[Veille technologique]

[IMIE RENNES – ADMINISTRATEUR SYSTEMES ET RESEAUX]

[2017-2018]

Nom et prénom de l'apprenti : AJDAINI Hatim

Nom et prénom du tuteur : JANY Franck

Entreprise : OTIMA

Sommaire :

I) La virtualisation traditionnelle 1

1.a) Principe 1

1.b) Avantages 1

1.c) Inconvénients 2

1.d) Conclusion 2

II) La technologie Docker 2

2.a) Présentation de Docker 2

2.b) Qu'est-ce qu'un conteneur 3

2.c) Qu'est-ce que Docker 3

2.d) Petite histoire des conteneurs 3

III) Docker et la virtualisation traditionnelle 4

3.a) Isolation 4

3.b) Avantages de Docker par rapport à la virtualisation traditionnelle 5

IV) Association de Red Hat et Docker *(membres de la OCI)* 5

V) Docker passe sous Libcontainer 5

VI) Docker repose sur son propre outil de versioning : 6

VII) Docker compose pour gérer plusieurs conteneurs : 7

VIII) Implémentation de Docker 7

8.a) Oracle aussi pousse ses applications dans les containers 7

8.b) Microsoft séduit par Docker 8

8.c) Amazon 8

8.d) IBM intègre Docker à son cloud 9

8.e) VMWare 10

8.f) Le ralliement de docker à kubernetes 10

IX) Docker et son concurrent Rocket 11

X) Conclusion 11

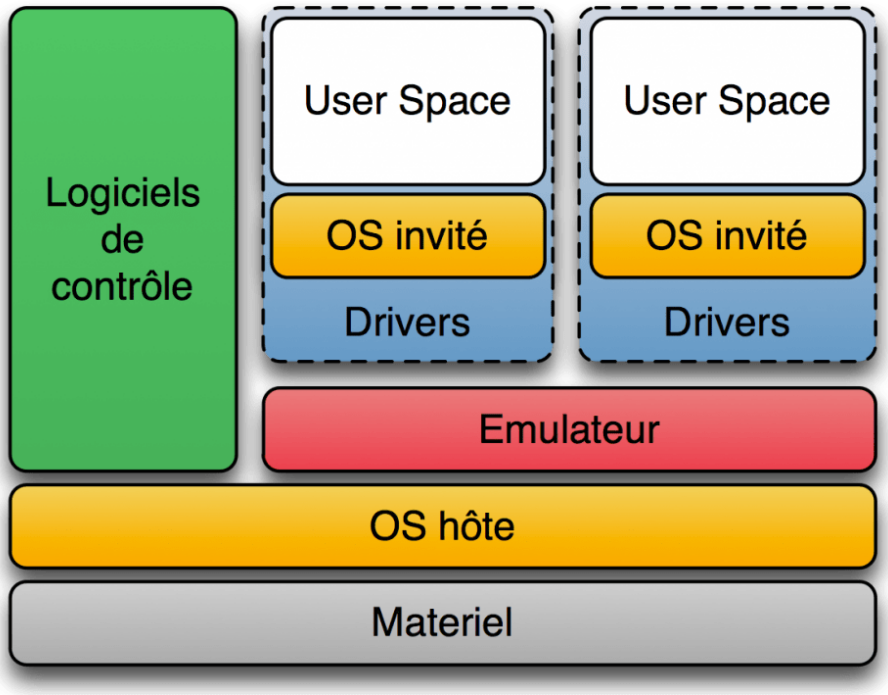
10.a) Légèreté des conteneurs 11

10.d) Orientation micro-services 12

I) La virtualisation traditionnelle

1.a) Principe

Le principe est simple c'est qu'au lieu d'avoir un serveur avec un système d'exploitation faisant tourner une ou plusieurs applications, on préférera mutualiser plusieurs serveurs virtuels depuis un serveur physique grâce à un logiciel nommé l'hyperviseur. L'hyperviseur permet d'émuler intégralement les différentes ressources matérielles d'un serveur ou d'un PC client, tels que l'unité centrale, la mémoire, le disque dur ou le réseau, et permet à des machines virtuelles de les partager. Ainsi ces machines virtuelles aussi nommées VM bénéficieront de ressources techniques selon leurs besoins (plus de puissance processeur, plus de mémoire vive, plus d'espace disque…). Si on le souhaite on peut aussi modifier les ressources techniques de ces VMs en quelques clics. De plus elles peuvent posséder leur propre système d'exploitation et leurs propres applications.



1.b) Avantages

- Consacrer les ressources adaptées selon les applications. On peut ainsi donner la puissance et la mémoire nécessaire et pas plus pour chaque VM.
- Les machines virtuelles sont simples à manier. On peut très bien altérer la VM d'une plateforme à une autre dans un autre lieu sans aucun impact visible pour les utilisateurs.

- La manipulation est plus simple, car une fois la machine virtuelle créée, on peut la sauvegarder, la dupliquer, et cela à volonté,
- La virtualisation réduit les dépenses en abaissant le besoin de systèmes matériels physiques. Elle permet ainsi de réduire la quantité d'équipement nécessaire et les coûts de maintenance, d'alimentation et de refroidissement des composants.
- Les machines virtuelles apportent également une aisance à l'administration car un matériel virtuel n'est pas sujet aux défaillances. Les administrateurs profitent des environnements virtuels pour faciliter les sauvegardes, la reprise après catastrophe.

1.c) Inconvénients

- Le fait d'accéder aux ressources de façon virtuelle affaiblie les performances cela est dû car on passe par une couche d'abstraction matérielle qui malheureusement doit faire des interprétations entre le matériel en place et celui simulé dans la machine virtuelle.
- Chaque machine virtuelle embarque un OS qui consomme à lui tout seul énormément de ressources alors qu'au final, ce qui nous intéresse c'est la ou les applications qui vont tourner dessus.

1.d) Conclusion

On se trouve alors avec une technologie très utile, malléable et économique pour les professionnels, mais malheureusement elle possède aussi son lot d'inconvénients, heureusement que d'autres personnes ont pensés à aller encore plus loin, et d'être encore plus efficace, et pour cela, Docker a été créé.

Sources :

- *06/10/2017 : https://www.supinfo.com/articles/single/5969-qu-est-ce-que-virtualisation*
- *https://www.vmware.com/fr/solutions/virtualization.html*

II) La technologie Docker

2.a) Présentation de Docker

Docker est une start-up spécialisée dans les conteneurs et fondée en France et devenue, ces derniers mois, l'une des étoiles montantes de la Silicon Valley. La technologie des conteneurs connaît depuis 2 ans un rythme d'adoption exponentiel poussé par le succès de Docker. Les levées de fonds spectaculaires enchaînées par la société Docker illustrent le potentiel attendu de cette technologie. Son rythme d'adoption serait l'un des plus rapide de toutes les technologies récentes, elle est déjà largement répandue chez les acteurs leaders des nouvelles technologies.

Docker est annoncée comme le prochain bouleversement de l'IT comme VMware a pu l'être il y a 15 ans avec la virtualisation d'applications.

2.b) Qu'est-ce qu'un conteneur

Un conteneur est une enveloppe virtuelle qui permet de packager une application avec tous les éléments dont elle a besoin pour fonctionner : fichiers source, runtime, librairies, outils et fichiers. Ils sont packagés en un ensemble cohérent et prêt à être déployé sur un serveur et son OS.

Contrairement à la virtualisation de serveurs et à une machine virtuelle, le conteneur n'intègre pas d'OS, il s'appuie directement sur le système d'exploitation du serveur sur lequel il est déployé.

2.c) Qu'est-ce que Docker

Docker est un projet Open Source proposant une surcouche qui automatise et simplifie le déploiement d'applications dans des conteneurs virtuels. C'est une technologie qui répond aux problématiques de la virtualisation traditionnelle. Le principe c'est d'avoir une machine (qui peut donc être virtuelle) hébergeant la plateforme docker et qui va nous permettre de gérer des conteneurs applicatifs. Ces conteneurs applicatifs standards qu'on peut dupliquer, déplacer, sauvegarder pour les utiliser par exemple sur une autre plateforme Docker, sur un autre serveur, dans un autre lieu. Docker nous permet de faire abstraction de l'OS, du moment que les ressources adéquates sont en place.

2.d) Petite histoire des conteneurs

La technologie que nous appelons aujourd'hui « conteneurs » est apparue en 2000 sous le nom de jail FreeBSD. L'objectif était de créer des processus dans un dans lequel l'accès au système de fichiers, au réseau et aux utilisateurs est virtualisé et de les y « emprisonner » afin qu'ils ne compromettent pas l'ensemble du système.

Très rapidement, d'autres technologies se sont greffées aux conteneurs pour concrétiser cette approche de l'isolement :

- La fonction du noyau croups (groupes de contrôle) permet de contrôler et de limiter l'utilisation des ressources pour un processus ou un groupe de processus.
- Le système d'initialisation systemd permet de définir l'espace utilisateur et de gérer les processus associés. Il est utilisé par la fonction cgroups pour offrir un niveau de contrôle plus élevé sur ces processus isolés. Ces deux technologies, qui permettent de mieux contrôler Linux, ont servi de base pour pouvoir exécuter des environnements bien qu'ils soient séparés.
- Les espaces de noms utilisateur les utilisateurs et des groupes peuvent avoir des privilèges pour effectuer certaines opérations au sein du conteneur, mais que ces privilèges ne leurs sont pas accordés en dehors du conteneur ». Le concept est similaire à celui d'un environnement jail, mais

avec un niveau de sécurité plus élevé, atteint grâce une isolation plus marquée des processus, au lieu d'un environnement modifié.

Sources :

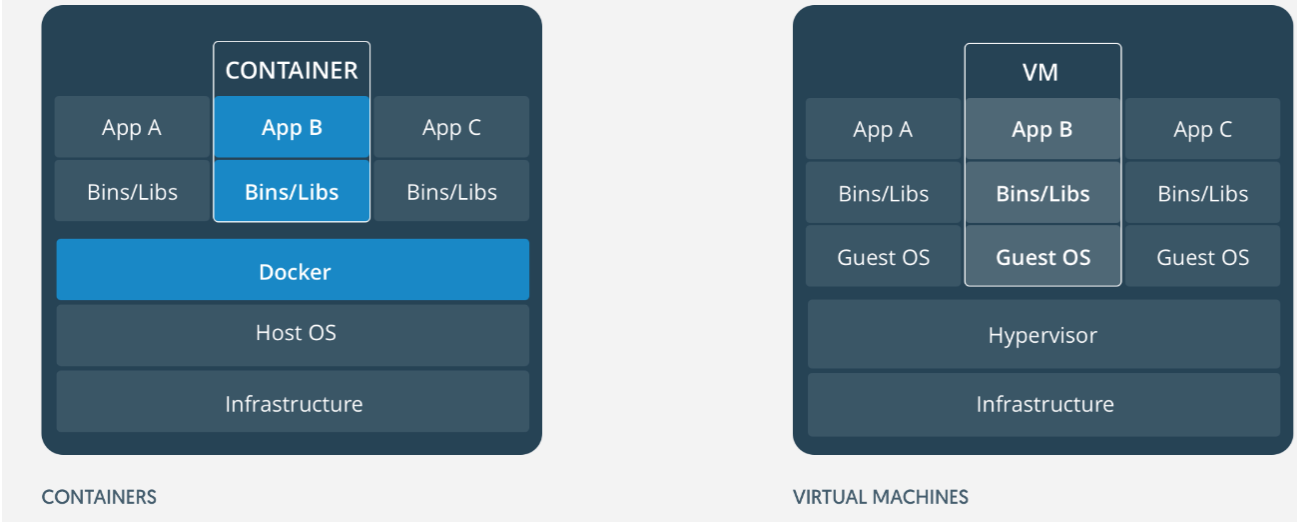
- 02/11/2015 :https://actu.alfa-safety.fr/devops/conteneur-docker-fonctionnement-et-avantages-pour-heberger-ses-applications/
- https://www.redhat.com/fr/topics/containers/whats-a-linux-container

III) Docker et la virtualisation traditionnelle

3.a) Isolation

Dans le cas de la virtualisation de serveurs (Vmware, Xen) l'isolation entre VMs est intrinsèque à la technologie, et l'on partage le serveur physique (CPU/RAM/Disque) en serveurs virtuels (VMs), ce sont des éléments très bas niveau que l'on virtualise.

Dans le cas des conteneurs, l'isolation se fait au niveau du système d'exploitation, nous mettons à disposition des éléments de plus haut niveau (pile réseau, dossier, zone mémoire, process namespace, etc.). Ce niveau d'isolation est aussi efficace du point de vue d'une application, car toutes les ressources sont utilisées à travers des éléments virtualisés. L'image ci-dessous illustre cette phase d'abstraction de l'OS.



3.b) Avantages de Docker par rapport à la virtualisation traditionnelle

- Ces machines virtuelles intègrent elles-mêmes un OS allant jusqu'à Giga-octets. Ce n'est pas le cas du container. Le container appel directement à l'OS pour réaliser ses appels système et exécuter ses applications. Il est beaucoup moins gourmand en ressources
- Les administrateurs ne vont plus gérer la sécurité et les configurations réseau de plusieurs serveurs, mais d'un seul, de plus on peut faire des tests sur nos applications toute en gardant une ancienne version de l'application.
- La portabilité est un des points clés de cette nouvelle voûte qu'est Docker, on peut déplacer les conteneurs d'un environnement à l'autre sans perte de données. À la manière de la machine virtuelle que l'on peut déplacer de serveurs à serveurs mais en gardant cette couche OS allant jusqu'à Giga-octets, alors que sur Docker on est sur des conteneurs de tailles beaucoup plus réduites que vous pouvez manipuler à votre gré !

Sources :

- *02/11/2015 :*https://actu.alfa-safety.fr/devops/conteneur-docker-fonctionnement-et-avantages-pour-heberger-ses-applications/

IV) Association de Red Hat et Docker *(membres de la OCI)*

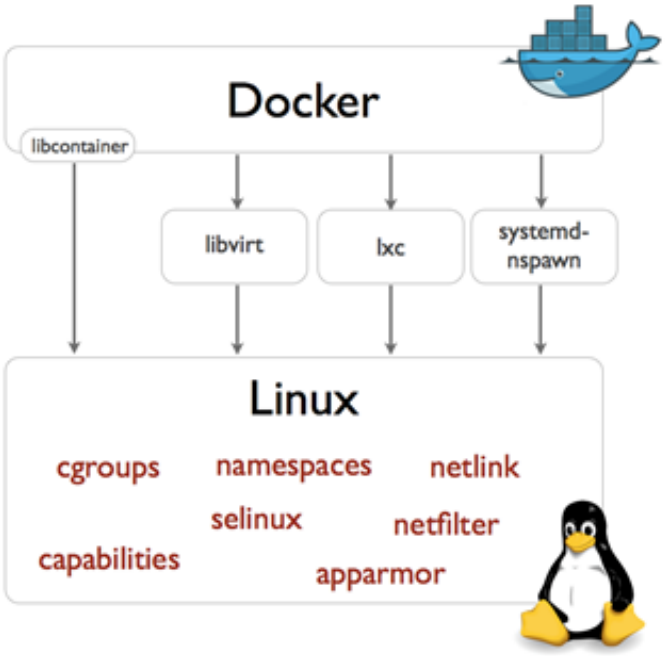
En 2008, la technologie de conteneurs Docker a fait son apparition (via dotCloud). Elle combine le travail du projet LXC (Linux container) à des outils améliorés pour les développeurs, qui augmentent le niveau de convivialité des conteneurs. La technologie Open Source Docker est actuellement le projet le plus connu et utilisé pour déployer et gérer des conteneurs Linux. Aujourd'hui, Red Hat et Docker font partie des nombreux membres de l'OCI (Open Container Initiative), qui travaille à la création de normes ouvertes pour les technologies de conteneurs à l'échelle du secteur.

Sources :

- *https://www.redhat.com/fr/topics/containers/whats-a-linux-container*

V) Docker passe sous Libcontainer

Docker est passé à Libcontainer après la v.1.8. L'intérêt est que Docker peut maintenant supporter plus de technologies d'isolation. Libcontainer est une bibliothèque écrite en Go pour la création de conteneurs avec des espaces de noms, les groupes de contrôle, les capacités et les contrôles d'accès du système de fichiers. Cette librairie a été développée pour faire le travail de LXC (linux container) tout en simplifiant l'installation de docker.



Source du fondateur de Docker :

- 10/07/2014 **:** https://blog.docker.com/2014/03/docker-0-9-introducing-execution-drivers-and-libcontainer/

VI) Docker repose sur son propre outil de versioning :

Docker inclut des fonctionnalités de type git pour le suivi des versions successives d'un conteneur, l'inspection de la différence entre les versions validant de nouvelles versions, le retour en arrière, etc. L'historique inclut également la manière dont un conteneur a été assemblé et par qui, vous obtenez ainsi une traçabilité complète depuis le serveur de production jusqu'au développeur en amont. Docker implémente également des téléchargements incrémentiels et des téléchargements, similaires à git pull, de sorte que les nouvelles versions d'un conteneur peuvent être transférées en envoyant uniquement des différences.

Sources

- 31/07/2016 : https://www.youtube.com/watch?v=NYfKsgDv_yA&index=2&list=PL6tu16kXT9PqdhOZk4MNVtQDJp6xFrotg

VII) Docker compose pour gérer plusieurs conteneurs :

Explication de l'utilité de docker compose par un exemple :

J'ai besoin d'une stack Nginx, PHP FPM, MariaDB et Memcached pour faire des tests. Dans ce cas Docker répond parfaitement à la demande. Pour cela nous avons besoin de créer quatre containers (un pour chaque service) et de les lancer ensemble. Néanmoins cette gestion de la création, ordonnancement et lancement des containers est fastidieuse car il faut le faire manuellement et respecter chaque étape. C'est là que docker-compose vous simplifie la vie.

En se basant sur un fichier docker-compose.yml contenant la configuration de votre stack, docker-compose va créer, lancer, ouvrir les ports, créer les volumes et lier vos containers tout seul.

Docker Compose est l'outil "officiel" proposé par *Docker, Inc* en ayant racheté Fig. Outil qui permet de définir et exécuter des applications docker multi-conteneur. Avec compose, vous pouvez utiliser un fichier Composer pour configurer les services de votre application. Ensuite, en utilisant une seule commande, vous créez et démarrez tous les services de votre configuration.

Sources

- 16/08/2016 : https://www.youtube.com/watch?v=k900NVwFfcA&t=1s

VIII) Implémentation de Docker

Docker est conçu pour les microservices et la portabilité en Cloud. Les microservices contribuent à empêcher les problèmes qui surviennent lorsqu'un service donné en affecte un autre, l'autre point clef est la portabilité des applications et à sa technologie open source, ce pourquoi Docker suscite un formidable engouement.

Docker assure une véritable portabilité aux applications par leur encapsulation dans des conteneurs. Ce qui explique que de nombreux fournisseurs de Cloud s'y intéressent. Google, Microsoft Azure, Amazon Web Services et d'autres encore. Tous ces fournisseurs utilisent Docker pour créer des infrastructures prenant en charge les conteneurs.

8.a) Oracle aussi pousse ses applications dans les containers

Oracle propose également plusieurs de ses applications phares en conteneurs via le Docker Store. Cela signifie que les clients qui utilisent une des versions commerciales de Docker peuvent provisionner

facilement des logiciels comme Oracle Database, Java SE 8 et WebLogic Server. Toutes ces annonces ont été faites lors du DockerCon 2017.

Sources :

- *20/04/2017 :* https://www.lemondeinformatique.fr/actualites/lire-un-programme-docker-pour-migrer-des-applications-vers-des-conteneurs-67982.html

8.b) Microsoft séduit par Docker

Microsoft s'intéresse depuis longtemps aux containers, notamment de type Docker. L'éditeur a déjà porté cette technologie sur son cloud Azure en vue d'en fluidifier les processus de déploiement. En lien avec Docker, il a aussi décliné le moteur Docker (initialement limité à Linux) pour Windows Server, donnant ainsi naissance à des containers natifs au format Windows.

Docker for Azure est un environnement intégré et facile à déployer pour créer, assembler et expédier des applications de conteneur sur Microsoft Azure. En tant qu'application Azure native, Docker for Azure est optimisé pour les services Azure IaaS et ne nécessite aucun logiciel à installer.

Docker et Microsoft se sont associés pour apporter l'agilité, la portabilité et les prestations de sécurité de la plateforme Docker à chaque édition de Windows Server 2016. Les conteneurs Windows Server 2016, propulsés par le moteur Docker, fournissent des conteneurs aux applications Windows natives et enrichissent la boîte à outils des développeurs Docker traditionnels qui codent sous Linux et des professionnels de l'IT.

Mais le groupe commence à pousser ce type de technologie au cœur de Windows 10 dans l'optique d'en faire bénéficier les appareils (PC, tablettes, objets connectés...). C'est ce qui ressort d'une conférence donnée en mai dernier par Alex Ionescu, directeur de la stratégie au sein de la société de sécurité informatique CrowdStrike. Expert en systèmes d'exploitation, Alex Ionescu est loin d'être un inconnu. Il est co-auteur de Windows Internals, un ouvrage de référence sur l'architecture de Windows... qui a même sa page de présentation sur le site de Microsoft.

Sources :

- *(21/09/2017) :* https://www.journaldunet.com/solutions/dsi/1197143-les-containers-docker-la-prochaine-grande-mutation-de-windows-10/
- https://www.docker.com/docker-azure
- https://www.docker.com/microsoft

8.c) Amazon

Docker et AWS se sont associés pour simplifier au maximum le déploiement d'un environnement Docker Containers en tant que service (CaaS = Container as a Service) sur l'infrastructure EC2 d'Amazon. L'exécution de Docker Datacenter sur AWS offre aux développeurs et aux opérations

informatiques un moyen très fiable et peu coûteux de déployer des charges de travail prêtes pour la production en un seul clic.

Les développeurs Cloud sont souvent aux prises avec des problèmes de configuration. Ainsi, un service donné peut parfaitement fonctionner en environnement de test et ne plus se comporter comme prévu en production. Amazon EC2 Container Service fournit des API hébergées, gérées et totalement programmables. Celles-ci permettent d'exécuter des conteneurs Docker au sein d'EC2. Les développeurs choisissent la taille et le nombre de serveurs dans le parc Docker et Amazon installe les conteneurs dans les instances.

En utilisant Docker sur AWS, les développeurs et les administrateurs disposent d'un moyen fiable et peu coûteux de développer, livrer et exécuter des applications distribuées à toutes les échelles. AWS prend en charge les deux modèles de licence Docker : open source avec Docker Community Edition (CE), et par abonnement avec Docker Enterprise Edition (EE).

Sources :

- *https://www.docker.com/aws*
- *Octobre 2015 : https://www.lemagit.fr/conseil/Amazon-EC2-Container-Service-de-lavantage-de-combiner-le-IaaS-dAWS-et-Docker*

8.d) IBM intègre Docker à son cloud

Après Amazon, Microsoft et Google, c'est au tour d'IBM de rentrer dans la course au support de Docker, la technologie de containers Open Source qui monte.

IBM et Docker sont partenaires depuis décembre 2014 offrant des solutions intégrées du datacenter au cloud et un support technique complet pour les entreprises du monde entier. Un partenariat technique approfondi permet la création et le déploiement de charges de travail multi-plateformes et multi-conteneurs dans des infrastructures hybrides,

IBM et Docker accélèrent la mise à disposition des applications et permettent la gestion du cycle de vie des applications pour les applications conteneurisées.

IBM a également noué un partenariat avec Docker, la société qui gère le logiciel Open Source de virtualisation, pour développer davantage Docker Hub Enterprise. Il s'agit d'un logiciel de gestion pour Docker qui peut être aussi bien être utilisé dans le cloud public, sur site que pour un déploiement dans le cloud privé. Les containers Docker peuvent être utilisées pour packager les applications de façon qu'elles puissent être facilement déplacées sur différents serveurs.

Sources :

- *05/12/2014 :* https://www.lemondeinformatique.fr/actualites/lire-ibm-integre-docker-a-son-cloud-59509.html
- https://www.ibm.com/cloud-computing/bluemix/fr/containers
- *04/12/2014 :* https://www-03.ibm.com/press/us/en/pressrelease/45597.wss

8.e) VMWare

Malgré la montée en puissance de Microsoft (avec Hyper-V), l'éditeur, vmware demeure de loin leader du domaine. Selon l'étude réalisé par Spiceworks , la technologie vSphere en tête des émulateurs de serveur les plus utilisés en entreprise. De 59% chez les TPE, son taux de pénétration augmente avec la taille des organisations pour atteindre 82% chez les grands comptes.

Dans ce contexte, on peut se demander quelle est la stratégie de la firme en matière de container logiciel, cette technologie dont l'américain Docker s'est fait le porte-drapeau, et qui est devenue en quelques années l'une des principales alternatives à la machine virtuelle (VM).

VMware pousse un peu plus loin ses pions sur le terrain de Docker avec l'annonce de deux projets Open Source qui permettront à ses utilisateurs de déployer plus facilement des conteneurs dans leurs infrastructures vSphere existantes. Sous le nom de projet Photon, il vient de livrer la préversion d'une technologie de conteneur s'appuyant sur une distribution Linux légère, conçu pour se lancer très rapidement sur son environnement de virtualisation. Photon supporte Docker. Le code source de Photon est accessible sur le référentiel Github*. L'éditeur espère ainsi favoriser la collaboration autour des applications exploitées en conteneurs dans un environnement virtualisé.

Sources :

- Projet Photon Github : https://github.com/vmware/photon
- *23/04/2015 :* https://www.lemondeinformatique.fr/actualites/lire-vmware-cloud-on-aws-arrive-en-europe-71084.html

8.f) Le ralliement de docker à kubernetes

Le fournisseur de conteneurs Docker a annoncé qu'il allait intégrer l'outil d'orchestration open source Kubernetes, offrant aux utilisateurs de sa solution Enterprise Edition le choix entre Kubernetes ou le gestionnaire de cluster maison Swarm. Le fondateur et CTO de Docker, Solomon Hykes, a indiqué à l'occasion de la DockerCon à Copenhague, que la décision était moins une réponse à une forte demande de la communauté, que le fait que, si Docker devait intégrer d'autres gestionnaires, le choix de Kubernetes a du sens.

L'association de Docker et de Kubernetes permet d'orchestrer une planification intelligente, une réparation spontanée et une mise à l'échelle horizontale en vue d'une livraison rapide des applications.

Sources :

18/10/2017 : https://www.lemondeinformatique.fr/actualites/lire-docker-ajoute-kubernetes-a-sa-plateforme-container-69728.html
https://blog.docker.com/2018/01/docker-ee-kubernetes/

IX) Docker et son concurrent Rocket

Editeur d'une distribution Linux orientée gestion de cluster, CoreOS dévoile son alternative aux containers de Docker. Cette société figurait pourtant parmi les soutiens de la première heure de la start-up de Solomon Hykes, séduite par sa volonté de proposer un container standard. CoreOS a d'ailleurs basé sa propre distribution sur Docker, et figure aussi parmi les principaux contributeurs du projet open source. "Mais Docker a depuis créé toute une série d'outils autour, pour provisionner des serveurs, des grappes de systèmes, et plein d'autres fonctions. Docker n'est plus le composant simple et réutilisable qu'il devrait être", regrette CoreOS. D'où la volonté de CoreOS de proposer une spécification de moteur de container plus dépouillée, visant à revenir à un projet "plus en phase avec la philosophie initiale de Docker". Baptisée Rocket,

Les deux offres de conteneurs ont leurs avantages uniques mais en dehors de rocket et de ses avantages de sécurité décroissants sur Docker et docker plus axé sur la simplicité, les offres des deux fournisseurs sont pour la plupart complémentaires.

Sources :

- *Lire le post sur le blog de la société CoreOS : https://coreos.com/blog/rocket.html*
- *Comparaisons détaillées : https://www.upguard.com/articles/docker-vs-coreos*
- *Pourquoi blabla car a choisit Rocket à la place de docker : http://blog.d2-si.fr/2017/04/19/blablacar-conteneurs-rocket/*

X) Conclusion

10.a) Légèreté des conteneurs

Contrairement à un serveur virtuel sous Linux, le conteneur n'a besoin que de quelques centaines Mo de disques. L'empreinte mémoire est aussi réduite, car nous n'avons que de la mémoire utilisée pour l'application (pas de couche OS). Il est ainsi plus rapide à démarrer mais aussi à déplacer d'une machine à une autre.

10.b) Rapidité et facilité de déploiement des applications

Après la récupération d'un template, qui se fait en une commande, il suffit d'une autre commande pour exécuter le conteneur, celui-ci démarre en quelques secondes. On peut ajouter des paramètres lors du démarrage qui seront transmis au conteneur. On va par exemple spécifier l'accès à la base de données ou d'autres paramètres d'automatisation.

L'idéal reste la mise en place d'une gestion de configuration, grâce à elle, nos conteneurs seront autonomes. On gère ses conteneurs comme on gère un package logiciel, cette facilité permet au développeur de pouvoir provisionner les environnements qui sont déployés en production par exemple sans connaissance particulière d'infrastructures ou d'administration système.

10.c) Portabilité et multi cloud

Il y a peu de formats d'enveloppes virtuelles qui soient nativement multi-cloud. Il est toujours compliqué de récupérer sur sa ferme VMware son POC fait sur AWS, ou encore migrer sa machine virtuelle sur une Ressource Group Azure…

Avec Docker, vous pouvez, en quelques secondes, déployer votre test fait en local sur votre poste, en production sur la très grande majorité des Clouds (pour ne pas dire tous).

10.d) Orientation micro-services

Les nouvelles applications métiers, sites Web et nouvelles applications IT, se composent maintenant de micro-services, à l'inverse d'une application contenue dans un seul serveur et une seule application.

Cette tendance répond à plusieurs objectifs :

- Améliorer l'élasticité d'une application,
- Augmenter la performance, chaque brique a un usage précis, dans laquelle elle est optimisée,
- L'architecture logicielle est aussi orientée microservices, le déploiement en continu est facilité car on n'impacte pas tout le code de l'application lors d'une modification, mais un seul module par exemple.

On fonctionne de manière modulaire, l'application n'est plus monolithique et cela apporte de la flexibilité et réduit les risques.

Docker facilite grandement ce mode de fonctionnement, que cela soit dans le déploiement et le maintien en condition opérationnelle des briques de l'application, que pour le déploiement et le développement. Chaque microservice de l'application pourra être packagé dans un conteneur.