



GNU LINUX

PRATIQUE

FRANCE MÉTRO. : 7,90 €
BELUX : 8,90 € CH : 13,40 CHF
ESP/IT/PORT-CONT : 8,90 €
DOM/S : 8,90 €
TUN : 20 TND MAR : 98 MAD
CAN : 14,99 \$CAD

L 18864 - 124 - F: 7,90 € - RD



LIGNE DE COMMANDES

Trucs et astuces pour optimiser sa configuration de Bash p. 36

OPEN DATA

Petit manuel de survie dans le milieu de la donnée ouverte p. 74

SYSTÈME / AUTOMATISATION

MAÎTRISEZ LE DEVOPS PAR LA PRATIQUE !

- Créez vos environnements de tests avec Vagrant
- Encapsulez les images des machines avec Packer
- Gérez vos conteneurs avec Docker...

p. 44



TUTORIELS

STOCKAGE

Générez et manipulez des images ISO p. 12

GIT / INTERFACE

Gérez vos projets en mode graphique avec GitKraken p. 62

BASE DE DONNÉES / BIG DATA

Créez vos premiers clusters avec Elasticsearch p. 22

INTÉGRATION CONTINUE

Déployez Jenkins CI avec Docker p. 56



JOYEUX ANNIVERSAIRE WIKIPÉDIA !

Retrouvez-nous pour célébrer un des plus beaux
projets collaboratifs du web d'aujourd'hui !

Rendez-vous à partir du 1er mars 2021 pour plus
d'informations sur www.wikimedia.fr/wp20

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 : www.linux-pratique.com
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 : <https://stock.adobe.com/fr>

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 : À parution

N° ISSN : 0183-0872

Commission Paritaire : K78 990

Périodicité : Bimestrielle

Prix de vente : 7,90 Euros

La rédaction n'est pas responsable des textes, illustrations et photos qui lui sont communiqués par leurs auteurs. La reproduction totale ou partielle des articles publiés dans Linux Pratique est interdite sans accord écrit de la société Les Éditions Diamond. Sauf accord particulier, les manuscrits, photos et dessins adressés à Linux Pratique, publiés ou non, ne sont ni rendus, ni renvoyés. Les indications de prix et d'adresses figurant dans les pages rédactionnelles sont données à titre d'information, sans aucun but publicitaire. Toutes les marques citées dans ce numéro sont déposées par leur propriétaire respectif. Tous les logos représentés dans le magazine sont la propriété de leur ayant droit respectif. Les articles non signés contenus dans ce numéro ont été rédigés par les membres de l'équipe rédactionnelle des Éditions Diamond.



ENCART CONNECT ENCARTÉ DANS LA COUVERTURE

Retrouvez-nous sur :



@linuxpratique - @editionsdiamond



@linuxpratique



www.linux-pratique.com



Le saviez-vous ? En ce début d'année, nous avons fêté la quinzième édition de la Journée européenne de la protection des données. Cette initiative est née au sein du Conseil de l'Europe dans le cadre de la « Convention pour la protection des personnes à l'égard du traitement automatisé des données à caractère personnel », texte signé à ce jour par une quarantaine de pays. L'objectif suivi est de sensibiliser les citoyens de ces pays à la protection de leurs données et les informer des droits existants en la matière.

Cette journée, fixée le 28 janvier de chaque année est donc l'occasion d'organiser des événements en Europe et au-delà, pour éduquer tout un chacun aux enjeux liés à la problématique de la protection des données dans un monde où les questions relatives à leur traitement et leur exploitation font régulièrement l'actualité. Vous n'aurez sans doute pas manqué l'épisode des nouvelles conditions de services de WhatsApp de ce début d'année. Cet événement ne sera sans doute que le premier d'une série qui risque de ponctuer 2021.

À cette date, le prix « protection de la vie privée » 2020 a été remis par la CNIL et l'INRIA à une équipe de recherche pour son article « Facebook Ads Monitor: An Independent Auditing System for Political Ads on Facebook »*. Cette récompense européenne vise à attirer l'attention de la communauté scientifique aux questions liées à la protection des données et à encourager les travaux de recherche en ce sens. L'article primé a permis de mettre en exergue la problématique de la réglementation de la publicité politique sur le réseau social et alerter ainsi administrations et organisations sur le sujet.

Ceci aura donc pour avantage de mettre en lumière un champ d'investigation parmi tant d'autres autour du vaste débat concernant les données. Nous soulèverons dans ce numéro un autre sujet de réflexion autour de ces dernières avec l'open data.

L'article phare de ce numéro de printemps s'intéressera quant à lui au DevOps, qui risque lui aussi de faire couler encore beaucoup d'encre cette année. Dans un contexte de transformation digitale accélérée avec des besoins d'automatisation importants, cette approche a en effet de beaux jours devant elle. Gageons que cet aperçu des possibilités qui s'offrent à vous en la matière vous permettra de mieux appréhender des solutions de plus en plus plébiscitées.

ALINE HOF

* pour les curieux, celui-ci est accessible ici : <https://hal.archives-ouvertes.fr/hal-03048337/>



www.ed-diamond.com

OFFRES D'ABONNEMENTS | ANCIENS NUMÉROS | HORS-SÉRIES | BASE DOCUMENTAIRE TECHNIQUE

Sommaire

Linux Pratique n°124

ACTUALITÉS & NOUVEAUTÉS

06 BRÈVES

SYSTÈME & CONFIGURATION

12 GÉNÉRER ET MANIPULER DES IMAGES ISO

Quand bien même leur usage s'est quelque peu atténué au profit d'autres supports de stockage de masse depuis plusieurs années, les CD-ROM et autres DVD-ROM ont toujours une utilité en 2020 : stockage de données, création de disques bootables (pour distributions GNU/Linux entre autres), disques multimédias...

22 CRÉEZ VOS CLUSTERS AVEC ELASTICSEARCH

Vous êtes-vous déjà demandé comment Facebook et Google gèrent leurs immenses masses de données tout en fournissant des services rapides ? Quelle serait votre solution pour gérer une base de données avec un trafic de données abondant ?

SHELL & SCRIPTS

36 OPTIMISER SA CONFIGURATION DE BASH

Nouveau sur bash, vous cherchez à apprendre de nouvelles astuces ? Utilisateur confirmé, vous voulez optimiser encore plus votre configuration ? Cet article est pour vous.



Intégration continue



56

Faciliter le déploiement de Jenkins CI avec Docker



36

Trucs et astuces pour optimiser son utilisation de Bash

Shell

Git



62

Tirer parti d'une interface graphique pour réaliser toutes ses opérations Git

DEVOPS & ORGANISATION

44 LE DEVOPS EN PRATIQUE

Le DevOps a été adopté par la plupart des entreprises ayant comme activité principale ou annexe, la digitalisation de leur système d'information pour faire face au défi que représente l'essor des technologies numériques. Dans cet article, nous ferons un premier tour d'horizon de la partie outillage du DevOps...

56 DÉPLOYER JENKINS CI AVEC DOCKER

Solution appréciée d'intégration continue, Jenkins est un puissant logiciel Java dont l'installation et l'exécution sur un système ne sont pas sans conséquence et à prendre à la légère. Afin de faciliter son déploiement et isoler proprement ce programme du reste de la machine qui l'héberge, nous allons illustrer ici comment le mettre en place, en seulement quelques commandes, à l'aide de Docker...

62 GÉREZ VOS PROJETS GIT EN MODE GRAPHIQUE AVEC GITKRAKEN

Certains de vos collaborateurs ne sont peut-être pas à l'aise avec la ligne de commandes, ne veulent tout simplement pas l'utiliser ou encore ne veulent pas apprendre la syntaxe des commandes Git. GitKraken est fait pour eux. Il leur permettra de réaliser toutes les opérations Git à l'aide d'une interface graphique...

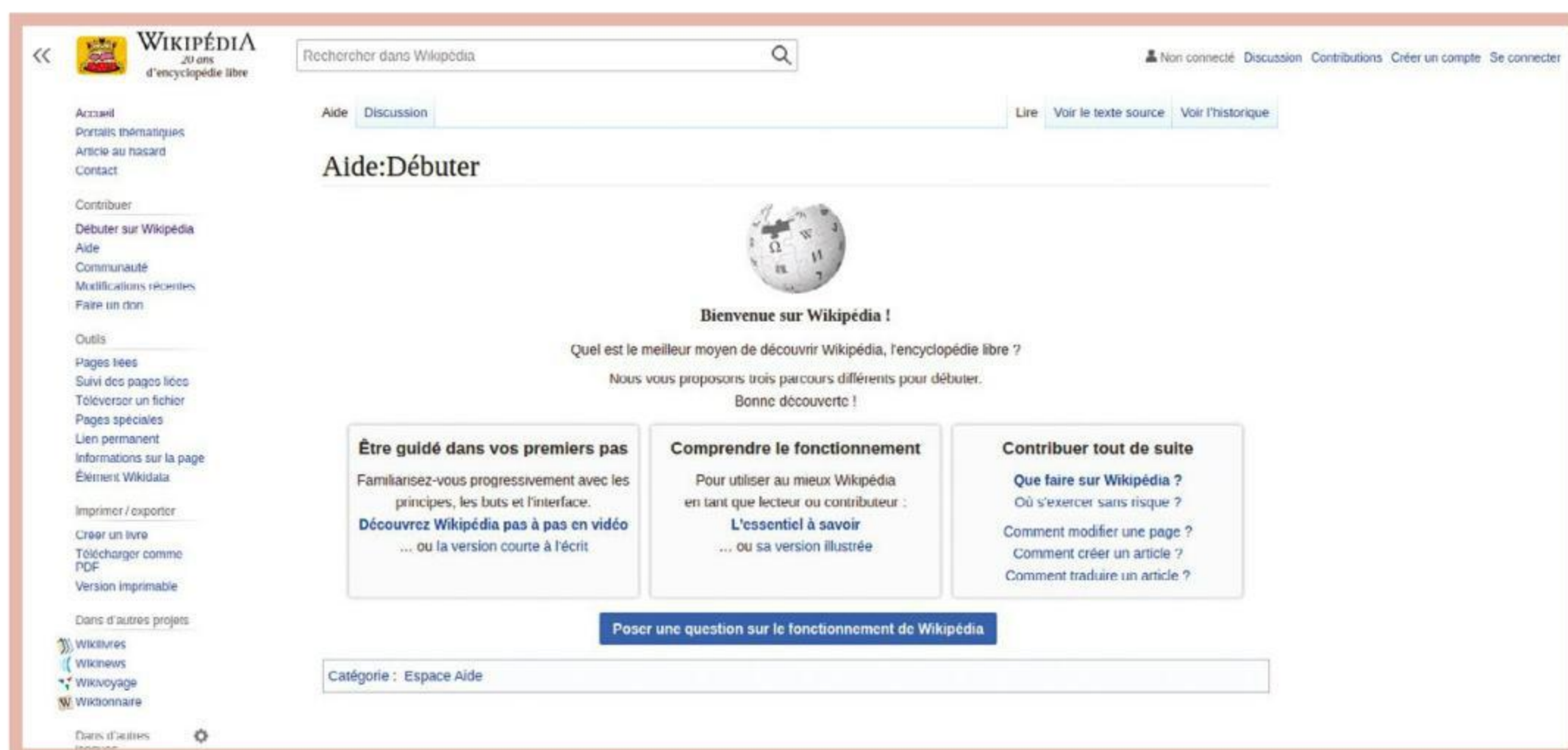
RÉFLEXION & SOCIÉTÉ

74 À LA DÉCOUVERTE DE L'OPEN DATA : PETIT MANUEL DE SURVIE DANS LE MILIEU DE LA DONNÉE

La remise au Premier ministre du rapport Bothorel sur la politique publique de la donnée, des algorithmes et des codes sources permet de procéder à un état des lieux de l'open data en France. De quoi parle-t-on quand on parle d'open data ? Quels sont les critères et les formats ? Comment une structure peut-elle se lancer dans l'open data ?

» 27 ABONNEMENTS

WIKIPÉDIA FÊTE SES 20 ANS



L'encyclopédie collaborative en ligne, libre (licence GFDL, puis CC BY-SA 3.0 depuis 2009) et gratuite, la plus connue au monde a fêté ses 20 ans le 15 janvier dernier. Pour la petite histoire, la version française de l'encyclopédie n'arrivera que deux mois après son lancement en anglais. Elle accueille aujourd'hui plus de 300 langues. À l'heure où nous rédigeons ces lignes, notre Wikipédia « français » compte près de 2 300 000 articles.

En moyenne, plus de 280 000 personnes contribuent à Wikipédia tous les mois à l'échelle mondiale. À l'occasion de l'anniversaire du projet, la fondation Wikimedia qui le chapeaute depuis 2003, a souhaité mettre à l'honneur ses contributeurs. On pourra ainsi découvrir le profil de certains d'entre eux sur <https://wikimediafoundation.org/wikipedia20/>. ■

L'ÉDITION 2021 DE L'AFUP DAY

La prochaine journée de conférences dédiée à PHP aura lieu le 28 mai prochain. Pour l'occasion, une quarantaine de conférences seront organisées à Lille, Rennes, Toulouse ainsi qu'à Tours. Au moment où nous rédigeons ces lignes, il est difficile de prévoir si elles auront bien lieu sur place ou en ligne. Le programme pour chacune des villes citées est en tout cas d'ores et déjà disponible. Il y sera notamment question de migration vers K8S, de gestion d'une architecture technique dans un environnement agile, des challenges de HTTP/3 ou encore d'architecture hexagonale...

Toutes les informations liées à cet événement sont disponibles sur <https://event.afup.org/> (billets, programmes, lieux, etc.). ■

UN NOUVEL INDICE DE RÉPARABILITÉ

Depuis le 1er janvier de cette année, un nouvel élément est apposé aux côtés des équipements électriques et électroniques en application de la loi n° 2020-105 du 10 février 2020 relative à la lutte contre le gaspillage et à l'économie circulaire : l'indice de réparabilité. Si vous cherchez à acquérir un nouvel ordinateur portable ou un smartphone (pour information, cela vaut aussi pour les lave-linges, téléviseurs et tondeuses à gazon), vous devriez ainsi voir ce score (une note de 1 à 10) affiché dans les spécifications du produit consulté.



Le but du gouvernement est d'atteindre un taux de 60 % de réparation des produits électriques et électroniques dans les 5 ans. Autre objectif pour le 1er janvier 2024 : proposer un indice de durabilité qui prendra en compte de nouveaux critères tels que la robustesse ou la fiabilité du produit. ■

DES RECOMMANDATIONS À DESTINATION DES SI À INFORMATIONS SENSIBLES OU DIFFUSION RESTREINTE

L'ANSSI a publié récemment un guide visant à fournir des indications techniques pour faciliter la mise en place d'architectures qui hébergent des informations sensibles ou à diffusion restreinte, relevant de l'II 901, une instruction ministérielle de janvier 2015 qui définit les objectifs et mesures de sécurité relatifs à ce type d'informations. Ce guide, destiné aux administrateurs système, DSI et RSSI, s'adresse aussi bien aux entités publiques que privées visées par ce contexte.

Le document, disponible à <https://www.ssi.gouv.fr/entreprise/guide/recommandations-pour-les-architectures-des-systemes-dinformation-sensibles-ou-diffusion-restreinte/>, ne couvre pas le sujet en cas d'hébergement des données dans un cloud. Y seront présentées les diverses architectures acceptables dans le cadre cité, les recommandations à suivre pour administrer un SI sensible, comment interconnecter celui-ci avec un autre SI, des bonnes pratiques pour la sécurisation interne des SI sensibles, etc.

UN FORUM INTERNATIONAL DE LA CYBERSÉCURITÉ REPORTÉ À JUIN

À l'heure où nous rédigeons ces lignes, l'édition 2021 du FIC a pour le moment été décalée aux 8, 9 et 10 juin 2021 prochains (à confirmer en fonction des dispositions liées à la crise sanitaire, une date de secours a également été réservée le 31 août, 1er et 2 septembre). Ce rendez-vous incontournable dédié à la cybersécurité en Europe, qui accueille en moyenne plus de 10 000 participants, est l'occasion d'assister à des tables rondes, conférences, démonstrations techniques et masterclass organisés pour l'occasion. Diverses thématiques y seront traitées sous l'angle de la sécurité : IoT, travail à distance, cloud, e-santé, cybercriminalité, automobile, etc.

Le FIC permet également d'échanger avec des professionnels du milieu dans son salon dédié (vous devriez d'ailleurs y trouver un stand dédié au bimestriel de la cybersécurité offensive et défensive, MISC). Pour en savoir plus sur cet événement, qui sera placé sous le thème « Pour une cybersécurité coopérative et collaborative », rendez-vous sur <https://www.forum-fic.com/accueil.htm>. ■

BYE BYE CENTOS, BIENVENUE ROCKY LINUX

Ce n'est plus un secret pour personne : en décembre dernier, Red Hat annonçait consacrer ses futurs investissements dans la version Stream de CentOS en tant que plateforme de « pré-test » de Red Hat Enterprise Linux et non plus dans CentOS Linux, utilisable en production. Le support de CentOS 8 sera arrêté fin décembre 2021 (alors qu'il aurait dû être maintenu jusqu'au 31 mai 2029). Autant dire que cette décision a fait énormément parler d'elle, CentOS étant largement adopté par bon nombre d'entreprises et d'organisations.

Greg Kurtzer, co-fondateur de la distribution CentOS a rapidement annoncé le lancement d'un fork du projet : Rocky Linux dont on peut suivre le développement ici : <https://rockylinux.org/fr/>. Il y a de fortes chances que d'autres dérivés risquent de se profiler à l'horizon dans les prochaines semaines/mois... ■

LES JOURNÉES DU LOGICIEL LIBRE

Organisées depuis 1998, une nouvelle édition des Journées du Logiciel Libre devrait se tenir à Lyon les 3 et 4 avril 2021 prochains (date à confirmer en fonction du contexte actuel, à l'heure où nous rédigeons ces lignes celle-ci est toujours maintenue). L'événement accueille durant tout un week-end des acteurs et passionnés du monde du Libre. Pendant ces deux jours, on pourra profiter de conférences, d'ateliers, de démonstrations. Un village associatif sera présent et des install parties seront également organisées.

La thématique retenue pour cette édition 2021 est la suivante : « Utopies concrètes et accessibles ». Le programme, une fois disponible, pourra être consulté sur <https://www.jdll.org/programme>. L'événement aura lieu de 10 à 18h à la Maison Pour Tous - Salle des Rancy, située à Lyon 3. Un tour sur <https://www.jdll.org/> vous permettra de préparer votre venue. ■

LE KIT PÉDAGOGIQUE DU CITOYEN NUMÉRIQUE

Fruit du travail de la CNIL, du CSA, du Défenseur des droits et de l'Hadopi, le kit pédagogique du citoyen numérique offre diverses ressources permettant de répondre à de nombreuses questions liées à l'usage du numérique. Il couvre principalement quatre grandes thématiques : les droits sur Internet, l'utilisation des écrans, le respect de la création et la protection de la vie privée en ligne. Le but est d'accompagner les plus jeunes dans l'apprentissage de leur citoyenneté numérique.

Les ressources ainsi mises à disposition seront aussi bien utiles aux parents qu'aux enseignants et formateurs afin d'avoir en main les réponses à apporter aux enjeux soulevés par l'usage d'Internet, mais aussi et surtout, préserver et éduquer les plus jeunes face aux risques en ligne. Tous ces éléments sont réunis sur <https://www.educnum.fr/kit-pedagogique-du-citoyen-numerique-retrouvez-toutes-les-ressources>.



PROPOSER UN CATALOGUE DE SOLUTIONS NUMÉRIQUES À DESTINATION DES ADMINISTRATIONS

La direction interministérielle du numérique (DINUM) entend pouvoir proposer un catalogue de solutions numériques à destination des administrations afin de faciliter la création de services publics numériques ou encore favoriser la mutualisation d'outils entre les entités publiques.

Dans l'attente de la publication de ce document, tous ceux qui disposent d'un outil pouvant être utile peuvent d'ores et déjà le faire référencer en se rendant sur <https://numerique.gouv.fr/services/identifier-et-utiliser-des-outils-performants-pour-les-services-publics-numeriques/>.

Chaque solution proposée pourra être auto-évaluée selon plusieurs critères parmi lesquels sa licence, sa conception, son traitement des données, sa prise en compte de la sécurité, sa conformité par rapport aux exigences du secteur public, etc. Les acteurs publics à la recherche de solutions peuvent également se rendre sur cette page pour être tenus au courant de l'ouverture du catalogue. ■

DÉCOUVREZ LA NOUVELLE VERSION !



CONNECT

LA DOCUMENTATION TECHNIQUE DES PROS DE L'IT

LES 6 NOUVEAUTÉS :

1

Nouvelle
ergonomie

2

Articles premiums
jamais publiés

3

Parcours thématiques
de la rédaction

4

Listes de lecture
personnalisées

5

Création d'alertes
de publication

6

Moteur de
recherche amélioré

ET TOUJOURS...

- ✓ Plus de 1770 articles disponibles
- ✓ Plus de 80 nouveaux articles chaque année
- ✓ Magazines et hors-séries d'hier et d'aujourd'hui
- ✓ Accès illimité 24h/24 - 7j/7
- ✓ Abonnements multi-lecteurs



CONTACTEZ-NOUS :

par téléphone :
03 67 10 00 28

par e-mail :
connect@ed-diamond.com



LINUX PRATIQUE



Lecture
en avant-première
des magazines en ligne !

LA DOCUMENTATION TECHNIQUE DES PROS DE L'IT

OUI, JE M'ABONNE

Abonnement à retourner avec votre règlement à :
Les Éditions Diamond Service des Abonnements 10 Place de la Cathédrale, 68000 Colmar, France

Réf. : LP+3

☐ **1 LECTEUR : 189 € HT* / AN**
soit : 15,75 € HT / MOIS / LECTEUR

Réf. : LP+3/5

☐ **5 LECTEURS : 259 € HT* / AN**
soit : 4,32 € HT / MOIS / LECTEUR

(France) TVA 20% =

Total =

* Tarifs France Métro.

Merci de consulter les tarifs hors France Métro. sur : www.ed-diamond.com

Société : Nom : Prénom :

Adresse :

Code Postal : Ville :

Pays : Téléphone :

Veuillez indiquer svp l'adresse e-mail du référent :

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

Générer et manipuler des IMAGES ISO

MALIK LAICHAOUI



QUAND BIEN MÊME LEUR USAGE S'EST QUELQUE PEU ATTÉNUÉ AU PROFIT D'AUTRES SUPPORTS DE STOCKAGE DE MASSE DEPUIS PLUSIEURS ANNÉES, LES CD-ROM ET AUTRES DVD-ROM ONT TOUJOURS UNE UTILITÉ EN 2020 : STOCKAGE DE DONNÉES, CRÉATION DE DISQUES BOOTABLES (POUR DISTRIBUTIONS GNU/LINUX ENTRE AUTRES), DISQUES MULTIMÉDIAS... S'IL EST VRAI QUE SA DURÉE DE VIE A ÉTÉ SURESTIMÉE À UNE ÉPOQUE (PEUT-ÊTRE POUR FAVORISER SON UTILISATION), LA BONNE VIEILLE GALETTE N'EST PAS ENCORE MORTE, ET IL LUI RESTE ENCORE DE BEAUX JOURS À VIVRE.

À la base, une image ISO (ou image disque au format ISO 9660 [1]) désigne le contenu d'un disque optique, qui se présente sous la forme d'un fichier archive au format .iso. Celui-ci reproduit fidèlement l'arborescence de l'ensemble des fichiers et dossiers contenus sur ce même disque. Cette image

ISO peut donc s'avérer très utile lors de la réalisation de sauvegardes des données extraites d'un disque optique, puisqu'elle évite à l'utilisateur d'avoir à tout récupérer manuellement.

À l'inverse, il est tout à fait possible de générer une image ISO à partir de dossiers et fichiers, en

vue de sa gravure sur un disque optique, ou même sur une clé USB pour en faire un périphérique bootable. Celui-ci permettra alors de lancer un système d'exploitation en mode « live », avant d'éventuellement l'installer sur le disque dur d'un ordinateur (ou d'une machine virtuelle).

1. KÉSAKO ISO ?

Les différentes variantes des formats de systèmes de fichiers utilisés sur les supports optiques sont : ISO 9660, Joliet, Rockridge, El Torito, et UDF.

ISO vient du nom de l'International Standardization Organisation, l'institution en charge des différentes normes régissant divers domaines, allant des technologies du papier aux chemins de fers, en passant par l'agriculture et les matériaux de construction. Dans ce cas précis qui concerne l'informatique, c'est le raccourci de la norme ISO 9660, qui définit en particulier le système de fichiers d'un CD-ROM (ou DVD-ROM), celui-ci se devant d'être compatible avec tous les systèmes informatiques, qu'il s'agisse de GNU/Linux, Windows, macOS et autres Unix-like. Pour ce format ISO 9660, les noms de fichiers sont limités dans le choix et le nombre de caractères (de 8 à 32), et les fichiers peuvent être coupés en plusieurs morceaux. L'arborescence du système de fichiers est aussi limitée à 8 niveaux de profondeur.

La norme Joliet, qui est en fait une extension de l'ISO 9660, permet de s'affranchir de la limite de 32 caractères pour les noms de fichiers, en la repoussant à 64 caractères (et même à 103 caractères si l'on en croit cette page de documentation dédiée au programme genisoimage [2]).

Rockridge est également une amélioration de l'ISO 9660, mais spécifiquement dédiée aux

systèmes de fichiers POSIX. Les noms de fichiers peuvent alors comprendre 255 caractères (avec plus de choix possibles), l'arborescence du système de fichiers peut être plus profonde, et il devient possible d'utiliser des liens symboliques. Par ailleurs, la gestion des droits sur les fichiers et dossiers est semblable à celle mise en œuvre sur les systèmes Unix.

La norme El Torito permet quant à elle de réaliser des supports optiques directement bootables, pour ce qui concerne les machines modernes, ou bien permet au BIOS d'émuler le démarrage sur disque dur si le matériel ou le système d'exploitation est ancien.

Enfin, pour compléter ce descriptif des différentes normes de système de fichiers utilisées sur supports optiques, je suis obligé de vous parler du format UDF (pour Universal Disk Format), car c'est un format ouvert, succédant à l'ISO 9660 (il utilise les normes ISO/IEC 13346 et ECMA-167, qui sont gérées par l'Optical Storage Technology Association), et qui permet entre autres de réaliser des CD-ROM dits « multisessions », grâce notamment au procédé d'écriture par paquets. Il devient alors possible d'ajouter ou supprimer des données, puisqu'en termes de gravure, le disque n'est alors pas « fermé » (c'est-à-dire finalisé en quelque sorte). Ce format, qui supporte des tailles de fichiers supérieures aux autres normes, permet ainsi de travailler sur de plus grands

disques, et peut intégrer beaucoup plus d'informations sur les fichiers et les répertoires.

2. VÉRIFIER UNE IMAGE ISO

Avant de passer à la production et la gravure des fichiers au format .iso, je me dois d'évoquer deux questions de sécurité. Les images ISO téléchargées depuis Internet doivent absolument être authentifiées ET vérifiées.

L'authentification consiste à s'assurer que le fichier .iso de la distribution GNU/Linux que l'utilisateur vient de télécharger est bien issu de la source d'où il est censé venir, et ce sans modification aucune de la part d'un tiers qui aurait pu y introduire des logiciels malveillants. Cela est beaucoup plus courant qu'on ne le pense, notamment sur les copies illégales de logiciels propriétaires (suivez mon regard ;-)).

La vérification permet quant à elle de valider la conformité d'une copie de notre fichier .iso téléchargé, par rapport au fichier source situé sur le site de téléchargement.

Dans le cas d'une Debian 10.7 Mate, nous allons faire appel à deux fichiers, qui se trouvent juste à côté des liens vers les fichiers .iso que l'on s'apprête à rapatrier sur notre ordinateur, SHA256SUMS.sign et SHA256SUMS.

On utilise d'abord la commande pour vérifier l'intégrité de l'image ISO téléchargée :


```
# sha256sum -b debian-live-10.7.0-amd64-  
mate.iso
```

qui doit alors retourner la valeur suivante :

```
274a32812efb6994149bfcddb05d6738c7e2782a3565e  
d261349d14a0525682f
```

Il suffit de comparer cette valeur avec celle indiquée dans le fichier SHA256SUMS, en les copiant toutes les deux dans un éditeur de texte par exemple, pour s'assurer de leur exacte correspondance.

Ensuite, pour authentifier l'image disque de la distribution Debian, on installe un paquet dédié à la récupération des clés de chiffrement GPG :

```
# sudo apt install debian-keyring
```

Puis, on utilise la commande suivante (qui fonctionne également avec les fichiers SHA512SUMS.sign et SHA512SUMS) :

```
# gpgv --keyring /usr/share/keyrings/  
debian-role-keys.gpg -- SHA256SUMS.sign  
SHA256SUMS
```

qui doit retourner la sortie suivante :

```
gpgv: Signature faite le dim. 06 déc. 2020  
02:46:48 CET  
gpgv: avec la clef RSA  
DF9B9C49EAA9298432589D76DA87E80D6294BE9B  
gpgv: Bonne signature de " Debian CD  
signing key <debian-cd@lists.debian.org> "
```

pour nous informer que le tout est bien conforme à l'originale.

En cas de doute, il est en plus possible de se rendre à l'adresse : <https://www.debian.org/CD/verify> pour vérifier que notre clé RSA est bien présente sur cette page officielle de Debian, ce qui est bien le cas ici.

3. LES COMMANDES DE BASE

Nous allons commencer par passer en revue les principales commandes à saisir dans un terminal pour générer et manipuler des images disques au format ISO.

3.1 genisoimage

La commande `genisoimage` remplace l'ancien générateur de fichiers .iso `mkisofs`, suite à un fork des développeurs à l'époque de la distribution Debian 4.0 « Etch ». Il est souvent précisé qu'elle propose les mêmes options que le programme `xorriso`, que nous verrons par la suite, à ceci près que ce dernier permet en plus la gravure des images ISO produites. Dans les faits, la commande `genisoimage` est surtout préférée si l'utilisateur a besoin de créer des disques UDF (Universal Disk Format) ou HFS (Apple). Cependant, sans option particulière, elle génère par défaut une image au format ISO 9660.

Pour créer une image disque à partir du contenu d'un dossier donné :

```
# genisoimage -o nom_image.iso /home/  
user/nom_dossier_source
```

Pour créer une image disque avec extension Joliet et Rockridge et compatible HFS, on utilisera :

```
# genisoimage -o nom_image.iso -J -R -hfs  
/home/user/nom_dossier_source
```

BON À SAVOIR

Si l'on grave notre image sur un disque optique, le dossier source devient la racine du CD-ROM, qui n'affichera donc que le contenu de ce dossier source quand on le consultera dans un explorateur de fichiers. Le format de CD hybride HFS propose nombre d'options, tout comme la commande `genisoimage` d'ailleurs. La lecture de la page de manuel pourra apporter de précieuses informations à ce sujet.

3.2 La commande dd

La commande Unix historique `dd` [3] est très puissante. Elle est très utile lors de la réalisation d'une sauvegarde parfaite et complète d'un support de type disque dur, puisqu'elle copie toutes les zones du disque, c'est-à-dire les secteurs de démarrage,


```

malik@core2:~/Téléchargements$ genisoimage -o clonezilla.iso clonezilla/
I: -input-charset not specified, using utf-8 (detected in locale settings)
Using FILES000.PAC;1 for clonezilla/live/filesystem.packages.remove (filesystem.packages)
Using DEJAV000.PF2;1 for clonezilla/boot/grub/dejavu-bold-14.pf2 (dejavu-bold-16.pf2)
Using XNU_U000.MOD;1 for clonezilla/boot/grub/i386-efi/xnu_uuid_test.mod (xnu_uuid.mod)
Using MDRAI000.MOD;1 for clonezilla/boot/grub/i386-efi/mdraid09_be.mod (mdraid09.mod)
Using PASSW000.MOD;1 for clonezilla/boot/grub/i386-efi/password.mod (password_pbkdf2.mod)
Using GCRY_000.MOD;1 for clonezilla/boot/grub/i386-efi/gcry_sha1.mod (gcry_sha256.mod)
Using GFXTE000.MOD;1 for clonezilla/boot/grub/i386-efi/gfxterm_background.mod (gfxterm_menu.mod)
Using USBSE000.MOD;1 for clonezilla/boot/grub/i386-efi/usbserial_common.mod (usbserial_ftdi.mod)
Using GCRY_001.MOD;1 for clonezilla/boot/grub/i386-efi/gcry_sha256.mod (gcry_sha512.mod)
Using USBSE001.MOD;1 for clonezilla/boot/grub/i386-efi/usbserial_ftdi.mod (usbserial_pl2303.mod)
Using USBSE002.MOD;1 for clonezilla/boot/grub/i386-efi/usbserial_pl2303.mod (usbserial_usbdebug.mod)
Using MULTI000.MOD;1 for clonezilla/boot/grub/i386-efi/multiboot.mod (multiboot2.mod)
Using PART_000.MOD;1 for clonezilla/boot/grub/i386-efi/part_sun.mod (part_sunpc.mod)
Using VIDE0000.MOD;1 for clonezilla/boot/grub/i386-efi/videotest.mod (videotest_checksum.mod)
Using XNU_U000.MOD;1 for clonezilla/boot/grub/x86_64-efi/xnu_uuid_test.mod (xnu_uuid.mod)
Using MDRAI000.MOD;1 for clonezilla/boot/grub/x86_64-efi/mdraid09_be.mod (mdraid09.mod)
Using PASSW000.MOD;1 for clonezilla/boot/grub/x86_64-efi/password.mod (password_pbkdf2.mod)
Using GCRY_000.MOD;1 for clonezilla/boot/grub/x86_64-efi/gcry_sha1.mod (gcry_sha256.mod)
Using GFXTE000.MOD;1 for clonezilla/boot/grub/x86_64-efi/gfxterm_background.mod (gfxterm_menu.mod)
Using USBSE000.MOD;1 for clonezilla/boot/grub/x86_64-efi/usbserial_common.mod (usbserial_ftdi.mod)
Using GCRY_001.MOD;1 for clonezilla/boot/grub/x86_64-efi/gcry_sha256.mod (gcry_sha512.mod)
Using USBSE001.MOD;1 for clonezilla/boot/grub/x86_64-efi/usbserial_ftdi.mod (usbserial_pl2303.mod)
Using USBSE002.MOD;1 for clonezilla/boot/grub/x86_64-efi/usbserial_pl2303.mod (usbserial_usbdebug.mod)
Using MULTI000.MOD;1 for clonezilla/boot/grub/x86_64-efi/multiboot.mod (multiboot2.mod)
Using PART_000.MOD;1 for clonezilla/boot/grub/x86_64-efi/part_sun.mod (part_sunpc.mod)
Using VIDE0000.MOD;1 for clonezilla/boot/grub/x86_64-efi/videotest.mod (videotest_checksum.mod)
3.42% done, estimate finish Tue Nov 17 18:35:21 2020
6.85% done, estimate finish Tue Nov 17 18:35:21 2020
10.27% done, estimate finish Tue Nov 17 18:35:21 2020
13.70% done, estimate finish Tue Nov 17 18:35:21 2020
17.12% done, estimate finish Tue Nov 17 18:35:21 2020
20.55% done, estimate finish Tue Nov 17 18:35:21 2020
23.97% done, estimate finish Tue Nov 17 18:35:21 2020
27.39% done, estimate finish Tue Nov 17 18:35:21 2020
30.81% done, estimate finish Tue Nov 17 18:35:21 2020
34.24% done, estimate finish Tue Nov 17 18:35:21 2020
37.66% done, estimate finish Tue Nov 17 18:35:21 2020
41.09% done, estimate finish Tue Nov 17 18:35:21 2020

```

FIGURE 1. La création d'une image disque (bootable) de la distribution Clonezilla, à partir du dossier téléchargé sur le net (sous forme d'archive zip), avec la commande `genisoimage`.

les tables de partition et les partitions elles-mêmes. Elle est aussi utilisée pour riper le contenu d'un support optique. Son utilisation est assez simple. Pour produire une copie bit à bit d'un disque dur repéré en tant que `/dev/sdb` :

```
#sudo dd if=/dev/sdb /home/user/image_disk.iso
```

Pour extraire le contenu d'un CD-ROM ou DVD-ROM, et le récupérer sous la forme d'un fichier unique au format `.iso` :

```
#sudo dd if=/dev/cdrom0 /home/user/image_disk.iso
```

La commande `dd` permet par ailleurs de s'affranchir des programmes avec interfaces graphiques que sont UNetbootin et autres USB disk creator pour créer une clé USB bootable à partir d'une image disque au format `.iso`.

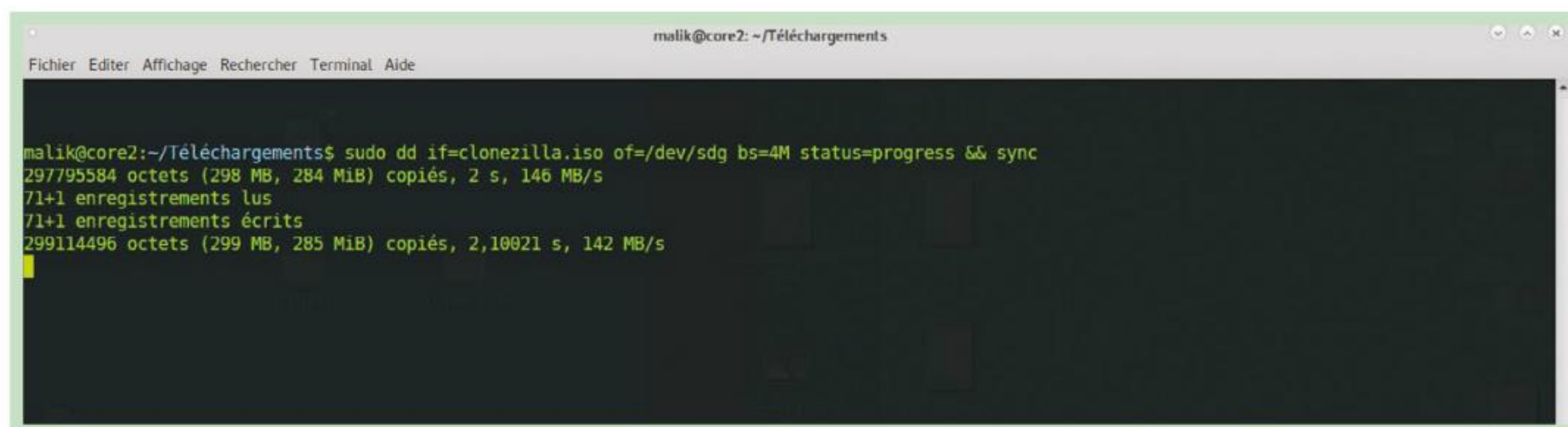
Elle s'utilise alors très simplement, depuis le répertoire où se situe l'image ISO d'une distribution de notre OS favori par exemple, avec la commande suivante :

```
#sudo dd if=image_disk_bootable.iso of=/dev/sdc bs=4M status=progress && sync
```

- `if` (input file) correspond à l'image ISO qui va être utilisée ;
- `of` (output file) correspond au périphérique sur lequel va être copiée l'image ISO, ici `/dev/sdc` correspond à la clé USB que l'on aura bien pris soin d'identifier avec la commande :

```
# sudo fdisk -l
```

- `bs` indique que la taille des blocs qui seront copiés sera de 4 mégaoctets.



```
malik@core2:~/Téléchargements$ sudo dd if=clonezilla.iso of=/dev/sdg bs=4M status=progress && sync
297795584 octets (298 MB, 284 MiB) copiés, 2 s, 146 MB/s
71+1 enregistrements lus
71+1 enregistrements écrits
299114496 octets (299 MB, 285 MiB) copiés, 2,10021 s, 142 MB/s
```

FIGURE 2. La progression de la création d'une clé USB bootable Clonezilla avec la commande `dd`. Simple et efficace.

L'argument `status=progress` informera de la progression de l'opération, puis la commande de synchronisation `sync` sera enchaînée directement après la réalisation de l'image ISO.

BON À SAVOIR

Vous l'aurez compris, la commande `dd` permet de tout faire, et bien plus encore. Il est possible de ne copier qu'une partition d'un disque, de cloner directement deux disques (attention aux capacités respectives de ceux-ci, ainsi qu'aux tailles des partitions), de créer un lecteur virtuel, d'effacer un lecteur, ou même d'écrire sur l'espace libre d'une partition donnée, afin de rendre complètement irrécupérables les fichiers précédemment supprimés. La consultation de la page de manuel de `dd` permettra de mieux comprendre la portée de ce véritable couteau suisse des systèmes Unix.

3.3 Montage d'une image ISO

On a parfois besoin de vérifier le contenu d'une image ISO avant de la graver ou de la copier sur une clé USB.

Il existe une solution simple, le montage de cette image dans un répertoire. Sur mon système, j'utilise un répertoire dédié à cette tâche, tout comme je le fais par ailleurs pour le partage NFS.

```
# sudo mkdir /mnt/ISO
```

va donc créer un dossier ISO dans `/mnt`, qui sert généralement au montage des supports amovibles, CD-ROM (`/mnt/cdrom`) ou autres partages NFS (`/mnt/NFS`).

On peut alors saisir la commande suivante pour monter notre fichier `.iso` dans le répertoire tout juste créé :

```
#sudo mount -o loop image.iso /mnt/ISO
```

Il est alors possible de lister les fichiers et dossiers contenus dans cette image ISO avec :

```
# ls -al /mnt/ISO
```

L'utilisateur peut parfois avoir besoin de naviguer visuellement dans l'arborescence ou de récupérer des éléments composant une image ISO. Il existe une solution très simple pour ce faire, à l'intérieur du navigateur de fichiers. Un clic droit puis **Extraire ici** (ou vers une destination autre), va recréer le dossier à partir duquel le fichier `.iso` a été généré. Notez que les navigateurs de fichiers



récents permettent même de cliquer directement sur le fichier .iso (clic gauche classique), repéré à l'aide d'une icône représentant un disque optique, afin d'accéder simplement à son contenu.

3.4 Graver en ligne de commandes

L'une de mes toutes premières activités sur un système GNU/Linux (une petite SUSE 6.2, il y a 20 ans déjà !) était d'extraire des pistes audios avec l'excellent `cdparanoia` (j'adore ce nom), puis de le graver sur CD avec `cdrecord`. Mettre les mains dans le système avec les lignes de commandes s'apparentait à de la magie je dois dire. Taper un simple `eject` et voir le tiroir du lecteur CD-ROM s'ouvrir était comme un miracle.

Si aujourd'hui il existe de belles interfaces graphiques pour réaliser ce type de tâches, il n'en reste pas moins que l'utilisation de la ligne de commandes est toujours d'une efficacité redoutable.

Dans un premier temps, il faut bien identifier nos lecteurs optiques, avec la commande :

```
# ls SCSI
```

qui retourne la liste des différents périphériques disponibles sur le système :

```
[0:0:0:0] disk ATA INTEL SSDSC2CT06 300i /dev/sda
[1:0:0:0] cd/dvd ATAPI DVD A DH16A1L KH39 /dev/sr0
[3:0:0:0] disk ATA ST3250820AS L /dev/sdb
[4:0:0:0] cd/dvd _ NEC DVD+-RW ND-6650A 1.23 /dev/sr1
[5:0:0:0] disk Generic- Compact Flash 1.00 /dev/sdc
[5:0:0:1] disk Generic- SM/xD-Picture 1.00 /dev/sdd
[5:0:0:2] disk Generic- SD/MMC 1.00 /dev/sde
[5:0:0:3] disk Generic- MS/MS-Pro 1.00 /dev/sdf
[6:0:0:0] disk SanDisk Cruzer Blade 1.00 /dev/sdg
```

Dans mon cas, deux lecteurs apparaissent clairement dans la catégorie « cd/dvd » : `/dev/sr0` et `/dev/sr1`, qui correspondent respectivement au lecteur/graveur interne de mon unité centrale en connexion SATA, et à un lecteur/graveur externe connecté sur port USB.

Il faut bien noter que l'ancienne commande `cdrecord` est à ce jour remplacée par `wodim`, qui reprend la syntaxe de son illustre ancêtre. Ainsi, pour graver une image ISO sur un CD-ROM, il suffira de saisir la ligne de commandes :

```
# wodim -v speed=10 dev=/dev/sr0 /home/user/clonezilla.iso
```

`/dev/sr0` correspondant au lecteur optique chargé de la gravure.

L'option `-v`, pour verbose, très utilisée en ligne de commandes, donne plusieurs informations avant et pendant les opérations, et l'argument

`speed` indique la vitesse maximale de gravure (en fonction de la vitesse maximale du graveur ET de celle du support à graver).

4. LES INTERFACES GRAPHIQUES

4.1 Isomaster

Le logiciel isomaster [4] est un programme très basique, dédié à une seule tâche comme souvent sous GNU/Linux. En l'occurrence, son rôle sera de générer (et manipuler) des images ISO à partir de dossiers et fichiers choisis sur le disque dur de l'utilisateur. Son interface est relativement simple et épurée. Elle se compose de deux parties, avec en haut de la fenêtre l'arborescence des répertoires et fichiers, et en bas, la zone de sélection.

Pour commencer, on crée un nouveau projet dans le menu **Fichier > Nouveau**, ce qui aura pour effet « d'activer » la zone basse de la fenêtre pour permettre d'y ajouter les éléments à intégrer dans l'image ISO finale. Attention, le glisser-déposer ne fonctionne pas. Il faudra donc cliquer sur le/les fichiers et dossiers souhaités, puis appuyer sur le bouton **Ajouter**.

Une fois que cela est fait, on se rend dans le menu **Fichier > Enregistrer sous**, et on sélectionne un dossier de destination ainsi qu'un nom de fichier .iso, et le tour est joué (Figure 3, page suivante).

Isomaster donne également la possibilité de manipuler/modifier des images ISO existantes. Si on

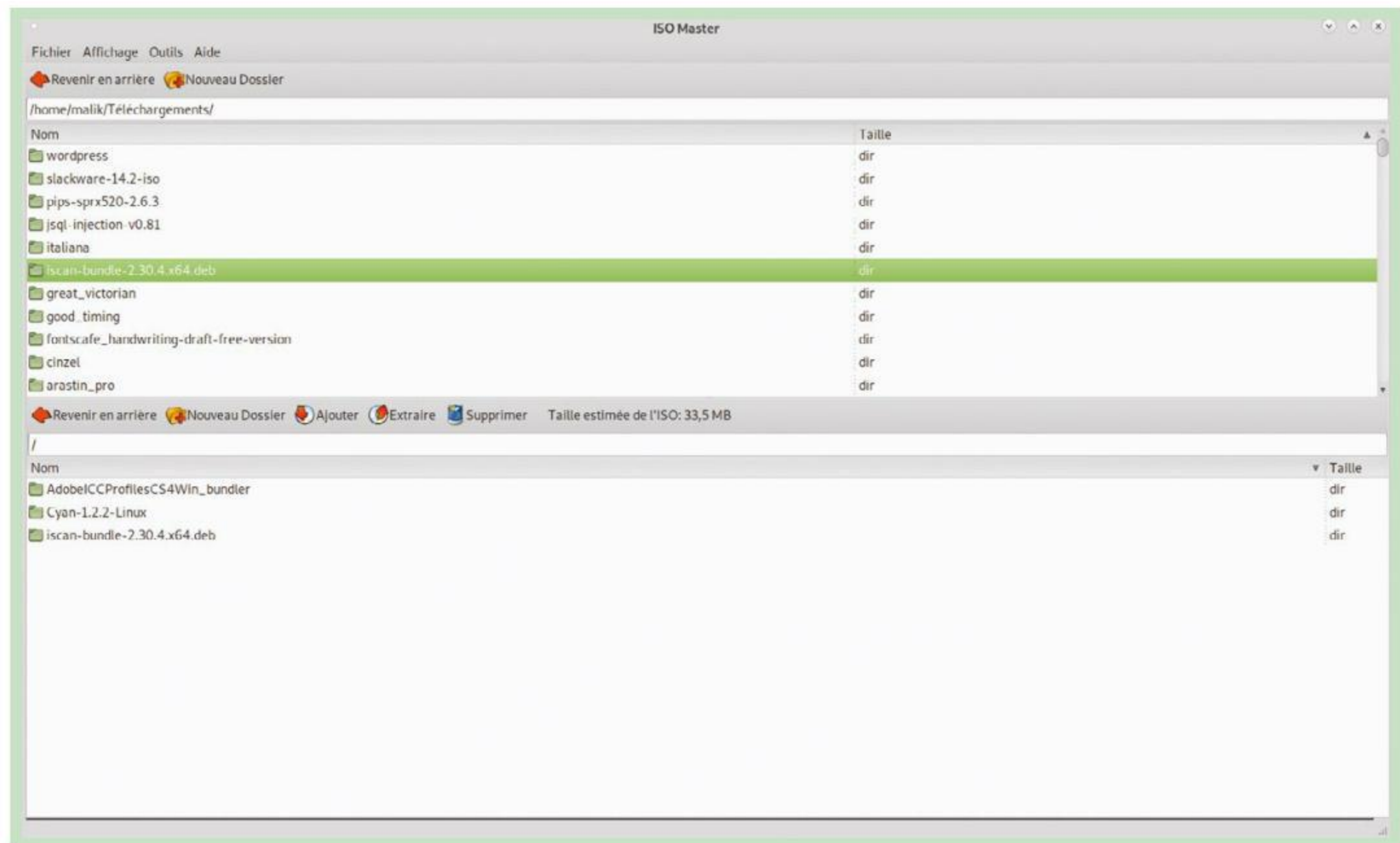


FIGURE 3. L'interface du programme Isomaster, très simple à prendre en main.

va dans le menu **Fichier > Ouvrir** et que l'on sélectionne un fichier .iso sur le disque dur, on va avoir accès à son contenu sous forme d'arborescence classique. Il est alors possible de choisir d'extraire un ou plusieurs éléments (fichier(s) et/ou dossier(s)), en les sélectionnant et en appuyant sur le bouton **Extraire** de la barre d'outils centrale. Les éléments seront ainsi extraits dans le répertoire de travail désigné dans la partie haute de l'interface d'isomaster. Cela devient tout à coup très pratique et ergonomique pour modifier une image ISO existante. Une fois l'arborescence extraite, il suffit de rajouter des éléments comme vu précédemment, et de générer une nouvelle image ISO.

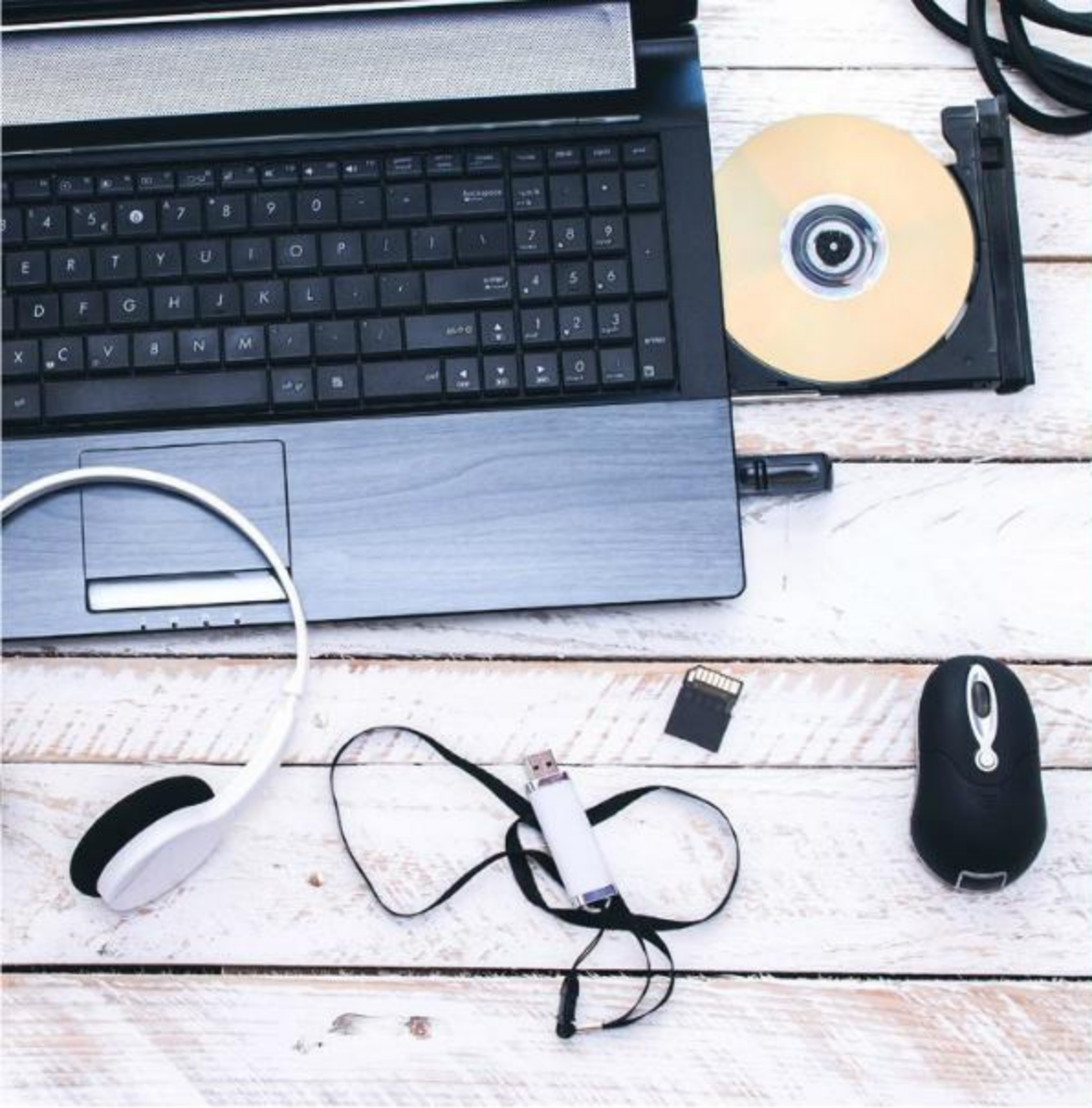
4.2 xorriso (en mode Tcl-Tk)

Le programme [xorriso](#) permet de produire et manipuler des images disque au format .iso, mais il est également capable de graver ces images sur support optique. Il peut aussi être utilisé pour effacer un CDRW, mais il ne grave pas de CD audio, et ne travaille pas avec le format UDF.

Si xorriso est bien utilisable en ligne de commandes seule (la lecture de la page de manuel est conseillée, ne serait-ce que pour avoir un aperçu de ses nombreuses capacités et options), nous allons passer en

revue son interface graphique [xorriso-tcltk](#), programmée comme son nom l'indique à l'aide du langage interprété Tcl-Tk [5] (« tickle-ticket » pour les initiés). En fait d'interface graphique, il s'agirait plutôt d'un « proof of concept de frontend de xorriso, fonctionnant en mode dialogue », d'après la description donnée par la documentation même du logiciel. D'où son aspect quelque peu austère, et en anglais qui plus est. Mais il en faut plus pour arrêter un linuxien motivé.

Pour exécuter le programme depuis un terminal (car il n'y a pas d'icône disponible dans le menu suite à l'installation) :



```
# xorriso-tcltk &
```

Une fois le programme lancé, on découvre une véritable petite usine à gaz, avec des boutons dans tous les recoins de la fenêtre. Première chose très pratique : un simple clic droit sur n'importe quel bouton,

et même sur une étiquette (un label), ouvre une aide de type infobulle plutôt complète.

L'interface se décompose en 3 parties :

- La partie haute qui offre la possibilité de passer des commandes manuelles. On peut surtout y observer la sortie et l'évolution des différentes manipulations, notamment les erreurs. Notez l'emplacement plutôt inhabituel du bouton **End**, qui sert à quitter le logiciel, en haut à gauche de la fenêtre.
- La partie centrale concerne la détection des lecteurs optiques (**Scan drives**) et leur sélection (**Pick Input et Output drives**), l'effacement de support optique (**Blank**), ainsi que l'écriture des données à proprement parler (**Write ISO session**).
- La partie inférieure est quant à elle utilisée pour la manipulation d'images ISO : extraction des données vers le disque dur, création d'images ISO à partir d'éléments du disque dur, modification de données composant une image ISO.

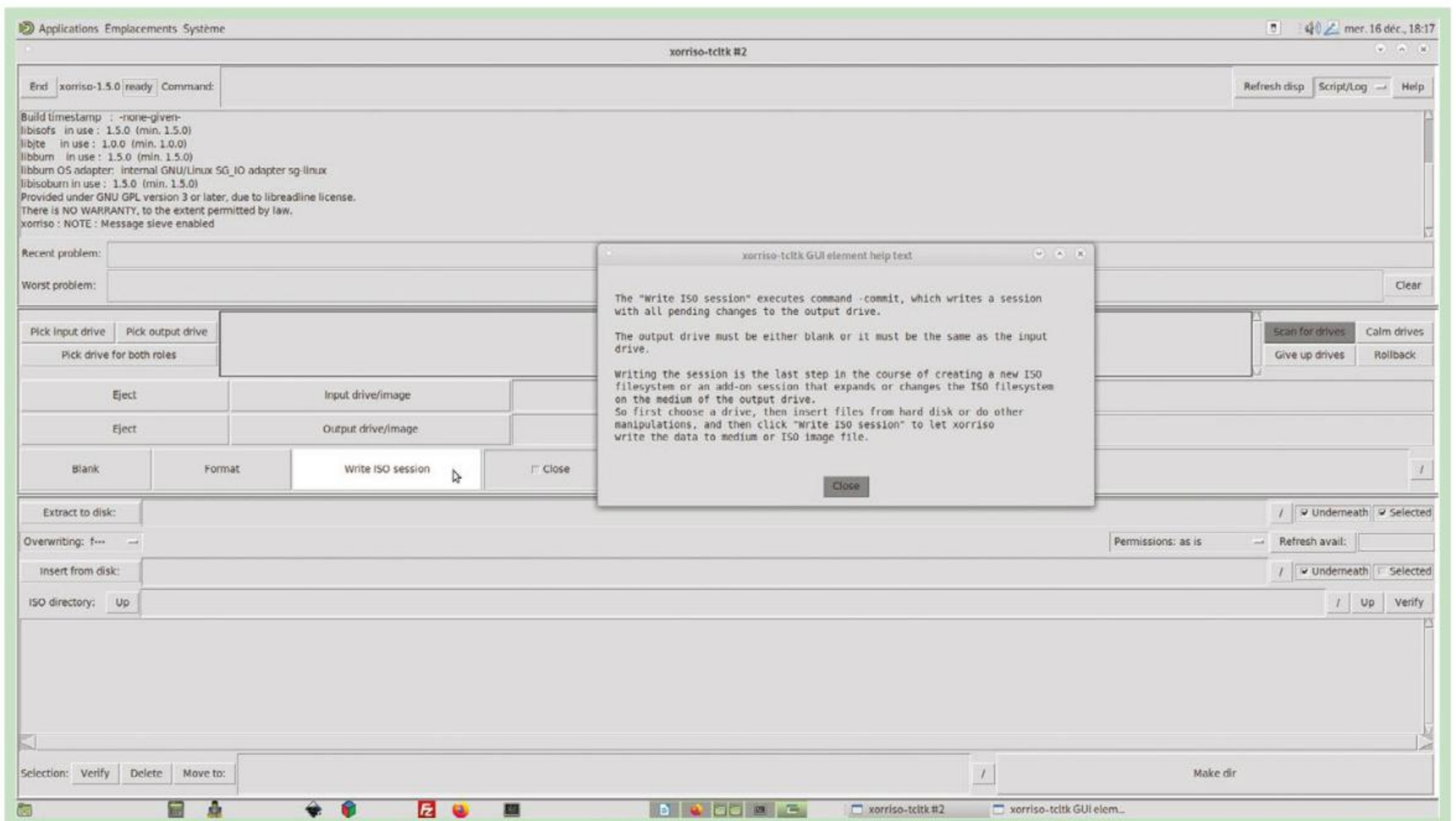


FIGURE 4. L'interface du programme xorriso-tcltk, avec ses aides sous la forme d'infobulles.

Passons aux choses concrètes en montrant comment générer et écrire une image ISO sur un disque optique.

4.2.1 Produire une image ISO

Il faut tout d'abord renseigner le champ **Output drive/image** pour nommer notre fichier ISO final. On appuie sur [Entrée] pour valider le choix.

Le champ **Insert from disk** va permettre d'ajouter les éléments que l'on souhaite intégrer à notre image ISO. Le petit bouton **/** ouvre un navigateur de fichiers pour réaliser cette tâche. Ces éléments apparaissent alors dans une liste qui est désignée par **ISO directory**, affichant l'arborescence de notre futur disque. Je conseille dans un premier temps de décocher les cases **Underneath** et **Selected**, qui donnent la possibilité d'ajouter des éléments soit à l'intérieur d'un dossier sélectionné, soit au même niveau que lui dans l'arborescence. In fine, quelques tests seront nécessaires pour bien comprendre l'action de ces deux options, et comment sont organisés les éléments au sein de l'arborescence.

Il suffit alors de cliquer sur le bouton **Write session** pour générer le fichier ISO.

4.2.2 Graver une image ISO

Avant de commencer, on valide la détection du matériel de gravure : c'est le rôle du bouton **Scan Drives** qui va faire apparaître les différents lecteurs/graveurs à notre disposition. On clique sur le lecteur de destination dans lequel sera introduit le CD-ROM

à graver, couramment **/dev/sr0** si vous n'en possédez qu'un seul, puis on confirme ce choix avec le bouton **Pick Output drive**.

Pour graver une image ISO, nous utilisons le bouton **Burn image file**, grâce auquel on indique le chemin vers le fichier .iso à graver, puis le bouton **Write ISO session**, qui va lancer l'opération de gravure du disque.

Attention, si vous utilisez un disque réinscriptible (CD-RW ou DVD-RW) déjà gravé, il vous sera rappelé de procéder à son effacement à l'aide du bouton **Blank**.

CONCLUSION

Comme toujours avec notre système favori, la prise en main des logiciels via la ligne de commandes permet de s'approprier les multiples possibilités offertes, ainsi qu'une meilleure compréhension de leur fonctionnement. Le programme **dd** en est un très bon exemple.

Du côté des interfaces graphiques, il existe aujourd'hui beaucoup de solutions à la disposition des utilisateurs désireux de graver des supports optiques : K3b a toujours été très efficace et complet, Xfburn est mis en œuvre avec le bureau Xfce, et Brasero est peut-être l'un des programmes les plus ergonomiques et les plus simples pour extraire ou graver des données. Vous n'avez maintenant plus d'excuses pour ne pas brûler les disques que vous avez retrouvés en faisant du rangement pendant le confinement. ■

RÉFÉRENCES

- [1] https://fr.wikipedia.org/wiki/ISO_9660
- [2] <https://linux.die.net/man/1/genisoimage>
- [3] [https://fr.wikipedia.org/wiki/Dd_\(Unix\)](https://fr.wikipedia.org/wiki/Dd_(Unix))
- [4] <https://doc.ubuntu-fr.org/isomaster>
- [5] https://fr.wikipedia.org/wiki/Tool_Command_Language





Chez votre marchand de journaux
et sur **www.ed-diamond.com**



en kiosque



FLIPBOOK HTML5

sur **www.ed-diamond.com**



CONNECT
LA DOCUMENTATION TECHNIQUE DES PROS DE L'IT

sur **connect.ed-diamond.com**

CRÉEZ VOS CLUSTERS avec Elasticsearch

STÉPHANE LONKENG TOULEPI

VOUS ÊTES-VOUS DÉJÀ DEMANDÉ COMMENT FACEBOOK ET GOOGLE GÈRENT LEURS IMMENSES MASSES DE DONNÉES TOUT EN FOURNISSANT DES SERVICES RAPIDES ? QUELLE SERAIT VOTRE SOLUTION POUR GÉRER UNE BASE DE DONNÉES AVEC UN TRAFIC DE DONNÉES ABONDANT ? VOICI QUELQUES ÉLÉMENTS DE RÉPONSE À CES QUESTIONS.



La croissance du nombre d'internautes, ajoutée à l'évolution des technologies et des objets connectés a entraîné un grand trafic de données sur le réseau Internet. Ces données depuis les années 2000 sont de plus en plus grandissantes, et les concepteurs d'applications se sont rendu compte qu'ils ne pouvaient plus les gérer avec les systèmes de gestion de bases de données (SGBD) qu'ils utilisaient à leur époque à cause des limites du

modèle relationnel de données. Ainsi, une nouvelle approche a été pensée et de nouveaux types de bases de données ont vu le jour, apportant ainsi plus de robustesse et de flexibilité. L'une d'entre elles a fait beaucoup parler d'elle, il s'agit d'Elasticsearch. Dans cet article, nous l'étudierons donc à travers son concept, et verrons comment l'installer sur un réseau d'ordinateurs.

1. PRÉSENTATION D'ELASTICSEARCH

1.1 Nature

Elasticsearch est un moteur de recherche distribué qui permet aux utilisateurs de créer des bases de données NoSQL et de les manager à l'aide d'une API Rest [1].

Dans le terme « moteur de recherche distribué », il y a « moteur de recherche » et « distribué ». Un moteur de recherche est un système qui indexe un ensemble d'informations (dans une sorte de base de données) et qui, lorsqu'il est appelé, permet de faire une recherche analytique et efficace dans sa base de données. Quant à « distribué », cela veut en fait dire que la base de données en question est décentralisée ou distribuée dans plusieurs unités, qui peuvent être des serveurs interconnectés. En d'autres termes, la recherche est décentralisée, et puis les informations remontent et sont reconstituées pour fournir le résultat final de la recherche.

Ensuite, nous avons le terme « bases de données NoSQL ». Une base de données NoSQL (qui signifie Not Only SQL) est un tout autre type de base de données qui en général ne suit pas le format relationnel, c'est-à-dire le format de tuple, relation, et algèbre relationnelle. Ces bases de données suivent en général des approches visant à optimiser la vitesse de recherche et sur le coup elles ne disposent pas d'un schéma figé. Toutefois, il y a une chose qui revient très souvent, c'est le système « key-value », qui est en fait une manière de collecter les données à l'aide d'une clé unique et d'une valeur qui est la donnée en question [2]. Le but étant de procéder aux recherches avec une complexité algorithmique égale à $O(1)$.

» REMARQUE

TUPLE

Un tuple est une liste finie d'éléments ordonnés, en mathématiques, c'est ce qu'on appelle un n-uplet. Par exemple, un couple (x, y) est un tuple à 2 éléments, où le premier élément est x et le second y . Dans un tuple, les éléments peuvent se répéter, ce qui n'est pas le cas dans son rival de toujours l'ensemble. Ainsi, on peut avoir le tuple (x, y, z, x, x) , de plus une permutation dans un tuple forme un autre tuple. En d'autres termes, les tuples (x, y, z) et (x, z, y) sont deux tuples différents.

Enfin, nous avons le terme « API REST ». Tout d'abord, une API (Application Programming Interface) est un composant (ou une boîte noire) qui fournit un ensemble de services et qui permet à l'utilisateur d'accéder à ces services à l'aide d'une interface simplifiée et sécurisée [3]. Elle peut se présenter sous plusieurs formes, comme une bibliothèque informatique ou comme un service web. Et justement, c'est de la forme web qu'interviennent les API REST.

Tout d'abord, le terme REST signifie Representational state transfer, ensuite une API REST est tout simplement une API qui fonctionne à l'aide de requêtes HTTP de type GET, HEAD, POST, PUT, etc.. La particularité de ces API c'est que les résultats sont formatés à l'aide d'une syntaxe standardisée telle que le format JSON ou XML. Ces API sont alors des services tournant sur des serveurs et qui fonctionnent en recevant des requêtes HTTP, et renvoyant des réponses formatées et utilisables dans nos applications personnelles.

En définitive, Elasticsearch est donc un système de gestion de base de données NoSQL fonctionnant sous forme d'un service web, et fournissant à son utilisateur une API REST comme décrite plus haut.

1.2 Cluster et nœuds

Elasticsearch est un SGBD distribué, cela signifie que la base de données en question est répartie sur plusieurs unités ou serveurs. Ces serveurs sont interconnectés entre eux pour ainsi former un

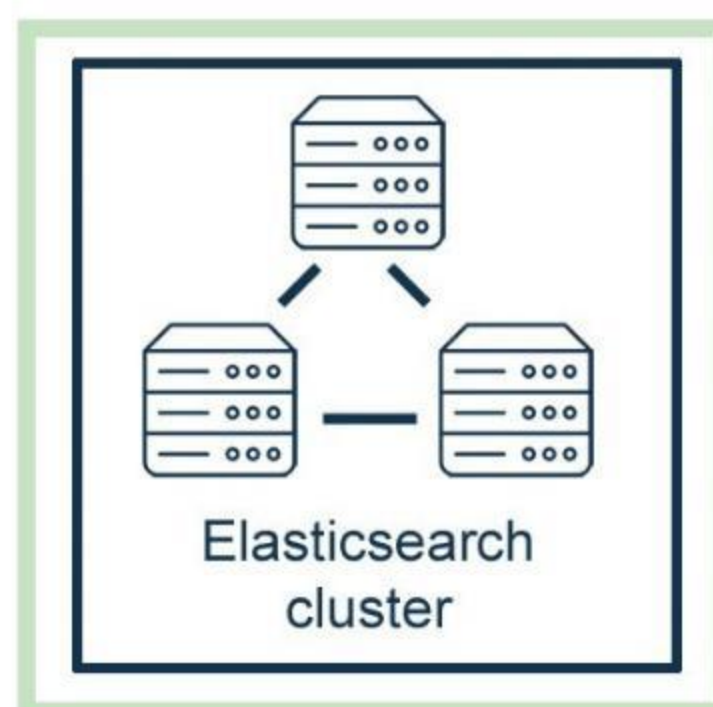


FIGURE 1. Trois nœuds interconnectés qui forment ensemble un cluster [5].

cluster. Elasticsearch nécessite toujours un cluster pour fonctionner et celui-ci doit disposer d'au moins 2 serveurs. Sur Elasticsearch, un cluster est défini par un nom, et toutes les unités possèdent ce nom qui permet de définir leur appartenance. Les unités dans un cluster sont appelées des nœuds (Figure 1, page précédente).

Il existe 3 principaux types de nœuds sur Elasticsearch : le master node, le data node et le ingest node.

1.2.1 Le nœud master (Master node)

Le nœud master est le nœud qui manage un cluster, c'est le nœud principal du cluster. Il donne les instructions aux autres nœuds, c'est lui qui lance les requêtes de recherche et d'insertion de données. C'est également lui qui distribue les parts de données dans tout le cluster. En plus de cela il partage l'état du cluster à tous les nœuds du réseau.

Par ailleurs, un cluster doit toujours posséder un et un seul nœud master. Au départ, les nœuds que l'on souhaite définir comme probables futurs masters, possèdent le titre de master-eligible node. En d'autres termes, ils sont éligibles au poste de master, un processus d'élection du master est opéré par la suite [1].

1.2.2 Le nœud data (Data Node)

Un nœud data dans un cluster est chargé de conserver les données et les copies. Il est important de le dissocier du nœud master, car sa mission est précise, conserver les données et effectuer les opérations de recherche et de modification sur ces données. Il reçoit les instructions du nœud master et les exécute sur ses données.

1.2.3 Le nœud ingest (Ingest node)

Les nœuds ingest sont des nœuds ayant une fonction très technique. Leur objectif est de créer des pipelines afin d'exécuter les requêtes plus efficacement dans le cluster, ils tiennent compte des ressources requises et du type d'opération à effectuer. Quand une requête arrive, elle est prétraitée et séquencée efficacement.

1.3 Index et document

En Elasticsearch, un index est une collection de données d'une même catégorie, ou d'un même concept. Il peut être vu en base de données SQL comme une table, mais le concept est plus large que ça.

Une indexation est en quelque sorte une catégorisation d'un ensemble de données afin de faciliter la recherche. En d'autres termes, il s'agit de poser une étiquette sur une donnée qui ici s'appelle document.

Ainsi, on peut avoir l'index des voitures, l'index des étudiants, etc.

Un document quant à lui est l'unité d'information. C'est l'entité unitaire d'un index, il est représenté sous le format JSON (JavaScript Object Notation). Les documents au sein d'un même index n'ont pas obligatoirement la même structure, cela dépend du concepteur de la base de données, seulement ceux-ci possèdent la même idéologie.

1.4 Fragments et répliques

L'une des politiques d'Elasticsearch et de la majorité des SGBD est de garantir la rapidité d'accès aux données. Ainsi, Elasticsearch a mis en place dans son système un principe de partage des données entre différents nœuds. Le but étant d'appliquer le principe de « diviser pour régner », en effet, Elasticsearch divise les index en ce qu'on appelle fragment (ou éclat, shard en anglais) pour les partager à différents nœuds du cluster, et pas n'importe quel nœud, il les partage aux nœuds data.

En gros, un fragment est, comme son nom l'indique, un fragment d'un index délocalisé dans un nœud. Cette notion est importante en sécurité car, imaginez que le cluster soit attaqué par des pirates informatiques, et des serveurs ainsi mis à terre. Si les données étaient concentrées dans un seul serveur, le concepteur court alors le risque de perdre toutes ses données, tandis que si les données étaient départagées dans plusieurs serveurs, le concepteur du cluster a moins de chance de tout perdre.

Dans la même optique de sécurisation des données, il existe un principe de duplication des données mis

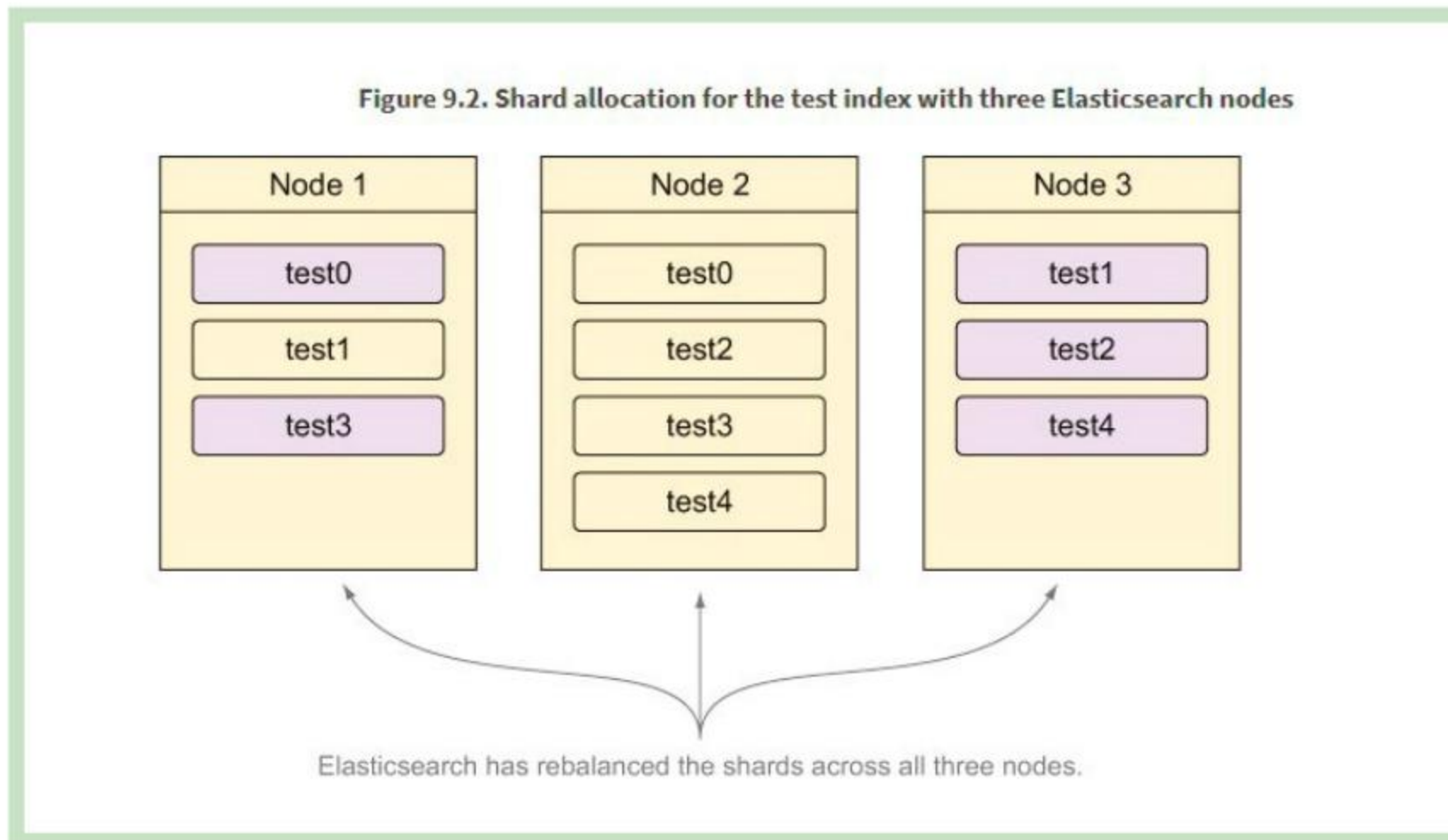


FIGURE 2.
Exemple de
répartition des
fragments et des
répliques [6].

en place par Elasticsearch. Le principe est que pour chaque fragment d'index, une réplique est créée qui contient exactement les mêmes données et qui est mise à jour à chaque modification de l'originale. Cette réplique est stockée dans un autre nœud du cluster. Le but étant de permettre au nœud contenant la réplique de prendre la relève, lorsque le nœud contenant le fragment original tombe en panne, ou est indisponible à un moment donné.

2. INSTALLATION D'ELASTICSEARCH

2.1 Téléchargement et installation

Elasticsearch est un outil gratuit d'Elastic, vous le trouverez ici : <https://www.elastic.co/fr/downloads/elasticsearch>.

Une fois que vous l'avez téléchargé, vous obtiendrez un fichier d'extension .tar.gz, décompressez ce fichier dans votre répertoire personnel avec la commande suivante :

```
tar -xzf <nom _ du _ fichier> -C ~
```

La décompression étant faite, il faut l'ajouter à vos variables d'environnement.

Pour cela, il existe dans votre répertoire personnel un fichier caché nommé .bashrc, ouvrez-le avec votre éditeur de texte. Allez à la fin du fichier et ajoutez cette ligne de code :

```
export ELASTICSEARCH_HOME=<chemin _  
absolu _ vers _ elasticsearch>
```

Où vous remplacerez <chemin_absolu_vers_elasticsearch>, par le chemin absolu vers votre répertoire d'Elasticsearch. Par exemple, dans mon cas c'est /home/xxx/elasticsearch-7.7.1.

Cette ligne a pour but de créer une variable contenant le chemin vers le dossier d'Elasticsearch.

» REMARQUE

ATTENTION

Souvent certains programmes que vous avez installés ont modifié votre fichier .bashrc comme nous sommes en train de le faire. En principe, si tout est bien fait, l'ajout de nouvelles lignes ne pose pas de problème, mais il faut vérifier que les lignes que vous ajoutez à la fin soient exécutées.

Ajoutez à la suite la ligne suivante :

```
export PATH=${ELASTICSEARCH_HOME}/bin:${PATH}
```

Cette ligne permet de rendre accessibles les programmes du dossier `bin` d'Elasticsearch dans le terminal. Une fois que c'est fait, sauvegardez le fichier et fermez-le.

Pour vérifier que tout s'est bien passé, exécutez la commande suivante dans votre terminal :

```
$ elasticsearch --version
```

Vous obtiendrez alors un message similaire à celui-ci :

```
Version: 7.7.1, Build: default/tar/ad56dce
891c901a492bb1ee393f12dfff473a423/2020-05-
28T16:30:01.040088Z, JVM: 1.8.0 _144
```

Elasticsearch dépend de Java présent dans votre machine.

2.2 Configuration

Dans cette section, nous allons configurer notre premier cluster. Notez que pour cette pratique il est important de disposer d'au moins 2 ordinateurs, en effet pour pouvoir utiliser les fonctionnalités d'Elasticsearch, il vous faut disposer de plusieurs nœuds.

La configuration d'Elasticsearch est plutôt simple, et elle se passe dans 3 fichiers se trouvant dans le dossier `config` d'Elasticsearch :

- `elasticsearch.yml` : ce fichier permet de configurer le statut et quelques propriétés personnelles du nœud en question ;

- `jvm.options` : ce fichier permet de configurer la machine virtuelle Java (JVM) qu'Elasticsearch créera (rappelez-vous Elasticsearch utilise Java) ;
- `log4j2.properties` : ce fichier permet de configurer les fichiers de logs.

Le fichier le plus important est `elasticsearch.yml`, dans le pire des cas on fera des configurations sur la JVM.

Avant de commencer la configuration et en supposant que vous disposez de deux postes, vous devez installer Elasticsearch comme vu précédemment sur les deux postes, c'est la première condition. Pour la seconde, il vous faut connecter vos 2 postes sur le même réseau local. En effet, n'oubliez pas qu'un cluster nécessite des postes interconnectés.

Pour le faire, c'est simple, si vous disposez d'un modem (un smartphone en général peut faire l'affaire en activant le hotspot), il vous suffit de connecter vos deux postes à ce modem. Pas besoin de connexion à Internet sur le modem.

Une fois que c'est fait, nous allons commencer par une petite modification sur les propriétés de la JVM. Ouvrons le fichier `jvm.options`.

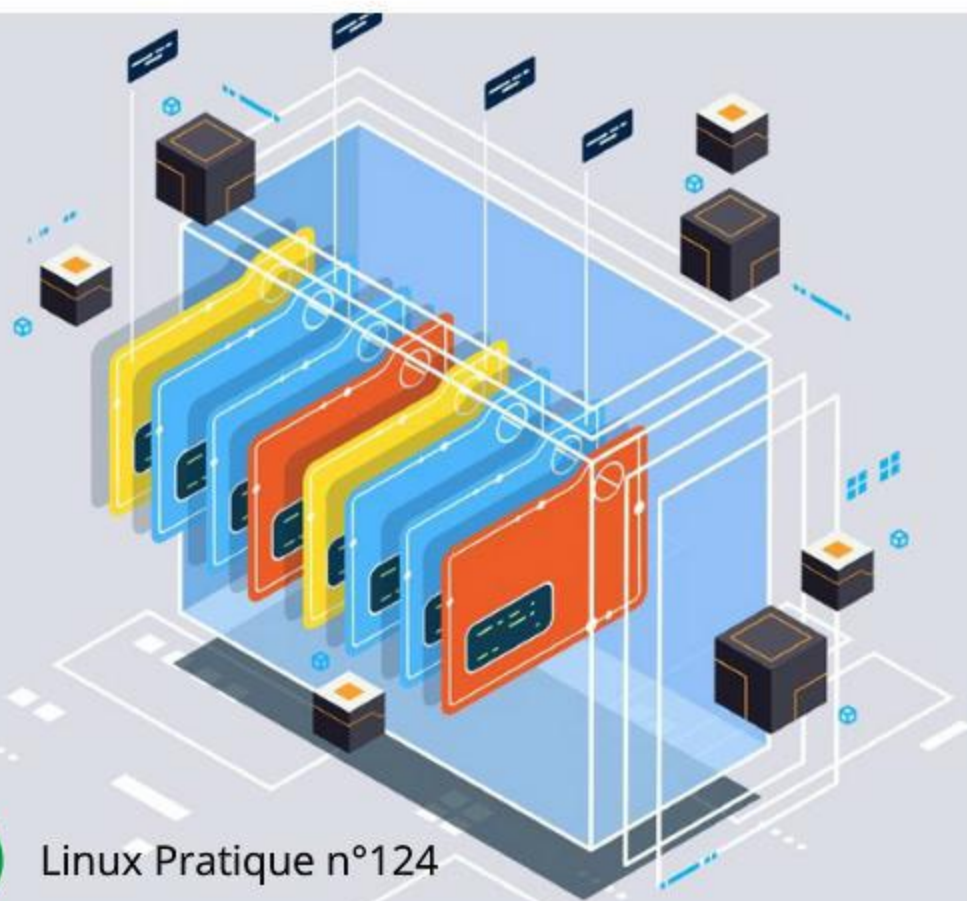
Il y a ici une configuration importante à faire, c'est la configuration de la taille du tas de la JVM :

```
#####
#####
## IMPORTANT: JVM heap size
#####
#####
```

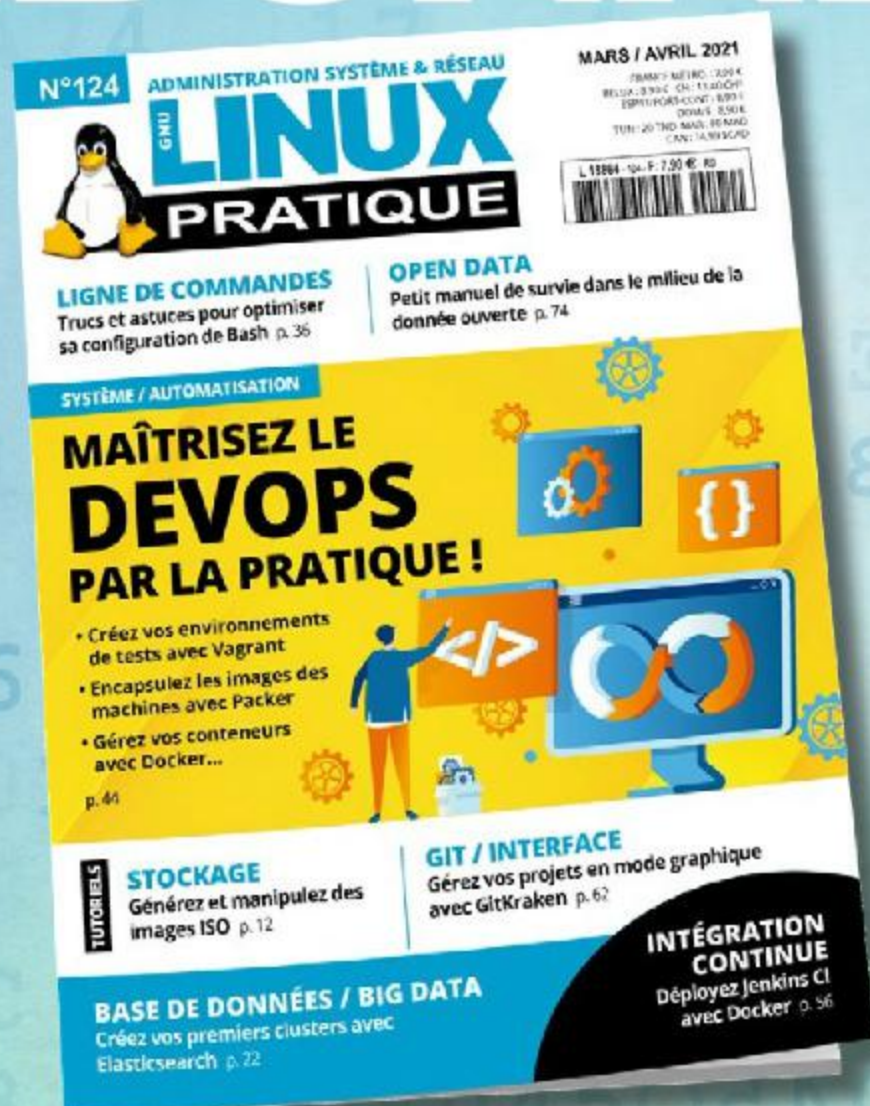
En programmation des systèmes, le tas d'un programme est l'espace dans lequel il fait des allocations de mémoire. Elasticsearch étant un programme, il fonctionne avec un tas, et on s'apprête justement à configurer la taille de son tas. Il y a deux valeurs à configurer, la taille minimale et la taille maximale. La configuration actuelle indique que les tailles minimales et maximales valent 1 Go.

Nous allons fixer ces valeurs à 2 Go, parce qu'en général 1 Go s'avère petit. Pour passer à 2 Go, il faut juste changer le 1 en 2 dans le code ci-dessus :

```
-Xms2g
-Xmx2g
```



ABONNEZ-VOUS !



6 MAGAZINES/AN

39 €*

au lieu de 47,40 €

**FRAIS
DE PORT
OFFERTS**

*Prix TTC en Euros / France Métropolitaine

**DISPONIBLE
EN VERSION
PAPIER OU
FLIPBOOK**

DISPONIBLE EN VERSION PAPIER OU FLIPBOOK



Offre Papier : LP1

Offre Flipbook : LP4

39 €*

Prix kiosque :
47,40 €

Économie :
8,40 €

*Prix TTC en Euros / France Métropolitaine

DISPONIBLE EN VERSION PAPIER OU FLIPBOOK



Offre Papier : LP+1

Offre Flipbook : LP+4

69 €*

Prix kiosque :
92,10 €

Économie :
23,10 €

*Prix TTC en Euros / France Métropolitaine



DÉCOUVREZ LE FLIPBOOK !

sur :
www.ed-diamond.com

BULLETIN D'ABONNEMENT

JE M'ABONNE À LINUX PRATIQUE

- ☐ Offre LP1 - 6 numéros pour 39 €* (papier)
☐ Offre LP4 - 6 numéros pour 39 €* (Flipbook)

JE M'ABONNE À LINUX PRATIQUE ET SES HORS-SÉRIES

- ☐ Offre LP+1 - 6 numéros et 3 hors-séries pour 69 €* (papier)
☐ Offre LP+4 - 6 numéros et 3 hors-séries pour 69 €* (Flipbook)

*Prix TTC en Euros / France Métropolitaine - Les tarifs hors France Métropolitaine, Europe, Asie, etc. sont disponibles en ligne !

À découper ou recopier et à renvoyer avec votre règlement à :

Les Éditions Diamond Service des Abonnements
10 Place de la Cathédrale - 68000 Colmar - France

JE RÈGLE €

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

COORDONNÉES DE L'ABONNÉ

Société :
Nom :
Prénom :
Adresse :
Code Postal :
Ville :
Pays :
Téléphone :
E-mail :

- ☐ J'autorise Linux Pratique à me contacter par e-mail ou par téléphone

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

FRAIS DE PORTS OFFERTS*

sur tous nos magazines en kiosque !



Pas envie de vous déplacer ? Profitez-en !



**LIVRÉS CHEZ VOUS
GRATUITEMENT !***



* Frais de ports offerts pour toute livraison en France métropolitaine, offre valable uniquement pour toutes les publications de la rubrique « En kiosque » de la boutique www.ed-diamond.com, appliquée une fois connecté à votre espace personnel.

Rendez-vous sur www.ed-diamond.com

» REMARQUE

AVERTISSEMENT

Pour chaque modification que vous faites dans les fichiers de configuration, si vous avez déjà démarré Elasticsearch, il va falloir l'arrêter puis le démarrer à nouveau.

Cette configuration doit être faite sur tous les postes.

Maintenant, passons à la configuration du cluster et des nœuds. Un cluster dans un réseau local est défini par son nom. Nous allons appeler notre cluster `mycluster`.

Pour faire cette configuration, ouvrez le fichier `elasticsearch.yml` dans les deux postes. Toutes ses lignes en commentaires décrivent comment le configurer. On apprend notamment que la configuration du cluster se fait par la configuration de la clé `cluster.name`. Ainsi, allons à la fin de notre fichier et ajoutons la ligne suivante :

```
cluster.name: mycluster
```

On a ainsi défini le nom de notre cluster. Cela permettra aux différents nœuds du réseau de se reconnaître entre eux.

Maintenant, passons à la configuration des propriétés des nœuds. Chaque nœud dans le cluster dispose d'un nom. Dans notre cas, nous possédons 2 nœuds. Le premier nœud, qui jouera le rôle de master, sera `node-1`, le second nœud, celui qui jouera le rôle de data, s'appellera `node-2`. Dans cette architecture, nous n'aurons pas de nœud ingest.

Pour faire ces configurations, ajoutons dans le fichier `elasticsearch.yml` du poste 1 la ligne suivante :

```
node.name: node-1
```

Ensuite dans le poste 2, ajoutons la ligne suivante :

```
node.name: node-2
```

Nous allons à présent configurer la nature des nœuds. Dans ce cluster, nous aurons besoin d'un master et de deux data. Pour cela, nous allons cumuler le rôle de

master et de data au nœud 1, et le nœud 2 sera juste data. J'en profite également pour vous dire qu'il est possible de cumuler certains rôles.

Dans le fichier de configuration du premier nœud, ajoutons le code suivant :

```
node.master: true
node.data: true
node.remote_cluster_client: false
cluster.routing.allocation.disk.threshold_enabled: false
```

Cela permet d'indiquer que le nœud sera un nœud master et un nœud data, tout en désactivant quelques configurations qui sont par défaut actives.

Dans le fichier de configuration du second nœud, ajoutons le code suivant :

```
node.master: false
node.data: true
node.remote_cluster_client: false
cluster.routing.allocation.disk.threshold_enabled: false
```

Si dans une autre configuration vous souhaitez disposer d'un nœud `voting-only` comme présenté plus haut, voici la configuration à apporter :

```
node.master: true
node.voting_only: true
```

Quant au nœud ingest, la configuration à apporter est la suivante :

```
node.ingest: true
```

Nous allons passer à la configuration de la connectivité. En d'autres termes, nous allons indiquer à Elasticsearch les identifiants des nœuds sur le réseau local.

Les ordinateurs disposent de plusieurs unités avec lesquels ils peuvent se connecter à un réseau. Par unité ici, on peut comprendre cartes réseaux. En général, nous avons au moins la carte Ethernet et la carte Wifi.

Ainsi, nous devons indiquer à Elasticsearch par quelle voie nous nous connectons au réseau local. Pour connaître toutes les voies configurées sur nos ordinateurs, installons premièrement le paquet `net-tools`.

Une fois le paquet installé, exécutons la commande suivante :

```
$ ifconfig
```

Vous aurez un résultat similaire à ceci :

```
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Boucle locale)
    RX packets 224384 bytes 11258905 (11.2 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 224384 bytes 11258905 (11.2 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlp2s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet XXX.XXX.XXX.XXX netmask 255.255.0.0 broadcast XXX.XXX.XXX.XXX
    inet6 XX:XX:XX:XX:XX prefixlen 64 scopeid 0x20<link>
    inet6 XX:XX:XX:XX:XX prefixlen 64 scopeid 0x20<link>
    ether XX:XX:XX:XX:XX txqueuelen 1000 (Ethernet)
    RX packets 11339176 bytes 3695244509 (3.6 GB)
    RX errors 0 dropped 0 overruns 0 frame 11406957
    TX packets 887855 bytes 119138027 (119.1 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 17
```

Selon les configurations de mes cartes réseaux, je dispose des sorties `wlp2s0`, `virbr0`, `lo` et `eno1`. Il s'agit des noms des ports par lesquels je peux me connecter à un réseau. Ces noms peuvent varier et/ou peuvent ne pas être tous présents :

- `wlp2s0` : il s'agit de la configuration de ma carte wifi ;
- `lo` : il s'agit du localhost, un hôte virtuel qui sert de connectivité interne ;
- `eno1` : il s'agit de la configuration de ma carte Ethernet.

Si vous disposez d'une carte wifi sur votre ordinateur, et si vous avez activé le wifi, vous devriez voir sa configuration apparaître. Si vous n'avez pas fait de modification particulière sur celle-ci, comme changer le nom explicitement, elle devrait commencer par « wl » pour wireless (par exemple : wlan0, wlan1, wlps2s0, etc.).

Il en est de même pour la configuration de la carte Ethernet. En général, c'est la première sur la liste, et elle commence par un « e » pour Ethernet (par exemple, eth0, eth1, en0, etc.).

Ainsi, par exemple, la configuration réseau de ma carte wifi est :

```
wlp2s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet XXX.XXX.XXX.XXX netmask 255.255.0.0 broadcast XXX.XXX.XXX.XXX
    inet6 XX:XX:XX:XX:XX prefixlen 64 scopeid 0x20<link>
    inet6 XX:XX:XX:XX:XX prefixlen 64 scopeid 0x20<link>
    ether XX:XX:XX:XX:XX txqueuelen 1000 (Ethernet)
    RX packets 11339176 bytes 3695244509 (3.6 GB)
    RX errors 0 dropped 0 overruns 0 frame 11406957
    TX packets 887855 bytes 119138027 (119.1 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 17
```




Pour indiquer à Elasticsearch notre identifiant sur le réseau local, il faut utiliser l'adresse IP de la carte réseau par laquelle nous sommes connectés à l'aide de la ligne suivante :

```
network.host: "<adresse_ip>"
```

La configuration de votre carte réseau vous permet de connaître votre adresse IP (information indiquée en rouge dans l'exemple de configuration ci-dessus).

Maintenant, nous allons permettre à nos nœuds de pouvoir se contacter dans le réseau. Pour cela, nous devons indiquer à chaque nœud l'adresse IP de son collègue. Pour le nœud 1, nous devons lui transmettre l'adresse IP du nœud 2, et réciproquement.

Dans ma configuration actuelle, le nœud 1 possède l'adresse IP 192.168.43.99, et le nœud 2 possède l'adresse IP 192.168.43.53.

Sachant cela, dans le fichier de configuration du master ajoutons donc la ligne suivante :

```
discovery.seed_hosts: ["192.168.43.53"]
```

Dans le fichier de configuration du nœud data, ajoutons la ligne suivante :

```
discovery.seed_hosts: ["192.168.43.99"]
```

Attention à mettre les bonnes adresses de votre côté.

Enfin, il est important d'indiquer dans le cluster les différents nœuds qui souhaitent devenir master. Dans notre cas, nous n'avons qu'un seul nœud. Pour le présenter comme candidat dans le cluster, ajoutons la ligne suivante dans le code de configuration du nœud master (il n'est pas nécessaire de l'ajouter dans le nœud data) :

```
cluster.initial_master_nodes: ["192.168.43.99"]
```

N'oubliez pas de remplacer l'adresse IP 192.168.43.99 par la vraie adresse IP de votre nœud master.

Les configurations des différents nœuds sont prêtes. Pour récapituler, à la fin du fichier de configuration du premier nœud, nous avons quelque chose de similaire à ceci :

```
cluster.name: mycluster
node.name: node-1
node.master: true
node.data: true
node.remote_cluster_client: false
cluster.routing.allocation.disk.threshold_enabled: false

network.host: "192.168.43.99"
discovery.seed_hosts: ["192.168.43.53"]
cluster.initial_master_nodes:
["192.168.43.99"]
```

Et dans le second nœud, nous avons ceci :

```
cluster.name: mycluster
node.name: node-2
node.master: false
node.data: true
node.remote_cluster_client: false
cluster.routing.allocation.disk.threshold_enabled: false

network.host: "192.168.43.53"
discovery.seed_hosts: ["192.168.43.99"]
```

Les configurations sont finies au niveau du cluster, toutefois il y a 2 dernières configurations à faire au niveau du système. Il s'agit du nombre maximum de fichiers ouvrables et de la taille maximale de la mémoire virtuelle de votre ordinateur.

Premièrement, vous devez modifier le nombre maximum de fichiers ouverts à 65536. Cette configuration se fait dans le fichier [/etc/security/limits.conf](#). Vous devez ouvrir ce fichier avec votre éditeur de texte puis le modifier. Comme il s'agit d'un fichier système, il vous faut les droits nécessaires, donc nous allons l'ouvrir en root avec la commande suivante :

```
$ sudo gedit /etc/security/limits.conf
```


Nous allons ajouter pour cela à la fin du fichier le code suivant :

```
<nom_du_compte> -  
nofile 65536
```

Où `<nom_du_compte>` est le nom du compte utilisateur sur lequel vous êtes connecté.

Cette instruction indique que pour le compte `<nom_du_compte>`, le nombre maximum de fichiers ouverts est 65536. Une fois que c'est fait, enregistrez le fichier et fermez gedit.

Passons à la seconde configuration, il s'agit de la taille de la mémoire virtuelle du système.

MÉMOIRE VIRTUELLE

En système d'exploitation, la mémoire physique (la RAM et éventuellement le disque dur) n'est pas directement utilisée par les programmes, car ceux-ci varient en fonction de plusieurs paramètres comme le fabricant, le principe de fonctionnement, etc.

Pour éviter que les développeurs se cassent la tête là-dessus, les systèmes d'exploitation intègrent la notion de mémoire virtuelle qui est une représentation de la mémoire physique, c'est cette mémoire qui est fournie aux programmes pour manipulation, et quant à la mémoire physique, c'est le système qui de manière transparente gère la correspondance entre la mémoire physique et la mémoire virtuelle.

La configuration se fait dans le fichier `/etc/sysctl.conf`.

Ainsi, nous allons à nouveau ouvrir notre fichier en mode root avec notre éditeur :

```
sudo gedit /etc/sysctl.conf
```

Nous allons ajouter la ligne suivante comme recommandé dans la documentation d'Elasticsearch :

```
vm.max_map_count=262144
```

Une fois que c'est fait, redémarrez vos ordinateurs. Après avoir relancé le système, assurez-vous que vos adresses IP sont restées inchangées. Si elles ont varié, reconfigurez les fichiers de configuration en conséquence.

Une fois que tout est OK, lancez la commande suivante sur les deux ordinateurs :

```
$ elasticsearch
```

Vous aurez alors un résultat similaire dans le terminal :

```
[2020-06-14T01:57:44,491][INFO ][o.e.e.NodeEnvironment ] [node-1] using [1] data paths, mounts [[/ (/dev/sda6)], net usable _space [13gb], net total _space [150.5gb], types [ext4]  
[2020-06-14T01:57:44,494][INFO ][o.e.e.NodeEnvironment ] [node-1] heap size [1.9gb], compressed ordinary object pointers [true]  
[2020-06-14T01:57:44,580][INFO ][o.e.n.Node ] [node-1] node name [node-1], node ID [QuFQD-0JQd6mR5BdMpLwmg], cluster name [mycluster]  
[2020-06-14T01:57:44,581][INFO ][o.e.n.Node ] [node-1] version[7.7.1], pid[30007], build[default/tar/ad56dce891c901a492bb1ee393f12dfff473a423/2020-05-28T16:30:01.040088Z], OS[Linux/4.15.0-101-generic/amd64], JVM[Oracle Corporation/Java HotSpot(TM) 64-Bit Server VM/1.8.0 _144/25.144-b01]  
[2020-06-14T01:57:44,582][INFO ][o.e.n.Node ] [node-1] JVM home [/home/xxx/opt/jdk-
```

Laissez les terminaux de chaque poste actif, Elasticsearch est ainsi lancé et les deux nœuds sont en communication.

2.3 Comment communiquer avec le cluster ?

Comme nous l'avons mentionné au début de l'article, Elasticsearch permet une communication à l'aide d'une API Rest. En d'autres termes, on communique avec elle à l'aide d'une requête http, et en retour elle nous renvoie des objets JSON (la plupart du temps).

Dans votre terminal exécutant Elasticsearch, ce dernier est en fait un serveur actif qui attend des requêtes de votre part. La communication réseau doit être maintenue, car les différents serveurs sur chaque nœud communiquent et se synchronisent constamment.

Il est aussi à noter que le serveur créé par Elasticsearch reçoit les requêtes par défaut sur le port 9200. Vous avez la possibilité de changer cela avec le code suivant :

```
http.port: <nouveau _port>
```

Il faut juste vous rassurer que le numéro de port que vous renseignez n'est pas un numéro de port réservé.

Dans une vraie application web ou mobile, il existe des API permettant de faire des requêtes vers des API REST. Dans notre cas, comme nous ne travaillons pas dans une application, nous allons utiliser le terminal et l'utilitaire `curl` qui permet de faire des requêtes http facilement à partir de celui-ci.

Faisons notre première communication avec Elasticsearch à l'aide de la commande suivante :

```
$ curl -X GET "192.168.43.99:9200/_cluster/health?pretty"
```

N'oubliez pas de prendre le soin de remplacer l'adresse IP de cette requête, par celle de votre nœud master.

Lorsque vous exécutez cette requête, vous obtenez en sortie le résultat suivant :

```
{
  "cluster_name" : "mycluster",
  "status" : "green",
  "timed_out" : false,
  "number_of_nodes" : 2,
  "number_of_data_nodes" : 2,
  "active_primary_shards" : 0,
  "active_shards" : 0,
  "relocating_shards" : 0,
  "initializing_shards" : 0,
  "unassigned_shards" : 0,
  "delayed_unassigned_shards" : 0,
  "number_of_pending_tasks" : 0,
  "number_of_in_flight_fetch" : 0,
  "task_max_waiting_in_queue_millis" : 0,
  "active_shards_percent_as_number" : 100.0
}
```

Ce résultat est un objet JSON, il nous donne l'état actuel du cluster. Voilà comment la communication avec Elasticsearch se passe. Dans la suite, nous verrons plusieurs autres commandes, et comment les construire.

Regardons attentivement la commande que nous avons exécutée. La partie de la requête qui nous intéresse est : `GET "192.168.43.99:9200/_cluster/health?pretty"`.

`GET` est utilisé lorsqu'on souhaite obtenir une information du destinataire, tandis que par exemple `POST` est utilisé pour transmettre des informations au destinataire. `DELETE` est utilisé lorsqu'on souhaite que le destinataire supprime des informations... vous avez compris comment ça marche !

Entre les guillemets, nous avons l'URL de la requête http. Observez dans l'URL ceci : `192.168.43.99:9200`. Ici, nous sommes en train de définir le destinataire de cette requête dans notre réseau local. L'adresse IP `192.168.43.99` permet de définir la carte réseau cible et par ricochet le nœud cible, tandis que le numéro de port `9200` permet de définir l'application cible, en l'occurrence Elasticsearch. Et enfin la dernière partie `/_cluster/health`, est tout simplement la route vers l'information que l'on souhaite avoir dans le serveur. Les informations dans une API REST s'obtiennent à travers une route précise.

Pour finir, parlons de `?pretty`. Dans une URL, tout ce qui suit le point d'interrogation `?` est un paramètre. Ainsi, dans notre requête, nous avons ajouté le paramètre `pretty` qui indique à Elasticsearch de retourner un résultat « joli », on entend par là un résultat bien affiché et bien lisible.

2.4 La santé du cluster

Lorsqu'un cluster vient d'être formé, il est important de connaître dans quel état de santé il se trouve. Il existe trois types d'états de santé pour un cluster :

- L'état Green : toutes les données du cluster sont distribuées et disponibles, de plus les répliques des fragments sont également présentes. En d'autres termes, c'est l'état de santé optimal.
- L'état Yellow : tous les fragments sont disponibles, mais des répliques n'ont pas encore été assignées ou sont indisponibles. Dans cet état, les données sont présentes, mais le risque de les perdre est plus grand.
- L'état Red : certains fragments de données sont indisponibles. Cet état est mauvais, car il manque des données.

La requête que nous avons exécutée dans la section précédente nous permet de déterminer l'état de santé du cluster. En effet, la route `/_cluster/health` permet d'obtenir les informations sur le cluster.

Le résultat de cette requête étant ceci :

```
{
  "cluster_name" : "mycluster",
  "status" : "green",
  "timed_out" : false,
  ...
}
```

Observez en particulier cette ligne :

```
"status" : "green",
```

Elle nous donne l'état du cluster. Elle nous dit notamment que toutes les données et les répliques sont disponibles. C'est un peu normal puisque nous n'avons encore aucune donnée.

De plus, à travers cette requête on peut obtenir d'autres informations comme le nombre de nœuds du cluster (`number_of_nodes`), le nombre de nœuds data (`number_of_data_nodes`), le nombre de fragments (`active_primary_shards`), etc.

Enfin, grâce à la route `/_nodes`, vous avez la possibilité d'obtenir un rendu des informations sur les nœuds du cluster.

La requête est la suivante :

```
$ curl -X GET "192.168.43.99:9200/_nodes?pretty"
```

Et voici un début de réponse du terminal :

```
{
  "_nodes" : {
    "total" : 2,
    "successful" : 2,
    "failed" : 0
  },
  "cluster_name" : "mycluster",
  "nodes" : {
    "QuFQD-0JQd6mR5BdMpLwmg" : {
      "name" : "node-1",
      "transport_address" :
"192.168.43.99:9300",
      "host" : "192.168.43.99",
      "ip" : "192.168.43.99",
      "version" : "7.7.1",
      "build_flavor" : "default",
```

```
"build_type" : "tar",
"build_hash" :
"ad56dce891c901a492bb1ee393f12dfff473a423",
...
```

Vous pouvez notamment voir au début le nombre de nœuds, le nombre de nœuds bien connectés, etc.

CONCLUSION

La conception d'une application nécessite de prendre en compte le nombre d'utilisateurs qui se connecteront à cette application, et ainsi d'utiliser une base de données appropriée. En effet, nous vivons dans une ère où apprendre à gérer des masses de données est primordial lorsqu'on souhaite développer des applications pour le grand public. Nous avons vu que les bases de données relationnelles bien qu'étant robustes, avaient des limites quand il s'agissait d'un immense public. L'utilisation d'une base de données NoSQL permettant de gérer des masses de données est donc une solution à toujours envisager. Ainsi, dans cet article nous avons étudié une solution parmi tant d'autres, il s'agit d'Elasticsearch qui, pour rappel est un SGBD NoSQL permettant du clustering et intégrant un moteur de recherche puissant. Elasticsearch s'offre donc à nous comme un choix de qualité pour accueillir à bras ouverts des big data. ■

RÉFÉRENCES

- [1] <https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html>
- [2] <https://fr.wikipedia.org/wiki/NoSQL>
- [3] https://en.wikipedia.org/wiki/Application_programming_interface
- [4] <https://en.wikipedia.org/wiki/Elasticsearch>
- [5] <https://engineering.bitnami.com/articles/add-scalable-high-performance-search-to-your-app-with-the-bitnami-elasticsearch-cluster.html>
- [6] <https://toungeafranck.com/chapitre-9-scaling-out/>



Chez votre marchand de journaux
et sur **www.ed-diamond.com**



en kiosque



FLIPBOOK HTML5

sur **www.ed-diamond.com**



CONNECT
LA DOCUMENTATION TECHNIQUE DES PROS DE L'IT

sur **connect.ed-diamond.com**

OPTIMISER SA CONFIGURATION de Bash

CÉLESTIN MATTE



NOUVEAU SUR BASH, VOUS CHERCHEZ À APPRENDRE DE NOUVELLES ASTUCES ? UTILISATEUR CONFIRMÉ, VOUS VOULEZ OPTIMISER ENCORE PLUS VOTRE CONFIGURATION ? CET ARTICLE EST POUR VOUS.

1. INTRODUCTION

Dans cet article, nous allons voir différentes astuces pour optimiser sa pratique de bash en fonction de ses besoins personnels. On supposera que vous connaissez déjà les rudiments de bash, son fonctionnement et ses raccourcis basiques. Utilisateurs avancés, ne fuyez pas ! Vous découvrirez certainement quelques options ou commandes utiles...

Le but de l'article est vraiment d'apprendre à faire sa propre configuration. Gardons à l'esprit le fait que des outils existent pour obtenir directement une configuration avancée **[Oh-my-bash]**.

Pour ne pas vous embêter à recopier manuellement les exemples, vous retrouverez tous les morceaux de code proposés dans l'article à l'adresse suivante : https://github.com/Perdu/lp_article_bash.

2. COMMANDES UTILES

Commençons par mentionner quelques commandes indispensables, soit parce qu'elles sont très utiles, soit parce qu'on s'en resserrera par la suite.

2.1 jobs

Je me dois d'introduire ici la commande `jobs`, car on va y revenir souvent dans la suite de l'article. Il est très utile d'apprendre à jongler entre plusieurs programmes dans un terminal avec celle-ci. Beaucoup de programmes peuvent être interrompus avec Ctrl-Z, ce qui les met en arrière-plan. `jobs` permet de lister ces programmes, `fg` de revenir aux précédents programmes interrompus, ou aux autres en donnant leur numéro (visible avec `jobs`) en argument. `bg` permet de continuer l'exécution indépendante d'un programme mis en arrière-plan, par exemple pour pouvoir continuer à utiliser un visualiseur de documents.

2.2 Options de cd

Vous savez vous déplacer avec `cd` ? Très bien ! Mais connaissez-vous tous ses arguments ? `cd` sans option permet de retourner directement à la racine de son `home`. `cd -` permet de retourner au dossier précédemment parcouru.

2.3 Fichiers temporaires

Sans doute connaissez-vous les sous-commandes, qu'on appelle avec la syntaxe `$(commande)` ou ``commande`` pour récupérer directe-

ment le résultat d'une commande en tant qu'argument d'une autre commande (particulièrement utiles dans les scripts pour affecter le résultat d'une commande à une variable). Mais savez-vous que de manière similaire, il est possible d'utiliser le résultat d'une commande en tant que fichier, avec la syntaxe `<(commande)` ? Cela peut éviter de créer manuellement des fichiers temporaires pour lancer une commande qui ne prend ses entrées que par des fichiers. Par exemple, pour comparer le résultat de deux scripts avec `diff` :

```
diff <(python un_script.py) <(python un_autre_script.py)
```

3. RACCOURCIS CLAVIER INDISPENSABLES

Vous savez vous déplacer dans la ligne de commandes, couper des mots, naviguer dans l'historique proche. Bien, mais après ? Voici quelques raccourcis qui font gagner de secondes précieuses au quotidien :

- Vous quittez votre terminal en fermant la fenêtre ? Peut-être tapez-vous `exit`, ou même avez-vous créé un alias pour fermer plus rapidement ? Sachez que la commande Ctrl-D envoie un caractère EOF (End-Of-File), qui indique la fin de la réception de caractères à lire. Autrement dit, en mode interactif, cela termine la session shell (si c'est le seul caractère de la ligne). Sachez que cette commande fonctionne avec de nombreux programmes en ligne de commandes : comme EOF est utilisé pour connaître la fin des entrées dans un script non-interactif, peu de programmes seront réticents à terminer de cette manière.
- Vous utilisez plusieurs commandes à la suite sur le même fichier ? Retenez bien ce raccourci : Ctrl-Maj-. (point). Cela colle le dernier argument de la commande précédente. Très utile !
- Marre de passer de la souris au clavier ? Shift+Insert colle les données du presse-papier primaire (celui rempli par une simple sélection à la souris) dans le terminal. Ce n'est certes pas un raccourci de bash lui-même et ce n'est pas supporté par tous les terminaux, mais c'est très utile.
- Alt-_ (souligné/underscore) annule la dernière action. Moins utile que dans un éditeur de texte, mais toujours bon à savoir !

4. .BASHRC : OPTIONS UTILES

Maintenant que nous avons vu quelques généralités, attardons-nous sur la configuration de bash lui-même, qui s'effectue dans le fichier `~/.bashrc` (ou `/etc/bashrc` pour une configuration générale appliquée à tous les utilisateurs).

Je passe sur la configuration de base, qui est déjà certainement fournie par votre distribution (lisez-la !), pour discuter directement d'astuces diverses.

4.1 Options utiles

Jaloux de votre collègue utilisant zsh qui frime avec son « globstar » ? Cette fonctionnalité permet d'utiliser « ** » comme super-wildcard permettant de chercher des correspondances dans tous les sous-dossiers, de manière récursive. Par exemple, `ls **/*.txt` affichera les fichiers .txt dans tous les sous-dossiers du dossier courant (et sous-sous-dossiers, etc.). Eh bien cette fonction existe dans bash ! Il suffit de l'activer :

```
shopt -s globstar
```

Ensuite, le `CDPATH` est un must. Dans cette variable, on peut ajouter des dossiers qui seront parcourus par bash en plus du dossier courant pour se déplacer avec `cd`. Par exemple, `CDPATH="/home/$USER"` permettra de se déplacer dans un sous-dossier de son `home` depuis n'importe où !

4.2 L'historique

L'historique est une partie indispensable de la ligne de commandes. Voici quelques configurations bien pratiques :

```
HISTCONTROL=ignoredups:ignorespace # ne pas conserver les
doublons et les lignes vides dans l'historique
HISTIGNORE="ls:[bf]g:cd _*:history*" # ne pas conserver de
commandes précises dans l'historique (à configurer)
HISTSIZE=1000000 # augmenter la taille maximale de
l'historique pour conserver toutes ses commandes
effectuées
```

Petite astuce supplémentaire : pour ne pas enregistrer une ligne dans l'historique, on peut rendre la variable `HISTFILE` vide (`HISTFILE=`), ou tout simplement débiter la ligne de commandes par un espace.

4.3 Le prompt

Le prompt est le début de la ligne de commandes. Par défaut, il a cette allure :

```
[user@machine dossier]$
```

Celle-ci peut se configurer. C'est là tout un art ! Par exemple, le mien ressemble à cela :

```
user:~[1]:)♥
```

Le nombre entre crochets sert à compter les jobs en arrière-plan. Le smiley indique le résultat de la précédente commande. Et un cœur, c'est quand même plus sympa qu'un dollar !

En exclusivité, je vous donne la définition complète, commentée :

```
# La fonction pour obtenir le nombre de programmes en
arrière-plan
nb_jobs() {
    nb_jobs=$(jobs -s | wc -l)
    if [ "$nb_jobs" -gt 0 ]
    then
        echo -n "[$nb_jobs]"
    fi
}
# Définition des couleurs et de la graisse (les crochets
évitent des problèmes de formatage)
couleur=31 # couleur spécifique à cette machine, à changer
pour chaque machine
col="\[\033[01;${couleur}m\]"
white='\[\033[00m\]'
grey='\[\033[01;90m\]'
bold="\[${tput bold}\]"
```



```

normal="\[${tput sgr0}\]"
# On récupère le code de retour de la commande précédente pour afficher un smiley triste en
cas d'erreur
smiley() {
    RC=$?
    [[ ${RC} == 0 ]] && echo -e ':' || echo -e ':'
}
# Et si on affichait un caractère différent en fin de prompt si on est en SSH sur une
machine distante ?
ssh_case() {
    if [ ! -z "${SSH_CLIENT}" ]
    then
        echo -e "♥"
    else
        echo -e "♥"
    fi
}
# La définition finale du prompt
PS1="$col\u$white:$col\w$grey\$(nb_jobs)$white$bold\$(smiley)$normal$white\$(ssh_case) "

```

N'hésitez pas à modifier tout cela selon votre propre convenance ! Au passage, je déconseille l'utilisation de caractères spéciaux ou de lignes trop longues, qui peuvent parfois poser quelques soucis.

5. ALIAS

Dans son fichier `.bashrc`, on peut configurer des alias, qui permettront d'appeler des commandes complexes en quelques caractères. Voici quelques alias parmi les plus utiles que j'ai pu définir en plusieurs années de configuration de bash :

```

alias ..='cd ..'
alias ai='sudo aptitude install' # ou le gestionnaire de paquets de votre distribution
alias c2p='xclip -o -selection clipboard | xclip' # passe les données du presse-papier clipboard
(copiées avec Ctrl-C) vers le presse-papier primaire (clic-roulette)
alias dl='youtube-dl -i --audio-format mp3 --extract-audio -o '\''%(title)s.%(ext)s'\'' # télécharger
une vidéo sur youtube et la convertir en mp3
alias dt='dmesg | tail' # voir les derniers messages du noyau
alias e='emacs -nw' # ouvrir un fichier avec emacs dans le terminal
alias get_cert_fingerprint='openssl x509 -noout -fingerprint -in' # extraire l'empreinte d'un
certificat
alias get_ssh_fingerprint='awk '\''{print $2}'\'' /etc/ssh/ssh_host_ecdsa_key.pub | base64 -d |
sha256sum -b | awk '\''{print }'\'' | xxd -r -p | base64' # obtenir l'empreinte du serveur SSH de la
machine en cours
alias jprint='python -mjson.tool' # afficher un fichier JSON de manière lisible
alias maj='. ~/.bashrc' # recharger bashrc (dont les alias) après une modification
alias p8='ping 8.8.8.8' # pour les tests réseau
alias random_pass='< /dev/urandom tr -dc _A-Z-a-z-0-9 | head -c8 ; echo' # générer un mot de
passe aléatoire
alias se='sudoedit' # éditer un fichier root sans faire tourner son éditeur en root
alias sus='systemctl suspend' # mettre la machine en veille
alias unixtime='date +%s' # obtenir le timestamp unix actuel
alias zkill='kill -9 $(jobs -lp)' # tuer les jobs en arrière-plan (utiles si bloqués)

```


C'est évidemment une sélection personnelle, et chacun trouvera divers alias plus ou moins utiles.

On peut aussi saisir l'opportunité pour gérer ses erreurs de frappe les plus courantes comme des alias pour éviter de se faire gronder par bash à chaque fois, par exemple :

```
alias gf='fg'
alias gti='git'
# etc.
```

Attention : les alias peuvent casser l'autocomplétion bash (quand vous utilisez TAB pour terminer une commande). Il faut bidouiller un peu pour la réparer avec `complete`. Comme c'est du bidouillage plutôt bancal, je vous mets mes solutions sur le dépôt posté plus haut, sans les détailler ici. Pour donner tout de même un exemple simple et montrer l'utilité de `complete` au-delà de ce problème, voilà comment on peut (re)créer la complétion pour des lecteurs PDF et leurs alias en ne permettant de compléter que les noms de fichiers PDF et les répertoires :

```
alias z='zathura'
alias ev='evince'
# compléter des lecteurs de pdf avec les
# fichiers pdf et les répertoires seulement
complete -f -X '!*.pdf' -o plusdirs
evince zathura ev z
```

5.1 Fonctions

Contrairement à d'autres shells, Bash ne permet pas d'insérer des arguments à l'intérieur d'un alias. Pour cela, il faudra utiliser une fonction. Par exemple :

```
function unix_to_date() {
    date -d @"$1"
}
```

Cet exemple ne fonctionnerait pas avec un simple alias (`alias unix_to_date='date -d @'`), car l'appel de ce dernier nécessite l'ajout d'un espace après l'arobase, ce qui ne respecte pas la syntaxe de `date`.

On peut évidemment définir des fonctions plus complexes, avec notamment des arguments. Comme ce n'est pas le but de cet article, je vous invite à lire un tutoriel sur la programmation bash pour en savoir plus, par exemple [\[prog-bash\]](#).

6. COULEURS

Une bonne manière de s'y retrouver plus facilement dans son environnement en ligne de commandes est d'activer les couleurs dans les programmes où c'est possible. Les configurations par défaut sont souvent en monochrome, pour éviter les problèmes de compatibilité éventuels. Si on peut se le permettre, autant les activer !

Dans bash lui-même, la gestion des couleurs est un peu spéciale. On écrit en couleur en faisant précéder le texte de la commande suivante : `"\033[01;${couleur}m"`. À la place de la variable `$couleur`, on indique un nombre de 0 à 255, par exemple 0 pour le blanc, 31 pour le rouge ou 32 pour le vert (on trouve facilement des tableaux de correspondance en ligne [\[couleurs\]](#)). Tout le texte suivant sera de la couleur choisie, jusqu'à ce qu'on en change. Savoir cela peut être utile pour, par exemple, configurer un prompt coloré (relisez le paragraphe 4.3, c'est ce que j'ai fait !).

Pour certaines commandes, la couleur s'active avec une option. On peut créer un alias pour l'activer par défaut, ce qui est parfois fait dans les fichiers `.bashrc` fournis par défaut par la distribution, par exemple sous Ubuntu. L'exemple suivant en est d'ailleurs tiré :

```
if [ -x /usr/bin/dircolors ]; then
    test -r ~/.dircolors && eval
    "$(dircolors -b ~/.dircolors)" || eval
    "$(dircolors -b)"
    alias ls='ls --color=auto'
    alias grep='grep --color=auto'
    alias fgrep='fgrep --color=auto'
    alias egrep='egrep --color=auto'
    alias diff='diff --color=auto'
fi
```

On peut aussi activer des couleurs pour `man` avec l'astuce suivante, tirée du wiki d'Arch Linux [\[arch-wiki\]](#) :

```
# Couleurs pour less, utilisées notamment
# par man
man() {
    LESS _TERMCAP _md='${e[01;31m' \
    LESS _TERMCAP _me='${e[0m' \
    LESS _TERMCAP _se='${e[0m' \
    LESS _TERMCAP _so='${e[01;44;33m' \
```



```
LESS _TERMCAP _ue=$'\e[0m' \
LESS _TERMCAP _us=$'\e[01;32m' \
command man "$@"
}
```

Pour `diff`, on peut installer le paquet `colordiff`, un petit script perl qui s'occupera de rajouter de la couleur à la sortie de la commande.

Enfin, votre gestionnaire de paquets doit peut-être être configuré pour utiliser la couleur, par exemple sur le résultat des recherches. Par exemple, sous Arch Linux, il faut décommenter la ligne `Color` dans `/etc/pacman.conf`. Pour `apt`, les possibilités de configuration sont plus minimes, puisque seule la barre de chargement peut être améliorée. Pour cela, il faut créer un fichier `/etc/apt/apt.conf.d/99colorbar` contenant les lignes suivantes :

```
Dpkg::Progress-Fancy "1";
APT::Color "1";
```

Notons au passage que la sortie de `git` est colorée par défaut, mais si au contraire vous souhaitez désactiver ce comportement, vous pouvez le faire avec la commande suivante :

```
git config --global color.ui false
```

7. .INPUTRC

Alors, préparez-vous. Dans cette section, on va faire de la magie !

`.inputrc` est le fichier de configuration de la librairie `readline`, utilisée par de nombreux logiciels en ligne de commandes (dont `bash`) pour lire les entrées clavier. C'est un fichier intéressant pour effectuer des configurations surprenantes, bien plus avancées que ce qu'on a vu dans les sections précédentes.

7.1 Un exemple pour vous convaincre

`.inputrc` permet de déclarer des raccourcis clavier effectuant une combinaison de commandes. On peut donc invoquer d'autres raccourcis dans un raccourci ! Par exemple, cette ligne à l'aspect barbare associe à `Alt-g` l'effacement de la ligne en cours (`Ctrl-A` puis

`Ctrl-K`), l'écriture de la commande `grep -r ""`, puis le déplacement du curseur entre les doubles quotes (plusieurs `Ctrl-B`). Rien que ça !

```
"\eg": '\C-a\C-kgrep -r "" \C-b\C-b\C-b'
```

7.2 Obtenir directement le fichier le plus récent

Et si on importait des informations contextuelles directement dans la ligne de commandes ? Par exemple, coller directement le nom du fichier le plus récent du dossier en cours, ça vous fait rêver ? Moi, oui ! C'est extrêmement utile ! Mais par quels arcanes obscurs peut-on obtenir ce résultat ?

Premièrement, on va définir un alias permettant d'obtenir le fichier le plus récent du dossier en cours dans notre `~/.bashrc` :

```
alias latest='/bin/ls -t -l -d * | head -n 1'
```

Ensuite, c'est là que la magie opère : on va utiliser l'expansion shell pour obtenir notre résultat. Dans notre `~/.inputrc`, on va définir la commande suivante :

```
"\e:": '\\\$(latest)\\\' \e\C-e'
```

Celle-ci lie à un raccourci clavier (ici `Alt-:`), une suite d'entrées clavier qui colle notre alias en tant que sous-commande (`$(latest)`), puis lance une expansion shell pour l'exécuter et la remplacer par son résultat (`Ctrl-Alt-E`). On ajoute des quotes (qu'on échappe) pour que le nom du fichier soit utilisable directement même s'il contient des espaces, et le tour est joué.

Attention : l'expansion shell résout tous les alias et sous-commandes. Si vous en avez d'autres dans votre ligne de commandes, ils seront également exécutés !

7.3 Naviguer dans l'historique

Vous aimez retrouver rapidement des commandes dans votre historique, mais vous trouvez l'utilisation de `Ctrl-R` lourde ? Essayez ceci :

```
"\e[1;5A": history-search-backward
"\e[1;5B": history-search-forward
```


À l'aide de Ctrl-flèche haut, vous obtiendrez directement la dernière commande commençant de la même manière que ce que vous avez déjà écrit ! Plus besoin d'effacer la ligne pour lancer Ctrl-R !

Note : en fonction des environnements, il faut parfois adapter l'appel à flèche haut et bas, par exemple avec `"\e[A"`. Pour trouver quel est votre cas, tapez les combinaisons de touches dans [sed -n l](#).

7.4 D'autres exemples

Là encore, je vous présente une sélection des raccourcis les plus utiles que j'ai pu inventer au cours des années :

```
"\ej": "\C-a\C-kjobs -sl\n" # Voir les jobs en arrière-plan
"\es": "\C-p\C-asudo \C-e" # Ajouter "sudo" à la dernière commande
"\ep": "\C-a\C-kps aux | grep -i ""\C-b" # Chercher un processus dans les processus en cours
"\ei": "\C-a\C-kssh -t machine LANG=fr_FR.UTF-8 tmux attach\C-m" # Se connecter en SSH à une machine
et ouvrir sa session tmux en un seul raccourci !
"\eF": "\C-a\C-kfind . -iname "***" | tee >(tr -d '\n' | xclip)\C-a\ef\ef\C-f\C-f\C-f" # Chercher des
fichiers par nom, et coller les résultats dans le presse-papier au passage
"\eh": "\C-a\C-khistory | grep '\C-b' # Chercher une commande dans son historique
"\em": "\C-a\C-kmake\C-m" # Lancer make avec Alt-m. Très agréable !
"\ea": ' \\\$(ls | dmenu)\\\e\C-e' # Utiliser dmenu pour sélectionner un fichier dans le dossier
courant. Pratique pour récupérer des noms de fichiers complexes ou contenant des caractères spéciaux
```

8. POUR ALLER PLUS LOIN

Voilà, nous avons vu comment améliorer notre environnement bash grâce à l'utilisation des alias, couleurs, raccourcis clavier peu connus, du fichier de configuration `.inputrc`... Mais tout cela n'était qu'un survol ! On peut toujours plus personnaliser sa configuration.

Je trouve cela utile de synchroniser sa configuration bash entre plusieurs machines en utilisant git, ce que je fais avec une méthode expliquée dans cet article [\[linuxfr-git\]](#). Ainsi, on se retrouve à un [git clone](#) d'importer notre configuration personnalisée sur une nouvelle machine !

Pour creuser plus, il existe de nombreux logiciels particulièrement adaptés à un environnement en ligne de commandes. Par exemple, apprendre à maîtriser et configurer tmux est un must, car ce logiciel permet entre autres de jongler entre des terminaux virtuels dans le même terminal (donc sans avoir à gérer une palanquée de fenêtres ou onglets divers). Cela permet aussi de créer des sessions persistantes, ce qui est très pratique quand on accède à une machine via SSH. Finalement, il existe de nombreux logiciels avec une interface ncurses, qui permettent d'interagir avec le logiciel sans sortir du terminal. On citera par exemple ranger pour la gestion des fichiers, ou encore vim ou emacs (avec son option `-nw`) pour les éditeurs de texte. Utiles en SSH, pour éviter le passage fréquent du clavier à la souris, ou par goût, tout simplement ! ■

RÉFÉRENCES

[\[Oh-my-bash\]](#) <https://ohmybash.nntoan.com/>, inspiré de son équivalent pour zsh

[\[Linuxfr-git\]](#) <https://linuxfr.org/users/perdu/journaux/gerer-ses-fichiers-de-config-avec-git>

[\[prog-bash\]](#) https://doc.ubuntu-fr.org/tutoriel/script_shell

[\[couleurs\]](#) https://misc.flogisoft.com/bash/tip_colors_and_formatting

[\[arch-wiki\]](#) https://wiki.archlinux.org/index.php/Color_output_in_console#Using_less



MISC

LE MAGAZINE DE LA CYBERSÉCURITÉ
OFFENSIVE ET DÉFENSIVE

N°114

MARS / AVRIL 2021

FRANCE MÉTRO : 9,90 € - BELUX : 10,90 € - CH : 15,80 CHF
ESP/IT/PORT-CONT : 10,90 € - DOM/S : 10,90 €
TUN : 28 TND - MAR : 110 MAD - CAN : 17,90 \$CAD

L 19018 - 114 - F : 9,90 € - RD



RÉSEAU :
OPENFLOW / SCAPY

Contourner les règles de
filtrage réseau en exploitant les
faiblesses des contrôleurs SDN

CODE :
DÉVELOPPEMENT / LEAK

Découvrez les bonnes
pratiques pour gérer les
secrets dans les dépôts Git

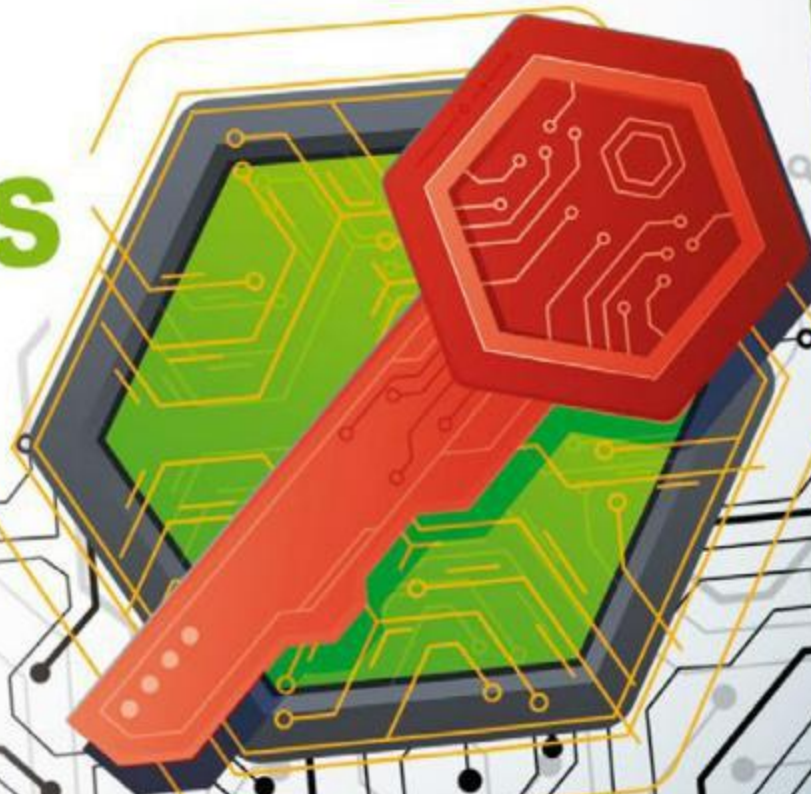
ORGANISATION :
CYBERMENACE / PROSPECTIVE

Pouvons-nous éviter
un Pearl Harbor
numérique ?

DOSSIER

PUCES SÉCURISÉES À LA DÉCOUVERTE DE LA SÉCURITÉ MATÉRIELLE

- 1 - SE, TPM, TEE : tour d'horizon
des environnements sécurisés
- 2 - Découverte de la puce sécurisée
de Google Titan M
- 3 - Les protections des Secure
Elements contre les attaques
physiques



FORENSIC CORNER

Analyse du bruit de
fond d'Internet avec un
honeypot

EXPLOIT CORNER

Impact de la vulnérabilité
Spectre sur la conception
d'applications web

PENTEST CORNER

Analyse du standard
HITAG2 pour déverrouiller
un véhicule



en kiosque



FLIPBOOK HTML5

sur www.ed-diamond.com



CONNECT

LA DOCUMENTATION TECHNIQUE DES PROS DE L'IT

sur connect.ed-diamond.com

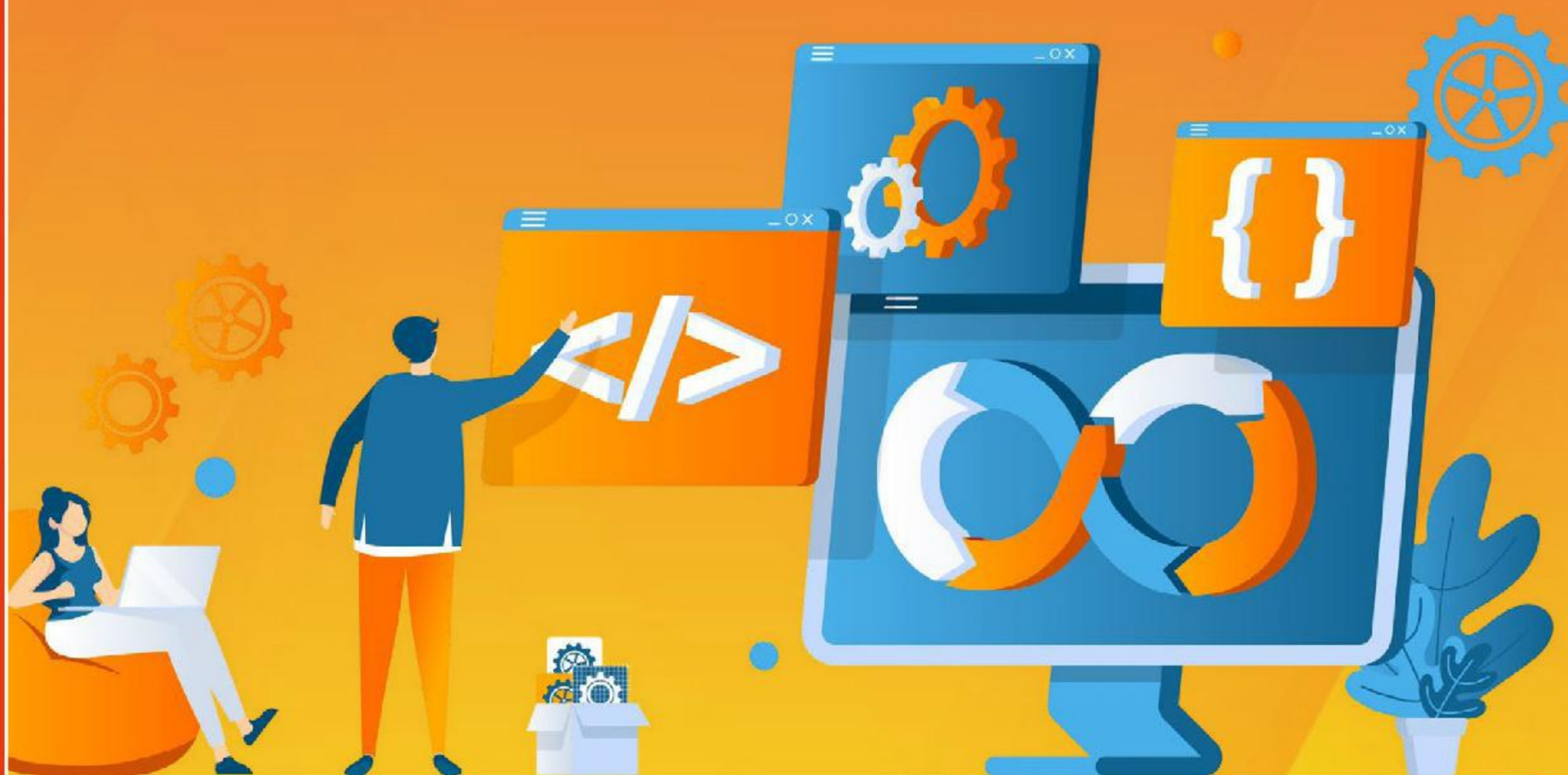
Chez votre marchand de journaux
et sur www.ed-diamond.com

LE DEVOPS

en pratique

ISSAM MEJRI

AUJOURD'HUI, LE TERME DEVOPS EST AU CŒUR DE LA TRANSFORMATION DIGITALE. EN EFFET, LORSQU'ON ÉVOQUE L'AGILITÉ DANS LE CYCLE DE VIE DE DÉVELOPPEMENT LOGICIEL, ON EST AMENÉ À PRATIQUER DES MÉTHODOLOGIES PERMETTANT D'ACCÉLÉRER D'UNE PART LE PROCESSUS DE DÉVELOPPEMENT ET DE FIABILISER LA PRODUCTION LOGICIELLE D'UN AUTRE CÔTÉ. PARMI CES MÉTHODOLOGIES (SCRUM, KANBAN...), LE DEVOPS EST CERTAINEMENT CELLE QUI A ÉTÉ ADOPTÉ PAR LA PLUPART DES ENTREPRISES AYANT COMME ACTIVITÉ PRINCIPALE OU ANNEXE, LA DIGITALISATION DE LEUR SYSTÈME D'INFORMATION POUR FAIRE FACE AU DÉFI QUE REPRÉSENTE L'ESSOR DES TECHNOLOGIES NUMÉRIQUES. DANS CET ARTICLE, NOUS FERONS UN PREMIER TOUR D'HORIZON DE LA PARTIE OUTILLAGE DU DEVOPS. IL EXISTE UNE PANOPLIE D'OUTILS ET IL EST IMPOSSIBLE DE LES ÉVOQUER TOUS DANS UN SEUL ARTICLE, CAR CHAQUE OUTIL NÉCESSITERAIT UN OUVRAGE À PART ENTÈRE.



Le DevOps est principalement une culture de collaboration entre les différents protagonistes qui interviennent sur un projet, mais aussi un ensemble de techniques et d'outils non standardisés qui ont un impact direct sur la façon de concevoir, développer, déployer les applications. Je vais choisir pour chaque technique, un outil qui correspond le mieux et dont la prise en main est simple.

L'intégration continue et le déploiement continu se feront avec Jenkins. Pour la création des machines et des environnements de développement et de tests, j'ai choisi Vagrant. Pour la création des images et serveurs immutables, ce sera Packer, et finalement pour les environnements à base de conteneurs ce sera Docker.

1. CRÉATION DE MACHINES DE TEST PAR VAGRANT

Vagrant est un outil permettant rapidement de déployer des environnements de développement et de tests. L'avantage est de rendre les équipes sur un projet plus autonomes. Autrement dit, les équipes système livrent un fichier contenant la description de l'environnement et les développeurs exécutent une seule commande pour créer les machines cibles, on parle de l'Infrastructure As Code ; Vagrant est l'un des outils qui implémente ce concept.

Vous l'avez compris, Vagrant permet de créer des serveurs, et de déployer les outils nécessaires

pour l'environnement de développement. Plus encore, il permet de créer des environnements à l'identique, chose très importante lorsque les équipes travaillent sur le même projet.

Vagrant est géré par un seul fichier de configuration, le Vagrantfile avec un V majuscule, et d'autres fichiers annexes optionnels pour déployer et provisionner des ressources, tel que l'installation des outils ou une configuration système nécessaire.

1.1 Installation de Vagrant

Commençons par installer Vagrant. Il faut noter que Vagrant nécessite la présence d'un logiciel de virtualisation pour pouvoir créer les machines (Virtual Machine dans le contexte d'une infrastructure virtualisée). Dans mon cas, je travaille sur une machine Linux Ubuntu 18.04 LTS.

```
root@vagrant-packer:~# apt update -y
root@vagrant-packer:~# apt install virtualbox
root@vagrant-packer:~# curl -O https://releases.hashicorp.com/vagrant/2.2.9/
vagrant_2.2.9_x86_64.deb
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 40.9M  100 40.9M    0     0  113M      0 --:--:-- --:--:-- --:--:-- 113M
root@vagrant-packer:~# apt install ./vagrant_2.2.9_x86_64.deb
root@vagrant-packer:~# vagrant --version
Vagrant 2.2.9
```

1.2 Création du projet Vagrant

Pour créer notre environnement, il faut commencer par créer un répertoire pour notre projet et un Vagrantfile. Le Vagrantfile correspond au code de notre environnement provisionné, il décrit les ressources pour notre machine, ainsi que les éventuelles configurations et logiciels déployés. Vagrant, pour cela, fait appel à des outils de gestion de configuration tels qu'Ansible, Puppet ou encore Chef.

Nous allons créer une machine de type Ubuntu, avec l'installation du serveur web nginx et de l'outil htop pour la supervision des ressources.

1.2.1 Création du Vagrantfile

```
root@vagrant-packer:~# mkdir article
root@vagrant-packer:~# cd article
root@vagrant-packer:~/article#
root@vagrant-packer:~/article# vagrant init ubuntu/xenial64
==> vagrant: A new version of Vagrant is available: 2.2.14
(installed version: 2.2.9)!
==> vagrant: To upgrade visit: https://www.vagrantup.com/
downloads.html
```


A `Vagrantfile` has been placed in this directory. You are now ready to `vagrant up` your first virtual environment! Please read the comments in the Vagrantfile as well as documentation on `vagrantup.com` for more information on using Vagrant.

Ces commandes permettent de créer un Vagrantfile dans le répertoire du projet article.

Voici à quoi correspond notre Vagrantfile :

```
Vagrant.configure("2") do |config|

  # Choix du système à installer
  config.vm.box = "ubuntu/xenial64"

  # Create a forwarded port mapping which
  # allows access to a specific port
  # within the machine from a port on the
  # host machine. In the example below,
  # accessing "localhost:8080" will access
  # port 80 on the guest machine.
  # NOTE: This will enable public access to
  # the opened port
  config.vm.network "forwarded__port", guest:
  80, host: 80

  # Création du réseau public pour exposer
  # notre VM à l'extérieur (Bridge)
  config.vm.network "public__network"

  # Provider-specific configuration so you can
  # fine-tune various
  # backing providers for Vagrant. These
  # expose provider-specific options.

  config.vm.provider "virtualbox" do |vb|
    # # Display the VirtualBox GUI when booting
    # the machine
    # vb.gui = true
    #
    # # Notre Vm aura une capacité mémoire de
    # 512 Mo
    vb.memory = "512"
  end
  # Installation de Nginx et de l'outil htop
  config.vm.provision "shell", inline: <<-SHELL
  apt-get update
  apt-get install -y nginx
  apt-get install -y htop
  SHELL
end
```

1.2.2 Création de la machine

```
root@vagrant-packer:~/article# vagrant up
Bringing machine 'default' up with
'virtualbox' provider...
==> default: Checking if box 'ubuntu/
xenial64' version '20210127.0.0' is up to
date...
==> default: Running provisioner: shell...
default: Running: inline script
```

Vagrant réalisera les tâches suivantes :

- téléchargement du box Ubuntu ;
- création de la VM dans VirtualBox avec les ressources et le réseau associé à notre Vagrantfile ;
- création des clés SSH pour la connexion à la machine ;
- installation des paquets nginx et htop.

1.2.3 Accès à la machine Vagrant

Pour se connecter à la machine, il faut passer `ssh` à la commande `vagrant` :

```
vagrant ssh
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux
4.4.0-201-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.
canonical.com
 * Support:       https://ubuntu.com/
advantage

1 package can be updated.
1 of these updates is a security update.
To see these additional updates run: apt
list --upgradable

New release '18.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun Jan 31 19:06:00 2021 from
10.0.2.2
```

C'est bon, je suis bien connecté à ma VM et je peux accéder à mes outils installés.



Vérifions que nginx et httpd sont installés :

```
vagrant@ubuntu-xenial:~$ systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2021-01-31 19:08:30 UTC; 12min ago
     Main PID: 2550 (nginx)
       Tasks: 3
      Memory: 3.7M
         CPU: 639ms
    CGroup: /system.slice/nginx.service
            └─2550 nginx: master process /usr/sbin/nginx -g daemon on; master _
               process on
                └─2551 nginx: worker process
                  └─2552 nginx: worker process

vagrant@ubuntu-xenial:~$ httpd
```

Pour vérifier les box disponibles dans le cache local, on exécute la commande suivante :

```
root@vagrant-packer:~/article# vagrant box list
ubuntu/xenial64 (virtualbox, 20210127.0.0)
```

Une box est une image d'un OS au sens Vagrant. Une fois que cette image ou box est installée, elle peut être réutilisée dans plusieurs projets sans pour autant être téléchargée à nouveau.

Pour arrêter la machine :

```
root@vagrant-packer:~/article# vagrant box list
ubuntu/xenial64 (virtualbox, 20210127.0.0)
root@vagrant-packer:~/article# vagrant halt
==> default: Attempting graceful shutdown of VM...
```

Enfin, pour détruire le projet :

```
root@vagrant-packer:~/article# vagrant halt
==> default: Attempting graceful shutdown of VM...
root@vagrant-packer:~/article# vagrant destroy
   default: Are you sure you want to destroy the 'default' VM? [y/N] y
==> default: Destroying VM and associated drives...
```

2. CRÉATION D'IMAGES AVEC PACKER

Alors que Vagrant permet de créer des machines, Packer permet d'encapsuler les images des machines. Avec Packer, on définit notre type de machine cible avec la configuration souhaitée et la plateforme cible. La plateforme cible correspondra à l'environnement sur lequel s'exécutera la machine image (VirtualBox, VMware, Cloud AWS, DigitalOcean...).

Dans notre cas, nous allons créer une image de type CentOS, stocker l'image dans le conteneur des images du Cloud, puis construire la machine à partir de cette image.

2.1 Installation de Packer

Commençons par télécharger Packer depuis le site officiel Hashicorp :

```
root@:~# curl -O https://releases.hashicorp.com/packer/0.12.2/packer_0.12.2_linux_amd64.zip
```

Si l'outil zip n'est pas installé sur votre machine, il faut l'installer via le gestionnaire APT :

```
root@vagrant-packer:~# sudo apt install -y unzip
```

L'installation s'est bien déroulée, vérifions qu'on peut lancer la commande `packer` :

```
root@vagrant-packer:~# packer version
Packer v0.12.2

Your version of Packer is out of date! The latest version is 1.6.6. You can update by downloading from www.packer.io
```


2.2 Configuration de Packer

Packer détient un fichier de configuration appelé `template` permettant de définir les propriétés de l'image. C'est un fichier JSON avec plusieurs sections. Voici le contenu de notre `template` :

```
{
  "builders": [
    {
      "type": "digitalocean",
      "ssh_username": "root",
      "api_token": "mon_token_digitalocean",
      "image": "centos-7-6-x64",
      "region": "ams3",
      "size": "1024mb"
    }
  ]
}
```

Le Builder permet de spécifier quel type de build utiliser pour créer l'image. Dans notre cas, c'est un builder de type DigitalOcean, ce qui permet ensuite de créer l'image dans le Cloud DigitalOcean.

L'image sera créée dans la région Amsterdam. Il faut créer un token pour pouvoir communiquer avec l'API DigitalOcean.

Une autre section très importante est la section `provisioners`, ajoutant de la configuration à l'image :

```
"provisioners": [
  {
    "type": "shell",
    "inline": [
      "sleep 30",
      "yum install -y redis-server",
      "yum install -y epel-release",
      "yum install -y redis",
      "systemctl enable redis",
      "systemctl start redis"
    ]
  }
]
```

Avant de lancer la création de l'image, vérifions la syntaxe de notre fichier `template.json` :

```
root@vagrant-packer:~# packer validate
template.json
Template validated successfully.
```

Création de l'image :

```
root@vagrant-packer:~# packer build
template.json
digitalocean output will be in this color.

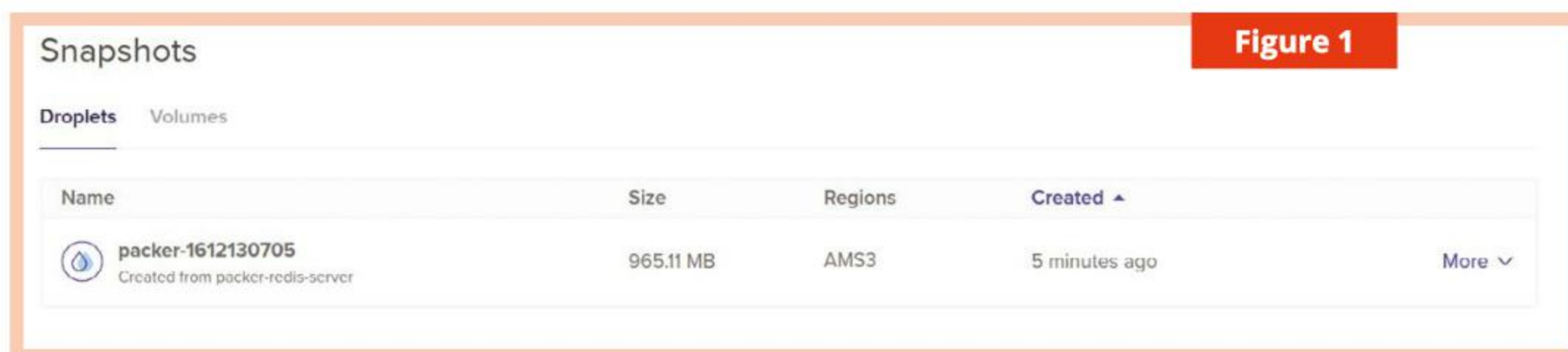
==> digitalocean: Creating temporary ssh
key for droplet...
==> digitalocean: Creating droplet...
==> digitalocean: Waiting for droplet to
become active...
```

Une fois la création de l'image et l'installation des packages finalisée (Figure 1), on peut utiliser ce modèle d'image pour déployer des machines à la demande. Cette image peut être également mise à la disposition pour déployer rapidement et accélérer le processus de mise en production.

3. CONTENEURISATION D'APPLICATIONS AVEC DOCKER

Docker permet rapidement la création des applications en mode conteneur. Il permet de tester rapidement et efficacement vos applications d'une part et de déployer indépendamment de la plateforme cible.

Pour personnaliser vos applications, il faut commencer par créer une image. Cette image est construite à partir d'un fichier modèle appelé `Dockerfile`. Nous créons l'image à partir d'un `Dockerfile`, ensuite on démarre l'application via les commandes docker.



3.1 Installation de Docker

Dans notre cas, nous installons Docker à partir du dépôt Docker de la distribution Ubuntu.

```
$ sudo apt-get update

$ sudo apt-get install \
  apt-transport-https \
  ca-certificates \
  curl \
  gnupg-agent \
  software-properties-common
```

On ajoute la clé GPG du dépôt :

```
$ curl -fsSL https://download.docker.com/
linux/ubuntu/gpg | sudo apt-key add -
$ sudo add-apt-repository \
  "deb [arch=amd64] https://download.
docker.com/linux/ubuntu \
  $(lsb_release -cs) \
  stable"
```

Maintenant, on peut installer les paquetages nécessaires :

```
$ sudo apt-get update
$ sudo apt-get install docker-ce docker-
ce-cli containerd.io
```

Vérifions que le démon docker est démarré :

```
root@vagrant-packer:~# systemctl status
docker
• docker.service - Docker Application
Container Engine
   Loaded: loaded (/lib/systemd/system/
docker.service; enabled; vendor preset:
enabled)
   Active: active (running) since Sun 2021-
01-31 22:49:33 UTC; 36s ago
     Docs: https://docs.docker.com
  Main PID: 5237 (dockerd)
    Tasks: 8
   CGroup: /system.slice/docker.service
           └─5237 /usr/bin/dockerd -H fd://
--containerd=/run/containerd/containerd.
sock
```

3.2 Création de l'image Docker

Pour créer l'image de l'application à partir du Dockerfile, je vais cloner mon dépôt GitHub qui contient les sources de mon application :

```
# git clone https://github.com/imejri/
psweb.git
Cloning into 'psweb'...
remote: Enumerating objects: 13, done.
remote: Counting objects: 100% (13/13),
done.
remote: Compressing objects: 100% (11/11),
done.
remote: Total 44 (delta 3), reused 0 (delta
0), pack-reused 31
Unpacking objects: 100% (44/44), done.

# cd psweb
```

Voici notre Dockerfile :

```
FROM alpine

LABEL maintainer="issam.mejri@monmail.com"

# Install Node and NPM
RUN apk add --update nodejs nodejs-npm

# Copy app to /src
COPY . /src

WORKDIR /src

# Install dependencies
RUN npm install

EXPOSE 8080

ENTRYPOINT ["node", "./app.js"]
```

Création de l'image :

```
# docker build -t psweb:v1.0
Successfully built 1f8c5f890037
Successfully tagged psweb:v1.0
```

Vérifions la présence de nos images. Normalement, on aura deux images, la première est l'image de base provenant de l'instruction **FROM** et notre image build à partir de celle-ci.


```
# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
psweb         v1.0     1f8c5f890037   6 minutes ago  84.1MB
alpine        latest   e50c909a8df2   3 days ago    5.61MB
```

3.3 Démarrage du conteneur

Maintenant que notre image est disponible, on peut démarrer notre conteneur :

```
root@vagrant-packer:~# docker run -d -p 8080:8080 --name psweb psweb:v1.0
e23e511b2ddd329cc72812bd87f7716630fc661f18dd158e0420f3c0da4b5c12
```

Vérifions le bon démarrage du conteneur :

```
root@vagrant-packer:~# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
e23e511b2ddd   psweb:v1.0     "node ./app.js"         2 minutes ago Up 2 minutes   0.0.0.0:8080->8080/tcp             psweb
```

Notre conteneur est bien en **UP**. Désormais, on peut accéder à notre application via l'adresse <http://hôte:8080> à partir du navigateur, ou encore effectuer un `curl http://hôte`.

3.4 Déployer des instances dockerisées avec Docker Machine

Lorsqu'on travaille dans un environnement de développement, qu'on a besoin d'avoir plusieurs serveurs sur lesquels installer Docker tout en administrant ces serveurs à partir d'un seul et même endroit, Docker Machine est la solution la plus adéquate.

Il facilite en effet énormément le déploiement des machines avec une administration simplifiée. Découvrons cela par la pratique, en commençant par son installation.

3.4.1 Installation de Docker Machine

```
# base=https://github.com/docker/machine/releases/download/v0.16.0 && \
curl -L $base/docker-machine-$(uname -s)-$(uname -m) >/tmp/docker-machine

# sudo mv /tmp/docker-machine /usr/local/bin/docker-machine && \
chmod +x /usr/local/bin/docker-machine
```

Testons notre installation :

```
# docker-machine --version
docker-machine version 0.16.0, build 702c267f
```

Docker Machine est conçu autour de plusieurs drivers pour s'intégrer avec la plupart des clouds publics et privés. En effet, il est possible de déployer nos machines dockerisées sur AWS, OpenStack, VMware vSphere...

Je resterai toujours sur un déploiement vers DigitalOcean pour notre article. La commande principale permettant de créer le Docker Host est la suivante :




```
$ docker-machine create --driver
digitalocean --digitalocean-access-token=a
a9399a2175a93b17b1c86c807e08d3fc4b79876545432
a629602f61cf6ccd6b dev-host
. . . . .
Creating CA: /root/.docker/machine/certs/
ca.pem
Creating client certificate: /root/.docker/
machine/certs/cert.pem
Running pre-create checks...
Creating machine...
(dev-host) Creating SSH key...
(dev-host) Creating Digital Ocean
droplet...
(dev-host) Waiting for IP address to be
assigned to the Droplet...
Waiting for machine to be running, this
may take a few minutes..
```

Les paramètres à passer à la commande sont :

- `create` : pour lancer la création de la machine ;
- `driver` : on choisit le driver qui correspond à notre besoin, dans mon cas c'est DigitalOcean ;
- `digitalocean-access-token` : le token d'accès à l'API DigitalOcean.

On peut désormais accéder en local à l'hôte fraîchement installé via la commande `docker-machine`.

```
# docker-machine ssh dev-host
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux
4.4.0-193-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.
canonical.com
 * Support:       https://ubuntu.com/
advantage

60 packages can be updated.
48 updates are security updates.

root@dev-host:~#
```

Super, on est bien connectés à notre machine avec un démon docker installé. On peut désormais jouer avec nos conteneurs.

4. CRÉER VOTRE PIPELINE AS CODE AVEC JENKINS ET JENKINSFILE

Jenkins est un serveur de build, mais principalement un outil d'intégration continue et de déploiement continu. La réputation de Jenkins réside dans le fait qu'il est open source et qu'il s'affranchit de toute dépendance vis-à-vis d'un environnement de compilation. Il est conçu aussi autour de plus d'un millier de plugins facilitant son intégration avec la plupart des outils de l'écosystème DevOps.

Dans cet article, on va exploiter une fonctionnalité intéressante de Jenkins, le Pipeline As Code. Le Pipeline As Code permet de décrire le cycle d'intégration et de déploiement d'une application (build, test, archivage, déploiement) dans un fichier appelé Jenkinsfile.

Dans le Jenkinsfile, on décrit l'ensemble des jobs ou tâches à réaliser dans des sections appelées stages.

Voici un aperçu de notre Jenkinsfile :

```
pipeline {
    agent any

    parameters {
        string(name: 'tomcat_dev',
        defaultValue: 'ip_staging_server',
        description: 'Staging Server')
        string(name: 'tomcat_prod',
        defaultValue: 'ip_prod_server',
        description: 'Production Server')
    }

    triggers {
        pollSCM('* * * * *')
    }

    stages{
        stage('Build'){
            steps {
                sh 'mvn clean package'
            }
            post {
                success {
                    echo 'Now Archiving...'
                    archiveArtifacts
                }
            }
            artifacts: '**/target/*.war'
        }
    }
}
```



```

    }
  }

  stage ('Deployments'){
    parallel{
      stage ('Deploy to Staging')
    {
      steps {
        sh "scp -i /home/
jenkins/tomcat-demo.pem **/target/*.war
ec2-user@${params.tomcat_dev}:/var/lib/
tomcat7/webapps"
      }
    }

    stage ("Deploy to
Production"){
      steps {
        sh "scp -i /home/
jenkins/tomcat-demo.pem **/target/*.war
ec2-user@${params.tomcat_prod}:/var/lib/
tomcat7/webapps"
      }
    }
  }
}
}
}

```

La version 2 de Jenkins supporte deux styles syntaxiques de pipeline, le script pipeline et la pipeline déclarative. La deuxième est la plus intuitive et la plus simple à comprendre au début, alors que la première nécessite quelques connaissances en programmation Groovy (un langage de programmation s'exécutant sous la machine virtuelle de Java, ce qui explique son utilisation par Jenkins qui est lui-même en Java) par contre elle est plus puissante en termes de personnalisation de la pipeline.

Notre Jenkinsfile commence toujours par la section pipeline. C'est la section parente.

La section `parameters` permet d'ajouter des paramètres à notre pipeline. Ces paramètres peuvent être de type string ou booléen.

La section `Triggers` permet de déclencher l'exécution du pipeline sous certaines conditions. Dans notre cas, c'est de vérifier un changement de code source toutes les minutes.



La section `stages` quant à elle encapsule d'autres sections `stages`, c'est dans ces sections qu'on décrit toutes les étapes de l'exécution de la pipeline. Dans notre code, on a une section permettant de compiler le code source (Build), une deuxième pour déployer dans l'environnement d'avant production (Deploy to Staging) et la dernière section lance un déploiement vers la production (Deploy to Production).

À l'intérieur de chaque section, on décrit les tâches à réaliser, ce qui explique la présence de la section `steps`.

Cette pipeline réalise :

- Le build d'une application java en se basant sur Maven (`sh 'mvn clean package'`).
- L'archivage de l'artefact (le binaire `.war`) sur le système de fichiers (`archiveArtifacts artifacts: '**/target/*.war'`) juste après l'opération du build via la directive `post` qui déclenche des actions suivant des résultats.
- Le déploiement sur un serveur d'applications tomcat (`sh "scp -i /home/jenkins/tomcat-demo.pem **/target/*.war ec2-user@${params.tomcat_dev}:/var/lib/tomcat7/webapps"`) dans les deux environnements cibles (staging et production).

4.1 Configuration de la pipeline

Après avoir maîtrisé notre Jenkinsfile, c'est le moment de déployer notre pipeline dans Jenkins. Je suppose que vous avez une installation du serveur Jenkins avec le plugin pipeline déjà installé.

The screenshot shows the 'Advanced Project Options' tab in Jenkins. At the top, there are tabs for 'General', 'Build Triggers', 'Advanced Project Options', and 'Pipeline'. Below the tabs, the 'Advanced Project Options' section has an 'Avancé...' button. The 'Pipeline' section contains a 'Definition' dropdown set to 'Pipeline script from SCM'. Below this, the 'SCM' dropdown is set to 'Git'. The 'Repositories' section shows a 'Repository URL' of 'https://github.com/imejri/m', 'Credentials' set to '- aucun -', and an 'Ajouter' button. There are 'Avancé...' and 'Add Repository' buttons. The 'Branches to build' section shows a 'Branch Specifier (blank for \'any\')' set to '*/master' and an 'Add' button. At the bottom left, there are 'Sauver' and 'Apply' buttons.

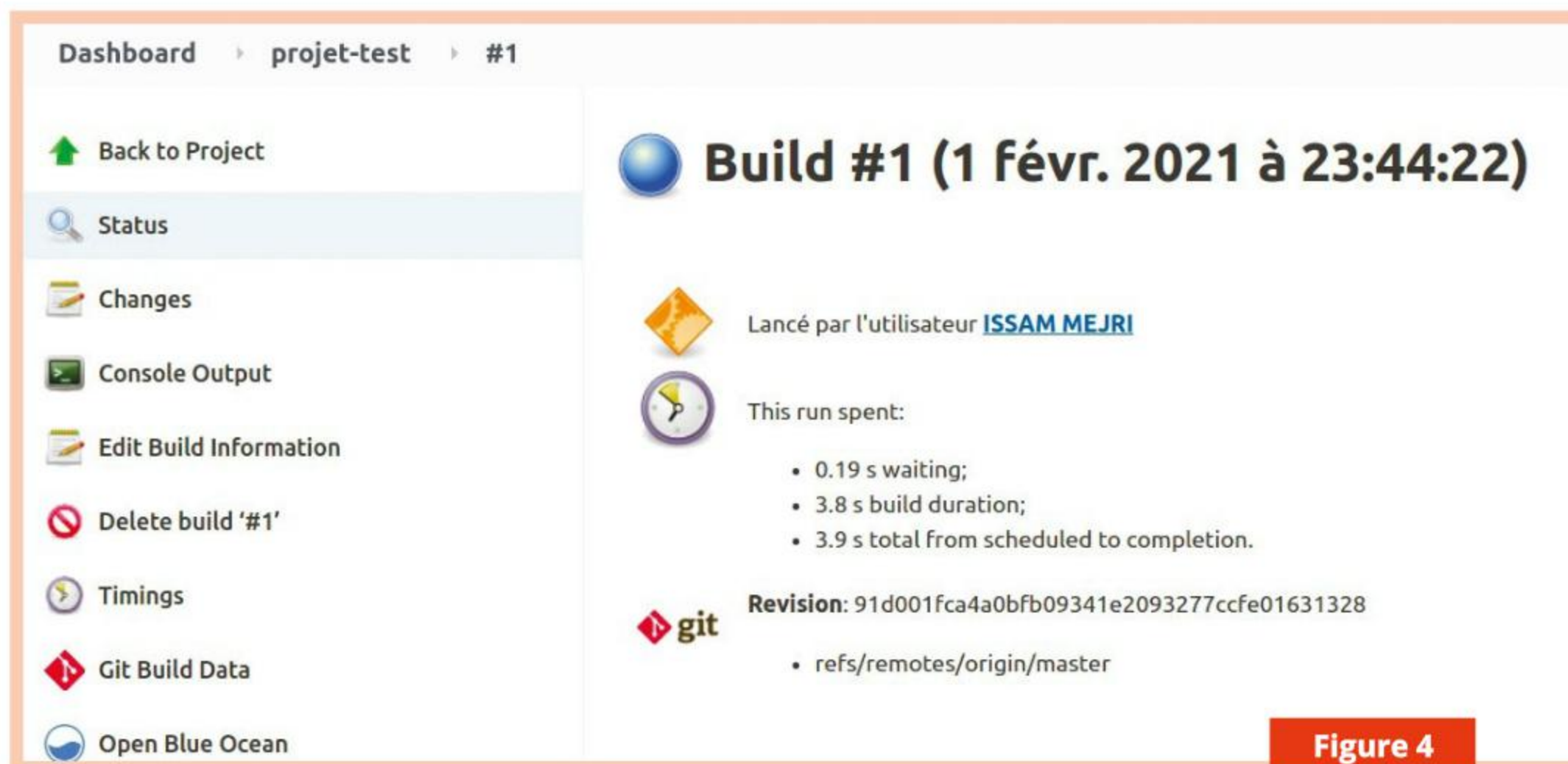
Figure 2

The screenshot shows the Jenkins dashboard for a project named 'projet-test'. The left sidebar contains links: 'Back to Dashboard', 'Status', 'Changes', 'Lancer un build', 'Configurer', 'Supprimer Pipeline', 'Full Stage View', 'Open Blue Ocean', 'Rename', and 'Pipeline Syntax'. The main content area is titled 'Pipeline projet-test' and includes a 'Recent Changes' section with a 'Recent Changes' link. Below this is a 'Stage View' section with a message: 'No data available. This Pipeline has not yet run.' At the bottom, there is a 'Liens permanents' section. A red box at the bottom right of the main content area contains the text 'Figure 3'.

Pour cela, il faut se connecter à Jenkins, cliquez sur **Nouveau item** puis dans la fenêtre qui s'affiche choisir **Pipeline**, donner un nom au projet, puis valider avec **OK**.

Dans la fenêtre qui s'affiche, cliquez sur l'onglet **Pipeline** comme dans la figure 2.

Dans **Définition**, choisir **Pipeline Script from SCM**, puis dans le menu déroulant **SCM** choisir **Git** comme **Source de dépôt**. Indiquez ensuite l'URL du code source dans votre dépôt git, laissez les autres champs par défaut, puis cliquez sur **Sauver**.



Dans la fenêtre de projet qui s'affiche (Figure 3), cliquez sur **Lancer un build** pour déclencher l'exécution de notre pipeline qui va exécuter toutes les étapes de création du projet.

En bas à gauche de la fenêtre un premier build de notre projet est créé (Figure 4).

Cliquez sur ce build, puis dans la fenêtre qui s'affiche, cliquez sur **Console Output** pour inspecter les journaux du déroulement de la pipeline. La Console Output permet également de diagnostiquer les erreurs qui peuvent se produire lors de l'exécution des jobs.

CONCLUSION

Les outils DevOps facilitent énormément la mise en place technique d'une démarche DevOps. Dans cette première partie, j'ai évoqué quelques-uns d'entre eux. Une seconde partie est d'ores et déjà prévue pour continuer à vous en faire découvrir d'autres parmi lesquels Ansible, Terraform, Git, OpenShift... ■

RÉFÉRENCES

- Tableau périodique des outils DevOps : <https://www.devops-cloud.fr/tableau-periodique-devops-tools-par-xebialabs/>
- Pipeline As Code avec Jenkins : <https://www.jenkins.io/solutions/pipeline/>
- Documentation Hashicorp de Vagrant : <https://www.vagrantup.com/docs>
- Documentation Hashicorp de Packer : <https://www.packer.io/docs>
- Guide Docker-Machine en local : <https://github.com/docker/machine>





HACKABLE MAGAZINE

DÉMONTEZ | COMPRENEZ | ADAPTEZ | PARTAGEZ

N° 36
JAN. / FÉV. / MARS
2021

FRANCE MÉTRO: 12 000 € - BELGIUM: 12 000 € - CH: 10 000 CHF
ESPAGNE/PORTUGAL: 13 000 € - DOMAINS: 13 000 € - TURK: 30 000 TL
MAR: 145 MAD - CAN: 21 000 CAD

L 19338 - 36 - F: 12,90 € - RD



MATÉRIEL / SIGNAUX

Découverte et utilisation de l'analyseur de réseau économique NanoVNA

DOMOTIQUE / ARDUINO

Pilotez des ventilateurs PC pour gérer la température d'un coffret réseau

STM32 / MÉMOIRE / RTOS

Comprenez et utilisez l'accès direct mémoire (DMA) avec STM32 et ChibiOS

SDR / POSITIONNEMENT

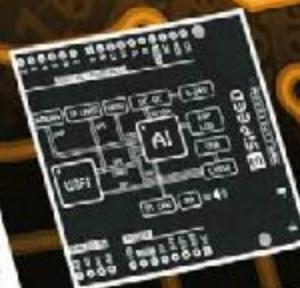
Décodage par radio logicielle du système de navigation aérienne VOR

ESP32 / Sipeed / K210

RISC-V + IA

Découvrez Maixduino,
la carte "Arduino" ultime !

- 1 - Prise en main du kit Maixduino
- 2 - Configuration de l'IDE Arduino
- 3 - Deep learning avec MicroPython



OPEN SOURCE / ASIC

2021, année du silicium libre ? Comment 20 ans d'histoire ont conduit à RISC-V et Google Skywater

RÉTRO / NINTENDO

Recréez le légendaire Pac-Man en assembleur 6502 pour votre NES



PAPIER

en kiosque



FLIPBOOK HTML5

sur www.ed-diamond.com



CONNECT

LA DOCUMENTATION TECHNIQUE DES PROS DE L'IT

sur connect.ed-diamond.com

Chez votre marchand de journaux
et sur www.ed-diamond.com

DÉPLOYER JENKINS CI avec Docker

ROMAIN PELISSE

SOLUTION APPRÉCIÉE D'INTÉGRATION CONTINUE, JENKINS EST UN PUISSANT LOGICIEL JAVA DONT L'INSTALLATION ET L'EXÉCUTION SUR UN SYSTÈME NE SONT PAS SANS CONSÉQUENCE ET À PRENDRE À LA LÉGÈRE. AFIN DE FACILITER SON DÉPLOIEMENT ET ISOLER PROPREMENT CE PROGRAMME DU RESTE DE LA MACHINE QUI L'HÉBERGE, NOUS ALLONS ILLUSTRER ICI COMMENT LE METTRE EN PLACE, EN SEULEMENT QUELQUES COMMANDES, À L'AIDE DE DOCKER.



Jenkins est un célèbre logiciel Java dédié à l'intégration continue. Dès ses premières heures, il a connu un immense succès et une large adoption, même au-delà de la communauté Java open source. L'une des raisons de ceci est la grande simplicité de mise en place de l'outil. On le paramètre en quelques clics

seulement et il bénéficie de nombreuses extensions pour adapter son comportement et enrichir ses fonctionnalités selon ses besoins.

Si cet aspect a énormément facilité la prise en main de l'outil d'intégration continue, il demeure cependant nécessaire, au préalable, d'installer le logiciel et de le mettre en place sur le système.

Avec l'arrivée de Docker et la mise à disposition d'images de conteneur prêtes à l'emploi par la communauté de Jenkins, nous allons pouvoir non seulement aisément installer Jenkins, avec le minimum de configuration sur la machine cible, mais aussi profiter de la souplesse de Docker pour gérer sa mise à jour. Bonus non

négligeable de cette approche, Docker va isoler ce service du reste du système qui l'héberge. Et ceci, en l'exécutant avec les privilèges d'un simple utilisateur, apportant ainsi un confort appréciable en matière de sécurité. Démonstration par l'exemple dans cet article.



1. RÉCUPÉRATION DE L'IMAGE

Jenkins est une solution complète et relativement monolithique. Son serveur peut bien sûr utiliser de nombreuses ressources externes afin d'exécuter les constructions logicielles dont il a la charge, mais lui-même est, en essence, un seul

(mais complexe) programme Java. En ce sens, il se prête très bien à la mise à disposition sous forme de conteneur Docker. Sa communauté propose d'ailleurs, sans surprise, des images prêtes à l'emploi pour la plupart de ses versions récentes.

Nous allons donc profiter de l'une de celles-ci pour déployer le service d'intégration continue. Commençons par installer cette image à l'aide de la commande `docker pull` et vérifier qu'elle fonctionne sans encombre sur notre système :

```
$ docker pull jenkins/jenkins:lts
lts: Pulling from jenkins/jenkins
3192219afd04: Pull complete
17c160265e75: Pull complete
cc4fe40d0e61: Pull complete
9d647f502a07: Pull complete
d108b8c498aa: Pull complete
1bfe918b8aa5: Pull complete
dafa1a7c0751: Pull complete
675a3a9eb833: Pull complete
53cdb29bcd2b: Pull complete
5f0910286082: Pull complete
77ab70638138: Pull complete
7be43348db56: Pull complete
86b25b001094: Pull complete
d7939f1aa909: Pull complete
57bf080fb868: Pull complete
9aaadcd7f7e0: Pull complete
ffa20079fc9a: Pull complete
34b6fb6eb6f62: Pull complete
13f37a4ccdaa: Pull complete
12c7a8ed0a76: Pull complete
Digest: sha256:63fde6791bc9e11dc8fc94830ac76228afbe8d31206ab8f83f251314a3f45ba9
Status: Downloaded newer image for jenkins/jenkins:lts
docker.io/jenkins/jenkins:lts
```

Le lecteur prendra soin de noter que nous avons sélectionné l'image étiquetée lts (acronyme qui signifie long time support). Par défaut, l'image proposée est le fruit de l'intégration continue hebdomadaire de Jenkins. Comme il s'agit ici d'automatiser la mise en place d'un serveur de production, nous avons évidemment opté pour utiliser une version stable de l'outil.

Une fois l'image installée sur le système, démarrons un conteneur à partir de celle-ci et vérifions le bon fonctionnement de Jenkins :

```
$ docker run -p 8080:8080 jenkins/jenkins:lts
Running from: /usr/share/jenkins/jenkins.war
webroot: EnvVars.masterEnvVars.get("JENKINS_HOME")
2021-01-19 10:03:05.865+0000 [id=1] INFO org.eclipse.jetty.
util.log.Log#initialized: Logging initialized @365ms to org.
eclipse.jetty.util.log.JavaUtilLog
```



```

2021-01-19 10:03:05.980+0000 [id=1] INFO winstone.Logger#logInternal: Beginning extraction from war
file
2021-01-19 10:03:07.148+0000 [id=1] WARNING o.e.j.s.handler.ContextHandler#setContextPath: Empty
contextPath
2021-01-19 10:03:07.206+0000 [id=1] INFO org.eclipse.jetty.server.Server#doStart: jetty-9.4.33.
v20201020; built: 2020-10-20T23:39:24.803Z; git: 1be68755656cef678b79a2ef1c2ebbca99e25420; jvm 1.8.0 _242-b08
2021-01-19 10:03:07.451+0000 [id=1] INFO o.e.j.w.StandardDescriptorProcessor#visitServlet: NO JSP
Support for /, did not find org.eclipse.jetty.jsp.JettyJspServlet
2021-01-19 10:03:07.490+0000 [id=1] INFO o.e.j.s.s.DefaultSessionIdManager#doStart:
DefaultSessionIdManager workerName=node0
2021-01-19 10:03:07.490+0000 [id=1] INFO o.e.j.s.s.DefaultSessionIdManager#doStart: No
SessionScavenger set, using defaults
2021-01-19 10:03:07.492+0000 [id=1] INFO o.e.j.server.session.HouseKeeper#startScavenging: node0
Scavenging every 600000ms
2021-01-19 10:03:07.887+0000 [id=1] INFO hudson.WebAppMain#contextInitialized: Jenkins home
directory: /var/jenkins_home found at: EnvVars.masterEnvVars.get("JENKINS_HOME")
2021-01-19 10:03:08.002+0000 [id=1] INFO o.e.j.s.handler.ContextHandler#doStart: Started
w.@1b58ff9e{Jenkins v2.263.2,,file:///var/jenkins_home/war/AVAILABLE}{/var/jenkins_home/war}
2021-01-19 10:03:08.018+0000 [id=1] INFO o.e.j.server.AbstractConnector#doStart: Started
ServerConnector@126253fd{HTTP/1.1, (http/1.1)}{0.0.0.0:8080}
2021-01-19 10:03:08.019+0000 [id=1] INFO org.eclipse.jetty.server.Server#doStart: Started @2519ms
2021-01-19 10:03:08.019+0000 [id=23] INFO winstone.Logger#logInternal: Winstone Servlet Engine
running: controlPort=disabled
2021-01-19 10:03:10.551+0000 [id=30] INFO jenkins.InitReactorRunner$1#onAttained: Started
initialization
2021-01-19 10:03:10.588+0000 [id=39] INFO jenkins.InitReactorRunner$1#onAttained: Listed all plugins
2021-01-19 10:03:12.083+0000 [id=29] INFO jenkins.InitReactorRunner$1#onAttained: Prepared all
plugins
2021-01-19 10:03:12.092+0000 [id=31] INFO jenkins.InitReactorRunner$1#onAttained: Started all plugins
2021-01-19 10:03:12.101+0000 [id=32] INFO jenkins.InitReactorRunner$1#onAttained: Augmented all
extensions
2021-01-19 10:03:13.010+0000 [id=37] INFO jenkins.InitReactorRunner$1#onAttained: System config
loaded
2021-01-19 10:03:13.011+0000 [id=28] INFO jenkins.InitReactorRunner$1#onAttained: System config
adapted
2021-01-19 10:03:13.011+0000 [id=28] INFO jenkins.InitReactorRunner$1#onAttained: Loaded all jobs
2021-01-19 10:03:13.012+0000 [id=28] INFO jenkins.InitReactorRunner$1#onAttained: Configuration for
all jobs updated
2021-01-19 10:03:13.023+0000 [id=56] INFO hudson.model.AsyncPeriodicWork#lambda$doRun$0: Started
Download metadata
2021-01-19 10:03:13.035+0000 [id=56] INFO hudson.util.Retrier#start: Attempt #1 to do the action
check updates server
2021-01-19 10:03:14.000+0000 [id=33] INFO o.s.c.s.AbstractApplicationContext#prepareRefresh:
Refreshing org.springframework.web.context.support.StaticWebApplicationContext@400c9591: display name
[Root WebApplicationContext]; startup date [Tue Jan 19 10:03:14 UTC 2021]; root of context hierarchy
2021-01-19 10:03:14.001+0000 [id=33] INFO o.s.c.s.AbstractApplicationContext#obtainFreshBeanFactory:
Bean factory for application context [org.springframework.web.context.support.StaticWebApplicationContext
@400c9591]: org.springframework.beans.factory.support.DefaultListableBeanFactory@f38075a

```



```

2021-01-19 10:03:14.015+0000 [id=33]      INFO    o.s.b.f.s.DefaultListableBeanFactory#preInstantiateSingletons:
Pre-instantiating singletons in org.springframework.beans.factory.support.DefaultListableBeanFactory@f38075a:
defining beans [authenticationManager]; root of factory hierarchy
2021-01-19 10:03:14.234+0000 [id=33]      INFO    o.s.c.s.AbstractApplicationContext#prepareRefresh:
Refreshing org.springframework.web.context.support.StaticWebApplicationContext@4bef20e3: display name [Root
WebApplicationContext]; startup date [Tue Jan 19 10:03:14 UTC 2021]; root of context hierarchy
2021-01-19 10:03:14.234+0000 [id=33]      INFO    o.s.c.s.AbstractApplicationContext#obtainFreshBeanFactory:
Bean factory for application context [org.springframework.web.context.support.StaticWebApplicationContext@4b
ef20e3]: org.springframework.beans.factory.support.DefaultListableBeanFactory@626e81cc
2021-01-19 10:03:14.235+0000 [id=33]      INFO    o.s.b.f.s.DefaultListableBeanFactory#preInstantiateSingletons:
Pre-instantiating singletons in org.springframework.beans.factory.support.DefaultListableBeanFactory@626e8
1cc: defining beans [filter,legacy]; root of factory hierarchy
2021-01-19 10:03:14.626+0000 [id=33]      INFO    jenkins.install.SetupWizard#init:

*****
*****
*****

Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:

0f4aea009bff4e81a2ecb3b26e43ba2c

This may also be found at: /var/jenkins_home/secrets/initialAdminPassword

*****
*****
*****

2021-01-19 10:03:20.030+0000 [id=56]      INFO    h.m.DownloadService$Downloadable#load: Obtained the updated
data file for hudson.tasks.Maven.MavenInstaller
2021-01-19 10:03:20.030+0000 [id=56]      INFO    hudson.util.Retrier#start: Performed the action check
updates server successfully at the attempt #1
2021-01-19 10:03:20.034+0000 [id=56]      INFO    hudson.model.AsyncPeriodicWork#lambda$doRun$0: Finished
Download metadata. 7,009 ms
2021-01-19 10:03:23.603+0000 [id=34]      INFO    jenkins.InitReactorRunner$1#onAttained: Completed initial

```

Une simple requête HTTP, effectuée à l'aide de la commande `curl`, nous permet de valider que Jenkins est accessible :

```
$ curl -I http://localhost:8080
```

2. GÉRER LES DONNÉES DE JENKINS À L'AIDE D'UN VOLUME

À partir de là, on peut utiliser un navigateur pour accéder à Jenkins et finir l'installation à l'aide d'une série de formulaires interactifs. La configuration du serveur n'étant pas le propos de cet article, nous allons ainsi sauter cette étape. D'autant plus que, en l'état, si nous arrêtons le conteneur, les données de Jenkins vont disparaître. Le travail de paramétrage sera alors perdu !

Il est certes possible de démarrer à nouveau ce même conteneur, mais ce n'est pas une solution optimum. L'idéal est d'héberger ces données sur le système lui-même et d'accorder l'accès à celles-ci au conteneur responsable de l'exécution de Jenkins.

Pour ce faire, nous allons donc ajouter un volume à la commande `docker run` :

```
$ docker run -p 8080:8080 -v /home/jenkins:/var/
jenkins_home:rw jenkins/jenkins:lts
Attention, pour que ce volume soit parfaitement
accessible depuis le conteneur, il faut créer un
utilisateur et un groupe jenkins, aux mêmes UID et
GID que ceux attribués au sein de l'image Docker
fournie :
# useradd -u 1000 jenkins -d /home/jenkins
# groupadd -g 1010 jenkins
```

Avec l'addition de ce volume, les données de Jenkins (sa configuration et plus tard la définition de ses tâches de construction logicielle) vont désormais survivre à l'arrêt ou au crash du conteneur.

Par souci d'exhaustivité, ajoutons deux arguments supplémentaires au démarrage de celui-ci :

- le paramètre `--rm` qui assure que le conteneur sera automatiquement effacé une fois son exécution suspendue (la présence du volume nous autorise à automatiser ceci en toute sérénité) ;
- et le paramètre `--name` qui nous permet de nommer l'instance lancée « jenkins ».

Voici donc ci-dessous la ligne de commandes complète de démarrage de Jenkins :

```
$ docker run -p 8080:8080 -v/home/
jenkins : /var/jenkins_home: rw --rm
--name jenkins jenkins/jenkins:lts
```

Avant de conclure, il nous faut attirer l'attention du lecteur sur un point important : l'image Docker a été lancée depuis un simple compte utilisateur sans privilège d'administration (root). En effet, l'exécution de Jenkins ne requiert en aucun cas de disposer d'un tel accès. Il est donc fortement recommandé d'utiliser le compte `jenkins` (créé un peu plus haut) afin d'exécuter le conteneur en lui-même.

Ceci garantit que si l'application dysfonctionne ou est simplement compromise, elle ne sera pas en mesure d'affecter le comportement des autres services hébergés sur la même machine.

CONCLUSION

À l'aide de seulement quelques commandes Docker, et au goût, faible, de la création d'un compte et d'un groupe utilisateur, nous avons pu déployer une instance complète de Jenkins, sans imposer l'installation et la gestion de ses dépendances sur notre système cible. Point appréciable, l'installation d'une machine virtuelle Java sur celui-ci n'est pas requise, car celle-ci est fournie par l'image Docker utilisée. De même le maintien à jour de la version de ce cadre d'exécution est à la responsabilité de la communauté qui s'occupe de cette image Docker.

En outre, l'ensemble de l'installation s'est effectuée depuis un compte utilisateur système, sans privilège particulier. Seule la création du compte utilisateur et du groupe `jenkins` exige de disposer de privilèges d'administration. On notera cependant que si cette approche permet d'exécuter le service sans être root, elle impose malheureusement l'utilisation d'un UID et d'un GID précis. Ceci vient donc avec des risques de conflits, si ces identifiants sont déjà utilisés par un autre utilisateur ou groupe sur la machine cible.

Il n'en demeure pas moins que cette mise en place a été rapide, aisée et apporte un cadre d'exécution sûr (confiné à un conteneur et à un simple utilisateur système) tout en réduisant au minimum le travail de maintien à jour du logiciel, qui est intégralement délégué à la communauté qui a fourni l'image utilisée. ■



Toujours disponible sur
www.ed-diamond.com



sur www.ed-diamond.com



sur connect.ed-diamond.com

GÉREZ VOS PROJETS GIT EN MODE GRAPHIQUE

avec GitKraken

MORÈRE YANN



JE VOUS AI PRÉCÉDEMMENT PRÉSENTÉ GITLAB [0], UNE FORGE LOGICIELLE UTILISANT GIT VOUS PERMETTANT UNE GESTION GLOBALE DE VOS PROJETS EN ÉQUIPE. COMME NOUS L'AVIONS VU, UNE CONNAISSANCE MINIMALE DE GIT ÉTAIT NÉCESSAIRE POUR L'UTILISATION DE GITLAB. CERTAINS DE VOS COLLABORATEURS NE SONT PEUT-ÊTRE PAS À L'AISE AVEC LA LIGNE DE COMMANDES, NE VEULENT TOUT SIMPLEMENT PAS L'UTILISER OU ENCORE NE VEULENT PAS APPRENDRE LA SYNTAXE DES COMMANDES GIT. GITKRAKEN [1] EST FAIT POUR EUX. IL LEUR PERMETTRA DE RÉALISER TOUTES LES OPÉRATIONS GIT À L'AIDE D'UNE INTERFACE GRAPHIQUE.

» ÉTAPE 0 : QU'EST-CE QUE GITKRAKEN

Git, comme nous l'avons vu dans l'article présentant GitLab est un vraiment très pratique pour le contrôle de sources distribuées. Cependant, son utilisation en ligne de commandes n'est pas forcément très simple

d'accès pour le commun des mortels. Il n'est d'ailleurs pas connu pour sa convivialité ou sa facilité d'apprentissage. La solution est alors d'utiliser un frontal graphique qui vous permettra de

réaliser toutes les actions nécessaires sans avoir à retenir la syntaxe des commandes Git. Ugit [2] est une première solution. Elle permet d'apporter un peu de convivialité à Git sans sacrifier sa polyvalence. Le logiciel basé web (il utilise Node.js) est composé d'une interface intuitive pour faciliter la compréhension de Git, il fonctionne sur toutes les plateformes et s'utilise directement à travers votre navigateur. Cependant, le fait que cet utilitaire soit 100 % web pourra rebuter certains utilisateurs.

Il existe de nombreuses autres applications qui permettent d'interfacer graphiquement Git, un tableau les répertoriant se trouve à la page [3]. De notre côté, nous allons nous intéresser à GitKraken [1]. Ce dernier est largement plébiscité par les utilisateurs pour ses nombreuses fonctionnalités – c'est pourquoi nous avons choisi de vous le présenter même s'il s'agit d'un projet propriétaire – et son interface intuitive : les fonctions les plus utilisées (« push », « pull », « branch », « stash », « commit ») sont accessibles en un clic, et sont les seuls boutons. Les autres fonctionnalités ne se trouvent pas dans des menus compliqués, mais sont affichées par l'intermédiaire d'un menu contextuel. Découvrons sans attendre cet outil.

» ÉTAPE 1 : INSTALLATION

En tant qu'enseignant, il est possible d'obtenir une licence professionnelle gratuite. Pour cela, il suffit d'un compte GitHub et de fournir une preuve (photo recto/verso de votre carte professionnelle par exemple). Vous aurez alors accès à la boîte à outils enseignants GitHub (GitHub Teacher Toolbox). Vous trouverez les informations sur cette offre à l'adresse [4].

Pour les étudiants, il est possible d'obtenir une version pro gratuite à travers le « GitHub Student Developer Pack ». Les informations sont disponibles à la page [5].

Sinon, il vous est possible d'utiliser GitKraken gratuitement, mais les fonctionnalités seront limitées (cf. la page [6]). Par exemple, il n'est pas possible de

travailler avec des dépôts privés, ni avec des dépôts autogérés (cf. votre installation de Git/GitLab sur votre serveur personnel décrite dans l'article [0]), ni avec des collaborateurs. Cependant, cela peut être largement suffisant dans le cadre d'une évaluation du logiciel.

L'installation est très simple. Sur une distribution Ubuntu, il suffit de télécharger le paquet [gitkraken-amd64.deb](#) sur la page [1]. Ensuite, vous pouvez utiliser l'application [gdebi](#) afin de l'installer graphiquement ou encore utiliser la ligne de commandes suivante :

```
$ sudo dpkg -i gitkraken-amd64.deb
```

Sur ma distribution Arch Linux, l'application est déjà présente dans les dépôts utilisateurs. Je l'installe à l'aide de la commande :

```
$ yaourt -S gitkraken
```

Vous pouvez lancer GitKraken par l'intermédiaire de la ligne de commandes ou encore via le menu programmation de votre bureau. L'application devrait s'y trouver.

Lors du premier lancement, il faut associer l'application à votre compte GitHub ou créer un compte GitKraken.

BON À SAVOIR

Si vous êtes déjà connecté avec vos identifiants sur les sites web GitHub/GitLab, un simple appui sur le bouton **Connect validera la connexion entre GitKraken et GitHub/GitLab.**

Ici, je prendrai en exemple GitHub. Il suffit de cliquer sur le bouton **Sign in with GitHub**, cela ouvre un navigateur vous demandant l'autorisation d'associer GitHub à Gitkraken. Il vous faudra ensuite renseigner vos identifiants et mots de passe GitHub afin de valider l'association.

De mon côté, j'ai ensuite activé mon compte GitLab, et associé mon GitLab autogéré (cf. l'article [0]).

Pour cela, il est possible de passer par le menu **Preferences > Integrations**. Il est alors nécessaire de générer un jeton d'accès (« access token ») sur votre GitLab en spécifiant les accès **api** et **read_user** comme demandé par GitKraken. Une fois le jeton généré, il suffit de le renseigner dans l'interface de GitKraken afin d'être connecté automatiquement à votre GitLab autogéré.

Enfin, il est nécessaire de configurer son avatar et ses données personnelles avant de pouvoir utiliser pleinement Gitkraken.

Tout est prêt, nous allons maintenant voir l'interface de GitKraken plus en détail.

» ÉTAPE 2 : DÉCOUVRONS L'INTERFACE

Disons-le tout de suite, l'interface de GitKraken n'est pas localisable et nous devons nous adapter à la langue de Shakespeare pour l'utilisation du logiciel. De toute manière, il est préférable de travailler dans la langue internationale pour les projets collaboratifs.

Commençons par les préférences de GitKraken. L'ensemble des réglages est disponible à partir du menu **File > Preferences** :

- L'onglet **General** vous permet de configurer les principales fonctionnalités de GitKraken, notamment les outils permettant la gestion des différences (« diff ») et de la fusion (« merge »), l'éditeur de texte (j'utilise ici atom) et le terminal par défaut.
- L'onglet **Profile** vous permet de définir différents profils afin de stocker les préférences de chacun (l'organisation de l'interface par exemple) ainsi que les informations Git.
- L'onglet **SSH** vous permet de choisir une paire de clés SSH en parcourant votre système de fichiers (local), ou de laisser GitKraken la générer pour vous. Avant de pouvoir cloner un projet, il est nécessaire de configurer vos clés SSH dans GitKraken. Ensuite, si ce n'est pas encore fait, il faudra copier la clé publique et la coller dans votre service d'hébergement (GitHub, GitLab, etc.) à distance afin de pouvoir accéder à ce dernier (normalement, cette partie est déjà configurée si vous avez un serveur GitLab autogéré). Les utilisateurs de GitHub pourront se tourner vers la fiche de l'adresse [7] pour une intégration simplifiée.

FUZZY FINDER

Le Fuzzy Finder est une fonctionnalité assez méconnue. Elle fournit un moyen plus rapide d'accéder aux fonctions de GitKraken. Vous pouvez utiliser le Fuzzy Finder pour ouvrir des dépôts, l'historique des fichiers dans le dépôt actuel et les paramètres. Pour ouvrir Fuzzy Finder, utilisez le raccourci clavier [Ctrl] + [P] et commencez à taper (Figure 1).

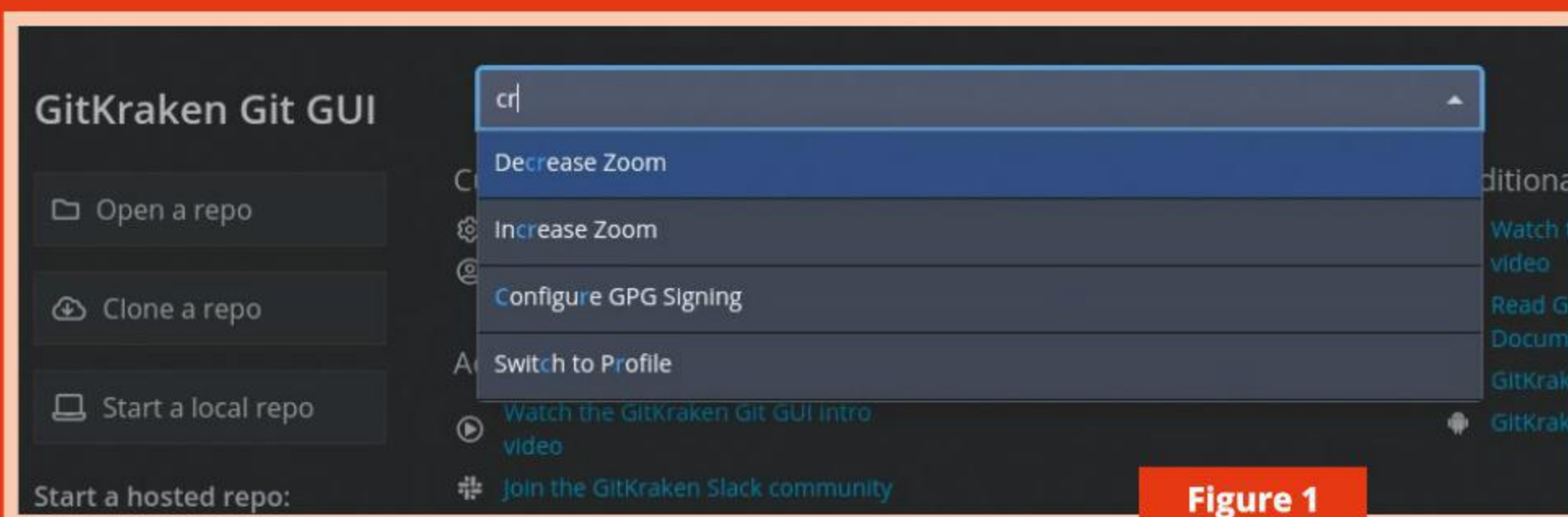
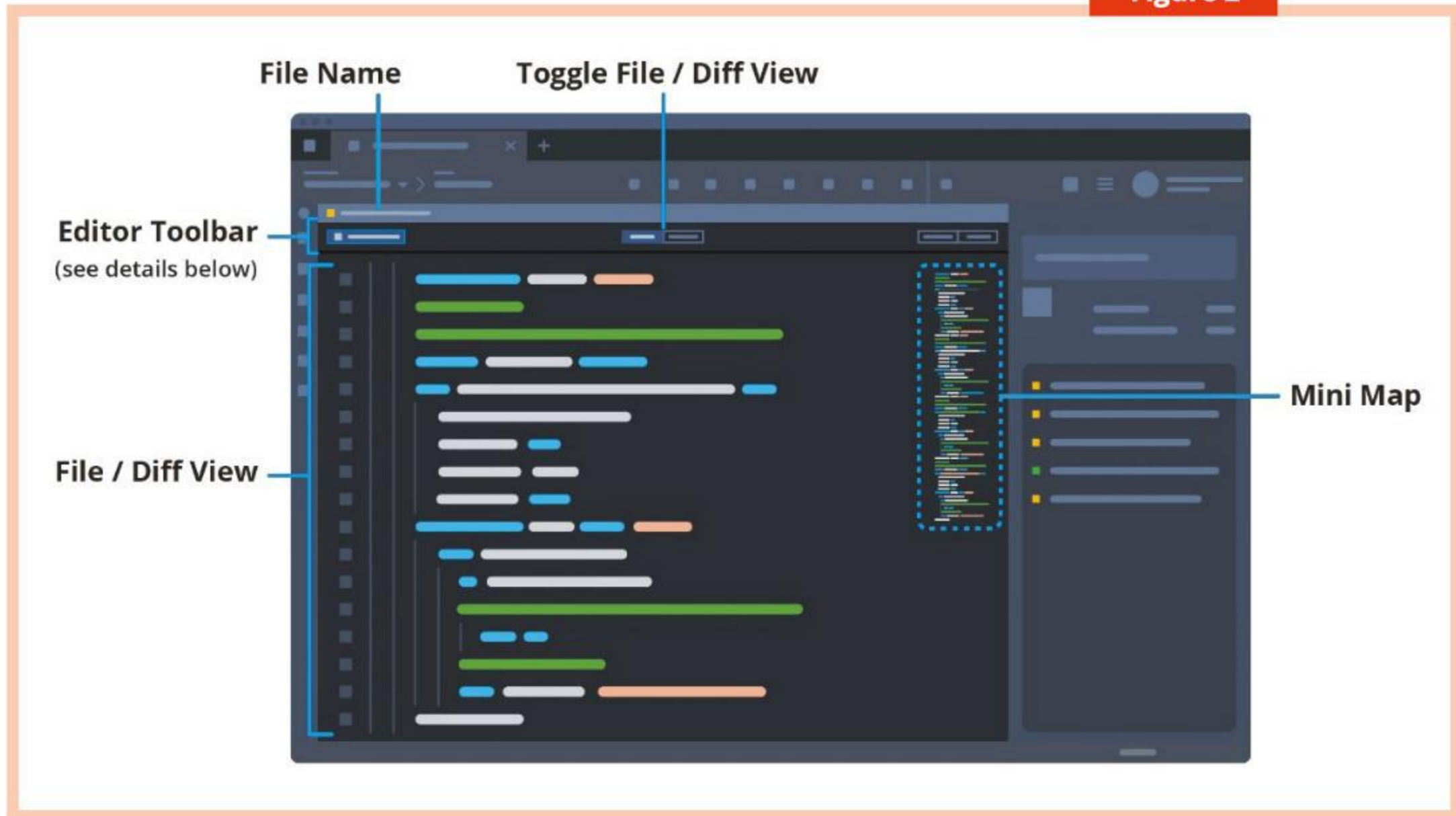


Figure 1

Figure 2



- L'onglet **Integration** vous permet de créer des comptes et de vous authentifier auprès de plusieurs services d'hébergement (GitHub, GitLab, etc.). C'est ce que nous avons fait dans l'étape d'installation. Vous pourrez alors accéder automatiquement à vos projets et les utiliser de manière transparente.
- L'onglet **UI customization** permet de choisir le thème graphique de votre interface ainsi que différentes options d'affichage. Les préférences suivantes sont disponibles : afficher les étiquettes des icônes, vérification orthographique, afficher les initiales de l'auteur au lieu des avatars (Gravatar), afficher l'auteur du commit dans le graphique, afficher la date/heure de l'engagement dans le graphique, etc.
- L'onglet **GPG** permet de configurer les paramètres de votre cryptosystème. Il vous permet d'assurer l'intégrité, la confidentialité et l'authenticité d'un message reçu. Si vous avez installé GPG sur votre système d'exploitation, vous pouvez alors configurer GitKraken afin qu'il l'utilise.
- L'onglet **Editor** permet de personnaliser les paramètres de votre éditeur : police, taille de la police,

taille des onglets, caractère de fin de ligne, mise en évidence de la syntaxe, afficher les numéros de ligne.

Le site GitKraken propose un fichier synthétisant les fonctionnalités les plus importantes du logiciel à l'adresse [7]. On y retrouve une vue synthétique de l'interface avec la description de chaque partie ainsi qu'un tableau répertoriant l'ensemble des raccourcis.

Cette interface est très complète et vous permet d'appréhender l'ensemble des zones de l'interface et leurs significations/fonctionnalités. Une bonne connaissance de Git est cependant nécessaire pour permettre la compréhension de l'ensemble des options. Pour cela, je vous renvoie vers l'adresse [9]. Dans cette série de vidéos d'apprentissage de Git, vous trouverez des tutoriels qui expliquent les concepts de Git pour niveaux débutants, intermédiaires et avancés.

Il est bien sûr possible d'éditer/comparer du code ou des versions de code dans GitKraken. Pour cela, il suffit de cliquer sur un fichier source de votre projet et l'interface se transforme comme dans le schéma de la figure 2.

Suivant la vue choisie **File View** ou **Diff View**, les options sont différentes. Nous verrons leur utilisation dans l'étape 4.

Deux boutons importants restent à être explorer : **GitKraken Boards** et **GitKraken Timelines**.

GitKraken Boards vous permet de créer des tableaux pour la gestion de vos projets. Depuis l'écran d'accueil de la figure 3, vous avez la possibilité de créer un tableau vide ou de créer un tableau à partir d'un modèle (Trello, CSV). Vous pouvez alors assigner des tâches quotidiennes et des dates butoirs aux collaborateurs de vos projets pour une gestion fine de l'avancement.

Les calendriers « Timelines GitKraken » sont conçus pour montrer les principales étapes sur une ligne continue qui représente le temps (Figure 4). Ceci

vous permet de communiquer les principaux objectifs ou jalons afin que les membres de l'équipe sachent ce qu'il va se passer, et quand. Alors que les outils de suivi des tâches comme les tableaux GitKraken sont parfaits pour gérer des tonnes de tâches quotidiennes, le calendrier GitKraken est le meilleur moyen de communiquer de manière plus générale sur le projet, de communiquer les initiatives et les échéances importantes lors des réunions.

Nous avons donc rapidement fait le tour des principales fonctionnalités de GitKraken. Pour aller plus loin dans les détails, je vous conseille le visionnage de la vidéo d'introduction à GitKraken à l'adresse [8] afin d'avoir un aperçu des fonctionnalités de ce logiciel et aussi la documentation officielle disponible à l'adresse [10].

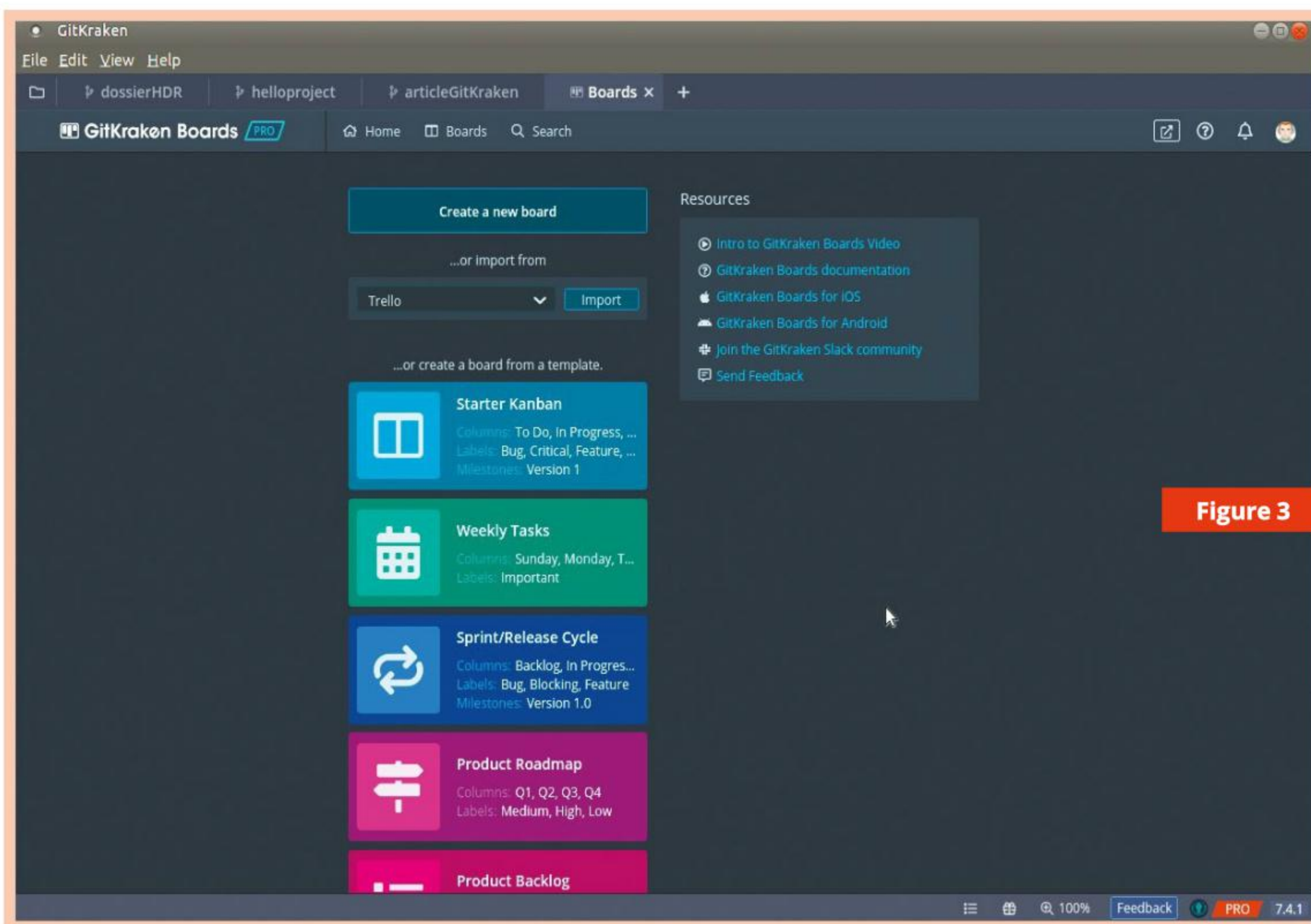


Figure 3

LISEZ DÈS À PRÉSENT TOUS NOS MAGAZINES EN LIGNE !



Simple. Rapide. Pratique.

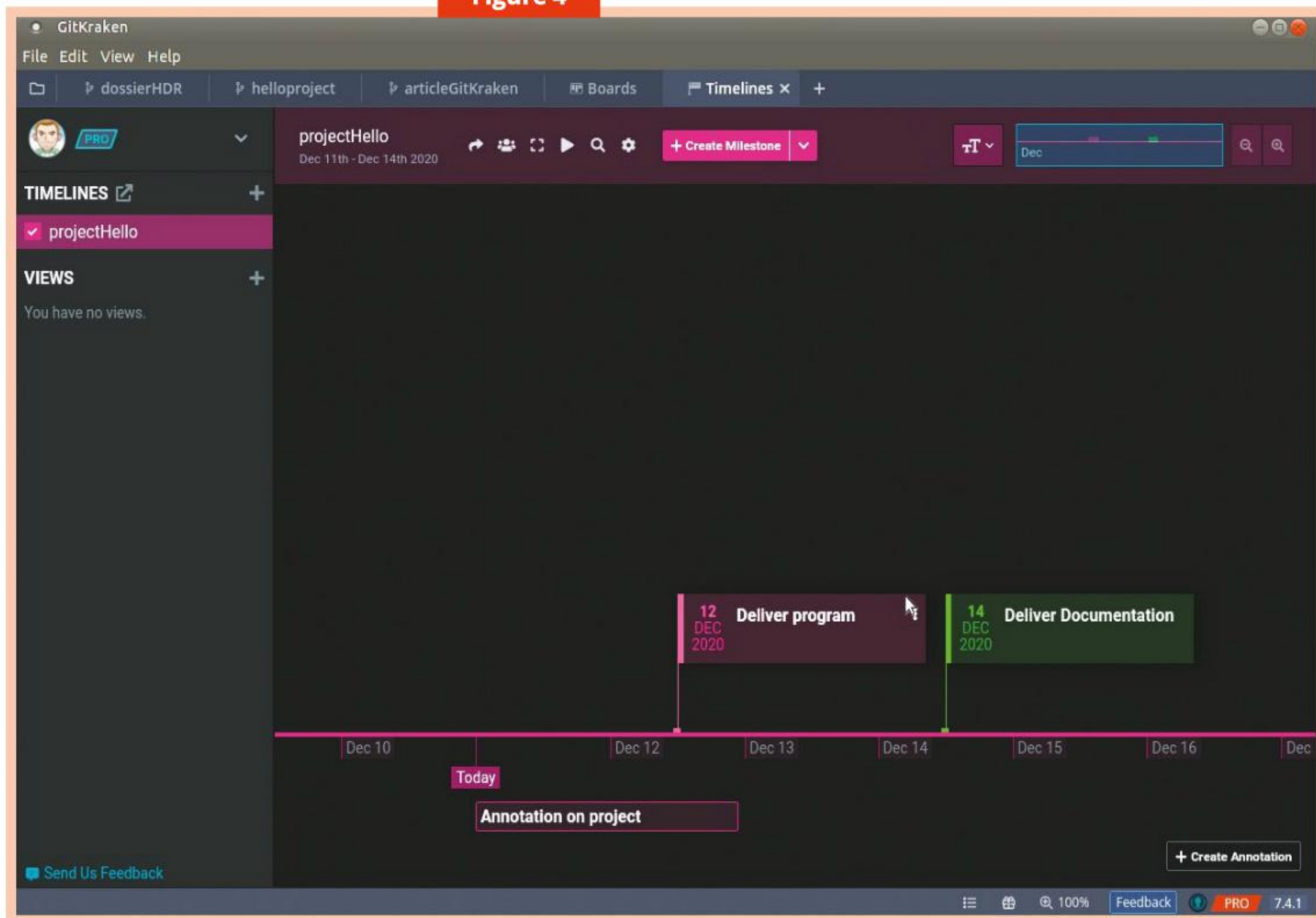


Tous nos magazines ont leur version Flipbook :



Rendez-vous sur www.ed-diamond.com

Figure 4



» ÉTAPE 3 : DÉMARRER UN PROJET SIMPLE AVEC GITKRAKEN

Dans la suite, nous allons créer un projet sur notre GitLab auto-hébergé (cf. [0]). Il s'agira du projet de cet article. En effet, j'utilise GitLab pour gérer l'ensemble de mes projets, que ce soit des projets rédactionnels (LaTeX, LibreOffice) ou de programmation.

La première étape consiste à créer le projet sur le serveur à l'aide de l'onglet **Init** de la fenêtre **Repository Management**. Sur mon dépôt autogéré, je sélectionne le groupe des articleslinux pour y placer le nouvel article nommé articleGitKraken. Il est en accès privé et je n'ai pas mis de description. J'ai ensuite choisi de clo-

ner localement le projet après sa création et j'ai aussi renseigné le chemin de stockage. On remarque que le nom de la branche par défaut est « main » et plus « master » afin de supprimer les références inutiles à l'esclavage et les remplacer par des termes plus inclusifs. Après avoir rempli la partie **Licence**, il ne reste plus qu'à appuyer sur le bouton **Create repository and clone**. Si tout se passe bien, votre nouveau projet apparaît dans GitKraken (Figure 5, page suivante).

Ayant déjà commencé l'article, je copie les fichiers dans le répertoire du projet. GitKraken m'indique alors que des fichiers ont été ajoutés au répertoire et m'invite à voir les changements puis à ajouter ces nouveaux fichiers à l'aide d'un simple clic sur le bouton **Stage all change**.

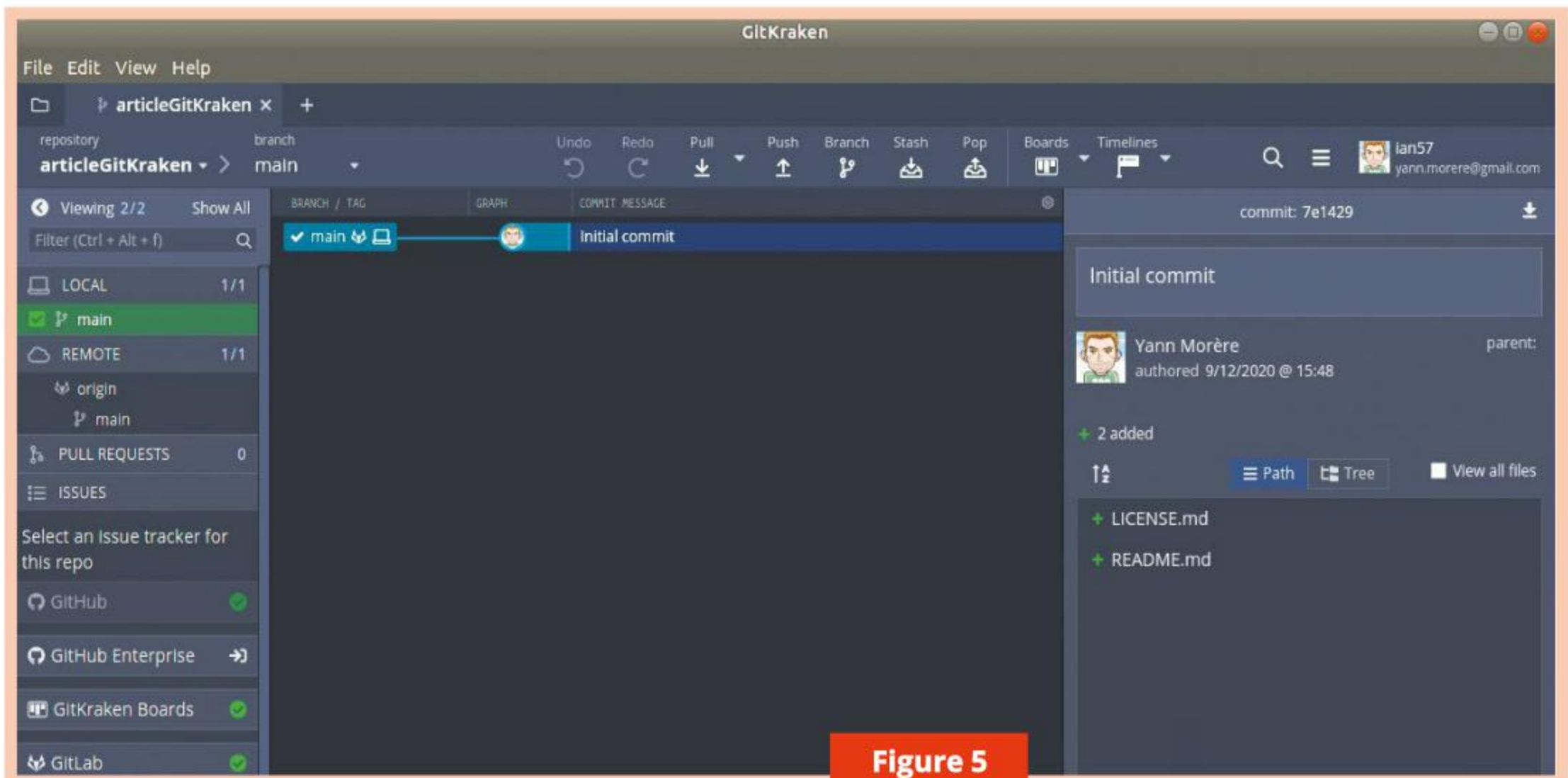


Figure 5

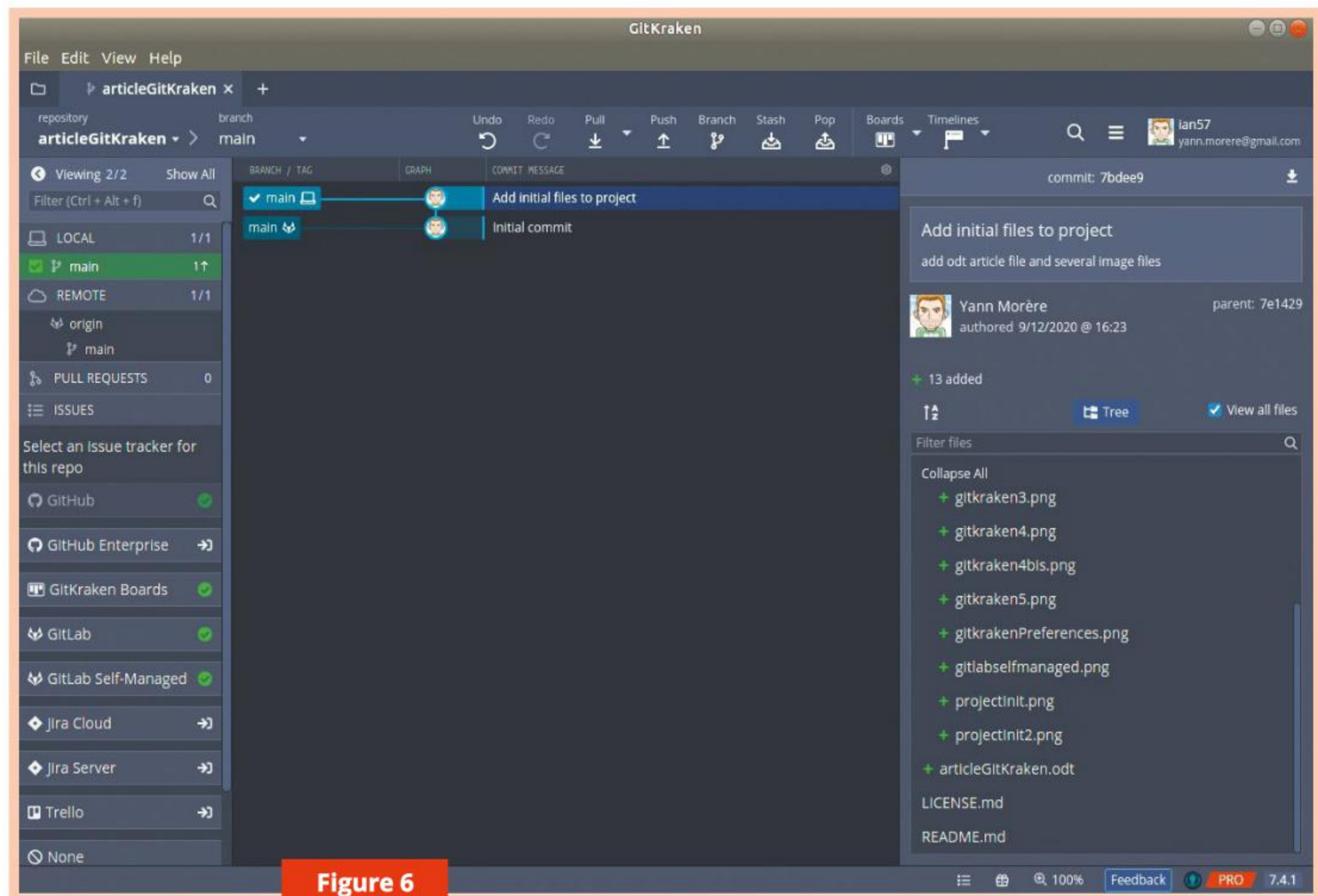


Figure 6

Une fois les fichiers ajoutés (« staging »), il faut remplir la partie « commit message » afin de renseigner les informations des modifications réalisées. Ensuite, il suffit d'appuyer sur le bouton **Commit changes to files** et la partie centrale de GitKraken affiche la nouvelle validation « commit » (Figure 6). Il est bien sûr aussi possible de valider les changements fichier par fichier.

Pour l'instant, toutes nos modifications sont locales et sur notre serveur aucun fichier ajouté n'est encore présent. Un simple appui sur le bouton **Push** va remonter nos modifications sur le serveur GitLab.

Il est possible de revenir simplement sur une des versions précédentes de notre travail pour récupérer une section effacée par exemple. Pour cela,

il suffit de placer la souris sur le « commit » désiré et un clic droit nous donne le menu contextuel. On choisit alors **Checkout this commit** (Figure 7). La commande **Revert commit** permet quant à elle de supprimer intelligemment un commit en déterminant comment annuler les changements introduits par le commit et ajoute un nouveau commit avec le contenu ainsi obtenu.

Il est aussi possible d'utiliser le bouton **Undo** présent au centre de l'interface.

Nous avons ici présenté une utilisation très simple de GitKraken sur un exemple utilisant des fichiers binaires et non du code source. Cette utilisation n'est pas la plus courante, mais vous permet d'avoir une sauvegarde et un historique de l'évolution de votre travail.

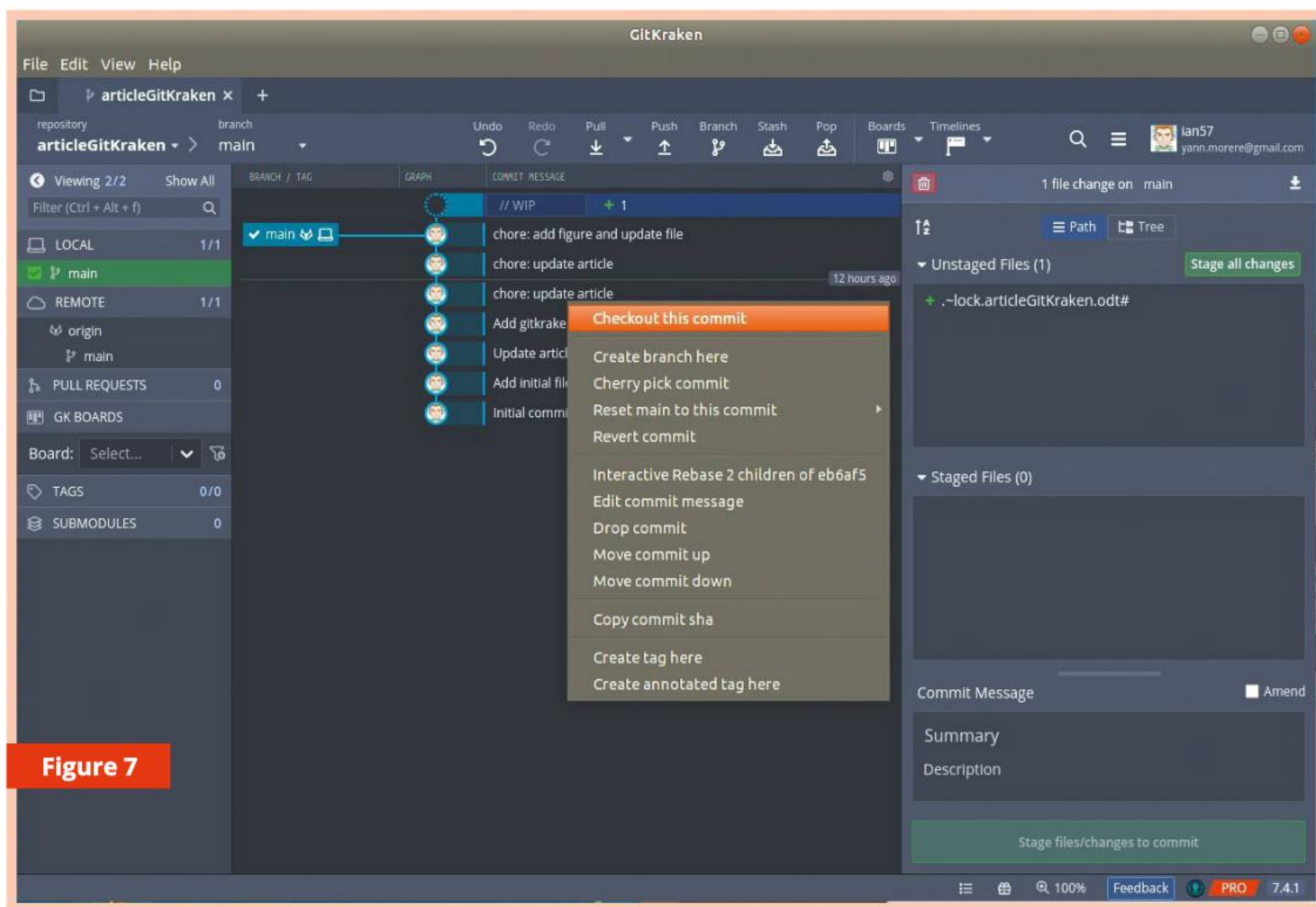


Figure 7

» ÉTAPE 4: DÉMARRER UN PROJET CODE SOURCE AVEC GITKRAKEN

Nous allons maintenant démarrer un mini projet basé sur du code source C et utiliser d'autres fonctionnalités de GitKraken. On commence par générer notre projet depuis le serveur GitLab comme précédemment.

Je vais utiliser l'éditeur de texte Atom afin d'ajouter le fichier source « [hello.c](#) ».

```
/* sample program for gitkraken article*/
/* yann morere 2020 */
#include <stdio.h>
int main(void){
    printf("Hello GitKraken!\n");
    return 0;
}
```

BON À SAVOIR

Pour l'ajout de code source, il est tout à fait possible de le faire depuis l'interface de GitKraken, ce dernier possédant son propre éditeur. Par exemple, ouvrez le Fuzzy Finder avec [Ctrl]+[P], tapez *Create File*, et appuyez sur [Entrée]. Tapez ensuite le nom de fichier désiré et cela crée un nouveau fichier texte dans le répertoire courant du projet.

Dans l'interface de GitKraken, on ajoute le nouveau fichier (« stage file ») à Git et on réalise la validation des modifications localement « commit ». On peut alors pousser « push » nos modifications sur le serveur.

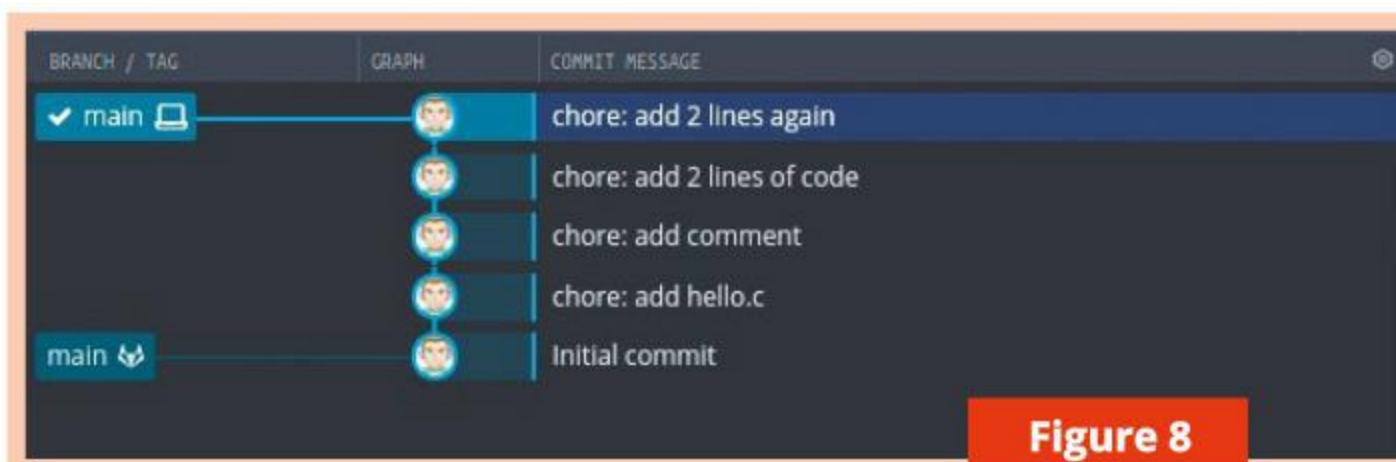


Figure 8

Continuons à faire quelques modifications dans notre fichier en ajoutant quelques lignes, soit avec votre éditeur préféré, soit directement avec l'éditeur de GitKraken. On réalisera une validation des modifications après chaque ajout (« stage » et « commit ») pour avoir au final le code suivant :

```
/* sample program for gitkraken article*/
/* yann morere 2020 */
/* test gitkraken editor */
#include <stdio.h>

int main(void){
    printf("Hello GitKraken!\n");
    /* commit for each line added*/
    printf("Hello world!\n");
    /* commit for each line added*/
    printf("Hello LinuxPratique!\n");
    return 0;
}
```

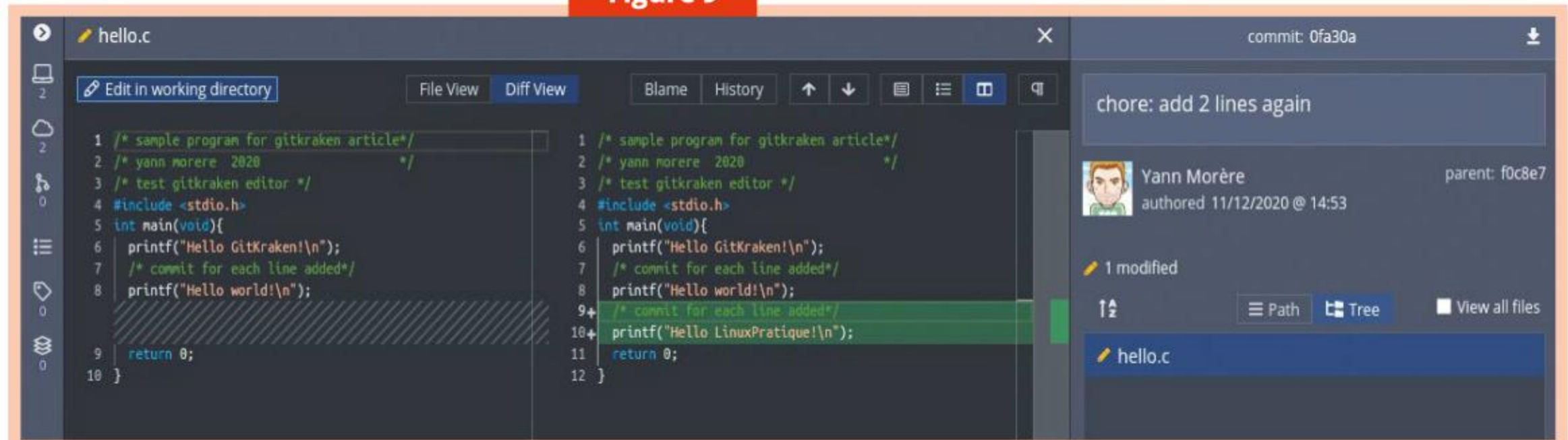
Dans la figure 8, on visualise l'évolution de notre projet. Chaque « commit » est un élément interactif et un clic droit vous affiche un menu contextuel avec les opérations réalisables à partir de cette étape du projet.

En sélectionnant un « commit », puis en cliquant sur le fichier source [hello.c](#) vous obtenez un aperçu des différences entre la version sélectionnée et la version précédente (Figure 9) d'une manière très pratique et très visuelle.

Imaginons que vous désiriez tester une nouvelle fonctionnalité dans votre programme en toute sécurité par rapport au code source existant et fonctionnel. Pour cela, nous allons créer une nouvelle branche de développement, travailler dans celle-ci sans risquer de corrompre le code source de la branche principale « main ».

La création d'une nouvelle branche se fait simplement par l'intermédiaire du menu contextuel d'un « commit ». Placez-vous sur le « commit » à partir duquel vous désirez créer la branche et faites un clic droit puis **Create Branch Here**. GitKraken vous propose alors

Figure 9



de saisir le nom de la nouvelle branche de développement (ici nommée « myTestBranch »). Une fois le nom validé, vous êtes alors automatiquement basculé sur la nouvelle branche (« checkout ») et toutes les modifications seront réalisées sur cette dernière. On peut alors travailler sans souci sur notre nouvelle branche.

Une fois les modifications testées et validées dans cette branche de test, on pourra la fusionner avec la branche principale « main ». Pour cela, on se place sur le commit de la branche de test « myTestBranch » et on sélectionne **Merge myTestBranch Into main** dans le menu contextuel. Puis, finalement, on bascule sur la branche « main » pour continuer à travailler sur la branche principale une fois les modifications intégrées.

Je m'arrête ici, et vous laisse découvrir par vous-même la puissance de ce logiciel. Maintenant c'est à vous de jouer...

CONCLUSION

Nous avons terminé ce petit tutoriel de mise en œuvre de GitKraken pour la gestion graphique de vos projets utilisant Git. Nous n'avons bien sûr pas exploré tous les arcanes de Git à travers GitKraken, notamment les « issue » et « merge request » par exemple, ou encore les « squash » de « commit » qui permettent de regrouper entre eux plusieurs « commit » successifs et autres « cherry pick » permettant d'intégrer des « commit » arbitraires. Je vous laisse découvrir cet outil en détail grâce à sa documentation très complète. ■

RÉFÉRENCES

- [0] <https://connect.ed-diamond.com/Linux-Pratique/LP-120/Utilisez-GitLab-pour-la-gestion-globale-de-vos-projets-en-equipe>
- [1] <https://www.gitkraken.com/>
- [2] <https://github.com/FredrikNoren/ungit>
- [3] https://en.wikipedia.org/wiki/Comparison_of_Git_GUIs
- [4] <https://www.gitkraken.com/teacher-resources>
- [5] <https://www.gitkraken.com/student-resources>
- [6] <https://www.gitkraken.com/pricing#git-gui-features>
- [7] <https://www.gitkraken.com/resources/gitkraken-github-cheat-sheet>
- [8] <https://youtu.be/ub9GfRziCtU>
- [9] <https://www.gitkraken.com/resources/learn-git>
- [10] <https://support.gitkraken.com/>



Chez votre marchand de journaux
et sur **www.ed-diamond.com**



en kiosque



FLIPBOOK HTML5

sur **www.ed-diamond.com**



CONNECT
LA DOCUMENTATION TECHNIQUE DES PROS DE L'IT

sur **connect.ed-diamond.com**

À LA DÉCOUVERTE DE L'OPEN DATA : PETIT MANUEL DE SURVIE DANS LE MILIEU DE LA DONNÉE

TRIS ACATRINEI

LA REMISE AU PREMIER MINISTRE DU RAPPORT BOTHOREL SUR LA POLITIQUE PUBLIQUE DE LA DONNÉE, DES ALGORITHMES ET DES CODES SOURCES PERMET DE PROCÉDER À UN ÉTAT DES LIEUX DE L'OPEN DATA EN FRANCE. DE QUOI PARLE-T-ON QUAND ON PARLE D'OPEN DATA ? QUELS SONT LES CRITÈRES ET LES FORMATS ? COMMENT UNE STRUCTURE PEUT-ELLE SE LANCER DANS L'OPEN DATA ?



L'open data représente un pan de la nouvelle économie, que certains qualifient de mine d'or, en fonction des domaines. Dans le contexte de l'épidémie

de COVID-19, il est instamment demandé que les résultats de la vaccination soient disponibles en open data. En matière immobilière, cela peut être un

véritable outil de support pour un achat, car le croisement de différentes données permet d'obtenir une expertise extrêmement fine du logement, de

sa valeur, de son emplacement et donc de sa qualité de vie. De quoi parle-t-on précisément quand on évoque l'open data ?

1. OPEN DATA : LES CRITÈRES

Lorsque l'on parle d'open data, on parle évidemment de données, mais si on se réfère à la définition juridique, « représentation d'une information sous une forme conventionnelle destinée à faciliter son traitement », on est face à une définition beaucoup plus large qui ajoute le code source d'une application, un algorithme ou encore une métadonnée. Dans cet article, on privilégiera uniquement la data, pour laisser de côté le champ des codes sources et des algorithmes.

La définition juridique est intéressante, car elle inclut la compétence juridictionnelle lorsque la donnée est publique — c'est-à-dire fournie par un organisme de droit public — à savoir la Commission d'Accès aux Documents Administratifs (CADA). Cette attribution est confirmée avec l'article L.300-2 et L.321-1 du Code des relations entre le public et l'administration (CRPA). Mieux encore, sous l'égide du droit communautaire, même si le producteur de la donnée n'est pas une entité publique, il est possible de saisir la CADA pour demander les données en open data, un code source ou un algorithme. Le critère retenu n'est pas la nature

Open data ne rime pas nécessairement avec public ou service public. En effet, il existe de plus en plus de structures de droit privé qui jouent le jeu de l'open data et proposent des bases de données aux internautes.

de l'établissement dans ce cas de figure, mais l'exécution d'une mission de service public.

Prenons l'hypothèse d'un prestataire qui gèrerait une mission de service public pour le compte de l'État, au hasard, les vœux des bacheliers pour leurs études supérieures. Si cette mission est dévolue à une entreprise privée, dans la mesure où il s'agit de la mise en œuvre d'une politique publique, et donc d'un service public, il est possible de saisir la CADA pour demander le code source de l'application qui gère les vœux, la documentation et toutes les informations nécessaires, visant à éclairer le public sur la décision publique.

Open data ne rime pas nécessairement avec public ou service public. En effet, il existe de plus en plus de structures de droit privé qui jouent le jeu de l'open data et proposent des bases de données aux internautes, sans pour autant qu'ils soient considérés comme un service public. Néanmoins, contrairement à

une entité de droit public, un individu ne pourra pas exiger d'une entreprise qu'elle libère des données. Par ailleurs, il existe certaines hypothèses où un organisme public ne pourra pas procéder à l'open data, c'est notamment le cas pour les données de santé. Ainsi, dans le cas de l'application « TousAntiCOVID », si le code est ouvert, les données ne peuvent pas être totalement libérées dans la mesure où des individus signalent leurs déplacements et d'éventuels cas contacts. L'état de santé d'une personne relève du champ de la vie privée, on ne peut donc pas exiger une ouverture totale des informations.

Une base de données mise à disposition du public, peut-elle être qualifiée d'open data ? Pour cela, elle doit répondre à certains critères. En premier lieu, les données doivent être accessibles, aussi bien sur le plan technique qu'humain. Une donnée qui nécessiterait l'installation et la maîtrise d'un certain type de logiciel ne peut pas être considérée comme étant en open data. De la même manière, cette donnée doit être interopérable, c'est-à-dire qu'elle ne doit pas être subordonnée à un environnement technique spécifique. Un fichier qui ne serait disponible qu'au format Excel n'est pas de l'open data. Évidemment, un fichier Excel peut être ouvert avec Libre Office ou Open Office — on laissera au lecteur le choix de sa suite bureautique

préférée — mais cela n'en fera pas de l'open data pour autant. Enfin, la donnée doit être lisible. Une base de données qui n'aurait que des matricules sans schéma de référence ni de documentation claire, permettant de comprendre à quoi correspond le matricule n'est pas de l'open data.

Enfin, il faut distinguer la donnée des autres productions intellectuelles. Une donnée n'est pas un document administratif. Pour illustrer le propos, on peut se référer aux différents calculs de temps de parole du CSA. Pour l'année 2007, le CSA fournit des PDF avec le temps de parole du Président de la République, de son gouvernement et des différents partis politiques. C'est un document administratif, mais pas une base de données en open data. De la même manière, une information contenue sur un site web public n'est pas de l'open data. Prenons l'exemple des rapporteurs de texte au Parlement et plus particulièrement, des rapporteurs de lois de finances.

Une donnée qui nécessiterait l'installation et la maîtrise d'un certain type de logiciel ne peut pas être considérée comme étant en open data.

Les habitués le savent : pour l'examen d'une loi de finances annuelle, il existe un rapport général et des rapporteurs thématiques dont le nombre varie en fonction des chambres. Sur le site de l'Assemblée nationale, la liste est affichée sous forme de carrousel ainsi qu'avec un tableau en bas de page. L'information est publique, mais n'est pas mise en open data dans le répertoire de l'Assemblée nationale. Il faut constituer cette base de données à la main. Enfin, un document public n'est pas non plus de l'open data. Restons au Palais Bourbon et prenons l'exemple de la feuille jaune. Ce document — qui tire son nom de sa couleur — est parfaitement public. Il s'agit d'une liste des orateurs qui s'exprimeront pendant une séance dans l'hémicycle. Elle est mise à jour en temps réel. Mais elle n'est pas « transformée » en base de données quelconque.

Quels sont les buts de l'open data ? Le premier est de permettre l'innovation, notamment l'innovation par le secteur privé. En matière de transports, certaines applications, qui ont vu le jour grâce à la libération des données, sont plus performantes que les applications du secteur public. Au niveau local, on observe que des initiatives privées ou associatives sont plus en adéquation avec les demandes des populations. Auraient-elles pu être développées en interne, au sein des administrations ?

Objectivement, oui, mais en pratique, les choses sont plus délicates. Tout d'abord, les administrations, même en ayant recours à des contractuels — donc soumis au droit privé — doivent répondre à certaines exigences et ne peuvent matériellement pas offrir la même flexibilité qu'une structure privée. L'illustration la plus simple est celle du télétravail. Dans une entreprise, peu importe la forme, un dirigeant peut mettre en place, de manière assez souple, pourvu qu'il ait la trésorerie, le télétravail à 100 %. Pendant les confinements, nombreux sont les agents titulaires de la fonction publique à avoir rapporté anonymement, être forcés de se rendre sur site.

Pour les décideurs publics, la mise en œuvre de l'open data répond à trois volontés. La première est de faire de la communication sur son action. En s'appuyant sur des données, il est plus facile de quantifier et de qualifier une politique publique. Cela permet également de la



transparence sur cette dernière. Il ne s'agit pas de se référer à des communiqués ou des dossiers de presse, mais de montrer et de prouver que telle décision a eu tel effet. Dans un État décentralisé et déconcentré, l'open data permet aux citoyens de s'emparer d'une thématique et éventuellement d'instaurer un lien de confiance. Le dernier point qui rejoint celui de l'innovation est celui de l'aide à la décision publique. En libérant les données, une administration peut permettre à une entité privée de répondre à un manque et prendre les décisions nécessaires à l'échelon local ou national.

L'un des avantages de l'open data — quand elle est correctement mise en œuvre — est sa facilité d'accès. Les puristes peuvent utiliser R ou Python, mais lorsque les administrations respectent à la lettre le principe d'accessibilité, Libre Office, Open Office ou même Excel peuvent largement faire l'affaire et cela, en raison des formats utilisés.

2. LES FORMATS DE L'OPEN DATA

En matière d'open data, la clef de voûte reste le format. Comme dit précédemment, même en tant qu'entité privée, il est tout à fait possible de contribuer à l'open data. Le choix des données à libérer est évidemment à la discrétion des structures. En dehors de la gestion des données personnelles et de l'anonymisation nécessaire, ce dernier est vaste. On recommandera néanmoins de prendre attache avec la CNIL afin de s'assurer que cette volonté d'ouverture ne rentre pas en contradiction avec les dispositions du RGPD.

Il faudra également étudier les différentes licences. La plus simple à mettre en œuvre est la Licence Ouverte/Open License. La licence Open Database License ou OBDL est également utilisée, mais dans le cadre d'une entreprise qui libérerait ses données, elle n'est pas la mieux adaptée. Là encore, en cas de doute sur la licence adéquate, le mieux est de prendre un conseil extérieur, notamment auprès des professionnels du droit.

Une fois cette analyse préalable franchie, quel format choisir ? Passons en revue les possibilités.

Le plus classique, le plus utilisé et le plus simple d'accès reste le CSV. Cet acronyme désigne « Comma-separated values » ou « valeurs séparées par une virgule ». C'est une feuille de calcul

” Cette donnée doit être interopérable, c'est-à-dire qu'elle ne doit pas être subordonnée à un environnement technique spécifique.

dans laquelle les informations sont séparées par une virgule. L'encodage le plus fréquent est l'UTF-8, qui a l'avantage d'être compris par la plupart des logiciels de base de données, dont MySQL. Il n'y a pas de spécifications formelles même s'il existe un RFC, à savoir le RFC 4180. Pour qu'il s'agisse d'un document CSV, les données doivent être enregistrées ainsi :

```
aaaa,bbbb,cccc,dddd
```

Les données peuvent être séparées par des virgules, mais peuvent également être enchâssées dans des doubles guillemets. Ainsi, cela évite la perte lors d'une intégration si l'expression en elle-même contient une virgule. L'exemple le plus courant est celui des adresses postales. Si les données ne sont séparées que par des virgules, on va éclater l'information en plusieurs colonnes. On n'écrira donc pas 126,rue de l'université, mais « 126, rue de l'université », afin de ne pas perdre l'information. Cela peut également avoir

une importance si les données en question sont des nombres avec des décimales. La virgule est la séparation la plus communément admise. Pour autant, utiliser le point-virgule avec ou sans doubles guillemets pour séparer les informations est également recevable sur le plan technique. À titre d'exemple, on peut se référer au module Feeds dans l'environnement Drupal. Ce dernier permet d'inclure des données dans des contenus de façon massive, sur la base d'un fichier CSV. Il suffit de lui indiquer quelle est la séparation pour qu'il interprète correctement le fichier.

Enfin, on notera que le CSV est celui qui est le plus simple à créer. Open Office et Libre Office le font parfaitement, pour peu qu'on indique les bons paramètres (séparation, encodage). Il existe de très nombreux convertisseurs en ligne gratuits. Quant à Excel, au-delà du fait qu'il s'agisse d'un logiciel propriétaire, on ne le recommandera pas pour

générer les CSV. Certaines administrations ont fait le choix de générer des CSV pour Excel, mais les systèmes de gestion de base de données tels que MySQL ont tendance à ne pas les comprendre.

Le CSV est adapté pour la donnée qui n'est pas hiérarchisée. Mais si vos données doivent faire apparaître une hiérarchie, le JSON et le XML sont recommandés.

JSON est l'acronyme de Javascript Object Notation. Si vous êtes allergique à JavaScript, passez directement à la partie qui aborde le XML. La force du JSON réside dans la hiérarchisation des informations, mais également le fait qu'il n'y a pas besoin de parser (procéder à l'analyse syntaxique), qu'il est simple à mettre en œuvre, à interpréter et rapide.

Petite subtilité du format, il possède deux normes concurrentes : le RFC 8259 et l'ECMA 404. Dans les lignes qui suivent, on se basera sur

le RFC 8259. On l'a dit, le JSON sert à hiérarchiser la donnée. En CSV, l'information est calibrée ainsi : entêtes : « entité », « attribut a », « attribut a1 » et corps : « 1 », « random », « random ». En JSON, la donnée va être schématisée ainsi :

```
{
  " entité " : " 1 "
  " attribut a " : " random "
  " attribut a1 " : " random "
}
```

Quel est l'intérêt ? Le CSV peut être maniable jusqu'à un certain niveau d'information. Imaginons un fichier CSV qui contiendrait 577 lignes, mais une cinquantaine de colonnes. Au-delà du fait qu'il va falloir un certain temps pour que le tableur puisse se charger correctement, il n'y a pas de hiérarchisation simple de l'informatique.

La séparation des éléments demande une certaine gymnastique intellectuelle, mais la plupart des éditeurs savent corriger si besoin les fichiers JSON. Pour les personnes qui sont plus habituées à manier le CSV que le JSON, il existe de nombreux convertisseurs en ligne. Tout va dépendre du besoin de la personne qui met en ligne les datas et de celles qui vont les récupérer.

Autre format, très concurrent de JSON : le XML, pour Extensible Markup Language. Son RFC est la 3470 et est utilisable pour les flux RSS. Il est très simple à mettre en œuvre et à utiliser, même s'il n'est pas aussi simple que les fichiers au





Chez votre marchand de journaux
et sur **www.ed-diamond.com**



en kiosque



FLIPBOOK HTML5

sur **www.ed-diamond.com**



CONNECT
LA DOCUMENTATION TECHNIQUE DES PROS DE L'IT

sur **connect.ed-diamond.com**

format CSV. Cela reste un fichier texte relativement basique et à partir du moment où vous avez compris le système de balisage des informations, vous avez compris l'essentiel. Le fichier se présente ainsi :

```
<<?xml version='1.0' encoding='utf-8' ?>
<Informations>
<Information-2><Elements>Texte</Elements></Information2>

</Informations>
```

En résumé, c'est assez similaire à une page web. Point commun entre le JSON et le XML, ils sont parfaitement adaptés pour les données géographiques. Si les informations que vous souhaitez mettre en open data sont des informations impliquant une géolocalisation, le plus simple est le geoJSON ou le KLM, qui est lui-même basé sur XML. On les retrouve d'ailleurs dans les outils de cartographie.

Certains services publics ou entreprises proposent les trois formats. Le choix se fait en fonction des systèmes d'information, mais aussi des demandes du public visé.

3. MISE EN ŒUVRE DE L'OPEN DATA

Pour une structure, mettre en place une politique d'open data revient aussi à s'interroger sur son déploiement. Il ne suffit pas de mettre quelques fichiers en open data, un lien en page renvoyant vers le répertoire, pour dire qu'on a mis en place une politique d'open data. La notion s'accompagne du concept d'innovation. Cette dernière implique de faire connaître et de faire participer. Plusieurs options s'offrent aux personnes désireuses de se lancer dans une telle aventure.

La première consiste à créer son propre écosystème. Si les données ne sont pas d'utilité publique — dans le sens où elles ne relèvent pas du service public — qu'elles s'adressent à une certaine catégorie de professionnels ou que la structure souhaite fidéliser des internautes, la solution de facilité est de créer sa propre plateforme.

On peut parfaitement recourir à un CMS (Content Management System) de type WordPress ou Drupal, en utilisant les modules et les extensions adéquates. On peut également utiliser un CMS dédié à ce type de problématiques : CKAN. Utilisée par plusieurs gouvernements et organismes publics, il s'agit d'un outil clef en main. Comme l'indique la documentation, au lieu d'avoir un site sur lequel on publie des articles, on publie des jeux de données. Le gros avantage est que l'outil moissonne lui-même les jeux de données qu'on lui indique.

Du côté des utilisateurs et des contributeurs de données, CKAN est très simple à utiliser et maniable. L'interface ressemble à n'importe





En matière d'open data, la clef de voûte reste le format.

quel autre CMS. C'est du côté de l'installation et de la configuration que les choses sont un peu complexes. En clair, si vous n'avez pas de compétences particulières en administration système, vous serez très perplexe devant l'outil. Par ailleurs, il n'est pas impossible que vous soyez dans l'obligation d'avoir votre propre serveur dédié afin de procéder à la configuration. Un hébergement

de type Simple Hosting chez Gandi ne sera peut-être pas adapté. Or, avoir un serveur dédié implique une ligne budgétaire supplémentaire et il vaut mieux avoir un administrateur système ou au moins une personne ayant quelques compétences dans ce domaine, dans l'équipe. C'est pourquoi la solution du recours à CKAN sera à privilégier pour les structures de taille conséquente ou avec un fort potentiel technique.

Si les ressources internes sont absentes, la possibilité de créer un point d'entrée sur un site web est faisable. Le portail de statistiques Eurostat, qui dépend de la Commission européenne, a fait le choix d'une plateforme, disponible en trois langues, sur laquelle, les statistiques et les données sont téléchargeables. L'institution propose des données mises à jour deux fois par jour, au format CSV, mais aussi une consultation interactive en ligne. Ce choix a été fait pour viser un

certain public. Utiliser un outil comme CKAN aurait pu être une option, mais l'objectif n'était pas nécessairement de viser des développeurs ou même des habitués de la data, mais plus simplement du personnel administratif, politique et des journalistes.

Le choix peut donc dépendre du public visé. Si les données à mettre en open data sont des données d'intérêt général, remplissant une mission de service public ou des informations locales, il est aussi possible de recourir à data.gouv.fr. Contrairement à ce que le lien peut laisser penser, la plateforme n'est pas réservée à des contributeurs issus de l'administration. Des citoyens, des associations et même des entreprises contribuent régulièrement à l'enrichissement des données. Comme l'indique la documentation « tout le monde peut publier des données sur data.gouv.fr, du moment qu'il s'agit de données d'intérêt public. ». Ainsi, si vous souhaitez constituer une liste des endroits où il est possible de recycler proprement du matériel informatique dans votre département, cela répond à une problématique d'intérêt public, cela sera donc accepté a priori. Vous pouvez parfaitement contacter Etalab pour leur poser directement la question. Veillez à indiquer précisément les jeux de données que vous souhaitez téléverser ainsi que la problématique à laquelle vous entendez répondre.

On n'y pense pas forcément, mais GitHub peut aussi être une bonne alternative pour héberger des répertoires de données. Imaginons que vous ayez des jeux de données, qui ne sont pas amenés à être mises à jour au quotidien, par exemple des données comptables pour une année spécifique. L'option peut paraître exotique, mais selon les cas, être adaptée. Évidemment, le libriste qui lira ses lignes risque de grincer des dents. En effet, GitHub a été racheté par Microsoft. Heureusement, il existe des alternatives libres ou open source. GitLab semble être celle qui est la plus plébiscitée.

En résumé, selon le public que l'on souhaite toucher, les moyens internes et les jeux de données, il existe des possibilités diverses, qui ont l'avantage d'être adaptées à quasiment tout le monde.

Néanmoins, une fois que les données sont hébergées quelque part, encore faut-il

en informer les personnes qui y ont intérêt. Il y a les newsletters, les réseaux sociaux et les forums. Pour les réseaux sociaux, on privilégiera Twitter, LinkedIn, mais aussi Mastodon, qui comporte une grande communauté de gens impliqués dans l'open data. Pour les forums, l'un des boosters de trafic reste clairement Reddit, mais également Hacker News. Le système d'agrégation et de mise en avant communautaire peuvent constituer un vrai lever de popularité.

Enfin, on peut toujours avoir recours aux événements physiques lorsque les circonstances le permettent : une conférence avec la présentation d'un outil qui a permis de créer les données, une rencontre entre professionnels ou un barcamp.

CONCLUSION

Comment améliorer l'open data en France et inciter plus d'acteurs à y contribuer ? Une des pistes de réflexion consisterait à donner un label, mais surtout un statut aux producteurs de données, quand ces derniers ne sont pas issus de l'administration. En effet, pour que les acteurs privés participent, ils doivent y trouver une motivation, que celle-ci soit honorifique, financière ou autre. Sur ce point, l'État n'est pas arrivé à maturité, mais le rapport Bothorel est un premier pas en ce sens. ■

” **Comment améliorer l'open data en France et inciter plus d'acteurs à y contribuer ? Une des pistes de réflexion consisterait à donner un label, mais surtout un statut aux producteurs de données.**



**Franck a le pouvoir
d'ouvrir l'école
à tous les enfants
pour 2,50€/mois***

Grâce au don mensuel
on a tous le pouvoir de **changer**
le monde, même à distance !

hi.fr/don.mensuel

Franck G., donateur régulier pour Handicap International depuis 2005



**handicap
international**



BON DE SOUTIEN RÉGULIER

À RENVOYER À L'ADRESSE :
HANDICAP INTERNATIONAL LIBRE RÉPONSE N° 45134 69129 LYON 08

☐ **OUI,** je souhaite soutenir dans la durée les actions de Handicap International

Merci de compléter vos informations ci-dessous :

☐ M. ☐ Mme ☐ Mlle

Nom :

Prénom :

E-mail :@.....

vosre adresse e-mail sera utilisee exclusivement par Handicap International

Adresse :

Code postal : | | | | | Ville :

Je choisis le montant de mon soutien

☐ **10€/mois**
soit 2,5€ après réduction fiscale

☐ **20€/mois**
soit 5€ après réduction fiscale

☐ **30€/mois**
soit 7,5€ après réduction fiscale

Le premier prélèvement devra avoir lieu le 10 du mois de2021
Je pourrai faire suspendre l'exécution de ce prélèvement par simple demande à l'association, signifiée au plus tard le 20 du mois précédent.
Je recevrai un reçu fiscal annuel pour l'ensemble de mes dons.

MANDAT DE PRÉLÈVEMENT SEPA Désignation du compte à débiter (merci de joindre un RIB) :

[illegible][illegible]

Signature :
(obligatoire)

Fait à

le | | / | / | | |

En signant ce formulaire de mandat, vous autorisez Handicap International à envoyer des instructions à votre banque pour qu'elle débite votre compte et votre banque à débiter votre compte conformément aux instructions de Handicap International. Vous bénéficiez du droit d'être remboursé(e) par votre banque selon les conditions décrites dans la convention que vous avez passée avec elle. Toute demande de remboursement doit être présentée dans les 8 semaines suivant la date de débit de votre compte pour un prélèvement autorisé et au plus tard dans les 13 mois en cas de prélèvement non autorisé. Vos droits concernant le présent mandat sont expliqués dans un document que vous pouvez obtenir auprès de votre banque. Les informations demandées sont enregistrées dans un fichier informatisé par Handicap International qui dispose d'un délégué à la protection des données (dpo). Elles sont nécessaires pour répondre à vos demandes ou faire appel à votre générosité. Elles sont conservées pendant la durée strictement nécessaire à la réalisation des finalités précitées. Conformément à la loi Informatique et Libertés et à la réglementation européenne, en vous adressant à donateurs@france.hi.org, vous bénéficiez d'un droit d'accès, rectification, limitation, portabilité, effacement et opposition à l'utilisation de vos données à caractère personnel. En cas de difficulté, vous pouvez introduire une réclamation auprès de la CNIL.

0

EUROPEAN CYBER CUP

LA 1^{ÈRE} COMPÉTITION DE ESPORT DÉDIÉE AU HACKING ÉTHIQUE

LE 9 & 10
JUN 2021
À LILLE
GRAND PALAIS

TO BE [PWNED]
OR NOT TO BE [PWNED]
THAT IS THE {CYBER}QUESTION

WWW.EUROPEAN-CYBERCUP.COM

Organisé par :



En partenariat avec :



Avec le soutien de :

