# Rendre l'auto-hébergement facile et sans douleur

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 20 mars 2013

https://www.bortzmeyer.org/presence-en-ligne.html

---

Aujourd'hui, des millions d'utilisateurs innocents sous-traitent leur **présence en ligne** à des gros industriels du "*cloud*" comme Google ou Facebook. Cela, au détriment de leur vie privée et au prix d'une complète perte de contrôle de leurs propres données. L'alternative libre est évidemment l'**auto-hébergement**, avoir un jeu d'applications qui mettent en œuvre cette présence en ligne, sur une machine qu'on contrôle. Mais tout le monde n'a pas forcément la compétence, ou tout simplement le temps ou l'envie, pour gérer cette machine et ces applications. N'est-il pas temps de développer un système tout fait pour cela ?

Les dangers de l'hébergement infonuagique ont déjà été décrits plein de fois. Par Richard Stallman <http://www.framablog.org/index.php/post/2011/01/11/stallman-google-chromeos-cloudcomputin John Harris <http://www.guardian.co.uk/commentisfree/2011/apr/25/hackers-spooks-cloud-antiaut et d'autres. Bruce Schneier a bien noté que l'hébergement en nuage était un retour à la féodalité <http://www.wired.com/opinion/2012/11/feudal-security> (on abandonne sa liberté en échange d'un service hypothétique). Avec l'hébergement dans le nuage, on peut être coupé brutalement des autres <http://seenthis.net/messages/122264>, on peut ne plus avoir accès à ses propres données <http://korben.info/google-facebook-proprietaire-donnees.html> alors que des polices étrangères le peuvent <http://yro.slashdot.org/story/12/12/05/0632258/researchers-patric Je ne vais pas revenir sur ces dangers : si vous n'êtes pas convaincu, si vous croyez que Google est une association humanitaire, et que Facebook a été créé pour aider bénévolement, lisez d'abord les articles cités plus haut. (Si c'est gratuit, c'est parce que vous n'êtes pas le client, mais la marchandise.)

Mais pourquoi les gens utilisent-ils massivement ces services, alors ? Parce qu'il n'y a pas d'alternative ? Ou bien celles-ci ont-elles des défauts sérieux ? L'alternative la plus souvent proposé par les « nébuloclastes » (les anti-cloud) est l'**auto-hébergement** : on prend une machine chez soi, ou bien hébergée chez un professionnel, on y met les logiciels nécessaires (il en existe plein, souvent sous une licence libre) et on a son petit nuage à soi, échappant aux problèmes notés plus haut. (Une petite nuance au passage : si la machine est chez un hébergeur - ce qui se nomme l'IaaS, celui-ci a quand même un certain contrôle sur votre présence en ligne, mais nettement moins direct que s'il manipule directement les données, comme il le fait avec le SaaS.)

Mais alors pourquoi tout le monde ne s'auto-héberge pas ? Une première raison est que cette solution nécessite certaines compétences techniques. Installer et gérer un MTA, pour pouvoir faire du courrier, et avec son identité indépendante du fournisseur, est plus compliqué que d'installer un nouveau navigateur Web. Mais il y a une autre raison derrière : même si on a les compétences nécessaires, installer, configurer et ensuite gérer ces services est très chronophage. Même un ingénieur système expérimenté a peut-être envie de consacrer son temps libre à autre chose.

D'où la vision que je voudrais proposer ici : ce qu'il manque pour développer l'auto-hébergement, et remplacer les méchants géants du nuage, c'est un paquetage tout fait, qui s'installe facilement et rapidement, et permet à chacun d'avoir une présence en ligne qu'il ou elle contrôle. Le reste de cet article est consacré à quelques détails pratiques.

Un tel paquetage devrait être sous forme d'une image qui s'installerait telle quelle sur une machine dédiée (les autres solutions sont moins simples, car elles doivent coexister avec les logiciels existants sur une machine). Les logiciels disponibles (mais pas forcément activés par défaut) devraient être :

— Un serveur de courrier par exemple Postfix (et Courier pour le IMAP),
— Un serveur de messagerie instantanée par exemple ejabberd,
— Un serveur de fichiers par exemple ownCloud ou SparkleShare <http://sparkleshare.org/>,
— Un serveur de calendrier, de contacts, etc (ownCloud, là aussi ?),
— Un logiciel de partage de fichiers (BitTorrent),
— Un serveur DNS ? Ce n'est pas évident, j'en discute plus loin,
— Un serveur Web permettant d'héberger des services comme webfinger (RFC 7033 [1]), un blog, un remplacement à Google Reader <http://www.techdirt.com/articles/20130313/17262322315/killing-google-reader-highlights-risk-relying-single-provider.shtml> comme NewsBlur <http://www.newsblur.com/>, etc,
— Des trucs d'échange comme StatusNet ?
— Les services dont ont besoin les services ci-dessus (par exemple un SGBD pour le blog), mais sans que l'utilisateur ait besoin de les voir ou de savoir qu'ils sont là,
— Et bien sûr une interface d'administration pour gérer le tout depuis un navigateur.

À mon avis, l'interface d'administration devrait être réduite : une part de configuration (indiquer le nom de domaine, les services qu'on souhaite activer, etc) et une de surveillance et de statistiques.

Est-ce réaliste, comme cahier des charges ? Les composants existent déjà presque tous et n'auraient pas besoin d'être modifiés. Le travail serait d'intégration et de développement de l'interface Web d'administration. Je n'ai pas le temps de m'y lancer mais je suggère l'idée aux développeurs ambitieux. Lorsque l'objectif est d'avoir un truc simple, la difficulté est la **qualité** : il faut du zéro bogue. Il existe déjà plein de logiciels « auto-hébergement » et qui sont faits à 95 % ce qui n'est pas suffisant. Une partie du succès de Google vient de sa qualité (pas d'erreur 500 en cours de navigation...)

Un problème difficile sera celui de la sécurité. Il y a la sécurité des applications (le logiciel de blog écrit en PHP qu'il faut mettre à jour sinon on est craqué et on se retrouve à héberger un site de hameçonnage ; ou bien l'interface Web ayant un mot de passe par défaut, et 99 % des utilisateurs qui ne le changeront pas). Et il y a le problème plus général de la responsabilité : être administrateur système, ce n'est pas uniquement de la technique, c'est de la responsabilité, suivre les problèmes, répondre aux messages envoyés à abuse <https://www.bortzmeyer.org/abuse-ne-repond-pas.html>, etc. Il ne faudrait pas que des millions de machines d'auto-hébergement se transforment en autant de zombies ! Pour limiter les risques, le logiciel doit donc être configuré de manière à être très blindé, et mis à jour automatiquement.

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc7033.txt

Reste le cas du DNS. C'est un support essentiel de l'identité en ligne (si votre identité est `facebook.com/BêteCompany` vous ne contrôlez pas votre identité, c'est Facebook qui le fait). Il semble donc logique qu'il y ait un serveur DNS dans le paquetage. Mais cela laisse deux problèmes difficiles : l'interaction avec le bureau d'enregistrement (la plupart des registres imposent un passage par cet intermédiaire, et il n'existe pas de protocole de communication standard entre le client et le BE, regardez comment cela complique la doc de YunoHost <https://yunohost.org/#/dns>), et les serveurs DNS secondaires (le RFC 1034 décrète sagement qu'il faut au moins deux serveurs par zone, pour des raisons de résilience). Idées bienvenues.

Une fois le logiciel défini et réalisé, il faut en faire une image. Le type d'image dépend de l'hébergement physique. Si on utilise l'IaaS, certains fournisseurs (pas tous, loin de là) permettent de créer ses propres images et de démarrer ensuite une machine virtuelle dessus. C'est de très loin le plus simple pour l'utilisateur. Pour Amazon EC2 <https://www.bortzmeyer.org/amazon-cloud.html>, le processus de création d'images est décrit dans de nombreux articles <http://onlamp.com/pub/a/onlamp/2008/05/13/creating-applications-with-amazon-ec2-and-s3.html> et la relative facilité de création et de publication d'images a mené au développement d'innombrables images toutes faites, pour tous les besoins. Pour Gandi, le processus est documenté <http://wiki.gandi.net/fr/hosting/create-server/private-image> mais, malheureusement, on ne peut pas mettre ces images à la disposition des autres, comme on peut le faire chez Amazon. Pour d'autres hébergeurs, je ne sais pas, n'hésitez pas à m'envoyer des informations.

Alors, évidemment, utiliser un hébergeur est simple et bon marché mais a des limites. Celui-ci, après tout, a le contrôle complet des machines et les données sur celles-ci ne sont donc pas forcément en sécurité. Et si on veut héberger ce paquetage de présence en ligne chez soi ? Pour que l'installation reste simple, il vaut mieux une image conçue pour une machine physique donnée. Par exemple, on pourrait imaginer des images pour les Soekris <http://soekris.com/>. Ou, pour un engin plus limité mais beaucoup moins cher et consommant très peu (donc pouvant être laissé branché en permanence), le Raspberry Pi <https://www.bortzmeyer.org/raspberry-pi.html>, on peut noter qu'il existe déjà énormément d'images pour le Pi <http://elinux.org/RPi_Distributions> dédiées à divers usages. On pourrait donc en ajouter une dédiée à la présence en ligne. (Un exemple d'installation de **beaucoup** de logiciels serveurs sur un Pi est documenté ici <http://seenthis.net/messages/123465#message123618>.)

Rien n'interdirait d'ailleurs de développer une "*appliance*" matérielle, ayant déjà l'image installée, sur le modèle de certains NAS grand public actuels (j'ai entendu plusieurs avis favorables sur les produits de Synology mais cela n'a pas l'air très libre.).

Notez qu'il n'y a pas de miracle : héberger chez soi ne résoud pas tous les problèmes (votre FAI peut vous couper) et, en IPv4, il faut prévoir des complications (comme d'ouvrir sur la "*box*" les ports permettant l'accès à distance).

Voilà, comme j'aime bien donner du travail aux autres, je vais m'arrêter là. Je serais ravi que des développeurs courageux s'emparent de cette idée et la réalisent. Notez bien qu'il n'est pas nécessaire (et sans doute pas souhaitable) qu'il n'y ait qu'un seul logiciel mettant en œuvre ce concept. Si tout le monde utilise des normes ouvertes comme HTTP et XMPP, il n'est pas nécessaire de n'avoir qu'un seul logiciel, ils pourront interagir. Et, dans l'intérêt de la sécurité, il vaut sans doute mieux ne pas mettre tous ses œufs dans le même panier et donc avoir plusieurs logiciels.

Quelques projets existants qui ont un rapport avec cette idée (merci à Daniel Le Bray pour la première liste) :

---

— BeedBox <http://www.beedbox.org/> (projet abandonné : comme beaucoup de projets en informatique, commencer est facile, terminer à 100 % est **difficile**). La description du projet <http://www.beedbox.org/a-propos> est très proche de ce que je propose.
— YunoHost <http://yunohost.org/>.
— CozyCloud <https://www.cozycloud.cc/>. Attention, ils ont deux offres, une hébergée (du SaaS classique) et une où on s[Caractère Unicode non montré [2] ]auto-héberge. Le logiciel est libre, mais pas encore fini (« "*In the future there will be a fully packaged virtual machine*" » c'est-à-dire exactement ce que je souhaitais).
— Helios.im <http://www.helios.im/>, qui est une "*appliance*" physique (en l'occurrence un Raspberry Pi).
— Sur le Rasperry Pi, voir aussi arkOS <https://arkos.io/> (encore expérimental).
— Amahi <http://www.amahi.org>, un logiciel pour transformer son PC en serveur d'auto-hébergement.
— Qy.share <http://www.qyshare.com/>. Cela semble très bien mais je ne trouve aucun détail sur leur site Web.
— FreedomBox <https://www.freedomboxfoundation.org/>, qui a des ambitions bien plus sécuritaires que l'idée que j'ai développé ici. Je pensais au contrôle des données, FreedomBox est plus dans une logique de sécurité.
— SME Server <http://wiki.contribs.org/Main_Page>.

On peut aussi noter qu'il n'y pas de choix qu'entre les gros monstres du "*cloud*" et l'auto-hébergement. On peut imaginer des systèmes coopératifs, où un certain nombre de citoyens se regroupent pour monter une plate-forme commune (merci à Fil pour avoir rappelé cette possibilité). Jean-Baptiste Favre a fait un excellent article sur ce thème <http://blog-notes.jbfavre.org/?lautohebergement-ou-le-risque-de-3034>. Un projet dans cet esprit est <http://the.re/>.

Comme ce blog n'a pas de système de commentaires <https://www.bortzmeyer.org/no-comment.html>, le mieux pour discuter de cette idée est d'aller sur SeenThis <http://seenthis.net/messages/123465>. (Une autre discussion - plus technique ? - a lieu sur LinuxFr <http://linuxfr.org/users/bortzmeyer/journaux/rendre-l-auto-hebergement-facile-et-sans-douleur>.) Enfin, dernier article, une intéressante réflexion sur l'auto-hébergement <http://seteici.ondule.fr/2013/07/introduction-a-lauto-hebergement/> chez soi.

2. Car trop difficile à faire afficher par LaTeX