

Amazon Web Services professionnalise le développement Cloud

Positionnée comme plate-forme technologique cloud, l'avenir d'Amazon Web Services passe immanquablement par les développeurs. Toutefois, les technologies cloud avec leur approche de micro-services et d'API (ou composants ouverts et exposables) amènent vite à une explosion des modules. En outre, une approche professionnelle d'entreprise s'impose pour contrecarrer l'image de bidouilleurs du web des développeurs cloud.

Conscient de ces phénomènes, AWS tente d'y remédier en apportant à la fois de la méthode et des outils. L'éditeur annonce donc de nouveaux services pour maîtriser les tests et les déploiements d'applications cloud. Des solutions qui s'inscrivent dans une approche de gestion du cycle de vie des applications (ou ALM pour Application Lifecycle management) : développer, concevoir et tester, superviser et analyser, et déployer.

Bref : la panoplie pour concevoir et manager des applications cloud d'entreprise dignes de ce nom. Le but consiste aussi à automatiser les maximum de tâches de ce type pour permettre aux développeurs de se consacrer pleinement à son application, "sans se soucier de la plomberie", mais tout en ayant un œil sur l'ensemble des processus.

Automatiser et contrôler le déploiement applicatif

« Comment exploiter pleinement la puissance de cette nouvelle plate-forme technologique? » lance Andy Jassy, senior vice-président chez Amazon Web Services. « Nous avons examiné comment nous concevons et déployons du logiciel chez Amazon Web Services. En fait nous utilisons Appolo, notre service interne de déploiement qui a servi à réaliser plus de 50 millions de déploiements au cours des douze derniers mois, soit plus de 95 par minutes. Nous proposons donc désormais ce service à nos clients sous le nom d'AWS CodeDeploy. »

AWS CodeDeploy se propose d'automatiser les déploiements et les mises à jour, y compris à grande échelle, des applications sur EC2. Désormais le développeur peut rapidement et simplement mettre à jour ses applications sur une ou plusieurs milliers d'instances EC2.

Le module peut effectuer des mises à jour par roulement afin de ne pas interrompre l'exécution des applications pour les utilisateurs. Son intégration avec AWS Auto Scaling assure l'adaptation dynamique de l'infrastructure, tandis que des contrôles de maintien en conditions opérationnelles sont effectués avec possibilité de retour à l'état précédent si nécessaire. Enfin, ses API permettent de rendre CodeDeploy compatible avec une autre solution du marché assurant une action de la chaîne du déploiement. D'ailleurs, plusieurs acteurs du marché proposent déjà des intégrations comme GitHub, CircleCI, Atlassian, Codeship, Solano CI, Travis CI ou CloudBees. La solution est totalement indépendante des langages de programmation utilisés.

Une console centralisée (celle d'AWS ou d'une autre solution) permet au développeur de lancer et de contrôler les processus, mais également d'afficher les historiques détaillés des déploiements. Le développeur peut stopper l'un des déploiements, sans avoir à se loguer sur l'instance concernée.

Cerise sur le gâteau : AWS CodeDeploy est gratuit pour les clients qui ne paient que les ressources déjà nécessaires pour le stockage et l'exécution de leurs applications.

Un code source mieux maîtrisé jusqu'à la mise en ligne

Pour compléter CodeDeploy qui assure les fonctions de supervision, d'analyse et de déploiement, AWS propose CodeCommit et CodePipeline.

Sécurisé et hautement disponible, le service **AWS CodeCommit** assure le contrôle du code source d'une application via l'hébergement de référentiels Git. L'utilisateur peut aussi utiliser ses propres outils pour accéder aux référentiels Git stockés dans CodeCommit.

Physiquement proche des environnements de tests et de production, CodeCommit n'impose aucune limite de taille sur les référentiels et fichiers. Ce service collaboratif sera disponible début 2015.

AWS CodePipeline automatise le déploiement continu de nouvelles versions de fonctions ou services auprès des utilisateurs. Cette automatisation de la conception, des tests et de la gestion des versions permet de standardiser ces tâches et détecter les modifications de code et les bogues simples à corriger. Le développeur utilise l'interface graphique de WS CodePipeline pour modéliser les différentes étapes de mise à jour de l'application. Les règles peuvent contenir –entre autres- des éléments temporels, ou des validations manuelles si nécessaire. Le workflow peut programmer des tests unitaires ou d'intégration conditionnel avant déploiement, le tout totalement automatisé. CodePipeline assure le processus de bout en bout, mais peut également intégrer des solutions tierces à toute étape.

Ce module sera également disponible début 2015.

De l'event Processing à portée du programmeur Lambda ?

« Quelles sont les bases de la programmation? » interroge Werner Vogels, CTO d'Amazon. « Des fonctions et des données entre lesquelles on trouve des interactions déclenchées à travers par la détection d'événements. Et la magie opère à l'intersection des fonctions, des événements et des données. Et pendant trente ans, nombre d'entre nous avons travaillé en ce sens pour dompter les données via des événements et des fonctions : concevoir des petites fonctions rapides, ajouter des fonctionnalités simplement et en toute sécurité, et maintenir dynamiquement les données à jour. Aujourd'hui, la situation est bien plus complexe avec une pile complète (stack) pour chaque fonction, et plus onéreuse avec des fonctions très peu utilisées. Pour y remédier, nous proposons AWS Lambda ! »

Présenté comme « un service applicatif basé sur la gestion d'événements pour des applications dynamiques », **AWS Lambda** peut recevoir des événements provenant de multiples sources et de tout format : notifications S3 (stockage), flux DynamoDB (base de données NoSQL), événements Kinesis (traitement en temps réel de gros flux de données), ou tout événement...

Dans la lignée de l'event processing, AWS Lambda permet d'associer une fonction et donc des actions lorsqu'un événement survient. Le principe consiste à « écouter » ce qui se passe dans l'environnement applicatif (ou système) pour déclencher automatiquement des opérations. En outre, AWS Lambda procure les ressources nécessaires de façon totalement transparente.

Enfin, le code n'étant activé que lorsque l'événement se produit, l'utilisateur en paie que pour cette durée d'exécution, sans avoir à se préoccuper ou à réserver ou monopoliser des ressources.

Le service est tarifé à la requête de une par mois à plusieurs milliers par seconde : 20 centimes de dollar par million de requête, 0,00000021 centimes de dollar pour 100 millisecondes d'exécution avec 128 Mo de mémoire. Des prix qui semblent très raisonnables, mais à étudier de près selon les scénarios. Et pour rendre l'offre attractive, AWS propose gratuitement à ses clients jusqu'à 3,2 millions de secondes d'exécution pour un million de requête par mois.

AWS Lambda est disponible en preview [ici](#).

A lire aussi :

[Cloud : LinkByNet veut faire cohabiter AWS, Azure et Numergy](#)

[Cloud : AWS épouse Docker avec EC2 container service](#)