

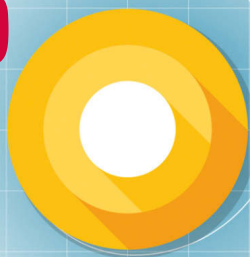
Amazon Alexa

Etendre et
personnaliser
Alexa



Android O

Toutes les
nouveau



Cloud Computing
+ DevOps :
le duo de choc



Monter un
cluster Docker
avec **ClusterHat**

L'art du debug



© jgroup



100% TECHNIQUE

WINDEV®



DSI

Développeurs

WebDesigners

Architectes logiciel

Ingénieurs

Start-ups...

Tech Tour

29

**SUJETS TECHNIQUES
D'ACTUALITÉ ET UTILES.**

Sur place, vous recevrez le **code source**
des applications présentées.

11 villes

du 2 mai au 31 mai

10.000 places

inscrivez-vous vite !

(gratuit)

www.pcsoft.fr

MONTPELLIER	mardi 2 mai
TOULOUSE	mardi 9 mai
BORDEAUX	mercredi 10 mai
NANTES	jeudi 11 mai
PARIS	mardi 16 mai
LILLE	mercredi 17 mai
BRUXELLES	jeudi 18 mai
STRASBOURG	mardi 23 mai
GENÈVE	mercredi 24 mai
MARSEILLE	mardi 30 mai
LYON	mercredi 31 mai

de 13h45 à 17h45

Inscrivez-vous vite ! (gratuit)

Tech
Tour

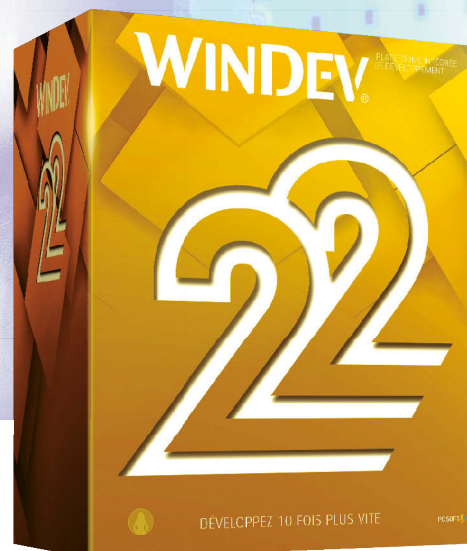
Logiciel professionnel

Elu
«Langage
le plus productif
du marché»

VOUS ÊTES INVITÉ !

**WINDEV
TECH TOUR 2017**

**SÉMINAIRE
100% TECHNIQUE
(10.000 PLACES)**



WINDEV TECH TOUR 2017:

VOUS AUSSI, PARTICIPEZ AU PLUS GRAND ÉVÉNEMENT
FRANCOPHONE DU DÉVELOPPEMENT PROFESSIONNEL.
(10.000 PLACES, GRATUIT)



WWW.PCSOFT.FR

Fracture numérique : le code peut-il être une (des) réponse(s) ?

Nous parlons chaque mois de technologies, de langages, d'outils, de machines. Mais nous avons aussi tendance à oublier une réalité : la fracture numérique. Avec la dématérialisation des démarches administratives, l'obligation de passer par Internet, le gouvernement veut moderniser l'accès à l'administration, fluidifier les démarches, éviter les attentes dans les locaux publics, optimiser les coûts et les traitements. La même démarche est faite à La Poste, les banques, les transports, etc. Même si au final, nous ne gagnons pas forcément du temps...














Si ces démarches se comprennent et se justifient, il ne faut pas les imposer bêtement et refuser la réalité : toute une population n'a pas d'ordinateurs ou d'accès à Internet, ou ne sait pas comment faire ni l'utiliser. La fracture numérique est une réalité trop souvent oubliée, ou que l'on évite soigneusement d'évoquer publiquement. Or, cette réalité doit trouver des réponses. Cette fracture ne concerne pas uniquement les personnes âgées mais tous les âges. Et avec une société de plus en plus technologique, informatisée, et bientôt robotisée, ces personnes doivent être entendues... Mais tout se passe comme si on préférerait imposer des services dématérialisés et attendre les conséquences sociales. L'accessibilité, l'ergonomie et la praticité de ces services sont parfois en cause, car certains sont tout sauf pratiques et simples à utiliser.

Le développement peut-il aider des jeunes et moins jeunes à trouver un travail, ou tout du moins une formation ? Oui, car, le marché est dynamique et la demande en techniciens et en développeurs reste forte et elle le restera. Non, car il est dangereux de faire croire que tout le monde sera développeur. Et tout le monde n'a pas vocation à être développeur. Le métier est (très) exigeant et la pression est forte. Mais oui il y a des opportunités pour celles et ceux qui sont motivés et qui en veulent. Celles et ceux qui sont motivés et prêts à s'investir dans notre univers s'ouvriront des métiers excitants. Mais il faut être honnête, la marche technique est parfois très haute et elle peut décourager. Persévérance tu apprendras.

Nous ne disons pas que devenir développeur sauvera le monde et réduira la fracture numérique, mais on peut redonner espoir aux personnes motivées qui veulent sortir d'une situation difficile. Et nous ne pouvons qu'encourager les initiatives autour du code.

François Tonic
ftonic@programmez.com



Agenda 4	Tableau de bord 6	Fabrique à Neuf 8	
	CloudFoundry Foundation 12	ABONNEZ-VOUS ! 11	
Echec de l'agile ? 14		Virtual Laval 16	 
ClusterHat 17	Conférence RSA 20	L'art du debug 21	
	NPM & Docker 46		Cloud+DevOps 27
	10 bonnes pratiques GreenIT 49		
Amazon Alexa 53		VoicelT 59	
Android O 62	Magento partie 2 67		API 70
	Programmation Atari ST 75	Maker partie 2 78	
	Atari 800 XL 80	Commitstrip 82	

**Dans le prochain numéro !
Programmez! #208, dès le 3 juin 2017**

Java 9

La prochaine version majeure de Java arrivera dans quelques semaines. Nous faisons le point complet sur Java 9 : les nouveautés, les migrations de codes, les outils.

API cognitives

Google propose de puissantes API cognitives. Dossier complet sur l'API Cloud Vision et l'analyse des images et des visages.

DevCon #3

CONFÉRENCE PROGRAMMEZ! LE 23 MAI

Programmez! organise sa 3e conférence technique sur 3 thèmes :

API, le Cloud Computing et les IoT.

- 2 keynotes
 - 4 sessions techniques
 - 1 pizza party
- + un track spécial sur la plateforme IoT "Constellation"

Informations & inscriptions sur
www.programmez.com

Où : la conférence se déroulera à Issy les Moulineaux.
 A -5 minutes des RER C et T2.

Accueil à 13h30. Début des conférences à 14h. Fin à 19h30.



PHP TOUR 2017

18 et 19 mai / Nantes

Le PHP Tour, le cycle de conférences itinérant de l'AFUP pour toutes les communautés PHP, professionnelles et open source, pose de nouveau cette année ses valises au C.C.O. de Nantes. Il sera question du futur ! PHP a célébré ses 20 ans en 2015, mais ne s'endort pas pour autant sur ses lauriers. Le langage a le regard obstinément tourné vers l'avenir et le programme du PHP Tour 2017 reflète cette vivacité : les conférences vont couvrir le large spectre de PHP.

Le site du PHP Tour 2017 à Nantes : <http://event.afup.org/>

Le programme : <http://event.afup.org/phptournantes2017/programme/>

La billetterie : <http://event.afup.org/phptournantes2017/tickets-inscriptions/>

Juin

L'APRÈS-MIDI DU DOMAIN-DRIVEN DESIGN

7 juin / Issy-les-Moulineaux

Deux ans après le succès de l'après-midi du Software Craftsmanship, nous reviendrons le 7 juin prochain chez Microsoft France avec la



même formule boostée en live-coding mais pour vous proposer cette fois une immersion dans le DDD que vous n'êtes pas prêt d'oublier. Fini le code métier noyauté par de la plomberie technique ou du code « générique » sans aucune saveur métier. Que vous ayez des connaissances ou non à propos du Domain Driven Design, cette conférence vous permettra de repartir avec les idées claires, mais surtout pleines d'astuces concrètes pour pouvoir les appliquer dès le lendemain sur vos projets (notamment « legacy »). Etudiants, développeurs et architectes sont les bienvenus. Inscription sur le Meetup DDD Paris : <http://tinyurl.com/192xtzx>

DEVFEST LILLE 2017

9 juin / Lille

Le Google Developer Group de Lille (@GDGLille) organisera la première édition du DevFest Lille. Au programme, des conférences et des codelabs autour des technologies que nous aimons tous : Web, Mobilité, Cloud, IoT, ...

Les inscriptions et le CFP sont ouverts.

Rendez-vous sur <https://devfest.gdglille.org/>

MAKER FAIRE PARIS 2017

du 9 au 11 juin / Paris

Cette nouvelle édition du Maker Faire se déplace à la Villette et se tiendra en dehors de la foire de Paris comme ce fut le cas les années passées. Comme chaque année, ce sera l'occasion de découvrir des centaines de makers et de projets divers et variés.

PARIS CONTAINER DAY

13 juin / Paris

2e édition de la conférence dédiée aux conteneurs et aux bonnes pratiques. Le principal thème de cette année sera « les conteneurs en production ».

Informations : paris-container-day.fr

[2017]

DÉVELOPPEZ 10 FOIS PLUS VITE

WINDEV®



WINDEV 22 : ATELIER DE DÉVELOPPEMENT PROFESSIONNEL, COMPLET EN STANDARD

Gestion du cycle de vie complet: Idée, Conception, Développement, Génération, Déploiement, Exploitation • Un code multi-plateformes Windows, Linux, Java, Internet, Mobiles • Environnement ALM complet • Toutes les bases de données sont supportées et cloud • Inclus: HFSQL, base de données locale, Client/Serveur, cluster, embarquée et Big Data • Puissant RAD • Intégration continue • Générateur de fenêtres (UI) visuel & intuitif • Utilisation d'Audit statique & dynamique • Héritage et surcharge d'interface • Tous les champs faciles de charte graphique • Générateur de champs de saisie, vers Word, envoi d'email, (contrôles) sont très puissants et livrés en standard: Champ de saisie, Champ Tableau croisé dynamique (cube), Champ Planning, Champ Diagramme de Gantt, Champ Tableau de bord, Champ Table, Champ Graphe, etc. • FAA: chaque application bénéficie automatiquement de Fonctionnalités Automatiques: Puissant générateur de rapports et codes-barres • Langage de 5ème génération: WLangage • Éditeur de code intuitif avec puissant débogueur • Tests unitaires et tests automatiques • Versioning (GDS/SCM) • Support de tous les standards: XML, USB, Bluetooth, NFC, J2EE, OLE, ActiveX, RPC, SaaS, SMTP, FTP, OPC, DLNA, IoT, Sockets, API, Webservices... • Lien avec des centaines d'exemples et d'assistants • Générateur de procédures d'installation: local, CD, USB, Télémétrie pour connaître l'utilisation réelle de vos applications • Support Technique Personnalisé Gratuit • ...

CONSULTEZ PLUS DE 100 TÉMOIGNAGES DE PROFESSIONNELS SUR LE SITE PCSOFT.FR

Elu
«Langage
le plus productif
du marché»

**VERSION
EXPRESS
GRATUITE**
Téléchargez-la !

**VU À LA TÉLÉ
EN 2017**

**SUR TF1, SUR BFMTV
ET SUR M6**



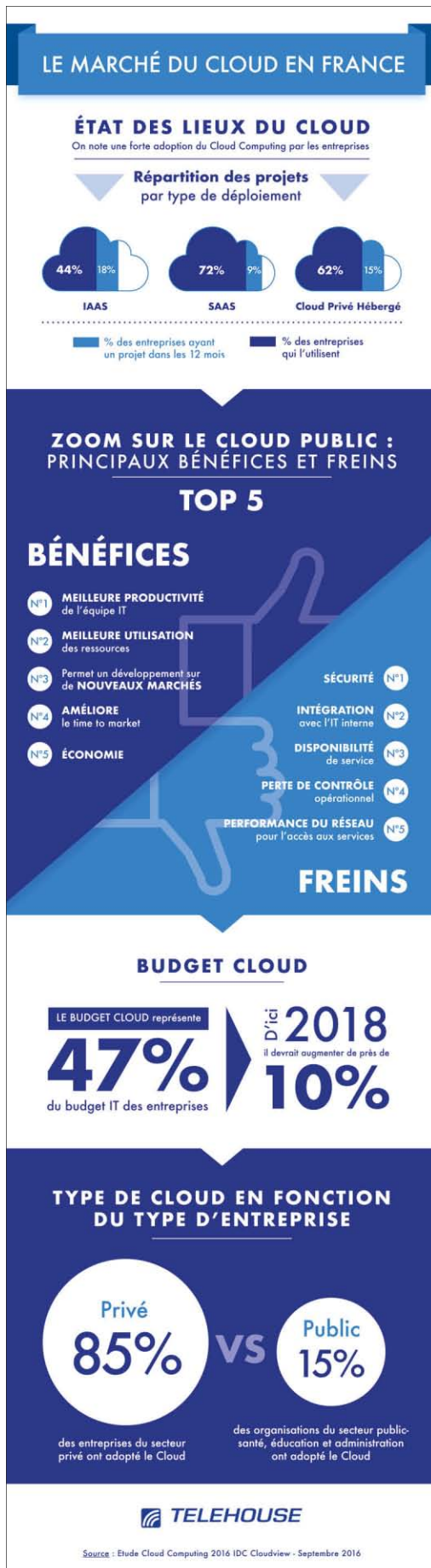
Tél Paris: 01 48 01 48 88

Tél Montpellier: 04 67 032 032

Plus de 100 témoignages
Dossier complet gratuit



WWW.PCSOFT.FR



Des salariés d'**Apple** estiment que les parkings de l'Apple Park sont trop loin. Chez **Tesla**, ils sont à côté mais trop petits...

Vaisseau mère bis. Apple Park provoque une pénurie d'arbres en Californie : ben oui, il faut planter +9000 arbres pour l'immense parc...

Un cours pour utiliser des apps développées en **Swift** sur Android est proposé par une école italienne. Android utilise Java comme langage.

Le 2 mai, **Microsoft** tient une conférence à New York normalement liée à l'éducation : Windows en mode Cloud pour concurrent ChromeOS comme rumeur et possiblement la nouvelle Xbox, le projet Scorpio.

Vroom. Uber passe de mauvaises semaines : plaintes de chauffeurs, interdictions diverses et maintenant l'espionnage d'un concurrent.

OUPS, ON OUBLIE UNE VERSION...

C'est une des grandes modes depuis quelques années : on saute d'une version à une autre, en oubliant une numérotation. Souvenez-vous de Windows 10 qui aurait du prendre un 9 ou encore PHP 7 qui fait l'impasse sur PHP 6, pour la bonne cause : il fallait oublier les errements du projet PHP 6... Aujourd'hui c'est Angular qui prend la version 4, en oubliant la v3. Mais promis, beaucoup de codes d'Angular 2 sera compatible, mais pas tout. Rassurez-vous, les versions 5 et 6 sont prévues et même très rapidement... Oracle aurait pu faire la même chose avec Java en passant directement de Java 8 à Java 10...

INDEX TIOBE

Pas de changement dans la popularité des recherches sur les langages en avril : les 6 premiers ne bougent pas sur 1 an. Quelques changements après : VB reprend quelques places, tout comme Delphi.

Avril 2017	Avril 2016	Evolution	Langage	%	Evolution
1	1	=	Java	15,568%	-5,28%
2	2	=	C	6,966%	-6,94%
3	3	=	C++	4,554%	-1,36%
4	4	=	C#	3,579%	-0,22%
5	5	=	Python	3,457%	+0,13%
6	6	=	PHP	3,376%	+0,38%
7	10	+3	Visual Basic .NET	3,251%	+0,98%
8	7	-1	JavaScript	2,851%	+0,28%
9	11	+2	Delphi/Object Pascal	2,816%	+0,60%
10	8	-2	Perl	2,413%	-0,11%

Swift se classe 12e et se rapproche du top 10.

CANONICAL FAIT DES CHOIX DIFFICILES

L'éditeur de la distribution Ubuntu a annoncé il y a quelques semaines des choix brutaux dans sa stratégie et les projets développés. La société arrête de développer la version mobile, la fin de son interface graphique Unity, et un retour à une interface standard, Gnome, même si le projet ne serait pas totalement mort. Dans le même temps, Canonical veut miser sur deux marchés importants : le Cloud et les objets connectés. Sur le premier, Ubuntu est présent mais bien moins que SuSe ou Red Hat. Sur la partie IoT, le marché est bien plus ouvert. L'éditeur annonce aussi la fin du projet Mir sur la partie desktop et va être soutenu sur les IoT. Des licenciements devraient suivre cette nouvelle stratégie. On parle de 80 personnes, voire, plus. Et enfin, preuve que ces changements sont profonds, le fondateur, Mark Shuttleworth, revient à la tête de Canonical. Mark avait quitté le poste de dirigeant en 2010.

Quoi de neuf dans les systèmes mobiles ?

Kantar a publié son rapport trimestriel sur les systèmes mobiles. Commençons par Windows Mobile qui n'en finit plus de chuter : 2,4 % en France, 3 en Allemagne, 1,7 aux USA, 0,2 en Chine, et, incroyable une augmentation énorme au Japon, passant de 0,5 à 1,3 %...

Dans les 5 grands marchés européens, Android est plus que jamais devant avec 75 % mais iOS réussit à attendre 24 %. Ces chiffres se font au détriment de Windows. Aux USA, Android atteint 55,9 % contre 42 pour Apple. Par contre en Chine, iOS continue de chuter, à peine 13 %. Au Japon, Android est désormais 1er avec 54 %.

SERVEURS DÉDIÉS XEON®

AVEC **ikoula**
HÉBERGEUR CLOUD

Optez pour un serveur dédié dernière génération et bénéficiez d'un support technique expérimenté.



POUR LES LECTEURS DE
PROGRAMMEZ*

OFFRE SPÉCIALE -60 %
À PARTIR DE

11,99€
HT/MOIS

~~29,99€~~

CODE PROMO
XEPRO17

- ✓ Assistance technique **en 24/7**
- ✓ Interface **Extranet** pour gérer vos prestations
- ✓ **KVM sur IP** pour garder l'accès
- ✓ Analyse et surveillance **de vos serveurs**
- ✓ **RAID Matériel** en option
- ✓ Large choix d'**OS** Linux et Windows

*Offre spéciale -60 % valable sur la première période de souscription avec un engagement de 1 ou 3 mois. Offre valable jusqu'au 31 décembre 2017 23h59 pour une seule personne physique ou morale, et non cumulable avec d'autres remises. Prix TTC 14,39 €. Par défaut les prix TTC affichés incluent la TVA française en vigueur.

CHOISISSEZ VOTRE XEON®

<https://express.ikoula.com/promoxeon-pro>



ikoula
HÉBERGEUR CLOUD



/ikoula



@ikoula



sales@ikoula.com



01 84 01 02 50

NOM DE DOMAINE | HÉBERGEMENT WEB | SERVEUR VPS | SERVEUR DÉDIÉ | CLOUD PUBLIC | MESSAGERIE | STOCKAGE | CERTIFICATS SSL

La **Fabrique à neuf** : redonner une vie aux objets

Nous parlons souvent de gaspillage, d'obsolescence programmée, de renouvellement accéléré de nos matériels, etc. Mais nous oublions trop souvent une délicate question sociétale : la fracture numérique. Cette réalité touche de nombreux habitants en France et ailleurs. Revenons sur le rôle de La Fabrique à neuf, ses objectifs et son travail au quotidien, avec Fabrice Boutard.

Fabrice, 34 ans, animateur radio de sa profession, est ce qu'on peut appeler un geek. Il a eu son premier ordinateur à 13 ans. Sensible au recyclage et plus généralement au réemploi des choses plutôt que de les jeter, toujours à chiner des trucs improbables dans les brocantes du coin, sa rencontre et son engagement dans la ressourcerie associative La Fabrique à Neuf était une évidence. Nous avons été impressionnés par cette initiative et son enracinement local.

Peux-tu nous parler de La Fabrique à neuf en quelques mots ? Son fonctionnement et les buts qu'elle a ?

La Fabrique à Neuf est un réseau associatif de 3 ressourceries (recycleries) situées à Corbeil-Essonnes, Ris-Orangis et Montgeron.

Le principe est relativement simple : nous nous évertuons à récupérer, collecter et essayer de donner une deuxième, voire une troisième vie aux objets plutôt que de les voir terminer à la poubelle ou à la déchetterie. Il faut avoir un chiffre en tête : en France près de la moitié de ce que nous jetons à la poubelle est enseveli sous terre, sans même être trié ou recyclé. La moitié ! À titre de comparaison, la Suède ensevelit moins de... 1% de ses déchets. Elle est même obligée d'importer ceux des voisins ! Donner une deuxième vie, car beaucoup d'objets sont réutilisables en l'état. Un simple coup de chiffon suffit à les proposer à tarif solidaire dans nos boutiques en Essonne (91).

Donner une troisième vie, car certains objets en mauvais état peuvent malgré tout servir de base pour créer quelque chose de nouveau. C'est ce qu'on appelle l'upcycling, ou « surcyclage » en bon français. Ça consiste souvent à détourner certains objets de leur utilisation principale par exemple un lampadaire-tambour-machine à laver, qu'on a fabriqué avec un vieux portemanteau, une lampe halogène cassée, et un tambour en inox de machine à laver en panne !



Lors d'un meetup il y a quelques semaines, tu parlais des PC remis à neuf à partir de 30 €, comment est-ce possible dans notre univers technologique ? Comment fais-tu pour réparer ou recycler ?

Le cœur de notre activité concerne essentiellement les meubles, les vêtements, les objets de décoration, mais de plus en plus également les objets électroniques et informatiques, technologie oblige.

C'est d'ailleurs comme ça que je suis rentré dans l'association, car j'ai vu que je pouvais y apporter une certaine expertise pour donner une seconde vie aux ordinateurs récupérés. Donner une seconde vie aux ordinateurs n'est pas quelque chose de très compliqué en soi. Une unité centrale n'est rien de plus qu'un jeu de LEGO, où les éléments s'emboîtent les uns dans les autres. Ça semble technique et compliqué quand on n'a jamais mis les mains dedans, alors qu'il n'y a rien de plus simple ! Le seul défi reste avant tout de proposer ensuite un ordinateur qui soit utilisable au quotidien, sans soucis de compatibilité ou de gros ralentissements.

Avant de lancer ce projet, j'ai passé pratique-

ment un trimestre de mon temps libre à télécharger, installer et tester différents systèmes d'exploitation pour trouver la solution logicielle la plus facile à déployer (pour ne pas perdre trop de temps par ordinateur), mais aussi la solution la moins onéreuse et la plus accessible pour le plus néophyte et technophobe des utilisateurs.

Mes choix se sont rapidement orientés vers des solutions Linux, même si cette communauté et ces outils sont particulièrement hermétiques aux débutants biberonnés à Windows depuis des années.

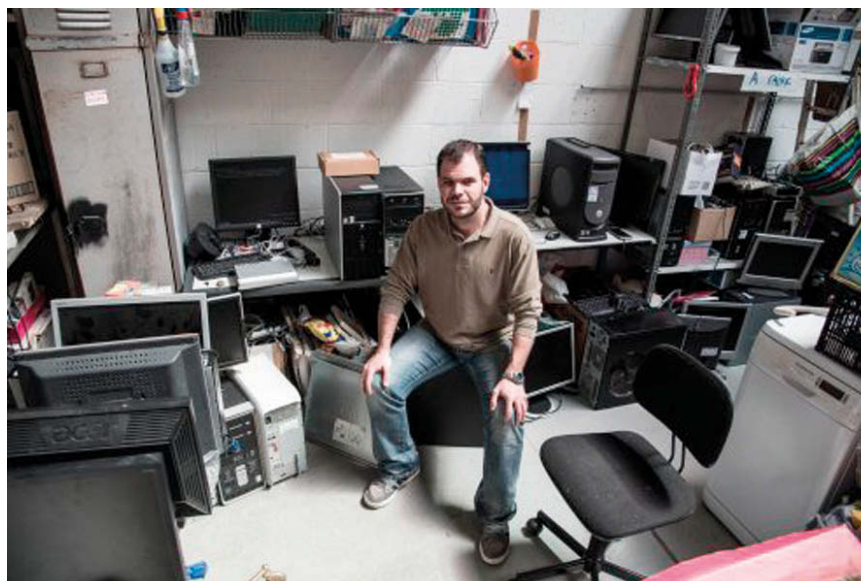
J'ai testé (entre autres) Puppy Linux, Ubuntu, Xubuntu, Ubuntu MATE et bien d'autres solutions aux noms tordus et dont encore aujourd'hui je ne perçois pas trop les différences en tant qu'utilisateur lambda souhaitant me lancer dans le monde du libre. Chacune a ses qualités et ses défauts, comme la compatibilité des drivers, l'accessibilité et l'ergonomie générale du système d'exploitation, l'austérité de l'interface, ou l'absence de communauté (ou non francophone). Et puis je suis tombé sur HandyLinux, une distrib franco-française (désormais devenue DFLINUX « Debian Facile Linux ») sous Debian, qui reste à ce jour mon

choix de prédilection. Je l'ai installée sur près de 150 ordinateurs, sans aucune difficulté d'installation, de driver, ou d'utilisation. Pour ceux qui ne connaissent pas, c'est un équivalent Windows très léger qui contient de nombreux drivers et de nombreux logiciels prêts à l'emploi. Il s'installe et est totalement fonctionnel en moins de 30 minutes, mise à jour comprise, là où Windows va demander des heures de mise à jour et d'interminables recherches de drivers... Je suis encore époustoufflé aujourd'hui du rendu du système d'exploitation au regard de la rapidité (et de la simplicité !) de l'installation, comparé à Windows ! Ça, c'était pour la partie logicielle. Concernant la partie hardware, je teste chacun des ordinateurs qui entre à la ressource (dons des particuliers ou des entreprises/collectivités locales qui changent leur parc informatique). Je diagnostique rapidement les éventuelles pannes, souvent les mêmes (alimentation, lecteur DVD HS...). Mais la plupart du temps les ordinateurs sont en excellent état de fonctionnement, ils sont simplement technologiquement dépassés. (obsolescence technique, des ordinateurs de +/- 10 ans). Je me fixe le strict minimum d'un Pentium 4, avec un lecteur DVD, 1 à 2 Go de RAM et un disque dur d'une centaine de Go. Mais la plupart du temps nous sommes plus sur des dual core avec 2 Go de RAM minimum et un disque dur de près de 300 Go. Tout dépend bien évidemment des apports que nous recevons.

Tout ce qui est en dessous de ces caractéristiques est démantelé pour être recyclé (Pentium 2, Pentium 3...). Tout ce qui est identifié comme étant en panne critique (carte mère, processeur) est aussi démantelé pour récupérer des pièces de rechange afin de procéder aux diverses réparations. Je n'ai pas fait de statistiques, mais je dirais que sur une dizaine d'ordinateurs reçus, j'arrive à en ressource facilement 8 ou 9 !

On parle beaucoup de fracture sociale en France depuis de nombreuses années, mais on oublie souvent un point crucial : la fracture numérique. Comment se concrétise-t-elle au quotidien ? Qui sont les plus touchés par cette fracture et aussi pour rompre un isolement des personnes ? Tes PC à partir de 30 € répondent à ce problème ?

Le projet de recycler des ordinateurs a pour origine un article que j'ai lu il y a quelques



temps de cela sur la ville de Nice, qui ne recevait plus personne aux guichets des services publics, car tous les services avaient été dématérialisés. Pile au moment où je rentre en contact avec l'association, c'est au tour de Pôle emploi de faire le virage du tout numérique. J'ai trouvé ça un peu choquant, car fait sans réelle politique numérique, et aussi, car Pôle emploi est un service public qui s'adresse justement à des personnes sans emploi et donc plus fragiles que d'autres et pour beaucoup ne pouvant pas prétendre acquérir un ordinateur à 800 € pour faire recherches et démarches. C'est ce qui m'a poussé à trouver une solution pour faire quelque chose de ces vieux PC qu'on accumulait. On a donc décidé d'en proposer certains à un tarif d'appel particulièrement solidaire de 30 € l'unité centrale seule, 50 € le PC complet, prêt à brancher. Vous imaginez bien que pour ce prix inutile d'espérer faire tourner le moindre jeu où lire la moindre vidéo Full HD sur YouTube, mais ce n'est absolument pas ce que j'ai prétendu vouloir faire.

Je voulais un ordinateur accessible, simple d'utilisation, avec lequel l'utilisateur -jeune ou beaucoup moins jeune- puisse créer et envoyer ses CV, ses cours, répondre aux offres d'emploi, et mettre à jour sa situation sur les sites Internet des différents services publics. Bien évidemment dans le lot nous récupérons parfois des ordinateurs bien plus puissants. Ils sont également vérifiés, formatés, réinstallés sous HandyLinux/DebianFacile et remis en vente à tarif solidaire au regard de leurs caractéristiques.

Finalement, nous n'avons pas besoin de la fuite en avant technologique : un nouveau PC tous les ans ou presque, pour les usages quotidiens, pas la peine d'avoir un PC exploitant à peine 10 % de sa puissance, comment tu regardes cela ?

Je m'inscris en vrai dans cette question et cette remarque. Il y a d'ailleurs une expression bien connue qui dit que les cordonniers sont les plus mal chaussés. J'en suis la preuve vivante puisque mon ordinateur de tous les jours date de 2004. C'est un vieux PC portable HP dual core, sur lequel je n'ai ajouté qu'une barrette de mémoire. Et ça suffit ! Bon, je dois avouer qu'au bout de 13 ans il commence un petit peu à manquer de souffle au regard de l'utilisation intensive que j'en ai, surtout qu'aujourd'hui je monte des vidéos en Full HD... Mais à part ça, il tourne comme une horloge !

Je confirme que nous n'avons pas besoin au quotidien d'une telle débauche de puissance. Surtout que les gains en puissance des PC aujourd'hui sont souvent peu perceptibles puisque d'une année sur l'autre les systèmes d'exploitation, les logiciels et nos usages Internet sont de plus en plus gourmands. Mon bon vieux Windows fait parfaitement l'affaire !

Pour les personnes qui cherchent à s'équiper et qui ont un minimum de connaissances informatiques, je ne peux que les inviter à se tourner sur le marché de l'occasion. D'ailleurs, on songe de plus en plus à créer des ateliers DIY pour monter/réparer/formater/réinstaller son PC à la Fabrique à Neuf. A suivre...

Si je regarde au-delà de La Fabrique à neuf et des PC à partir de 30 €, plusieurs initiatives existaient ou existent pour proposer des ordinateurs « de base » à -100 €, notamment avec des solutions basées sur des cartes de type Raspberry Pi. Qu'en penses-tu ?

Effectivement il existe encore d'autres solutions pour acquérir un ordinateur peu cher et suffisant pour 90 % des usages.

Il y a 2 ans je me suis lancé à acheter un petit PC portable 11" tout léger pour toujours l'avoir avec moi. Même si je n'étais pas convaincu au moment de l'achat, j'ai tenté ma chance en prenant un Chromebook de chez Google (ASUS C200).

Bien m'en a pris ! Cette machine est une merveille de technologie.

C'est simple : à l'intérieur il n'y a que le navigateur Chrome. Rien d'autre. Pas d'OS, pas de logiciels, pas de... rien. Ce choix est assez intelligible puisque désormais 95% de nos utilisations sont connectées (webmail, réseaux sociaux, vidéos en ligne, shopping, démarches... Même les logiciels de bureautique et d'édition (basique) audio/photo peuvent se trouver en ligne ! C'est un ordinateur qui démarre en 2 secondes, tout le temps disponible, ultra réactif et dispos en plus d'une autonomie gargantuesque d'une dizaine d'heures sur batterie. Tout ça pour la somme faramineuse de... 150 €. Neuf.

C'est pour moi un énorme coup de cœur, et un produit qui mérite au moins de se poser la question. D'autant plus que Google a annoncé sa volonté de rapprocher les Chromebook

avec son système d'exploitation Android.

De nombreux modèles vont bientôt d'ailleurs avoir accès au Google Store, afin de pouvoir profiter des mêmes applications que sur son smartphone.

Je dois quand même avouer qu'il ne remplit pas toutes les missions que je peux attendre d'un ordinateur comme imprimer, ou faire du montage photo, audio ou vidéo. Pas bien grave, au regard du temps qu'il me fait gagner !

Je pense avant tout qu'il faut un ordinateur en adéquation avec ses moyens et surtout ses besoins. Avoir une formule 1 ne vous permettra pas d'aller plus vite faire vos courses à Carrefour...

Pour ma part je crois énormément dans l'avenir d'Android, et de Linux. Android est un système d'exploitation hyper user-friendly qui sort de plus en plus de son usage téléphonique, comme le futur Samsung Galaxy S8 que l'on pourra connecter à un écran et utiliser tel un ordinateur de bureau avec le traditionnel combo clavier/souris. Ce n'est pas sans rappeler le concept de Jide, qui commercialise (trop discrètement malheureusement) depuis quelques temps déjà un mini-ordinateur à brancher sur un écran d'ordinateur ou de télé pour « PC-iser » Android via Remix-OS. Coût : entre 80 et 120 €, pour un PC sous Android franchement très puissant ! Le logiciel peut être téléchargé gratuitement.

Ensuite il existe bien sûr la solution de passer par LA fameuse carte à tout faire, le Raspberry Pi 3.

Je suis définitivement fan de ce concept. Je

m'en sers d'ailleurs quotidiennement pour jouer aux jeux vidéo de mon enfance grâce à Recalbox, un OS entièrement dédié au rétro-gaming ! C'est une vraie réussite, et incroyablement puissant comme bestiole pour le prix !

Je sais qu'il existe plusieurs systèmes d'exploitation à installer directement dessus, pour faire de cette petite carte vendue une trentaine d'euros un véritable ordinateur polyvalent. Le problème de ces solutions reste l'accessibilité grand public aux néophytes/technophobes. Ce côté hyper user-friendly qu'ont réussi à créer Android et iOS.

On dira ce qu'on voudra --et je vais probablement m'attirer les foudres de certains en disant ça--, mais malheureusement Linux jouit d'une très mauvaise image auprès du grand public. « Un truc de geek, compliqué, incompatible, incompréhensible... »

Il existe une quantité pléthorique de distributions, à tel point qu'il en devient impossible de faire un choix quand on a été biberonné à Windows depuis des années.

Et quand bien même on arrive à faire le choix d'une version de Linux, s'offrent à nous des versions X86, X64, ARM, i386, i486, i586, la personne préfère retomber dans la solution de simplicité de Windows, bien plus lisible.

J'ai un coup de cœur à ce titre pour l'association PIQO, qui oeuvre à rendre accessible Linux, en proposant entre autres des Raspberry Pi 3 préconfigurés en ordinateur de bureau clés en main. J'aime beaucoup l'initiative, même si je sais qu'elle n'est pas au goût de certains. Tant pis pour eux. Je retiens pour ma part que c'est du Made In France, c'est écologique, économique, ça marche très bien et ça contribue à populariser chaque jour un peu plus Linux et le monde du libre dans les foyers !

Ressources

<https://www.facebook.com/LaFabriqueNeuf/>
http://www.francetvinfo.fr/monde/environnement/la-suede-manque-tellement-de-dechets-qu-elle-en-importe_1698591.html
<https://debian-facile.org/projets:live-df>
<https://lescahiersdudebutant.fr/src/doku.php/dflinux#téléchargements>
<http://www.jide.com/remixos>
www.recalbox.com
<http://www.mypiqo.com/fr/products/piqo-one-se/>



Abonnez-vous à **programmez!**

le magazine des développeurs

Nos classiques

1 an 49€*
11 numéros

2 ans 79€*
22 numéros

Etudiant 39€*
1 an - 11 numéros * Tarifs France métropolitaine

Abonnement numérique

PDF 35€*
1 an - 11 numéros

Souscription uniquement sur
www.programmez.com

Option :
accès aux archives 10€

Nos offres spéciales printemps 2017

1 an 59€
11 numéros + 1 vidéo ENI au choix :

- Big Data avec Hadoop
 - Framework Spring
- (Valeur de la vidéo de 29,99 à 59,99 €)



2 ans 89€
22 numéros + 1 vidéo ENI au choix :

- Big Data avec Hadoop
 - Framework Spring
- (Valeur de la vidéo de 29,99 à 59,99 €)

Offre limitée à la France métropolitaine

Toutes nos offres sur www.programmez.com

Oui, je m'abonne

ABONNEMENT à retourner avec votre règlement à :
Service Abonnements PROGRAMMEZ, 4 Rue de Mouchy, 60438 Noailles Cedex.

- ☐ **Abonnement 1 an : 49 €**
☐ **Abonnement 2 ans : 79 €**
☐ **Abonnement 1 an Etudiant : 39 €**
Photocopie de la carte d'étudiant à joindre

- ☐ **Abonnement 1 an : 59 €**
11 numéros + 1 vidéo ENI au choix :
☐ **Abonnement 2 ans : 89 €**
22 numéros + 1 vidéo ENI au choix :

- ☐ Vidéo : Big Data avec Hadoop
☐ Vidéo : Framework Spring

☐ Mme ☐ M. Entreprise : _____ Fonction : _____

Prénom : _____ Nom : _____

Adresse : _____

Code postal : _____ Ville : _____

email indispensable pour l'envoi d'informations relatives à votre abonnement

E-mail : _____ @ _____

☐ Je joins mon règlement par chèque à l'ordre de Programmez !

☐ Je souhaite régler à réception de facture

* Tarifs France métropolitaine



Faisons le point sur **Cloud Foundry** avec Abby Kearns

La fondation Cloud Foundry est assez discrète mais pas inactive pour autant. Nous avons fait le point avec Abby Kearns qui est au cœur de la fondation.

Quelle est la raison d'être de la fondation et comment la fondation soutient et travaille avec Cloud Foundry ?

La fondation existe pour sensibiliser et faire adopter la plateforme open source du même nom. Il s'agit aussi de favoriser et d'aider la communauté et de créer une stratégie cohérente avec toutes les sociétés membres de la fondation. Nous avons des activités commerciales et techniques pour l'ensemble de la communauté : gouvernance du projet avec équité, croissance de notre écosystème, disponibilité des applications, développement des compétences. La communauté technique travaille avec les autres projets open source pour partager les standards, intégrer les technologies des communautés telles que l'Open Container Initiative (OCI) ou encore le Cloud Native Computing Foundation (CNCF). Notre conseil d'administration est en charge de la stratégie et de la gouvernance commerciale de la fondation. Cloud Foundry est une des plateformes applicatives les plus matures sur les conteneurs. La plateforme s'appuie sur les conteneurs pour déployer et gérer les applications en s'appuyant sur les dernières avancées de l'OCI et du CNCF.

Comment Cloud Foundry fait-elle entendre sa voix et ses choix techniques face aux grands fournisseurs tels que Google, Amazon, Microsoft, Salesforce, etc. qui proposent du IaaS et du PaaS ?

Un des avantages de l'open source est que nous travaillons avec la plupart des grandes entreprises technologiques. Google est membre de la fondation par exemple. Nous avons des Cloud Platform Interface (CLI) avec Microsoft Azure et Amazon Web Services. Les éditeurs ont collaboré à ces développements et ils contribuent à Cloud Foundry. Nous croyons



en l'open source. Nous n'avons pas plus à nous inquiéter de faire entendre notre voix que de veiller à l'adoption des standards par l'industrie technologique. Gage de réussite pour tout le monde.

IBM utilise Cloud Foundry au cœur de Bluemix. L'usage de Cloud Foundry est-il dans ce type de déploiement et d'utilisation ?

Bluemix est une implémentation d'une architecture open source par IBM, basée sur Cloud Foundry. Il permet aux utilisateurs / développeurs de créer, déployer et gérer des applications Cloud. Comme Bluemix est basé sur Cloud Foundry, les utilisateurs peuvent profiter de notre écosystème. Certains services sont proposés par la fondation, d'autres par des sociétés telles que IBM.

Dans le PaaS Java, la concurrence est vive, pour vous, quels sont les atouts de la plateforme ?

Une plateforme permet aux développeurs de se concentrer sur les fonctionnalités à rajouter / à développer, et de résoudre tous les problèmes liés à l'activité ainsi qu'à apporter de la valeur. Nous utilisons la technologie des conteneurs Linux, et avec une simple commande « push », la plateforme créera le conteneur pour le développeur. C'est énorme car, encore une fois, le développeur se concentre sur son

code et non sur les problèmes d'infrastructures. La plateforme s'occupe de la plomberie et simplifie au maximum la configuration et la gestion des applications dans l'environnement.

Comment séduire les développeurs et les inciter à vous utiliser et pas une autre plateforme ?

Il faut comprendre que Cloud Foundry a été créée pour faciliter la vie des développeurs et nous pensons qu'ils sont intéressés par la plateforme. Les développeurs aiment avoir le contrôle et la flexibilité pour utiliser le langage, le framework ou le Cloud qu'il souhaite, sans être bloqués par l'infrastructure. Bien que cela soit très important, ils veulent que l'infrastructure soit simple et gérée pour eux. Cloud Foundry est là pour fournir le continuous delivery sans se préoccuper du reste.

Quelles seront les grandes annonces et les grands mouvements pour 2017 ?

Nous venons d'annoncer une initiative pour la certification des développeurs Cloud natifs : Cloud Foundry Certified Developer (CFCD). Elle sera disponible pour tous les développeurs durant notre conférence à la Silicon Valley en juin prochain. Nous annoncerons d'autres choses mais je ne peux pas en parler ici.

Comment jugez-vous l'écosystème Cloud Foundry (hébergeurs, fournisseurs de Cloud, SSII, etc.) ? HPE a récemment vendu les activités Cloud Foundry à SuSe par exemple.

Notre écosystème est soutenu par les sociétés membres venant de diverses industries et les développeurs. HPE reste engagé sur la plateforme malgré les ventes de ses actifs à SuSe. HPE fut un des premiers soutiens à Cloud Foundry et à mobiliser des ressources de développement pour en faire une solution d'entreprise. SuSe est un acteur important dans le monde de l'entreprise et dans l'open source depuis plus de 20 ans. Nous sommes impatients de voir l'impact de leur distribution Cloud Foundry auprès des entreprises et des développeurs.



NOUVEAU

HÉBERGEMENT DE HAUT NIVEAU

NOUVEAU : performance flexible et évolutive ! Bénéficiez du niveau de performance idéal pour votre projet Web. Que vous utilisiez des applications gourmandes en ressources ou que vous prévoyiez un pic de visites suite à l'envoi d'une newsletter, la publication d'un article de blog ou l'intégration d'une boutique en ligne, ajustez simplement le niveau de performance de votre site en fonction de vos besoins.

✓ Performance de haut niveau

- **NOUVEAU** : 2,5 Go de RAM
- **NOUVEAU** : performance ajustable en quelques clics jusqu'à 19 Go de RAM
- Suivi de performance depuis l'Espace Client 1&1
- Modification du niveau de performance sans interruption de service

✓ Vitesse de haut niveau

- **NOUVEAU** : temps de chargement accélérés avec HTTP/2
- **NOUVEAU** : PHP 7.1 + OPcache
- 1&1 CDN
- Assistance 24/7

✓ Sécurité de haut niveau

- Certificat SSL inclus
- Protection DDoS
- Géo-redondance

À partir de
~~4,99~~ **0,99** € HT/mois
 (1,19 € TTC)*



☎ 0970 808 911
 (appel non surtaxé)



1and1.fr

* À partir de 0,99 € HT/mois (1,19 € TTC) la 1^{ère} année au lieu de 4,99 € HT/mois (5,99 € TTC) pour le pack hébergement 1&1 Basic avec un engagement minimum de 12 mois. À l'issue des 12 premiers mois, les prix habituels s'appliquent. Offres sans durée minimale d'engagement également disponibles. Conditions détaillées sur 1and1.fr. 1&1 Internet SARL, RCS Sarreguemines B 431 303 775.

L'agilité en échec, ou l'anarchie dans l'entreprise



Fabien Maury
Software Gardener chez **Arolla**
@MrCafetux

S'adapter à un environnement changeant et surtout rectifier le tir pendant qu'il en est encore temps : c'est la première caractéristique qui me vient à l'esprit.

Le feedback permet de vérifier que l'on s'est bien compris, d'apprendre et de s'adapter. Et si la boucle de feedback est rapide, elle permet d'apprendre rapidement et donc de s'améliorer. Les pratiques agiles sont souvent mises en opposition avec la gestion de projet traditionnelle (dite "waterfall", "cycle en V"), car elles permettent d'éviter le fameux "effet tunnel", qui consiste à ne pouvoir recevoir de feedback qu'en fin de projet, donc quand il est trop tard.

Le fait de devoir construire la route qu'emprunte le projet, au fur et à mesure qu'il avance, me fait penser à l'anarchiste Kropotkine qui écrivait que l'harmonie dans la société se ferait "par un incessant mouvement d'ajustement et de réajustement entre une multitude de forces et d'influences". Kropotkine se refusait, comme la plupart des anarchistes, à établir des plans précis et définitifs sur le fonctionnement que devrait avoir la société. En effet, les anarchistes croient en l'autogestion et en l'autorégulation, et Kropotkine avait bien compris cette valeur du manifeste agile : **l'adaptation au changement, plus que le suivi d'un plan.**

L'ordre moins le pouvoir

L'anarchie est aujourd'hui souvent associée et réduite au chaos et à la violence. Afin d'éviter toute confusion, il me semble donc important de revenir sur les principes de base de l'anarchie. Tout d'abord, l'anarchie prend sa source dans la contradiction entre l'Etat et la société (l'Etat ayant des buts et des objectifs en décalage avec les attentes et besoins du peuple) et souhaite éliminer cette contradiction en supprimant l'Etat. Cela ferait de notre société une réunion libre d'êtres libres et égaux. En revanche, absence de pouvoir ne veut pas dire absence d'organisation : on parle donc bien ici d'auto-organisation. J'imagi-

L'agilité. Tout le monde en parle, tout le monde en fait ... mais presque tout le monde affiche un sourire crispé quand on lui en parle, met des guillemets en l'énonçant, s'excuse de le faire approximativement : 'On fait presque du Scrum', 'oui, on est "agile" (petit signe en guillemets)', 'oui, oui, on est agile, enfin...' Mais au fond de quoi parle-t-on ? Qu'est-ce que ce fameux "agile" ?

ne que vous voyez où je veux en venir. Par exemple, les anarchistes n'aiment pas vraiment la police car elle est là pour faire respecter les lois... Et les anarchistes n'aiment pas vraiment les lois. Seraient-ils donc pour le chaos ? Pour une société où chacun pourrait tuer son prochain sur un coup de tête et voler autrui par plaisir ? Bien sûr que non. Dans **La morale anarchiste**, Kropotkine explique que les lois définissent ce qui est bien ou mal dans la société, le concept de légalité se substituant ainsi à la morale. En outre, il dénonce le fait que les lois ont tendance à se mêler de choses qui ne regardent pas l'Etat (la sexualité, les croyances, etc.) et punissent des actions hors de leur contexte. Kropotkine nous propose donc une autre vision, basée sur le bien commun.



En effet, la seule vraie question qui semble retenir l'attention de Kropotkine est la suivante : "Est-ce bon pour la communauté ?". Sans la barrière entre légalité et illégalité et sans la présence des forces de l'ordre pour faire respecter ces lois, il serait à la charge de chaque individu de se poser la question et d'intervenir le cas échéant. Il faut donc arriver à faire en sorte que les gens aient confiance en leur prise d'initiative pour qu'ils deviennent acteurs de leur société. On retrouve là la notion de Leadership portée par le lean et donc l'agilité. Ainsi, une personne incivique dans les transports ou s'accaparant trop de ressources serait aussitôt stoppée par la population. Cette dernière ne pouvant plus se reposer sur une autorité tierce pour faire respecter l'ordre. Cela remettrait l'éthique et la morale au centre de la société et des préoccupations des citoyens. L'effet positif serait de remettre toutes les formes de violences (physique, sociale, etc.) sur un pied d'égalité.

Agilité et anarchie contre le dogmatisme

L'agilité prône les équipes auto-organisées et le pragmatisme. Elle propose de s'adapter plutôt

que de suivre dogmatiquement des process. Mais au final, qu'est-ce qu'un process sinon une loi dans l'entreprise ? Les agilités n'auraient-ils pas quelques valeurs anarchistes ?

Serions-nous en train d'arriver à cette autre valeur du manifeste agile : **Les individus et leurs interactions plus que les processus et les outils ?**

C'est pour cela que l'on préfère, au sein des équipes agiles, respecter des règles d'équipe (règles de vie définies par l'équipe lors des rétrospectives), que des process dictés par une tierce autorité. D'ailleurs, le 11ème principe sous-jacent au manifeste Agile l'explique bien : **Les meilleures architectures, spécifications et conceptions émergent d'équipes auto-organisées.**

Et l'on peut noter que les anarchistes ne rejettent pas toute forme d'autorité, seulement l'autorité imposée, ce qu'explique admirablement Bakounin :

" Nous acceptons toutes les autorités naturelles et toutes les influences de fait, aucune de droit, car toute autorité ou toute influence de droit, et comme telle officiellement imposée devenant aussitôt une oppression et un mensonge, nous imposerait infailliblement, comme je crois l'avoir suffisamment démontré, l'esclavage et l'absurdité. En un mot, nous repoussons toute législation toute autorité et toute influence privilégiée, patentée, officielle et légale, même sortie du suffrage universel, convaincus qu'elles ne pourront tourner jamais qu'au profit d'une minorité dominante et exploitante, contre les intérêts de l'immense majorité asservie. Voilà dans quel sens nous sommes réellement des anarchistes."

Mikhail Bakounin

L'absence d'autorité n'est donc pas incompatible avec la présence de leaders, en revanche, elle demande une double responsabilité : celle de vouloir mener le groupe d'une part, et pour le groupe, celle de vouloir suivre les propositions du leader. Tout le monde est ainsi respon-

sabilisé. Dans un registre plus capillotracté, on pourrait aussi considérer la loi comme une documentation exhaustive.

L'anarchie prônerait alors le fait d'avoir **une société opérationnelle plus qu'une documentation exhaustive**, c'est-à-dire une société qui fonctionne pour les personnes qui y vivent plutôt que d'avoir des lois qui expliquent comment les gens devraient se comporter.

Une histoire de confiance ?

C'est la valeur qui se cache derrière la 3e valeur du manifeste agile : **la collaboration avec les clients plus que la négociation contractuelle**. Un contrat est généralement mis en œuvre pour protéger les parties l'une de l'autre. Difficile de vraiment collaborer lorsque ce qui nous unit est justement ce qui nous sépare.

On ne peut, en effet, se passer de négociations contractuelles que si les personnes présentes autour de la table ont le même but (la fameuse "communauté" par rapport à laquelle on positionne les notions de bien et de mal), hors intérêts externes. C'est pour ça qu'il faut être avec les "bonnes personnes", centrés autour des mêmes valeurs. Autant la tâche est plus complexe dans une société car la "communauté" doit être inclusive (sinon la notion de bien par rapport à la communauté peut vite déraiper), autant sur un projet, il est possible de choisir votre "communauté" ainsi que vos clients. Profitez-en ! Car c'est le vrai point bloquant dans la vision décrite dans **la morale anarchiste** et vous avez le luxe de pouvoir vous mettre en conditions quasi idéales.

L'agilité, la résurrection de l'anarchie ?

Haymarket, la révolution russe, la guerre d'Espagne... A force d'assassiner les anarchistes, le mouvement était tombé en désuétude avec ses valeurs et sa définition dénaturées.

Mais j'ai l'impression que cette philosophie a réapparu sous le nom d'agilité. Et ce n'est pas par hasard car au final, quoi de plus logique que demander aux personnes qui font (et donc savent) plutôt que de leur imposer une vision depuis l'extérieur ? En constatant l'échec des anciennes méthodes, l'industrie a commencé à faire preuve de pragmatisme et a préféré :

- Mettre en place des boucles de feedback pour pouvoir s'adapter ;
- Faire confiance aux équipes pour agir au mieux pour le projet.

C'est donc pour moi la philosophie anarchiste qui est née à nouveau de nos projets : lorsqu'ils se sont heurtés aux limites du système hiérar-

chique pyramidal ainsi qu'aux plans établis à l'avance par une élite dite « sachante ». Là où agilité et anarchie diffèrent, c'est que les équipes agiles sont auto-organisées, alors que l'anarchie prône l'auto-gestion. La nuance entre ces termes se trouve dans la définition des objectifs. Dans l'auto-gestion, les objectifs sont définis par la communauté, alors que dans l'auto-organisation, on vous donne la liberté de moyens pour atteindre un objectif donné. En effet, en entreprise, il semble pertinent que les objectifs à atteindre soient définis par votre patron (c'est le principe-même du salariat : l'enrôlement de personnes via un salaire pour les mettre en mouvement vers un objectif que vous désirez atteindre). D'ailleurs, c'est souvent en se trompant d'objectifs que l'on envoie ses équipes dans le mur ("implémenter Scrum", "utiliser des micro-services", etc. ne sont pas des objectifs).

Pourquoi l'agilité ne fonctionne-t-elle pas ?

Revenons enfin à notre sujet de départ :

Pourquoi ce malaise autour de l'agilité ? Alors certes, le malaise n'est pas ressenti partout, l'agilité fonctionnant dans beaucoup d'équipes et de petites structures. Mais pourquoi est-elle source de malaise dans certaines structures (votre grand compte par exemple) ? Pourquoi rien ne ressort de vos cérémonies ? Pourquoi avez-vous l'impression que le sprint est devenu une course insoutenable à la productivité ?

Peut-être avez-vous tenté d'appliquer dogmatiquement Scrum, en renommant votre chef de projet "Scrum master", en obligeant les équipes à participer à des cérémonies qu'elles ne comprennent pas ? En fait, la réponse est assez simple : vous essayez de mettre en place l'anarchie dans une structure qui veut tout contrôler. Par exemple : prévoir le coût des choses avant de les avoir définies (et donc au moment où l'on a le moins d'informations sur le sujet), avoir l'impression de savoir où on va, que tout est sous contrôle. C'est pourtant l'origine de l'agilité ! On sait que ça ne va pas se passer comme prévu, réfléchissons plutôt à comment nous adapter.

Donc oui, vous essayez de mettre en place l'anarchie dans une structure capitaliste, une de ces nombreuses entreprises qui base sa croissance sur une sur-interprétation du Darwinisme pouvant se résumer par "manger ou être mangé". Vous demandez à des personnes qui ont été recrutées pour leur docilité de faire preuve de leadership ? Et à celles recrutées pour leur autorité de déléguer ? Vous n'y arrivez pas ? Vous vous confrontez à une culture contraire ? Etes forcés de mener une "conduite

du changement"...? C'est étonnant !

L'agilité, comme elle devrait l'être effectivement, grince et a du mal à passer. Tout simplement car elle n'est pas adaptée à notre système.

On peut aisément l'expliquer par le fait que l'agilité n'est pas une évolution de l'approche traditionnelle mais une rupture avec la pensée déterministe de cette approche (à l'image de la physique quantique qui fut une rupture avec la physique traditionnelle).

L'agilité est en rupture avec l'entreprise de la même façon que l'anarchie serait en rupture avec notre société actuelle. On parle ici de remettre l'Humain au centre des stratégies.

Abandonner l'agilité ?

L'agilité va à l'encontre de la façon dont est construite notre société. C'est ce qui explique cette zone de friction permanente. Je pense sincèrement que la philosophie agile est la bonne approche, mais qu'elle n'est pas adaptée à tous les environnements. Si une méthode ne vous convient pas, arrêtez de la tordre, de la cabosser pour la faire rentrer dans votre moule. Laissez tomber et arrêtez de faire souffrir vos équipes plus qu'elles ne souffrent déjà ! Après, en ce qui



me concerne, j'aime bien cette "agilité" et ses principes. Dans l'idéal, je préférerais que l'on change de société, pour aller vers plus de confiance, plus de réalisme et plus de collaboration, plutôt que de l'abandonner. Cela permettrait qu'elle ne soit plus en rupture avec le modèle des entreprises, et on y gagnerait tous. Mais c'est un idéal encore lointain...

Sources et inspirations :

- Manifeste agile
- Baillargeon, Normand. "L'ordre moins le pouvoir : Histoire & actualité de l'anarchisme". Agone, 2008. ISBN-10: 2748900979
- Petr Alekseevitch, Kropotkine. "la morale anarchiste". Fayard (1001 nuits) ISBN-10: 2842058372
- Mikhail, Bakounine. "Dieu et l'état". Fayard (1001 nuits) . ISBN-10: 2842050746
- "Ni dieu Ni Maitre", Les sales magestés. <https://play.spotify.com/track/2qhmZuHIUmTh0yAZE3zVbY>

Merci à Olivier pour avoir collaboré à enrichir cet article.

Une journée au **Laval Virtual**



Romain Gaborieau



Johann Brancourt

Concepteurs Développeurs
SQLI Nantes



D'universitaires japonais aux grandes entreprises internationales comme HTC ou Microsoft en passant par de petits studios français, il y a de quoi plaire au plus grand nombre. Fort de l'intérêt croissant des visiteurs (professionnels et grand public) pour ces technologies, le Laval Virtual n'a cessé de croître et proposait en 2016 5200m² de surface d'exposition. Un nouveau record de fréquentation a été atteint cette année, avec 17700 visiteurs qui se sont précipités dans une ville qui a reçu le label French Tech en juillet dernier. [1]

Quelques stands intéressants

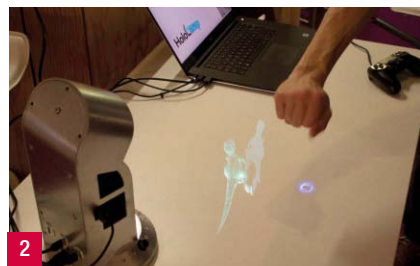
Lors de cette édition 2017, nous avons pu rencontrer bon nombre d'entreprises, de groupes de travail ou des développeurs solitaires venus présenter leurs produits ou leurs recherches. Nous ne pouvons malheureusement que regretter la floraison de projets utilisant le HTC Vive, le casque de réalité virtuelle en vogue, sorti l'été dernier. Dommage, car de nombreuses démonstrations utilisaient le même type de support ludique, ce qui laissait moins de place aux innovations. Cependant, d'autres inventions ont retenu notre attention, comme celui présenté par un entrepreneur vêtu du Perception Neuron, une combinaison de capture de mouvement. Ce créateur venu des Pays-Bas nous a présenté le framework qu'il développait depuis plus de trois ans. Il permet d'intégrer facilement et de manière réaliste plusieurs technologies de capture de mouvement (Microsoft Kinect, Leap Motion...) dans l'environnement Unity3D.

Un autre stand proposait de jouer aux échecs ou de donner à manger à un dinosaure sur sa table de salon. Comment ça marche? Simple, mais ingénieux, un cube contenant un projecteur et des caméras, détecte la position de l'utilisateur et projette sur la table les objets 3D. Comme il suit le mouvement de cet utilisateur, la projection est mise à jour selon son angle de vue, ce qui donne une impression de 3D sans avoir à porter de lunettes [2]. Le Laval Virtual,

Le Laval Virtual est le rendez-vous incontournable des nouvelles technologies et usages du virtuel en Europe. Né en 1999, il réunit tous les ans à Laval, de grands acteurs du monde de la réalité virtuelle et propose des tables-rondes, conférences et des démonstrations.



1 Le futur des fêtes foraines



2 Ces petits dinosaures donnent l'impression de sortir de la table

ce n'est pas que des casques ou des projecteurs, il y avait également le stand d'eMotion Tech, société française présentant des imprimantes 3D. Ces imprimantes sont développées et assemblées en France tout en restant accessibles (à partir de 400 €). De type delta, elles sont précises et rapides, mais il faudra supporter ses 8kg d'acier, de câbles et moteurs. Assurément solide et stable, mais un peu lourd.

Focus sur l'Hololens

Enfin, le stand qui nous a le plus marqué est celui de Holoforge Interactive qui présentait l'Hololens de Microsoft. Parmi les quelques stands qui présentaient l'appareil, c'est le seul qui avait un appareil encore chargé. En effet, victime de son succès, l'Hololens nécessite une heure de recharge pour une autonomie d'environ 3h, il est donc difficile de le faire tester à tous les visiteurs. Cette société basée à Bordeaux développe essentiellement des jeux vidéo et des applications de réalité augmentée. À l'occasion du salon, le stand réel était pour ainsi dire vide : des murs en relief et une table en son centre. Ce n'est qu'en se coiffant du casque Hololens que l'espace du stand se remplit. Les murs deviennent des boutons



3 Un clic et c'est parti...

permettant de démarrer une démo; la table au centre se retrouve à supporter la maquette d'un immeuble que l'on peut découper pour voir l'intérieur; on lève les yeux et on voit une grue de chantier taille réelle au-dessus de nous... Pour une première version de l'appareil, on notera un champ de vision "virtuel" pas très large, soit 120° (NDLR: Je doute du 120°), qui oblige à tourner la tête pour voir l'ensemble de la scène, mais dans l'ensemble, la qualité de l'image est satisfaisante, le tracking des mouvements et de l'environnement sont précis même s'ils manquent de fluidité. Un point étonnant, c'est son poids, à peu près 400g, ce qui paraît lourd à porter sur la tête, mais qui après utilisation se révèle confortable et très bien équilibré. En effet, les autres appareils de ce genre ont tendance à trop peser sur l'avant, ce qui rend leur utilisation désagréable. On appréciera également l'absence de câblage du fait qu'il embarque un Windows 10 indépendant. [3]

En bref

Le salon nous informe bien de l'évolution du marché et des technologies. Pour notre part, nous voyons plus de débouchés et de cas d'utilisation dans la réalité augmentée que dans la réalité virtuelle (en dehors des jeux vidéo); soit plus de chance pour l'Hololens que pour le HTC Vive de percer dans le milieu. Mais rien n'est joué, c'est un domaine encore très jeune et les technologies qui sont annoncées vont sûrement révolutionner notre façon de voir la réalité virtuelle.

ClusterHat : monter un cluster de Raspberry Pi Zéro c'est facile



François Tonic

Aujourd'hui, l'offre autour des mini et nano serveurs est particulièrement séduisante. Nous avons évoqué il y a quelques mois le VoCore 2, un ordinateur de la taille d'une pièce de monnaie offrant des ressources suffisantes pour en faire un serveur NAS ou un serveur Web. Mais jusqu'à présent, nous n'avions pas réellement trouvé une utilité au Pi Zéro, c'est chose faite avec le ClusterHAT.

ClusterHAT se veut une solution simple pour créer un mini cluster et pour apprendre le fonctionnement des architectures distribuées, avec une carte d'extension à connecter à sa Raspberry Pi et enfilant jusqu'à 4 Raspberry Pi Zero. Le montage prend – 5 minutes. Le plus long est de flasher les cartes micro SD, une par une.

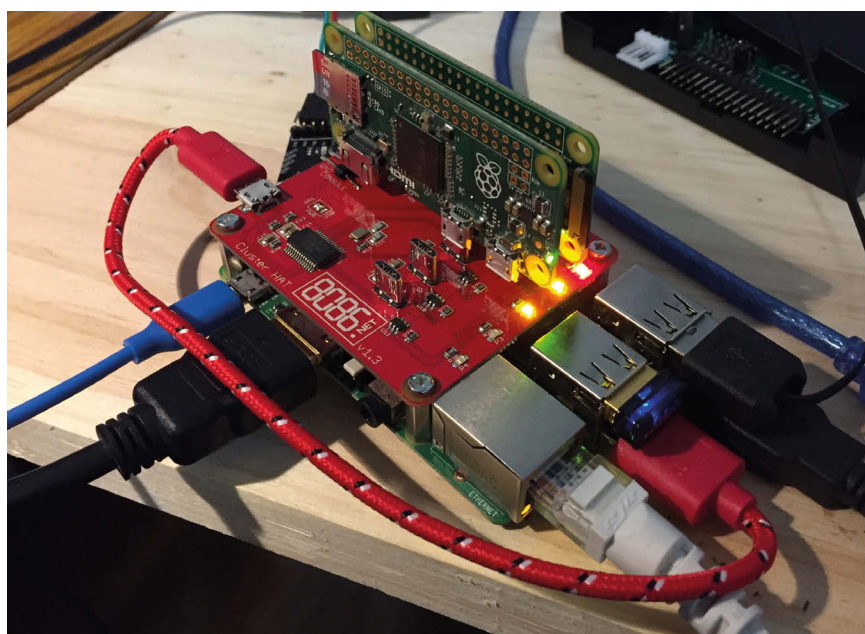
Le système

ClusterHAT se présente sous la forme d'une simple carte d'extension dotée du connecteur adapté au GPIO de la Pi. Pour stabiliser le montage, il suffit de fixer les vis, fournies. Puis de connecter le câble USB du ClusterHAT, sur un port USB de la Pi. Très simple. Sur le dessus, le ClusterHAT possède 4 connecteurs micro-USB. Il suffit d'y enfoncer ces Pi Zero. Et c'est tout. Votre cluster est monté. Attention à bien connecter le port USB de la Pi Zero et non le port micro USB servant à l'alimentation...

Vérifiez bien que les Pi Zéro soient bien enfoncées sur les connecteurs et qu'elles soient bien horizontales. C'est le petit défaut du ClusterHAT, aucun support, hormis le connecteur, ne stabilise les Pi Zero. Honnêtement ce n'est pas gênant sauf quand vous transportez sans boîtier. Chaque extension est numérotée : P1 à P4. Une LED est associée à chaque P. Si vous ne remplissez pas tous les ports du cluster, partez du P1.

Mon premier cluster

Voilà, tout est monté. Maintenant, il faut s'occuper de la partie logicielle. ClusterHAT propose une version modifiée de la Raspbian : pour le contrôleur et pour les Pi Zero. Il faut les télécharger et flasher les micro SD avec ses distributions.



1- Télécharger la distribution Raspbian modifiée : chaque carte a sa distribution ;

2- Flasher les micro SD avec la distribution et insérez la SD dans la carte concernée.

N'oubliez pas d'activer le serveur SSH qui est désactivé par défaut. Sinon, vous ne pourrez pas vous connecter en SSH. Si vous n'êtes pas habitué à Linux, bootez sur chaque SD depuis le contrôleur Pi et affichez l'outil de configuration : `sudo raspi-config`

Puis dans le menu 5 interfacing options choisir P2 SSH Enable / Disable. Vous pouvez aussi allouer tout l'espace de la SD à votre partition. Faites-le de suite et rebootez : 7 Advanced Options -> A1 Expand Filesystem

3- Connectez votre câble Ethernet puis allumez le contrôleur Pi ;

4- Vous devez activer les Pi Zero sur le ClusterHat avec la commande : `clusterhat on P[numéro]`. Vous pouvez toutes les activer via le paramètre `all` ou indiquer les ports du cluster, par exemple : `clusterhat on P1 P2`. Le système met quelques minutes pour se charger, soyez patient ;

5- Connectez-vous sur les 3 cartes via SSH : `ssh pi@[hostname].local`. Le mot de passe par défaut est `raspberrypi` ;

6- Installez sur toutes les cartes Docker : `curl -sSL https://get.docker.com | sh`

Installez curl si nécessaire ;

Vous pouvez aussi automatiser le déploiement et l'installation docker en faisant :

`for host in controller P1 P2; do ssh pi@$host curl -sSL https://get.docker.com | sh; done`

ID	NAME	IMAGE	NODE	DESIRED STATE	CURRENT STATE	ERROR
ubfeioi8tapv	counter.1	alexellis2/arm_redis_counter:latest	p2	Running	Preparing 10 seconds ago	
9isw7dlhrtzg	_ counter.1	alexellis2/arm_redis_counter:latest	p1	Shutdown	Preparing 9 minutes ago	
ub4pcw8sc4jt	counter.2	alexellis2/arm_redis_counter:latest	p2	Running	Preparing 9 minutes ago	
sfnj3dnt7b0r	counter.3	alexellis2/arm_redis_counter:latest	controller	Running	Running 1 second ago	
lbyzdld5g8hj	_ counter.3	alexellis2/arm_redis_counter:latest	p1	Shutdown	Preparing 2 minutes ago	
upxcoh93q72	counter.4	alexellis2/arm_redis_counter:latest	controller	Running	Running 2 minutes ago	

Si tout se passe bien la commande docker version fonctionnera sans problème.

7- sudo iptables -P FORWARD ACCEPT ;

8- docker info | grep Swarm : opération pour toutes les cartes. Puis nous activons et créons le cluster swarm via la commande docker swarm init : docker swarm init -advertise-addr [IP du contrôleur]. On déclare en premier le leader du cluster qui sera le controller (donc notre Pi) ;

9- Le swarm init fournit le docker swarm join avec le token et l'ip pour permettre de créer les nœuds sur nos Pi Zero. Copiez la commande dans sa globalité sur les Pi Zero.

Par exemple : docker swarm join \

```
--token SWMTKN-1-1b5f0zeuw0k2xyns55c2qsriqodzi-  
flwb6ku46i9t8i4w66s5k-7h6lkjw20yjke6p3gg0cwl4md \
```

192.168.1.95:2377

Si tout se passe bien, sur votre pi@controller en faisant un docker node ls vous verrez tous les nœuds de votre cluster docker.

10- Il ne reste plus qu'à déployer vos conteneurs sur le cluster. Les nœuds et les compteurs sont gérés dynamiquement comme vous le verrez en montant et en démontant les Pi Zero. Cependant, il faut prendre en compte le temps démarrage du nœud pour le voir de nouveau dans le docker node.

Attention : vous aurez peut-être besoin de fixer les hosts et hostnames et les IP :

sudo nano /etc/hosts

Vous pouvez y modifier et rajouter les IP et les hosts correspondants. Pour les hostname, il faut modifier le fichier /etc/hostname et y rajouter tous les hostnames nécessaires, dans notre cas : Controller

P1

P2

Et faites un reboot pour prendre en compte les modifications.

PS : bien entendu, le sudo reste votre ami.

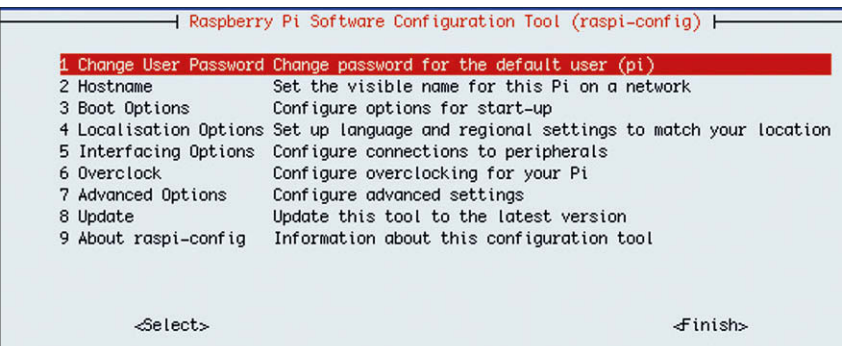
Le budget

Matériel	Quantité	Prix total (€)
Raspberry Pi 3	1	39,60
Raspberry Pi Zéro	4	22
Carte micro SD (8 Go)	5	20
ClusterHAT	1	40,90
Câble réseau Ethernet 3m	1	6
Total		128,50

Dans notre montage, nous avons 2 Pi Zéro et 3 micro SD de 16 Go et nous avons déjà le câble. Le budget est légèrement optimisé : 113,50 € TTC. Ce qui reste bien moins cher qu'un barebone de type Intel NUC ou MSI.

```
pi@controller:~$ docker node ls
```

ID	HOSTNAME	STATUS	AVAILABILITY	MANAGER STATUS
kk1hkks8noxjxcjf1pw6qxqfw *	controller	Ready	Active	Leader
npj3bmncwv3bf6f2p3rhf62e1	p1	Ready	Active	
s9vzbz4q1bm44b0eub2l43x	p2	Ready	Active	



PICOCLUSTER : L'AUTRE CLUSTER PI

PicoCluster propose une large gamme de clusters "économiques" pour Pi 3, Pine64 et Odroid C2. Ce sont des cubes à monter ou préconfigurés. Ils contiennent les cartes, le switch réseau, le cube, les cartes SD et les câbles. Bref, un cluster complet qu'il suffit de brancher et d'administrer. Les options sont nombreuses : nombre de cartes, réseau (100 ou 1000), capacité des SD. La version 3 cartes démarre à 273 \$, sans les cartes SD. La version 10 cartes, est proposée à partir de 909 \$ (en Pi 3). Des dissipateurs de chaleur sont inclus car les Pi 3 ont tendances à chauffer.

L'avantage de ces Pico est de proposer un package complet et des montages plutôt propres, cartes incluses. Cette offre se destine plutôt pour les utilisateurs avancés ou les entreprises. Difficile de faire une comparaison directe avec le ClusterHAT qui est clairement en retrait sur le design et les prestations.

Ces clusters visent les entreprises.

Rajoutez à ce tarif le boîtier de protection, si nécessaire.

Notre avis

Honnêtement, pour le prix de la carte, le résultat est plutôt satisfaisant même s'il faut passer par la distribution conseillée par le constructeur pour pouvoir manipuler les nœuds et activer chaque Pi Zero. Le montage est propre et relativement stable, si on n'y touche pas trop. Monter un docker swarm n'est pas forcément très compliqué mais si vous n'êtes pas trop habitué au clustering et à docker on perd du temps à tout déployer et faire les bonnes commandes ; mais ensuite, on va très vite. Les adresses IP restent un point sensible ainsi que la gestion de l'espace disponible sur les SD. Ce point est à surveiller. Entre le manque d'espace (pensez à étendre les partitions et les tmp) et les problèmes d'IP quand vous rebootez le cluster, vous perdez beaucoup de temps.

Le manque de documentations pénalise aussi, même si on utilise du Raspbian, le constructeur

doit mieux faire même s'il existe un groupe Google actif autour du ClusterHat : <https://groups.google.com/forum/#!forum/clusterhat>

C'est le côté frustrant du ClusterHat : on sent le potentiel du montage et son intérêt pour comprendre les mécanismes mais quand on ne connaît pas le monde du cluster, on est vite perdu.

Les +

- Le prix
- La compacité de la carte
- Facilité d'installation
- Qualité de fabrication

Les -

- Pas de miracles sur les performances
- Administration sensible
- Le design pour connecter le câble USB
- Documentation et tutoriaux limités
- Instabilité

Découvrez **tangente** le magazine des mathématiques

l'aventure mathématique



Pour mieux comprendre le monde

tangente
l'aventure mathématique

Aussi en version numérique !

Tous les deux mois
chez votre marchand de journaux

Politique, économie, finance,
jeux, musique, littérature,
arts plastiques, architecture,
informatique, physique,
biologie, géographie :
les mathématiques sont partout !!!
Le magazine *Tangente*
et ses hors séries vous aident à
redécouvrir notre quotidien.



Les hors séries « kiosque »

4 fois par an, un hors série
d'au moins 56 pages, explore
un grand dossier de savoir
ou de culture. Derniers parus :
Les angles - Les graphes -
Les démonstrations - Les fonctions -
Maths des assurances -
Maths et médecine - La Droite -
Maths et Architecture - Les ensembles -
Disponibles chez votre marchand de journaux
ou avec l'abonnement PLUS.

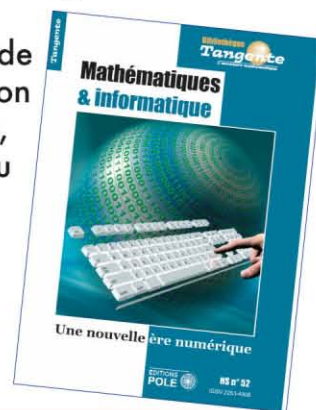
Vous voulez vous rendre compte
de ce qu'est la consultation numérique
d'un numéro de *Tangente* ?

**L'accès au numéro 167
vous est offert !**

Rendez-vous sur <http://tangente-mag.com>
(identifiez-vous)

La « Bibliothèque Tangente »

Pour les lecteurs les plus curieux,
les articles des hors séries de *Tangente* sont
repris et complétés dans la
Bibliothèque Tangente, avec de
magnifiques ouvrages d'environ
160 pages, richement illustrés,
disponibles sur la boutique du
site www.infinimath.com
ou chez votre libraire ou avec
l'abonnement SUPERPLUS.



<http://tangente-mag.com>

POUR OBTENIR « MATHÉMATIQUES ET INFORMATIQUE » -

Rendez-vous sur www.infinimath.com/librairie et achetez-le avec le code promo PROG.

Vous aurez la gratuité des frais de port sur tout votre panier. Vous pouvez aussi envoyer ce bulletin
à TANGENTE - 80 BD SAINT-MICHEL - 75006 PARIS accompagné d'un chèque de 19,80 €

NOM* PRÉNOM*
ADRESSE*
CODE POSTAL* VILLE* PAYS
MAIL* TÉLÉPHONE*

RSA San Francisco

Machine learning & IA, les alliés de la cybersécurité

• Véronique Loquet
@vloquet

La conférence RSA 2017 a affiché un nombre record de visiteurs. 43 000 professionnels de l'industrie de la sécurité informatique ont convergé au Moscone Center de San Francisco pour une semaine infernale. Parmi les sujets qui secouent l'infosec circus, l'IA et le machine learning.



Les systèmes intelligents et autonomes s'imposent dans notre paysage et vont devenir omniprésents dans le domaine de la sécurité des systèmes d'information. Une pléiade d'outils de détection s'appuient désormais sur l'automatisation. RSA accueillait des dizaines d'offres basées sur le machine learning, et nombre d'entre elles ont prouvé leur efficacité dans la détection de logiciels malveillants, c'est par exemple le cas de Cylance qui parviendrait à en intercepter 99 %, vulnérabilités inédites comprises. L'usage de la technologie d'apprentissage automatique est un gain de temps considérable pour les équipes de sécurité. Le machine learning se positionne en renfort sur des tâches où l'automatisation est mature, comme moyen de réduire le bruit des alertes, de trier les faux-positifs, en mettant en évidence les problèmes de sécurité à traiter en priorité. Principalement utilisés dans la détection des logiciels malveillants, l'analyse dynamique des risques et la découverte d'anomalies, ces systèmes reposent sur l'activité de surveillance en temps réel et les usages non adaptés. L'analyse du comportement des utilisateurs fait l'objet de recherches poussées pour évaluer les risques et détecter les attaques internes où ciblées, à l'aide d'algorithmes et d'une collecte de données souvent massives. Au delà de la détection d'intrusion, l'IA peut prédire et signaler les intentions malveillantes. Lorsqu'il n'y a pas de modèles pré-établis, il s'agit d'apprentissage automatique «non supervisé». Lorsque les

modèles sont formés à l'avance cet apprentissage machine est dit «supervisé». Les compétences machines s'améliorent au fil de la pratique, après un certain nombre de répétitions, le résultat souhaité devient automatique.

Quid de la sécurisation de l'Internet des objets (IoT) ?

Lors de sa keynote à RSA, Eric Schmidt, co-fondateur de Google et Chairman de Alphabet, avançait que « l'intelligence Artificielle serait utile pour apporter de la sécurité aux objets connectés. ». Un défi car ils seront plus de 30 milliards en 2020 selon IDC, et ne pourront être uniquement sécurisés par l'homme. L'intelligence artificielle et l'apprentissage machine seraient donc une aide précieuse pour la sécurité des IoT. La course pour leur sécurité est d'ores et déjà lancée car malgré leurs vulnérabilités, l'usage des périphériques connectés augmente de façon spectaculaire dans tous les domaines : domotique, santé, finance, armement, énergie, transport... Une étude de Hewlett Packard Enterprise relève que plus de 70% d'entre eux ne possèdent pas de chiffrement et que la sécurité n'est simplement pas intégrée à leur conception, menaçant ainsi une quantité énorme de données. Bien que dans certains cas le maillon faible peut se trouver du côté applicatif client ou serveur, les attaques sur les dispositifs IoT diffèrent des attaques traditionnelles car elles ne nécessitent pas obligatoirement d'accéder aux données et aux

outils que nous avons pris l'habitude de protéger. L'attaquant peut agir directement sur le matériel, où à distance via les réseaux et protocoles de communication, il peut aussi détourner l'équipement pour l'intégrer à un botnet, comme on l'a vu avec l'attaque géante par déni de service du malware Mirai infiltré dans plus de 150 000 caméras de surveillance. Si l'absence de normes dans le secteur des IoT ne semble pas être un frein à l'adoption, elle l'est pour la sécurité.

L'Intelligence Artificielle pose question

Dans le domaine de l'IA, les considérations éthiques sont un facteur déterminant. L'apprentissage algorithmique des machines s'associe à une collecte massive de données pour déterminer les profils et les comportements humains, met en jeu la protection des données à caractère personnel, mais pas seulement puisque l'industrie développe aussi la prise de décisions autonome des machines. L'US Air Force par exemple, prévoit l'utilisation de drones de combats entièrement autonomes, dotés d'une capacité de décision ultra rapide pour frapper l'ennemi. Par ailleurs le flou juridique devrait évoluer vers un niveau de sécurité standard. Le Parlement européen vient de prendre une mesure pour introduire une nouvelle réglementation via la Commission européenne sur la robotique et l'intelligence artificielle. Le rapport aborde différents types de robots et objets connectés devant répondre aux exigences de sécurité, afin que l'UE et ses États membres puissent maintenir le contrôle des normes réglementaires. Les États-Unis, le Japon, la Chine et la Corée du Sud, développent déjà leurs propres règles en matière d'intelligence artificielle. De son côté l'UE vise à prendre des mesures préventives pour s'assurer que tout vaste déploiement de la robotique ne tombe pas en dessous des normes établies. Ces projets de régulations seront sans doute suivis par les développeurs, les fabricants et les utilisateurs d'IA, avec la difficulté pour le législateur de se mettre à niveau et faire preuve de souplesse pour ne pas freiner l'innovation. •

L'art du Debug

• Sarah Buisson
Ingénieur Java et passionnée
de Craftmanship,
chez **Xebia**.



Souvent ignorés, le debug et la maintenance applicative sont pourtant des domaines de compétence à part entière : avec des outils et des méthodologies qui leur sont propres. En raison de ce manque de reconnaissance, la plupart des développeurs debugge sans chercher à étendre ou à transmettre leurs compétences dans ce domaine. Cet article vous présentera les outils et méthodologies de debugging les plus courants, applicables avec tous les IDE et langages modernes.

Qu'est-ce qu'un bug ?

Un bug est une différence entre le comportement attendu et le comportement obtenu. Ce n'est pas à proprement parler une erreur dans le code. C'est son symptôme. Quelque part en amont du projet se cache une instruction erronée : l'origine du bug (ou root-cause).

Déboguer, c'est donc chercher et corriger l'origine d'un bug.

UN PEU DE THÉORIE ET DE MÉTHODOLOGIE

La base de toutes les stratégies de debugging repose sur 4 points :

- Reproduire le bug ;
- Diagnostiquer son origine ;
- Corriger ;
- Conclure.

Étape 1 : reproduction de l'erreur

Avant toute chose, vous devez commencer par essayer de reproduire l'erreur. Pourquoi ? Tout simplement parce que si vous ne reproduisez pas l'erreur :

- Vous ne saurez pas si l'anomalie existe réellement ;
- Vous n'aurez jamais la certitude que votre correction ait été efficace.

Dans un monde idéal, la personne remontant le bug donne un compte-rendu détaillé de comment reproduire le bug, sa présence est confirmée par plusieurs personnes, le comportement attendu est précisé.

Dans la réalité, c'est parfois plus compliqué.

Un scénario de reproduction peut être la suite d'instructions remontée par la recette pour produire le bug, un test unitaire qui ne passe plus, etc. Tout ce qui vous permettra de reproduire facilement et à volonté l'erreur.

L'importance du contexte

Déboguer en local (dans un environnement de développement) est plus simple que de debugger un test d'intégration continue retourné en erreur. Cependant déboguer l'Intégration Continue reste plus facile que de corriger une erreur remontée par l'équipe de recette pendant les tests manuels. Erreur qui sera moins coûteuse que le bug remonté par un utilisateur final depuis la production. Plus vous montez dans les environnements, moins vous aurez d'outils et plus vous aurez un impact sur le travail des autres. L'idée est de parvenir à détecter les bugs le plus tôt possible d'une part, mais surtout de les reproduire sur des environnements les plus proches possibles de vous afin de bénéficier de plus de liberté et des meilleurs outils à votre disposition.

Le processus de remontée de l'anomalie

Le processus de remontée des anomalies influence énormément la facilité avec laquelle vous reproduirez le bug. N'hésitez pas à remonter vos

besoins aux équipes de recettes. Idéalement, chaque anomalie remontée doit :

- Indiquer comment reproduire l'anomalie ;
- Préciser quels sont le comportement attendu (spécifications fonctionnelles) et constaté (capture d'écran, stack-trace, identifiants, etc.) dans quel contexte (sur quelle plateforme, pour quel utilisateur, sous quel navigateur, etc.) ;
- Indiquer la priorité de l'anomalie (mineure, prioritaire, bloquante) ;
- Ne remonter qu'une anomalie à la fois (avec un éventuel lien vers d'autres bugs liés).

Kiss: Keep It Simple And Stupid: reproduction et tests unitaires

Reproduire le bug, c'est bien. Le reproduire facilement, c'est mieux. Lors des phases de diagnostics et de correction, vous allez devoir reproduire le bug à de nombreuses reprises. C'est pour cela qu'il vous faut alléger le scénario de reproduction au maximum, jusqu'à aboutir à un cas de test le plus simple possible (mais qui reproduit toujours l'erreur).

L'idéal est de reproduire le bug via un test unitaire, afin de pouvoir le tester d'un simple clic. Cela vous permettra de vous focaliser sur l'analyse et la correction du bug, et non sur un scénario de reproduction trop compliqué.

Et une fois le bug corrigé, que faire du test ?

Idéalement, gardez-le et modifiez les cas de tests existants. En effet, ce n'est pas un seul bug que vous avez mais deux. L'un dans le programme, l'autre dans vos tests, qui ont laissé le bug passer inaperçu jusqu'à présent. Les modifications dans les tests font donc partie intégrante du correctif : elles doivent être livrées avec.

Delta debugging

Supposons que votre bug ait un scénario de reproduction très complexe. Le delta debugging consiste à supprimer et remettre une-à-une les étapes du scénario de reproduction, afin de déterminer lesquelles sont nécessaires à la reproduction de l'anomalie. Si la disparition d'une étape cause la disparition du bug (ou l'apparition d'un autre bug) alors nous la gardons, sinon nous la retirons du scénario.

Les bugs impossibles à reproduire

Parfois, vous ne parviendrez pas à reproduire le bug. Cela ne veut pas dire qu'il est corrigé pour autant. Vous devez trouver pourquoi le bug n'est pas reproductible :

- Si le bug a été corrigé par un autre correctif, vous devez trouver lequel ;
- Si le bug est intermittent, vous devez isoler les causes qui le rendent inconstant, afin de pouvoir reproduire le bug systématiquement. Nous reparlerons du traitement des bugs intermittent dans la suite de l'article ;

- Si c'est une question de données, vous devez importer ou fabriquer les données nécessaires en local ;
- Si c'est une question d'habilitation, "pair-debuggez" avec quelqu'un habilité à reproduire le bug ;
- Si c'est une question de plateforme, vous devrez vous contenter des outils disponibles sur la plateforme sur laquelle est reproductible le bug ;
- Si c'est une question de timing : vous devez trouver le bon enchaînement d'actions qui provoque le bug, en vous basant sur les logs existants. À noter que l'ajout de logs peut perturber la reproduction des bugs de timings.

Étape 2 : formulation d'hypothèses et diagnostics

Une fois l'erreur reproduite, vous pouvez l'analyser pour déterminer son origine. Lire le code et l'exécuter ligne par ligne est certes une bonne pratique, et vous permettra de découvrir le projet, mais implique de lire, comprendre et maîtriser pratiquement l'intégralité du code avant de trouver la cause. Heureusement, il existe d'autres stratégies vous permettant de restreindre votre champ de recherche et de "cibler" la cause de l'erreur.

Notions de base

Vérifiez ce qui a été modifié récemment.

Les bugs n'apparaissent pas spontanément. Une anomalie qui ne s'est jamais produite auparavant n'a que 3 causes possibles :

- Soit le cas de tests n'avait jamais été vérifié auparavant ;
- Soit le contexte du projet a changé (nouvelles données, nouvelles dépendances) ;
- Soit une erreur a été récemment introduite dans le code.

Voyez large.

Ne vous limitez pas à une seule hypothèse sur la source du bug. Formulez-en plusieurs, triezy-les de la plus évidente à la moins probable et vérifiez-les toutes. Ne vous restreignez pas à une seule section du code. L'erreur n'est pas forcément là où vous regardez.

Générez plus de données pour plus d'hypothèses.

Ne vous limitez pas au scénario de reproduction original. Essayez de reproduire le bug de plusieurs façons, et variez les données de vos tests. Ainsi, vous pourrez déterminer les points communs entre les différents scénarii reproduisant l'erreur. N'oubliez pas de tirer des conclusions des scénarii ne reproduisant pas le bug : ils vous permettent d'invalidier certaines de vos hypothèses.

Isolez votre code.

Mockez ("Bouchonner") les parties du projet, vous permettra de réduire les interférences et ainsi de concentrer votre attention sur l'erreur.

Parlez du bug à vos collègues.

Expliquer le problème à quelqu'un d'autre aide à la compréhension du bug ("Rubber Duck Debugging"). Il est possible qu'ils aient eu affaire à des situations similaires dans le passé et qu'ils puissent vous conseiller certains outils.

Diagnostic par observation

Il s'agit d'obtenir autant d'informations possibles sur le bug en se basant sur l'exécution du programme : soit en consultant les logs, soit en pla-

çant des points d'arrêt sur le chemin du bug, afin de trouver la cause de l'erreur. Cette approche nécessite d'avoir une bonne connaissance du programme ou de l'architecture à déboguer. Il faut être capable de cibler rapidement les classes susceptibles de contenir l'origine du bug.

Au sujet du log-debugging

Debugger en ajoutant des logs est sujet à débat : alors que certains utilisent cette pratique régulièrement, d'autres la rejettent fortement.

Le problème est que les logs sont un outil de debug aussi puissant que dangereux :

Ils vous donneront des informations précieuses sur les exécutions passées, mais ils ont un impact sur l'exécution du programme et sont soumis à diverses réglementations suivant les projets. À noter que pour les bugs reproductibles en local et une connaissance avancée de votre Environnement de Développement Intégré (IDE), vous permettront d'arriver au même résultat que les logs aussi facilement, sans recompilation ni risque d'effets de bord. Si en revanche l'anomalie ne se manifeste que sur une machine distante, il sera souvent plus simple de mettre en place davantage de logs plutôt qu'un remote debug.

Diagnostic par induction

Pour rappel, l'induction représente la logique suivante : "si A, alors B" entraîne "Si non B, alors non A".

L'idée consiste à partir d'une version de l'application qui fonctionne, pour progressivement arriver à la version sur laquelle le bug se produit. Prenons l'exemple d'une régression. L'anomalie est brusquement apparue avec la dernière release. Nous allons charger la dernière version du programme sans le bug, tester la présence de l'anomalie, puis récupérer le commit suivant, retester, etc. Jusqu'à isoler sur le commit qui contient l'anomalie. On pourra alors centrer la recherche exclusivement sur les fichiers modifiés par le commit ayant introduit le bug.

Tous les outils de versionning conviennent à cette stratégie, mais les fonctionnalités telles que git blame et git bisect peuvent vous faire gagner du temps en ciblant plus efficacement les commits à tester.

Stratégie Back-tracking : le debug en marche arrière

C'est une approche très populaire de debugging.

Il s'agit de chercher la cause directe du bug remonté (le symptôme), puis la cause de cette cause, etc. plusieurs fois de suite, jusqu'à trouver l'origine initiale du bug.

Nous trouvons ainsi l'enchaînement d'instructions ayant provoqué le bug pour aboutir à son origine.

Dans la pratique :

- Commencez par placer un point d'arrêt sur une ligne de code où le bug a été constaté ;
- Exécutez le scénario de reproduction ;
- À partir des variables d'exécution et du code, cherchez ce qui a pu provoquer le bug, et placez un point d'arrêt sur chaque ligne de code responsable ;
- Exécutez le scénario de reproduction. Analysez. Ne gardez que les points d'arrêt où le bug a été constaté ;
- Recommencez.

Vous finirez ainsi par trouver l'origine du bug. À noter que certains langages et IDE offrent la possibilité de remonter le fil d'exécution en marche arrière, évitant ainsi de refaire plusieurs fois le scénario de reproduction. Cette approche n'est malheureusement efficace que si l'on a réussi à isoler, au moins en partie, la portion de code contenant le bug, au risque de devoir parcourir l'intégralité de l'application.

Où placer vos points d'arrêt

Chaque programme possède des sections critiques qui doivent être analysées en priorité : les points d'entrée, les appels aux APIs tierces, le début d'une méthode publique, l'entrée dans un algorithme complexe, etc. Placer un point d'arrêt dans ces sections du code vous donnera de précieuses informations sur l'état du programme : si l'anomalie est déjà survenue ou non, mais aussi si des signes précurseurs de l'anomalie sont déjà apparus (nombreux champs vides, listes d'erreurs pleines, données incohérentes, etc). Trouver ces points vient avec l'expérience sur le projet.

Saff Squeeze

Le Saff Squeeze (du nom d'une technique de plaquage au football américain) consiste à valider et retirer un à un les éléments valides d'une méthode à tester, afin de prendre "en sandwich" l'erreur. En partant d'un test reproduisant l'anomalie, avec une assertion qui échoue :

- "Inlinez" la fonction à tester, (remplacez un appel de la fonction par son code, par exemple via la fonction *inline* de votre IDE) ;
- Ajoutez de nouvelles assertions au cœur du code de la fonction inline, testant le bon fonctionnement de cette méthode ;
- Lancez le test : certaines assertions vont passer, d'autres vont échouer ;
- Nettoyez : bouchonnez (mockez) les fonctions dont les assertions sont passées, et supprimez le code après l'assertion échouée. Les fonctions ayant échoué se retrouvent ainsi prises en sandwich entre les mocks d'un côté et les assertions de l'autre ;
- Recommencez : Inlinez les fonctions restantes et ajoutez de nouvelles assertions.

Progressivement, vous arriverez à un test de plus en plus unitaire, qui à la fois reproduit l'erreur et cible précisément son origine. À noter qu'il n'y a que peu d'intérêt à comiter le test unitaire final généré : il ne fait que dupliquer le code en erreur. Il est souvent plus intéressant de garder celui d'une ou deux itérations précédentes.

Étape 3 : Correction du bug

Vous avez réussi à reproduire et identifier la cause de l'erreur, félicitations ! Nous allons enfin pouvoir littéralement "déboguer", c'est-à-dire retirer le bug. La correction implique de concevoir et mettre en place les modifications qui corrigent le problème, sans induire de régression, et en respectant les standards de qualité du projet.

Comment éviter les régressions ?

Les régressions sont un sujet complexe et récurrent, auquel il faudrait plus d'un article pour y répondre. Néanmoins, une bonne compréhension du projet, aussi bien d'un point de vue technique que fonctionnel, une bonne couverture de tests (manuels et/ou automatiques) indépendante du développeur et des processus de qualité stricts (relecture de code, continuous testing / integration, métriques de qualité, alertes automatisées) permettent d'éviter la plupart des régressions.

Au sujet des quickfixs

Parfois, vous n'aurez pas le temps (ou l'opportunité) de faire les choses bien. Il est alors possible d'utiliser le quickfix : une rustine, un code temporaire ne respectant pas les standards de qualité habituelle ou tous les cas de test. Cependant, ayez en tête :

- L'utilisation du quickfix a un impact fort sur la qualité du projet et son futur. Cette décision ne doit pas être prise unilatéralement par le développeur. En revanche, il devra argumenter les avantages et inconvénients de son utilisation ;

- Un quickfix doit être temporaire : il doit être associé à une nouvelle story, qui apportera la correction finale propre ;
- L'information doit être conservée : votre quickfix va masquer une partie des informations relatives à l'erreur, rendant difficile la correction finale. Stockez quelque part les logs d'avant le quickfix, vos hypothèses concernant le bug, etc. ;
- Le code d'un quickfix ne doit pas être confondu avec le reste du projet et doit être visible du premier coup d'oeil : par exemple en les commentant d'un `//TODO: Quickfix de la STORY-123`

Étape 4 : la conclusion du bug

Votre travail ne s'arrête pas une fois le bug corrigé.

Comment ce bug a-t-il pu se produire ? Comment a-t-il pu passer inaperçu jusqu'à aujourd'hui ? N'y a-t-il pas d'autres bugs similaires déjà existants ? Comment aurais-je pu identifier le problème plus facilement (plus d'informations sur le scénario de reproduction, plus de documentation) ? Comment s'assurer que ni ce bug, ni un bug semblable ne se produise à nouveau ? La documentation a-t-elle été mise à jour ?

L'équipe de qualification a-t-elle toutes les informations nécessaires pour tester la correction ?

Il vous faudra répondre à ces différentes questions, en apportant des solutions si nécessaire.

En Bonus : la stratégie complète pour les bugs intermittents

Vous avez essayé d'isoler toutes les causes possibles, mais rien à faire : votre bug reste intermittent ou impossible à reproduire sur votre machine locale. Le problème est double : non seulement l'anomalie va être difficile à diagnostiquer, faute de pouvoir être observée "in vivo", mais en plus il va être difficile de confirmer l'efficacité du correctif.

Pour reproduire, diagnostiquer et corriger un bug intermittent, cette stratégie repose sur des outils de monitoring (via des points d'arrêt, des metrics, des logs, etc. suivant la plateforme où le bug est reproduit) :

- À chaque fois que l'anomalie est reproduite, nous la monitorons ;
- Une fois suffisamment de données de monitoring recueillies, l'analyse de ces données permettra de trouver l'origine du bug ;
- Cette erreur originelle va également être monitorée afin de valider qu'il s'agit bien de la cause du bug : à chaque trace de l'erreur originelle, devrait correspondre une trace de l'anomalie reproduite ; Si les deux monitoring ne correspondent pas, alors l'hypothèse est fausse.
- Une fois l'origine du bug identifiée, nous allons effectuer la correction de l'anomalie, tout en gardant les traces. Il est en effet important de vérifier que les traces de l'erreur originelle ne soient plus associées à celles confirmant la présence de l'anomalie.
- Une fois seulement cette vérification effectuée, nous pouvons retirer les outils de monitoring.

Cette méthode de résolution est très lente. Il n'est pas inhabituel de devoir attendre plusieurs semaines avant de recueillir suffisamment d'information de monitoring. Vous pouvez éventuellement envisager de livrer un quickfix en attendant.

Un exemple d'utilisation :

La situation : tous les dossiers doivent être affichés avec un utilisateur.

Le bug : certains dossiers sont affichés sans utilisateurs.

Le problème : je ne parviens pas à reproduire l'anomalie, tous mes dossiers sont créés avec utilisateur.

Je vais donc suivre la méthodologie précédente :

- Je rajoute un log à chaque fois qu'un dossier est affiché, indiquant s'il a ou non un utilisateur. Au bout d'une semaine, je récupère les logs sur les différents dossiers sans utilisateur. Leur analyse me permet de constater qu'il s'agit de dossiers sous-traités, avec un fort turnover des utilisateurs.
 - Je formule 3 hypothèses :
 - Soit ces dossiers n'ont jamais eu d'utilisateurs, (hypothèse 1) ;
 - Soit les dossiers ont eu leur utilisateur retiré, (hypothèse 2) ;
 - Soit les utilisateurs ont été supprimés. (hypothèse 3).
 - Je rajoute des logs pour chacune de ces hypothèses :
 - Pour chaque création de dossier utilisateur, (scénario 1) ;
 - Pour chaque modification de dossier sans utilisateur, (scénario 2) ;
 - Pour chaque suppression d'utilisateur, (scénario 3).
- Au bout d'une semaine, je consulte à nouveau ces logs et constate :
- Aucun dossier n'a été créé sans utilisateur, mais des dossiers ont néanmoins été affichés sans utilisateur (hypothèse 1 invalidée).
 - Certains dossiers ont eu leurs utilisateurs retirés, mais aucun de ces dossiers n'a été affiché par la suite (hypothèse 2 invalidée).
 - À chaque fois qu'un utilisateur est supprimé, le dossier est affiché sans utilisateur (hypothèse 3 validée).
 - Je corrige donc l'hypothèse 3. Désormais la suppression d'un utilisateur n'est possible que s'il n'a plus aucun dossier.
 - Une semaine plus tard, je consulte les logs :
 - Des utilisateurs ont de nouveau été supprimés .
 - Aucun dossier sans utilisateurs n'a été affiché.

L'anomalie est donc corrigée, je peux retirer les logs.

DEBUG ET DEBUGGER : AU-DELÀ DES POINTS D'ARRÊT

Comment fonctionne un debugger ?

Pour la majorité des langages managés/interprétés modernes, ce n'est pas l'IDE mais la machine d'exécution elle-même qui s'exécute en mode "debug", en mettant à disposition un port distant. L'IDE se contente de se connecter à ce port, et ce que l'on appelle communément "débugger" dans un IDE désigne en fait l'IHM entre le programme et l'utilisateur, qui fait en permanence le mapping entre le code source et le code compilé en cours d'exécution, donnant ainsi l'illusion que c'est l'IDE qui interprète et exécute le programme.

Machine d'exécution : la JVM de Java, la CLR .NET, le moteur Javascript etc., c'est-à-dire le tourne-disques sur lequel se joue votre programme

Pour les langages compilés et sans machine d'exécution virtuelle, le code déjà compilé est modifié directement en mémoire par le debugger lors de l'exécution, afin de lever une exception à chaque ligne de code. Ces exceptions seront attrapées par le debugger qui choisira de marquer l'arrêt, exécuter la ligne, etc.

Dans tous les cas, le debugger a un accès complet à la mémoire, aux threads et au code (source ou compilé) du programme en cours d'exécution.

Fonctionnalités méconnues du debugger

Au-delà du point d'arrêt et du pas à pas, la plupart des debuggers modernes proposent des méthodes avancées.

Points d'arrêt avancés

Au-delà du point d'arrêt standard, vous avez également :

- Le conditionnel, qui ne s'activera que si l'instruction de votre choix est validée ;
- L'itératif, actif après seulement un certain nombre d'itérations ;
- Des points d'arrêt à chaque accès/modification d'une variable ;
- ... À chaque instanciation d'une classe ;
- ... À chaque levée d'exception ;
- Seulement actif pour une instance donnée (breakpoint Instance) ;
- Des points d'arrêt reliés entre eux, actifs seulement si un autre point d'arrêt lié a été activé précédemment ;
- Des filtres, afin de stopper le process seulement si l'exécution est passée par certaines classes/packages au préalable ;
- Des points d'arrêt valables seulement sur une instance d'une classe avec un hashcode particulier.

Il est bien évidemment possible de combiner les différents types de points d'arrêt (par exemple : s'arrêter uniquement sur les `NullPointerException` lancés depuis le package `monprojet.service`). Prenez le temps de chercher ces options dans votre environnement de développement.

Navigation avancée du debugger

Outre les classiques "aller à la ligne suivante", "rentrer dans la méthode" et "ignorer tous les points d'arrêt", il vous sera possible de :

- Exécuter un extrait de code sélectionné / saisis ;
- Aller jusqu'à la ligne de mon curseur ;
- Continuer jusqu'à la fin du bloc ;
- Ignorer une ligne de code ;
- Retourner en arrière, jusqu'au début du bloc d'exécution (avec ou sans conservation de la mémoire).

Encore une fois, ces fonctionnalités ne sont pas présentes sur tous les IDE ni possibles pour tous les langages.

Une bonne connaissance des raccourcis clavier de la navigation avec le débogueur vous permettra de gagner en efficacité lors de vos sessions de debugs.

Call Stack, Variables et Watcher avancés

La plupart des IDE propose de visualiser la call stack en cours (la "suite de méthodes appelantes"), et les variables accessibles pour le scope en cours. Certains proposent également de filtrer les étapes de la call stack pour masquer les méthodes de votre framework d'inversion de dépendance par exemple. Il est également possible de modifier ces variables "à la volée", et de créer vos propres "vues" dans le Watch, ainsi que de positionner des points d'arrêt dessus actifs à la consultation et/ou changement sur les variables et vues.

Spécificités des debugger front

Les debugger intégrés aux navigateurs Web sont généralement moins avancés que ceux intégrés aux IDE, mais proposent d'autres fonctionnalités :

- Des points d'arrêt sur un élément d'un DOM : toute modification de l'élément arrêtera le script qui en est à l'origine ;
- Des points d'arrêt sur des events et listeners ;
- Des points d'arrêt sur des requêtes HTTP (XHR breakpoint) ;
- La possibilité de décomposer le CSS d'un élément du dom ;
- La possibilité de simuler l'aspect du site Web sur un autre device, version du navigateur, etc. ;
- L'état de la mémoire, des caches, etc. ;

- Des outils de profiling ;
- Un aperçu de la timeline de chargement et des trames réseau échangées.

ET À PART LE DEBUGGER, QUELS SONT LES AUTRES OUTILS ?

Logs : comment en tirer le meilleur

Les logs sont le plus ancien et le plus universel des outils de debugger. Leur utilité est quadruple. En vous donnant des informations sur leur contexte, ils vous permettent de constater les erreurs, de comprendre leurs causes et enfin de constater la disparition de l'erreur, après correction. Et ce, sans que l'utilisateur final ne s'en rende compte. Pensez à spécifier un niveau de criticité approprié à chaque log. Cela vous permettra d'isoler rapidement les comportements anormaux des simples traces informatives. Vous pouvez facilement augmenter l'utilité de vos logs en apprenant à les requêter efficacement. Des outils d'analyse de logs peuvent également vous faciliter la tâche. Exemples : Logstash, Kibana

Logs des flux d'entrées/sorties

Lister les entrées-sorties d'un programme vous aidera à comprendre les données qui transitent dans votre système et à isoler les causes d'un bug. À noter que l'AOP permet de mettre rapidement en place des logs des appels entrants et sortants. Des outils d'analyse-réseau tels que Wireshark et Fiddler peuvent vous aider à obtenir le même résultat, en local.

Serveur de mock & proxy

Un serveur de mock permet de simuler facilement un serveur distant à un instant donné, et retourner exactement les réponses HTTP souhaitées pour simuler un scénario.

Exemple : Wiremock (utilisable à la fois en serveur indépendant ou en lib de mock durant les tests d'intégration).

À noter que certains de ces serveurs ont une fonctionnalité "d'aspiration". Le serveur fait office de proxy entre le véritable serveur et le programme pendant l'étape de configuration, puis se transforme en véritable serveur de mock, renvoyant pour une requête donnée, la même réponse que celle qu'il a enregistrée précédemment.

Exemples : api-mocking-proxy, MockServer

Citons un exemple : une anomalie liée aux réponses d'un serveur distant ne peut être reproduite qu'une fois par jour. Nous mettons en place un MockServer en tant que proxy entre notre application et le serveur distant, et nous reproduisons l'erreur pour enregistrer les réponses renvoyées. Ensuite, nous n'aurons qu'à passer le MockServer en mode "mock" afin de reproduire le contexte de l'anomalie à volonté.

Thread Dump, Memory Dump et Profiler

Un dump permet de récupérer une copie de l'état d'un programme à un instant donné dans le but de l'analyser plus tard via un logiciel dédié. Ils sont généralement utilisés pour analyser les problèmes de performance et les fuites mémoire. La plupart des machines virtuelles d'exécution sont livrées avec des outils de dump, qui leur sont propres.

Exemples : jmap, Managed Stack Explorer

Citons un exemple. L'application connaît d'importants ralentissements tous les jours à 17h depuis la dernière mise en prod. Un memory dump nous informe que la majorité de la mémoire est occupée par 360 000 instances de la classe "TeaTimeSingleton".

Class	Instance Count	Total Size
class [Ljava.lang.Object;	595	48516
class java.lang.String	361997	43464
class fr.programmez.service.TeaTimeSingleton	360000	2880000
class IT	1057	207560

Les outils de profilage ("profiler") permettent une analyse dynamique d'un programme en cours d'exécution, aussi bien des threads que de la mémoire. Les plus basiques proposent juste des graphiques montrant l'évolution des principales métriques du système, les plus avancées permettent d'avoir un memory-dump actualisé en temps réel.

Exemple : Jvmti, dotTrace

A noter que certains environnements permettent d'effectuer un snapshot de vos programmes, puis d'en refaire une nouvelle instance qui reprend l'exécution là où elle s'était arrêtée.

Exemple : Compute Engine de Google

Modification du code à chaud

Modifier votre code et le déployer sans avoir à redémarrer votre machine d'exécution ni perdre vos données s'appelle le hot deployment. Cela vous permet de tester vos correctifs sans avoir à redémarrer votre machine d'exécution, et ainsi de gagner du temps.

C'est un domaine avec trop de variations et spécificités suivant les langages et framework pour qu'il soit abordé en détail dans cet article.

Exemples : JRebel, WatchJS.

Les tests automatisés

Comme vu précédemment, les tests automatisés sont l'outil idéal pour reproduire un bug à moindre frais. N'oubliez pas non plus les mocks, qui vous permettront de simuler des pans entiers de votre système, tout en vous concentrant uniquement sur les parties buggées.

Quelques bibliothèques et outils supplémentaires permettront de réaliser facilement des tests reproduisant vos cas d'erreurs :

- En embarquant une base de donnée avec vos tests : H2 ;
- En mockant dynamiquement tout un serveur : wiremock.

Analyseur de code statique

Une analyse statique du code permet d'éviter efficacement les erreurs les plus courantes, dans une démarche "préventive". Ils peuvent être configurés pour cibler une problématique en particulier (par exemple : la concurrence.). Ces outils sont disponibles aussi bien sur serveur distant que sous forme de plugin IDE, ces derniers ayant l'avantage de vérifier la qualité du code pendant la saisie.

Exemple : Sonar, Findbug

Remote debug

Il est possible de lancer tout programme sur une machine à distance ou un container, et de le debugger depuis votre IDE.

Cela est particulièrement utile dans le cas d'un bug impossible à reproduire sur la machine de dev, mais reproductible en intégration ou recette. (Attention au risque d'impact sur les autres utilisateurs).

Cela est particulièrement utile lorsque vous développez un programme pour un périphérique ou une machine dédiée qui ne peut héberger la plateforme de développement. Malheureusement, dans la pratique, cet outil est lourd à mettre en place. Vous aurez probablement besoin du Sysadmin pour ouvrir des ports et relancer la machine d'exécution.

Par exemple : un projet Java est lancé en mode debug sur une plateforme distante :

```
java -jar ./build/libs/gs-spring-boot-0.1.0.jar -agentlib:jdwp=transport=dt_socket,
server=y,suspend=n,address=5005 -Xdebug
```

L'IDE lance un process "Remote" afin de pouvoir debugger le processus.

... Ou debugger en lignes de commandes.

Si jamais les restrictions de la machine distante sont trop importantes, il est également possible de debugger en ligne de commande sur la plupart des machines d'exécution.

Exemples : gdb(C, .NET) , mais aussi jdb (java), ndb (nodeJs)

Et le reste ? Au sujet de vos outils, frameworks et autres technologies

Tous les frameworks, toutes les technologies, tous les outils que vous utilisez sont fournis avec des outils et fonctionnalités de debug. Sans exception. Citons quelques-unes de ces fonctionnalités :

- Vous aurez toujours un niveau de logs avancé (et désactivé par défaut). Que ce soit pour votre compilateur, votre IDE, le framework que vous utilisez, etc. ;
- De nombreux outils tournant sur une machine virtuelle d'exécution ont un mode ou une version alternative compatible avec le debugger de votre IDE (mvndebug, gulp-debug) ;
- Souvent les outils d'"Infra As Code" ont un mode "plan" qui détaillera le plan d'exécution prévu par le script, sans pour autant l'exécuter (Puppet, Terraform, Ansible, etc.) ;
- Les ORM et frameworks d'injection de dépendance récents ont un code suffisamment lisible pour qu'il soit intéressant de rentrer dans leur code via le débogueur afin de comprendre certains comportements et anomalies, et d'observer le contenu de leurs données internes (Spring, Hibernate, etc.) ;
- Tous les SGBD ont une vue des requêtes en cours d'exécution, et une vue des requêtes les plus lentes. Toutes ont une commande "explain", qui déroulera le plan d'exécution d'une requête.

```
explain select * from users u
left join employees e
on e.user_id = u.user_id
```

QUERY PLAN

```
Merge Left Join (cost=149.93..625.13 rows=25680 width=12)
Merge Cond: (u.user_id = e.user_id)
-> Index Only Scan using userid on users u (cost=0.15..84.16 rows=2400 width=4)
-> Sort (cost=149.78..155.13 rows=2140 width=8)
Sort Key: e.user_id
-> Seq Scan on employees e (cost=0.00..31.40 rows=2140 width=8)
```

Exemple : postgresql

QUAND LE DÉVELOPPEUR PASSE EN MODE DEBUG

On dit parfois qu'il faut concevoir son logiciel comme un architecte, coder comme un ingénieur et tester comme un barbare. Nous ajoutons "debugger comme un détective." Debugger implique un état d'esprit particulier, différent de celui de produire du code :

- N'ayez pas d'idées préconçues. Tant que vous n'avez pas trouvé l'origine du bug, toutes les parties du programme et de l'environnement sont "présumées coupables" ;
- Restez rigoureux et méthodique. Pour arriver à trouver l'instruction fautive, aucun élément ne doit vous échapper ;
- Connaissiez vos outils : vous ne devez pas passer à côté d'une information vitale, juste parce que vous ne savez pas bien vous servir d'un outil ;
- Écoutez vos intuitions, mais vous devez toujours les vérifier. Une intuition validée est un fait. Vos conclusions doivent se fonder sur des faits, pas sur des intuitions ;
- Ne vous attendez pas toujours à une explication complexe. Même une faute de frappe peut causer un bug ;
- Ne modifiez pas votre code pendant la phase d'analyse du bug. Vous risquez d'introduire de nouvelles erreurs ou de modifier le bug initial par inadvertance ;
- Même si vous devez être rigoureux dans votre recherche, essayez de prioriser les sources potentielles de l'erreur. Vous ne pourrez jamais relire et analyser minutieusement chaque ligne du programme. Utilisez votre expérience pour gagner du temps et vérifiez les méthodes que vous suspectez le plus en premier. Mais ne négligez aucune piste pour autant ;
- Vous ne serez efficace que si vous gardez l'esprit clair et alerte. Debugger un code demande plus d'énergie que de l'écrire. Prenez des pauses, mangez, assurez-vous d'avoir suffisamment de sommeil, alternez avec d'autres tâches. Assurez-vous d'avoir un environnement de travail sain et de ne pas être interrompu pendant vos sessions de debugger. Si vous ne trouvez pas la cause du problème, réessayez le lendemain. Vous aurez un oeil neuf sur votre code ;
- N'en faites pas une affaire personnelle. Le programme ne tente pas de vous piéger, il traite juste les informations qu'il a de la manière dont cela lui a été demandé ;
- Même si, pris individuellement, chaque bug aurait pu être évité, la présence globale de bugs au sein du code que vous produisez est inévitable. Voyez vos bugs comme une opportunité d'apprendre, pas comme un échec personnel.

Un dernier conseil : la principale difficulté dont souffre le debug est qu'il n'est pas considéré comme un véritable domaine d'expertise. En conséquence, les développeurs échangent peu sur le sujet, la passation de compétence reste limitée, la littérature est assez pauvre, et bien souvent les pratiques tombent dans l'oubli. C'est une erreur ! Vous devez partager ce que vous savez ! Toujours ! Faites du "pair-debugging" ! Transmettez vos pratiques lors des passations. Écrivez une vraie "documentation du debug" pour votre projet, qui listera les requêtes les plus utilisées et les outils disponibles, plateforme par plateforme. Le debug est un domaine d'expertise à part entière, une discipline universelle, un art. À vous de le conquérir.

Sources :

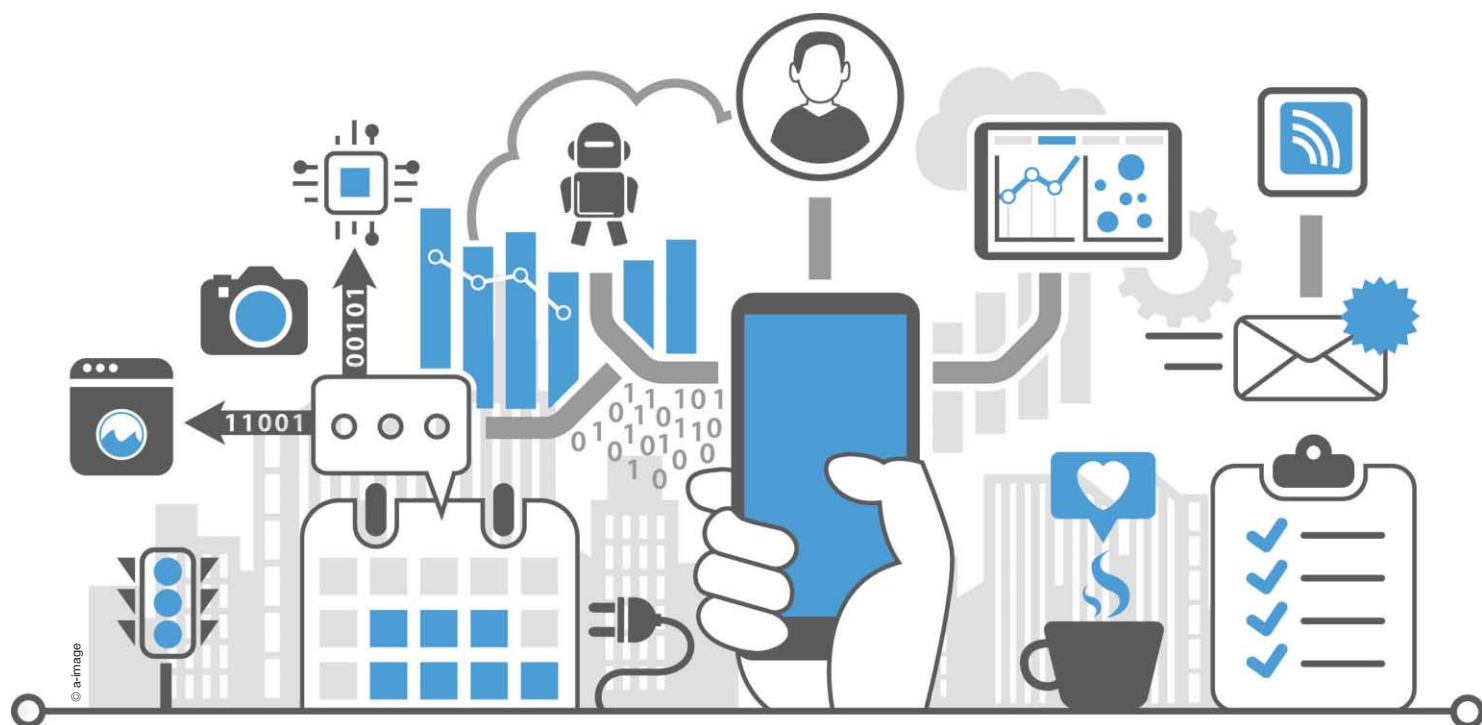
Le debugger front : <http://devtoolsecrets.com/>

L'importance de la remontée de l'anomalie :

<http://blog.xebia.fr/2015/11/23/qualifier-une-anomalie-pour-faciliter-sa-correction/>

Saff squeeze : <https://www.infoq.com/news/2008/11/beck-saff-squeeze>

Cloud Computing + DevOps



Le Cloud Computing et le DevOps n'ont plus de secrets pour vous, ou presque. Le Cloud est une entité mouvante qui évolue constamment, et, désormais, impossible d'échapper à la matrice.

Le Cloud, quelle que soit sa forme est présent partout :

- Logiciels ;
- Site / app web ;
- Stockage ;
- App mobile.

De très nombreuses applications utilisent des services Cloud même si l'utilisateur ne le sait pas. Dans les environnements de développements et de tests, le Cloud est devenu incontournable pour déployer des environnements de tests, faire du build, etc. Pour les entreprises, il s'agit de ne plus supporter en interne toute une partie de leurs infrastructures. Pour une nouvelle entreprise, une startup, les services Cloud seront bien utiles. La dématérialisation de l'informatique a réellement débuté avec les App Store mobiles qui ont bouleversé notre manière de consommer les

logiciels. Aujourd'hui, quoi de plus facile que d'installer une application sur son smartphone ou son desktop ? Des pures players comme Salesforce ont bouleversé la manière de faire de l'ERP et du CRM. Amazon, Google ont imposé un modèle Cloud que les concurrents ont mis en place. Aujourd'hui, Microsoft, IBM, Oracle, SAP, ont fait du Cloud un enjeu essentiel.

Pourquoi encore installer et déployer une suite bureautique classique quand on peut facilement déployer et gérer des milliers d'utilisateurs avec Google Docs ou Office 365. Tout est lié et interconnecté. Les nouvelles architectures s'appuient massivement sur le Cloud. Et fini le temps où le Cloud était de l'infrastructure, le fameux IaaS où on faisait de la machine virtuelle. Les conteneurs, le PaaS (plateforme) et la fusion entre le PaaS et l'IaaS, transforment notre paysage technique et le déploiement.

Dans ce dossier, nous avons volontairement additionné Cloud et DevOps car finalement, les deux travaillent très bien ensemble et nous verrons pourquoi.

La rédaction.

Les impacts du Cloud sur l'hébergement d'une application

• Fabien Didot

Cloud & Datacenter MVP et Azure P-Seller. L'officier en tant qu'architecte solutions Azure chez Orange Application for Business où j'œuvre pour construire les meilleurs hébergements pour les applications de mes clients.

Le métier d'hébergeur d'applications évolue. En effet, l'avènement des solutions de PaaS sur des Clouds publics offre maintenant des avantages que l'hébergement traditionnel ne permet pas car la dette technique est souvent trop lourde à compenser. Les architectures microservices également apportent leur lot de nouveautés.

Le rôle d'un hébergeur est de garantir la disponibilité des applications, chose faisable relativement simplement lorsque nous maîtrisons toutes les briques (réseau, stockage, hyperviseur, etc.). L'utilisation d'un Cloud public pour garantir le même niveau de satisfaction client n'est pas chose aisée car chaque fournisseur dispose de ses propres règles d'architecture permettant de garantir les SLA maximaux. Ces SLA sont d'ailleurs souvent suivis d'étoile pour nous informer que même s'ils ne sont pas respectés, le client n'aura souvent que ses yeux pour pleurer. Donc en tant qu'hébergeur notre mission est d'utiliser les mécanismes similaires de haute disponibilité que nous avons toujours mis en place dans nos datacenters (et que nous continuons à mettre en place) afin de garantir ces SLA. Nous parlons ici de PCA (Plan de Continuité d'Activité) et de PRA (Plan de Reprise d'Activité) qui, associés au respect des patterns d'architecture des Clouds publics permettent de faire du fog computing (utilisation

de plusieurs fournisseurs de Cloud public pour la création d'une plateforme) tout en garantissant des SLA très élevés à des coûts défiant toute concurrence.

La création de la plateforme est de maintenir les SLA, c'est la partie simple dans ce changement de paradigme. Il faut aussi accompagner les employés avec ces nouveautés. Le taux de pénétration du Cloud et de ses concepts est au moins aussi important que celui de ses fausses croyances, ce qui accroît le sentiment de certaines personnes à se sentir mises de côté alors que cette évolution dans leur travail n'en sera que bénéfique. En effet la partie récurrente est scriptée/automatisée ; de ce fait, il reste beaucoup plus de temps pour la mise en œuvre de projets à haute valeur ajoutée. C'est du gagnant-gagnant. Pour cela il est important, non seulement de mettre en œuvre un plan de formation, ce qui semble être une évidence, mais qui n'est pas toujours réalisé, mais il faut aussi y rajouter l'intégration des équipes dans les réflexions et le choix des solutions à mettre en

œuvre notamment sur la partie reporting et monitoring, qui sont les éléments majeurs garantissant leur travail et la qualité de la plateforme.

C'est le point primordial pour les équipes opérationnelles. Deux solutions s'offrent à nous avec chacune leurs avantages :

- Continuer à mettre en œuvre le monitoring avec les solutions actuelles. Cela implique souvent de mettre des collecteurs qui vont centraliser les données sur les différents environnements clients et à moindre frais de pouvoir fonctionner à périmètre équivalent, ou alors, d'étendre son SI par le biais de liaisons MPLS et/ou VPN. Le problème majeur de ce choix est que nous ne pourrions pas profiter des scénarios de monitoring poussés que ce soit de la plateforme ou applicatif par les mécanismes de la plateforme.
- Utiliser une solution externe en SaaS pour ne pas avoir une nouvelle plateforme de monitoring à passer en CAPEX. Les solutions sur le marché sont nombreuses, plus hétérogènes quant aux plateformes supportées, cependant il peut parfois être délicat d'intégrer leur modèle de licensing à un Business Model existant.

Quoi qu'il en soit, le changement sur l'environnement de travail n'est pas anodin. L'idée n'est pas de calquer le modèle actuel sur le Cloud, mais bien de faire évoluer son SI vers la modernité en remettant le client au centre des intérêts, en proposant l'accès à des dashboards en temps réel, faisant le lien entre tickets ouverts et SLA en utilisant le Big Data pour agréger tous les éléments.

Attention aux contrats

La dernière chose et non des moindres à prendre en compte c'est l'aspect contractuel. Si je prends l'exemple d'Azure, plusieurs types de contrats existent, chacun avec ses avantages et inconvénients.



Cloud + DevOps : pourquoi ?



François Tonic

Revenons rapidement sur la notion de DevOps. Comme vous le savez, DevOps n'est pas un outil, ni une méthode mais une culture, une philosophie. En soi, DevOps n'est pas une nouveauté et reprend les bases de l'agilité et des outils d'intégration.

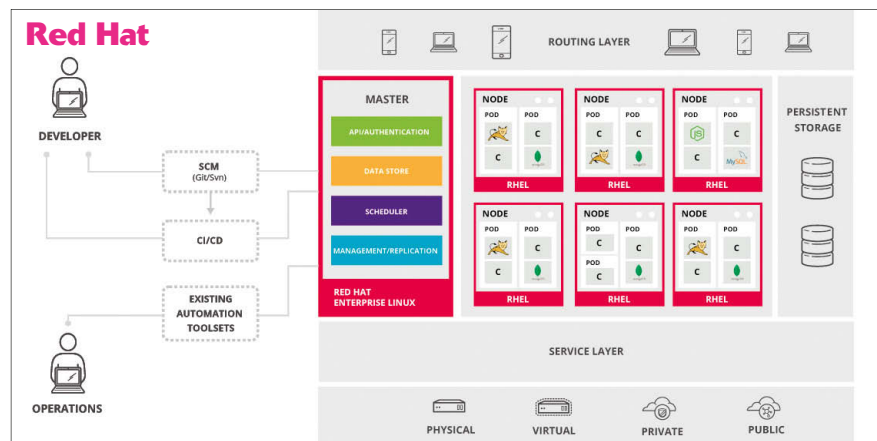
Quand nous regardons ce que recouvre cette philosophie, nous y trouvons : les méthodes agiles, les cycles itératifs, les outils que nous avions dans l'approche ALM et tout ce qui est Continuous (intégration, génération, livraison, déploiement), avec les notions de tests et de sécurité. DevOps redonne un "cadre" plus global à ceci avec pour objectif de faire communiquer et travailler ensemble les développeurs (Dev) et les opérationnels (Ops), deux mondes qui ont parfois du mal à s'entendre. A cela, il faut rajouter les utilisateurs, les métiers, les testeurs. Il ne faut jamais oublier que DevOps ne se limite pas aux Devs et aux Ops mais concerne l'ensemble des équipes d'une entreprise. Cette réalité a parfois du mal à être comprise.

DevOps s'appuie effectivement sur plusieurs fondations incontournables :

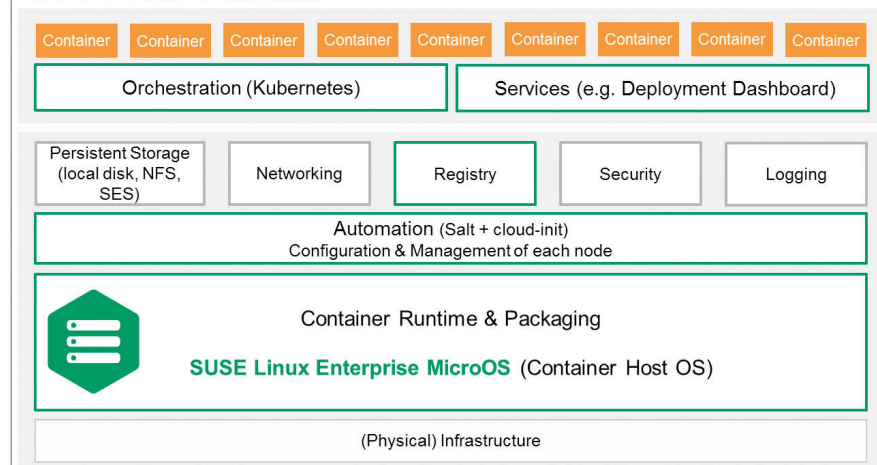
- L'agilité et les cycles itératifs : typiquement la méthode Scrum avec les notions d'itération / de sprint. Mais toute méthode agile peut convenir ;
- La notion de Continuous : c'était une des bases essentielles de l'agilité, de l'ALM et de la Software Factory ;
- Des outils communs, un langage commun, la communication entre les équipes.

Une philosophie simple mais une réalité complexe

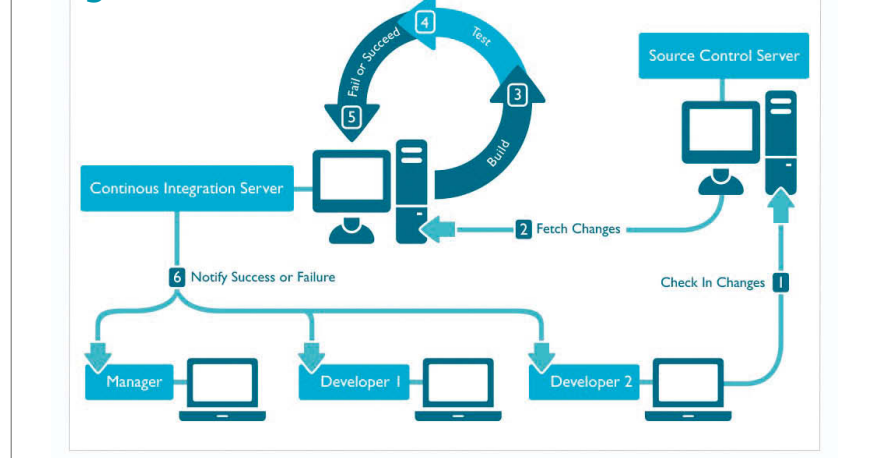
On dit souvent que la simplicité est complexe. Nous pouvons dire la même chose de DevOps. L'idée générale est de casser les fameux silos où chaque équipe travaille dans son coin sans se soucier des autres. Mais, avec la mobilité, le Web, l'exigence de proposer plus rapidement des nouveaux services, des mises à jour, oblige à repenser la manière de développer et de déployer. La dématérialisation de l'informatique et particulièrement du logiciel a profondément changé notre manière de gérer les projets et de consommer l'informatique. En effet, les App Store mobiles ont créé une nouvelle manière de consommer l'application



SUSE CaaS Platform



Intégration continue



qui est désormais jetable. Le cycle produit traditionnel avec une version majeure tous les 12 – 18 mois, c'est fini, tout comme le support physique. Les utilisateurs sont pressés et ne veulent pas attendre.

Les pures players du Web ont compris depuis leur création qu'il fallait repenser le développement des services Web. Il faut mettre à jour et ajouter des codes et de nouvelles fonctions très régulièrement.

Le cycle ne s'arrête pas. Pour arriver à cela, il faut être totalement agile, flexible et avoir un développement continu.

Mais il faut aussi rapprocher le monde du dev et le monde des ops : le développeur doit comprendre comment on déploie, comment on met en production un service, un nouveau code. L'automatisation a permis cette fluidité :

- L'intégration continue intègre les différents codes, effectue les tests nécessaires et génère les binaires nécessaires ;
- La livraison continue et le déploiement continu permettent d'automatiser les environnements de production et tout le processus de génération et de déploiement.

Le DevOps préconise cette automatisation et ces cycles continus.

Le Cloud facilite aussi cette automatisation et toutes les processus de Continuous Delivery. En effet, il est facile de provisionner de l'intégration continue ou même un moteur de build directement en services Cloud. Ainsi, vous pouvez déployer en quelques minutes un Jenkins entier sur le Cloud.

Vous avez besoin d'un référentiel de code ? GitHub. Des outils de tests ? Là encore, vous aurez l'embarras du choix ! Des environnements de déploiement ? Le Cloud sera là pour vous aider et vous simplifier le travail.

Le Cloud rentre totalement dans la culture DevOps et de l'agilité. Le but est d'être le plus souple possible, de provisionner et déprovisionner en quelques minutes.

Le Cloud a la capacité de s'adapter à vos besoins, selon les contraintes de chaque projet, de chaque application. Pareillement, il ne faut pas appliquer bêtement le DevOps. Adaptez-le à vos besoins réels, ne l'imposez pas. C'est la meilleure solution pour échouer. Déjà avec Scrum, ou toute autre méthode agile, il ne fallait pas l'appliquer bêtement mais utiliser ce qui était utile à chaque équipe, à chaque projet et l'adapter selon le contexte.

LINUX EST BIEN LÀ

L'open source et Linux sont bien entendus des acteurs majeurs du Cloud même si côté Linux ce n'est pas toujours très visible. Mais le système demeure leader chez les hébergeurs et le marché serveur.

Côté Red Hat

Red Hat est le poids lourd du monde Linux en entreprise. Et sur la partie Cloud, l'éditeur travaille depuis plusieurs années à sa solution OpenShift. La plateforme a su se développer et proposer une bonne richesse fonctionnelle. Bien entendu, l'objectif est de s'appuyer sur les standards et outils Open Source du marché. Il s'agit d'un PaaS utilisant un socle orienté conteneur (Kubernetes et Docker). OpenShift s'adresse aussi bien aux entreprises qu'aux équipes devs et ops. OpenShift se compose de 3 grandes éditions :

- Openshift online : la partie publique ;
- Openshift dedicated : la partie privée ;
- Openshift container platform : idéale pour les datacenters et pour contruire un PaaS.

Bien entendu, le DevOps n'est pas oublié avec tous les outils d'automatisation, le Continuous. Sur la partie

Infrastructure as Code, pour Red Hat, cela concerne aussi bien les devs que les ops et l'outil central est Ansible. Pour les équipes Red Hat, Ansible ne présente pas de réels soucis de montée en charge même avec des milliers de machines / conteneurs à gérer. Côté Terraform, ce n'est pas la priorité pour Red Hat.

Côté langage, OpenShift supporte : PHP, Perl, Python, Ruby, Java et de multiples instances d'outils et de bases de données. Côté .Net, la plateforme supporte ASP.Net et .Net Core. On peut utiliser .Net Core 1.0 et 1.1.

Une édition ARM est actuellement en préversion. Elle supporte les puces A8-A 64-bit et inclura KVM, stockage Ceph et les outils dédiés aux développeurs.

Côté Suse

Depuis 2003, Suse a connu plusieurs évolutions majeures : rachat par Novell, puis rachat de Novell par Attachmate et séparation de SuSe et de Novell puis rachat d'Attachmate par Micro Focus. Même si Suse était déjà présent dans la partie Cloud et DevOps, l'éditeur restait discret, surtout auprès des développeurs. Le rachat

des activités OpenStack et CloudFoundry de HPE (ex-HP), pour plusieurs milliards remplace Suse dans le marché Cloud et va permettre de repositionner Suse sur le PaaS et donc auprès des développeurs. Si la question de voir une distribution CloudFoundry sous le logo Suse n'est qu'une question de temps, l'éditeur devra dévoiler ses ambitions, une roadmap et les outils.

Suse avoue elle-même qu'elle est toujours très orientée ops et entreprises et peu développeurs. Mais l'éditeur veut s'investir auprès d'eux pour adresser les deux mondes. Il faut bâtir une offre attractive et des services dédiés aux développeurs. Et Suse sait qu'un énorme travail est à accomplir pour être plus visible.

Suse a fait le choix de Kubernetes et de Docker, du projet Magnum d'OpenStack. Pour la partie infra as code, l'éditeur mise actuellement sur l'outil Salt. Les outils tels que Ansible et Terraform ne sont pas supportés officiellement. L'éditeur parle de plateforme CaaS, Container as a Service et mise beaucoup sur cette approche, comme les autres fournisseurs technologiques.

Vers l'Infrastructure as Code

Cette "fusion" est une réalité. Et le Cloud fait partie de la vie du développeur, qu'il le veuille ou non. Dans le développement, les services backend mobiles sont pléthores. En entreprise, on peut dématérialiser tout ou partie de son infrastructure en Cloud public ou monter des environnements d'analyses Big Data ou des outils de développement pour x heures.

L'Infrastructure as Code est une approche par le code de l'infrastructure à provisionner, à déployer pour exécuter les applications et les piles techniques. Cette infrastructure peut être écrite par le dev mais aussi par l'ops. Bien entendu, il faut avoir des notions côté applicatif et côté production. Et le Dev + Ops prend tout son sens car pour réussir son code d'infrastructure, il faut connaître les deux mondes, comme nous le verrons dans l'article dédié à ce sujet. •

Professionaliser et fiabiliser ses déploiements avec VSTS et Azure



Sébastien Boinot
DevOps & Agile Coach
chez **Avanade**
@BoinotSebastien

Mettre en place un Release Pipeline qui intègre et déploie en continu est un premier pas vers l'automatisation et l'approche DevOps. Mais ce n'est pas suffisant. Vous devez professionnaliser et fiabiliser les livraisons et les déploiements afin de vous assurer que ce que vous livrez en production est d'un haut niveau de qualité.

L'approche DevOps passe par la mise en place d'outils d'automatisation afin d'industrialiser les livraisons et être ainsi capable d'intégrer, livrer et déployer en continu. L'outil Visual Studio Team Services (VSTS) de Microsoft, qui gère le cycle complet d'une application, du développement jusqu'à la mise en production, facilite l'implémentation d'une usine logicielle, en pratiquant certains concepts fondamentaux de la philosophie DevOps :

- **Continuous Integration (CI)**, la pratique qui consiste à vérifier à chaque changement du code source fait par le développeur, grâce à une build qui compile et exécute les tests unitaires automatiquement, que le résultat ne produit pas de régression dans l'application développée ;
- **Continuous Delivery**, la pratique qui consiste à rendre chaque changement « releasable » et « déployable » à tout moment en production ;
- **Continuous Deployment (CD)**, le principe de déployer automatiquement en production chaque changement qui passe avec succès les tests automatisés.

La création d'une Release Pipeline dans VSTS avec les modules Build et Release s'est simplifiée ces dernières années et permet aujourd'hui de créer un environnement complet d'intégration et de déploiement continu dans Azure en seulement quelques minutes. [1]

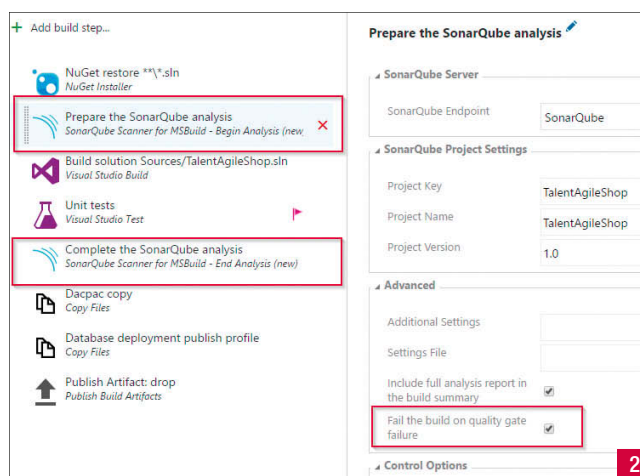
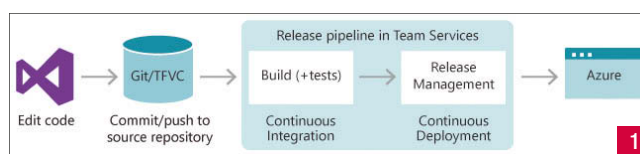
Mais implémenter uniquement cette automatisation n'est pas suffisant. Avoir à disposition un système qui déploie automatiquement en production, sans intervention humaine, va permettre d'augmenter la productivité et de réduire le temps de mise à disposition en production mais peut engendrer d'énormes dégâts en cas de livraison d'un code bogué. Vous devez fiabiliser et maîtriser vos déploiements en mettant en place des contrôles et des indicateurs de qualité afin de valider chaque étape de votre Release pipeline. Vous devez stopper la promotion d'une étape vers la suivante si elle ne répond pas aux exigences attendues.

Nous allons voir ensemble comment procéder avec VSTS et le PaaS d'Azure.

Mesurer et contrôler sa dette technique

L'intégration continue est implémentée dans VSTS par une build qui se déclenche à chaque archivage d'un développeur. La build compile entièrement les sources avec la dernière modification archivée et exécute automatiquement les tests unitaires. Ce système vérifie en continuation l'intégration des changements mais ne permet pas de savoir si le code source est bien écrit. Nous devons prendre en compte une composante souvent oubliée : la dette technique. Elle va nous garantir que le code est propre, maintenable, évolutif dans le temps et que la situation ne se dégrade pas.

Les équipes de SonarQube, l'outil incontournable de mesure de dette technique, ont développé une extension sous la forme de tâches de build



qui permet d'analyser le code source pendant la build.

Il suffit d'ajouter dans la définition de build les 2 tâches « SonarQube Scanner for MSBuild - Begin Analysis (new) » et « SonarQube Scanner for MSBuild - End Analysis (new) », la première avant la compilation de la solution, la deuxième après l'exécution des tests unitaires, de les paramétrer vers votre serveur SonarQube et d'activer l'option « Fail the build on quality gate failure » afin de casser la build si le changement de code ne correspond pas aux normes et aux règles de qualité définies dans votre serveur SonarQube. [2]

Désormais à chaque déclenchement de la build d'intégration continue (build CI), votre code est analysé et rejeté s'il ne correspond pas aux règles de qualité exigées.

Versionner ses assemblies

La mise en place d'une build nous apporte un versioning automatique. A chaque nouvelle build un numéro de build est généré et une étiquette, reprenant ce numéro de build, est appliquée sur le code source afin de les associer et de garder une traçabilité.

En appliquant le système « gestion sémantique de version » défini par sa convention MAJEUR.MINEUR.CORRECTIF, MAJEUR quand il y a des changements non rétro compatibles, MINEUR pour ceux rétro compatibles et CORRECTIF pour des corrections d'anomalies rétro compatibles, vos numéros de version, et la façon dont ils changent, donneront du sens au code sous-jacent et à ce qui a été modifié d'une version à l'autre.

Pour conserver une traçabilité de bout en bout, code source -> build -> assemblies, nous devons aussi versionner les assemblies au moment de la build. Ne vous êtes-vous pas déjà retrouvé dans la situation où la version d'une assembly déployée sur le serveur de production est 1.0.0.0 ? Et de ce fait être dans l'impossibilité de remonter jusqu'à la version du code source utilisée ?

Ce versioning peut se faire à l'aide d'un script PowerShell :

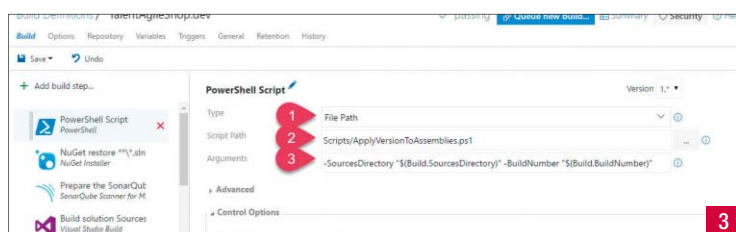
```
Param(
    [string]$SourcesDirectory,
    [string]$BuildNumber
)

# Make sure path to source code directory is available
if (-not $SourcesDirectory)
{
    Write-Error ("SourcesDirectory variable is missing.")
    exit 1
}
elseif (-not (Test-Path $SourcesDirectory))
{
    Write-Error "SourcesDirectory does not exist: $SourcesDirectory"
    exit 1
}
Write-Host "SourcesDirectory: $SourcesDirectory"

# Make sure there is a build number
if (-not $BuildNumber)
{
    Write-Error ("BuildNumber variable is missing.")
    exit 1
}
Write-Host "BuildNumber: $BuildNumber"

# Regular expression pattern to find the version in the build number
# and then apply it to the assemblies
$VersionRegex = "\d+\.\d+\.\d+\.\d+"

# Get and validate the version data
$VersionData = [regex]::matches($BuildNumber,$VersionRegex)
switch($VersionData.Count)
{
    0
    {
        Write-Error "Could not find version number data in BuildNumber."
        exit 1
    }
    1 {}
    default
```



```
{
    Write-Warning "Found more than instance of version data in BuildNumber."
    Write-Warning "Will assume first instance is version."
}
}

$NewVersion = $VersionData[0]
Write-Host "Version: $NewVersion"

# Apply the version to the assembly property files
$files = Get-ChildItem -Path $SourcesDirectory -Recurse -include AssemblyInfo.*

Write-Host "$($files.Count) files to update with the new version '$NewVersion'"

foreach ($file in $files)
{
    Write-Host "Processing $file"
    $filecontent = Get-Content($file)
    $filecontent -replace $VersionRegex $NewVersion | Out-File $file
}
}
```

On ajoute ensuite ce script en tant que première étape de la définition de build CI à l'aide d'une tâche « PowerShell Script » prenant en arguments les variables systèmes de build : « Build.SourcesDirectory » et « Build.BuildNumber ». [3]

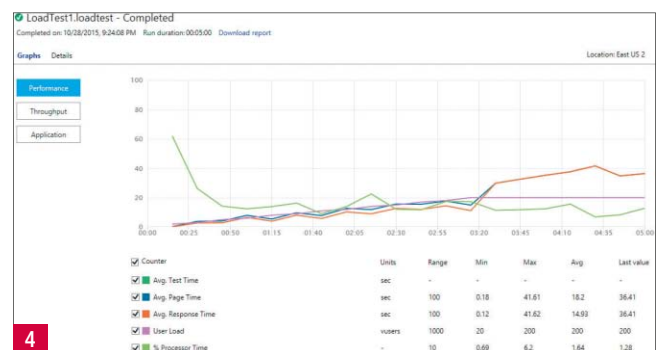
Tests de charge et de performances

Vous devez vous assurer que votre application supportera bien la charge d'utilisateurs attendue. Les tests de charge et de performances doivent être effectués sur un environnement isolé de la production pour ne pas impacter les utilisateurs en cours. L'environnement doit être dimensionné au plus proche de celui de production, avec le même volume de données, si possible avec une copie de la base de données de production en ayant au préalable anonymisé les données sensibles.

Avec VSTS et Azure, la mise en place et la gestion d'une infrastructure on-premise afin de simuler les utilisateurs n'est plus nécessaire. Elle est provisionnée à la volée et dimensionnée automatiquement dans le Cloud avant chaque lancement de tests de charge et de performances. Les rapports d'exécution et l'évolution des tests sont consultables depuis VSTS dans le module Test. [4]

Blue-green Deployments avec AZURE WEB APPS SLOTS

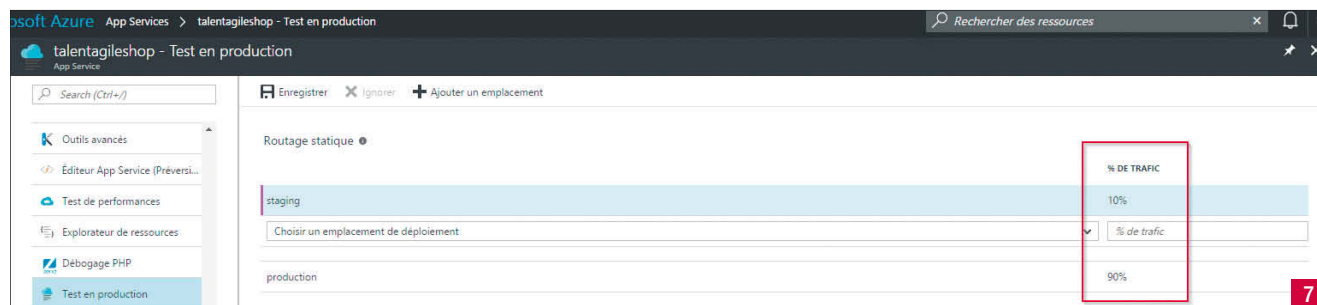
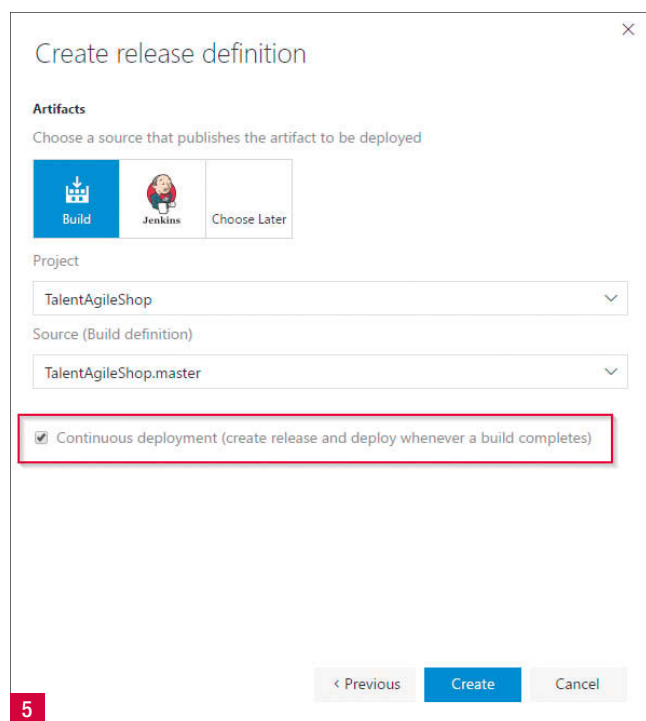
Concevoir un déploiement continu vers une application ou site Web hébergé dans le PaaS (Platform as a service) Azure Web Service, est devenu abordable avec VSTS Release Management et les modèles de



déploiements Azure Web Service. Créez votre définition de release en définissant votre release en tant que déploiement continu de manière à déclencher une release à chaque build CI exécutée avec succès. [5]

Le déploiement est désormais en continu jusqu'à la production. Mais comment être sûr que la mise en production s'est bien passée, et que ce que nous venons de livrer est opérationnel ? Et si nous rencontrons des erreurs en production, est-il encore possible de réinstaller la version précédente ? Pour éviter toutes ces incertitudes et réduire le risque lors des mises en production, une bonne pratique peut être mise en place : le « Blue-Green Deployments ». Le « Blue-Green Deployments » est une technique qui s'appuie sur deux lignes de production identiques, la bleue et la verte, dont le principe est de déployer l'application sur une ligne uniquement dans un premier temps. Cela permet de toujours disposer d'un environnement de production accessible pour les utilisateurs le temps du déploiement, du démarrage du site, de la validation par quelques tests d'exploration sur le deuxième et de faire un rollback en cas de problème. Sans aucun impact pour les utilisateurs.

Azure Web Services nous offre la possibilité de mettre en place le Blue-Green Deployment grâce aux environnements intermédiaires appelés « Slots » dans Azure Web Services. Une fois la configuration de l'emplacement de déploiement, nommé « staging » par exemple, créé au niveau de votre Web App de Production [6], il suffit ensuite de le prendre en compte en tant que nouvelle étape de déploiement dans votre Release pipeline définie dans VSTS Release Management.



A/B Testing avec AZURE WEB APPS Traffic Routing

L'A/B Testing et le Blue-Green déploiement, qui sont parfois confondus de par leurs méthodes de déploiement, ont deux utilités très différentes. Tandis que le Blue Green déploiement sécurise les mises en production et supporte de fait un rollback en cas de problème, l'A/B testing permet de tester sur un échantillon de « vrais » utilisateurs la qualité des nouvelles fonctionnalités, les changements d'ergonomie ou d'interface utilisateur, afin de récupérer des retours sur leur utilisation, leur adoption, leur performance. Cette première phase va permettre de valider ces nouveautés et de les publier sur l'ensemble des sites de production dans un second temps. Vous pouvez bien sûr combiner les deux en utilisant le Blue-Green déploiement pour déployer les nouvelles fonctionnalités à tester en A/B Testing.

Traffic Routing qui est une fonctionnalité sur une Web APP va nous permettre d'implémenter l'A/B Testing en paramétrant le pourcentage de redirections vers le slot de Staging et celui de Production [7]. Traffic routing applique une méthode de routage de pondération du trafic, basée sur le pourcentage renseigné, à chaque requête DNS.

La collecte des données d'utilisation pourra ensuite s'effectuer depuis Application Insights.

Monitorer son application avec Application Insights

Le monitoring est un des piliers de l'approche DevOps. Il est important et nécessaire de superviser son application en production afin de détecter au plus vite les éventuelles erreurs et d'être capable d'en analyser la cause grâce aux données récoltées.

Application Performance Monitoring

Azure Application Insights est un APM (Application Performance Monitoring) puissant permettant de superviser son application en termes de performance : temps moyen de réponse des pages, nombres de vues par page, top 5 des pages les plus lentes, etc.

Dans Azure Web Services, Application Insights est nativement intégré. Il

suffit de l'activer lors de la création de la Web APP. Vous pourrez ensuite suivre en temps réel les informations concernant votre site Web. [8] Afin d'augmenter l'efficacité de la supervision de votre application, vous devez renforcer celle-ci par une mise en place d'alertes vous prévenant des erreurs et des seuils de performances dépassés.

Tests de disponibilité et de réactivité

Avec Application Insights vous pouvez aussi vérifier la disponibilité et la réactivité de votre site Web. Le principe est de mettre en place un test qui ping à intervalle régulier l'url de votre site Web. [9]

De la même manière que l'APM, vous devez configurer des alertes afin d'être averti en temps réel et d'être informé avant vos clients.

Haute disponibilité avec AZURE TRAFFIC MANAGER

Dernier point, la qualité de service de votre site Web doit être irréprochable. Délivrer et mettre à disposition une application répondant aux besoins des utilisateurs est bien sûr impératif, mais vous devez aussi proposer un service de qualité et performant. Rien de plus énervant en tant qu'utilisateur de tomber nez à nez sur la page "notre site est actuellement en maintenance" ou bien attendre plusieurs secondes que la page se charge. Vous devez identifier les problèmes avant les utilisateurs. Une des solutions est d'implémenter une topologie distribuée sur plusieurs instances du site Web hébergées dans une ou plusieurs régions. [10] Azure Traffic Manager nous propose ce service et permet de mettre en place un tel dispositif. Traffic Manager applique la méthode de routage du trafic à chaque requête DNS qu'il reçoit. Quatre méthodes de routa-

ge existent : priorité, pondération, performance et géographique. Choisissez la méthode de routage qui correspond à votre besoin, par exemple « Priorité » pour obtenir une réduction des temps d'arrêt des applications, « Pondération » pour faire de l'A/B Testing, « Géographique » pour distribuer le trafic utilisateur entre plusieurs emplacements selon la localisation.

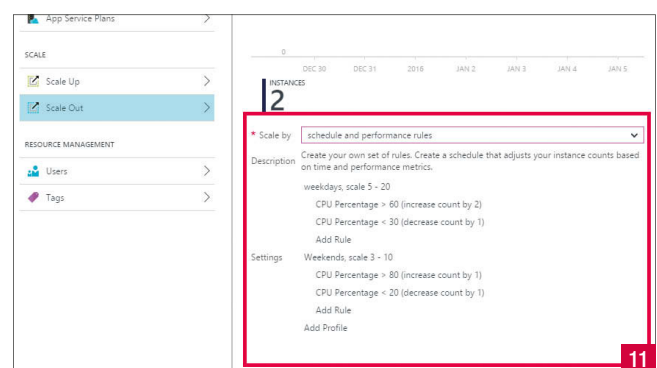
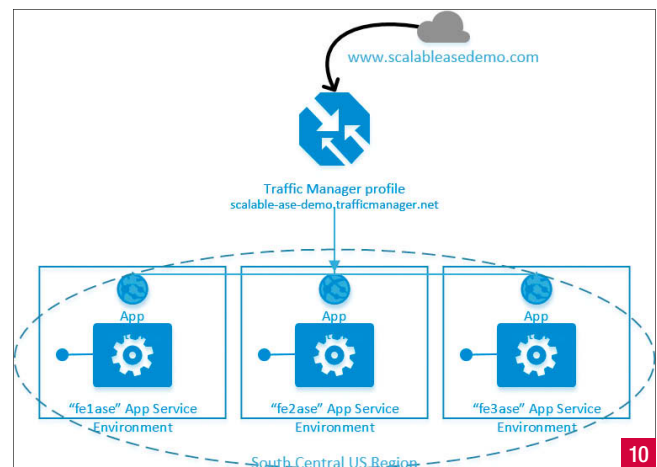
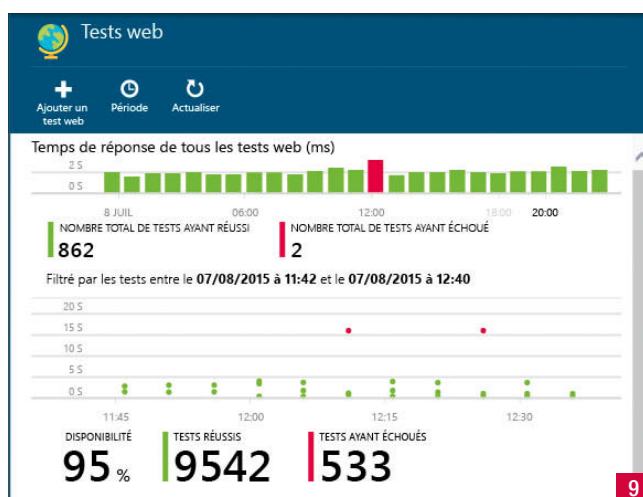
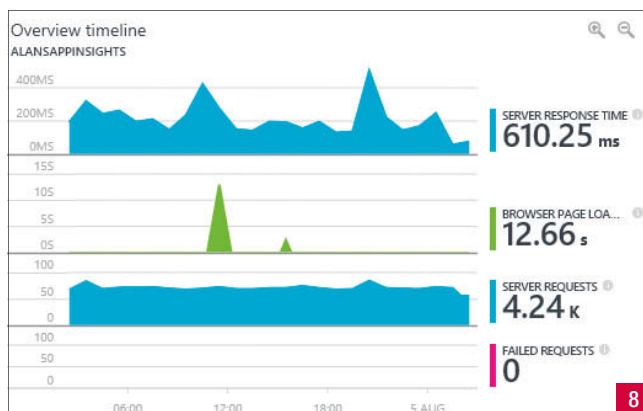
Auto Scaling avec AZURE APPS

Prenons l'exemple concret du pic de fréquentation sur un site e-commerce en période de soldes. Ce pic intervient seulement 3-4 fois dans l'année sur une période de quelques semaines. Monter une infrastructure afin de supporter cette charge temporaire n'est pas envisageable financièrement pour la plupart des entreprises.

Les équipes Microsoft Azure proposent le service d'auto scaling dans Azure App Services. Ce service est capable d'optimiser vos ressources en augmentant ou diminuant automatiquement le nombre d'instances de votre site Web ; soit en s'adaptant en temps réel à la charge d'utilisateurs, selon l'activité du CPU par exemple [11], soit en paramétrant des plages horaires spécifiques.

CONCLUSION

Vous devez automatiser les déploiements afin de fluidifier les mises en production, mais vous devez aussi surveiller étroitement l'exploitation et la qualité de votre application par des métriques et indicateurs clés. Avec le PaaS d'Azure et VSTS, implémenter ces bonnes pratiques et approches réservées habituellement à des experts est maintenant accessible à tous. Ils nous permettent leur mise en place, et nous aident efficacement à fiabiliser et professionnaliser notre release pipeline et nos mises en production.



Service Fabric : une plateforme PaaS sous stéroïdes Partie 3

• Maxime CAROUL
caroul@infinitesquare.com
<https://blogs.infinitesquare.com/users/mcaroul>

• Thibaut RANISE
tranise@infinitesquare.com
<https://blogs.infinitesquare.com/users/tranise>



Derrière ce titre quelque peu marketing se dresse un constat simple, l'arrivée d'une nouvelle génération de solutions Platform as a Service (PaaS) qui offre davantage de performance, de souplesse et d'évolutivité à nos applications. Cet article a pour objectif de vous présenter les avantages et inconvénients de cette offre avec, aussi souvent que possible, un élément de comparaison ou une mise en situation pour mieux comprendre le PaaS v2.

Gestion du cycle de vie des applications Les conteneurs

Service Fabric supporte actuellement les conteneurs Docker sur Linux et les conteneurs Windows Server sur Windows Server 2016. Le support des conteneurs Hyper-V est prévu lorsque Windows Server 2016 sera en disponibilité générale (prévu mi-octobre 2016). Prudence, tous les modèles de services ne sont pas disponibles selon l'OS. L'intérêt d'utiliser les conteneurs dans Service Fabric est le même que dans un cas classique : portabilité, haute densité, rapidité de création et plus léger.

Mise à niveau propagée

Lorsqu'un nouveau package d'application est publié, Azure Service Fabric compare le nouveau manifeste d'application avec celui déjà présent (s'il existe) afin de déterminer quels sont le ou les services qui ont réellement été modifiés pour ne mettre à jour que ce qui est nécessaire. Rappelez-vous, dans la partie modèle d'application, j'avais précisé que les fichiers manifestes contenaient les éléments constitutifs d'une application avec leur version. Afin d'offrir une haute disponibilité, Azure Service Fabric procède à la mise à jour successive de chaque domaine de mise à jour (sous-ensemble de nœud) d'une application, c'est pourquoi elle est dite propagée. La seule « contrainte » inhérente à cette continuité de service, est que la nouvelle version de l'application doit être rétro-compatible avec la version précédente. Il est cependant possible de mettre à jour d'une seule traite le cluster en ne spécifiant qu'un seul domaine de mise à jour dans le manifeste du cluster. Néanmoins, cela implique une interruption de service due à l'arrêt complet du cluster. L'intelligence réside dans les contrôles effectués pour déterminer si le déploiement s'est bien passé, que le package a bien été copié et que l'instance a démarré. Dès lors que tout est ok, Service Fabric continue la mise à niveau au prochain domaine de mise à jour. Sachez qu'il existe trois niveaux de suivi pour les mises à niveau :

- Surveillé : Mode expliqué auparavant ;
- Non surveillé automatique : Service Fabric n'attend pas que le domaine soit pleinement fonctionnel et n'effectue pas les vérifications d'intégrité. Par conséquent les mises à jour sont plus rapides ;
- Non surveillé manuel : idem que le précédent sauf que cela requiert une intervention humaine pour sélectionner le domaine de mise à jour suivant.

Mise à l'échelle Planification des capacités

Lorsque vient le moment de créer son cluster Service Fabric, il vous faut choisir le type de machine et leur nombre. Pour déterminer le nombre de nœuds, Microsoft recommande la règle de calcul suivante :

Nombre de nœuds = (Taille de l'état en Go * nombre de réplique) / Mémoire en Go disponible sur un nœud

Pour de meilleures performances, il est conseillé de choisir des nœuds capables de garder l'état complet en mémoire. De plus, il est préférable de sur provisionner pour gérer les pics de charge ou les pannes matérielles. Sans connaissance des pics de charge, il est conseillé d'ajouter 5 à 10%.

Service sans état vs avec état

Pour augmenter la capacité de traitement d'un service sans état, il suffit de rajouter une instance supplémentaire.

[Pour le service avec état, c'est plus compliqué car cela implique de répliquer son état à travers plusieurs services afin de garantir une tolérance à la panne. Dans un contexte où ce service possède plusieurs centaines d'instances, cela peut engendrer une surcharge non négligeable.]

Instance homogène vs hétérogène

Dans un environnement On Premise, il est plus difficile de maintenir un cluster avec des nœuds homogènes, c'est-à-dire avec les mêmes capacités matérielles. Les pannes matérielles et autres feront qu'avec le temps celui-ci deviendra inévitablement hétérogène. Est-ce un mal ? Pas forcément mais il est plus difficile d'appliquer une configuration juste et précise du service d'allocation de ressources pour que chaque nœud ne soit pas sous-utilisé et/ou que d'autres ne puissent traiter la charge.

Monitoring

Dans une offre PaaS, le monitoring est une fonctionnalité clé puisque l'infrastructure sous-jacente n'est pas accessible. Avec Service Fabric, votre outil principal est l'interface Web de gestion du cluster disponible à l'adresse suivante : [http\(s\)://\[domaine du cluster\]:19080/explorer](http(s)://[domaine du cluster]:19080/explorer). Celui-ci permet de visualiser la santé de chaque entité selon quatre états : OK, avertissement, erreur ou inconnu. Ces états sont obtenus de manière quasi temps réel afin de vous faciliter la gestion du cluster.

Testing

Une pratique qui a déjà prouvé son efficacité et qui encourage les développeurs à écrire du code résilient, est l'exécution de tests en environnement de production. D'ailleurs Netflix a été l'un des précurseurs dans ce domaine avec les outils Chaos Monkey et ensuite Simian Army (disponible sur GitHub). Le but de ces outils est double. D'abord, s'assurer que votre infrastructure est suffisamment redondée en cas de panne mais aussi de valider le bon fonctionnement de vos applications, en générant de manière aléatoire un certain nombre de pannes maté-

rielles. Néanmoins, toutes les entreprises sont assez réticentes à lancer ces tests en environnement de production car il existe toujours un risque de dégradation de la qualité de service ou même de déclencher ce qu'il est censé prévenir, c'est-à-dire une réelle interruption de service. C'est pourquoi, il est important d'être capable de rétablir votre infrastructure rapidement et cela ne concerne pas uniquement le code mais aussi les données car celles-ci peuvent être corrompues. Azure Service Fabric intègre nativement un système de test similaire afin de fournir des services toujours plus robustes. Voici les scénarios pris en charge :

- Redémarrage d'un nœud ;
- Redémarrage d'un package de code déployé ;
- Redémarrage d'un réplica (primaire ou secondaire) ;
- Suppression d'un réplica (primaire ou secondaire) ;
- Déplacement d'un réplica (primaire ou secondaire).

Backup

Même si une plateforme PaaS offre une tolérance à la panne, la fonctionnalité de backup reste indispensable. En effet, des événements comme la corruption des données suite à un bug applicatif ou tout simplement la perte du datacenter peuvent survenir. Dans le premier cas, une sauvegarde locale suffit et pour l'autre il est possible de déplacer celles-ci vers un répertoire externe comme Azure Storage.

CONCLUSION

Azure Service Fabric est le résultat de la métamorphose de Microsoft d'un éditeur de logiciel (en général monolithique) en fournisseur de services Cloud. Fort de cette double compétence, cela permet à Microsoft de proposer à ses utilisateurs des solutions uniques sur le marché par rapport aux autres fournisseurs Cloud comme par exemple Azure Stack. Et maintenant what's next ? Une prise en charge complète des conteneurs et l'intégration dans Azure Stack qui est prévue lors de sa disponibilité générale mi-2017.

DevCon #3

CONFÉRENCE PROGRAMMEZ! LE 23 MAI

API, Cloud & IoT

Programmez! organise sa 3e conférence technique sur 3 thèmes : les API, le Cloud Computing et les IoT.

- 2 keynotes
- 6 sessions techniques
- 1 pizza party

+ un track spécial sur la plateforme IoT "Constellation"

Informations & inscriptions sur
www.programmez.com section DevCon

Où : la conférence se déroulera
à Issy les Moulineaux.
A -5 minutes des RER C et T2.

Accueil à 13h30.
Début des conférences à 14h.
Fin à 19h30.



Le « **serverless computing** » avec OpenWhisk

• Anne-Cécile Baudriller
IBM

• Catherine Ezvan
IBM

La nouvelle solution de service Cloud a fait son apparition il y a quelques mois, il s'agit du « serverless computing », plus communément appelé « serverless ». Le serverless est un mode d'exécution à la fonction, c'est-à-dire que le code est confié à un environnement d'exécution qui va allouer à la volée des ressources quand le code aura besoin de s'exécuter. A la différence des anciennes technologies Cloud, le serverless est basé sur de l'événementiel. La fonction écrite ne sera exécutée qu'à la condition qu'elle soit appelée. Le concept de "serverless" requiert bien sûr des services d'infrastructure, mais ceux-ci restent en arrière-plan (on ne les voit pas). C'est pourquoi le "serverless" est aussi appelé "Function-as-a-Service".

Les applications sont conçues de deux manières : soit entièrement en mode « serverless », soit en utilisant un mélange de code traditionnel et de code serverless. Le haut niveau d'abstraction des infrastructures et du middleware permet de se concentrer sur la seule écriture du code.

Pourquoi cela fait le buzz ? La valeur ajoutée du serverless

Etre notifié quand vous avez un mail, faire de la reconnaissance d'image directement après « upload », être alerté quand la température de l'eau de votre poisson rouge est excessive... Toutes ces fonctionnalités peuvent être mises en œuvre grâce au serverless. Et plus précisément dans OpenWhisk, quand on opte pour cette implémentation Open Source de la fondation Apache.

A partir d'un événement, une fonction est appelée. Cette fonction peut être utilisée pour tester une valeur liée à l'événement avant de déclencher une action, le cas échéant. Prenons un exemple : nous pouvons imaginer un capteur de température qui prélève la température d'une pièce tous les quarts d'heure. Un scénario pourrait être celui-ci :

- Si $T < 15^{\circ}\text{C}$, allumer le radiateur ;
- Si $T > 20^{\circ}\text{C}$, éteindre le radiateur.

Dans ce cas, on voit que si la température est comprise entre 15 et 20 °C, aucune action ne sera réalisée.

C'est pourquoi on considère que le serverless apporte les promesses du PaaS : on ne paie que ce qui est réellement utilisé, c'est à dire les exécutions des actions.

Dans quel cas l'utiliser ?

Des microservices au cognitif, le serverless offre de nombreux cas d'utilisation.

Tout d'abord pour les microservices : l'appel à des APIs étant la base des microservices, le serverless donne la possibilité de les intégrer facilement. Les grosses applications sont divisées en plusieurs plus petites, améliorant ainsi le suivi de chacune. Avec la solution IBM, il est possible de faire à la fois des microservices et du cognitif : en effet, des services Watson sont disponibles en sortie des actions, en tant que services tiers. Ensuite pour l'IoT : même s'il est possible d'utiliser les solutions traditionnelles d'architecture pour l'IoT, le serverless s'avère être plus adapté à ce type d'utilisation. Ici aussi, le manque de standardisation des réseaux, de la sécurité des données transitant et des différentes technologies amène

les développeurs vers des solutions plus optimisées, portées notamment par une conception « event-driven » et une utilisation élastique des serveurs. Un pic de traitement peut être envisagé sans poser de problèmes ni d'énormes contraintes, comme c'est le cas avec les architectures classiques.

Une utilisation sympathique est aussi le backend d'APIs, construites rapidement sans serveur grâce au serverless OpenWhisk. Ce dernier supporte la génération automatique d'API REST pour chaque action et s'implémente facilement avec le système de management d'API de votre choix. De même pour ce qui est du backend mobile, du cognitif ou encore du data processing. Voici quelques exemples d'applications Web qui illustrent ces propos : <http://www.sitespirit.nl/> , <https://12fprod.github.io/openwhisk-emoting/> , <http://openwhisk-darkvision.mybluemix.net/#/videos>.

De manière générale, on peut donc dire que le serverless est parfaitement adapté pour des traitements courts, stateless et de préférence basés sur des événements. Mais il peut être utilisé à bien plus large échelle.

Par la conception même du serverless, la plateforme OpenWhisk est facturée à la demande, c'est-à-dire qu'on ne paie que ce qu'on utilise. Les besoins en ressources sont ajustés de manière automatique selon les invocations individuelles. Plus besoin d'infrastructure propriétaire, ni de machines virtuelles, ni de containers.

OpenWhisk

La plateforme OpenWhisk est Open Source et c'est ce qu'IBM a privilégié pour sa plateforme Bluemix. L'évolution d'OpenWhisk est gérée par la fondation Apache. Au-delà de sa nature Open Source, elle se distingue par le support de langages multiples (Node.js, Swift, Python) et l'exécution de code arbitraire dans des containers Docker. Par ailleurs, cette architecture bénéficie d'intégrations multiples avec des services de la plateforme Bluemix. Le schéma suivant illustre le « serverless » à haut-niveau dans l'environnement Bluemix : [1]

OpenWhisk repose sur des actions générées par des événements. Elle permet aux développeurs de se focaliser sur le code tandis que les questions de résilience, de scalabilité, de disponibilité sont laissées à la charge de la plateforme. Les technologies très ouvertes permettent de développer sur OpenWhisk puis d'aller sur d'autres supports.

Les actions sont des fonctions, donc des snippets ou du code embarqué dans un container Docker. Quand les actions sont invoquées, elles sont instantanément déployées et exécutées quand un trigger est déclenché.

Logiquement, plus un trigger se déclenche souvent, plus l'action s'exécutera régulièrement. Si aucun trigger n'est invoqué, l'action ne se lancera pas et les coûts stagneront. Il est possible d'invoquer des actions par la commande OpenWhisk (wsk), des API ou le SDK d'iOS.

Sur OpenWhisk, vous pouvez créer des actions, des séquences, des règles et des triggers. Les actions sont appelées par des événements, et sont composées d'instructions qui s'exécutent quand l'action est lancée. Une règle associe un trigger à une action, avec chaque activation du trigger déclenchant l'exécution de l'action avec les paramètres passés par le trigger. A un trigger peuvent être associées plusieurs règles, et donc plusieurs déclenchements d'actions. Et plusieurs triggers peuvent être associés à la même action.

Les règles sont utilisées pour canaliser les sources d'événements internes comme externes. Il peut s'agir de règles de temps (prélever la température toutes les 5 minutes) ou règles de gestion (à chaque fichier enregistré dans une base de données, le nom et la date de téléchargement du document sont sauvegardés). Les séquences sont des enchaînements d'actions, c'est-à-dire que la sortie d'une action sera l'entrée de la suivante.

Il est possible de se connecter à des packages avec des services additionnels, qui lient une action OpenWhisk au monde extérieur. Les services de Bluemix, la plateforme PaaS d'IBM, peuvent alors être utilisés pour relier des informations. Un catalogue permet de se connecter à des packages utiles comme GitHub, un service de bases de données Cloudant, Slack ou The Weather Company.

La solution Open Source OpenWhisk repose sur des géants de l'open source comme Nginx, Kafka, Consul, Docker, et CouchDB. Voici un schéma du process de lancement pour une action : [2]

Concrètement, quand une action est invoquée, elle tourne dans un container qui avait été préalablement provisionné. OpenWhisk peut être

complété avec des services Bluemix comme les services cognitifs de Watson ou des bases de données.

TUTORIEL : PUSH TO SLACK

Cet exemple se base sur une action appelée par un trigger GitHub. Le trigger est invoqué ici par un push dans un répertoire défini de Git. Quand le trigger sera invoqué, l'action se déclenchera et publiera un message dans un channel Slack en donnant quelques informations à propos de ce push.

Avant tout, allons faire un tour sur la plateforme OpenWhisk dans Bluemix : <https://console.ng.bluemix.net/openwhisk/>. Si vous n'avez pas encore de compte Bluemix, c'est le moment de vous en créer un !

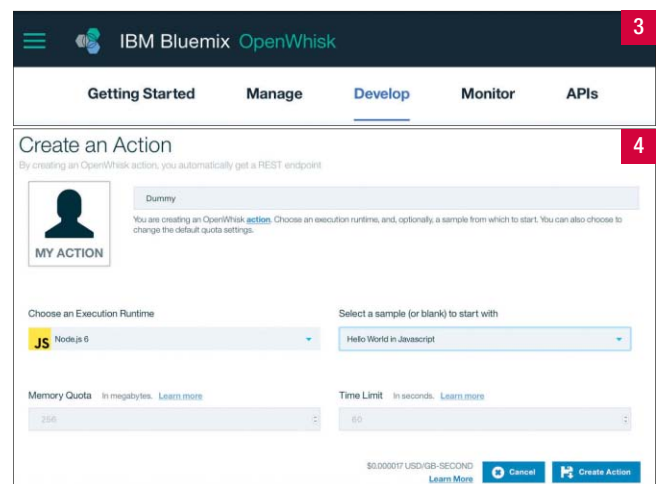
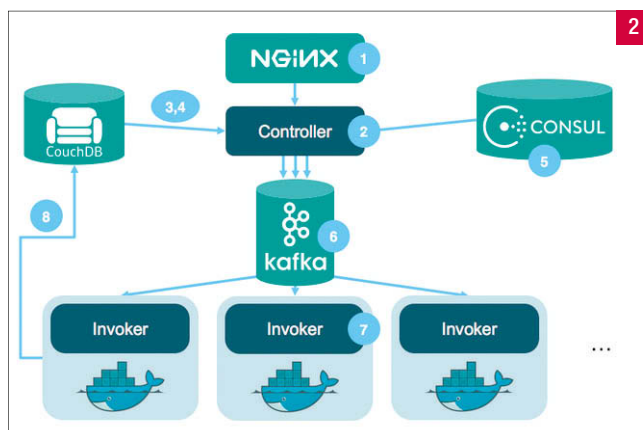
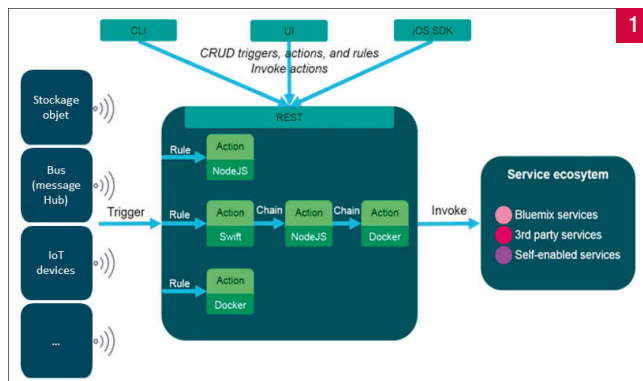
Le premier maillon de la chaîne : de GitHub à OpenWhisk

Dans la plateforme Bluemix, aller dans le menu OpenWhisk et plus précisément dans l'éditeur Web. Choisir l'onglet « Develop » dans le menu OpenWhisk. [3]

Tout d'abord, il faut créer une action. Dans un premier temps, on crée une action « Dummy » avec le code Node.js « Hello World ». On travaillera sur le lien vers Slack dans un 2ème temps. [4]

Après avoir cliqué sur « Create Action », l'action est effectivement affichée. La prochaine étape consiste à créer le trigger. Pas besoin de séquence ici, une seule action devant être exécutée. On clique donc sur « automatiser cette action » ou « Automate this Action ». A noter que dans Bluemix, la langue de l'interface utilisateur (UI) est celle du navigateur. Pour les copies d'écran, on a utilisé un navigateur anglais (un browser). [5]

Dans l'étape suivante, on choisit GitHub comme type de trigger et on clique sur « New trigger ». [6]



Ci-dessous, dans les paramètres, on renseigne le nom du trigger, les informations relatives au repository GitHub, et les actions pour lesquelles on veut être notifié (uniquement pour le push ici). [7]

Une fois la configuration sauvegardée, on est prêt à passer à l'étape suivante : [8]

Cette étape 3 consiste à confirmer la création de la règle qui associe l'action au trigger, ici "Dummy" à "PushInMyRepo". [9]

L'étape 4 est tout aussi simple, puisqu'il s'agit de donner un nom à la règle. Pour faire simple, on garde le nom proposé par l'éditeur graphique de Bluemix.

Et voilà ! La règle est créée à son tour. Nous avons donc : une action, un trigger et une règle. [10]

Il ne reste plus qu'à tester. C'est le moment de faire des modifications dans le projet et de les pousser sur le répertoire Git. Comment vérifier que cela fonctionne ? Facile : il suffit de regarder les logs. Grâce à la console graphique de Bluemix, il est facile de visualiser les logs. On les trouve dans l'onglet

« Monitor » du tableau de bord OpenWhisk. [11]

Comme attendu, trois étapes sont affichées dans la console (pas nécessairement par ordre d'exécution) :

- L'invocation de l'action « Dummy » ;
- Le trigger PushinMyrepo est déclenché, suite au push sur le repository Git ;
- La règle associée au trigger est activée, à savoir « On github-webhook – using PushinMyRepo – invoke Dummy ».

A ce stade, l'action est un simple « Hello World ». L'étape suivante consiste à effectivement implémenter le lien vers Slack.

Le deuxième maillon de la chaîne : d'OpenWhisk à Slack

Dans un second temps, on cherche à publier un message dans un channel Slack pour annoncer les modifications dans GitHub, plus exactement le « push » dans Git.

La publication dans Slack peut se faire en faisant appel à un webhook Slack. Concrètement, cela passe par une requête HTTP avec un payload en JSON qui inclut le texte du message à publier. Ce texte est construit dans une nouvelle action NotifyInSlack, à partir des paramètres passés à l'action par le trigger.

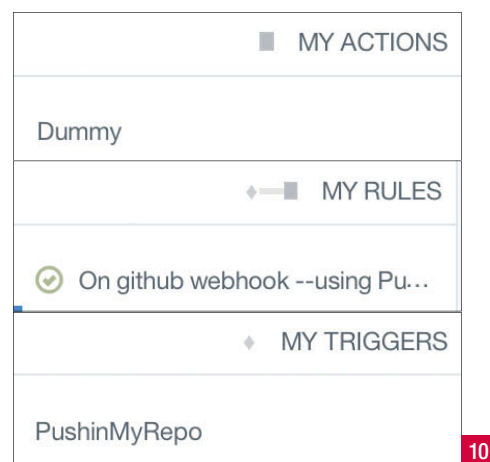
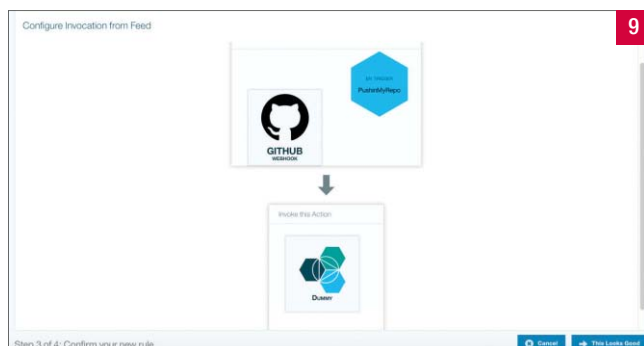
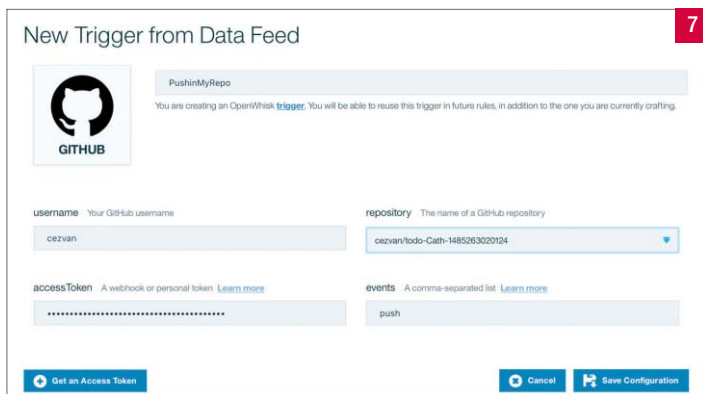
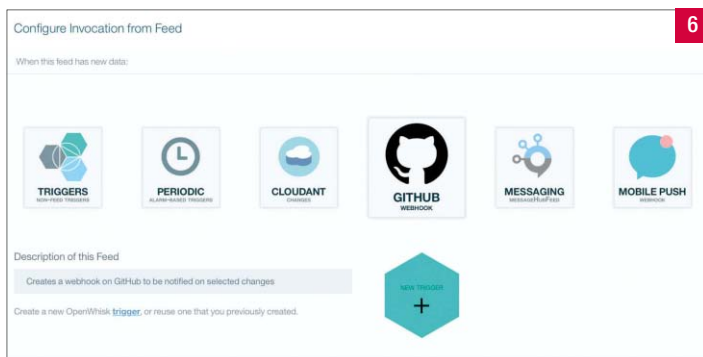
Comme dans la 1ère partie, on commence par créer une action. Cette nouvelle action s'appelle « NotifyInSlack ». [12]

Une fois l'action créée, on va se servir d'un package tout prêt pour effectivement publier dans Slack. Dans la page « Actions », il suffit d'aller dans « Parcourir les packages publics » ou « Browse Public Packages ».

[Browse Public Packages](#)

Ces packages permettent de lier vos actions avec des services utiles, dans Bluemix ou en dehors de Bluemix. [13]

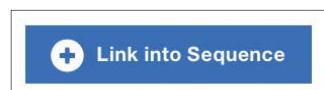
On a choisi de vous montrer comment cela se faisait avec Slack, cas



d'usage utile en particulier dans le cadre d'un travail de groupe. On sélectionne donc le package Slack. [14]

On est ainsi prêt à configurer l'action pour poster dans Slack. Pour créer un nouveau lien, on clique sur « New Binding + »

On configure la nouvelle connexion comme représenté ci-dessous. On renseigne les informations pour accéder au channel, à savoir : le username Slack, le nom du channel concerné (sans le # !) et l'url du webhook précédemment généré. [15]. Dans l'étape suivante, on crée le lien de ce package avec l'action dans une séquence.



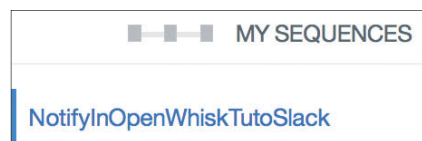
On va retrouver l'action dans « My Actions ». [16]

On choisit l'action "NotifyInSlack" et on clique sur « Add to Sequence ».

On peut alors choisir le sens de l'invocation. Ici on veut que l'action « Slack Post » soit invoquée une fois l'action NotifyInSlack exécutée. Donc on change le sens (et donc l'ordre de la séquence). [17]

This Looks Good! Ça s'annonce bien ! On donne un nom à la séquence, et on sauvegarde. [18]

Cette nouvelle séquence apparaît dans la liste.



Il va falloir coder un peu pour mettre en forme le message qui va être publié dans Slack. On ne veut pas que « Hello World » apparaisse dans Slack ! ;-)

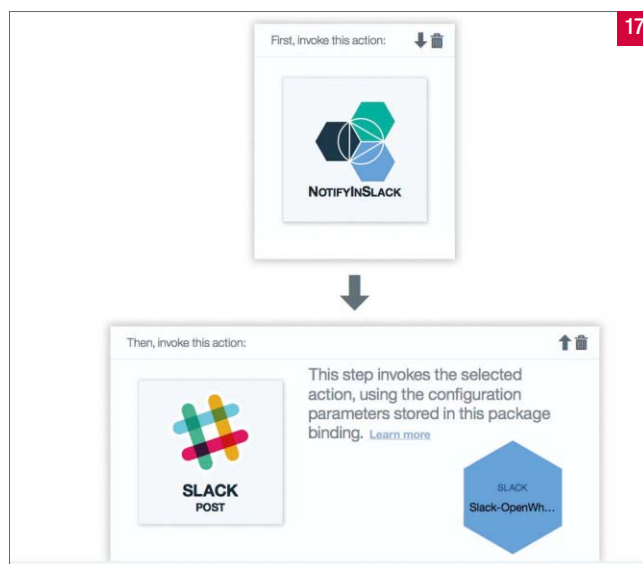
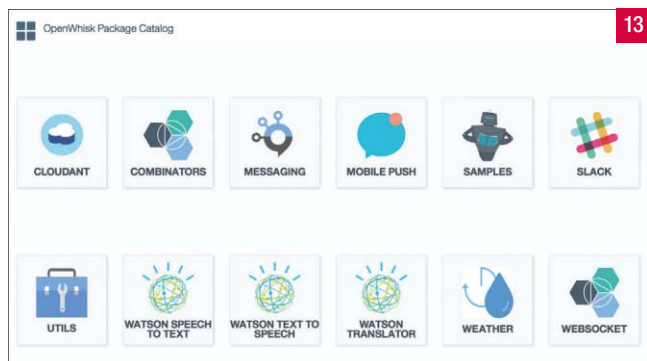
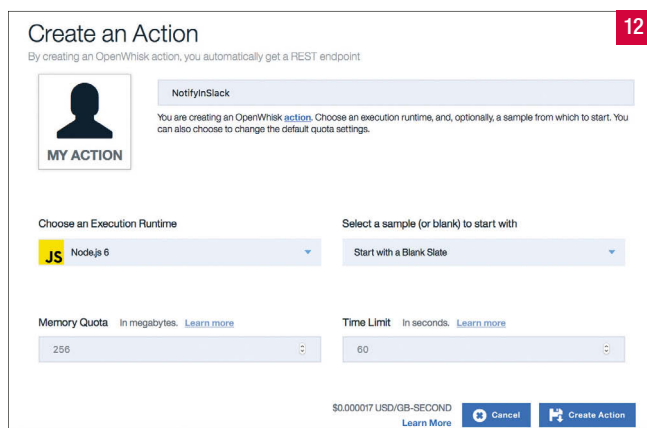
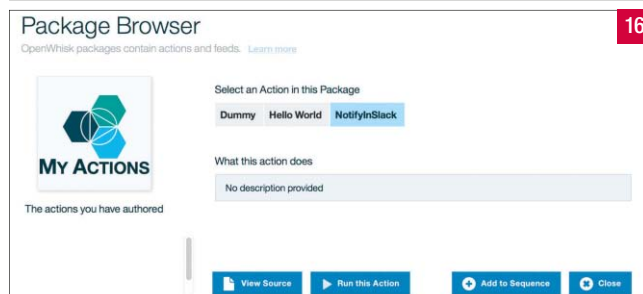
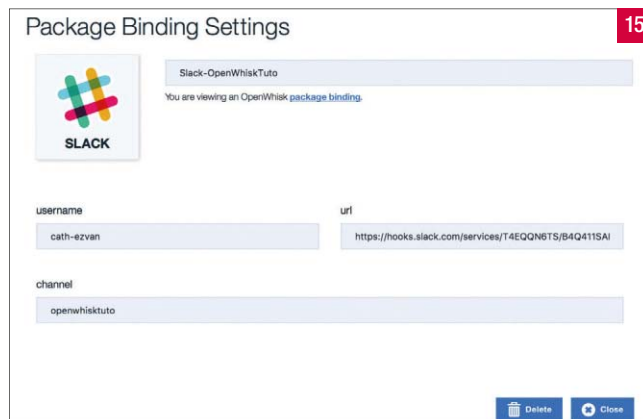
L'action NotifyInSlack, implémentée ici en Node.js, reçoit en paramètre le message passé par le trigger. Ce message n'est autre que celui prove-

nant du webhook GitHub. Pour connaître le format du message et avoir un exemple de message JSON, on se réfère à la documentation :

<https://developer.github.com/v3/activity/events/types/#pushevent>.

Et l'action NotifyInSlack doit passer un message à l'action suivante dans la séquence, ici l'action « Slack Push ». Ce nouveau message doit contenir le texte à publier dans Slack. On va donc prendre quelques éléments significatifs du message Git en entrée et retourner un petit texte.

Le code par défaut :



```
function main(params) {
  return { message: 'Hello World' };
}
```

est donc remplacé par :

```
function main(params) {

  var publication = params.pusher.name + ' a fait un push sur le répertoire ' +
  params.repository.name + ' le ' + params.repository.updated_at + ' et l'url est la
  suivante : ' + params.repository.git_url;

  return { text: publication };
}
```

Il ne reste plus qu'à déclarer une nouvelle règle pour lier le trigger et la nouvelle séquence. C'est notre prochaine étape. On clique sur « Automate ».

Il suffit de sélectionner la catégorie GitHub et de préciser le trigger GitHub à lier à une action. On réutilise le trigger de notre 1ère partie de tutoriel, à savoir « PushinMyRepo ».

On passe à l'étape suivante dans laquelle on retrouve la règle qui définit l'enchaînement suivant : [19]

A l'étape suivante, on donne un nom à cette règle. On garde celui par défaut, qui veut bien dire ce qu'il veut dire : « On github webhook -- using PushinMyRepo --invoke NotifyInSlack »

On sauvegarde la règle. Et voilà ! On est prêt pour le test. Une nouvelle modification dans le projet Git. Et on regarde dans Slack. [20]

VERS DU GREEN IT

Depuis une vingtaine d'années, on a vu le nombre et la taille des « data centers » exploser. L'énergie consommée par ces centres est très élevée, augmentée par tous les serveurs qui sont provisionnés seulement « au cas où ... » en prévision d'un dommage. Les grands acteurs du numérique travaillent à des serveurs plus « verts ». Et le Cloud permet un provisionnement plus vert, parce qu'adapté aux besoins de l'utilisateur. Le magazine Forbes le mentionnait : « les serveurs classiques des entreprises et des data centers utilisent en moyenne entre 5 et 15% de leur maximum de production sur un an. ». Ce qui a un très fort impact environnemental.

“Typical servers in business and enterprise data centers deliver between 5 and 15 percent of their maximum computing output on average over the course of the year.” Forbes

Dans cette optique, des progrès ont déjà été faits avec l'infrastructure

Cloud qui permet aux entreprises de provisionner des serveurs « à la demande ». A un plus haut niveau, le serverless est un fonctionnement encore meilleur pour la planète parce que seules les actions nécessaires sont exécutées. Le client n'est plus chargé lui-même du choix de son infrastructure, mais il laisse au fournisseur le soin d'ajuster les infrastructures nécessaires en temps réel. De cette manière, il n'y a plus de serveurs provisionnés systématiquement pour gérer les cas rares de pics de charge. La demande est gérée automatiquement et les serveurs sont provisionnés au fur et à mesure des besoins, que ce soit en charge nominale ou en cas de forte demande.

CONCLUSION

Pour aller plus loin : apprenez à utiliser les services OpenWhisk sur Bluemix :

https://console.ng.bluemix.net/docs/openwhisk/openwhisk_catalog.html#openwhisk_ecosystem ou suivez les tutoriaux de Niklas Heidloff

<http://heidloff.net/?s=openwhisk&submit>.

Vous pouvez aussi voir des exemples d'applications construites pour l'IoT : <https://developer.ibm.com/tv/project-openridge-improving-customer-service-iot-event-driven-cloud-computing/>

Références :

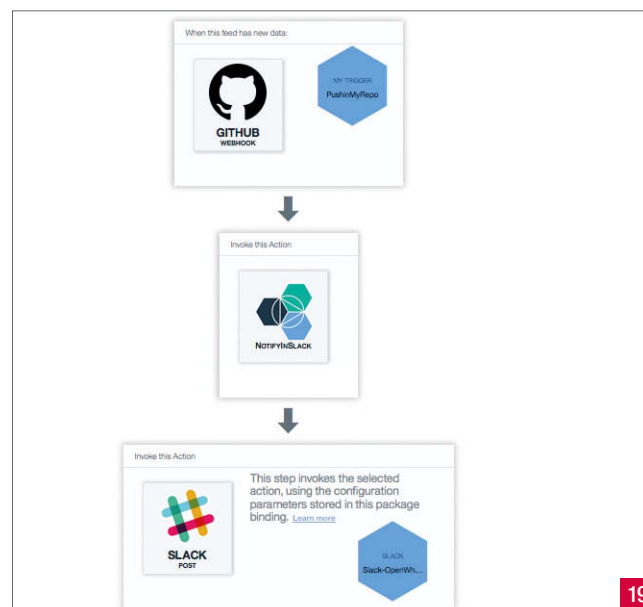
[The Wall Street Journal: Look Ma, No Server: Corporate IT Expects 'Serverless' Computing to Trigger Big

Changes](<http://www.wsj.com/articles/look-ma-no-server-corporate-it-expects-serverless-computing-to-trigger-big-changes-1477509674>)

[Forbes: Three Ways That "Serverless" Computing Will Transform App Development In 2017](<http://www.forbes.com/sites/ibm/2016/11/17/three-ways-that-serverless-computing-will-transform-app-development-in-2017/#3ac20ba8565a>)

[Forbes: Why Enterprises Should Care About Serverless Computing](<http://www.forbes.com/sites/janakirammsv/2016/10/12/why-enterprises-should-care-about-serverless-computing/#16a186385555>)

<https://martinfowler.com/articles/serverless.html>



cath-ezvan APP 8:11 PM
cezvan a fait un push sur le répertoire todo-Cath-1485263020124 le 2017-01-24T13:04:5Z et l'url est la suivante : [git://github.com/cezvan/todo-Cath-1485263020124.git](https://github.com/cezvan/todo-Cath-1485263020124.git)

Infrastructure as Code

- Michel Hubert,
 - Manon Pernin,
 - Etienne Deneuve
- cellenae**
DOESITRETER | Services & Solutions



Quel informaticien n'a jamais voulu automatiser les installations de ses serveurs ? L'automatisation du déploiement des ressources a toujours été un enjeu dans de nombreuses entreprises. En effet, à l'heure du Cloud Computing, il devient difficile d'annoncer un délai de 3 mois pour deux serveurs. Plutôt que de trouver des nouvelles méthodes, pourquoi ne pas employer celles des équipes de développement qui ont déjà fait leurs preuves ?

Les concepts

De façon simple, l'Infrastructure as Code est le fait de « coder » son infrastructure, d'exécuter des scripts pour gérer son infrastructure. L'utilisation de scripts par les équipes Infrastructure n'est pas nouvelle ; l'émergence du Cloud, les démarches Agile et DevOps ou encore les architectures Microservices vont tirer profit de l'Infrastructure as Code.

Cycle de vie des applications

Dans une démarche DevOps, les développeurs et les « Ops » doivent collaborer pour mener à bien un projet. L'infrastructure va ainsi évoluer pour s'adapter à l'évolution des applications.

Exemples :

- La nouvelle version de l'application nécessite l'installation d'un nouveau framework.
- Une nouvelle application doit être déployée sur un nouveau site Web et s'appuyer sur une nouvelle base de données.

Les besoins en termes d'infrastructure dépendent des applications gérées, l'infrastructure as Code doit donc s'intégrer dans le cycle de vie des applications !

Infrastructure as Code = Code

Comme son nom l'indique, l'Infrastructure as Code est avant tout du code. Il convient alors de penser son infrastructure comme on pense son code. Le code de l'infrastructure va entrer dans la chaîne logicielle de développement et on va pouvoir utiliser les mêmes outils que les développeurs pour archiver son code dans un contrôleur de sources et le versionner. Le code de l'infrastructure peut même être intégré dans le package des applications ! La collaboration entre développeurs et « Ops » est dans ce cas primordiale.

Scripts déclaratifs et Idempotence

L'Infrastructure as Code, est du code ... avec ses spécificités. Puisque le déploiement d'un environnement peut être assez long, on va éviter de supprimer l'ensemble d'un environnement si cela n'est pas nécessaire. Avant de déployer son application, on veut simplement s'assurer que l'infrastructure est conforme à ce que nécessite l'application. On va donc comparer un état actuel de l'infrastructure à un état attendu.

Dans un script impératif (scripts traditionnels), il fallait bien penser à tester cet état actuel (existence d'un élément et configuration) avec l'état attendu avant d'éventuellement lancer la création des éléments manquants ou la mise à jour de certains éléments pour avoir les caractéristiques attendues. Pour faciliter la réalisation de ces scripts, l'utilisation des

scripts déclaratifs est conseillée. Dans ce type, il suffit de « déclarer » la liste des éléments souhaités (VM, Base de données, Framework...) avec leurs caractéristiques ainsi que la dépendance entre ces éléments. Lors de l'exécution du script, une comparaison est faite entre l'état actuel et l'état attendu, et seul l'écart entre ces 2 états est réalisé. Ce type de script permet donc d'aboutir toujours au même résultat, peu importe l'état initial et peu importe le nombre de fois que ce script est exécuté. **C'est ce qu'on appelle l'idempotence.**

Fonction mathématique de l'idempotence : $f(f(x)) = f(x)$

Déploiement

La mise à jour de l'infrastructure étant réalisée par l'exécution du script, l'assistance d'un « Ops » n'est plus nécessaire. Le script aura été validé et testé au préalable par les « Ops » et l'exécution de ce dernier peut donc être réalisée par les développeurs qui déploient leur application. Les process de déploiements des applications sont généralement automatisés à l'aide d'outils de Release Management. Dans ce cas, une étape liée à l'infrastructure doit être ajoutée au process pour s'assurer de l'adéquation entre l'infrastructure en place et l'application déployée.

Bénéfices

En résumé, l'Infrastructure as Code doit permettre d'améliorer :

- **Le délai la livraison d'une application** : plus besoin d'attendre la disponibilité des « Ops » pour vérifier ou mettre à jour l'infrastructure avant le déploiement des applications ;
- **La qualité du déploiement de mes applications** : l'infrastructure est la même sur mes environnements de tests et ceux de Production (avec des caractéristiques différentes évidemment : l'environnement de Production est plus puissant que les autres par exemple) ;
- **L'utilisation des ressources et le budget** : il est désormais possible de supprimer des environnements lorsqu'ils ne sont pas utilisés, et de les recréer lorsqu'on en a besoin. Ceci est d'autant plus vrai dans un environnement Cloud au sein duquel la facturation se fait à l'utilisation.

Les cas d'utilisation

Le périmètre couvert par l'Infrastructure as Code ne se limite pas au déploiement du socle de l'infrastructure uniquement. C'est en fait une partie du Release Management qui peut être couvert par l'Infrastructure as Code. [1]

Si on prend un process de déploiement simple, l'Infrastructure as Code permet de gérer les étapes de :

- **Déploiement de l'infrastructure** : le socle de l'infrastructure, c'est-à-dire les type d'éléments que l'on souhaite créer (VMs, Réseau...) et leurs caractéristiques (CPU, Mémoire...) ;
- **Configuration de l'infrastructure** : l'ensemble des fonctionnalités et des logiciels qui doivent être présents pour que l'application puisse tourner (Server Web, Sites Web, Bases de données, Framework...) ;
- **Déploiement et configuration de l'application** : cette étape est souvent déjà gérée par un outil de Release Management, néanmoins il est possible de l'inclure dans vos scripts d'Infrastructure as Code.

Selon le contexte dans lequel on se trouve et la stratégie choisie, il est possible de mettre en place de l'Infrastructure as Code uniquement pour la partie déploiement du socle de l'infrastructure, uniquement pour la partie configuration mais aussi pour gérer l'ensemble des étapes via de l'Infrastructure as Code.

Hébergement On-Premise

Si vos VMs sont hébergées On-Premise (ou chez un hébergeur), le déploiement de VMs et la configuration réseau risquent d'être difficilement automatisables. Par contre, il est possible de tirer profit de l'Infrastructure as Code pour gérer la configuration de vos environnements. Il sera par exemple possible de définir la configuration souhaitée pour toutes les VMs utilisées pour héberger des sites Web et celle pour toutes les VMs hébergeant des bases de données. Vous pourrez également superviser la configuration de vos VMs et être alertés en cas de configuration manquante.

Hébergement dans le Cloud

Dans le cas d'un hébergement dans le Cloud, autant profiter de la souplesse que cela offre pour automatiser le déploiement du socle de l'Infrastructure. Vous pourrez ainsi optimiser les coûts liés à la consommation, uniquement, de vos ressources. Pour les environnements de test, par exemple, il vous sera possible de supprimer les environnements lorsqu'ils ne sont pas utilisés et de les créer uniquement pour tester une nouvelle version d'une application.

Remarque : il est également possible d'utiliser des fonctionnalités du Cloud pour gérer la configuration des VMs hébergées On-Premise !

Les outils

Déploiement de l'infrastructure	Configuration de l'infrastructure	Orchestration
	Azure Resource Manager	Release Management (VSTS ou TFS)
Terraform	PowerShell DSC Azure Automation DSC	Azure Automation
Ansible	Chef Puppet	Orchestrator Octopus Deploy
	Ansible	

Techniques (+ Exemples)

Microsoft

Plusieurs outils Microsoft permettent de faire de l'Infrastructure as Code, que les environnements soient On-Premise ou dans le Cloud Microsoft Azure. Voyons quelques exemples d'utilisation de ces outils.

Template Azure Resource Manager

Comme décrit dans les cas d'utilisation, un hébergement dans le Cloud offrira plus de souplesse au niveau de la suppression et création des environnements. Le Cloud Microsoft Azure propose pour gérer le déploiement des environnements d'utiliser **Azure Resource Manager**. Azure Resource Manager considère chaque composant de votre environnement comme une ressource (une VM, un réseau virtuel, une base SQL Azure...). Vous pouvez les organiser de façon purement logique dans des « Resource Group » pour faciliter la gestion d'un ensemble de ressources qui fonctionnent ensemble. Une bonne pratique est de regrouper des éléments qui doivent être déployés en même temps. Pour déployer un groupe de ressources, laas et/ou PaaS, vous avez la possibilité de construire un modèle Resource Manager (ou

« template ARM »). Ce template est un fichier JSON qui définit de façon déclarative les ressources à déployer ainsi que les dépendances entre ses ressources. L'exemple ci-dessous contient la déclaration de ressources de type « Microsoft.Sql/servers/databases », il s'agit d'une base SQL Azure. La propriété « copy » permet de définir la création de plusieurs bases à partir d'une liste passée en paramètre. La création de ces bases dépend de la création du serveur SQL Azure sur lequel seront hébergées ces bases, cette dépendance est définie à l'aide de la propriété « dependsOn ».

```
{
  "name": "[concat(parameters('sqlserverName'), '/', parameters('databasesList')[copyIndex()])]",
  "type": "Microsoft.Sql/servers/databases",
  "location": "[resourceGroup().location]",
  "tags": {
    "displayName": "[parameters('databasesList')[copyIndex()]]"
  },
  "copy": {
    "name": "databasesCopy",
    "count": "[length(parameters('databasesList'))]"
  },
  "apiVersion": "2014-04-01-preview",
  "dependsOn": [
    "[concat('Microsoft.Sql/servers/', parameters('sqlserverName'))]"
  ],
  "properties": {
    "edition": "[parameters('edition')]",
    "collation": "[parameters('collation')]",
    "maxSizeBytes": "[parameters('maxSizeBytes')]",
    "requestedServiceObjectiveName": "[parameters('requestedServiceObjectiveName')]"
  }
}
```

Le déploiement de groupes de ressources à partir de template ARM peut ensuite se faire de différentes façons :

- Directement via Visual Studio ;
- Par un script PowerShell et donc dans n'importe quel outil qui peut exécuter du PowerShell (Orchestrator, Azure Automation ou autres) ;
- Dans un process de déploiement (Release Management (VSTS ou TFS), Octopus Deploy...).

PowerShell DSC (Desired State Configuration)

Pour tout ce qui est configuration, **PowerShell DSC (Desired State Configuration)** est installé avec PowerShell dès la version 4.0 et vous permettra de gérer les configurations des VMs de vos environnements, qu'ils soient On-Premise ou dans le Cloud.

PowerShell DSC se base sur des scripts PowerShell, eux aussi en mode déclaratif : les configurations souhaitées sont listées et les dépendances entre les configurations peuvent être précisées. Dans l'exemple ci-dessous, les différents blocs définissent une configuration : installation de la fonctionnalité DNS, configuration de l'adresse DNS, formatage d'un disque, installation de la fonctionnalité Active Directory.

```
Node localhost
{
  WindowsFeature DNS
  {
    Ensure = "Present"
```

```

Name = "DNS"
}
xDnsServerAddress DnsServerAddress
{
  Address = '127.0.0.1'
  InterfaceAlias = $InterfaceAlias
  AddressFamily = 'IPv4'
}
xWaitforDisk Disk2
{
  DiskNumber = 2
  RetryIntervalSec = $RetryIntervalSec
  RetryCount = $RetryCount
}
cDiskNoRestart ADDDataDisk
{
  DiskNumber = 2
  DriveLetter = "F"
}
WindowsFeature ADDSInstall
{
  Ensure = "Present"
  Name = "AD-Domain-Services"
}

```

Une fois la déclaration des configurations décrite dans un script PowerShell DSC, la configuration doit être appliquée à une VM. Pour cela, vous avez la possibilité d'appliquer la configuration directement à votre VM à l'aide de son moteur DSC appelé « Local Configuration Manager ». Vous pouvez aussi passer par un « Pull Server » sur lequel vous pourrez centraliser vos scripts PowerShell DSC, ce qui peut être utile dans des environnements où un grand nombre de VMs doivent être configurées de la même manière. [2]

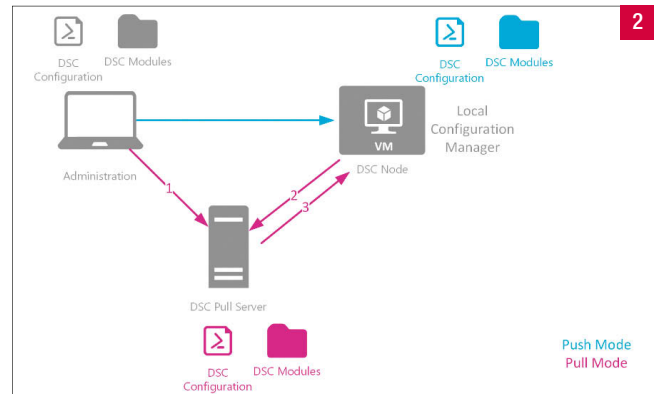
Pour faciliter la configuration d'un « Pull Server », Microsoft Azure propose la version PaaS : Azure Automation DSC. Azure Automation est un outil d'automatisation des processus hébergé dans le Cloud Microsoft Azure, c'est en quelque sorte la version Cloud de Orchestrator. Azure Automation DSC est une de ses fonctionnalités grâce à laquelle vous pouvez stocker vos scripts PowerShell DSC et gérer la configuration de vos VMs qui sont hébergées dans Azure **mais aussi celles hébergées On-Premise !**

Release Management (VSTS ou TFS)

Enfin, pour intégrer le déploiement et la configuration de l'infrastructure à vos process de déploiements, des étapes liées à l'infrastructure doivent être ajoutées. La fonctionnalité Release Management des outils Microsoft VSTS et TFS est utilisée pour définir vos process de déploiements. Des étapes spécifiques aux déploiements Azure Resource Manager existent, vous pourrez alors configurer votre template ARM avec vos scripts PowerShell DSC. Vous pourrez aussi utiliser d'autres outils pour ajouter des étapes de déploiement Azure Resource Manager. Octopus Deploy, par exemple, propose également des étapes spécifiques pour les déploiements ARM. Et sinon, vous avez toujours la possibilité de configurer le déploiement à l'aide d'une étape PowerShell.

RedHat : Ansible

Ansible dans sa version Open Source, permet d'automatiser le déplo-



2 modes de fonctionnement de PowerShell DSC : Push et Pull

ment de ressources Linux et Windows. La particularité d'Ansible est d'être Agentless, il supporte le SSH et le WinRM. Les fichiers d'infrastructure seront en YAML (si vous ne connaissez pas encore le YAML, attention !). Par exemple, si nous souhaitons sur une machine Linux un Nginx nous aurons un playbook du type :

```

...
- name: Configure instances
  hosts: (machines dans /etc/ansible/hosts)
  tasks:
    # On installe nginx
    - name: Installs nginx web server
      apt: pkg=nginx state=installed update_cache=true
      notify:
        - start nginx
      become: true

```

Il est également possible avec Ansible de déployer des machines dans les différents Clouds, AWS, Azure, Google Cloud Engine pour ne citer que les plus courants. Par exemple, si nous souhaitons créer une instance Linux dans Azure nous pouvons rédiger un YAML tel que : voir code complet sur www.programmez.com.

Pour exécuter ce playbook, il suffira (en plus de renseigner ses credentials d'accès en fonction du module utilisé) de lancer la commande « ansible-playbook playbook.yml ». Ansible possède aussi des fonctionnalités pour faire l'inventaire de vos ressources existantes. Cette fonctionnalité est très intéressante mais nous ne pourrions pas détailler tout son fonctionnement ici.

Hashicorp : Terraform

Terraform est disponible, en version entreprise et open source. Édité par HashiCorp, également connu pour Vagrant, Packer... Terraform n'est pas un outil similaire à DSC ou à Ansible puisqu'il permet de déployer de l'infrastructure sans nécessairement la configurer via Terraform. Les langages des fichiers de configuration Terraform sont assez simples à comprendre, il s'agit du HCL (HashiCorp Configuration Language). Avec 62 providers supportés, Terraform permet d'imaginer des scénarios intéressants (Alicloud, AWS, Docker, Kubernetes, Azure, Google, VMware vSphere pour n'en citer que quelques uns). Par exemple si on souhaite déployer une machine dans Azure avec Ansible de pré-installé : voir code complet sur www.programmez.com.

En suite, on exécute la commande « terraform plan <chemin vers le dossier> » et Terraform vérifie l'état actuel sans déployer ni modifier quoi ce soit. Lors du déploiement (terraform apply), il réutilise les informations

pour modifier que le strict nécessaire. Petite mention de la commande « terraform destroy » qui détruit bien tout ce qui a été déployé.

Docker

Comment parler d'infra as Code sans aborder Docker ? De nombreux articles ont déjà couvert le sujet, nous n'allons pas faire un article de plus là-dessus. En revanche si on souhaite déployer une instance Nginx dans un container sans plus de configuration, nous pouvons le faire très simplement :

Docker pull nginx

Docker run -d -p 80:80 nginx

Nous pouvons également faire un Dockerfile (fichier de configuration Docker) tel que :

```
# Pull base image.
FROM dockerfile/ubuntu

# Install Nginx.
RUN \
  add-apt-repository -y ppa:nginx/stable && \
  apt-get update && \
  apt-get install -y nginx && \
  rm -rf /var/lib/apt/lists/* && \
  echo "ndaemon off;" >> /etc/nginx/nginx.conf && \
  chown -R www-data:www-data /var/lib/nginx

# Define mountable directories.
VOLUME ["/etc/nginx/sites-enabled", "/etc/nginx/certs", "/etc/nginx/conf.d", "/var/log/
```

```
nginx", "/var/www/html"]
```

```
# Define working directory.
```

```
WORKDIR /etc/nginx
```

```
# Define default command.
```

```
CMD ["nginx"]
```

```
# Expose ports.
```

```
EXPOSE 80
```

```
EXPOSE 443
```

CONCLUSION

Pour faire de l'infra as code, comme toujours, la technique ne sera qu'un détail. Les outils sont globalement des choix à faire en fonction de votre environnement et des connaissances de vos équipes. Il n'est en revanche pas nécessaire d'être « DevOps » ou « Agile Ready », les outils peuvent vous aider à suivre les principes de ces démarches plus rapidement, mais ne construiront pas une belle usine logicielle et ne régleront certainement pas les problèmes structurels de votre entreprise si la communication n'est pas au rendez-vous. Il faudra les utiliser à tout le niveau de votre production, du « dev » à la prod. Pour bien comprendre, les enjeux et les problématiques, je vous recommande la lecture du livre « The Phoenix Project: A Novel about IT, DevOps, and Helping Your Business Win » écrit par Kevin Behr, George Spafford et Gene Kim.

Tous les numéros de
programmez!
le magazine des développeurs
sur une clé USB (depuis le n° 100).



34,99 €*

Clé USB 4 Go.
Photo non contractuelle.
Testé sur Linux, OS X, Windows. Les magazines sont au format PDF.

* tarif pour l'Europe uniquement.
Pour les autres pays, voir la boutique en ligne

☐ Clé USB PROGRAMMEZ 34,99 €

Commande à envoyer à : Programmez!
7, avenue Roger Chambonnet - 91220 Brétigny sur Orge

☐ M. ☐ Mme ☐ Mlle Entreprise : _____ Fonction : _____

Prénom : _____ Nom : _____

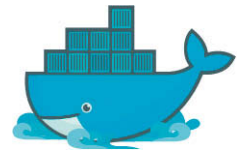
Adresse : _____

Code postal : _____ Ville : _____

E-mail : _____ @ _____

Règlement par chèque à l'ordre de Programmez !

Un “npm install” pas si trivial dans Docker...



• Jonathan Petitcolas
(@Sethpolma)
Développeur agile @marmelab
www.marmelab.com

En tant que développeurs, nous sommes souvent amenés à travailler sur divers projets hétéroclites, chacun utilisant un jeu de technologies différent. Et, lorsque deux clients partagent la même technologie, il n'est pas rare que les deux versions soient différentes. Installer toutes les technologies utilisées en plusieurs versions serait un vrai cauchemar sans Docker.

Un container Docker est une machine virtuelle très légère (pour simplifier, au risque de m'attirer les foudres des puristes). Docker permet ainsi de virtualiser les différents environnements utilisés par nos clients de manière très simple, sans risque de conflits, grâce à l'isolation des différents containers entre eux. Cela est d'autant plus simple que Docker fournit un *repository* centralisant les différents containers déjà configurés par la communauté. Que vous cherchiez un serveur PSQ, un serveur optimisé pour WordPress ou encore un serveur Go prêt à l'emploi, nul besoin de réinventer la roue : il suffit de télécharger le container correspondant...

Dans la suite de cet article, nous considérerons que Docker est installé et prêt à l'emploi sur votre machine.

Installation d'un serveur Node via Docker

Par exemple, imaginons que nous ne voulions pas installer Node.js en local sur notre machine. Il nous suffit de récupérer l'image Docker pour Node.js et de la lancer comme ceci :

```
docker run -it --rm -v "$PWD":/app -w /app node:7 node index.js
```

Nous utilisons ici l'image `node:7` récupérée du *Docker Repository*. L'option `--rm` indique que le container (une « instance » de l'image) doit être supprimé à la fin de la commande. L'option `-it` sert quant à elle à relier l'entrée standard de l'hôte à celles du container.

Grâce à l'option `-v`, nous partageons le dossier courant (`$PWD`) avec notre container (dossier `/app`). Nous nous plaçons à l'intérieur de celui-ci en spécifiant le *workdir* (option `-w`). Enfin, nous exécutons la commande `node index.js` à l'intérieur de notre container, dans son dossier courant. Avec un simple container, cette méthode est très simple et fonctionne parfaitement. Néanmoins, si nous commençons à ajouter d'autres containers (comme une base de données à travers un container PSQ, par exemple), l'utilisation de *docker-compose* s'avère cruciale. Docker-compose est un orchestrateur de containers. Grâce à lui, nous définissons les liens entre nos différents containers.

La configuration de `docker-compose` est rédigée au format YAML :

```
# /docker-compose.yml
version: '2'

services:
  db:
    # Quelle image doit-on télécharger ?
    image: postgres

    # Configuration des accès via des variables d'environnement
```

```
environment:
  - POSTGRES_USER=username
  - POSTGRES_PASSWORD=password

node:
  image: node:7
  working_dir: /app
  command: "node index.js"

# Partage du dossier courant dans /app du container
volumes:
  - ./app

# Le container sera accessible depuis l'hôte sur le port 3000
expose:
  - "3000"

# On ne démarre pas tant que le container db n'est pas dispo
depends_on:
  - db

# `db` et `node` doivent pouvoir communiquer
links:
  - db
```

Grâce à ce fichier, nous pouvons désormais démarrer nos deux containers (Node et PostgreSQL) en une simple commande : `docker-compose up`. L'architecture de notre projet est à présent définie dans la configuration de notre projet. Nous pouvons donc la versionner, et ainsi permettre à n'importe quel nouveau développeur sur le projet d'installer tout l'environnement nécessaire (avec les bonnes versions) en quelques minutes seulement. Un gain certain de productivité.

Droits des fichiers et Containers

À présent, installons les dépendances de notre projet Node. Comme nous n'avons pas Node en local, nous devons passer par le container. `docker-compose` fournit également une commande `run` fonctionnant de manière similaire à la commande Docker :

```
docker-compose run --rm --no-deps node bash -ci 'npm install'
```

Ici, pas besoin de l'option `-it`. En revanche, nous utilisons l'option `--no-deps` qui indique que nous pouvons démarrer le container individuellement. Cela nous évite un délai supplémentaire dans la mesure où cette tâche ne nécessite aucune connexion à la base de données.

Cette commande devrait s'exécuter sans problèmes. Néanmoins, si nous vérifions les permissions de nos fichiers créés par le container dans `node_modules`, une désagréable surprise nous attend :

```
$ ls -al node_modules

total 3184
drwxrwxrwx 784 root root 32768 Jan 26 17:47 .
drwxr-xr-x 19 root root 4096 Jan 26 17:46 ..
drwxr-xr-x 3 root root 4096 Jan 26 17:47 abab
drwxr-xr-x 2 root root 4096 Jan 26 17:47 abbrev
drwxr-xr-x 3 root root 4096 Jan 26 17:47 accepts
```

On peut constater que Docker a créé tous les fichiers en tant que `root`. Ces permissions ne sont pas gênantes en local : il suffit d'exécuter un `sudo chown`. Néanmoins, cela peut poser des problèmes après déploiement. En cas de sous-traitance de l'exploitation, nous ne possédons pas (ou du moins, nous ne devrions pas) les droits root sur les machines, rendant impossible la manipulation des `node_modules`. Ce qui est utile par exemple pour supprimer les vieilles releases. Il n'y a pas de solutions simples à ce problème. La plus rapide consiste à ajouter l'option `user` à notre fichier YAML. Ainsi, nous indiquerons à Docker quel UID (`user id`) et GID (`group id`) utiliser pour lancer le container.

```
services:
  node:
    user: "${UID}:${GID}"
```

Comme nous avons utilisé l'écriture ``$XXX``, ``docker-compose`` remplacera automatiquement ces deux variables par les deux variables d'environnement éponymes. Il nous suffit donc d'exporter manuellement nos identifiants d'utilisateur et de groupe pour corriger ce problème de permissions.

Néanmoins, assigner ces deux valeurs se fait manuellement pour l'instance. Il y a de nombreuses chances que nous oublions de les exporter, causant de nouveau des problèmes de droit. C'est pourquoi nous allons utiliser un `Makefile` afin de gérer tout cela automatiquement :

```
export $UID = $(id -u)
export $GID = $(id -g)

install:
  docker-compose run --rm --no-deps node bash -ci 'npm install'
```

Les commandes ``id -u`` et ``id -g`` retournent les identifiants respectifs de l'utilisateur courant et de son groupe. Il est nécessaire de les exporter afin qu'elles soient présentes dans l'environnement courant, permettant ainsi à Docker de les récupérer.

Pro-tip : généralement, pour chacun de nos projets sur lesquels nous intervenons, nous mettons en place trois commandes assez explicites : ``make install``, ``make run`` et ``make test``. Ainsi, n'importe quel nouveau venu sur le projet est capable de démarrer rapidement sur le projet. Si nous tentons d'installer de nouveau nos dépendances :

```
# On supprime les anciennes dépendances installées en root
$ sudo rm -Rf node_modules/
```

```
# Et on les réinstalle avec notre nouvelle commande.
$ make install
```

```
# On vérifie les permissions...
$ ls -al node_modules
```

```
total 3184
drwxrwxrwx 784 jpetitcolas jpetitcolas 32768 Jan 26 17:47 .
drwxr-xr-x 19 jpetitcolas jpetitcolas 4096 Jan 26 17:46 ..
drwxr-xr-x 3 jpetitcolas jpetitcolas 4096 Jan 26 17:47 abab
drwxr-xr-x 2 jpetitcolas jpetitcolas 4096 Jan 26 17:47 abbrev
drwxr-xr-x 3 jpetitcolas jpetitcolas 4096 Jan 26 17:47 accepts
```

Celles-ci ont désormais les permissions de l'utilisateur courant. Victoire ! \o/

Installer une dépendance privée depuis GitHub

Tout fonctionne correctement. Néanmoins, les ennuis commencent à partir du moment où l'on désire installer une dépendance privée. Notre dépendance privée n'a pas encore été entièrement finalisée et nous ne voulons donc pas la publier immédiatement sur npm. Nous décidons donc de l'installer depuis son *repository* GitHub.

Pour installer un package depuis GitHub, il suffit d'indiquer le nom d'utilisateur (ou d'organisation) et le nom de projet dans notre fichier `package.json` :

```
{
  // ...
  "my-private-bundle": "Acme/my-private-bundle#develop"
}
```

Il est également possible de spécifier une version (un tag, une branche, un commit) en suffixant l'emplacement de la dépendance par un dièse suivi du nom de la référence.

Si nous refaisons un `make install`, nous recevrons une erreur 401 de la part de GitHub, nous indiquant que nous ne sommes pas autorisés à accéder à ce *repository* privé.

Transfert d'agent SSH entre hôte et container

Ce comportement est logique. Nous lançons notre `npm install` depuis notre container. Or, celui-ci n'a pas connaissance de la clef SSH nous permettant d'y accéder. En effet, cette clef est uniquement disponible sur notre host.

La solution la plus simple serait de lier notre dossier `~/.ssh` (où se situent nos clefs) dans notre container. Cependant, je ne suis pas du tout à l'aise à l'idée de partager ma clef personnelle avec un container, même s'il est privé. Heureusement : une meilleure solution existe : le transfert d'agent SSH. Cette technique permet de transférer au container les autorisations reçues grâce à une clef de l'hôte.

Afin de transférer notre agent SSH, il nous suffit de lier l'emplacement de notre socket SSH de notre hôte vers notre container. Cet emplacement est fourni à travers la variable d'environnement `SSH_AUTH_SOCK` (par exemple : `./run/user/1000/keyring/ssh`)

Ce partage se résume donc à modifier notre configuration :


```
services:
  node:
    volumes:
      - $(SSH_AUTH_SOCKET):/ssh_agent
    environment:
      - SSH_AUTH_SOCKET=/ssh_agent
```

Avant de tester directement l'installation de nos dépendances, testons tout d'abord notre connexion à GitHub :

```
docker-compose run --rm --no-deps node bash -ci 'ssh -T git@github.com'
```

Le container refuse de se connecter et nous retourne le message d'erreur suivant :

```
No user exists for uid 1000
```

Créer un utilisateur lambda pour se connecter en SSH

En effet, afin de résoudre nos problèmes de droits sur nos fichiers, nous avons utilisé un identifiant de 1000 pour notre utilisateur. Or, aucun utilisateur ayant cet identifiant n'existe sur notre container. Et SSH n'apprécie pas cela.

La solution est de créer un utilisateur ayant cet identifiant. Néanmoins, pour ce faire, il nous faut les droits *root*. Nous ne pouvons donc pas utiliser *docker-compose* puisqu'il associe automatiquement l'utilisateur courant à l'id 1000, qui n'est donc pas *root*. Rien ne nous empêche cependant de réutiliser *docker* directement :

```
docker run --rm -it \
-v `pwd`:/app \
-v $(SSH_AUTH_SOCKET):/ssh_agent \
-e SSH_AUTH_SOCKET=/ssh_agent \
-e USER_ID=$USER_ID \
node bash -ci " \
  useradd -u $USER_ID dummy && \
  mkdir -p /home/dummy/.ssh && \
  chown -R dummy /home/dummy && \
  su dummy -c 'ssh -T git@github.com' \
"
```

Cette commande peut paraître intimidante au premier abord. Néanmoins, il ne s'agit "que" de l'exécution de quelques lignes Bash. Les cinq premières lignes permettent la création d'un container temporaire. Nous lui transférons la variable d'environnement correspondant à notre UID (-e USER_ID), et y activons le transfert d'agent SSH.

Quant à la commande Bash exécutée, dans l'ordre, elle crée un utilisateur *dummy* correspondant à notre UID, elle crée un dossier *.ssh* dans son home, lui attribue les droits correspondants, et exécute notre test de connexion en utilisant l'utilisateur *dummy* (via *switch user*, ou *su*).

Si nous exécutons la commande précédente, GitHub nous retourne une nouvelle erreur :

```
Host key verification failed.
```

Désactiver la vérification SSH pour GitHub

Cette dernière erreur n'est pas problématique. Il nous suffit de désactiver la vérification de l'hôte pour le domaine *github.com*.

```
docker run --rm -it \
-v `pwd`:/app \
-v $(SSH_AUTH_SOCKET):/ssh_agent \
-e SSH_AUTH_SOCKET=/ssh_agent \
-e USER_ID=$USER_ID \
node bash -ci " \
[...]\
cp -f /app/docker/node/ssh_config /home/dummy/.ssh/config && \
su dummy -c 'ssh -T git@github.com' \
"
```

Seule différence par rapport au script précédent : la copie d'un fichier *ssh_config* vers le fichier *.ssh/config* du container. Ce fichier contient les lignes suivantes :

```
Host github.com
  StrictHostKeyChecking no
```

Cela permet de désactiver la vérification du host distant pour tous les domaines en *github.com*.

Si nous relançons une nouvelle fois notre commande de test, nous pouvons alors nous assurer que cela fonctionne :

```
Hi jpetitcolas! You've successfully authenticated, but GitHub does not provide shell access.
```

Installation d'un package npm privé GitHub depuis Docker

La connexion à GitHub est à présent fonctionnelle. Néanmoins, nous n'avons fait qu'un test de connexion, et nous n'avons pas encore essayé un *npm install*. Si nous nous contentons de remplacer notre commande *ssh -T* par *npm install*, nous aurions une erreur indiquant que le dossier *~/npm*, requis par *npm*, n'existe pas. Qu'à cela ne tienne, il nous suffit de le créer avec les permissions correctes.

Ainsi, notre commande devient :

```
docker run --rm -it \
-v `pwd`:/app \
-v $(SSH_AUTH_SOCKET):/ssh_agent \
-e SSH_AUTH_SOCKET=/ssh_agent \
-e USER_ID=$USER_ID \
node bash -ci " \
  useradd -u $USER_ID dummy && \
  mkdir -p /home/dummy/.ssh /home/dummy/.npm && \
  chown -R dummy /home/dummy && \
  cat /app/docker/node/ssh_config > /home/dummy/.ssh/config && \
  su dummy -c 'npm install' \
"
```

Et voilà ! Nous avons réussi à faire un "simple" npm install, en étant capables de cloner des repositories privés, et en conservant les bons droits d'installation.

Top 10 des Best Practices pour un développement Web plus Green



Guillaume Renaudin
Architecte SI, Web, Cloud



Nous commençons juste à en prendre conscience de manière globale : la gestion de l'énergie, du besoin à l'utilisation, est l'un des enjeux majeurs de notre siècle. Et Internet n'échappe pas à ce défi. Avec un poids moyen de page multiplié par plus de 100 en 20 ans passant ainsi de 14 ko en 1995 à plus de 2,3 Mo en 2016. En comparaison, c'est autant que Doom sorti en 1993 ! (Pour être plus précis c'est autant que le binaire d'installation de la version shareware du fameux FPS). On estime qu'en 2030 les besoins en énergie pour faire fonctionner Internet seront équivalents aux dépenses mondiales de 2008 tous secteurs confondus (...) !

Les initiatives fleurissent un peu partout dans le secteur des applications numériques et des sites Web :

- Des hackathons orientés "Green Code" pour optimiser l'efficacité d'une application ou pour trouver des nouveaux usages, par exemple, au sein des Smart Cities ;
- Des associations telles que le club Green IT regroupant des entreprises du Numérique qui mettent en commun leurs compétences et expériences dans ce domaine pour identifier et formuler les bonnes pratiques ainsi que les normes qui en découlent ;
- Des projets d'entreprise tels que celui de la banque de Fribourg qui a donné lieu à l'ouvrage "Eco-conception Web/les 115 bonnes pratiques".

Cet article est un résumé de ce référentiel en 10 bonnes pratiques essentielles. A qui s'adresse-t-il ? A nous, développeurs de sites Web, d'applications mobiles ou encore de services applicatifs online ! Quel rôle pouvons-nous jouer dans cette recherche de l'efficacité des logiciels et des outils numériques ?

C'est ce que je souhaite vous montrer à travers ces 10 conseils applicables de la phase de conception jusqu'à l'hébergement ; avant, pendant et après la phase de développement. Nous allons nous concentrer sur les plus marquantes afin qu'elles puissent éventuellement vous donner des idées pour les appliquer et les adapter à votre contexte. Certaines font déjà partie de vos pratiques car elles permettent également d'améliorer ou de moins dégrader les performances.

Attention cependant, car avec la recherche de l'efficacité énergétique, vous serez parfois amenés à faire des choix entre efficacité et sécurité absolue, performance maximum ou encore disponibilité continue du service. Tout est ici une question de pertinence dans un contexte d'utilisation donné. L'efficacité est l'efficacité d'utilisation des ressources pour parvenir au résultat attendu.

Conception et Design

Les 3 premières recommandations sont générales et s'appliquent dans la phase précédant le développement : la conception. Aujourd'hui, environ 80% des gains escomptés dans les consommations énergétiques seront la conséquence d'efforts réalisés pendant les phases amont et aval du développement.

#BestPractice 1 : ne développez et ne maintenez que les fonctionnalités essentielles

Partons d'un constat intéressant : presque 50% des fonctionnalités développées ne sont pas utilisées. Il faut donc se demander quel est l'impact de ce superflu : données stockées et code exécuté pour l'alimentation sont autant de sources de consommation d'espace de stockage de RAM et de CPU qui ont une empreinte énergétique inutile. Et si l'argument énergétique n'est pas suffisant, l'argument économique devrait permettre de convaincre la majorité d'entre nous car un code en production est une source de bugs ou de maintenance évolutive qui a un coût non négligeable. Si vous êtes en capacité d'identifier le code associé à une fonctionnalité non nécessaire : débarrassez-vous en maintenant !

Pour connaître les fonctionnalités qui intéressent vos utilisateurs, mettez en place une remontée d'indicateurs métier pertinents grâce à des outils. Et s'ils ne sont pas assez fins ou adaptés à votre contexte d'usage, pourquoi ne pas vous essayer au développement d'aspect par exemple ?

#BestPractice 2 : choisissez judicieusement vos frameworks et bibliothèques

Certains frameworks, bibliothèques ou API sont connus pour être très gourmands en ressources. Par exemple, pendant longtemps en Java, le parsing XML n'était réalisable qu'à travers de deux API : JDom et SAX. La première était réputée très consommatrice en mémoire car l'ensemble du fichier XML y était chargé sous forme d'arbre (DOM). Cependant elle offrait des méthodes de parcours beaucoup plus puissantes que son homologue plus économe, SAX qui reposait sur une notion de streaming. Finalement est apparue la JSR 173 définissant l'API StAX qui regroupe le meilleur des deux précédentes, réconciliant ainsi l'utilisabilité et les performances. Mon conseil : faites le tour de vos communautés, réseaux sociaux, blogs favoris pour effectuer un choix éclairé. N'hésitez pas également à partager vos trouvailles ou expériences et à alimenter les pull requests, issues, etc. des solutions que vous utilisez, Open Source ou non.

Développement Client :

C'est avec la partie "Client" et plus particulièrement le navigateur Web que nous abordons les bonnes pratiques orientées Dev avec plusieurs domaines : les CSS, le JS et le DOM.

#BestPractice 3 : codez avec style

Le standard CSS existe depuis 20 ans et ne cesse d'évoluer pour nous apporter toujours plus de nouvelles fonctionnalités dans la définition du style de nos sites. La dernière version CSS3 apporte un lot important d'améliorations que nous avons l'habitude de coder à grand coup d'images, JS et autres hacks : transparence, bords arrondis, positionnements ou encore transitions et animations. Leur rendu est, grâce à cette standardisation, optimisé par les moteurs de rendu des navigateurs, il est donc important de toujours préférer une propriété CSS à :

Une image pour, par exemple, dessiner des cadres :

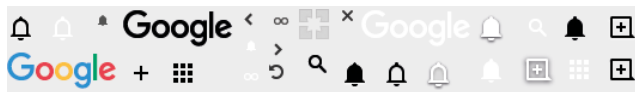
```
#cadre {
  border-radius: 10px;
}
```

Ou du Javascript pour les animations

Les spritesheets

Le CSS via la notion de spritesheet permet de limiter le nombre d'allers-et-retours nécessaires pour télécharger des images, car l'idée est d'avoir une seule grande image qui regroupe toutes les images de notre site.

Voici une spritesheet issue de www.google.fr:



Pour afficher le titre **Google**, il suffira d'écrire :

```
<div id="title"></div>
#title{
  background:url("google_sprites.png") 0 0 no-repeat;
  background-position: 0 100%;
  background-size : 500%;
  height: 10%;
  width: 10%;
  padding:0;
}
```

On charge donc l'image complète et on ajuste la position de l'arrière-plan et la taille du conteneur pour la limiter à la zone que l'on souhaite afficher. Notez au passage que c'est l'image d'arrière-plan que l'on décale en position verticale... Les ajustements en pourcentage permettent quant à eux de rendre le tout "responsive".

Ce procédé a également l'avantage de limiter l'effet de clignotement lors des animations sur un `mouseover` car on ne change pas d'image mais l'on effectue une transformation CSS. Les sprites sont aussi souvent utilisés par les développeurs de Jeux Vidéo.

Les sélecteurs, déclarations ...

Quelques autres bonnes pratiques simples et efficaces avec les CSS : Utilisez des sélecteurs qui ciblent directement une classe ou un id plutôt qu'une hiérarchie ou une sélection par attributs.

On pourrait choisir une colonne d'entête par :

```
table[title="multicolor"] > thead > tr > th { ... }
```

Mais la méthode la plus efficace serait de créer une classe qui cible directement ces éléments :

```
.table-multicolor-header { ... }
```

Groupez les déclarations CSS identiques ou communes

Par exemple :

```
h1 {
  color : blue;
  font-family: Verdana;
}
h2 {
  color : blue;
  font-family: Verdana;
}
h3 {
  color : blue;
  font-family: Verdana;
}
```

Peut être remplacé par :

```
h1, h2, h3 {
  color : blue;
  font-family: Verdana;
}
```

Utilisez les notations CSS abrégées

Ainsi :

```
{
  margin-top : 10px;
  margin-right : 20px;
  margin-bottom : 30px;
  margin-left : 40px;
}
```

Sera simplifié par :

```
{ margin : 10px 20px 30px 40px; }
```

Modifiez les propriétés CSS en une seule fois en groupant les propriétés en classes en modifiant dynamiquement la classe affectée, cela évitera les activités de repaint/reflow du navigateur. Préférez :

```
error-highlight {
  color : red;
  font-weight: bold;
}

$.addClass('error-highlight');
```

à :

```
$.css('color','red');
$.css('font-weight', 'bold');
```

#BestPractice 4 : dynamisez efficacement

Le Javascript subit beaucoup d'évolutions tant au niveau du langage que de son interprétation par les navigateurs. Avec l'apparition d'ECMAScript 6, il devient un vrai langage objet et beaucoup de constats sur l'efficacité actuelle du langage pourraient être amenés à être revus dans les années à venir. Néanmoins, il reste quelques règles qui ne seront probablement jamais complètement obsolètes :

- Validez votre code avec JSLint car une mauvaise syntaxe est souvent source d'approximation ou de mauvaise interprétation par le navigateur, ce qui induit des cycles CPUs inutiles pour l'exécution de la page ;

- Privilégiez les fonctions anonymes si cela ne nuit pas à la maintenabilité ou si l'efficacité revêt une priorité plus élevée. L'interpréteur économisera la résolution du nom de la fonction ;
- Pour les appels des fonctions `setTimeout()` et `setInterval()` utilisez des fonctions plutôt que des chaînes de caractères. D'une manière générale, l'utilisation d'éval a un coût non négligeable en termes de consommation CPU de par l'inspection qu'elle nécessite.

#BestPractice 5 : avec prudence le DOM tu manipuleras

Le DOM est un outil très puissant qui facilite la création et la maintenance de pages complexes et dynamiques. Il est nécessaire d'observer quelques règles de prudence dès lors qu'on le manipule directement :

- Mettez en cache les objets souvent accédés, cela permettra des opérations coûteuses de parcours de l'arbre à chaque besoin de lire ou modifier ces éléments :

```
menu = document.getElementById('menu');
menu.property1 = 'foo';
menu.property2 = 'bar';
```

- Masquez les éléments du DOM lors de leur modification, cela vous permettra de limiter le nombre de repaint/reflow. Pour cela vous pourrez utiliser un `display : none`, effectuer vos modifications de style sur l'élément ciblé puis réappliquer un `display : block` par exemple. Il faudra bien sûr que le masquage ne génère pas d'artefacts visuels trop importants.
- Utilisez la délégation d'évènement à un nœud parent *central* pour éviter de multiplier le nombre de listeners actifs sur certains événements. C'est le nœud parent, qui se chargera de déléguer les changements d'état aux descendants.

Développement Serveur :

Dans cette seconde partie, nous nous concentrons sur les optimisations à apporter côté serveur : Web, applicatif ou bases de données. Ces bonnes pratiques ne sont que quelques extraits, car bien des efforts ont déjà été menés de ce côté, souvent tirés par la volonté d'améliorer les performances.

#BestPractice 6 : optimisez l'accès aux ressources statiques

La première étape consiste à externaliser l'ensemble du CSS ou du Javascript dans des fichiers dédiés. Cela permettra notamment au serveur Web puis au navigateur mais aussi aux différents proxys de mettre en cache ces fichiers et ainsi de préserver la bande passante des réseaux. En second lieu, minifier et compresser ces mêmes fichiers. Cela permettra également de préserver la bande passante lors des premiers téléchargeurs ou à l'expiration des caches.

Enfin la dernière étape relève d'une recommandation aussi valable dans le développement d'outils ou de bibliothèques utilitaires : modularisez les différentes feuilles de styles ou fichiers de déclarations de vos fonctions JS. Ainsi, seuls les fichiers nécessaires à la page ou à l'application courante seront chargés. Un bon framework disposera toujours d'une partie *core* puis de modules ou plugins additionnels.

#BestPractice 7 : allégez les contenus multimédias

Cette bonne pratique cible principalement les CMS ou les sites de publication et diffusion de contenu tels que les forums et les blogs. Force est de constater que l'apparition de la haute définition qu'elle soit visuelle ou auditive a fortement impacté la volumétrie des échanges sur Internet. Il

est donc toujours intéressant de se demander si le matériel ciblé aura toujours besoin d'une définition 1080p pour une vidéo ou plus de 320 Kbps pour un mp3. Voici 3 conseils pour rendre la diffusion de contenu plus efficiente :

- Encodez vos images/vidéos/sons avant leur diffusion dans votre CMS ou votre site, ne le faites pas de manière dynamique depuis votre site ;
- Publiez différentes versions d'un même contenu afin de proposer une version parfaitement adaptée au contexte de visualisation ou d'écoute de votre utilisateur (matériel, écran, réseau, bande passante, environnement). Vous pourrez par exemple vous baser sur les media queries HTML5/CSS3 pour cibler le niveau de définition ;
- Optimisez également le contenu publié par les utilisateurs : soit par des quotas sur les tailles d'upload, soit par des outils de compression à l'envoi ou encore au moment de l'enregistrement en base de données avec le recours à des outils tel que [ImageMagic](#) pour les images.

#BestPractice 8 : réduisez l'empreinte de vos services métier

Un grand nombre d'optimisations ont déjà lieu dans le cadre du développement des services métier et l'accès aux bases de données. Il s'agit là uniquement de quelques rappels :

- Que ce soit dans les modèles de domaines ou dans la base de données, choisissez des formats de données adaptés :
 - Choisir un entier sur 4 pour stocker l'effectif d'une salle de classe est inadéquat quand on sait que l'on peut alors aller jusqu'à $2^{31}-1$ (2 147 483 647) ;
 - Faire une recherche par numéro de sécurité sociale avec une chaîne de caractère à la place d'un nombre est bien moins performant et plus coûteux en énergie (parcours d'un arbre a priori plus long).
- N'effectuez pas de requêtes à la base de données à l'intérieur d'une itération. Préférez plutôt un regroupement de la recherche au sein d'une même requête suivie d'un parcours applicatif de la liste de résultat. L'autre solution consiste à faire confiance à son ORM en ayant adapté les niveaux de cache à votre contexte.
- Cherchez les astuces des langages, que ce soit C, PHP, Java, Javascript, etc. Certaines règles sont simples à automatiser et à appliquer. Par exemple :
 - Ne faites pas appel à des fonctions dans la déclaration d'une boucle `for`.
 - N'écrivez pas :

```
for(int i=0; i < maliste.size(); i++)
```

- Mais plutôt :

```
int tailleliste = maliste.size();
for(int i=0; i < tailleliste; i++)
```

- Remplacez les `i++ ($i++)` par `++i (++$i)`
- Dans beaucoup de langages, la forme `i++` créera une variable temporaire de plus que `++i`

Hébergement et Infrastructure :

L'un des derniers axes d'amélioration de la consommation énergétique d'un site est bien sûr tout ce qui concerne l'hébergement et l'architecture de déploiement.

#BestPractice 9 : trouvez un hébergeur Green

Aujourd'hui, il n'existe malheureusement pas de label pour normaliser ce qu'est un hébergeur Green. Il faudra donc se fier aux critères présentés

par des articles tels que celui de GreenIt.fr [Quels critères pour identifier un hébergeur « vert » ?](#) ou bien encore parcourir les listes proposées telles que celle d'Ekopedia.

Mais si l'on se concentre sur l'un des critères essentiels qui est l'efficacité énergétique du matériel proposé, les solutions BareMetal à base d'ARM proposées par [Scaleway](#) sont très attractives. Rappelons au passage que l'un des gros avantages de la plateforme ARM par rapport à Intel est sa plus faible consommation énergétique. De plus, elles sont aujourd'hui quasiment imbattables économiquement parlant si vous souhaitez une solution BareMetal.

#BestPractice 10 : virtualisez & conteneurisez

L'architecture de déploiement joue un rôle important dans la consommation énergétique d'un site Web ou d'une application en ligne.

On pourra donc profiter de la virtualisation ou mieux encore de la conteneurisation proposée en standard par la majorité des fournisseurs de Cloud public pour limiter l'empreinte énergétique de nos sites et adapter leur consommation aux besoins de traitement des requêtes. Fini le serveur utilisé à seulement 20% de ses capacités CPU et RAM mais dont la consommation énergétique stagne à plus de 50% même en période d'inactivité ; il est désormais possible d'allumer ou d'éteindre nos "serveurs" à la demande.

#BestPractice 11 : pensez aux CDN

Si vous avez bien suivi, le nombre de notre Top 10 est atteint et même dépassé ! Je souhaitais tout de même vous faire part de ce dernier conseil car même si non prioritaire, il est intéressant de le connaître aussi bien

du point de vue technique que philosophique.

Il s'agit du Content Delivery Network qui est un réseau spécialisé dans la diffusion de contenu. Il est constitué de nos serveurs Web ainsi que de nœuds secondaires dont le rôle est de faire office de cache pour servir nos ressources à des clients. L'idée est que les requêtes d'obtention de ressources statiques d'un client seront routées vers le nœud du CDN géographiquement le plus proche permettant d'améliorer les performances mais également limitant les sollicitations réseaux. Par conséquent, l'empreinte énergétique de la requête est moindre.

On peut effectuer le parallèle avec des initiatives actuelles qui visent à limiter le transport des marchandises dans le monde et privilégier les circuits de distribution les plus courts, dès que cela est envisageable.

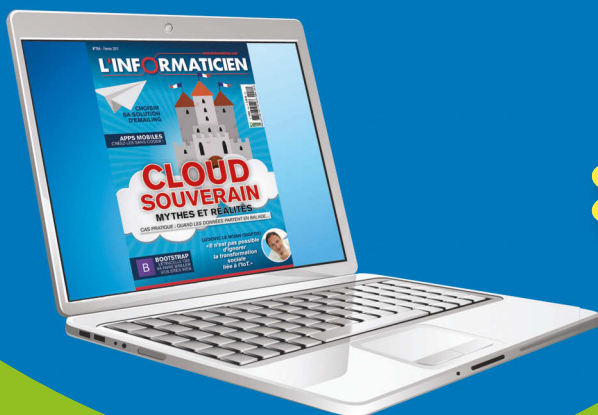
CONCLUSION

Pour résumer ce Top 10, un développement "Green" repose sur 3 facteurs : la recherche de simplicité, la pertinence des choix et solutions, et l'automatisation pour l'application de règles de base. Il ne faut pas chercher à sur-optimiser au risque de se retrouver avec l'effet contraire, mais à appliquer des concepts simples et efficaces, à rester informé et s'adapter aux nouvelles bonnes pratiques.

Chaque acteur de la conception au déploiement a un rôle à jouer dans l'amélioration de l'efficacité de nos systèmes pour que demain, nous puissions bénéficier de toujours plus de services et d'outils innovants dans un monde où les ressources pour les exécuter sont limitées. Finalement, nos sites Web ont aujourd'hui un poids 100 fois plus important qu'il y a 20 ans. Mais est-ce que pour autant nous commandons des billets de train, d'avion ou d'hôtel 100 fois plus vite qu'à cette période ? •

L'INFORMATICIEN + PROGRAMMEZ

versions numériques



2 magazines mensuels
22 parutions / an
+ accès aux archives PDF

PRIX NORMAL POUR UN AN : 69 €
POUR VOUS : 49 € SEULEMENT*

Souscription sur www.programmez.com

Alexa, l'assistant vocal d'Amazon vu de près !



Ouadie Lahdioui
Soat

Consultant Architect ChatBot.
Ouadie accompagne nos clients
grands comptes dans leur
transformation digitale. Notamment
dans la mise en place des produits
logiciel innovants genre les agents
conversationnels @lahdiouiouadie

SOAT

Depuis quelques mois, nous assistons peu à peu à la naissance d'une nouvelle ère en termes d'intelligence artificielle et d'expérience utilisateur. Nous vivons la renaissance d'une tendance qui a révolutionné la manière d'interagir avec les machines au quotidien. On parle de plus en plus des agents conversationnels, des assistants virtuels ou encore des ChatBots.

Ce phénomène a pour vocation de fournir une expérience utilisateur conversationnelle basée sur la voix permettant à la fois un accès rapide à l'information et offre une relation de confiance en mode one-to-one entre les entreprises et ses clients. Dans cet article, on va mettre en lumière le développement et le déploiement d'une application pour l'assistant vocal d'Amazon 'Alexa' qui s'étioffe progressivement depuis peu de temps. Qu'est-ce qu'un ChatBot ? Etes-vous convaincu par ses promesses et voyez-vous clairement ses domaines d'utilisations ? Le rapport avec Alexa ?

Un Chatbot : qu'est-ce que c'est ?

Littéralement, un ChatBot est la contraction de « Chat » et de « Robot », et digitalement c'est un logiciel doté d'intelligence artificielle capable de réagir aux demandes de ses utilisateurs en analysant le texte de la conversation ou la voix provenant d'un canal de communication qui pourrait être une application de messagerie instantanée ou bien un appareil physique.

La plus value des ChatBots qui sont aussi appelés des assistants virtuels, ou encore des robots conversationnels, est de fournir une nouvelle expérience utilisateur fluide basée sur des dialogues humains-robots. Les agents conversationnels commencent ainsi à séduire les marques et à s'infiltrer dans plusieurs domaines.

Le domaine bancaire

Les Fintechs dégagent encore une fois en premier et donnent vie à la banque conversationnelle. C'est le cas pour plusieurs banques qui ont créé des services de gestion financière et permis d'obtenir le solde des comptes, les dernières transactions et les virements effectués en discutant directement avec un chatbot. Parmi les banques qui ont sauté le pas, je citerai le projet Skill de l'américain "Capital One" pour interroger ses comptes en mode vocal, et le projet AmexBot d'American Express sur Facebook Messenger. Outre les établissements bancaires et les Fintechs, plusieurs porteurs de projets innovants dans le domaine de la banque conversationnelle entrent en jeu, comme Finn.ai et Trim, qui sont de véritables coaches financiers pour accompagner le client au quotidien dans la gestion de son argent.

Le domaine du commerce

Grâce à l'évolution des messageries instantanées, qui sont devenues des intermédiaires privilégiés entre les marques et les consommateurs, le commerce conversationnel en mode one-to-one existe en Asie depuis plusieurs années. Avec ce pattern de commerce conversationnel, les marques peuvent créer un lien fort avec leurs consommateurs et établir une relation de proximité avec eux. Puisque les applications de

messagerie vont maintenant bien au-delà des échanges de texte et d'images, et offrent des composants graphiques très complets, le tunnel d'achat d'un site e-commerce classique devient de plus en plus simple et efficace, et une simple affirmation écrite ou verbale pourrait même suffire à effectuer l'opération de paiement !

Les chatbots existent depuis les années 60

Les chatbots existent depuis longtemps. Si l'on en parle autant et qu'ils frappent fort cette année, c'est parce que les géants du Web autrement connus sous le nom de GAFAM se sont tous lancés sur le créneau. Le premier programme de type ChatBot a été conçu au milieu des années 1960 par le laboratoire d'intelligence artificielle MIT qui l'a nommé "ELIZA". Ce robot "thérapeute" était programmé pour reformuler les affirmations du patient en question. Cette première expérience fut un tel succès qu'elle donna naissance à une nouvelle expression pour désigner la tendance à assimiler de manière inconsciente le comportement d'un ordinateur à celui d'un être humain : L'effet ELIZA.

En avril dernier, sous l'impulsion de la messagerie chinoise WeChat, qui avait pris l'initiative de lancer un système de chatbots, Facebook annonçait, lors de sa conférence F8, l'arrivée des chatbots sur Messenger. Microsoft lançait également un framework de développement de bots à disposition des développeurs pour qu'ils créent leurs propres chatbots, et pour en accroître la robustesse, il n'a pas tardé à sortir une suite de services d'intelligence baptisée "Cognitive Services". Plus récemment Oracle rejoint la vague comme a annoncé le CTO Larry Ellison lors de la conférence OpenWorld 2016.

Les 3 grands types de ChatBots à connaître

Le mot 'ChatBot' est abstrait et peut faire référence à des types variés, pour cela, il faut bien connaître les différentes catégories des ChatBots qui se différencient selon la nature de la communication, la manière de demander l'information et le lieu de résidence du bot.

1 • Le ChatBot textuel : ce type de ChatBot réside dans les logiciels de messagerie instantanée où l'accès à l'information se fait par le biais des commandes textuelles. Parmi les logiciels de messagerie qui sont ouverts à mettre en place des ChatBots on cite : FaceBook, Slack, Skype, Kik, Viber, ...

2 • Le ChatBot vocal : à l'instar de son frère jumeau, ce type de ChatBot se base sur des commandes vocales pour accéder à l'information, car il est parfois plus facile de demander l'information à

l'oral que de l'écrire ; imaginez par exemple que vous vouliez réserver un restaurant ou consulter les dernières transactions bancaires en conduisant. Le ChatBot vocal peut aussi résider dans les logiciels de messagerie instantanée comme il peut prendre la forme d'une application à part entière.

3 • Les assistants virtuels : se sont des interfaces vocales disponibles sur smartphone ou intégrées nativement dans le système d'exploitation ; la plupart d'entre eux sont semi-ouverts dans la mesure où seuls les partenaires peuvent greffer leur service.

Les assistants virtuels peuvent être intégrés dans les applications de messagerie ou dans nos maisons via des smart hubs. C'est le cas des assistants vocaux d'Amazon : Echo, Dot et Tap.

Alexa, l'assistant vocal d'Amazon embarqué dans les enceintes d'Amazon

Amazon a lancé 3 nouvelles enceintes connectées interactives capables de répondre aux commandes vocales qui se positionnent comme des véritables assistants pour toute la famille : Amazon Echo, Amazon Tap et Amazon Dot. [1]

L'enceinte Amazon Echo, est un gros cylindre haut de 23,5 cm constitué d'un puissant haut-parleur et de sept micros intégrés pour bien capter les commandes vocales : [2]



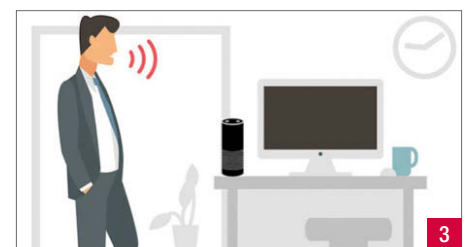
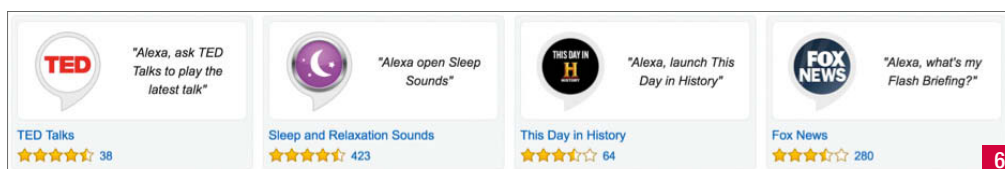
La deuxième enceinte (Echo Tap) est un périphérique portable tandis que la troisième est une version miniature de l'Echo (Amazon Dot). Tous ces assistants vocaux sont capables de parler à leurs utilisateurs, de leur donner des informations et même d'anticiper leurs besoins. Ils offrent également une couverture de son optimale à 360° afin qu'ils puissent aisément capter les commandes dans toute la pièce ; l'utilisateur est alors entendu de n'importe quel coin, il suffit seulement qu'il prononce le mot magique 'Alexa' ! [3]

Alexa, qui êtes-vous ?

Une fois l'enceinte Amazon branchée, il suffit à l'utilisateur de dire 'Alexa' pour qu'il soit relié directement à iTunes, Spotify ou autre. Par ce fait, l'enceinte est allumée et se connectera automatiquement au service de reconnaissance vocale pour répondre à la demande instantanément d'une manière à la fois rapide et simple. Alexa représente donc le cerveau du matériel, c'est lui qui va chercher les informations sur Internet, jouer de la musique, télécommander la maison connectée (éclairage, chauffage...) ou encore faire passer une commande en ligne. [4]

À vrai dire, Alexa est le système intelligent embarqué dans les enceintes d'Amazon, c'est à lui d'interpréter la demande de l'utilisateur et de chercher l'information demandée. [5]

Amazon permet aux développeurs de créer des skills, c'est l'équivalent des applications pour iOS ou Android. De nos jours, il dispose désormais plus de 10 000 applications sur le store et ce nombre ne cesse pas d'agrandir. [6]



Pour avoir la liste complète des skills déjà disponibles, rendez-vous sur le site du store Amazon.

Alexa : comment développer ma propre application ?

Amazon ouvre sa plateforme Alexa pour les développeurs pour qu'ils créent leurs propres applications (autrement connue sous le nom Skills). A disposition des développeurs, Amazon offre trois principaux programmes : Amazon Skills Kit, Alexa Voice Service et Alexa Found. [7]

Amazon Found est le programme qui permet de fournir une aide financière jusqu'à 100 millions de dollars à tous les porteurs de projets innovants qui fournissent des nouvelles expériences vocales avec Alexa.

Le deuxième programme est 'Alexa Voice Service (AVS)', cela permet aux entreprises fabriquant des objets connectés, des applications mobiles, et des voitures intelligentes, d'ajouter le support d'Alexa sur leurs produits à condition que le produit dispose d'un microphone et de haut-parleurs. [8]

Le troisième programme est celui qui va nous intéresser le plus, Amazon Alexa Skills Kit (ASK) ; c'est la boîte à outils et le point d'entrée des développeurs pour développer des skills pour Alexa.

Exemple d'application : dans cet exemple, on va créer une application pour le suivi de sinistres afin de fournir à l'utilisateur une nouvelle expérience utilisateur en lui permettant de consulter ses sinistres juste en demandant à son enceinte intelligente. On va également déployer l'application créée dans AWS et la faire tester sur Amazon echo.

1 - La configuration de l'enceinte Amazon Echo

La première étape est de connecter l'enceinte Amazon Echo au réseau WIFI, cela se fait en entrant au site www.amazon.co.uk puis de se connecter avec son compte Amazon : [9]

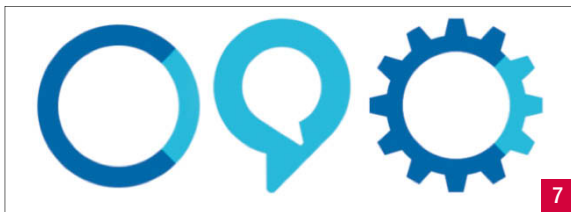
Une fois cela fait, vous verrez l'écran qui vous permettra de choisir le type de votre enceinte, dans mon cas, il s'agit d'une Amazon Echo : [10]

Choisissez votre enceinte puis votre langue : [11]

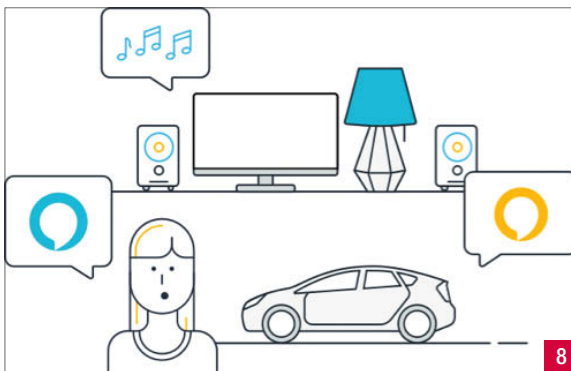
Connectez-vous au même réseau WIFI que votre enceinte et cliquez sur 'connect to Wi-Fi' : [12]

A ce moment-là, Amazon Echo va vous informer qu'elle est prête à être configurée : [13]

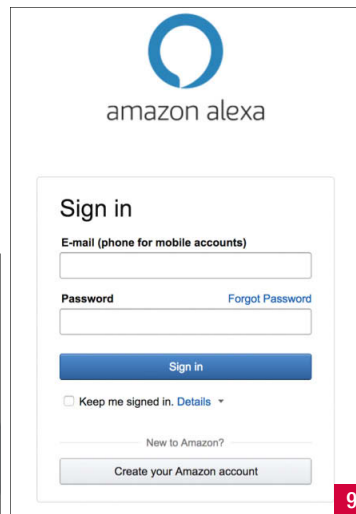
Mais si elle ne le fait pas, il faut seulement cliquer sur le bouton d'action pendant 5 secondes environ : [14]



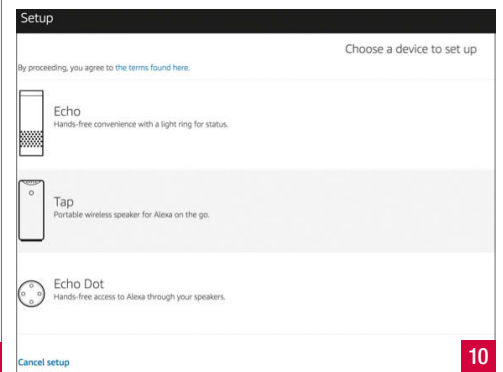
7



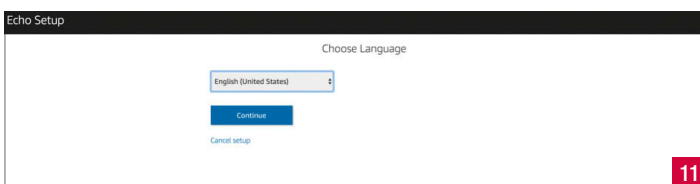
8



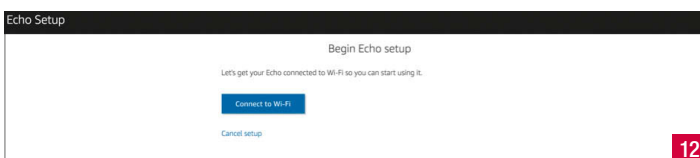
9



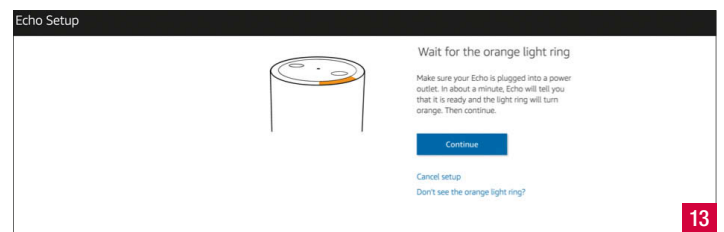
10



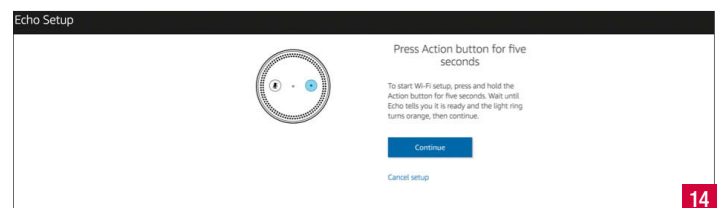
11



12



13



14

Amazon Echo va par la suite créer un réseau baptisé 'Amazon-XXX' : [15]

Votre machine est connectée au nouveau réseau créé par Amazon Echo.

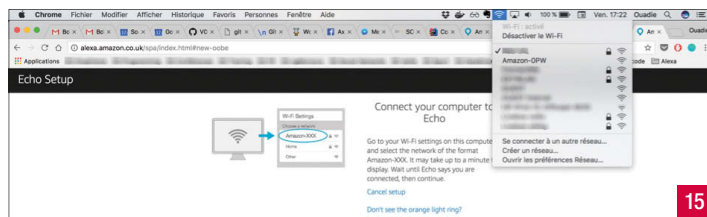
Maintenant l'enceinte vous redemande de choisir le réseau sur lequel vous voulez la connecter, dans mon cas c'est le réseau R&D DIL : [16]

La prise en charge de la nouvelle config prend quelques secondes : [17]

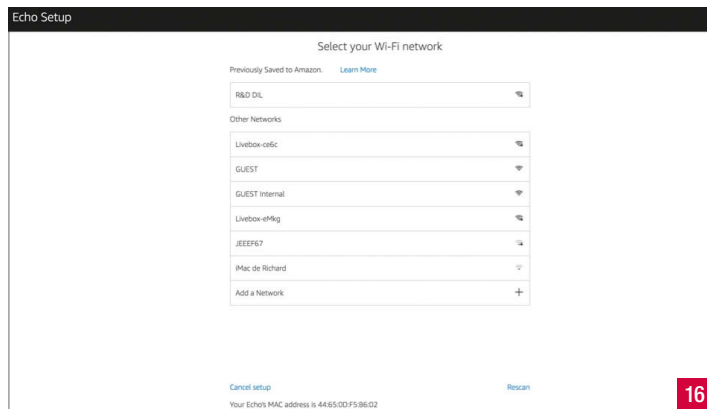
Vers la fin de la procédure de configuration, Amazon Echo demande de vous connecter à nouveau à votre réseau initial (pas le Amazon-XXX) : [18]

Et voilà, votre Amazon Echo est prête à être utilisée.

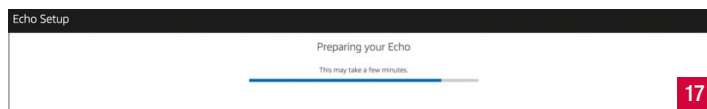
Remarque : si vous essayez de vous connecter à votre enceinte et qu'elle n'est pas accessible pour une raison ou une autre, vous verrez un message indiquant que l'enceinte est hors ligne : [19]



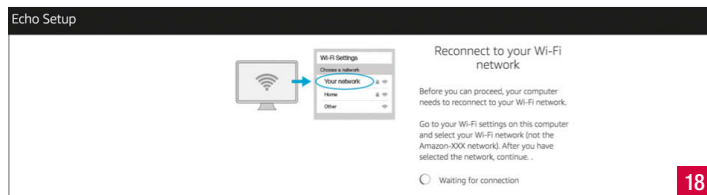
15



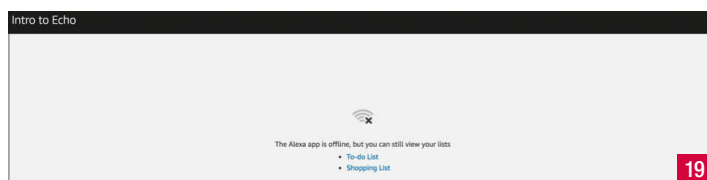
16



17



18



19

Cependant, vous pouvez quand même visualiser la liste de vos applications installées et un historique de vos interactions avec l'enceinte : [20]

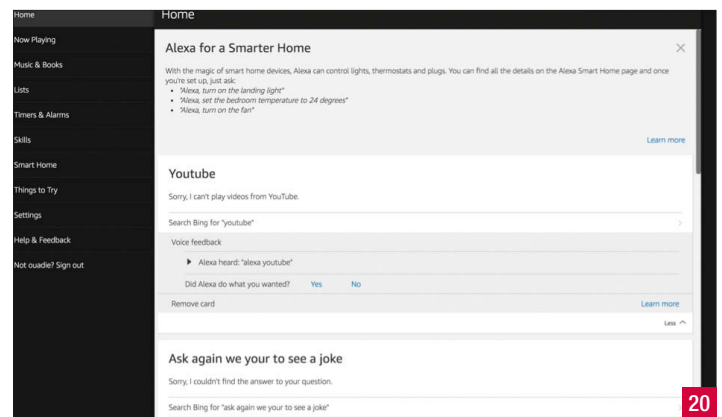
A partir du site dédié à la configuration d'Alexa, vous pouvez aussi accéder au store et télécharger des applications.

2 – Création d'une fonction Lambda

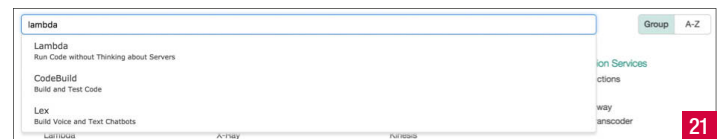
Connectez-vous à la console Amazon Web Services (<https://aws.amazon.com/>) et cherchez le service Amazon Lambda : [21]

Changez la région en US East (N. Virginia) : [22]

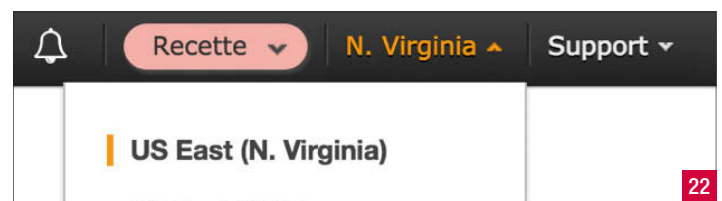
Vous avez la possibilité de créer une fonction éventuellement en vous basant sur un template : [23]



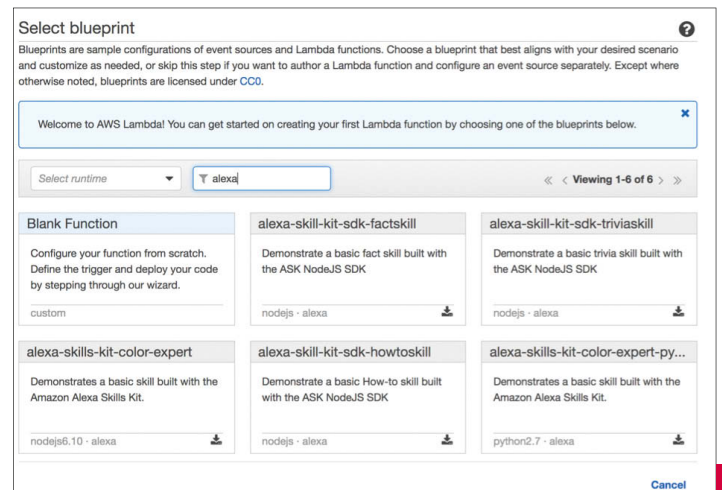
20



21



22



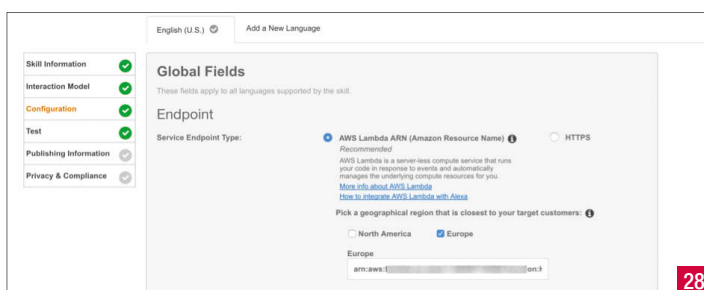
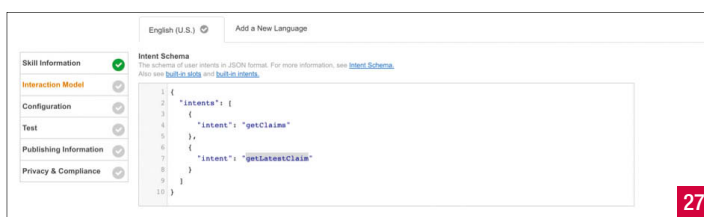
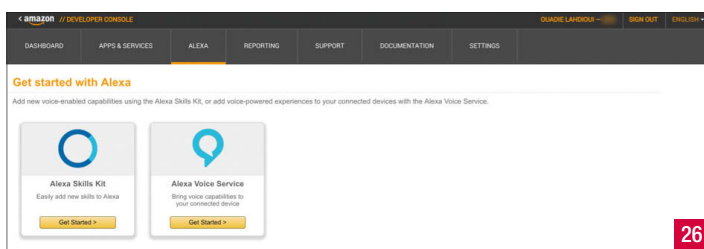
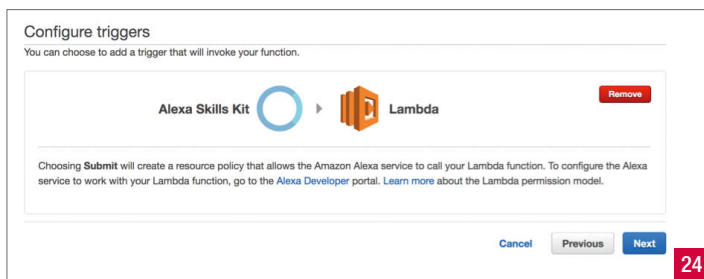
23

Dans notre exemple, on va se baser sur un template de façon à vous illustrer toutes les étapes. Cliquez sur 'Blank Function' puis sélectionnez 'Alexa Skills Kit' : [24]

Choisissez un nom et une description pour votre fonction Lambda : [25]

Sur ce Github (<https://github.com/ouadie-lahdioui/Amazon-Alexa>), je vous ai partagé un exemple de code source de la fonction lambda pour récupérer la liste des sinistres.

Une fois votre fonction Lambda créée, gardez bien le code ARN qu'on va utiliser dans un second temps.



3 – Création du Skill sur le portail du développeur Amazon :

Rendez-vous sur le portail du développeur Amazon

(<https://developer.amazon.com>) et cliquez sur la rubrique Alexa : [26]

Cliquez sur le bouton 'Add New Skill'.

Choisissez un nom à votre application et chaîne d'invocation, dans mon cas, l'application que je suis en train de créer est appelée 'Claims API', et je peux par la suite l'invoquer par le chaîne 'Alexa ask claims bot ...'

Définissez le schéma d'intention qui est un objet de type JSON et qui représente la liste des intentions gérées par l'application.

Dans notre exemple, l'application gère les deux intentions 'getClaims' et 'getLatestClaim' : [27]

Maintenant, il faut définir les chaînes d'invocations que va dire l'utilisateur pour entrer en contact avec notre application, le format de la chaîne d'invocation est le suivant :

[INOM_INTENTION][CHAINE_INVOCATION]

Tapez le code ARN de votre fonction AWS Lambda déjà créée auparavant.

Une fois la création du Skill sur le portail du développeur Amazon faite, vous pouvez lister l'ensemble des skills déjà créés et installés sur votre enceinte intelligente sur le site (www.alexa.amazon.co.uk) : [28]

UN PEU DE CODE...

Nous avons vu toutes les étapes nécessaires à la création et au déploiement d'une application pour l'assistant intelligent Amazon Alexa. Nous avons également déployé une fonction lambda AWS pour pouvoir réagir aux commandes vocales de l'utilisateur du Skill développé. Alors, comprenons bien ce que l'on vient de déployer ! Il est possible de développer un Skill pour Alexa avec plusieurs langages de programmation. Si nous avons choisi pour l'exemple Node.js, c'est parce que sa nature non bloquante et monothreadée est bien adaptée à ce genre d'application. En plus de ça, il s'interface rapidement avec Amazon Lambda. Rappelez-vous qu'au moment de la déclaration du Skill dans le portail des développeurs Amazon, et plus exactement dans la partie modèle d'interaction, nous avons configuré le Skill pour gérer les deux intentions 'getClaims' et 'getLatestClaim' :

```
{
  "intents": [
    {
      "intent": "getClaims"
    },
    {
      "intent": "getLatestClaim"
    }
  ]
}
```

Pour cela, il faut à un certain moment dans notre code source faire le traitement métier nécessaire qui va former la réponse adéquate par intention d'utilisateur, c'est notre fonction Lambda.

Lorsque vous créez une fonction Lambda, vous spécifiez un gestionnaire, qui est une fonction de votre code qu'AWS Lambda peut appeler lorsque le service exécute ce code. Utilisez la structure syntaxique générale suivante lorsque vous créez une fonction de gestionnaire en Node.js.

```
exports.handler = (event, context) => {
  // Code
}
```

Le paramètre 'event' est utilisé pour transmettre les données d'événement au gestionnaire, tandis que 'context' fournit au gestionnaire les informations d'exécution de la fonction.

Le Type des requêtes envoyées par Alexa

Alexa est capable d'envoyer à notre fonction lambda trois types de requêtes :

• LaunchRequest :

Cette événement est envoyé par Alexa lorsque l'utilisateur invoque le Skill sans spécifier l'intention, par exemple :

Utilisateur dit : Alexa, Open Claims Bot

Avec Node.js, on peut réagir à l'événement 'LaunchRequest' comme celui là :

```
exports.handler = (event, context) => {

  switch (event.request.type) {

    case "LaunchRequest" :
      console.log(` LAUNCH REQUEST `)
      context.succeed(
        generateResponse(
          buildSpeechletResponse("Welcome to an Alexa Skill, this is running on a
            deployed lambda function", true),
          {}
        )
      )
      break;

    default:
      context.fail(` INVALID REQUEST TYPE: ${event.request.type} `)

  }

}

buildSpeechletResponse = (outputText, shouldEndSession) => {

  return {
    outputSpeech: {
      type: "PlainText",
      text: outputText
    },
    shouldEndSession: shouldEndSession
  }

}

generateResponse = (speechletResponse, sessionAttributes) => {

  return {
```

• IntentRequest :

Cet événement est déclenché par Alexa lorsque l'utilisateur invoque le Skill en disant une phrase qui corresponde bien avec une intention, par exemple :

Utilisateur dit : Alexa, Ask Claims Bot to show my latest claim

De ce fait, il est possible de faire des traitements métiers par intention dans la fonction lambda :

```
exports.handler = (event, context) => {

  switch (event.request.type) {

    case "IntentRequest":
      console.log(` INTENT REQUEST `)
      switch(event.request.intent.name) {
        case "getClaims":
          context.succeed(
            generateResponse(
              buildSpeechletResponse(` You have 3 open claims ` , true),
              {}
            )
          )
          break;
        case "getLatestClaim":
          context.succeed(
            generateResponse(
              buildSpeechletResponse(` Your latest claim has been open last month ` , true),
              {}
            )
          )
          break;
      }

  }

}
```

• SessionEndedRequest :

Alexa envoie cet événement une fois la session terminée, et cela peut se faire pour plusieurs raisons. La première est le fait que l'utilisateur demande de terminer la session en disant 'Alexa, Exit', la seconde dans le cas où l'utilisateur n'as pas répondu par une réponse compréhensible par Alexa lorsqu'il attend une réponse.

```
exports.handler = (event, context) => {

  switch (event.request.type) {

    case "SessionEndedRequest":
      // Session Ended Request
      console.log(` SESSION ENDED REQUEST `)
      break;

    default:
      context.fail(` INVALID REQUEST TYPE: ${event.request.type} `)

  }

}

buildSpeechletResponse = (outputText, shouldEndSession) => {

  return {
    outputSpeech: {
```


Pierre Dac disait *“Avec de la méthode et de la logique on peut arriver à tout aussi bien qu'à rien.”*

- Principes :

- Nous allons à l'essentiel ;
- L'équipe décide ensemble.
- Outils
 - Des User Stories ;
 - Un tableau Kanban ;
 - Un daily meeting ;
 - Des démos (dès qu'on peut, quand on peut).
- Repères
 - Une vision claire ;
 - De l'envie et du fun !

Nous devons procéder méthodologiquement et réaliser un produit qui fonctionne. Notre robot doit comprendre ce qu'on lui dit, pas uniquement obéir à quelques ordres prédéfinis. Il doit reconnaître le sens des mots et donc les analyser en se basant sur une approche sémantique.

La sémantique est une branche de la linguistique qui est à l'opposé de la branche qui se base sur la syntaxe. Elle étudie les signifiés : ce dont on parle et ce que l'on veut énoncer.

Pourquoi a-t-on besoin de la sémantique dans ce type de projet ? Elle va permettre de comprendre de quoi on parle. Elle permet de démêler les choses compliquées du langage comme les mots composés, les antonymes, les synonymes, etc. Dans le prototype, c'est ce qui va permettre au drone d'interpréter par exemple « tourne, mais pas à droite » ce qui donne en résultat une redirection à gauche grâce à l'interprétation sémantique. C'est cette capacité inédite qui est garante de l'innovation. L'ontologie est un autre outil souvent utilisé par la sémantique. Elle permet de décrire la nature et les propriétés des objets du monde réel à partir de concepts et de relations entre ces concepts. En philosophie, l'ontologie est l'étude de l'être en tant qu'être. En science, c'est l'étude des propriétés générales de ce qui existe. C'est l'une des approches de modélisation qui permet d'inférer des faits à partir d'autres faits et surtout de donner du sens à une syntaxe.

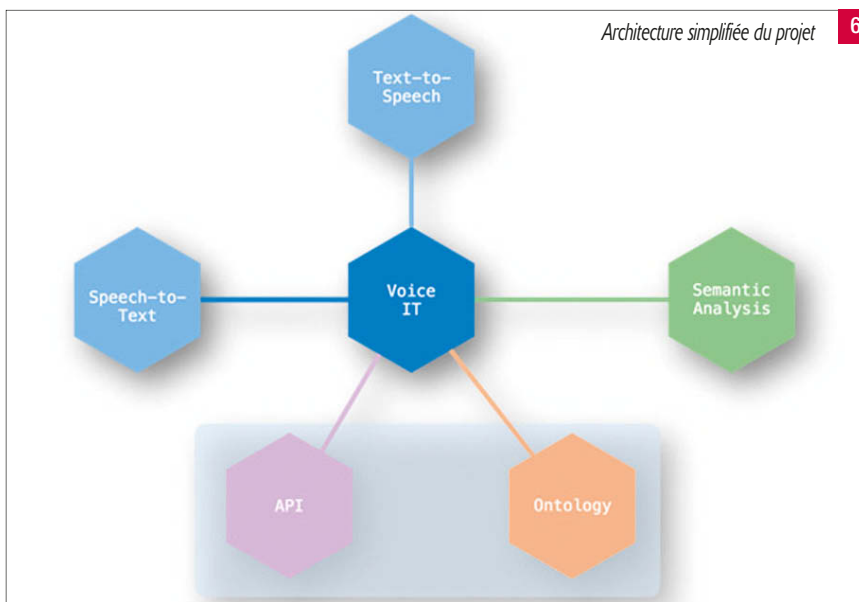
Si nous allons un peu plus loin, une ontologie est un ensemble de concepts reliés entre eux. Ce genre de réseau de relations permet de faire ce qu'on appelle une inférence. C'est tout simplement une déduction à partir de conditions initiales.

[illegible]

Grappe
Sémantique

5

6



tions entre concepts est précieuse pour amener les machines à des comportements riches, proches de ce qu'on appelle l'intelligence. Avec des ontologies très complexes et riches en relations, le modèle serait analogue à un réseau de nœuds capable de générer des réponses à des interrogations sur l'ensemble de bases de connaissances. C'est à l'image du cerveau : plus il est grand et complexe, plus l'être vivant qui s'en sert, est intelligent. À partir d'un certain degré de complexité (nombre d'éléments et de relations) un modèle sémantique devient capable de faire émerger des connaissances inconnues jusqu'alors. [5]

L'idée était d'avoir quelque chose de réutilisable, facile à intégrer dans différents métiers. En séparant les différents concepts, on reste agnostique par rapport au problème à résoudre : la seule chose à modifier pour traiter un nouveau client/API/Service, c'est l'ontologie

et son mapping vers l'API finale.

Procédons étape par étape dans le processus d'interactions entre les différents composants :

- Un dispositif reconnaît la voix et la traduit vers du texte ;
- Notre application le reçoit et l'envoie vers le module du matching sémantique pour en extraire des actions et entités ;
- L'application essaye de matcher ces actions dans l'ontologie ;
- Si elle a reconnu une commande, on l'exécute avec l'API cible ;
- Le programme produit un feed-back utilisateur via text-to-speech ou un écran ;
- Dans le cas où l'application n'a pas compris, on peut commencer une 'conversation' avec l'utilisateur en demandant soit de répéter soit de donner plus de détails, exemple :
 - Utilisateur : "Tournez",
 - Robot : "Je peux tourner à gauche ou à droite, je fais quoi ?",
 - Utilisateur : "à droite !" [6]

7 CMU Sphinx Project by Carnegie Mellon University

Nous avons commencé par explorer des solutions Open Source pour la reconnaissance vocale et l'analyse sémantique fonctionnant avec un Raspberry Pi. Nous avons étudié les outils suivants : CMU (Carnegie Mellon University) Sphinx (Voice-to-Text) et Stanford NLP.

- Caractéristiques de CMU Sphinx : [7]
 - Le Pocketsphinx offre une version prête à utiliser sur Android (avec 20% de taux d'erreur parmi ~ 10 mille mots) ;
 - Il peut fonctionner en Offline ;
 - Le réglage (tuning) du modèle est assez compliqué ;
 - Le support de la langue française n'est pas assez complet, beaucoup moins que le support de l'anglais

Caractéristiques de Stanford NLP [8]



- Le support complet des modules que pour l'anglais, cependant nos commandes seront en français ;
- Stanford NLP n'a pas deux des éléments les plus importants pour nous : les Lemmes (*Lemmas*) et les *Named Entities*.
- Les *Lemmas* permettent d'extraire l'origine d'un mot depuis une conjugaison : en anglais : "was" -> "to be" ;
- Les *Named Entities* permettent de 'taguer' des mots, d'ajouter de la metadata, "yesterday" -> 26-10-2016.[9]

Il existe des outils Cloud permettant d'analyser et d'extraire des actions depuis du texte :

ANNOTATOR	AR	ZH	EN	FR	DE	ES
Tokenize / Segment	✓	✓	✓	✓	✓	✓
Sentence Split	✓	✓	✓	✓	✓	✓
Part of Speech	✓	✓	✓	✓	✓	✓
Lemma			✓			
Named Entities		✓	✓		✓	✓
Constituency Parsing	✓	✓	✓	✓	✓	✓
Dependency Parsing		✓	✓	✓	✓	
Sentiment Analysis			✓			
Mention Detection		✓	✓			
Coreference		✓	✓			
Open IE			✓			

Supports des modules linguistiques

Google, Wolfram sont les plus connus, mais il existe également des services comme api.ai et wit.ai. Tous ces services sont payants, mais l'analyse sémantique est bien au-delà de notre simple P.o.C. (Proof of Concept). Afin de bénéficier d'une analyse sémantique de qualité, il faut prendre en compte un coût d'implémentation assez important.

Choix de la plateforme du PoC : Tablette Android et Reconnaissance vocale.

Comme nous n'avions qu'un temps limité, nous avons repriorisé le backlog et décidé, au vu de ses différents avantages, de se lancer sur la plateforme Android d'une tablette :

- Speech-to-Text : il permet de télécharger les packs multilingues hors-ligne sur Android. Pour notre projet, il était nécessaire de pouvoir reconnaître la voix de manière offline, car le drone est connecté à la tablette via WiFi du drone. Dans la Raspberry, ça serait différent, car il y du WiFi ET de l'Ethernet ;
- Text-to-Speech : Non seulement on peut choisir la langue, mais on peut aussi choisir parmi différents types de voix (féminine/masculine, rapide/lente, etc.) ce qui facilite la personnalisation du produit pour ses utilisateurs (ceci peut être adapté aux capacités de l'utilisateur : enfant, adulte, etc.) et la communication avec ses interlocuteurs (ceci est très utile pour s'adapter au type du produit final : drone, robot, station fixe, application mobile / assistant virtuel, bot, etc.) ;
- Écran + Micro + Haut-Parleurs intégrés ;
- Librairie GAST : encapsule des outils pour utiliser les capacités de détection d'événements sur les capteurs (dans notre cas le son) de la tablette Android. Il contient des exemples de codes et les algorithmes dont vous avez besoin pour utiliser correctement les capteurs pris en charge par Android.

Le Drone

Parrot propose une API très complète avec laquelle on peut contrôler des mouvements,



10 Jumping Sumo Parrot

prendre des photos, transférer de la vidéo en temps réel, réaliser des animations, etc. Pour les modèles de drones plus avancés, l'API a la possibilité de définir un parcours via des coordonnées GPS, d'envoyer et de recevoir de l'audio, etc. De plus, l'API est écrite en C et propose des interfaces en Java, Unix et iOS. [10]

Où est l'innovation ?

Notre prototype utilise une interprétation sémantique et non juste syntaxique. Notre solution sait reconnaître un ensemble de mots et les interpréter dans un ensemble de concepts reconnus dans le domaine métier du client. Nous modélisons donc le champ sémantique de notre client pour que la reconnaissance soit vraiment ouverte et pertinente. Ceci peut être réalisé rapidement en adoptant un modèle sémantique déjà existant ; aujourd'hui il existe des modèles sémantiques génériques ou spécialisés pour plusieurs disciplines / activités (ex, médecine, tourisme, réservation en ligne, etc.) et qui sont même disponibles en libre-service sur des moteurs de recherche comme Google. On peut aussi les adopter et les améliorer pour les adapter. Il est également possible de construire son propre modèle.

L'avantage de cette approche est d'avoir une solution simple, ouverte, performante sans avoir besoin d'une phase d'apprentissage. En ce qui concerne la partie applicative, il n'y a pas de code spécifique au métier. À tout moment, on peut utiliser cette application pour un autre métier : il suffit alors de créer un nouveau lexique et de changer les appels d'API. L'innovation est là !

Perspectives

Nos perspectives d'évolution sur ce projet sont les suivantes :

- Modulariser le code ;
- Analyse sémantique et extraction des actions/paramètres ;
- Mode conversation ;
- Étendre l'ontologie et l'exploiter via Apache Jena ;
- Reconnaissance vocale.
- Intégrer tout ça dans un Raspberry Pi ;
- Utiliser la caméra du drone pour reconnaître des objets, couleurs, etc.

Liens :

<http://blog.palo-it.com/2016/11/09/la-video-du-barcamp-ok-palo-presente-moi-ton-assistant-virtuel-est-en-ligne>

Android O : tour du propriétaire côté développeur

• Sylvain Saurel
sylvain.saurel@gmail.com
Développeur Android
<https://www.ssaurel.com>

Lancé pour la première fois avec Android Nougat l'an passé, le programme Developer Preview aura connu un vif succès auprès des développeurs et aura permis à Google de sortir une version d'Android encore plus stable en Août 2016. Pour sa nouvelle mouture, répondant au nom de code Android O, Google a décidé de relancer son programme et vient tout juste de lancer la première Developer Preview d'Android O à la fin du mois de Mars. Dans cet article, nous vous proposons un tour d'horizon des nouveautés apportées par cette nouvelle version.

L'an passé à pareille époque, Google innovait en lançant un programme Developer Preview pour Android Nougat, la version 2016 de son OS pour appareils mobiles. Force est de constater que cette innovation a fait mouche auprès des développeurs qui ont apprécié pouvoir tester en avant-première les nouvelles fonctionnalités de leur OS favori tout en préparant au mieux leurs applications pour un support optimal d'Android Nougat au lancement officiel en Août 2016. Côté Google, outre le fait de renforcer sa relation avec la communauté des développeurs, ce programme Developer Preview aura permis d'avoir des retours du terrain au plus tôt et de corriger ainsi au plus vite les imperfections d'Android Nougat pour offrir aux utilisateurs finaux un système optimisé. Fort de ce succès, Google a décidé de continuer sur sa lancée en 2017 pour la nouvelle version de son OS avec la sortie fin Mars de la première Developer Preview d'Android O. Cette version initiale est qualifiée d'alpha par Google qui a également publié une roadmap précise pour les mois à venir du programme Developer Preview (figure 1).

Ainsi, Google a prévu de mettre à disposition des développeurs 4 versions de la Developer Preview. La première ayant été sortie fin Mars, il reste désormais 3 versions qui seront mises à disposition de Mai à Juillet. Chaque version devant se rapprocher de la version finale stabilisée qui, elle, doit être rendue publique à la fin de l'été 2017.

Afin de permettre aux développeurs d'utiliser au mieux cette première Developer Preview d'Android O, Google a également mis à disposition tous les outils nécessaires avec notamment la dernière preview d'Android Studio 2.4 permettant d'accéder aux éléments indispensables suivants :

- Le SDK et les outils de la Developer Preview d'Android O ;
- L'image système de l'émulateur de la Developer Preview d'Android O avec un support 32 bits et 64 bits ;
- L'image système de l'émulateur pour Android TV de la Developer Preview d'Android O avec un support 32 bits uniquement ;
- Les bibliothèques de support pour la Developer Preview d'Android O.

Une fois ces éléments récupérés sur le site développeur dédié à Android, vous serez fin prêt pour tester la première Developer Preview d'Android O. Si cette dernière ne nous renseigne guère sur le nom qui sera donné à Android O, elle nous donne en revanche un bon aperçu des nouvelles fonctionnalités du millésime 2017 d'Android.

Mise à jour de votre configuration de Build

Pour commencer, il va être nécessaire de mettre à jour la configuration de votre projet Android pour tester Android O. Pour ce faire, il faudra modifier les propriétés `compileSdkVersion`, `buildToolsVersion` et

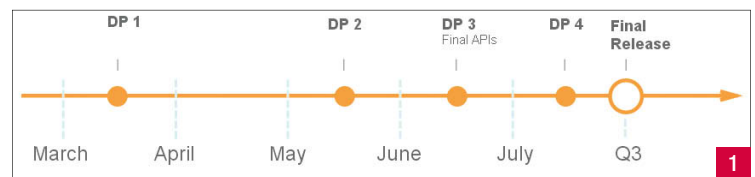


Figure 1 : Roadmap Developer Preview Android O

`targetSdkVersion` tout en changeant la version de la Support Library en dépendance, ce qui nous donne l'Android Manifest suivant :

```
android {
    compileSdkVersion 'android-O'
    buildToolsVersion '26.0.0-rc1'

    defaultConfig {
        targetSdkVersion 'O'
    }
    ...
}

dependencies {
    compile 'com.android.support:appcompat-v7:26.0.0-alpha1'
}
```

Il est bon de noter qu'actuellement il est impossible de publier une application ciblant Android O sur le Google Play Store et que la lettre O sera remplacée par un numéro d'API lorsque Android O sera utilisable en production.

Notifications repensées et améliorées

En évolution permanente depuis Android Lollipop, le système de notifications de l'OS est encore mis sous le feu des projecteurs par Google dans cette version. Avec Android O, Google a repensé et amélioré les notifications en profondeur pour proposer aux développeurs et aux utilisateurs une manière plus simple et plus consistante pour gérer leur comportement et leurs propriétés. La notion de channels de notifications est ainsi introduite offrant aux utilisateurs plus de contrôle et d'efficacité dans la gestion des notifications. Le niveau de contrôle bascule du mode individuel au mode groupe (figure 2).

Pour le développeur, cette nouveauté s'avère intéressante. Précédemment, les utilisateurs pouvaient complètement bloquer une application pour l'empêcher d'afficher des notifications s'ils trouvaient la fréquence d'envoi trop élevée par exemple. Maintenant, il devient possible de créer différents channels pour une même application afin de séparer les notifications par catégories. L'utilisateur pourra ainsi bloquer uniquement le channel ne lui convenant pas et laisser la possibilité à l'application d'utiliser d'autres channels de notifications. Chaque application peut créer autant de channels que nécessaire via la méthode `createNotificationChannel` du `NotificationManager` comme suit :

```
NotificationManager notificationManager = (NotificationManager) getSystemService(
    Context.NOTIFICATION_SERVICE);
String id = "ID_DU_CHANNEL";
CharSequence name = "Le nom du channel visible par l'utilisateur";
int importance = NotificationManager.IMPORTANCE_HIGH;
NotificationChannel channel = new NotificationChannel(id, name, importance);

// Configuration du channel puis création
channel.enableLights(true);
channel.setLightColor(Color.RED);
channel.enableVibration(true);
notificationManager.createNotificationChannel(channel);
```

Au niveau de l'envoi des notifications, il sera nécessaire de définir à quel channel la notification est rattachée. Le rattachement se faisant en utilisant l'identifiant du channel de notifications utilisé à sa création. Avec Android O, l'envoi d'une notification aura donc l'allure suivante :

```
NotificationManager notificationManager = (NotificationManager) getSystemService(
    Context.NOTIFICATION_SERVICE);
// Id de notification qui peut être mis à jour
int notild = 1;
// Id du channel de rattachement
String CHANNEL_ID = "ID_DU_CHANNEL";

// Création de la notification et rattachement au channel
Notification notification = new Notification.Builder(this)
    .setContentTitle("Ma Notification")
    .setContentText("Android O c'est top avec SSaurel")
    .setSmallIcon(R.drawable.ic_notif)
    .setChannel(CHANNEL_ID)
    .build();

// Envoi
notificationManager.notify(notild, notification);
```

Avec les channels de notifications, toutes les notifications envoyées par les applications seront proprement regroupées par catégories. Les utilisateurs auront ensuite la possibilité de paramétrer le comportement et les caractéristiques de ces channels en modifiant le niveau d'importance, le son associé, les lumières associées, ou encore si le channel doit être affiché sur l'écran de verrouillage. Cette customisation des channels se faisant au niveau de l'objet `NotificationChannel`. Comme expliqué précédemment, il sera donc également possible de bloquer totalement un channel de notifications si l'utilisateur le souhaite. Une fois le channel de notifications créé, l'utilisateur peut modifier à sa

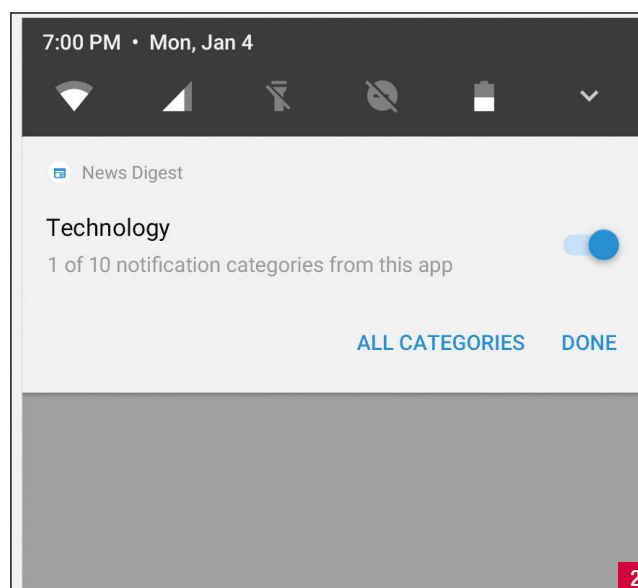


Figure 2 : Channel de Notifications

guise son comportement et ses propriétés. Une application ne peut alors plus modifier les choix faits par l'utilisateur. Il est néanmoins possible de connaître ces réglages afin de s'y adapter au mieux en récupérant l'instance d'un channel particulier :

```
NotificationManager notificationManager = (NotificationManager) getSystemService(
    Context.NOTIFICATION_SERVICE);
// Récupération du channel
NotificationChannel notificationChannel = notificationManager.getNotificationChannel("ID_DU_CHANNEL");
// Récupération du paramétrage du channel
Uri soundUri = notificationChannel.getSound();
int lightColor = notificationChannel.getLightColor();
long[] vibrationPattern = notificationChannel.getVibrationPattern();
```

Il demeure possible de proposer à l'utilisateur de mettre à jour les propriétés d'un channel de notifications en le redirigeant vers l'activité dédiée en passant en paramètre le nom de package de l'application ainsi que l'identifiant du channel :

```
Intent intent = new Intent(Settings.ACTION_CHANNEL_NOTIFICATION_SETTINGS);
intent.putExtra(Settings.EXTRA_CHANNEL_ID, mChannel.getId());
intent.putExtra(Settings.EXTRA_APP_PACKAGE, getPackageName());
startActivity(intent);
```

Enfin, la suppression d'un channel de notifications peut se faire en récupérant une instance donnée via son identifiant puis en passant cette dernière en entrée de la méthode `deleteNotificationChannel` du `NotificationManager` :

```
NotificationManager mNotificationManager = (NotificationManager) getSystemService(
    Context.NOTIFICATION_SERVICE);
String id = "ID_DU_CHANNEL";
NotificationChannel mChannel = mNotificationManager.getNotificationChannel(id);
mNotificationManager.deleteNotificationChannel(mChannel);
```

D'autres modifications plus mineures ont été apportées aux notifications comme la possibilité de définir une couleur d'arrière-plan personnalisée ou encore une fonctionnalité de snoozing permettant de faire réapparaître une notification plus tard. Néanmoins, les channels constituent la nouveauté essentielle en matière de notifications qui viendra améliorer leur gestion au quotidien pour l'utilisateur et pour le développeur.

AutoFill Framework

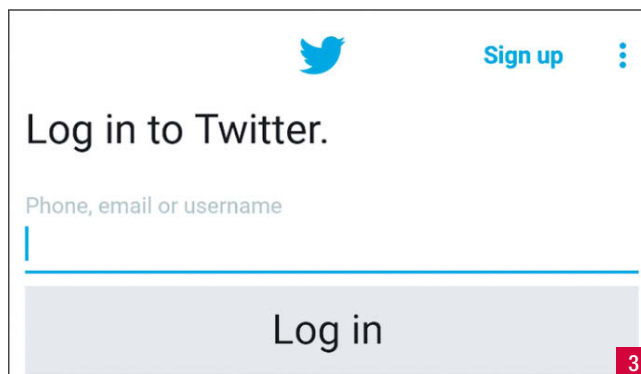
La création de compte, le remplissage de formulaires de login ou de paiement prend du temps tout en étant source d'erreurs pour l'utilisateur. Ce dernier peut être frustré de répéter ces tâches à de multiples reprises au sein d'applications installées sur son appareil. Afin d'adresser cette problématique, Android O introduit l'AutoFill Framework visant à simplifier la vie des utilisateurs en proposant une solution unifiée d'auto complétion de formulaires sous Android.

Pour que votre application puisse bénéficier de l'AutoFill Framework, un service d'auto complétion doit être activé par l'utilisateur au sein des préférences systèmes. Un seul service de ce type peut être actif à la fois sur un appareil Android. Ce service va se charger de parcourir la hiérarchie de vues d'une activité dans le but d'y trouver les vues pour lesquelles une auto complétion sera proposée. Si votre interface utilisateur s'appuie uniquement sur des vues de type standard, l'AutoFill Framework fonctionnera sans travail supplémentaire. Dans le cas où vous avez défini des vues personnalisées, il sera nécessaire d'utiliser l'API AutoFill pour spécifier la structure de l'information au sein de l'application avant que le Framework puisse opérer.

Lorsqu'un évènement AutoFill est déclenché, le Framework envoie la valeur de type AutoFill à votre vue personnalisée. Il devient nécessaire de surcharger la méthode `autoFill` pour définir la manière dont sera gérée et exploitée cette valeur. En outre, la vue devra surcharger les méthodes `getAutoFillType` et `getAutoFillValue` pour préciser au Framework de quel type de vue il s'agit et le type de valeur automatiquement complétée attendue. Une vue personnalisée aura alors l'allure suivante :

```
public MyView extends View {
    public MyView(Context context, AttributeSet attrs) {
        super(context, attrs);
        // ...
    }

    public void autoFill(AutoFillValue value) {
        // Gestion de la valeur auto complétée ...
    }
}
```



AutoFill Framework en action

```
public AutoFillType getAutoFillType() {
    // définition de l'objet AutoFillType à retourner
    return ...;
}

public AutoFillValue getAutoFillValue() {
    // définition de l'objet AutoFillValue à retourner
    return ...;
}
}
```

L'AutoFill Framework profitera principalement aux utilisateurs mais le support permettra aux développeurs d'améliorer l'expérience utilisateur au sein de leurs applications. Sur la [figure 3](#), on peut voir un écran de connexion à Twitter où l'AutoFill Framework pourra apporter un confort d'utilisation majeur.

Picture-In-Picture Mode

Introduit pour les utilisateurs d'Android TV avec Android Nougat, le mode Picture-In-Picture (PIP) permet à un utilisateur d'Android TV d'épingler une vidéo dans un coin de l'écran tout en continuant à naviguer entre les différentes applications de son système. Ce mode améliore l'expérience multi-tâches de l'utilisateur puisqu'il lui donne la possibilité de consommer des contenus multimédias tout en utilisant en parallèle l'application de son choix. Las, ce mode étant réservé à la plateforme Android TV, peu de développeurs avaient franchi le pas pour intégrer cette nouveauté.

En étendant la fonctionnalité pour Android, Android O change la donne et l'on devrait voir l'usage du mode PIP se répandre notamment pour les applications proposant des lectures vidéo ou des slideshows d'images. Afin de mettre en oeuvre cette fonctionnalité au sein de son application, il faudra tout d'abord l'activer pour une activité particulière au sein de l'Android Manifest :

```
<activity android:name="VideoPlaybackActivity"
    android:supportsPictureInPicture="true"
    ...>
```

Le mode PIP est un cas particulier du support multi-fenêtres introduit avec Android Nougat qu'il fallait activer au sein de l'Android Manifest en positionnant à true la propriété `android:resizeableActivity` d'une activité. Ici, le fait d'activer le mode PIP suffit à préciser au système que l'activité supporte le mode multi-fenêtres. Cependant, l'emploi de la propriété `android:resizeableActivity` demeure nécessaire pour une activité supportant le mode multi-fenêtres mais pas le mode PIP. Le passage en mode PIP pour une activité de votre application doit ensuite se faire de manière explicite en appelant manuellement la méthode `enterPictureInPictureMode` de la classe `Activity`. Cette dernière méthode accepte un objet `PictureInPictureArgs` en entrée permettant de configurer le rendu de votre fenêtre PIP (aspect ratio ou encore les actions disponibles).

Bien qu'il soit séduisant, le mode PIP a certaines particularités à prendre en compte pour ne pas nuire à l'expérience utilisateur. La première concerne la taille de la fenêtre PIP qui est de 240 par 135dp. Il faudra donc retirer les éléments de l'interface utilisateur superflus pour un affichage en mode PIP. La seconde spécificité concerne toujours l'interface

utilisateur puisque ce sera à vous de restaurer convenablement les éléments retirés lors du retour en affichage plein écran. Afin de gérer au mieux ce basculement, il sera nécessaire de surcharger la méthode `onPictureInPictureModeChanged` comme suit :

```
@Override
public void onPictureInPictureModeChanged(boolean isInPictureMode) {
    if (isInPictureMode) {
        // Entrée Mode PIP : on cache éléments UI superflus
        ...
    } else {
        // Sortie Mode PIP : on restaure les éléments UI
        ...
    }
}
```

Un autre point d'attention particulier dans la mise en place du mode PIP est la gestion correcte du cycle de vie en mode multi-fenêtres. Ainsi, lorsque votre activité entre en mode PIP, la méthode `onPause()` de l'activité est appelée et l'activité est considérée comme étant en pause. Habituellement, il est de coutume de mettre en pause la lecture de contenus multimédias au sein de la méthode `onPause()`. Cependant, dans le cas du mode PIP, cela reviendrait à le rendre totalement inutile. Une solution peut être de n'interrompre les contenus multimédias qu'au sein de la méthode `onStop()`. Si cela s'avère problématique pour votre application, il est possible de détecter si l'activité est en mode PIP ou non dans la méthode `onPause()` et d'agir en conséquence :

```
@Override
public void onPause() {
    // Pas de pause de la lecture vidéo en mode PIP
    if (!isInPictureInPictureMode()) {
        ...
    } else {
        // On n'est pas en mode PIP, pause de la vidéo
        ...
    }
}
```

Durée de vie de la batterie

Impactant directement l'expérience utilisateur en limitant l'usage d'un appareil, l'autonomie sous Android est un problème crucial que Google a pris à bras le corps depuis Android Marshmallow en introduisant par exemple le mode Doze. Ce dernier avait permis une amélioration significative de la durée de vie de la batterie et avait connu quelques améliorations sous Android Nougat avec le bien nommé Extended Doze Mode.

Pour aller plus loin dans la préservation de la durée de vie de la batterie, Android O s'attaque cette fois directement aux causes de la consommation excessive de la batterie. Ainsi, Android O va limiter l'accès à des fonctionnalités consommatrices s'exécutant en arrière-plan. Dans cette première Developer Preview, les background services, les broadcasts implicites et les mises à jour de la position de l'utilisateur sont concernés. Android O restreint la fenêtre d'exécution des background services à quelques minutes lorsque l'application n'est pas en premier plan. Passé ce délai, le service sera stoppé par l'OS pour éviter tout usage excessif de la batterie. La solution naturelle est de basculer vers l'API `JobScheduler`

introduite sous Android Lollipop qui permet de planifier des jobs et des traitements batchs de manière intelligente en définissant des conditions d'exécutions spécifiques telles que le type de connexion Internet ou encore le fait que l'appareil soit en charge ou non via l'objet `JobInfo`. La planification d'un service en tâche de fond pourra être réalisée comme suit avec cette API :

```
ComponentName myService = new ComponentName(this, MyService.class);
JobInfo myJob = new JobInfo.Builder(myService)
    .setRequiresCharging(true)
    .setRequiredNetworkType(JobInfo.NETWORK_TYPE_ANY)
    .setPersisted(true)
    .build();

JobScheduler jobScheduler = (JobScheduler) getSystemService(Context.JOB_SCHEDULER_SERVICE);
jobScheduler.schedule(myJob);
```

Dans le cadre des limitations liées à l'exécution de traitements en arrière-plan, les applications ciblant Android O ne pourront plus enregistrer des broadcasts receivers ciblant des broadcasts implicites dans leur Manifest. Il faudra désormais réaliser ces enregistrements de manière programmatique en recourant à la méthode `registerReceiver`. Il est bon de noter que ces limitations ne concernent pas les broadcasts explicites. Ce choix réalisé par Google est judicieux dans le sens où de nombreux broadcasts inutiles sont envoyés par des applications n'ayant pas besoin d'écouter le retour obtenu à l'instant présent ce qui entraîne une consommation inutile de la batterie. Dans la pratique, considérons la déclaration du `BroadcastReceiver` suivant dans l'Android Manifest :

```
...
<receiver android:name=".ChargerReceiver">
    <intent-filter>
        <action android:name="android.intent.action.ACTION_POWER_CONNECTED" />
    </intent-filter>
</receiver>
...
```

Pour garder ce `BroadcastReceiver` opérationnel après le passage à Android O, il faut retirer la déclaration précédente de l'Android Manifest et déplacer la logique d'enregistrement au sein d'une activité de votre application :

```
public class MainActivity extends AppCompatActivity {
    private BroadcastReceiver chargerReceiver;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        chargerReceiver = new BroadcastReceiver() {
            @Override
            public void onReceive(Context context, Intent intent) {
                // code ...
            }
        };
    }
};
```



```

registerReceiver(chargerReceiver,
    new IntentFilter(Intent.ACTION_POWER_CONNECTED)
);
}

@Override
protected void onDestroy() {
    super.onDestroy();
    unregisterReceiver(chargerReceiver);
}
}

```

Les dernières restrictions imposées par Android O concernent la récupération de la position de l'utilisateur par des applications en arrière-plan. À partir d'Android O, une application en arrière-plan ne pourra plus récupérer les mises à jour de la position de l'utilisateur qu'un certain nombre de fois par heure. Ce nombre devant être fixé d'ici à la sortie officielle d'Android O. En outre, toutes les applications seront impactées par cette restriction qu'elles ciblent ou non Android O comme SDK de référence à la compilation. Les développeurs abusant de ce type de requêtes vont devoir corriger le tir dans les mois à venir. Pour les applications nécessitant plus de mises à jour de positions, il faudra faire passer l'application au premier plan, utiliser un foreground service ou bien s'appuyer sur l'API GeoFencing spécifiquement conçue pour préserver la batterie.

Divers

Outre les nouveautés majeures détaillées au cours de l'article, Android O propose également quelques nouveautés sympathiques qui faciliteront la vie des développeurs. On peut ainsi citer la fonctionnalité "Fonts in XML" permettant d'utiliser des polices de caractères personnalisées en tant que ressources au sein d'une application Android. Les icônes adaptatives permettent quant à elles de proposer des icônes différentes suivant le

constructeur de l'appareil pour une même application en s'appuyant sur le système de l'appareil sous-jacent sensé fournir un masque spécifique utilisé par Android O pour le rendu des icônes sous différentes formes. Ces icônes supportent également les effets visuels de type zoom ou rebond lors de l'interaction avec l'utilisateur. Pour créer une telle icône, il faudra définir un fichier XML sous `res/mipmap-anydpi/ic_launcher.xml` contenant une icône masquable avec 2 couches :

```

<maskable-icon>
  <background android:drawable="@color/ic_background"/>
  <foreground android:drawable="@mipmap/ic_foreground"/>
</maskable-icon>

```

Enfin, on citera le composant `TextView`, mis à jour pour supporter le redimensionnement automatique avec des possibilités de configuration intéressantes, ainsi que le composant `WebView` qui se voit ajouter des APIs améliorant sa stabilité et sa sécurité.

CONCLUSION

Fort du succès de son programme Developer Preview initié en 2016, Google reconduit l'expérience en 2017 pour Android O dont le nom définitif ne sera lui connu qu'à la sortie officielle de cette nouvelle mouture à la fin de l'été 2017. Avec Android O, pas de révolutions majeures mais des évolutions importantes qui préfigurent ce que sera le futur d'Android avec l'arrivée du mode Picture-In-Picture et une amélioration du support du multi-fenêtres. Les nouveautés concernant les notifications visent à mieux catégoriser l'information affichée à l'utilisateur pour lui simplifier la vie. Enfin, l'amélioration de la durée de vie de la batterie est un point essentiel pour tous les utilisateurs et les efforts de Google en la matière ne s'arrêteront pas avec Android O. Le tour du propriétaire réalisé, vous pouvez vous lancer dès à présent dans Android O afin de préparer vos applications pour qu'elles soient opérationnelles dès la sortie de la nouvelle mouture.

Restez connecté(e) à l'actualité !

- **L'actu** de Programmez.com : le fil d'info **quotidien**
- La **newsletter hebdo** : la synthèse des informations indispensables.
- **Agenda** : Tous les salons, barcamp et conférences.

The screenshot shows the Programmez.com website. At the top, there's a search bar and navigation links like 'Actualités', 'Avis d'experts', 'Livres blancs', 'Boutique', 'Livres', 'Logiciels', 'Tutoriels', 'Emploi', 'Agenda', and 'Programmez! Collectors #1'. A featured article titled 'Souriez ! Vous êtes analysé(e)s ! Les technologies cognitives envahissent notre quotidien' is prominent. Other articles include 'Intel lance son premier Bug Bounty : jusqu'à 30 000 dollars de récompense' and 'Programmez! Collectors #1 - Hors-série vintage'. There are also buttons for 'ACHETER' and 'S'ABONNER'.

Abonnez-vous, c'est gratuit ! www.programmez.com

Un support client optimisé en connectant la VoIP à votre Back Office Magento

Partie 2

• Jean-Marie Heitz,
Ingénieur études et développement
e-commerce chez
Netapsys Grand Est
Jean-marie.heitz@netapsys.fr

*Dans cette seconde partie,
nous allons parler d'Asterisk et de CTI.*

Asterisk et le CTI

Asterisk est un IPBX open source très populaire. Il a la particularité de s'interfacer avec un grand nombre de technologies, dont les technologies de télécoms traditionnelles, via des cartes d'extension. En outre, le cœur d'Asterisk est le plan de numérotation, qui régit le traitement des numéros appelés (au sens large) en fonction du contexte d'appel de l'appelant. Asterisk est prévu pour être extrêmement souple, et de ce fait, peut être d'une très grande aide au couplage téléphonie informatique.

PLAN DE NUMEROTATION : intégration ODBC et CURL

Par exemple, depuis le plan de numérotation, on peut sans problème effectuer des requêtes SQL en ODBC à la fois en *read* et en *write*, ce qui signifie également que si cela est nécessaire, on peut par exemple implémenter certaines fonctions via des procédures stockées. Dans notre cas, on pourrait très bien attaquer la BD de Magento depuis le plan de numérotation, et ainsi par exemple récupérer un nom de client pour le mettre en place pour l'affichage sur les postes clients, voire augmenter l'information avec des informations courtes, comme par exemple le groupe de clients auquel il appartient.

De même, on pourrait très bien envisager de récupérer une assignation du commercial responsable du client, de sorte à diriger l'appel en priorité au responsable du compte, si un développement spécifique a été fait dans Magento dans ce sens.

Dans le même esprit, on peut également réaliser des appels CURL via le plan de numérotation pour récupérer les informations de cette manière, notamment si la logique à suivre est trop complexe pour être mise en base de données ou en plan de numérotation ; Magento est particulièrement rompu à cet usage, puisque le framework propose des URL structurées avec des paramètres facilement récupérables.

```
odbc read ODBC_MAGE_NAME_LOOKUP 1235550001 exec
[Aug 4 17:55:55] DEBUG[1459]: pbx_variables.c:474 ast_str_substitute_variables
_full: Evaluating 'ARG1' (from 'ARG1') left join customer_address_entity_varchar f
on f.entity_type_id=t.entity_type_id and f.attribute_id=(select a.attribute_id from
eav_attribute a where a.entity_type_id=f.entity_type_id and a.attribute_code=
'firstname' limit 1) and e.entity_id=f.entity_id left join customer_address_entity
_varchar n on n.entity_type_id=t.entity_type_id and n.attribute_id=(select b.
attribute_id from eav_attribute b where b.entity_type_id=f.entity_type_id and b.
attribute_code='lastname') and e.entity_id=n.entity_id limit 1' len 4)
[Aug 4 17:55:55] DEBUG[1459]: pbx_variables.c:381 ast_str_retrieve_variable:
Result of 'ARG1' is '1235550001'
[Aug 4 17:55:55] DEBUG[1459]: res_odbc.c:864 _ast_odbc_request_obj2: Reusing
ODBC handle 0xa3faa9c from class 'magento'
[Aug 4 17:55:55] DEBUG[1459]: func_odbc.c:1502 cli_odbc_read: Found handle
magento
```

```
name Gogi Cooper
Returned 1 row. Query executed on handle 0 [magento]
*CL> [Aug 4 17:55:55] DEBUG[1459]: res_odbc.c:713 ast_odbc_release_obj:
Releasing ODBC handle 0xa3faa9c into pool
```

Exemple de fonction ODBC définie dans Asterisk exécutée sur la ligne de commande, afin de récupérer prénom et nom depuis la BD Magento.

MENUS VOCAUX ET TRANSFERT D'INFORMATION

Les menus vocaux peuvent être utilisés comme premier niveau de qualification d'appels entrants. Les informations saisies par l'utilisateur grâce aux codes DTMF générés par l'appui sur les touches doivent pouvoir être transmises à d'autres périphériques. Ces informations peuvent être stockées en BD, ou transmises par CURL, mais néanmoins comme ces informations sont en relation directe avec l'appel, il peut aussi être intéressant de les faire transiter directement dans la signalisation de l'appel au niveau SIP. Ceci est possible car on peut, au niveau d'Asterisk, ajouter des *headers* de son choix qui seront transmis dans la requête INVITE d'établissement de l'appel. Charge cependant à l'UA de traiter ces *headers*.

Dans le cadre de Magento, il ne faut pas négliger ce type de solution, car elle permet de décharger un opérateur de tâches répétitives sans grande valeur ajoutée, comme par exemple la saisie d'un numéro de client ou la saisie d'un numéro de commande. Cette démarche est d'autant plus judicieuse que Magento utilise à de très nombreux endroits des *entity_id* et que l'on peut automatiser des contrôles de cohérences en vérifiant par exemple la cohérence entre un numéro de commande, un numéro de téléphone et un numéro de client.

FICHIERS D'APPEL (CALL FILES)

Les fichiers d'appel sont des fichiers qui définissent un canal que l'on doit tout d'abord contacter, et s'il répond, Asterisk lance la communication vers l'endroit spécifié dans le fichier (c'est à dire, fait démarrer l'appel dans un endroit du plan de numérotation, comme si le canal avait appelé). Grâce à la technologie des canaux locaux d'Asterisk, cette technique n'est pas limitée à un canal bien défini, car on peut utiliser le canal local pour que la partie "futur appelant" soit l'objet d'un traitement au niveau du plan de numérotation.

Cette technique est assez simple, mais peut éventuellement poser des questions quant à son utilisation dans le cadre d'un couplage avec deux systèmes isolés, à cause de l'accès requis au système de fichiers. Il est même nécessaire d'avoir accès à deux répertoires : un répertoire pour préparer les fichiers, de sorte que les fichiers soient entièrement écrits d'abord, et ensuite on peut les déplacer vers le répertoire de *spool* d'Asterisk ; les deux devant être dans le même système de fichiers de sorte que le déplacement du fichier soit une opération atomique ne pouvant pas donner lieu à un problème de fichier lu alors qu'il n'est qu'à moitié écrit.

Avec l'existence d'une telle possibilité, on peut déjà améliorer le quotidien des commerciaux, puisqu'elle réduit les risques d'erreur lors de la saisie des numéros de téléphone et réduit le temps nécessaire pour composer les numéros de téléphone.

On pourrait également envisager des campagnes d'information ciblées en fonction des produits achetés, en préparant des annonces enregistrées que l'on souhaite diffuser aux clients.

ASTERISK GATEWAY INTERFACE

L'allusion à CURL fait penser à un autre type de solution : l'utilisation de l'Asterisk Gateway Interface, qui est le fait que pendant un appel, Asterisk arrive à s'interfacer avec des scripts qui communiquent avec lui au travers de *stdin* et *stdout*, pour faire exécuter diverses commandes, dont des assignations de variables, l'exécution d'applications du plan de numérotation, ou une redirection de l'appel à un autre endroit du plan de numérotation.

Il existe même des variantes qui permettent d'utiliser des connexions TCP à la place de *stdin* et *stdout*, ce qui permet de garder la spécialisation des serveurs tout en gardant les possibilités d'interactions accrues d'AGI.

On en déduit qu'on pourrait ainsi créer des scripts qui allient la puissance de Magento (avec la classe Mage) et les commandes AGI d'Asterisk. Ce type de fonctionnement permet également de répondre à un autre type de problématique: la délimitation des secteurs de responsabilité. En effet, on constate régulièrement que même si la tendance DevOps est une tendance de plus en plus forte, où les développeurs doivent avoir une très bonne idée du fonctionnement de l'hébergement pour comprendre, et résoudre les problématiques qui peuvent en résulter, le développement de solutions et leur hébergement sur une infrastructure restent encore des métiers séparés. Il en va de même avec la téléphonie : on ne peut pas laisser par exemple la main sur l'ensemble d'un plan de numérotation à des développeurs, pour des questions de responsabilité. On peut donc utiliser AGI pour définir une ligne de démarcation entre les domaines de responsabilité des développeurs des scripts AGI et de l'équipe infrastructure en charge du PABX.

ASTERISK MANAGER INTERFACE

L'Asterisk Manager Interface est un autre type d'interface, qui s'apparente plutôt à une sorte de telnet. L'AMI s'exécute hors contexte d'appel, mais permet néanmoins d'être une source d'informations sur les appels, dans la mesure où l'on peut demander à récupérer des événements que l'on peut également filtrer. Dans le cadre de l'AMI, il est également possible de travailler sur des appels en cours, ou d'émettre de nouveaux appels. Du fait que l'on a accès à l'ensemble des appels, on peut par exemple utiliser l'AMI pour monitorer les appels en cours, comme le fait le Flash Operator Panel, qui est un outil complémentaire d'Asterisk assez connu. De plus, cet outil a une gestion de droits, donc il est possible de créer des comptes avec des droits spécifiques. L'utilisation de l'AMI pourrait donc être une source centralisée d'informations sur tous les appels en cours en mode push, puisque c'est Asterisk qui fournit les événements. Toutefois, cette interface a comme inconvénient d'être une interface de type telnet, ce qui nécessite certaines pratiques dans le développement de l'interfaçage avec AMI, comme par exemple de bien re-segmenter les "blocs d'information", et de bien distinguer une remontée d'information, d'une réponse à une commande.

L'AMI dispose d'un système de contrôle d'accès, avec une authentification pouvant utiliser différentes méthodes, ainsi qu'une gestion des droits : on peut donc préparer des comptes pour des applications. Un exemple

est la gestion des appels évoquée précédemment : on peut par exemple souhaiter limiter le nombre de ressources simultanées utilisées pour les appels : l'AMI peut fournir des informations et peut permettre de déclencher des appels à bon escient (en ne les déclenchant que si le nombre de canaux que l'on souhaite utiliser n'est pas atteint).

ASTERISK REST INTERFACE

L'Asterisk Rest Interface n'est pas qu'une simple interface RESTful : certes, cette interface existe, et est exposée au travers du serveur HTTP intégré à Asterisk, serveur qui peut également être utilisé pour accéder à l'AMI par exemple. Mais cette interface REST est complétée par un service événementiel qui est implémenté au-dessus de WebSockets, avec des messages encapsulés dans du JSON, ainsi qu'une application particulière au niveau du plan de numérotation, qui est l'application Stasis.

Asterisk propose à travers cet ensemble de technologies un moyen qui permet de déporter l'intelligence du plan de numérotation vers une application externe. Pour ce faire, l'application doit ouvrir une connexion WebSockets et doit s'enregistrer pour souscrire aux événements liés à un "nom d'application" défini par le développeur. Une fois que cette souscription est active, un appel routé vers l'application Stasis va envoyer des événements à l'application à travers les WebSockets pour signifier de manière asynchrone des informations telles que le début d'appel ou un raccrochage. L'application devra piloter entièrement ce qui doit être fait à travers les commandes implémentées dans l'interface REST, et se tiendra au courant via les événements asynchrones envoyés.

De par la nature asynchrone des notifications d'événements, il est la plupart du temps utile de définir des ID uniques au niveau de l'application afin de faire des correspondances entre les commandes envoyées et les événements reçus. Cette commande est donc une étape extrêmement aboutie de couplage téléphonie informatique, la partie informatique représentée par l'application créée contrôlant l'appel de son côté et pouvant avoir des informations sur le fonctionnement du PABX lui-même, ; la partie téléphonie, fournie par Asterisk, faisant juste le lien avec le tiers impliqué dans l'appel.

```
<?php
require_once('abstract.php');
require_once('../lib/autoload.php');
class Jmh_Ari extends Mage_Shell_Abstract
{
    protected $_stasisClient;
    protected $_stasisEvents;
    /**
     * @var phpari $_ariObject
     */
    protected $_ariObject;
    protected $_stasisLoop;
    public function run()
    {
        $this->_ariObject = new phpari('version');
        $this->_stasisClient = $this->_ariObject->stasisClient;
        $this->_stasisEvents = $this->_ariObject->stasisEvents;
        $this->_stasisEvents->on('StasisStart',function($event)
        {
            $channelId=$event->channel->id;
            $this->_ariObject->channels()->channel_set_variable($channelId,'mage_version',
            Mage::getVersion());
```

```
$this->_ariObject->channels()->channel_continue($channelId,'ari','10',3);
}
);
$this->_stasisClient->on("message", function ($message) {
    $event = json_decode($message->getData());
    $this->_stasisEvents->emit($event->type, array($event));
});
$this->_stasisLoop = $this->_ariObject->stasisLoop;
$this->_stasisClient->open();
$this->_stasisLoop->run();
}
}
$ari = new Jmh_Ari();
$ari->run();
```

Exemple d'utilisation de la librairie PHPARI qui utilise l'ARI d'Asterisk : dans ce code shell Magento, on cherche à transmettre la version courante de Magento. A l'utilisation :

```
[Aug 15 13:16:23] DEBUG[1383]: ari/ari_websockets.c:177 ast_ari_websocket
_session_write: Examining ARI event (length 561):
{
    "variable": "mage_version",
    "value": "1.9.2.4",
    "type": "ChannelVarset",
    "timestamp": "2016-08-15T13:16:23.750+0200",
    "channel": {
        "id": "1471259781.2",
        "name": "SIP/jm-00000002",
        "state": "Up",
        "caller": {
            "name": "",
            "number": ""
        },
        "connected": {
            "name": "",
            "number": ""
        },
        "accountcode": "",
        "dialplan": {
            "context": "ari",
            "exten": "10",
            "priority": 2
        },
        "creationtime": "2016-08-15T13:16:21.769+0200",
        "language": "fr"
    },
    "application": "version"
}
```

On constate dans la capture de debug que la version utilisée est bien remontée au niveau d'Asterisk.

CALL DETAIL RECORD

Les CDR sont des enregistrements des informations vis à vis d'un appel, ou éventuellement d'une partie du trajet de l'appel. Ces informations peuvent éventuellement être intéressantes pour enrichir un CRM, en complément des informations qu'un commercial aura saisies sur l'appel. En revanche, l'intérêt des CDR sous Asterisk reste quand même limité,

car la pertinence des informations loguées dépend également du plan de numérotation : plus un plan de numérotation est complexe (par exemple, avec l'utilisation de canaux de type Local), plus les CDR seront compliqués à exploiter - et ce, sans parler des cas de transfert d'appel. En plus, les informations ne sont pas disponibles avant la fin de l'appel, voire même avant un certain temps (par exemple si on utilise le batching). Cependant, on notera que la structure de données standard propose quand même un champ défini par l'utilisateur, champ qui pourrait par exemple être rempli automatiquement par le système pour se noter la qualification d'un appel réalisé par un menu vocal.

Les CDR seront utiles si on les récupère à des endroits stratégiques et où le plan de numérotation est simple ; éventuellement par exemple sur une passerelle SIP / TDM. Comme ils peuvent être stockés dans des bases de données comme MySQL, on peut envisager plusieurs types d'intégrations (vues, moteur de stockage MySQL permettant d'attaquer un autre serveur, mais aussi connexion vers une deuxième BDD au niveau de Magento, en utilisant une connexion spécifique).

CHANNEL EVENT LOGGING

Le CEL est une technologie plus récente qui est apparue en raison des cas non assurés par les CDR : l'idée est de générer des événements liés à la vie des canaux, et même si on le souhaite des événements personnalisés, qui devraient pouvoir être plus exploitables pour tirer de la valeur ajoutée des informations loguées. Ceci signifie que l'on peut chercher en fonction du contexte à définir des événements que l'on va lancer à travers le plan de numérotation pour les exploiter ensuite. Même si les CEL suivent en fait l'évolution des canaux, il faut quand même admettre que leur exploitation peut s'avérer compliquée, et que l'instrumentation complète d'un plan de numérotation à l'aide d'événements personnalisés est difficile à obtenir - aussi difficile qu'il est de prévoir tous les cas qui peuvent se présenter. Mais on peut profiter de cette possibilité pour qualifier un appel par exemple, même en cours de communication avec un agent (via des fonctionnalités spécifiques en cours de communication configurées dans features.conf, par exemple). Comme les CEL peuvent également être stockés en base de données, on pourra par exemple utiliser les possibilités qu'offre Magento en termes de grid, d'export de données, et plus généralement, de reporting, pour exploiter ces données.

ENVOI DE SMS

L'envoi de SMS est en théorie possible avec certaines applications du plan de numérotation d'Asterisk, mais je n'ai pour l'instant pas réussi à le faire fonctionner. Ceci n'est cependant pas la seule possibilité d'envoyer des SMS : on peut par exemple envisager de transformer un téléphone Android en passerelle SMS, en fonction des volumes que l'on souhaite atteindre - et également si l'opérateur permet un tel usage. Sinon, il existe des API "propriétaires" d'entreprises pour envoyer des SMS.

L'envoi de SMS a un coût, mais un coût réduit, et est dans une certaine mesure, l'équivalent d'un mail, à ceci près qu'un SMS est court et dispose généralement et historiquement d'une notification immédiate à l'utilisateur qui a un téléphone portable : c'est donc un moyen de communication précieux puisque pouvant accéder de manière privilégiée au client. De plus, Magento dispose déjà d'un système assez sophistiqué pour générer des mails à partir de templates. Comme il s'agit à la base dans les deux cas de produire un texte personnalisé, il sera donc envisageable de profiter de ce système existant pour l'adapter et l'utiliser pour l'envoi de SMS personnalisés.

• **La suite le mois prochain**

Créer une API en quelques minutes



Philippe THOMAS
IT Architect Cloud, Analytics and
Monitoring Business Development & ISV
- Business Solution Center Nice Meridia -
IBM France
thomas1@fr.ibm.com

Les interfaces de programmation (API) font le lien numérique entre les services, les applications et les systèmes pour créer des expériences client uniques. Elles vous permettent de créer rapidement des interfaces entre les systèmes de gestion et les applications. Partager ces interfaces avec les clients et les développeurs peut vous aider à lancer de nouveaux services numériques sur le marché, à ouvrir de nouveaux canaux de revenus et à dépasser les attentes de vos clients.

Il est essentiel d'avoir une solution de gestion capable de traiter les aspects critiques du cycle de vie des APIs pour les environnements sur site et dans le cloud. IBM API Connect propose des fonctions pour créer, exécuter, gérer et sécuriser les APIs et les microservices. Elle vous permet également de déployer rapidement et de simplifier l'administration des APIs à l'échelle de votre entreprise.

IBM API Connect propose les fonctions de gestion du cycle de vie des APIs suivantes :

- **Créer** — Créez des APIs de haute qualité, évolutives et sécurisées pour les serveurs d'applications, les bases de données, les bus de services d'entreprise (ESB) et les mainframes en quelques minutes.
- **Exécuter** — Profitez d'outils intégrés pour créer, déboguer et déployer des APIs et des microservices avec Node.js ou Java.
- **Gérer** — Créez et gérez des portails qui permettent aux développeurs de reconnaître et de consommer rapidement des APIs, d'accéder aux données d'entreprise en toute sécurité et de surveiller les APIs pour améliorer les performances.
- **Sécuriser** — Les administrateurs peuvent gérer la sécurité et la gouvernance des APIs et des microservices. L'IT peut définir et appliquer des règles API pour protéger les actifs informationnels d'arrière-plan et respecter la gouvernance et les obligations réglementaires.

Dans cet article, nous vous proposons de créer une interface et le micro-service Node.js associé, de les tester et de découvrir une majorité des fonctionnalités de notre solution qui concernent le développeur et facilite la programmation. La gestion et la sécurisation des APIs ne seront pas couverts dans cet article.

Pour vous aider à rapidement créer, développer, améliorer et gérer vos APIs de façon sécurisée, la plateforme PaaS Bluemix basée sur les standards ouverts du marché est particulièrement adaptée.

Installer API Connect

API Connect est constitué de plusieurs modules fournis dans différents formats (VM, Cloud, Docker, npm). Dans cet article, nous allons installer API Connect sur l'ordinateur du développeur afin qu'il puisse rapidement interagir avec la solution. Il pourra par la suite déployer ses APIs vers un environnement de production disponible dans son centre informatique ou sur le Cloud. L'API Connect Developer Toolkit, que nous allons installer, présuppose que Node.js version 4.x LTS soit déjà installé sur votre poste de travail.

Pour vérifier cela, ouvrir un terminal ou une ligne de commande :

```
>node -v
v4.5.0
>npm -v
v3.8.6
```

Si npm est installé dans une version **différente** de la v3, utiliser cette commande :

```
$ sudo npm install -g npm
ou
>npm install -g npm
```

Installer ensuite API Connect Developer Toolkit avec la commande suivante :

```
>npm install -g apiconnect
```

Cette commande permet d'installer rapidement les composants suivants :

- API Connect Developer Toolkit (la ligne de commande apic et l'interface graphique API Designer) ;
- Le framework Node.js LoopBack ;
- Le microgateway (nous verrons plus loin de quoi il s'agit).

Vous pourrez passer la commande suivante pour vérifier la bonne fin de l'installation :

```
>apic -v
API Connect: v5.0.3.0 iFix 1
```

Création d'une API simple

Dans cette partie, nous allons créer une application Node.js simple (**mynotes**) et son API en utilisant la commande en ligne apic et le framework LoopBack.

```
>apic loopback -n mynotes
[?] What's the name of your application? mynotes
[?] Enter name of the directory to contain the project: mynotes
    create mynotes/
info change the working directory to my-notes
[?] What kind of application do you have in mind? (Use arrow keys)
```

```
empty-server (An empty LoopBack API, without any configured models or datasources)
hello-world (A project containing a controller, including a single vanilla Message
and a single remote method)
> notes (A project containing a basic working example, including a memory database)
```

Cette ligne de commande est interactive et permet de choisir entre différentes options en utilisant son curseur. Après les questions, npm est automatiquement démarrée et le répertoire **mynotes** est généré puis rempli de sous-répertoires et de fichiers (scaffolding).

```
child process npm install exited with code 0

Updating swagger and product definitions
Created C:\sample\phil\mynotes\definitions\mynotes.yaml swagger description
Created mynotes-product.yaml product definition [mynotes:1.0.0]

Next steps:

Change directory to your app
$ cd C:\sample\phil\mynotes
Create a model in your app
$ apic create --type model
Compose your API, run, manage, enforce and deploy it with API Connect
$ apic edit
Run the app
$ apic start

C:\sample\phil>
```

Déplaçons-nous un instant dans le répertoire **mynotes** afin d'en observer le contenu. Si vous êtes familier avec Node.js, vous devriez reconnaître le contenu d'une application. Voici quelques explications sur les sous-répertoires et fichiers les plus intéressants :

client

Si l'application a un front-end, c'est là que l'on va mettre les fichiers HTML, CSS, JavaScript, etc. Dans notre exemple, il n'y a que le fichier README.md.

common

Il s'agit des fichiers communs au serveur et au client dans l'application.

common/models

Ce sous-répertoire contient tous les modèles (fichiers JSON et JavaScript). Lorsqu'on parle de modèle, on pense à un modèle de données.

common/models/note.js

Il s'agit d'un script personnalisable (Custom script) associé à notre modèle note.

common/models/note.json

C'est le fichier de définition associé à notre modèle de données note.

definitions

Ce répertoire contient toutes les définitions en relation avec les APIs et le produit associé et contenu dans cette application mynotes.

definitions/mynotesproduct.yaml

C'est le fichier YAML pour le produit mynotes. Un produit identifie les caractéristiques de limitation ou de commercialisation d'une API.

definitions/mynotes.yaml

Il s'agit de la définition Swagger (OpenAPI) correspondant à l'API mynotes. Il contient tous les chemins REST, les opérations, les schémas pour les data models et les contraintes de sécurité ...

node_modules

Ce répertoire contient tous les modules pour faire fonctionner cette application.

package.json

Cela correspond à la spécification standard Node.js pour la description du package et des dépendances entre les modules.

server

Ce répertoire contient les modules concernant le côté serveur de notre application.

server/server.js

Il s'agit du principal script pour le serveur.

server/config.json

Ce fichier contient la configuration globale de notre application comme par exemple le root REST API, le hostname et le port.

server/model-config.json

Finalement, ce fichier sert de lien entre les sources de données et spécifie comment elles sont exposées en REST notamment.

server/datasources.json

Ce fichier contient les descriptions des sources de données.

Utilisation de l'API Designer

A ce stade, nous avons créé une application Node.js baptisée **mynotes**, une API et une base de données en mémoire. Cette application très simple permet d'accéder aux notes enregistrées dans cette base (en mode CRUD), de les créer, supprimer et modifier. Il est possible de gérer cette API en continuant à utiliser notre commande apic ou bien à l'aide de l'**API Designer** (équivalent fonctionnel de la commande apic). Pour accéder à l'API Designer, taper :

Sur Windows

```
>cd mynotes
>set SKIP_LOGIN=true
```

Ou bien sur les autres systèmes (Linux et OSX)

```
$ cd mynotes
$ export SKIP_LOGIN=true
```

Puis :

```
>apic edit
Express server listening on http://127.0.0.1:9000
```

Une fenêtre devrait s'ouvrir dans votre navigateur : [1]

Quelques explications avant de lancer notre application. Dans la barre

bleue en haut de l'écran, vous trouverez quelques fonctions d'administration concernant notre API : [2]

- **Publish** : permet de déployer notre API, notre produit et les modèles associés vers une solution API Connect qui tourne sur un autre système ou dans le Cloud en production par exemple. API Connect propose différents modes de fonctionnement (VM, Docker, Bluemix ou autres Clouds).
- **Explore** : permet de visualiser les différentes URI et les opérations associées.
- **La roue dentelée** : elle permet de reconfigurer le répertoire et les noms des APIs.
- **Le point d'interrogation** : c'est le point d'entrée pour l'aide, la documentation en ligne, les forums ...
- **Le visage** : donne accès au profil utilisateur.

Un peu plus bas, vous trouverez une barre avec plusieurs onglets permettant de gérer les différents aspects d'une API : [3]

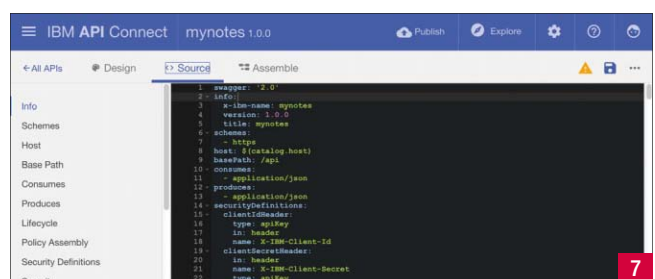
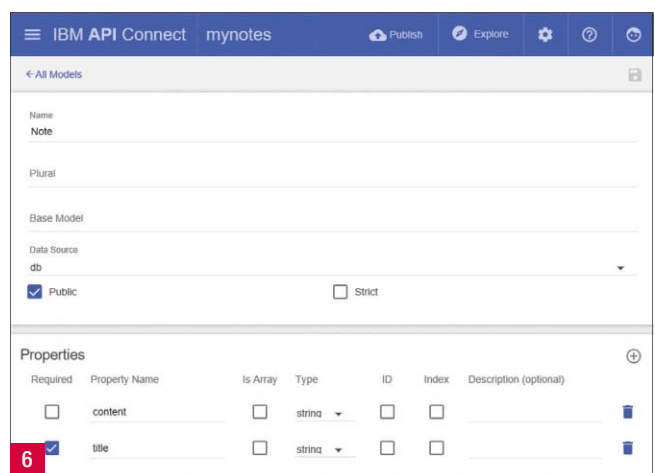
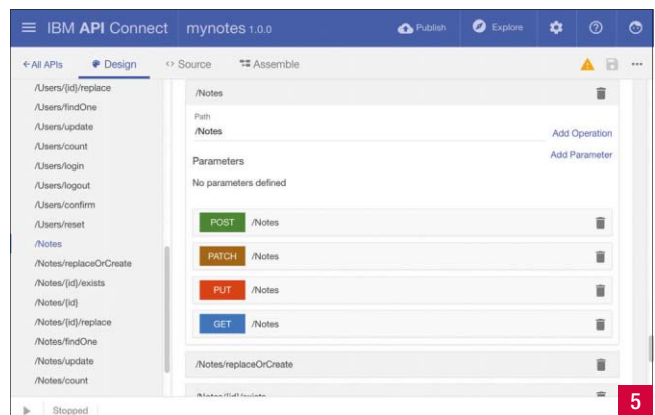
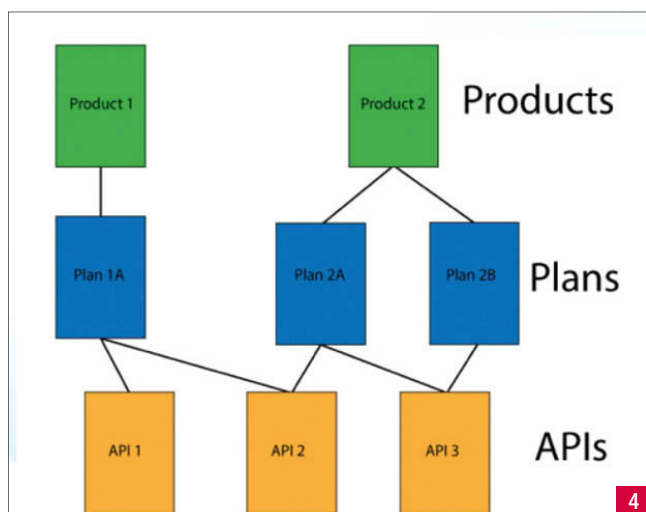
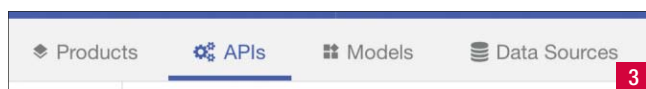
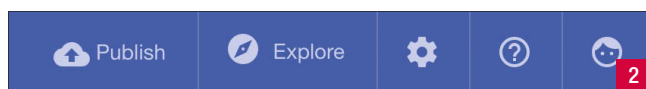
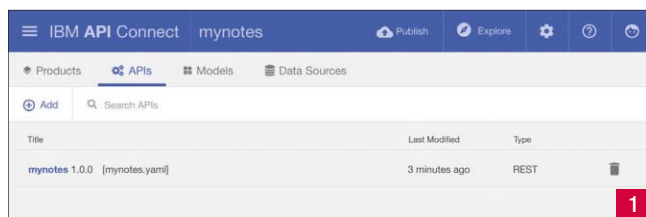
- **Products** : les produits permettent de gérer les aspects de monétisation et de limitation des APIs vis à vis des développeurs externes. Chaque produit contient généralement un ou plusieurs plans décrivant les limitations concernant une ou plusieurs APIs (SLA, limitations) et offrant des possibilités et des prix différents. [4]
- **APIs** : cet onglet permet de prendre en compte toute la définition technique de l'API conformément au standard OpenAPI (Swagger v2). Cliquer sur l'onglet **APIs** puis sur la ligne **mynotes** puis sur **Design**. [5]
- **Models** : ils définissent les modèles de données, typiquement la table et les colonnes pour les bases de données notamment. Dans notre exemple, nous avons créé une base de données en mémoire (in-

memory DB) qui contiendra des notes (title et content) : [6]

- **Data Sources** : elles correspondent à la définition de tous types de bases de données ou de moyens de transport. Par exemple : in-memory DB, Email, DB2, DashDB, MQLight, Cloudant DB, DB2 on z/OS, Mongo DB, MySQL, PostgreSQL, Oracle, MS SQL, Rest Services, SOAP Webservices, CouchBase, Neo4j, Kafka, SAP HANA ... Pour chaque source, vous devrez spécifier les paramètres d'accès (adresse, url, userID, password, base de données ...) et vous devrez également importer le connecteur correspondant (en un clic) afin qu'il soit présent dans l'application Node.js.

On aura noté qu'il est possible de gérer (ajouter, modifier, supprimer) des éléments de notre API et de notre application Node.js associée (mynotes). C'est un réel plaisir de voir que le répertoire mynotes et ses fichiers de configuration sont modifiés automatiquement et dynamiquement en fonction des modifications apportées dans l'interface graphique.

Si vous cliquez sur l'onglet **APIs** puis sur la ligne **mynotes** puis sur l'onglet **Source**, vous devriez voir le code source contenu dans .../mynotes/definition/mynotes.yaml [7]



Le gateway : un composant indispensable pour notre API

Pour faire tourner notre API, nous avons besoin de deux choses : notre application mynotes et un **gateway**. Le gateway est l'élément indispensable qui permet la sécurisation, le filtrage, l'assemblage et la gestion complète du flux http/s avant qu'il n'atteigne l'application.

Différents types de gateway sont disponibles :

- Le **microgateway** : il s'agit d'un programme Node.js qui se charge de ces tâches de sécurisation, d'assemblage, de filtrage ... Ce gateway est programmable en Javascript.
- Le **DataPower gateway** : il s'agissait au départ d'une Appliance physique. En plus du boîtier, ce gateway est également présenté sous d'autres formats : VM, Docker, sur Bluemix ... Il présente des garanties de sécurisation et des fonctionnalités plus importantes que le microgateway.

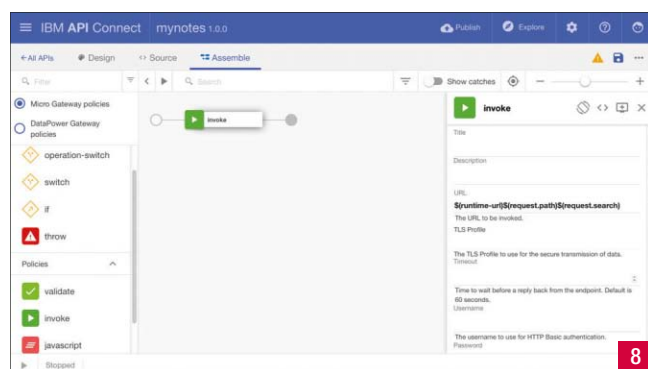
Les fonctionnalités du gateway sont accessibles dans le Toolkit à travers l'onglet Assemble. Cliquez sur l'onglet **APIs** puis sur la ligne **mynotes** puis sur l'onglet **Assemble**. Cliquez sur la politique de traitement **Invoke** dans le centre afin de visualiser sur la droite le contenu de cette politique. [8]

Afin de gérer un flux http/s qui arrive sur le gateway, nous aurons à notre disposition une palette de politiques (icônes sur la gauche). A ce stade, vous pouvez choisir le type de gateway, les opérateurs logiques et les politiques les plus appropriés selon vos notions de sécurité (par copier/coller au centre).

Dans notre cas de figure, nous n'allons pas faire de modification et utiliser la politique **Invoke** telle que mentionnée ici (passage de l'URI telle quelle vers l'application mynotes).

Activation de notre API

Pour activer en **local** notre API (et donc l'application mynotes) et un microgateway :



- Cliquez en bas à droite sur la **roue dentelée** pour faire apparaître les logs (vides dans un premier temps) : [9]
- Cliquez en bas à gauche sur le bouton de **lecture** pour démarrer le microgateway (port 4002) et notre application mynotes (port 4001). Il est possible de visualiser les logs et le statut **Running** doit apparaître en vert : [10]

Explorer notre API

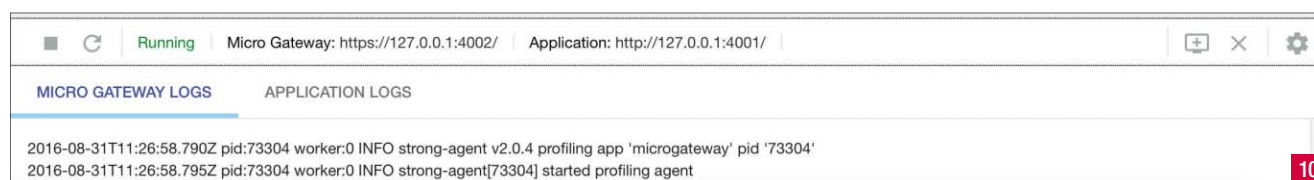
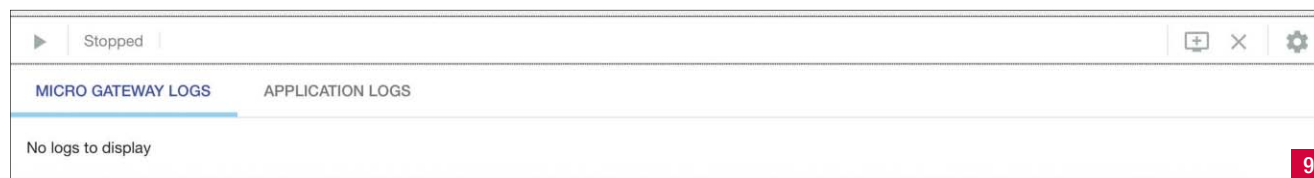
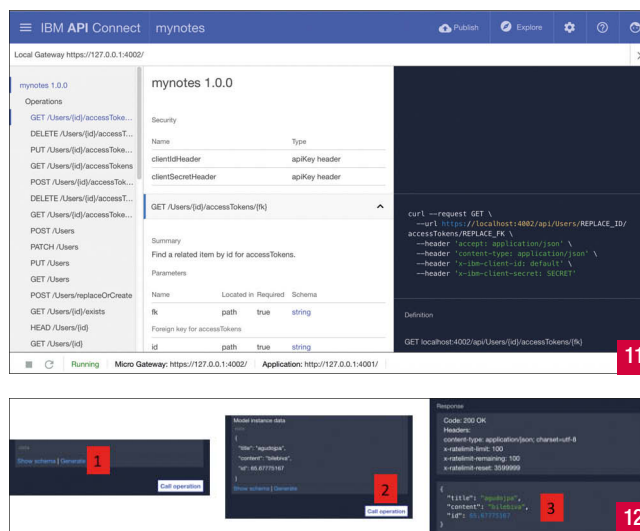
Pour découvrir et tester cette API, nous allons continuer à utiliser l'API Designer. Cliquez sur le bouton bleu **Explore** en haut à droite. [11]

Cet écran est divisé en trois parties : à gauche, les différentes opérations, au centre une information détaillée sur l'opération en cours et à droite des snippets pour tester notre API en live. Faites défiler les opérations à gauche, trouvez **POST /Notes** et cliquez dessus. Nous allons créer quelques notes qui seront stockées dans notre base de données (in-memory DB). Sur la partie à droite (fond noir), on trouve une commande **curl** et des exemples d'utilisations. Descendez doucement et cliquez sur le lien **Generate** (cela permettra de générer des données conformes au modèle), puis cliquez sur **Call Operation** : [12]

La réponse est un code 200 OK (succès). Si vous recevez un **code -1**, vous devrez exécuter le lien <https://localhost:4002/api/Notes> dans une autre tab de votre navigateur afin d'accepter les certificats.

On pourra répéter l'opération de POST (avec generate) plusieurs fois afin de créer plusieurs notes. Notre API implémente une gestion CRUD (Create, Read, Update, Delete) dans notre base de données.

Pour visualiser le nombre de notes dans la base, on pourra utiliser une opération de lecture **GET /Notes/count** que vous trouverez dans la liste à gauche. [13]



Il sera possible également de tester d'autres opérations comme la lecture d'une note en particulier ou bien toutes les notes. En utilisant **GET /Notes**, vous pourrez voir la liste de toutes les notes dans un format JSON.

CONCLUSION

Dans cet exercice, grâce à API Connect (et en particulier ce module Developer Toolkit), nous avons créé en quelques minutes une simple API de gestion de notes (CRUD) et nous avons pu la tester rapidement. Une fois créée, notre API peut ensuite être déployée depuis le poste de travail du développeur ou de l'administrateur vers un environnement API de production sur des systèmes on-premise ou dans le Cloud.

Pour développer et gérer vos Applications et vos APIs dans le Cloud, la plateforme Bluemix basée sur les standards ouverts et comprenant plus de 140 services (dont le service API Connect) est particulièrement adaptée, vous pouvez bénéficier de 30 jours gratuits sur le lien suivant : ibm.biz/BluemixFranceTrial

D'autres aspects importants de la gestion des APIs sont également couverts par **API Connect** comme : le portail des développeurs, les plans (et les produits), la sécurité au niveau des APIs (avec OAuth2 par exemple), le support des web services (SOAP), l'assemblage conditionnel de différentes APIs dans une seule API synthétique, la gestion de la scalabilité et de la haute disponibilité.

API Connect propose une approche complète et unique dans la gestion du cycle de vie des APIs de la création à l'exécution, la gestion et la sécurisation des APIs dans le respect des standards ouverts. [14]

Autres liens utiles (site, vidéo, documentations, tutoriels) :

<https://developer.ibm.com/apiconnect/>

<https://developer.ibm.com/apiconnect/doc/>

<https://developer.ibm.com/apiconnect/videos/>

Response

Code: 200 OK

Headers:

content-type: application/json; charset=utf-8

x-ratelimit-limit: 100

x-ratelimit-remaining: 98

x-ratelimit-reset: 3569595

```
{
  "count": 3
}
```

13

14

Restez connecté(e) à l'actualité !

- ▶ **L'actu** de Programmez.com : le fil d'info **quotidien**
- ▶ La **newsletter hebdo** : la synthèse des informations indispensables.
- ▶ **Agenda** : Tous les salons, barcamp et conférences.

Abonnez-vous, c'est gratuit ! www.programmez.com

Atari ST : menu et compilation



Frédéric Sagez
Codeur du groupe
NoExtra-Team
fsagez@gmail.com

Vous avez eu un ATARI ST et vous avez déjà utilisé des compilations de groupes comme FUZION, ELITE, ZUUL ou AUTOMATION, alors cet article tombe bien car il vous explique comment on fabrique ce type d'interface. Et le tout en assembleur s'il vous plait ! Aujourd'hui je vais vous expliquer comment faire une compilation d'intros tout en utilisant différents outils tel que Néochrome Master pour la partie graphique, le GFA BASIC pour générer nos données brutes telles que la fonte ainsi que l'assembleur DEVPAC pour compiler nos programmes.

Comme indiqué dans le numéro 203 de janvier, nous allons faire une incursion dans le monde de la compilation. Une compilation ou menu contient des éléments légaux et/ou illégaux, cela va du jeu ou de la version crackée (vies infinies ou déplombé), de démonstrations, ou encore de logiciels, voire de sharewares suivant le support proposé. Le menu est souvent interactif, on sélectionne l'item que l'on veut voir via la souris ou le clavier et hop c'est parti !

Comment ça marche ?

C'est assez simple : nous avons besoin d'un Loader qui va se charger d'exécuter autant de programme que l'utilisateur le désire via un Menu. En fait le Loader est agencé comme ceci : une fois lancé, il va afficher une image ou des informations, ensuite il va charger en mémoire et exécuter le Menu. Généralement bien compressé à cause de la taille du fichier, il s'exécute assez rapidement lors du chargement de celui-ci via la disquette. A partir du Menu, une fois que l'utilisateur a choisi ce qu'il veut voir, il va renvoyer en mémoire la sélection de l'utilisateur et rendre la main au Loader qui va se charger d'exécuter le programme sélectionné et ensuite de recharger le Menu une fois que l'utilisateur aura fini de regarder le programme. La sortie du Menu et le retour au bureau GEM sont prévus lorsque vous cliquez sur la touche

ESCAPE dans le cadre de notre démonstration. La figure [1] vous explique le principe du fonctionnement continu du chargement et de l'exécution des programmes.

Le LOADER

Le Loader est le fichier de chargement, il permet de charger un ou plusieurs fichiers en mémoire et de l'exécuter ensuite. Pour s'assurer que le Loader s'exécutera à chaque fois que l'on insère la disquette et qu'il sera exécuté automatiquement au démarrage, on créera un répertoire **AUTO** à la racine du lecteur A et on le copiera le fichier **LOADER.PRG** à l'intérieur. A chaque action de chargement, nous afficherons une information. Au premier démarrage de celui-ci on affichera le Logo [2] du groupe NoExtra-Team, puis pour chaque chargement de programme, nous indiquerons **LOADING** en blanc ou **ERROR** en rouge si il y a eu une erreur au chargement du fichier.

Pour charger nos programmes, nous utiliserons la fonction **Pexec0** de l'Atari ST, c'est une fonction **GEMDOS** qui permet de charger des programmes à partir de la disquette. Nous lui passons 4 paramètres via la pile mais seuls 2 paramètres sont utiles pour notre Loader : le nom du fichier à charger ainsi que le mode de chargement. Celui-ci est à zéro, soit en mode charge et exécute. Nous utilisons le code retour de cette fonction qui est mis dans le registre de

données D0 afin de vérifier s'il y a eu un problème lors du chargement d'un fichier, ceci afin d'indiquer une erreur à l'utilisateur. La gestion de l'erreur de chargement d'un fichier nous assure de toujours revenir au Menu afin de ne pas laisser l'utilisateur sur un écran final décoré de jolies bombes.

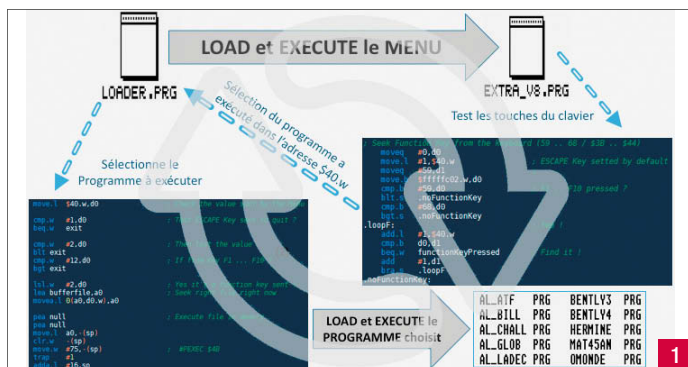
Le MENU

Le Menu est le programme qui permet de sélectionner et de choisir ce que l'on veut voir à partir de celui-ci. Dans notre cas, nous allons afficher du texte qui permettra de sélectionner des intros que voudra visionner l'utilisateur. A part la partie graphique, sonore et visuelle, chaque menu est composé généralement de textes, indiquant des informations relatives à son contenu ainsi qu'un scrolltext pour faire passer des messages à différentes personnes, groupes, etc.

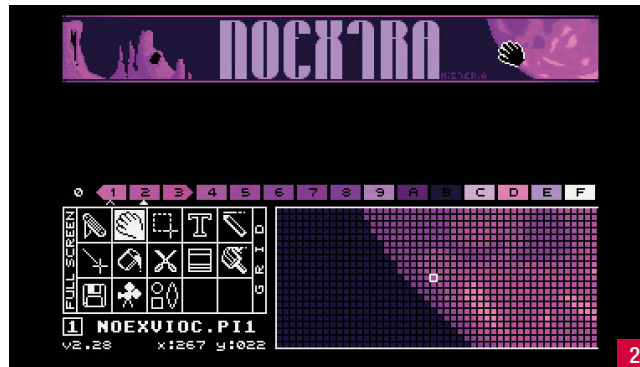
Le menu va contenir principalement :

- Un texte proposant le contenu du Menu,
- Une animation avec un effet 3D,
- Un scrolltext contenant des messages, de la pub, etc. ,
- Et de la musique : un module Amiga 4 voies sera joué en fond.

Une interaction avec les touches de Fonction de F1 à F10 du clavier du ST sera géré pour sélectionner l'introduction à visualiser. La touche Escape permettra de sortir du menu et de



Description du fonctionnement du Loader



Logo 16 couleurs en 320x40 par Mister.A

retourner sur le bureau GEM. Le Menu pourra fonctionner soit avec un Atari STF ou STE avec 512 ko de mémoire grâce à la détection de la machine et du Player de module utilisé. [3]

Pour créer le menu, nous utiliserons notre propre Framework qui se trouve à l'adresse suivante <https://github.com/NoExtra-Team/framework> et nous rajouterons nos effets d'affichage de textes, de scrolltext, effet 3d affichant un cercle sur deux plans et une musique de type Module 4 voies. Et le plus important : nous allons gérer les touches de fonctions de F1 à F10 de l'Atari ST, nous détecterons quelle touche sera appuyée pour la sélection du programme à visualiser et nous retournerons le numéro du programme à exécuter via le Loader tout en indiquant cette information en mémoire à l'adresse \$40.w.

Le code Assembleur

La fonte qui servira pour le texte et le scrolltext sera créée avec un logiciel de dessin [4]. Nous utiliserons une fonte un plan - soit une couleur - de taille 8 par 8 pixels provenant du jeu Bobble Bubble. Notre fonte commence toujours par un ESPACE et finit par la lettre minuscule 'z'. L'ESPACE correspondant au caractère hexadécimal \$20 et la lettre 'z' correspondant au caractère hexadécimal \$7A, la chaîne de caractères ainsi affichée correspondra avec les caractères standard ASCII utilisés par le ST. En fait, on fera correspondre la chaîne de caractères à afficher via les codes hexadécimaux des caractères ASCII. Exemple, pour afficher le caractère majuscule 'A', correspondant au code hexadécimal \$41, nous afficherons le 33ème caractère sur la première ligne de l'image de la fonte. (33 en décimal correspondant à 21 en hexadécimal plus 20 comme valeur de base - c'est le caractère ESPACE - qui donnera 41 en hexadécimal)

Nous utiliserons le **GFA BASIC** qui est le Basic standard sur Atari ST pour transformer notre

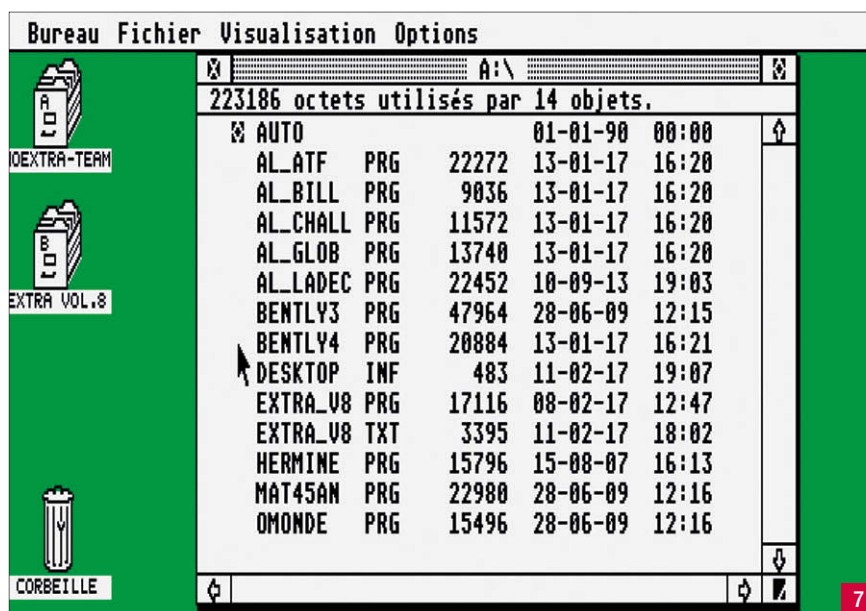
image Degas *BOBBLE.PI1* en un fichier binaire qui sera utilisé dans le Menu. Le programme *CODE8_8.LST* permet de générer un fichier de type RAW de la fonte de taille 8 pixels sur les 3 lignes de l'image Degas. [5]

Dans le code de notre menu, fichier *EXTRA_V8.S*, la routine *print_text* va servir à imprimer le texte du menu. Cette routine est dédiée aux résolutions Basse ou Moyenne nécessitant un plan pour afficher les caractères et pourra utiliser des fontes de taille 8 pixels ou 6 pixels. (La taille réelle de la fonte utilisée est de 8x8 pixels). [6]

La routine scrolltext utilise aussi la fonte générée par le code écrit en GFA-BASIC. Tout comme l'affichage du texte, nous prenons le code hexadécimal du caractère, on le multiplie par 8 - taille de la fonte en largeur - puis on le soustrait à 256 pour pointer directement sur le caractère au format binaire issu des données sur l'écran final. Par contre pour le scrolltext nous passerons par un buffer pour gérer le

décalage à gauche de notre chaîne de caractères puis nous l'afficherons sur l'écran physique à la position et au plan désiré. Concernant la routine 3d, nous afficherons un peu plus de 400 points sous la forme d'une sphère dont nous allons calculer la courbe et la forme dans l'initialisation via la routine *init_sphere*. Suivant la position du point et suivant l'axe des X, nous afficherons des points de couleurs bleu et violet pour indiquer le fond de la sphère afin de lui donner de la profondeur via la routine *play_sphere_3d*.

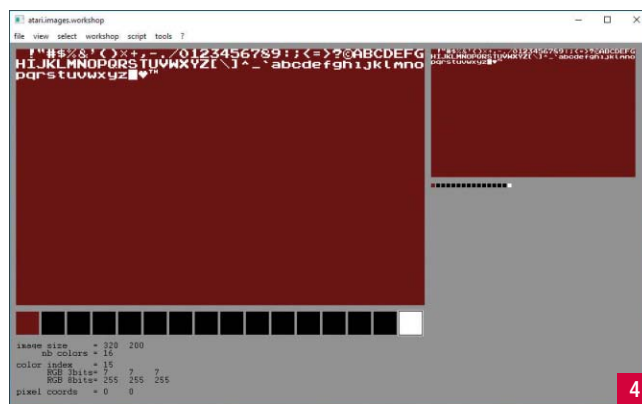
Au niveau sonore, nous utiliserons une routine qui permet de rejouer des modules 4 voies via le TIMER A et avec le TIMER D suivant le type de l'Atari ST détecté par la routine *Play_Module*. La routine Soundtrack du groupe **WizzCat/Delta Force** permet de jouer soit sur un Atari STF via le composant YM2149 sur 3 voies Mono ou sur Atari STE via le composant DMA permettant une meilleure sonorité : 4 voies Stéréo et une meilleure gestion des



Contenu de la Disquette



Menu de la compilation EXTRA



Fonte 8x8 pixels 1 plan du jeu Bubble Bobble


```

691 *****
692 ***      DISPLAY TEXT FONT 8x8 or 6x6      ***
693 ***      MED OR LOW RESOLUTION - ONE BITPLANE ***
694 ***      ZOPR02/NOEXTRA-TEAM      ***
695 *****
696 CHARS      EQU 40      ; chars per line, 80=for med res, 40 for low res *
697 LINES      EQU 33      ; 33 for 8x8 font, 45 with 6x6 font *
698 FONTSIZE    EQU 8      ; 8=8x8, 6=6x6 font *
699 SHIFTSIZE   EQU 4      ; 2=MED RESOLUTION, 4=LOW RESOLUTION *
700 POS_SCR     EQU 160*46 ; Top Position in the Screens *
701 PLAN_SCR    EQU 4      ; Bitplane used to display the text *
702 *****
703 print_text: lea      message,a2      ; Read the text or hexa
704 new_char:   bsr      _x_conversion   ; Convert / Resolution
705             moveq    #0,d0
706             move.b    (a2)+,d0      ; If zero, stop routine
707             cmp.b     #0,d0
708             beq       line_feed     ; Next line / RC
709             cmpi.b    $fd,d0      ; Bitplane + 0
710             bne.s     .test_plan_1
711             move.w    #0,pointeur_plan
712             bra.s     new_char
713             bra.s     .end_of_line
714             cmpi.b    $fc,d0      ; Bitplane + 2
715             bne.s     .end_of_line
716             move.w    #2,pointeur_plan
717             bra.s     new_char
718             cmpi.b    $ff,d0      ; End of the text ?
719             bne.s     process_char
720             rts
721
722 process_char: asl.w     #3,d0      ; Find ASCII value
723             lea      fonts,a1      ; and choose character font
724             sub.w    #256,d0
725             adda.w    d0,a1
726             moveq    #0,d1      ; Position of the letter
727             add.w     pointeur_plan,d1 ; in the two Screens
728             add.l     y_offset,d1
729             add.l     x_offset,d1
730             movea.l    physique,a0 ; In First screen
731             lea      POS_SCR+PLAN_SCR(a0),a0
732             adda.l     d1,a0
733             movea.l    physique+4,a3 ; In Second screen
734             lea      POS_SCR+PLAN_SCR(a3),a3
735             adda.l     d1,a3
736             rept     FONTSIZE      ; Print letter on Screens
737             move.b    (a1),(a0)
738             move.b    (a1)+,(a3)
739             lea      160(a0),a0
740             lea      160(a3),a3
741             endr
742             addq.w    #1,x_curs
743             cmpi.w    #CHARS,x_curs ; 79 for MED res
744             bils     new_char
745             move.w    #CHARS,x_curs ; 79 for MED res
746             bra      new_char
747
748 line_feed:   clr.w     x_curs      ; back to first char
749             addi.l    #FONTSIZE*160+160,y_offset ; when u reached ',0'
750             cmpi.l    #LINES*FONTSIZE*160,y_offset
751             bils     new_char
752             move.l    #LINES*FONTSIZE*160,y_offset
753             bra      new_char

```

Le code source en assembleur 68K

```

1  REM -----
2  REM --- Encodage fonte 8x8 pixels in one bitplane ---
3  REM -----
4  ' Initialisations
5  RESERVE 10000
6  buffer%=MALLOC(2*8*16*3)
7  buff%=buffer%
8  width%=8
9  height%=8
10 ' Charge le dessin de la fonte (PI1)
11 CLS
12 FILESELECT #"PICTURE", "C:\MENU\font\*.PI1", "", fontpic$
13 BLOAD fontpic$,XBIO$(2)-34
14 ~XBIO$(6,L:XBIO$(2)-32)
15 ' Encodage de la fonte
16 ' --- Ligne 1 ---
17 FOR n%=0 TO 152 STEP width%
18   adr%=XBIO$(2)+n%
19   adr2%=adr%+1
20   @encodage
21 NEXT n%
22 ' --- Ligne 2 ---
23 FOR n%=0 TO 152 STEP width%
24   adr%=XBIO$(2)+height%*160+n%
25   adr2%=adr%+1
26   @encodage
27 NEXT n%
28 ' --- Ligne 3 ---
29 FOR n%=0 TO 152 STEP width%
30   adr%=XBIO$(2)+2*height%*160+n%
31   adr2%=adr%+1
32   @encodage
33 NEXT n%
34 ' Sauvegarde de la fonte
35 BSAVE "C:\MENU\font\FONT8_8.DAT",buffer%,2*8*16*3
36 ~MFREE(buffer%)
37 EDIT
38 ' Procédure Encodage
39 PROCEDURE encodage
40   FOR t%=0 TO height%-1
41     POKE buf%,PEEK(adr%)
42     ADD buf%,1
43     ADD adr%,160
44   NEXT t%
45   FOR t%=0 TO height%-1
46     POKE buf%,PEEK(adr2%)
47     ADD buf%,1
48     ADD adr2%,160
49   NEXT t%
50 RETURN

```

Génération de la fonte au format binaire

effets. Vous pouvez télécharger des modules via les sites **Mod Archive** (<https://modarchive.org/>) ou **Amiga Music Preservation** (<http://amp.das-cene.net/>) si vous voulez utiliser votre Soundtrack préféré.

Les Programmes

Assurez-vous bien que les programmes utilisés rendent toujours bien la main une fois exécutés sous le **GEM**. (Portant l'extension « .PRG ») Pour un meilleur chargement via la disquette, n'hésitez pas à les compresser avec un outil tel qu'**UPX**, c'est un compresseur puissant multi-plateforme qui compresse bien les exécutables Atari comme sur PC. (<https://upx.github.io/>)

Dans le menu que je vous présente, huitième édition déjà ! Il sera composé d'introductions faites pour la Team *Atari Legend* lors de la diffusion de jeux sur Atari ST via leur site Internet <http://www.atarilegend.com/>.

Comme on le voit dans [7], le répertoire **AUTO** contient le Loader qui sera exécuté au boot de la disquette ainsi que tous les pro-

grammes qui seront appelés par le Menu qui est le programme *EXTRA_V8.PRG*.

Les Sources

Tous les sources sont disponibles sur le **Github** de NoExtra-Team à l'adresse suivante : <https://github.com/NoExtra-Team/framework/sources/menu>. Il contient 3 répertoires :

- Le répertoire /font contient les données pour encoder la fonte (routines écrites en GFA-BASIC plus le fichier Degas de la fonte utilisée) ;
- Le répertoire /intros contient toutes les introductions de jeux contenues dans la compilation ;
- Le répertoire /loader contient tout le code source du LOADER (routine assembleur plus les datas graphiques) ;
- Le répertoire /menu contient tous les éléments pour compiler le menu (routines assembleurs - core + les effets + replayer de module - avec les datas).

Ressources

Dead Hacker Society, le site Internet de la démoscène Atari : <http://dhs.nu/>

Plein de sources assembleurs tout azimut sur <http://freddo.chez.com/Sources/Sources.html>

Pour les Cours en assembleur 68000, je vous conseille ceux fait par Féroce Lapin. Ils sont visibles en ligne sur <http://ataniste.free.fr/asm/assembleur.html> ou consultable au format PDF sur http://sasfepu78.fr/articles/Cours_asm_68000_Feroce%20Lapin.pdf

Le référentiel Github de sources de groupes connus sur Atari ST : https://github.com/ggnkua/Atari_ST_Sources

Sinon tous les projets Atari sont référencés sur le site The Orphaned Projects Page sur <http://www.ataricq.org/>.

Petit oubli de ma part dans mon dernier article : les fichiers musicaux de type Soundchip sont disponibles sur le site YM2149 Archive de *Phil Graham* à l'adresse suivante : <http://sndh.atari.org/>. Le site Web regorge en effet de toutes les musiques 3 voies de jeux ou de démos spécifiques au composant Yamaha de l'Atari ST. •

Module HM-10 : du Bluetooth LE pour votre Arduino !

• Renaud Pinon

Suite et presque fin de notre article concernant le module HM-10 qui apporte le Bluetooth Low Energy à vos montages électroniques.

Petit retour dans le passé : dans la première partie, nous avons vu comment souder et brancher un module *HM-10* (ou sa contrefaçon très répandue : le *CC41*) à un *Arduino*, et comment lui envoyer les commandes « *AT* » permettant de changer son nom. Pour rappel, nous utilisons un programme *Arduino* qui ne fait que transférer les commandes envoyées par le moniteur série d'*Arduino IDE* vers le module *Bluetooth*. La principale subtilité est que pour un module *HM-10*, les commandes n'ont pas de séparateur, alors que pour le *CC41* chaque commande se termine obligatoirement par un retour chariot suivi d'une nouvelle ligne (*\r\n*).

Le code *Arduino* et le projet *Xcode* utilisés dans cet article peuvent être trouvés à l'adresse suivante : <https://github.com/renaudpinon/HM10>.

Mode *iBeacon*

Continuons nos aventures palpitantes parmi les commandes *AT* avec le mode *iBeacon*. C'est un mode de fonctionnement autonome qui permet de connaître la distance entre un module *HM-10* et votre *smartphone*. Dans ce mode, vous ne pourrez en aucun cas échanger de données avec le module mais vous pourrez, en branchant uniquement ses bornes *VCC* et *GND* à une pile ou batterie 3,7V, l'accrocher à vos clés de voiture afin de ne plus jamais avoir à les chercher un quart d'heure chaque matin. Un autre usage pourrait être, si on considère un musée, une application spécifique qui, pour chaque visiteur s'approchant à moins de deux mètres d'un tableau, affiche sur son téléphone une description qui lui est propre. Pour configurer notre module en mode *iBeacon*, nous allons envoyer successivement les commandes suivantes. Ne cherchez pas la logique, cela ne s'invente pas :)

HM-10	CC41	Commentaire
AT+RENEW	AT+RENEW	Retour paramètres usine
AT+RESET	AT+RESET	Reboot du module
AT	AT	OK doit apparaître
AT+MARJ=0x1111	AT+MARJ0x1111	Major number (non il n'y a pas de faute de frappe). Remplacez 0x1111 par la valeur voulue.
AT+MINO=0xBA34	AT+MINO0xBA34	Minor number. Remplacez 0xBA34 par la valeur voulue.
AT+ADVI=5	AT+ADVI5	Intervalle de rafraîchissement de publication (en 10 ^{ième} de sec, soit ici 500 ms)
AT+NAME=Programmez0001	AT+NAMEProgrammez0001	Nouveau nom
AT+ADTY=3	(n'existe pas. Ignorer)	Rend le module non connectable
AT+IBEA=1	AT+IBEA1	Active le mode beacon
AT+DELO=2	(n'existe pas. Ignorer)	Mode diffusion simple
AT+PWRM=0	AT+PWRM0	Active la veille automatique
AT+RESET	AT+RESET	Reboot du module

Les trois commandes *AT+DELO*, *AT+ADVI* et *AT+PWRM* vous permet-

tent de modifier la consommation d'énergie du module : grâce à cela, la batterie de votre *beacon* tiendra infiniment plus longtemps (attention : *AT+DELO* n'existe pas sur *CC41*). La commande *AT+IBEA* est celle qui active à proprement parler le mode *iBeacon*.

La détection du *beacon* fonctionne grâce à un identifiant utilisateur unique (*UUID*) codé sur 16 octets et donc représenté par 32 caractères hexadécimaux. Cet identifiant, le même pour tous vos *beacons* et les différencie de ceux appartenant aux autres personnes. Pour l'exemple, considérons un magasin qui aurait un *beacon* pour chacun de ses produits : tous les *beacons* de notre magasin auront exactement le même *UUID*. Par contre, ceux du magasin voisin et concurrent du nôtre en auront un autre. La distinction entre chacun des modules d'un même magasin se fera grâce aux nombres *Major* et *Minor*. Ce sont deux nombres encodés chacun sur 2 octets : *Major* sert à identifier une catégorie de *beacon* tandis que *Minor* identifie un *beacon* parmi ceux de cette catégorie. Pour reprendre l'exemple de notre magasin, tous les produits alimentaires auront le même numéro *Major*, mais chaque produit alimentaire aura son propre numéro *Minor*. Notez qu'en plus de pouvoir modifier le nombre *Major* et *Minor*, les commandes *AT* vous permettent également de lire et modifier l'*UUID* (commandes *AT+IBEO*, 1, 2 et 3). Un *UUID* est déjà affecté par défaut mais rien ne vous empêche de le modifier à votre guise... ne rentrez juste pas le même que celui du voisin ;)

Votre module est donc maintenant configuré pour le mode *iBeacon*. Encore une fois, si vous le souhaitez, vous pouvez rendre votre module totalement autonome en débranchant tous les câbles le reliant à l'*Arduino* et en raccordant uniquement son *VCC* et *GND* à une batterie 3,7 V. Allumez votre *smartphone* (*iOS* ou *Android*) et téléchargez gratuitement une application quelconque sur le *Store* de votre appareil qui permette de trouver les objets *iBeacon*. Je ne veux pas faire de pub mais, sur *iPhone*, celle du nom de *Dartle* a plutôt bien fonctionné pour moi (seul le nom du module détecté était folklorique). Si tout va bien vous devriez voir votre module *Programmez0001* dans la liste et sa distance par rapport à votre téléphone. Sinon, suivant votre application, il faudra peut-être l'ajouter manuellement en indiquant l'*UUID* (les 4 résultats formés par *AT+IBEO*, 1, 2 et 3), ainsi que le *Major* et le *Minor*. Attention : ces deux derniers sont souvent demandés en décimal, il faudra donc probablement que vous fassiez la conversion à partir de l'hexadécimal.

Envoi et réception de données depuis un iPhone.

Passons désormais totalement à autre chose, à savoir l'envoi et réception de données entre un *smartphone* et un module *HM-10*. Nous allons créer une application en *Objective-C* afin de faire communiquer notre module avec un *iPhone*. Nous allons d'abord rebrancher (si nécessaire) notre *HM-10* à l'*Arduino*, désactiver le mode *iBeacon* en le réinitialisant avec ses paramètres d'origine, puis le renommer et enfin le rebooter. Pour cela, envoyez-donc dans l'ordre les commandes *AT* suivantes :

HM-10	CC41
AT+RENEW	AT+RENEW
AT+NAME=Programmez0001	AT+NAMEProgrammez0001
AT+RESET	AT+RESET

a. Modèle GATT

Le *Bluetooth LE* fonctionne suivant le modèle *GATT* - pour *Generic ATributes* - qui permet de décrire le « profil » du matériel, les « services » qu'il propose et les « caractéristiques » propres à chacun de ses services. Par exemple, un bracelet de fitness serait décrit comme un profil « sport et fitness », avec un service de « vitesse et cadence de course », comprenant lui-même les deux caractéristiques « localisation » (= coordonnées GPS actuelles) et « point de contrôle » (= vitesse, calories, élévation, ... à un instant T). Chacun de ses attributs (profil, service ou caractéristique) est identifié par un nombre entier codé sur 2 octets et dont la liste pré-définie se trouve sur le site officiel du *Bluetooth* : <https://www.bluetooth.com/specifications/assigned-numbers>.

Voici un schéma résumant le fonctionnement du *GATT* : [1]

Maintenant que vous avez bien en tête ce principe de multi-services ayant chacun plusieurs caractéristiques, vous pouvez... l'oublier. Car en fait le *HM-10* ne supporte qu'un seul service et une seule caractéristique à la fois, mais qu'il est évidemment possible de configurer. Pour obtenir/modifier le numéro du service, il s'agit de la commande *AT+UUID* (attention: rien à voir avec l'*UUID* du mode *iBeacon* !) et pour le numéro de la caractéristique c'est *AT+CHAR*. Leurs valeurs par défaut pour le *HM-10* sont respectivement *0xFFE0* et *0xFFE1*.

b. Code Arduino

Bonne nouvelle, le montage électronique avec notre Arduino reste le même. Par contre le programme interne change. Le voici (et notez qu'il fonctionnera que vous ayez un *HM-10* ou un *CC41*) :

```
// Fichier HM10_utilisation.ino
```

```
#include <SoftwareSerial.h>
```

```
// Defines:
```

```
#define kBTPinRx 10
```

```
#define kBTPinTx 11
```

```
// Variables globales:
SoftwareSerial _btSerial(kBTPinRx, kBTPinTx);
String _btBuffer = "";

// Initialisation:
void setup()
{
  _btSerial.begin(9600);
  while (!_btSerial) { /* attente interface dispo */ }

  Serial.begin(9600);
  while (!Serial) { /* attente interface dispo */ }
}

// Fonction principale :
void loop()
{
  // Réception des données du smartphone vers HM-10 :
  while (_btSerial.available() > 0)
  {
    char letter = (char)_btSerial.read();
    if (letter != '\r' && letter != '\n')
    {
      _btBuffer += String(letter);
    }
    else if (letter == '\n')
    {
      // Fin de commande (\n = séparateur de commandes):
      Serial.println(_btBuffer); // pour le debug.

      // TODO: on peut ici interpréter les données
      // et faire agir l'Arduino en conséquence ou
      // afficher la chaîne reçue sur un écran LCD par exemple.

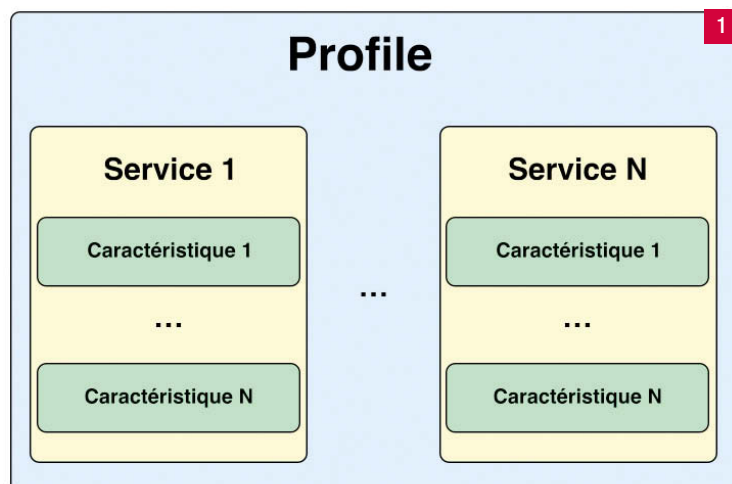
      _btSerial.println("Bien reçu ! »); // accusé de réception.
      _btBuffer = ""; // réinitialisation du buffer.
    }
  }
}
```

Dans ce code, nous attendons simplement de recevoir des données par le port série du module *Bluetooth* et lorsque c'est le cas, nous mettons les données dans un *buffer*. Si le caractère reçu est un retour chariot, nous estimons qu'il s'agit d'un séparateur de commandes et nous affichons ce que nous avons reçu dans le moniteur série d'*Arduino IDE*. Evidemment nous pouvons à ce stade faire interpréter par notre *Arduino* les données reçues et en faire ce que nous voulons : ouvrir une porte à l'aide d'un moteur, faire tourner notre robot, allumer une LED, afficher le texte reçu sur un écran LCD, etc.

c. Code iOS

En ce qui concerne le code *iOS* de notre application, nous ne listerons pas tout ici car il y en aurait pour plusieurs pages. Vous pouvez toutefois trouver le projet *Xcode* complet à l'adresse : <https://github.com/renaudpinon/HM10>. Pour ne pas être perdu, je vous conseille de lire les explications avec le projet ouvert à côté.

Suite et fin le mois prochain



Atari 800 XL : les derniers Atari 8-bit

- *Timeline : 1983*
- *Objet : ordinateur*
- *Constructeur : Atari*

Vous avez peut-être débuté dans le monde informatique avec les fameux Atari ST, une des machines les plus mythiques des années 80. Mais on oublie souvent, et moi, le premier, qu'Atari avait sorti plusieurs gammes d'ordinateurs personnels, ou des consoles, que l'on appelle souvent la famille des Atari 8-bits. Souvent, on ne parle pas d'ordinateurs mais de consoles pour les désigner.

Atari avait de grandes ambitions dans l'informatique personnelle. Et c'est en 1979 que le constructeur va sortir sa première gamme de machines 8-bit : les modèles 400 et 800. Ils utilisent, comme sur le reste de la gamme, un processeur MOS 6502 1,77 MHz. Ce sont des machines qui se veulent Plug&Play grâce à un port d'extension : le SIO, un bus série. La machine était chère à cause d'une structure en aluminium. Ce design évoluera par la suite. Pour les programmes, on peut utiliser un lecteur de cassettes (lecteur externe), des cartouches et des disquettes (lecteur externe). Le clavier du 400 est un clavier à membrane, peu pratique, comme sur le ZX81 et l'Oric. La frappe est assez dure. Le clavier mécanique est adopté pour le 800. Une trappe sur le dessus permettait d'accéder au port cartouche (2 ports sur le 800) et aux extensions (RAM, ROM, CPU).

Début 1983, une nouvelle ligne d'ordinateurs apparaît : les XL. Ils héritent d'un nouveau design qui peut ressembler à un Apple II. Il est plus plat, avec un clavier mécanique, plusieurs touches de fonctions et une connectique redessinée. Le premier modèle est le 1200XL, mais son prix élevé l'handicape alors que les 400 et 800 continuent à être vendus. Le 1200XL va rapidement être retiré de la vente et deux modèles vont le remplacer : les 400 et 800XL.

La guerre avec Commodore : le rôle des modèles XL

Atari affronte une vive concurrence avec Commodore et Texas Instruments, avec son modèle TI99 et TI99/A. Atari travaillait sur plusieurs projets et particulièrement sur un nouveau design : le LIZ NY. L'objectif était de créer un ordinateur peu cher. Ce projet ne sera



pas finalisé. Cependant, la gamme XL reprendra les grandes lignes du LIZ NY.

Un modèle devait reprendre tel quel le LIZ NY : le 600XL. Ce modèle apparaît dès le printemps 1983. Très rapidement, le constructeur sort le 800 XL. Le 800 XL est une évolution du 600 XL avec une quantité de mémoire supérieure : 64 Ko (24 Ko de ROM). Ces modèles étaient capables de faire fonctionner la plupart des programmes des 400 et 800 même si un traducteur était nécessaire. Il possédait un langage de programmation : Atari Basic, accessible dès le démarrage (touche option, sans cartouche connectée). Une évolution du 800 XL inclura un nouveau gestionnaire de mémoire (puce Freddy). Il fonctionne avec un système DOS (2, 2.5 ou 3). Le Basic était directement intégré à la ROM contrairement aux modèles précédents nécessitant une cartouche dédiée. L'Atari Basic est directement interprété. Il avait été conçu par Shepardson Microsystems. Trois grandes versions ont été distribuées : Revision A, Revision B et Revision C. Les version B et C étaient shipées pour les XL. La version C fixait des problèmes de mémoires et se retrouvera tardivement dans les XL et les modèles XE. L'Atari Basic autorisait l'accès matériel.

Autre nouveauté du 800XL, un bus parallèle haute vitesse (PBI), permettant par exemple de connecter un boîtier d'extension (Atari 1090)

autorisant jusqu'à 5 cartes d'extensions ! Une petite révolution pour le grand public.

Un 3e modèle était prévu, mais il ne fut jamais commercialisé : le 1450 XLD avec lecteur de disquettes.

Le rachat

Jack Tramiel, fondateur de Commodore en 1954, va racheter Atari (sic!) en 1984, quelques mois auparavant, Tramiel est obligé de quitter le constructeur. Une première gamme sort en 1985 : les XE. Ils sortent en même temps que les Atari ST, au design si reconnaissable, mais avec des différences notamment sur le clavier, le processeur, le lecteur de disquettes (absent). Les XE sont des modèles améliorés des XL et restent des machines 8-bit contrairement aux ST. Les XE embarquent plus de mémoire vive et le nouveau design (celui du ST). La gamme est compatible avec les cartouches des XL.

Les Atari 8-bits se vendront à plusieurs millions d'exemplaires entre 1979 et 1985. Mais la concurrence très forte dès 1980/81 fait mal à Atari qui cherche à se relancer avec la gamme XL. C'est finalement le fondateur de Commodore, quelle ironie de l'histoire, qui va relancer la machine Atari avec les mythiques ST et passer aux plateformes 16 bits et aux processeurs 68000.

Mais ceci est une autre histoire. •

Tous les numéros de programmez! le magazine des développeurs

sur une clé USB (depuis le n°100).



34,99 €*

Clé USB 4 Go.
Photo non contractuelle.
Testé sur Linux, OS X, Windows. Les magazines sont au format PDF.

* tarif pour l'Europe uniquement.
Pour les autres pays, voir la boutique en ligne

Commandez la directement sur notre site internet : www.programmez.com

Complétez votre collection

Prix unitaire : 6,50 €



- | | |
|---|---|
| <input type="checkbox"/> 191 : <input type="text"/> exemplaire(s) | <input type="checkbox"/> 201 : <input type="text"/> exemplaire(s) |
| <input type="checkbox"/> 194 : <input type="text"/> exemplaire(s) | <input type="checkbox"/> 202 : <input type="text"/> exemplaire(s) |
| <input type="checkbox"/> 196 : <input type="text"/> exemplaire(s) | <input type="checkbox"/> 203 : <input type="text"/> exemplaire(s) |
| <input type="checkbox"/> 197 : <input type="text"/> exemplaire(s) | <input type="checkbox"/> 204 : <input type="text"/> exemplaire(s) |
| <input type="checkbox"/> 200 : <input type="text"/> exemplaire(s) | <input type="checkbox"/> 205 : <input type="text"/> exemplaire(s) |

soit exemplaires x 6,50 € = € soit au **TOTAL** = €

Commande à envoyer à :
Programmez!

7, avenue Roger Chambonnet - 91220 Brétigny sur Orge

Prix unitaire : 6,50 € (Frais postaux inclus)

☐ M. ☐ Mme ☐ Mlle Entreprise : Fonction :

Prénom : Nom :

Adresse :

Code postal : Ville :

Règlement par chèque à l'ordre de Programmez !

Un de plus



CommitStrip.com



Une publication **Nefer-IT**, 7 avenue Roger Chambonnet, 91220 Brétigny sur Orge - redaction@programmez.com

Tél. : 01 60 85 39 96 - Directeur de la publication & Rédacteur en chef : François Tonic

Secrétaire de rédaction : Olivier Pavie

Ont collaboré à ce numéro : S. Saurel

Nos experts techniques : F. Maury, R. Gaborieau, J. Brancourt, S. Buisson, F. Didot, V. Loquet, S. Boinot, M. Carol, T.

Ranise, A-C Baudriller, C. Ezvan, M. Hubert, M. Pemin, E. Deneuve, J. Petitcolas, G. Renaudin, O. Lahdioui, Y. Chabeb, J-M Heitz, P. Thomas, F. Sagez

Couverture : © jgroup - Amazon - Google - François Tonic - Maquette : Pierre Sandré

Publicité : PC Presse, Tél. : 01 74 70 16 30, Fax : 01 40 90 70 81 - pub@programmez.com

Imprimeur : S.A. Corelio Nevada Printing, 30 allée de la recherche, 1070 Bruxelles, Belgique

Marketing et promotion des ventes : Agence BOCONSEIL - Analyse Media Etude - Directeur : Otto BORSCHA oborscha@boconseilame.fr

Responsable titre : Terry MATTARD Téléphone : 09 67 32 09 34

Contacts : Rédacteur en chef : ftonic@programmez.com - Rédaction : redaction@programmez.com - Webmaster : webmaster@programmez.com -

Publicité : pub@programmez.com - Evenements / agenda : redaction@programmez.com

Dépôt légal : à parution - Commission paritaire : 1220K78366 - ISSN : 1627-0908 - © NEFER-IT / Programmez, avril 2017

Toute reproduction intégrale ou partielle est interdite sans accord des auteurs et du directeur de la publication.

Encart, catalogue Pcsot jeté sur une partie du tirage abonné du magazine Programmez!

Abonnement : Service Abonnements PROGRAMMEZ, 4 Rue de Mouchy, 60438 Noailles Cedex. - Tél. : 01 55 56 70 55 - abonnements.programmez@groupe-gli.com - Fax : 01 55 56 70 91 - du lundi au jeudi de 9h30 à 12h30 et de 13h30 à 17h00, le vendredi de 9h00 à 12h00 et de 14h00 à 16h30. **Tarifs** abonnement (magazine seul) : 1 an - 11 numéros France métropolitaine : 49 € - Etudiant : 39 € CEE et Suisse : 55,82 € - Algérie, Maroc, Tunisie : 59,89 € Canada : 68,36 € - Tom : 83,65 € - Dom : 66,82 € - Autres pays : nous consulter.

PDF : 35 € (monde entier) souscription sur www.programmez.com



Sur abonnement ou en kiosque

Le magazine des pros de l'IT

Mais aussi sur le web



Ou encore sur votre tablette

L'INFORMATICIEN

ikoula
HÉBERGEUR CLOUD

PRÉSENTE

CLOUD **IKOULA** ONE



Le succès est votre prochaine destination

MIAMI SINGAPOUR PARIS
AMSTERDAM FRANCFORT _ _ _

CLOUD **IKOULA** ONE est une solution de Cloud public, privé et hybride qui vous permet de déployer **en 1 clic et en moins de 30 secondes** des machines virtuelles à travers le monde sur des infrastructures SSD haute performance.



www.ikoula.com



sales@ikoula.com



01 84 01 02 50

ikoula
HÉBERGEUR CLOUD



NOM DE DOMAINE | HÉBERGEMENT WEB | SERVEUR VPS | SERVEUR DÉDIÉ | CLOUD PUBLIC | MESSAGERIE | STOCKAGE | CERTIFICATS SSL