

Comment les conteneurs Linux Changer le développement et Livraison des candidatures



1. RÉSUMÉ

Dans le but de créer des applications de manière fiable, une séparation s'est développée dans la plupart des organisations entre le « Développement », qui crée les applications, et les « Opérations », qui mettent en service et exécutent ces applications.

Il y a de bonnes raisons à cette séparation des préoccupations, mais elle est de plus en plus difficile à maintenir face aux défis auxquels sont confrontées les entreprises.

Les organisations doivent désormais répondre rapidement aux demandes et fournir rapidement des services flexibles. Cela est difficile à réaliser lorsqu'ils disposent d'une structure rigide pour produire et livrer des applications.

Le développement évolue vers de nouveaux modèles de développement « agile », mais un mouvement plus radical a émergé, selon lequel la séparation rigide entre le développement et l'exploitation elle-même freine la production de nouveaux services et rend les entreprises moins agiles.

Le « DevOps » a été proposé comme une nouvelle façon de travailler basée sur la collaboration et la communication entre les différentes parties de l'organisation.

Mais DevOps n'est qu'une théorie, et il ne deviendra pas quelque chose de plus tangible à moins qu'il ne soit adopté par une organisation qui comprend clairement les raisons pour lesquelles le développement et l'exploitation ont été séparés en premier lieu - et qui est prête à adopter des technologies qui leur permettent pour être réunis en toute sécurité.

Cet article soutient que DevOps peut être mis en œuvre de manière productive si les deux disciplines adoptent un modèle « cloud » pour la production et la consommation de services. Il soutient également que les conteneurs Linux constituent un changement technologique très puissant et pratique pour permettre ce modèle, à condition qu'ils soient utilisés avec des technologies d'automatisation et de gestion appropriées.

2. COMMENT EN SOMMES-NOUS ARRIVÉS ?

À première vue, il existe de très bonnes raisons de séparer les activités informatiques. Développer du code est une tâche très différente de la maintenance et du support de ce code. Les voitures sont construites dans une usine ; ils sont entretenus et soutenus par un garage. Pourquoi le code devrait-il être différent ?

En fin de compte, les utilisateurs ont besoin de choses différentes de la part des développeurs et des services opérationnels. Les délais de mise sur le marché des utilisateurs professionnels et la flexibilité de l'entreprise sont des facteurs clés. Les développeurs doivent comprendre leurs besoins et savoir concevoir et créer les systèmes qui y répondent. Les responsables des opérations doivent être capables de maintenir le statu quo et d'assurer un fonctionnement fiable et sécurisé.

Le développement doit créer rapidement de nouvelles fonctionnalités, tandis que les opérations ont pour objectif la stabilité, le contrôle et la gouvernance à long terme. Cette séparation est importante – et elle est d'autant plus importante que votre organisation est grande. Une entreprise développant et fournissant une myriade d'applications doit réellement disposer d'une structure fiable pour les prendre en charge et les gérer.

Les développeurs sont souvent incités sur la rapidité avec laquelle ils peuvent fournir du code, tandis que les opérations sont généralement incitées sur la durée entre les échecs. Emmené à leur (il) Extrêmement logique, cela conduirait à un service de développement fournissant du code qui n'est pas prêt, et à un service des opérations ayant tout intérêt à ne jamais autoriser la mise en œuvre de nouveau code !

Dans le modèle DevOps, le côté opérations doit prendre en charge la livraison rapide de plusieurs versions d'une application. Dans le même temps, les tests et le souci de fiabilité doivent être intégrés tout au long du processus de développement.

En fin de compte, les objectifs des deux départements ne sont pas très différents, mais les équipes les abordent sous un angle différent.

Une bonne infrastructure cloud prend en charge les deux rôles lorsqu'il s'agit d'atteindre leurs objectifs en matière de stabilité, de sécurité, d'infrastructure fiable, de provisionnement rapide et de mise en scène efficace.

3. POURQUOI DEVOP MAINTENANT ?

Si le développement et les opérations sont séparés depuis des années, pourquoi cela pose-t-il soudainement un problème ?

Premièrement, le problème n'est pas nouveau. Les projets informatiques importants et complexes ont tendance à être traités avec des méthodologies de gestion de projet purement classiques. Par conséquent, ces projets ont tendance à être moins flexibles pour s'adapter à l'évolution des exigences, les responsables obtiennent des retours très tard et, parfois, les projets peuvent dépasser considérablement les délais et les budgets.

Un développement plus rapide et plus progressif évite ce problème. C'est une bonne idée.

Mais c'est bien plus que cela. C'est essentiel. Chaque organisation est confrontée à la concurrence d'autres organismes qui appliquent des techniques plus agiles. Il y a un changement dans la vitesse des affaires. Des services cloud sont apparus, permettant aux applications d'évoluer rapidement, et de nouvelles start-ups fournissent des services rapidement.

Alors que les grands systèmes n'étaient auparavant déployés que quelques fois par an, on s'attend désormais à ce que les nouvelles fonctionnalités puissent être mises sur le marché en quelques jours ou semaines au maximum.

4. DEVOPS - ACTIVÉ PAR LE CLOUD

À ce stade, toute personne extérieure au département informatique se posera une question évidente. Nous comprenons POURQUOI un développement plus rapide est nécessaire. Nous avons maintenant une idée générale de ce qu'est le DevOps : il s'agit d'un alignement plus étroit des intérêts du développement et des opérations.

Mais la question est : COMMENT cela se produit-il ?

Le DevOps n'est-il qu'une autre stratégie de gestion ? Est-ce encore un autre culte des processus métiers ? Sommes-nous censés croire que nous pouvons changer nos organisations en organisant un ensemble différent de réunions interdépartementales ?

Non. DevOps nécessite une approche différente de la gestion du processus de développement et un changement dans la façon dont l'entreprise gère l'informatique, mais c'est bien plus qu'un simple mot à la mode en matière de gestion.

DevOps est rendu possible par une solide technologie d'infrastructure cloud. En fait, elle est fondamentalement soutenue par ces mêmes technologies qui menacent le statu quo.

D'une part, nous pouvons considérer DevOps comme la mise en œuvre d'un modèle cloud au sein du service informatique. Les technologies cloud incluent l'automatisation et la virtualisation qui permettent à n'importe quelle fonction

être fourni sous la forme d'un service hautement standardisé et massivement évolutif dans le cloud. Le stockage en tant que service et l'infrastructure en tant que service ne sont que deux exemples.

Le développement produit des applications, mais il a besoin de plates-formes sur lesquelles développer et fournir ces applications. Si une organisation adopte le cloud en interne, alors les opérations peuvent être le service cloud qui fournit et prend en charge les plates-formes sur lesquelles le développement effectue son travail.

Étant donné que les services cloud sont standard et peuvent être répliqués et mis à l'échelle, le développement travaille désormais sur une plate-forme fonctionnellement identique à la plate-forme de livraison éventuelle. La livraison incrémentielle rapide des applications est désormais possible.

Un modèle de service cloud établit également une relation claire entre le développement (le consommateur) et les opérations (le fournisseur).

Il fournit un cadre général pour les équipes Dev et Ops, avec des rôles et des responsabilités définis. Associé à une vision des objectifs commerciaux communs, cela conduit à une meilleure communication et collaboration à mesure que les besoins et les incitations s'alignent sur les objectifs commerciaux supérieurs.

Si les opérations sont automatisées dans la mesure du possible, cela réduit le besoin de personnel spécialisé et permet au développement d'adopter une approche en libre-service.

5. FOURNIR CE CLOUD VIA DES CONTENEURS

Lorsque nous parlons de modèle cloud, nous pouvons être un peu plus précis. La plupart des services cloud sont basés sur la « virtualisation », dans laquelle un serveur physique prend en charge plusieurs « machines virtuelles ». Un logiciel appelé hyperviseur crée ces machines virtuelles, de sorte que tout programme semble disposer d'un serveur complet.

Lorsque vous louez un serveur sur Amazon ou Google Compute Engine (GCE), voici ce que vous obtenez.

Mais il existe un moyen plus efficace de procéder. L'exécution d'une application ne nécessite pas une copie complète d'un serveur virtuel. C'est pourquoi l'idée de « conteneurs » a émergé.

Les conteneurs ressemblent à un serveur pour l'application qui s'y exécute, mais sont en réalité allégés avec une surcharge beaucoup plus faible qu'un serveur virtuel complet. Ils fournissent la puissance de traitement du processeur, la mémoire, l'accès au réseau et le stockage sur disque.

Les conteneurs sont exactement le type de service que le développement attend des opérations. Ils offrent un espace pour le travail de développement, peuvent fournir les bibliothèques et composants requis

pour les applications, et une application peut être facilement déplacée vers un environnement opérationnel complet prenant en charge la même technologie de conteneur.

Les conteneurs signifient que le même environnement peut être utilisé pour le développement et la livraison – exactement ce qu'exige DevOps. Ce sont des blocs standardisés dans le cycle de vie du développement : le développeur prend le conteneur, et une fois l'application opérationnelle, une application peut être déplacée vers les autres phases de son cycle de vie.

Les conteneurs peuvent même être adaptés à des tâches particulières, telles que Java, Java EE, Ruby, etc.

6. DE QUOI AVONS-NOUS BESOIN D'AUTRE ?

Les conteneurs ont connu un essor rapide au sein de la communauté open source et sont rapidement devenus le standard de facto pour la grande majorité des services cloud.

Aujourd'hui, la technologie Linux permet l'allocation de ressources telles que le processeur et la mémoire qui peuvent être isolées et distribuées entre les conteneurs. Le pool de ressources sous-jacent peut provenir de machines physiques ou virtuelles. Ils constituent un élément essentiel de Linux, mais ils nécessitent une standardisation, une gestion et une automatisation. Ils nécessitent également la mise en œuvre de considérations de sécurité minutieuses pour les environnements multi-locataires.

Pour que les opérations fournissent un service efficace de type cloud, la gestion des conteneurs doit être automatisée autant que possible, afin que les conteneurs puissent être fournis en libre-service et gérés de manière indépendante pour la création et la livraison des applications.

Cela nécessite un format standard pour le packaging et la distribution des applications avec dépendances. Le projet Docker est aujourd'hui un leader incontesté dans la fourniture d'un tel format.

Docker utilise un concept de superposition d'images pour permettre aux opérations de créer et de gérer des conteneurs, et permet au développement d'y insérer du contenu - l'application.

Docker propose un format de packaging qui devient rapidement populaire et les conteneurs créés au format Docker fournissent une norme, de sorte qu'ils peuvent facilement être mis en œuvre au sein d'une organisation ou même entre des organisations travaillant sur des projets communs.

Mais il arrive un moment où ces projets deviennent trop complexes pour les outils de base de Docker.

Dans un environnement de grande envergure, les organisations auront besoin d'un environnement de plateforme en tant que service sophistiqué, dans lequel leurs conteneurs peuvent être pris en charge tout au long du cycle de vie complet des applications - développement, tests, assurance qualité, production, etc. un environnement capable de connecter et de coordonner des applications exécutées dans plusieurs conteneurs.

OpenShift, disponible pour Red Hat Linux, est un tel service. Il maintient les services sous-jacents à l'application et les fait évoluer si nécessaire.

Essentiellement, OpenShift est une structure de gestion parfaitement adaptée aux conteneurs. Il utilisait la technologie des conteneurs dès le début et est maintenant amélioré pour permettre la création et le déploiement fluides de conteneurs au format Docker sur la plate-forme.

7. CONCLUSIONS

DevOps est un grand idéal. Il peut même s'agir d'une question de vie ou de mort pour l'entreprise. Une collaboration accrue est essentielle entre les opérations et le développement si les organisations veulent répondre, rapidement et à grande échelle, à l'environnement commercial actuel.

Mais toute adoption de DevOps échouera à moins qu'elle ne réponde à trois critères :

1. Une compréhension de la nécessaire séparation des compétences qui a conduit à la création de Fonctions de développement et d'exploitation
2. Une technologie appropriée pour assurer la collaboration nécessaire tout en préservant Vivre la séparation des compétences
3. Une culture habilitante et une technologie habilitante comme fondement

Les conteneurs Linux, Docker et OpenShift se combinent pour fournir un environnement dans lequel les compétences nécessaires peuvent être déployées de la bonne manière.

Les conteneurs permettent le DevOps tout en préservant les préoccupations fondamentales – et nécessaires – du développement et des opérations.