noarch      3.4.3-154.el7.centos.1       updates      1.2 M

Transaction Summary
================================================================================
Upgrade  11 Packages

Total download size: 14 M
Downloading packages:
Delta RPMs disabled because /usr/bin/applydeltarpm not installed.
warning: /var/cache/yum/x86_64/7/updates/packages/bind-license-9.9.4-51.el7_4.2.
↪noarch.rpm: Header V3 RSA/SHA256 Signature, key ID f4a80eb5: NOKEY
Public key for bind-license-9.9.4-51.el7_4.2.noarch.rpm is not installed
--------------------------------------------------------------------------------
Total                                              1.5 MB/s |  14 MB  00:09
Retrieving key from file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
Importing GPG key 0xF4A80EB5:
Userid     : "CentOS-7 Key (CentOS 7 Official Signing Key) <security@centos.org>"
Fingerprint: 6341 ab27 53d7 8a78 a7c2 7bb1 24c6 a8a7 f4a8 0eb5
Package    : centos-release-7-4.1708.el7.centos.x86_64 (@CentOS)
From       : /etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Updating   : libdb-5.3.21-21.el7_4.x86_64                                1/22
Updating   : binutils-2.25.1-32.base.el7_4.2.x86_64                      2/22
Updating   : kmod-20-15.el7_4.7.x86_64                                   3/22
Updating   : systemd-libs-219-42.el7_4.7.x86_64                          4/22
Updating   : kmod-libs-20-15.el7_4.7.x86_64                              5/22
Updating   : systemd-219-42.el7_4.7.x86_64                               6/22
Updating   : libdb-utils-5.3.21-21.el7_4.x86_64                          7/22
Updating   : yum-3.4.3-154.el7.centos.1.noarch                           8/22
Updating   : 32:bind-license-9.9.4-51.el7_4.2.noarch                     9/22
Updating   : tzdata-2018c-1.el7.noarch                                  10/22
Updating   : kpartx-0.4.9-111.el7_4.2.x86_64                            11/22
Cleanup    : systemd-219-42.el7_4.4.x86_64                              12/22
Cleanup    : kmod-20-15.el7_4.6.x86_64                                  13/22
Cleanup    : libdb-utils-5.3.21-20.el7.x86_64                           14/22
Cleanup    : yum-3.4.3-154.el7.centos.noarch                            15/22
Cleanup    : 32:bind-license-9.9.4-51.el7_4.1.noarch                    16/22
Cleanup    : tzdata-2017c-1.el7.noarch                                  17/22
Cleanup    : libdb-5.3.21-20.el7.x86_64                                 18/22
Cleanup    : binutils-2.25.1-32.base.el7_4.1.x86_64                     19/22
Cleanup    : kmod-libs-20-15.el7_4.6.x86_64                             20/22
Cleanup    : systemd-libs-219-42.el7_4.4.x86_64                         21/22
Cleanup    : kpartx-0.4.9-111.el7.x86_64                                22/22
Verifying  : kmod-20-15.el7_4.7.x86_64                                   1/22
Verifying  : kmod-libs-20-15.el7_4.7.x86_64                              2/22
Verifying  : libdb-utils-5.3.21-21.el7_4.x86_64                          3/22
Verifying  : systemd-219-42.el7_4.7.x86_64                               4/22
Verifying  : kpartx-0.4.9-111.el7_4.2.x86_64                             5/22
Verifying  : tzdata-2018c-1.el7.noarch                                   6/22
Verifying  : 32:bind-license-9.9.4-51.el7_4.2.noarch                     7/22
Verifying  : systemd-libs-219-42.el7_4.7.x86_64                          8/22
Verifying  : binutils-2.25.1-32.base.el7_4.2.x86_64                      9/22
Verifying  : libdb-5.3.21-21.el7_4.x86_64                               10/22
Verifying  : yum-3.4.3-154.el7.centos.1.noarch                          11/22
Verifying  : binutils-2.25.1-32.base.el7_4.1.x86_64                     12/22
Verifying  : 32:bind-license-9.9.4-51.el7_4.1.noarch                    13/22
```

(continues on next page)

(continued from previous page)

```
Verifying  : systemd-libs-219-42.el7_4.4.x86_64                          14/22
Verifying  : kmod-20-15.el7_4.6.x86_64                                   15/22
Verifying  : systemd-219-42.el7_4.4.x86_64                               16/22
Verifying  : libdb-utils-5.3.21-20.el7.x86_64                            17/22
Verifying  : kmod-libs-20-15.el7_4.6.x86_64                              18/22
Verifying  : tzdata-2017c-1.el7.noarch                                   19/22
Verifying  : kpartx-0.4.9-111.el7.x86_64                                 20/22
Verifying  : yum-3.4.3-154.el7.centos.noarch                             21/22
Verifying  : libdb-5.3.21-20.el7.x86_64                                  22/22

Updated:
bind-license.noarch 32:9.9.4-51.el7_4.2
binutils.x86_64 0:2.25.1-32.base.el7_4.2
kmod.x86_64 0:20-15.el7_4.7
kmod-libs.x86_64 0:20-15.el7_4.7
kpartx.x86_64 0:0.4.9-111.el7_4.2
libdb.x86_64 0:5.3.21-21.el7_4
libdb-utils.x86_64 0:5.3.21-21.el7_4
systemd.x86_64 0:219-42.el7_4.7
systemd-libs.x86_64 0:219-42.el7_4.7
tzdata.noarch 0:2018c-1.el7
yum.noarch 0:3.4.3-154.el7.centos.1

Complete!
Loaded plugins: fastestmirror, ovl
Examining /var/tmp/yum-root-CU9Amb/ius-release.rpm: ius-release-1.0-15.ius.centos7.
↪noarch
Marking /var/tmp/yum-root-CU9Amb/ius-release.rpm to be installed
Resolving Dependencies
--> Running transaction check
---> Package ius-release.noarch 0:1.0-15.ius.centos7 will be installed
--> Processing Dependency: epel-release = 7 for package: ius-release-1.0-15.ius.
↪centos7.noarch
Loading mirror speeds from cached hostfile
* base: centos.quelquesmots.fr
* extras: ftp.ciril.fr
* updates: centos.quelquesmots.fr
--> Running transaction check
---> Package epel-release.noarch 0:7-9 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
Package           Arch        Version                  Repository         Size
================================================================================
Installing:
ius-release       noarch      1.0-15.ius.centos7       /ius-release      8.5 k
Installing for dependencies:
epel-release      noarch      7-9                      extras             14 k

Transaction Summary
================================================================================
Install  1 Package (+1 Dependent package)

Total size: 23 k
Total download size: 14 k
Installed size: 33 k
Downloading packages:
Running transaction check
Running transaction test
```

(continues on next page)

(continued from previous page)

```
Transaction test succeeded
Running transaction
Installing : epel-release-7-9.noarch                                      1/2
Installing : ius-release-1.0-15.ius.centos7.noarch                        2/2
Verifying  : ius-release-1.0-15.ius.centos7.noarch                        1/2
Verifying  : epel-release-7-9.noarch                                      2/2

Installed:
ius-release.noarch 0:1.0-15.ius.centos7

Dependency Installed:
epel-release.noarch 0:7-9

Complete!
Loaded plugins: fastestmirror, ovl
Loading mirror speeds from cached hostfile
* base: centos.quelquesmots.fr
* epel: fr.mirror.babylon.network
* extras: ftp.ciril.fr
* ius: mirrors.ircam.fr
* updates: centos.quelquesmots.fr
Resolving Dependencies
--> Running transaction check
---> Package python36u.x86_64 0:3.6.4-1.ius.centos7 will be installed
---> Package python36u-devel.x86_64 0:3.6.4-1.ius.centos7 will be installed
---> Package python36u-libs.x86_64 0:3.6.4-1.ius.centos7 will be installed
---> Package python36u-pip.noarch 0:9.0.1-1.ius.centos7 will be installed
--> Processing Dependency: python36u-setuptools for package: python36u-pip-9.0.1-1.
 ↪ius.centos7.noarch
--> Running transaction check
---> Package python36u-setuptools.noarch 0:36.6.0-1.ius.centos7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
Package                    Arch        Version                    Repository
                                                                              ␣
 ↪                                                        Size
================================================================================
Installing:
python36u                  x86_64      3.6.4-1.ius.centos7        ius        56 k
python36u-devel            x86_64      3.6.4-1.ius.centos7        ius       839 k
python36u-libs             x86_64      3.6.4-1.ius.centos7        ius       8.7 M
python36u-pip              noarch      9.0.1-1.ius.centos7        ius       1.8 M
Installing for dependencies:
python36u-setuptools       noarch      36.6.0-1.ius.centos7       ius       587 k

Transaction Summary
================================================================================
Install  4 Packages (+1 Dependent package)

Total download size: 12 M
Installed size: 53 M
Downloading packages:
warning: /var/cache/yum/x86_64/7/ius/packages/python36u-setuptools-36.6.0-1.ius.
 ↪centos7.noarch.rpm: Header V4 DSA/SHA1 Signature, key ID 9cd4953f: NOKEY
Public key for python36u-setuptools-36.6.0-1.ius.centos7.noarch.rpm is not␣
 ↪installed
--------------------------------------------------------------------------------
Total                                              634 kB/s |  12 MB  00:19
```

(continues on next page)

(continued from previous page)

```
Retrieving key from file:///etc/pki/rpm-gpg/IUS-COMMUNITY-GPG-KEY
Importing GPG key 0x9CD4953F:
Userid     : "IUS Community Project <coredev@iuscommunity.org>"
Fingerprint: 8b84 6e3a b3fe 6462 74e8 670f da22 1cdf 9cd4 953f
Package    : ius-release-1.0-15.ius.centos7.noarch (installed)
From       : /etc/pki/rpm-gpg/IUS-COMMUNITY-GPG-KEY
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : python36u-libs-3.6.4-1.ius.centos7.x86_64                    1/5
Installing : python36u-3.6.4-1.ius.centos7.x86_64                         2/5
Installing : python36u-setuptools-36.6.0-1.ius.centos7.noarch             3/5
Installing : python36u-pip-9.0.1-1.ius.centos7.noarch                     4/5
Installing : python36u-devel-3.6.4-1.ius.centos7.x86_64                   5/5
Verifying  : python36u-setuptools-36.6.0-1.ius.centos7.noarch             1/5
Verifying  : python36u-pip-9.0.1-1.ius.centos7.noarch                     2/5
Verifying  : python36u-3.6.4-1.ius.centos7.x86_64                         3/5
Verifying  : python36u-libs-3.6.4-1.ius.centos7.x86_64                    4/5
Verifying  : python36u-devel-3.6.4-1.ius.centos7.x86_64                   5/5

Installed:
python36u.x86_64 0:3.6.4-1.ius.centos7
python36u-devel.x86_64 0:3.6.4-1.ius.centos7
python36u-libs.x86_64 0:3.6.4-1.ius.centos7
python36u-pip.noarch 0:9.0.1-1.ius.centos7

Dependency Installed:
python36u-setuptools.noarch 0:36.6.0-1.ius.centos7

Complete!
Loaded plugins: fastestmirror, ovl
Loading mirror speeds from cached hostfile
* base: centos.quelquesmots.fr
* epel: fr.mirror.babylon.network
* extras: ftp.ciril.fr
* ius: mirrors.ircam.fr
* updates: centos.quelquesmots.fr
Resolving Dependencies
--> Running transaction check
---> Package gcc.x86_64 0:4.8.5-16.el7_4.1 will be installed
--> Processing Dependency: libgomp = 4.8.5-16.el7_4.1 for package: gcc-4.8.5-16.
↪el7_4.1.x86_64
--> Processing Dependency: cpp = 4.8.5-16.el7_4.1 for package: gcc-4.8.5-16.el7_4.
↪1.x86_64
--> Processing Dependency: glibc-devel >= 2.2.90-12 for package: gcc-4.8.5-16.el7_
↪4.1.x86_64
--> Processing Dependency: libmpfr.so.4()(64bit) for package: gcc-4.8.5-16.el7_4.1.
↪x86_64
--> Processing Dependency: libmpc.so.3()(64bit) for package: gcc-4.8.5-16.el7_4.1.
↪x86_64
--> Processing Dependency: libgomp.so.1()(64bit) for package: gcc-4.8.5-16.el7_4.1.
↪x86_64
---> Package which.x86_64 0:2.20-7.el7 will be installed
--> Running transaction check
---> Package cpp.x86_64 0:4.8.5-16.el7_4.1 will be installed
---> Package glibc-devel.x86_64 0:2.17-196.el7_4.2 will be installed
--> Processing Dependency: glibc-headers = 2.17-196.el7_4.2 for package: glibc-
↪devel-2.17-196.el7_4.2.x86_64
--> Processing Dependency: glibc-headers for package: glibc-devel-2.17-196.el7_4.2.
↪x86_64
```

(continues on next page)

(continued from previous page)

```
---> Package libgomp.x86_64 0:4.8.5-16.el7_4.1 will be installed
---> Package libmpc.x86_64 0:1.0.1-3.el7 will be installed
---> Package mpfr.x86_64 0:3.1.1-4.el7 will be installed
--> Running transaction check
---> Package glibc-headers.x86_64 0:2.17-196.el7_4.2 will be installed
--> Processing Dependency: kernel-headers >= 2.2.1 for package: glibc-headers-2.17-
↪196.el7_4.2.x86_64
--> Processing Dependency: kernel-headers for package: glibc-headers-2.17-196.el7_
↪4.2.x86_64
--> Running transaction check
---> Package kernel-headers.x86_64 0:3.10.0-693.17.1.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
Package             Arch        Version                     Repository    Size
================================================================================
Installing:
gcc                 x86_64      4.8.5-16.el7_4.1            updates       16 M
which               x86_64      2.20-7.el7                  base          41 k
Installing for dependencies:
cpp                 x86_64      4.8.5-16.el7_4.1            updates      5.9 M
glibc-devel         x86_64      2.17-196.el7_4.2            updates      1.1 M
glibc-headers       x86_64      2.17-196.el7_4.2            updates      676 k
kernel-headers      x86_64      3.10.0-693.17.1.el7         updates      6.0 M
libgomp             x86_64      4.8.5-16.el7_4.1            updates      154 k
libmpc              x86_64      1.0.1-3.el7                 base          51 k
mpfr                x86_64      3.1.1-4.el7                 base         203 k

Transaction Summary
================================================================================
Install  2 Packages (+7 Dependent packages)

Total download size: 30 M
Installed size: 60 M
Downloading packages:
--------------------------------------------------------------------------------
Total                                           1.3 MB/s |  30 MB  00:23
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : mpfr-3.1.1-4.el7.x86_64                                       1/9
Installing : libmpc-1.0.1-3.el7.x86_64                                     2/9
Installing : cpp-4.8.5-16.el7_4.1.x86_64                                   3/9
Installing : kernel-headers-3.10.0-693.17.1.el7.x86_64                     4/9
Installing : glibc-headers-2.17-196.el7_4.2.x86_64                         5/9
Installing : glibc-devel-2.17-196.el7_4.2.x86_64                           6/9
Installing : libgomp-4.8.5-16.el7_4.1.x86_64                               7/9
Installing : gcc-4.8.5-16.el7_4.1.x86_64                                   8/9
Installing : which-2.20-7.el7.x86_64                                       9/9
install-info: No such file or directory for /usr/share/info/which.info.gz
Verifying  : cpp-4.8.5-16.el7_4.1.x86_64                                   1/9
Verifying  : glibc-devel-2.17-196.el7_4.2.x86_64                           2/9
Verifying  : which-2.20-7.el7.x86_64                                       3/9
Verifying  : mpfr-3.1.1-4.el7.x86_64                                       4/9
Verifying  : libgomp-4.8.5-16.el7_4.1.x86_64                               5/9
Verifying  : libmpc-1.0.1-3.el7.x86_64                                     6/9
Verifying  : kernel-headers-3.10.0-693.17.1.el7.x86_64                     7/9
Verifying  : glibc-headers-2.17-196.el7_4.2.x86_64                         8/9
```

(continues on next page)

(continued from previous page)

```
Verifying  : gcc-4.8.5-16.el7_4.1.x86_64                                  9/9

Installed:
gcc.x86_64 0:4.8.5-16.el7_4.1            which.x86_64 0:2.20-7.el7

Dependency Installed:
cpp.x86_64 0:4.8.5-16.el7_4.1
glibc-devel.x86_64 0:2.17-196.el7_4.2
glibc-headers.x86_64 0:2.17-196.el7_4.2
kernel-headers.x86_64 0:3.10.0-693.17.1.el7
libgomp.x86_64 0:4.8.5-16.el7_4.1
libmpc.x86_64 0:1.0.1-3.el7
mpfr.x86_64 0:3.1.1-4.el7

Complete!
Loaded plugins: fastestmirror, ovl
Loading mirror speeds from cached hostfile
* base: centos.quelquesmots.fr
* epel: fr.mirror.babylon.network
* extras: ftp.ciril.fr
* ius: mirrors.ircam.fr
* updates: centos.quelquesmots.fr
Resolving Dependencies
--> Running transaction check
---> Package openldap-devel.x86_64 0:2.4.44-5.el7 will be installed
--> Processing Dependency: cyrus-sasl-devel(x86-64) for package: openldap-devel-2.
↪4.44-5.el7.x86_64
--> Running transaction check
---> Package cyrus-sasl-devel.x86_64 0:2.1.26-21.el7 will be installed
--> Processing Dependency: cyrus-sasl(x86-64) = 2.1.26-21.el7 for package: cyrus-
↪sasl-devel-2.1.26-21.el7.x86_64
--> Running transaction check
---> Package cyrus-sasl.x86_64 0:2.1.26-21.el7 will be installed
--> Processing Dependency: /sbin/service for package: cyrus-sasl-2.1.26-21.el7.x86_
↪64
--> Running transaction check
---> Package initscripts.x86_64 0:9.49.39-1.el7_4.1 will be installed
--> Processing Dependency: sysvinit-tools >= 2.87-5 for package: initscripts-9.49.
↪39-1.el7_4.1.x86_64
--> Processing Dependency: iproute for package: initscripts-9.49.39-1.el7_4.1.x86_
↪64
--> Running transaction check
---> Package iproute.x86_64 0:3.10.0-87.el7 will be installed
--> Processing Dependency: libmnl.so.0(LIBMNL_1.0)(64bit) for package: iproute-3.
↪10.0-87.el7.x86_64
--> Processing Dependency: libxtables.so.10()(64bit) for package: iproute-3.10.0-
↪87.el7.x86_64
--> Processing Dependency: libmnl.so.0()(64bit) for package: iproute-3.10.0-87.el7.
↪x86_64
---> Package sysvinit-tools.x86_64 0:2.88-14.dsf.el7 will be installed
--> Running transaction check
---> Package iptables.x86_64 0:1.4.21-18.2.el7_4 will be installed
--> Processing Dependency: libnfnetlink.so.0()(64bit) for package: iptables-1.4.21-
↪18.2.el7_4.x86_64
--> Processing Dependency: libnetfilter_conntrack.so.3()(64bit) for package:␣
↪iptables-1.4.21-18.2.el7_4.x86_64
---> Package libmnl.x86_64 0:1.0.3-7.el7 will be installed
--> Running transaction check
---> Package libnetfilter_conntrack.x86_64 0:1.0.6-1.el7_3 will be installed
---> Package libnfnetlink.x86_64 0:1.0.1-4.el7 will be installed
--> Finished Dependency Resolution
```

(continues on next page)

(continued from previous page)

```
Dependencies Resolved

================================================================================
Package                     Arch       Version                 Repository   Size
================================================================================
Installing:
openldap-devel              x86_64     2.4.44-5.el7            base        801 k
Installing for dependencies:
cyrus-sasl                  x86_64     2.1.26-21.el7           base         88 k
cyrus-sasl-devel            x86_64     2.1.26-21.el7           base        310 k
initscripts                 x86_64     9.49.39-1.el7_4.1       updates     435 k
iproute                     x86_64     3.10.0-87.el7           base        651 k
iptables                    x86_64     1.4.21-18.2.el7_4       updates     428 k
libmnl                      x86_64     1.0.3-7.el7             base         23 k
libnetfilter_conntrack      x86_64     1.0.6-1.el7_3           base         55 k
libnfnetlink                x86_64     1.0.1-4.el7             base         26 k
sysvinit-tools              x86_64     2.88-14.dsf.el7         base         63 k

Transaction Summary
================================================================================
Install  1 Package (+9 Dependent packages)

Total download size: 2.8 M
Installed size: 9.5 M
Downloading packages:
--------------------------------------------------------------------------------
Total                                              1.2 MB/s | 2.8 MB  00:02
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : libnfnetlink-1.0.1-4.el7.x86_64                               1/10
Installing : libmnl-1.0.3-7.el7.x86_64                                     2/10
Installing : libnetfilter_conntrack-1.0.6-1.el7_3.x86_64                   3/10
Installing : iptables-1.4.21-18.2.el7_4.x86_64                             4/10
Installing : iproute-3.10.0-87.el7.x86_64                                  5/10
Installing : sysvinit-tools-2.88-14.dsf.el7.x86_64                         6/10
Installing : initscripts-9.49.39-1.el7_4.1.x86_64                          7/10
Installing : cyrus-sasl-2.1.26-21.el7.x86_64                               8/10
Installing : cyrus-sasl-devel-2.1.26-21.el7.x86_64                         9/10
Installing : openldap-devel-2.4.44-5.el7.x86_64                           10/10
Verifying  : iptables-1.4.21-18.2.el7_4.x86_64                             1/10
Verifying  : libmnl-1.0.3-7.el7.x86_64                                     2/10
Verifying  : iproute-3.10.0-87.el7.x86_64                                  3/10
Verifying  : initscripts-9.49.39-1.el7_4.1.x86_64                          4/10
Verifying  : cyrus-sasl-devel-2.1.26-21.el7.x86_64                         5/10
Verifying  : libnfnetlink-1.0.1-4.el7.x86_64                               6/10
Verifying  : sysvinit-tools-2.88-14.dsf.el7.x86_64                         7/10
Verifying  : libnetfilter_conntrack-1.0.6-1.el7_3.x86_64                   8/10
Verifying  : openldap-devel-2.4.44-5.el7.x86_64                            9/10
Verifying  : cyrus-sasl-2.1.26-21.el7.x86_64                              10/10

Installed:
openldap-devel.x86_64 0:2.4.44-5.el7

Dependency Installed:
cyrus-sasl.x86_64 0:2.1.26-21.el7
cyrus-sasl-devel.x86_64 0:2.1.26-21.el7
initscripts.x86_64 0:9.49.39-1.el7_4.1
iproute.x86_64 0:3.10.0-87.el7
```

(continues on next page)

(continued from previous page)

```
iptables.x86_64 0:1.4.21-18.2.el7_4
libmnl.x86_64 0:1.0.3-7.el7
libnetfilter_conntrack.x86_64 0:1.0.6-1.el7_3
libnfnetlink.x86_64 0:1.0.1-4.el7
sysvinit-tools.x86_64 0:2.88-14.dsf.el7

Complete!
Removing intermediate container 531a6dcb0ab1
---> 0cfdf4200049
Step 5/5 : RUN python3.6 -m pip install pipenv
---> Running in 222c51c8c187
Collecting pipenv
Downloading pipenv-9.0.3.tar.gz (3.9MB)
Collecting virtualenv (from pipenv)
Downloading virtualenv-15.1.0-py2.py3-none-any.whl (1.8MB)
Collecting pew>=0.1.26 (from pipenv)
Downloading pew-1.1.2-py2.py3-none-any.whl
Requirement already satisfied: pip>=9.0.1 in /usr/lib/python3.6/site-packages␣
↪(from pipenv)
Collecting requests>2.18.0 (from pipenv)
Downloading requests-2.18.4-py2.py3-none-any.whl (88kB)
Collecting flake8>=3.0.0 (from pipenv)
Downloading flake8-3.5.0-py2.py3-none-any.whl (69kB)
Collecting urllib3>=1.21.1 (from pipenv)
Downloading urllib3-1.22-py2.py3-none-any.whl (132kB)
Requirement already satisfied: setuptools>=17.1 in /usr/lib/python3.6/site-
↪packages (from pew>=0.1.26->pipenv)
Collecting virtualenv-clone>=0.2.5 (from pew>=0.1.26->pipenv)
Downloading virtualenv-clone-0.2.6.tar.gz
Collecting certifi>=2017.4.17 (from requests>2.18.0->pipenv)
Downloading certifi-2018.1.18-py2.py3-none-any.whl (151kB)
Collecting chardet<3.1.0,>=3.0.2 (from requests>2.18.0->pipenv)
Downloading chardet-3.0.4-py2.py3-none-any.whl (133kB)
Collecting idna<2.7,>=2.5 (from requests>2.18.0->pipenv)
Downloading idna-2.6-py2.py3-none-any.whl (56kB)
Collecting pycodestyle<2.4.0,>=2.0.0 (from flake8>=3.0.0->pipenv)
Downloading pycodestyle-2.3.1-py2.py3-none-any.whl (45kB)
Collecting mccabe<0.7.0,>=0.6.0 (from flake8>=3.0.0->pipenv)
Downloading mccabe-0.6.1-py2.py3-none-any.whl
Collecting pyflakes<1.7.0,>=1.5.0 (from flake8>=3.0.0->pipenv)
Downloading pyflakes-1.6.0-py2.py3-none-any.whl (227kB)
Installing collected packages: virtualenv, virtualenv-clone, pew, certifi, chardet,
↪ idna, urllib3, requests, pycodestyle, mccabe, pyflakes, flake8, pipenv
Running setup.py install for virtualenv-clone: started
Running setup.py install for virtualenv-clone: finished with status 'done'
Running setup.py install for pipenv: started
Running setup.py install for pipenv: finished with status 'done'
Successfully installed certifi-2018.1.18 chardet-3.0.4 flake8-3.5.0 idna-2.6␣
↪mccabe-0.6.1 pew-1.1.2 pipenv-9.0.3 pycodestyle-2.3.1 pyflakes-1.6.0 requests-2.
↪18.4 urllib3-1.22 virtualenv-15.1.0 virtualenv-clone-0.2.6
Removing intermediate container 222c51c8c187
---> 9965dbca3f49
Successfully built 9965dbca3f49
Successfully tagged id3centos7:0.1.1
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 Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\centos7>
```

4.12.17 Nouveau fichier Dockerfile

4.12.17.1 Dockerfile

```
# Use an official centos7 image
FROM centos:7

RUN localedef -i fr_FR -c -f UTF-8 -A /usr/share/locale/locale.alias fr_FR.UTF-8
ENV LANG fr_FR.utf8

# gcc because we need regex and pyldap
# openldap-devel because we need pyldap
RUN yum update -y \
        && yum install -y https://centos7.iuscommunity.org/ius-release.rpm \
        && yum install -y python36u python36u-libs python36u-devel python36u-pip \
        && yum install -y which gcc \
        && yum install -y openldap-devel

RUN python3.6 -m pip install pipenv

WORKDIR /opt/intranet
COPY Pipfile /opt/intranet/
```

4.12.17.2 Constuction de l'image docker build -t id3centos7:0.1.2 .

```
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\centos7> docker build -t id3centos7:0.1.2 .
```

```
Sending build context to Docker daemon  195.1kB
Step 1/7 : FROM centos:7
 ---> ff426288ea90
Step 2/7 : RUN localedef -i fr_FR -c -f UTF-8 -A /usr/share/locale/locale.alias fr_
↪FR.UTF-8
 ---> Using cache
 ---> b7dac1f044e3
Step 3/7 : ENV LANG fr_FR.utf8
 ---> Using cache
 ---> e28a88050b8f
Step 4/7 : RUN yum update -y     && yum install -y https://centos7.iuscommunity.
↪org/ius-release.rpm     && yum install -y python36u python36u-libs python36u-
↪devel python36u-pip     && yum install -y which gcc     && yum install -y␣
↪openldap-devel
 ---> Using cache
 ---> 0cfdf4200049
Step 5/7 : RUN python3.6 -m pip install pipenv
 ---> Using cache
 ---> 9965dbca3f49
Step 6/7 : WORKDIR /opt/intranet
Removing intermediate container ffc087754a0c
 ---> aecca04b51f8
Step 7/7 : COPY Pipfile /opt/intranet/
 ---> e126ba1ca5f5
Successfully built e126ba1ca5f5
Successfully tagged id3centos7:0.1.2
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.
```

4.12.17.3 docker run –name id3centos7.1.2 -it id3centos7:0.1.2

```
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\centos7> docker run --name id3centos7.1.2 -it id3centos7:0.1.2
```

```
[root@8586df0dcb8e intranet]# pwd
/opt/intranet
```

```
[root@8586df0dcb8e intranet]# ls -als
```

```
total 12
4 drwxr-xr-x 1 root root 4096 févr.  2 13:43 .
4 drwxr-xr-x 1 root root 4096 févr.  2 13:43 ..
4 -rwxr-xr-x 1 root root  910 févr.  2 11:23 Pipfile
```

Problème : la commande pipenv

4.12.18 Nouveau dockerfile

4.12.18.1 Dockerfile

```
# Use an official centos7 image
FROM centos:7

RUN localedef -i fr_FR -c -f UTF-8 -A /usr/share/locale/locale.alias fr_FR.UTF-8
ENV LANG fr_FR.utf8

# gcc because we need regex and pyldap
# openldap-devel because we need pyldap
RUN yum update -y \
        && yum install -y https://centos7.iuscommunity.org/ius-release.rpm \
        && yum install -y python36u python36u-libs python36u-devel python36u-pip \
        && yum install -y which gcc \
        && yum install -y openldap-devel

RUN python3.6 -m pip install pipenv

WORKDIR /opt/intranet

# copy the Pipfile to the working directory
COPY Pipfile /opt/intranet/
# https://docs.pipenv.org/advanced/
# This is useful for Docker containers, and deployment infrastructure (e.g. Heroku␣
↪does this)
RUN pipenv install
```

```
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\centos7> docker build -t id3centos7:0.1.3 .
```

```
Sending build context to Docker daemon  198.1kB
Step 1/8 : FROM centos:7
 ---> ff426288ea90
Step 2/8 : RUN localedef -i fr_FR -c -f UTF-8 -A /usr/share/locale/locale.alias fr_
↪FR.UTF-8
 ---> Using cache
 ---> b7dac1f044e3
Step 3/8 : ENV LANG fr_FR.utf8
 ---> Using cache
```

(continues on next page)

(continued from previous page)

```
 ---> e28a88050b8f
Step 4/8 : RUN yum update -y     && yum install -y https://centos7.iuscommunity.
↪org/ius-release.rpm     && yum install -y python36u python36u-libs python36u-
↪devel python36u-pip     && yum install -y which gcc     && yum install -y␣
↪openldap-devel
 ---> Using cache
 ---> 0cfdf4200049
Step 5/8 : RUN python3.6 -m pip install pipenv
 ---> Using cache
 ---> 9965dbca3f49
Step 6/8 : WORKDIR /opt/intranet
 ---> Using cache
 ---> aecca04b51f8
Step 7/8 : COPY Pipfile /opt/intranet/
 ---> Using cache
 ---> 188cff4aa6e9
Step 8/8 : RUN pipenv install
 ---> Running in cdc65d965685
Creating a virtualenv for this project...
Using base prefix '/usr'
New python executable in /root/.local/share/virtualenvs/intranet-6TUV_xiL/bin/
↪python3.6
Also creating executable in /root/.local/share/virtualenvs/intranet-6TUV_xiL/bin/
↪python
Installing setuptools, pip, wheel...done.

Virtualenv location: /root/.local/share/virtualenvs/intranet-6TUV_xiL
Pipfile.lock not found, creating...
Locking [dev-packages] dependencies...
Locking [packages] dependencies...
Updated Pipfile.lock (326c76)!
Installing dependencies from Pipfile.lock (326c76)...
To activate this project's virtualenv, run the following:
 $ pipenv shell
Removing intermediate container cdc65d965685
 ---> 179eac6f62c1
Successfully built 179eac6f62c1
Successfully tagged id3centos7:0.1.3
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 Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\centos7>
```

4.12.19 Nouveau fichier Dockerfile

4.12.19.1 Dockerfile

```
# Use an official centos7 image
FROM centos:7

RUN localedef -i fr_FR -c -f UTF-8 -A /usr/share/locale/locale.alias fr_FR.UTF-8
ENV LANG fr_FR.utf8

# gcc because we need regex and pyldap
# openldap-devel because we need pyldap
```

(continues on next page)

(continued from previous page)

```
RUN yum update -y \
        && yum install -y https://centos7.iuscommunity.org/ius-release.rpm \
        && yum install -y python36u python36u-libs python36u-devel python36u-pip \
        && yum install -y which gcc \
        && yum install -y openldap-devel

RUN python3.6 -m pip install pipenv

WORKDIR /opt/intranet

# copy the Pipfile to the working directory
ONBUILD COPY Pipfile /opt/intranet/
# https://docs.pipenv.org/advanced/
# https://github.com/pypa/pipenv/issues/1385
# This is useful for Docker containers, and deployment infrastructure (e.g. Heroku␣
↪does this)
ONBUILD RUN pipenv install --system
```

```
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\centos7> docker build -t id3centos7:0.1.4 .
```

```
Sending build context to Docker daemon  201.2kB
Step 1/8 : FROM centos:7
 ---> ff426288ea90
Step 2/8 : RUN localedef -i fr_FR -c -f UTF-8 -A /usr/share/locale/locale.alias fr_
↪FR.UTF-8
 ---> Using cache
 ---> b7dac1f044e3
Step 3/8 : ENV LANG fr_FR.utf8
 ---> Using cache
 ---> e28a88050b8f
Step 4/8 : RUN yum update -y     && yum install -y https://centos7.iuscommunity.
↪org/ius-release.rpm     && yum install -y python36u python36u-libs python36u-
↪devel python36u-pip     && yum install -y which gcc     && yum install -y␣
↪openldap-devel
 ---> Using cache
 ---> 0cfdf4200049
Step 5/8 : RUN python3.6 -m pip install pipenv
 ---> Using cache
 ---> 9965dbca3f49
Step 6/8 : WORKDIR /opt/intranet
 ---> Using cache
 ---> aecca04b51f8
Step 7/8 : ONBUILD COPY Pipfile /opt/intranet/
 ---> Running in 0d30cd780e8c
Removing intermediate container 0d30cd780e8c
 ---> c4a15216b54b
Step 8/8 : ONBUILD RUN pipenv install --system
 ---> Running in 9bb757ba3d15
Removing intermediate container 9bb757ba3d15
 ---> 237ec53f0462
Successfully built 237ec53f0462
Successfully tagged id3centos7:0.1.4
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.
```

4.12.20 Nouveau fichier Dockerfile

```
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\centos7> docker build -t id3centos7:0.1.6 .
```

```
Sending build context to Docker daemon  240.6kB
Step 1/8 : FROM centos:7
---> ff426288ea90
Step 2/8 : RUN localedef -i fr_FR -c -f UTF-8 -A /usr/share/locale/locale.alias fr_
↪FR.UTF-8
---> Using cache
---> b7dac1f044e3
Step 3/8 : ENV LANG fr_FR.utf8
---> Using cache
---> e28a88050b8f
Step 4/8 : RUN yum update -y     && yum install -y https://centos7.iuscommunity.
↪org/ius-release.rpm     && yum install -y python36u python36u-libs python36u-
↪devel python36u-pip     && yum install -y which gcc     && yum install -y␣
↪openldap-devel
---> Using cache
---> 0cfdf4200049
Step 5/8 : RUN python3.6 -m pip install pipenv
---> Using cache
---> 9965dbca3f49
Step 6/8 : WORKDIR /opt/intranet
---> Using cache
---> aecca04b51f8
Step 7/8 : COPY requirements.txt /opt/intranet/
---> 8ae3427dbfca
Step 8/8 : RUN pip install -r requirements.txt
---> Running in 555693a8d7bb
/bin/sh: pip: command not found
The command '/bin/sh -c pip install -r requirements.txt' returned a non-zero code:␣
↪127
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\centos7> docker build -t id3centos7:0.1.6 .
Sending build context to Docker daemon  240.6kB
Step 1/7 : FROM centos:7
---> ff426288ea90
Step 2/7 : RUN localedef -i fr_FR -c -f UTF-8 -A /usr/share/locale/locale.alias fr_
↪FR.UTF-8
---> Using cache
---> b7dac1f044e3
Step 3/7 : ENV LANG fr_FR.utf8
---> Using cache
---> e28a88050b8f
Step 4/7 : RUN yum update -y     && yum install -y https://centos7.iuscommunity.
↪org/ius-release.rpm     && yum install -y python36u python36u-libs python36u-
↪devel python36u-pip     && yum install -y which gcc     && yum install -y␣
↪openldap-devel
---> Using cache
---> 0cfdf4200049
Step 5/7 : WORKDIR /opt/intranet
Removing intermediate container 2af4e31fb8ed
---> 7fb09cc14c29
Step 6/7 : COPY requirements.txt /opt/intranet/
---> eecebec115f4
Step 7/7 : RUN python3.6 -m pip install -r requirements.txt
---> Running in 8400df97d2aa
Collecting arrow==0.12.1 (from -r requirements.txt (line 1))
Downloading arrow-0.12.1.tar.gz (65kB)
Collecting babel==2.5.3 (from -r requirements.txt (line 2))
```

(continues on next page)

(continued from previous page)

```
Downloading Babel-2.5.3-py2.py3-none-any.whl (6.8MB)
Collecting certifi==2018.1.18 (from -r requirements.txt (line 3))
Downloading certifi-2018.1.18-py2.py3-none-any.whl (151kB)
Collecting chardet==3.0.4 (from -r requirements.txt (line 4))
Downloading chardet-3.0.4-py2.py3-none-any.whl (133kB)
Collecting dateparser==0.6.0 (from -r requirements.txt (line 5))
Downloading dateparser-0.6.0-py2.py3-none-any.whl (68kB)
Collecting diff-match-patch==20121119 (from -r requirements.txt (line 6))
Downloading diff-match-patch-20121119.tar.gz (54kB)
Collecting django==2.0.2 (from -r requirements.txt (line 7))
Downloading Django-2.0.2-py3-none-any.whl (7.1MB)
Collecting django-ajax-selects==1.7.0 (from -r requirements.txt (line 8))
Downloading django_ajax_selects-1.7.0-py3-none-any.whl
Collecting django-autocomplete-light==3.2.10 (from -r requirements.txt (line 9))
Downloading django-autocomplete-light-3.2.10.tar.gz (428kB)
Collecting django-bootstrap4==0.0.5 (from -r requirements.txt (line 10))
Downloading django-bootstrap4-0.0.5.tar.gz
Collecting django-braces==1.12.0 (from -r requirements.txt (line 11))
Downloading django_braces-1.12.0-py2.py3-none-any.whl
Collecting django-countries==5.1.1 (from -r requirements.txt (line 12))
Downloading django_countries-5.1.1-py2.py3-none-any.whl (682kB)
Collecting django-crispy-forms==1.7.0 (from -r requirements.txt (line 13))
Downloading django_crispy_forms-1.7.0-py2.py3-none-any.whl (104kB)
Collecting django-embed-video==1.1.2 (from -r requirements.txt (line 14))
Downloading django-embed-video-1.1.2.tar.gz
Collecting django-environ==0.4.4 (from -r requirements.txt (line 15))
Downloading django_environ-0.4.4-py2.py3-none-any.whl
Collecting django-extended-choices==1.2 (from -r requirements.txt (line 16))
Downloading django_extended_choices-1.2-py2.py3-none-any.whl
Collecting django-extensions==1.9.9 (from -r requirements.txt (line 17))
Downloading django_extensions-1.9.9-py2.py3-none-any.whl (213kB)
Collecting django-import-export==0.7.0 (from -r requirements.txt (line 18))
Downloading django_import_export-0.7.0-py2.py3-none-any.whl (72kB)
Collecting django-localflavor==2.0 (from -r requirements.txt (line 19))
Downloading django_localflavor-2.0-py2.py3-none-any.whl (2.4MB)
Collecting django-money==0.12.3 (from -r requirements.txt (line 20))
Downloading django_money-0.12.3-py2.py3-none-any.whl
Collecting django-phonenumber-field==2.0.0 (from -r requirements.txt (line 21))
Downloading django-phonenumber-field-2.0.0.tar.gz
Collecting djangorestframework==3.7.7 (from -r requirements.txt (line 22))
Downloading djangorestframework-3.7.7-py2.py3-none-any.whl (1.1MB)
Collecting et-xmlfile==1.0.1 (from -r requirements.txt (line 23))
Downloading et_xmlfile-1.0.1.tar.gz
Collecting ftfy==5.3.0 (from -r requirements.txt (line 24))
Downloading ftfy-5.3.0.tar.gz (53kB)
Collecting future==0.16.0 (from -r requirements.txt (line 25))
Downloading future-0.16.0.tar.gz (824kB)
Collecting idna==2.6 (from -r requirements.txt (line 26))
Downloading idna-2.6-py2.py3-none-any.whl (56kB)
Collecting jdcal==1.3 (from -r requirements.txt (line 27))
Downloading jdcal-1.3.tar.gz
Collecting odfpy==1.3.6 (from -r requirements.txt (line 28))
Downloading odfpy-1.3.6.tar.gz (691kB)
Collecting openpyxl==2.5.0 (from -r requirements.txt (line 29))
Downloading openpyxl-2.5.0.tar.gz (169kB)
Collecting pendulum==1.4.0 (from -r requirements.txt (line 30))
Downloading pendulum-1.4.0-cp36-cp36m-manylinux1_x86_64.whl (127kB)
Collecting phonenumberslite==8.8.10 (from -r requirements.txt (line 31))
Downloading phonenumberslite-8.8.10-py2.py3-none-any.whl (429kB)
Collecting pillow==5.0.0 (from -r requirements.txt (line 32))
Downloading Pillow-5.0.0-cp36-cp36m-manylinux1_x86_64.whl (5.9MB)
```

(continues on next page)

(continued from previous page)

```
Collecting prettytable==0.7.2 (from -r requirements.txt (line 33))
Downloading prettytable-0.7.2.zip
Collecting psycopg2==2.7.3.2 (from -r requirements.txt (line 34))
Downloading psycopg2-2.7.3.2-cp36-cp36m-manylinux1_x86_64.whl (2.7MB)
Collecting py-moneyed==0.7.0 (from -r requirements.txt (line 35))
Downloading py_moneyed-0.7.0-py3-none-any.whl
Collecting python-dateutil==2.6.1 (from -r requirements.txt (line 36))
Downloading python_dateutil-2.6.1-py2.py3-none-any.whl (194kB)
Collecting pytz==2017.3 (from -r requirements.txt (line 37))
Downloading pytz-2017.3-py2.py3-none-any.whl (511kB)
Collecting pytzdata==2018.3 (from -r requirements.txt (line 38))
Downloading pytzdata-2018.3-py2.py3-none-any.whl (492kB)
Collecting pyyaml==3.12 (from -r requirements.txt (line 39))
Downloading PyYAML-3.12.tar.gz (253kB)
Collecting regex==2018.1.10 (from -r requirements.txt (line 40))
Downloading regex-2018.01.10.tar.gz (612kB)
Collecting requests==2.18.4 (from -r requirements.txt (line 41))
Downloading requests-2.18.4-py2.py3-none-any.whl (88kB)
Collecting ruamel.yaml==0.15.35 (from -r requirements.txt (line 42))
Downloading ruamel.yaml-0.15.35-cp36-cp36m-manylinux1_x86_64.whl (558kB)
Collecting six==1.11.0 (from -r requirements.txt (line 43))
Downloading six-1.11.0-py2.py3-none-any.whl
Collecting sorl-thumbnail==12.4.1 (from -r requirements.txt (line 44))
Downloading sorl_thumbnail-12.4.1-py2.py3-none-any.whl (44kB)
Collecting sqlanydb==1.0.9 (from -r requirements.txt (line 45))
Downloading sqlanydb-1.0.9.tar.gz
Collecting tablib==0.12.1 (from -r requirements.txt (line 46))
Downloading tablib-0.12.1.tar.gz (63kB)
Collecting typing==3.6.4 (from -r requirements.txt (line 47))
Downloading typing-3.6.4-py3-none-any.whl
Collecting tzlocal==1.5.1 (from -r requirements.txt (line 48))
Downloading tzlocal-1.5.1.tar.gz
Collecting unicodecsv==0.14.1 (from -r requirements.txt (line 49))
Downloading unicodecsv-0.14.1.tar.gz
Collecting urllib3==1.22 (from -r requirements.txt (line 50))
Downloading urllib3-1.22-py2.py3-none-any.whl (132kB)
Collecting wcwidth==0.1.7 (from -r requirements.txt (line 51))
Downloading wcwidth-0.1.7-py2.py3-none-any.whl
Collecting xlrd==1.1.0 (from -r requirements.txt (line 52))
Downloading xlrd-1.1.0-py2.py3-none-any.whl (108kB)
Collecting xlwt==1.3.0 (from -r requirements.txt (line 53))
Downloading xlwt-1.3.0-py2.py3-none-any.whl (99kB)
Requirement already satisfied: setuptools in /usr/lib/python3.6/site-packages␣
↪(from django-money==0.12.3->-r requirements.txt (line 20))
Installing collected packages: six, python-dateutil, arrow, pytz, babel, certifi,␣
↪chardet, regex, ruamel.yaml, tzlocal, dateparser, diff-match-patch, django,␣
↪django-ajax-selects, django-autocomplete-light, django-bootstrap4, django-braces,
↪ django-countries, django-crispy-forms, idna, urllib3, requests, django-embed-
↪video, django-environ, future, django-extended-choices, typing, django-
↪extensions, odfpy, jdcal, et-xmlfile, openpyxl, unicodecsv, xlrd, xlwt, pyyaml,␣
↪tablib, django-import-export, django-localflavor, py-moneyed, django-money,␣
↪phonenumberslite, django-phonenumber-field, djangorestframework, wcwidth, ftfy,␣
↪pytzdata, pendulum, pillow, prettytable, psycopg2, sorl-thumbnail, sqlanydb
Running setup.py install for arrow: started
Running setup.py install for arrow: finished with status 'done'
Running setup.py install for regex: started
Running setup.py install for regex: finished with status 'done'
Running setup.py install for tzlocal: started
Running setup.py install for tzlocal: finished with status 'done'
Running setup.py install for diff-match-patch: started
Running setup.py install for diff-match-patch: finished with status 'done'
```

(continues on next page)

(continued from previous page)

```
Running setup.py install for django-autocomplete-light: started
Running setup.py install for django-autocomplete-light: finished with status 'done'
Running setup.py install for django-bootstrap4: started
Running setup.py install for django-bootstrap4: finished with status 'done'
Running setup.py install for django-embed-video: started
Running setup.py install for django-embed-video: finished with status 'done'
Running setup.py install for future: started
Running setup.py install for future: finished with status 'done'
Running setup.py install for odfpy: started
Running setup.py install for odfpy: finished with status 'done'
Running setup.py install for jdcal: started
Running setup.py install for jdcal: finished with status 'done'
Running setup.py install for et-xmlfile: started
Running setup.py install for et-xmlfile: finished with status 'done'
Running setup.py install for openpyxl: started
Running setup.py install for openpyxl: finished with status 'done'
Running setup.py install for unicodecsv: started
Running setup.py install for unicodecsv: finished with status 'done'
Running setup.py install for pyyaml: started
Running setup.py install for pyyaml: finished with status 'done'
Running setup.py install for tablib: started
Running setup.py install for tablib: finished with status 'done'
Running setup.py install for django-phonenumber-field: started
Running setup.py install for django-phonenumber-field: finished with status 'done'
Running setup.py install for ftfy: started
Running setup.py install for ftfy: finished with status 'done'
Running setup.py install for prettytable: started
Running setup.py install for prettytable: finished with status 'done'
Running setup.py install for sqlanydb: started
Running setup.py install for sqlanydb: finished with status 'done'
Successfully installed arrow-0.12.1 babel-2.5.3 certifi-2018.1.18 chardet-3.0.4␣
↪dateparser-0.6.0 diff-match-patch-20121119 django-2.0.2 django-ajax-selects-1.7.
↪0 django-autocomplete-light-3.2.10 django-bootstrap4-0.0.5 django-braces-1.12.0␣
↪django-countries-5.1.1 django-crispy-forms-1.7.0 django-embed-video-1.1.2 django-
↪environ-0.4.4 django-extended-choices-1.2 django-extensions-1.9.9 django-import-
↪export-0.7.0 django-localflavor-2.0 django-money-0.12.3 django-phonenumber-field-
↪2.0.0 djangorestframework-3.7.7 et-xmlfile-1.0.1 ftfy-5.3.0 future-0.16.0 idna-2.
↪6 jdcal-1.3 odfpy-1.3.6 openpyxl-2.5.0 pendulum-1.4.0 phonenumberslite-8.8.10␣
↪pillow-5.0.0 prettytable-0.7.2 psycopg2-2.7.3.2 py-moneyed-0.7.0 python-dateutil-
↪2.6.1 pytz-2017.3 pytzdata-2018.3 pyyaml-3.12 regex-2018.1.10 requests-2.18.4␣
↪ruamel.yaml-0.15.35 six-1.11.0 sorl-thumbnail-12.4.1 sqlanydb-1.0.9 tablib-0.12.
↪1 typing-3.6.4 tzlocal-1.5.1 unicodecsv-0.14.1 urllib3-1.22 wcwidth-0.1.7 xlrd-1.
↪1.0 xlwt-1.3.0
Removing intermediate container 8400df97d2aa
---> bf91ebbc265a
Successfully built bf91ebbc265a
Successfully tagged id3centos7:0.1.6
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 Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\centos7>
```

4.13 Tutoriel Docker et Postgresql

See also:

- *Images PostgreSQL*

- https://wsvincent.com/django-docker-postgresql/
- https://github.com/wsvincent/djangoforbeginners
- https://stackoverflow.com/questions/29852583/docker-compose-accessing-postgres-shell-psql
- *Tutoriel Docker et Postgresql*
- *Mardi 30 janvier 2018 : écriture des fichiers Dockerfile et docker-compose.yml*
- *Images PostgreSQL*
- https://github.com/slardiere
- https://docs.postgresql.fr/10/charset.html

Contents

        * *docker-compose stop*
        * *docker-compose build*
    - *CREATE DATABASE db_id3_save WITH TEMPLATE = template0 ENCODING = 'UTF8' LC_COLLATE = 'fr_FR.UTF-8' LC_CTYPE = 'fr_FR.UTF-8';*

4.13.1 Modèle de fichier docker-compose.yml

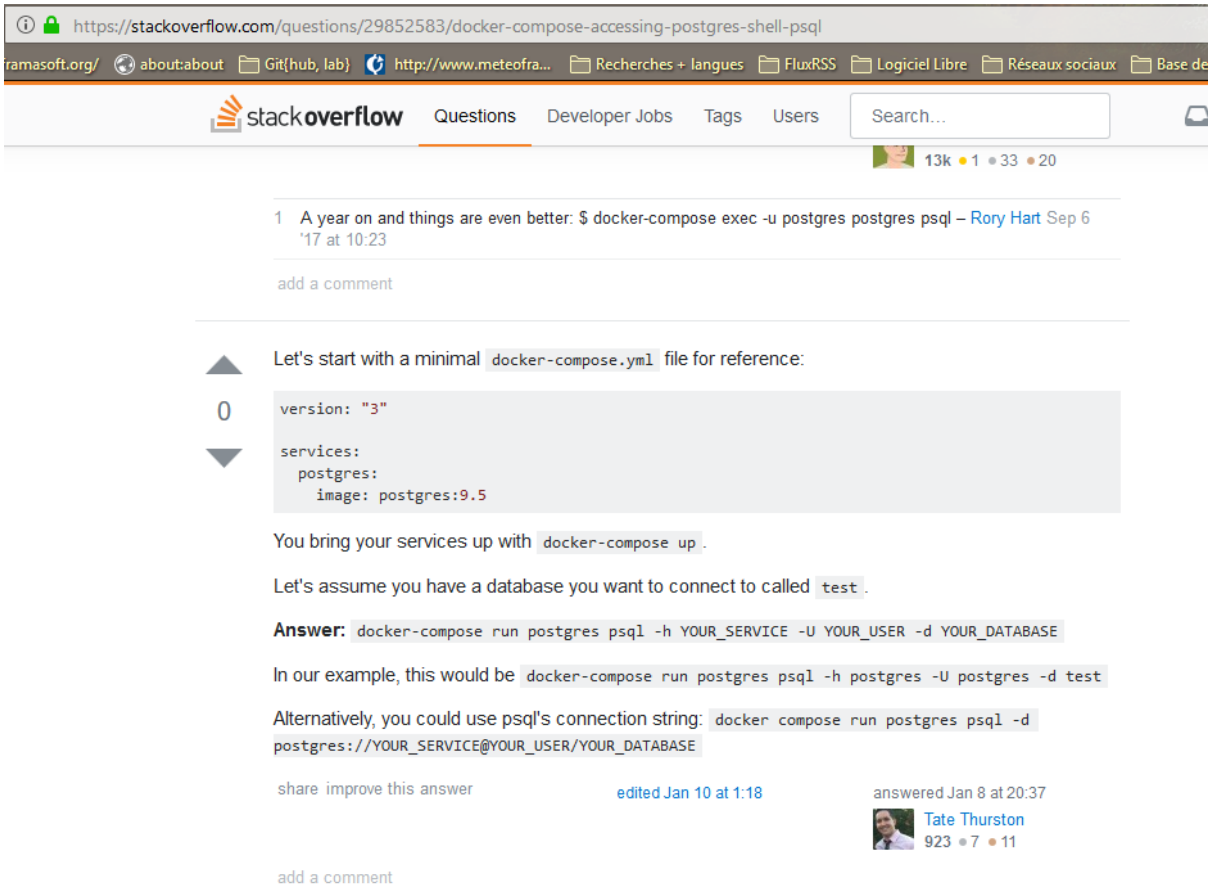


Fig. 12: stack_overflow_postgres.png

```
version: "3"

services:
  postgres:
        image: postgres:9.5
```

4.13.2 docker-compose up

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\postgresql>docker-compose up
```

```
WARNING: The Docker Engine you're using is running in swarm mode.
```

(continues on next page)

(continued from previous page)

```
Compose does not use swarm mode to deploy services to multiple nodes in a swarm.␣
↪All containers will be scheduled on the current node.

To deploy your application across the swarm, use `docker stack deploy`.

Creating network "postgresql_default" with the default driver
Pulling postgres (postgres:10)...
10: Pulling from library/postgres
Digest: sha256:3f4441460029e12905a5d447a3549ae2ac13323d045391b0cb0cf8b48ea17463
Status: Downloaded newer image for postgres:10
Creating postgresql_postgres_1 ... done
Attaching to postgresql_postgres_1
postgres_1  | The files belonging to this database system will be owned by user
↪"postgres".
postgres_1  | This user must also own the server process.
postgres_1  |
postgres_1  | The database cluster will be initialized with locale "en_US.utf8".
postgres_1  | The default database encoding has accordingly been set to "UTF8".
postgres_1  | The default text search configuration will be set to "english".
postgres_1  |
postgres_1  | Data page checksums are disabled.
postgres_1  |
postgres_1  | fixing permissions on existing directory /var/lib/postgresql/data ...
↪ ok
postgres_1  | creating subdirectories ... ok
postgres_1  | selecting default max_connections ... 100
postgres_1  | selecting default shared_buffers ... 128MB
postgres_1  | selecting dynamic shared memory implementation ... posix
postgres_1  | creating configuration files ... ok
postgres_1  | running bootstrap script ... ok
postgres_1  | performing post-bootstrap initialization ... ok
postgres_1  | syncing data to disk ...
postgres_1  | WARNING: enabling "trust" authentication for local connections
postgres_1  | You can change this by editing pg_hba.conf or using the option -A, or
postgres_1  | --auth-local and --auth-host, the next time you run initdb.
postgres_1  | ok
postgres_1  |
postgres_1  | Success. You can now start the database server using:
postgres_1  |
postgres_1  |     pg_ctl -D /var/lib/postgresql/data -l logfile start
postgres_1  |
postgres_1  | ****************************************************
postgres_1  | WARNING: No password has been set for the database.
postgres_1  |          This will allow anyone with access to the
postgres_1  |          Postgres port to access your database. In
postgres_1  |          Docker's default configuration, this is
postgres_1  |          effectively any other container on the same
postgres_1  |          system.
postgres_1  |
postgres_1  |          Use "-e POSTGRES_PASSWORD=password" to set
postgres_1  |          it in "docker run".
postgres_1  | ****************************************************
postgres_1  | waiting for server to start....2018-01-22 11:51:28.410 UTC [37] LOG:␣
↪ listening on IPv4 address "127.0.0.1", port 5432
postgres_1  | 2018-01-22 11:51:28.410 UTC [37] LOG:  could not bind IPv6 address
↪"::1": Cannot assign requested address
postgres_1  | 2018-01-22 11:51:28.410 UTC [37] HINT:  Is another postmaster␣
↪already running on port 5432? If not, wait a few seconds and retry.
postgres_1  | 2018-01-22 11:51:28.510 UTC [37] LOG:  listening on Unix socket "/
↪var/run/postgresql/.s.PGSQL.5432"
postgres_1  | 2018-01-22 11:51:28.712 UTC [38] LOG:  database system was shut down␣
↪at 2018-01-22 11:51:26 UTC
```

(continues on next page)

(continued from previous page)

```
postgres_1  | 2018-01-22 11:51:28.780 UTC [37] LOG:  database system is ready to␣
↪accept connections
postgres_1  |  done
postgres_1  | server started
postgres_1  | ALTER ROLE
postgres_1  |
postgres_1  |
postgres_1  | /usr/local/bin/docker-entrypoint.sh: ignoring /docker-entrypoint-
↪initdb.d/*
postgres_1  |
postgres_1  | 2018-01-22 11:51:28.985 UTC [37] LOG:  received fast shutdown request
postgres_1  | waiting for server to shut down....2018-01-22 11:51:29.037 UTC [37]␣
↪LOG:  aborting any active transactions
postgres_1  | 2018-01-22 11:51:29.042 UTC [37] LOG:  worker process: logical␣
↪replication launcher (PID 44) exited with exit code 1
postgres_1  | 2018-01-22 11:51:29.042 UTC [39] LOG:  shutting down
postgres_1  | 2018-01-22 11:51:29.405 UTC [37] LOG:  database system is shut down
postgres_1  |  done
postgres_1  | server stopped
postgres_1  |
postgres_1  | PostgreSQL init process complete; ready for start up.
postgres_1  |
postgres_1  | 2018-01-22 11:51:29.565 UTC [1] LOG:  listening on IPv4 address "0.0.
↪0.0", port 5432
postgres_1  | 2018-01-22 11:51:29.565 UTC [1] LOG:  listening on IPv6 address "::",
↪ port 5432
postgres_1  | 2018-01-22 11:51:29.665 UTC [1] LOG:  listening on Unix socket "/var/
↪run/postgresql/.s.PGSQL.5432"
postgres_1  | 2018-01-22 11:51:29.825 UTC [55] LOG:  database system was shut down␣
↪at 2018-01-22 11:51:29 UTC
postgres_1  | 2018-01-22 11:51:29.878 UTC [1] LOG:  database system is ready to␣
↪accept connections
```

4.13.3 docker-compose run postgres psql -h postgres -U postgres

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\postgresql>docker-compose run postgres psql -h postgres -U␣
↪postgres
```

```
psql (10.1)
Type "help" for help.

postgres=#
```

```
postgres=# help
```

```
You are using psql, the command-line interface to PostgreSQL.
Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help with psql commands
       \g or terminate with semicolon to execute query
       \q to quit
postgres=#
```

4.13.4 docker-compose down

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\postgresql>docker-compose down
```

```
Stopping postgresql_postgres_1 ... done
Removing postgresql_postgres_run_2 ... done
Removing postgresql_postgres_run_1 ... done
Removing postgresql_postgres_1     ... done
Removing network postgresql_default
```

```
postgres_1  | 2018-01-22 11:51:29.565 UTC [1] LOG:  listening on IPv4 address "0.0.
↪0.0", port 5432
postgres_1  | 2018-01-22 11:51:29.565 UTC [1] LOG:  listening on IPv6 address "::",
↪ port 5432
postgres_1  | 2018-01-22 11:51:29.665 UTC [1] LOG:  listening on Unix socket "/var/
↪run/postgresql/.s.PGSQL.5432"
postgres_1  | 2018-01-22 11:51:29.825 UTC [55] LOG:  database system was shut down␣
↪at 2018-01-22 11:51:29 UTC
postgres_1  | 2018-01-22 11:51:29.878 UTC [1] LOG:  database system is ready to␣
↪accept connections
postgres_1  | 2018-01-22 11:56:12.567 UTC [66] FATAL:  database "test" does not␣
↪exist
postgres_1  | 2018-01-22 12:08:39.698 UTC [1] LOG:  received smart shutdown request
postgres_1  | 2018-01-22 12:08:39.749 UTC [1] LOG:  worker process: logical␣
↪replication launcher (PID 61) exited with exit code 1
postgres_1  | 2018-01-22 12:08:39.750 UTC [56] LOG:  shutting down
postgres_1  | 2018-01-22 12:08:39.965 UTC [1] LOG:  database system is shut down
postgresql_postgres_1 exited with code 0
```

```
version: "3"

services:
  db:
        image: postgres:10.1
        volumes:
          - postgres_data:/var/lib/postgresql/data/
```

4.13.5 docker-compose build

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\postgresql>docker-compose build
```

```
db uses an image, skipping
```

4.13.6 docker-compose up

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\postgresql>docker-compose up
```

```
WARNING: The Docker Engine you're using is running in swarm mode.

Compose does not use swarm mode to deploy services to multiple nodes in a swarm.␣
↪All containers will be scheduled on the current node.

To deploy your application across the swarm, use `docker stack deploy`.
```

(continues on next page)

(continued from previous page)

```
Creating network "postgresql_default" with the default driver
Creating volume "postgresql_postgres_data" with default driver
Creating postgresql_db_1 ... done
Attaching to postgresql_db_1
db_1  | The files belonging to this database system will be owned by user "postgres
↪".
db_1  | This user must also own the server process.
db_1  |
db_1  | The database cluster will be initialized with locale "en_US.utf8".
db_1  | The default database encoding has accordingly been set to "UTF8".
db_1  | The default text search configuration will be set to "english".
db_1  |
db_1  | Data page checksums are disabled.
db_1  |
db_1  | fixing permissions on existing directory /var/lib/postgresql/data ... ok
db_1  | creating subdirectories ... ok
db_1  | selecting default max_connections ... 100
db_1  | selecting default shared_buffers ... 128MB
db_1  | selecting dynamic shared memory implementation ... posix
```

4.13.7 docker-compose exec -u postgres db psql

```
psql (10.1)
Type "help" for help.
```

```
postgres=# help
```

```
You are using psql, the command-line interface to PostgreSQL.
Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help with psql commands
       \g or terminate with semicolon to execute query
       \q to quit
```

```
postgres=# \h
```

```
Available help:
  ABORT                            ALTER TRIGGER                    CREATE RULE
↪                  DROP GROUP                       LISTEN
  ALTER AGGREGATE                  ALTER TYPE                       CREATE SCHEMA
↪                  DROP INDEX                       LOAD
  ALTER COLLATION                  ALTER USER                       CREATE
↪SEQUENCE                  DROP LANGUAGE                    LOCK
  ALTER CONVERSION                 ALTER USER MAPPING               CREATE SERVER
↪                  DROP MATERIALIZED VIEW           MOVE
  ALTER DATABASE                   ALTER VIEW                       CREATE
↪STATISTICS                DROP OPERATOR                    NOTIFY
  ALTER DEFAULT PRIVILEGES         ANALYZE                          CREATE
↪SUBSCRIPTION              DROP OPERATOR CLASS              PREPARE
  ALTER DOMAIN                     BEGIN                            CREATE TABLE
↪                  DROP OPERATOR FAMILY             PREPARE TRANSACTION
  ALTER EVENT TRIGGER              CHECKPOINT                       CREATE TABLE
↪AS                  DROP OWNED                       REASSIGN OWNED
  ALTER EXTENSION                  CLOSE                            CREATE
↪TABLESPACE                DROP POLICY                      REFRESH MATERIALIZED
↪VIEW
  ALTER FOREIGN DATA WRAPPER       CLUSTER                          CREATE TEXT
↪SEARCH CONFIGURATION DROP PUBLICATION                REINDEX
```

(continues on next page)

(continued from previous page)

```
  ALTER FOREIGN TABLE              COMMENT                          CREATE TEXT␣
↪SEARCH DICTIONARY    DROP ROLE                          RELEASE SAVEPOINT
  ALTER FUNCTION                   COMMIT                           CREATE TEXT␣
↪SEARCH PARSER        DROP RULE                          RESET
  ALTER GROUP                      COMMIT PREPARED                  CREATE TEXT␣
↪SEARCH TEMPLATE      DROP SCHEMA                        REVOKE
  ALTER INDEX                      COPY                             CREATE␣
↪TRANSFORM                 DROP SEQUENCE                      ROLLBACK
  ALTER LANGUAGE                   CREATE ACCESS METHOD             CREATE TRIGGER␣
↪                   DROP SERVER                        ROLLBACK PREPARED
  ALTER LARGE OBJECT               CREATE AGGREGATE                 CREATE TYPE    ␣
↪                   DROP STATISTICS                    ROLLBACK TO SAVEPOINT
  ALTER MATERIALIZED VIEW          CREATE CAST                      CREATE USER    ␣
↪                   DROP SUBSCRIPTION                  SAVEPOINT
  ALTER OPERATOR                   CREATE COLLATION                 CREATE USER␣
↪MAPPING              DROP TABLE                         SECURITY LABEL
  ALTER OPERATOR CLASS             CREATE CONVERSION                CREATE VIEW    ␣
↪                   DROP TABLESPACE                    SELECT
  ALTER OPERATOR FAMILY            CREATE DATABASE                  DEALLOCATE     ␣
↪                   DROP TEXT SEARCH CONFIGURATION     SELECT INTO
  ALTER POLICY                     CREATE DOMAIN                    DECLARE        ␣
↪                   DROP TEXT SEARCH DICTIONARY        SET
  ALTER PUBLICATION                CREATE EVENT TRIGGER             DELETE         ␣
↪                   DROP TEXT SEARCH PARSER            SET CONSTRAINTS
  ALTER ROLE                       CREATE EXTENSION                 DISCARD        ␣
↪                   DROP TEXT SEARCH TEMPLATE          SET ROLE
  ALTER RULE                       CREATE FOREIGN DATA WRAPPER      DO             ␣
↪                   DROP TRANSFORM                     SET SESSION AUTHORIZATION
  ALTER SCHEMA                     CREATE FOREIGN TABLE             DROP ACCESS␣
↪METHOD                 DROP TRIGGER                       SET TRANSACTION
  ALTER SEQUENCE                   CREATE FUNCTION                  DROP AGGREGATE␣
↪                   DROP TYPE                          SHOW
  ALTER SERVER                     CREATE GROUP                     DROP CAST      ␣
↪                   DROP USER                          START TRANSACTION
  ALTER STATISTICS                 CREATE INDEX                     DROP COLLATION␣
↪                   DROP USER MAPPING                  TABLE
  ALTER SUBSCRIPTION               CREATE LANGUAGE                  DROP␣
↪CONVERSION                 DROP VIEW                          TRUNCATE
  ALTER SYSTEM                     CREATE MATERIALIZED VIEW         DROP DATABASE  ␣
↪                   END                                UNLISTEN
  ALTER TABLE                      CREATE OPERATOR                  DROP DOMAIN    ␣
↪                   EXECUTE                            UPDATE
  ALTER TABLESPACE                 CREATE OPERATOR CLASS            DROP EVENT␣
↪TRIGGER               EXPLAIN                            VACUUM
  ALTER TEXT SEARCH CONFIGURATION  CREATE OPERATOR FAMILY           DROP EXTENSION␣
↪                   FETCH                              VALUES
  ALTER TEXT SEARCH DICTIONARY     CREATE POLICY                    DROP FOREIGN␣
↪DATA WRAPPER        GRANT                              WITH
  ALTER TEXT SEARCH PARSER         CREATE PUBLICATION               DROP FOREIGN␣
↪TABLE                IMPORT FOREIGN SCHEMA
  ALTER TEXT SEARCH TEMPLATE       CREATE ROLE                      DROP FUNCTION  ␣
↪                   INSERT
```

4.13.8 docker ps

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\postgresql>docker ps
```

```
CONTAINER ID        IMAGE               COMMAND                  CREATED
↪   STATUS              PORTS               NAMES
d205b9239366        postgres:10         "docker-entrypoint.s..."   6 minutes ago
↪     Up 6 minutes        5432/tcp            postgresql_db_1
```

4.13.9 docker exec -it d205b9239366 bash

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\postgresql>docker exec -it d205b9239366 bash
```

```
root@d205b9239366:/# ps -ef
```

```
UID          PID    PPID  C STIME TTY          TIME CMD
postgres       1       0  0 12:23 ?        00:00:00 postgres
postgres      56       1  0 12:23 ?        00:00:00 postgres: checkpointer process
postgres      57       1  0 12:23 ?        00:00:00 postgres: writer process
postgres      58       1  0 12:23 ?        00:00:00 postgres: wal writer process
postgres      59       1  0 12:23 ?        00:00:00 postgres: autovacuum launcher
↪process
postgres      60       1  0 12:23 ?        00:00:00 postgres: stats collector process
postgres      61       1  0 12:23 ?        00:00:00 postgres: bgworker: logical
↪replication launcher
postgres      66       0  0 12:28 pts/0    00:00:00 /usr/lib/postgresql/10/bin/psql
postgres      78       1  0 12:28 ?        00:00:00 postgres: postgres postgres
↪[local] idle
root         110       0  0 12:45 pts/1    00:00:00 bash
root         114     110  0 12:45 pts/1    00:00:00 ps -ef
```

```
root@d205b9239366:/# uname -a
```

```
Linux d205b9239366 4.9.60-linuxkit-aufs #1 SMP Mon Nov 6 16:00:12 UTC 2017 x86_64
↪GNU/Linux
```

```
root@d205b9239366:/# which psql
```

```
/usr/bin/psql
```

4.13.10 Mardi 30 janvier 2018

```
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\postgresql> docker ps
```

```
CONTAINER ID        IMAGE               COMMAND                  CREATED
↪   STATUS              PORTS               NAMES
02b2487f304e        postgres:10.1       "docker-entrypoint.s..."   18 seconds ago
↪     Up 16 seconds       5432/tcp            postgres_test
```

```
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\postgresql> docker exec -it 02b2487f304e bash
```

```
root@02b2487f304e:/# psql -U postgres
psql (10.1)
Type "help" for help.
```

(continues on next page)

(continued from previous page)

```
postgres=# \dt
Did not find any relations.
postgres=# \l
                                                                    List of␣
↪databases
         Name       |  Owner   | Encoding |  Collate   |   Ctype    |   Access␣
↪privileges
----------------+----------+----------+------------+------------+-----------------
↪------
 db_id3_intranet | id3admin | UTF8     | en_US.utf8 | en_US.utf8 |
 postgres        | postgres | UTF8     | en_US.utf8 | en_US.utf8 |
 template0       | postgres | UTF8     | en_US.utf8 | en_US.utf8 | =c/postgres␣
↪     +
                                  |          |            |            |            |␣
↪postgres=CTc/postgres
 template1       | postgres | UTF8     | en_US.utf8 | en_US.utf8 | =c/postgres␣
↪     +
                                  |          |            |            |            |␣
↪postgres=CTc/postgres
(4 rows)
```

4.13.10.1 docker-compose.yml

```
version: "3"

services:
  db:
        image: postgres:10.1
        container_name: container_intranet
        volumes:
          - volume_intranet:/var/lib/postgresql/data/

volumes:
  volume_intranet:
```

```
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\postgresql> docker ps
```

```
CONTAINER ID        IMAGE               COMMAND                  CREATED␣
↪   STATUS              PORTS               NAMES
47501acda106        postgres:10.1       "docker-entrypoint.s..."   15 minutes ago␣
↪     Up 15 minutes       5432/tcp            container_intranet
```

4.13.10.2 docker volume ls

```
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\postgresql> docker volume ls
```

```
DRIVER              VOLUME NAME
local               postgresql_volume_intranet
```

4.13.10.3 docker volume inspect postgresql_volume_intranet

```
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\postgresql> docker volume inspect  postgresql_volume_intranet
```

```
[
        {
                "CreatedAt": "2018-01-30T12:14:30Z",
                "Driver": "local",
                "Labels": {
                        "com.docker.compose.project": "postgresql",
                        "com.docker.compose.volume": "volume_intranet"
                },
                "Mountpoint": "/var/lib/docker/volumes/postgresql_volume_intranet/_
↪data",
                "Name": "postgresql_volume_intranet",
                "Options": {},
                "Scope": "local"
        }
]
```

4.13.10.4 docker exec -it 47501acda106 bash

```
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\postgresql> docker exec -it 47501acda106 bash
```

4.13.10.5 psql -U postgres

```
root@47501acda106:/# psql -U postgres
```

```
psql (10.1)
Type "help" for help.
```

4.13.10.6 l (liste des bases de données)

```
postgres=# \l
```

```
                                                 List of databases
   Name    |  Owner   | Encoding |  Collate   |   Ctype    |   Access privileges
-----------+----------+----------+------------+------------+-----------------------
 postgres  | postgres | UTF8     | en_US.utf8 | en_US.utf8 |
 template0 | postgres | UTF8     | en_US.utf8 | en_US.utf8 | =c/postgres          +
           |          |          |            |            | postgres=CTc/
↪postgres
 template1 | postgres | UTF8     | en_US.utf8 | en_US.utf8 | =c/postgres          +
           |          |          |            |            | postgres=CTc/
↪postgres
(3 rows)
```

4.13.10.7 CREATE USER id3admin WITH PASSWORD 'id338';

```
postgres=# CREATE USER id3admin WITH PASSWORD 'id338';
```

```
CREATE ROLE
```

4.13.10.8 CREATE DATABASE db_id3_intranet WITH OWNER = id3admin ENCODING = 'UTF8' CONNECTION LIMIT = -1;

```
postgres=# CREATE DATABASE db_id3_intranet WITH OWNER = id3admin ENCODING = 'UTF8'␣
↪CONNECTION LIMIT = -1;
```

```
CREATE DATABASE
```

4.13.10.9 l

```
postgres=# \l
```

```
                                                                    List of␣
↪databases
          Name       |  Owner   | Encoding |  Collate   |   Ctype    |   Access␣
↪privileges
-----------------+----------+----------+------------+------------+-----------------
↪------
 db_id3_intranet | id3admin | UTF8     | en_US.utf8 | en_US.utf8 |
 postgres        | postgres | UTF8     | en_US.utf8 | en_US.utf8 |
 template0       | postgres | UTF8     | en_US.utf8 | en_US.utf8 | =c/postgres     ␣
↪     +
                                  |          |          |            |            |␣
↪postgres=CTc/postgres
 template1       | postgres | UTF8     | en_US.utf8 | en_US.utf8 | =c/postgres     ␣
↪     +
                                  |          |          |            |            |␣
↪postgres=CTc/postgres
(4 rows)
```

4.13.10.10 docker-compose run db env

See also:

- https://realpython.com/blog/python/django-development-with-docker-compose-and-machine/

```
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\postgresql> docker-compose run db env
```

```
LANG=en_US.utf8
HOSTNAME=7dc6fce71c87
PG_MAJOR=10
PWD=/
HOME=/root
PG_VERSION=10.1-1.pgdg90+1
GOSU_VERSION=1.10
PGDATA=/var/lib/postgresql/data
TERM=xterm
SHLVL=0
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/lib/
↪postgresql/10/bin
```

4.13.10.11 docker-compose config

```
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\postgresql> docker-compose config
```

```
services:
  db:
        container_name: container_intranet
        environment:
          LANG: fr_FR.utf8
        image: postgres:10.1
        ports:
        - 5432:5432/tcp
        volumes:
        - volume_intranet:/var/lib/postgresql/data/:rw
        - Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\postgresql:/code:rw
version: '3.0'
volumes:
  volume_intranet: {}
```

4.13.11 Import de la base de données

```
pg_restore --dbname=db_id3_intranet  --username=id3admin  -f db_id3_intranet.sql
```

4.13.12 Mercredi 31 janvier 2018 : export/import d'une base de données PostgreSQL (tutoriel PostgreSQL)

4.13.12.1 pg_dump -U postgres –clean –create -f db.dump.sql db_id3_intranet

```
pg_dump -U postgres --clean --create -f db.dump.sql db_id3_intranet
```

```
PS C:\Tmp> pg_dump -U postgres --clean --create -f db.dump.sql db_id3_intranet
Mot de passe :
PS C:\Tmp>
```

Fig. 13: pg_dump -U postgres –clean –create -f db.dump.sql db_id3_intranet

4.13.12.2 Entête de db.dump

C'est du format texte.

```
--
-- PostgreSQL database dump
--

-- Dumped from database version 10.1
-- Dumped by pg_dump version 10.1

-- Started on 2018-01-31 10:16:48

SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SET check_function_bodies = false;
SET client_min_messages = warning;
SET row_security = off;
```

(continues on next page)

(continued from previous page)

```
DROP DATABASE db_id3_intranet;
--
-- TOC entry 3644 (class 1262 OID 16394)
-- Name: db_id3_intranet; Type: DATABASE; Schema: -; Owner: id3admin
--

CREATE DATABASE db_id3_intranet WITH TEMPLATE = template0 ENCODING = 'UTF8' LC_
↪COLLATE = 'French_France.1252' LC_CTYPE = 'French_France.1252';


ALTER DATABASE db_id3_intranet OWNER TO id3admin;

\connect db_id3_intranet

SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SET check_function_bodies = false;
SET client_min_messages = warning;
SET row_security = off;
```

4.13.12.3 Expérience substitution de db_id3_save à db_id3_intranet

On substitue db_id3_save à db_id3_intranet. On espère donc créer une copie de la base de données db_id3_intranet. Comme le fichier est au format texte, on peut utiliser psql pour l'import.

```
--
-- PostgreSQL database dump
--

-- Dumped from database version 10.1
-- Dumped by pg_dump version 10.1

SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SET check_function_bodies = false;
SET client_min_messages = warning;
SET row_security = off;

--
-- Name: db_id3_save; Type: DATABASE; Schema: -; Owner: id3admin
--

CREATE DATABASE db_id3_save WITH TEMPLATE = template0 ENCODING = 'UTF8' LC_COLLATE␣
↪= 'French_France.1252' LC_CTYPE = 'French_France.1252';


ALTER DATABASE db_id3_save OWNER TO id3admin;

\connect db_id3_save

SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET client_encoding = 'UTF8';
```

(continues on next page)

(continued from previous page)

```
SET standard_conforming_strings = on;
SET check_function_bodies = false;
SET client_min_messages = warning;
SET row_security = off;

--
-- Name: db_id3_save; Type: COMMENT; Schema: -; Owner: id3admin
--

COMMENT ON DATABASE db_id3_save IS 'La base db_id3_save';
```

4.13.12.4 psql -U postgres -f .db.dump.sql

```
psql -U postgres -f .\db.dump.sql
```

```
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
GRANT
```

OK, tout s'est bien passé.

```
PS C:\Tmp> psql -U postgres
Mot de passe pour l'utilisateur postgres :
psql (10.1)
Attention : l'encodage console (850) diffère de l'encodage Windows (1252).
            Les caractères 8 bits peuvent ne pas fonctionner correctement.
            Voir la section « Notes aux utilisateurs de Windows » de la page
            référence de psql pour les détails.
Saisissez « help » pour l'aide.

postgres=# \l
                                    Liste des bases de donnÚes
      Nom       | PropriÚtaire | Encodage |  Collationnement   |    Type caract.    |    Droits d'accÞs
----------------+--------------+----------+--------------------+--------------------+----------------------
 db_id3_intranet | id3admin     | UTF8     | French_France.1252 | French_France.1252 |
 db_id3_save     | id3admin     | UTF8     | French_France.1252 | French_France.1252 |
 db_test         | id3admin     | UTF8     | French_France.1252 | French_France.1252 |
 postgres        | postgres     | UTF8     | French_France.1252 | French_France.1252 |
 template0       | postgres     | UTF8     | French_France.1252 | French_France.1252 | =c/postgres          +
                 |              |          |                    |                    | postgres=CTc/postgres
 template1       | postgres     | UTF8     | French_France.1252 | French_France.1252 | =c/postgres          +
                 |              |          |                    |                    | postgres=CTc/postgres
(6 lignes)
```

Fig. 14: psql -U postgres -f .db.dump.sql

On voit aussi que l'encodage French_France.1252 va peut-être poser des problèmes dans l'image Docker actuelle.

```
postgres=# \l

↪Liste des bases de donnÚes
          Nom       | PropriÚtaire | Encodage |  Collationnement   |    Type␣
↪caract.    |    Droits d'accÞs
----------------+--------------+----------+--------------------+-------------------
↪--+-----------------------
 db_id3_intranet | id3admin     | UTF8     | French_France.1252 | French_France.
↪1252 |
 db_id3_save     | id3admin     | UTF8     | French_France.1252 | French_France.
↪1252 |
 db_test         | id3admin     | UTF8     | French_France.1252 | French_France.
↪1252 |
```

(continues on next page)

(continued from previous page)

```
 postgres         | postgres      | UTF8      | French_France.1252 | French_France.
↪1252 |
 template0        | postgres      | UTF8      | French_France.1252 | French_France.
↪1252 | =c/postgres          +
                                  |               |          |                    | ␣
↪                   | postgres=CTc/postgres
 template1        | postgres      | UTF8      | French_France.1252 | French_France.
↪1252 | =c/postgres          +
                                  |               |          |                    | ␣
↪                   | postgres=CTc/postgres
(6 lignes)
```

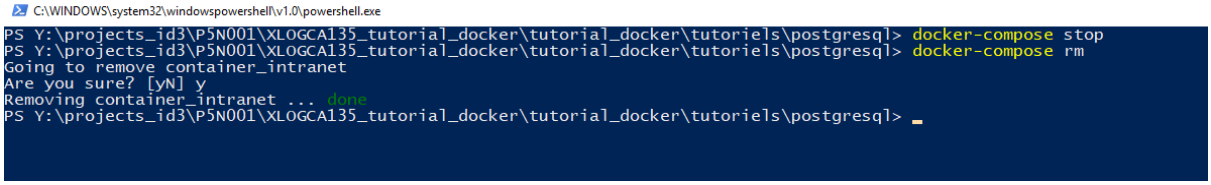
Sur Docker on a:

```
root@02b2487f304e:/# psql -U postgres
psql (10.1)
Type "help" for help.

postgres=# \dt
Did not find any relations.
postgres=# \l
                                                                        List of␣
↪databases
          Name       |  Owner   | Encoding |  Collate   |   Ctype    |   Access␣
↪privileges
---------------------+----------+----------+------------+------------+-----------------
↪------
 db_id3_intranet     | id3admin | UTF8     | en_US.utf8 | en_US.utf8 |
 postgres            | postgres | UTF8     | en_US.utf8 | en_US.utf8 |
 template0           | postgres | UTF8     | en_US.utf8 | en_US.utf8 | =c/postgres     ␣
↪     +
                                |          |            |            |            |␣
↪postgres=CTc/postgres
 template1           | postgres | UTF8     | en_US.utf8 | en_US.utf8 | =c/postgres     ␣
↪     +
                                |          |            |            |            |␣
↪postgres=CTc/postgres
(4 rows)
```

On suit les conseils donnés *ici*: On essaye déjà avec la langue allemande et on essayera avec *French_France.1252* ?

Dockerfile:

```
FROM postgres:10.1
RUN localedef -i de_DE -c -f UTF-8 -A /usr/share/locale/locale.alias de_DE.UTF-8
ENV LANG de_DE.utf8
```



4.13.12.5 docker-compose stop

```
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\postgresql> docker-compose stop
```

```
Stopping container_intranet ... done
```

4.13.12.6 docker-compose build

```
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\postgresql> docker-compose build
```

```
Building db
Step 1/3 : FROM postgres:10.1
 ---> ec61d13c8566
Step 2/3 : RUN localedef -i de_DE -c -f UTF-8 -A /usr/share/locale/locale.alias de_
↪DE.UTF-8
 ---> Running in 19e95836a1ce
Removing intermediate container 19e95836a1ce
 ---> 331ee9213868
Step 3/3 : ENV LANG de_DE.utf8
 ---> Running in 852054da9e27
Removing intermediate container 852054da9e27
 ---> 56dd534c98f7
Successfully built 56dd534c98f7
Successfully tagged postgres:10.1
```

4.13.13 CREATE DATABASE db_id3_save WITH TEMPLATE = template0 ENCODING = 'UTF8' LC_COLLATE = 'fr_FR.UTF-8' LC_CTYPE = 'fr_FR.UTF-8';

```
postgres=# CREATE DATABASE db_id3_save WITH TEMPLATE = template0 ENCODING = 'UTF8'␣
↪LC_COLLATE = 'fr_FR.UTF-8' LC_CTYPE = 'fr_FR.UTF-8';
```

```
CREATE DATABASE
```

4.14 Docker OpenLDAP

See also:

- https://github.com/osixia/docker-openldap

Contents

Chapter 5

Exemples Docker labs

See also:

- https://docs.docker.com/samples/#tutorial-labs

5.1 Samples Docker labs

5.1.1 Samples Docker labs beginner

See also:

- https://github.com/docker/labs/tree/master/beginner/
- https://hub.docker.com/_/hello-world/
- https://raw.githubusercontent.com/docker-library/hello-world/master/hello.c

Contents

5.1.1.1 Setup

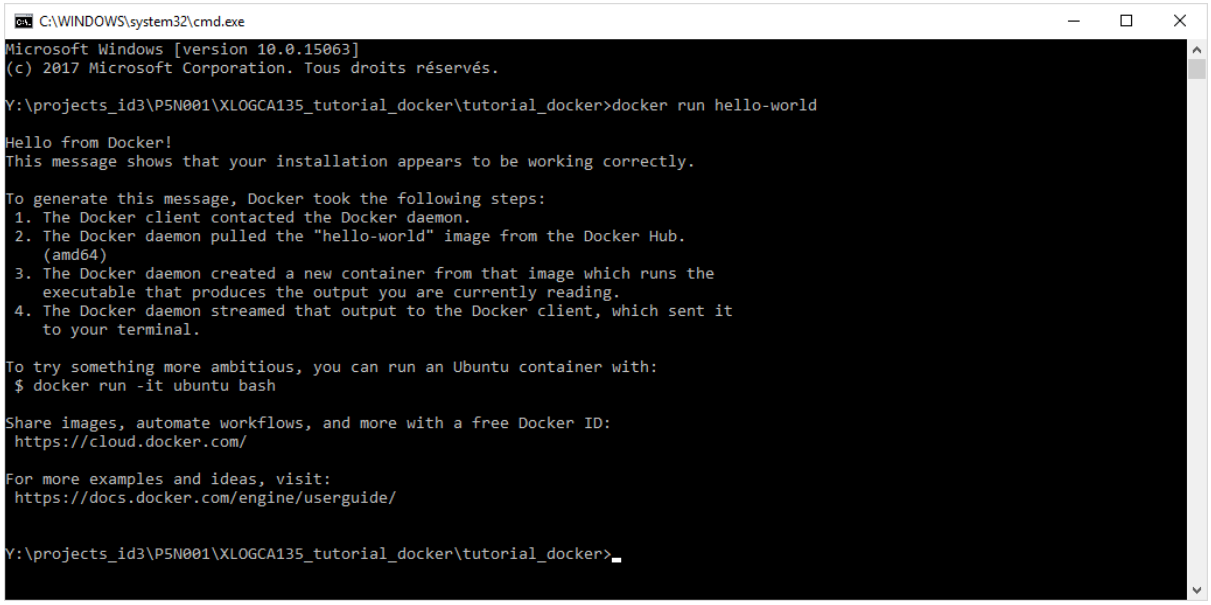
See also:

- https://github.com/docker/labs/blob/master/beginner/chapters/setup.md

5.1.1.2 docker run hello-world

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_docker>docker run hello-
↪world
```

```
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.
        (amd64)
 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/
```



5.1.1.2.1 hello.c

See also:

- https://github.com/docker-library/hello-world/blob/master/hello.c

```
    //#include <unistd.h>
    #include <sys/syscall.h>

    #ifndef DOCKER_IMAGE
            #define DOCKER_IMAGE "hello-world"
    #endif

    #ifndef DOCKER_GREETING
            #define DOCKER_GREETING "Hello from Docker!"
    #endif

    #ifndef DOCKER_ARCH
            #define DOCKER_ARCH "amd64"
    #endif

    const char message[] =
            "\n"
            DOCKER_GREETING "\n"
            "This message shows that your installation appears to be working correctly.\n"
            "\n"
            "To generate this message, Docker took the following steps:\n"
            " 1. The Docker client contacted the Docker daemon.\n"
            " 2. The Docker daemon pulled the \"" DOCKER_IMAGE "\" image from the Docker Hub.\n"
            "    (" DOCKER_ARCH ")\n"
            " 3. The Docker daemon created a new container from that image which runs the\n"
            "    executable that produces the output you are currently reading.\n"
            " 4. The Docker daemon streamed that output to the Docker client, which sent it\n"
            "    to your terminal.\n"
            "\n"
            "To try something more ambitious, you can run an Ubuntu container with:\n"
            " $ docker run -it ubuntu bash\n"
            "\n"
            "Share images, automate workflows, and more with a free Docker ID:\n"
            " https://cloud.docker.com/\n"
            "\n"
            "For more examples and ideas, visit:\n"
            " https://docs.docker.com/engine/userguide/\n"
            "\n";

    void _start() {
            //write(1, message, sizeof(message) - 1);
            syscall(SYS_write, 1, message, sizeof(message) - 1);

            //_exit(0);
            syscall(SYS_exit, 0);
    }
```

5.1.1.2.2 Dockerfile.build

```
# explicitly use Debian for maximum cross-architecture compatibility
FROM debian:stretch-slim

RUN dpkg --add-architecture i386

RUN apt-get update && apt-get install -y --no-install-recommends \
```

(continues on next page)

(continued from previous page)

```
                gcc \
                libc6-dev \
                make \
                \
                libc6-dev:i386 \
                libgcc-6-dev:i386 \
                \
                libc6-dev-arm64-cross \
                libc6-dev-armel-cross \
                libc6-dev-armhf-cross \
                libc6-dev-ppc64el-cross \
                libc6-dev-s390x-cross \
                \
                gcc-aarch64-linux-gnu \
                gcc-arm-linux-gnueabi \
                gcc-arm-linux-gnueabihf \
                gcc-powerpc64le-linux-gnu \
                gcc-s390x-linux-gnu \
                \
                file \
        && rm -rf /var/lib/apt/lists/*

WORKDIR /usr/src/hello
COPY . .

RUN set -ex; \
        make clean all test \
                TARGET_ARCH='amd64' \
                CC='x86_64-linux-gnu-gcc' \
                STRIP='x86_64-linux-gnu-strip'

RUN set -ex; \
        make clean all \
                TARGET_ARCH='arm32v5' \
                CC='arm-linux-gnueabi-gcc' \
                STRIP='arm-linux-gnueabi-strip'

RUN set -ex; \
        make clean all \
                TARGET_ARCH='arm32v7' \
                CC='arm-linux-gnueabihf-gcc' \
                STRIP='arm-linux-gnueabihf-strip'

RUN set -ex; \
        make clean all \
                TARGET_ARCH='arm64v8' \
                CC='aarch64-linux-gnu-gcc' \
                STRIP='aarch64-linux-gnu-strip'

RUN set -ex; \
        make clean all test \
                TARGET_ARCH='i386' \
                CC='gcc -m32 -L/usr/lib/gcc/i686-linux-gnu/6' \
                STRIP='x86_64-linux-gnu-strip'

RUN set -ex; \
        make clean all \
                TARGET_ARCH='ppc64le' \
                CC='powerpc64le-linux-gnu-gcc' \
                STRIP='powerpc64le-linux-gnu-strip'
```

(continues on next page)

(continued from previous page)

```
RUN set -ex; \
        make clean all \
                TARGET_ARCH='s390x' \
                CC='s390x-linux-gnu-gcc' \
                STRIP='s390x-linux-gnu-strip'

RUN find \( -name 'hello' -or -name 'hello.txt' \) -exec file '{}' + -exec ls -lh '
↪{}' +

CMD ["./amd64/hello-world/hello"]
```

5.1.1.3 Running your first container : docker pull alpine

See also:

- https://github.com/docker/labs/blob/master/beginner/chapters/alpine.md
- *Images Alpine*

5.1.1.3.1 docker pull alpine

```
docker pull alpine
```

5.1.1.3.2 docker images

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_docker>docker images
```

```
REPOSITORY                  TAG                 IMAGE ID            CREATED
↪     SIZE
id3pvergain/get-started     part2               ed5b70620e49        25 hours ago
↪     148MB
friendlyhello               latest              ed5b70620e49        25 hours ago
↪     148MB
alpine                      latest              3fd9065eaf02        6 days ago
↪     4.15MB
wordpress                   latest              28084cde273b        7 days ago
↪     408MB
centos                      latest              ff426288ea90        7 days ago
↪     207MB
nginx                       latest              3f8a4339aadd        2 weeks ago
↪     108MB
ubuntu                      latest              00fd29ccc6f1        4 weeks ago
↪     111MB
python                      2.7-slim            4fd30fc83117        5 weeks ago
↪     138MB
hello-world                 latest              f2a91732366c        8 weeks ago
↪     1.85kB
docker4w/nsenter-dockerd    latest              cae870735e91        2 months ago
↪     187kB
```

5.1.1.3.3 docker run alpine ls -l

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_docker>docker run alpine
↪ls -l
```

```
total 52
drwxr-xr-x    2 root     root          4096 Jan  9 19:37 bin
drwxr-xr-x    5 root     root           340 Jan 16 08:57 dev
drwxr-xr-x    1 root     root          4096 Jan 16 08:57 etc
drwxr-xr-x    2 root     root          4096 Jan  9 19:37 home
drwxr-xr-x    5 root     root          4096 Jan  9 19:37 lib
drwxr-xr-x    5 root     root          4096 Jan  9 19:37 media
drwxr-xr-x    2 root     root          4096 Jan  9 19:37 mnt
dr-xr-xr-x  127 root     root             0 Jan 16 08:57 proc
drwx------    2 root     root          4096 Jan  9 19:37 root
drwxr-xr-x    2 root     root          4096 Jan  9 19:37 run
drwxr-xr-x    2 root     root          4096 Jan  9 19:37 sbin
drwxr-xr-x    2 root     root          4096 Jan  9 19:37 srv
dr-xr-xr-x   13 root     root             0 Jan 15 15:33 sys
drwxrwxrwt    2 root     root          4096 Jan  9 19:37 tmp
drwxr-xr-x    7 root     root          4096 Jan  9 19:37 usr
drwxr-xr-x   11 root     root          4096 Jan  9 19:37 var
```

What happened? Behind the scenes, a lot of stuff happened. When you call run:

- The Docker client contacts the Docker daemon
- The Docker daemon checks local store if the image (alpine in this case) is available locally, and if not, downloads it from Docker Store. (Since we have issued docker pull alpine before, the download step is not necessary)
- The Docker daemon creates the container and then runs a command in that container.
- The Docker daemon streams the output of the command to the Docker client

When you run docker run alpine, you provided a command (ls -l), so Docker started the command specified and you saw the listing.

5.1.1.3.4 docker ps -a

Liste des conteneurs qui ont tourné à un moment donné.

```
C:\Tmp>docker ps -a
```

```
CONTAINER ID        IMAGE                                COMMAND
↪CREATED             STATUS                          PORTS                    NAMES
cb62ace67ba4        alpine                               "ls -l"                  20␣
↪minutes ago      Exited (0) 20 minutes ago                                 eager_
↪heisenberg
685915373a4c        hello-world                          "/hello"                 2␣
↪hours ago         Exited (0) 2 hours ago                                    gallant_
↪wright
e150d0531321        alpine                               "/bin/sh"                18␣
↪hours ago        Exited (0) 18 hours ago                                   objective_
↪curran
7d6e93a39de5        alpine                               "/bin/sh"                18␣
↪hours ago        Exited (0) 18 hours ago                                   amazing_knuth
807d38ada261        ubuntu                               "/bin/bash"              18␣
↪hours ago        Exited (127) 18 hours ago                                 confident_
↪bassi
eebf7e801b96        ubuntu                               "/bin/bash"              18␣
↪hours ago        Exited (0) 13 minutes ago                                 wonderful_
↪blackwell
c31e71b41bdb        id3pvergain/get-started:part2        "python app.py"          22␣
↪hours ago        Exited (137) 20 hours ago                                 ␣
↪getstartedlab_web.3.kv05oigiytufm5wsuvnp4guoj
8780b68999cf        id3pvergain/get-started:part2        "python app.py"          22␣
↪hours ago        Exited (137) 20 hours ago                                 ␣
↪getstartedlab_web.4.as0f73cwv518fibwnjd60yfyw
```

(continues on next page)

(continued from previous page)

```
f45453da50cf        id3pvergain/get-started:part2   "python app.py"          23␣
↪hours ago          Exited (137) 20 hours ago                                youthful_
↪wilson
b47fd081642e        id3pvergain/get-started:part2   "python app.py"          23␣
↪hours ago          Exited (137) 20 hours ago                                admiring_
↪lumiere
06193b763075        friendlyhello                   "python app.py"          24␣
↪hours ago          Exited (137) 23 hours ago                                boring_
↪goodall
16eca9f1274e        friendlyhello                   "python app.py"          26␣
↪hours ago          Exited (255) 24 hours ago   0.0.0.0:4000->80/tcp         stoic_lalande
fb92255412cf        hello-world                     "/hello"                 3␣
↪days ago           Exited (0) 3 days ago                                    infallible_
↪kepler
dd8ca306fb5b        hello-world                     "/hello"                 4␣
↪days ago           Exited (0) 4 days ago                                    musing_
↪hopper
4d1e5f24ba8e        nginx                           "nginx -g 'daemon of..."   4␣
↪days ago           Exited (0) 4 days ago                                    webserver
```

5.1.1.3.5 docker run -it alpine /bin/sh

```
C:\Tmp>docker run -it alpine /bin/sh
```

```
/ # uname -a
```

```
Linux 2b8fff5f4068 4.9.60-linuxkit-aufs #1 SMP Mon Nov 6 16:00:12 UTC 2017 x86_64␣
↪Linux
```

```
/ # ls
```

```
bin    dev    etc    home   lib    media  mnt    proc   root   run    sbin   srv  ␣
↪ sys    tmp    usr    var
```

Running the run command with the -it flags attaches us to an interactive tty in the container. Now you can run as many commands in the container as you want. Take some time to run your favorite commands.

That concludes a whirlwind tour of the docker run command which would most likely be the command you'll use most often.

It makes sense to spend some time getting comfortable with it.

To find out more about run, use docker run –help to see a list of all flags it supports.

As you proceed further, we'll see a few more variants of docker run.

5.1.1.4 docker run –help

```
Usage:  docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

Run a command in a new container

Options:
          --add-host list                  Add a custom host-to-IP mapping
                                                                           ␣
↪(host:ip)
  -a, --attach list                    Attach to STDIN, STDOUT or STDERR
```

(continues on next page)

(continued from previous page)

```
          --blkio-weight uint16            Block IO (relative weight),
                                                                    between␣
↪10 and 1000, or 0 to
                                                                    disable␣
↪(default 0)
          --blkio-weight-device list       Block IO weight (relative device
                                                                    weight)␣
↪(default [])
          --cap-add list                   Add Linux capabilities
          --cap-drop list                  Drop Linux capabilities
          --cgroup-parent string           Optional parent cgroup for the
                                                                           ␣
↪container
          --cidfile string                 Write the container ID to the file
          --cpu-period int                 Limit CPU CFS (Completely Fair
                                                                           ␣
↪Scheduler) period
          --cpu-quota int                  Limit CPU CFS (Completely Fair
                                                                           ␣
↪Scheduler) quota
          --cpu-rt-period int              Limit CPU real-time period in
                                                                           ␣
↪microseconds
          --cpu-rt-runtime int             Limit CPU real-time runtime in
                                                                           ␣
↪microseconds
  -c, --cpu-shares int                 CPU shares (relative weight)
          --cpus decimal                   Number of CPUs
          --cpuset-cpus string             CPUs in which to allow execution
                                                                    (0-3, 0,
↪1)
          --cpuset-mems string             MEMs in which to allow execution
                                                                    (0-3, 0,
↪1)
  -d, --detach                         Run container in background and
                                                                    print␣
↪container ID
          --detach-keys string             Override the key sequence for
                                                                           ␣
↪detaching a container
          --device list                    Add a host device to the container
          --device-cgroup-rule list        Add a rule to the cgroup allowed
                                                                    devices␣
↪list
          --device-read-bps list           Limit read rate (bytes per second)
                                                                    from a␣
↪device (default [])
          --device-read-iops list          Limit read rate (IO per second)
                                                                    from a␣
↪device (default [])
          --device-write-bps list          Limit write rate (bytes per
                                                                    second)␣
↪to a device (default [])
          --device-write-iops list         Limit write rate (IO per second)
                                                                    to a␣
↪device (default [])
          --disable-content-trust          Skip image verification (default true)
          --dns list                       Set custom DNS servers
          --dns-option list                Set DNS options
          --dns-search list                Set custom DNS search domains
          --entrypoint string              Overwrite the default ENTRYPOINT
```

(continues on next page)

(continued from previous page)

```
                                                                    of the␣
↪image
  -e, --env list                       Set environment variables
          --env-file list                  Read in a file of environment variables
          --expose list                    Expose a port or a range of ports
          --group-add list                 Add additional groups to join
          --health-cmd string              Command to run to check health
          --health-interval duration       Time between running the check
                                                                           ␣
↪(ms|s|m|h) (default 0s)
          --health-retries int             Consecutive failures needed to
                                                                    report␣
↪unhealthy
          --health-start-period duration   Start period for the container to
                                                                           ␣
↪initialize before starting
                                                                    health-
↪retries countdown
                                                                           ␣
↪(ms|s|m|h) (default 0s)
          --health-timeout duration        Maximum time to allow one check to
                                                                    run␣
↪(ms|s|m|h) (default 0s)
          --help                           Print usage
  -h, --hostname string                Container host name
          --init                           Run an init inside the container
                                                                    that␣
↪forwards signals and reaps
                                                                           ␣
↪processes
  -i, --interactive                    Keep STDIN open even if not attached
          --ip string                      IPv4 address (e.g., 172.30.100.104)
          --ip6 string                     IPv6 address (e.g., 2001:db8::33)
          --ipc string                     IPC mode to use
          --isolation string               Container isolation technology
          --kernel-memory bytes            Kernel memory limit
  -l, --label list                     Set meta data on a container
          --label-file list                Read in a line delimited file of labels
          --link list                      Add link to another container
          --link-local-ip list             Container IPv4/IPv6 link-local
                                                                           ␣
↪addresses
          --log-driver string              Logging driver for the container
          --log-opt list                   Log driver options
          --mac-address string             Container MAC address (e.g.,
                                                                           ␣
↪92:d0:c6:0a:29:33)
  -m, --memory bytes                   Memory limit
          --memory-reservation bytes       Memory soft limit
          --memory-swap bytes              Swap limit equal to memory plus
                                                                    swap: '-
↪1' to enable unlimited swap
          --memory-swappiness int          Tune container memory swappiness
                                                                    (0 to␣
↪100) (default -1)
          --mount mount                    Attach a filesystem mount to the
                                                                           ␣
↪container
          --name string                    Assign a name to the container
          --network string                 Connect a container to a network
                                                                           ␣
↪(default "default")
```

(continues on next page)

(continued from previous page)

```
          --network-alias list              Add network-scoped alias for the
                                                                              ␣
↪container
          --no-healthcheck                  Disable any container-specified
                                                                              ␣
↪HEALTHCHECK
          --oom-kill-disable                Disable OOM Killer
          --oom-score-adj int               Tune host's OOM preferences (-1000
                                                                              to 1000)
          --pid string                      PID namespace to use
          --pids-limit int                  Tune container pids limit (set -1
                                                                              for␣
↪unlimited)
          --platform string                 Set platform if server is
                                                                              multi-
↪platform capable
          --privileged                      Give extended privileges to this
                                                                              ␣
↪container
  -p, --publish list                    Publish a container's port(s) to
                                                                              the host
  -P, --publish-all                     Publish all exposed ports to
                                                                              random␣
↪ports
          --read-only                       Mount the container's root
                                                                              ␣
↪filesystem as read only
          --restart string                  Restart policy to apply when a
                                                                              ␣
↪container exits (default "no")
          --rm                              Automatically remove the container
                                                                              when it␣
↪exits
          --runtime string                  Runtime to use for this container
          --security-opt list               Security Options
          --shm-size bytes                  Size of /dev/shm
          --sig-proxy                       Proxy received signals to the
                                                                              process␣
↪(default true)
          --stop-signal string              Signal to stop a container
                                                                              ␣
↪(default "15")
          --stop-timeout int                Timeout (in seconds) to stop a
                                                                              ␣
↪container
          --storage-opt list                Storage driver options for the
                                                                              ␣
↪container
          --sysctl map                      Sysctl options (default map[])
          --tmpfs list                      Mount a tmpfs directory
  -t, --tty                             Allocate a pseudo-TTY
          --ulimit ulimit                   Ulimit options (default [])
  -u, --user string                     Username or UID (format:

↪<name|uid>[:<group|gid>])
          --userns string                   User namespace to use
          --uts string                      UTS namespace to use
  -v, --volume list                     Bind mount a volume
          --volume-driver string            Optional volume driver for the
                                                                              ␣
↪container
```

(continues on next page)

(continued from previous page)

```
          --volumes-from list              Mount volumes from the specified
                                                                            ␣
↪container(s)
  -w, --workdir string                  Working directory inside the container
```

5.1.1.5 docker inspect alpine

```
C:\Tmp>docker inspect alpine
```

```
[
        {
                "Id":
↪"sha256:3fd9065eaf02feaf94d68376da52541925650b81698c53c6824d92ff63f98353",
                "RepoTags": [
                        "alpine:latest"
                ],
                "RepoDigests": [

↪"alpine@sha256:7df6db5aa61ae9480f52f0b3a06a140ab98d427f86d8d5de0bedab9b8df6b1c0"
                ],
                "Parent": "",
                "Comment": "",
                "Created": "2018-01-09T21:10:58.579708634Z",
                "Container":
↪"30e1a2427aa2325727a092488d304505780501585a6ccf5a6a53c4d83a826101",
                "ContainerConfig": {
                        "Hostname": "30e1a2427aa2",
                        "Domainname": "",
                        "User": "",
                        "AttachStdin": false,
                        "AttachStdout": false,
                        "AttachStderr": false,
                        "Tty": false,
                        "OpenStdin": false,
                        "StdinOnce": false,
                        "Env": [
                                "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/
↪usr/bin:/sbin:/bin"
                        ],
                        "Cmd": [
                                "/bin/sh",
                                "-c",
                                "#(nop) ",
                                "CMD [\"/bin/sh\"]"
                        ],
                        "ArgsEscaped": true,
                        "Image":
↪"sha256:fbef17698ac8605733924d5662f0cbfc0b27a51e83ab7d7a4b8d8a9a9fe0d1c2",
                        "Volumes": null,
                        "WorkingDir": "",
                        "Entrypoint": null,
                        "OnBuild": null,
                        "Labels": {}
                },
                "DockerVersion": "17.06.2-ce",
                "Author": "",
                "Config": {
                        "Hostname": "",
                        "Domainname": "",
```

(continues on next page)

(continued from previous page)


```
                        "User": "",
                        "AttachStdin": false,
                        "AttachStdout": false,
                        "AttachStderr": false,
                        "Tty": false,
                        "OpenStdin": false,
                        "StdinOnce": false,
                        "Env": [
                                "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/
↪usr/bin:/sbin:/bin"
                        ],
                        "Cmd": [
                                "/bin/sh"
                        ],
                        "ArgsEscaped": true,
                        "Image":
↪"sha256:fbef17698ac8605733924d5662f0cbfc0b27a51e83ab7d7a4b8d8a9a9fe0d1c2",
                        "Volumes": null,
                        "WorkingDir": "",
                        "Entrypoint": null,
                        "OnBuild": null,
                        "Labels": null
                },
                "Architecture": "amd64",
                "Os": "linux",
                "Size": 4147781,
                "VirtualSize": 4147781,
                "GraphDriver": {
                        "Data": {
                                "MergedDir": "/var/lib/docker/overlay2/
↪e4af82b9362c03a84a71a8449c41a37c94592f1e5c2ef1d4f43a255b0a4ee2bd/merged",
                                "UpperDir": "/var/lib/docker/overlay2/
↪e4af82b9362c03a84a71a8449c41a37c94592f1e5c2ef1d4f43a255b0a4ee2bd/diff",
                                "WorkDir": "/var/lib/docker/overlay2/
↪e4af82b9362c03a84a71a8449c41a37c94592f1e5c2ef1d4f43a255b0a4ee2bd/work"
                        },
                        "Name": "overlay2"
                },
                "RootFS": {
                        "Type": "layers",
                        "Layers": [

↪"sha256:cd7100a72410606589a54b932cabd804a17f9ae5b42a1882bd56d263e02b6215"
                        ]
                },
                "Metadata": {
                        "LastTagTime": "0001-01-01T00:00:00Z"
                }
        }
]
```


5.1.1.6 Next Steps: 2.0 Webapps with Docker

See also:

https://github.com/docker/labs/blob/master/beginner/chapters/webapps.md

For the next step in the tutorial, head over to *2.0 Webapps with Docker*.

5.1.2 2) Webapps with Docker (Python + Flask)

See also:

- https://github.com/docker/labs/blob/master/beginner/chapters/webapps.md
- https://github.com/docker/labs/tree/master/beginner/static-site
- https://hub.docker.com/r/dockersamples/static-site/

Contents

5.1.2.1 Introduction

Great! So you have now looked at docker run, played with a Docker container and also got the hang of some terminology.

Armed with all this knowledge, you are now ready to get to the real stuff, deploying web applications with Docker.

5.1.2.2 Run a static website in a container : docker run -d dockersamples/static-site

Note: Code for this section is in this repo in the static-site directory[12].

Let's start by taking baby-steps. First, we'll use Docker to run a static website in a container.

The website is based on an existing image.

We'll pull a Docker image from Docker Store, run the container, and see how easy it is to set up a web server.

The image that you are going to use is a single-page website that was already created for this demo and is available on the Docker Store as dockersamples/static-site.

You can download and run the image directly in one go using docker run as follows:

```
docker run -d dockersamples/static-site
```

```
C:\Tmp>docker run -d dockersamples/static-site
```

```
Unable to find image 'dockersamples/static-site:latest' locally
latest: Pulling from dockersamples/static-site
fdd5d7827f33: Pull complete
a3ed95caeb02: Pull complete
716f7a5f3082: Pull complete
7b10f03a0309: Pull complete
aff3ab7e9c39: Pull complete
Digest: sha256:daa686c61d7d239b7977e72157997489db49f316b9b9af3909d9f10fd28b2dec
Status: Downloaded newer image for dockersamples/static-site:latest
3bf76a82d6127dfd775f0eb6a5ed20ce275ad7eaf02b18b2ce50bd96df1432ba
```

[12] https://github.com/docker/labs/tree/master/beginner/static-site

5.1.2.3 docker images

```
docker images
```

```
REPOSITORY                    TAG                 IMAGE ID            CREATED
↪      SIZE
ubuntu                        trusty              02a63d8b2bfa        17 hours ago
↪      222MB
id3pvergain/get-started       part2               ed5b70620e49        31 hours ago
↪      148MB
friendlyhello                 latest              ed5b70620e49        31 hours ago
↪      148MB
alpine                        latest              3fd9065eaf02        6 days ago
↪      4.15MB
wordpress                     latest              28084cde273b        7 days ago
↪      408MB
centos                        latest              ff426288ea90        7 days ago
↪      207MB
nginx                         latest              3f8a4339aadd        2 weeks ago
↪      108MB
ubuntu                        latest              00fd29ccc6f1        4 weeks ago
↪      111MB
python                        2.7-slim            4fd30fc83117        5 weeks ago
↪      138MB
hello-world                   latest              f2a91732366c        8 weeks ago
↪      1.85kB
docker4w/nsenter-dockerd      latest              cae870735e91        2 months ago
↪      187kB
dockersamples/static-site     latest              f589ccde7957        22 months ago
↪      191MB
```

5.1.2.4 docker run –name static-site -e AUTHOR=”patrick.vergain” -d -P dockersamples/static-site

```
C:\Tmp>docker run --name static-site -e AUTHOR="patrick.vergain" -d -P
↪dockersamples/static-site
```

```
554e21d4b723a49e4b2019497d4411d955de2175e8b216a126d3a0c214ca9458
```

In the above command:

- -d will create a container with the process detached from our terminal
- -P will publish all the exposed container ports to random ports on the Docker host
- -e is how you pass environment variables to the container
- –name allows you to specify a container name
- AUTHOR is the environment variable name and Your Name is the value that you can pass

5.1.2.5 docker port static-site

```
docker port static-site
```

```
443/tcp -> 0.0.0.0:32768
80/tcp -> 0.0.0.0:32769
```

If you are running Docker for Mac, Docker for Windows, or Docker on Linux, you can open http://localhost:{[}YOUR_PORT_FOR 80/tcp]. For our example this is http://localhost:32769/

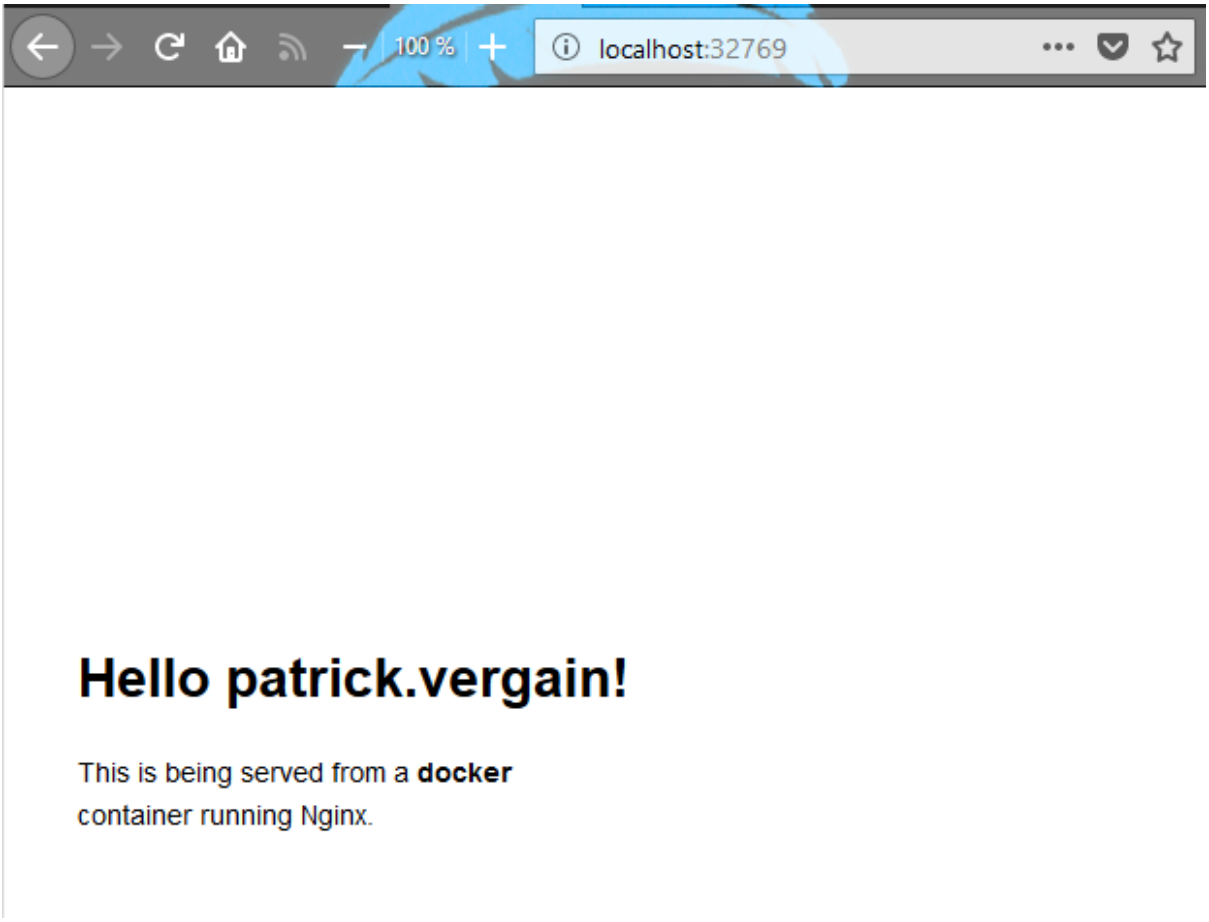
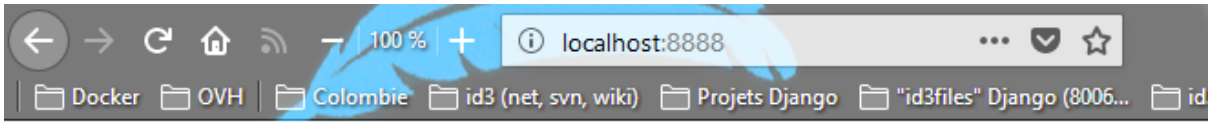


Fig. 1: http://localhost:32769/

5.1.2.6 docker run –name static-site-2 -e AUTHOR=”patrick.vergain” -d -p 8888:80 dockersamples/static-site

```
C:\Tmp>docker run --name static-site-2 -e AUTHOR="patrick.vergain" -d -p 8888:80␣
↪dockersamples/static-site
```

```
839649f1be575ec442f9fe94d6957b0f218b63af3dfaa8df989f413e86896d16
```



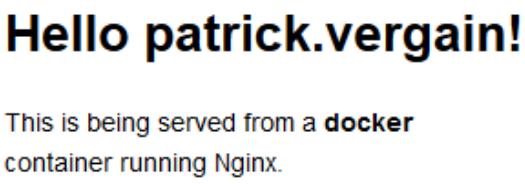


Fig. 2: http://localhost:8888/

To deploy this on a real server you would just need to install Docker, and run the above docker command(as in this case you can see the AUTHOR is Docker which we passed as an environment variable).

Now that you've seen how to run a webserver inside a Docker container, **how do you create your own Docker image ?**

This is the question we'll explore in the next section.

But first, let's stop and remove the containers since you won't be using them anymore.

5.1.2.7 docker stop static-site

```
docker stop static-site
```

```
static-site
```

5.1.2.8 docker rm static-site

```
docker rm static-site
```

```
static-site
```

5.1.2.9 Let's use a shortcut to remove the second site: docker rm -f static-site-2

```
docker rm -f static-site-2
```

```
static-site-2
```

5.1.2.10 Docker Images

See also:

- http://linuxfr.org/news/sortie-d-ubuntu-16-04-lts-xenial-xerus

In this section, let's dive deeper into what Docker images are.

You will build your own image, use that image to run an application locally, and finally, push some of your own images to Docker Cloud.

Docker images are the basis of containers. In the previous example, you pulled the dockersamples/static-site image from the registry and asked the Docker client to run a container based on that image.

To see the list of images that are available locally on your system, run the docker images command.

```
C:\Tmp>docker images
```

```
REPOSITORY                  TAG                 IMAGE ID            CREATED
↪      SIZE
ubuntu                      trusty              02a63d8b2bfa        18 hours ago
↪      222MB
id3pvergain/get-started     part2               ed5b70620e49        32 hours ago
↪      148MB
friendlyhello               latest              ed5b70620e49        32 hours ago
↪      148MB
alpine                      latest              3fd9065eaf02        6 days ago
↪      4.15MB
wordpress                   latest              28084cde273b        7 days ago
↪      408MB
centos                      latest              ff426288ea90        7 days ago
↪      207MB
nginx                       latest              3f8a4339aadd        2 weeks ago
↪      108MB
ubuntu                      latest              00fd29ccc6f1        4 weeks ago
↪      111MB
python                      2.7-slim            4fd30fc83117        5 weeks ago
↪      138MB
hello-world                 latest              f2a91732366c        8 weeks ago
↪      1.85kB
docker4w/nsenter-dockerd    latest              cae870735e91        2 months ago
↪      187kB
dockersamples/static-site   latest              f589ccde7957        22 months ago
↪      191MB
```

Above is a list of images that I've pulled from the registry and those I've created myself (we'll shortly see how). You will have a different list of images on your machine. The TAG refers to a particular snapshot of the image and the ID is the corresponding unique identifier for that image.

For simplicity, you can think of an image akin to a git repository - images can be committed with changes and have multiple versions. When you do not provide a specific version number, the client defaults to latest.

For example you could pull a specific version of ubuntu image as follows:

5.1.2.11 docker pull ubuntu:16.04

```
docker pull ubuntu:16.04
```

```
16.04: Pulling from library/ubuntu
8f7c85c2269a: Pull complete
9e72e494a6dd: Pull complete
3009ec50c887: Pull complete
9d5ffccbec91: Pull complete
e872a2642ce1: Pull complete
Digest: sha256:d3fdf5b1f8e8a155c17d5786280af1f5a04c10e95145a515279cf17abdf0191f
Status: Downloaded newer image for ubuntu:16.04
```

If you do not specify the version number of the image then, as mentioned, the Docker client will default to a version named latest.

So for example, the docker pull command given below will pull an image named ubuntu:latest:

```
docker pull ubuntu
```

To get a new Docker image you can either get it from a registry (such as the Docker Store) or create your own. There are hundreds of thousands of images available on Docker Store. You can also search for images directly from the command line using docker search.

An important distinction with regard to images is between base images and child images.

- Base images are images that have no parent images, usually images with an OS like ubuntu, alpine or debian.
- Child images are images that build on base images and add additional functionality.

Another key concept is the idea of official images and user images. (Both of which can be base images or child images.)

Official images are Docker sanctioned images. Docker, Inc. sponsors a dedicated team that is responsible for reviewing and publishing all Official Repositories content. This team works in collaboration with upstream software maintainers, security experts, and the broader Docker community.

These are not prefixed by an organization or user name. In the list of images above, the python, node, alpine and nginx images are official (base) images. To find out more about them, check out the Official Images Documentation.

User images are images created and shared by users like you. They build on base images and add additional functionality. Typically these are formatted as user/image-name. The user value in the image name is your Docker Store user or organization name.

5.1.2.12 Create your first image

Note: The code for this section is in this repository in the flask-app directory.

Now that you have a better understanding of images, it's time to create your own. Our goal here is to create an image that sandboxes a small Flask application.

The goal of this exercise is to create a Docker image which will run a Flask app.

We'll do this by first pulling together the components for a random cat picture generator built with Python Flask, then dockerizing it by writing a Dockerfile.

Finally, we'll build the image, and then run it.

- Create a Python Flask app that displays random cat pix
- Write a Dockerfile

- Build the image
- Run your image
- Dockerfile commands summary

5.1.2.13 Create a Python Flask app that displays random cat pix

For the purposes of this workshop, we've created a fun little Python Flask app that displays a random cat .gif every time it is loaded because, you know, who doesn't like cats ?

Start by creating a directory called flask-app where we'll create the following files:

- app.py
- requirements.txt
- templates/index.html
- Dockerfile

Make sure to cd flask-app before you start creating the files, because you don't want to start adding a whole bunch of other random files to your image.

5.1.2.13.1 app.py

Create the app.py with the following content.

```
"""app.py



"""

from flask import Flask, render_template
import random

app = Flask(__name__)

# list of cat images
images = [
    "http://ak-hdl.buzzfed.com/static/2013-10/enhanced/webdr05/15/9/anigif_
↪enhanced-buzz-26388-1381844103-11.gif",
    "http://ak-hdl.buzzfed.com/static/2013-10/enhanced/webdr01/15/9/anigif_
↪enhanced-buzz-31540-1381844535-8.gif",
    "http://ak-hdl.buzzfed.com/static/2013-10/enhanced/webdr05/15/9/anigif_
↪enhanced-buzz-26390-1381844163-18.gif",
    "http://ak-hdl.buzzfed.com/static/2013-10/enhanced/webdr06/15/10/anigif_
↪enhanced-buzz-1376-1381846217-0.gif",
    "http://ak-hdl.buzzfed.com/static/2013-10/enhanced/webdr03/15/9/anigif_
↪enhanced-buzz-3391-1381844336-26.gif",
    "http://ak-hdl.buzzfed.com/static/2013-10/enhanced/webdr06/15/10/anigif_
↪enhanced-buzz-29111-1381845968-0.gif",
    "http://ak-hdl.buzzfed.com/static/2013-10/enhanced/webdr03/15/9/anigif_
↪enhanced-buzz-3409-1381844582-13.gif",
    "http://ak-hdl.buzzfed.com/static/2013-10/enhanced/webdr02/15/9/anigif_
↪enhanced-buzz-19667-1381844937-10.gif",
    "http://ak-hdl.buzzfed.com/static/2013-10/enhanced/webdr05/15/9/anigif_
↪enhanced-buzz-26358-1381845043-13.gif",
    "http://ak-hdl.buzzfed.com/static/2013-10/enhanced/webdr06/15/9/anigif_
↪enhanced-buzz-18774-1381844645-6.gif",
    "http://ak-hdl.buzzfed.com/static/2013-10/enhanced/webdr06/15/9/anigif_
↪enhanced-buzz-25158-1381844793-0.gif",
```

(continues on next page)

(continued from previous page)

```
    "http://ak-hdl.buzzfed.com/static/2013-10/enhanced/webdr03/15/10/anigif_enhanced-buzz-11980-1381846269-1.gif"
]

@app.route('/')
def index():
    url = random.choice(images)
    return render_template('index.html', url=url)

if __name__ == "__main__":
    app.run(host="0.0.0.0")
```

5.1.2.13.2 requirements.txt

In order to install the Python modules required for our app, we need to create a file called requirements.txt and add the following line to that file

```
Flask==0.10.1
```

5.1.2.13.3 templates/index.html

Create a directory called templates and create an index.html file in that directory with the following content in it.

```
<html>
  <head>
    <style type="text/css">
      body {
        background: black;
        color: white;
      }
      div.container {
        max-width: 500px;
        margin: 100px auto;
        border: 20px solid white;
        padding: 10px;
        text-align: center;
      }
      h4 {
        text-transform: uppercase;
      }
    </style>
  </head>
  <body>
    <div class="container">
      <h4>Cat Gif of the day</h4>
      <img src="{{url}}" />
      <p><small>Courtesy: <a href="http://www.buzzfeed.com/copyranter/the-best-cat-gif-post-in-the-history-of-cat-gifs">Buzzfeed</a></small></p>
    </div>
  </body>
</html>
```

5.1.2.13.4 Write a Dockerfile

See also:

- https://docs.docker.com/engine/reference/builder/

We want to create a Docker image with this web app. As mentioned above, all user images are based on a base image. Since our application is written in Python, we will build our own Python image based on *Alpine*. We'll do that using a Dockerfile.

A Dockerfile is a text file that contains a list of commands that the Docker daemon calls while creating an image. The Dockerfile contains all the information that Docker needs to know to run the app, a base Docker image to run from, location of your project code, any dependencies it has, and what commands to run at start-up. It is a simple way to automate the image creation process. The best part is that the commands you write in a Dockerfile are almost identical to their equivalent Linux commands. This means you don't really have to learn new syntax to create your own Dockerfiles.

5.1.2.13.4.1 FROM alpine:3.5

We'll start by specifying our base image, using the FROM keyword:

```
FROM alpine:3.5
```

5.1.2.13.4.2 RUN apk add –update py2-pip

The next step usually is to write the commands of copying the files and installing the dependencies. But first we will install the Python pip package to the alpine linux distribution. This will not just install the pip package but any other dependencies too, which includes the python interpreter. Add the following RUN command next:

```
RUN apk add --update py2-pip
```

Let's add the files that make up the Flask Application.

5.1.2.13.4.3 COPY requirements.txt /usr/src/app/

Install all Python requirements for our app to run. This will be accomplished by adding the lines:

```
COPY requirements.txt /usr/src/app/
RUN pip install --no-cache-dir -r /usr/src/app/requirements.txt
```

5.1.2.13.4.4 COPY app.py /usr/src/app/

Copy the files you have created earlier into our image by using COPY command.

```
COPY app.py /usr/src/app/
COPY templates/index.html /usr/src/app/templates/
```

5.1.2.13.4.5 EXPOSE 5000

Specify the port number which needs to be exposed. Since our flask app is running on 5000 that's what we'll expose.

```
EXPOSE 5000
```

5.1.2.13.4.6 CMD ["python", "/usr/src/app/app.py"]

The last step is the command for running the application which is simply python ./app.py. Use the CMD command to do that:

```
CMD ["python", "/usr/src/app/app.py"]
```

The primary purpose of CMD is to tell the container which command it should run by default when it is started.

Verify your Dockerfile.

Our Dockerfile is now ready. This is how it looks:

```
# our base image
FROM alpine:3.5

# Install python and pip
RUN apk add --update py2-pip

# install Python modules needed by the Python app
COPY requirements.txt /usr/src/app/
RUN pip install --no-cache-dir -r /usr/src/app/requirements.txt

# copy files required for the app to run
COPY app.py /usr/src/app/
COPY templates/index.html /usr/src/app/templates/

# tell the port number the container should expose
EXPOSE 5000

# run the application
CMD ["python", "/usr/src/app/app.py"]
```

5.1.2.13.5 Build the image (docker build -t id3pvergain/myfirstapp)

Now that you have your Dockerfile, you can build your image.

The docker build command does the heavy-lifting of creating a docker image from a Dockerfile.

When you run the docker build command given below, make sure to replace <YOUR_USERNAME> with your username.

This username should be the same one you created when registering on Docker Cloud. If you haven't done that yet, please go ahead and create an account.

The docker build command is quite simple - it takes an optional tag name with the -t flag, and the location of the directory containing the Dockerfile - the . indicates the current directory:

```
docker build -t id3pvergain/myfirstapp .
```

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\samples\labs\webapps\app_flask>docker build -t id3pvergain/myfirstapp .
```

```
Sending build context to Docker daemon   7.68kB
Step 1/8 : FROM alpine:3.5
3.5: Pulling from library/alpine
550fe1bea624: Pull complete
Digest: sha256:9148d069e50eee519ec45e5683e56a1c217b61a52ed90eb77bdce674cc212f1e
Status: Downloaded newer image for alpine:3.5
 ---> 6c6084ed97e5
Step 2/8 : RUN apk add --update py2-pip
 ---> Running in 1fe5bd53d58d
fetch http://dl-cdn.alpinelinux.org/alpine/v3.5/main/x86_64/APKINDEX.tar.gz
fetch http://dl-cdn.alpinelinux.org/alpine/v3.5/community/x86_64/APKINDEX.tar.gz
(1/12) Installing libbz2 (1.0.6-r5)
(2/12) Installing expat (2.2.0-r1)
```

(continues on next page)

(continued from previous page)

```
(3/12) Installing libffi (3.2.1-r2)
(4/12) Installing gdbm (1.12-r0)
(5/12) Installing ncurses-terminfo-base (6.0_p20170701-r0)
(6/12) Installing ncurses-terminfo (6.0_p20170701-r0)
(7/12) Installing ncurses-libs (6.0_p20170701-r0)
(8/12) Installing readline (6.3.008-r4)
(9/12) Installing sqlite-libs (3.15.2-r1)
(10/12) Installing python2 (2.7.13-r0)
(11/12) Installing py-setuptools (29.0.1-r0)
(12/12) Installing py2-pip (9.0.0-r1)
Executing busybox-1.25.1-r1.trigger
OK: 61 MiB in 23 packages
Removing intermediate container 1fe5bd53d58d
 ---> 23504d4e2c59
Step 3/8 : COPY requirements.txt /usr/src/app/
 ---> 1be30128b66f
Step 4/8 : RUN pip install --no-cache-dir -r /usr/src/app/requirements.txt
 ---> Running in a5f6ada2483d
Collecting Flask==0.10.1 (from -r /usr/src/app/requirements.txt (line 1))
  Downloading Flask-0.10.1.tar.gz (544kB)
Collecting Werkzeug>=0.7 (from Flask==0.10.1->-r /usr/src/app/requirements.txt␣
↪(line 1))
  Downloading Werkzeug-0.14.1-py2.py3-none-any.whl (322kB)
Collecting Jinja2>=2.4 (from Flask==0.10.1->-r /usr/src/app/requirements.txt (line␣
↪1))
  Downloading Jinja2-2.10-py2.py3-none-any.whl (126kB)
Collecting itsdangerous>=0.21 (from Flask==0.10.1->-r /usr/src/app/requirements.
↪txt (line 1))
  Downloading itsdangerous-0.24.tar.gz (46kB)
Collecting MarkupSafe>=0.23 (from Jinja2>=2.4->Flask==0.10.1->-r /usr/src/app/
↪requirements.txt (line 1))
  Downloading MarkupSafe-1.0.tar.gz
Installing collected packages: Werkzeug, MarkupSafe, Jinja2, itsdangerous, Flask
  Running setup.py install for MarkupSafe: started
       Running setup.py install for MarkupSafe: finished with status 'done'
  Running setup.py install for itsdangerous: started
       Running setup.py install for itsdangerous: finished with status 'done'
  Running setup.py install for Flask: started
       Running setup.py install for Flask: finished with status 'done'
Successfully installed Flask-0.10.1 Jinja2-2.10 MarkupSafe-1.0 Werkzeug-0.14.1␣
↪itsdangerous-0.24
You are using pip version 9.0.0, however version 9.0.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
Removing intermediate container a5f6ada2483d
 ---> 68467d64c546
Step 5/8 : COPY app.py /usr/src/app/
 ---> 62a6a857c6cd
Step 6/8 : COPY templates/index.html /usr/src/app/templates/
 ---> 639c61ea4a4b
Step 7/8 : EXPOSE 5000
 ---> Running in c15c0178577c
Removing intermediate container c15c0178577c
 ---> f6d0fdcd6c29
Step 8/8 : CMD ["python", "/usr/src/app/app.py"]
 ---> Running in 222f91658593
Removing intermediate container 222f91658593
 ---> 0ce3c7641c9a
Successfully built 0ce3c7641c9a
Successfully tagged id3pvergain/myfirstapp: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.
```

(continues on next page)

(continued from previous page)

If you don't have the alpine:3.5 image, the client will first pull the image and then create your image. Therefore, your output on running the command will look different from mine. If everything went well, your image should be ready!

5.1.2.13.6 docker images

Run docker images and see if your image (<YOUR_USERNAME>/myfirstapp) shows.

Y:projects_id3P5N001XLOGCA135_tutorial_dockertutorial_dockersampleslabswebappsapp_flask>docker images

```
REPOSITORY                  TAG                 IMAGE ID            CREATED
↪      SIZE
id3pvergain/myfirstapp      latest              0ce3c7641c9a        2 minutes ago
↪      56.4MB
ubuntu                      16.04               2a4cca5ac898        38 hours ago
↪      111MB
ubuntu                      trusty              02a63d8b2bfa        38 hours ago
↪      222MB
friendlyhello               latest              ed5b70620e49        2 days ago
↪      148MB
id3pvergain/get-started     part2               ed5b70620e49        2 days ago
↪      148MB
alpine                      3.5                 6c6084ed97e5        7 days ago
↪      3.99MB
alpine                      latest              3fd9065eaf02        7 days ago
↪      4.15MB
wordpress                   latest              28084cde273b        8 days ago
↪      408MB
centos                      latest              ff426288ea90        8 days ago
↪      207MB
nginx                       latest              3f8a4339aadd        3 weeks ago
↪      108MB
ubuntu                      latest              00fd29ccc6f1        4 weeks ago
↪      111MB
python                      2.7-slim            4fd30fc83117        5 weeks ago
↪      138MB
hello-world                 latest              f2a91732366c        8 weeks ago
↪      1.85kB
docker4w/nsenter-dockerd    latest              cae870735e91        2 months ago
↪      187kB
dockersamples/static-site   latest              f589ccde7957        22 months ago
↪      191MB
```

5.1.2.13.7 Run your image (docker run -p 8888:5000 –name myfirstapp id3pvergain/myfirstapp)

The next step in this section is to run the image and see if it actually works.

```
   docker run -p 8888:5000 --name myfirstapp id3pvergain/myfirstapp


* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```

Head over to http://localhost:8888 and your app should be live.

Note: If you are using Docker Machine, you may need to open up another terminal and determine the container

ip address using docker-machine ip default.

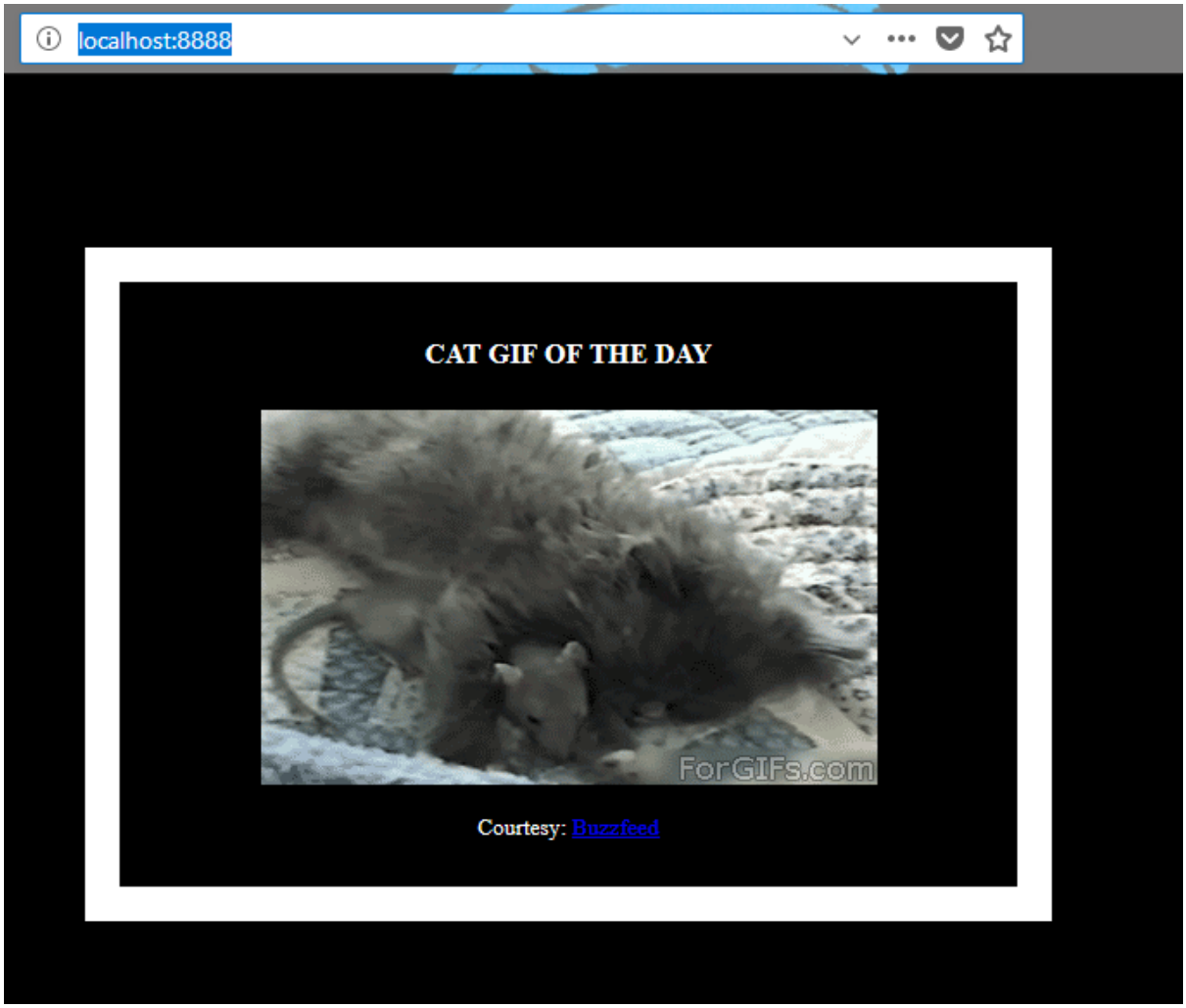


Fig. 3: http://localhost:8888/

Hit the Refresh button in the web browser to see a few more cat images.

5.1.2.13.8 Push your image (docker push id3pvergain/myfirstapp)

Now that you've created and tested your image, you can push it to Docker Cloud.

First you have to login to your Docker Cloud account, to do that:

5.1.2.13.8.1 docker login

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\samples\labs\webapps\app_flask>docker login
```

```
Login with your Docker ID to push and pull images from Docker Hub. If you don't␣
↪have a Docker ID, head over to https://hub.docker.com to create one.
Username (id3pvergain):
Password:
Login Succeeded
```

5.1.2.13.8.2 docker push id3pvergain/myfirstapp

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\samples\labs\webapps\app_flask>docker push id3pvergain/myfirstapp
```

```
The push refers to repository [docker.io/id3pvergain/myfirstapp]
b7591dd05809: Pushed
cd36128c70d4: Pushed
cea459424f6e: Pushed
6ac80674ef6a: Pushed
de7b45529bcb: Pushed
d39d92664027: Mounted from library/alpine
latest: digest:␣
↪sha256:8f945ed63e2dc3ef3fa178fe4dded5a68eae07c5c9e854ec278c7cfa2c6bc6bb size:␣
↪1572
```

5.1.2.13.9 docker rm -f myfirstapp

Now that you are done with this container, stop and remove it since you won't be using it again.

Open another terminal window and execute the following commands:

```
docker stop myfirstapp
docker rm myfirstapp
```

or:

```
docker rm -f myfirstapp
```

```
myfirstapp
```

5.1.2.13.10 docker ps

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\samples\labs\webapps\app_flask>docker ps
```

```
CONTAINER ID        IMAGE               COMMAND             CREATED             ␣
↪STATUS              PORTS               NAMES
```

5.1.2.14 Dockerfile commands summary

Here's a quick summary of the few basic commands we used in our Dockerfile.

5.1.2.14.1 FROM

FROM starts the Dockerfile. It is a requirement that the Dockerfile must start with the FROM command. Images are created in layers, which means you can use another image as the base image for your own. The FROM command defines your base layer. As arguments, it takes the name of the image. Optionally, you can add the Docker Cloud username of the maintainer and image version, in the format username/imagename:version.

5.1.2.14.2 RUN

RUN is used to build up the Image you're creating. For each RUN command, Docker will run the command then create a new layer of the image. This way you can roll back your image to previous states easily. The syntax for a RUN instruction is to place the full text of the shell command after the RUN (e.g., RUN mkdir /user/local/foo). This will automatically run in a /bin/sh shell. You can define a different shell like this: RUN /bin/bash -c 'mkdir /user/local/foo'

5.1.2.14.3 COPY

COPY copies local files into the container.

5.1.2.14.4 CMD

CMD defines the commands that will run on the Image at start-up.

Unlike a RUN, *this does not create a new layer for the Image*, but simply runs the command.

There can only be one CMD per a Dockerfile/Image.

If you need to run multiple commands, the best way to do that is to have the CMD run a script. CMD requires that you tell it where to run the command, unlike RUN.

So example CMD commands would be:

CMD ["python", "./app.py"]

CMD ["/bin/bash", "echo", "Hello World"]

5.1.2.14.5 EXPOSE

EXPOSE creates a hint for users of an image which ports provide services. It is included in the information which can be retrieved via docker inspect <container-id>.

Note: The EXPOSE command does not actually make any ports accessible to the host! Instead, this requires publishing ports by means of the -p flag when using $ docker run.

5.1.2.14.6 PUSH

PUSH pushes your image to Docker Cloud, or alternately to a private registry

Note: If you want to learn more about Dockerfiles, check out Best practices for writing Dockerfiles.

5.1.2.15 Next Steps : Deploying an app to a Swarm

See also:

- *3.0) Deploying an app to a Swarm*

For the next step in the tutorial head over to *3.0 Deploying an app to a Swarm*

5.1.3 3.0) Deploying an app to a Swarm

See also:

- https://github.com/docker/labs/blob/master/beginner/chapters/votingapp.md
- https://github.com/dockersamples/example-voting-app
- *2) Webapps with Docker (Python + Flask)*

Contents

5.1.3.1 Introduction

This portion of the tutorial will guide you through the creation and customization of a voting app. It's important that you follow the steps in order, and make sure to customize the portions that are customizable.

Warning: To complete this section, you will need to have Docker installed on your machine as mentioned in the Setup section. You'll also need to have git installed. There are many options for installing it. For instance, you can get it from GitHub.

5.1.3.2 Voting app

For this application we will use the Docker Example Voting App.

This app consists of five components:

- Python webapp which lets you vote between two options
- Redis queue which collects new votes
- DotNET worker which consumes votes and stores them in
- Postgres database backed by a Docker volume
- Node.js webapp which shows the results of the voting in real time

Clone the repository onto your machine and cd into the directory:

```
git clone https://github.com/docker/example-voting-app.git
```

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\samples\labs\votingapp>git clone https://github.com/docker/example-voting-
↪app.git
```

```
Cloning into 'example-voting-app'...
remote: Counting objects: 463, done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 463 (delta 4), reused 12 (delta 4), pack-reused 447
Receiving objects: 100% (463/463), 226.49 KiB | 318.00 KiB/s, done.
Resolving deltas: 100% (167/167), done.
```

```
cd example-voting-app
```

5.1.3.3 Deploying the app

See also:

- https://docs.docker.com/engine/swarm/

For this first stage, we will use existing images that are in Docker Store.

This app relies on Docker Swarm mode[13]. Swarm mode is the cluster management and orchestration features embedded in the Docker engine. You can easily deploy to a swarm using a file that declares your desired state for the app.

Swarm allows you to run your containers on more than one machine.

In this tutorial, *you can run on just one machine*, or you can use something like Docker for AWS or Docker for Azure to quickly create a multiple node machine. Alternately, you can use Docker Machine to create a number of local nodes on your development machine. See the Swarm Mode lab for more information.

5.1.3.3.1 docker swarm init

First, create a Swarm.

docker swarm init

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\samples\labs\votingapp\example-voting-app>docker swarm init
```

[13] https://docs.docker.com/engine/swarm/

```
Swarm initialized: current node (pfx5nyrmtv0m5twcz4dv4oypg) is now a manager.

To add a worker to this swarm, run the following command:

        docker swarm join --token SWMTKN-1-
↪1a5pls76a0tyfn9tybruku4naqaa1vldvw0iy76hw9t6uw931w-098lzv69ozqce3v6eiptieeta 192.
↪168.65.3:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow␣
↪the instructions.
```

Next, you will need a Docker Compose file. You don't need Docker Compose installed, though if you are using Docker for Mac or Docker for Windows you have it installed. However, docker stack deploy accepts a file in the Docker Compose format. The file you need is in Docker Example Voting App at the root level. It's called docker-stack.yml.

5.1.3.3.2 Docker compose file : docker-stack.yml

See also:

- https://github.com/dockersamples/example-voting-app/blob/master/docker-stack.yml

```
version: "3"
services:

  redis:
        image: redis:alpine
        ports:
          - "6379"
        networks:
          - frontend
        deploy:
          replicas: 1
          update_config:
                parallelism: 2
                delay: 10s
          restart_policy:
                condition: on-failure
  db:
        image: postgres:9.4
        volumes:
          - db-data:/var/lib/postgresql/data
        networks:
          - backend
        deploy:
          placement:
                constraints: [node.role == manager]
  vote:
        image: dockersamples/examplevotingapp_vote:before
        ports:
          - 5000:80
        networks:
          - frontend
        depends_on:
          - redis
        deploy:
          replicas: 2
          update_config:
                parallelism: 2
          restart_policy:
                condition: on-failure
```

(continues on next page)

(continued from previous page)

```
  result:
        image: dockersamples/examplevotingapp_result:before
        ports:
          - 5001:80
        networks:
          - backend
        depends_on:
          - db
        deploy:
          replicas: 1
          update_config:
                parallelism: 2
                delay: 10s
          restart_policy:
                condition: on-failure

  worker:
        image: dockersamples/examplevotingapp_worker
        networks:
          - frontend
          - backend
        deploy:
          mode: replicated
          replicas: 1
          labels: [APP=VOTING]
          restart_policy:
                condition: on-failure
                delay: 10s
                max_attempts: 3
                window: 120s
          placement:
                constraints: [node.role == manager]

  visualizer:
        image: dockersamples/visualizer:stable
        ports:
          - "8080:8080"
        stop_grace_period: 1m30s
        volumes:
          - "/var/run/docker.sock:/var/run/docker.sock"
        deploy:
          placement:
                constraints: [node.role == manager]

networks:
  frontend:
  backend:

volumes:
  db-data:
```

5.1.3.3.3 docker stack deploy –compose-file docker-stack.yml vote

First deploy it, and then we will look more deeply into the details:

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\samples\labs\votingapp\example-voting-app>docker stack deploy --compose-
↪file docker-stack.yml vote
```

```
Creating network vote_backend
Creating network vote_default
Creating network vote_frontend
Creating service vote_visualizer
Creating service vote_redis
Creating service vote_db
Creating service vote_vote
Creating service vote_result
Creating service vote_worker
```

5.1.3.3.4 docker stack services vote

to verify your stack has deployed, use *docker stack services vote*

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\samples\labs\votingapp\example-voting-app>docker stack services vote
```

```
ID                  NAME                MODE                REPLICAS
↪IMAGE                                          PORTS
d7ovptjpvv3y        vote_vote           replicated          0/2
↪dockersamples/examplevotingapp_vote:before     *:5000->80/tcp
lve7cp7gxvwg        vote_result         replicated          0/1
↪dockersamples/examplevotingapp_result:before   *:5001->80/tcp
r2mhivfbyaun        vote_redis          replicated          1/1
↪redis:alpine                                   *:30000->6379/tcp
szzocr20dyfc        vote_visualizer     replicated          1/1
↪dockersamples/visualizer:stable                *:8080->8080/tcp
vgv0iucy6fx9        vote_db             replicated          0/1
↪postgres:9.4
vlieeu7ru24a        vote_worker         replicated          0/1
↪dockersamples/examplevotingapp_worker:latest
```

5.1.3.3.5 Analyse du fichier Docker compose file : docker-stack.yml

See also:

- https://github.com/docker/labs/tree/master/networking

If you take a look at docker-stack.yml, you will see that the file defines

- vote container based on a Python image
- result container based on a Node.js image
- redis container based on a redis image, to temporarily store the data.
- DotNET based worker app based on a .NET image
- Postgres container based on a postgres image

The Compose file also defines two networks, front-tier and back-tier.

Each container is placed on one or two networks.

Once on those networks, they can access other services on that network in code just by using the name of the service.

Services can be on any number of networks.

Services are isolated on their network.

Services are only able to discover each other by name if they are on the same network.

To learn more about networking check out the Networking Lab[14].

5.1.3.3.5.1 compose-file: "3"

See also:

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

Take a look at the file again. You'll see it starts with:

```
version: "3"
```

It's important that you use **version 3** of compose files, as docker stack deploy won't support use of earlier versions.

5.1.3.3.5.2 compose-file: *services*

You will see there's also a **services** key, under which there is a separate key for each of the services. Such as:

```
vote:
        image: dockersamples/examplevotingapp_vote:before
        ports:
          - 5000:80
        networks:
          - frontend
        depends_on:
          - redis
        deploy:
          replicas: 2
          update_config:
                parallelism: 2
          restart_policy:
                condition: on-failure
```

5.1.3.3.5.3 compose-file: *image*

See also:

https://docs.docker.com/compose/compose-file/#image

The image key there specifies which image you can use, in this case the image dockersamples/examplevotingapp_vote:before.

If you're familiar with Compose, you may know that there's a build key, which builds based on a Dockerfile.

However, docker stack deploy does not suppport build, so you need to use pre-built images.

5.1.3.3.5.4 compose-file: *ports and networks depends_on*

See also:

- https://docs.docker.com/compose/compose-file/#ports

Much like docker run you will see you can define ports and networks.

There's also a depends_on key which allows you to specify that a service is only deployed after another service, in this case vote only deploys after redis.

[14] https://github.com/docker/labs/tree/master/networking

5.1.3.3.5.5 compose-file: *deploy*

See also:

https://docs.docker.com/compose/compose-file/#deploy

The deploy key is new in version 3.

It allows you to specify various properties of the deployment to the Swarm.

In this case, you are specifying that you want two replicas, that is two containers are deployed on the Swarm. You can specify other properties, like when to restart, what healthcheck to use, placement constraints, resources.

5.1.3.3.5.6 Test run : http://localhost:5000/

Now that the app is running, you can go to http://localhost:5000 to see:

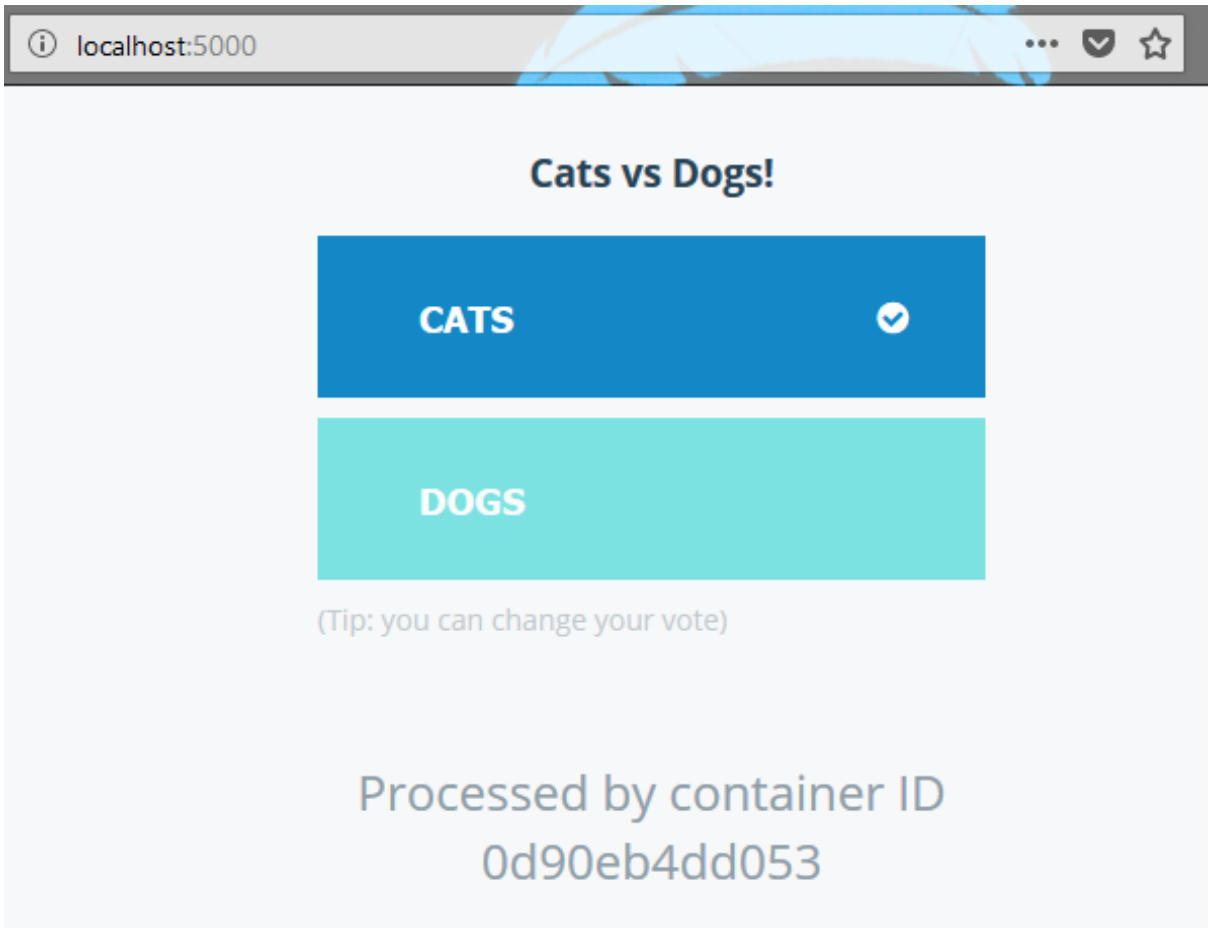


Fig. 4: http://localhost:5000/

Click on one to vote. You can check the results at http://localhost:5001

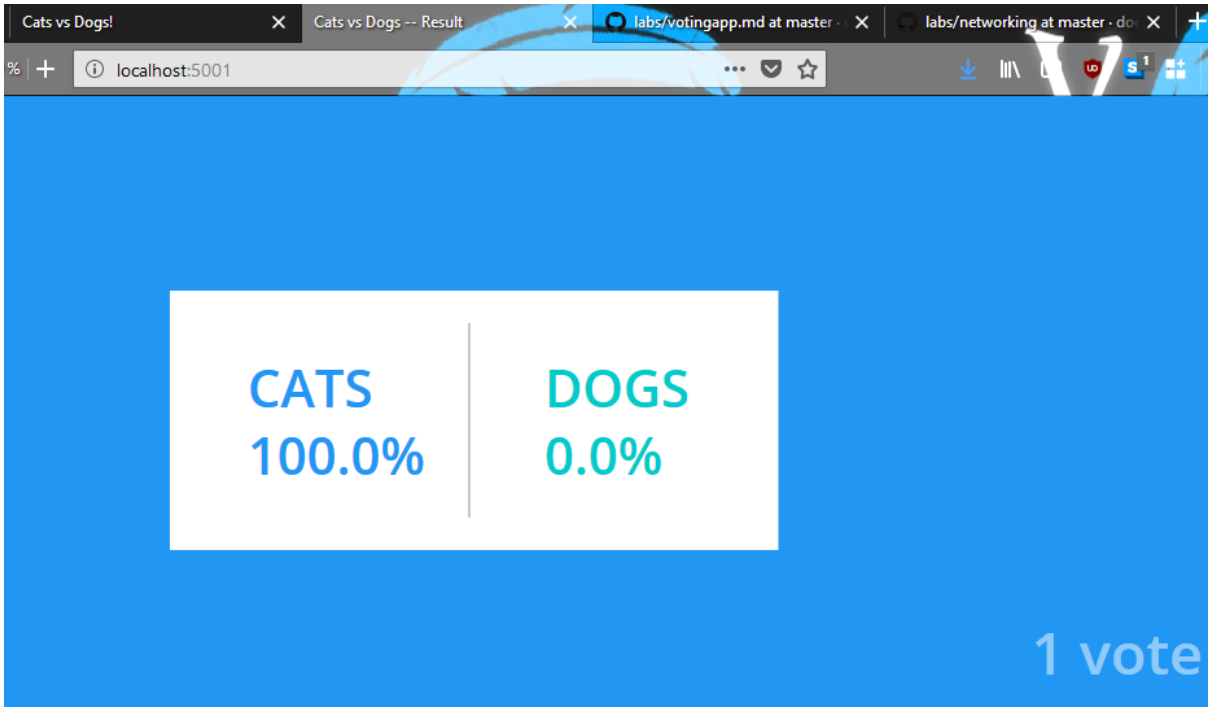
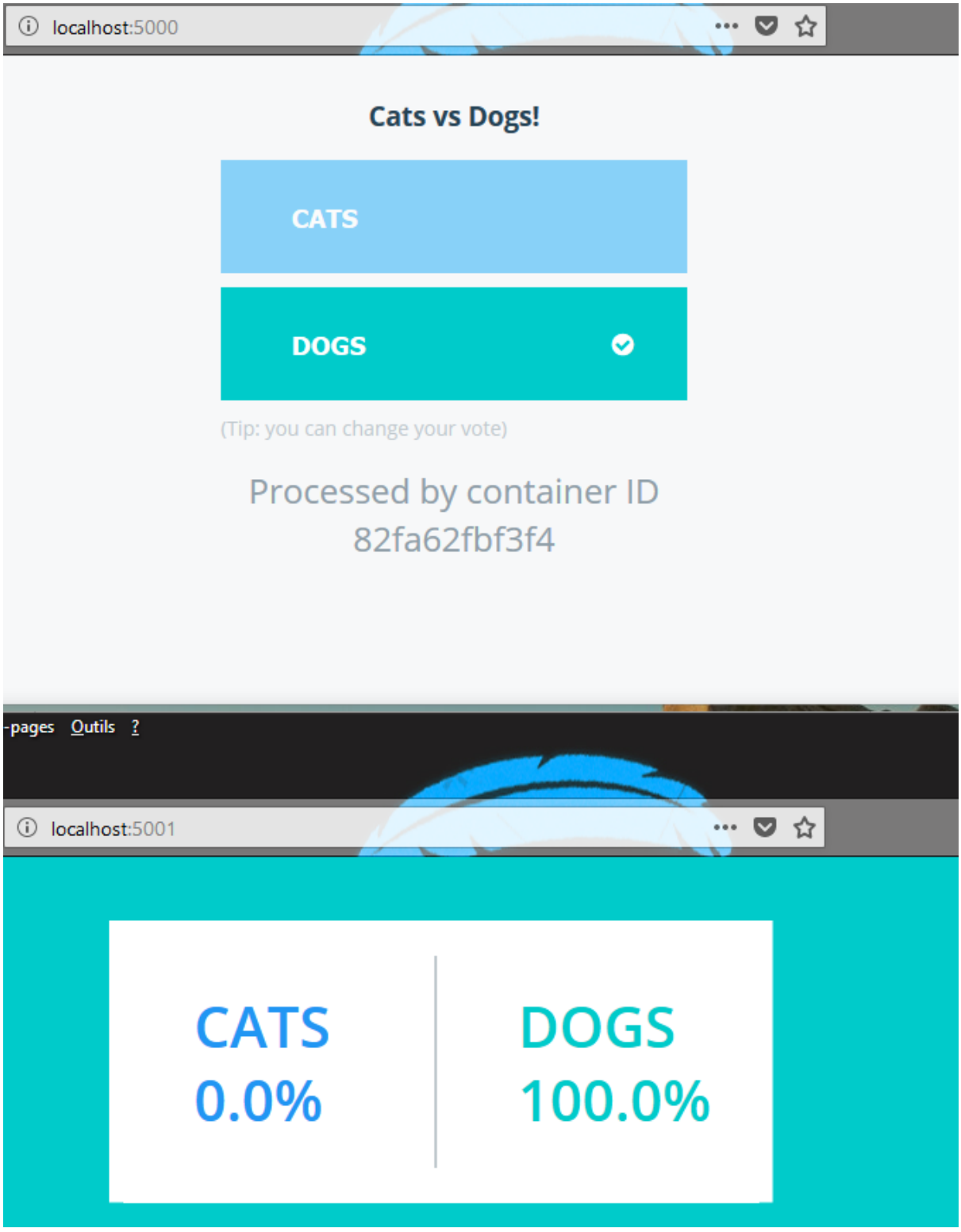


Fig. 5: http://localhost:5001/

Note: If you are running this tutorial in a cloud environment like AWS, Azure, Digital Ocean, or GCE you will not have direct access to localhost or 127.0.0.1 via a browser. A work around for this is to leverage ssh port forwarding.

Below is an example for Mac OS. Similarly this can be done for Windows and Putty users:

```
ssh -L 5000:localhost:5000 <ssh-user>@<CLOUD_INSTANCE_IP_ADDRESS>
```



5.1.3.4 Customize the app

In this step, you will customize the app and redeploy it.

We've supplied the same images but with the votes changed from Cats and Dogs to Java and .NET using the after tag.

5.1.3.4.1 Change the images used

Going back to docker-stack.yml, change the vote and result images to use the after tag, so they look like this:

```
vote:
  image: dockersamples/examplevotingapp_vote:after
  ports:
    - 5000:80
  networks:
    - frontend
  depends_on:
    - redis
  deploy:
    replicas: 2
    update_config:
      parallelism: 2
    restart_policy:
      condition: on-failure
result:
  image: dockersamples/examplevotingapp_result:after
  ports:
    - 5001:80
  networks:
    - backend
  depends_on:
    - db
  deploy:
    replicas: 2
    update_config:
      parallelism: 2
      delay: 10s
    restart_policy:
      condition: on-failure
```

5.1.3.4.2 Redeploy: docker stack deploy –compose-file docker-stack.yml vote

Redeployment is the same as deploying:

```
docker stack deploy --compose-file docker-stack.yml vote
```

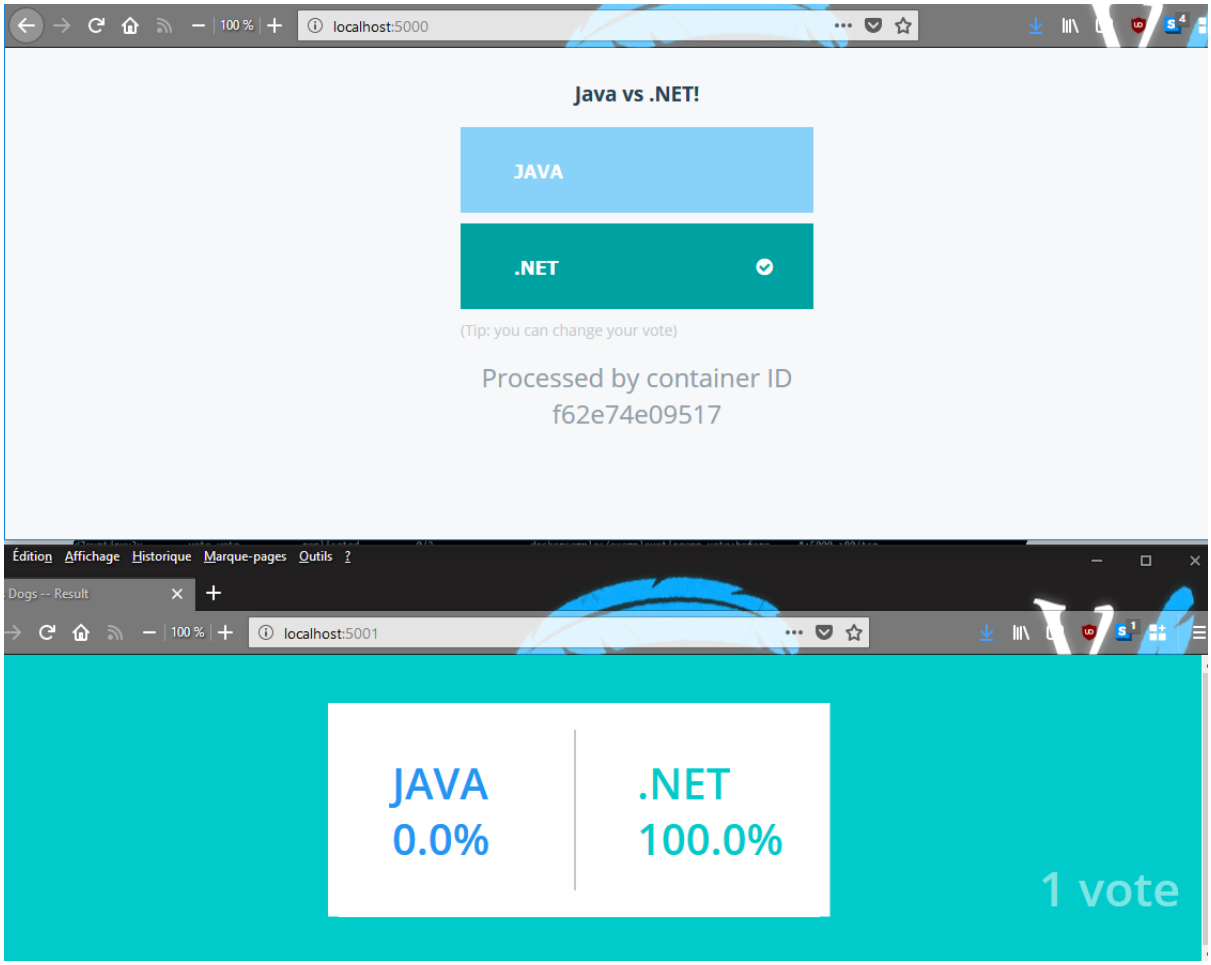
```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\samples\labs\votingapp\example-voting-app>docker stack deploy --compose-
↪file docker-stack.yml vote
```

```
Updating service vote_db (id: vgv0iucy6fx9ih4so6ufdzqh4)
Updating service vote_vote (id: d7ovptjpvv3ylpxb30hitxd1g)
Updating service vote_result (id: lve7cp7gxvwge1qhesjwuyon1)
Updating service vote_worker (id: vlieeu7ru24a8kc4vouwa0i5r)
Updating service vote_visualizer (id: szzocr20dyfc6ux0vdmamo5e1)
Updating service vote_redis (id: r2mhivfbyaunnd5szq5kh5fm7)
```

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_docker\samples\labs\votingapp\example-voting-app>docker swarm init
Swarm initialized: current node (pfx5nyrmtv0m5twcz4dv4oypg) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-1a5pls76a0tyfn9tybruku4naqaa1vldvw0iy76hw9t6uw931w-098lzv69ozqce3v6eiptieeta 192.168.65.3:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.


Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_docker\samples\labs\votingapp\example-voting-app>docker stack deploy --compose-file docker-stack.yml vote
Creating network vote_backend
Creating network vote_default
Creating network vote_frontend
Creating service vote_visualizer
Creating service vote_redis
Creating service vote_db
Creating service vote_vote
Creating service vote_result
Creating service vote_worker

Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_docker\samples\labs\votingapp\example-voting-app>docker stack services vote
ID                  NAME                MODE                REPLICAS            IMAGE                                          PORTS
d7ovptjpvv3y        vote_vote           replicated          0/2                 dockersamples/examplevotingapp_vote:before     *:5000->80/tcp
lve7cp7gxvwg        vote_result         replicated          0/1                 dockersamples/examplevotingapp_result:before   *:5001->80/tcp
r2mhivfbyaun        vote_redis          replicated          1/1                 redis:alpine                                   *:30000->6379/tcp
szzocr20dyfc        vote_visualizer     replicated          1/1                 dockersamples/visualizer:stable                *:8080->8080/tcp
vgv0iucy6fx9        vote_db             replicated          0/1                 postgres:9.4
vlieeu7ru24a        vote_worker         replicated          0/1                 dockersamples/examplevotingapp_worker:latest

Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_docker\samples\labs\votingapp\example-voting-app>docker stack deploy --compose-file docker-stack.yml vote
Updating service vote_db (id: vgv0iucy6fx9ih4so6ufdzqh4)
Updating service vote_vote (id: d7ovptjpvv3ylpxb30hitxd1g)
Updating service vote_result (id: lve7cp7gxvwge1qhesjwuyon1)
Updating service vote_worker (id: vlieeu7ru24a8kc4vouwa0i5r)
Updating service vote_visualizer (id: szzocr20dyfc6ux0vdmamo5e1)
Updating service vote_redis (id: r2mhivfbyaunnd5szq5kh5fm7)

Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_docker\samples\labs\votingapp\example-voting-app>
```

5.1.3.4.3 Another test run

Now take it for a spin again. Go to the URLs you used in section 3.1 and see the new votes.



5.1.3.4.4 Remove the stack

Remove the stack from the swarm:

```
docker stack rm vote
```

```
Removing service vote_db
Removing service vote_redis
Removing service vote_result
Removing service vote_visualizer
Removing service vote_vote
Removing service vote_worker
Removing network vote_frontend
Removing network vote_default
Removing network vote_backend
```

5.1.3.5 Next steps

See also:

- https://docs.docker.com/
- https://forums.docker.com/
- https://stackoverflow.com/tags/docker/

Now that you've built some images and pushed them to Docker Cloud, and learned the basics of Swarm mode, you can explore more of Docker by checking out the documentation[15].

And if you need any help, check out the Docker Forums[16] or StackOverflow[17].

5.2 Exemples sur Windows 10

See also:

- https://docs.microsoft.com/fr-fr/virtualization/windowscontainers/quick-start/quick-start-windows-10

[15] https://docs.docker.com/
[16] https://forums.docker.com/
[17] https://stackoverflow.com/tags/docker/

Chapter 6

Exemples Docker compose

See also:

- https://docs.docker.com/compose/samples-for-compose/

Contents

6.1 Concepts clés

Key concepts these samples cover

The samples should help you to:

- define services based on Docker images using Compose files docker-compose.yml and docker-stack.yml files
- understand the relationship between docker-compose.yml and Dockerfiles
- learn how to make calls to your application services from Compose files
- learn how to deploy applications and services to a swarm

6.2 Exemples

6.2.1 Quickstart: Compose and Django

See also:

- https://docs.docker.com/compose/django/
- https://docs.docker.com/compose/install/
- https://docs.docker.com/engine/tutorials/dockerimages/#building-an-image-from-a-dockerfile
- https://docs.docker.com/engine/reference/builder/
- https://store.docker.com/images/python

- https://docs.docker.com/compose/compose-file/
- https://docs.djangoproject.com/en/1.11/ref/settings/#allowed-hosts
- https://docs.docker.com/compose/reference/down/

Contents

6.2.1.1 Overview of Docker Compose

See also:

- https://github.com/docker/compose
- https://github.com/docker/docker.github.io/blob/master/compose/django.md
- https://docs.docker.com/compose/overview/
- https://docs.docker.com/compose/compose-file/
- https://github.com/docker/docker.github.io/blob/master/compose/overview.md#common-use-cases

Looking for Compose file reference? Find the latest version here[18].

Compose is a tool for defining and running multi-container Docker applications.

With Compose, you use a YAML file to configure your **application's services**.

[18] https://docs.docker.com/compose/compose-file/

Then, with a single command, you create and start all the services from your configuration. To learn more about all the features of Compose, see the list of features.

Compose works in all environments:

- production,
- staging,
- development,
- testing,
- as well as CI workflows.

You can learn more about each case in Common Use Cases.

Using Compose is basically a three-step process:

- Define your app's environment with a Dockerfile so it can be reproduced anywhere.
- Define the services that make up your app in docker-compose.yml so they can be run together in an isolated environment.
- Lastly, run **docker-compose up** and Compose will start and run your entire app.

6.2.1.2 Introduction

This quick-start guide demonstrates how to use Docker Compose to set up and run a simple Django/*PostgreSQL app*.

Before starting, you'll need to have Compose installed[19].

6.2.1.3 Define the project components

For this project, you need to create a Dockerfile, a Python dependencies file, and a docker-compose.yml file. (You can use either a .yml or .yaml extension for this file.)

6.2.1.3.1 mkdir django_app

Create an empty project directory.

You can name the directory something easy for you to remember. This directory is the context for your application image. The directory should only contain resources to build that image.

```
mkdir django_app
```

6.2.1.3.2 Create a Dockerfile

Create a new file called Dockerfile in your project directory.

```
FROM python:3
ENV PYTHONUNBUFFERED 1
RUN mkdir /code
WORKDIR /code
ADD requirements.txt /code/
RUN pip install -r requirements.txt
ADD . /code/
```

[19] https://docs.docker.com/compose/install/

The Dockerfile defines an application's image content via one or more build commands that configure that image.

Once built, you can run the image in a container.

For more information on Dockerfile, see the Docker user guide[20] and the Dockerfile reference[21].

This Dockerfile starts with a Python 3 parent image.

6.2.1.3.2.1 Les images Python

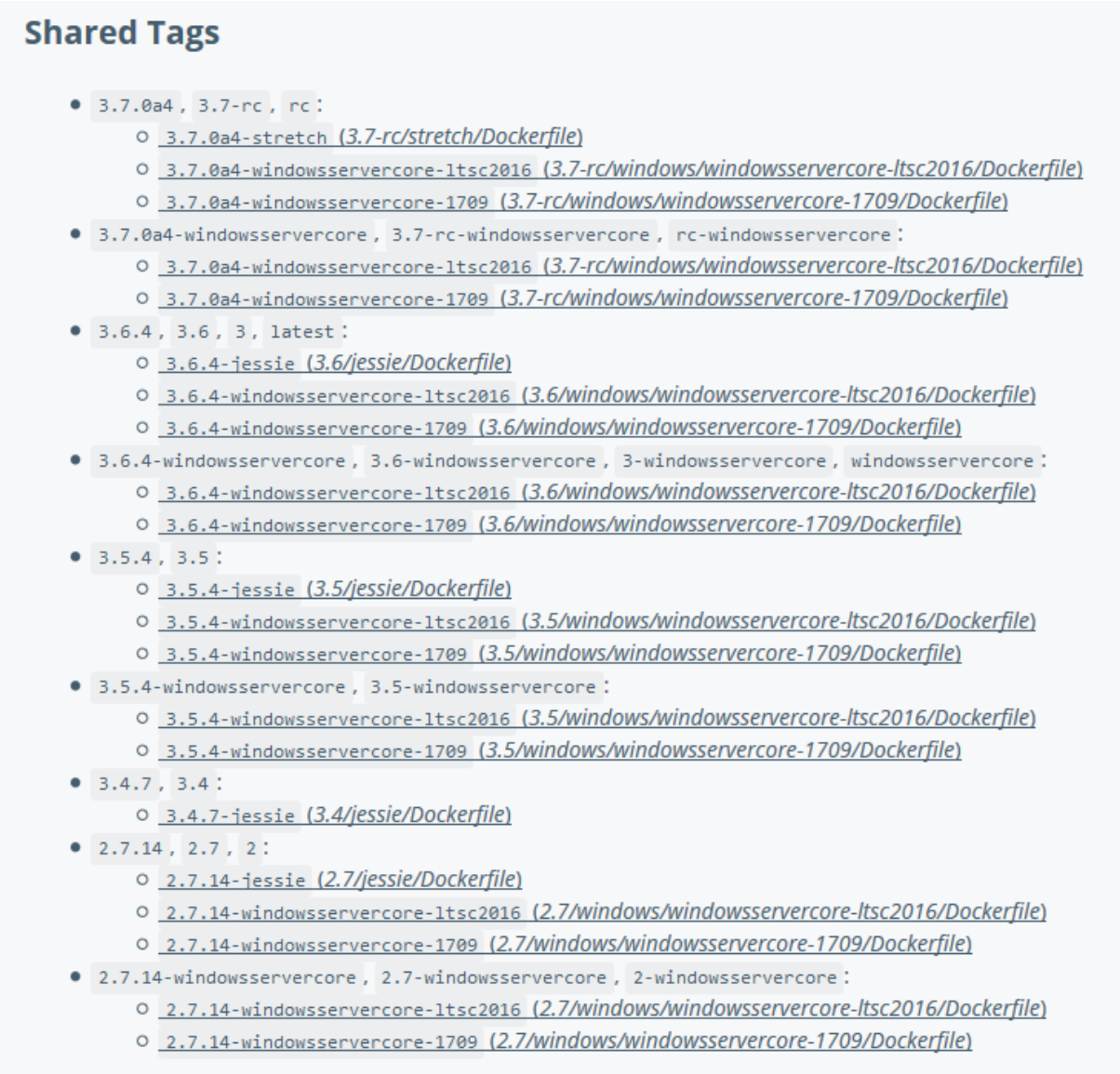


Fig. 1: Les images Python voir https://store.docker.com/images/python

Le tag **python:3** correspond à la version courante en 2018 c'est à dire **3.6**.

The parent image is modified by adding a new code directory. The parent image is further modified by installing the Python requirements defined in the requirements.txt file.

[20] https://docs.docker.com/engine/tutorials/dockerimages/#building-an-image-from-a-dockerfile
[21] https://docs.docker.com/engine/reference/builder/

6.2.1.3.3 Create a requirements.txt in your project directory

This file is used by the RUN pip install -r requirements.txt command in your Dockerfile.

```
django
psycopg2
```

6.2.1.3.4 Create a file called docker-compose.yml in your project directory

See also:

- https://docs.docker.com/compose/compose-file/
- https://store.docker.com/images/postgres

The docker-compose.yml file describes the services that make your app.

```
version: '3'

services:
  db:
    image: postgres
  web:
    build: .
    command: python3 manage.py runserver 0.0.0.0:8000
    volumes:
      - .:/code
    ports:
      - "8000:8000"
    depends_on:
      - db
```

This file defines two services: The **db** service and the **web** service.

6.2.1.3.4.1 Les images postresql

Fig. 2: Les images PostgreSQL voir https://store.docker.com/images/postgres

The compose file also describes which Docker images these services use, how they link together, any volumes they might need mounted inside the containers.

See the docker-compose.yml reference[22] for more information on how this file works

6.2.1.4 Create a Django project

In this step, you create a Django starter project by building the image from the build context defined in the previous procedure.

6.2.1.4.1 cd django_app

Change to the root of your project directory.

6.2.1.4.2 docker-compose run web django-admin.py startproject composeexample .

This instructs Compose to run django-admin.py startproject composeexample in a container, using the web service's image and configuration.

Because the web image doesn't exist yet, Compose builds it from the current directory, as specified by the build: . line in docker-compose.yml.

```
docker-compose run web django-admin.py startproject composeexample .
```

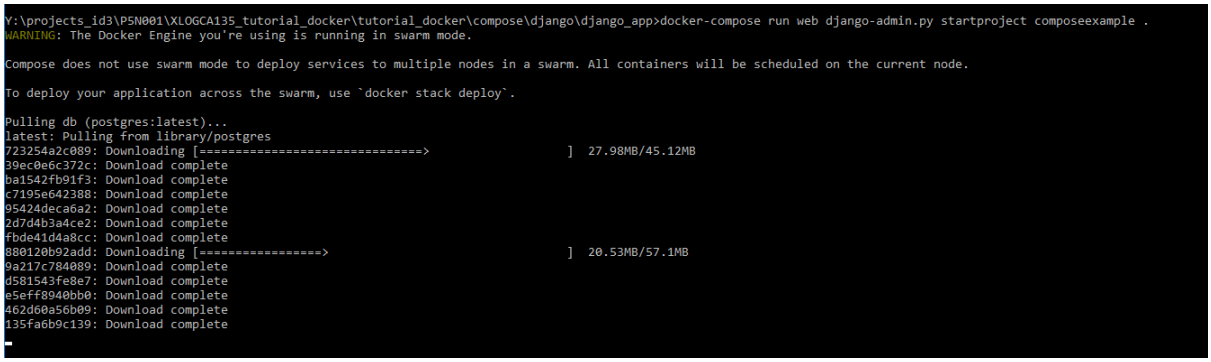


Fig. 3: docker-compose run web django-admin.py startproject composeexample .

```
Pulling db (postgres:latest)...
latest: Pulling from library/postgres
723254a2c089: Pull complete
39ec0e6c372c: Pull complete
ba1542fb91f3: Pull complete
c7195e642388: Pull complete
95424deca6a2: Pull complete
2d7d4b3a4ce2: Pull complete
fbde41d4a8cc: Pull complete
880120b92add: Pull complete
9a217c784089: Pull complete
d581543fe8e7: Pull complete
e5eff8940bb0: Pull complete
462d60a56b09: Pull complete
135fa6b9c139: Pull complete
Digest: sha256:3f4441460029e12905a5d447a3549ae2ac13323d045391b0cb0cf8b48ea17463
Status: Downloaded newer image for postgres:latest
Creating djangoapp_db_1 ... done
Building web
```

(continues on next page)

[22] https://docs.docker.com/compose/compose-file/

(continued from previous page)

```
Step 1/7 : FROM python:3
3: Pulling from library/python
f49cf87b52c1: Already exists
7b491c575b06: Pull complete
b313b08bab3b: Pull complete
51d6678c3f0e: Pull complete
09f35bd58db2: Pull complete
0f9de702e222: Pull complete
73911d37fcde: Pull complete
99a87e214c92: Pull complete
Digest: sha256:98149ed5f37f48ea3fad26ae6c0042dd2b08228d58edc95ef0fce35f1b3d9e9f
Status: Downloaded newer image for python:3
 ---> c1e459c00dc3
Step 2/7 : ENV PYTHONUNBUFFERED 1
 ---> Running in 94847219310a
Removing intermediate container 94847219310a
 ---> 221d2e9ab9e4
Step 3/7 : RUN mkdir /code
 ---> Running in a65c8bf5e5a9
Removing intermediate container a65c8bf5e5a9
 ---> 589950689c7a
Step 4/7 : WORKDIR /code
Removing intermediate container f7b978400775
 ---> e039064473fb
Step 5/7 : ADD requirements.txt /code/
 ---> 4305caf141b9
Step 6/7 : RUN pip install -r requirements.txt
 ---> Running in 0705839561d0
Collecting django (from -r requirements.txt (line 1))
  Downloading Django-2.0.1-py3-none-any.whl (7.1MB)
Collecting psycopg2 (from -r requirements.txt (line 2))
  Downloading psycopg2-2.7.3.2-cp36-cp36m-manylinux1_x86_64.whl (2.7MB)
Collecting pytz (from django->-r requirements.txt (line 1))
  Downloading pytz-2017.3-py2.py3-none-any.whl (511kB)
Installing collected packages: pytz, django, psycopg2
Successfully installed django-2.0.1 psycopg2-2.7.3.2 pytz-2017.3
Removing intermediate container 0705839561d0
 ---> fa8182703037
Step 7/7 : ADD . /code/
 ---> 72d70c82ea04
Successfully built 72d70c82ea04
Successfully tagged djangoapp_web:latest
WARNING: Image for service web was built because it did not already exist.
To rebuild this image you must use `docker-compose build` or `docker-compose up --
↪build`.
```

Once the web service image is built, Compose runs it and executes the django-admin.py startproject command in the container. This command instructs Django to create a set of files and directories representing a Django project.

6.2.1.4.2.1 tree /a /f .

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\compose\django\django_app>tree /a /f .
```

```
Y:\PROJECTS_ID3\P5N001\XLOGCA135_TUTORIAL_DOCKER\TUTORIAL_
↪DOCKER\COMPOSE\DJANGO\DJANGO_APP
|   docker-compose.yml
|   Dockerfile
|   manage.py
```

(continues on next page)

(continued from previous page)

```
|   requirements.txt
|
\---composeexample
                settings.py
                urls.py
                wsgi.py
                __init__.py
```

6.2.1.5 Connect the database

See also:

- https://store.docker.com/images/postgres

In this section, you set up the database connection for Django.

6.2.1.5.1 Edit the composeexample/settings.py file

In your project directory, edit the composeexample/settings.py file.

Replace the DATABASES = ... with the following:

```
DATABASES = {
        'default': {
                'ENGINE': 'django.db.backends.postgresql',
                'NAME': 'postgres',
                'USER': 'postgres',
                'HOST': 'db',
                'PORT': 5432,
        }
}
```

These settings are determined by the postgres Docker image specified in docker-compose.yml.

6.2.1.5.2 django_app> docker-compose up

Run the docker-compose up command from the top level directory for your project.

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\compose\django\django_app>docker-compose up
```

```
WARNING: The Docker Engine you're using is running in swarm mode.

Compose does not use swarm mode to deploy services to multiple nodes in a swarm.␣
↪All containers will be scheduled on the current node.

To deploy your application across the swarm, use `docker stack deploy`.

djangoapp_db_1 is up-to-date
Creating djangoapp_web_1 ... done
Attaching to djangoapp_db_1, djangoapp_web_1
db_1    | The files belonging to this database system will be owned by user
↪"postgres".
db_1    | This user must also own the server process.
db_1    |
db_1    | The database cluster will be initialized with locale "en_US.utf8".
db_1    | The default database encoding has accordingly been set to "UTF8".
```

(continues on next page)

(continued from previous page)

```
db_1    | The default text search configuration will be set to "english".
db_1    |
db_1    | Data page checksums are disabled.
db_1    |
db_1    | fixing permissions on existing directory /var/lib/postgresql/data ... ok
db_1    | creating subdirectories ... ok
db_1    | selecting default max_connections ... 100
db_1    | selecting default shared_buffers ... 128MB
db_1    | selecting dynamic shared memory implementation ... posix
db_1    | creating configuration files ... ok
db_1    | running bootstrap script ... ok
db_1    | performing post-bootstrap initialization ... ok
db_1    | syncing data to disk ... ok
db_1    |
db_1    | WARNING: enabling "trust" authentication for local connections
db_1    | You can change this by editing pg_hba.conf or using the option -A, or
db_1    | --auth-local and --auth-host, the next time you run initdb.
db_1    |
db_1    | Success. You can now start the database server using:
db_1    |
db_1    |     pg_ctl -D /var/lib/postgresql/data -l logfile start
db_1    |
db_1    | ****************************************************
db_1    | WARNING: No password has been set for the database.
db_1    |          This will allow anyone with access to the
db_1    |          Postgres port to access your database. In
db_1    |          Docker's default configuration, this is
db_1    |          effectively any other container on the same
db_1    |          system.
db_1    |
db_1    |          Use "-e POSTGRES_PASSWORD=password" to set
db_1    |          it in "docker run".
db_1    | ****************************************************
db_1    | waiting for server to start....2018-01-18 09:51:04.629 UTC [37] LOG:
↪listening on IPv4 address "127.0.0.1", port 5432
db_1    | 2018-01-18 09:51:04.630 UTC [37] LOG:  could not bind IPv6 address "::1":
↪Cannot assign requested address
db_1    | 2018-01-18 09:51:04.630 UTC [37] HINT:  Is another postmaster already
↪running on port 5432? If not, wait a few seconds and retry.
db_1    | 2018-01-18 09:51:04.755 UTC [37] LOG:  listening on Unix socket "/var/run/
↪postgresql/.s.PGSQL.5432"
db_1    | 2018-01-18 09:51:04.916 UTC [38] LOG:  database system was shut down at
↪2018-01-18 09:51:02 UTC
db_1    | 2018-01-18 09:51:04.976 UTC [37] LOG:  database system is ready to accept
↪connections
db_1    |  done
db_1    | server started
db_1    | ALTER ROLE
db_1    |
db_1    |
db_1    | /usr/local/bin/docker-entrypoint.sh: ignoring /docker-entrypoint-initdb.d/
↪*
db_1    |
db_1    | 2018-01-18 09:51:05.165 UTC [37] LOG:  received fast shutdown request
db_1    | waiting for server to shut down....2018-01-18 09:51:05.224 UTC [37] LOG:
↪aborting any active transactions
db_1    | 2018-01-18 09:51:05.226 UTC [37] LOG:  worker process: logical
↪replication launcher (PID 44) exited with exit code 1
db_1    | 2018-01-18 09:51:05.228 UTC [39] LOG:  shutting down
db_1    | 2018-01-18 09:51:05.860 UTC [37] LOG:  database system is shut down
db_1    |  done
```

(continues on next page)

(continued from previous page)

```
db_1   | server stopped
db_1   |
db_1   | PostgreSQL init process complete; ready for start up.
db_1   |
db_1   | 2018-01-18 09:51:05.947 UTC [1] LOG:  listening on IPv4 address "0.0.0.0",
↪ port 5432
db_1   | 2018-01-18 09:51:05.947 UTC [1] LOG:  listening on IPv6 address "::",␣
↪port 5432
db_1   | 2018-01-18 09:51:06.080 UTC [1] LOG:  listening on Unix socket "/var/run/
↪postgresql/.s.PGSQL.5432"
db_1   | 2018-01-18 09:51:06.278 UTC [55] LOG:  database system was shut down at␣
↪2018-01-18 09:51:05 UTC
db_1   | 2018-01-18 09:51:06.340 UTC [1] LOG:  database system is ready to accept␣
↪connections
web_1  | Performing system checks...
web_1  |
web_1  | System check identified no issues (0 silenced).
web_1  |
web_1  | You have 14 unapplied migration(s). Your project may not work properly␣
↪until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
web_1  | Run 'python manage.py migrate' to apply them.
web_1  | January 18, 2018 - 10:46:37
web_1  | Django version 2.0.1, using settings 'composeexample.settings'
web_1  | Starting development server at http://0.0.0.0:8000/
web_1  | Quit the server with CONTROL-C.
```

At this point, your Django app should be running at port 8000 on your Docker host.

On Docker for Mac and Docker for Windows, go to http://localhost:8000 on a web browser to see the Django welcome page

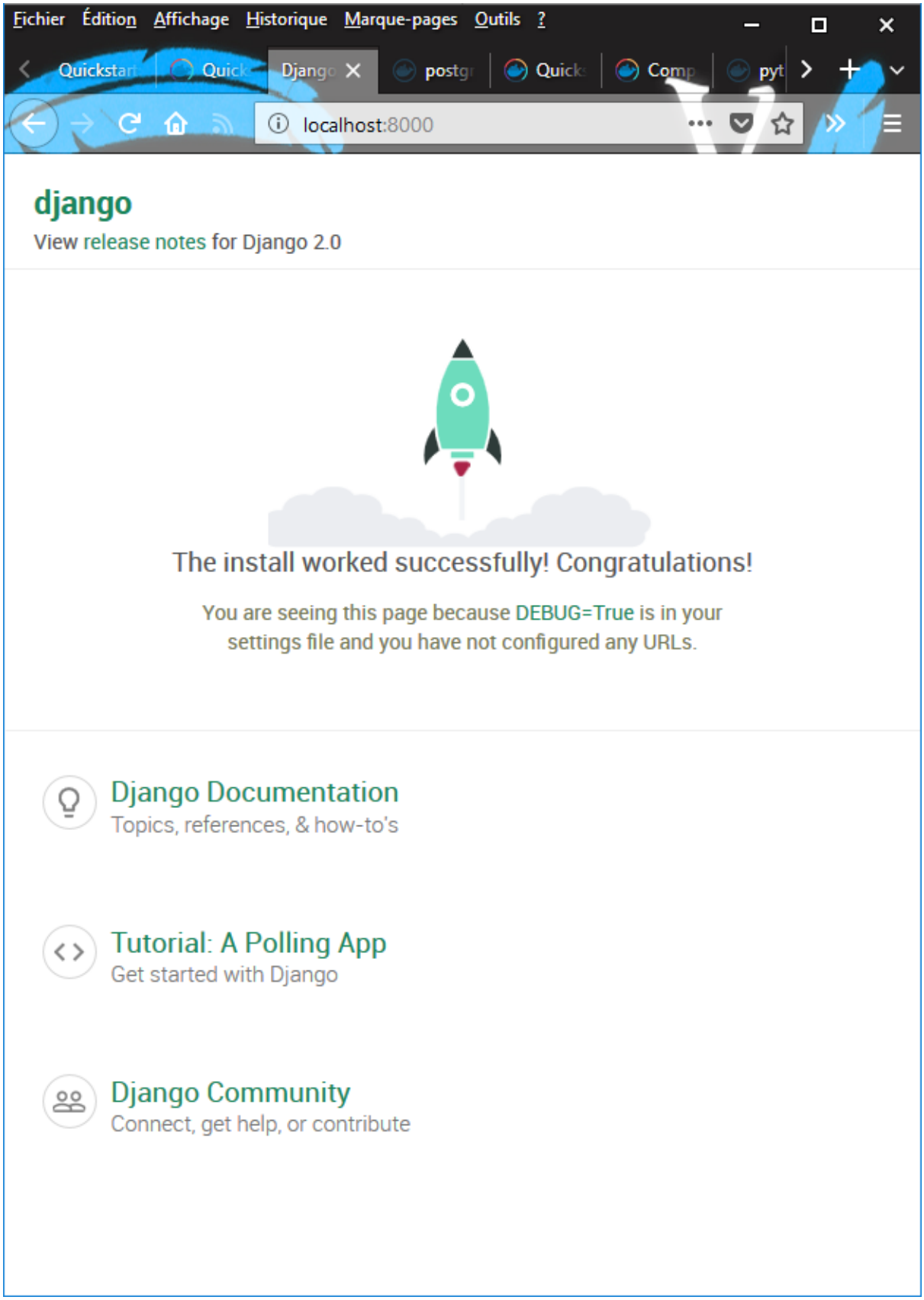


Fig. 4: http://localhost:8000/

6.2.1.5.3 docker ps

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_docker>docker ps
```

```
CONTAINER ID        IMAGE               COMMAND                  CREATED
↪  STATUS              PORTS                    NAMES
30b4922c00b2        djangoapp_web       "python3 manage.py r..."   About an hour␣
↪ago   Up About an hour    0.0.0.0:8000->8000/tcp   djangoapp_web_1
0883a9ef1c3b        postgres            "docker-entrypoint.s..."   2 hours ago
↪    Up 2 hours          5432/tcp                 djangoapp_db_1
```

6.2.1.5.4 django_app> docker-compose down

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\compose\django\django_app>docker-compose down
```

```
Stopping djangoapp_web_1 ... done
Stopping djangoapp_db_1  ... done
Removing djangoapp_web_1     ... done
Removing djangoapp_web_run_1 ... done
Removing djangoapp_db_1      ... done
Removing network djangoapp_default
```

6.2.1.6 Compose file examples

6.2.1.6.1 Compose file examples

6.2.1.6.1.1 Compose file example 1

See also:

- https://github.com/pydanny/cookiecutter-django/issues/1258

Contents

6.2.1.6.1.2 base.yml

```
version: '3.2'
services:
  postgres:
        build: ./compose/postgres
        environment:
          - POSTGRES_USER_FILE=/run/secrets/pg_username
          - POSTGRES_PASSWORD_FILE=/run/secrets/pg_password
        secrets:
          - pg_username
          - pg_password
```

(continues on next page)

(continued from previous page)

```
  django:
        command: /gunicorn.sh
        environment:
          - USE_DOCKER=$DAPI_VAR:-yes
          - DATABASE_URL=postgres://{username}:{password}@postgres:5432/{username}
          - SECRETS_FILE=/run/secrets/django_s
          - POSTGRES_USER_FILE=/run/secrets/pg_username
          - POSTGRES_PASSWORD_FILE=/run/secrets/pg_password
        # My Deploy
        deploy:
                  replicas: 1
                  restart_policy:
                        condition: on-failure
        secrets:
          - pg_username
          - pg_password
          - django_s

secrets:
        django_s:
                external: True
        pg_username:
                external: True
        pg_password:
                external: True
```

6.2.1.6.1.3 dev.yml

```
version: '3.2'

volumes:
  postgres_data_dev: {}
  postgres_backup_dev: {}

services:
  postgres:
        image: apple_postgres
        volumes:
          - postgres_data_dev:/var/lib/postgresql/data
          - postgres_backup_dev:/backups

  django:
        image: apple_django
        build:
          context: .
          dockerfile: ./compose/django/Dockerfile-dev
        command: /start-dev.sh
        volumes:
          - .:/app
        ports:
          - "8000:8000"
        secrets:
          - pg_username
          - pg_password
          - source: django_s
                #target: /app//.env

  node:
```

(continues on next page)

(continued from previous page)

```
        image: apple_node
        #user: $USER:-0
        build:
          context: .
          dockerfile: ./compose/node/Dockerfile-dev
        volumes:
          - .:/app
          - ${PWD}/gulpfile.js:/app/gulpfile.js
          # http://jdlm.info/articles/2016/03/06/lessons-building-node-app-docker.
↪html
          - /app/node_modules
          - /app/vendor
        command: "gulp"
        ports:
          # BrowserSync port.
          - "3000:3000"
          # BrowserSync UI port.
          - "3001:3001"
```

6.2.1.6.1.4 Compose file example 2

See also:

- http://ramkulkarni.com/blog/docker-project-for-python3-djaongo-and-apache2-setup/

Contents

Chapter 7

Bonnes pratiques Docker

Contents

7.1 actu.alfa-safety.fr

See also:

- https://actu.alfa-safety.fr/devops/docker-en-production/

Docker est largement utilisé en développement mais bien souvent les choses se compliquent en production:

- d'abord l'application ne fonctionne pas toujours correctement en prod,
- les performances ne suivent pas,
- ensuite on s'aperçoit que l'on a oublié de prendre en compte un certain nombre d'éléments indispensables en production: monitoring, scalabilité, contraintes réseaux.

La facilité est alors de dire : Docker fonctionne bien en Dev mais n'est pas un outil adapté à la production. **Bien au contraire**, Docker en production doit permettre de faciliter et sécuriser les déploiements tout en rendant votre application scalable.

Mais pour cela, il faut bien fonctionner en mode Devops et respecter un certain nombre de bonnes pratiques. C'est en tant que telle une compétence ou expertise Docker en production qu'il faut développer.

Enfin quand votre production atteint une certaine complexité, et que le nombre de conteneurs que vous gérez se compte en dizaines, il faut envisager de passer sur un orchestrateur de conteneurs.

Avant d'attaquer le vif du sujet, vous pouvez revenir sur notre précédent article sur les bases de Docker.

7.2 Best practices for writing Dockerfiles

See also:

- https://docs.docker.com/engine/userguide/eng-image/dockerfile_best-practices/
- https://docs.docker.com/engine/reference/builder/

Docker can build images automatically by reading the instructions from a Dockerfile, a text file that contains all the commands, in order, needed to build a given image. Dockerfiles adhere to a specific format and use a specific set of instructions.

You can learn the basics on the Dockerfile Reference page[23]. If you're new to writing Dockerfiles, you should start there.

This document covers the best practices and methods recommended by Docker, Inc. and the Docker community for building efficient images.

To see many of these practices and recommendations in action, check out the Dockerfile for buildpack-deps.

Note: for more detailed explanations of any of the Dockerfile commands mentioned here, visit the Dockerfile Reference page.

---

[23] https://docs.docker.com/engine/reference/builder/

Chapter 8

Images Docker (Store Docker, ex Hub docker)

See also:

- https://store.docker.com/
- https://docs.docker.com/engine/userguide/eng-image/
- https://github.com/docker-library
- https://github.com/docker-library/official-images
- https://hub.docker.com/explore/

Contents

8.1 Nouveau: le docker store: https://store.docker.com/

Fig. 1: https://store.docker.com/

8.2 Ancien: le hub docker https://hub.docker.com/explore/

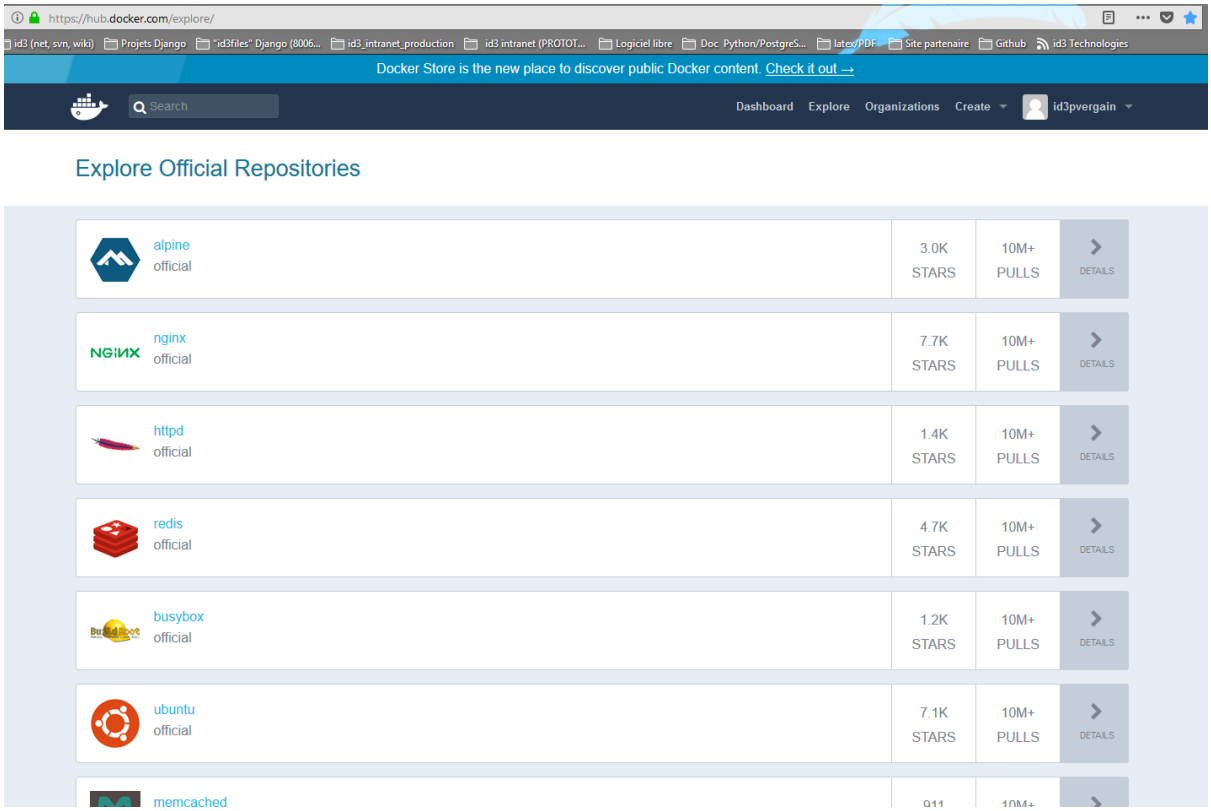


Fig. 2: https://hub.docker.com/explore/

8.3 Gitlab registry

8.3.1 GitLab Container Registry

See also:

- https://docs.gitlab.com/ce/user/project/container_registry.html

Contents

8.3.1.1 Introduction

With the Docker Container Registry integrated into GitLab, **every project can have its own space to store its Docker images**.

8.4 Images OS

8.4.1 Images Alpine

See also:

- https://store.docker.com/images/alpine
- https://hub.docker.com/_/alpine/
- https://www.alpinelinux.org/
- https://fr.wikipedia.org/wiki/Alpine_Linux
- https://github.com/gliderlabs/docker-alpine
- http://gliderlabs.viewdocs.io/docker-alpine/

Contents

Fig. 3: Le logo Alpine-linux

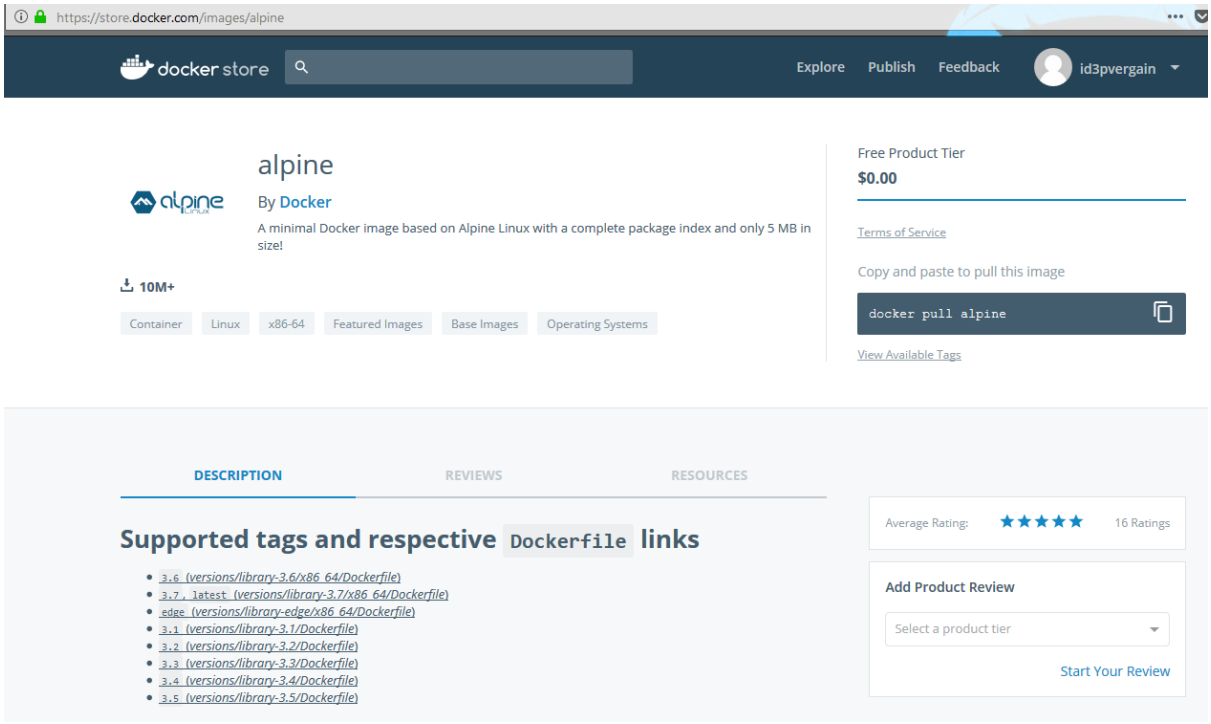


Fig. 4: https://store.docker.com/images/alpine

8.4.1.1 Short Description

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

8.4.1.2 Description

See also:

- https://fr.wikipedia.org/wiki/Alpine_Linux

Alpine Linux est une distribution Linux ultra-légère, orientée sécurité et basée sur Musl et BusyBox, principalement conçue pour “Utilisateur intensif qui apprécie la sécurité, la simplicité et l’efficacité des ressources”.

Elle utilise les patches PaX et Grsecurity du noyau par défaut et compile tous les binaires de l’espace utilisateur et exécutables indépendants de la position (dits “portables”) avec protection de destruction de la pile.

Cette distribution se prête particulièrement, en raison de sa légèreté, à la création d’images de containers Docker. La distribution Alpine Linux est particulièrement populaire pour cet usage .

8.4.1.3 Dockerfile

```
FROM scratch
ADD rootfs.tar.xz /
CMD ["/bin/sh"]
```

8.4.2 Images Debian

See also:

- https://store.docker.com/images/debian
- https://hub.docker.com/_/debian/

- https://fr.wikipedia.org/wiki/Debian

Contents

Fig. 5: Logo Debian

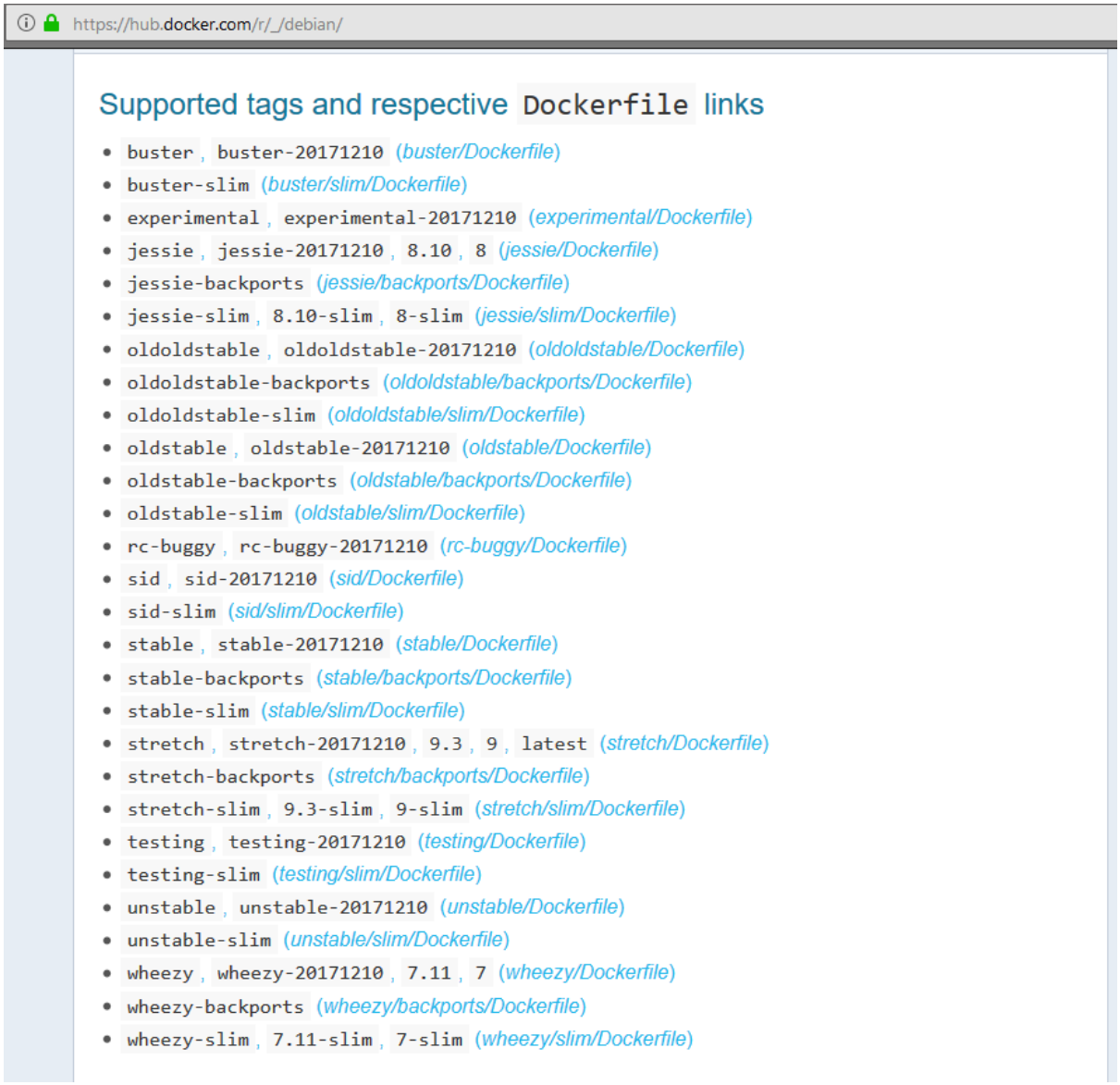


Fig. 6: https://hub.docker.com/_/debian/

8.4.2.1 Short Description

Debian is a Linux distribution that's composed entirely of free and open-source software.

8.4.2.2 Description

See also:

- https://fr.wikipedia.org/wiki/Debian

Debian (/de.bjan/) est une organisation communautaire et démocratique, dont le but est le développement de systèmes d'exploitation basés exclusivement sur des logiciels libres.

Chaque système, lui-même nommé Debian, réunit autour d'un noyau de système d'exploitation de nombreux éléments pouvant être développés indépendamment les uns des autres, pour plusieurs architectures matérielles. Ces éléments, programmes de base complétant le noyau et logiciels applicatifs, se présentent sous forme de « paquets » qui peuvent être installés en fonction des besoins (voir Distribution des logiciels). L'ensemble système d'exploitation plus logiciels s'appelle une distribution.

On assimile généralement ces systèmes d'exploitation au système Debian GNU/Linux, la distribution GNU/Linux de Debian, car jusqu'en 2009 c'était la seule branche parfaitement fonctionnelle. Mais d'autres distributions Debian sont en cours de développement en 2013 : Debian GNU/Hurd3, et Debian GNU/kFreeBSD5. La version Debian *Squeeze* est la première à être distribuée avec le noyau kFreeBSD en plus du noyau Linux6.

Debian est utilisée comme base de nombreuses autres distributions telles que Knoppix et Ubuntu qui rencontrent un grand succès.

8.4.3 Images Ubuntu

See also:

- https://store.docker.com/images/ubuntu
- https://hub.docker.com/_/ubuntu/
- https://fr.wikipedia.org/wiki/Ubuntu_(syst%C3%A8me_d%27exploitation)

Contents

Fig. 7: Le logo Ubuntu

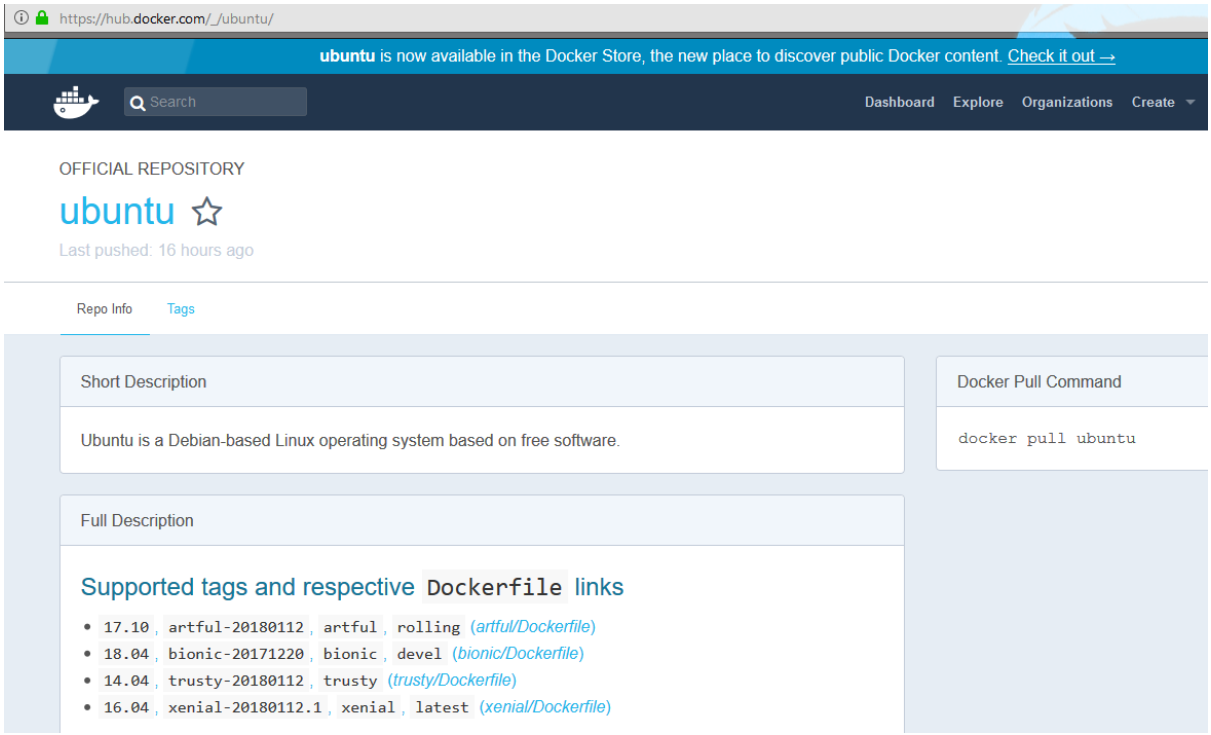


Fig. 8: https://hub.docker.com/_/ubuntu/

8.4.3.1 Short Description

Ubuntu is a Debian-based Linux operating system based on free software.

8.4.3.2 Description

See also:

- https://fr.wikipedia.org/wiki/Ubuntu_(syst%C3%A8me_d%27exploitation)

Ubuntu (prononciation : /u.bun.tu/) est un système d'exploitation GNU/Linux basé sur la distribution Linux Debian. Il est développé, commercialisé et maintenu pour les ordinateurs individuels par la société Canonical.

Ubuntu se définit comme « un système d'exploitation utilisé par des millions de PC à travers le monde »10 et avec une interface « simple, intuitive, et sécurisée ».

Elle est la distribution la plus consultée sur Internet d'après le site Alexa. Et est le système d'exploitation le plus utilisé sur les systèmes Cloud ainsi que sur les serveurs informatiques.

Ubuntu se divise en deux branches :

- La branche principale stable dit LTS. Avec mise à niveau tous les six mois et mise à jour majeure tous les 2 ans. La dernière version 16.04.3, nom de code *Xenial Xerus* est sortie le 3 août 2017.
- La branche secondaire instable avec mise à jour majeure tous les six mois. La dernière version en date est la 17.10, nom de code *Artful Aardvark** est sortie le 19 octobre 2017.

8.4.3.3 La Philosophie d'Ubuntu

Le mot ubuntu provient d'un ancien mot bantou (famille de langues africaines) qui désigne une personne qui prend conscience que son *moi* est intimement lié à ce que sont les autres. Autrement dit : **Je suis ce que je suis grâce à ce que nous sommes tous**.

C'est un concept fondamental de la « philosophie de la réconciliation » développée par Desmond Mpilo Tutu avec l'abolition de l'apartheid.

Ubuntu signifie par ailleurs en kinyarwanda (langue rwandaise) et en kirundi (langue burundaise) *humanité*, *générosité* ou *gratuité*.

On dit d'une chose qu'elle est *k'ubuntu* si elle est obtenue gratuitement.

En informatique, on considère qu'une distribution existe aux travers des apports des différentes communautés Linux. Et tel qu'il se trouve expliqué dans le travail de la Commission de la vérité et de la réconciliation. Elles permettent de mieux saisir par exemple la mission de la Fondation Shuttleworth relayée en France par les travaux de philosophes comme Barbara Cassin et Philippe-Joseph Salazar.

8.4.4 Images CentOS

See also:

- https://store.docker.com/images/centos
- https://hub.docker.com/_/centos/
- https://fr.wikipedia.org/wiki/CentOS
- https://www.centos.org/

Contents

Fig. 9: Logo CentOS

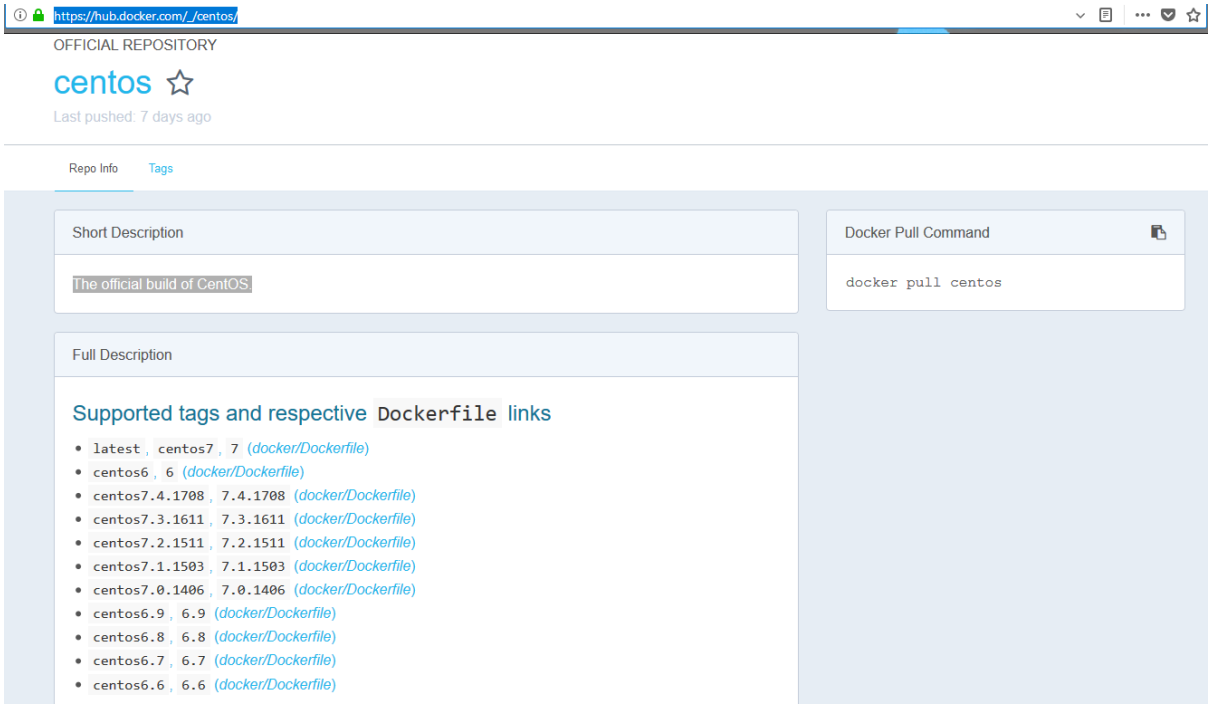


Fig. 10: https://hub.docker.com/_/centos/

8.4.4.1 Short Description

The official build of CentOS.

8.4.4.2 Description

See also:

- https://fr.wikipedia.org/wiki/CentOS

CentOS (Community enterprise Operating System) est une distribution GNU/Linux principalement destinée aux serveurs. Tous ses paquets, à l'exception du logo, sont des paquets compilés à partir des sources de la distribution RHEL (Red Hat Enterprise Linux), éditée par la société Red Hat. Elle est donc quasiment identique à celle-ci et se veut 100 % compatible d'un point de vue binaire.

Utilisée par 20 % des serveurs web Linux, elle est l'une des distributions Linux les plus populaires pour les serveurs web. Depuis novembre 2013, elle est la troisième distribution la plus utilisée sur les serveurs web ; en avril 2017, elle était installée sur 20,6 % d'entre eux ; les principales autres distributions étaient Debian (31,8 %), Ubuntu (35,8 %) et Red Hat (3,3 %).

8.4.4.3 Structures

La RHEL en version binaire, directement installable et exploitable, ne peut être obtenue que par achat d'une souscription auprès de Red Hat ou de ses revendeurs. La plupart des programmes inclus et livrés avec la Red Hat sont publiés sous la licence GPL, qui impose au redistributeur (sous certaines conditions) de fournir les sources. CentOS utilise donc les sources de la RHEL (accessibles librement sur Internet) pour regénérer la Red Hat à l'identique.

On peut donc considérer la CentOS comme une version gratuite de la Red Hat. Le support technique est de type communautaire : il se fait gratuitement et ouvertement via les listes de diffusion et les forums de la communauté CentOS.

Depuis le 7 janvier 2014, Red Hat et CentOS se sont fortement rapprochées, puisque la plupart des principaux membres maintenant la CentOS ont été embauchés par Red Hat.

8.5 Images langages

8.5.1 Images Python

See also:

- https://store.docker.com/images/python
- https://hub.docker.com/_/python/

Contents

Fig. 11: Le logo Python

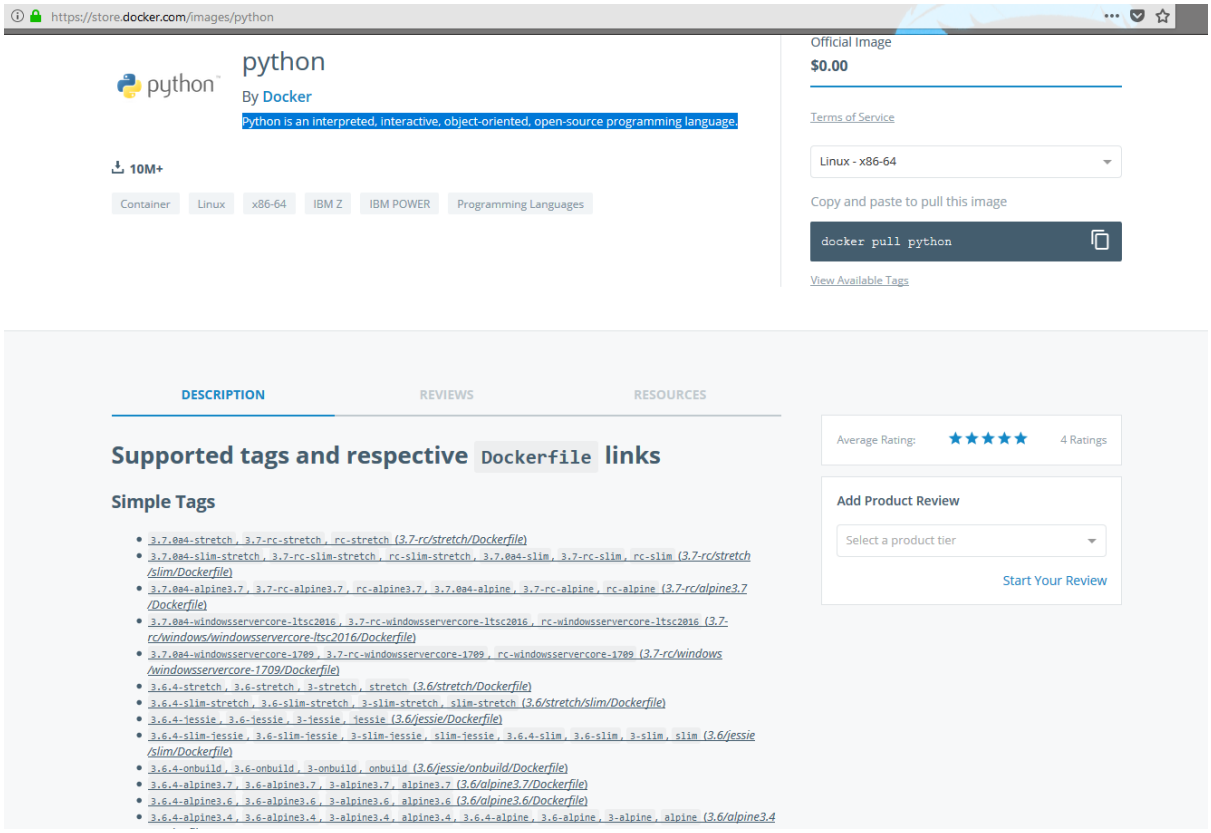


Fig. 12: https://store.docker.com/images/python

8.5.1.1 Short Description

Python is an interpreted, interactive, object-oriented, open-source programming language.

8.5.1.2 What is Python ?

Python is an interpreted, interactive, object-oriented, open-source programming language.

It incorporates modules, exceptions, dynamic typing, very high level dynamic data types, and classes. Python combines remarkable power with very clear syntax.

It has interfaces to many system calls and libraries, as well as to various window systems, and is extensible in C or C++.

It is also usable as an extension language for applications that need a programmable interface.

Finally, Python is portable: it runs on many Unix variants, on the Mac, and on Windows 2000 and later.

8.5.1.3 How to use this image

Create a Dockerfile in your Python app project

```
FROM python:3

WORKDIR /usr/src/app

COPY requirements.txt ./
RUN pip install --no-cache-dir -r requirements.txt

COPY . .

CMD [ "python", "./your-daemon-or-script.py" ]

You can then build and run the Docker image:
```

```
docker build -t my-python-app .
```

```
docker run -it --rm --name my-running-app my-python-app
```

Run a single Python script

For many simple, single file projects, you may find it inconvenient to write a complete Dockerfile. In such cases, you can run a Python script by using the Python Docker image directly:

```
docker run -it --rm --name my-running-script -v "$PWD":/usr/src/myapp -w /usr/src/
↪myapp python:3 python your-daemon-or-script.py
```

8.5.2 Images PHP

See also:

- https://store.docker.com/images/php
- https://en.wikipedia.org/wiki/PHP

Contents

Fig. 13: Le logo PHP

8.5.2.1 Short Description

While designed for web development, the PHP scripting language also provides general-purpose use.

8.5.2.2 What is PHP ?

See also:

- https://en.wikipedia.org/wiki/PHP

PHP is a server-side scripting language designed for web development, but which can also be used as a general-purpose programming language. PHP can be added to straight HTML or it can be used with a variety of templating engines and web frameworks.

PHP code is usually processed by an interpreter, which is either implemented as a native module on the web-server or as a common gateway interface (CGI).

8.5.3 Images Ruby

See also:

- https://store.docker.com/images/ruby
- https://en.wikipedia.org/wiki/Ruby_%28programming_language%29

Contents

Fig. 14: Le logo Ruby

8.5.3.1 Short Description

Ruby is a dynamic, reflective, object-oriented, general-purpose, open-source programming language.

8.5.3.2 What is Ruby ?

See also:

- https://en.wikipedia.org/wiki/Ruby_%28programming_language%29

Ruby is a dynamic, reflective, object-oriented, general-purpose, open-source programming language.

According to its authors, Ruby was influenced by Perl, Smalltalk, Eiffel, Ada, and Lisp. It supports multiple programming paradigms, including functional, object-oriented, and imperative. It also has a dynamic type system and automatic memory management.

8.5.4 Images Node

See also:

- https://store.docker.com/images/node
- https://en.wikipedia.org/wiki/Ruby_%28programming_language%29

Contents

Fig. 15: Le logo Node.js

8.5.4.1 Short Description

Node.js is a JavaScript-based platform for server-side and networking applications.

8.5.4.2 What is Node.js ?

See also:

- https://en.wikipedia.org/wiki/Node.js

Node.js is a software platform for scalable server-side and networking applications.

Node.js applications are written in JavaScript and can be run within the Node.js runtime on Mac OS X, Windows, and Linux without changes.

Node.js applications are designed to maximize throughput and efficiency, using non-blocking I/O and asynchronous events.

Node.js applications run single-threaded, although Node.js uses multiple threads for file and network events.

Node.js is commonly used for real-time applications due to its asynchronous nature.

Node.js internally uses the Google V8 JavaScript engine to execute code; a large percentage of the basic modules are written in JavaScript.

Node.js contains a built-in, asynchronous I/O library for file, socket, and HTTP communication.

The HTTP and socket support allows Node.js to act as a **web server** without additional software such as Apache.

8.5.5 Images Go (Golang)

See also:

- https://store.docker.com/images/golang
- https://en.wikipedia.org/wiki/Go_%28programming_language%29

Contents

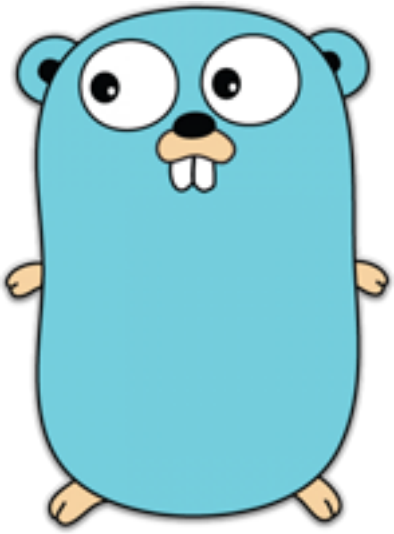


Fig. 16: Le logo Golang

8.5.5.1 Short Description

Node.js is a JavaScript-based platform for server-side and networking applications.

8.5.5.2 What is Go ?

See also:

- https://en.wikipedia.org/wiki/Go_%28programming_language%29

Go (a.k.a., Golang) is a programming language first developed at Google.

It is a statically-typed language with syntax loosely derived from C, but with additional features such as garbage collection, type safety, some dynamic-typing capabilities, additional built-in types (e.g., variable-length arrays and key-value maps), and a large standard library.

8.5.6 Images OpenJDK (Java)

See also:

- https://hub.docker.com/_/openjdk/
- https://en.wikipedia.org/wiki/OpenJDK
- http://openjdk.java.net/
- https://github.com/docker-library/openjdk
- https://twitter.com/OpenJDK

Contents

Fig. 17: Le logo OpenJDK

Full Description

Supported tags and respective `Dockerfile` links

Simple Tags

- `10-ea-32-jre-experimental`, `10-ea-jre-experimental`, `10-jre-experimental`, `10-ea-32-jre`, `10-ea-jre`, `10-jre` *(10-jre/Dockerfile)*
- `10-ea-32-jre-slim-experimental`, `10-ea-jre-slim-experimental`, `10-jre-slim-experimental`, `10-ea-32-jre-slim`, `10-ea-jre-slim`, `10-jre-slim` *(10-jre/slim/Dockerfile)*
- `10-ea-32-jdk-experimental`, `10-ea-32-experimental`, `10-ea-jdk-experimental`, `10-ea-experimental`, `10-jdk-experimental`, `10-experimental`, `10-ea-32-jdk`, `10-ea-32`, `10-ea-jdk`, `10-ea`, `10-jdk`, `10` *(10-jdk/Dockerfile)*
- `10-ea-32-jdk-slim-experimental`, `10-ea-32-slim-experimental`, `10-ea-jdk-slim-experimental`, `10-ea-slim-experimental`, `10-jdk-slim-experimental`, `10-slim-experimental`, `10-ea-32-jdk-slim`, `10-ea-32-slim`, `10-ea-jdk-slim`, `10-ea-slim`, `10-jdk-slim`, `10-slim` *(10-jdk/slim/Dockerfile)*
- `9.0.1-11-jre-sid`, `9.0.1-jre-sid`, `9.0-jre-sid`, `9-jre-sid`, `9.0.1-11-jre`, `9.0.1-jre`, `9.0-jre`, `9-jre` *(9-jre/Dockerfile)*
- `9.0.1-11-jre-slim-sid`, `9.0.1-jre-slim-sid`, `9.0-jre-slim-sid`, `9-jre-slim-sid`, `9.0.1-11-jre-slim`, `9.0.1-jre-slim`, `9.0-jre-slim`, `9-jre-slim` *(9-jre/slim/Dockerfile)*
- `9.0.1-11-jdk-sid`, `9.0.1-11-sid`, `9.0.1-jdk-sid`, `9.0.1-sid`, `9.0-jdk-sid`, `9.0-sid`, `9-jdk-sid`, `9-sid`, `9.0.1-11-jdk`, `9.0.1-11`, `9.0.1-jdk`, `9.0.1`, `9.0-jdk`, `9.0`, `9-jdk`, `9` *(9-jdk/Dockerfile)*
- `9.0.1-11-jdk-slim-sid`, `9.0.1-11-slim-sid`, `9.0.1-jdk-slim-sid`, `9.0.1-slim-sid`, `9.0-jdk-slim-sid`, `9.0-slim-sid`, `9-jdk-slim-sid`, `9-slim-sid`, `9.0.1-11-jdk-slim`, `9.0.1-11-slim`, `9.0.1-jdk-slim`, `9.0.1-slim`, `9.0-jdk-slim`, `9.0-slim`, `9-jdk-slim`, `9-slim` *(9-jdk/slim/Dockerfile)*
- `9.0.1-jdk-windowsservercore-ltsc2016`, `9.0.1-windowsservercore-ltsc2016`, `9.0-jdk-windowsservercore-ltsc2016`, `9.0-windowsservercore-ltsc2016`, `9-jdk-windowsservercore-`

Fig. 18: https://hub.docker.com/_/openjdk/

8.5.6.1 Short Description

OpenJDK is an open-source implementation of the Java Platform, Standard Edition

8.5.6.2 What is OpenJDK ?

See also:

- https://en.wikipedia.org/wiki/OpenJDK

OpenJDK (Open Java Development Kit) is a free and open source implementation of the Java Platform, Standard Edition (Java SE).

OpenJDK is the official reference implementation of Java SE since version 7.

8.5.6.3 How to use this image

Start a Java instance in your app

The most straightforward way to use this image is to use a Java container as both the build and runtime environment. In your Dockerfile, writing something along the lines of the following will compile and run your project:

```
FROM openjdk:7
COPY . /usr/src/myapp
WORKDIR /usr/src/myapp
RUN javac Main.java
CMD ["java", "Main"]
```

You can then run and build the Docker image:

```
$ docker build -t my-java-app .
$ docker run -it --rm --name my-running-app my-java-app
```

8.6 Images webserver: serveurs HTTP (serveurs Web)

See also:

- https://fr.wikipedia.org/wiki/Serveur_HTTP
- https://fr.wikipedia.org/wiki/Serveur_HTTP#Logiciels_de_serveur_HTTP
- https://en.wikipedia.org/wiki/Web_server

Le serveur HTTP le plus utilisé est Apache HTTP Server qui sert environ 55% des sites web en janvier 2013 selon Netcraft.

Le serveur HTTP le plus utilisé dans les 1 000 sites les plus actifs est en revanche Nginx avec 38,2% de parts de marché en 2016 selon w3techs et 53,9% en avril 2017.

8.6.1 Images Apache HTTPD

See also:

- https://hub.docker.com/_/httpd/
- https://en.wikipedia.org/wiki/Apache_HTTP_Server
- https://httpd.apache.org/

Contents

Fig. 19: Le logo Apache HTTPD

Full Description

Supported tags and respective `Dockerfile` links

- `2.2.34`, `2.2` (*2.2/Dockerfile*)
- `2.2.34-alpine`, `2.2-alpine` (*2.2/alpine/Dockerfile*)
- `2.4.29`, `2.4`, `2`, `latest` (*2.4/Dockerfile*)
- `2.4.29-alpine`, `2.4-alpine`, `2-alpine`, `alpine` (*2.4/alpine/Dockerfile*)

Fig. 20: https://hub.docker.com/_/httpd/

8.6.1.1 Short Description

The Apache HTTP Server Project.

8.6.1.2 What is httpd ?

The Apache HTTP Server, colloquially called **Apache**, is a Web server application notable for playing a key role in the initial growth of the World Wide Web.

Originally based on the NCSA HTTPd server, development of Apache began in early 1995 after work on the NCSA code stalled.

Apache quickly overtook NCSA HTTPd as the dominant HTTP server, and has remained the most popular HTTP server in use since April 1996.

8.6.2 Images apache Tomcat

See also:

- https://hub.docker.com/_/tomcat/
- https://fr.wikipedia.org/wiki/Apache_Tomcat

Contents

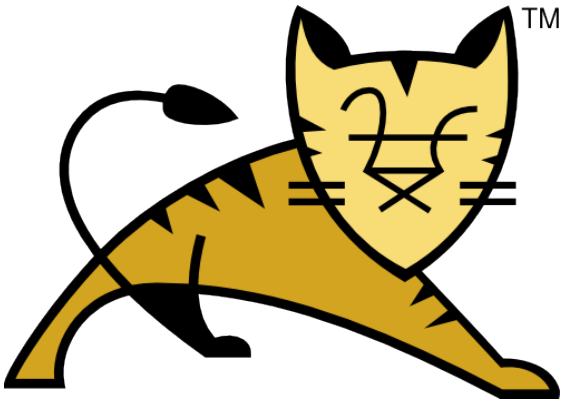


Fig. 21: Le logo Apache Tomcat

Supported tags and respective `Dockerfile` links

- `7.0.82-jre7`, `7.0-jre7`, `7-jre7`, `7.0.82`, `7.0`, `7` *(7/jre7/Dockerfile)*
- `7.0.82-jre7-slim`, `7.0-jre7-slim`, `7-jre7-slim`, `7.0.82-slim`, `7.0-slim`, `7-slim` *(7/jre7-slim/Dockerfile)*
- `7.0.82-jre7-alpine`, `7.0-jre7-alpine`, `7-jre7-alpine`, `7.0.82-alpine`, `7.0-alpine`, `7-alpine` *(7/jre7-alpine/Dockerfile)*
- `7.0.82-jre8`, `7.0-jre8`, `7-jre8` *(7/jre8/Dockerfile)*
- `7.0.82-jre8-slim`, `7.0-jre8-slim`, `7-jre8-slim` *(7/jre8-slim/Dockerfile)*
- `7.0.82-jre8-alpine`, `7.0-jre8-alpine`, `7-jre8-alpine` *(7/jre8-alpine/Dockerfile)*
- `8.0.48-jre7`, `8.0-jre7`, `8.0.48`, `8.0` *(8.0/jre7/Dockerfile)*
- `8.0.48-jre7-slim`, `8.0-jre7-slim`, `8.0.48-slim`, `8.0-slim` *(8.0/jre7-slim/Dockerfile)*
- `8.0.48-jre7-alpine`, `8.0-jre7-alpine`, `8.0.48-alpine`, `8.0-alpine` *(8.0/jre7-alpine/Dockerfile)*
- `8.0.48-jre8`, `8.0-jre8` *(8.0/jre8/Dockerfile)*
- `8.0.48-jre8-slim`, `8.0-jre8-slim` *(8.0/jre8-slim/Dockerfile)*
- `8.0.48-jre8-alpine`, `8.0-jre8-alpine` *(8.0/jre8-alpine/Dockerfile)*
- `8.5.24-jre8`, `8.5-jre8`, `8-jre8`, `jre8`, `8.5.24`, `8.5`, `8`, `latest` *(8.5/jre8/Dockerfile)*
- `8.5.24-jre8-slim`, `8.5-jre8-slim`, `8-jre8-slim`, `jre8-slim`, `8.5.24-slim`, `8.5-slim`, `8-slim`, `slim` *(8.5/jre8-slim/Dockerfile)*
- `8.5.24-jre8-alpine`, `8.5-jre8-alpine`, `8-jre8-alpine`, `jre8-alpine`, `8.5.24-alpine`, `8.5-alpine`, `8-alpine`, `alpine` *(8.5/jre8-alpine/Dockerfile)*
- `8.5.24-jre9`, `8.5-jre9`, `8-jre9`, `jre9` *(8.5/jre9/Dockerfile)*
- `8.5.24-jre9-slim`, `8.5-jre9-slim`, `8-jre9-slim`, `jre9-slim` *(8.5/jre9-slim/Dockerfile)*
- `9.0.2-jre8`, `9.0-jre8`, `9-jre8` *(9.0/jre8/Dockerfile)*
- `9.0.2-jre8-slim`, `9.0-jre8-slim`, `9-jre8-slim` *(9.0/jre8-slim/Dockerfile)*
- `9.0.2-jre8-alpine`, `9.0-jre8-alpine`, `9-jre8-alpine` *(9.0/jre8-alpine/Dockerfile)*
- `9.0.2-jre9`, `9.0-jre9`, `9-jre9`, `9.0.2`, `9.0`, `9` *(9.0/jre9/Dockerfile)*
- `9.0.2-jre9-slim`, `9.0-jre9-slim`, `9-jre9-slim`, `9.0.2-slim`, `9.0-slim`, `9-slim` *(9.0/jre9-slim/Dockerfile)*

Fig. 22: https://hub.docker.com/_/tomcat/

8.6.2.1 Short Description

Apache Tomcat is an open source implementation of the Java Servlet and JavaServer Pages technologies

8.6.2.2 What is Apache Tomcat ?

Apache Tomcat (or simply Tomcat) is an open source web server and servlet container developed by the Apache Software Foundation (ASF).

Tomcat implements the Java Servlet and the JavaServer Pages (JSP) specifications from Oracle, and provides a "pure Java" HTTP web server environment for Java code to run in.

In the simplest config Tomcat runs in a single operating system process.

The process runs a Java virtual machine (JVM).

Every single HTTP request from a browser to Tomcat is processed in the Tomcat process in a separate thread.

8.6.3 Images webserver : serveurs Web + reverse proxy + load balancer

See also:

- https://en.wikipedia.org/wiki/Reverse_proxy
- https://fr.wikipedia.org/wiki/Proxy_inverse
- https://en.wikipedia.org/wiki/Load_balancer
- https://en.wikipedia.org/wiki/HTTP_cache

See also:

- https://en.wikipedia.org/wiki/Web_server

8.6.3.1 Apache HTTP Server + mod_proxy

Apache HTTP Server, serveur HTTP libre configurable en *proxy inverse* avec le module mod_proxy.

8.6.3.2 Nginx

8.6.3.2.1 Images nginx (engine-x)

See also:

- https://hub.docker.com/_/nginx/
- https://en.wikipedia.org/wiki/Nginx

Contents

Fig. 23: Le logo Nginx

8.6.3.2.1.1 Short Description

Official build of Nginx.

8.6.3.2.1.2 What is nginx ?

See also:

- https://en.wikipedia.org/wiki/Web_server
- https://en.wikipedia.org/wiki/Reverse_proxy
- https://en.wikipedia.org/wiki/Load_balancer
- https://en.wikipedia.org/wiki/HTTP_cache

Nginx (pronounced “engine-x”) is an:

- open source reverse proxy server for HTTP, HTTPS, SMTP, POP3, and IMAP protocols,
- as well as a load balancer, HTTP cache,
- and a **web server** (origin server).

The nginx project started with a strong focus on high concurrency, high performance and low memory usage.

It is licensed under the 2-clause BSD-like license and it runs on Linux, BSD variants, Mac OS X, Solaris, AIX, HP-UX, as well as on other nix flavors.

It also has a proof of concept port for Microsoft Windows.

A large fraction of web servers use NGINX often as a load balancer.

8.7 Images db : bases de données

8.7.1 Images PostgreSQL

See also:

- https://store.docker.com/images/postgres
- https://hub.docker.com/_/postgres/
- https://fr.wikipedia.org/wiki/PostgreSQL
- https://en.wikipedia.org/wiki/PostgreSQL

Contents

Fig. 24: Le logo PostgreSQL

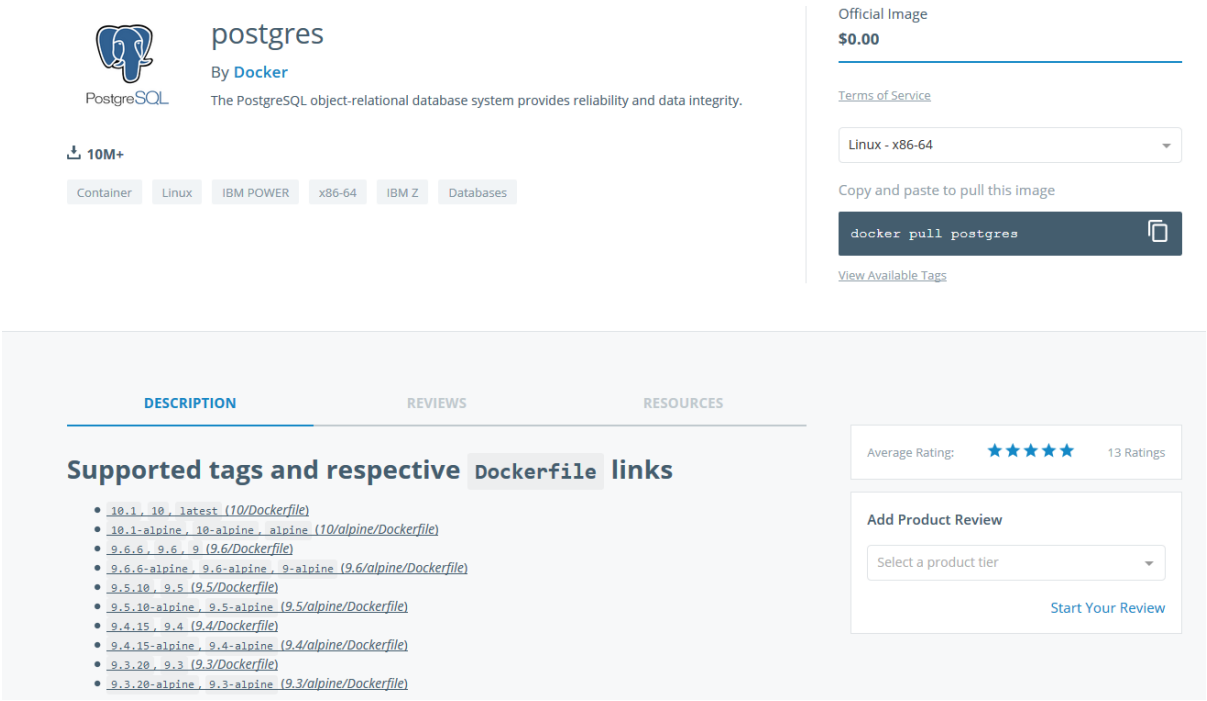


Fig. 25: https://store.docker.com/images/postgres

8.7.1.1 Short Description

The PostgreSQL object-relational database system provides reliability and data integrity.

8.7.1.2 Description

PostgreSQL est un système de gestion de base de données relationnelle et objet (SGBDRO). C'est un outil libre disponible selon les termes d'une licence de type BSD.

Ce système est concurrent d'autres systèmes de gestion de base de données, qu'ils soient libres (comme MariaDB, MySQL et Firebird), ou propriétaires (comme Oracle, Sybase, DB2, Informix et Microsoft SQL Server).

Comme les projets libres Apache et Linux, PostgreSQL n'est pas contrôlé par une seule entreprise, mais est fondé sur une communauté mondiale de développeurs et d'entreprises

8.7.1.3 What is PostgreSQL ?

See also:

- https://en.wikipedia.org/wiki/PostgreSQL

PostgreSQL, often simply "Postgres", is an object-relational database management system (ORDBMS) with an emphasis on extensibility and standards-compliance.

As a database server, its primary function is to store data, securely and supporting best practices, and retrieve it later, as requested by other software applications, be it those on the same computer or those running on another computer across a network (including the Internet).

It can handle workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users.

Recent versions also provide replication of the database itself for security and scalability.

PostgreSQL implements the majority of the SQL:2011 standard, is ACID-compliant and transactional (including most DDL statements) avoiding locking issues using multiversion concurrency control (MVCC), provides immunity to dirty reads and full serializability; handles complex SQL queries using many indexing methods that are not available in other databases; has updateable views and materialized views, triggers, foreign keys; supports functions and stored procedures, and other expandability, and has a large number of extensions written by third parties. In addition to the possibility of working with the major proprietary and open source databases, PostgreSQL supports migration from them, by its extensive standard SQL support and available migration tools. And if proprietary extensions had been used, by its extensibility that can emulate many through some built-in and third-party open source compatibility extensions, such as for Oracle.

8.7.1.4 Environment Variables

The PostgreSQL image uses several environment variables which are easy to miss. While none of the variables are required, they may significantly aid you in using the image.

8.7.1.4.1 POSTGRES_PASSWORD

This environment variable is recommended for you to use the PostgreSQL image. This environment variable sets the superuser password for PostgreSQL. The default superuser is defined by the POSTGRES_USER environment variable. In the above example, it is being set to "mysecretpassword".

Note 1: The PostgreSQL image sets up trust authentication locally so you may notice a password is not required when connecting from localhost (inside the same container). However, a password will be required if connecting from a different host/container.

Note 2: This variable defines the superuser password in the PostgreSQL instance, as set by the initdb script during inital container startup. It has no effect on the PGPASSWORD environment variable that may be used by the psql client at runtime, as described at

https://www.postgresql.org/docs/10/static/libpq-envars.html. PGPASSWORD, if used, will be specified as a separate environment variable.

8.7.1.4.2 POSTGRES_USER

This optional environment variable is used in conjunction with POSTGRES_PASSWORD to set a user and its password. This variable will create the specified user with superuser power and a database with the same name. If it is not specified, then the default user of postgres will be used.

8.7.1.4.3 PGDATA

This optional environment variable can be used to define another location - like a subdirectory - for the database files. The default is /var/lib/postgresql/data, but if the data volume you're using is a fs mountpoint (like with GCE persistent disks), Postgres initdb recommends a subdirectory (for example /var/lib/postgresql/data/pgdata) be created to contain the data.

8.7.1.4.4 POSTGRES_DB

This optional environment variable can be used to define a different name for the default database that is created when the image is first started. If it is not specified, then the value of POSTGRES_USER will be used. POSTGRES_INITDB_ARGS

This optional environment variable can be used to send arguments to postgres initdb. The value is a space separated string of arguments as postgres initdb would expect them. This is useful for adding functionality like data page checksums: -e POSTGRES_INITDB_ARGS="–data-checksums".

8.7.1.4.5 POSTGRES_INITDB_WALDIR

This optional environment variable can be used to define another location for the Postgres transaction log. By default the transaction log is stored in a subdirectory of the main Postgres data folder (PGDATA). Sometimes it can be desireable to store the transaction log in a different directory which may be backed by storage with different performance or reliability characteristics.

Note: on PostgreSQL 9.x, this variable is POSTGRES_INITDB_XLOGDIR (reflecting the changed name of the –xlogdir flag to –waldir in PostgreSQL 10+).

8.7.1.5 Docker Secrets

As an alternative to passing sensitive information via environment variables, _FILE may be appended to the previously listed environment variables, causing the initialization script to load the values for those variables from files present in the container. In particular, this can be used to load passwords from Docker secrets stored in /run/secrets/<secret_name> files. For example:

```
$ docker run --name some-postgres -e POSTGRES_PASSWORD_FILE=/run/secrets/postgres-
↪passwd -d postgres
```

Currently, this is only supported for POSTGRES_INITDB_ARGS, POSTGRES_PASSWORD, POSTGRES_USER, and POSTGRES_DB

8.7.1.6 How to extend this image

If you would like to do additional initialization in an image derived from this one, add one or more *.sql, .sql.gz, or .sh scripts under /docker-entrypoint-initdb.d (creating the directory if necessary).

After the entrypoint calls initdb to create the default postgres user and database, it will run any .sql files and source any .sh scripts found in that directory to do further initialization before starting the service.

For example, to add an additional user and database, add the following to /docker-entrypoint-initdb.d/init-user-db.sh:

```
#!/bin/bash
set -e

psql -v ON_ERROR_STOP=1 --username "$POSTGRES_USER" <<-EOSQL
        CREATE USER docker;
        CREATE DATABASE docker;
        GRANT ALL PRIVILEGES ON DATABASE docker TO docker;
EOSQL
```

These initialization files will be executed in sorted name order as defined by the current locale, which defaults to **en_US.utf8**.

Any .sql files will be executed by POSTGRES_USER, which defaults to the postgres superuser.

It is recommended that any psql commands that are run inside of a .sh script be executed as POSTGRES_USER by using the –username "$POSTGRES_USER" flag. This user will be able to connect without a password due to the presence of trust authentication for Unix socket connections made inside the container.

Additionally, as of docker-library/postgres#253, these initialization scripts are run as the postgres user (or as the "semi-arbitrary user" specified with the –user flag to docker run; see the section titled "Arbitrary –user Notes" for more details).

8.7.1.6.1 Extends with a Dockerfile

You can also **extend the image with a simple Dockerfile** to set a different locale. The following example will set the default locale to de_DE.utf8:

```
FROM postgres:9.4
RUN localedef -i de_DE -c -f UTF-8 -A /usr/share/locale/locale.alias de_DE.UTF-8
ENV LANG de_DE.utf8
```

Since database initialization only happens on container startup, this allows us to set the language before it is created.

8.7.1.7 docker-compose up

```
FROM postgres:10.1
RUN localedef -i fr_FR -c -f UTF-8 -A /usr/share/locale/locale.alias fr_FR.UTF-8
ENV LANG fr_FR.utf8
```

```
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\postgresql> docker-compose up
```

```
WARNING: The Docker Engine you're using is running in swarm mode.

Compose does not use swarm mode to deploy services to multiple nodes in a swarm.␣
↪All containers will be scheduled on the current node.

To deploy your application across the swarm, use `docker stack deploy`.

Building db
Step 1/3 : FROM postgres:10.1
10.1: Pulling from library/postgres
723254a2c089: Pull complete
39ec0e6c372c: Pull complete
ba1542fb91f3: Pull complete
c7195e642388: Pull complete
95424deca6a2: Pull complete
```

(continues on next page)

(continued from previous page)

```
2d7d4b3a4ce2: Pull complete
fbde41d4a8cc: Pull complete
880120b92add: Pull complete
9a217c784089: Pull complete
d581543fe8e7: Pull complete
e5eff8940bb0: Pull complete
462d60a56b09: Pull complete
135fa6b9c139: Pull complete
Digest: sha256:3f4441460029e12905a5d447a3549ae2ac13323d045391b0cb0cf8b48ea17463
Status: Downloaded newer image for postgres:10.1
 ---> ec61d13c8566
Step 2/3 : RUN localedef -i fr_FR -c -f UTF-8 -A /usr/share/locale/locale.alias fr_
↪FR.UTF-8
 ---> Running in 18aa6161e381
Removing intermediate container 18aa6161e381
 ---> a20322020edd
Step 3/3 : ENV LANG fr_FR.utf8
 ---> Running in 0245352c15af
Removing intermediate container 0245352c15af
 ---> b738f47d14a3
Successfully built b738f47d14a3
Successfully tagged postgres:10.1
WARNING: Image for service db was built because it did not already exist. To␣
↪rebuild this image you must use `docker-compose build` or `docker-compose up --
↪build`.
Creating container_intranet ... done
Attaching to container_intranet
container_intranet | 2018-01-31 12:09:54.628 UTC [1] LOG:  listening on IPv4␣
↪address "0.0.0.0", port 5432
container_intranet | 2018-01-31 12:09:54.628 UTC [1] LOG:  listening on IPv6␣
↪address "::", port 5432
container_intranet | 2018-01-31 12:09:54.839 UTC [1] LOG:  listening on Unix␣
↪socket "/var/run/postgresql/.s.PGSQL.5432"
container_intranet | 2018-01-31 12:09:55.034 UTC [20] LOG:  database system was␣
↪shut down at 2018-01-31 12:03:16 UTC
container_intranet | 2018-01-31 12:09:55.135 UTC [1] LOG:  database system is␣
↪ready to accept connections
```

```
PS Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_
↪docker\tutoriels\postgresql> docker exec -ti dda260532cd7 bash
```

```
root@dda260532cd7:/# psql -U postgres
psql (10.1)
Saisissez « help » pour l'aide.

postgres=# \l
                                                                 Liste des bases␣
↪de données
   Nom    | Propriétaire | Encodage | Collationnement | Type caract. |    ␣
↪Droits d'accès
-----------+--------------+----------+-----------------+--------------+------------
↪-----------
 postgres  | postgres     | UTF8     | en_US.utf8      | en_US.utf8   |
 template0 | postgres     | UTF8     | en_US.utf8      | en_US.utf8   | =c/
↪postgres          +
           |              |          |                 |              |␣
↪postgres=CTc/postgres
 template1 | postgres     | UTF8     | en_US.utf8      | en_US.utf8   | =c/
↪postgres          +
           |              |          |                 |              |␣
↪postgres=CTc/postgres
```

(continues on next page)

(continued from previous page)

```
(3 lignes)
```

8.7.2 Images MariaDB

See also:

- https://store.docker.com/images/mariadb
- https://en.wikipedia.org/wiki/MariaDB
- https://fr.wikipedia.org/wiki/MariaDB

Contents

Fig. 26: Le logo MariaDB

Fig. 27: https://store.docker.com/images/mariadb

8.7.2.1 Short Description

MariaDB is a community-developed fork of MySQL intended to remain free under the GNU GPL

8.7.2.2 What is MariaDB ?

MariaDB is a community-developed fork of the MySQL relational database management system intended to remain free under the GNU GPL.

Being a fork of a leading open source software system, it is notable for being led by the original developers of MySQL, who forked it due to concerns over its acquisition by Oracle.

Contributors are required to share their copyright with the MariaDB Foundation.

The intent is also to maintain high compatibility with MySQL, ensuring a "drop-in" replacement capability with library binary equivalency and exact matching with MySQL APIs and commands. It includes the XtraDB storage engine for replacing InnoDB, as well as a new storage engine, Aria, that intends to be both a transactional and non-transactional engine perhaps even included in future versions of MySQL.

8.7.2.3 How to use this image

Start a mariadb server instance

Starting a MariaDB instance is simple:

```
$ docker run --name some-mariadb -e MYSQL_ROOT_PASSWORD=my-secret-pw -d mariadb:tag
```

where some-mariadb is the name you want to assign to your container, my-secret-pw is the password to be set for the MySQL root user and tag is the tag specifying the MySQL version you want.

See the list above for relevant tags. Connect to MySQL from an application in another Docker container

Since MariaDB is intended as a drop-in replacement for MySQL, it can be used with many applications.

This image exposes the standard MySQL port (3306), so container linking makes the MySQL instance available to other application containers. Start your application container like this in order to link it to the MySQL container:

```
$ docker run --name some-app --link some-mariadb:mysql -d application-that-uses-
↪mysql
```

8.7.3 Docker sybase

See also:

- https://github.com/cbsan/docker-sybase
- https://github.com/search?utf8=%E2%9C%93&q=docker+sybase&type=

Contents

8.8 Images outils collaboratifs

8.8.1 Images Gitlab community edition

See also:

- https://store.docker.com/images/gitlab-community-edition
- https://about.gitlab.com/features/

Contents

Fig. 28: Le logo Gitlab

8.8.1.1 Short Description

GitLab includes Git repository management, issue tracking, code review, an IDE, activity streams, wikis, and more.

Open source collaboration and source control management: code, test, and deploy together! More details on features can be found on https://about.gitlab.com/features/

8.8.2 Images Redmine

See also:

- https://store.docker.com/images/redmine

Contents

Fig. 29: Le logo redmine

8.8.2.1 Short Description

Redmine is a flexible project management web application written using Ruby on Rails framework.

8.8.3 Images Wordpress

See also:

- https://store.docker.com/images/wordpress

Contents

Fig. 30: Le logo redmine

8.8.3.1 Short Description

The WordPress rich content management system can utilize plugins, widgets, and themes.

8.9 Images “documentation”

8.9.1 Images MiKTeX

See also:

- https://store.docker.com/community/images/miktex/miktex
- https://github.com/MiKTeX/docker-miktex
- https://en.wikipedia.org/wiki/MiKTeX

Contents

8.9.1.1 Short Description

MiKTeX is a distribution of the TeX/LaTeX typesetting system for Microsoft Windows. It also contains a set of related programs.

MiKTeX provides the tools necessary to prepare documents using the TeX/LaTeX markup language, as well as a simple tex editor: TeXworks.

The name comes from Christian Schenk’s login: MiK for Micro-Kid.

8.10 Images outils scientifiques

8.10.1 Images Anaconda3

See also:

- https://docs.anaconda.com/anaconda/user-guide/tasks/integration/docker
- https://hub.docker.com/r/continuumio/
- https://hub.docker.com/r/continuumio/anaconda3/
- https://github.com/ContinuumIO/docker-images
- https://docs.anaconda.com/anaconda/glossary

Contents

Fig. 31: Le logo Continuumio

8.10.1.1 Short Description

Powerful and flexible python distribution.

8.10.1.2 Usage

You can download and run this image using the following commands:

```
C:\Tmp>docker pull continuumio/anaconda3
```

```
    Using default tag: latest
    latest: Pulling from continuumio/anaconda3
    85b1f47fba49: Pull complete
    f4070d96116d: Pull complete
    8b1142e4866d: Pull complete
```

(continues on next page)

(continued from previous page)

```
    924a14505c9a: Pull complete
    Digest: sha256:c6fb10532fe2efac2f61bd4941896b917ad7b7f197bda9bddd3943aee434d281
Status: Downloaded newer image for continuumio/anaconda3:latest
```

```
C:\Tmp>docker run -i -t continuumio/anaconda3 /bin/bash
```

```
root@8ffcde2f70f6:/# uname -a
```

```
Linux 8ffcde2f70f6 4.9.60-linuxkit-aufs #1 SMP Mon Nov 6 16:00:12 UTC 2017 x86_64␣
↪GNU/Linux
```

```
root@8ffcde2f70f6:/# which python
```

```
/opt/conda/bin/python
```

```
root@8ffcde2f70f6:/# python
```

```
Python 3.6.3 |Anaconda, Inc.| (default, Oct 13 2017, 12:02:49)
[GCC 7.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

8.11 Images apprentissage

8.11.1 Image dockersamples/static-site

See also:

- https://hub.docker.com/r/dockersamples/static-site/

Contents

8.11.2 Image hello world

See also:

- https://hub.docker.com/_/hello-world/
- https://store.docker.com/images/hello-world
- https://github.com/docker/labs/blob/master/beginner/chapters/setup.md

Contents

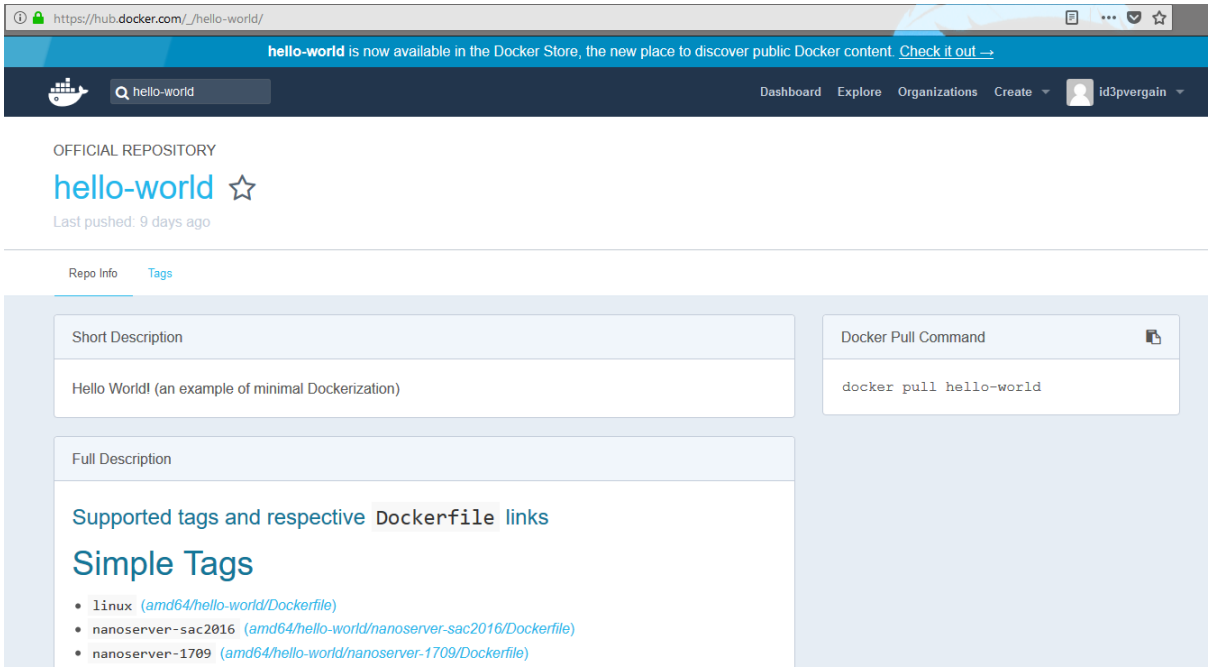


Fig. 32: https://hub.docker.com/_/hello-world/

8.11.2.1 Short Description

Hello World! (an example of minimal Dockerization).

Chapter 9

Docker commands

Contents

9.1 docker build

Chapter 10

Docker network

See also:

- https://github.com/vrde/notes/tree/master/docker-playground
- https://github.com/docker/labs/tree/master/networking

Contents

10.1 Las networking

See also:

- https://github.com/docker/labs/tree/master/networking

Chapter 11

Volumes Docker

See also:

- https://docs.docker.com/engine/admin/volumes/volumes/
- http://www.lemagit.fr/conseil/Docker-quelles-sont-les-options-pour-le-stockage-persistant
- http://xataz.developpez.com/tutoriels/utilisation-docker/

Contents

11.1 Use volumes

Estimated reading time: 12 minutes

Volumes are the preferred mechanism for persisting data generated by and used by Docker containers.

While bind mounts are dependent on the directory structure of the host machine, volumes are completely managed by Docker.

Volumes have several advantages over bind mounts:

- Volumes are easier to back up or migrate than bind mounts.
- You can manage volumes using Docker CLI commands or the Docker API.
- Volumes work on both Linux and Windows containers.
- Volumes can be more safely shared among multiple containers.
- Volume drivers allow you to store volumes on remote hosts or cloud providers, to encrypt the contents of volumes, or to add other functionality.
- A new volume's contents can be pre-populated by a container.

In addition, volumes are often a better choice than persisting data in a container's writable layer, because using a volume does not increase the size of containers using it, and the volume's contents exist outside the lifecycle of a given container.

11.2 Create and manage volumes

Unlike a bind mount, you can create and manage volumes outside the scope of any container.

11.2.1 docker volume create

Create a volume:

```
docker volume create my-vol
```

11.2.2 docker volume ls

```
Y:\projects_id3\P5N001\XLOGCA135_tutorial_docker\tutorial_docker\tutoriels\pipenv>
↪docker volume ls
```

```
DRIVER              VOLUME NAME
local               ␣
↪03f1a1ed0555015e51863dbed5f6c7847099fd33449b9d83919bd7028cdfdd9b
local               ␣
↪4bc5fb631c6af81f5ba84a8465b3c2805ca713541fe736faf3a232ef4b24ae72
local               ␣
↪56295a3bb8a90d260864c258e6b174755338543a614f206e5082c066d22eb197
local               ␣
↪67871ba2f3b3a9e75fdbfcf2fe5ec36ba7a10cd5930a60e8227abc7110e62ca4
local               ␣
↪b6432532ff915143ede0b7169abf7690790ce0227277013d7c5ab00007d68703
local               ␣
↪bbef076d429a90ca7bfd7a751d7c8aa1ea3d08e0b7b4036bb296681545940a0b
local               ␣
↪bf69b1f1164c09d7dc0f3e6011f3116e7bc197e0e9341e645a15fdc7566489f3
local               ␣
↪cee0d9feda75150bda5e6b32c5eeaad4e433afe01165bf822eae8413b1f4e861
local               pgdata
local               postgresql_postgres_data
local               vote_db-data
```

Chapter 12

Révisions

Auteurs	Révision	Date	Description
Patrick Vergain	*0.1.0*	2018-01-12	Création du document CA135

12.1 Version 0.1.0 (2018-01-12) : création du projet

On commence l'écriture du tutoriel Docker. Ecriture du chapitre *introduction*.

Chapter 13

Glossaire Docker

See also:

- https://docs.anaconda.com/anaconda/glossary

Agile Software Development A set of concepts, practices and principles for the development of software under which both requirements and the software that meets them evolve during the development life-cycle by processes of collaboration, as opposed to being defined at milestones within it

Containers Running instances of Docker images — containers run the actual applications. A container includes an application and all of its dependencies. It shares the kernel with other containers, and runs as an isolated process in user space on the host OS. You created a container using docker run which you did using the alpine image that you downloaded. A list of running containers can be seen using the docker ps command.

Docker

Définition 1 (anglais) Docker allows us to easily create clean, pre-installed images of our application in an isolated state, like a binary application build, rather than having to worry about virtual environments and system packages of whatever server we are deploying to. This build can then be tested and deployed as if it was an isolated artifact in and of itself.

Source: https://peakwinter.net/blog/modern-devops-django/

Définition 2 With Docker, you can run your Django project on an Ubuntu server in a container on your laptop, and because Docker is available for Mac, Linux, and Windows, your choice of operating system really comes down to preference. When it comes time to push your code to a staging or production server, you can be sure it'll run exactly the same as it did on your laptop, because you can configure a Dockerfileto exactly match these environments.

Source: https://medium.com/@adamzeitcode/a-simple-recipe-for-django-development-in-docker-bonus-testing-with-s

Docker daemon The background service running on the host that manages building, running and distributing Docker containers.

Docker client The command line tool that allows the user to interact with the Docker daemon.

docker-compose.yml

Definition 1 (français) Le docker compose est un fichier de configuration de l'ensemble des Dockers que vous souhaitez déployer pour votre application, il sert à les déployer et à gérer les liens entre les conteneurs ainsi que les volumes de data.

Definition 2 (anglais) The file where you can set up your database, automatically start your server when you start your container, and cool stuff like that.

Source: https://www.revsys.com/tidbits/brief-intro-docker-djangonauts/

Définition 3 (anglais) Docker Compose lets you run more than one container in a Docker application. It's especially useful if you want to have a database, like Postgres, running in a container alongside your

web app. (Docker's overview of Compose is helpful.) Compose allows you to define several services that will make up your app and run them all together.

Source: https://www.revsys.com/tidbits/brief-intro-docker-djangonauts/

Dockerfile

Definition 1 (français) C'est le fichier texte qui décrit la configuration de votre docker, en général , on part d'une image standard et on ajoute les éléments propres à la configuration de l'application que l'on veut déployer ; une fois le Dockerfile finalisé, on *build* le conteneur.

Definition 2 (anglais) The name of the file that contains the instructions for setting up your image. Source: https://www.revsys.com/tidbits/brief-intro-docker-djangonauts/

Docker image

Definition 1 (français) C'est l'élément de base d'un docker, on utilise une Docker image à deux stades :

- Au départ, on va chercher une image de base standard pour l'applicatif choisi (Nginx, Php, Redis), le plus souvent dans un repository public, on complète ensuite cette image standard des éléments de configuration de votre application, vous pouvez ensuite enregistrer la nouvelle image dans un repository public, ou privé,

Definition 2 (anglais) The file system and configuration of our application which are used to create containers. To find out more about a Docker image, run:

```
docker inspect alpine
```

In the demo above, you used the docker pull command to download the alpine image. When you executed the command docker run hello-world, it also did a docker pull behind the scenes to download the hello-world image.

Definition 3 (anglais) A **lightweight, stand-alone, executable package that includes everything needed to run a piece of software.** You will set up a specific image for each project you work on that will tell Docker which packages your project needs, where your code lives, etc.

Source: https://www.revsys.com/tidbits/brief-intro-docker-djangonauts/

Docker Store A registry of Docker images, where you can find trusted and enterprise ready containers, plugins, and Docker editions. You'll be using this later in this tutorial.

hyperviseur

Hyperviseur En informatique, un hyperviseur est une plate-forme de virtualisation qui permet à plusieurs systèmes d'exploitation de travailler sur une même machine physique en même temps.

Hyper-V Microsoft Hyper-V, codenamed Viridian and formerly known as Windows Server Virtualization, is a native hypervisor; it can create virtual machines on x86-64 systems running Windows.

Hyper-V, également connu sous le nom de Windows Server Virtualisation, est un système de virtualisation basé sur un hyperviseur 64 bits de la version de Windows Server 2008.

Orchestrateur de conteneurs L'orchestrateur est un peu au conteneur ce que vSphere/vCenter est à VMware pour des VMs, c'est le logiciel de gestion de l'ensemble des conteneurs sur un pool de ressources serveurs, avec davantage de fonctionnalités que vSphere[24]/vCenter. C'est en quelque sorte un PaaS[25] pour les conteneurs

[24] https://fr.wikipedia.org/wiki/VMware_vSphere
[25] https://fr.wikipedia.org/wiki/Plate-forme_en_tant_que_service

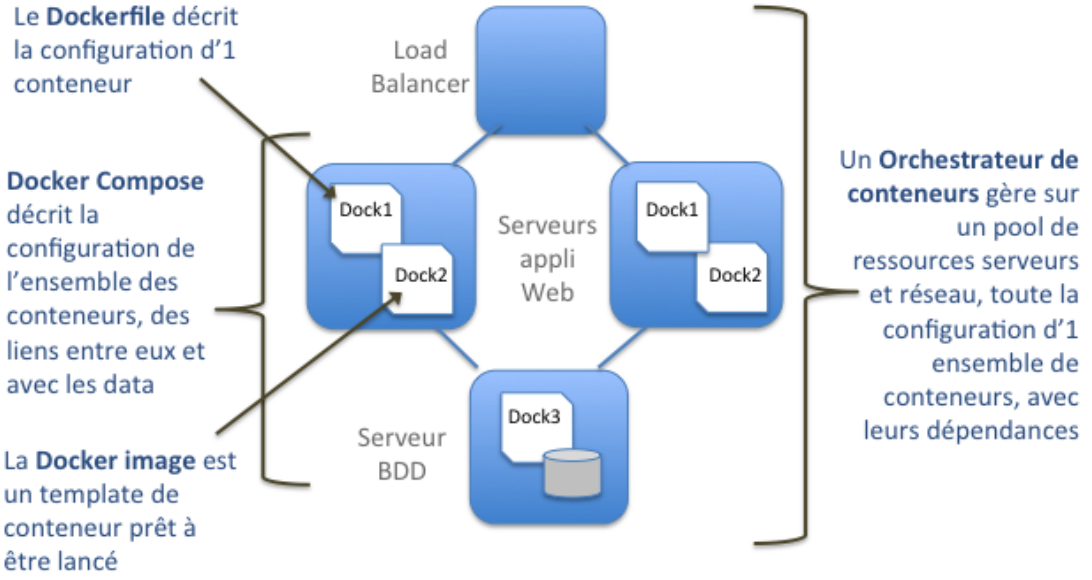


Fig. 1: Un orchestrateur gère un pool de ressources serveurs et réseau .. seealso:: https://actu.alfa-safety.fr/devops/docker-en-production/

reverse proxy

proxy inverse Un proxy inverse (reverse proxy) est un type de serveur, habituellement placé en frontal de serveurs web. Contrairement au serveur proxy qui permet à un utilisateur d'accéder au réseau Internet, le proxy inverse permet à un utilisateur d'Internet d'accéder à des serveurs internes, une des applications courantes du proxy inverse est la répartition de charge (load-balancing).

Le proxy inverse est installé du côté des serveurs Internet. L'utilisateur du Web passe par son intermédiaire pour accéder aux applications de serveurs internes. Le proxy inverse est parfois appelé substitut (surrogate)

See also:

https://fr.wikipedia.org/wiki/Proxy_inverse

Essaim

Swarm

swarm A swarm is a cluster of one or more Docker Engines running in swarm mode. En français *swarm* est un essaim

Fig. 2: Un essaim de docker engines

Virtual machine Have you ever seen someone boot up Windows on a Mac ? That process of running one complete OS on top of another OS called running a **virtual machine**.

See also:

http://erick.matsen.org/2018/04/19/docker.html

Chapter 14

Hyper-V (Viridian, Windows Server Virtualisation)

See also:

- https://fr.wikipedia.org/wiki/Hyper-V

14.1 Définition

14.1.1 En français

See also:

- https://fr.wikipedia.org/wiki/Hyper-V

Hyper-V, également connu sous le nom de Windows Server Virtualisation, est un système de virtualisation basé sur un hyperviseur 64 bits de la version de Windows Server 2008.

Il permet à un serveur physique de devenir Hyperviseur et ainsi gérer et héberger des machines virtuelles communément appelées VM (virtual machines).

Grâce à cette technologie il est possible d'exécuter virtuellement plusieurs systèmes d'exploitation sur une même machine physique et ainsi d'isoler ces systèmes d'exploitation les uns des autres.

14.1.2 En anglais

See also:

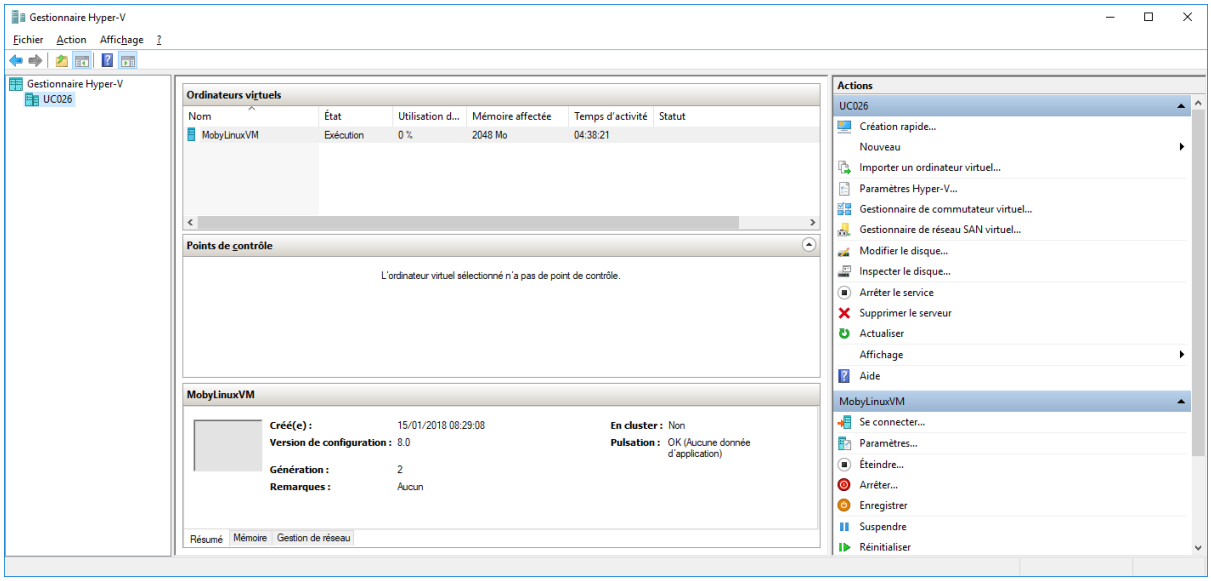
- https://en.wikipedia.org/wiki/Hyper-V

Microsoft Hyper-V, codenamed Viridian and formerly known as Windows Server Virtualization, is a native hypervisor; it can create virtual machines on x86-64 systems running Windows.

Starting with Windows 8, Hyper-V superseded Windows Virtual PC as the hardware virtualization component of the client editions of Windows NT. A server computer running Hyper-V can be configured to expose individual virtual machines to one or more networks.

Hyper-V was first released alongside Windows Server 2008, and has been available without charge for all the Windows Server and some client operating systems since.

14.2 Le gestionnaire Hyper-V




```
%windir%\System32\mmc.exe "%windir%\System32\virtmgmt.msc"
```

Chapter 15

Hébergeurs Docker

Contents

15.1 Amazon

See also:

- http://www.journaldunet.com/solutions/cloud-computing/1205896-comment-aws-supporte-t-il-vraiment-docker/

Chapter 16

Docker documentation

16.1 Docker aquasec documentation

See also:

- https://www.aquasec.com/wiki

16.1.1 About this Site

This website brings together thousands of online resources about container technology.

Containers are nothing new: as early as 1982 Unix administrators could launch isolated processes, similar to today's containers, using the chroot command.

The first modern container was probably Linux-VServer released in 2001.

Containers matured considerably in the 12 years that followed, until the rise of Docker which finally took containers to the mainstream.

Today cloud computing, deployment, DevOps and agile development are almost synonymous with containers. So much has been written on this complex subject, and few have attempted to organize this corpus into a meaningful format.

At Aqua Security, a pioneer in container security, we took upon ourselves to fill this gap and collect the most important writings about container technology - from conceptual articles and best practices to vendor information and how to guides - to help the growing community make sense of the space. The end result will include over 200 sub-topics around containers, container platforms, container orchestration and more. With a special focus on Docker and Kubernetes which are becoming ubiquitous in modern container setups.

Chapter 17

Docker videos

See also:

- https://docs.docker.com/docker-for-windows/

17.1 2018

See also:

- https://www.youtube.com/watch?v=YFl2mCHdv24

Index

Symbols

Numbers

A

B

C

D

E

G

H

I

J

U

V

W

X