

h e g

Haute école de gestion
Genève

Docker®

Travail de Bachelor réalisé en vue de l'obtention du Bachelor HES

par :

Karim LAGHZAoui

Conseiller au travail de Bachelor :

Peter DAEHNE, Professeur HES

Genève, le 6 mars 2017

Haute École de Gestion de Genève (HEG-GE)

Filière Informatique de gestion

Déclaration

Ce travail de Bachelor est réalisé dans le cadre de l'examen final de la Haute école de gestion de Genève, en vue de l'obtention du titre Bachelor of Science HES-SO en Informatique de gestion.

L'étudiant atteste que son travail a été vérifié par un logiciel de détection de plagiat.

L'étudiant accepte, le cas échéant, la clause de confidentialité. L'utilisation des conclusions et recommandations formulées dans le travail de Bachelor, sans préjuger de leur valeur, n'engage ni la responsabilité de l'auteur, ni celle du conseiller au travail de Bachelor, du juré et de la HEG.

« J'atteste avoir réalisé seul le présent travail, sans avoir utilisé des sources autres que celles citées dans la bibliographie. »

Fait à Genève, le 6 mars 2017

Karim Laghzaoui

Remerciements

Je tiens tout d'abord à remercier M. Peter Daehne pour sa disponibilité et ses précieux conseils qui m'ont permis d'aborder les sujets traités dans ce travail de Bachelor de façon optimale tant au niveau de la recherche que de la rédaction.

Par ailleurs, je remercie mes parents qui m'ont toujours soutenu durant mes études ainsi que pour ce mémoire.

Résumé

Depuis des dizaines d'années, les avancées technologiques ont eu une influence certaine sur l'augmentation exponentielle de l'utilisation des outils informatiques tant par les entreprises que par les particuliers.

Pour en arriver là, plusieurs étapes majeures ont marqué de leur empreinte l'histoire récente de l'informatique. Certaines sont très connues du grand public comme l'Internet, d'autres le sont moins comme la virtualisation.

Bien qu'il soit moins connu, le domaine de la virtualisation a pris une place très importante pour les entreprises dotées d'une infrastructure informatique conséquente.

Les intérêts liés à son utilisation vont de la réduction des coûts informatiques à l'optimisation du développement d'applications en passant par la sécurisation d'un système d'exploitation physique protégé par un système d'exploitation virtuel.

Dans le marché de la virtualisation, l'entreprise VMWare® est leader depuis plus d'une quinzaine d'années en mettant en œuvre une virtualisation qui utilise le concept de machines virtuelles.

Depuis presque cinq ans, une nouvelle société s'est introduite dans le marché de la virtualisation. Docker Inc® ne s'est pas contenté de reprendre le concept de machines virtuelles utilisé par un grand nombre d'entreprises du marché de la virtualisation.

En effet, elle a apporté une nouvelle façon de concevoir la virtualisation avec un système plus léger et plus simple d'utilisation que les machines virtuelles : le « conteneur ».

L'objectif de ce travail de Bachelor, est d'analyser les deux façons de virtualiser évoquées ci-dessus afin de définir quels sont leurs avantages, leurs inconvénients et dans quels cas elles peuvent être utilisées.

Table des matières

Déclaration.....	i
Remerciements	ii
Résumé	iii
Liste des tableaux	vi
Liste des figures.....	vi
1. Introduction.....	1
2. La virtualisation	4
2.1 Concept	4
2.2 Historique	5
3. Cloud computing	7
3.1 Définition	7
3.2 Historique	8
3.3 Propriétés fondamentales	10
3.3.1 Libre-service à la demande	10
3.3.2 Accès au réseau depuis partout	10
3.3.3 Mutualisation des ressources	10
3.3.4 Élasticité	10
3.3.5 Service mesurable	10
3.4 Notion de « service »	11
3.4.1 SaaS : Software as a Service.....	11
3.4.2 PaaS : Platform as a Service	11
3.4.3 IaaS : Infrastructure as a Service	12
3.4.4 CaaS : Containers as a Service	12
3.5 Modèles du cloud computing.....	13
3.5.1 Cloud public	13
3.5.2 Cloud privé.....	13
3.5.3 Cloud communautaire	13
3.5.4 Cloud hybride.....	13
4. Marché de la virtualisation.....	14
4.1 VMware®.....	14
4.2 Microsoft®	15
4.3 Citrix®	16
4.4 Oracle®	16
5. Avantages liés à la virtualisation	17
5.1 Améliorer le déploiement	17
5.2 Optimiser la portabilité	18
5.3 Utiliser des logiciels prévus pour divers environnements.....	18

5.4 Réduire les coûts	18
6. Virtualisation en pratique.....	20
6.1 Serveurs.....	20
6.2 Postes de travail.....	20
6.3 Applications	21
6.4 Réseau	21
7. Docker®	22
7.1 Présentation	22
7.2 Historique	23
7.3 Partenariat avec Microsoft®	25
7.4 Docker® et Linux™	26
7.5 Fonctionnement	27
7.5.1 Docker Engine™	29
7.5.2 Conteneur	30
7.5.3 Docker® image	32
7.5.4 Docker® registry	35
7.5.5 Libcontainer	36
7.5.6 Namespaces	37
7.5.7 CGroups	37
8. Comparaison des solutions de virtualisation	38
8.1 Conteneurs VS machines virtuelles.....	38
8.1.1 Analogie avec le monde réel	40
8.1.2 Caractéristiques	41
8.2 Cas pratiques d'utilisation.....	42
9. Conclusion	43
Bibliographie	44

Liste des tableaux

Tableau 1 : Comparaison entre les machines virtuelles et les conteneurs.....	41
--	----

Liste des figures

Figure 1 : Évolution du nombre d'utilisateurs de l'Internet dans le monde	1
Figure 2 : Évolution du trafic de données à partir d'appareils fixes et mobiles	2
Figure 3 : Architecture de Docker®	28
Figure 4 : Docker Engine™	29
Figure 5 : Conteneur Docker®	30
Figure 6 : Cycle de vie d'un conteneur	31
Figure 7 : Exemple d'images officielles disponibles dans Docker Hub™	35
Figure 8 : Architecture d'un conteneur Docker®	36
Figure 9 : Hyperviseurs de type 1 et 2	39
Figure 10 : Machines virtuelles et conteneurs.....	40

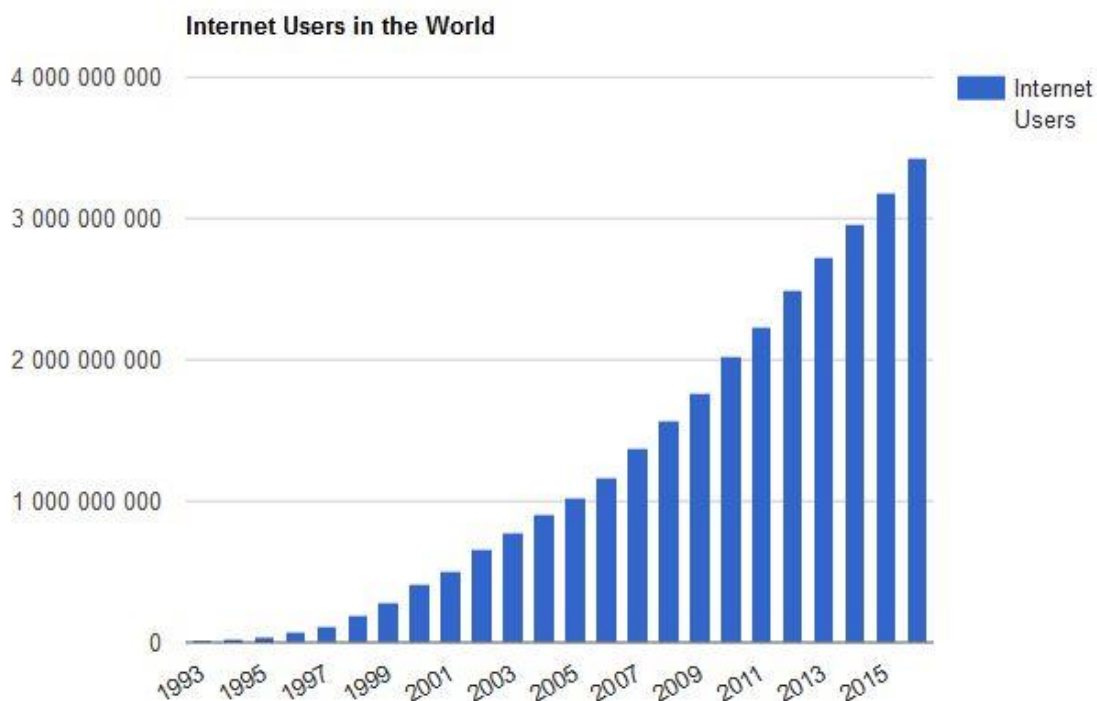
1. Introduction

L'informatique s'est considérablement développée les vingt dernières années, jusqu'à devenir quasiment indispensable pour chacun d'entre nous et davantage encore pour les entreprises.

Parmi les nouvelles technologies qui ont un impact important aujourd'hui, il y a notamment les ordinateurs personnels, les téléphones portables, les tablettes tactiles et les objets connectés. C'est une fois qu'ils sont connectés à l'Internet que l'utilisation de ces appareils devient réellement intéressante.

Le graphique ci-dessous (cf. Figure 1 : Évolution du nombre d'utilisateurs de l'Internet dans le monde) nous permet de constater que le nombre d'utilisateurs connectés à l'Internet a augmenté très rapidement à partir du début des années 2000.

Figure 1 : Évolution du nombre d'utilisateurs de l'Internet dans le monde



(<http://www.internetlivestats.com/internet-users/>)

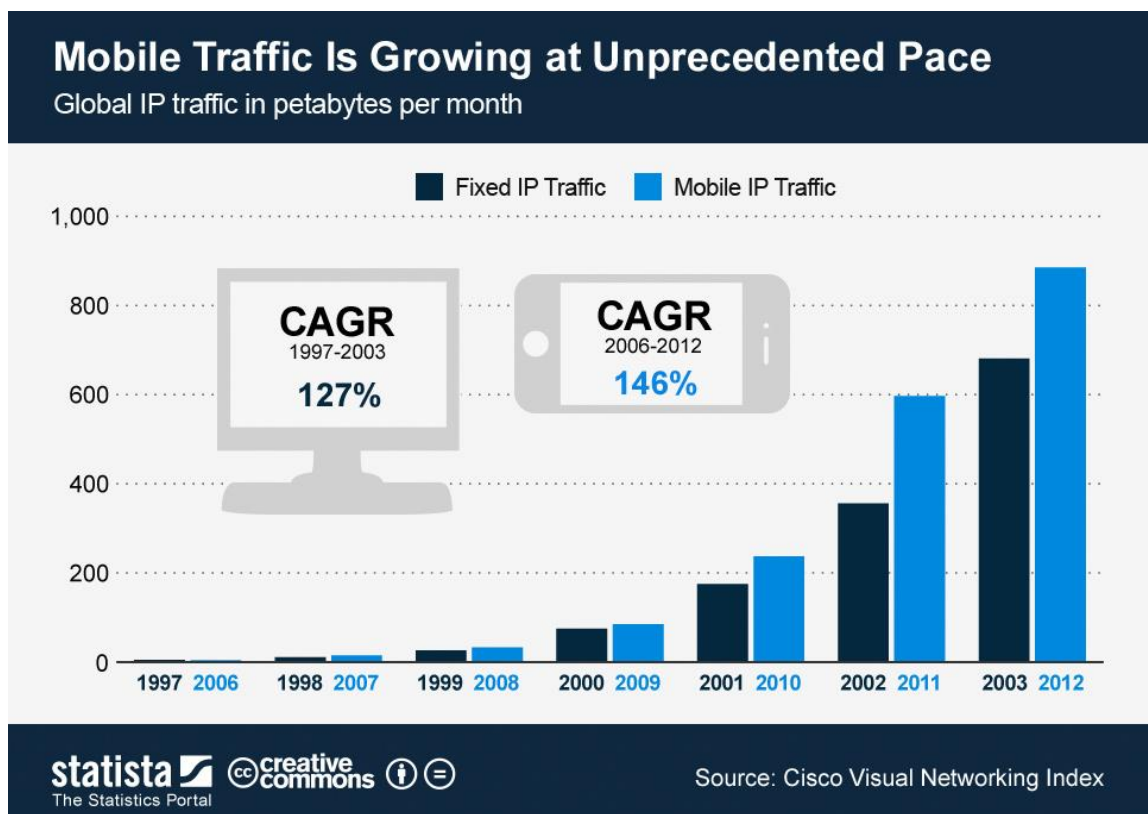
En parallèle à cette importante augmentation du nombre d'utilisateurs connectés à l'Internet, il faut également considérer la sortie des premiers smartphones et la généralisation des ordinateurs portables ainsi que des tablettes tactiles.

Désormais, tout ce qui pouvait être réalisé avec un ordinateur fixe et encombrant peut être effectué depuis n'importe quel lieu grâce à la démocratisation des appareils mobiles.

Cette formidable avancée technologique ainsi que l'accroissement important du nombre d'appareils mobiles connecté à l'Internet dans le monde ont généré une augmentation exponentielle du trafic de données.

Le graphique ci-dessous (cf. Figure 2 : Évolution du trafic de données à partir d'appareils fixes et mobiles) illustre clairement que les appareils mobiles ont généré plus de trafic de données en 6 ans (entre 2006 et 2012) que les ordinateurs fixes dans le même laps de temps (entre 1997 et 2003).

Figure 2 : Évolution du trafic de données à partir d'appareils fixes et mobiles



[\(https://www.statista.com/chart/1009/mobile-internet-traffic-growth/\)](https://www.statista.com/chart/1009/mobile-internet-traffic-growth/)

Par conséquent, il a fallu trouver des solutions pour stocker la masse gigantesque de données présente sur Internet et concevoir de nouvelles façons d'accéder à ces données de façon simple.

Depuis plusieurs années, la virtualisation (cf. [2. La virtualisation](#)) et le cloud computing (cf. [3. Cloud computing](#)) font partie des solutions pour stocker de grandes quantités de données et les rendre disponibles pour un client indépendamment de sa localisation et de l'appareil qu'il utilise.

Par ailleurs, beaucoup d'avantages sont liés à la mise en œuvre de la virtualisation, parmi lesquelles on peut noter la réduction des coûts d'une infrastructure technique et l'utilisation de logiciels prévus pour des environnements différents par exemple (cf. [5. Avantages liés à la virtualisation](#)).

Ensuite, nous allons étudier plusieurs types de virtualisation différents tels que la virtualisation des serveurs ou des applications (cf. [6. Virtualisation en pratique](#)).

Dans ce mémoire, les différentes notions évoquées précédemment sont étudiées dans le but d'analyser un nouveau venu dans le domaine de la virtualisation.

Il s'agit de Docker® (cf. [7. Docker®](#)), un logiciel de virtualisation différent des autres de par son fonctionnement (cf. [7.5 Fonctionnement](#)) basé sur des conteneurs (cf. [7.5.2 Conteneur](#)); il se démarque clairement de la virtualisation « classique » basée sur des machines virtuelles.

Pour terminer, nous ferons une analyse comparative de deux façons de faire de la virtualisation (machines virtuelles et conteneurs) afin de déterminer leurs caractéristiques et dans quels cas chacune d'entre elles peut être mise en œuvre (cf. [8. Comparaison des solutions de virtualisation](#)).

2. La virtualisation

2.1 Concept

Tout d'abord, il est nécessaire de définir la virtualisation étant donné que c'est une notion fondamentale de ce travail.

La virtualisation permet d'utiliser une ressource informatique virtuelle à partir d'une machine physique réelle.

Par conséquent, une seule machine physique peut exécuter plusieurs ressources informatiques virtuelles différentes.

Les ressources physiques tels qu'un processeur, une interface réseau ou un disque dur sont allouées à une machine virtuelle pour que celle-ci fonctionne comme une machine physique.

Aujourd'hui, pour qu'un ordinateur puisse héberger plusieurs machines virtuelles de façon optimale, il est nécessaire qu'il soit composé de ressources matérielles suffisamment puissantes.

En résumé, la virtualisation consiste à utiliser les technologies de l'information et de la communication, matérielles et logicielles, dans le but d'héberger plusieurs systèmes d'exploitation différents sur une unique machine physique.

2.2 Historique

C'est dans les années soixante qu'apparaît la première forme de virtualisation, à un moment où le matériel informatique est encore très peu présent dans les entreprises.

A l'époque, la virtualisation concerne les mainframes, qui sont des ordinateurs puissants permettant d'effectuer un grand nombre de traitements.

Au vu de la grande capacité de calcul de ces grosses machines, il était important d'optimiser l'utilisation de celles-ci afin d'éviter qu'elles ne travaillent inutilement.

C'est d'ailleurs un problème qui est toujours d'actualité aujourd'hui avec nos serveurs modernes.

Dans le but de partitionner ses mainframes afin qu'ils puissent réaliser plusieurs tâches en même temps, la première entreprise ayant développé la virtualisation est IBM®.

La société a développé l'IBM System/360 model 67 qui permettait le fonctionnement du temps partagé qui n'a pas réellement abouti.

Cependant, IBM® a lancé un nouveau système qui utilisait la virtualisation pour mettre en œuvre le temps partagé et observer l'interaction entre les systèmes d'exploitation et le matériel informatique.

Ce système apparu en 1967 est le CP/CMS (Control Program/Cambridge Monitor System puis Console Monitor System).

La particularité qui le caractérise est que son code source était libre, par conséquent l'ensemble des clients d'IBM® pouvait y accéder gratuitement.

Le terme CP (Control Program) correspondait à un programme de contrôle permettant de créer l'environnement d'une machine virtuelle. CP permettait à un utilisateur de gérer une machine virtuelle depuis un terminal.

Le CMS (Console Monitor System) était défini comme un simple système d'exploitation interactif mono-utilisateur.

La qualité de ce système CP/CMS était perceptible car, à cette époque, il était déjà possible d'exécuter plusieurs copies de CMS simultanément sur les machines virtuelles CP sans impacter les performances.

Le système CP/CMS a largement évolué durant cette période et était donc une innovation importante à l'époque.

En effet, c'est un système qui a beaucoup inspiré la virtualisation telle qu'on la connaît de nos jours.

L'environnement était grandement virtualisé avec CP/CMS, toute machine virtuelle était dotée de ses propres périphériques virtuels sur la base du type de matériel physique du système.

En résumé, il y a eu plusieurs versions de CP/CMS. La première a été CP-40/CMS, c'est celle qui a permis de mettre en œuvre l'architecture de la machine virtuelle CP/CMS.

Par la suite, la CP-67/CMS a été développée pour l'IBM System/360 model 67. Enfin, la CP-370/CMS a été développée pour l'IBM System/370 mais n'a jamais été mise sur le marché.

Cependant, c'est à partir de celle-ci que le système d'exploitation VM/370 a été commercialisé en 1972.

Dans la version « moderne » de la virtualisation, c'est l'hyperviseur qui a pris le rôle du CP d'IBM®. L'entreprise qui a introduit et dévoilé au grand public le concept de virtualisation tel qu'il existe de nos jours est VMWare®.

C'est en 1999 que cette nouvelle société sort VMWare Workstation™ 1.0. Dès l'an 2000, IBM®, Dell® et Compaq® rejoignent VMWare® en tant que partenaires pour publier en 2001 VMWare GSX Server™ 1.0 et ESX Server™ 1.0.

VMWare® se développe rapidement et atteint le nombre d'un million d'utilisateurs en 2002.

Le produit VMotion™, lancé en 2003, est considéré comme étant une avancée importante dans le domaine car il permet de transférer des machines virtuelles depuis une machine physique vers une autre.

Intel® et Cisco® deviennent actionnaires de la société en 2007, après son introduction en bourse.

Au-delà des intérêts financiers que ces sociétés ont en commun, cela permettra de mettre en place des partenariats basés sur la technologie dans le futur.

3. Cloud computing

3.1 Définition

Dans le but d'avoir une définition complète du cloud computing, nous utiliserons la définition datant de septembre 2011 du NIST (National Institute of Standards and Technology) :

« Le cloud computing est un modèle qui permet l'ubiquité, l'accès à un réseau à la demande d'un pool partagé de ressources informatiques configurables (par exemple : réseaux, serveurs, stockage, applications, services), qui peuvent être rapidement fournis et publiés avec un effort minime de la gestion des interactions de la part du fournisseur de service. »

En d'autres termes, le cloud computing ou informatique en nuage permet d'accéder à des ressources informatiques ou d'exécuter des applications hébergées sur des serveurs depuis un client (ordinateur, tablette, smartphone, etc) sans devoir gérer l'infrastructure informatique permettant de mettre en œuvre ce système.

Par ailleurs, il est intéressant de constater le lien entre le cloud computing et la virtualisation des ressources dans différents domaines qui ne concernent pas nécessairement l'informatique.

Par exemple, nous pouvons notamment noter que les films et la musique sont de plus en plus consommés via des plateformes de streaming comme avec Netflix® ou Spotify®.

De plus, l'utilisation de véhicules est souvent optimisée, comme avec Mobility® et Autolib'™ qui permettent d'utiliser une voiture et de payer uniquement pour l'utilisation qu'on en fait sans devoir payer son entretien.

Il y a aussi des services de covoiturage comme BlaBlaCar® avec lesquels nous pouvons effectuer de longs trajets à moindre coût en voyageant à plusieurs.

3.2 Historique

Bien que son concept ait plus de 50 ans, le cloud computing tel qu'il existe de nos jours s'est réellement développé au début des années 2000 en parallèle au développement de l'Internet.

En effet, entre la fin des années 1990 et le début des années 2000, les ordinateurs personnels et l'Internet sont de plus en plus présents dans les entreprises et également dans les foyers.

C'est donc durant cette période, que Salesforce[®], un des pionniers du cloud computing, est lancé par Marc Benioff, Parker Harris, Dave Moellenhoff et Frank Dominguez en 1999. Le site internet de l'entreprise, salesforce.com, a été le premier à fournir l'accès à des applications depuis un navigateur web.

En 2002, Amazon Web Services^{™1} est lancé. C'est la première version de ce service que la société Amazon[®] met sur le marché.

Amazon Web Services[™] permettra aux utilisateurs de stocker des données en ligne et avec Amazon Mechanical Turk^{™2} de diviser des tâches en sous-tâches pour qu'elles soient effectuées via cette plateforme par une personne rémunérée. Amazon Mechanical Turk[™] est une des premières formes de travail en ligne.

Par la suite, Amazon's Elastic Compute Cloud^{™3} (EC2) est lancé en 2006. Ce service permet aux utilisateurs d'exécuter leurs applications sur des serveurs qu'ils louent à Amazon.

De plus, le client peut créer des machines virtuelles et paie par rapport au temps d'utilisation du service. C'est à ceci que fait référence le terme « Elastic » composant le nom de ce service.

Le prochain service développé par Amazon sera nommé Amazon S3^{™4}. Le terme « S3 » correspond aux trois mots suivants commençant par la lettre « s » :

- Simple
- Storage
- Service

¹ <https://aws.amazon.com/fr/>

² <https://www.mturk.com/mturk/welcome>

³ <https://aws.amazon.com/fr/ec2/>

⁴ https://aws.amazon.com/fr/s3/?nc1=h_ls

Comme son nom l'indique, le service permet le stockage de données informatiques à grande échelle.

D'ailleurs, de grandes entreprises utilisent ce service aujourd'hui. Parmi les plus connues nous retrouvons notamment, Dropbox[®], Netflix[®] et Tumblr[®].

La société Salesforce[®] lance en 2007 force.com[™]. Force.com^{™5} est un PaaS (cf. [3.4.2 PaaS : Platform as a Service](#)) fait pour que les développeurs puissent créer, héberger et exécuter des logiciels.

C'est à la même période que Google développe considérablement son offre en matière de cloud computing. En effet, en 2006, Google[®] met sur le marché Google Apps for Your Domain[®] (appelé G Suite^{®6} aujourd'hui).

Aujourd'hui, G Suite[®] est composé de différents produits utilisés par des millions de personnes dans le monde. Parmi les plus connus, il y a notamment Gmail^{®7}, Google Drive^{®8}, Google Docs^{®9}.

⁵ <https://www.salesforce.com/products/platform/products/force/>

⁶ <https://gsuite.google.com/index.html>

⁷ <https://www.google.com/intl/fr/mail/help/about.html>

⁸ https://www.google.com/intl/fr_ALL/drive/

⁹ https://www.google.com/intl/fr_ch/docs/about/

3.3 Propriétés fondamentales

Le cloud computing est une technologie en constante évolution. Elle est basée sur cinq caractéristiques principales qui sont décrites dans ce chapitre.

3.3.1 Libre-service à la demande

Les capacités de calcul comme le temps serveur et le stockage réseau sont accessibles de façon unilatérale par un utilisateur en fonction de ses besoins sans devoir interagir avec un être humain pour chacun des services fournis.

3.3.2 Accès au réseau depuis partout

L'utilisation de plateformes hétérogènes de client léger ou lourd tels que des ordinateurs, des tablettes ou des smartphones est permise par la mise en place de la disponibilité et de l'accessibilité sur le réseau des capacités de traitement par des mécanismes standard.

3.3.3 Mutualisation des ressources

Les ressources informatiques du fournisseur sont mises en commun afin de garantir leur accès à plusieurs utilisateurs en utilisant un modèle multi-entité, avec différentes ressources physiques et virtuelles affectées dynamiquement selon la demande de l'utilisateur.

En règle générale, l'utilisateur n'a pas connaissance et n'a pas la possibilité de contrôler l'emplacement des ressources fournies mais il peut définir l'emplacement à un niveau d'abstraction plus élevé (pays, région, data center).

3.3.4 Élasticité

Les capacités du cloud peuvent être augmentées ou diminuées automatiquement pour que les ressources soient affectées en fonction des besoins du système.

Pour le client, les capacités disponibles pour l'approvisionnement semblent être illimitées et peuvent être allouées en quantité suffisante à tout moment.

3.3.5 Service mesurable

Les systèmes de cloud computing gèrent et optimisent automatiquement l'utilisation des ressources en exploitant une capacité de comptage à un niveau d'abstraction adapté à un type de service (stockage, traitement, compte utilisateur, bande passante).

L'utilisation des ressources peut être surveillée en garantissant la transparence pour le prestataire de service ainsi que pour le client.

3.4 Notion de « service »

Cette partie est consacrée aux principaux modèles de services existant dans le domaine du cloud computing.

Les trois principaux types qui sont concernés sont le Software as a Service (SaaS), le Platform as a Service (PaaS) et l'Infrastructure as a Service (IaaS).

Par ailleurs, un quatrième modèle de service sera abordé, c'est le CaaS (Containers as a Service). Ce dernier est le plus récent et il est particulièrement important car c'est avec celui-ci que fonctionne le logiciel Docker® qui fait l'objet de ce travail de Bachelor.

3.4.1 SaaS : Software as a Service

L'utilisation d'un logiciel en tant que service permet à un client d'y accéder par Internet via un serveur sans que le client ait besoin de l'installer localement sur son ordinateur.

Ce système est souvent accessible par un navigateur web sous forme de client léger. En général, il faut payer un abonnement pour utiliser le logiciel plutôt qu'acheter une licence comme dans la plupart des cas où le logiciel est installé sur l'ordinateur du client.

Ce type de service est considéré comme le plus complet car c'est tant le matériel que le logiciel qui sont vendus au client.

La partie émergée de l'iceberg est le logiciel utilisé concrètement par le client mais il y a aussi une partie immergée qui correspond à toute l'infrastructure technique qui n'est pas directement utilisée par le client mais qui lui permet d'exécuter et d'utiliser la partie logicielle fournie.

3.4.2 PaaS : Platform as a Service

Ce type de service est particulièrement intéressant dans le cadre de ce travail car il fournit notamment des machines virtuelles, des systèmes d'exploitation ou encore des applications.

En fournissant un environnement de développement, il permet aux utilisateurs de développer des applications et de les intégrer dans le cloud sans devoir gérer la plateforme qu'ils utilisent.

En effet, c'est le rôle du fournisseur de maintenir la plateforme sur laquelle travaille le client. Le client peut donc se concentrer uniquement sur son travail sans s'occuper de la partie technique qui lui permet de l'effectuer.

3.4.3 IaaS : Infrastructure as a Service

L'infrastructure en tant que service est une forme de cloud computing dans laquelle les ressources matérielles sont virtualisées.

En effet, en utilisant ce type de service, le client paie uniquement pour les infrastructures techniques dont il a besoin au moment où il en a besoin mais c'est le fournisseur qui est le propriétaire de l'infrastructure.

Parmi les différents éléments techniques qui peuvent être utilisés par le client, il y a notamment les serveurs, les disques durs et le matériel lié au réseau.

3.4.4 CaaS : Containers as a Service

L'utilisation de conteneurs est le type de service sur lequel est basé le logiciel Docker®. Dans la pratique, les conteneurs se basent sur le système IaaS décrit précédemment tout en développant d'autres façons de faire, décrites dans le chapitre sur Docker® (cf. [7.5.2 Conteneur](#)), pour améliorer le service.

3.5 Modèles du cloud computing

Il existe plusieurs modèles de déploiement du cloud computing. Nous allons ici les lister et les définir pour comprendre à quel type d'utilisation chacun d'entre eux correspond.

3.5.1 Cloud public

Le modèle du cloud public permet au grand public de l'utiliser librement. Le propriétaire de l'infrastructure est le fournisseur de service. Les plus connus sont les géants du web comme Google®, Microsoft® et Apple®.

3.5.2 Cloud privé

Une infrastructure cloud privé est faite pour être utilisée par une seule organisation. La gestion d'un cloud privé peut être effectuée en interne par l'organisation qui l'utilise ou par un acteur externe à l'organisation.

3.5.3 Cloud communautaire

Dans le cas du cloud computing communautaire, les utilisateurs sont considérés comme un groupe spécifique formé de personnes pouvant provenir d'organisations différentes mais ayant des intérêts communs tels que des exigences de sécurité, de politique ou de conformité.

Tout comme dans le cloud privé, la gestion peut être confiée à la charge d'une organisation ou d'un prestataire de service.

Par ailleurs, la spécificité de la gestion d'un cloud communautaire est qu'elle peut également être faite par plusieurs organisations qui partagent ce cloud.

3.5.4 Cloud hybride

Un cloud hybride est composé de deux ou plus des différents modèles du cloud computing décrit ci-dessus.

Bien qu'il y ait différents types de cloud computing, chacun d'entre eux reste une entité à part entière tout en étant lié les uns aux autres par des technologies propriétaires ou normalisées qui permettent la portabilité des applications et des données.

4. Marché de la virtualisation

La virtualisation a pris davantage d'importance durant les 15 dernières années depuis l'émergence de VMWare® sur le marché.

Afin de pouvoir mesurer l'ampleur qu'a pris la virtualisation, le thème que nous allons aborder à présent concerne le marché actuel de la virtualisation.

4.1 VMware®

En sortant le logiciel vSphere™ 6.0 en février 2015, VMWare® consolide son statut de leader du marché de la virtualisation.

VMWare® est l'entreprise qui a la plus grande part de marché dans ce domaine et continue d'innover au niveau des services et fonctionnalités qu'elle propose.

Néanmoins, selon Gartner®¹⁰ ce sont les fournisseurs de services de cloud computing qui représentent environ 15% du déploiement des machines virtuelles de nos jours. En 2011, ce chiffre était de 3% seulement.

Les entreprises du domaine du cloud computing fournissant du cloud de type IaaS se sont considérablement développées pour atteindre 15% du déploiement des machines virtuelles en quelques années seulement.

C'est la société Amazon®, avec sa division Amazon Web Services™ qui est en grande partie actrice de ce récent phénomène qui a pris de l'ampleur de façon très rapide.

Malgré cela, VMWare® reste le leader du domaine de la virtualisation, que ce soit par son nombre de clients ou au niveau des investissements importants effectués dans la recherche et le développement.

¹⁰ <https://www.gartner.com/doc/reprints?ct=160707&id=1-3B9FAM0&st=sb>

4.2 Microsoft®

Le géant de l'informatique qu'est Microsoft® n'est pas le leader du marché de la virtualisation même si avec son système de virtualisation Hyper-V™, il est tout de même un acteur important du secteur.

C'est avec Windows Server™ 2012 R2, sorti en 2013 que Microsoft® a pu s'installer solidement et augmenter sa part de marché.

Bien que les prix restent plus compétitifs chez Microsoft®, par rapport à VMWare®, les clients constatent plusieurs problèmes importants pour Microsoft® Hyper-V™ :

- Durée d'arrêt planifié plus longue à cause de la taille du système d'exploitation parent.
- Complexité opérationnelle ainsi que problèmes de basculement avec haute disponibilité.
- Manque d'alternative concernant la gestion de la disponibilité contrairement à VMWare® avec son Site Recovery Manager™¹¹.

Le dernier né des systèmes d'exploitation orienté serveur de Microsoft® est Windows Server™ 2016, il est sorti en octobre 2016, il n'a donc pas encore eu le temps de s'installer dans le monde de la virtualisation.

Microsoft est constamment en train de tenter de récupérer des clients de VMWare® avec difficulté étant donné que ceux-ci sont en grande partie satisfaits des services du leader du marché de la virtualisation.

En outre, mis à part VMWare® et les autres entreprises du domaine, Microsoft® va devoir affronter une concurrence nouvelle avec les produits open source et les fournisseurs de cloud computing.

Cependant, Microsoft compte un avantage de taille qui pourra lui permettre de se développer dans les années à venir en ce qui concerne le marché de la virtualisation : un grand nombre d'entreprises travaillent uniquement avec des produits Microsoft® et il y a beaucoup de systèmes d'exploitation Windows Server™ installés.

¹¹ <http://www.vmware.com/fr/products/site-recovery-manager.html>

4.3 Citrix®

La société Citrix® compte deux produits très utilisés dans la virtualisation des postes de travail (cf. [6.2 Postes de travail](#)), XenApp®¹² et XenDesktop™¹³. XenApp® permet la virtualisation d'applications et l'accès à distance de celles-ci.

En ce qui concerne XenDesktop™, il offre la possibilité d'intégrer un système de virtualisation de postes de travail en plus des fonctionnalités proposées par XenApp®.

Pour la virtualisation de serveurs, Citrix® propose le produit Citrix XenServer™¹⁴, qui est une plateforme de gestion de la virtualisation.

Dans le but de développer ses activités également dans le domaine du cloud computing, Citrix a conclu un accord avec la société Accelerite™ au début de l'année 2016 afin d'acquérir les produits CloudPlatform™¹⁵ et CloudPortal™¹⁶.

4.4 Oracle®

La société Oracle® propose diverses technologies permettant de virtualiser. Les deux principales sont la virtualisation avec un hyperviseur et celle par conteneur (cf. [7.5.2 Conteneur](#)).

Oracle VM™¹⁷, le serveur de virtualisation de la société Oracle®, est muni d'un ensemble d'outils permettant de gérer l'ensemble des éléments logiciels et matériels développés par la société Oracle® appelé Oracle Enterprise Manager™¹⁸.

Avec ce produit, les administrateurs peuvent gérer la totalité de l'environnement de virtualisation, de l'infrastructure informatique physique, jusqu'aux applications déployées sur les machines.

Parmi les avantages qu'offre Oracle® dans le domaine de la virtualisation, il y a l'interopérabilité avec les produits Microsoft®.

Par ailleurs, Oracle® propose également Oracle Solaris Zones™ et Oracle Linux Containers™ qui sont respectivement dédiés aux systèmes d'exploitation Solaris (propriété d'Oracle®) et à Linux™.

¹² <https://www.citrix.fr/downloads/xenapp/>

¹³ <https://www.citrix.fr/downloads/xendesktop/>

¹⁴ <https://www.citrix.fr/products/xenserver/>

¹⁵ https://www.citrix.com/content/dam/citrix/en_us/documents/products-solutions/citrix-cloudplatform-powered-by-apache-cloudstack.pdf

¹⁶ <https://www.citrix.fr/products/cloudportal-services-manager/>

¹⁷ <http://www.oracle.com/technetwork/server-storage/virtualbox/overview/index.html>

¹⁸ <http://www.oracle.com/technetwork/oem/enterprise-manager/overview/index.html>

5. Avantages liés à la virtualisation

Depuis plusieurs années, la notion de virtualisation se développe considérablement jusqu'à devenir un élément très important dans le domaine de l'informatique.

Si la virtualisation a pris une telle place aujourd'hui, c'est parce qu'elle comporte différents avantages qui ne peuvent pas être négligés, notamment par les entreprises qui ont un grand besoin en ressources informatiques.

En effet, la virtualisation est avant tout très intéressante pour faire des économies de différentes sortes (matérielles, énergétiques, financières).

Dans le but de comprendre plus concrètement les avantages liés à la virtualisation, nous allons détailler ci-dessous plusieurs aspects de manière non exhaustive, qui démontrent pour quelles raisons la virtualisation a pris autant d'ampleur de nos jours.

5.1 Améliorer le déploiement

L'intégration d'éléments informatiques matériels ou logiciels et la migration vers de nouveaux systèmes d'exploitation peuvent souvent devenir des sources de problèmes pour les administrateurs systèmes.

En effet, l'installation et la configuration de nouveaux serveurs physiques et systèmes d'exploitation peuvent prendre un certain temps, sans parler de la migration des applications hébergées sur d'anciens serveurs.

Par ailleurs, dans certains cas, une seule application peut bloquer la mise à jour du système d'exploitation d'un serveur et par conséquent, toutes les autres applications sont également condamnées à rester sur le système d'exploitation actuel.

Pour l'exemple expliqué ci-dessus, la virtualisation permettrait que le système ne pouvant pas être mis à jour devienne une machine virtuelle sur laquelle serait hébergée l'application qui fonctionne uniquement sur ce système d'exploitation.

Par contre, les autres applications pourraient être hébergées sur une autre machine virtuelle fonctionnant avec le système d'exploitation mis à jour.

5.2 Optimiser la portabilité

La rapidité de la gestion des machines virtuelles fait partie des avantages de l'utilisation de la virtualisation. Avec la virtualisation, il est possible de déplacer des machines virtuelles d'un serveur physique à un autre de façon simple et rapide.

C'est un élément important lors d'opération de maintenance ou d'indisponibilités imprévues d'une machine physique.

Par ailleurs, les machines virtuelles s'adaptent simplement en fonction des différents types d'infrastructures physiques qui les hébergent.

Dans le cas de la virtualisation sur un ordinateur, les machines virtuelles s'exécutent et fonctionnent indépendamment du type de matériel physique comme les cartes mères ou les différents types de périphériques.

5.3 Utiliser des logiciels prévus pour divers environnements

En virtualisant, la majorité des systèmes d'exploitation sont exécutables sur d'autres systèmes d'exploitation. Ce principe permet à un utilisateur d'avoir accès à des applications prévues à la base pour un grand nombre d'autres systèmes d'exploitation. Cela est réalisable en déployant plusieurs machines virtuelles sur un unique ordinateur physique.

Sur la base de ce principe, les possibilités de travail avec une multitude de logiciels prévus pour différents systèmes d'exploitation sont très grandes et deviennent accessibles à tous sans devoir acquérir du matériel physique dédié à chaque système utilisé.

Pour illustrer cela, prenons le cas d'un utilisateur qui possède un ordinateur fonctionnant sous Windows®. Avec la virtualisation, son ordinateur peut exécuter une machine virtuelle Linux™ et par extension, les logiciels prévus pour ce dernier.

5.4 Réduire les coûts

En faisant le choix de la virtualisation dans leurs infrastructures techniques, les entreprises ne négligent pas les importantes économies qu'elles peuvent potentiellement réaliser.

Tout d'abord, la réduction des coûts est clairement perceptible au niveau du matériel, car avec l'utilisation de machines virtuelles, le besoin d'acquisition de machines physiques a logiquement tendance à diminuer.

Par conséquent, étant donné que les machines virtuelles utilisent les éléments d'une même machine physique et communiquent via ceux-ci, l'achat de matériel réseau tel que les routeurs est aussi considérablement réduit.

Par ailleurs, le fait de posséder moins de machines physiques implique également un gain de place dans les locaux de l'entreprise ainsi que des économies d'énergie importantes, notamment dans les domaines du refroidissement et de l'hébergement.

Grâce à la simplification du déploiement, de la maintenance, de la sauvegarde et d'autres opérations, la virtualisation permet aux administrateurs de diminuer le temps de travail lié à ces sujets afin de participer à d'autres projets.

6. Virtualisation en pratique

Avec l'évolution constante du concept de virtualisation, cette notion s'est fortement développée et ne concerne plus uniquement la virtualisation des serveurs comme auparavant.

Dans ce chapitre, nous allons aborder les principales façons de mettre en œuvre la virtualisation actuellement.

6.1 Serveurs

Selon l'entreprise VMWare®, les serveurs utilisent moins de 15% de leurs capacités¹⁹, la virtualisation est une solution afin d'éviter la sous-utilisation des serveurs.

En effet, la virtualisation des serveurs permet à un serveur physique d'héberger plusieurs serveurs virtuels.

En fonctionnant ainsi, les capacités d'un serveur physique sont davantage utilisées car tous les serveurs virtuels utilisent une partie des ressources du serveur physique.

6.2 Postes de travail

La virtualisation des postes de travail permet à un client d'exécuter une machine virtuelle qui est hébergée sur un serveur distant.

La machine virtuelle peut être exécutée sur différents appareils. Les principaux sont l'ordinateur de bureau, l'ordinateur portable, la tablette ainsi que le smartphone.

En accédant à cette machine virtuelle, le client travaille avec le système d'exploitation virtualisé, les logiciels et les données comme dans le cas d'une utilisation locale.

Les entreprises dotées de la virtualisation de postes de travail voient la gestion de leur parc d'ordinateurs largement simplifiée.

En effet, en utilisant une même image de machine virtuelle pour plusieurs dizaines ou centaines d'ordinateurs, les opérations de maintenance, les mises à jour et autres tâches deviennent plus simples à réaliser.

¹⁹ <http://www.vmware.com/fr/solutions/virtualization.html>

6.3 Applications

La virtualisation d'applications permet de déployer des logiciels de manière indépendante vis-à-vis de la machine hôte et sans devoir modifier les éléments locaux comme le système d'exploitation, le système de fichiers et le registre.

La mise à disposition des applications est facilitée pour les entreprises mettant en œuvre la virtualisation d'applications.

Le déploiement des applications se fait sans installation locale sur les ordinateurs des utilisateurs.

Les applications sont isolées les unes des autres et peuvent être exécutées simultanément sur un ordinateur même si elles ont des problèmes d'incompatibilité.

Étant indépendantes du système d'exploitation local, les applications virtualisées permettent de simplifier la gestion des risques liés à la protection du système.

6.4 Réseau

La virtualisation du réseau gère les serveurs et les éléments du réseau comme un ensemble unique de ressources auxquelles il est possible d'accéder malgré les différents types de composants physiques.

Un réseau virtualisé est basé sur un seul réseau physique qui peut contenir plusieurs réseaux virtuels. Chaque réseau virtuel est considéré comme un ensemble de nœuds et liens virtuels et comme un sous-ensemble de ressources du réseau physique.

7. Docker®

7.1 Présentation

Tout d'abord, il est indispensable de définir les deux significations différentes lors de l'emploi du terme « Docker ».

Le premier des deux termes à distinguer est Docker® qui représente le projet informatique qui est décrit ci-dessous.

Le deuxième de ces termes est Docker Inc® qui correspond à l'entreprise qui est également présentée par la suite et qui développe la plateforme Docker®.

Docker® est une plateforme open source qui permet à des applications d'être déployées à partir de conteneurs.

Le concept de Docker® est d'avoir une couche d'abstraction permettant aux développeurs d'applications d'empaqueter n'importe quelle application et c'est ensuite la technologie de conteneurisation qui gère le déploiement, indépendamment de l'infrastructure physique.

La plateforme Docker® facilite la portabilité et l'intégration d'applications dans différents environnements.

Avec Docker®, un développeur peut configurer une application sur son ordinateur et la transmettre à une autre personne qui pourra la déployer et l'exécuter sur un serveur Docker® sans différence de fonctionnement avec la version du développeur.

7.2 Historique

Le fondateur de l'entreprise qui deviendra Docker Inc[®] est Solomon Hykes, un franco-américain né à New York et ayant fait ses études en France. Solomon Hykes a cofondé la société dotCloud[™] en 2008.

La société est active dans le secteur du cloud computing en tant que fournisseur de PaaS (cf. [3.4.2 Paas : Platform as a Service](#)).

En 2010, dotCloud[™] s'inscrit auprès de l'entreprise américaine Y Combinator^{™20} dans le but de lever des fonds pour développer ses projets.

Le concept proposé par Solomon Hykes est Docker[®], un outil qui utilise la conteneurisation pour faire de la virtualisation.

Y Combinator[™] fait le choix de retenir le projet Docker[®] et permet à Solomon Hykes de le développer avec la somme de 20'000 dollars reçue.

Par la suite, l'équipe ayant développé le projet Docker[®] propose ses services aux autres entreprises inscrites chez Y Combinator[™] afin que leurs applications soient hébergées par la plateforme qu'elle a mise en place.

C'est ainsi que Solomon Hykes a réussi à convaincre et à lever des fonds à hauteur de 800'000 dollars de la part d'investisseurs séduits par le projet.

Au vu de l'ampleur et du développement de l'entreprise de Solomon Hykes, dotCloud[™] s'installe en 2011 dans le lieu incontournable du développement d'applications: la Silicon Valley.

Le projet Docker[®] est sorti en open source au mois de mars 2013 et la technologie innovante proposée a largement intéressé la communauté du domaine.

En octobre 2013, la société dotCloud[™] devient Docker Inc[®] et le projet initial de dotCloud[™] est vendu à la compagnie cloudControl[™] quelques mois plus tard.

Docker Inc[®] parvient à lever 15 millions de dollars en janvier 2014 pour développer le projet open source ainsi que la plateforme communautaire, puis 40 millions de dollars au mois de septembre de la même année.

²⁰ <https://www.ycombinator.com/about/>

Ubuntu™ 14.04 est devenue la première distribution Linux® d'entreprise à être livrée avec Docker®. En juin, la version 1.0 de Docker® est sortie lors de la première conférence organisée par l'entreprise.

Au mois de septembre, une levée de fonds de 40 millions de dollars monte la valeur du projet à 400 millions de dollars selon Docker Inc®²¹.

À la fin de l'année 2014, Docker Inc® a organisé sa première conférence en Europe qui a eu lieu à Amsterdam. Lors de cette conférence, plusieurs nouveaux projets ont été annoncés tel que Docker Hub™ Entreprise™.

C'est à cette période que de grandes entreprises du domaine informatique vont se rapprocher de Docker Inc® et devenir partenaires. Parmi celles-ci, il y a notamment Microsoft®, Amazon®, et IBM®.

Docker Inc® termine l'année 2014 en sortant la version 1.4 de son projet et en devenant le 24^{ème} projet le plus apprécié sur GitHub™.

Durant l'année 2015, Docker Inc® a comptabilisé 1,2 milliard de conteneurs téléchargés avec son projet et a acquis plusieurs starts-up comme Kitematic™ et Tutum™.

En 2016, Docker Inc® annonce un nouveau produit, c'est un service de cloud computing nommé Docker Cloud™ permettant de gérer facilement l'intégration continue d'applications.

²¹ <https://www.bloomberg.com/news/articles/2014-09-16/docker-said-to-be-valued-at-400-million-in-funding-round>

7.3 Partenariat avec Microsoft®

Après avoir commencé un partenariat en 2014 pour développer les liens entre la plateforme Docker® et Windows Server™, Microsoft® et Docker Inc® ont conclu un nouvel accord commercial en septembre 2016.

Ce nouveau partenariat a pour but de permettre aux développeurs Windows et aux entreprises spécialisées dans l'informatique d'étendre les capacités de l'utilisation du projet Docker® avec des systèmes Windows®.

Les éléments compris dans le partenariat concernent la disponibilité de Docker Engine™ pour les clients de Windows Server™ 2016 sans frais supplémentaires et le support de celui-ci.

Cet accord permettra aux entreprises d'accélérer dans l'innovation et la valorisation lors de la création, le déploiement et l'exécution d'applications.

De plus, Docker Inc® et Microsoft® ont décidé de promouvoir Docker Datacenter™²² pour que les sociétés informatiques puissent sécuriser la chaîne d'approvisionnement logicielle et gérer la conteneurisation des produits Windows Server™ physiques ou virtuels.

Avec sa plateforme intégrée, Docker Datacenter™ permet aux développeurs de gérer efficacement le cycle de vie des applications sans devoir négliger la portabilité et l'agilité lors de leur développement.

²² <https://www.docker.com/products/docker-datacenter>

7.4 Docker® et Linux™

Pour comprendre ce qu'est Docker® et comment il fonctionne, il est essentiel de savoir qu'il est directement lié à Linux™, par son noyau appelé Linux Kernel²³ qui contient notamment deux services importants dans l'architecture de Docker®.

Ces services sont nommés « namespaces » (cf. [7.5.6 Namespaces](#)) et « cgroups » (cf. [7.5.7 CGroups](#)).

Linux Kernel™ est un logiciel qui compose une partie fondamentale du système d'exploitation Linux™. Il est responsable de la gestion des ressources d'un ordinateur et permet aux composants matériels et logiciels de celui-ci de communiquer entre eux.

Docker® est basé sur une technologie de virtualisation appelée Linux Containers™ (LXC). Chaque conteneur est considéré comme étant un système d'exploitation isolé.

Lorsqu'un conteneur est en cours d'exécution, il nécessite d'être lié au noyau Linux™ pour être isolé et pouvoir gérer ses ressources.

De cette façon, chaque conteneur sera exécuté sous un même noyau Linux™ tout en possédant un système de fichiers, de la mémoire ou des processus qui lui seront privés.

Par conséquent, tous les conteneurs exécutés sur un même hôte utilisent le même noyau contrairement aux machines virtuelles « classiques ».

Ceci permet d'améliorer les performances du système, notamment lorsqu'il s'agit de démarrer ou d'arrêter un conteneur.

²³ <https://www.kernel.org/doc/html/latest/>

7.5 Fonctionnement

L'architecture de Docker[®] est basée sur une architecture de type client-serveur (cf. Figure 3 : Architecture de Docker[®]). Il y a un client Docker[®] et un Docker[®] daemon qui prend le rôle de serveur.

Le client Docker[®] ne peut pas communiquer directement avec les conteneurs. Pour cela, il communique avec un élément de l'architecture appelé Docker[®] daemon.

Le client Docker[®] peut être installé sur la même machine que le Docker[®] daemon ou sur une machine différente.

Docker[®] daemon est un élément du présent sur le serveur Docker[®] considéré comme un gestionnaire de conteneurs qui fait le lien entre le client, les conteneurs et les images qui composent les conteneurs.

C'est un composant essentiel qui a pour responsabilité d'effectuer toutes les opérations liées à un conteneur comme la création, l'exécution ou l'arrêt de celui-ci.

Le Docker[®] daemon peut être exécuté sur plusieurs serveurs dans une infrastructure et un client peut interagir avec plusieurs serveurs également.

La gestion des communications appartient totalement aux clients, mais les serveurs Docker[®] peuvent interagir directement avec les Docker[®] images si un client le demande.

Les clients sont responsables et doivent gérer les tâches exécutées par les serveurs. Le rôle des serveurs est d'héberger les applications sous forme de conteneurs.

La communication entre le client et le serveur se fait généralement par l'outil de ligne de commande de Docker[®]. C'est un outil qui est fonctionnel avec la majorité des systèmes d'exploitation.

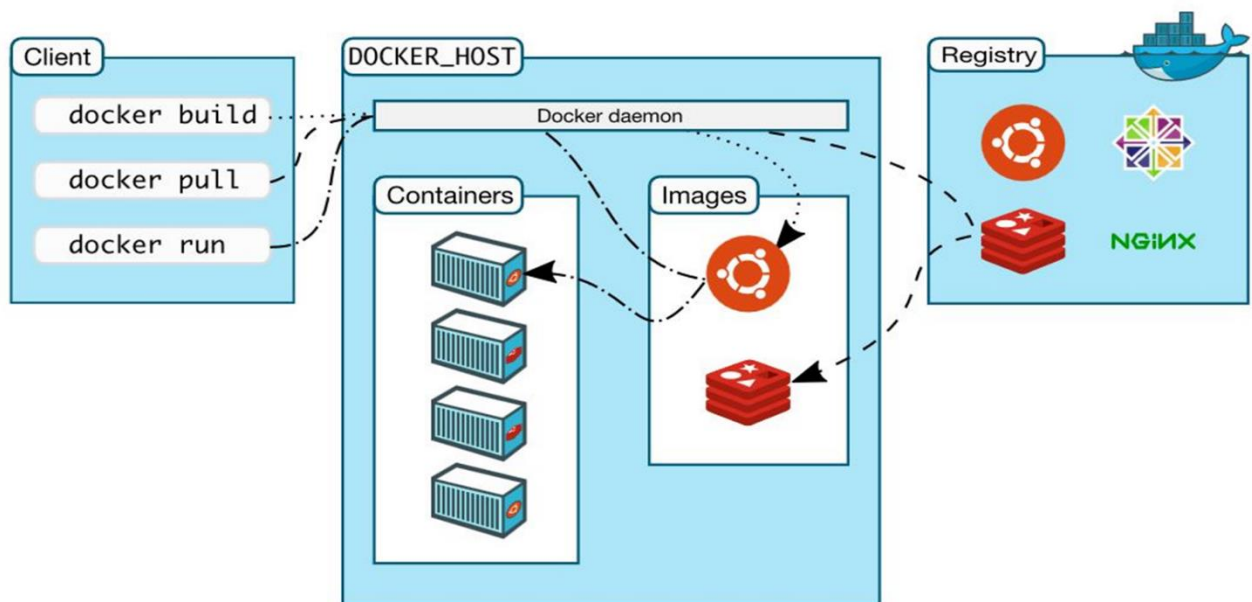
Voici les principales actions qu'il est possible de réaliser avec l'outil de ligne de commande de Docker[®] :

- Récupérer une image depuis un registre et la transférer vers un Docker[®] daemon ou transférer une image vers un registre à partir d'un Docker[®] daemon.
- Créer une image Docker[®]
- Démarrer un conteneur sur un serveur Docker[®]

L'outil de ligne de commande de Docker® n'est pas le seul moyen de communication entre un client et un serveur Docker®.

L'utilisation d'APIs permet aussi de communiquer avec le serveur et de gérer et contrôler l'exécution des applications.

Figure 3 : Architecture de Docker®



[\(https://docs.docker.com/engine/understanding-docker/\)](https://docs.docker.com/engine/understanding-docker/)

7.5.1 Docker Engine™

Le Docker Engine™ est une partie fondamentale de la plateforme Docker®. C'est lui qui permet notamment d'exécuter son propre conteneur, de construire une Docker® image, ou encore de partager ses Docker® images avec d'autres via Docker Hub™.

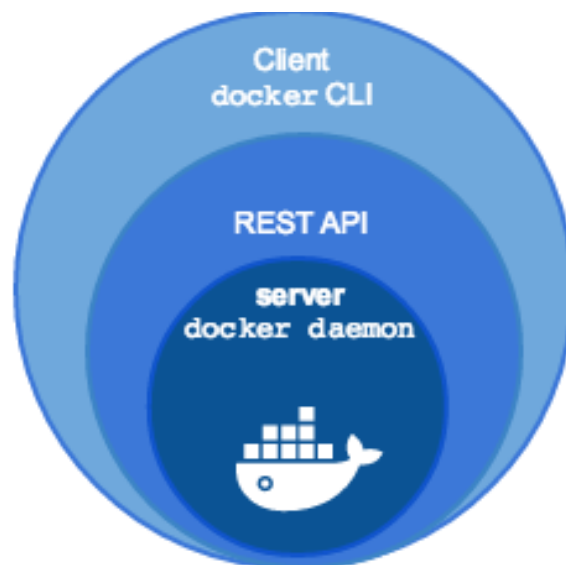
En fait, le Docker Engine™ correspond à une partie spécifique de l'architecture de Docker® qui concerne précisément la communication entre le client et le serveur.

Cette application client-serveur est composée de trois éléments principaux :

- Un outil de ligne de commande qui a le rôle du client
- Une API REST qui gère les interfaces que les programmes utilisent pour communiquer avec le Docker® daemon
- Un serveur appelé Docker® daemon

Ces trois éléments du Docker Engine™ sont illustrés par le schéma ci-dessous.

Figure 4 : Docker Engine™

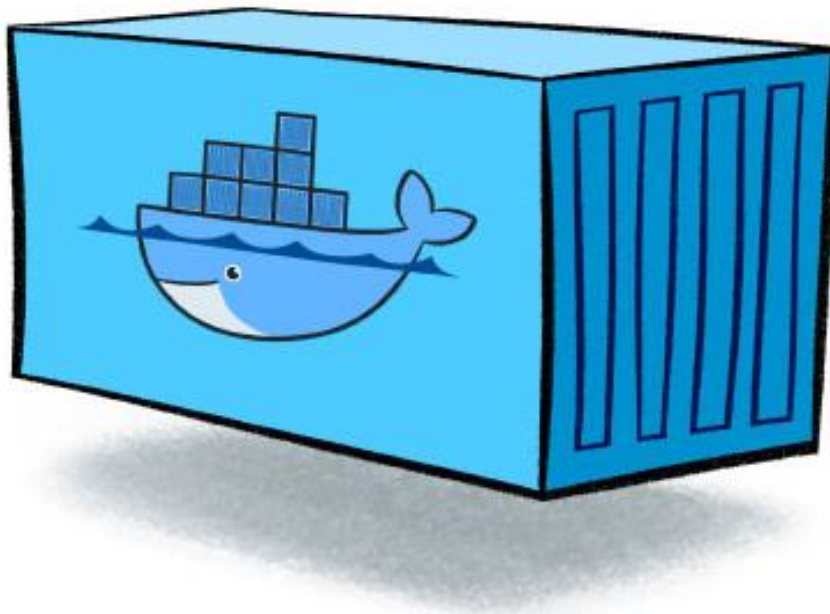


[\(https://docs.docker.com/machine/overview/\)](https://docs.docker.com/machine/overview/)

7.5.2 Conteneur

À l'origine, le concept de conteneurisation utilisé par Docker® pour ses solutions logicielles vient des conteneurs physiques utilisés depuis des dizaines d'années dans le domaine du transport de marchandises.

Figure 5 : Conteneur Docker®



[\(http://www.nebulaworks.com/blog/2016/08/04/docker-container-basics-operations-guide-part-1-3/\)](http://www.nebulaworks.com/blog/2016/08/04/docker-container-basics-operations-guide-part-1-3/)

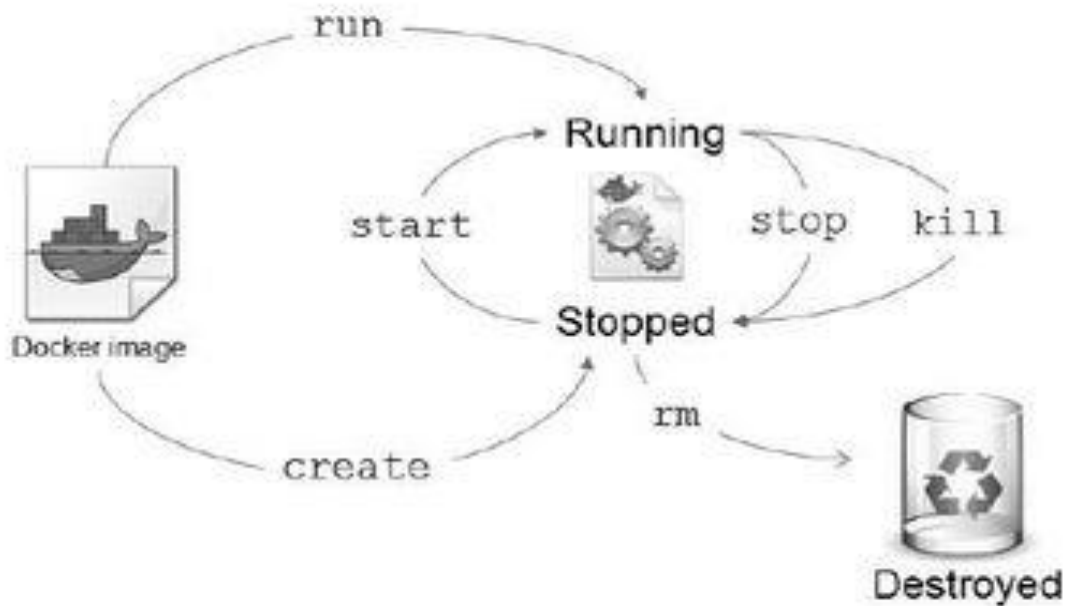
Un conteneur Docker® est une instance exécutable d'une Docker® image. Les différentes actions suivantes peuvent être effectuées sur un conteneur à partir de l'outil de ligne de commande de Docker® ou d'une API :

- Exécuter un conteneur
- Arrêter un conteneur
- Déplacer un conteneur
- Supprimer un conteneur

Chaque conteneur est une plateforme d'applications indépendante et sécurisée à partir de laquelle on peut accéder à des bases de données ainsi qu'à des ressources présentes dans un autre conteneur.

Pour fonctionner, un conteneur utilise le noyau Linux™ de la machine hôte et doit être composé de tous les fichiers liés à la création d'une image, ainsi que des métadonnées concernant le conteneur lorsque celui-ci est créé ou exécuté.

Figure 6 : Cycle de vie d'un conteneur



(CLOUX, Pierre-Yves, GARLOT, Thomas, KOHLER, Johann, 2016. *Docker, pratique des architectures à base de conteneurs*. Malakoff : Dunod. ISBN: 978-2-10-074744-3)

L'illustration ci-dessus (cf. Figure 6 : Cycle de vie d'un conteneur) représente les différentes étapes de la vie d'un conteneur Docker®.

Les différentes flèches représentent les actions réalisées par l'utilisateur à partir de l'outil de ligne de commande faisant office de client dans l'architecture de Docker® (cf. Figure 3 : Architecture de Docker®).

Sur ce schéma, nous pouvons clairement constater que la création d'un conteneur se fait à partir d'une Docker® image.

Par ailleurs, les autres commandes présentes sur ce schéma sont détaillées ci-dessous :

- **CREATE** : crée un conteneur à partir d'une image.
- **RUN** : exécute une image dans un conteneur.
- **START** : démarre un conteneur qui se trouve en état d'arrêt.
- **STOP** : arrête un conteneur en cours d'exécution.
- **KILL** : correspond à **STOP** mais supprime instantanément le processus principal du conteneur.
- **RM** : supprime un conteneur.

7.5.3 Docker® image

Une Docker® image est un modèle permettant de créer un conteneur. Par exemple, une image peut contenir un système d'exploitation Ubuntu™ avec un serveur web et une application web.

En fait, une Docker® image correspond à une manière simple de réutiliser et déplacer des données. Ces données représentées par une Docker® image correspondent au contenu d'un conteneur.

Il est possible de créer des images ex nihilo ou de télécharger et d'utiliser des images créées et mises à disposition par d'autres personnes.

La description d'une Docker® image est faite dans un simple fichier texte appelé Dockerfile.

Un Dockerfile est un fichier de configuration contenant plusieurs instructions permettant de construire une Docker® image. Les instructions concernent plusieurs tâches comme la création d'un répertoire, la copie de fichiers, etc.

Ci-dessous, nous allons voir plus en détails quels sont les principaux éléments qui composent un Dockerfile et comment faire pour créer un Dockerfile de base :

- **FROM** : communique au Docker® daemon l'image de base qui permettra de construire une nouvelle image.
- **MAINTAINER** : cette instruction n'est pas exécutable, elle indique le nom de l'auteur du Dockerfile.
- **RUN** : permet d'exécuter une application.
- **CMD** : diffère de **RUN** car cette instruction n'exécute rien au moment de la construction d'une image. **CMD** spécifie seulement la commande prévue pour l'image tandis que **RUN** exécute durant la construction de l'image.
- **LABEL** : affecte des métadonnées à l'image.

- **EXPOSE** : permet d'ouvrir un port réseau durant l'exécution de l'image dans un conteneur.
- **ENV** : définit les variables d'environnement avec des paires de type clé-valeur.
- **COPY** : copie des dossiers et fichiers d'une source vers une destination dans le système de fichiers du conteneur.
- **ADD** : ressemble à **COPY** mais prend en charge en plus la présence d'une adresse URL en tant que source d'un élément à copier.
- **ENTRYPOINT** : représente la commande principale de l'image.
- **VOLUME** : autorise l'accès à un emplacement sur l'hôte à partir d'un conteneur.
- **USER** : indique le nom d'utilisateur à utiliser lorsque l'image est en cours d'exécution.
- **WORKDIR** : définit le dossier actif pour les autres instructions.
- **ONBUILD** : ajoute une instruction à exécuter lorsque l'image sert de modèle pour une autre image. Cette commande fonctionne comme si **RUN** était inséré directement après **FROM** dans le Dockerfile de la nouvelle image. Cela permet de construire régulièrement une nouvelle image contenant des valeurs qui changent de façon dynamique.

Afin de mieux comprendre le fonctionnement d'un Dockerfile dans la pratique, on peut voir ci-dessous un exemple d'un Dockerfile qui se base sur une image Ubuntu™ pour construire une nouvelle image.

Par ailleurs, nous pouvons constater qu'il n'est pas nécessaire de mettre en œuvre toutes les commandes décrites précédemment pour qu'un Dockerfile soit fonctionnel.

```
#
# Ubuntu Dockerfile
#
# https://github.com/dockerfile/ubuntu
#

# Pull base image.
FROM ubuntu:14.04

# Install.
RUN \
  sed -i 's/# \(.multiverse$\)/\1/g' /etc/apt/sources.list && \
  apt-get update && \
  apt-get -y upgrade && \
  apt-get install -y build-essential && \
  apt-get install -y software-properties-common && \
  apt-get install -y byobu curl git htop man unzip vim wget && \
  rm -rf /var/lib/apt/lists/*

# Add files.
ADD root/.bashrc /root/.bashrc
ADD root/.gitconfig /root/.gitconfig
ADD root/.scripts /root/.scripts

# Set environment variables.
ENV HOME /root

# Define working directory.
WORKDIR /root

# Define default command.
CMD ["bash"]
```

(<https://dockerfile.github.io/#/ubuntu>)

7.5.4 Docker® registry

Le Docker® registry est une bibliothèque de Docker® images. Il peut être public ou privé.

Le Docker® registry public géré par Docker® s'appelle Docker Hub™ ; il contient les images officielles et permet aux développeurs de stocker et de partager des images qu'ils ont créées.

Il est possible de mettre à disposition une image de façon publique sur Docker Hub™ ou de la livrer dans un registre privé.

Figure 7 : Exemple d'images officielles disponibles dans Docker Hub™



<https://plus.google.com/+DockerIo/posts/3jDXtTEpZYU>

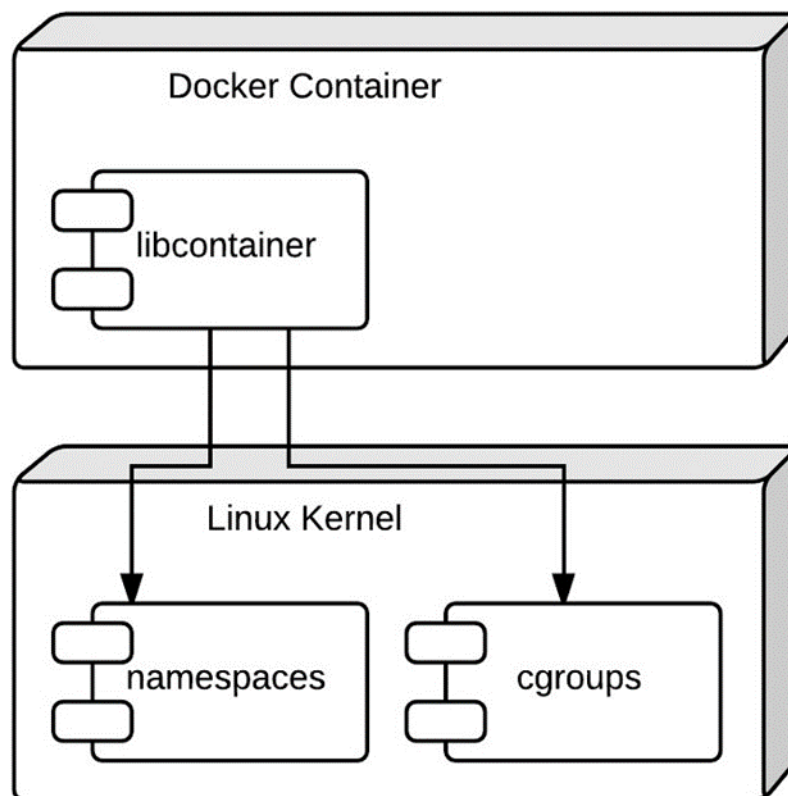
7.5.5 Libcontainer

Libcontainer est l'environnement d'exécution par défaut concernant les conteneurs de la plateforme Docker®.

Libcontainer est une bibliothèque de pilotes permettant de lier directement, sans autre dépendance, un conteneur Docker® avec le noyau Linux™ et ses composants comme on peut le constater sur l'illustration ci-dessous (cf. Figure 8 : Architecture d'un conteneur Docker®).

Grâce à Libcontainer, les conteneurs Docker® peuvent fonctionner efficacement avec les namespaces (cf. [7.5.6 Namespaces](#)), les CGroups (cf. [7.5.7 CGroups](#)), les interfaces réseaux ou les règles de pare-feu.

Figure 8 : Architecture d'un conteneur Docker®



<https://medium.com/aws-activate-startup-blog/a-better-dev-test-experience-docker-and-aws-291da5ab1238#.ljf4h2a07>

7.5.6 Namespaces

Comme expliqué précédemment, le fonctionnement de Docker® est lié à Linux™ par des éléments du noyau Linux™.

Le terme « namespaces²⁴ » correspond à une caractéristique du noyau Linux™ comme illustré ci-dessus (cf. Figure 8 : Architecture d'un conteneur Docker®).

Les namespaces permettent d'isoler les conteneurs et de virtualiser des ressources telles que l'identifiant de processus, identifiant d'utilisateur ou le système de fichiers du conteneur.

Par conséquent, ils peuvent alors avoir leurs propres périphériques réseaux, une adresse IP et un nom d'hôte unique. Cela permet aux conteneurs de s'exécuter indépendamment les uns des autres.

À l'exécution d'un conteneur, Docker® crée un ensemble de namespaces pour celui-ci. Ces namespaces fournissent une couche d'isolement qui fait que chaque aspect d'un conteneur est exécuté dans un namespace séparé et son accès est limité à ce namespace.

7.5.7 CGroups

Les CGroups²⁵ (Control Groups) représentent un autre élément présent dans le noyau Linux™ (cf. Figure 8 : Architecture d'un conteneur Docker®) sur lequel se base la plateforme Docker® pour mettre en œuvre son concept de virtualisation par conteneur.

CGroups est une caractéristique du noyau Linux qui permet de fixer des limites et de partitionner les ressources physiques d'un hôte aux conteneurs.

Docker® l'utilise afin de contrôler et prioriser l'utilisation de différents éléments comme la mémoire, la puissance de calcul, et l'accès au réseau.

Cela est très utile dans le cas où une application web et une base de données sont installées sur un serveur.

Dans ce cas-là, CGroups permet de répartir la mémoire et la puissance de calcul pour que l'exécution des processus soit optimisée.

Chaque conteneur a un CGroup qui lui est unique. Tous les processus du conteneur font partie du même CGroup.

²⁴ <http://man7.org/linux/man-pages/man7/namespaces.7.html>

²⁵ <https://www.kernel.org/doc/Documentation/cgroup-v1/cgroups.txt>

8. Comparaison des solutions de virtualisation

Dans le but d'étudier quelles sont les solutions adaptées à différents cas, ce chapitre fera l'analyse des différents types de virtualisation évoqués dans ce mémoire pour définir à quels besoins chacun d'entre eux correspond.

8.1 Conteneurs VS machines virtuelles

Pour savoir quels sont les cas d'utilisation adaptés à la virtualisation basée sur des conteneurs ou à la virtualisation avec des machines virtuelles, nous allons analyser les forces et les faiblesses de ces deux façons de mettre en œuvre la virtualisation.

Dans l'architecture des machines virtuelles, l'élément essentiel est l'hyperviseur. C'est une plateforme de virtualisation qui permet de gérer l'exécution des machines virtuelles.

Le concept de ce type de virtualisation est d'émuler le matériel physique pour que chaque machine virtuelle ait l'impression de fonctionner avec son propre matériel.

Les deux types d'hyperviseurs qui existent sont représentés ci-dessous (cf. Figure 9 : Hyperviseurs de type 1 et 2).

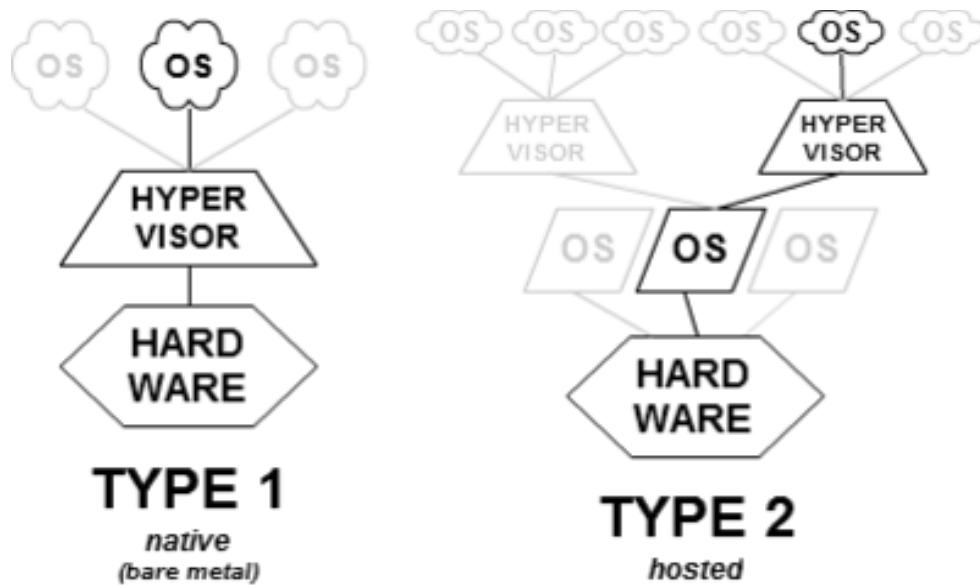
L'hyperviseur de type 1, appelé aussi « bare metal » (« métal nu » en français) est exécuté directement sur le matériel de la machine hôte. Il permet de gérer l'exécution des machines virtuelles et des systèmes d'exploitation invités.

Contrairement à l'hyperviseur de type 1, l'hyperviseur de type 2 ne s'exécute pas directement sur le matériel mais à l'intérieur d'un autre système d'exploitation.

Dans ce cas, la machine virtuelle et le système d'exploitation invité se situent à un niveau plus élevé par rapport au matériel de la machine physique.

La majorité des logiciels de virtualisation fonctionne avec un hyperviseur de type 2. Son avantage est de pouvoir exécuter plusieurs hyperviseurs simultanément étant donné qu'ils ne sont pas directement liés à la couche matérielle.

Figure 9 : Hyperviseurs de type 1 et 2



<https://en.wikipedia.org/wiki/Hypervisor>

La virtualisation avec des conteneurs qu'utilise Docker® se différencie de la virtualisation « classique » avec des machines virtuelles (cf. Figure 10 : Machines virtuelles et conteneurs).

Dans la virtualisation, l'hyperviseur gère les ressources de la machine hôte. Il est donc possible de partager les ressources mémoires car les machines virtuelles ne consomment pas la totalité de leur mémoire allouée.

Les machines virtuelles sont composées d'un système d'exploitation complet avec leur propre gestion de mémoire, pilotes de périphériques et bibliothèques.

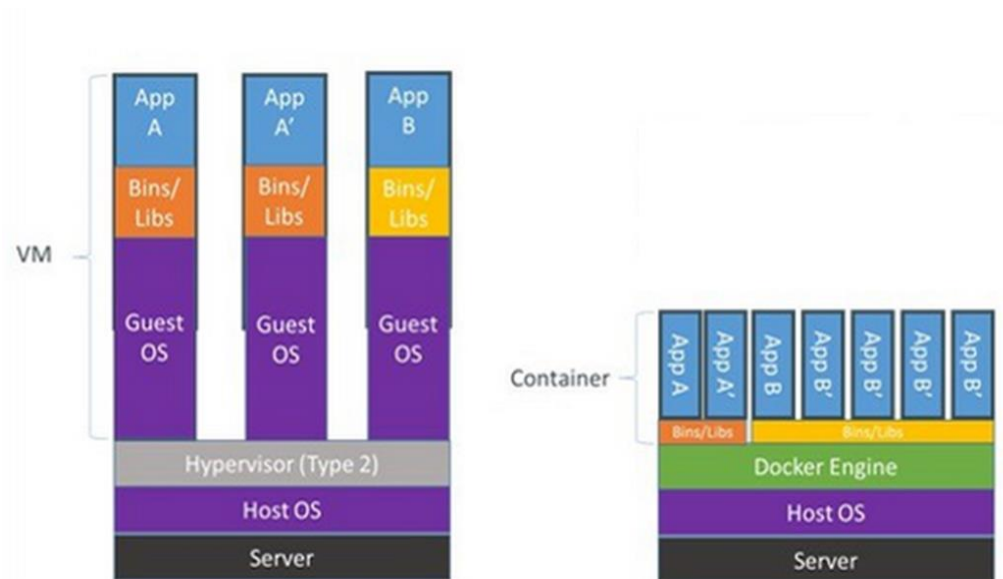
Par contre, les conteneurs partagent le même système d'exploitation, celui de la machine hôte et également des bibliothèques si elles sont communes à plusieurs conteneurs.

Par conséquent, les conteneurs représentent une solution de virtualisation plus légère et portable car ils utilisent moins de ressources que les machines virtuelles.

D'ailleurs, les conteneurs Docker® sont souvent appelés « machines virtuelles légères » car les deux concepts ont un même objectif qui est de virtualiser des systèmes.

La principale différence entre les machines virtuelles et les conteneurs est leur architecture.

Figure 10 : Machines virtuelles et conteneurs



(<https://www.upguard.com/articles/docker-vs.-vmware-how-do-they-stack-up>)

8.1.1 Analogie avec le monde réel

Afin d'aborder les différences entre les concepts de machines virtuelles et de conteneurs d'un autre point de vue, nous pouvons faire une analogie avec le monde réel, où les machines virtuelles seraient des maisons et les conteneurs correspondraient à des appartements.

Les maisons sont autonomes et protégées des intrus, chacune d'entre elle possède sa propre infrastructure (électricité, chauffage, etc.).

En ce qui concerne les appartements, ils sont également protégés des intrus mais sont construits autour d'une infrastructure partagée.

Le bâtiment, représentant le serveur dans l'environnement de Docker®, fournit entre autres l'électricité et le chauffage à chaque appartement.

Par ailleurs, les appartements peuvent être de tailles très différentes (par exemple : d'un studio à un 6 pièces). C'est-à-dire que nous pouvons louer exactement ce dont nous avons besoin et choisir la taille de notre appartement (conteneur) alors qu'une maison (machine virtuelle) est composée de plusieurs pièces et sa surface est généralement plutôt importante.

8.1.2 Caractéristiques

Dans le contexte du logiciel Docker®, lorsqu'un développeur crée une image pour l'exécuter dans un conteneur, il utilise uniquement ce dont il a besoin. Il utilise une image de base et y ajoute seulement les applications nécessaires.

Au contraire, les machines virtuelles sont constituées d'un système d'exploitation complet auquel des éléments inutiles peuvent être retirés selon les cas.

Tableau 1 : Comparaison entre les machines virtuelles et les conteneurs

Caractéristiques	Machines virtuelles	Conteneurs
Système d'exploitation invité	Chaque machine virtuelle est exécutée directement sur le matériel	Tous les systèmes d'exploitation invités partagent le même système d'exploitation hôte et le même noyau Linux™
Sécurité	Plus sécurisé car chaque machine virtuelle a son propre noyau de système d'exploitation	Vulnérable à cause du partage du noyau Linux™ par les différents conteneurs
Performance	Performances limitées à cause du temps de latence entre le système d'exploitation invité et hôte	Performances proches du système d'exploitation hôte
Isolation	Partage de bibliothèques et de fichiers entre les invités et l'hôte impossible	Les conteneurs sont isolés mais peuvent partager un système d'exploitation, des bibliothèques ou des fichiers.
Temps de démarrage	Quelques minutes	Quelques secondes
Stockage	Beaucoup de mémoire car tout le système d'exploitation et ses programmes sont installés et exécutés	Les conteneurs prennent peu de mémoire car le système d'exploitation hôte est partagé
Déploiement	Lourd et difficile	Léger et facile

(<https://jyx.jyu.fi/dspace/bitstream/handle/123456789/48029/URN:NBN:fi:jyu-201512093942.pdf?sequence=1>)

8.2 Cas pratiques d'utilisation

Les deux types de virtualisation traités dans ce mémoire ont chacun leur propre fonctionnement qui correspond et peut être adapté à différents cas pratiques d'utilisation de la virtualisation.

Tout d'abord, les conteneurs utilisés par le logiciel Docker® ont des caractéristiques qui permettent notamment de l'utiliser dans les cas suivants :

- Développement d'applications
- Test d'applications sur divers systèmes d'exploitation
- Serveur web

Mis à part Docker®, il existe plusieurs logiciels qui fournissent la mise en œuvre de la virtualisation avec des conteneurs :

- LXC™ (Linux Containers™)
- OpenVZ™
- Solaris Containers™

Par ailleurs, la virtualisation basée sur les machines virtuelles est utile dans les différents cas suivants :

- Partager un système informatique entre plusieurs utilisateurs
- Sauvegarder des systèmes d'exploitation
- Créer un service de cloud computing

Les principaux logiciels qui fournissent la mise en œuvre de la virtualisation avec des machines virtuelles sont :

- Oracle VirtualBox™
- Citrix XenServer™
- VMWare Workstation™

9. Conclusion

La virtualisation est une technologie en constante évolution qui est très intéressante pour les entreprises de par les avantages qu'elle apporte. En effet, les économies qui peuvent être réalisées grâce à la virtualisation sont très importantes. En utilisant la virtualisation, les besoins en matériels informatiques baissent, ce qui engendre une réduction des coûts plus globale liée notamment aux frais d'électricité et aux frais de maintenance des machines physiques.

Par ailleurs, la nouvelle façon de virtualiser que propose Docker® permet notamment aux développeurs de créer et tester des applications et aux administrateurs système de gérer un environnement virtualisé avec des conteneurs.

De plus, il est possible de transférer rapidement et d'héberger un grand nombre de conteneurs sur des machines physiques grâce à leur légèreté. Au contraire, étant donné le fait qu'elles sont composées d'un système d'exploitation complet, les machines virtuelles deviennent plus lourdes, ce qui diminue la portabilité et la rapidité de déploiement de celles-ci.

En ce qui concerne l'avenir de la virtualisation, bien que Docker® gagne de plus en plus de terrain dans ce domaine, nous avons constaté que, malgré les avantages liés à l'utilisation de conteneurs, ils ne constituent pas la solution idéale pour tous les cas d'utilisation.

Par conséquent, les machines virtuelles resteront probablement importantes dans ce domaine dans les années à venir. Depuis quelques temps, de nouveaux développements se font pour essayer de trouver une solution de virtualisation qui conviendrait à une majorité de cas. Celle-ci est appelée hybride parce qu'elle mélange les deux façons de virtualiser que sont les machines virtuelles et les conteneurs.

Pour terminer, ce mémoire m'a permis de développer mes connaissances dans le domaine du cloud computing et de la virtualisation, ainsi que de comprendre de manière plus globale leur importance dans le monde d'aujourd'hui.

À mon avis, au vu de l'ampleur qu'ont pris ces deux domaines depuis quelques années, ils sont devenus nécessaires dans l'époque d'informatisation dans laquelle nous vivons et qui se développe continuellement notamment avec l'arrivée des objets connectés et de la robotique.

Bibliographie

Livres

SOSINSKY, Barrie, 2011. *Cloud Computing Bible*. Indianapolis : Wiley Publishing, Inc. ISBN 978-0-470-90356-8

OLZAK, Thomas, BOOMER, Jason, KEEFER, Robert M., SABOVIK, James, 2010. *Microsoft Virtualization, Master Microsoft Server, Desktop, Application, and Presentation Virtualization*. Burlington : Elsevier Inc. ISBN 978-1-59749-431-1

CLOUX, Pierre-Yves, GARLOT, Thomas, KOHLER, Johann, 2016. *Docker, pratique des architectures à base de conteneurs*. Malakoff : Dunod. ISBN 978-2-10-074744-3

HENNION, Romain, TOURNIER, Hubert, BOURGEOIS, Eric, 2012. *Cloud computing, décider, concevoir, piloter, améliorer*. Paris : Eyrolles. ISBN 978-2-212-13404-9

HESS, Kenneth, NEWMAN Amy, 2010. *Virtualisation en pratique*. Paris : Pearson Education France. ISBN 978-2-7440-4154-9

PLOUIN, Guillaume, 2016. *Cloud computing, sécurité, gouvernance du SI hybride et panorama du marché*. Paris : Dunod. ISBN 978-2-10-074511-1

Sites web

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY, 2011. *The NIST Definition of Cloud Computing* [en ligne]. [Consulté le 21 novembre 2016]. Disponible à l'adresse : <http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>

SDXCENTRAL, 2016. What is Network Virtualization? – Definition [en ligne]. [Consulté le 2 janvier 2017]. Disponible à l'adresse : <https://www.sdxcentral.com/sdn/network-virtualization/definitions/whats-network-virtualization/>

GUERSAN, Guillaume, 2016. Où en est la virtualisation de poste de travail en 2016 ? [en ligne]. [Consulté le 2 janvier 2017]. Disponible à l'adresse : <http://guersanguillaume.com/virtualisation-de-poste-de-travail-2016/>

EUREVA, 2009. *La virtualisation de poste de travail et d'application, concepts & utilisation rapide* [en ligne]. [Consulté le 2 janvier 2017]. Disponible à l'adresse : <http://www.eureva.fr/media/682/La%20virtualisation%20v1.2.pdf>

LIPPIS CONSULTING, 2007. *Network Virtualization : The New Building Blocks of Network Design* [en ligne]. [Consulté le 2 janvier 2017]. Disponible à l'adresse : http://www.cisco.com/c/dam/en/us/solutions/collateral/enterprise-networks/network-fabric/net_implementation_white_paper0900aecd80707cb6.pdf

VMWARE, 2006. *Virtualization Overview* [en ligne]. [Consulté le 3 janvier 2017]. Disponible à l'adresse : <https://www.vmware.com/pdf/virtualization.pdf>

IBM, 2012. *z/VM – A Brief Review of Its 40 Year History* [en ligne]. [Consulté le 8 novembre 2016]. Disponible à l'adresse : <http://www.vm.ibm.com/vm40hist.pdf>

IBM, 2012. *45 (40) Years of Mainframe Virtualization : CP-67/CMS and VM/370 to z/VM* [en ligne]. [Consulté le 8 novembre 2016]. Disponible à l'adresse : [https://www-01.ibm.com/events/wwc/grp/grp019.nsf/vLookupPDFs/7%20-%20VM-45-JahreHistory-EA-J-Elliott%20%5BKompatibilit%C3%A4tsmodus%5D/\\$file/7%20-%20VM-45-JahreHistory-EA-J-Elliott%20%5BKompatibilit%C3%A4tsmodus%5D.pdf](https://www-01.ibm.com/events/wwc/grp/grp019.nsf/vLookupPDFs/7%20-%20VM-45-JahreHistory-EA-J-Elliott%20%5BKompatibilit%C3%A4tsmodus%5D/$file/7%20-%20VM-45-JahreHistory-EA-J-Elliott%20%5BKompatibilit%C3%A4tsmodus%5D.pdf)

EXAKIS, 2013. *Point sur la virtualisation, objectif virtualisation* [en ligne]. [Consulté le 8 novembre 2016]. Disponible à l'adresse : <https://123virtualization.files.wordpress.com/2013/04/1-0-objectif-virtualisation-point-sur-la-virtualisation.pdf>

EVERYTHING VM, 2011. History of Virtualization [en ligne]. [Consulté le 8 novembre 2016]. Disponible à l'adresse : <http://www.everythingvm.com/content/history-virtualization>

VMBLOG, 2012. Virtualization History [en ligne]. [Consulté le 9 novembre 2016]. Disponible à l'adresse : <http://vmblog.com/archive/2012/02/02/virtualization-history-has-an-impact-on-windows-server-backup.aspx#.WLBQp3-8o9p>

WIKIPEDIA, 2016. Conversational Monitor System [en ligne]. [Consulté le 9 novembre 2016]. Disponible à l'adresse : https://en.wikipedia.org/wiki/Conversational_Monitor_System

WIKIPEDIA, 2016. CP/CMS [en ligne]. [Consulté le 9 novembre 2016]. Disponible à l'adresse : <https://en.wikipedia.org/wiki/CP/CMS>

WIKIPEDIA, 2016. History of CP/CMS [en ligne]. [Consulté le 9 novembre 2016]. Disponible à l'adresse : <https://en.wikipedia.org/wiki/CP/CMS>

BAMBA, Khadidjatou Iman, 2014. *Les fondamentaux du cloud computing* [en ligne]. [Consulté le 21 novembre 2016]. Disponible à l'adresse : <https://itandsi.files.wordpress.com/2014/09/les-fondamentaux-du-cloud-computing.pdf>

STANFORD UNIVERSITY, 2008. *Amazon enters the cloud computing business* [en ligne]. [Consulté le 21 novembre 2016]. Disponible à l'adresse : <http://web.stanford.edu/class/ee204/Publications/Amazon-EE353-2008-1.pdf>

CMS-SPIP, 2010. SaaS, PaaS, IaaS, Cloud : définitions [en ligne]. [Consulté le 21 novembre 2016]. Disponible à l'adresse : <http://icp.ge.ch/sem/cms-spip/spip.php?article962>

WIKIPEDIA, 2013. Amazon Elastic Compute Cloud [en ligne]. [Consulté le 22 novembre 2016]. Disponible à l'adresse : https://fr.wikipedia.org/wiki/Amazon_Elastic_Compute_Cloud

WIKIPEDIA, 2016. Amazon S3 [en ligne]. [Consulté le 22 novembre 2016]. Disponible à l'adresse : https://fr.wikipedia.org/wiki/Amazon_S3

WIKIPEDIA, 2016. G Suite [en ligne]. [Consulté le 22 novembre 2016]. Disponible à l'adresse : https://en.wikipedia.org/wiki/G_Suite

SALESFORCE, 2015. Cloud computing – applis à la demande [en ligne]. [Consulté le 24 novembre 2016]. Disponible à l'adresse : <http://www.salesforce.com/fr/cloudcomputing/#where>

AITP – RESEARCH AND STRATEGY ADVISORY GROUP, 2008. *Virtualization and Its Benefits* [en ligne]. [Consulté le 15 décembre 2016]. Disponible à l'adresse : https://c.ymcdn.com/sites/www.aitp.org/resource/resmgr/research/virtualization_and_its_benef.pdf

VMWARE, 2016. Qu'est-ce que la virtualisation [en ligne]. [Consulté le 15 décembre 2016]. Disponible à l'adresse : <http://www.vmware.com/fr/solutions/virtualization.html>

INTECH, 2012. *Business Benefits from the Virtualization of an ICT Infrastructure* [en ligne]. [Consulté le 15 décembre 2016]. Disponible à l'adresse : <http://cdn.intechopen.com/pdfs/41074.pdf>

GARTNER, 2016. Magic Quadrant for x86 Server Virtualization Infrastructure [en ligne]. [Consulté le 28 novembre 2016]. Disponible à l'adresse : <https://www.gartner.com/doc/reprints?ct=160707&id=1-3B9FAM0&st=sb>

MARKETREALIST, 2014. Why "big data" and "virtualization" go hand in hand? [en ligne]. [Consulté le 28 novembre 2016]. Disponible à l'adresse : <http://marketrealist.com/2014/07/big-data-virtualization-go-hand-hand/>

ZDNET, 2014. Microsoft met les bouchées doubles pour convertir à Hyper-V [en ligne]. [Consulté le 28 novembre 2016]. Disponible à l'adresse : <http://www.zdnet.fr/actualites/microsoft-met-les-bouchees-doubles-pour-convertir-a-hyper-v-39799787.htm>

IDC, 2016. *Worldwide Datacenter Automation Software Market Shares, 2015 : Year of Suite Success* [en ligne]. [Consulté le 30 novembre 2016]. Disponible à l'adresse : <https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/products/vrealiz-e-suite/vmware-idc-worlwide-datacenter-automation-software-market-shares-2015.pdf>

WIKIPEDIA, 2016. Hyper-V [en ligne]. [Consulté le 30 novembre 2016]. Disponible à l'adresse : <https://en.wikipedia.org/wiki/Hyper-V>

JANCORG, 2015. LibContainer Overview [en ligne]. [Consulté le 22 février 2017]. Disponible à l'adresse : <http://jancorg.github.io/blog/2015/01/03/libcontainer-overview/>

DOCKER, 2014. Docker 0.9: introducing execution drivers and libcontainer [en ligne]. [Consulté le 22 février 2017]. Disponible à l'adresse : <https://blog.docker.com/2014/03/docker-0-9-introducing-execution-drivers-and-libcontainer/>

DOCKER. Dockerfile reference [en ligne]. [Consulté le 24 février 2017]. Disponible à l'adresse : <https://docs.docker.com/engine/reference/builder/>

DEIS, 2015. Dockerfile reference [en ligne]. [Consulté le 24 février 2017]. Disponible à l'adresse : <https://deis.com/blog/2015/dockerfile-instructions-syntax/>

DIGITAL OCEAN, 2013. Docker Explained: Using Dockerfiles to Automate Building of Images [en ligne]. [Consulté le 24 février 2017]. Disponible à l'adresse : <https://www.digitalocean.com/community/tutorials/docker-explained-using-dockerfiles-to-automate-building-of-images>

MEDIUM, 2016. Docking A Docker Container—Part 2 : Namespace, cgroup [en ligne]. [Consulté le 5 février 2017]. Disponible à l'adresse : <https://medium.com/@tomdeore/docking-a-docker-container-part-2-c1206e7c6677#.ltbdsloxf>

MEDIUM, 2014. A Better Dev/Test Experience: Docker and AWS [en ligne]. [Consulté le 5 février 2017]. Disponible à l'adresse : <https://medium.com/aws-activate-startup-blog/a-better-dev-test-experience-docker-and-aws-291da5ab1238#.qrpl7hkib>

CLOUDANDBIGDATALAB, 2014. Dockers and Micro services - CGroups and Namespaces [en ligne]. [Consulté le 5 février 2017]. Disponible à l'adresse : <http://cloudandbigdatalab.github.io/chameleon/docker/cgroups-namespaces/>

DOCKER. Docker Overview [en ligne]. [Consulté le 18 janvier 2017]. Disponible à l'adresse : <https://docs.docker.com/engine/understanding-docker/>

DOCKER. Docker Machine Overview [en ligne]. [Consulté le 18 janvier 2017]. Disponible à l'adresse : <https://docs.docker.com/machine/overview/>

DOCKER. Engine user guide [en ligne]. [Consulté le 18 janvier 2017]. Disponible à l'adresse : <https://docs.docker.com/engine/userguide/intro/>

DOCKER, 2016. Docker Announces Commercial Partnership with Microsoft to Double Container Market by Extending Docker Engine to Windows Server [en ligne]. [Consulté le 23 janvier 2017]. Disponible à l'adresse : <https://blog.docker.com/2016/09/docker-microsoft-partnership/>

1&1, 2016. Les conteneurs informatiques : la nouvelle virtualisation [en ligne]. [Consulté le 20 janvier 2017]. Disponible à l'adresse : <https://www.1and1.fr/digitalguide/serveur/know-how/conteneurs-informatiques-virtualisation-sans-emulation/>

DOCKER, 2016. Docker Announces Commercial Partnership with Microsoft to Drive Adoption of Containerized Applications in the Enterprise [en ligne]. [Consulté le 23 janvier 2017]. Disponible à l'adresse : <https://www.docker.com/docker-news-and-press/docker-announces-commercial-partnership-microsoft-drive-adoption-containerized>

WIKIPEDIA, 2016. Docker (software) [en ligne]. [Consulté le 17 janvier 2017]. Disponible à l'adresse : [https://en.wikipedia.org/wiki/Docker_\(software\)](https://en.wikipedia.org/wiki/Docker_(software))

WIKIPEDIA, 2016. Docker, Inc. [en ligne]. [Consulté le 17 janvier 2017]. Disponible à l'adresse : https://en.wikipedia.org/wiki/Docker,_Inc.

WIKIPEDIA, 2016. Docker (logiciel) [en ligne]. [Consulté le 17 janvier 2017]. Disponible à l'adresse : [https://fr.wikipedia.org/wiki/Docker_\(logiciel\)](https://fr.wikipedia.org/wiki/Docker_(logiciel))

WIKIPEDIA, 2016. Solomon Hykes [en ligne]. [Consulté le 17 janvier 2017]. Disponible à l'adresse : https://fr.wikipedia.org/wiki/Solomon_Hykes

WIKIPEDIA, 2017. Hypervisor [en ligne]. [Consulté le 15 février 2017]. Disponible à l'adresse : <https://en.wikipedia.org/wiki/Hypervisor>

ALFA-SAFETY, 2015. Conteneur, Docker, fonctionnement et avantages pour héberger ses applications ? [en ligne]. [Consulté le 20 janvier 2017]. Disponible à l'adresse : <http://actu.alfa-safety.fr/produit/hebergement-cloud-produit/conteneur-docker-fonctionnement-et-avantages-pour-heberger-ses-applications/>

SLASHROOT, 2014. Difference between Hypervisor Virtualization and Container Virtualization [en ligne]. [Consulté le 10 février 2017]. Disponible à l'adresse : <http://www.slashroot.in/difference-between-hypervisor-virtualization-and-container-virtualization>

LIBRE EN QUESTIONS, 2016. VMWare vs Docker : comment virtualiser efficace ? [en ligne]. [Consulté le 13 février 2017]. Disponible à l'adresse : <http://www.libre-en-questions.net/administration/vmware-docker-virtualiser-efficace/>

BOBCARES, 2016. Docker vs Virtual machines – What's best for your business ? [en ligne]. [Consulté le 13 février 2017]. Disponible à l'adresse : <https://bobcares.com/blog/docker-vs-virtual-machines/>

AIRPAIR, 2015. 8 Proven Real-World Ways to Use Docker [en ligne]. [Consulté le 15 février 2017]. Disponible à l'adresse : <https://www.airpair.com/docker/posts/8-proven-real-world-ways-to-use-docker>

TECHGLIMPSE, 2016. What is Docker, Difference between Docker and VM, Installation of Docker and its usage [en ligne]. [Consulté le 15 février 2017]. Disponible à l'adresse : <https://techglimpse.com/docker-installation-tutorial-centos/>

LINKEDIN, 2016. A Beginner Friendly Intro to Containers, VM and Docker [en ligne]. [Consulté le 15 février 2017]. Disponible à l'adresse : <https://www.linkedin.com/pulse/beginner-friendly-intro-containers-vm-docker-preethi-kasireddy>

DOCKER, 2016. *Docker for the Virtualization Admin* [en ligne]. [Consulté le 14 février 2017]. Disponible à l'adresse : https://goto.docker.com/rs/929-FJL-178/images/Docker-for-Virtualization-Admin-eBook.pdf?mkt_tok=eyJpIjoiTkRka1I6VTNnbVV6WTJZdyIsInQiOiJncCtDYldzMFk2N3RSTkwyMTFKMvc4cnpIQUdlZk1zZFRWNGx3a05VZGdlEc2emZQZ081YnNncGhqcWlyY2xMZ09NbkdqM3JXQWk4bnRGUINyQW9xaUVTWIArazd0VGszYlwwV3JJTmVYUUpSU3A1dUVPTE45OGJKa2kwZXIwOEMifQ%3D%3D