N° 113

COMPRENDRE, UTILISER & ADMINISTRER LINUX

GNU LINUX PRATIQUE

MAI JUIN 2019

FRANCE MÉTRO. : 7,90 €
DOM/TOM : 8,50 €
BEL/LUX/PORT. CONT. : 8,90 €
CH : 13 CHF
CAN : 14 $CAD
MAR : 98 MAD
TUN : 20 TND

ÉDUCATION

Apprenez à créer un jeu de réflexion avec Pygame Zero p. 80

CODE

Intégrez une gestion des ACL à vos applications PHP p. 67

ENTREPRISE

Comment organiser votre communication sur Mastodon ? p. 90

SYSTÈME

Installez automatiquement Debian sans support d'installation p. 22

SYSTÈME

SIMPLIFIEZ L'ADMINISTRATION DE VOS MACHINES AVEC ANSIBLE ! p. 30

- Pourquoi automatiser ?
- Configurer Ansible
- Cas pratique d'automatisation

L 18864 - 113 - F: 7,90 € - RD

TUTORIELS

BUREAUTIQUE

Recadrez, annotez, compressez, protégez, éditez et gérez vos fichiers PDF p. 08

PHOTOS

Optimisez vos photographies avec GIMP avant leur publication sur Internet p. 16

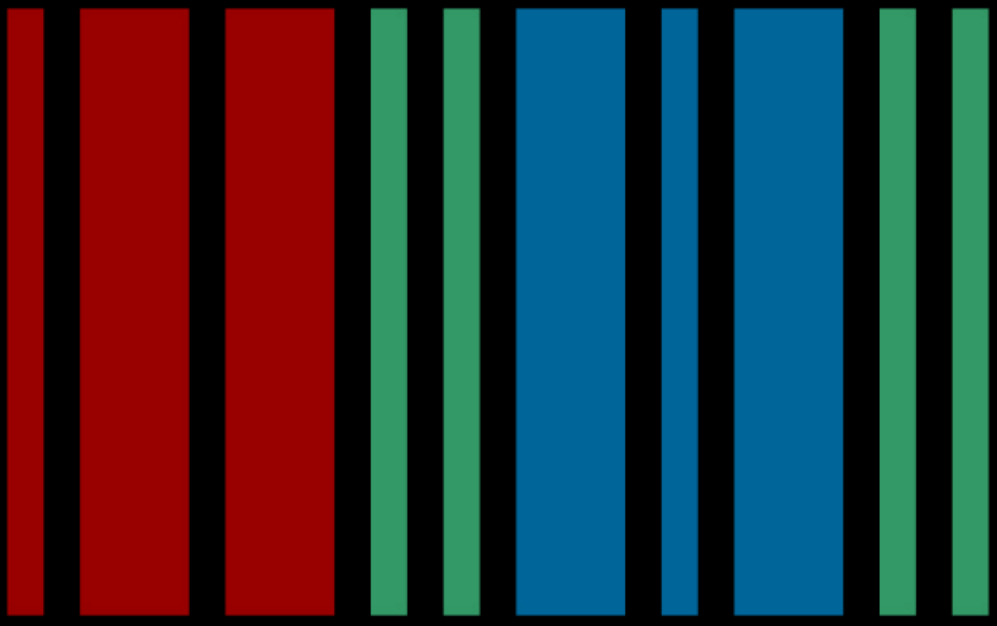
AUTO-HÉBERGEMENT

Centralisez la gestion de vos applications auto-hébergées avec Sandstorm p. 58

SÉCURITÉ

Isolez vos processus grâce à AppArmor p. 50

WIKI_____



_DATA

LA BASE DE DONNÉES LIBRE

Participez à Wikidata sur www.wikidata.org
ou lors de nos ateliers tous les 3e vendredi du mois au 40 rue de Cléry 75002 Paris
Toutes les infos sur wikimedia.fr

WIKIMEDIA
FRANCE

LINUX PRATIQUE EN NUMÉRIQUE ?

BIENVENUE SUR CONNECT !

LINUX PRATIQUE est édité par
Les Éditions Diamond

10, Place de la Cathédrale - 68000 Colmar - France
Tél. : 03 67 10 00 20 | **Fax** : 03 67 10 00 21

E-mail : cial@ed-diamond.com
lecteurs@linux-pratique.com
Service commercial :
abo@linux-pratique.com
Sites : https://www.linux-pratique.com
https://www.ed-diamond.com

Directeur de publication : Arnaud Metzler
Chef des rédactions : Denis Bodor
Rédactrice en chef : Aline Hof
Responsable service infographie : Kathrin Scali
Responsable publicité : Tél. : 03 67 10 00 27
Service abonnement : Tél. : 03 67 10 00 20
Photographie et images : http://www.fotolia.com
Impression : pva, Landau, Allemagne
Distribution France :
(uniquement pour les dépositaires de presse)
MLP Réassort :
Plate-forme de Saint-Barthélemy-d'Anjou
Tél. : 02 41 27 53 12
Plate-forme de Saint-Quentin-Fallavier
Tél. : 04 74 82 63 04
Service des ventes :
Distri-médias : Tél. : 05 34 52 34 01

IMPRIMÉ en Allemagne - PRINTED in Germany

Dépôt légal : A parution
N° ISSN : 0183-0872
Commission Paritaire : K78 990
Périodicité : Bimestrielle
Prix de vente : 7,90 Euros

Retrouvez-nous sur :

@linuxpratique
@editionsdiamond

@linuxpratique

https://connect.ed-diamond.com

https://www.linux-pratique.com
https://www.editions-diamond.fr

ENCART CONNECT ENCARTÉ DANS LA COUVERTURE

édito

Linux Pratique n°113

Un monde où la donnée est reine, peut-il influencer nos comportements ? À l'heure où nous parlons de plus en plus de capitalisme de données lorsque de grandes firmes ont érigé leur modèle économique sur le business de ces dernières, cette question est finalement loin d'être anodine.

Le big data et l'IA permettent de simplifier la gestion et le traitement de nos données. Grâce aux algorithmes, il est désormais possible d'établir des tendances, fruits de vos achats, de votre historique de navigation, des posts que vous avez aimés et/ou commentés, de vos contacts…

Cette accumulation de données peut donner naissance à une forme de « surveillance ». On pourra citer à cet effet le crédit social chinois **[1]** qui vise à observer le comportement des citoyens dans leur vie quotidienne et à noter ce dernier. En fonction de cette évaluation, diverses récompenses et sanctions peuvent être appliquées. Ce système – dont la mise en place est facilitée par notre mode de vie tout connecté et par les technologies citées précédemment – devrait se généraliser d'ici l'année prochaine en Chine.

Un procédé tel que celui-ci peut évidemment avoir des répercussions sur notre manière de nous comporter **[2]** : développement de l'auto-censure, d'une forme de conformisme, restriction des libertés, repli sur soi, etc. On notera que la législation ou encore l'essor des réseaux sociaux a malheureusement déjà largement contribué à tout cela.

Né le 12 mars 1989, le Web a fêté dernièrement ses 30 ans. L'effervescence des débuts a malheureusement laissé la place à une forme de désillusion pour son fondateur Tim Berners Lee, qui, je vous en parlais en novembre dernier, travaille actuellement sur un projet qui devrait permettre aux internautes d'avoir une meilleure gestion de leur identité numérique **[3]**. Les initiatives de ce type doivent être encouragées pour lutter contre l'émergence de sociétés où *Big Brother is watching you* **[4]**.

ALINE HOF

[1] https://fr.wikipedia.org/wiki/Système_de_crédit_social

[2] Pour aller plus loin sur le sujet : https://socialcooling.fr/

[3] https://solid.mit.edu/

[4] https://fr.wikipedia.org/wiki/1984_(roman)

Sommaire

Linux Pratique n°113

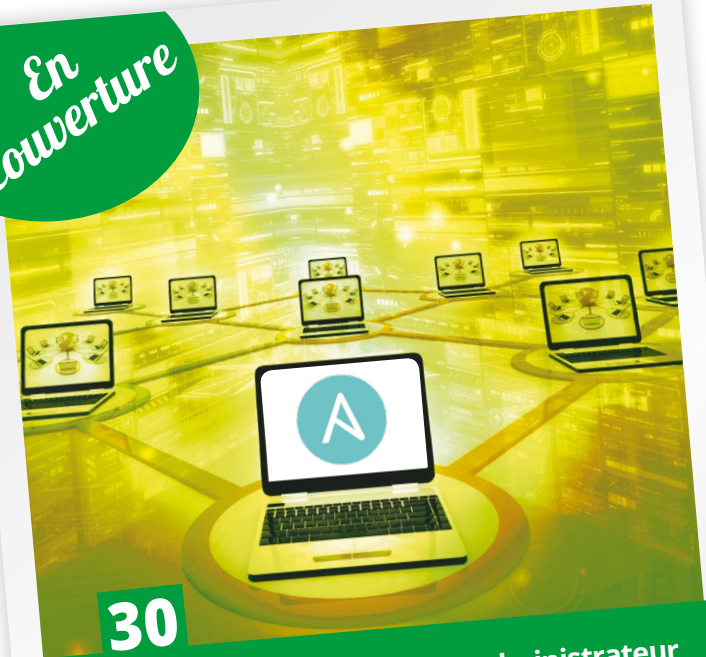
ACTUALITÉS & NOUVEAUTÉS

LOGITHÈQUE & APPLICATIF

SYSTÈME & PERSONNALISATION

SHELL & SCRIPTS

En couverture



30

Outil incontournable pour tout administrateur système, faites vos premiers pas avec Ansible pour automatiser tout ou presque sur vos systèmes d'information.

Sécurité

50

Diminuez la surface d'attaque globale de votre système en restreignant les actions de vos programmes avec AppArmor.

Système

22

Automatisez l'installation de Debian en créant votre fichier de préconfiguration Preseed.

Programmation

80

Découvrez le plaisir de programmer en créant un jeu de réflexion à l'aide du module Python Pygame Zero.

Web

58

Gros plan sur Sandstorm, une solution open source pour auto-héberger et gérer votre propre suite d'applications web.

SÉCURITÉ & PROTECTION

SERVEUR & CLOUD

CODE & DÉVELOPPEMENT

MOBILITÉ & OBJETS CONNECTÉS

ENTREPRISE & ORGANISATION

En bref...

■ L'Agence Nationale de la Sécurité des Systèmes d'Information (l'ANSSI) a publié une nouvelle version de son guide de recommandations de sécurité relatives à un système GNU/Linux. Y sont notamment décrits 5 recommandations minimales à mettre en place pour sécuriser son système : la réduction de la surface d'attaque, l'application d'un principe de défense en profondeur, la mise en place de mesures de cloisonnement applicatif, l'établissement de procédures d'administration sécurisées et l'instauration d'une politique de journalisation d'évènements cohérente. Le PDF mis à jour, pourra être consulté et téléchargé depuis **https://www.ssi.gouv.fr/guide/recommandations-de-securite-relatives-a-un-systeme-gnulinux/**.

■ Le projet Common Voice qui vise à développer et rendre accessibles les technologies de reconnaissance vocale suit son cours. Mené pour rappel par Mozilla, ce dernier permet de donner de sa voix pour enrichir une base de données vocale libre et publique. Il est désormais possible de télécharger un jeu de données vocales. Ce dernier, multilingue et open source, permet à tout un chacun d'entraîner ses applications et ainsi, favoriser le développement de technologies et d'applications utilisant la voix. À l'heure où nous rédigeons ces lignes, ce sont ainsi 1087 heures qui sont validées en 18 langues. Pour en savoir plus : **https://voice.mozilla.org/fr/datasets.**

■ Après deux ans de développement, le couteau suisse de la gestion d'images, digiKam, publie sa version 6.0. En plus de la photo, digiKam permet désormais de gérer aussi les fichiers vidéos. De nouvelles possibilités d'exportation sont disponibles, avec l'arrivée des options Pinterest, OneDrive et Box. Par ailleurs, l'accès à ces services en ligne est grandement facilité depuis les divers outils intégrés à digiKam. On notera aussi une meilleure gestion du format RAW avec la prise en charge d'un plus grand nombre d'appareils notamment grâce à la bibliothèque libraw 0.19. Toutes les nouveautés de ce logiciel open source très complet pour gérer vos clichés sont détaillées sur **https://www.digikam.org/news/2019-02-10-6.0.0_release_announcement/.** ■

Agenda

MAI – JUIN 2019

» 11 mai 2019 : Les différentes solutions d'hébergement web pour mettre en service un site internet

Rendez-vous le samedi 11 mai de 9h30 à 12h à l'Ecospace de Beauvais pour assister à un atelier consacré aux différents types d'hébergement web possibles pour mettre en place votre site internet. Il sera notamment question d'hébergement simple avec la solution que propose WordPress.com, mais aussi d'hébergement à la carte ou de gestion de serveurs. Pour consulter tous les évènements organisés par l'association Oisux, rendez-vous sur **https://www.oisux.org/**. ■

» 17 mai : AFUP Day

Cette édition 2019 dédiée à PHP se déroule simultanément dans trois villes différentes. À Rennes, on se rendra du côté de la Fabrique de 8h30 à 18h. À Lille, on pourra se diriger à l'auberge Stéphane Hessel. Et enfin à Lyon, cela se déroulera chez SupInfo. Dans ces villes, on pourra profiter d'un cycle de conférences composé de présentations techniques et de retours d'expérience dédiés au langage de programmation PHP. Plus d'informations sur cet évènement sur **https://event.afup.org/.** ■

» 12-13 juin : OW2con'19

C'est du côté de Châtillon qu'aura lieu la conférence annuelle d'OW2 – communauté open source internationale et indépendante regroupant plusieurs entreprises, collectivités et organisations – le mercredi 12 et le jeudi 13 juin. La thématique de cette nouvelle édition portera sur l'open source et la maturité industrielle. Ce rendez-vous, gratuit et ouvert à tous, proposera des conférences en anglais sur divers sujets : l'open source dans les grandes villes, l'intelligence artificielle, la gouvernance des logiciels libres, l'accessibilité, l'IoT, la sécurité, etc. Le programme de ces deux journées pourra être consulté sur **https://www.ow2con.org/view/2019.** ■

DISPONIBLE DÈS LE 31 MAI

LINUX PRATIQUE HORS-SÉRIE N°45

NOUVEAU ! ACHETEZ NOS HORS-SÉRIES DÈS MAINTENANT SUR IOS ET ANDROID :

Disponible en App Store

DISPONIBLE SUR Google Play

Sous réserve de toutes modifications.

NE LE MANQUEZ PAS

CHEZ VOTRE MARCHAND DE JOURNAUX ET SUR :

https://www.ed-diamond.com

Panorama d'outils pour LA MANIPULATION DE FICHIERS PDF

ANTHONY CARRÉ

LES FICHIERS PDF NE SONT PAS VRAIMENT CONÇUS POUR ÊTRE ÉDITÉS, MAIS IL ARRIVE TRÈS RÉGULIÈREMENT QUE NOUS AYONS BESOIN D'ISOLER UNE PAGE, DE RECADRER, D'ANNOTER OU DE CONCATÉNER, ETC. DE NOMBREUX LOGICIELS LIBRES SONT À DISPOSITION POUR CES OPÉRATIONS – ET BIEN D'AUTRES ENCORE –, MAIS ENCORE FAUT-IL EN CONNAÎTRE L'EXISTENCE. POUR VOUS AIDER À VOUS Y RETROUVER, VOICI UNE PRÉSENTATION DE DIFFÉRENTES INTERFACES GRAPHIQUES ET DIVERSES SOLUTIONS EN LIGNE DE COMMANDES DISPONIBLES SOUS GNU/LINUX. PLUS AUCUN FICHIER PDF NE VOUS RÉSISTERA.

Le but n'est pas de lister l'ensemble des applications, mais de présenter au moins une solution pour chaque besoin, plusieurs lorsqu'elles apportent des fonctionnalités particulières, en détaillant succinctement l'usage de l'application. Pour l'ensemble des exemples de commandes, nous prendrons des fichiers d'entrée in.pdf (in1.pdf, in2.pdf… éventuellement) et out.pdf, ces noms peuvent bien évidemment être modifiés.

Pour chaque commande citée, nous ne pouvons pas détailler l'ensemble des options ou la syntaxe, nous vous invitons donc à compléter nos informations via man, tldr ou votre moteur de recherche préféré.

» ÉTAPE 1
LE RECADRAGE

Krop **[1]** est certainement la solution la plus complète pour recadrer un PDF. Son interface est intuitive et efficace : vous ouvrez un fichier depuis le sélecteur de fichiers, vous indiquez un nom de fichier pour l'enregistrement, sur quelles pages appliquer les cadrages que vous dessinez directement sur le rendu du fichier, et voilà ! Quelques options supplémentaires vous permettent entre autres d'appliquer une rotation ou d'utiliser Ghostscript pour compresser le fichier de sortie, un onglet ***Advanced*** vous permet un accès à quelques options avancées. Vous pouvez créer plusieurs cadres dans une même page ou créer des cadres différents pour chaque page…

Autre solution disponible pour le recadrage : pdfhandoutcrop **[2]**, petite application simple permettant de définir plusieurs cadres par page. Pas franchement sexy, l'application a néanmoins une fonction de cadrage automatique qui peut se révéler intéressante.

En ligne de commandes, la référence est sans conteste, pdf-crop **[3]** (que vous trouverez dans le paquet texlive-extra-utils) :

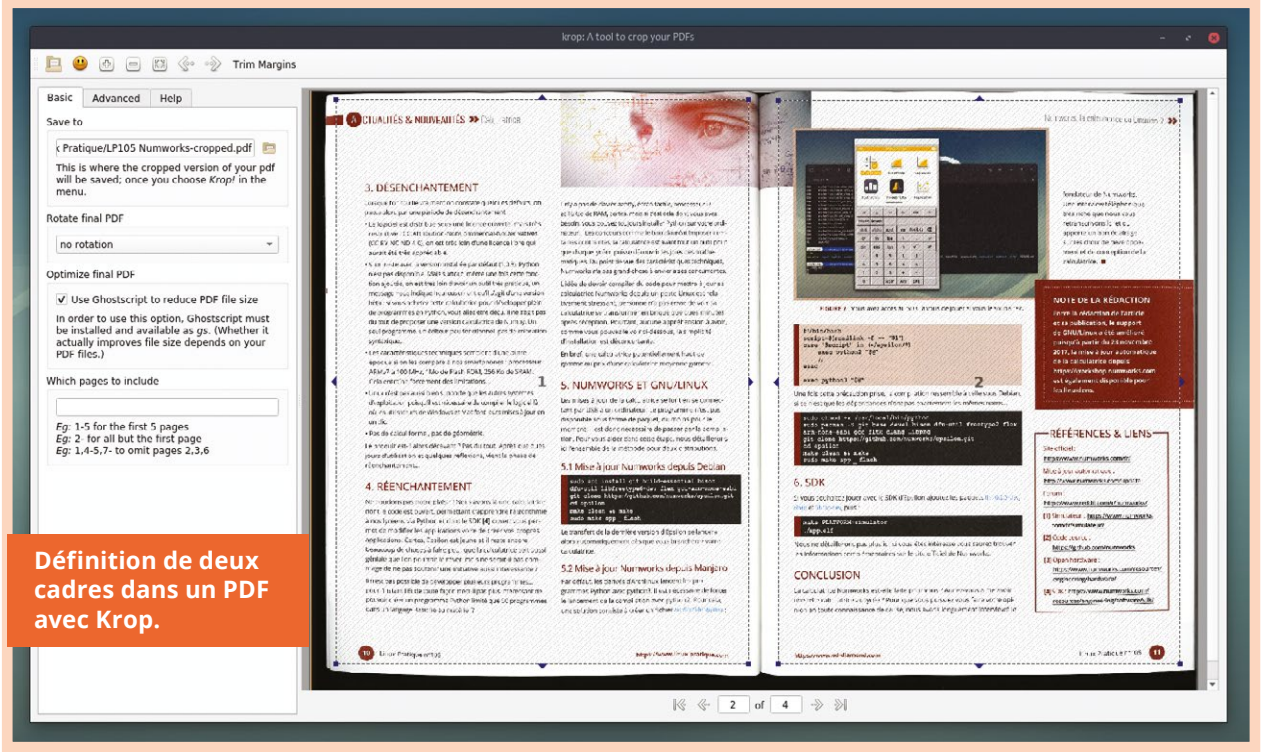
```
pdfcrop --margins '-10
-20 30 40' in.pdf out.pdf
```

Cette commande découpe 10 points à gauche et 20 en haut, ajoutera une marge de 30 à droite et de 40 en bas. Si un seul nombre est donné, il est appliqué à toutes les marges, utilisez pdfcrop --help pour plus de détails. L'unité bp (*big point* qui vaut 1/72 de pouce) n'est pas forcément idéale pour les néophytes, celle-ci est bien connue des utilisateurs de TeX.

Notez que vous pouvez également faire des recadrages dans pdfshuffler **[4]**, un outil plus complet de manipulation de fichiers PDF (voir section suivante).

» ÉTAPE 2
LA MANIPULATION DE PAGES

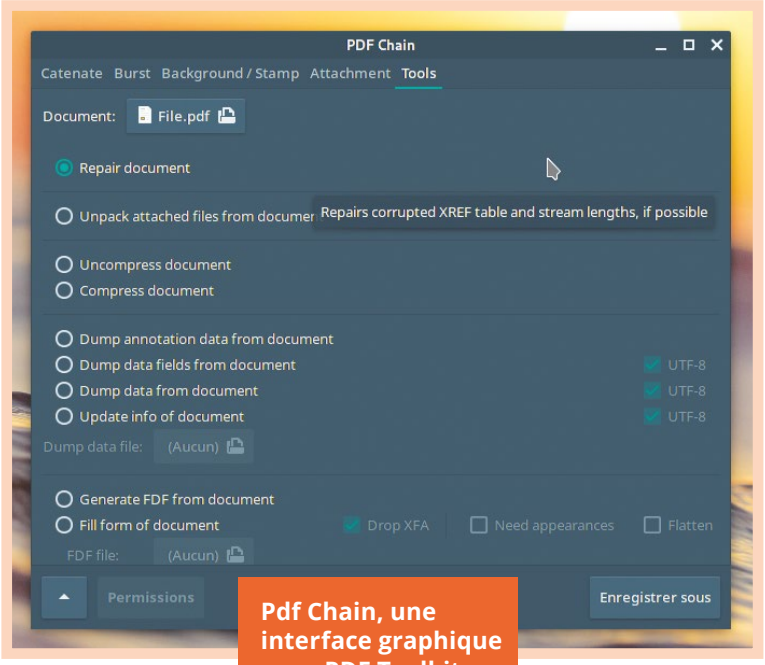
Trois outils très proches permettent de manipuler les pages de vos PDF en toute simplicité. Que ce soit pour supprimer une page, en modifier l'ordre, assembler deux fichiers, tourner une



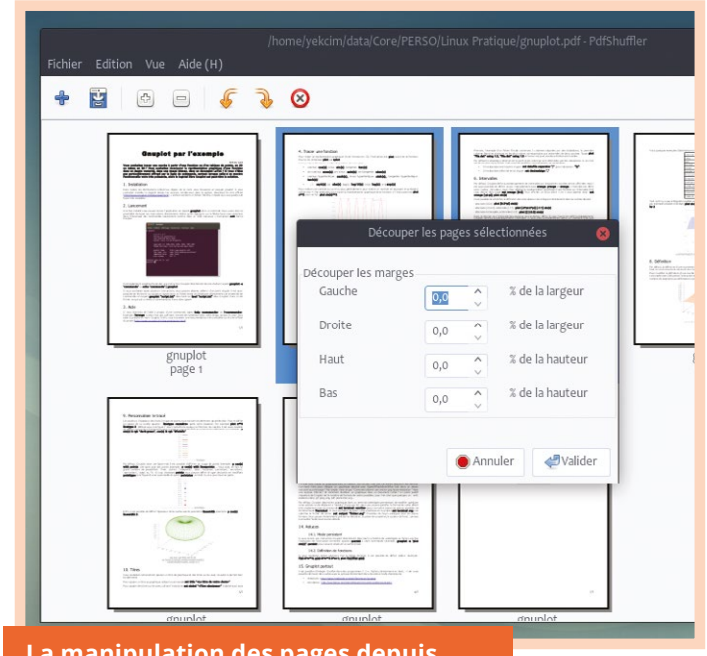
Définition de deux cadres dans un PDF avec Krop.

page en mode paysage ou en extraire une partie, PdfShuffler, pdfarranger **[5]** (fork de PdfShuffler) et PDF Mod **[6]** proposent tous les trois des interfaces très similaires, intuitives et efficaces. Le dernier du trio est peut-être le moins intéressant puisqu'il ne propose pas de fonction de recadrage (et sa dernière version date de 2011). Les pages du fichier sont affichées comme dans une trieuse, vous pouvez appliquer directement vos modifications par glisser-déposer ou via le menu contextuel qui apparaît lors d'un clic droit.

Un peu moins complète, l'alternative PDF Mix Tool **[7]** a une interface différente : ce ne sont pas les pages qui sont affichées directement, mais les fichiers, cela rend les opérations moins intuitives. Puisqu'il n'est pas nécessaire d'afficher l'ensemble des pages des documents, l'outil peut se révéler utile pour les fichiers volumineux dont le rendu complet peut parfois s'avérer difficile dans une vue d'ensemble.



Pdf Chain, une interface graphique pour PDF Toolkit.



La manipulation des pages depuis une visionneuse est particulièrement intuitive avec PdfShuffler.

Le logiciel CLI incontournable pour les opérations sur les pages des PDF est le célèbre pdftk (*pdftk server* pour être précis, mais le paquet est disponible sous son nom abrégé) **[8]**. Plusieurs distributions GNU/Linux proposent son port java pdftk-java **[9]**. De très nombreuses opérations sont possibles depuis ce logiciel très puissant !

Supprimer la page 4 d'un fichier :

```
pdftk in.pdf cat 1-3 5-end output out.pdf
```

Concaténer deux fichiers :

```
pdftk in1.pdf in2.pdf cat output out.pdf
```

Extraire chaque page d'un fichier dans un PDF spécifique (et écrire les metadata dans un fichier texte) :

```
pdftk in.pdf burst
```

Et encore de très nombreuses opérations ! Notez que l'outil étant puissant et multiplateforme, plusieurs interfaces graphiques s'appuient directement sur celui-ci, sous GNU/Linux, vous pouvez utiliser PDF Chain **[10]** qui propose un grand nombre de possibilités : concaténation, séparation, ajout d'un filigrane, fichier joint, compression, réparation de fichier corrompu…

Autres solutions CLI intéressantes, alternatives peut-être un peu moins connues que pdftk : QPDF **[11]** et PDFjam **[12]**. À titre d'exemple, voici comment :

• combiner les 5 premières pages d'un fichier avec les pages 11 à 15 d'un autre avec QPDF :

```
qpdf --empty --pages in1.pdf 1-5 in2.pdf
11-15 -- out.pdf
```

• extraire la page 2 d'un fichier avec pdfjam :

```
pdfjam in.pdf '2'
```

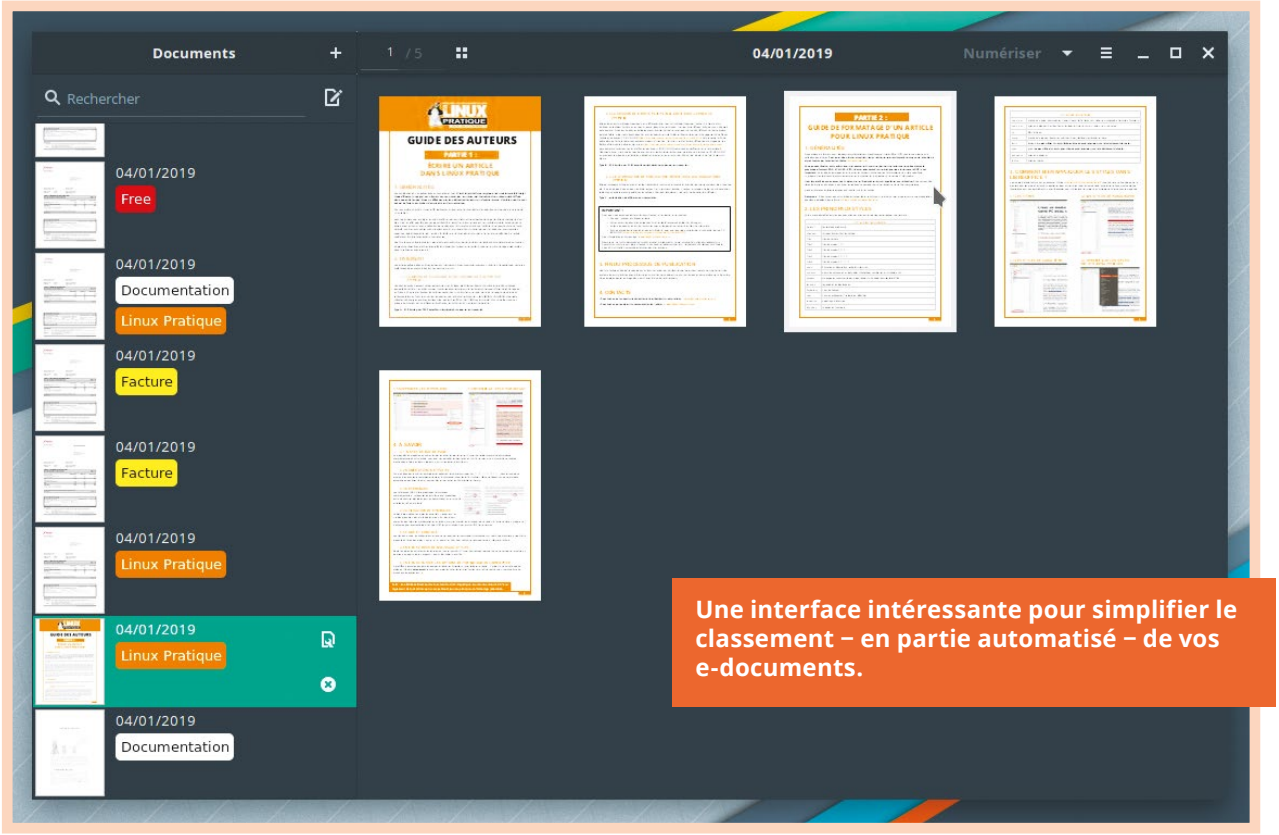
» ÉTAPE 3
L'ORGANISATION

Trier l'ensemble des factures ou autre paperasse importante afin de pouvoir les retrouver aisément au moment opportun, voilà bien une tâche qui mérite son logiciel. Depuis l'interface de Paperwork **[13]**, vous pouvez scanner directement vos documents, mais vous pouvez également les dater et taguer, et effectuer des recherches par mots-clés, la reconnaissance de caractères étant appliquée automatiquement sur les documents ajoutés ! Un logiciel indispensable pour tous ceux qui souhaitent simplifier leur organisation de documents électroniques.

Moins de fonctions, mais plus simple, booksorg **[14]** vous permet également d'organiser vos PDF. Plus particulièrement pensé pour la gestion de livres, le logiciel vous permet toutefois de gérer n'importe quelle série de PDF.

» ÉTAPE 4
L'EXTRACTION DE TEXTE, L'EXTRACTION D'IMAGES ET L'OCR

Si votre fichier contient du texte directement (n'est pas un scan), vous pouvez utiliser PDFMiner **[15]**, outil permettant d'extraire des informations et le texte de documents PDF. Nous vous laissons regarder les options disponibles via `pdftotext --help`, mais pour obtenir un fichier texte contenant la prose d'un PDF, il vous suffit de taper la très courte commande :



Une interface intéressante pour simplifier le classement – en partie automatisé – de vos e-documents.

```
pdftotext in.pdf
```

Si vous souhaitez récupérer l'ensemble des images d'un PDF, c'est vers pdfimages **[16]** qu'il faudra se tourner. En une seule commande, l'ensemble des images d'un fichier PDF sera extrait dans un dossier :

```
pdfimages -all -- in.pdf nom_dossier_out
```

Pour extraire le texte d'un scan de document, vous connaissez très certainement Tesseract-OCR, logiciel de reconnaissance de caractères, qui vous permet d'en extraire le texte. OCRmyPDF **[17]** utilise Tesseract pour ajouter un calque à votre fichier PDF. Ce calque se superpose à l'image qu'il contient, vous pouvez le sélectionner lorsque vous utilisez l'outil sélection de texte. Ainsi, votre texte est directement intégré à votre fichier, ce qui permet par exemple de le copier ou d'utiliser la fonction recherche de votre visionneuse.

```
ocrmypdf -l fra in.pdf out.pdf
```

» ÉTAPE 5

LES ÉQUIVALENTS DE COMMANDES BASH (DIFF, GREP) DANS DES PDF

Pour rechercher un motif dans des documents texte, grep est sans aucun doute un outil indispensable et très puissant. Pdfgrep **[18]** est au PDF ce que grep est au texte, d'ailleurs la plupart de ses options sont calquées sur la syntaxe de grep pour un apprentissage le plus simple possible. Il est possible de rechercher dans un document, dans une page spécifique de celui-ci, dans l'ensemble des PDF d'un dossier, récursivement ou non, éventuellement sous forme d'expression régulière (regex).

```
pdfgrep --with-filename --page-number --recursive mot_cherché
```

diffpdf **[19]** est une interface graphique vous permettant de comparer deux fichiers directement. La comparaison peut se faire sur le texte ou visuellement. Vous pouvez sélectionner les fichiers à analyser depuis le logiciel ou en lançant la commande :

```
diffpdf fichier1.pdf fichier2.pdf
```

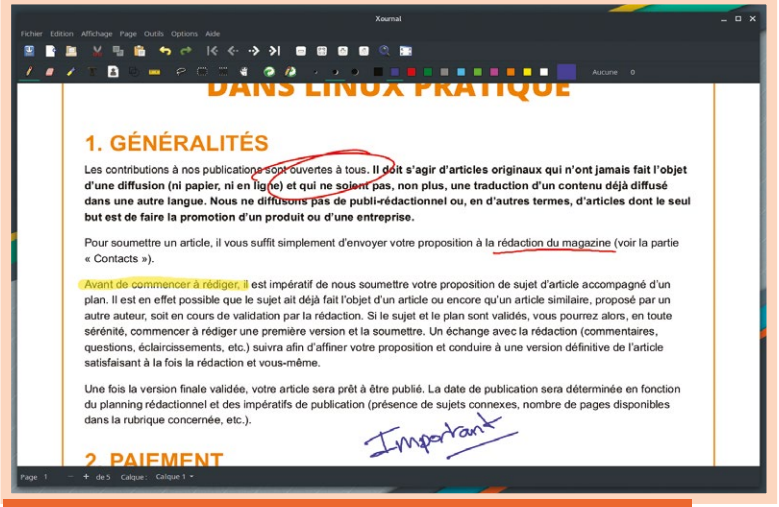
Hélas, même s'il reste toujours utilisable, il est à noter que le logiciel n'est plus maintenu par son auteur depuis quelques années. On pourra alors se rabattre sur une solution en ligne de commandes comme **[20] [21] [22]**.

» ÉTAPE 6

L'ANNOTATION ET LE GRIBOUILLAGE

Xournal **[23]** est un logiciel de prise de notes. S'il permet d'écrire au stylet (donc plutôt destiné aux possesseurs de tablette graphique) dans une page blanche virtuelle, il nous intéresse ici, car il permet également de faire directement des annotations sur un PDF, que ce soit au stylet ou à la souris, sous forme de texte, d'un coup de surligneur virtuel, etc.

Xournal++ **[24]** est un fork de Xournal proposant quelques fonctionnalités supplémentaires : [Ctrl]+molette pour le zoom, possibilité d'intercaler des pages, insertion d'images, dessins de formes géométriques, etc.



Annotez vos PDF ou prenez des notes avec Xournal.

» ÉTAPE 7
LA COMPRESSION, L'OPTIMISATION

Pdfsizeopt **[25]** permet d'optimiser en toute simplicité la taille de vos fichiers PDF :

```
pdfsizeopt in.pdf out.pdf
```

Vous pouvez également utiliser Ghostscript **[26]** pour compresser vos fichiers. Hélas, les commandes gs ne sont pas forcément très intuitives. Selon la qualité désirée, choisissez screen, ebook, printer ou prepress :

```
gs -sDEVICE=pdfwrite -dCompatibilityLevel=1.4
-dPDFSETTINGS=/ebook -dSAFER -dNOPAUSE -dQUIET
-dBATCH -sOutputFile="out.pdf" "in.pdf"
```

Heureusement, il est possible de simplifier la syntaxe en passant par ps2pdf :

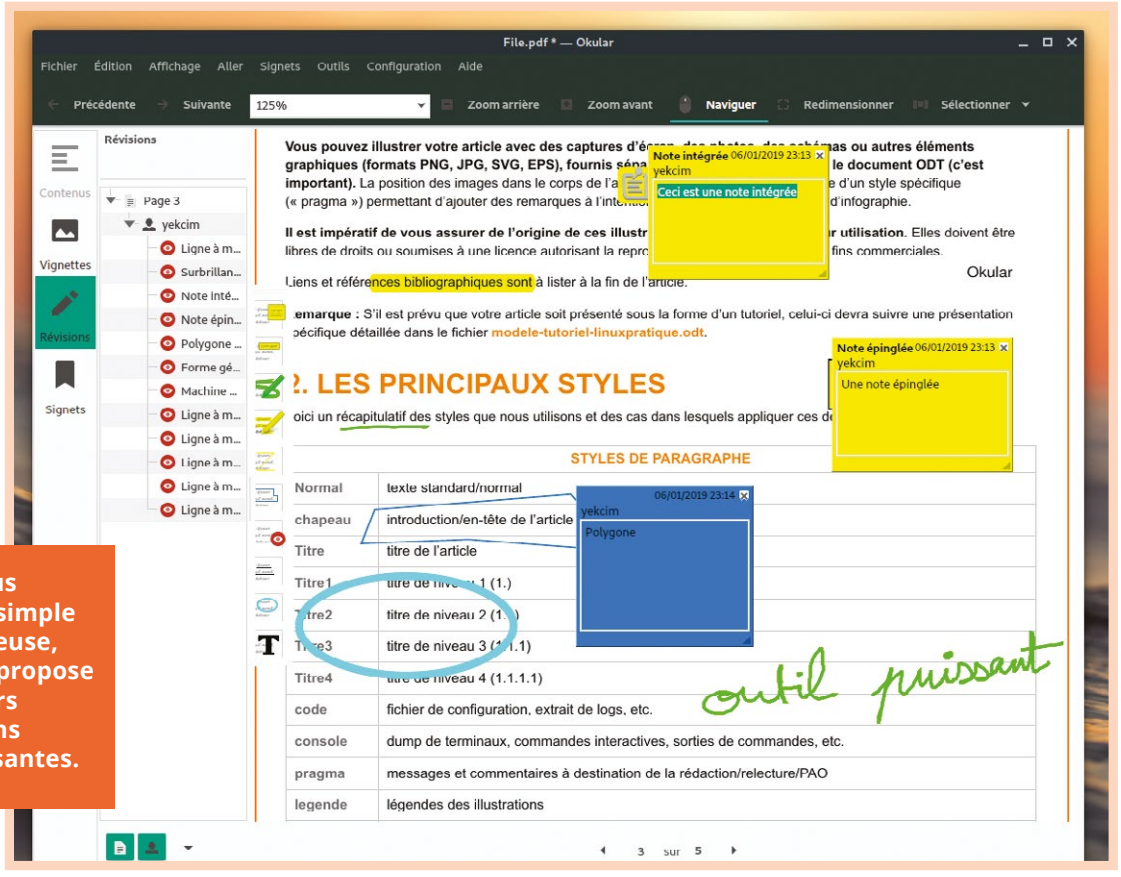
```
ps2pdf -dPDFSETTINGS=/ebook in.pdf out.pdf
```

» ÉTAPE 8
LA VISUALISATION (ET PLUS)

Evince **[27]** est la visionneuse PDF du projet GNOME. Outre l'affichage, Evince permet la gestion des annotations et le surlignage de texte (cliquez sur le crayon pour afficher ces fonctionnalités). Un mode nuit (fond sombre), un mode présentation en diaporama, la gestion de formulaires et une gestion de signets sont également disponibles.

Okular **[28]** est la visionneuse PDF du projet KDE. Elle propose des fonctionnalités similaires à Evince et quelques outils supplémentaires. En activant les annotations dans le menu ***Outils > Révision*** (raccourci [F6]) vous pouvez en effet dessiner sur votre document et ajouter différents types de notes ou un filigrane.

Zathura **[29]** est le lecteur de PDF qui sera le plus apprécié par les utilisateurs de vim. Minimaliste, il se pilote au clavier avec les mêmes principes que le célèbre éditeur de texte. Un mode présentation, un mode plein écran, / pour



Bien plus qu'une simple visionneuse, Okular propose plusieurs fonctions intéressantes.

rechercher, n pour la prochaine occurrence, hkjl pour se déplacer, [Ctrl+r] pour obtenir un mode sombre, o pour choisir le fichier à ouvrir, r pour tourner, f pour suivre un lien, : pour exécuter une commande…

Si vous cherchez un lecteur minimaliste plus standard, nous vous conseillons d'essayer mupdf **[30]** ou Xpdf **[31]**.

» ÉTAPE 9
LES PETITS OUTILS BIEN PRATIQUES

Quasiment toutes les applications permettent d'obtenir un PDF avec l'option « Imprimer dans un fichier ». Mais il peut arriver que certaines ne proposent pas cette option. Si vous rencontrez ce problème, vous pouvez installer cups-pdf **[32]**, qui ajoute à votre système une imprimante virtuelle accessible alors via le menu ***Impression*** de toutes les applications.

Pdf-remove-blank-pages **[33]** supprime les pages vides des documents PDF (ne fonctionne que sur les PDF contenant du texte, pas des images). La commande suivante supprime les pages sans texte du fichier (après en avoir fait une copie de sauvegarde) :

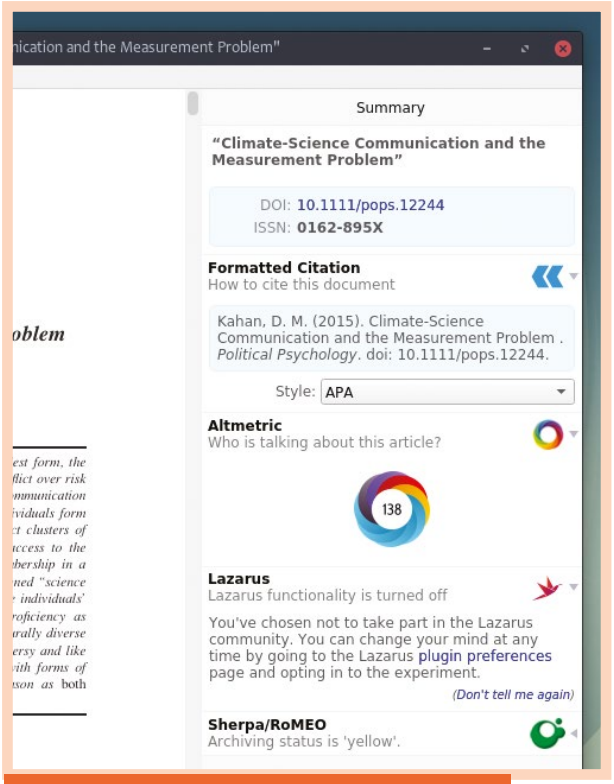
```
pdf-remove-blank-pages -f in.pdf
```

pdf-zip **[34]** est un petit script bash basé sur pdftk, déniché dans le dépôt AUR d'archlinux (mais utilisable depuis n'importe quelle distribution). Il permet de combiner deux fichiers PDF en alternant successivement une page de l'un puis de l'autre, particulièrement pratique lorsque vous scannez un document recto verso en deux fois (sur un scanner ne proposant pas l'option).

```
pdf-zip in1.pdf in2.pdf out.pdf
```

» ÉTAPE 10
UN COMPLÉMENT D'INFORMATIONS POUR LES SCIENTIFIQUES

Outil très spécifique à réserver aux scientifiques, Utopia Documents **[35]** permet l'analyse de publications au format PDF. L'outil recherche directement des publications en ligne à partir de mots-clés, mais surtout cherche automatiquement dans vos PDF certaines informations pertinentes pour en compléter l'affichage avec des renseignements importés depuis Internet : format de citation, DOI, nombre de citations de l'article sur Twitter ou des blogs…



Analysez automatiquement vos PDF correspondants à des publications scientifiques.

» ÉTAPE 11
PROTECTION ET SIGNATURE

Vous avez un PDF protégé par mot de passe et souhaitez supprimer cette protection, plusieurs commandes présentées précédemment peuvent être utilisées :

```
qpdf --decrypt --password=votremotdepasse
in.pdf out.pdf
pdftk in.pdf input_pw "votremotdepasse"
output out.pdf
```

Pour ajouter un mot de passe, les mêmes logiciels peuvent évidemment être utilisés.

Si vous avez oublié le mot de passe, si celui-ci n'est pas trop complexe ou si vous n'êtes pas trop pressé, l'attaque par force brute peut permettre de retrouver le précieux sésame. Pdfcrack **[36]** peut venir à votre rescousse assez simplement :

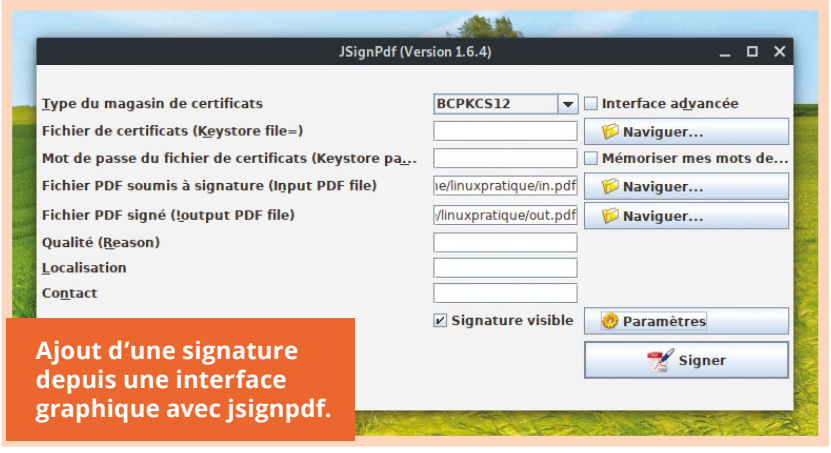
```
pdfcrack in.pdf
```

Des options permettent d'accélérer le processus, par exemple en définissant un dictionnaire ou une longueur minimum et maximum. Par défaut, seules les lettres minuscules, majuscules et les chiffres sont testés. Pour ajouter des caractères spéciaux, définissez la liste des caractères avec l'option --charset.

```
pdfcrack -c 'abcdefghijklmnopqrstuvwxyzABCDEFGH
IJKLMNOPQRSTUVWXYZ0123456789.-@#~%$&' in.pdf
```

Pour vous donner un ordre d'idée, lors de nos essais (dépendant grandement de la puissance de la machine), le mot de passe « aaa », a été trouvé en 0,07 secondes, « test » en 85 secondes, mais pour un mot de passe fort, plusieurs jours/semaines/mois peuvent être nécessaires.

Enfin, si vous devez ajouter une signature numérique à un document, jsignpdf **[37]** peut vous simplifier la tâche.



Ajout d'une signature depuis une interface graphique avec jsignpdf.

» ÉTAPE 12
L'ÉDITION

Le format PDF n'est pas vraiment fait pour être édité. Il existe toutefois quelques solutions pour apporter des modifications. PDFedit **[38]** est une application spécifiquement définie pour l'édition de PDF, mais il est devenu difficile à installer (la dernière version date de 2010), il est donc préférable de se tourner vers des alternatives.

Pour les fichiers contenant peu de pages, il est possible d'utiliser Inkscape **[39]**, logiciel incontournable de dessin vectoriel. L'importation ne peut se faire qu'une page à la fois, mais les données sont souvent très bien importées.

Pour l'édition d'un document multipages, la solution la plus commode semble être LibreOffice Draw **[40]**.

CONCLUSION

Cette liste bien que relativement abondante n'est bien sûr pas exhaustive, ne soyez donc pas choqué si vous n'y avez pas trouvé votre logiciel préféré. Plutôt qu'une présentation complète d'un logiciel, nous avons tenté de vous présenter un ensemble de solutions répondant à la quasi-totalité des besoins en rapport avec le format de fichier PDF.

Adobe Systems, éditeur des célèbres Reader et Acrobat ne supporte plus GNU/Linux, nous espérons que cet article vous aura convaincu que ce n'est toutefois pas pénalisant outre mesure au quotidien pour un usage courant, car les logiciels libres ont su répondre à cette absence pour un très grand nombre d'actions. ■

Les références de cet article sont disponibles sur : https://www.linux-pratique.com.

RETOUCHER VOS PHOTOGRAPHIES

avec GIMP avant leur publication sur Internet

SÉBASTIEN COLAS

Site du projet : https://www.gimp.org/fr/

DANS CET ARTICLE, JE VOUS PROPOSE QUELQUES ÉTAPES TRÈS SIMPLES À RÉALISER POUR AVOIR UN RENDU OPTIMAL SUR VOS PHOTOGRAPHIES LORS D'UNE PUBLICATION SUR UN SITE WEB.

» ÉTAPE 1
LE RECADRAGE

À moins d'être photographe professionnel, les photos ont souvent des problèmes de cadrage. Pour ma part, j'opte pour un recadrage qui conserve les proportions de la photographie.

On sélectionne ***Outil de découpage*** dans les outils. Comme le but est de garder les proportions, il faut cliquer sur ***Fixé*** dans le menu (Figure 1).

Dans la photographie d'exemple, le cœur de l'action se situe à gauche et le personnage de droite sur la photo originale déborde de la photographie donc autant le supprimer (Figure 2).

Figure 1

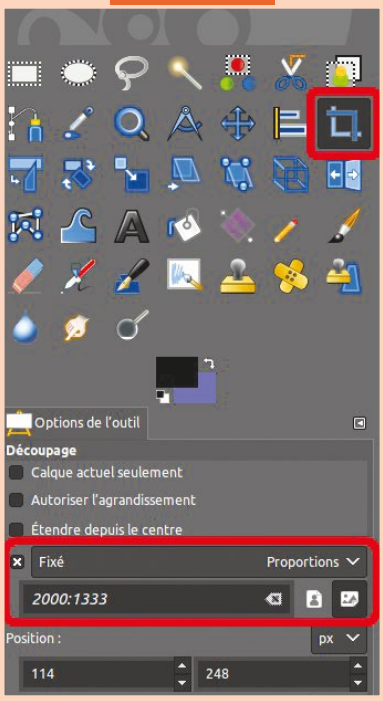


Figure 2

» ÉTAPE 2
LE RÉGLAGE DE LA SATURATION

Par manque d'éclairage les photographies ont tendance à virer à l'orange. Pour corriger ce problème, il faut modifier la saturation de l'image. Pour ce faire, on clique avec le bouton droit, puis on sélectionne le menu ***Couleurs***. Ensuite, on sélectionne ***Teinte-Saturation***. Le menu de réglage apparaît alors et là le réglage est assez simple, il suffit de baisser la saturation (dans l'exemple -15,2) (Figure 3). Pour vous rendre compte en temps réel des modifications, je vous invite à cliquer sur ***Aperçu*** et ***Éclater la vue***.

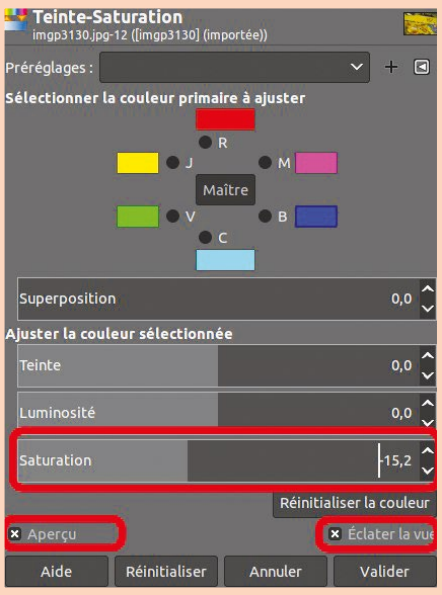


Figure 3

Figure 4

Sur la Figure 4 page précédente, la partie gauche nous montre un aperçu du changement de la saturation, la partie droite nous montre l'image avant la correction. Certes les couleurs ont l'air plus ternes après les modifications, mais nous allons régler ce problème dans l'étape suivante.

» ÉTAPE 3
LE RÉGLAGE DES NIVEAUX

Un autre effet du manque d'éclairage est d'avoir des photographies ternes/sombres. Pour corriger ce problème, nous allons modifier les niveaux. Il faut donc faire un clic droit sur la photographie, puis on sélectionne le menu ***Couleurs > Niveaux***. Le menu de réglage apparaît alors. Un histogramme nous permet de visualiser les couleurs du plus sombre au plus clair.

Voici comment je fais mes réglages. Je ramène le niveau de sortie des couleurs claires (l'index de droite) à la fin des couleurs, cela éclaircira

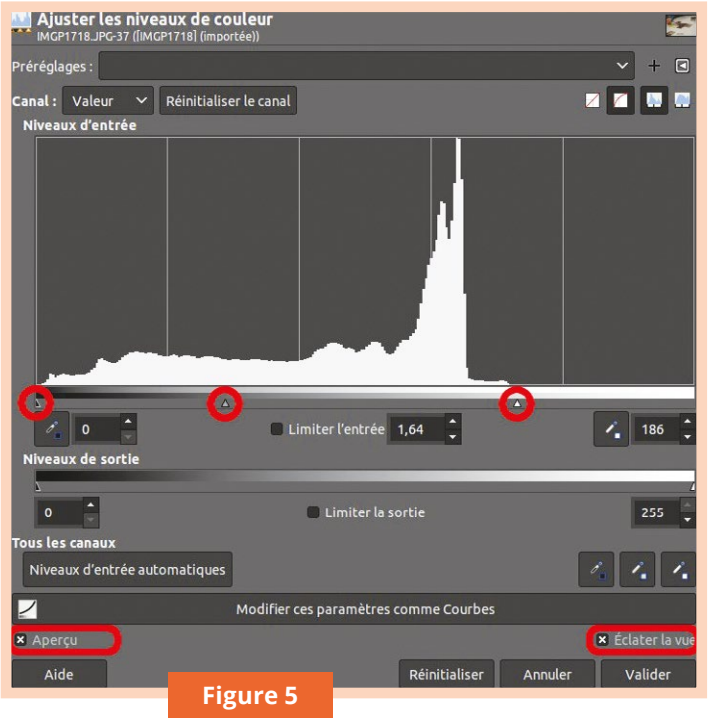


Figure 5

toute l'image (Figure 5). En général, il faut aussi éclaircir les couleurs médianes. Pour ce faire, on sélectionne l'index du milieu et on le décale vers la gauche. Comme d'habitude, je vous conseille de cocher les menus ***Aperçu*** et ***Éclater la vue*** pour se rendre compte en temps réel du futur rendu. Sur la photo, voici un aperçu des modifications (Figure 6).

Figure 6

» ÉTAPE 4
LE REDIMENSIONNEMENT

Avant de publier sur Internet, il est important d'avoir une dimension d'image correcte. Mon appareil photo crée des photographies d'une résolution de 5472x3648, ce qui est beaucoup trop pour une image Internet.

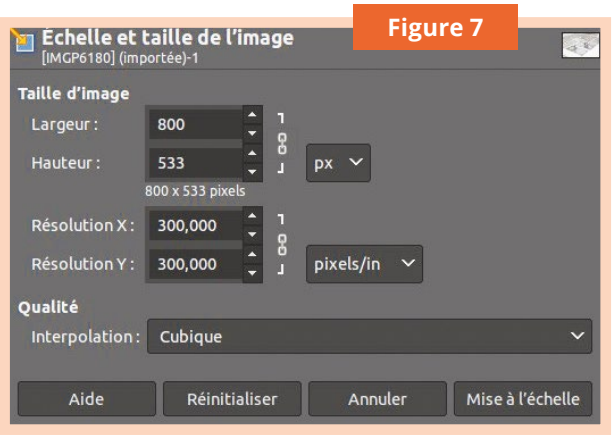


Figure 7

On fait donc un clic droit sur la photographie, puis ***Image > Échelle et taille de l'image***. Le menu de réglage apparaît alors. Je choisis en général une largeur de 800px (Figure 7). L'intérêt de cette modification est de pouvoir afficher une image sur la page web avec un écran classique.
Si par contre on dispose d'un écran 4K, on pourra alors afficher deux photographies côte à côte.

» ÉTAPE 5
LA COMPRESSION

La dernière étape consiste à réduire la taille du fichier image. Le mieux est d'opter pour un format d'image permettant une compression destructrice tel que le JPEG. Pour enregistrer notre photographie, on clique

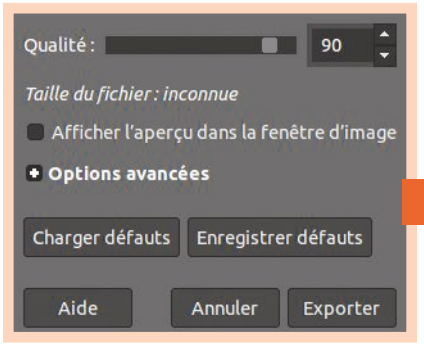


Figure 8

droit sur l'image, puis ***Fichier > Exporter sous***. On donne un nom de fichier avec l'extension .jpg et pour finir le menu de réglage nous propose la qualité de l'image (Figure 8). Plus la qualité est faible, plus l'image sera de petite taille, mais plus la compression sera visible. Une qualité raisonnable se situe entre 70 et 90, le mieux c'est de faire des tests.

On peut voir sur la partie gauche de la photographie une faible qualité d'image (Figure 9).

CONCLUSION

Voici donc 5 étapes très simples vous permettant de rapidement tirer le meilleur de vos photos en vue de les publier sur un site internet. ■

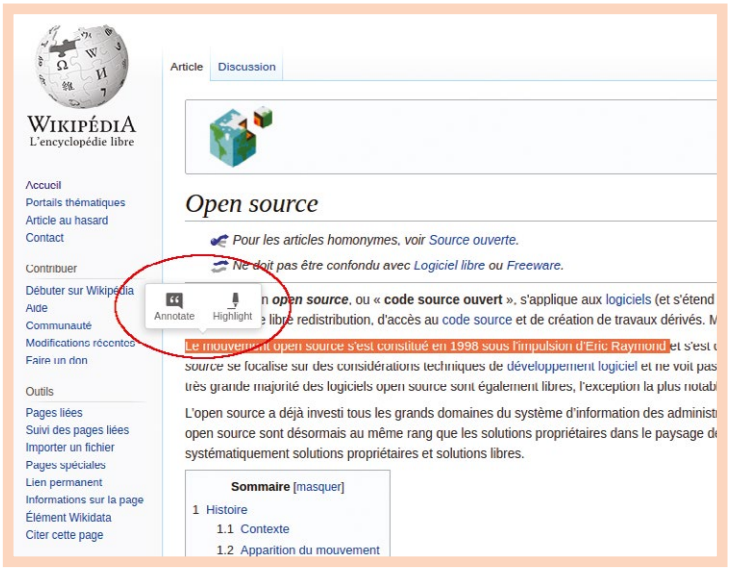
Figure 9

Hypothesis

ANNOTEZ LE WEB ET PARTAGEZ VOS RÉFLEXIONS

À l'heure où nous rédigeons ces lignes, Hypothesis se présente sous la forme d'une extension pour Chrome (le travail est en cours pour le développement d'un add-on Firefox) et d'un bookmarklet pour les autres navigateurs web. Open source, il a été développé pour fournir entre autres une solution aux étudiants, scientifiques, professeurs, éditeurs et journalistes qui leur permette de partager plus facilement l'information, de faciliter leurs recherches et d'échanger simplement sur des sujets variés. Il permet en effet de transformer le Web en un vaste espace de discussion.

Après installation, il faudra se créer un compte sur le site du projet. Une fois connecté, on pourra commencer à utiliser l'outil. Surfez sur le Web et annotez les pages ou les passages de votre choix. Vous pouvez aussi simplement choisir de surligner des termes. Des tags peuvent être ajoutés à vos notes, mais aussi des URL, des images ou encore des formules mathématiques. Vos notes peuvent être publiques, privées ou partagées avec les personnes de votre choix (dans ce cas-là, vous pourrez simplement leur envoyer un lien par e-mail ou via Facebook/Twitter).



Le point fort d'Hypothesis est sa fonctionnalité collaborative. Vous pouvez créer vos propres groupes et y inviter les personnes de votre choix pour partager ensemble vos notes. Lors de votre exploration du Web, vous pourrez aussi tomber sur les notes publiques d'autres utilisateurs, l'occasion d'entamer une discussion sur un sujet en particulier. Toutes vos notes restent accessibles dans un panneau latéral à droite de votre navigateur ou sur la page de votre compte sur le site du projet. Un outil de recherche est de la partie pour vous faciliter l'exploration de ces dernières. ■

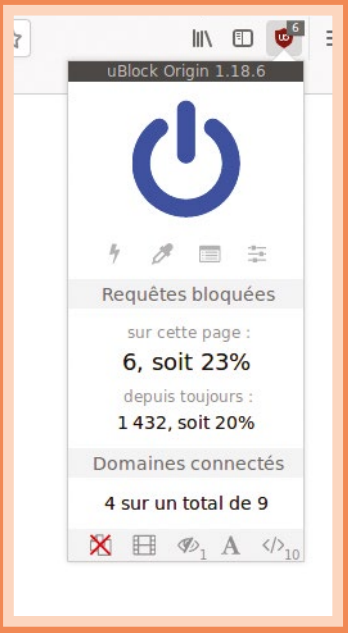
VERSION TESTÉE : 1.136.0.2 **SITE DU PROJET :** https://web.hypothes.is/ **COMPATIBILITÉ :** Chrome/Chromium

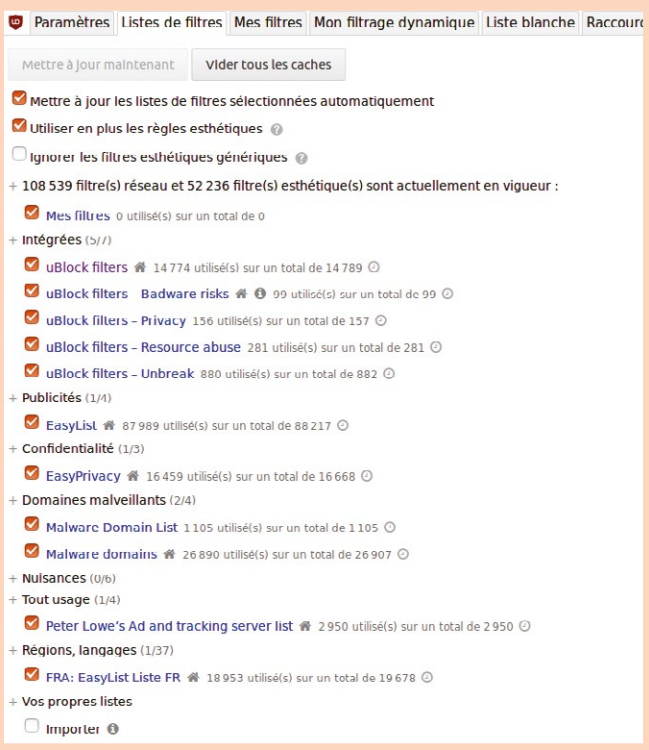
Ublock Origin

L'ARMURE ANTI-PUBLICITÉS ET PISTEURS

Ublock Origin est une extension libre qui vise à vous fournir de quoi vous libérer des publicités et pisteurs qui nuisent à votre navigation en ligne (et accessoirement à votre vie privée). Une fois installé, il est accessible depuis la barre d'outils de votre navigateur, vous pourrez ainsi facilement l'activer ou le désactiver en fonction des pages web consultées.

En mode actif, il vous indiquera le nombre de requêtes bloquées sur la page visitée ainsi que le nombre de domaines connectés. Des options supplémentaires vous permettront de bloquer les pop-ups du site, les médias de grande taille, désactiver JavaScript, bloquer les polices d'écriture distantes ou encore désactiver le filtrage esthétique du site.





D'autres fonctionnalités vous permettront encore d'aller plus loin. Vous pourrez par exemple activer le mode *Zappeur* qui vous offrira la possibilité de masquer les éléments de votre choix sur la page consultée, mais aussi recourir au mode *Sélecteur* pour analyser les zones sélectionnées (filtres esthétiques). Le plus intéressant reste cependant les paramètres de l'extension et surtout tout ce qui concerne la gestion des filtres utilisés par ce bloqueur. Un certain nombre de filtres, composés par la communauté d'utilisateurs du module, sont d'ores et déjà activés. Il ne tient qu'à vous de les compléter en créant les vôtres ou à désactiver ceux que vous ne jugeriez pas utiles. Prendre le temps de se constituer une liste blanche des sites à ne pas filtrer peut également être utile, tout comme configurer finement le comportement de l'extension dans les paramètres généraux. Tout cela devrait améliorer votre confort de navigation ! ■

VERSION TESTÉE : 1.18.6 **SITE DU PROJET :** https://old.reddit.com/r/uBlockOrigin/

COMPATIBILITÉ : Firefox, Chrome/Chromium

AUTOMATISER L'INSTALLATION de Debian

SÉBASTIEN LAMY

Outil utilisé : Debian Dnsmasq Preseed

Site du projet : https://www.debian.org/releases/stable/i386/apb.html

DEPUIS LONGTEMPS, LA DISTRIBUTION DEBIAN PERMET UNE INSTALLATION AUTOMATISÉE. CETTE FONCTION EST UTILE POUR INSTALLER PLUSIEURS MACHINES SANS AVOIR À RÉPONDRE AUX QUESTIONS DE L'INSTALLEUR. ELLE SE RÉVÈLE ÉGALEMENT TRÈS PRATIQUE POUR INSTALLER DIRECTEMENT À PARTIR D'UN BOOT RÉSEAU EN PXE ET AINSI NE PLUS AVOIR BESOIN DU SUPPORT D'INSTALLATION (CLÉ USB OU CD-ROM). C'EST CE QUE JE VOUS PROPOSE DE COMPRENDRE ENSEMBLE.

ÉTAPE 1

COMPRENDRE LE FONCTIONNEMENT

Le fichier Preseed est semblable à un fichier de configuration. Il définit un ensemble de champs et de valeurs qui sont tous interprétés au moment de l'installation par l'installeur Debian. On y retrouve donc le choix de la langue, la configuration réseau, l'heure, le partitionnement, etc. Si l'installeur ne comprend pas certaines valeurs, il arrêtera l'installation automatisée pour continuer en mode interactif. Ce fichier peut être chargé, au lancement de l'installation, de différentes façons : directement intégré au support d'installation ; hébergé sur un serveur web dont l'URL est saisie lors d'une installation en mode automatique ; hébergé par un serveur TFTP et chargé au moment du boot PXE.

Attention, toute modification de ce fichier vous obligera à relancer entièrement l'installation sur le client.

ÉTAPE 2

RÉDACTION DE NOTRE PRESEED.CFG DE BASE

Avec l'éditeur de texte de notre choix, nous pouvons commencer la saisie d'un nouveau fichier qui sera nommé preseed.cfg.

Pour démarrer, nous allons configurer la langue et le clavier à utiliser. Pour la langue, il suffit de modifier fr_FR par en_US si nous souhaitions obtenir la langue anglaise. La deuxième instruction, pour le clavier, demande à l'installeur de sélectionner automatiquement celui qui est lié à la langue choisie. Dans notre cas, langue fr_FR, ce sera le clavier français (azerty donc) :

```
d-i debian-installer/locale string fr_FR
d-i keyboard-configuration/xkb-keymap seen
true
```

Pour la configuration réseau, nous demandons la configuration en mode DHCP automatique de notre première interface :

```
d-i netcfg/choose_interface select auto
```

Concernant le dépôt d'installation, nous demandons le miroir ftp.fr.debian.org sur le port 80 et nous indiquons vouloir les paquets de la distribution debian/stretch :

```
d-i mirror/country string manual
d-i mirror/http/hostname string ftp.
fr.debian.org:80
d-i mirror/http/directory string /debian
d-i mirror/codename string stretch
d-i mirror/suite string stretch
d-i mirror/udeb/suite string stretch
```

Ensuite, nous réglons l'heure sur la valeur universelle UTC et nous paramétrons notre fuseau horaire sur Paris. L'heure sera synchronisée via le serveur NTP 0.debian.pool.ntp.org :

```
d-i clock-setup/utc boolean true
d-i time/zone string Europe/Paris
d-i clock-setup/ntp boolean true
d-i clock-setup/ntp-server string 0.debian.
pool.ntp.org
```

Pour le partitionnement des disques, nous demandons un paramétrage classique, le plus simple, qui utilisera le seul disque présent dans son intégralité. Attention, gardez en tête que ce disque sera intégralement formaté pour recevoir uniquement le système Debian. Par conséquent, pensez à utiliser une VM ou faire une sauvegarde du disque avant ! Ce sera également le moment de donner vos instructions pour confirmer les différentes modifications sur le disque :

```
d-i partman-auto/method string regular
d-i partman-auto/choose_recipe select atomic
d-i partman/confirm_write_new_label boolean true
d-i partman/choose_partition select finish
d-i partman/confirm boolean true
d-i partman/confirm_nooverwrite boolean true
```

Si nous devions avoir plusieurs disques, il conviendra d'avertir l'installeur du disque à utiliser pour le partitionnement :

```
d-i partman-auto/disk string /dev/vda
```

À présent, passons aux comptes utilisateurs. Vous constaterez ici que le compte root est créé avec le mot de passe « uiop » ; nous demandons également qu'aucun compte utilisateur ne soit créé :

```
d-i passwd/root-login boolean true
d-i passwd/make-user boolean false
d-i passwd/root-password-crypted password uiop
```

Poursuivons notre rédaction du fichier avec le choix des paquets à installer. Préférez l'installation minimale, qui n'inclue pas l'environnement graphique, et qui sera en effet parfaite pour nos tests. Il est également possible d'autoriser la remontée de nos choix de logiciels, utiles aux mainteneurs de la distribution pour faire évoluer les choix par défaut. Enfin, nous ne demandons pas la mise à jour du système au moment de son installation :

```
tasksel tasksel/first multiselect standard
popularity-contest popularity-contest/
participate boolean true
d-i pkgsel/upgrade select none
```

Le chargeur de démarrage, GRUB, s'installera dans le MBR du premier disque trouvé et proposera tout autre OS détecté :

```
d-i grub-installer/only _ debian boolean true
d-i grub-installer/with _ other _ os boolean true
d-i grub-installer/bootdev string default
```

Il est possible de définir le disque ou les disques que GRUB utilisera :

```
d-i grub-installer/bootdev string /dev/vda
#d-i grub-installer/bootdev string /dev/
vda /dev/vdb
```

La demande de redémarrer le système qui apparaît en fin d'installation n'est pas affichée, le système redémarrera donc automatiquement :

```
d-i finish-install/reboot _ in _ progress note
```

Toutes ces instructions doivent être intégrées dans votre fichier Preseed pour que vous puissiez entamer l'installation de Debian sans que l'installeur ne vous pose de questions intermédiaires.

ÉTAPE 3

CONFIGURATION D'UN SERVEUR TFTP ET DHCP

Pour automatiser nos installations, nous allons nous intéresser à la méthode qui consiste à héberger notre fichier preseed.cfg sur un serveur TFTP. Cela permettra de se passer complètement de tout support d'installation sur nos clients. Depuis longtemps maintenant, les ordinateurs savent booter en réseau. Ce mécanisme, appelé boot PXE, impose qu'un serveur DHCP soit disponible et que ce serveur donne à ses clients l'adresse IP de notre serveur TFTP. Dans mon cas, j'utilise ma machine hôte pour héberger les VM et les services TFTP et DHCP, sous l'adresse IP 10.0.0.1. La VM cliente est associée à un réseau partagé avec l'hôte. Elle récupèrera une adresse IP en 10.0.0.xxx et pourra communiquer librement avec son hôte.

NOTE : Il est tout à fait possible d'utiliser votre serveur DHCP déjà en place dans votre réseau. Pour cela, il faudra le configurer pour que votre client soit bien redirigé vers le serveur TFTP que nous allons configurer. À vous d'adapter la configuration qui suit à votre environnement. De même, vous pouvez utiliser votre serveur TFTP existant, et pour cela, vous adapterez simplement la configuration suivante.

Il existe un logiciel très complet et efficace qui propose les deux fonctionnalités TFTP et DHCP : dnsmasq.

Nous allons voir comment l'installer et modifier sa configuration pour remplir ce rôle :

```
seb@jza:~$ sudo apt install dnsmasq
```

Le fichier de configuration se situe à cet emplacement : /etc/dnsmasq.conf. Ce fichier d'origine est entièrement commenté. Nous le laisserons tel quel, car nous ferons notre configuration dans un fichier séparé : /etc/dnsmasq.d/preseed.conf.

Créons ce nouveau fichier et remplissons-le de la manière suivante :

```
interface=br0
listen-address=10.0.0.1
bind-dynamic
bogus-priv
enable-tftp
tftp-root=/srv/tftp
dhcp-authoritative
dhcp-range=10.0.0.100,10.0.0.199,255.255.255.0
,12h
dhcp-option=option:dns-server,10.0.0.1
dhcp-option=option:router,10.0.0.1
dhcp-boot=pxelinux.0,10.0.0.1
```

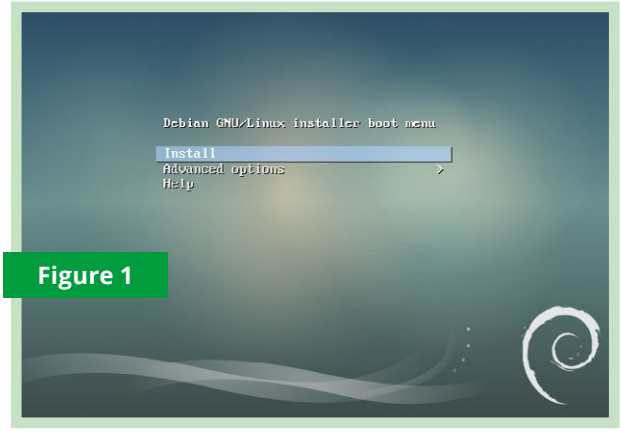


Figure 1

À présent, nous devons lui indiquer de prendre en compte ce nouveau fichier de configuration. Nous éditons le fichier /etc/dnsmasq.conf et nous décommentons la ligne : conf-dir=/etc/dnsmasq.d à la fin du fichier. Nous pouvons à présent redémarrer le service. Nous pouvons passer à notre arborescence dans /srv/tftp (si ce dossier n'existe pas encore, il vous faut le créer).

```
seb@jza:~$ cd /srv/tftp
seb@jza:~$ sudo wget http://ftp.debian.org/
debian/dists/stretch/main/installer-amd64/
current/images/netboot/netboot.tar.gz
seb@jza:~$ sudo tar zxf netboot.tar.gz &&
sudo rm netboot.tar.gz
```

Enfin, nous pouvons lancer notre VM cliente en lui demandant de booter en PXE. Si votre installation s'est bien passée, vous constaterez que votre VM cliente démarre en PXE, qu'elle obtient une configuration réseau du DHCP de dnsmasq et qu'elle affiche le menu de démarrage de notre serveur TFTP.

ÉTAPE 4

CRÉATION D'UNE ENTRÉE TFTP POUR NOTRE FICHIER PRESEED

À ce stade, nous avons tout ce qu'il faut pour démarrer notre première installation automatisée de Debian en démarrant sur le réseau. Il nous manque uniquement un élément dans notre menu TFTP qui permettra le lancement automatique de cette installation. Voyons ensemble comment y remédier.

Nous devons tout d'abord créer le fichier /srv/tftp/debian-installer/amd64/boot-screens/demolp.cfg et le remplir de la manière suivante :

```
label demolp
        menu label ^DEMO Linux Pratique Preseed
        kernel debian-installer/amd64/linux
        append vga=788 initrd=debian-installer/
amd64/initrd.gz auto=true priority=critical
preseed/url=tftp://10.0.0.1/demoLPpreseed.cfg
```

Observez bien qu'à la dernière ligne, append vga=... , il ne faut pas insérer de saut de ligne.

À présent, nous pouvons indiquer que nous disposons de ce nouveau fichier de configuration en ajoutant la ligne suivante au fichier /srv/tftp/pxelinux.cfg/default :

```
# D-I config version 2.0
# search path for the c32 support libraries (libcom32,
libutil etc.)
path debian-installer/amd64/boot-screens/
include debian-installer/amd64/boot-screens/menu.cfg
default debian-installer/amd64/boot-screens/vesamenu.c32
include debian-installer/amd64/boot-screens/demolp.cfg
prompt 0
timeout 0
```

Pour terminer, il faut déposer notre fichier Preseed travaillé en amont directement dans /srv/tftp et le renommer : demoLPpreseed.cfg.

C'est le moment de tester notre configuration. Normalement, après avoir redémarré la VM cliente, toujours en PXE, vous devriez retrouver le menu TFTP muni de sa nouvelle entrée : *DEMO Linux Pratique Preseed*.

NOTE : Nous partons du principe que tout se déroule correctement, sans entrave. Cependant, il peut arriver que vous soyez coincé à une étape. En particulier, le boot en PXE peut s'avérer pénible à mettre en place. Ne vous découragez pas, souvent, c'est un problème de réseau entre la VM et votre hôte qui pose un souci. Pour y pallier, assurez-vous, grâce à un LiveUSB, que la VM communique bien avec votre DHCP et de fait, qu'elle récupère une bonne configuration. Par la suite, vous pourrez terminer le test en lançant dans un terminal un programme client TFTP qui vous donnera des informations plus détaillées.

ÉTAPE 5

UN FICHIER PRESEED PLUS COMPLEXE

Notre première version du fichier preseed.cfg, certes minimale, est suffisante pour automatiser nos installations, mais nous allons tout de même voir comment l'améliorer. Pour cela, nous détaillerons quelques points pertinents destinés à maintenir cette solution en entreprise, dans un environnement de production.

Premièrement, si nous avons un serveur de cache APT comme apt-cacher-ng nous devons l'indiquer avec la ligne suivante :

```
d-i mirror/http/proxy string
http://10.0.0.2:3142
```

Ensuite, pour ne pas écrire en clair le mot de passe du compte root, nous pouvons le mentionner en chiffré :

```
d-i passwd/root-password-crypted password
$6$S6ZUr345RC6hrt.b$0ls3i5XiecKV.lySL8vbXtw
qXe7pATvog0jkiKrfPJp5r1AkAzLD8epDgG2G7mI5w
XuyjpIdS/6HhIBmwhXKQ1
```

Python et son module crypt vous seront nécessaires pour obtenir la version chiffrée du mot de passe « uiop »:

```
seb@jza:~$ python3 -c 'import crypt;
print(crypt.crypt("uiop", crypt.
mksalt(crypt.METHOD _ SHA512)))'
```

Troisièmement, vous disposez de différentes options pour créer un utilisateur supplémentaire :

```
d-i passwd/make-user boolean true
d-i passwd/user-fullname string Moi Sympa
d-i passwd/username string moi
d-i passwd/user-password password 123456
d-i passwd/user-password-again password 123456
```

Enfin, pour mettre à jour le système et ajouter des paquets, il faut procéder comme suit :

```
d-i pkgsel/upgrade select full-upgrade
d-i pkgsel/include string curl sudo
```

ÉTAPE 6

REMPLACER LE COMPTE ROOT PAR UN COMPTE SUDO

Comme le fait la distribution Ubuntu (et d'autres) depuis quelque temps, nous pouvons éviter la création du compte root avec un mot de passe. Pour cela, nous allons créer un compte utilisateur qui aura le privilège d'exécuter toutes les commandes normalement réservées à l'utilisateur root :

```
d-i passwd/root-login boolean false
```

Bien entendu, nous devons avoir saisi les instructions de création d'un compte utilisateur.

ÉTAPE 7

MÉTHODES DE PARTITIONNEMENT

Dans notre exemple initial, nous avons choisi d'utiliser l'intégralité du premier disque et de laisser l'installeur créer les partitions idéales. Il existe d'autres méthodes qui sont proposées par défaut comme LVM et RAID. De même qu'il est possible de rendre la partition /home indépendante ou bien de séparer /home, /var et /tmp.

Les configurations suivantes sont à intégrer dans le Preseed, en remplaçant les valeurs en place.

Premier exemple : l'utilisation de LVM avec une partition /home indépendante :

```
d-i partman-auto/method string lvm
d-i partman-auto/choose _ recipe select home
```

Deuxième exemple : sans LVM, mais avec les partitions /home, /var et /tmp séparées :

```
d-i partman-auto/method string regular
d-i partman-auto/choose _ recipe select multi
```

À noter que si vos disques ont précédemment été formatés avec LVM ou en RAID, les instructions suivantes détruisent ces configurations existantes :

```
d-i partman-lvm/device_remove_lvm boolean true
d-i partman-md/device_remove_md boolean true
d-i partman-lvm/confirm boolean true
d-i partman-lvm/confirm_nooverwrite boolean true
```

Pour aller plus loin dans le partitionnement, vous pouvez utiliser le partitionnement expert, dont la prise en main est réservée à un public averti, ce que vous allez immédiatement constater.

ÉTAPE 8

PARTITIONNEMENT EXPERT

La syntaxe utilisée dans notre Preseed concernant la gestion manuelle des partitions peut s'avérer obscure. Pourtant, elle permet de partitionner très finement nos disques.

Prenons l'exemple du partitionnement suivant :

```
d-i partman-auto/method string regular
d-i partman-auto/choose_recipe select perso
d-i partman-auto/expert_recipe string \
  perso :: \
    256 1 256 ext2 \
      $primary{ } $bootable{ } method{ format
} format{ }
      use_filesystem{ } filesystem{ ext3 }
mountpoint{ /boot } \
    . \
    512 1 512 linux-swap \
      method{ swap } format{ } \
    . \
    4096 2 -1 ext4 \
      method{ format } format{ } use_
filesystem{ } filesystem{ ext4 } mountpoint{
/ } \
    .
```

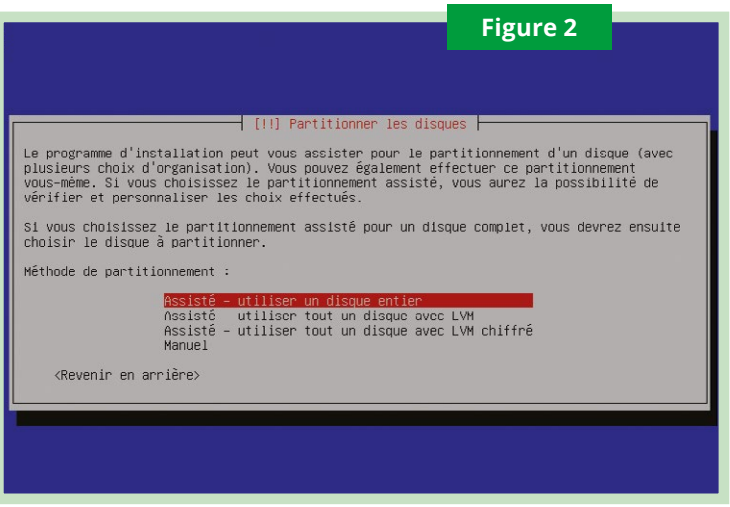
Nous avons demandé une installation sans LVM afin d'obtenir le résultat suivant : une première partition de 256Mo formatée en ext2 qui aura /boot pour point de montage ; une seconde partition pour le swap de 512Mo ; une dernière partition d'une taille minimum de 4Go qui utilisera tout l'espace restant et qui accueillera la racine du système sera formatée en ext4.

À présent, voyons un exemple de partitionnement en RAID 1 pour nos deux partitions système uniquement :

```
d-i partman-auto/disk string /dev/vda /dev/vdb
d-i partman-auto/method string raid
d-i partman-auto/choose_recipe select multiraid
d-i partman-auto/expert_recipe string       \
    multiraid ::                            \
        2048 1 2048  linux-swap $primary{ } method{
swap } format{ } . \
        4096 2 -1    raid       $primary{ } method{
raid } .
d-i partman-auto-raid/recipe string     \
    1 2 0 ext4 / /dev/vda2#/dev/vdb2 .

d-i mdadm/boot_degraded boolean true
d-i partman-md/confirm boolean true
d-i partman-md/confirm_nooverwrite boolean true
d-i grub-installer/bootdev string /dev/vda /dev/vdb
```

La gestion est ici plus complexe puisqu'il faut définir à l'avance les disques sur lesquels nous souhaitons effectuer un partitionnement. À présent, nous pouvons définir les partitions à savoir une partition *swap* de 2Go et le reste que nous allons dédier justement au RAID. C'est à ce moment que le partitionnement RAID intervient pour créer les systèmes de fichiers ainsi que leurs points de montage. Enfin, nous pouvons procéder à l'installation de GRUB sur les deux disques.

Figure 2



Si l'installeur rencontre un problème lors de l'interprétation de vos instructions, il mettra fin à l'installation en mode automatique et reprendra en mode interactif comme vous pouvez l'observer sur la Figure 2, page précédente.

NOTE : Il arrive que GRUB ne s'installe pas sur tous les disques demandés. Je vous encourage fortement à tester cela au démarrage. Si le résultat est négatif, il convient de l'installer manuellement avec la commande grub-install <device>.

ÉTAPE 9

FRAGMENTATION DU PRESEED

Imaginons que nous souhaitions servir plusieurs Preseed avec des paramètres différents. Il est probable qu'un certain nombre de valeurs restent identiques. Parmi elles, nous pouvons relever la langue, le clavier, le réseau en DHCP, etc. En conséquence, nous devons conserver tous les paramètres communs dans notre premier fichier, lequel s'appellera toujours demoLPpreseed.cfg, mais nous allons devoir créer d'autres fichiers Preseed qui contiendront seulement les valeurs qui seront particulières au cas.

Notre premier cas porte sur le lancement de l'installation automatisée en définissant le partitionnement simple, sous le nom partauto-demoLPpreseed.cfg :

```
d-i preseed/include string demoLPpreseed.cfg
d-i partman-auto/method string regular
d-i partman-auto/choose _ recipe select home
d-i grub-installer/bootdev string default
```

La première instruction fait appel aux paramètres communs dans le Preseed de base demoLPpreseed.cfg. Les trois suivantes définissent le partitionnement simple comme nous l'avons expliqué plus haut dans cet article.

Notre deuxième cas concerne le lancement de l'installation avec cette fois un partitionnement plus complexe en RAID1, sous le nom partraid1-demoLPpreseed.cfg :

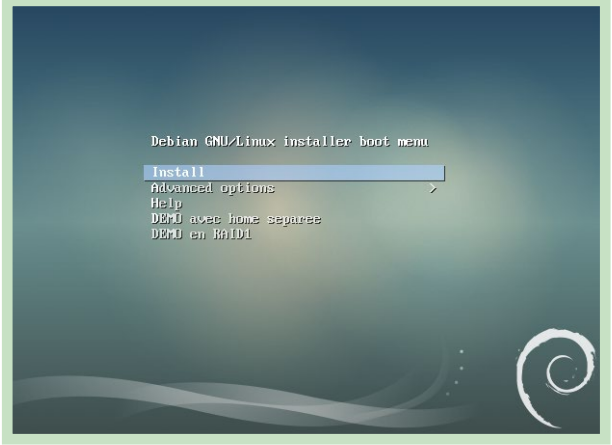
```
d-i preseed/include string demoLPpreseed.cfg

d-i partman-auto/disk string /dev/vda /dev/vdb
d-i partman-auto/method string raid
d-i partman-auto/choose _ recipe select
multiraid
d-i partman-auto/expert _ recipe string       \
    multiraid ::                              \
        2048 1 2048  linux-swap $primary{ }
method{ swap } format{ } . \
        4096 2 -1    raid       $primary{ }
method{ raid } .
d-i partman-auto-raid/recipe string     \
    1 2 0 ext4 / /dev/vda2#/dev/vdb2 .

d-i mdadm/boot _ degraded boolean true
d-i partman-md/confirm boolean true
d-i partman-md/confirm _ nooverwrite boolean
true
d-i grub-installer/bootdev string /dev/vda
/dev/vdb
```

La première instruction est identique, elle fait appel au Preseed qui contient les paramètres communs à nos deux cas d'installation. Les suivantes reprennent la définition d'un RAID1 que nous avons détaillé précédemment.

Il nous faut maintenant réécrire notre entrée de menu TFTP pour afficher nos deux installations possibles. Pour cela, nous devons éditer le fichier /srv/tftp/debian-installer/amd64/boot-screens/demolp.cfg :

```
label demopartauto
        menu label ^DEMO avec home separee
        kernel debian-installer/amd64/linux
        append vga=788 initrd=debian-
installer/amd64/initrd.gz auto=true
priority=critical preseed/url=tftp://10.0.0.1/
partauto-demoLPpreseed.cfg

label demoraid1
        menu label ^DEMO en RAID1
        kernel debian-installer/amd64/linux
        append vga=788 initrd=debian-
installer/amd64/initrd.gz auto=true
priority=critical preseed/url=tftp://10.0.0.1/
partraid1-demoLPpreseed.cfg
```

Après avoir redémarré notre VM cliente en PXE, nous découvrons nos deux nouvelles entrées, chacune lançant l'installation qui lui correspond. En fin d'installation, nous obtenons le partitionnement souhaité ainsi que les configurations de la langue, du réseau, etc. qui sont communes à nos deux cas.



ÉTAPE 10

EXÉCUTER DES COMMANDES SHELL

L'installation automatisée prévoit que nous puissions exécuter des commandes en fin d'installation, avant le redémarrage. Ces commandes nous servent à configurer le système comme si nous étions devant son terminal. Pour illustrer cette fonctionnalité, nous allons demander à l'installeur d'installer et de configurer un agent Puppet, chargé de configurer le système au redémarrage. Ajoutons simplement ces instructions à la fin de notre Preseed commun aux deux installations :

```
# Puppet agent install and configure
d-i preseed/late _ command string \
in-target wget http://apt.puppetlabs.com/
puppet5-release-stretch.deb ; \
in-target dpkg -i puppet5-release-stretch.
deb ; \
in-target apt update ; \
in-target apt install -y puppet-agent ; \
echo -e "[main]\nserver = puppetmaster.
amatiq.fr\n" >> /target/etc/puppetlabs/
puppet/puppet.conf ;
```

L'instruction utilisée est late_command. Il est impératif que sa valeur tienne sur une ligne. Ce comportement est identique à l'écriture du partitionnement vue plus haut. Dans le détail, rien d'extraordinaire dans la mesure où l'instruction in-target exécute des commandes dans l'environnement du client qui vient tout juste d'être installé. ■

POUR ALLER PLUS LOIN

Nous venons de passer en revue les instructions minimales à écrire pour installer Debian de façon automatisée non interactive. Sachez qu'il existe d'autres instructions dont vous pourriez avoir besoin. Ainsi, je vous recommande de parcourir la documentation Debian sur le sujet ici https://www.debian.org/releases/stable/i386/apb.html.fr pour les découvrir. De même, si vous utilisez Ubuntu ou une autre distribution basée sur Debian, vous pourrez être confronté à quelques différences pour lesquelles une recherche spécifique sera nécessaire. Par ailleurs, la famille Debian n'est pas la seule à proposer les installations automatisées. En effet, si vous êtres familiers des Red Hat, pensez aux Kickstart qui jouent le rôle de Preseed. Il ne me reste qu'à vous souhaiter une bonne installation !

L'automatisation dans vos systèmes d'informations avec Ansible

THOMAS BOURCEY

S'IL EST DES OUTILS QUI ONT EU UN EFFET RÉVOLUTIONNAIRE POUR LES ADMINISTRATEURS SYSTÈMES, CE SONT BIEN LES OUTILS D'AUTOMATISATION. ON PEUT CITER PAR EXEMPLE CHEF OU PUPPET AVEC LESQUELS J'AI COMMENCÉ MES PREMIÈRES TÂCHES D'AUTOMATISATION, MAIS IL FAUT SE RENDRE À L'ÉVIDENCE QUE DEPUIS QUELQUE TEMPS, C'EST ANSIBLE QUI (SELON MOI) A GAGNÉ LA GUERRE DES OUTILS D'AUTOMATISATION. MAIS QU'EST-CE DONC QU'ANSIBLE ET COMMENT FONCTIONNE-T-IL ?

1. PRÉSENTATION DE L'OUTIL

Ansible est une plateforme libre et open source de gestion de configuration pour vos serveurs. Je simplifie un peu, mais cet outil vous permettra donc d'automatiser l'installation et la maintenance de vos serveurs, voire d'infrastructures complètes, et vous permettra de gagner du temps pour l'administration de votre système d'information. Créée et développée en 2012 par Michael DeHaan, rachetée en 2015 par Red Hat, la plateforme Ansible tire son nom du terme *Ansible* choisi par Ursula Le Guin dans ses romans de science-fiction pour désigner un moyen de communication plus rapide que la lumière (source : Wikipédia).

Ansible propose plusieurs arguments par rapport à la concurrence :

- Sans agent : si la plupart des autres outils nécessitent un client sur la machine cible, Ansible n'a besoin que d'une simple connexion SSH, il est ce qu'on appelle *agent-less*.
- C'est simple : comme nous le verrons plus tard dans l'article, Ansible utilise une syntaxe très simple écrite en YAML, appelée *Playbooks*. YAML (*Yet Another Mark-up Language*) est un langage de sérialisation des données lisible par l'homme. Vous n'avez pas besoin de compétences particulières en programmation pour coder et comprendre les playbooks. Il est très facile d'installer et d'exécuter les tâches dans l'ordre.
- Modulaire : il est modulaire, car vous n'avez besoin que d'un programme par script. Vous pouvez ainsi répartir vos programmes sur différents serveurs.
- Efficace : aucun logiciel supplémentaire n'est requis sur vos serveurs, ce qui signifie plus d'espace pour vos ressources.
- Puissant et flexible : doté de puissantes fonctionnalités vous permettant de modéliser même les flux de travail informatique complexes en moins de temps, ainsi que de gérer l'infrastructure, les réseaux, les systèmes d'exploitation et les services déjà utilisés.

2. DÉBUTER AVEC ANSIBLE

Ansible utilise une simple connexion SSH pour réaliser les différentes actions à mener. Ces actions sont codées dans des playbooks grâce à la syntaxe YAML. L'avantage majeur de ce langage est qu'il est extrêmement simple à prendre en main.

Voici quelques terminologies qui seront utilisées pendant la lecture de l'article :

- *Control Machine* : la machine maître, celle depuis laquelle nous lancerons nos playbooks ;
- *Host inventory* : les hôtes sur lesquels seront lancées les actions ;
- *Playbooks* : fichier décrivant des tâches qui seront lancées sur nos hôtes ;
- *Tâches* : ce sont les actions que nous ferons sur nos hôtes (ex. : installer un package) ;
- *Rôles* : ensemble de tâches réparties dans différents dossiers. Ces dossiers seront des rôles (je simplifie, mais nous en parlerons plus en détail dans la fin de l'article) ;
- *Modules* : les modules sont appelés par des tâches. Ce sont les modules qui permettent d'exécuter une action (`apt`, `copy`, `cmd`...). Voir la liste de tous les modules existants : **https://docs.ansible.com/ansible/latest/modules/modules_by_category.html**.

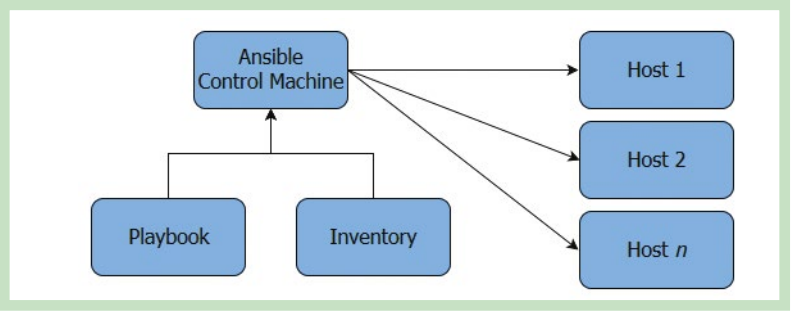


FIGURE 1. Schéma d'architecture Ansible.

2.1 Installation et configuration d'Ansible

2.1.1 Installation

La première chose à faire avant tout est, vous vous doutez bien, d'installer Ansible. Pour cela, nous l'installerons sur ce qu'on appelle un Control Machine. Celui-ci sera notre point d'entrée. Bien entendu, dans la pratique, il faut que cette machine ait accès via SSH aux *Managed Nodes*, aussi appelés hôtes.

Ansible est disponible sur plusieurs systèmes d'exploitation. Ici, on l'installera sur Debian 9. Pour les autres OS, je vous redirige vers le site officiel : **https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html**.

Pour avoir la dernière version disponible sur les dépôts du mainteneur officiel, ajoutez le dépôt ci-dessous dans le fichier /etc/apt/sources.list.d/ansible.list qu'il vous faudra au préalable créer.

```
deb http://ppa.launchpad.net/ansible/
ansible/ubuntu trusty main
```

Puis importez la clé GPG :

```
sudo apt-key adv --keyserver keyserver.
ubuntu.com --recv-keys 93C4A3FD7BB9C367
```

Enfin, on installe Ansible :

```
sudo apt update
sudo apt install ansible
```

2.1.2 Configuration d'Ansible

Maintenant que vous avez installé Ansible sur votre control machine, la première chose que nous allons faire est d'ajouter l'adresse d'un ou plusieurs hôtes. Par défaut, le fichier que nous allons renseigner est le fichier /etc/ansible/hosts.

Nous allons regrouper les serveurs par « domaine fonctionnel » dans le fichier hosts (libre à vous pour votre utilisation de grouper les serveurs comme bon vous semble). Commençons par éditer le fichier hosts. Celui par défaut est déjà rempli d'exemples. Je vous conseille de jeter un œil sur toutes les options

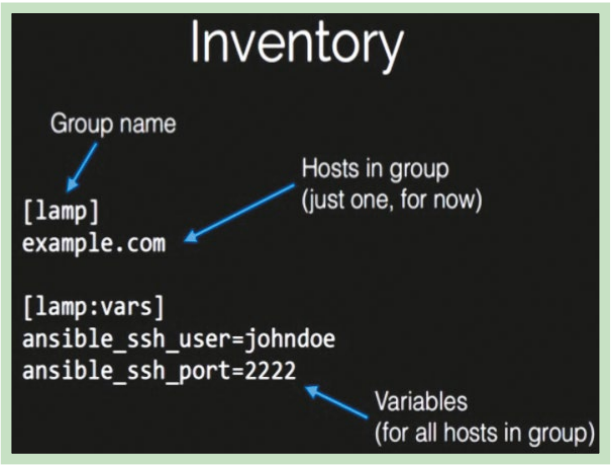


FIGURE 2

disponibles dans le fichier d'inventaire des hôtes (host inventory) ainsi que sur la documentation officielle : **http://docs.ansible.com/ansible/intro_inventory.html**.

Exemple :

```
# Fichier de configuration des hosts
Ansible
# les crochets permettent de définir des
groupes
# un serveur peut appartenir à plusieurs
groupes

[production:children]
webservers
dbservers
proxies

[webservers]
foo.example.com
bar.example.com

[webservers:vars]
ansible _ ssh _ user=johndoe
ansible _ ssh _ port=2222

[dbservers]
db[01:03].example.com

[dbservers:vars]
pgsql _ bind _ nic=eth1

[proxies]
192.168.1.1
```

2.1.3 Configuration de SSH

Comme je l'expliquais plus haut, les échanges entre Ansible et les hôtes se font en SSH. Afin de nous faciliter la vie (encore et toujours), nous allons créer une paire de clés publique/privée et l'envoyer sur nos serveurs pour que l'application ne demande plus de mot de passe. Vous remarquerez qu'ici tout sera fait en tant que root pour simplifier l'article, mais il est bien entendu possible d'utiliser un simple utilisateur et de lancer des commandes sudo.

```
ssh-keygen -t rsa -b 4096
ssh-copy-id -i ~/.ssh/id _ rsa.pub root@
webserver01
ssh-copy-id -i ~/.ssh/id _ rsa.pub root@
webserver02
ssh-copy-id -i ~/.ssh/id _ rsa.pub root@
dbserver01
[...]
```

2.1.4 Notre première commande

Maintenant que notre fichier /etc/ansible/hosts est correctement renseigné, nous allons tester que tout fonctionne bien. Pour cela, nous allons envoyer un ping sur nos serveurs et nous devrions recevoir un pong ;-) Ici, c'est le module ping qui est utilisé, rien à voir avec le ping ICMP.

```
ansible all -m ping -u root
```

2.2 Ansible CLI : la ligne de commandes

Tout d'abord, sachez que vous pouvez utiliser Ansible de deux manières différentes :

- en ligne de commandes (avec la commande ansible) ;
- à partir de playbooks (avec la commande ansible-playbook).

À la différence des Playbooks, la ligne de commandes Ansible est utilisée pour définir et exécuter une seule tâche sur un ou plusieurs hôtes. Nous ferons ainsi du one-shot. Par exemple, copier un fichier sur des serveurs, rebooter un groupe de serveurs, etc. Mais on ne l'utilisera qu'une seule fois pour un besoin plutôt ponctuel.

Quelques exemples :

- copier un fichier sur tous vos serveurs web (une mise à jour du fichier httpd.conf par exemple) :

```
ansible webservers -m copy -a "src=/
home/tom/webservers/httpd.conf dest=/etc/
apache2/httpd.conf"
```

- rebooter un groupe de serveurs :

```
ansible dbservers -b -a "reboot"
```

- installer un paquet sur un serveur :

```
ansible webserver02 -m apt -a "name=htop
state=present"
```

Au final, la ligne de commandes sera identique en terme de syntaxe à ce qu'il vous faudra utiliser dans les playbooks. Nous allons voir cela plus en détail dans le chapitre suivant. Il est possible d'aller plus loin avec les lignes de commandes, mais nous n'en parlerons pas dans cet article. Pour cela, il y a le man (on ne le présente plus) ou la documentation officielle : **https://docs.ansible.com/command_line_tools.html**.

3. LES PLAYBOOKS

3.1 Utiliser les Playbooks

Un Playbook, pour faire simple, est un fichier qui va automatiser des tâches de manière séquentielle et/ou conditionnelle (par exemple, installer un programme, mais ne l'installer que s'il n'est pas

déjà installé). Comme je le disais précédemment, le Playbook est écrit en YAML, donc très simple à lire, et à écrire.

Globalement, un playbook est composé de tâches comme ceci :

```
- name: Texte qui décrit votre tâche
  module: option=value
```

Un exemple sorti de son contexte, mais compréhensible :

```
- name: Upgrade packages
  apt:
    update_cache: yes
    upgrade: safe
```

Dans cet exemple, nous utilisons le module apt pour mettre à jour notre système, soit l'équivalent de la commande : apt-get update && apt upgrade.

Il est tout à faire possible de créer le premier fichier Playbook dans le répertoire de votre choix, cependant, comme souvent d'ailleurs, il est préférable de suivre les bonnes pratiques d'Ansible quant à la structure des répertoires et des fichiers. J'y reviendrai un peu plus tard.

Mais comment fonctionne donc un playbook ? Quelle syntaxe adopter ? Rien de bien compliqué. Les playbooks utilisent une syntaxe simple : on définit les hôtes ; on définit les éventuelles variables ; on définit notre tâche ou nos tâches. Chaque tâche possède un nom et fait appel à un ou plusieurs modules.

Nous allons utiliser un Playbook pour installer notre serveur web. Rien de compliqué : un serveur apache, quelques modules, notre fichier virtual host.

Ci-dessous, un exemple d'un Playbook :

```
# Les documents YAML commencent toujours
par "---"
---

# Le nom de l'hôte ou du groupe concerné
par le playbook. Ici nos serveurs web
- hosts: webservers

  # Les variables
  vars:
    http_port: 80
    domain: tomzone.fr

  # become et become_user permettent
d'indiquer que l'on veut réaliser nos
opérations en tant que root
  become: true
  become_user: root

  # La liste de nos tâches
  tasks:

  # On nomme nos tâches
  - name: Update server
    # nom du module à utiliser
    apt:
      update_cache: yes
      upgrade: full

  # On installe Apache
  - name: Install Apache
    # il existe une syntaxe alternative,
plus condensée
    apt:
      name: apache2
      state: latest

  # On active le mod rewrite s'il ne l'est
pas déjà
  - name: Enabled mod_rewrite
    apache2_module:
      name: rewrite
      state: present

  # On utilise le module "template" pour
copier les fichiers vhosts sur le serveur
  # Les variables à l'intérieur du fichier
seront remplies par les valeurs de nos
variables définies en haut du playbook
  - name: Copy vhost Apache config file
    template:
```

```
      src: /home/tom/Ansible/templates/
vhost.conf.j2
      dest: /etc/apache2/sites-available/
tomzone.fr.conf

  - name: enable vhost
    command: a2ensite tomzone.fr

  - name: restart apache
    service:
      name: httpd
      state: restarted
```

Et notre fichier vhost apache vhost.conf.j2 :

```
<VirtualHost *:{{ http _ port }}>
    ServerAdmin webmaster@{{ domain }}
    ServerName {{ domain }}
    ServerAlias www.{{ domain }}
    DocumentRoot /var/www/{{ domain }}

    ErrorLog ${APACHE _ LOG _ DIR}/error.{{
domain }}.log
    CustomLog ${APACHE _ LOG _ DIR}/access.{{
domain }}.log combined

    <Directory /var/www/{{ domain }}/>
       Options -Indexes +FollowSymLinks
       AllowOverride All
    </Directory>
</VirtualHost>
```

Comme vous pouvez le voir, le playbook décrit parfaitement notre tâche et le module « template » va charger sur le système distant notre fichier virtual host. Chaque {{ variable }} du fichier vhost.conf.j2 est parsée par Ansible et la valeur correspondante est injectée dans le fichier grâce notamment aux variables que nous avons préalablement renseignées dans notre playbook. Magique, non ?

Pourquoi une extension en *.j2 me direz-vous ? Jinja2 est un moteur de template pour le langage de programmation Python. L'avantage de Jinja2 est qu'il permet de gérer les boucles ou les listes de variables. C'est pourquoi il est recommandé d'utiliser Jinja2 pour vos templates de fichier qui doivent être parsés avec Ansible afin d'intégrer vos variables.

Sachez que vous pouvez lancer vos playbooks avec l'option --syntax-check, qui comme son nom l'indique, s'assure qu'il n'y a pas d'erreur dans la syntaxe de vos playbooks. Vous pouvez également utiliser l'option --check pour simuler un play sans effectuer aucun changement.

3.2 Les variables

Comme nous l'avons vu dans le chapitre précédent, il est possible de définir manuellement des variables dans nos playbooks. Nos variables Ansible, au même titre que des variables dans un script bash, vont nous permettre de définir des valeurs de configuration pour les différentes tâches. Mais Ansible fournit également de nombreuses variables avec ces différents modules.

Dans Ansible, les variables peuvent être définies à différents endroits : dans les playbooks, dans les rôles, dans l'inventaire, dans des fichiers séparés, en ligne de commandes, etc. Concernant les variables et leur organisation, il faut savoir qu'il y a une priorité, selon où se trouve la variable, cette notion de priorité est très bien détaillée dans la documentation d'Ansible : **https://docs.ansible.com/ansible/latest/user_guide/playbooks_variables.html**.

Prenons l'exemple des variables fournies par le module setup. Celui-ci permet de collecter de données sur l'hôte distant :

```
$ ansible webserver01  -m setup
webserver01 | SUCCESS => {
    "ansible _ facts": {
        "ansible _ all _ ipv4 _ addresses": [
            "10.31.0.133"
        ],
        "ansible _ all _ ipv6 _ addresses": [
            "fe80::471:85ff:fe22:3c70"
        ],

        [...]

        "ansible _ architecture": "x86 _ 64",
        "ansible _ bios _ date": "08/24/2006",
        "ansible _ bios _ version":
"4.2.amazon",

        [...]

        "ansible _ lsb": {
            "codename": "stretch",
```

```
            "description": "Debian GNU/
Linux 9.6 (stretch)",
            "id": "Debian",
            "major _ release": "9",
            "release": "9.6"
        },
        [...]
}
```

Pour des raisons de longueur, nous avons volontairement découpé la sortie de la commande ansible, mais comme vous pouvez le constater, le module setup nous renvoie de nombreuses variables système. Rien de plus simple pour les utiliser dans nos playbooks, il suffit de les appeler comme ceci :

```
# variable simple
{{ ansible _ architecture }}

# variable pour accéder à une propriété
{{ ansible _ lsb.codename }}

# variable pour accéder à un tableau
(première propriété)
{{ ansible _ all _ ipv4 _ addresses[0] }}
```

Mais la ou les variables deviennent vraiment puissantes quand on peut les définir de manière dynamique. Par exemple, vous souhaitez récupérer la liste des virtuals hosts installés sur l'un de vos serveurs et mettre cette liste dans des variables pour les réutiliser plus tard :

```
---
- hosts: webservers
  tasks:
    - name: List Virtuals Hosts
      command: /bin/ls /etc/apache2/sites-
enabled/
      register: vhosts
```

La valeur vhosts contient la liste des éléments présents dans /etc/apache2/sites-enabled/ renvoyés par la commande ls. Très pratique, n'est-ce pas ?

Mais, me direz-vous, comment être sûr de ce qu'il y a dans nos variables ? Grâce à l'utilisation de l'objet debug. En reprenant l'exemple précédent :

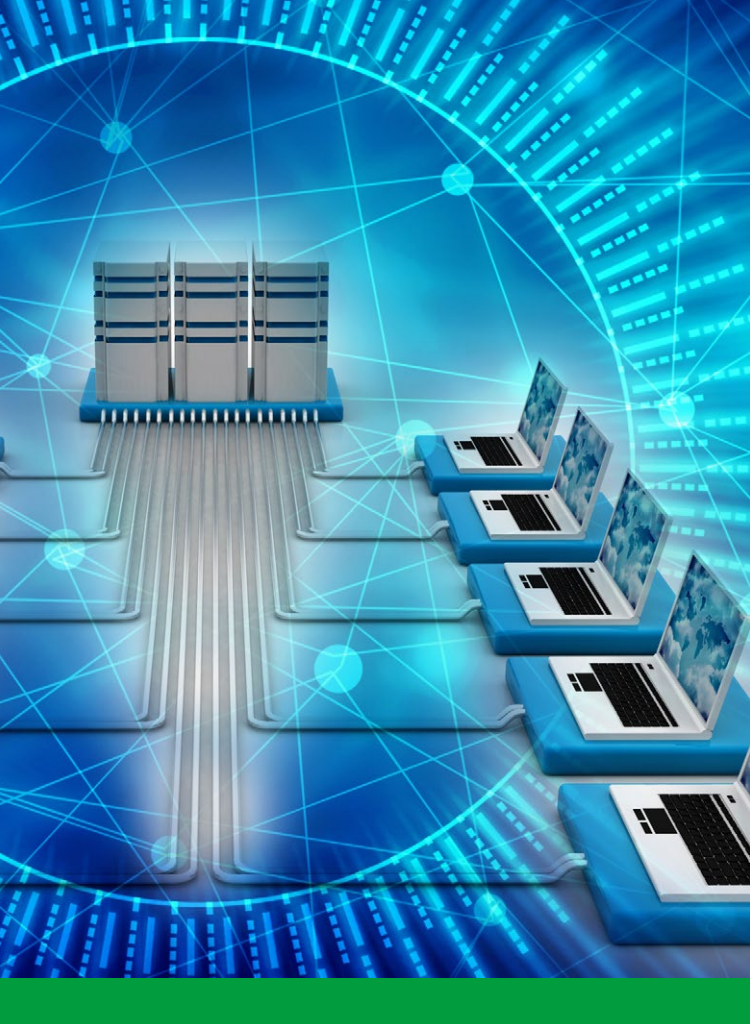
```
# test.yml
---
- hosts: webservers
  tasks:
    - name: List Virtuals Hosts
      command: /bin/ls /etc/apache2/sites-
enabled/
      register: vhosts
    - debug: msg="{{ vhosts }}"
```

On tape dans un terminal la commande ci-dessous :

```
ansible-playbook test.yml
```

Résultat de la commande :

```
PLAY [webservers] ***************************
**************************************

TASK [Gathering Facts] **********************
**************************************
ok: [webserver01]

TASK [determine vhosts] *********************
**************************************
changed: [webserver01]

TASK [debug] ********************************
**************************************
ok: [webserver01] => {
    "msg": {
        "changed": true,
        "cmd": [
            "/bin/ls",
            "/etc/apache2/sites-enabled/"
        ],
        "delta": "0:00:00.002109",
        "end": "2019-03-24 21:54:46.307948",
        "failed": false,
        "rc": 0,
        "start": "2019-03-24 21:54:46.305839",
        "stderr": "",
        "stderr _ lines": [],
        "stdout": "000-default.conf\
ntomzone.fr.conf",
        "stdout _ lines": [
            "000-default.conf",
            "tomzone.fr.conf"
        ]
    }
}

PLAY RECAP **********************************
**************************************
webserver01                : ok=3
changed=1    unreachable=0    failed=0
```



Le résultat de la commande est relativement verbeux. Comme vu précédemment, pour récupérer la valeur de notre vhost tomzone.fr, il faudra appeler la variable {{ vhosts.stdout_lines[1] }}. Si nous avions voulu récupérer la valeur 000-default.conf, alors notre variable, vous l'aurez compris, serait : {{ vhosts.stdout_lines[0] }}.

3.3 Les Handlers

Les handlers ressemblent aux tâches dans un playbook Ansible, mais ne sont exécutés que si la tâche contient la directive notify et que la tâche reçoit le signal que quelque chose a été changé. Par exemple, toujours avec notre serveur web, plutôt que de redémarrer Apache à chaque fois qu'une tâche est exécutée, alors, avec les handlers, le service redémarrera uniquement s'il y a eu un changement de configuration. Autre avantage et pas des moindres, l'action en question ne sera lancée qu'après l'exécution de tous les blocs tâches.

Reprenons notre exemple avec le fichier yaml du serveur web :

```
---
- hosts: webservers

  vars:
    http_port: 80
    domain: tomzone.fr

  become: true
  become_user: root

  tasks:
  [...]
  - name: Enabled mod_rewrite
    apache2_module:
      name: rewrite
      state: present
    notify:
      - restart apache2

  - name: Copy vhost Apache config file
    template:
      src: /home/tom/Ansible/templates/
vhost.conf.j2
      dest: /etc/apache2/sites-available/
tomzone.fr.conf


  - name: enable vhost
    command: a2ensite tomzone.fr
    notify:
      - restart apache2

- name: disable default vhost
    command: a2dissite 000-default
    notify:
      - restart apache2

handlers:
  - name: restart apache2
    service:
      name: apache2
      state: restarted
```

Dans l'exemple ci-dessus, nous allons redémarrer Apache dès qu'on va activer le mod_rewrite, dès qu'on va activer le virtual host tomzone.fr et supprimer le virtual host par défaut. Si nous avions référencé directement le module apache2 avec l'instruction de redémarrer dans chacun des blocs de tâches, nous aurions redémarré Apache trois fois.

Ceci reste qu'un simple exemple, car dans notre cas tout se trouve dans un seul playbook, ce qui limite

l'intérêt d'utiliser les handlers, mais dès que vos playbooks commenceront à être plus complexes, séparés en plusieurs briques logiques, alors vous verrez l'intérêt de les utiliser.

4. LES RÔLES ANSIBLE

Un playbook peut être monolithique ou divisé en plusieurs parties. Cela dépendra bien évidemment des tâches à réaliser, mais dès qu'on commence vraiment à mettre les mains dans les playbooks, qu'on a plusieurs tâches différentes, cela devient vite le bazar. C'est là qu'interviennent les rôles Ansible.

Les rôles définis par Ansible sont un ensemble de tâches qui s'assurent de la présence/absence d'une fonctionnalité spécifique (allant de la création d'un utilisateur Linux, en passant par l'installation et la configuration d'un serveur LAMP et jusqu'à l'instanciation d'un cluster Kubernetes d'une centaine de nœuds). Chaque rôle doit être capable de fonctionner en autonomie (sans compter les dépendances vers d'autres rôles) et inclut pour ça, dans son arborescence, un certain nombre de choses.

Ansible met à disposition un outil en CLI afin de préparer l'arborescence d'un rôle vide. Nous allons créer pour commencer le rôle common. Ce rôle sera notre base pour les serveurs.

```
ansible-galaxy init common
```

La commande va générer une arborescence telle que :

```
└── common
    ├── defaults
    │   └── main.yml
    ├── files
    ├── handlers
    │   └── main.yml
    ├── meta
    │   └── main.yml
    ├── README.md
    ├── tasks
    │   └── main.yml
    ├── templates
    ├── tests
    │   ├── inventory
    │   └── test.yml
    └── vars
        └── main.yml
```

Nous avons donc :

- un fichier README.md pour documenter l'utilisation du rôle ;
- un dossier defaults pour définir les variables par défaut ;
- un dossier files où l'on mettra les fichiers hors playbooks dont on aura besoin (par exemple, des fichiers de configuration) ;
- un dossier handlers qui recensera les événements Ansible liés à ce rôle ;
- un dossier meta qui contient, comme son nom l'indique, les métadonnées de ce rôle telles que son auteur ou ses dépendances à d'autres rôles ;
- le dossier le plus important tasks contient les tâches à exécuter ;
- on mettra dans le dossier template les templates Jinja2 qui permettent de générer des fichiers textes de façon procédurale comme vu précédemment ;
- et enfin, on déclare les variables du rôle dans vars (ce qui supplante évidemment les définitions du dossier defaults).

L'utilisation des rôles Ansible semble donc parfaite pour s'assurer qu'une partie du code sera le plus générique possible et réutilisable, pour peu que l'on suive quelques principes de base.

Il n'y a pas d'obligation quant à l'utilisation des répertoires. S'ils sont vides, nous les supprimerons. Cela ne sert à rien de les conserver pour rien.

4.1 Cas concret : installer un serveur web LAMP

Pour illustrer cet article, rien de mieux que de vous montrer comment définir dans la bonne pratique vos rôles et playbooks pour installer un serveur LAMP « de base » sur Debian (c'est pour l'exemple). En nous servant de ce que nous avons déjà vu précédemment, nous allons créer plusieurs rôles qui vont permettre au final de :

- configurer un serveur ;
- installer un serveur web LAMP ;
- instancier une page web (et vous verrez qu'il serait très simple d'automatiser le déploiement d'un blog WordPress dans le futur).

Pour cet atelier pratique, nous avons hébergé le code sur un GitHub. Vous pourrez trouver tous les fichiers nécessaires ici : **https://github.com/SckyzO/LinuxPratique-Ansible**.

```
git clone https://github.com/SckyzO/LinuxPratique-Ansible
```

Pour commencer, nous allons nous servir du rôle common créé précédemment (et vous verrez, nous n'utiliserons pas tous les répertoires créés par la commande ansible-galaxy init, car comme je le disais, si les dossiers ne sont pas utiles, autant les supprimer). Nous allons créer des tâches et utiliser des fichiers de configuration.

```
# common/tasks/main.yml
---
# Mise à jour du système
- name: Keep server up to date
  apt:
    update_cache: yes
    upgrade: dist

# On copie le fichier sources.list avec tous les dépôts nécessaires
- name: Add all required repositories
  copy:
    src: sources.list.j2
    dest: /etc/apt/sources.list
    mode: 0644

# On installe les paquets nécessaires
- name: Install base packages and utils for servers
  apt:
    name:
    - fail2ban
    - htop
    - vim
    - man
    - manpages-fr
    - manpages-fr-extra
    - manpages-fr-dev
    - bash-completion
    - locales
    - python-pip
    - wget
    - zip
    - bzip2
    - lsof
    - sudo
    - tree
    - ccze
    - vim-common
    - screen
    - curl
    - git
    - unzip

# On supprime les packages non nécessaires
- name: Remove useless stuff
  apt:
    name:
      - bind9
      - telnet
      - ftp
    state: absent

# On copie notre fichier pour fail2ban
- name: Upload fail2ban config file
  template:
    src: fail2ban.conf.j2
    dest: /etc/fail2ban/jail.d/defaults-debian.conf
    mode: 0644
```

Bien sûr, il s'agit d'une version très light de ce que vous pouvez faire, mais c'est pour vous montrer ce que ça peut donner. Ce rôle sera donc utilisé sur

l'installation de tous nos serveurs. Libre à vous de vous servir de ce rôle pour déployer votre .bashrc personnalisé, installer les applications que vous utilisez au quotidien, définir des règles de sécurité, paramétrer un firewall… la seule limite est votre imagination.

Si vous avez cloné le dépôt précédent, vous devriez vous retrouver avec l'arborescence suivante :

```
.
├── lampSetup.yml
├── LICENSE
├── README.md
└── roles
    ├── apache
    │   ├── files
    │   │   └── phpinfo.php
    │   ├── handlers
    │   │   └── main.yml
    │   ├── README.md
    │   ├── tasks
    │   │   └── main.yml
    │   └── templates
    │       └── tomzone.fr.conf.j2
    ├── common
    │   ├── README.md
    │   ├── tasks
    │   │   └── main.yml
    │   └── templates
    │       └── fail2ban.conf.j2
    ├── lampserver
    │   ├── meta
    │   │   └── main.yml
    │   └── tasks
    │       └── main.yml
    └── mariadb
        ├── defaults
        │   └── main.yml
        ├── handlers
        │   └── main.yml
        ├── README.md
        └── tasks
            └── main.yml
```

Comme vous pouvez le remarquer, nous allons créer un rôle qui s'appellera lampserver. Celui-ci se chargera d'installer Apache2, PHP, MariaDB. Et cerise sur le gâteau, eux-mêmes sont des rôles. On peut donc imbriquer des rôles dans des rôles ! On appellerait ceci l'*inception* si Christopher Nolan ne m'avait pas déjà volé l'idée ;-)

Nous allons créer ensuite notre fichier meta dans le rôle lampserver et renseigner les différentes dépendances de ce rôle :

```
# lampserver/meta/main.yml
dependencies:
  - { role: mariadb }
  - { role: apache }
```

Tout simplement…

Enfin, nous allons créer le playbook principal : lampSetup.yml. Puis, nous allons appeler les différents rôles à l'intérieur de ce playbook pour instancier notre serveur web :

```
—
# On définit un hôte ou un groupe d'hôtes
où sera joué notre playbook
- hosts: webserver01
  roles:
     - common
     - lampserver

  # Initialisation d'une variable globale,
elle sera utilisée notamment dans le rôle
'common'
  vars:
    email: tom@mail.fr
```

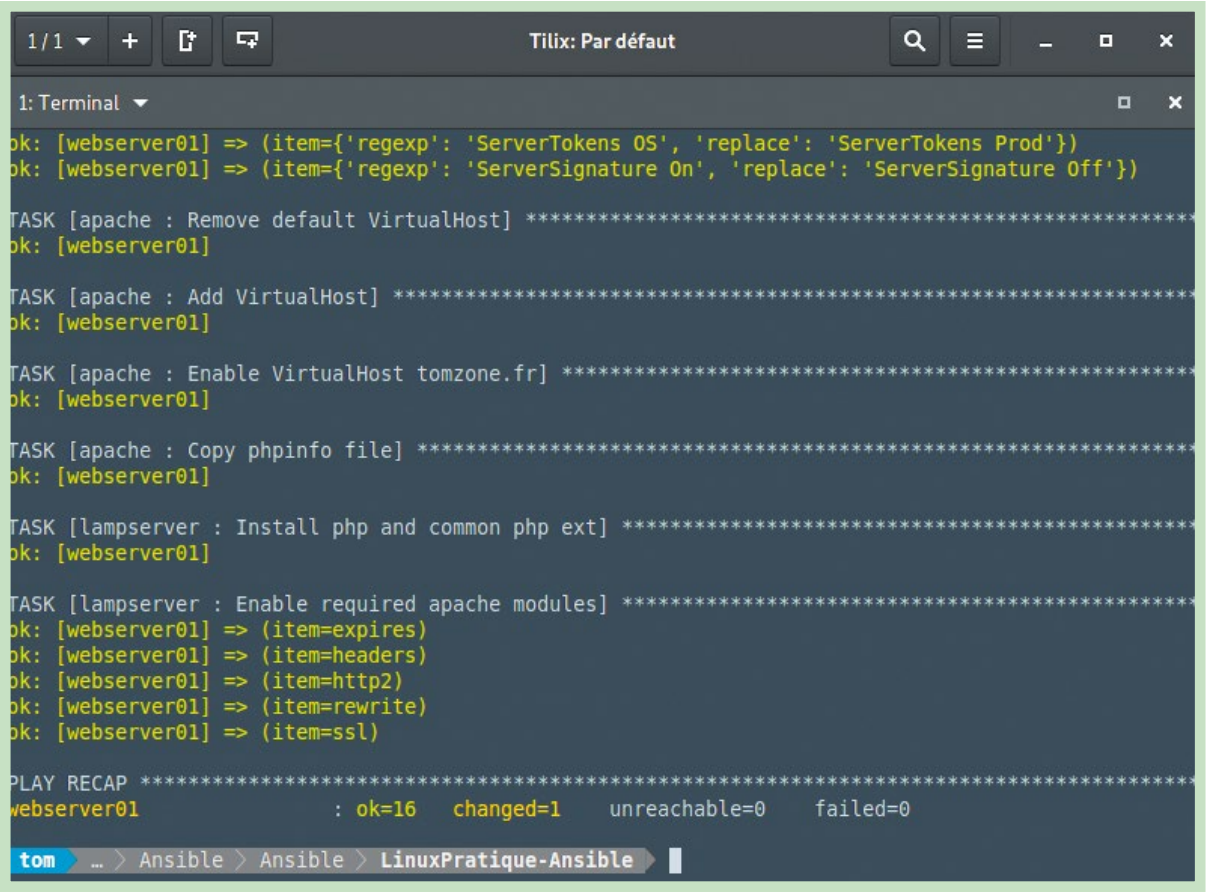


FIGURE 3

```
  # ici, on demande directement à
l'exécution de renseigner une variable
  vars_prompt:
    - name: "mysqlRootPassword"
      prompt: "Please enter your MySQL
root password"
```

Je le disais précédemment, nous avons mis en place un dépôt GitHub pour faciliter les choses, car je ne pourrai pas, dans cet article, détailler la totalité des playbooks, sinon je prendrai trop de pages. Mais je pense que maintenant vous commencez à comprendre la logique. Écrire des tâches et pouvoir les réutiliser à l'infini dans vos différents playbooks. Un gain de temps pour tout administrateur qui se respecte.

Je pense que nous avons fait le tour des explications sur les différents rôles et la façon de fonctionner d'Ansible. Pour installer le serveur, il suffit de taper ensuite dans un terminal :

```
ansible-playbook lampSetup.yml
```

Et vous pouvez admirer le résultat.

CONCLUSION

J'espère maintenant que les principes fondamentaux d'Ansible sont plus clairs. Ansible regorge de modules et est sans cesse en évolution. Ce fabuleux outil vous permettra d'automatiser tout ou quasiment sur vos systèmes d'information. À mon avis, il est indispensable de nos jours et si vous hésitez à franchir le cap, faites comme moi. Donnez-vous un objectif simple, réalisez-le, puis au fur et à mesure agrémentez vos playbooks et vos rôles de nouvelles fonctionnalités. Personnellement, je l'utilise au travail pour instancier des VMs Linux, des clusters Kubernetes. Je ne peux plus m'en passer. ■

Comment obtenir de L'AIDE ?

ANTHONY CARRÉ

RTFM, READ THE FUCKING MANUAL, CONSEIL PEU LOQUACE QUI POURRAIT SE TRADUIRE « POURRIEZ-VOUS AVOIR L'AMABILITÉ DE CHERCHER PRÉALABLEMENT PAR VOUS-MÊME ET ÉVITER AINSI D'IMPORTUNER VOS CAMARADES POUR OBTENIR UNE RÉPONSE QUI NOUS SEMBLE ÊTRE DÉJÀ RENSEIGNÉE DANS LES PAGES DU MANUEL ». QUI FRÉQUENTE FORUMS OU SALONS DE DISCUSSION AURA TRÈS CERTAINEMENT REÇU, AU MOINS UNE FOIS, CETTE LACONIQUE INJONCTION (OU UN DÉRIVÉ DU GENRE « GOOGLE EST TON AMI ») EN RÉPONSE À UNE QUESTION. COMME VOUS ESTIMEZ À JUSTE TITRE QU'IL N'EST PAS CORRECT D'ATTENDRE UNE RÉPONSE D'AUTRUI SANS FAIRE VOUS-MÊME UN EFFORT, VOUS TENTEZ ALORS DE LIRE LEDIT MANUEL. MAIS SA PRISE EN MAIN, COMME SA COMPRÉHENSION, N'EST NI AISÉE NI INTUITIVE. CET ARTICLE PROPOSE DE VOUS GUIDER DANS L'USAGE DE MAN ET PRÉSENTE QUELQUES SOLUTIONS COMPLÉMENTAIRES POUR VOUS AIDER À L'UTILISATION DE COMMANDES DANS VOTRE TERMINAL, MAIS PAS UNIQUEMENT.

1. MAN : PROGRAMME DE VISUALISATION DES PAGES DE MANUEL

man est le programme de visualisation des pages de manuel **[1]**. Il affiche, via un pager, la documentation fournie par la plupart des programmes. La majorité des distributions utilise less comme pager par défaut, c'est donc très probablement avec celui-ci que la commande man vous affichera la page de manuel désirée. Si vous souhaitez afficher le manuel de la commande unzip, vous devrez simplement taper : man unzip.

man est installé par défaut, toutefois, vous pouvez ajouter quelques pages de manuels complémentaires en ajoutant manpages-fr, manpages-fr-dev et manpages-fr-extra.

Si vous êtes débutant, il n'est pas forcément évident de comprendre comment naviguer dans le pager, voici le minimum à connaître : naviguez avec [Haut], [Bas], [Pageup] et [Pagedown], appuyez sur [q] pour quitter. Pour commencer une recherche, tapez [/] ([?] pour une recherche arrière) le motif désiré puis [Entrée]. Tapez [n] pour aller à la prochaine occurrence et [N] pour la précédente. Pour afficher toutes les possibilités offertes par less, nous vous invitons à consulter l'aide du pager ([h] une fois celui-ci lancé).

Par défaut, less n'affiche pas les manuels en couleur, cela ne simplifie pas la lecture, mais il existe plusieurs solutions pour arranger cela. Vous pouvez configurer less pour activer la coloration ou utiliser un autre pager (most est le plus souvent cité comme alternative, même si celui-ci implique d'autres difficultés : pas de redimensionnement automatique, raccourcis clavier non configurables… mais vous pouvez également essayer mcview, vimpager, moar, etc.). Les utilisateurs d'OhMyZSH peuvent activer le plugin colored-man-pages en l'ajoutant dans leur fichier ~/.zshrc, ceux qui utilise le shell fish activeront fish-colored-man.

Notez qu'il existe une commande apropos (équivalent à man -k) qui permet de rechercher un terme dans l'ensemble des manuels. Par exemple, apropos task retournera une liste des pages de manuels contenant le mot « task ». En complément, il existe également l'argument --help que vous pouvez utiliser après quasiment n'importe quelle commande pour obtenir une aide succincte, généralement une description rapide des arguments qui sont à votre disposition (vous pouvez essayer firefox --help ou gimp --help par exemple).

Vous pouvez assez facilement exporter une page de manuel. En HTML avec man2html, en PDF via une simple commande (ici par exemple le manuel de tar) :

```
man -t tar | ps2pdf - "Manuel tar.pdf"
```

Pour plus d'informations, tapez man man dans votre terminal.

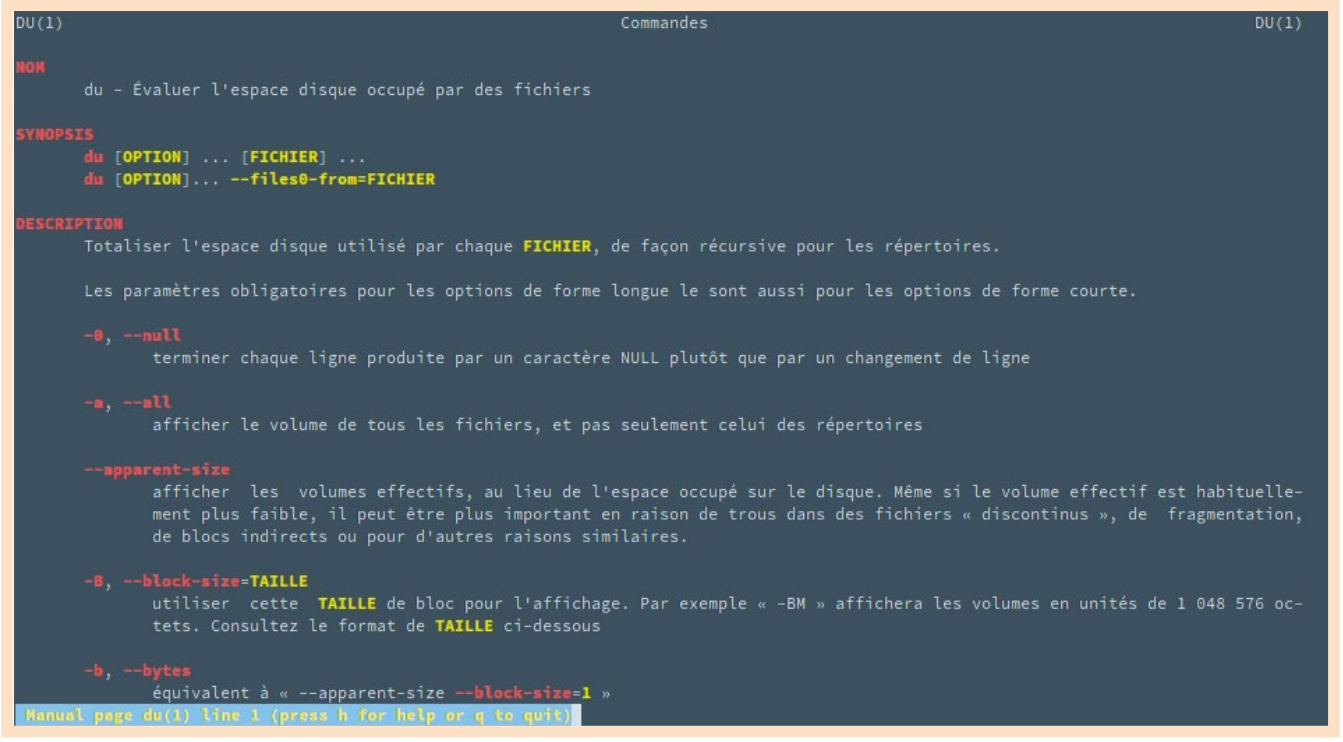


FIGURE 1. Affichage d'une page de manuel via la commande man.

2. TLDR : PAGES DE MANUEL SIMPLIFIÉES, PILOTÉES PAR LA COMMUNAUTÉ

Une page de manuel est rarement aussi passionnante qu'un roman d'Adams, Egan, Pennac ou Thompson. Il existe même un sigle parfaitement adapté à la sensation que l'on peut avoir a priori en voyant la quantité de texte à lire : TL;DR (*Too Long; Didn't Read*, ce qui pourrait se traduire : « Trop long, pas lu »). Parfois, on aimerait, en effet, pouvoir extraire facilement directement l'information pertinente à l'usage d'une commande sans passer par la lecture d'un pavé. TLDR **[2]** est justement là pour proposer cela : quelques exemples caractéristiques d'utilisation en lieu et place de longues et denses explications exhaustives. Un excellent complément au manuel.

La plupart des distributions disposent d'un paquet tldr, mais vous pouvez également l'installer via npm, pip, gem, stack, brew, snap ou autre… les solutions ne manquent pas. Vous trouverez même une app tldroid pour Android et tldr-pages pour iOS ou une version en ligne sur **https://tldr.ostera.io** !

Plusieurs centaines de fiches sont disponibles, couvrant un grand nombre de commandes. Pour ajouter vos propres commandes ou proposer vos traductions (les commandes sont accompagnées de commentaires qu'il peut être intéressant de traduire), la solution la plus simple est probablement d'éditer directement la page depuis l'interface web de GitHub **[3]**. N'hésitez pas à proposer vos meilleurs aide-mémoires.

Pour plus d'informations, tapez tldr tldr dans votre terminal.

```
$ tldr pdftk
pdftk
PDF toolkit.

- Extract pages 1-3, 5 and 6-10 from a PDF file and save them as another one:
  pdftk {{input.pdf}} cat {{1-3 5 6-10}} output {{output.pdf}}

- Merge (concatenate) a list of PDF files and save the result as another one:
  pdftk {{file1.pdf}} {{file2.pdf}} ... cat output {{output.pdf}}

- Split each page of a PDF file into a separate file, with a given filename output pattern:
  pdftk {{input.pdf}} burst output {{out_%d.pdf}}

- Rotate all pages by 180 degrees clockwise:
  pdftk {{input.pdf}} cat {{1-endsouth}} output {{output.pdf}}

- Rotate third page by 90 degrees clockwise and leave others unchanged:
  pdftk {{input.pdf}} cat {{1-2 3east 4-end}} output {{output.pdf}}
```

FIGURE 2. Peut-être que la lecture de tldr pdftk suffira pour votre usage et vous évitera alors la lecture des 436 lignes du man…

3. CHEAT.SH : LA SEULE ANTISÈCHE DONT VOUS AVEZ BESOIN

Sur le même principe que TLDR, Cheat.sh (abrégeable en cht.sh) propose des fiches présentant des exemples d'utilisations. Mais il faut admettre que ce dernier va bien plus loin : pas moins de 1800 entrées, des exemples pour des commandes, mais également de nombreuses classes pour plusieurs langages de programmation, utilisable même sans installation…

Il est possible d'accéder à l'ensemble des fiches de Cheat.sh juste avec curl (pour la fiche de la commande wc : curl cht.sh/wc) ou depuis votre navigateur **[4]**, mais il est bien sûr également possible de l'installer (et éventuellement de profiter de l'auto-complétion. Nous vous invitons à suivre les instructions disponibles). Il est également possible d'intégrer cht.sh à certains éditeurs (vim, emacs, sublime…).

Pour l'installation, nous vous encourageons à suivre les instructions disponibles via curl cht.sh/:intro ou dans la page GitHub du projet. Une fois installée, la commande cht.sh simplifie alors votre saisie.

Nous avons voulu tester la pertinence de l'affirmation que le

FIGURE 3. Le nombre d'entrées disponibles grâce à cheat.sh est impressionnant !

projet est suffisamment complet pour rendre les autres solutions obsolètes, nous n'avons pas été déçus. Quelle autre solution donne accès aussi facilement à des exemples d'utilisation de gnuplot, unrar ou grep que de la classe TransferFunction de la bibliothèque Signal du projet Scipy du langage Python ?

```
cht.sh gnuplot
curl cht.sh/python/signal
curl cheat.sh/python/signal.TransferFunction
```

Il est possible de faire des recherches de termes en les précédant d'un ~, lister les entrées avec :list… Pour plus d'informations, tapez curl cheat.sh dans votre terminal.

Vous pouvez apporter des modifications ou de nouvelles entrées, pour cela, lisez les instructions obtenues via curl cht.sh/:post ou rendez-vous sur le GitHub du projet **[5]**.

4. BROPAGES : OBTENEZ JUSTE L'ESSENTIEL !

Toujours sur le principe des exemples d'utilisation, un peu plus limité que cheat.sh, en particulier parce qu'il ne propose pas de fiches pour les modules de langages de programmation, bropages **[6]** est toutefois un outil à surveiller de près. Il propose des exemples pour un grand nombre de commandes (plus de 700 entrées) et propose une très forte implication de ses utilisateurs, en particulier via un système de vote intégré pour vos exemples préférés.

Après avoir installé ruby-dev, vous pouvez installer bropages via gem install bropages.

Une fois installé, vous affichez les exemples d'utilisation en tapant bro <commande> (bro dd par exemple). Les fiches s'ouvrent avec votre pager (less par défaut). Tapez [h] pour afficher l'aide, [q] pour quitter.

Chaque fois que plusieurs exemples sont disponibles pour une commande, vous avez la possibilité de voter. Le premier vote

FIGURE 4. Exemple d'utilisation de bro : les exemples d'utilisation de mutt.

nécessite de renseigner une adresse courriel valide. Les commandes `bro thanks 2` et `bro ...no 2` permettent respectivement de voter pour améliorer ou dégrader le référencement de la deuxième fiche. Vous pouvez également proposer vos propres exemples via `bro add <commande>`. L'intégration directe de ce système de vote est une bonne idée, même si on pourra regretter un affichage interminablement long, ce qui est contradictoire avec la concision recherchée.

Pour plus d'informations, tapez `bro bro` dans votre terminal (Figure 4).

5. QUELQUES SITES POUR DÉCORTIQUER DES COMMANDES

Il arrive parfois que l'on nous fournisse une commande, mais qu'on ne comprenne pas vraiment celle-ci, parce que sa syntaxe est complexe (bourrée de |, > et autres &) ou parce qu'elle utilise des commandes et arguments qui nous sont inconnus. On préfèrera assurément comprendre avant de copier-coller ce `sudo rm -rf /` trouvé

FIGURE 5. Apprenez les structures de commandes complexes, décortiquez les commandes dont vous ne comprenez pas le fonctionnement avant de faire une bêtise.

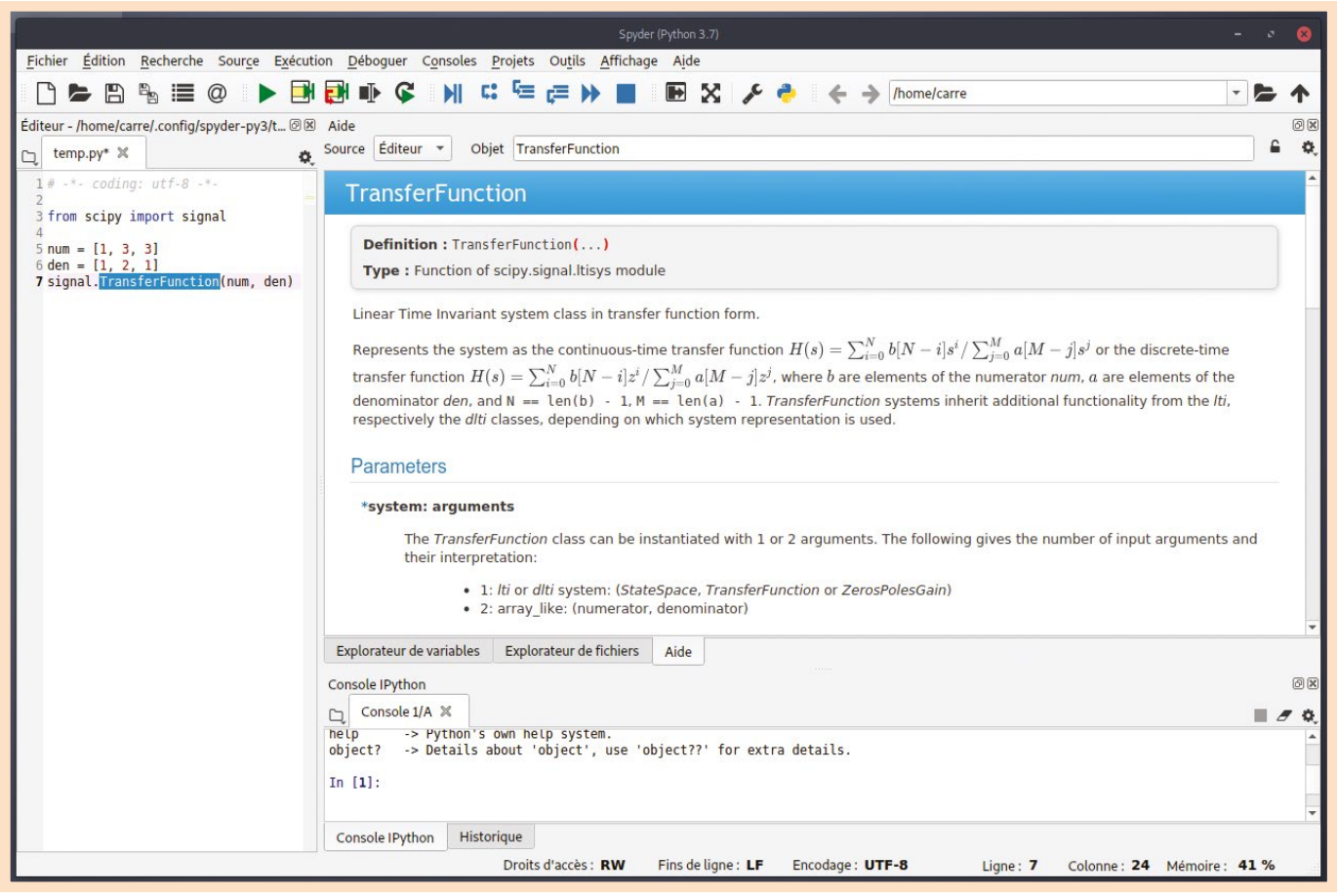


FIGURE 6. L'utilisation d'une fonction n'est pas évidente, sélectionnez-la et affichez son aide d'un simple [Ctrl+i].

sur la Toile et conseillé par un inconnu. Pour cela, il existe différents sites qui proposent de décortiquer vos commandes :

- tapez votre commande sur explainshell **[7]** et la page en détaillera séparément chaque partie ;
- plusieurs sites comme commandlinefu **[8]** ou kommandr **[9]** permettent de trouver de nombreux exemples de commandes, une bonne solution pour découvrir des astuces ;
- s'il y a bien un domaine qui nécessite manuels, aides et explications, c'est bien les expressions régulières aussi connues sous le nom « regex ». Sur regex101 **[10]**, vous pourrez décortiquer ces expressions incroyablement puissantes (mais humainement quasi inintelligibles, non ?).

6. DANS L'APPLICATION

On a parfois tendance à l'oublier, mais la plupart des applications graphiques intègrent directement une aide. Très souvent accessible directement via la touche [F1], elle est souvent également disponible via la dernière entrée du menu principal. N'oubliez pas de consulter cette aide, pour des applications comme Inkscape ou GIMP par exemple, celle-ci simplifie largement la prise en main.

7. PROGRAMMATION

Si vous codez, il vous arrive très certainement d'utiliser des bibliothèques dont vous ignorez a priori la syntaxe… Vous pouvez lire le manuel, mais trouver l'aide

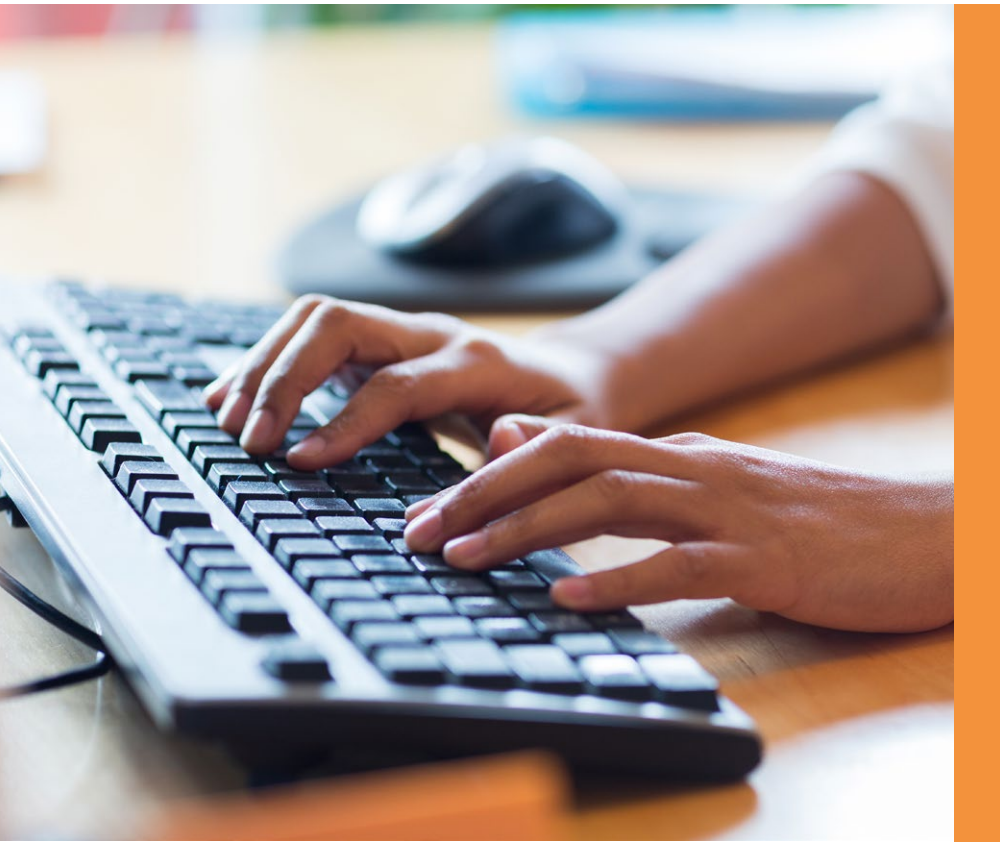
spécifique de la commande n'est pas forcément aisé. Prenons l'exemple de Python.

Il est assez simple d'obtenir l'aide d'une fonction avec Python puisqu'il suffit d'écrire la fonction de votre choix précédée d'un point d'interrogation pour que la documentation s'affiche directement dans votre console. Il existe également une solution plus accessible : dans Spyder **[11]**, si vous tapez (reprenons le même exemple que précédemment) « signal.TransferFunction » (après un `from scipy import signal` bien sûr), que vous sélectionnez ce texte et utilisez le raccourci [Ctrl+i], vous obtenez la page d'aide directement, mise en forme, avec coloration syntaxique, avec des exemples, dans l'onglet d'aide.

D'autres IDE propose une aide intégrée, cela peut se faire sous différentes formes (pop-up au survol par exemple), pensez à utiliser ces aides directes avant d'avoir le réflexe moteur de recherche et/ou forum.

CONCLUSION

Le but de cet article n'est pas de vous empêcher de poser vos questions dans un forum ou un salon, sur Stack Overflow **[12]** ou dans votre moteur de recherche préféré, ces solutions sont évidemment utiles, voire indispensables. Nous avons simplement tenté de mettre en avant des solutions qui sont étrangement passées du statut de premier réflexe à alternative (`man`, menu d'aide) ainsi que des compléments qui nous semblent intéressants de connaître. ■



RÉFÉRENCES

[1] Site officiel man : https://www.kernel.org/doc/man-pages/

[2] Site officiel TLDR : https://tldr.sh/

[3] Code TLDR : https://github.com/tldr-pages/tldr

[4] Site officiel Cheat.sh : http://cht.sh/

[5] Code Cheat.sh : https://github.com/chubin/cheat.sh

[6] Site officiel bropages : http://bropages.org/

[7] https://www.explainshell.com

[8] https://www.commandlinefu.com/

[9] http://kommandr.com/

[10] https://regex101.com/

[11] https://www.spyder-ide.org/

[12] https://stackoverflow.com/

ACTUELLEMENT DISPONIBLE
HACKABLE N°29 !

NOUVEAU ! ACHETEZ-LE DÈS MAINTENANT SUR IOS ET ANDROID :

Disponible en App Store

DISPONIBLE SUR Google Play

CRÉEZ UNE SONDE DE TEMPÉRATURE AUTONOME !

NE LE MANQUEZ PAS

CHEZ VOTRE MARCHAND DE JOURNAUX ET SUR :

https://www.ed-diamond.com

ISOLEZ VOS PROCESSUS
grâce à AppArmor

GAPZ

APPARMOR EST UN PROJET QUI PERMET D'AUGMENTER LA SÉCURITÉ D'UN SYSTÈME GNU/LINUX GRÂCE À DES MÉCANISMES DE RESTRICTION DE DROITS QUE L'ON APPLIQUE DIRECTEMENT AUX PROGRAMMES. QU'IL S'AGISSE DE PROTÉGER UN SERVICE RÉSEAU OU UN SIMPLE CLIENT DE MESSAGERIE, APPARMOR EST UNE SOLUTION SIMPLE ET ACCESSIBLE. CET ARTICLE EST UNE INTRODUCTION À LA MISE EN ŒUVRE DE CET OUTIL.

INTRODUCTION

La sécurité de base d'un système GNU/Linux repose sur de nombreux mécanismes historiques, notamment sur ce que l'on nomme le contrôle d'accès discrétionnaire, ou DAC (pour *Discretionary Access Control*). C'est ce modèle qui est utilisé pour contrôler l'accès aux fichiers (et rappelez-vous, sous Unix tout est fichier, il s'agit donc d'un mécanisme important), avec comme concept central que les permissions seront définies par rapport au propriétaire d'un fichier, ou au groupe auquel il appartient. Pour rappel :

```
$ ls -l /etc/shadow /home/gapz/.bashrc
-rw-r----- 1 root shadow 1220 Jan 15 14:00 /etc/shadow
-rw-r--r-- 1 gapz gapz   3526 Dec 29 21:47 /home/gapz/.bashrc
```

L'avantage de ce modèle est qu'il est très facile à configurer ; il suffit en effet de définir des accès en lecture/écriture/exécution (le fameux « rwx »), pour le propriétaire, le groupe et pour les autres utilisateurs qui ne rentrent pas dans les deux premières catégories. Cependant, on est vite confronté aux limites des possibilités offertes par une telle approche si l'on souhaite faire du cas par cas : par exemple, définir précisément à quels fichiers certains processus ont accès. Certes, il est possible dans

une certaine mesure de créer des groupes et des utilisateurs spécifiques, mais la mise en place devient vite complexe et difficilement maintenable.

C'est entre autres sur ce constat qu'a été développé AppArmor (mais aussi sur le fait que les solutions existantes, comme SELinux, étaient beaucoup trop complexes à mettre en œuvre). Cet article va présenter, à grand renfort d'exemples, comment utiliser AppArmor, que l'on soit un simple utilisateur ou un administrateur système. Certains éléments nécessiteront parfois une connaissance avancée du fonctionnement d'un système GNU/Linux, mais dans la plupart des cas un niveau d'abstraction permettra malgré tout d'utiliser AppArmor sans difficulté.

1. QU'EST-CE QU'APPARMOR ?

Pour rester simple, AppArmor va permettre d'autoriser un programme à effectuer ce qu'il a besoin de faire, et d'interdire tout le reste. Il a donc besoin de savoir ce que le programme a le droit d'effectuer, et utilisera pour cela un profil (fichier texte indiquant les permissions d'un programme, que l'on va découvrir plus loin dans l'article). Cette approche repose sur un modèle dit de contrôle d'accès obligatoire ou MAC (pour *Mandatory Access Control*). On va donc disposer d'un mécanisme permettant d'effectuer un contrôle fin directement au niveau des processus.

En substance, AppArmor va permettre d'effectuer un contrôle sur les éléments suivants :

- accès aux fichiers ;
- accès réseau ;
- *capabilities* (mécanisme décrit plus loin dans l'article).

D'un point de vue sécurité, il s'agit donc d'effectuer une minimisation de la surface d'attaque en restreignant les actions possibles des programmes lancés sur un système.

Ce projet connaît un certain succès les dernières années, avec comme fait marquant son intégration par défaut dans la dernière version de Debian (*buster*, encore en *testing* à l'heure où ces lignes sont écrites).

1.1 Prérequis : installation

Avant de rentrer dans le vif du sujet, il est nécessaire d'installer quelques outils pour pouvoir se servir d'AppArmor. Tout d'abord, le contrôle qu'effectue AppArmor se situe au niveau du noyau ; il se présente donc sous la forme d'un module noyau Linux (en l'occurrence, un LSM, *Linux Security Modules*). Le plus simple est d'utiliser une distribution qui fournit AppArmor dans l'installation par défaut, par exemple Ubuntu ou Debian.

Pour vérifier qu'AppArmor est bien activé, vous pouvez lancer la commande `apparmor_status`. Si une erreur se produit, vérifiez qu'AppArmor est bien activé au lancement du système, typiquement via la présence des options `apparmor=1 security=apparmor` dans grub :

```
# cat /etc/default/grub.d/apparmor.cfg
GRUB _ CMDLINE _ LINUX _ DEFAULT="$GRUB _ CMDLINE _ LINUX _ DEFAULT
apparmor=1 security=apparmor"
```

Si les options ne sont pas présentes, rajoutez-les, effectuez un `update-grub` et redémarrez. Concernant les outils en ligne de commandes : si vous utilisez Debian ou Ubuntu, vous devez installer les paquets `apparmor-utils` et `apparmor`.

1.2 Fonctionnement d'AppArmor

1.2.1 Les différents modes : enforce et complain

AppArmor applique un profil permettant d'autoriser ou interdire un ensemble d'éléments, comme l'accès aux fichiers, ou encore au réseau. Pour cela, il fonctionne en deux modes différents : *enforce*, c'est-à-dire qu'il va appliquer strictement les règles, autoriser ou refuser l'accès à un fichier par exemple. Le second mode se nomme *complain* et va permettre une simple notification lorsqu'une règle du profil ne sera pas respectée. Cela va permettre notamment d'analyser un programme et produire un ensemble de règles pour écrire un profil ou de surveiller certains accès d'une manière générale.

1.2.2 Les utilitaires en ligne de commandes

AppArmor dispose d'un ensemble d'outils en ligne de commandes avec chacun sa spécificité : lister les profils, en charger, changer de mode, etc. La liste est assez longue et peut être déroutante de prime abord :

```
# aa-<tab>
aa-audit        aa-decode       aa-exec         aa-remove-unknown
aa-autodep      aa-disable      aa-genprof      aa-status
aa-cleanprof    aa-enabled      aa-logprof      aa-unconfined
aa-complain     aa-enforce      aa-mergeprof    aa-update-browser
```

Le premier contact que l'on a généralement avec AppArmor va être d'effectuer aa-status ou apparmor_status :

```
# aa-status
apparmor module is loaded.
16 profiles are loaded.
13 profiles are in enforce mode.
   /root/lala.sh
   /usr/bin/irssi
   /usr/sbin/apt-cacher-ng
   /usr/sbin/haveged
   /usr/sbin/ntpd
   /usr/sbin/tcpdump
   gst _ plugin _ scanner
2 processes have profiles defined.
3 processes are in enforce mode.
   /usr/sbin/haveged (309)
   /usr/sbin/ntpd (983)
   /usr/bin/irssi (18059)
0 processes are in complain mode.
0 processes are unconfined but have a profile defined.
```

Cette commande nous affiche l'ensemble des profils chargés, le mode dans lequel ils sont, ainsi que l'ensemble des processus auxquels ces profils sont appliqués avec le mode correspondant. Dans un second temps, on souhaitera manipuler les modes (on prend ici comme exemple le profil pour le programme irssi, un simple client IRC) :

```
# aa-complain /usr/bin/irssi
Setting /usr/bin/irssi to complain mode.
```

On pourra vérifier avec aa-status de nouveau comme le mode a bien changé :

```
# aa-status
[...]
1 profiles are in complain mode.
   /usr/bin/irssi
[...]
1 processes are in complain mode.
   /usr/bin/irssi (18059)
[...]
```

De la même manière, on pourra utiliser aa-enforce pour passer en mode *enforce*. Il est également possible de recharger un profil après l'avoir modifié :

```
# apparmor_parser -r /usr/bin/irssi
```

Il peut aussi être intéressant d'avoir les informations sur le statut des programmes directement via la sortie de la commande ps :

```
$ ps auxZ | grep irssi
/usr/bin/irssi (complain)       root     18059  0.0  0.8
49460  8668 pts/0    S+   23:58   0:00 irssi
unconfined                      root     18094  0.0  0.0
12780   932 pts/1    S+   23:59   0:00 grep irssi
```

On verra plus loin que d'autres outils sont essentiels à AppArmor, notamment pour la création de profils, comme aa-genprof ou aa-autodep.

1.2.3 Structure et contenu des profils

Jusqu'à présent on a joué avec les modes, les profils, mais nous ne sommes pas rentrés dans les détails des règles que l'on est capable d'utiliser. L'objet de cette section est de décrire rapidement les possibilités offertes par les profils AppArmor. De nombreuses options sont disponibles et l'on propose aux lecteurs intéressés d'aller lire la documentation officielle **[2]** ou celle d'openSUSE **[0]**, qui maintient une liste exhaustive de ce que l'on peut faire avec les profils.

Certaines fonctionnalités sont évidentes, tandis que d'autres nécessitent une compréhension plus profonde du fonctionnement du système. Dans tous les cas, il est d'usage de se baser sur le résultat produit par les outils pour vous assister à la création de profil (que l'on va découvrir dans la partie suivante).

Il y a typiquement 3 parties que l'on va manipuler plus souvent que les autres :

- les permissions sur les fichiers ;
- les permissions sur le réseau ;
- les *capabilities*.

Pour cette dernière partie, les *capabilities* sont un mécanisme qui permet de segmenter sous forme de « fonctionnalité » ce que peut faire un programme (cela permet d'avoir un contrôle plus fin que l'approche traditionnelle processus privilégié/non privilégié). Pour avoir une liste exhaustive des *capabilities,* vous pouvez lire man 7 capabilities. Certaines sont facilement compréhensibles, comme CAP_CHOWN (permettre de modifier le propriétaire d'un fichier), tandis que d'autres sont autrement plus complexes, telles que CAP_SYS_ADMIN.

Concernant le réseau, il va s'agir essentiellement d'autoriser ou non l'utilisation des différents types et protocoles usuels (stream/raw/packet et ip/icmp/tcp/udp).

Pour ce qui est des fichiers, l'approche est d'appliquer les permissions qui seront indiquées, fichier par fichier.

Pour chaque règle, qu'il s'agisse de réseau, de fichier ou de capability, on va effectuer un deny (qui sera forcément indiqué) ou un allow (qui, lui, est implicite).

2. PREMIERS EXEMPLES DE PROFILS

2.1 Hello world et découverte des permissions et modes d'exécution

Voyons donc un premier profil qui va permettre d'effectuer une restriction sur un programme très simple : un script shell. Prenons le code suivant, situé dans /root/script.sh :

```
#!/bin/dash

echo "hello world" > /tmp/test
cat /tmp/test
```

Le script est on ne peut plus simple : il va écrire « hello world » dans le fichier /tmp/test et ensuite afficher son contenu. Admettons que l'on souhaite restreindre au maximum les permissions de ce script, voici ce que donnerait un exemple de profil AppArmor :

```
#include <tunables/global>

/root/script.sh {

  #include <abstractions/base>
  #include <abstractions/bash>
  #include <abstractions/consoles>

  /bin/cat mr,
  /bin/cat rix,
  /bin/dash ix,
  /lib/x86_64-linux-gnu/ld-*.so mr,
  /root/script.sh r,
  /tmp/test rw,
}
```

On constate plusieurs éléments de base d'un profil :

- des directives include, permettant d'inclure d'autres éléments ;
- des permissions sur des fichiers ;
- des permissions d'exécution.

Pour ce qui est des include, pour le tout premier il s'agit de variables pour nous faciliter la vie (répertoire home de l'utilisateur, répertoire /proc...). Concernant les trois include qui se suivent, il s'agit d'abstractions, c'est-à-dire un ensemble de règles communes à de nombreux processus que l'on va rencontrer régulièrement. Dans le cas présent, base/bash/consoles permettent essentiellement d'accéder à des fichiers usuels pour le bon fonctionnement de notre script.

Pour ce qui est des permissions sur les fichiers, elles sont on ne peut plus évidentes :

- r : lecture ;
- w : écriture, ne peut pas être utilisé avec a ;
- a : ajout, ne peut pas être utilisé avec w ;
- k : « locking » mode ;
- l : lien.

Pour ce qui est des modes d'exécution, l'approche est moins triviale, on distingue entre autres :

- Px : l'exécution du programme appelé se fait avec un autre profil ;
- Cx : l'exécution du programme appelé se fait avec un profil local (ie. dans le profil du programme courant) ;
- Ux : l'exécution du programme appelé se fait sans profil (*unconfined*) ;
- ix : l'exécution du programme appelé se fait avec le profil courant ;
- m : permet de mapper le fichier en mémoire en utilisant mmap avec PROT_EXEC (on verra souvent ce flag pour les bibliothèques partagées par exemple).

Dans l'exemple présent, on permet à cat et à dash de s'exécuter avec le profil courant. Pour une description exhaustive des permissions et mode d'exécution, on ne saura trop recommander la lecture de la documentation d'openSUSE à ce sujet **[2]**.

2.2 À chaque usage son profil : exemple avec ping

Un point qu'il est important de comprendre quand on utilise AppArmor est qu'il faut adapter les profils à vos besoins, à l'usage attendu du programme que vous souhaitez confiner. En d'autres termes, les profils « génériques » vont certainement permettre plus de choses que vous ne le désirez. Par exemple, l'outil ping est généralement setuid sur la plupart des systèmes, et l'on souhaite également pouvoir faire de l'IPv4 et de l'IPv6. Sur une configuration minimaliste, on pourrait imaginer ne faire que de l'IPv4 et avoir uniquement l'utilisateur root ; notre profil pour ping pourrait se limiter à :

```
#include <tunables/global>

/bin/ping {
  #include <abstractions/base>

  capability net _ raw,

  /bin/ping mr,
  /lib/x86 _ 64-linux-gnu/ld-*.so mr,
}
```

Voici, à titre comparatif, le profil officiel de ping (celui maintenu par Canonical) :

```
#include <tunables/global>
profile ping /{usr/,}bin/{,iputils-}ping
flags=(complain) {
  #include <abstractions/base>
  #include <abstractions/consoles>
  #include <abstractions/nameservice>

  capability net _ raw,
  capability setuid,
  network inet raw,
  network inet6 raw,

  /{,usr/}bin/{,iputils-}ping mixr,
  /etc/modules.conf r,

  # Site-specific additions and overrides.
See local/README for details.
  #include <local/bin.ping>
}
```

On note bien l'utilisation d'inet6 et la capability setuid. Bref, même si des profils existent pour vos programmes, il n'est jamais certain qu'ils répondent précisément à vos besoins.

2.3 Vue d'ensemble à l'échelle du système

Les profils (avec les abstractions et autres déclarations de variables) sont fournis dans Debian ou Ubuntu dans le répertoire /etc/apparmor.d/. Ce sont ces profils qui sont chargés par AppArmor par défaut. Les profils ont comme convention de nommage le chemin vers l'application séparée par des points et se terminent par le nom de celle-ci. Typiquement, pour /usr/bin/irssi, le profil se nomme usr.bin.irssi. Les outils en ligne de commandes, notamment pour générer des profils, respectent cette convention.

3. QUELQUES EXEMPLES PRATIQUES

3.1 Création d'un profil pour /bin/date et utilisation de « capability »

Lorsque l'on veut créer un profil pour une application, il existe deux méthodes, qui impliquent dans les deux cas l'usage d'un outil dédié :

- le profilage *stand-alone*, pour effectuer une analyse application par application ;
- le profilage *systemic*, à l'échelle du système entier (généralement pour analyser les différents démons lancés sur l'ensemble du système).

On s'intéresse ici uniquement à la première méthode, qui utilise aa-genprof. La procédure est simple : dans un premier terminal, lancez aa-genprof avec comme option le chemin vers le programme à profiler :

```
# aa-genprof /bin/date
[...]
[(S)can system log for AppArmor events] / (F)inish
```

Dans un second terminal, effectuez des exécutions de l'application que vous souhaitez profiler, en essayant de couvrir au maximum les usages que vous allez en faire :

```
$ date
$ date -s ''
$ date --date=@1
$ date +%s
```

Une fois les différentes commandes lancées et terminées, retournez dans votre terminal où vous avez lancé aa-genprof et appuyez sur [S] afin de lancer l'analyse des logs pour générer les règles de notre futur profil :

```
Reading log entries from /var/log/syslog.
Updating AppArmor profiles in /etc/apparmor.d.
Complain-mode changes:

Profile:    /bin/date
Capability: sys _ time
Severity:   8

 [1 - capability sys _ time,]
(A)llow / [(D)eny] / (I)gnore / Audi(t) / Abo(r)t /
(F)inish
```

Dans le cas présent, on nous propose plusieurs possibilités pour la capability sys_time (qui permet de configurer l'heure du système). On choisit d'autoriser cette dernière ; voici le profil résultant :

```
#include <tunables/global>

/bin/date {
  #include <abstractions/base>

  capability sys _ time,

  /bin/date mr,
  /lib/x86 _ 64-linux-gnu/ld-*.so mr,
}
```

Si l'on veut empêcher la possibilité de changer la date, il suffit d'interdire la capability sys_time via :

```
deny capability sys _ time,
```

On recharge le profil :

```
# apparmor _ parser -r /etc/apparmor.d/bin.date
```

Et l'on tente de changer la date :

```
# date -s ''
date: cannot set date: Operation not permitted
Mon Mar 25 00:00:00 UTC 2019
```

Si l'on souhaite avoir une trace de l'erreur dans les logs, rien de plus simple : il suffit de préfixer le mot clef audit avant la ligne concernée dans le profil :

```
audit deny capability sys _ time,
```

Ce qui aura pour résultat de produire la ligne suivante dans /var/log/message lorsque le programme date se fera refuser la possibilité de modifier la date du système :

```
[358254.537202] audit: type=1400 audit(1553531556.694:132):
apparmor="DENIED" operation="capable" profile="/bin/date"
pid=10502 comm="date" capability=25  capname="sys _ time"
```

3.2 Exemple concret : protection d'un WordPress

Pour les lecteurs qui souhaitent mettre en place un exemple avancé, Docker labs propose un exercice dont le but est de protéger un WordPress lancé dans un conteneur Docker **[1]**. AppArmor s'intègre parfaitement à Docker (après tout, un conteneur n'est qu'un processus), et la manipulation des profils se fait via des options à passer au client docker où à indiquer dans le fichier yaml de docker-compose.

Dans l'exercice, l'idée est de durcir la sécurité d'un WordPress par la prévention de l'ajout de plugin (ou la modification d'un plugin existant) en refusant l'accès en écriture dans le répertoire correspondant. En substance, une simple ligne dans un profil AppArmor :

```
deny /var/www/html/
wp-content/plugins/** wlx,
```

CONCLUSION

Même si un ensemble de profils AppArmor est mis à disposition sur un système, il est bien souvent nécessaire d'effectuer quelques modifications afin que les règles correspondent bien à l'usage attendu, le tout dans l'optique de restreindre au maximum les programmes ciblés et diminuer ainsi la surface d'attaque globale du système. ■

RÉFÉRENCES

[0] https://doc.opensuse.org/documentation/leap/security/html/book.security/cha.apparmor.profiles.htm

[1] https://github.com/docker/labs/tree/master/security/apparmor

[2] https://gitlab.com/apparmor/apparmor/wikis/home/

ACTUELLEMENT DISPONIBLE

MISC N°103

NOUVEAU ! ACHETEZ-LE DÈS MAINTENANT SUR IOS ET ANDROID :

Disponible en App Store

DISPONIBLE SUR Google Play

PENTEST WINDOWS

OUTILS & TECHNIQUES

NE LE MANQUEZ PAS

CHEZ VOTRE MARCHAND DE JOURNAUX ET SUR :

https://www.ed-diamond.com

AUTO-HÉBERGER SES SERVICES EN LIGNE
sur Sandstorm

BENOÎT BENEDETTI

VOUS REPRENDREZ BIEN UN PETIT PEU DE GAFA ? POUR VOS E-MAILS, VOS DOCUMENTS, VOS RÉSEAUX SOCIAUX ? GOOGLE, APPLE, FACEBOOK ET AMAZON, POUR EN NOMMER QUELQUES-UNS, GAFA COMME ON LES SURNOMME. CES SERVICES WEB FERMÉS, CENTRALISÉS ET PROPRIÉTAIRES FONT PARTIE DE NOTRE QUOTIDIEN. CELA FAIT DES ANNÉES QU'ON LE SAIT, ET QUE DES LIBRISTES PROPOSENT DES ALTERNATIVES AUTO-HÉBERGEABLES POUR LUTTER CONTRE CETTE HÉGÉMONIE. MAIS SI VOUS VOULEZ AUTO-HÉBERGER L'ÉQUIVALENT DE TOUTES CES APPLICATIONS, IL VA FALLOIR MAINTENIR BEAUCOUP DE SERVICES WEB DIFFÉRENTS. CE QUI DEMANDE BEAUCOUP DE TEMPS, VOIRE DE COMPÉTENCES TECHNIQUES. SI VOUS N'AVEZ NI L'UN NI L'AUTRE, FRAMASOFT EST DÉJÀ UNE TRÈS BONNE ALTERNATIVE HÉBERGÉE POUR QUITTER LE GIRON DES GAFA. SI VOUS POSSÉDEZ JUSTE UN MINIMUM DES DEUX, JE VOUS PROPOSE DE DÉCOUVRIR DANS CET ARTICLE STANDSTORM, POUR HÉBERGER ET GÉRER VOUS-MÊME VOTRE VIE EN LIGNE.

1. INTRODUCTION

Sandstorm vous permet d'auto-héberger votre propre suite d'applications web en ligne. Chaque application est une version packagée, suivant les spécifications Sandstorm, d'une application connue, que vous pourrez activer ou désactiver dans votre Sandstorm. Vous et vos utilisateurs pourrez ensuite lancer une instance de ces applications simplement à l'aide d'un clic dans l'interface

de Sandstorm : créer votre propre chat (à l'aide de Rocket.Chat), héberger et partager vos fichiers (Davros), gérer vos projets (Wekan), et rédiger des documents de manière collaborative (Etherpad). Et bien plus encore en installant des applications disponibles depuis un App Market **[1]**. Plusieurs dizaines d'applications sont déjà disponibles, et de nouvelles applications sont portées sur Sandstorm et ajoutées à l'App Market régulièrement.

Sandstorm présente donc de nombreux avantages :

- gain en temps et en administration : vous installez juste Sandstorm sur votre serveur. L'installation et la gestion des applications se font facilement via l'interface d'administration ;
- authentification unifiée : vous vous authentifiez une fois avec votre compte Sandstorm, qui vous donne accès automatiquement à vos instances d'application ;
- sécurité entre les instances : par défaut, les applications sont cloisonnées les unes aux autres ; aussi bien vos applications entre elles, que les applications entre utilisateurs ;
- contrôle d'accès flexibles : vous pouvez bien sûr donner des permissions aux applications.

Le cloisonnement est rendu possible grâce au mode de fonctionnement **[2]**. Là où une infrastructure traditionnelle isole les applications entre elles, Sandstorm pousse l'isolation au niveau des données : à l'aide des mêmes fonctionnalités du noyau Linux utilisées par des technologies de conteneurs comme Docker ou LXC, Sandstorm va isoler chacune de vos chatrooms, de vos documents Etherpad ou de vos dépôts GitWeb. Ce qui fait que chacun de ces documents, données, etc. appelés *grain* dans la terminologie Sandstorm, est isolé, sécurisé par défaut. Et peut potentiellement voir ses permissions finement données individuellement.

Pour fonctionner de cette manière isolée, chaque grain a besoin de son sous-domaine DNS : ce qui veut dire que si vous voulez héberger votre Sandstorm à l'adresse **exemple.fr**, vous devrez activer un Wildcard DNS ***.exemple.fr** pour que chaque grain soit accessible **[3]**. Pour vous aider à auto-héberger votre serveur, l'équipe de Sandstorm propose un service de DNS dynamique **[4]** : lorsque vous installez Sanstorm sur votre serveur, il vous est proposé d'utiliser un sous-domaine gratuit **xxx.sandcats.io**. Un Wildcard ***.xxx.sandcats.io** vous sera automatiquement attribué pour le bon fonctionnement de votre serveur. Dans ce cas, celui-ci doit bien sûr être accessible depuis Internet pour utiliser ce service : si vous utilisez un serveur privé sur un réseau local privé, vous devrez gérer vous-même votre Wildcard.

2. INSTALLATION

Il existe un service hébergé officiel, payant, pour utiliser Sandstorm **[5]** sans l'installer. Mais il est plus sécurisé d'auto-héberger ses données, et la procédure d'installation est tellement simple, qu'il serait dommage de s'en passer. Il suffit d'exécuter la commande suivante (si vous n'avez pas confiance dans le script d'installation, la documentation officielle en ligne vous donne les commandes pour récupérer le script et l'exécuter manuellement après l'avoir lu et vérifié **[6]**) :

```
$ curl https://install.sandstorm.io | bash
```

Une série de questions vous sera ensuite posée pour installer Sandstorm dans /opt/sandstorm. Comme par exemple si vous désirez utiliser le service de DNS Sandcats, ou si vous désirez gérer vous-même votre nom de domaine et son wildcard DNS :

```
What *.sandcats.io subdomain would you like? [] none
URL users will enter in browser: [http://debian:6080] http://
exemple.fr
Wildcard host: [*.exemple.fr]
```

Une fois l'installation de Sandstorm sur votre serveur finie, une URL temporaire vous est donnée pour accéder à son interface web et pour finaliser la configuration initiale de votre serveur : authentification des utilisateurs, application(s) activée(s) par défaut (Figure 1, page suivante), et création d'un compte administrateur.

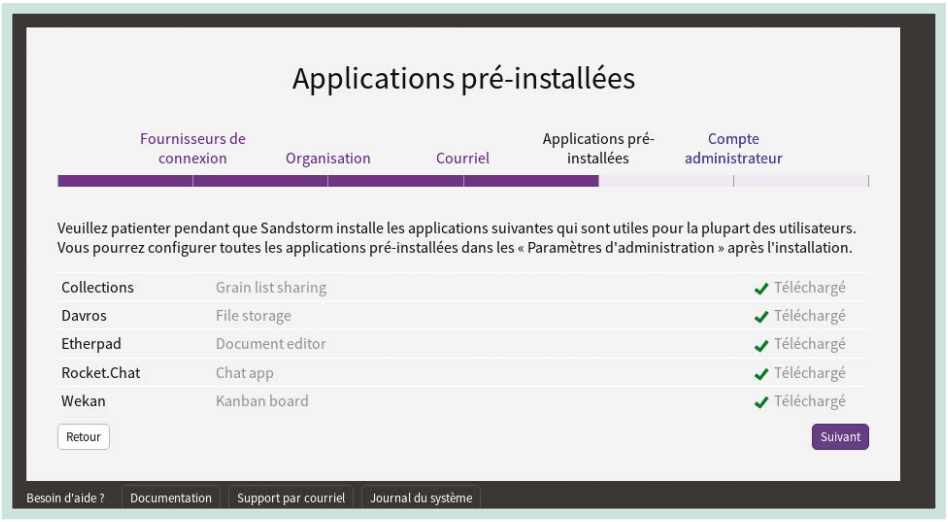


FIGURE 1. Activation des applications par défaut.

3. APPLICATIONS

Vous arrivez sur la page d'accueil de votre instance Sandstorm (Figure 2). L'interface de Sandstorm est minimaliste : un menu latéral gauche vous permet d'afficher la page des grains ou des applications. C'est cette page qui est affichée par défaut lorsque vous vous connectez. Elle liste toutes les applications disponibles sur votre Sandstorm : on retrouve les applications activées lors de l'installation de l'instance. Un clic sur l'icône en forme de gros plus vous redirige vers l'App Market et vous permet d'installer des applications supplémentaires (Figure 3). Il vous suffit de cliquer sur le bouton ***INSTALL*** sous l'application désirée, vous êtes automatiquement redirigé vers votre instance Sandstorm qui s'occupe de télécharger le paquet de l'application et de son installation.

4. GRAINS

Quand vous exécutez une application dans Sandstorm, vous créez un grain de l'application, c'est-à-dire une exécution isolée d'une instance de l'application. Pour exécuter un grain d'une application, cliquez sur l'icône de cette application depuis la page des applications. S'affiche la page particulière à cette application : elle liste tous les grains de cette application que vous avez créés ou que l'on vous a partagés sur cette instance. Dans la figure 4, on peut voir que j'ai déjà un grain Etherpad nommé *LPM*. Pour y accéder rapidement, une entrée dans le menu de gauche est créée pour chaque grain auquel vous avez accès et en cours de fonctionnement : cliquez sur la croix en regard d'un grain dans cette liste pour arrêter l'exécution de ce grain et soulager la charge de votre serveur. Vous pouvez également lister tous vos grains, en fonctionnement ou non, en cliquant le menu général ***Grains*** dans le menu latéral gauche

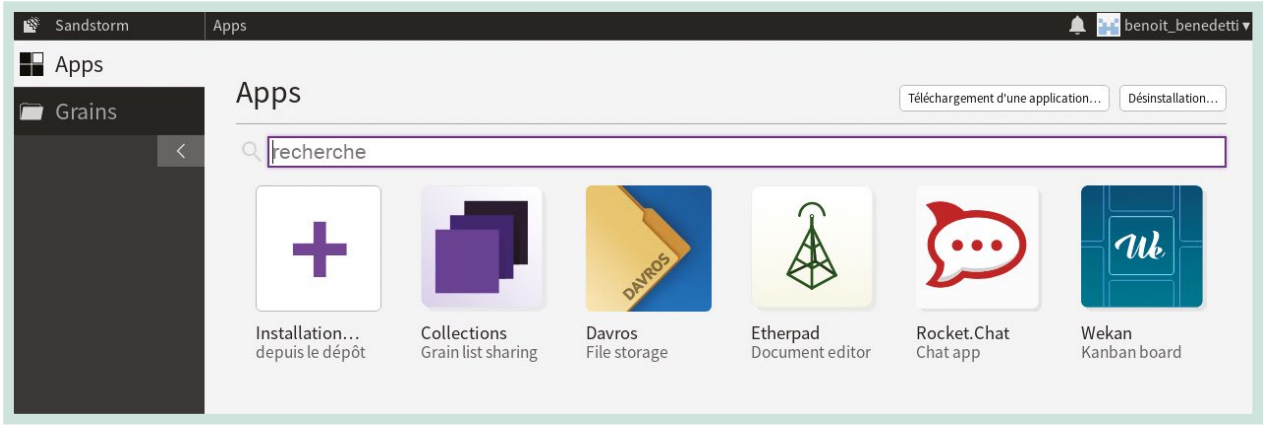


FIGURE 2. Page d'accueil et Apps.

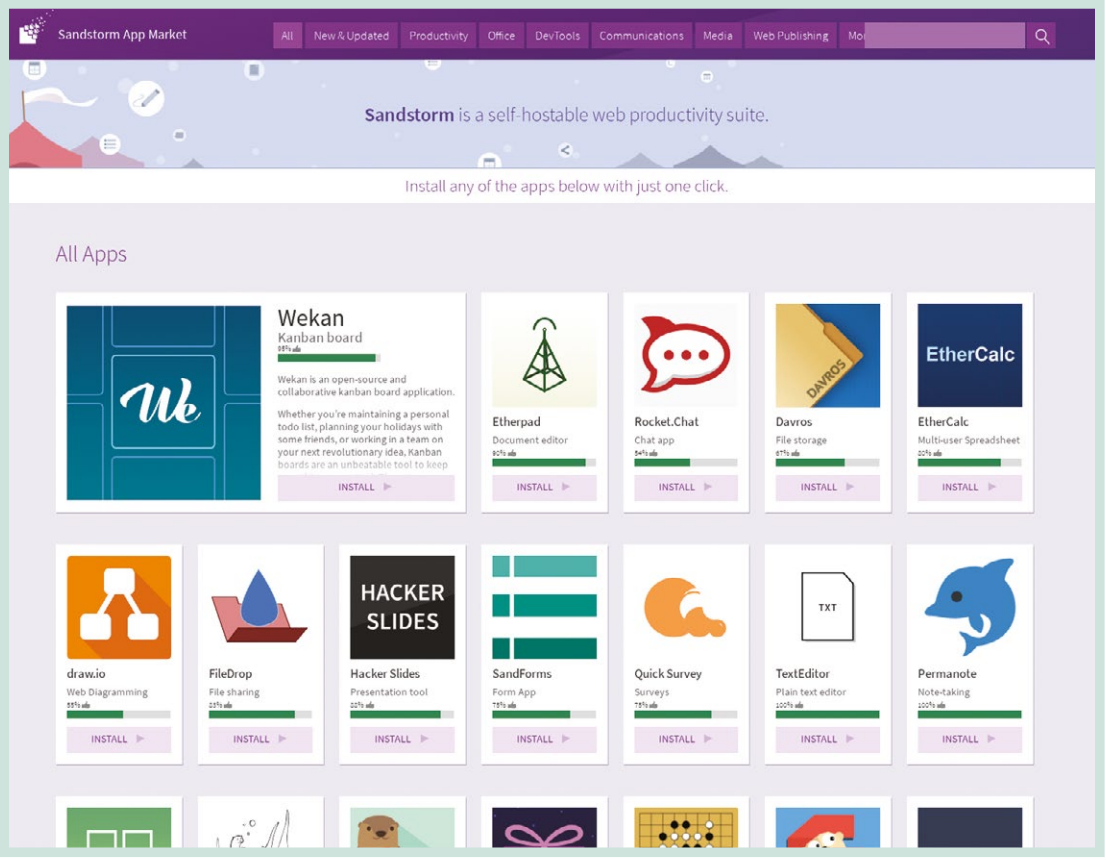


FIGURE 3.
App Market.

(Figure 5, page suivante). Tous ces grains (et donc ces applications) ont été créés et déployés en l'espace de quelques secondes à l'aide de quelques clics !

Pour déployer un grain, cliquez juste sur le plus dans la page d'une application : un nouveau pad Etherpad sera automatiquement créé et ouvert, dans son propre grain.

5. ADMINISTRATION

Connecté en tant qu'administrateur, vous pouvez gérer votre instance Sandstorm, depuis le menu en haut à droite. Vous pourrez configurer certains paramètres

FIGURE 4. Grains Etherpad.

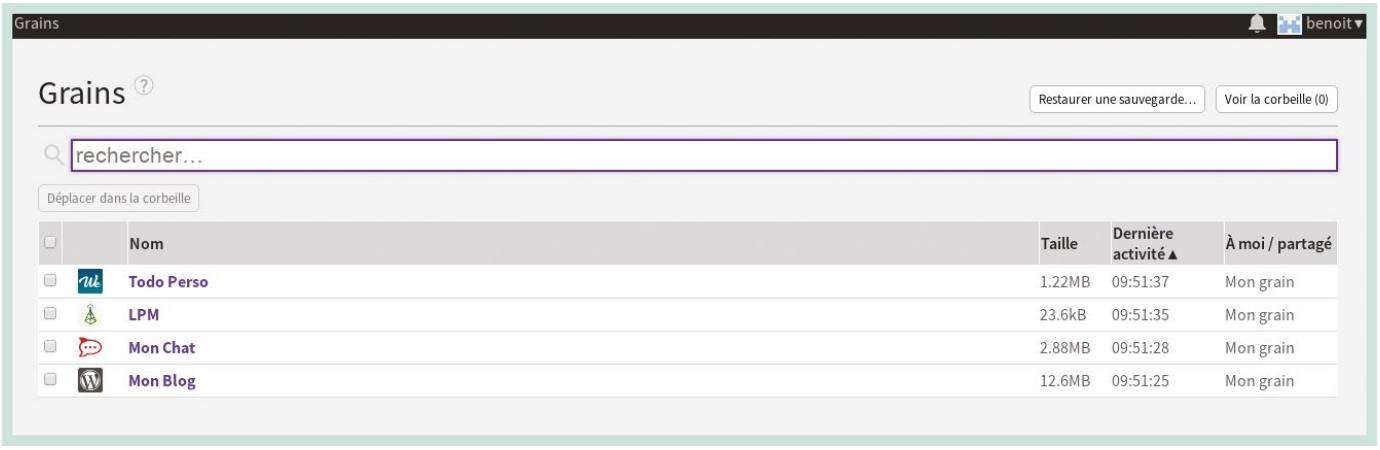


FIGURE 5. Tous mes grains sur cette instance.

déjà vus lors de l'installation : authentification, organisation, e-mail sortant, etc. Vous pourrez également personnaliser l'interface de votre instance, afficher des statistiques d'utilisation, mettre l'instance en maintenance, et définir quelles applications activer par défaut pour les nouveaux utilisateurs. Utilisateurs que vous pourrez également gérer depuis cette page d'administration.

Il existe trois niveaux de compte dans Sandstorm :

- ***Visiteur*** : la personne a un compte et peut se connecter à l'instance, elle ne peut pas exécuter ses propres grains, mais peut utiliser ceux qui lui ont été partagés ;
- ***Utilisateur*** : la personne peut en plus utiliser et exécuter des grains pour son usage, et les partager avec d'autres ;

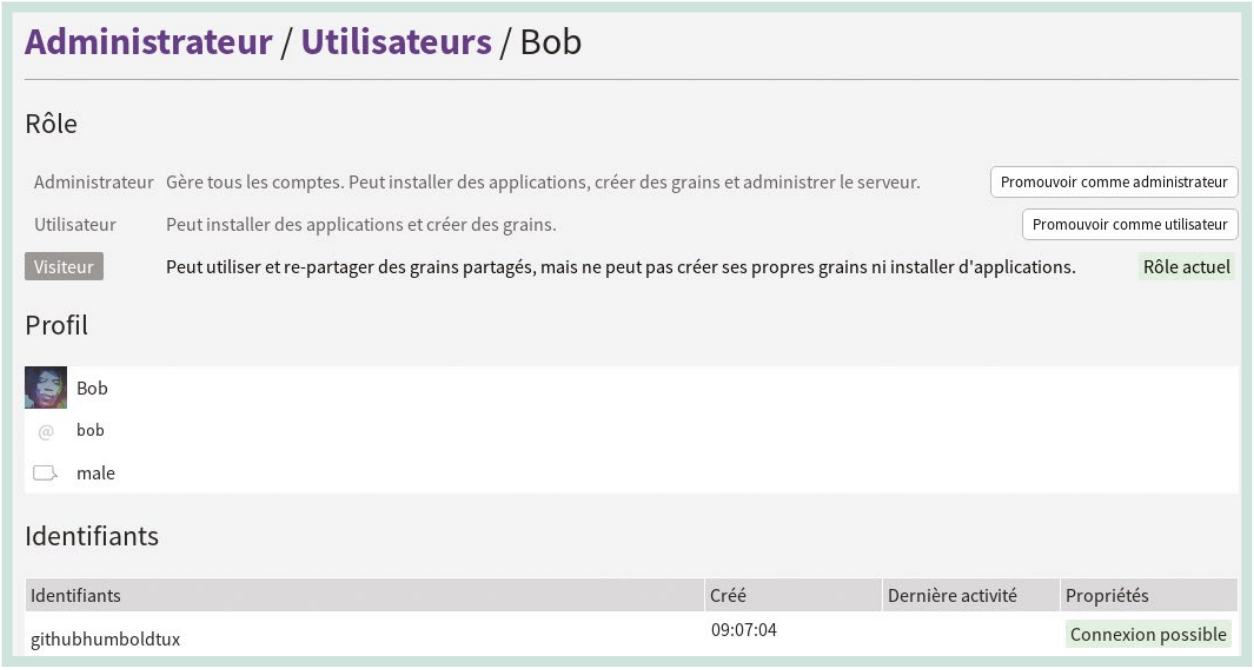


FIGURE 6. Gestion d'un compte utilisateur.

• ***Administrateur*** : la personne peut en plus gérer l'instance.

Par défaut, une instance Sandstorm est publique. C'est-à-dire que n'importe qui peut créer un compte, s'il ou elle a l'adresse de votre instance, suivant la ou les sources d'authentification activées. Le compte sera par défaut un compte ***Visiteur***, donc limité, et n'aura aucune application accessible. Un administrateur peut aussi depuis la page d'administration des utilisateurs envoyer une invitation à quelqu'un : ce compte aura les privilèges ***Utilisateur***, et pourra exécuter ses propres grains. Un administrateur peut élever les privilèges d'un compte depuis la page de gestion individuelle d'un compte (Figure 6), en cliquant sur le compte de son choix depuis la page de gestion des utilisateurs.

6. PARTAGE

Dans Sandstorm, chaque application est privée par défaut, et accessible via l'interface de Sandstorm. Plutôt que de créer des utilisateurs et donner des permissions depuis l'application elle-même, comme ce serait fait naturellement si on l'utilisait directement, on passe par le système de partage de grain de Sandstorm. Chaque application est modifiée et adaptée pour tirer parti du modèle d'exécution et d'isolation sous forme de grain propre à Sandstorm. Ainsi, pour chaque application, Sandstorm redéfinit des capacités que le propriétaire d'un grain peut déléguer à une autre personne. Par exemple, pour notre grain Etherpad, on peut déléguer le droit de lecture, de commenter ou d'écriture. On délègue des capacités depuis le lien ***Partage d'accès*** dans la barre supérieure du grain désiré (Figure 7, page suivante). On peut déléguer de deux manières, en envoyant un lien de partage directement par e-mail, ou en créant et partageant le lien de partage manuellement.

La personne qui ouvrira le lien pourra dans le cas d'un Etherpad l'utiliser suivant les capacités déléguées. Si la personne n'a pas de compte sur le Sandstorm, il lui sera proposé

FIGURE 7. Délégation de capacités pour un grain.

soit d'utiliser l'Etherpad en mode anonyme, soit de créer un compte sur le Sandstorm (ou d'utiliser son compte si elle en a un). Dans ce cas, le grain sera ajouté à la liste de ses grains dans l'application. Elle pourra en plus déléguer les mêmes capacités à un autre utilisateur si elle le désire, toujours sans passer par le système interne d'utilisateurs proposé originellement par l'application.

Les capacités déléguables et la manière d'utiliser un grain dépendent fortement néanmoins de l'application utilisée : par exemple, pour un site WordPress, si on vous délègue des droits administrateurs ou d'édition, il faudra vous connecter ou créer un compte Sandstorm pour pouvoir utiliser le site WordPress.

Un autre moyen de partager l'utilisation et les données d'un grain est de créer une clé API Sandstorm

FIGURE 8. Génération d'une clé web.

depuis la barre supérieure en cliquant l'icône en forme de clé (Figure 8). Clé que vous pourrez utiliser ensuite depuis une autre application pour vous connecter à ce grain.

CONCLUSION

Nous avons vu comment exécuter et partager des applications sur Sandstorm. La grande force de la plateforme est l'isolation de l'exécution de chaque application. C'est aussi cela qui la rend particulière, et nécessite un empaquetage spécial d'une application pour qu'elle puisse s'exécuter sur Sandstorm. Ce qui freine l'adoption et la mise à jour des applications. C'est pourquoi je vous invite à rejoindre la communauté d'utilisateurs de Sandstorm **[7]**, et pourquoi pas celle de ses développeurs **[8]**. ■

RÉFÉRENCES

[1] https://apps.sandstorm.io

[2] https://sandstorm.io/how-it-works

[3] https://docs.sandstorm.io/en/latest/administering/wildcard

[4] https://docs.sandstorm.io/en/latest/administering/sandcats

[5] https://oasis.sandstorm.io

[6] https://docs.sandstorm.io/en/latest/install

[7] https://sandstorm.io/community

[8] https://docs.sandstorm.io/en/latest/developing

GNU LINUX PRATIQUE

Abonnez-vous !

NOS TARIFS s'entendent TTC et en euros*		PAPIER	
OFFRE	ABONNEMENT	Réf	Tarif TTC
LP	6n° LP	☐ LP1	39 €
LP+	6n° LP + 3n° HS	☐ LP+1	69 €
	COUPLAGES AVEC NOS AUTRES MAGAZINES		
A+	6n° LP + 3n° HS + 11n° GLMF + 6n° HS	☐ A+1	198 €
C+	6n° LP + 3n° HS + 11n° GLMF + 6n° HS + 6n° MISC + 2n° HS	☐ C+1	263 €
K+	6n° LP + 3n° HS + 4n° HK	☐ K+1	111 €
L+	11n° GLMF + 6n° HS + 6n° LP + 3n° HS + 6n° MISC + 2n° HS + 4n° HK	☐ L+1	305 €

CONSULTEZ NOS OFFRES D'ABONNEMENT CONNECT SUR : www.ed-diamond.com

Les abréviations des offres sont les suivantes :
GLMF = GNU/Linux Magazine France
HS = Hors-Série | LP = Linux Pratique
HK = Hackable

J'indique l'offre si différente que celles ci-dessus :

J'indique la somme due (Total) : €

Je choisis de régler par :

- ☐ **Chèque bancaire ou postal à l'ordre des Éditions Diamond (uniquement France et DOM TOM)**
- ☐ **Pour les règlements par virements, veuillez nous contacter par e-mail : cial@ed-diamond.com ou par téléphone : +33 (0)3 67 10 00 20**

SÉLECTIONNEZ VOTRE OFFRE DANS LA GRILLE CI-DESSUS ET RENVOYEZ CE DOCUMENT COMPLET À L'ADRESSE CI-DESSOUS !

Voici mes coordonnées postales :	
Société :	
Nom :	
Prénom :	
Adresse :	
Code Postal :	
Ville :	
Pays :	
Téléphone :	
E-mail :	

- ☐ Je souhaite recevoir les offres promotionnelles et newsletters des Éditions Diamond.
- ☐ Je souhaite recevoir les offres promotionnelles des partenaires des Éditions Diamond.

En envoyant ce bon de commande, je reconnais avoir pris connaissance des conditions générales de vente des Éditions Diamond à l'adresse internet suivante : http://**boutique.ed-diamond.com/content/3-conditions-generales-de-ventes** et reconnais que ces conditions de vente me sont opposables.

Les Éditions Diamond
Service des Abonnements
10, Place de la Cathédrale
68000 Colmar – France
Tél. : + 33 (0) 3 67 10 00 20
Fax : + 33 (0) 3 67 10 00 21

Vos remarques :

.....

.....

RETROUVEZ TOUTES NOS OFFRES SUR : www.ed-diamond.com !

*Les tarifs hors France Métropolitaine, Europe, Asie, etc. sont disponibles en ligne !

COMMENT LIRE GNU LINUX PRATIQUE ?

➡ EN VERSION PAPIER...

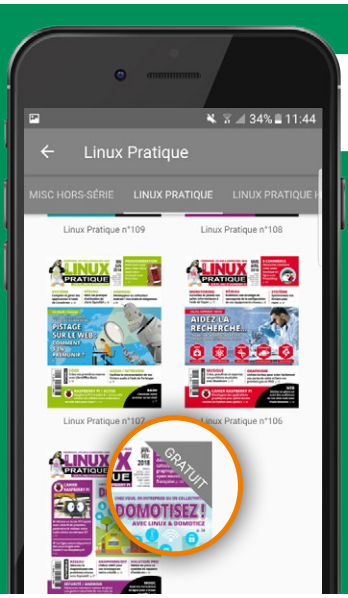
M'ABONNER ! ME RÉABONNER ! COMPLÉTER MA COLLECTION ! ...

Rendez-vous sur :

www.ed-diamond.com

pour consulter toutes les offres d'abonnements ou renvoyez-nous ce document complété !

➡ SUR VOTRE SMARTPHONE OU TABLETTE...



NOUVEAU !

Téléchargez notre application

DIAMOND KIOSK

DISPONIBLE SUR Google Play

Télécharger dans l'App Store

1 N°offert pour découvrir l'application

➡ OU EN NUMÉRIQUE...

CONNECT GNU LINUX PRATIQUE

Découvrez Connect

la plateforme de documentation numérique !

Retrouvez tous les articles de Linux Pratique dès leur parution en ligne et accédez aux archives du magazine !

Pour plus de renseignements, contactez-nous :
par téléphone au +33 (0) 3 67 10 00 20 ou par e-mail à cial@ed-diamond.com

À découvrir sur : **connect.ed-diamond.com**

ÉCRIRE UNE APPLICATION utilisant une gestion de droits

MICHAËL BERTOCCHI

LORSQUE VOUS DÉVELOPPEZ UNE APPLICATION, VOUS POUVEZ AVOIR BESOIN DE DIFFÉRENCIER/HIÉRARCHISER DES NIVEAUX DE DROITS EN FONCTION DE L'UTILISATEUR CONNECTÉ, C'EST CE QUE NOUS VERRONS DANS CET ARTICLE.

INTRODUCTION

Lorsque nous développons une application, dans certains cas, nous devons restreindre les actions de certains groupes d'utilisateurs. Pour cela, nous utilisons des ACL (*Access Control List*) : c'est la mise en place d'une gestion de droits applicative. Dans les faits : nous définissons des objets ainsi que des niveaux de droits dessus (lecture, écriture…) et enfin, nous déterminons quels groupes y sont habilités.

Dans cet article, nous allons créer une application de catalogue de produits :

- certains utilisateurs pourront en ajouter (l'équipe commerciale) ;
- d'autres pourront mettre à jour les stocks (l'équipe logistique) ;
- enfin, les derniers pourront administrer les utilisateurs et les droits (les administrateurs).

Nous utiliserons ici un framework open source : le mkframework. Créé en 2009, ce framework PHP facile à prendre en main propose un générateur web (le *Builder*) plutôt qu'une invite de commande.

1. UNE BASE DE DONNÉES POUR LES STOCKER TOUS

Comme dans la plupart des applications, nous allons avoir besoin de stocker des informations, pour cela nous utiliserons ici une base de données MySQL (ou MariaDB). Nous aurons besoin de tables contenant les produits, une autre les catégories et nous verrons plus tard pour la suite (utilisateurs et gestion de droits) (voir figure 1, page suivante).

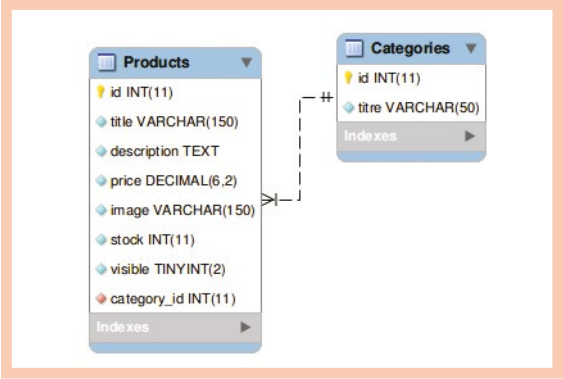


FIGURE 1. Nos tables de départ.

2. AVANT DE CONSTRUIRE NOTRE ÉDIFICE, COMMENÇONS PAR BÂTIR LES FONDATIONS

Rendez-vous sur le site du mkframework : **http://mkframework.com/**, rubrique ***Télécharger***, cliquez sur le package contenant l'ensemble pour démarrer. Vous avez une archive tar.gz que je vous invite à désarchiver dans votre répertoire web.

Ouvrez votre navigateur à l'adresse de votre installation Apache (ou Nginx). Vous avez ici 3 répertoires :

- Builder/ qui va vous permettre de générer vos applications, les administrer et générer d'autres éléments très facilement (couche modèle, modules CRUD…) ;
- Lib/ qui contient les fichiers du framework (seule partie à copier en production) ;
- Projects/ qui contiendra les applications que vous aurez créées.

Cliquez sur le répertoire Builder/ pour commencer à créer notre application (figure 2).

Vous avez ici l'application de génération de code du framework, notez une inscription rouge qui vous demande de rendre inscriptible votre répertoire Projects par l'utilisateur web (www-data, apache ou autre : le nom de l'utilisateur exécutant votre serveur web). Le builder vous propose plusieurs templates d'application, nous allons nous concentrer sur la dernière (la plus complexe, mais la plus pérenne).

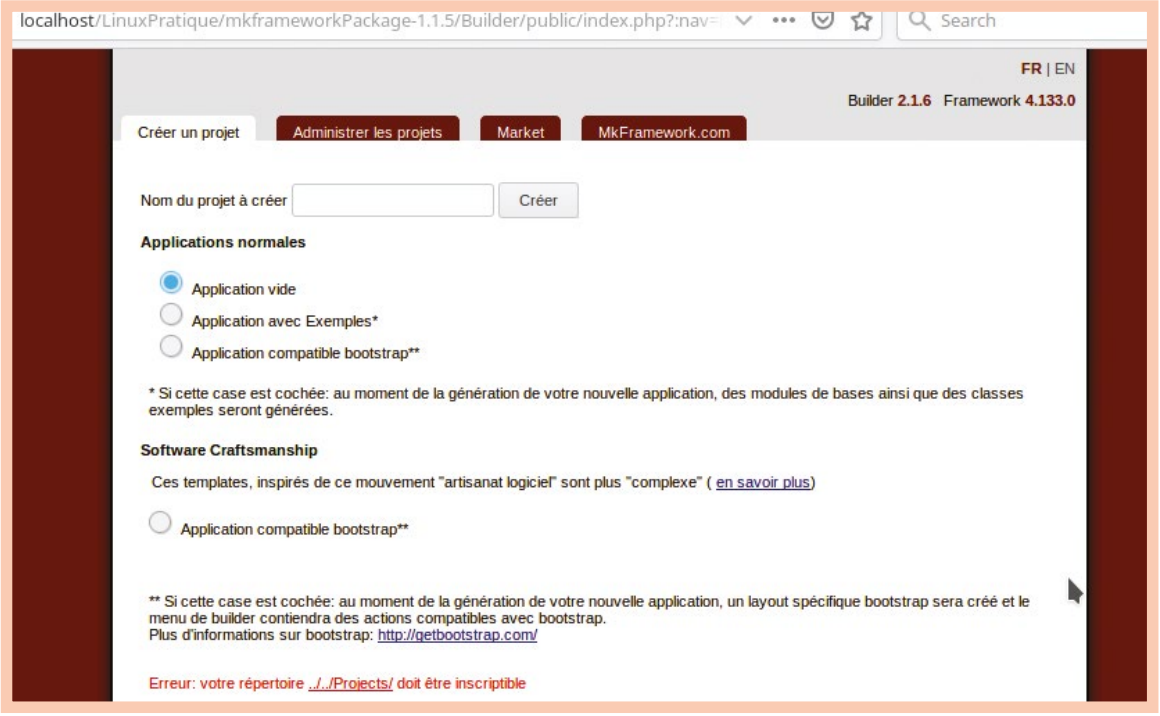
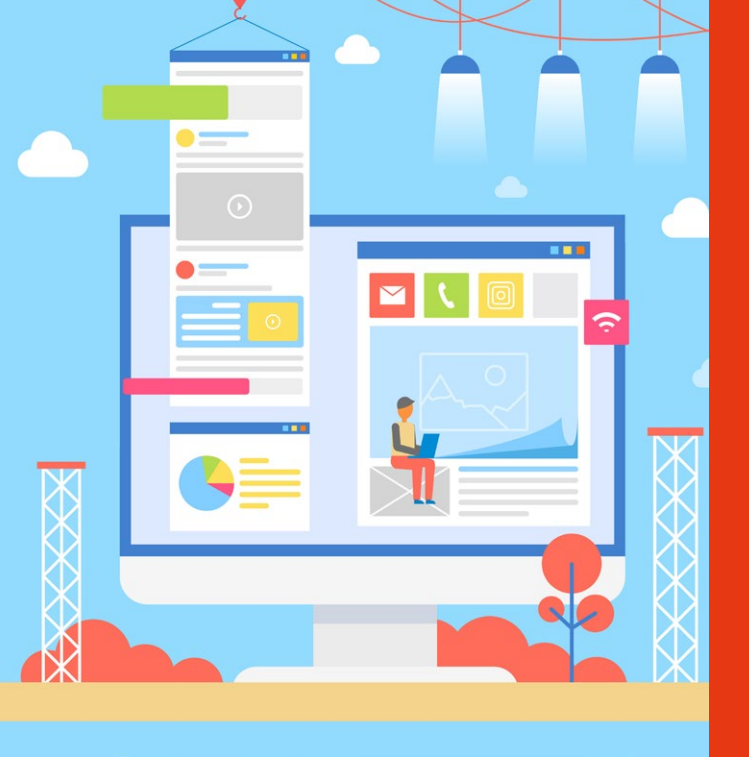


FIGURE 2. Page du Builder du mkframework.



Indiquez le nom de votre application et validez, vous allez être redirigé sur la page d'administration de vos projets. Vous avez trois boutons permettant d'administrer votre projet, de l'explorer ou de le visualiser. Votre application a été créée dans le répertoire Projects/.

3. PRÉSENTATION DE NOS DEUX PROTAGONISTES

Pour indiquer où est la base de données et comment s'y connecter, éditez le fichier de configuration conf/connexion.ini.php (soit via le bouton ***Explorer le projet***, soit avec votre éditeur de code). Modifiez l'exemple pdo/mysql et supprimez les autres exemples (figure 3).

4. INSTALLEZ-VOUS CONFORTABLEMENT ET LAISSEZ VOUS GUIDER

Sur notre application, nous voulons gérer des autorisations d'accès, mais pour cela, il faut d'abord identifier des utilisateurs avec leur groupe d'habilitation, nous devons donc démarrer par l'ajout d'une mécanique d'authentification.

Une des forces des frameworks est de vous simplifier le démarrage d'une application, ici, le framework ne déroge pas à la règle en vous proposant de vous mâcher le début du travail.

Ouvrez la page sur le Builder, et cliquez sur générer le lien ***Créer un module d'authentification + gestion de droits*** (figure 4, page suivante).

Le Builder vous informe des tables qui seront nécessaires pour générer cette mécanique : vous avez un schéma avec des indications graphiques plus le code MySQL pour générer ces tables.

Note : Vous pouvez créer ces mêmes tables avec d'autres noms : vous aurez un formulaire d'association de champs dans l'étape d'après : ce n'est qu'une proposition « clé en main ».

Cliquez sur ***Suivant*** une fois vos tables créées, vous allez ensuite avoir un formulaire de renseignement du module d'authentification à créer (figure 5, page suivante).

Le mkframework utilise le design pattern MVC (Modèle Vue Contrôleur), il nécessite donc un ensemble de

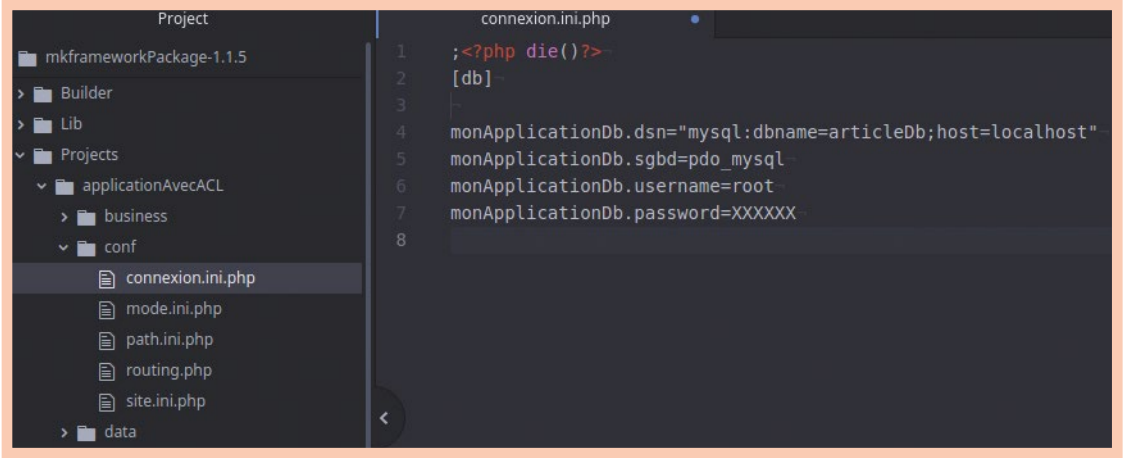


FIGURE 3. Fichier de configuration des connexions.

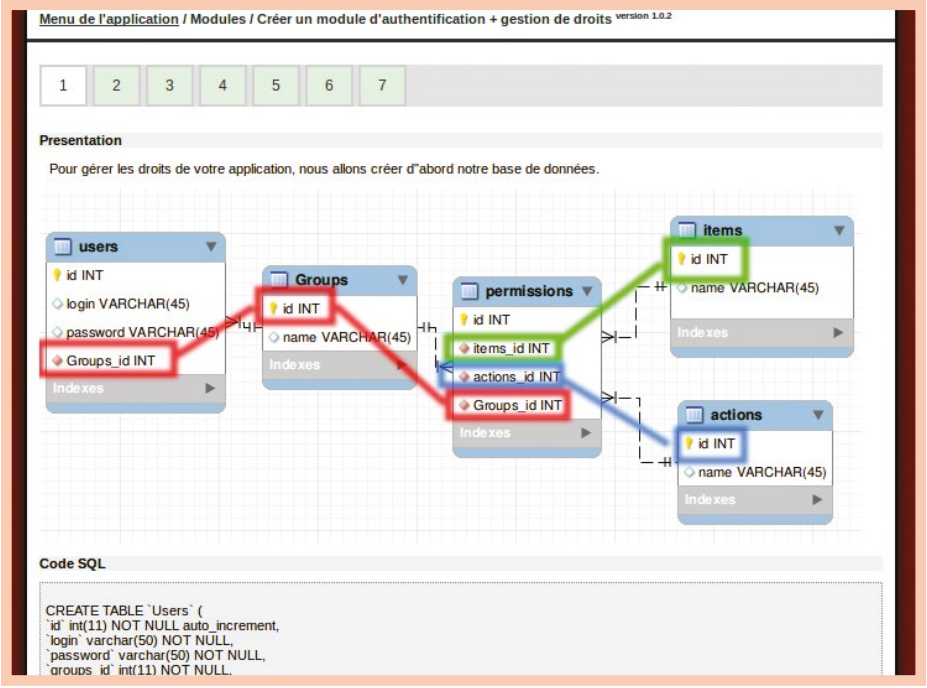


FIGURE 4. Page 1 de création d'un module d'authentification dans le Builder.

classes d'abstraction pour interagir avec la partie hébergeant les données (ici, notre base MySQL, mais ce peut être des fichiers XML, CSV, JSON…). Cliquez sur le profil de connexion à utiliser.

Validez cette étape pour voir s'afficher le formulaire d'association de vos tables en base de données avec les tables/champs dont il a besoin pour créer le module. Une fois les tables validées, il fera de même pour les champs à associer. Enfin, il va vous demander vers quel module parent rediriger une fois connecté (ce template permet l'héritage de modules) : pour l'instant, nous n'avons encore rien écrit, nous allons rediriger vers le module par défaut. Validez pour voir lister les fichiers et répertoires ajoutés.

Le Builder vous indique qu'il vous reste à modifier votre fichier de configuration pour activer l'authentification sur votre application et indiquer le couple module/action qui implémente le formulaire de connexion.

Note : Modifiez juste le champ module à `private_auth::login`, pas besoin de passer le champ `enabled` à `1`, vous verrez plus tard pourquoi.

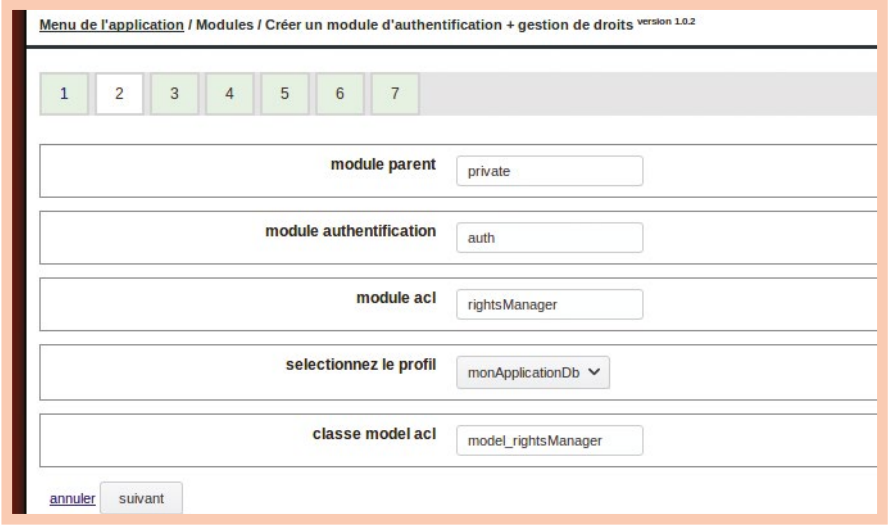


FIGURE 5. Page 2 de création d'un module d'authentification dans le Builder.

5. L'ASSIETTE VIENT D'ÊTRE POSÉE SUR LA TABLE, POSONS NOS COUVERTS ET OBSERVONS UN PEU TOUT ÇA

5.1 Un module d'authentification

Le but de l'authentification est double ici : d'une part, identifier qui se connecte (a-t-il les bons identifiants confirmant son identité) et d'autre part récupérer son groupe d'habilitation pour lui attribuer les droits qui lui reviennent. Ici, le Builder vous génère un

module : c'est un répertoire incluant un contrôleur (fichier main.php), les fichiers de langue (répertoire i18n), ainsi que le(s) vue(s) correspondante(s) (répertoire view).

Fichier module/private/auth/main.php (généré par le Builder) :

```
<?php
class module _ private _ auth extends module _ private {

        protected $ _ sModulePath = 'private/auth';

        public function getView($sView _ ) {
                return new _ view($this-> _ sModulePath .'::'.$sView _ );
        }

        public function _ login() {

                $sMessage = $this->checkLoginPass();

                $oView = $this->getView('login');
                $oView->sError = $sMessage;

                $this->oLayout->add('main', $oView);
        }

        private function checkLoginPass() {
                //if form is not sent, we stop there
                if (! _ root::getRequest()->isPost()) {
                        return null;
                }

                $sLogin = _ root::getParam('login');
                $sPassword = _ root::getParam('password');

                $oBusinessAuth = new business _ auth(model _ auth::getInstance(), _
root::getAuth(), _ root::getI18n() );
                if (false === $oBusinessAuth->checkCredentials($sLogin, $sPassword)) {
                        return $oBusinessAuth->getReturn()->getData('error');
                }

                $oUser=$oBusinessAuth->getReturn()->getData('oAccount');
                $oBusinessRightManager = new business _ rightsManager(model _
rightsManager::getInstance(), _ root::getACL(), _ root::getI18n(), new plugin _ sc _ valid() );
                $oBusinessRightManager->loadForUser($oUser);

                _ root::redirect('global _ default::index');
        }

        public function _ logout() {
                _ root::getAuth()->logout();
        }
}
```

On peut voir ici que ce module inclut deux pages (ou « actions » selon), on les distingue par le fait que ce sont deux méthodes publiques commençant par le caractère « underscore » : _login et _logout, qui comme leur nom l'indique sont les pages de connexion et de déconnexion.

Notez que le module module_private_auth hérite du module du module parent module_private :

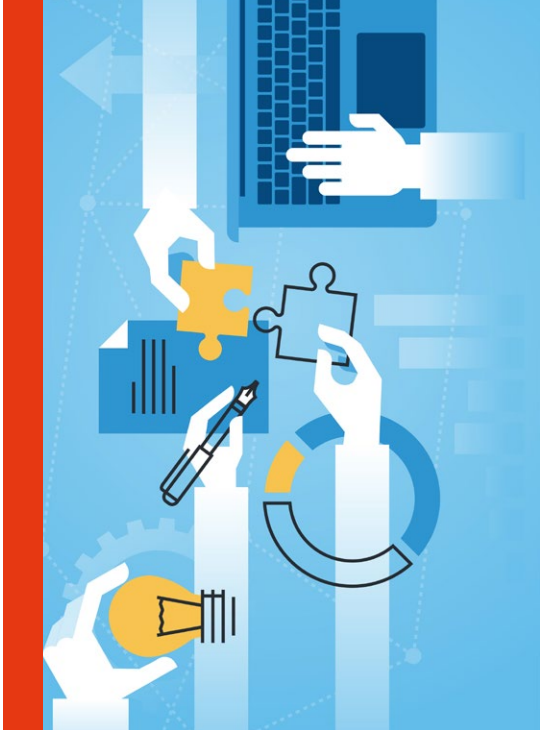
```
<?php
class module _ private extends abstract _ module {

        protected $oLayout;

        public function before() {
                //we enable authentification
                _ root::getAuth()->enable();

                $sLang = _ root::getConfigVar('language.default');
                if (isset($ _ SESSION['lang'])) {
                        $sLang = $ _ SESSION['lang'];
                }
                _ root::getI18n()->load($sLang);
                _ root::getI18n()->loadFromDir( _
root::getConfigVar('path.module').$this-> _ sModulePath.'/i18n/');

                $this->oLayout = new _ layout('bootstrap');
        }


        public function after() {
                $this->oLayout->show();
        }

}
```

On y distingue principalement deux méthodes « before » et « after », qui permettent de spécifier ce qu'il faut faire avant et après l'appel de la page demandée. Dans l'idée, on appelle d'abord la méthode « before » (de la classe parente, ou du module dans le cas d'une surcharge, puis la méthode de l'action/page demandée (ici la page « login », donc la méthode _login()) et enfin la méthode « after ».

Avec l'héritage, on mutualise cette méthode « before », on y initialise le layout/template du site, on charge le/les fichiers de langue, on charge si besoin le(s) menu(s), et ici on active l'authentification : tous les modules qui en hériteront nécessiteront que l'internaute soit authentifié pour y accéder. Puis, on appelle la méthode de l'action demandée, ici donc méthode _login() où l'on va charger la vue du formulaire de connexion, l'assigner dans un emplacement (placeholder) sur notre layout (précédemment initié), on vérifie également les identifiants afin de charger les permissions de l'utilisateur identifié. Enfin, on appelle la méthode after() pour afficher l'ensemble (layout + vues assignées) à l'écran.

5.2 Un module d'administration de droits

Un module permettant de gérer les droits est également généré, ceci vous facilitera la suite.



5.3 La couche modèle

Le Builder génère une classe modèle intégrant les différentes requêtes nécessaires pour vérifier les identifiants, trouver les différentes permissions ainsi qu'ajouter les différents éléments de paramétrage des ACL en base.

5.4 Des classes business et leurs interfaces

Nous avons fait le choix ici de créer une application de type « Software Craftsmanship », celle-ci privilégie une architecture hexagonale, ce faisant : on déporte la logique métier dans des classes extérieures (préfixées business_), celles-ci reçoivent tous les éléments nécessaires à leur fonctionnement et doivent respecter une interface. Ces deux types de classes ont été générés par le Builder, ici business_auth et business_rightManager.

Ici les avantages sont nombreux :

- centraliser le code métier à un seul endroit, et donc alléger le code du contrôleur, n'ayant qu'à appeler ces classes métier ;

- faciliter la mise en place de tests unitaires ;
- permettre de pouvoir facilement faire évoluer ces briques (par exemple, passer à une authentification Active Directory/LDAP ou via un SSO/Webservice tiers) ;
- ajouter un/plusieurs appels à ce code métier, par exemple via une API Rest/Soap/ligne de commandes...

5.5 Des tests unitaires

Toujours dans l'esprit « Software Craftsmanship », des tests unitaires sont générés afin de sécuriser vos classes business.

6. LA CRÉATION DU CATALOGUE DE PRODUITS

Créez la couche modèle de la table produits et catégories en passant par le Builder (pensez pour la table « Categories » à activer la génération de la méthode getSelect() qui simplifiera la mise en place du CRUD en retournant un tableau indexé facilitant la mise en place de menu déroulant).

Créons une classe model_YesNo.php qui permettra d'avoir un menu déroulant oui/non :

```
<?php
class model _ YesNo
{
    protected $tValue=array();

    protected static $oInstance=null;

    public function _ _ construct()
    {
        $this->tValue=array(
                        tr('label _ no'),
                        tr('label _ yes'),
                       );
    }

    public static function getInstance()
    {
        if (self::$oInstance===null) {
            self::$oInstance=new model _
YesNo();
        }
        return self::$oInstance;
    }

    public function getSelect()
    {
        return $this->tValue;
    }
}
```

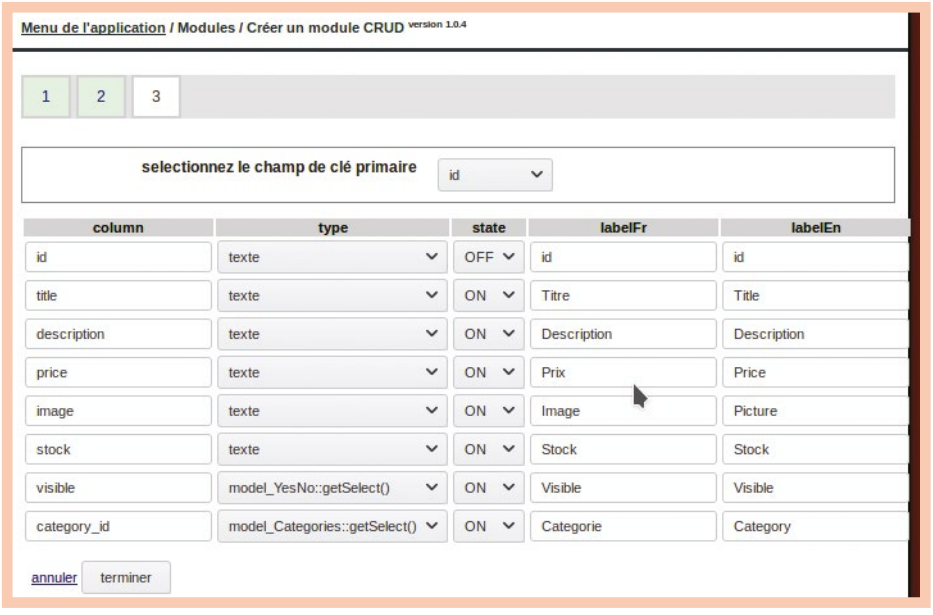


FIGURE 6. Page de création d'un module CRUD.

Dans le Builder, cliquez sur le lien ***Créer un module CRUD***, sélectionnez votre classe modèle model_Products, choisissez comme module parent private et finalisez le formulaire ainsi (figure 6).

Nous avons mis off sur le champ de clé primaire (pour ne pas l'afficher à l'écran), nous avons sélectionné pour les champs « visibles » et « category_id » une classe avec sa méthode getSelect : vous aurez un menu déroulant dans les formulaires et dans les pages d'affichage un joli

label plutôt qu'un id. Enfin, notez que vous pouvez dans ce formulaire saisir un libellé en français et en anglais pour chacune des colonnes (en effet, l'application est naturellement multilingue). Faites de même pour la table « Categories ».

7. UNE PETITE PAUSE S'IMPOSE : OBSERVONS CE QUE NOUS AVONS

Si, dans le Builder vous cliquez sur le bouton ***Voir le site***, vous allez voir une simple page quasiment vide, en bas à gauche, un bouton permet d'afficher la barre de debug. Nous pouvons voir que nous sommes sur l'action index du module global_default. Nous pouvons modifier ceci dans le fichier conf/site.ini.php :

```
[navigation]
scriptname=index.php
var=:nav
module.default=private _ Products
action.default=index
```

Si vous réactualisez la page, vous devriez atterrir sur la page de connexion : en effet, vous êtes redirigé sur la page de login, car le module Products hérite du module module_private. Il nous faut au moins un utilisateur pour pouvoir commencer, pour cela nous allons temporairement désactiver l'authentification, le temps de renseigner un utilisateur (en commentant l'authentification). Pour cela, éditez le fichier module/private/main.php, et désactivez la première ligne de la méthode before() :

```
class module _ private extends abstract _ module {

        protected $oLayout;

        public function before() {
                //we enable authentification
                //nous desactivons temporairement
l'authentification
                // _ root::getAuth()->enable();
```

Cette fois-ci, si vous cliquez dans le Builder sur ***Voir le site***, vous arrivez bien sur le module Products, mais il manque un menu. Cliquez sur ***Créer un module menu***. Vous pouvez ici cocher les cases des actions à faire apparaître dans le menu ainsi que le libellé affiché, validez (n'oubliez pas la ligne sur la méthode logout pour vous déconnecter).

Le builder vous affiche à la fois la liste des fichiers générés, mais également le code à inclure dans notre application pour voir apparaître ce menu. Copiez ce code, et collez-le dans la méthode before() du module parent module_private (module parent de Products et Categories). Éditez la méthode before() de notre fichier module/connecter/main.php :

```
    public function before(){
        //we enable authentification
        //nous desactivons temporairement
l'authentification
        // _ root::getAuth()->enable();

        $sLang = _ root::getConfigVar('language.default');
        if (isset($ _ SESSION['lang'])) {
            $sLang = $ _ SESSION['lang'];
        }
        _ root::getI18n()->load($sLang);
        _ root::getI18n()->loadFromDir( _
root::getConfigVar('path.module').$this-> _ sModulePath.
'/i18n/');

        $this->oLayout = new _ layout('bootstrap');
        //assignez le menu a l'emplacement menu
        $this->oLayout->addModule('menu', 'menu::index');
    }
```

Si vous réactualisez la page, vous allez voir apparaître le menu en haut de la page, modifiable dans son fichier module/menu/main.php :

```
<?php
class module _ menu extends abstract _ moduleembedded
{
    public function _ index(){
        $tLink=array(
          'Produits' => 'private _ Products::index',
          'Categories' => 'private _ Categories::index',
```

```
            'Droits' => 'private _
rightsManager::index',
            'Se deconnecter' => 'private _
auth::logout',
                );

        $oView=new _ view('menu::index');
        $oView->tLink=$tLink;

        return $oView;
    }
}
```

Pour permettre d'avoir un peu de matière, je vous invite à profiter de ce menu pour vous rendre dans le module Catégories et en ajouter quelques-unes.

8. PARAMÉTRONS NOS NIVEAUX DE DROITS

Dans votre application, cliquez sur le menu ***Droits***, vous voyez deux sections : liste des permissions et liste des utilisateurs. Commençons par les permissions, vous avec un formulaire comportant trois colonnes : groupe, action et élément. L'idée est la suivante : on autorise un groupe d'utilisateur à faire telle action sur tel élément.

Nous allons ici paramétrer quelques droits (liste non exhaustive) :

GROUPE	ACTION	ÉLÉMENT
Admin	*access*	module_private_rightsManager
Commercial	*access*	module_private_Products
Commercial	*add*	db_Product
...	...	...
Logistique	*access*	module_private_Products
Logistique	*update*	db_Product
Logistique	*write*	db_Product.stock

Voilà une petite liste intéressante de permissions pour commencer, notez que seul le groupe admin a accès à cette page d'administration des droits, que les commerciaux peuvent ajouter et supprimer des produits ainsi que des catégories, et que la logistique ne peut que modifier le champ stock.

Il ne reste plus qu'à nous permettre de créer des utilisateurs et leur assigner un groupe de droits, pour cela il nous faut créer un module CRUD héritant du module « private » (comme vu précédemment via le Builder : en commençant par la couche modèle). Puis éditons notre menu pour l'ajouter (fichier module/menu/main.php) :

```
class module _ menu extends abstract _ moduleembedded{
    public function _ index(){
        $tLink=array(
            'Produits' => 'private _ Products::index',
            'Categories' => 'private _ Categories::index',
            'Droits' => 'private _ rightsManager::index',
            'Utilisateurs' => 'private _ Users::index',
            'Se deconnecter' => 'private _ auth::logout',
        );
```

Si vous vous rendez sur le module utilisateur pour ajouter un utilisateur, ceci ne fonctionnera pas, car pour des raisons de sécurité le mot de passe n'est pas stocké en clair, on stocke une empreinte/hash.

Vous avez généré un module Users, qui comme vu précédemment, fait appel à une classe business pour ses différentes actions. Si vous éditez la classe model model_auth, vous pourrez voir une méthode hashPassword(), qui est appelée dans la classe business_auth pour vérifier les identifiants. Nous allons donc ajouter sur notre classe business fraîchement générée business_Users un appel à cette méthode pour générer l'empreinte de mot de passe à enregistrer en base. Commençons par modifier notre classe business business_crudUsers, remplaçons :

```
<?php
class business _ crudUsers extends business _
abstract {

        protected $ _ oModel;
        protected $ _ oAuth;
        protected $ _ oI18n;
        protected $ _ oValid;
        protected $ _ tColumn = array( 'login',
'password',);

        public function _ _ construct(interface _
model $oModel _ , interface _ i18n $oI18n _ ,
interface _ valid $oValid _ ) {
                $this-> _ oModel = $oModel _ ;
                $this-> _ oI18n=$oI18n _ ;
                $this-> _ oValid=$oValid _ ;
        }
```

par :

```
<?php
class business _ crudUsers extends business _
abstract
{
    protected $ _ oModel;
    protected $ _ oAuth;
    protected $ _ oI18n;
    protected $ _ oValid;
    protected $ _ oAuthModel;
    protected $ _ tColumn = array(    'login',
'password');

    public function _ _ construct(interface _
model $oModel _ , interface _ i18n $oI18n _ ,
interface _ valid $oValid _ , interface _
businessAuthModelUser $oAuthModel _ ){
        $this-> _ oModel = $oModel _ ;
        $this-> _ oI18n=$oI18n _ ;
        $this-> _ oValid=$oValid _ ;
        $this-> _ oAuthModel=$oAuthModel _ ;
    }
```

Nous ajoutons dans le constructeur un objet respectant l'interface interface_businessAuthModelUser : c'est la classe modèle qui contiendra la fameuse méthode de génération d'empreinte. Puis, nous modifions la méthode d'insertion pour ajouter cet appel au générateur d'empreinte si et seulement si le champ est le mot de passe :

```
public function insertItem($oUsers _ , $tParam _ )
    {
        $oValid = $this->getCheck($tParam _ );
        if (!$oValid->isValid()) {
            return $this->sendReturn(false,
array('tError' => $oValid->getListError()));
        }

        foreach ($this-> _ tColumn as $sColummn) {
            if ($sColummn==='password') {
                $oUsers _ ->$sColummn = $this-> _
oAuthModel->hashPassword($tParam _ [$sColummn]);
            } else {
                $oUsers _ ->$sColummn = $tParam _
[$sColummn];
            }
        }

        $this-> _ oModel->insert($oUsers _ );

        return true;
    }
```

Enfin, pour fonctionner, il faut modifier l'appel à cette classe business modifiée dans le module concerné : ici il faut modifier la méthode processSave dans le fichier module/private/Users/main.php. Remplacez :

```
$oBusiness = new business _
crudUsers(model _ Users::getInstance(), _
root::getI18n(), new plugin _ sc _ valid() );
```

par :

```
$oBusiness = new business _
crudUsers(model _ Users::getInstance(),
_ root::getI18n(), new plugin _ sc _ valid(),
model _ auth::getInstance());
```

Une fois effectué, vous pourrez créer un premier utilisateur qui sera l'administrateur par exemple (le mot de passe est une empreinte/hash). Cliquez sur le menu ***Droits***, et attribuez à l'utilisateur tout juste ajouté le groupe « Admin » (qui doit être présent dans la liste des utilisateurs). Maintenant que nous avons un utilisateur enregistré, nous pouvons réactiver l'authentification (mise en commentaire précédemment) dans le fichier module/private/main.php.

Pour éviter d'être, à la connexion, redirigé sur le module global_default, éditez la méthode checkLoginPass du fichier module/private/auth/main.php. Remplacez la ligne :

```
_ root::redirect('global _ default::index');
```

par :

```
_ root::redirect('private _
Products::index');
```

Après avoir enregistré, nous pouvons ajouter un lien vers la page de connexion sur cette même page en éditant le fichier module/global/default/view/index.php :

```
<p><a href="<?php echo _
root::getLink('private _ auth::login')?>">Se
connecter</a></p>
```

9. LA LIMITATION D'ACCÈS SELON L'UTILISATEUR CONNECTÉ

Continuons en limitant l'accès aux différents modules, et ceci en ajoutant ces trois lignes dans votre module module_private (fichier module/private/main.php) :

```
public function before(){
        //we enable authentification
        //nous desactivons temporairement
l'authentification
        _ root::getAuth()->enable();

        $sLang = _
root::getConfigVar('language.default');
        if (isset($ _ SESSION['lang'])) {
            $sLang = $ _ SESSION['lang'];
        }
        _ root::getI18n()->load($sLang);
        _ root::getI18n()->loadFromDir( _
root::getConfigVar('path.module').$this-> _
sModulePath.'/i18n/');

        $this->oLayout = new _
layout('bootstrap');

        //assignez le menu a l'emplacement
menu
        $this->oLayout->addModule('menu',
'menu::index');
        //ici on verifie que l'on a acces
au module concerne
        if ( _ root::getModule()!='private _
auth' and false=== _ root::getACL()-
>can('acces', 'module _ '. _
root::getModule())) {
            die('Not allowed to acces to
module: '. _ root::getModule());
        }
    }
```

On a déjà une première étape ici, mais nous avons deux soucis : la page d'arrivée est le module Products auquel l'administrateur n'a pas accès et on affiche les liens des menus même ceux auxquels on n'a pas accès (on devrait les masquer) . Commençons par modifier la redirection lors de l'authentification, éditez le fichier module/private/auth/main.php. Modifiez dans la méthode checkLoginPass() (en fin de méthode) :

```
        if ( _ root::getACL()->can('acces', 'module _
private _ Products')) {
            _ root::redirect('private _
Products::index');
        } elseif ( _ root::getACL()->can('acces',
'module _ private _ rightsManager')) {
            _ root::redirect('private _
rightsManager::index');
        } else {
            die('Cas non gere: pas de module
autorise');
        }
```

Nous vérifions si l'utilisateur a accès au module Products, si oui, on redirige dessus, et sinon on vérifie l'accès au menu ***Droits*** de la même manière. Filtrons maintenant les liens du menu de la même manière. Éditez la méthode _index() du fichier module/menu/main.php :

```
    public function _ index(){
        $tLink=array(
            'Produits' => 'private _
Products::index',
            'Categories' => 'private _
Categories::index',
            'Droits' => 'private _
rightsManager::index',
            'Utilisateurs' => 'private _
Users::index',
            'Se deconnecter' => 'private _
auth::logout',

        );

        //bouclons sur les liens pour supprimer
ceux auxquels l'utilisateur n'a pas acces
        foreach ($tLink as $label => $link) {
            list($sModule, $sAction)=explode('::',
$link);

            if ($sAction!='logout' and false=== _
root::getACL()->can('acces', 'module _ '.$sModule)) {
                unset($tLink[$label]);
            }
        }
```

Nous avons ajouté une boucle sur le tableau des liens pour supprimer ceux auxquels l'utilisateur n'a pas accès (en excluant bien sûr le lien de déconnexion). Si vous réactualisez la page, vous pourrez voir disparaître tous les liens non autorisés. Il ne reste plus qu'à ajouter la gestion de droits sur le cœur de l'application, le module Products. Mais pour cela, il nous faudrait deux nouveaux utilisateurs : un commercial, et une personne de la logistique, disons Ben (commercial) et Nuts (logistique).

Dans le menu ***Droits***, vous pouvez ajouter la permission suivante :

GROUPE	ACTION	ÉLÉMENT
Admin	*access*	module_private_Users

En vous reconnectant, vous verrez à nouveau le menu ***Utilisateurs*** vous permettant d'ajouter ces deux nouveaux utilisateurs. Retournez sur le menu ***Droits***, section ***Utilisateurs*** pour leur définir leur groupe. Commençons par limiter l'affichage des boutons d'ajout, de modification et de suppression. Éditez le fichier module/private/Products/view/list.php :

```
<?php /*ACL*/if ( _ root::getACL()->can('update',
'db _ Product')):?>
       <a class="btn btn-success" href="<?php
              echo $this->getLink(
                     'private _ Products::edit',
                      array('id' => $oProducts->id)
              )
        ?>"><?php echo tr('edit') ?></a>
<?php endif;?>
```

Par exemple ici, on vérifie s'il a le droit de mettre à jour le produit pour lui afficher ou non le bouton d'édition, idem pour l'ajout d'un produit. Ben peut ajouter des produits et les modifier, il peut également ajouter des catégories, alors que Nuts n'a accès qu'aux produits et n'a pas le bouton ajouter. On peut faire de même pour le formulaire de produit (fichier module/private/Products/view/new.php) :

```
<?php /*ACL*/if ( _ root::getACL()-
>can('write', 'db _ Product.title')):?>
        <div class="form-group">
                <label class="col-sm-2
control-label"><?php echo tr('field.
title')?></label>
                <div class="col-sm-10"><?php
echo $oForm->getInputText('title',
array('class'=>'form-control')) ?></div>
        </div>
<?php endif;?>
```

Note : Cacher le bouton c'est bien, mais il faut penser que l'utilisateur pourrait deviner l'URL (assez prédictible en effet).

Pour pallier ce problème, il faut également mettre le contrôle de droits sur les méthodes _new() et _edit() du module Products. En modifiant la méthode _new() du fichier module/private/Products/main.php :

```
public function _ new(){
        if (false=== _ root::getACL()->can('add',
'db _ Product')) {
                die('Not allowed');
        }
```

Mais attention, en faisant cela, vous avez bien la disparition des champs non autorisés selon l'utilisateur, en revanche vous ne pouvez pas enregistrer de produit, car la classe business attend que tous les champs soient renseignés (même ceux non autorisés).

En effet, les classes CRUD business générées par le Builder incluent un mécanisme de vérification de la donnée, et par défaut la seule contrainte activée est la vérification de remplissage de chacun des champs (vous pouvez modifier/ajouter des contraintes dans la méthode getCheck() de cette classe business).

Deux choix : désactiver la vérification de remplissage du stock (non autorisé pour notre commercial) ou prendre en compte dans la classe business de la gestion de droits (c'est mieux). Pour cela, nous allons dans un premier temps modifier notre classe business_crudProducts comme suit :

```
class business _ crudProducts extends
business _ abstract{
    protected $ _ oModel;
    protected $ _ oAuth;
    protected $ _ oI18n;
    protected $ _ oValid;
    protected $ _ tColumn = array(
'title', 'description',  'price', 'image',
'stock', 'visible',      'category _ id',);
    protected $ _ oACL;
```

```
    public function _ _ construct(interface _ model
$oModel _ , interface _ i18n $oI18n _ , interface _ valid
$oValid _ , interface _ acl $oACL _ ){
        $this-> _ oModel = $oModel _ ;
        $this-> _ oI18n=$oI18n _ ;
        $this-> _ oValid=$oValid _ ;
        $this-> _ oACL=$oACL _ ;

       //desactivation de la liste des colonnes non
autorisees
       foreach ($this-> _ tColumn as $key => $sColumn) {
            if (false===$this-> _ oACL->can('write', 'db _
Product.'.$sColumn)) {
                unset($this-> _ tColumn[$key]);
            }
        }
    }
```

Notez ici l'ajout d'un argument à notre constructeur : l'objet de gestion d'ACL tout simplement (celui-ci implémentant l'interface des ACL bien entendu). Une boucle sur les colonnes prise en compte par cette classe désactive celle où l'utilisateur n'a pas le droit. Il ne reste plus qu'à modifier le constructeur au moment de l'appel dans notre module comme suit :

```
$oBusiness = new business _ crudProducts(model _
Products::getInstance(), _ root::getI18n(), new plugin _ sc _
valid(), _ root::getACL());
```

On ajoute simplement en dernier argument notre objet de gestion de droits et le tour est joué. En vous connectant avec Ben, vous pouvez bien ajouter/modifier un produit, interagir avec tous les champs sauf le stock, et à l'inverse, avec Nuts vous ne pouvez que modifier le stock. Bien sûr, vous pouvez mettre des sinon à chaque fois pour afficher le champ en lecture seule, car sinon il sera difficile pour le service logistique de savoir quel stock ils administrent.

```
<div class="form-group">
       <label class="col-sm-2 control-label"><?php echo
tr('field.title')?></label>
       <?php /*ACL*/if ( _ root::getACL()->can('write',
'db _ Product.title')):?>
              <div class="col-sm-10"><?php echo $oForm->
getInputText('title', array('class'=>'form-control')) ?></div>
       <?php else:?>
              <div class="col-sm-10"><?php echo
$this->oProducts->title ?></div>
       <?php endif;?>
</div>
```

En cliquant sur la barre de debug, vous pouvez voir un bouton ***Permissions*** : les permissions demandées, accordées ou non (rouge/vert) ; un bouton ***Session*** listant les variables de session et notamment les permissions chargées.

UN CAFÉ ET L'ADDITION POUR TERMINER

Vous avez pu voir de quelle manière on peut ajouter des droits à des groupes d'utilisateurs, pensez bien à veiller à la fois à l'affichage, mais également aux vrais méthodes/code appelés : vous n'êtes jamais à l'abri d'une tentative de piratage. Vous pouvez facilement passer d'un modèle monogroupe à un modèle multigroupe, mais pour cela, il faudra définir une règle addition/soustraction dans le cas de conflit de droits.

Vous pourrez bien évidemment écrire de vraies pages d'accès restreints respectant la charte graphique de votre site.

J'espère également vous avoir fait découvrir ce framework PHP, qui se veut différent et très orienté application professionnelle/sécurisée.

Nous avons ici utilisé un template de site qui est multilingue nativement, mais pour ne pas surcharger l'article, j'ai préféré ne pas détailler son fonctionnement ici.

Idem pour les tests unitaires qui ont bien été générés par le Builder, mais doivent ici échouer au vu des différentes modifications apportées à nos classes Business. ■

CONCEVEZ ET PROGRAMMEZ UN JEU DE RÉFLEXION

avec Pygame Zero

LAURENT DELMAS

LA CRÉATION D'UN JEU EST TOUJOURS AUSSI PASSIONNANTE, MAIS PEUT PARAÎTRE COMPLEXE BIEN QU'IL EXISTE UNE MULTITUDE DE LIBRAIRIES FACILITANT CETTE TÂCHE. POURQUOI NE PAS L'APPRENDRE VIA LA CRÉATION DU JEU MASTERMIND ? C'EST CE QUE NOUS ALLONS VOIR DANS CET ARTICLE.

Nous allons concevoir et programmer un jeu simple en nous appuyant sur le module Python Pygame.Zero **[1]**. Le jeu de réflexion est simple en lui-même et bien connu de tous. Il s'agit du MasterMind dont l'objectif et de retrouver un code couleur. Tout en gardant le principe, il est possible d'élaborer un jeu plus complexe comme ajouter des niveaux, un temps imparti pour retrouver le code, des bonus, des combinaisons de codes magiques voire même une aventure dont le but ultime serait d'obtenir un trésor ou bien de délivrer le village d'un dragon tyran ou tout simplement retrouver le code.

1. INSTALLATION

Dans l'ensemble de cet article, nous utilisons une version de Python égale ou supérieure à la version 3.6. Pour ne pas polluer notre système, nous allons créer un environnement virtuel Python avec VirtualEnv. Nous ne détaillerons pas ici toutes les subtilités

et intérêts des environnements virtuels **[2]**. Installons simplement les paquets virtualenv et virtualwrapper.

```
laurent@ASUS:~$ sudo pip install virtualenv virtualwrapper
```

Nous ajoutons quelques variables à notre profil en éditant le fichier .bashrc qui se trouve dans votre répertoire utilisateur.

```
laurent@ASUS:~$ nano ~/.bashrc
export WORKON_HOME=$HOME/.virtualenvs
export VIRTUALWRAPPER_PYTHON=/usr/bin/python3
source /usr/local/bin/virtualenvwrapper.sh
```

Puis créons notre environnement virtuel que nous appelons pygame avec la version 3 de Python.

```
laurent@ASUS:~$ mkvirtualenv pygame -p python3
```

Nous voyons apparaître devant notre prompt dans le terminal, le nom de notre environnement virtuel entouré de parenthèses. Cela signifie que toutes les commandes que nous allons lancer et en particulier les installations de modules Python le seront dans notre environnement virtuel. Pour en sortir, il suffit de saisir la commande deactivate.

```
(pygame) laurent@ASUS:~$ deactivate
laurent@ASUS:~$
```

Pour entrer à nouveau dans un environnement virtuel, il suffit d'utiliser la commande workon suivie du nom de l'environnement virtuel.

```
laurent@ASUS:~$ workon pygame
(pygame) laurent@ASUS:~$
```

Commençons tout d'abord par installer Pygame.Zero et ses dépendances.

```
(pygame) laurent@ASUS:~$ pip install pgzero
```

Voilà tout est prêt, nous pouvons passer aux choses intéressantes.

2. PRINCIPE DU JEU

Le principe du jeu est relativement simple. Il s'agit de retrouver le code couleur défini aléatoirement par l'ordinateur. Afin d'agrémenter le jeu, il est possible de lui ajouter certaines fonctionnalités telles que la gestion de niveaux de difficulté. Par exemple :

- Niveau 1 : l'ordinateur précisera pour chaque position de couleur si elle est correcte ou pas et si elle est bien placée ou non.
- Niveau 2 : l'ordinateur donnera uniquement le nombre de couleurs correctes bien placées et mal placées sans pour autant préciser l'emplacement de ces dernières. Au joueur de les retrouver.

Il est également possible de permettre au joueur de choisir la longueur du code à retrouver : 4, 5 ou 6 couleurs puis ensuite de choisir un niveau de difficulté comme nous venons de le voir.

L'historique des tentatives de code ainsi que les réponses associées doivent être visibles tout au long de la partie pour aider la déduction du code. De plus, il est facile de mettre en place un fond d'écran sous forme d'image pour égayer le jeu.

Avant de commencer, une étape de préparation est indispensable.

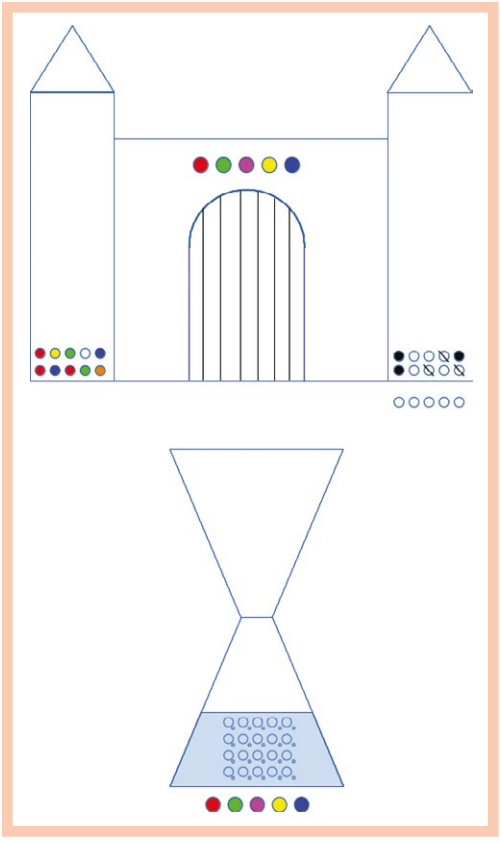
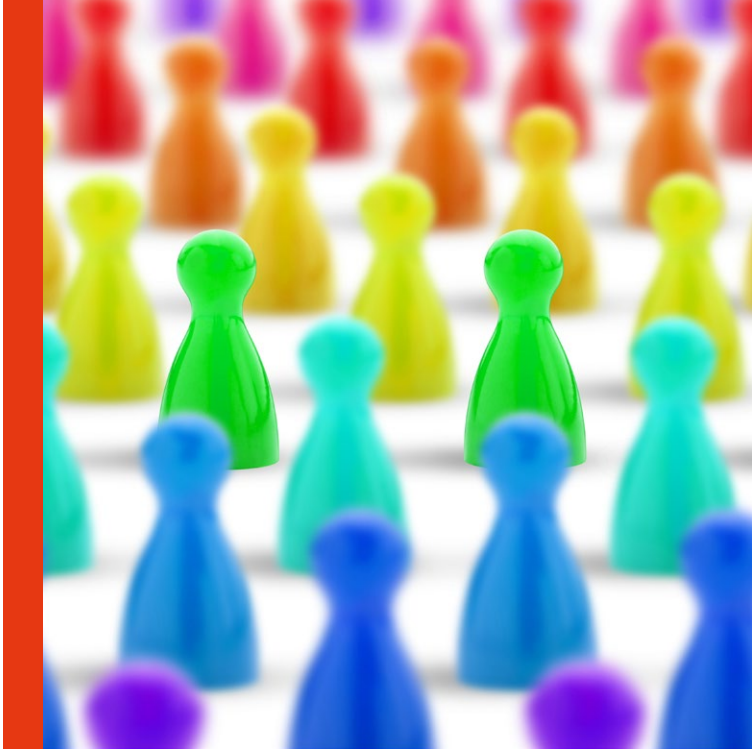


FIGURE 1. Exemples d'esquisses d'écran principal du jeu.



Elle consiste à collecter l'ensemble des éléments nécessaires pour la construction du jeu et si possible avoir une esquisse ou croquis de l'écran principal souhaité (Figure 1, page précédente).

Dans notre cas, nous avons besoin de jetons ou pions de différentes couleurs pour composer le code ainsi qu'une table de jeu où sera affiché l'ensemble des tentatives de code avec les réponses correspondantes. Dans notre cas, chaque pion sera représenté par une image avec un disque de couleur. Les réponses seront données de façon suivante :

- un pion bien placé s'affichera de manière normale ;
- un pion mal placé sera grisé ;
- un pion n'étant pas dans le code sera barré d'une croix.

En haut de l'écran se trouvera l'ensemble des couleurs disponibles et juste au-dessous des cases vides pour composer le code. La Figure 2 représente l'esquisse de l'écran de jeu.

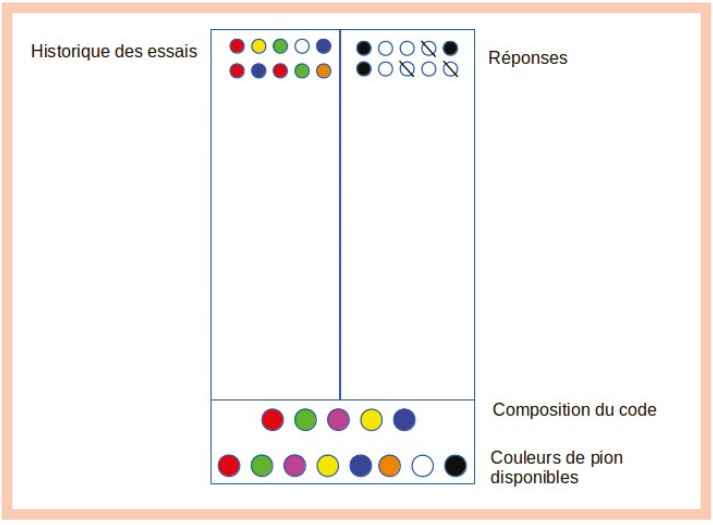


FIGURE 2. Esquisse de notre écran principal.

3. PYGAME ZERO

Pygame Zero a pour but de simplifier la création de jeux ou d'applications. Contrairement à Pygame, il n'est nul besoin de créer une boucle principale **[3][4][5][6][7]**. Cette dernière étant gérée de façon automatique via la fonction d'affichage `draw()` qui, comme son nom l'indique, gère l'affichage de la fenêtre principale du jeu.

Cependant ces simplifications ont des conséquences comme la limitation des entrées sorties. Pygame Zero prend en compte uniquement le clavier et la souris. Il n'est pas possible d'utiliser une manette de jeu. Un autre défaut, et particulièrement des versions antérieures à la version 1.2 de Pygame Zero, est l'impossibilité de lancer une application directement depuis l'interpréteur interactif Python IDLE ou PyCharm par exemple. Il faut donc utiliser la commande suivante dans un terminal :

```
(pygame) laurent@ASUS:~$ pgzrun main.py
```

La version 1.2 de Pygame Zero corrige en partie ce désagrément. Pour cela, il faut ajouter à la dernière ligne de notre programme la commande suivante :

```
pyzrun.go()
```

Tout au long de cet article, nous allons voir et implémenter différentes caractéristiques principales de Pygame Zero, en ajoutant progressivement les fonctionnalités à notre jeu.

Créons tout d'abord notre programme principal que nous appelons par exemple `master.py`. Importons les modules `pgzrun` et `pgzero` pour Pygame Zero. Définissons ensuite la fonction `draw()` qui, comme nous l'avons mentionné, gère directement la boucle principale.

```
import pgzrun
import pgzero

def draw() :
    screen.clear()

pgzrun.go()
```

Démarrons notre programme pour le constater. Vous voyez une fenêtre noire s'afficher. Pour changer la dimension de la fenêtre, il suffit de spécifier avant l'appel à la fonction draw() les variables HEIGHT et WIDTH qui correspondent respectivement à la hauteur et largeur de la fenêtre.

```
HEIGHT=800
WIDTH=1024
```

Pour changer la couleur de fond, il suffit d'utiliser la fonction screen.fill() en précisant les composantes RGB de la couleur voulue. Par exemple, screen.clear((255,255,128)) pour un fond jaune pâle.

3.1 Création des pions

Les pions seront représentés par des disques de couleur. Ces pions sont des images 50 x 50 au format PNG. Afin de savoir quelle couleur est sélectionnée, nous allons ajouter à chacun des boutons un état, ON ou OFF et lui attribuer un identifiant dont nous verrons par la suite l'utilité. L'état du bouton nous permettra de changer l'image associée au pion sélectionné afin que le joueur puisse voir ladite sélection.

Dans Pygame Zero, tous les éléments dynamiques sont déclarés avec la fonction Actor() à laquelle un certain nombre de paramètres sont passés tels que l'image, la position… comme nous allons le voir dans la suite. Toutes les images doivent se trouver dans un répertoire nommé images à la racine du programme. De plus, les fichiers doivent être impérativement en minuscules. Voici la déclaration de nos pions de couleur :

```
OFF=False
ON=True

pions=[]
pions.append(Actor('pion _ r0'))
pions[0].status=OFF
pions[0].id=0
pions.append(Actor('pion _ b0'))
pions[1].status=OFF
pions[1].id=1
pions.append(Actor('pion _ v0'))
pions[2].status=OFF
pions[2].id=2
pions.append(Actor('pion _ j0'))
pions[3].status=OFF
pions[3].id=3
pions.append(Actor('pion _ o0'))
pions[4].status=OFF
pions[4].id=4
pions.append(Actor('pion _ vi0'))
pions[5].status=OFF
pions[5].id=5
```

Nous ajoutons à cela cinq listes contenant chacune les noms des fichiers correspondant respectivement aux pions sélectionnés, non sélectionnés, bien placés, non présents et mal placés.

```
pionsON=['pion _ r0','pion _ b0','pion _ v0','pion _ j0','pion _ o0','pion _ vi0']
pionsOFF=['pion _ r1','pion _ b1','pion _ v1','pion _ j1','pion _ o1','pion _ vi1']
pionsOK=['pion _ r0','pion _ b0','pion _ v0','pion _ j0','pion _ o0','pion _ vi0']
pionsNOK='pion _ r0nok','pion _ b0nok','pion _ v0nok','pion _ j0nok','pion _
o0nok','pion _ vi0nok']
pionsMPL='pion _ r0mpl','pion _ b0mpl','pion _ v0mpl','pion _ j0mpl','pion _
o0mpl','pion _ vi0mpl']
```

3.2 Création du code

De même que pour les pions, notre code sera une liste initialisée avec des cases vides. Ici, le code sera composé de cinq éléments fixes. Pour ne pas ajouter de complexité, nous ne détaillerons pas les fonctions de niveaux de difficulté.

```
code=[]
code.append(Actor('pion _ vide'))
code[0].id=-1
code.append(Actor('pion _ vide'))
code[1].id=-1
code.append(Actor('pion _ vide'))
code[2].id=-1
code.append(Actor('pion _ vide'))
code[3].id=-1
code.append(Actor('pion _ vide'))
code[4].id=-1
```

3.3 Affichage du bandeau de sélection

L'affichage des éléments se fait, comme nous l'avons vu précédemment, au travers de la fonction draw(). Nous allons mettre les pions disponibles en haut. Chaque pion étant affiché par défaut depuis les coordonnées de son centre. Il suffit de les décaler les uns par rapport aux autres dans la boucle principale de la taille de l'image. Voilà les quelques lignes permettant cela :

```
centreX=25
centreY=25

#affichage bandeau selection
for i in range(0,len(pions)):
    pions[i].pos=((centreX+50*i),centreY)
    pions[i].draw()
```

De même pour l'affichage des cases vides pour la saisie du code qui se trouvent au-dessous des pions.

```
for i in range(0,len(code)):
    code[i].pos=((centreX+50*i),centreY+50)
    code[i].draw()
```

3.4 Composition du code

Pour pouvoir composer le code, il faut tout d'abord sélectionner le pion de la couleur voulue parmi celles disponibles puis le déposer dans la case de composition du code. Cela passe par l'utilisation de la souris. Fort heureusement, Pygame Zero nous facile la tâche en proposant l'implémentation des fonctions on_mouse_down(), on_mouse_up() et on_mouse_move() qui respectivement gèrent : l'appui, le relâchement et le déplacement de la souris. Il en est de même avec le clavier que nous n'utiliserons pas ici.

Nous allons implémenter la fonction on_mouse_down() pour réagir aux différents appuis des boutons de la souris. Lorsque nous appuierons sur la souris, la fonction vérifiera la position de celle-ci et si elle est sur l'un des pions de sélection ou sur une case du code. Pour ce faire, Pygame Zero utilise une propriété nommée collidepoint qui vérifie la collision des éléments du jeu, cela peut être le curseur de la souris avec un *Actor* ou bien plusieurs *Actors* entre eux.

Dans le premier cas (détection du pion sélectionné), nous changeons le statut du pion et le passons à ON afin d'informer le joueur. Dans le second cas (dépose du pion dans une case du code), nous appliquons à la case vide ou non l'image du pion sélectionné et lui attribuons également l'identifiant du pion.

```
def on _ mouse _ down(pos):
    global pions,sel,code
    for b in pions:
    #for i in range(1,len(pions)):
        if b.collidepoint(pos):
            b.status=ON
            sel[0].id=b.id
            #print("ON") #info pour test
        else:
            b.status=OFF
            #print("OFF") #info pour test
    for a in code:
        if a.collidepoint(pos):
           a.image=pionsOFF[sel[0].id]
           a.id=sel[0].id
```

Le changement de statut du pion a pour effet, dans la fonction update(), de mettre à jour le nom de l'image du pion en utilisant la liste correspondante que nous avons définie en tout début d'article.

```
def update():
    global pions
    for i in range(0,len(pions)):
        if pions[i].status==ON:
            pions[i].image=pionsON[i]
        else:
            pions[i].image=pionsOFF[i]
```

3.5 Vérification du code

Maintenant que nous avons fini de composer notre code, nous avons besoin d'un bouton pour en demander la vérification. Rien de plus simple, définissons comme pour les pions et le code un élément bouton en tant qu'*Actor* avec un statut initialisé à OFF.

```
bouton=[]
bouton.append(Actor('bouton _ off.png'))
bouton[0].status=OFF
```

Ajoutons également dans la fonction draw() la position d'affichage du bouton, juste à côté des pions à sélectionner.

```
bouton[0].pos=((centreX+50*(len(pions)+2)),c
entreY)
bouton[0].draw()
```

Il nous faut également détecter lorsque nous cliquons sur le bouton via la fonction de gestion de la souris (on_mouse_down()) afin que le joueur voit l'action sur le bouton. Pour cela, nous changeons l'image et passons le statut à ON.

```
if (bouton[0].collidepoint(pos)):
    bouton[0].status=ON
    bouton[0].image='bouton _ on.png'
```

Au fait, nous n'avons pas encore défini de code. Pallions à cet oubli en ajoutant une liste hidden en début de programme avec comme premier élément la valeur OFF. Celle-ci signifie qu'aucun code n'est défini.

```
hidden=[]
hidden.append(OFF)
```

Ensuite, créons la fonction secret() dont le but est de générer un code aléatoire. Ce code aléatoire est une liste dont chaque élément correspond à l'identifiant d'un pion. Cet identifiant est un nombre aléatoire obtenu avec la fonction randint du module random que nous nous empressons d'ajouter au-dessous des modules pgzero et pgzrun. Le premier élément ON ou OFF permet de signaler si un code est existant ou non.

```
def secret():
    global hidden
    hidden[0]=ON
    for i in range(0,len(code)):
        hidden.append(random.randint(0,5))
```

Nous appelons cette fonction en début de notre boucle principale c'est-à-dire dans la fonction draw(). Avant de générer un nouveau code, nous vérifions s'il en existe un ou pas avec le statut. S'il en existe un, il ne faut pas en régénérer un nouveau sinon à chaque passage dans la boucle principale le code serait changé.

```
if (hidden[0]==OFF):
    secret()
```

Il ne nous reste plus qu'à créer une fonction check() qui vérifie le code composé avec celui généré automatiquement. Certes la fonction ci-dessous n'est pas parfaite et il persiste probablement quelques « ratés » en particulier dans la recherche des pions mal placés. Le but de cet article n'étant pas de faire une fonction de comparaison optimisée et parfaite, je vous laisse donc le soin d'y apporter les corrections que vous jugez bonnes. Cette fonction nous suffira pour la suite.

```
def check(hidden,code):
    mal = []
    result=[]
    for i in range(0, len(code)):
        mal.append(-1)
        result.append(-1)
    for i in range(0, len(code)):
        print("i=" + str(i) + " code=" +
str(code[i].id) + " hidden=" + str(hidden[i + 1]))
        if code[i].id == hidden[i + 1]:
            result[i] = 1
            print(result)
            if mal[i] > -1:
                result[mal[i]] = -1
            # remplace mal place par -1 dans
result a la position j
```

```
        else:
            for j in range(i + 1,
len(hidden)):
                print("i=" + str(i) + "
code=" + str(code[i].id) + " j=" + str(j) +
" hidden=" + str(hidden[j]))
                if code[i].id == hidden[j]:
                    result[i] = 2
                    mal[j - 1] = i
                    # liste contenant la
position du code et de hidden pour le pion
mal placé
                    # au cas ou il y
aurait un pion bien placé par la suite à
la même position de                 #
hidden
                    print(result)
    return result
```

3.6 Création de la table de l'historique

Afin de pouvoir retrouver le code, il est indispensable d'avoir l'historique des tentatives précédentes ainsi que le résultat vis-à-vis du code. L'historique des différents essais avec les résultats associés va être représenté sous forme de table dont chaque ligne correspondra à une tentative de code. Définissons arbitrairement un historique de 12 tentatives. En début de la partie, la table est vide. La table est représentée sous forme d'une liste dont chaque élément basique contient l'identifiant du pion et le résultat correspondant. Voici un extrait de la liste :

```
>>> table= [['pion _ b0OK','pion _
bNOK','pion _ bMPL'], ['pion _ b0OK','pion _
bNOK','pion _ bMPL']]
```

En début de programme, nous remplissons une table de cases vides comme suit :

```
Table=[]
centreXtab=135
table.append(0) #position actuelle dans la
table
for i in range(0,12):
    ligne=[]
    for i in range(0,len(code)):
        ligne.append(Actor('pion _ vide'))
    table.append(ligne)
```

Ensuite, implémentons l'affichage de la table dans la fonction draw() :

```
for i in range(1,len(table)):
    for j in range(0,len(code)):
        table[i][j].pos=((centreX+50*j),(cent
reYtab+50*(i-1)))
        table[i][j].draw()
```

À chaque vérification (appui sur le bouton check), nous transposerons le résultat, pions bien placés, pions mal placés ou pions non présents, à partir des listes que nous avons définies en début d'article : pionsOK, pionsNOK et pionsMPL.

```
def update _ table(result):
    global table,pionsMPL,pionsNOK,pionsOK,
code
    posligne=table[0]
    for i in range(0,len(code)):
        if result[i]==1:
            #bien place
            table[posligne+1][i].
image=pionsOK[code[i].id]
        elif result[i]==2:
            #mal place
            table[posligne+1][i].
image=pionsMPL[code[i].id]
        else:
            table[posligne+1][i].
image=pionsNOK[code[i].id]
    table[0]=+1
```

Appelons également notre fonction update_table() en fin de fonction update() pour mettre à jour l'historique.

Et voilà, le jeu est terminé ! La Figure 3 montre le résultat obtenu.

CONCLUSION

Dans cet article, outre le fait d'avoir découvert les bases de Pygame.zero qui n'est autre qu'une simplification de Pygame, l'intérêt était de pouvoir se focaliser sur l'application ou le jeu plutôt que sur les aspects techniques (boucle principale, gestion des évènements...). Vous avez également vu

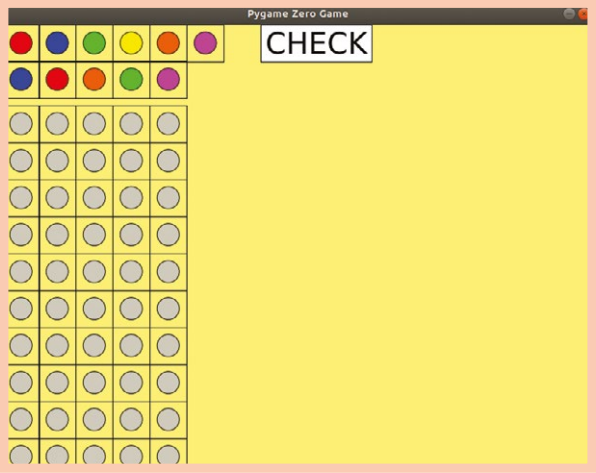


FIGURE 3. Écran du jeu terminé.

quelques étapes de construction d'un jeu. Certes, la structuration du jeu elle-même est réalisée dans un seul fichier principal, ce qui n'est pas la bonne méthode de programmation, mais ce n'était pas le but de cet article et cela pourra faire l'objet d'un article dédié. ■

RÉFÉRENCES

[1] Pygame.Zero, **https://pygame-zero.readthedocs.io/en/stable/**

[2] Laurent Delmas, « Découvrez le monde fabuleux d'OpenCV », *GNU/Linux Magazine HS n°96*, juin 2018.

[3] Tristan Colombo, « Créez un jeu en Python avec Pygame », *GNU/Linux Magazine n°117*, juin 2009.

[4] Tristan Colombo, « Animation et contrôle de sprites avec Pacman », *Linux Pratique n°99*, janvier 2017.

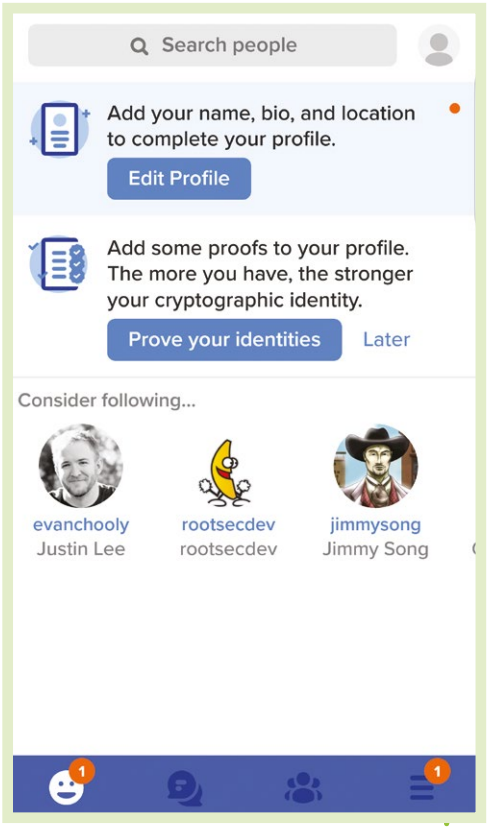
[5] Tristan Colombo, « Un peu d'organisation pour Pacman », *Linux Pratique n°100*, mars 2017.

[6] Tristan Colombo, « Il est rond comme un ballon, il est jaune comme citron c'est... un objet », *Linux Pratique n°100*, mars 2017.

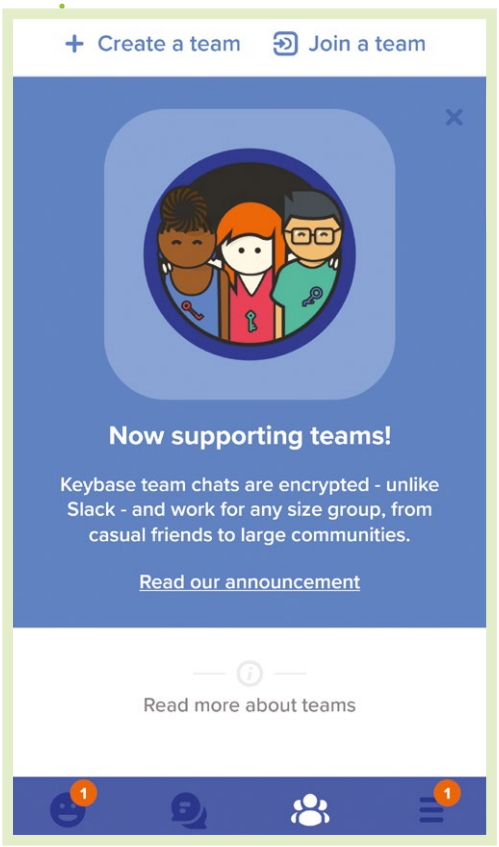
[7] Tristan Colombo, « Rafraîchir une fenêtre graphique Pygame », *GNU/Linux Magazine HS n°95*, mars 2018.

KEYBASE

PARTAGE, STOCKAGE ET ESPACES DE DISCUSSIONS CHIFFRÉS



Keybase est un service open source qui utilise GPG afin de vous permettre de chiffrer vos communications. Pour garantir l'authenticité de votre compte, Keybase vous invitera à vous authentifier à vos différentes identités numériques (comptes Twitter, GitHub, Reddit, Hacker News, clé PGP, etc.).



Parmi les fonctionnalités de ce projet se trouve une messagerie sécurisée qui vous permettra de débuter une conversation avec n'importe quel contact depuis son nom d'utilisateur Keybase, Twitter, Facebook, GitHub, Reddit ou Hacker News. Il n'est pas indispensable que votre interlocuteur dispose d'un compte Keybase, c'est l'une des grandes particularités de cet outil. Autre option pratique qui pourra plaire aussi bien aux particuliers qu'aux professionnels : la possibilité de créer ses propres espaces de discussion chiffrés où l'on pourra inviter par e-mail les membres de son choix. Il sera aussi possible d'ajouter vos dépôts Git chiffrés ou encore de stocker vos fichiers (les répertoires `private`, `public` et `team` ont d'ores et déjà été créés à cet effet). À noter que Keybase utilise son propre système de fichiers chiffré, KBFS. En bref, voilà là un vrai couteau suisse sécurisé avec des arguments intéressants qui mérite bien qu'on lui accorde un peu d'attention. ■

SITE : https://keybase.io/

VERSION TESTÉE : 3.1.2

QWANT JUNIOR

UN MOTEUR DE RECHERCHE POUR LES 6-12 ANS

Un moteur de recherche qui permette à vos enfants de découvrir toutes les ressources d'Internet en toute confiance ? Le français Qwant a travaillé sur le sujet en proposant une version de son moteur de recherche adaptée aux jeunes utilisateurs. Vous pourrez peut-être ainsi en faire le moteur de recherche par défaut sur votre ordinateur familial. Sur mobile, l'application lance un navigateur utilisant les outils open source de Mozilla.

Qwant Junior va bloquer par défaut tous les sites dont le contenu n'est pas adapté aux enfants. Par ailleurs, il fait un point d'honneur à respecter la vie privée de ses utilisateurs en ne collectant pas leurs données personnelles. Côté utilisation, l'enfant se trouve face à une page d'accueil conçue spécialement pour lui avec une sélection d'actualités adaptées et de jeux éducatifs. Les résultats peuvent être filtrés (***Web***, ***Actualités***, ***Éducation***, ***Images***, ***Vidéos***) et le moteur de recherche met en avant le contenu de l'encyclopédie Vikidia, destinée aux 8-13 ans. La fonctionnalité ***Carnets***, permet aux utilisateurs disposant d'un compte Qwant Junior (qui offre également la possibilité de sauvegarder toutes ses préférences du moteur de recherche) de partager des textes, images, vidéos et sites sur les thématiques de leur choix. N'hésitez pas à faire un rapide tour sur **https://www.qwantjunior.com/** pour vous faire un premier avis sur ce moteur de recherche ! ■

SITE : https://www.qwantjunior.com/

VERSION TESTÉE : 1.0.14

COMMUNIQUER sur Mastodon

TRIS ACATRINEI

À CÔTÉ DES RÉSEAUX SOCIAUX CENTRALISÉS, AU FONCTIONNEMENT ET À LA MODÉRATION PARFOIS ÉSOTÉRIQUE, UN PETIT VILLAGE NUMÉRIQUE RÉSISTE BRILLAMMENT À L'ENVAHISSEUR GAFAM ET IL S'APPELLE MASTODON.

Petit mammouth deviendra grand et le réseau social décentralisé qui a vu le jour en octobre 2016 continue à grandir et à fédérer. Nous allons vous faire un retour d'expérience de cette plateforme et vous donner quelques outils pour y faire vos premiers pas.

Un point de vocabulaire en premier lieu s'impose. Les termes utilisés ne sont pas tout à fait les mêmes que sur Twitter. Pour un message posté sur Mastodon, on parle de toot. Quand on partage un toot, on parle de retoot. Autre détail qui peut avoir son importance : vous n'avez pas de statistiques comme sur Twitter, pour évaluer la viralité de vos contenus. Comme nous le verrons, vous pourrez éventuellement le constater dans vos statistiques via Matomo. Entrons dans le vif du sujet.

1. MASTODON, POUR QUOI FAIRE ET POUR QUI ?

Que vous soyez à votre compte, une association ou tout simplement une personne qui a envie de faire connaître ses projets, surtout si ces derniers ont un lien avec le logiciel libre, vous inscrire sur

Mastodon peut être une bonne idée. Le public n'est pas du tout le même que sur Twitter, Facebook ou LinkedIn : moins nombreux, certes, mais plus qualitatif et pour utiliser des mots à la mode : plus engagé. Selon ce que vous faites, vous aurez de meilleurs retours que sur Twitter.

Autre point essentiel : le public y est globalement moins agressif que sur les réseaux sociaux plus connus. Je ne dis pas que vous n'y trouverez pas des trolls et des énervés, mais si on prend les choses avec hauteur, les discussions et les échanges sont plus agréables, moins épidermiques et cela tient au business model même de ce réseau social : il n'en a pas. L'objectif sur Mastodon n'est pas de vendre un produit quel qu'il soit, mais bien de permettre à des individus d'échanger. Nous ne sommes pas dans l'économie de l'attention et de l'émotion, mais dans un environnement collaboratif et j'oserais même dire, pacifique.

À ce stade de la lecture, vous l'avez déduit vous-même : inutile de vous inscrire si votre objectif est de pousser au clic pour vendre quelque chose : il n'y a pas de publicité, pas de posts sponsorisés ni même de statistiques. Par contre, si vous bloguez, que vous avez un projet libre, que vous êtes une association ou même une classe d'école primaire, vous y serez beaucoup mieux que sur Twitter, surtout si vous choisissez correctement l'instance sur laquelle vous allez « tooter ».

Quid des médias et des politiques ? Sur Twitter, chaque personnalité politique possède son compte, parfois contraint et forcé, car comme me le disent plus ou moins les députés eux-mêmes, sans Twitter

FIGURE 1. L'interface de paramétrage de profil sur Mastodon.

ni Facebook, point de salut. Quelques élus ont créé, non pas des comptes activement gérés, mais des robots qui repostent sur Mastodon, des contenus initialement envoyés sur Twitter. Certains grands médias comme Libération l'ont également fait, ainsi que ARTE, car ils ont compris qu'ils y avaient un lectorat potentiel. Ces comptes ont clairement indiqué être des bots avec la mention « [bot] » dans leur nom de profil, afin que les personnes comprennent qu'il n'y aura pas d'interaction directe, ce qui contribue aussi à calmer les esprits. Pour indiquer que votre compte est un robot, il suffit de vous rendre dans vos ***Paramètres > Modifier le profil*** et cocher la case ***Ceci est un robot***. Vous pouvez également vous inscrire dans l'annuaire de votre instance si vous le souhaitez.

Si vous êtes une « personnalité » sur Twitter ou que vous êtes clairement identifié comme appartenant à une entreprise, parce que vous faites des relations publiques ou que vous êtes journaliste, bref, que vous souhaitez vous aussi avoir un espace de liberté numérique, sans que cela vous porte préjudice, Mastodon peut s'avérer être une bonne option, car, tant que vous respectez les règles inhérentes à l'instance sur laquelle vous êtes inscrit, vous ne serez pas harcelé comme sur Twitter, à condition évidemment que votre pseudonyme sur Mastodon ne soit pas identique à celui que vous avez sur Twitter.

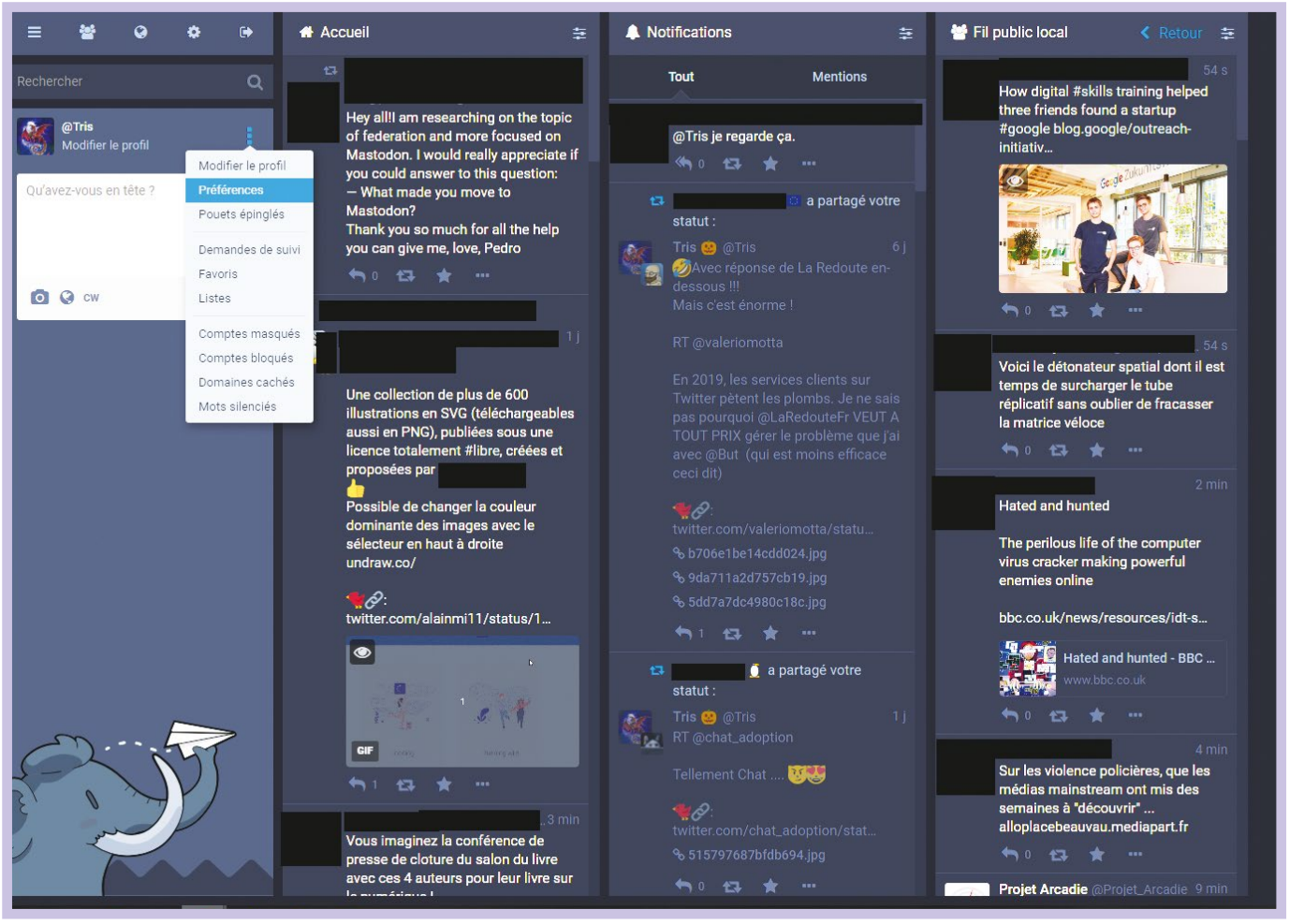


FIGURE 2. L'interface de Mastodon.

Enfin, si vous êtes freelance ou à la recherche d'un emploi, Mastodon peut être tout aussi efficace que Twitter, car il y a moins d'intermédiaires véreux et un hashtag dédié circule régulièrement avec des offres d'emploi, axées vers le logiciel libre.

2. JE TOOTE, TU TOOTES, NOUS TOOTONS

La force de Mastodon réside dans sa simplicité d'utilisation. Si vous êtes habitué à utiliser Twitter, vous ne serez pas perdu. Basiquement, quand vous vous connectez sur l'interface web, vous avez 4 colonnes. La première, la plus à gauche est la vôtre, avec un espace où écrire votre message, une barre de recherche et différentes icônes vous permettant de voir votre fil public local, un tutoriel et vos paramètres.

Sur la deuxième colonne, vous avez votre flux : les messages publics des gens auxquels vous vous êtes abonnés. Sur la troisième, vous avez vos notifications, vos mentions, mais aussi — contrairement à Twitter — vos messages directs privés. Sur l'interface web, ils sont d'un gris plus clair, ce qui vous permet de les identifier du premier coup d'œil.

Enfin, sur la quatrième colonne, vous avez le choix d'afficher ce que vous voulez : votre fil public local — c'est-à-dire les messages des personnes qui sont inscrites sur la même instance que la vôtre — ou le fil public global, qui affiche tous les messages de tous les utilisateurs de Mastodon. Personnellement, je préfère le fil public local, car le global est trop actif et dans des langues que je ne maîtrise pas comme le japonais.

Passons à la rédaction des messages. Contrairement à Twitter, vous pouvez poster des messages de 500 caractères. J'ai tendance à penser que c'est aussi l'un des facteurs apaisants de ce réseau social : on peut tourner les choses avec précaution et éviter les erreurs d'interprétation.

Vous pouvez aussi poster des photos ainsi que des GIF et des émoticônes, comme sur Twitter, mais il n'y a pas de bibliothèque de GIF comme sur Twitter, donc à vous d'importer les vôtres. Mais il y a un atout maître sur Mastodon : le *content warning* que j'apprécie beaucoup. Sur Twitter, vous voyez parfois des contenus un peu sensibles. Par exemple, il y a eu des violences policières pendant les manifestations des gilets jaunes et les photos des blessés ont été postées sur le réseau à l'oiseau bleu. Devant le déluge de photos, j'ai été contrainte de masquer le journaliste travaillant sur ce sujet, car je ne voulais pas voir ces images.

Sur Mastodon, le problème ne se pose pas grâce au *content warning*, abrégé en CW. C'est une façon urbaine de prévenir les autres utilisateurs que le message que vous avez posté peut heurter leurs sensibilités : photos gores, sexualité, viande, Windows, messages plus personnels, etc. Rien n'oblige les utilisateurs à utiliser le CW pour leur message, mais généralement, si quelqu'un envoie un message un peu limite sans utiliser le CW, ses abonnés ont tendance à le rappeler à l'ordre. Par ailleurs, vous pouvez donner un titre à votre CW, afin d'indiquer aux autres la nature du message. Si vous « cross-postez » entre Twitter et Mastodon, vous pouvez configurer un CW générique, afin de prévenir vos abonnés sur Mastodon qu'ils lisent un message initialement envoyé sur Twitter.

Autre point très fort de Mastodon : l'adaptabilité du degré de confidentialité. Sur Twitter, vous n'avez que trois options : laisser votre compte complètement ouvert à tous, le verrouiller pour n'échanger qu'avec vos abonnés et vos abonnements et les messages privés. Sur Mastodon, vous avez 4 niveaux de confidentialité : public, non-listé, abonnés uniquement, direct.

Le niveau public permet à votre message d'être affiché dans les

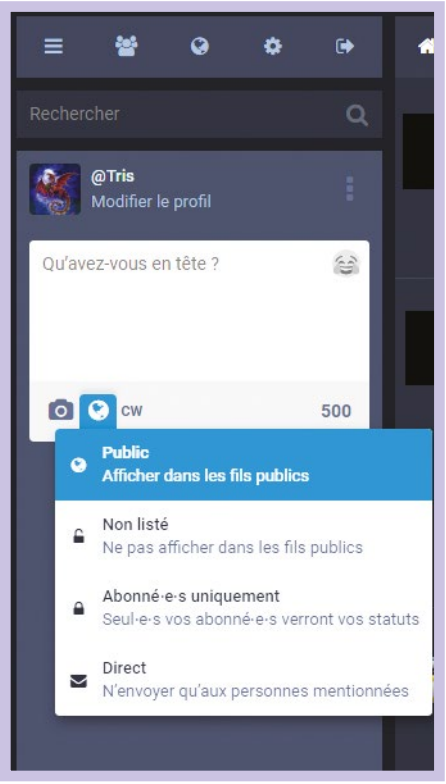


FIGURE 3. La confidentialité des messages sur Mastodon.

fils publics, local ou global. Il est symbolisé par une mappemonde. Le niveau non-listé, illustré par un cadenas ouvert, permet d'envoyer des messages à vos abonnés, mais ne sera pas intégré dans les fils publics. Il peut, par contre, être « retooté » contrairement aux messages réservés à vos abonnés. Ce dernier est symbolisé par un cadenas fermé. Enfin, les messages directs, illustrés par une enveloppe, sont les messages privés.

Attention à un point de détail : si vous lisez vos messages sur des applications mobiles, les messages directs ne seront pas grisés plus clairs comme sur l'interface web, donc prêtez attention aux icônes.

Passons aux images : cela fonctionne comme sur Twitter et vous pouvez ajouter une description à vos images pour les personnes malvoyantes. Vous pouvez également modifier l'aperçu en déplaçant le curseur sur l'endroit à partir duquel le focus de l'aperçu va se faire.

Comme sur Twitter, on peut épingler des messages ou utiliser des favoris et contrairement à Twitter, c'est parfaitement lisible et clair. En effet, depuis un moment, Twitter a remplacé la petite étoile, qui indiquait assez clairement qu'on mettait un message en marque-page ou pense-bête par un cœur et selon ce qu'on utilise pour le consulter, on voit défiler sur sa timeline, les messages « aimés » par les autres utilisateurs, ce qui brouille les cartes avec la fonction retweet.

Aucune confusion de ce type avec Mastodon : ce que vous mettez en étoile n'apparaît pas dans votre fil global et ce que vous retootez, apparaît. Avisez les trois points verticaux à côté de votre photo de profil sur la première colonne. Cliquez dessus et vous verrez apparaître ***Favoris***.

L'une des fonctionnalités que j'apprécie le plus sur Mastodon et qui n'existe pas en natif sur Twitter : effacer et réécrire. Rien de plus rageant sur Twitter que de se rendre compte qu'on a posté un message avec une faute d'orthographe ou de grammaire et de devoir l'effacer. Sur Mastodon, l'option est disponible.

Que faire des casse-pieds ? Là aussi, les choses sont mieux faites que sur Twitter, car dès sa création, le réseau social a intégré la possibilité de bloquer et de masquer des comptes, mais également de « silencier » des mots et de cacher intégralement des domaines, c'est-à-dire des instances. Aucun risque donc de subir des live-tweets d'émissions de téléréalité ou des messages de militants politiques, si cela ne vous inspire pas.

Seule limitation sur Mastodon : ce qu'on appelle le subtweet, c'est-à-dire, partager un tweet en le commentant. Sur Mastodon, cette option n'est pas encore disponible, mais il est possible que ce soit aussi une volonté des développeurs pour ne pas créer une atmosphère anxiogène.

Pour le reste, si vous avez déjà une grande habitude de Twitter, vous ne serez pas dépaysé.

3. PROPRIÉTAIRE OU LOCATAIRE ?

On l'a dit : Mastodon est un réseau social décentralisé. Pour vous y inscrire, vous devrez choisir une instance sur laquelle créer votre compte ou créer votre propre instance. On peut utiliser l'analogie du bien immobilier : être locataire d'une instance ou propriétaire. Commençons par balayer les deux options.

Si vous vous inscrivez sur une instance déjà existante, vous n'aurez pas à gérer les mises à jour, l'administration, ni la modération. Vous serez un utilisateur comme un autre, qui devra respecter les

ACTUELLEMENT DISPONIBLE

GNU/LINUX MAGAZINE HORS-SÉRIE N°102

NOUVEAU ! ACHETEZ-LE DÈS MAINTENANT SUR IOS ET ANDROID :

Disponible en App Store

DISPONIBLE SUR Google Play

NE LE MANQUEZ PAS

CHEZ VOTRE MARCHAND DE JOURNAUX ET SUR :

https://www.ed-diamond.com

règles de l'instance. Généralement, elles se trouvent dans ***About***. Sur certaines instances, la publicité est interdite, sur d'autres, ce sont les contenus à caractère sexuel, le spam ou encore les liens des contenus illégaux. Vous devrez également veiller à vous inscrire sur une instance où les inscriptions sont encore ouvertes. En effet, la plupart des instances sont gérées par des particuliers donc le nombre de membres est limité.

Autre point à ne pas négliger : la taille. Si vous êtes sur une instance très modeste, vous risquez d'être confronté à des indisponibilités de serveurs ou à des mises à jour non déployées. Par ailleurs, cela reste un réseau social donc le but est bien de se créer un nouveau réseau. On ne doit cependant pas oublier que Mastodon est un réseau social décentralisé donc le but n'est pas de concentrer tous les utilisateurs sur une même instance.

Comment s'y retrouver ? Il existe une plateforme qui recense toutes les instances Mastodon, avec le nombre d'utilisateurs, l'état des inscriptions — ouvertes ou fermées — et les versions utilisées. Vous pouvez également choisir la langue utilisée — ce qui peut être très important en cas de difficultés — la taille de l'instance et les règles de modération applicables. Partant de là, vous avez le choix.

Détail amusant : depuis que Tumblr a banni la pornographie sur sa plateforme, des instances spécialement dédiées aux contenus pour adultes ont vu le jour. L'avantage est que la couleur est clairement affichée.

Mais, tout comme lorsque l'on est locataire, si on ne respecte pas les règles, on peut être banni de l'instance que l'on a choisie, du jour au lendemain. Rassurons le lecteur : dans l'ensemble, la modération est moins fantaisiste que sur Twitter et la majorité des administrateurs est assez intelligente pour distinguer une parole malheureuse dans une nuée de toots d'un comportement réellement déviant.

Si vous souhaitez être propriétaire de votre instance, vous le pouvez. Si vous êtes une petite structure professionnelle ou associative, sans administrateur système à temps complet sous la main, vous pouvez recourir à un service qui se charge pour vous de créer et d'administrer votre instance Mastodon. Vous vous en doutez : ce service est payant, mais cela reste assez abordable. L'option la moins onéreuse est à 7 € par mois et la plus chère, à 35 €. C'est un service professionnel, qui passe par OVH et géré par une équipe française, ce qui n'est pas plus mal.

Enfin, il vous reste une option : tout construire vous-même *from scratch* et là, attention. Vous aurez de toute façon besoin d'un serveur, que vous serez amené à payer, sauf si votre structure a déjà un serveur vacant qui traîne dans un coin. Vous trouverez plusieurs tutoriels vous proposant l'installation sur un VPS Ubuntu, Debian et via Docker. Vous êtes donc assez libres de vos mouvements, mais n'hésitez surtout pas à lire les retours d'expériences des autres utilisateurs, notamment en ce qui concerne les mises à jour du logiciel à proprement parler. Même pour les plus aguerris à l'administration système, cela ne se passe pas toujours sans heurts. À moins que cela soit vraiment votre métier ou que vous avez envie de vous faire la main, privilégiez la situation de « locataire » ou de « propriétaire » avec prise en charge d'un service dédié. N'étant pas moi-même administrateur système, je me garderais bien de prétendre vous fournir un tutoriel clef en main, je vous invite cependant à consulter l'article **[1]** précédemment rédigé sur le sujet dans *Linux Pratique*. Je me suis inscrite sur l'instance gérée par La Quadrature Du Net (LQDN), ce qui m'amène à vous alerter sur un point.

La plupart des instances sont gérées par des personnes de façon bénévole ou par des structures associatives. Si vous faites le choix de vous y inscrire, essayez, à la mesure de vos moyens, de soutenir financièrement ces structures, afin qu'elles puissent aussi pérenniser ce type de services.

Que vous soyez locataire ou propriétaire, vous restez maître de vos données. Ainsi, dans vos

paramètres de compte, vous pouvez importer des données, mais aussi les exporter.

Vous êtes sur Twitter, Facebook, LinkedIn et vous devez vous demander pourquoi vous devriez vous inscrire sur Mastodon et gérer un autre réseau social ? La meilleure façon de vous convaincre est de vous raconter comment et pourquoi je m'y suis inscrite.

4. COMMENT J'AI ATTERRI SUR MASTODON

Un soir où je regardais une chaîne d'information en continu, j'ai tweeté sur le hashtag de l'émission, ce qui a amené un troll bien velu dans mes mentions. Au lieu de le masquer, je lui ai répondu assez vertement et assez violemment. Disons que j'étais prête à faire un remake de tous les Saw avec son humble personne. Léger problème : le tweet a été signalé et mon compte a été définitivement suspendu, malgré une procédure « d'appel ». La plupart des gens que je suivais sur Twitter s'étaient inscrits en masse sur Mastodon, je m'y suis donc inscrite et pour les autres, je les ai récupérés sur mon compte Twitter professionnel, à savoir Projet Arcadie.

Après quelques péripéties, j'ai récupéré un compte Twitter personnel, mais j'ai gardé Mastodon, car l'ambiance y est plus chaleureuse, moins agressive et beaucoup plus conviviale. Certains utilisateurs m'ont demandé de faire un compte pour le Projet Arcadie, ce qui fut fait dans le courant de l'année 2018. Sur Twitter, j'ai beaucoup plus de followers que sur Mastodon, on pourrait presque se dire que c'est anecdotique.

Là où les utilisateurs de Twitter peuvent être assez passifs, sauf quand il s'agit de déclencher des torrents de boue, les utilisateurs de Mastodon sont beaucoup plus réactifs. Dans le cadre du Projet Arcadie, qui n'est financé que par des dons de particuliers, c'est en grande partie de Mastodon que viennent mes donateurs, surtout lorsque certains mois sont plus difficiles que d'autres.

J'ai également regardé mes statistiques de fréquentation des trois plateformes : les parlementaires, les partis politiques, le blog. Si Mastodon n'est pas classé dans la section « réseaux sociaux » sur Matomo (anciennement Piwik), il est indexé dans la section « sites Web » et on voit que Mastodon se détache clairement. Sur certaines périodes, Mastodon m'a apporté plus de trafic sur Projet Arcadie qu'un article dans *Le Figaro*. Que le lecteur ne se méprenne pas : il ne va pas forcément faire exploser ses statistiques en s'inscrivant sur Mastodon, mais cela peut être un bon complément et apporter un autre public, notamment celui qui a déserté Twitter parce que l'ambiance est y de plus en plus lourde.

Si je vous ai convaincu de vous inscrire, vous devez vous demander comment gérer ce réseau social aussi facilement que les autres. Rassurez-vous : il y a des outils.

5. DES OUTILS POUR MAMMOUTH

Ce qui m'a amené le plus de followers sur Twitter – sur mon compte professionnel – est mon suivi des séances à l'Assemblée nationale et plusieurs personnes m'avaient demandé que ce live-tweet soit également disponible sur Mastodon. Sur les séances peu productives – appelons-les ainsi – faire des copier-coller est faisable et ça occupe quand les discussions générales sont à rallonge. Mais comment faire quand les tweets s'enchaînent à vitesse grand V ?

Un outil a été créé pour connecter à la fois votre compte Twitter et votre compte Mastodon, vous permettant de poster en simultané un même message sur les deux plateformes. Seul détail qui peut rebuter certaines personnes : l'application a besoin d'avoir une autorisation sur vos deux comptes.

L'interface est très simple à gérer. Vous connectez les deux comptes et ensuite, vous pouvez paramétrer les éléments dont vous

souhaitez qu'ils soient automatiquement envoyés sur les deux réseaux sociaux. Vous pouvez publier vos tweets sur Mastodon et à l'inverse, publier vos toots sur Twitter. Chaque onglet vous permet de personnaliser les envois, par exemple, les réponses aux tweets ou aux toots, selon les cas.

Sur ce point, il y a un détail à relever. Si vous faites ce que l'on appelle un *thread*, à savoir un enchaînement de tweets sur un sujet et que vous n'avez sélectionné l'option de réponse ***Publier les réponses à mes propres tweets sur Mastodon (fils de discussion)***, seul le premier tweet apparaîtra sur Mastodon, mais pas le reste.

Si vous « cross-postez » entre Twitter et Mastodon, vous pouvez configurer un CW générique, afin de prévenir vos abonnés sur Mastodon qu'ils lisent un message initialement envoyé sur Twitter.

Veillez également à regarder vos notifications sur Mastodon si vous prenez le pli de cross-poster, car il serait dommage de snober accidentellement des utilisateurs. Enfin, il sera explicitement indiqué sur les messages qu'il s'agit d'un cross-post et même chose si vous utilisez une application mobile.

Si vous êtes accro à votre smartphone, vous pouvez utiliser Tusky comme application mobile ou Subway Tooter. À une époque, les photos mettaient du temps à charger via Tusky, j'ai donc utilisé les deux applications et les deux se valent. L'utilisateur fera son choix en fonction de ses propres critères.

Si vous avez recours à Tweetdeck pour programmer vos tweets, vous ne rencontrerez aucune difficulté : le cross-post se passe sans difficulté. Par contre, je déconseille de cross-poster entre Twitter, Facebook, LinkedIn et Mastodon. Si Twitter et Mastodon ont des similitudes et répondent tous les deux à la définition du microblogging, Facebook et LinkedIn sont trop différents et il vaut mieux calibrer ses messages spécifiquement pour ces deux réseaux sociaux.

Qu'en est-il des métadonnées ? Le balisage sémantique d'OpenGraph fonctionne parfaitement sur Mastodon. Par exemple, si vous postez un article et que vous avez correctement renseigné tous les champs pour qu'un aperçu de l'article apparaisse sur Twitter, il apparaitra également sur Mastodon.

Qu'en est-il du partage de contenu ? En effet, sur la plupart des sites, on peut partager un contenu sur Facebook, Twitter, LinkedIn ou autres. Mais pour le moment, il n'y a pas forcément de bouton de partage partout ni de module qui intègrerait cette fonctionnalité même si on peut la créer avec quelques lignes de PHP. Il existe une extension de navigateur : Mastodon Share, disponible pour Firefox et pour Chrome.

CONCLUSION

Mastodon est là pour durer et c'est tout le mal qu'on lui souhaite, ne serait-ce parce qu'on a besoin d'un réseau social qui nous rappelle la glorieuse époque où les gens discutaient sur le Web, sans être guidés uniquement par une volonté de rabaisser les autres ni leur mettre sous le nez n'importe quel produit. ■

LA BOÎTE À OUTILS

[1] **https://connect.ed-diamond.com/Linux-Pratique/LP-102/Oubliez-les-gazouillis-proprietaires-et-pouetez-en-toute-liberte-avec-Mastodon**

- Le catalogue des instances : **https://instances.social/**
- Héberger son instance, clef en main : **https://masto.host/**
- Cross-poster entre Twitter et Mastodon : **https://crossposter.masto.donte.com.br/**
- Mastodon Share : **https://addons.mozilla.org/fr/firefox/addon/mastodon-share/** ou **https://chrome.google.com/webstore/detail/mastodon-share/ngkommdldcakheaeoafgakbbiinkohom**
- Tusky : **https://framalibre.org/content/tusky**
- Subway Tooter : **https://play.google.com/store/apps/details?id=jp.juggler.subwaytooter&hl=fr**


Anaïs a le pouvoir
de détecter
les mines avant
qu'elles n'explosent
pour 2,50 €/mois*
Grâce au don mensuel
on a tous le pouvoir de changer
le monde, même à distance !
hi.fr/don.mensuel
Anaïs C., donatrice régulière pour handicap international depuis 2010
handicap
international
*Après réduction fiscale.
Pamplemousse.com - Crédit photo : M.E. Brouet

PROPULSEZ-VOUS DANS LA CYBER DÉFENSE !

À Rennes,

DES POSTES D'INGÉNIEURS EN CDI (H/F) SONT OUVERTS

- ♦ Responsable Datacenter
- ♦ Architecte des systèmes d'information
- ♦ Intégrateur d'infrastructures

#SDS #DEVOPS #ADMIN #VIRTUALISATION #BIG DATA #HPC #SDN

DGA Maîtrise de l'information
POSTULEZ !

www.defense.gouv.fr/dga

Retrouvez-nous sur LinkedIn et APEC

DGA

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

MINISTÈRE DES ARMÉES