

10 outils Open Source indispensables pour maîtriser le Cloud

Dans le monde du Cloud, assez peu de gens connaissent les noms de OsV ou Salt, Ansible ou Puppet et pourtant l'environnement Open Source dans ce domaine **ne se cantonne pas à OpenStack**. Il est vrai que ce projet a attiré les lumières avec l'adhésion de plusieurs grands noms de l'IT comme HP, IBM, Red Hat, etc.

Il existe cependant d'autres projets qui sont parfois complémentaires ou concurrents mais qui ont pour vocation d'optimiser les infrastructures et plus globalement la stratégie Cloud. Nous avons ici répertorié 10 outils Open Source qui pourront trouver leur place dans l'arsenal indispensable pour administrer, gérer, piloter le Cloud. Pour une meilleure lecture, nous avons segmenté ces outils en **trois thèmes, les OS, les outils de développement et enfin les solutions d'orchestration**.

I- Des OS entièrement dédiés au Cloud

Le développement du Cloud a permis à plusieurs systèmes d'exploitation de se créer avec, pour certains, une orientation vers les machines virtuelles ou les conteneurs.

1- CoreOS mêle virtualisation et conteneurs au sein des Cloud d'entreprise

CoreOS est un système d'exploitation Linux conçu spécifiquement pour les infrastructures serveur massives. Il se veut idéal pour les offres Cloud. L'éditeur se trouvant derrière ce projet s'est appuyé sur les stratégies et techniques appliquées par les grandes sociétés du web (Google, Facebook ou encore Twitter) pour mettre au point cette solution.

Cette offre est tout d'abord conçue pour être **administrée depuis un point unique**, quel que soit le nombre d'instances. L'OS se veut minimal et aisé à maintenir. Ainsi, lors de l'application d'une mise à jour, c'est l'ensemble du système qui est modifié.

Le principe appliqué ici est simple : **deux partitions** sont proposées pour CoreOS. Lors de l'arrivée d'une mise à jour, l'OS patché est installé sur la partition inutilisée. Lors du prochain redémarrage de la machine (ou l'utilisation d'un reboot à chaud, via kexec), c'est cette dernière qui sera employée. La partition de démarrage utilisée précédemment sert alors de système de secours, permettant un retour aisé vers l'ancienne version de l'OS.

Le système n'a donc plus à gérer une foule de petites mises à jour. Il se met à niveau dans son ensemble. Les partitions système ne sont accessibles qu'en lecture, permettant ainsi de renforcer le niveau de sécurité de l'ensemble.

Technologie clé : les conteneurs

Tout ceci est rendu possible par la présence **d'un OS simplifié au maximum**, afin de ne pas ralentir l'infrastructure IT par des mises à jour qui présenteraient une taille trop imposante. Ses besoins de base en mémoire vive restent mesurés (114 Mo) et son temps démarrage se compte en

secondes.

En contrepartie, CoreOS ne saura pas jongler avec des dépendances spécifiques. Mais il n'en a pas besoin. En effet, toutes les applications doivent être packagées au sein de conteneurs Linux, via **Docker**. Les dépendances seront donc prises en compte par les conteneurs et non par CoreOS.

Les paramètres système sont centralisés par un outil qui se charge de découvrir et configurer les nœuds CoreOS au sein d'un ensemble de machines. Une technologie prête pour les clusters... et le cloud. Les mises à jour sont assurées par l'outil CoreUpdate, qui propose tableaux de bord et rapports pour suivre le fonctionnement d'une infrastructure de serveurs. Des outils spécifiques au monde des clusters sont également en cours de test.

Les serveurs CoreOS pourront être physiques ou virtuels. Il est ainsi possible de déployer et gérer des instances sur **les Cloud d'Amazon (EC2), de Google (Compute Engine)** ou encore de **Rackspace**. CoreUpdate gèrera sans distinction les instances physiques ou virtuelles du système.

CoreOS pourra donc à la fois servir à déployer des conteneurs sur des serveurs physiques ou virtuels. C'est bien évidemment dans la configuration « **virtualisation + conteneurs** » que cette solution donnera les résultats les plus probants, en alliant les bénéfices des deux mondes.

Au cœur de l'offre *bare metal* de Rackspace

Les mises à jour de base de CoreOS sont gratuites. CoreOS Inc. propose des offres de support « à la Red Hat » autour de sa solution. « Managed Linux » (de 1,25 à 10 dollars par serveur et par mois) permet de disposer des mises à jour de l'OS et d'outils offrant de disposer de plus de contrôle sur la phase d'update des serveurs. « **Premium Managed Linux** » (de 2,10 à 84 dollars par serveur et par mois) ajoute à ceci un support étendu (téléphone, chat et e-mail).

Le cas d'utilisation de cette technologie le plus marquant à ce jour est Rackspace, qui emploie CoreOS pour animer les machines de sa plate-forme OnMetal. DEIS l'utilise également au sein de son PaaS open source. Enfin, MemSQL Inc. a mis en place un cluster de 107 machines pilotées par CoreOS, qui sont exploitées pour des opérations de tests sur la base de données de l'éditeur.

2- OSv, un système d'exploitation Cloud dédié aux machines virtuelles

Dans le monde du cloud, la virtualisation tient un rôle essentiel. C'est elle qui permet d'obtenir la flexibilité requise pour répartir les tâches entre les serveurs. Afin de réduire la charge des machines, il est possible d'opter pour un hyperviseur de bas niveau, s'appuyant sur un OS minimal. Cette approche, [OSv](#) la transpose aux machines virtuelles (VM).

OSv est en effet un système d'exploitation minimaliste, dédié spécifiquement aux environnements virtualisés. Il n'intègre aucun pilote pour du matériel physique et ne propose aucun support multiutilisateur. Une offre adaptée au monde des machines virtuelles, où un seul logiciel est en général installé au sein d'une VM.

Posix et Java

Les applications tournent entièrement dans l'espace mémoire réservé au kernel. Ceci élimine l'impact lié aux changements de contexte, courants dans le monde Unix traditionnel. La pile de

base s'appuie en grande partie sur FreeBSD. Elle comprend un noyau, une couche réseau et le système de fichiers ZFS.

Côté applications, OSv est compatible avec les titres Posix et Java, via l'intégration en standard d'une JVM. Un choix judicieux pour les entreprises, qui utilisent massivement Java. Ceci ouvre également la voie à l'installation d'un serveur d'applications Java (qui, lui, sera adapté au lancement de plusieurs logiciels au sein d'une même VM).

OSv permet de créer des images virtuelles compatibles avec les hyperviseurs KVM, VirtualBox, VMware (ESXi et Workstation) et Xen. Elles pourront être également directement déployées sur le cloud public EC2 d'Amazon, ainsi que sur le Compute Engine de Google.

L'outil Capstan offre de créer des machines virtuelles OSv, avec la même facilité que celle apportée par Docker dans le monde des conteneurs Linux. Notez que cet outil est accessible sous Windows, OS X et Linux. La gestion des VM passe au choix par une console SSH ou l'utilisation d'une API REST.

Un OS turbo

Allégé au maximum, ce système d'exploitation se veut particulièrement rapide. Sous memcached, les performances sont ainsi jusqu'à quatre fois supérieures à celles obtenues à l'aide d'une VM Linux.

Le benchmark Spec JVM montre lui aussi quelques gains en vitesse. Notez que la machine virtuelle Java intégrée à cette offre devrait prochainement être optimisée, via une communication directe avec le kernel du système.

Le test Netperf montre des gains pouvant aller jusqu'à 77 %. Cette offre peut ainsi viser le marché des solutions NFV (*Network Function Virtualization*), qui requière une disponibilité à 99,9999 %, des temps de latence inférieurs à 20 microsecondes et une bande passante supérieure au Tb/s.

Un long chemin restera toutefois à parcourir. OSv n'est en effet aujourd'hui disponible qu'en version alpha. L'intégration de fonctions de monitoring et d'audit est la prochaine étape attendue. La livraison d'appliances prêtes à l'emploi (JBoss, MongoDB, Nginx, Redis, etc.) est également sur la feuille de route de ce produit. OSv sera alors très proche de la version bêta.

Notez que cette solution n'est actuellement accessible qu'en version x86 64 bits. Son adaptation aux serveurs ARM 64 bits et IBM Power est toutefois d'ores et déjà programmée.

3 -SmartOS : le clone libre de Solaris dédié au Cloud

Lors du lancement de Solaris 11 fin 2011, Oracle déclarait proposer le premier système d'exploitation Cloud du marché (voir l'article « [Oracle présente la première version de Solaris développée en interne](#) »).

Toutefois, c'est oublier un peu vite que le premier système conçu pour le Cloud n'est pas signé Oracle, mais Joyent avec SmartOS apparu dès l'été 2011 (voir « [Joyent lance son OS Cloud open source, SmartOS](#) »). Ironiquement, cette offre utilise les mêmes technologies que Solaris, puisqu'elle s'appuie sur un noyau Illumos dérivé d'OpenSolaris.

ZFS + DTrace + KVM + Zones

À ce jour, SmartOS est le seul hyperviseur de bas niveau (type 1) basé sur le code issu d'OpenSolaris. Cette offre est conçue pour pouvoir être déployée sur un grand nombre de serveurs, dont elle pilotera les machines virtuelles. Quatre technologies clés sont présentes dans cette offre : le système de fichiers ZFS, les sondes DTrace, l'hyperviseur KVM et les conteneurs Zones. À noter également, l'intégration de la technologie de virtualisation des ressources réseau (SDN) Crossbow.

Les Zones restent limitées à l'utilisation de Solaris ou d'OpenSolaris (KVM permettra pour sa part de faire fonctionner d'autres OS comme Windows ou Linux). Elles se montrent toutefois plus légères que la virtualisation traditionnelle. Un usage classique des Zones, popularisé au sein de Solaris, consiste à les combiner avec la virtualisation, en faisant tourner plusieurs OS en conteneurs ou au sein de chaque machine virtuelle. Le meilleur des deux mondes.

Notez que des solutions comme FiFo sont adaptées spécifiquement à l'administration de clouds opérés par SmartOS.

Une alternative à Linux ?

Les partisans du monde Linux mettront en avant le fait que le kernel de Linus Torvalds permet aujourd'hui d'atteindre un même niveau fonctionnel. Le système de fichiers Btrfs pourra en effet remplacer efficacement le ZFS, Kprobes les sondes DTrace et le couple KVM + LXC l'ensemble KVM + Zones. Il est même possible d'utiliser ZFS et DTrace au sein de Linux.

Toutefois, le niveau de maturité des deux offres n'est pas le même. ZFS, DTrace et Zones sont tous trois présents de longue date au sein de Solaris. Linux peut toutefois opposer ici le fait qu'il dispose d'un plus grand nombre d'utilisateurs, permettant de stabiliser plus rapidement les solutions qu'il propose.

Tout est donc question de choix. [En 2012](#), OVH avouait utiliser des technologies issues de Solaris au sein de ses infrastructures. L'hébergeur proposait alors à ses clients de démarrer leurs serveurs sous SmartOS. Faute d'intérêt, cette solution n'a plus guère évolué depuis chez OVH. Elle reste pourtant une alternative intéressante au monde Linux.

Un Cloud centré sur SmartOS

Si elle reste embryonnaire chez OVH, l'utilisation de la solution de virtualisation SmartOS est au cœur du [cloud public](#) et de l'offre [Smart Data Center](#) de Joyent. La société propose également divers services autour de cet OS, comme des formations, qui devraient bientôt être accessibles sur la Toile.

Joyent reste assez peu présent en Europe, l'essentiel de ses clients se trouvant en Amérique du Nord. Si la société demeure plutôt discrète, sur son blogue comme sur les réseaux sociaux, elle reste très active, les modifications apportées à SmartOS étant régulières.

II- Orchestration et administration du Cloud

Devant la complexité du déploiement des solutions d'infrastructure-as-a-service (IaaS), plusieurs solutions sont apparues. Orchestration et administration sont au menu de cette salve d'outils Open Source.

1- Puppet mène à la baguette les infrastructures Cloud

[Puppet](#) est un outil Open Source permettant de gérer la configuration des serveurs présents dans des infrastructures IT de grande taille, telles celles à l'œuvre au sein de Cloud, qu'ils soient privés, publics ou mixtes.

Puppet gère la mise en place **des serveurs, des ressources réseau et des systèmes de stockage** présents dans un datacenter. Il peut également prendre en charge des ressources virtualisées, par exemple des machines virtuelles Amazon EC2.

Écrit en Ruby, Puppet utilise son propre langage dédié (DSL pour *Domain specific language*), lequel permet de créer des fiches de configuration. Un agent récupère les informations système en provenance des serveurs et crée un catalogue des ressources disponibles.

Puppet met en œuvre des concepts **d'architecture pilotée par des modèles**, qui requièrent peu de connaissances en programmation. Toutefois, l'IDE Geppetto – basé sur Eclipse – permet de faciliter la mise au point des 'manifestes' Puppet.

Compatible avec quasiment tous les OS

La force de Puppet réside dans son très large support des systèmes d'exploitation présents sur le marché. Et ce aussi bien pour les serveurs administrés que les machines servant à contrôler l'infrastructure.

Puppet met avant tout l'accent sur le monde **Linux**, avec le support des offres de Red Hat (RHEL 5+), SUSE (SLES 11+) et Canonical (Ubuntu LTS 10.04+). Les solutions communautaires sont également de la partie : ArchLinux, CentOS, Debian, Fedora, Gentoo, etc.

Les systèmes **Unix** des grands constructeurs sont supportés : Apple OS X (10.5+), HP-UX, IBM AIX (5.3+), Oracle Linux (5+) et Oracle Solaris (10+). À noter également le support de FreeBSD et d'OpenBSD.

Même le monde **Windows** n'est pas oublié, avec un large ensemble d'OS pris en compte : de Windows Vista à Windows 8 dans le secteur desktop ; de Windows Server 2003 à Windows Server 2012 R2 pour les OS serveur.

Une offre commerciale... pour les grands comptes ?

Puppet Labs propose une version commerciale de son offre, proposant des fonctionnalités étendues, comme la prise en charge du provisionnement de nouvelles ressources et leur administration. Bref, **il ajoute une couche d'orchestration complète** à cet outil de configuration des infrastructures IT.

L'application ne s'installera plus que sous Linux, avec des agents pour Linux, Windows, OS X, AIX et Solaris. Notez qu'un agent est également proposé pour la version 4 de l'offre de Red Hat (et ses dérivés CentOS, Oracle Linux et Scientific Linux).

Côté clients, de grands spécialistes du monde IT utilisent aujourd'hui Puppet Enterprise, comme **Cisco, EMC, Intel, Red Hat ou encore Salesforce et VMware**. Toutefois aucune référence n'est citée en France, même si nous notons la présence de quelques grands clients en Europe : ARM, BT, Nestlé ou encore le CERN. Signalons également l'adoption de cette solution au sein de certaines places financières, comme le New York Stock Exchange et le London Stock Exchange.

2- Chef donne ses recettes pour administrer le Cloud

[Chef](#) est un outil d'orchestration des infrastructures de Cloud Computing. Livré sous licence Open Source Apache, cette offre monte en puissance, grâce à sa capacité à piloter de nombreux Cloud et systèmes d'exploitation. Chef est aujourd'hui utilisé par certains grands noms comme Bloomberg, Facebook ou encore Yahoo.

Appliqué aux infrastructures serveur de l'entreprise, le Cloud offre bien des avantages. La mise en place d'un Cloud privé permet en effet de faciliter le déploiement et la gestion des machines virtuelles ou applications. Fini le silotage et bienvenue dans le monde de la ressource accessible à la demande, l'IaaS (Infrastructure as a Service).

Entrée : les serveurs physiques

Reste que ceci ne fait que déporter le problème d'administration de l'infrastructure à d'autres niveaux. Il faut en effet gérer la configuration et la mise en place des serveurs, physiques ou virtuels.

C'est ici qu'entre en jeu Chef. Il permet de créer des recettes (via un langage de programmation métier), utilisables pour déployer les serveurs. Il est ainsi possible de définir le rôle qu'aura un nœud physique au sein d'un Cloud : compute, stockage, réseau, etc.

L'offre SUSE Cloud emploie Chef pour assurer la configuration des nœuds composants une infrastructure de Cloud privé (voir notre présentation « [SUSE Cloud en avant-première et en images](#) »). HP l'utilise également au sein de sa solution de Cloud public.

Plat de résistance : les serveurs virtuels

Mais ce n'est pas tout. Chef peut aussi provisionner (et déprovisionner) des machines virtuelles. Vous pourrez ainsi définir des **recettes permettant de préconfigurer et déployer** des serveurs virtuels, voire des conteneurs. Apache pour un serveur web, MariaDB pour une base de données, ou encore Hadoop pour du Big Data.

Il est à noter que Chef peut déployer ses recettes sous une foultitude d'OS. Linux est assez largement supporté (CentOS, Debian, RedHat, Ubuntu...), mais OS X et Windows sont également de la partie. Certaines solutions d'orchestration d'infrastructures Cloud sont même en mesure de réutiliser les recettes Chef. C'est ainsi le cas de Juju, l'offre liée à Ubuntu (voir notre article « [Open World Forum 2011 : l'Ubuntu veut révolutionner la gestion des services](#) »).

Dessert : les infrastructures hybrides

Si un Cloud est vu comme une ressource cohérente et uniforme, il peut s'appuyer sur des serveurs employant des technologies différentes, voir mixer des Cloud différents (infrastructure privée et Cloud publics par exemple).

Chef est capable de prendre cette problématique en compte. Il supporte en effet les plates-formes Cloud d'Amazon (EC2), Google (Cloud Platform), Microsoft (Azure) et Rackspace. Il sera en mesure de déployer ses recettes et donc de provisionner des serveurs virtuels sur toutes ces offres de Cloud public.

L'administrateur système a ainsi la possibilité de piloter l'ensemble de sa brigade de serveurs depuis une unique station de travail et un seul outil. À charge pour lui de définir les rôles des serveurs, puis de créer les recettes permettant de couvrir les différents cas possibles. Il pourra pour cela s'appuyer sur le vaste ensemble de recettes déjà existantes.

Un chef à la demande

L'automatisation des infrastructures suppose l'acquisition de certaines connaissances spécifiques. Notez que des SSII comme AlterWay proposent aujourd'hui une formation dédiée à Chef. Un sujet que nous avons évoqué dans un de nos précédents articles : « [Solutions Linux 2014 : L'Open Source raisonne Cloud et rêve des métiers](#) ».

Notez que Chef est proposé en version commerciale par son créateur. Soit sous forme on premise, soit en mode SaaS.

3- Salt, une pincée de dynamisme pour gérer les infrastructures Cloud

Dans le monde de la gestion des infrastructures serveur, il y a des marionnettes (Puppet), des chefs cuistots (Chef) et **Salt**, l'administrateur turbo des infrastructures de grandes tailles.

Salt a une particularité : il est le champion des datacenters ultra-massifs. Il se montre en effet capable d'administrer des infrastructures serveur comprenant **plusieurs dizaines de milliers de machines**. Une offre définitivement taillée pour le Cloud.

Un bus de communication...

À la base, Salt est **un outil de gestion des serveurs à distance** : administration et lancement de tâches définissent son quotidien. Salt mise sur sa vitesse de fonctionnement. À cet effet, il utilisait jusqu'alors le système de messagerie de haute performance ZeroMQ. Toutefois, depuis cet été, il offre sa propre solution de communication, **RAET** (Reliable Asynchronous Event Transport Protocol).

Il s'est également ouvert aux tâches de collecte d'informations de configuration et de découverte de services, ce qui le propulse sur le marché **des outils de configuration de serveurs** (à l'instar de Chef et Puppet). « *Salt propose un bus de communication dynamique pour les infrastructures, qui peut être utilisé pour l'orchestration, l'exécution à distance, la gestion de la configuration et bien plus encore* », résumant les promoteurs du projet.

... et des modules

Salt se veut aujourd'hui un outil générique pour interagir avec un large ensemble de serveurs. Il propose des modules de gestion de la configuration, d'exécution des tâches à distance, des solutions permettant aux machines de retourner des informations à Salt, etc.. Ce système de modules est extensible à souhait, le tout s'architecturant autour du bus de communication à haute vitesse proposé par Salt.

De quoi gérer et automatiser une infrastructure IT complète. Une API spécifique permet d'accéder aux services offerts par Salt. **Pepper** propose une librairie de fonctions et de scripts facilitant la mise en œuvre de cette solution. Un outil encore à l'état de version de test. De nombreux projets se sont greffés autour de Salt afin d'en étendre les fonctionnalités.

Des utilisateurs prestigieux, dont Orange

[SaltStack](#) est une société fondée par les créateurs de Salt. Elle propose **SaltStack Enterprise**, une offre commerciale, ainsi que diverses offres de services (intégration, formations, certifications, etc.).

Mais qui utilise aujourd'hui Salt ? De très grands comptes, comme Apple, LinkedIn, la Nasa ou encore le département de la Défense américaine. En France, une référence de choix est au tableau d'honneur de la société : Orange. À noter, Salt est utilisé par CloudFlare, un fournisseur de CDN.

Nous retrouvons également cette solution à l'œuvre chez certains fournisseurs de Cloud, comme Rackspace ou HP, qui l'utilise pour ses Cloud Services.

4- Ansible : JSON et SSH au service du Cloud

Ansible est une plate-forme de configuration et d'administration des ordinateurs, accessible sous licence Open Source. Cette offre permet de gérer l'inventaire et la configuration des serveurs, le déploiement des logiciels et le lancement de tâches.

D'autres offres existent dans ce secteur : **Chef** s'appuie sur le déploiement de recettes permettant de configurer les machines ; **Puppet** propose une puissante approche pilotée par des modèles ; **Salt** mise sur son bus de communication ultra rapide.

Ansible parie pour sa part sur la simplicité. Pour mettre en place cette solution, il suffit de disposer de SSH sur chacune des machines. Un outil quasi systématiquement mis en place sur les serveurs, afin de proposer un accès distant sécurisé. Aucun agent n'est nécessaire sur les serveurs. Python devra toutefois être installé, Ansible étant écrit dans ce langage de programmation.

Organisation via JSON et YAML

Minimaliste par design et sécurisé par défaut (via l'utilisation de SSH), Ansible se veut simple d'emploi et facile à mettre en place. Cette offre sait toutefois aussi se montrer puissante dans ses fonctionnalités.

Ansible déploie de façon dynamique les outils nécessaires à la gestion des serveurs. Les modules peuvent être écrits dans n'importe quel langage. Ils communiquent entre eux en employant le format de données **JSON**, garant de l'interopérabilité entre les différents composants mis en

œuvre.

Cette volonté de standardisation autour de formats reconnus, nous la retrouvons également dans les fichiers de configuration. Le format de données **YAML** est ici mis en œuvre pour écrire les fiches de configuration ou d'orchestration.

Notez que cette offre se montre à l'aise aussi bien avec les infrastructures de serveurs physiques que les solutions virtualisées ou Cloud. **CloudStack comme OpenStack** ne lui font pas peur. Pas plus que les offres de Cloud public, comme **Amazon EC2**.

Twitter et Evernote

Compatible avec la plupart des OS Linux, Ansible a su s'attirer les faveurs du projet Fedora, mais également de l'écosystème Red Hat Enterprise Linux, qui propose cette solution au travers d'EPEL (*Extra Packages for Enterprise Linux*). Ansible se montre capable de gérer la plupart des machines Linux, Unix et Windows composant une infrastructure de serveurs.

Parmi les utilisateurs de cette solution, nous trouvons de grands noms, comme **EA, Evernote, Motorola, Spotify ou encore Twitter**.

Derrière cette offre se cache une société – du même nom – proposant du service : implémentation, consulting, formations, etc. Des activités dédiées à la fois aux entreprises souhaitant utiliser Ansible, et aux spécialistes de cette technologie (au travers de l'offre Ansible Guru).

La solution **Ansible Tower** est également proposée aux clients de la société. Cet outil est une offre clé en main permettant de gérer une infrastructure de serveurs, avec monitoring et tableaux de bord intégrés. Ansible Tower revendique son approche 'presse bouton' qui met Ansible à la portée de tous les administrateurs système. De nombreux grands comptes l'utilisent aujourd'hui, comme **Juniper Networks ou la NASA**.

5- Piston plonge OpenStack dans le monde du business

OpenStack est un projet Open Source permettant de mettre en place des infrastructures cloud (IaaS pour Infrastructure as a Service). Toutefois, son utilisation n'est pas toujours simple.

Divers éditeurs se sont engouffrés dans la brèche, afin de fournir des distributions OpenStack, qui, à l'instar des distributions Linux, permettent de simplifier la mise en œuvre de cette technologie. La start-up **Piston** n'est pas l'acteur le plus connu du marché. Il s'avère pourtant intéressant à plus d'un titre.

Son équipe respire le sérieux. Le Pdg, **Jim Morrisroe**, était le patron de Zimbra ; le directeur technique, **Christopher MacGown**, un des responsables du projet OpenStack chez Rackspace ; la directrice marketing, **Gretchen Curtis**, une ancienne de la Nasa, œuvrant autour du projet Nebula. De nombreux employés de Piston viennent soit de Rackspace, soit de la Nasa, les deux acteurs à l'origine du projet OpenStack.

Une offre unifiée

Le principal produit de la société est **Piston OpenStack**, actuellement accessible dans sa troisième

livrée. Cette offre reste proche d'OpenStack, tout en apportant des outils permettant de mieux gérer une infrastructure cloud. L'approche choisie par l'éditeur se veut également totalement intégrée et 'convergée'. Chaque serveur peut ainsi **indifféremment fournir du calcul, du stockage ou du réseau**.

À cet effet, un OS Linux minimaliste est installé sur chaque machine. Pilotable à distance, il permet de définir à la volée le rôle des serveurs. Notez que les machines virtuelles pourront accueillir des hôtes Linux ou Windows.

Plus d'intégration signifie moins de différences dans le matériel. Piston OpenStack s'appuie ainsi quasi exclusivement sur des serveurs x86 classiques. Un cluster pourra comprendre entre 5 et 250 nœuds. Dans sa taille maximale, un cluster pourra proposer jusqu'à 4 Po de stockage et 7 500 CPU virtuels. Les outils de la jeune pousse permettront de gérer **jusqu'à 30 000 nœuds physiques**, soit un grand nombre de clusters.

Chaque nœud devra au minimum comprendre 24 Go de RAM (Piston en recommande 96 Go ou plus), une ou plusieurs connexions Ethernet à 10 Gb/s, un lien Ethernet Gigabit pour gérer le serveur et au moins deux disques durs (un ou plusieurs SSD sont recommandés). Le tout devra bien entendu être animé par un processeur x86 supportant la virtualisation matérielle.

Des API standard

Côté interfaces de programmation, Piston joue la carte de la compatibilité avec les standards en vigueur. Ainsi, ce sont les API **d'OpenStack Icehouse** qui sont aux commandes de Piston OpenStack 3.5, dernière version en date de l'offre de l'éditeur :

- Compute : Nova.
- Image : Glance.
- Storage : Cinder (block) et Ceph (object).
- Network : Neutron.
- Identity : Keystone.

À noter également, le support de la technologie TXT d'Intel (Trusted Execution Technology), laquelle permet de garantir l'intégrité du serveur dans ses couches les plus basses (Bios comme OS de base). La mise à jour de Piston OpenStack se fait à chaud, sans rupture de service.

Bien entendu, la société propose **des offres de support en 24 x 7**. La plupart des grands comptes utilisant les solutions de Piston sont inconnus de ce côté de l'Atlantique. Il est vrai qu'ils sont actuellement concentrés sur le territoire américain. Reste que la jeune pousse commence à s'étendre en dehors des frontières de son pays d'origine. Elle compte ainsi plusieurs clients en Asie, et un beaucoup plus près de nous, l'opérateur télécoms suisse **Swisscom**.

III- Programmation et développement

Après les OS et les outils d'orchestration du Cloud, voici deux projets qui sont devenus incontournables dans le domaine de la programmation et du développement.

1- Vagrant s'étend de la virtualisation desktop au monde du Cloud

Les machines virtuelles restent un des outils préférés des développeurs. Elles permettent en effet de tester une application dans un environnement de production simulé, voir sur un pool de serveurs virtuels.

Toutefois, créer et configurer ces machines virtuelles peut rapidement devenir fastidieux. C'est ici qu'entre en jeu le projet Open Source **Vagrant**, lancé en 2010 par Mitchell Hashimoto. Initialement, cet outil permettait de créer des machines virtuelles dédiées à VirtualBox, en conjonction avec des solutions de gestion de la configuration comme Chef, Puppet et Salt (dont nous avons parlé précédemment, voir « [Open Source : Chef donne ses recettes pour administrer le Cloud](#) », « [Orchestration : Puppet mène à la baguette les infrastructures Cloud](#) » et « [Salt : une pincée de dynamisme pour gérer les infrastructures Cloud](#) »).

Toutefois, cette offre s'est rapidement ouverte à d'autres solutions. Elle propose en effet le support des technologies de virtualisation de **VMware et Microsoft**, et permet également de déployer les machines virtuelles sur le cloud **EC2 d'Amazon**.

Depuis la version 1.6, Vagrant peut même générer des conteneurs **Docker**, qui se veulent plus légers que les VM classiques et ouvrent de nouvelles possibilités, comme le déploiement d'environnements de test – via les conteneurs – dans une machine virtuelle.

Plus de flexibilité

Le support de nouvelles offres de virtualisation passe par des '**Providers**'. Une offre hautement extensible. Cette souplesse, nous la retrouvons également dans les '**Provisioners**', qui se chargent d'automatiser l'installation et la configuration de logiciels sur les machines virtuelles.

En plus de Chef, Puppet et Salt (déjà évoqués), des 'Provisioners' sont proposés pour les scripts en ligne de commande et la solution CFEngine. La liste des 'Providers' comme des 'Provisioners' pourra être étendue par l'entremise de plug-ins.

Vagrant a su en très peu de temps devenir une solution de choix pour les développeurs. Une évolution rendue possible par l'investissement de l'auteur dans sa solution. Mitchell Hashimoto a en effet créé HashiCorp fin 2012, et travaille maintenant à temps complet sur l'évolution de Vagrant.

Un business naissant

Cette société vise le secteur des outils de gestion des configurations hétérogènes : physiques, virtuelles et Cloud. Une activité qui est centrée sur Vagrant, mais également sur des outils Open Source de gestion de la configuration et d'orchestration des serveurs. En plus de ces diverses solutions, la société propose Vagrant Cloud, une offre permettant de partager des environnements Vagrant en un tournemain.

Une offre de **support à 79 dollars par siège** (poste utilisateur) permet de disposer d'une assistance par mail à J+2. Elle reste toutefois limitée aux utilisateurs de solutions VMware desktop (Workstation ou Fusion). D'autres offres de support, plus étoffées, devraient voir le jour prochainement précise la société.

De nombreux grands comptes utilisent aujourd'hui Vagrant : O'Reilly, Mozilla ou encore Yammer.

Plus près de nous se trouvent des acteurs comme la BBC ou Nokia. Certaines sociétés contribuent également de façon importante à l'évolution de ce projet. Nous trouvons quelques grands noms du monde de l'infrastructure IT, comme Fastly (CDN) ou SoftLayer (Cloud).

2- Jenkins, l'outil d'intégration continue frappe aux portes du Cloud

[Jenkins](#) est né d'une dispute. Celle entre la communauté des logiciels libres et Oracle, suite au rachat de Sun Microsystems par la firme en 2010. 'fork' d'Hudson, Jenkins est **un outil d'intégration continue**.

Sous ce terme barbare se cache l'une des techniques les plus représentatives de l'Extreme Programming : les développeurs déposent leurs modifications dans un creuset unique, les compilations sont gérées automatiquement et des tests automatiques sont appliqués au binaire créé.

De quoi s'assurer que 1/ le cycle de développement sera raccourci par l'intégration rapide des modifications apportées au code et que 2/ des régressions n'apparaîtront pas dans le projet, tant il vrai que l'objectif premier est d'aller de l'avant... et non en arrière.

Des clients prestigieux

Cette recette, Jenkins l'applique avec brio. À un point tel qu'il est rapidement devenu l'un des chouchous des équipes de programmation. Jenkins propose un atout clé : il est extensible via des greffons. Plus de 1 000 sont référencés à ce jour. Grâce aux **plug-ins**, les entreprises pourront implémenter leurs processus au cœur de Jenkins. Un point très intéressant, en particulier pour les grands comptes.

Jenkins propose également des passerelles vers les systèmes de gestion des versions les plus populaires, comme **CVS, Git, Mercurial ou Subversion**. Le tout est facile à installer et à configurer. Une simplicité d'accès que l'on retrouve également lors de son utilisation. Il est ainsi possible de suivre les tâches de contrôle du code en temps réel par courrier électronique, ou via des flux RSS.

Avec toutes ces qualités, il n'est guère étonnant que Jenkins soit aujourd'hui aussi populaire. Il est utilisé par des centaines de grands comptes, dont – en France – **Alcatel-Lucent, Michelin ou encore Thales**. Notez que si Jenkins n'est pas réservé aux projets **Java**, c'est dans ce secteur qu'il se taille la part du lion, grâce à son support avancé des frameworks de test Junit et TestNG.

Jenkins, la technologie qui PaaS au Cloud

Intégration continue, développement agile, travail en équipe sur de gros projets. Tout ceci fleure le Cloud à plein nez. Le PaaS plus précisément (Platform-as-a-Service). Une étape que ce projet a allègrement franchie, en grande partie **sous l'impulsion de CloudBees**, qui en a fait un des piliers de sa plate-forme de développement en mode Cloud.

Si l'éditeur distribue des offres Jenkins on premise (Jenkins Enterprise et Jenkins Operations Center), il a également proposé une rupture avec sa solution DEV@cloud. L'idée est ici d'offrir aux développeurs des services leur permettant de mener à bien leurs projets plus aisément, grâce aux avantages apportés par le Cloud : haute disponibilité et capacité à traiter des projets de grande taille, en adaptant à la demande les ressources disponibles.

Là encore, c'est une opportunité pour les entreprises, dont les programmeurs ont pour vocation de créer des logiciels et non d'opérer une infrastructure de développement.

Un tour de 'PaaS PaaS'

Le choix de Jenkins pour le PaaS de CloudBees n'est pas innocent. Ce dernier dispose en effet en standard de fonctionnalités qui le rendent compatible avec un usage en mode Cloud : il peut ainsi **déléguer les opérations de compilation et de test à d'autres machines**. De plus, il peut surveiller à distance des travaux menés sur d'autres serveurs. Remplacez 'serveurs physiques' par 'machines virtuelles' et Jenkins bascule naturellement vers le Cloud.

Notez que le PaaS DEV@cloud n'est pas limité à la gestion de projets Java en mode Cloud, même si ce duo répond à une tendance forte. Il permettra en effet de créer également des applications classiques, développées avec d'autres solutions de programmation, comme le C/C++, PHP ou encore Python.

CloudBees a récemment décidé de se centrer encore plus sur Jenkins. L'offre DEV@cloud sera ainsi renommée « Cloudbees Jenkins as a service » (version cloud de Jenkins Enterprise). La solution de déploiement des applications RUN@cloud disparaît pour sa part purement et simplement du catalogue de la société. Lire à ce propos l'article « [Cloudbees enterre son Paas pour se concentrer sur Jenkins](#) ».

A lire aussi :

[OpenStack : le framework Open Source du Cloud paré pour l'entreprise ?](#)