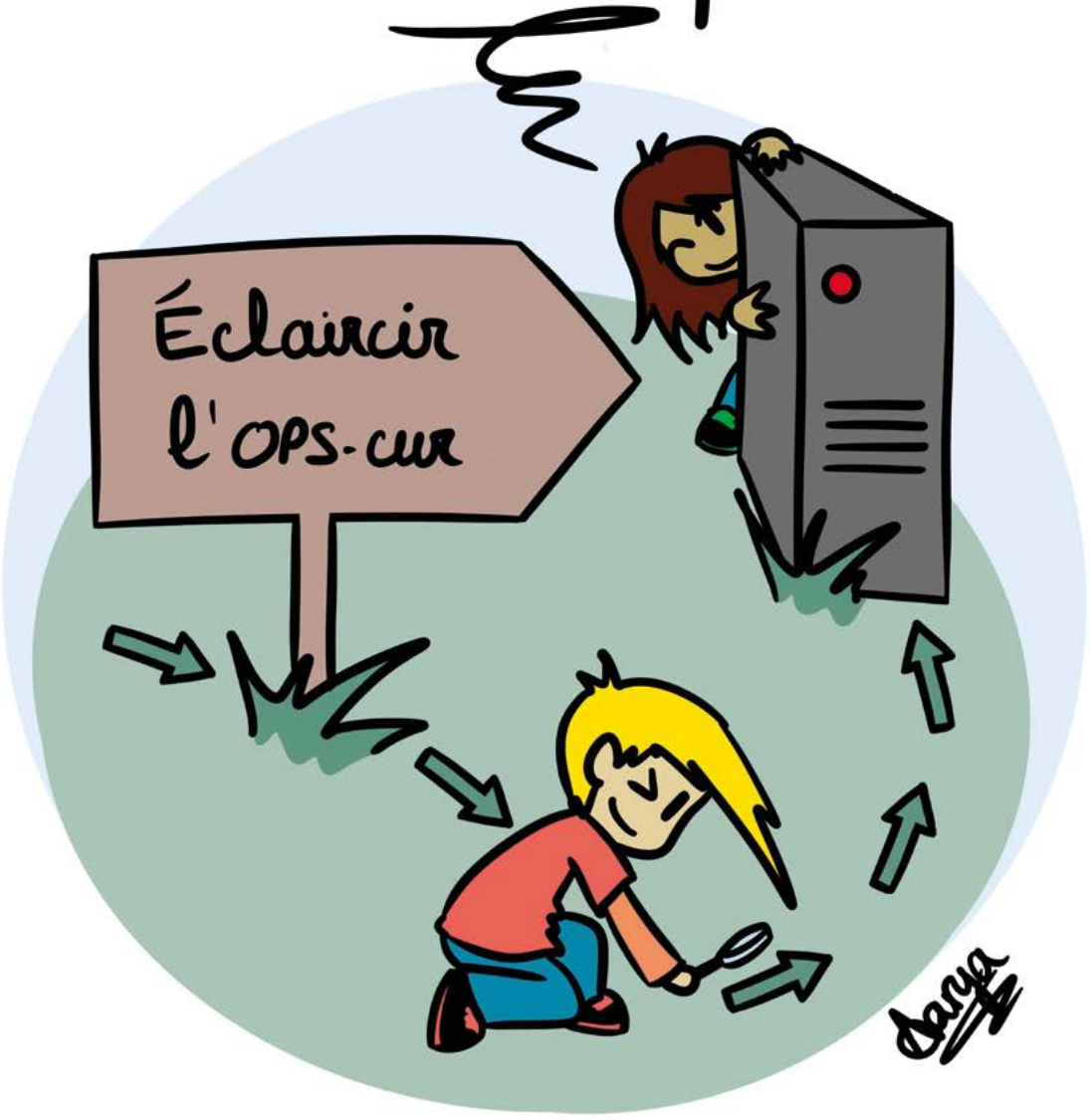
OCTO PRESENTS

A la découverte du DevOps !



Projecteur sur Aryana Pezé

THERE IS A BETTER WAY

Derrière OCTO, il y a des centaines d'Octos, et autant de talents cachés…

À travers cette bande dessinée, la collection OCTO Presents tourne ses projecteurs sur la talentueuse Octo Aryana Pezé.

Après ses débuts dans le recrutement, Aryana trouve sa voie : devenir développeuse !

Elle intègre ainsi OCTO en tant que dev, puis rejoint la Tribu Ops.

Passionnée par le dessin, elle décide de mettre en scène le DevOps à travers ses BD.

À la découverte du DevOps

Partie 1

Partie 2

Pour aller plus loin 26 - 43

À la découverte du DevOps !

Introduction

"Et toi? Tu fais quoi dans la vie?"
J'ai beau adorer mon métier, j'en suis arrivée à redouter la question.
Comment expliquer le métier d'Ops en quelques mots à des personnes qui n'ont jamais fait d'informatique, alors que je peine déjà à l'expliquer à d'autres personnes du secteur?
Et quand je demandais à d'autres Ops...

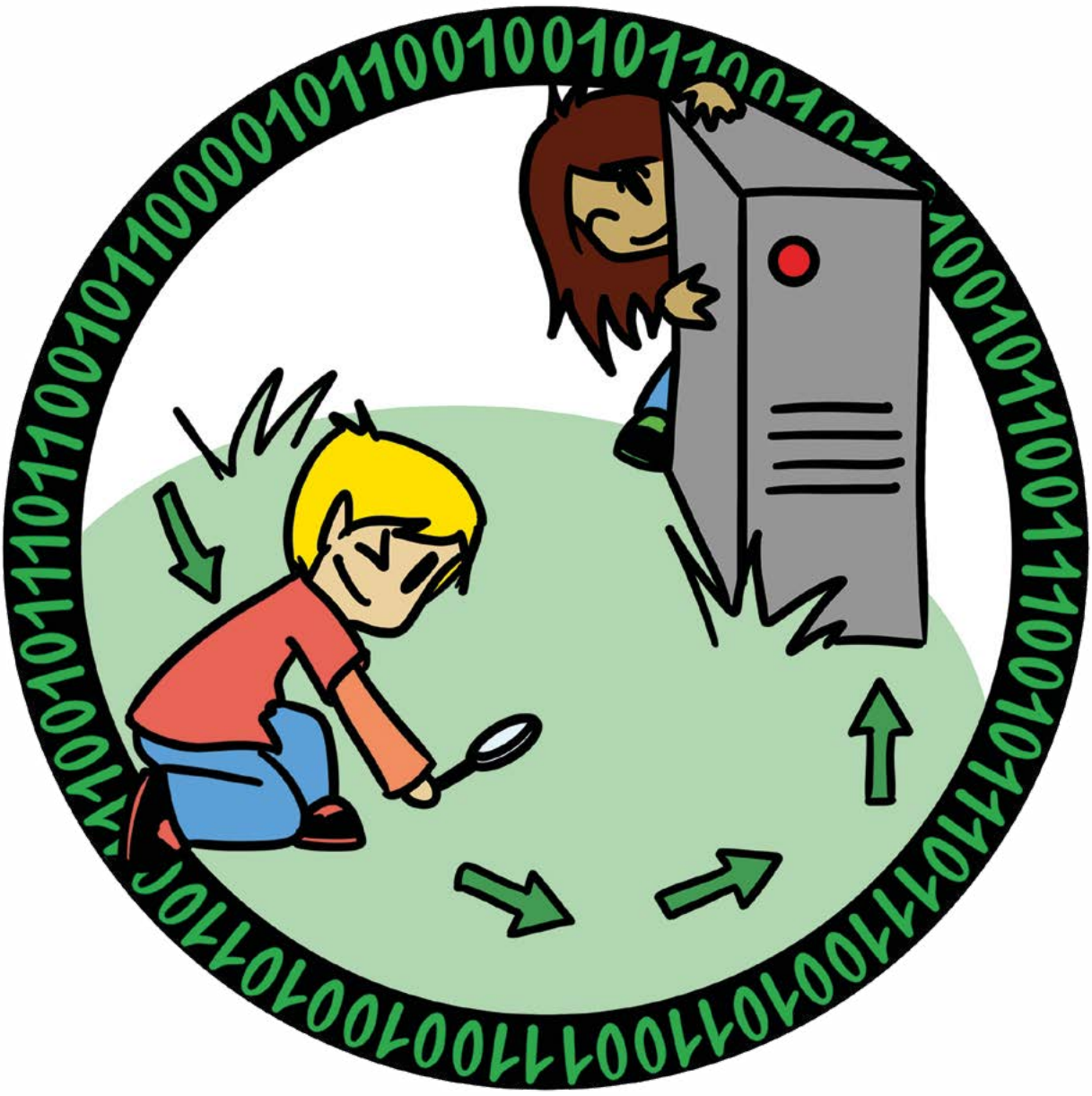
Ou bien encore:

... Un peu frustrant pour quelqu'un qui souhaite débuter dans le métier...!

Et voici comment cette BD a vu le jour!
L'objectif? Vulgariser certains sujets clefs de l'informatique. Le monde de l'informatique étant vaste, il ne s'agit que d'un premier aperçu de ces sujets.
A vous d'approfondir ceux qui vous intéressent!

Partie 1

Je me lance dans le DevOps 6-25

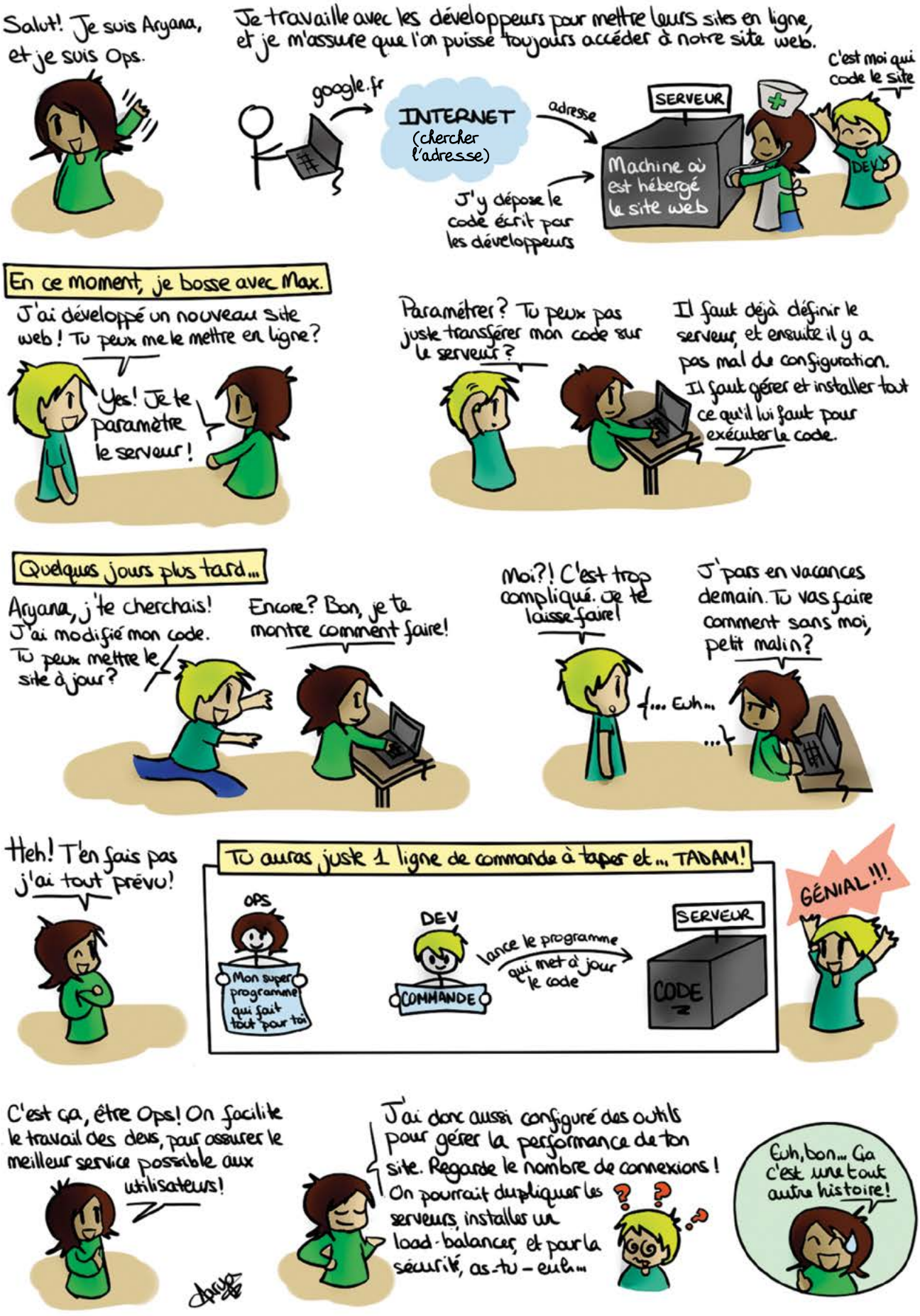


Lorsque j'ai décidé de faire de l'Ops, je ne savais pas vraiment par où commencer. J'ai dû apprendre sur le tas, et heureusement, j'étais bien entourée!

Cette première partie vise à mettre en lumière les différents sujets que nous sommes amenés à aborder en tant qu'Ops. Chaque sujet est abordé différemment selon le contexte, mais cela donne une première idée de ce que peut être notre métier.

A vous ensuite d'aller creuser les sujets qui vous intéressent!

Le DevOps ? C'est quoi ?

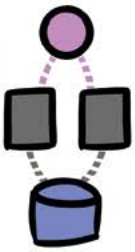


Quelques concepts clefs - 1

SERVEUR

Matériel informatique qui fournit des informations ou logiciels à d'autres ordinateurs reliés au même réseau (filaire, Intranet, Internet...).

INFRASTRUCTURE INFORMATIQUE



Ensemble des équipements et logiciels qui, connectés entre eux, vont former le squelette d'un système informatique.
On parle aussi d'architecture informatique.

OS

Système d'Exploitation : logiciel dans un appareil électronique qui pilote l'utilisation des ressources d'un outil informatique (mémoire, processeur, périphériques...).

TERMINAL

Point d'accès de communication entre l'homme et la machine. Il a souvent l'aspect d'un périphérique de saisie.

Quelques concepts clefs - 2

HÉBERGEUR

Un hébergeur met à disposition l'infrastructure nécessaire pour déployer un site web conçu par un tiers.

ENVIRONNEMENT DE DÉVELOPPEMENT

Ensemble des outils que les développeurs installent sur leur machine pour travailler (éditeur de texte, librairies...).

ENVIRONNEMENT DE PRODUCTION

Environnement final sur lequel fonctionne la version finale du site, disponible aux utilisateurs.

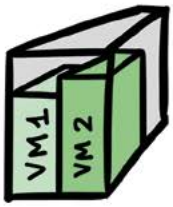
IMAGE

"Photo" de l'état souhaité du serveur de production: code et dépendances.

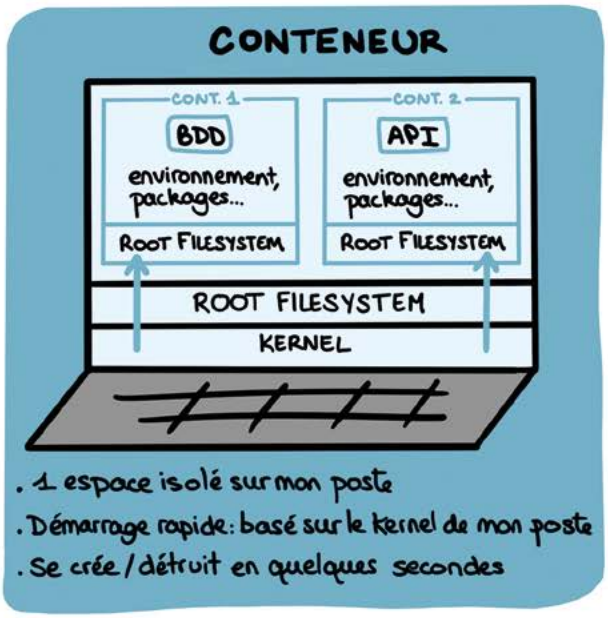
CONTENEUR

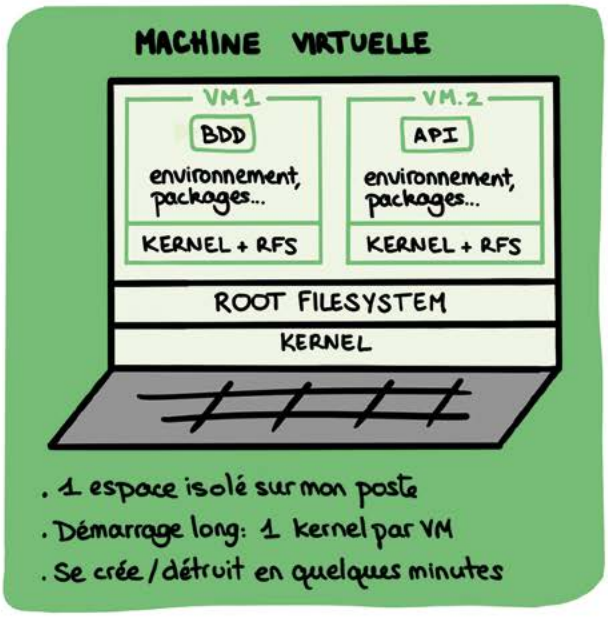
"Boîte" générée à partir d'une image. C'est un environnement isolé sur lequel tourne le code contenu dans l'image.

VM

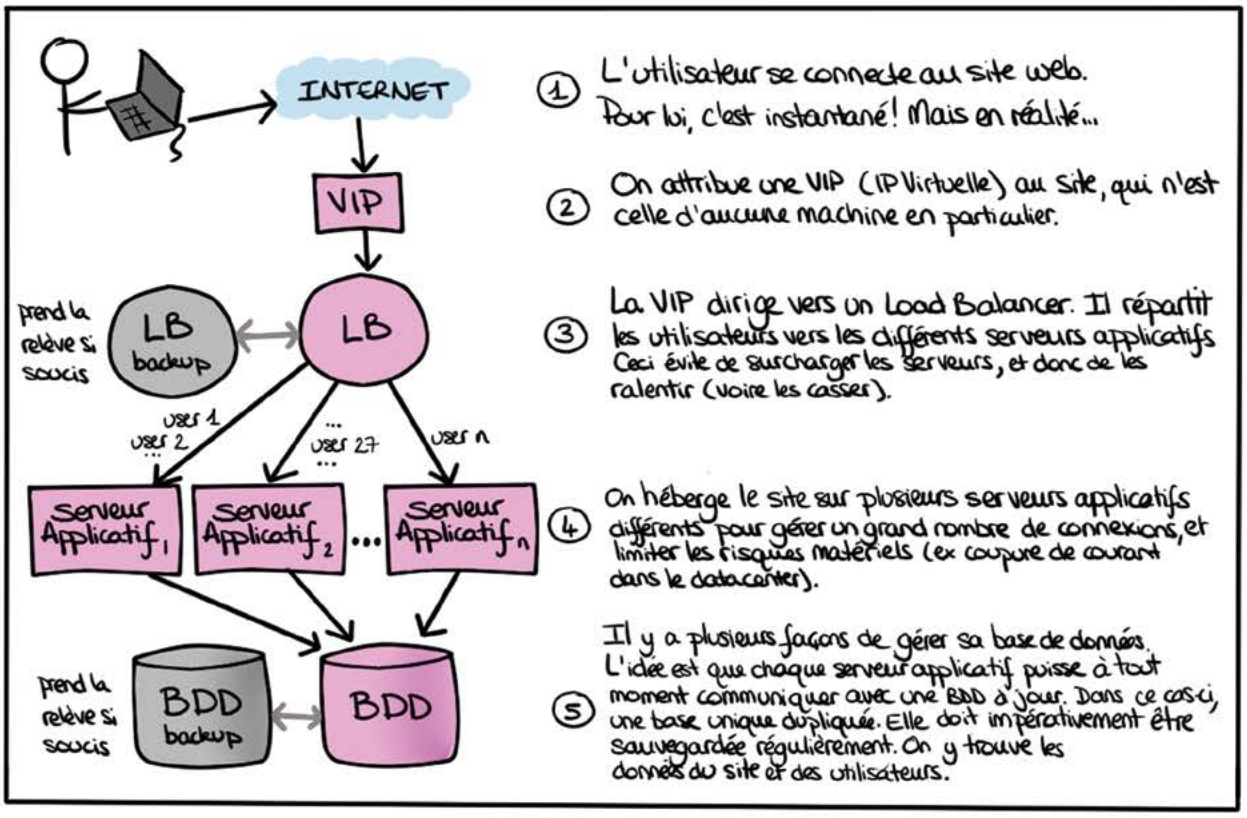


Machine Virtuelle, qui simule un serveur. On peut donc avoir plusieurs VMs isolées les unes des autres sur un même serveur.





La logique derrière mon infra



Les modèles Cloud

PAAS: Platform As A Service — l'ami des petites applis! €€€

L'hébergeur (Heroku, OVH, Scalingo...) met des infrastructures à disposition. Ses Ops gèrent tout sauf ton serveur applicatif.

Tu choisis l'action qui te convient... et en quelques clics, tout est créé!

Tu déploies ton code quand tu veux – même pas besoin d'un Ops!

IAAS: Infrastructure As A Service — mon infra sur mesure! €

L'hébergeur (AWS, Azure, GCP) met des outils (serveurs, BDD) à disposition - que ses Ops vont gérer.

On discute pour définir tes besoins..

J'automatise la création de l'infra, et le paramétrage des machines...

Je pousse mon code, et ton infra est prête!

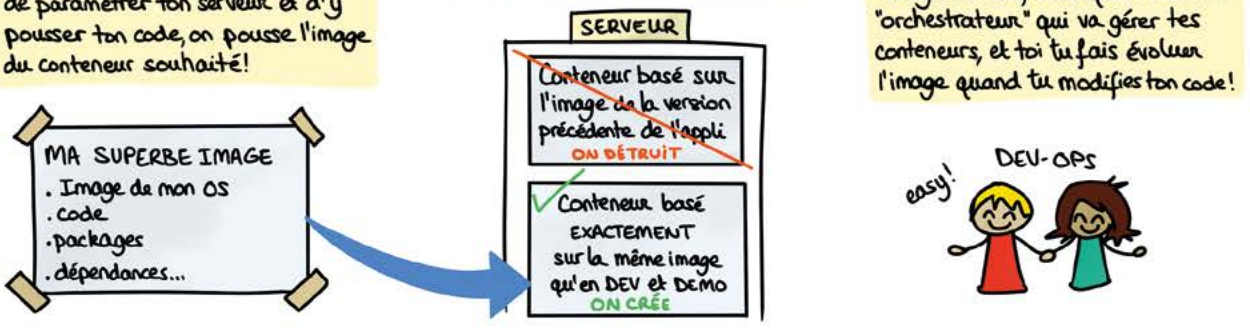
Et on la maintient ensemble!

CAAS: Containers As A Service — le service du futur? €

Entre le PAAS et le IAAS (hébergé par AWS, Azure, GCP...). Mais au lieu de paramétrer ton serveur et d'y pousser ton code, on pousse l'image du conteneur souhaité!

On a ainsi exactement la même image sur tous nos environnements!

Je t'aide à configurer ton image initiale, ainsi que l'éventuel "orchestrateur" qui va gérer tes conteneurs, et toi tu fais évoluer l'image quand tu modifies ton code!



Donc tout dépend de la complexité de mon infra? Oui, et du budget.

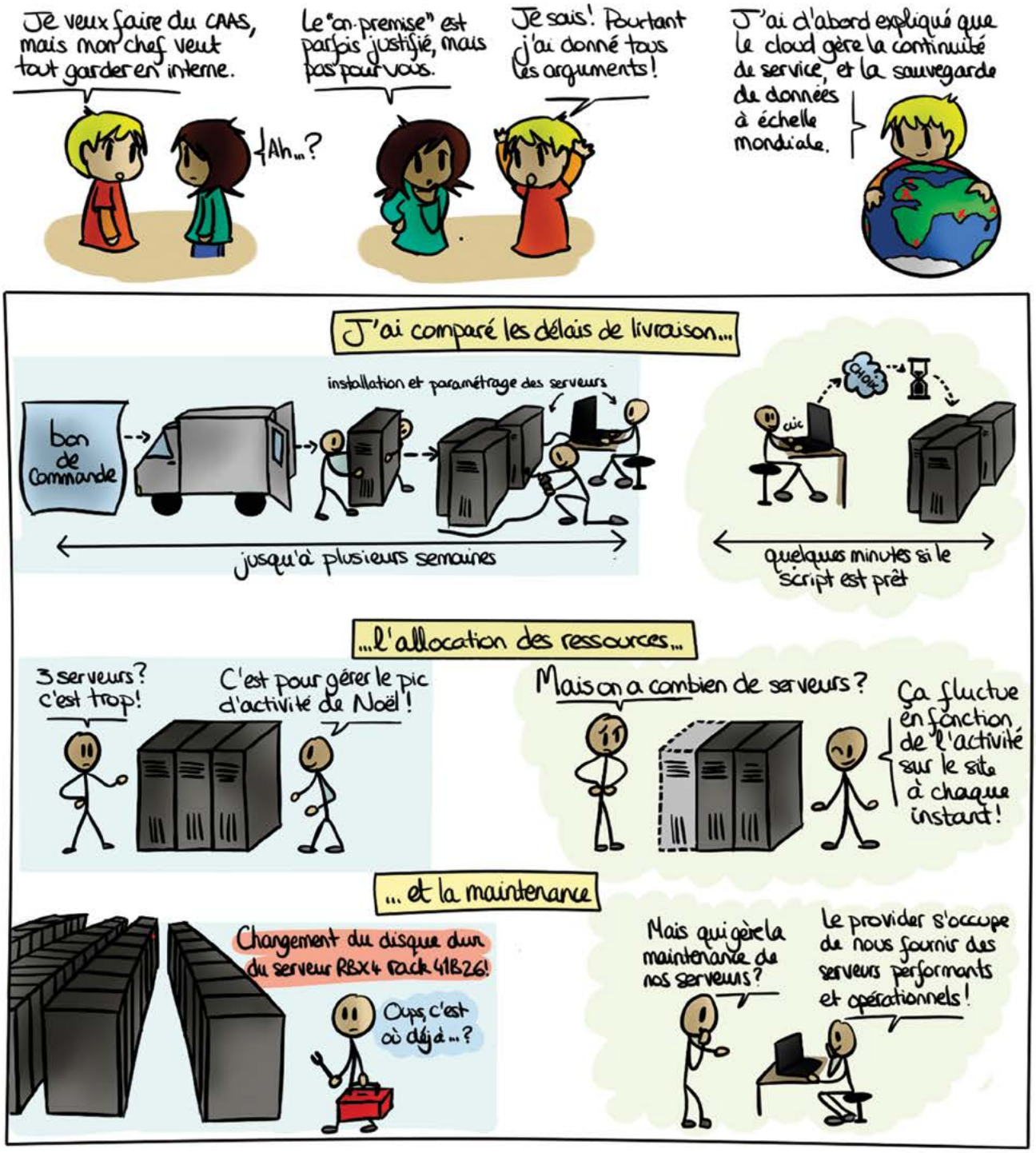
Même si on paye chaque service en fonction de notre utilisation, certains sont plus onéreux que d'autres.

Ok pour le choix IAAS-PAAS. Mais pour le CAAS...?

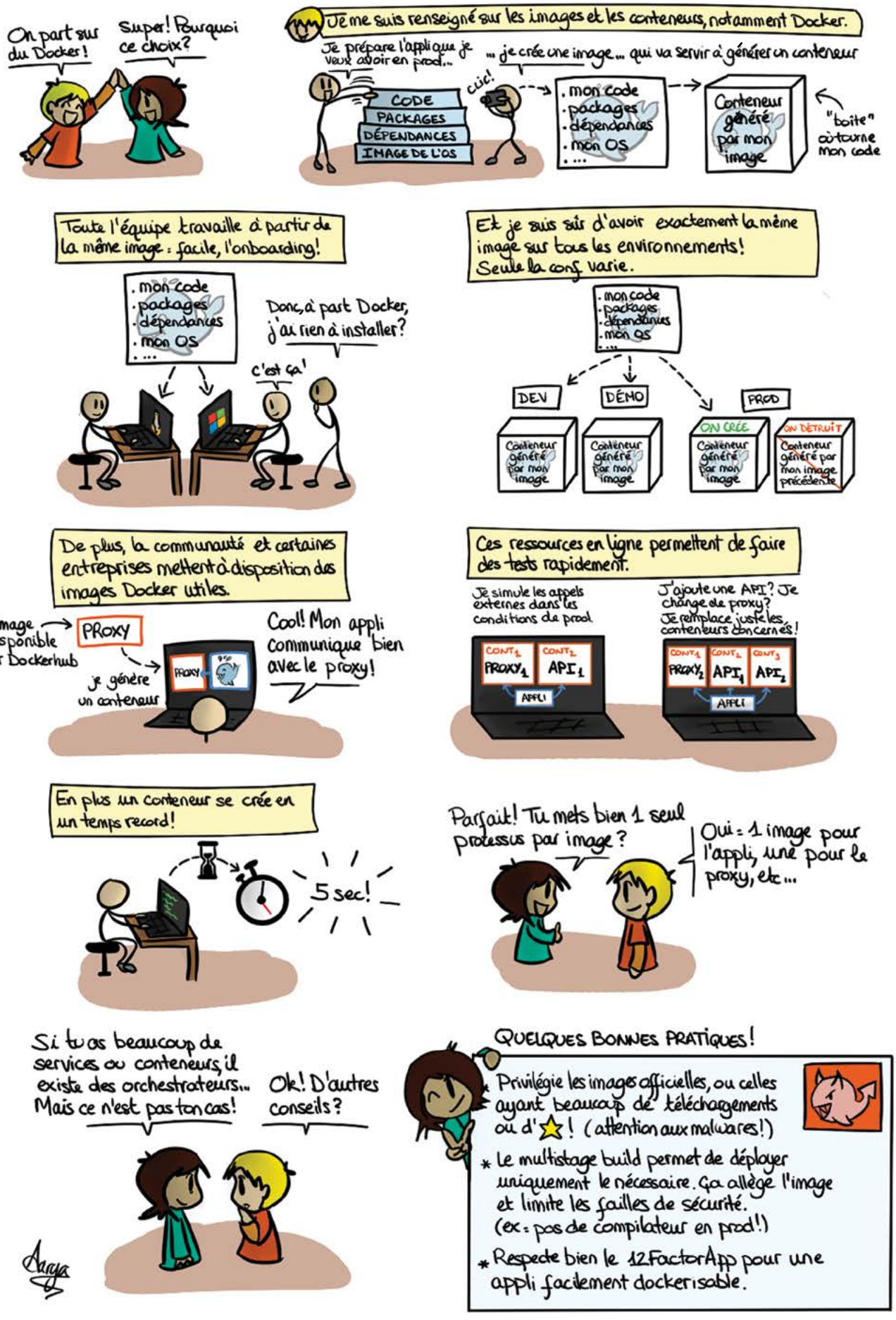
Beaucoup d'Ops le conseillent. C'est simple et efficace, du moment que ton appli est 'stateless'. Sa popularité ne cesse de croître! Affaire à suivre!

Aarya

Cloud vs On-Premise



Introduction à Docker



Introduction au réseau : les adresses IP - 1

yo! J'aimerais avoir une petite intro sur les IP et le réseau.

Ah, top! Commençons par les adresses IP.

Tout matériel connecté à un réseau a une carte réseau et est identifié par une adresse IP.

L'IPV4 (Internet Protocol Version 4) est la plus déployée.

format = XXX.XXX.XXX.XXX

1 octet

4 nombres de 0 à 255

Pour ce qui est du réseau, tu connais déjà le plus grand...!

Les fournisseurs d'accès à Internet allouent des adresses IP à chaque appareil connecté, via le protocole DHCP.

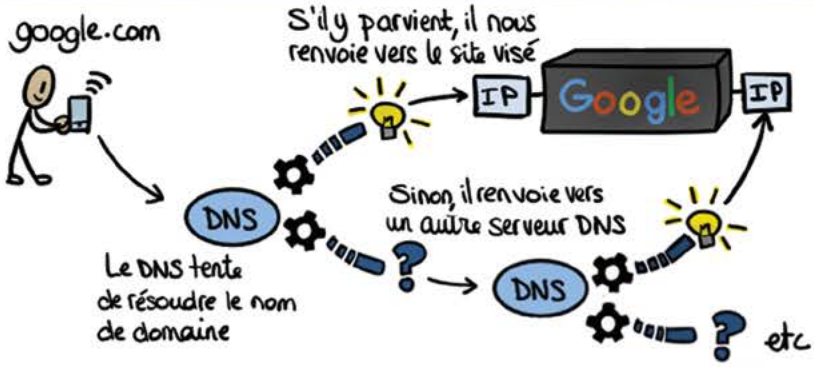
Si c'est une "adresse", ça signifie que je peux la contacter?

C'est ça! Pareil qu'une adresse postale!

Pour les sites web, pour plus de simplicité, on associe un nom de domaine à une adresse IP. Le service DNS (Domain Name System) est chargé de la traduction.

Mais c'est difficile à retenir...

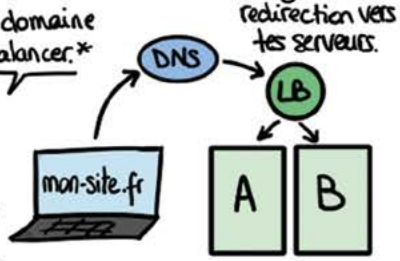
Oui, d'où l'utilisation de noms de domaine!



S'il ne parvient pas à le résoudre (nom de domaine erroné, serveur indisponible...) tu auras un message d'erreur.

Et pour mon site? J'ai 2 serveurs en parallèle!

Aucun souci! Il suffit d'associer le nom de domaine au load-balancer.*

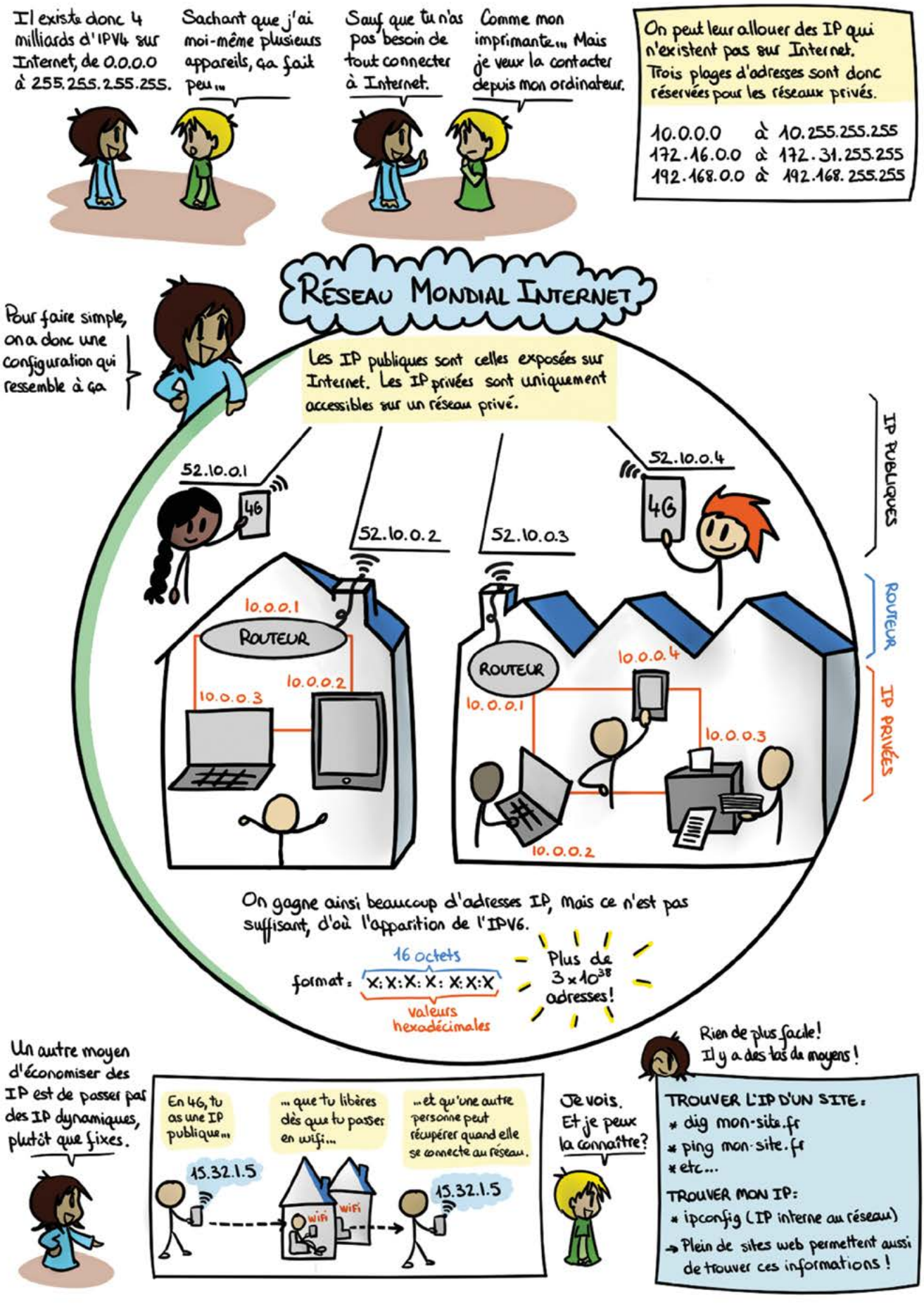


*Composant informatique, ici utilisé pour répartir le trafic entre les 2 serveurs.

Ah! Mais si on connait l'IP de mes serveurs, on peut contourner mon load-balancer?

On pourrait, si tes serveurs avaient une IP publique - mais je leur ai donné des IP privées! Laisse-moi t'expliquer...

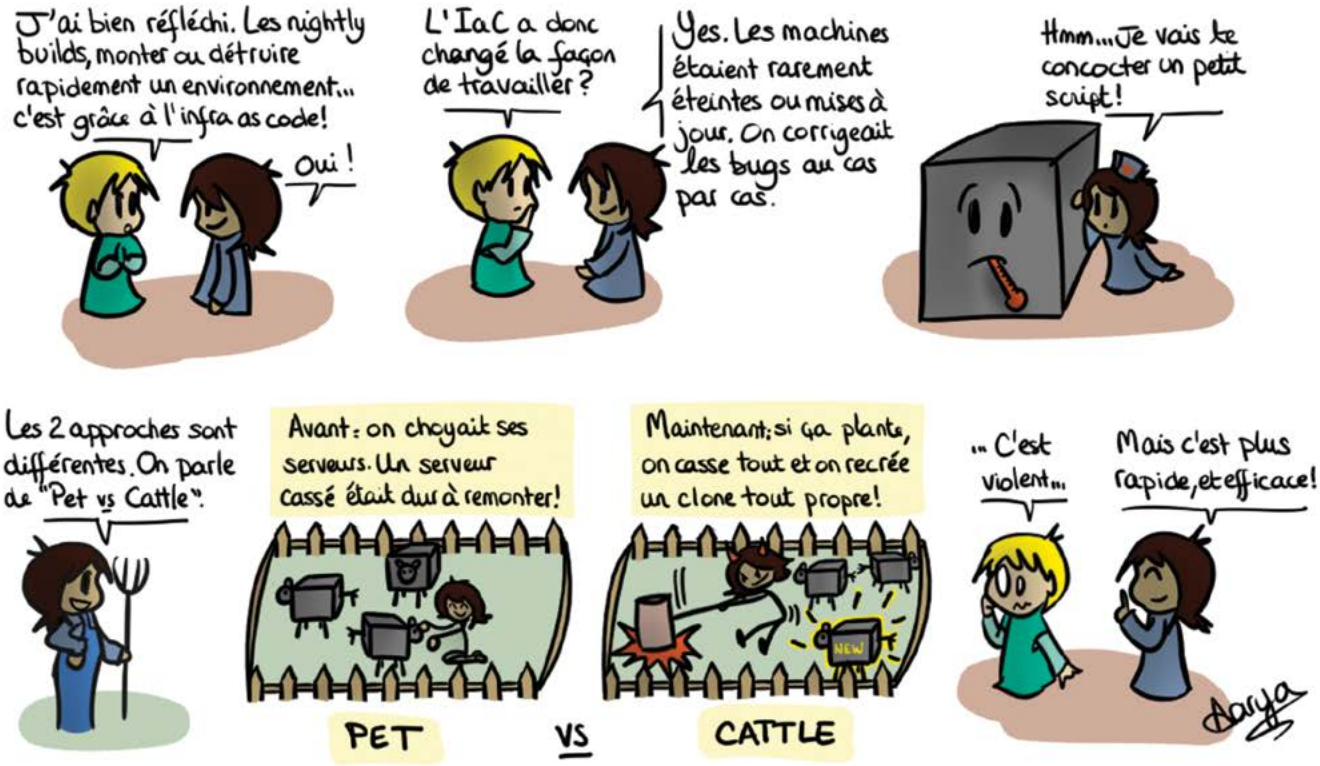
Introduction au réseau : les adresses IP-2



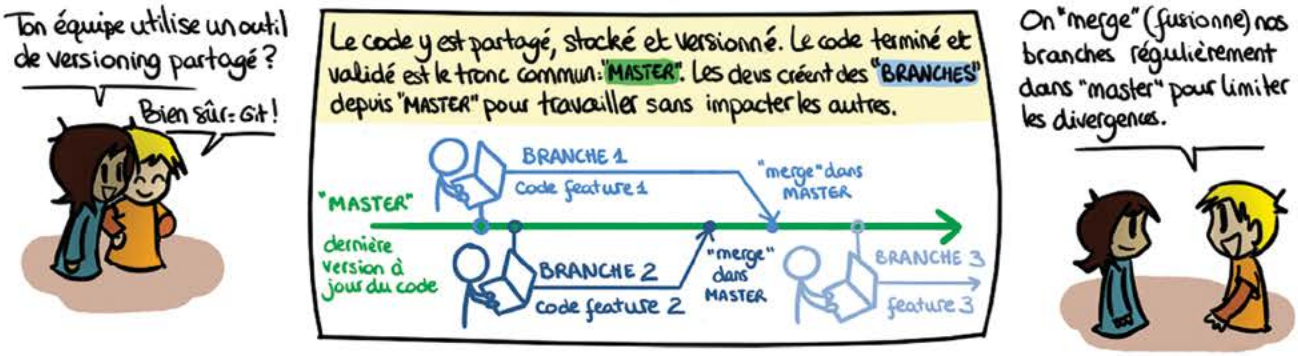
L'Infrastructure as Code (IaC)

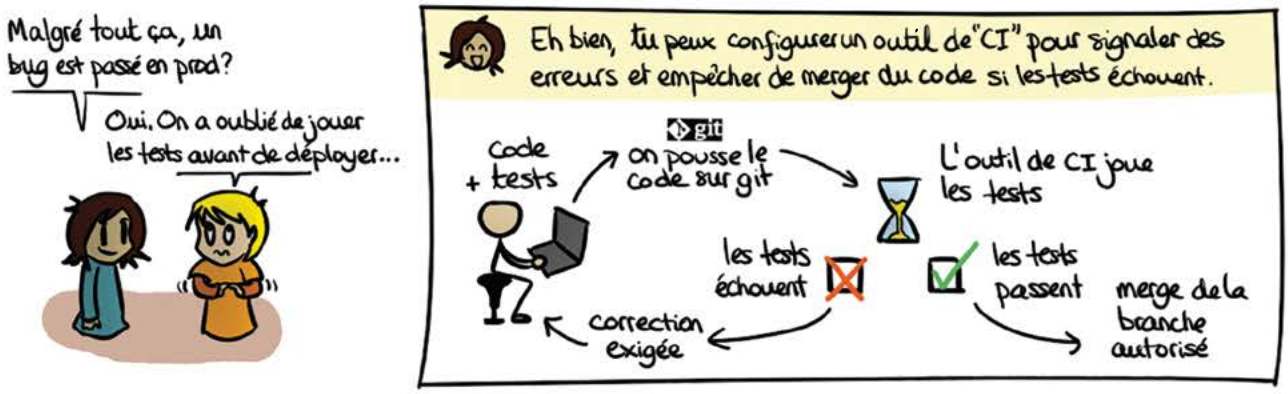
Pet vs Cattle

L'Infrastructure as Code (IAC) nous permet de facilement et rapidement détruire et remonter les machines. Cela a donc grandement impacté la façon de travailler des Ops! Plutôt que de travailler sur des serveurs spécifiques qu'on maintient et qu'on bichonne des mois durant, on préfère détruire et remonter les machines régulièrement.
On parle de « Pet vs Cattle ».

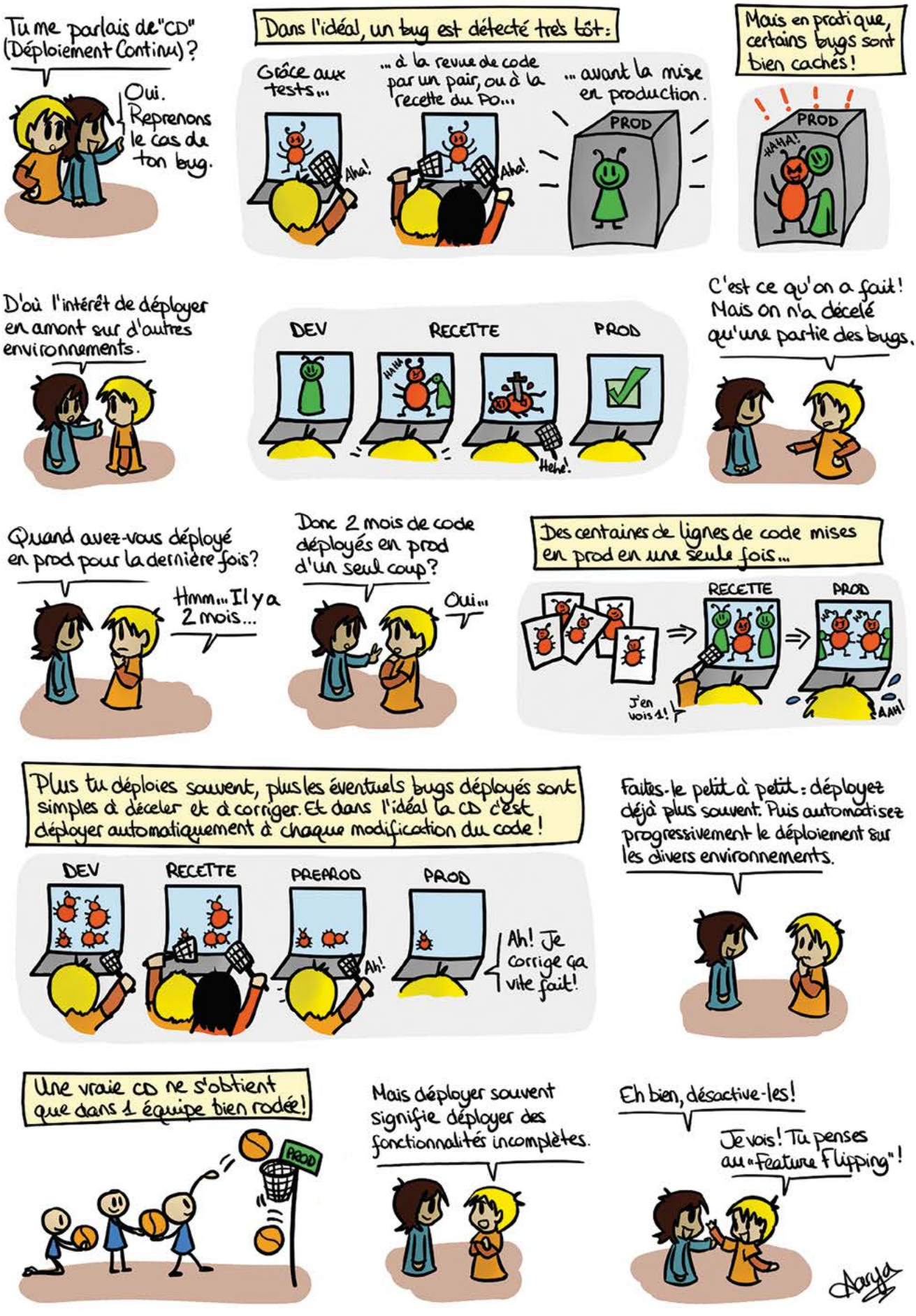


L'Intégration Continue (CI)

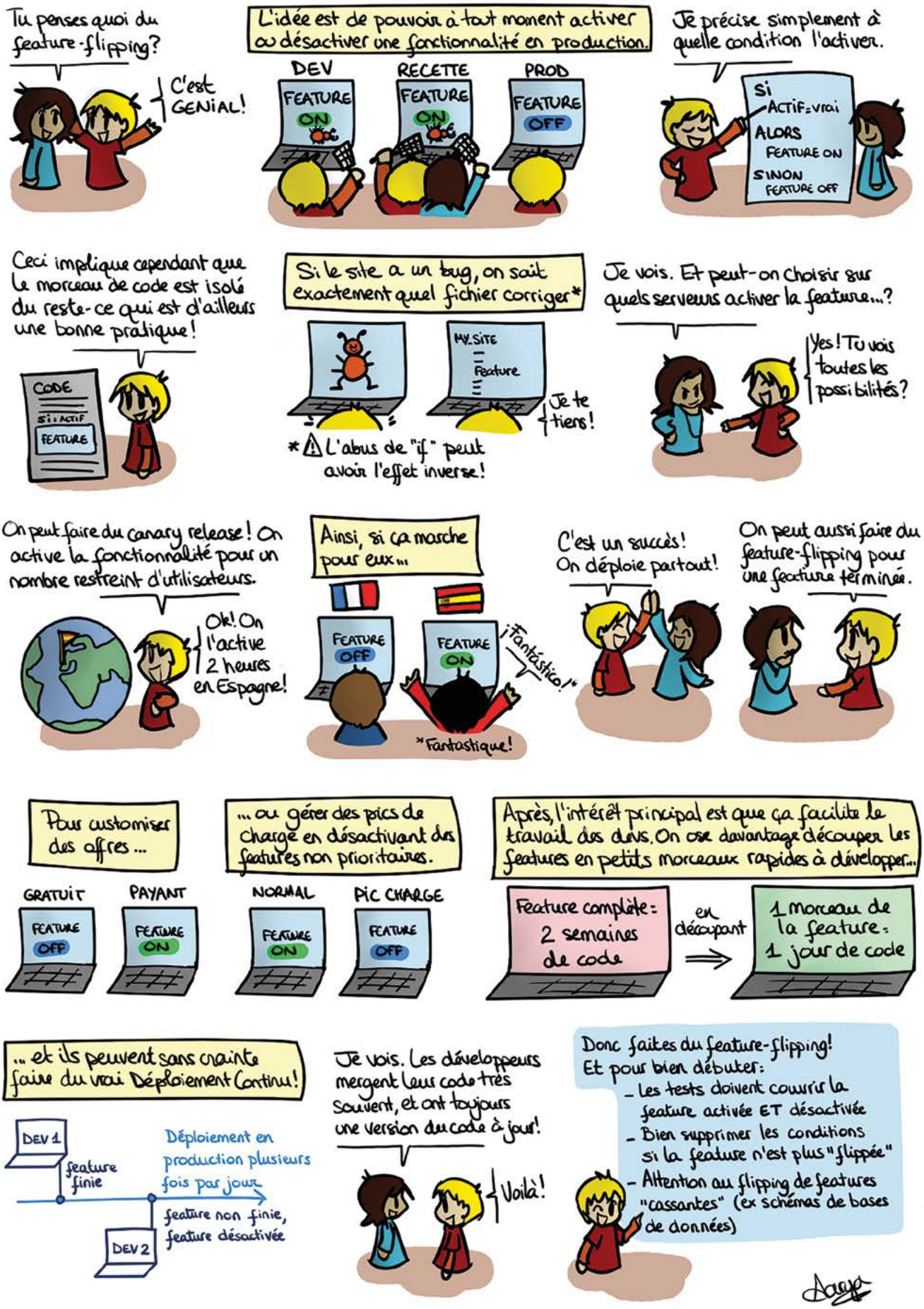




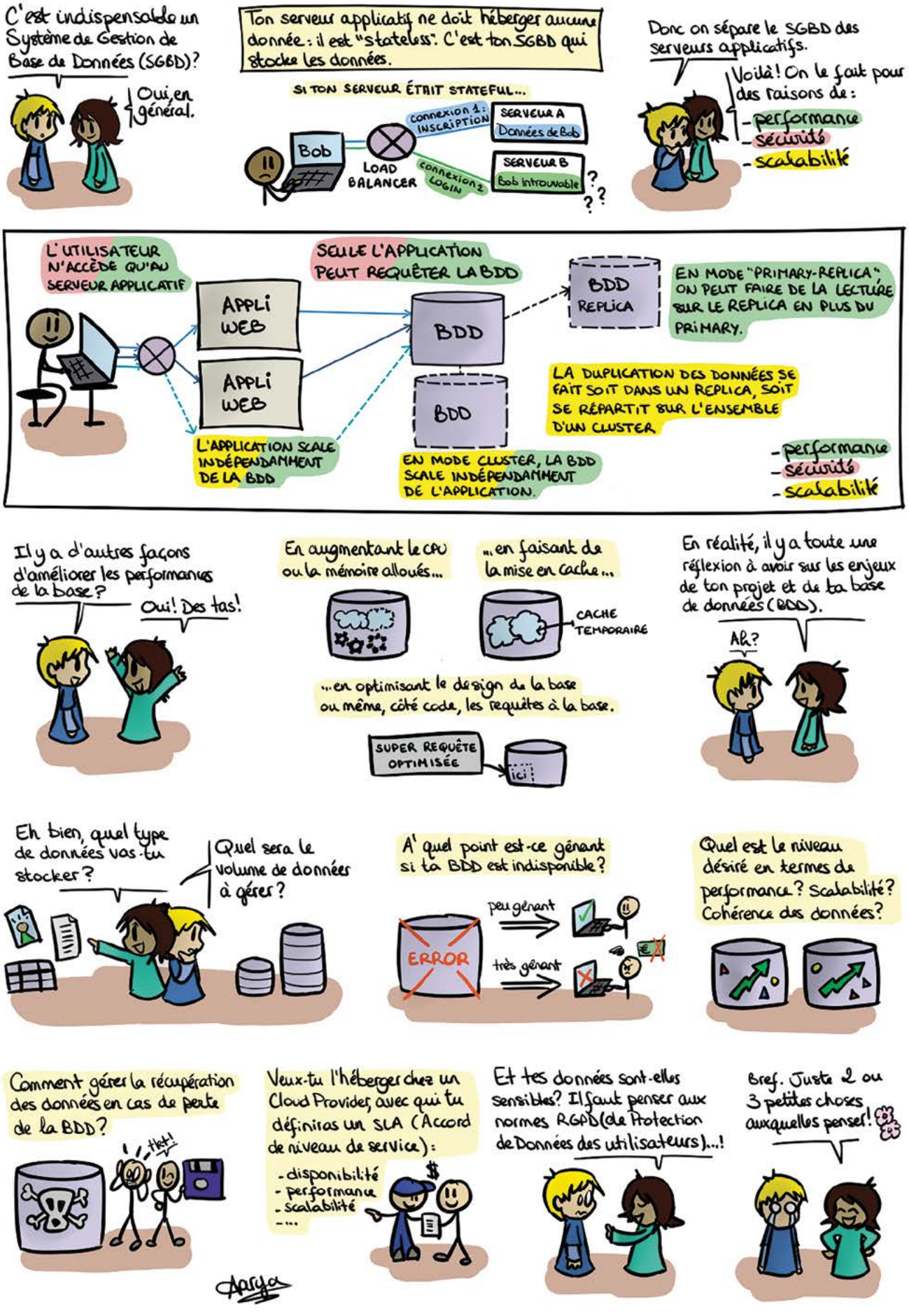
Le Déploiement Continu



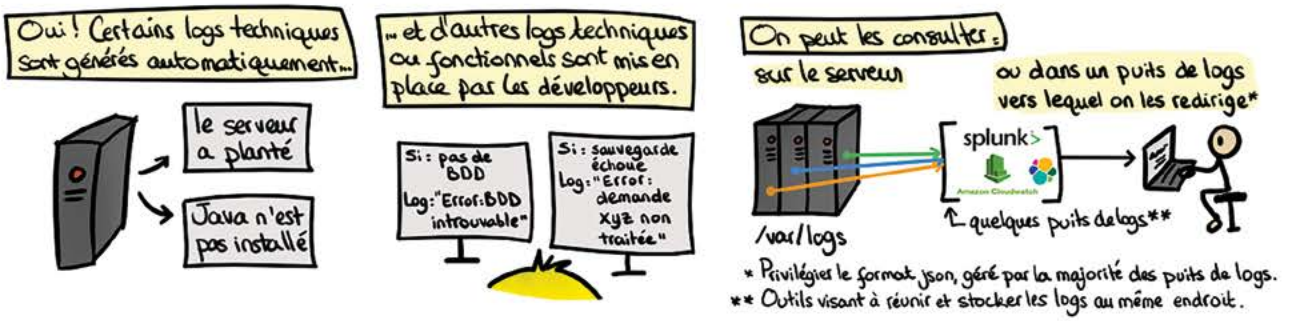
Le Feature Flipping

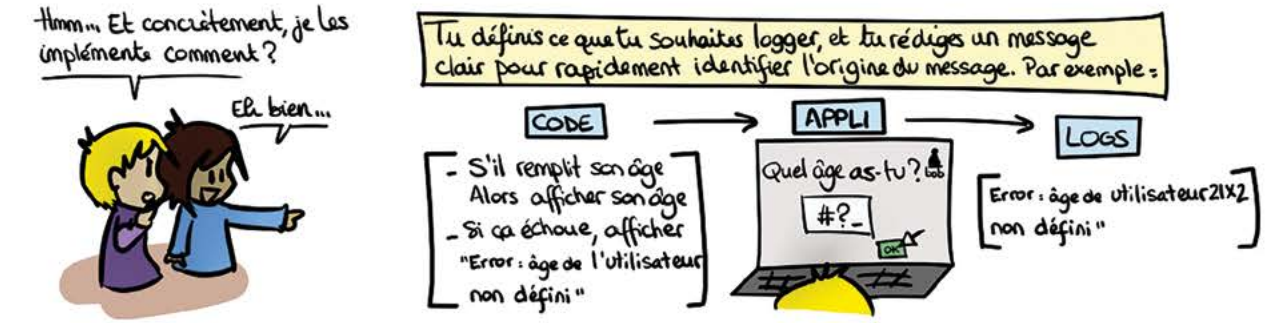


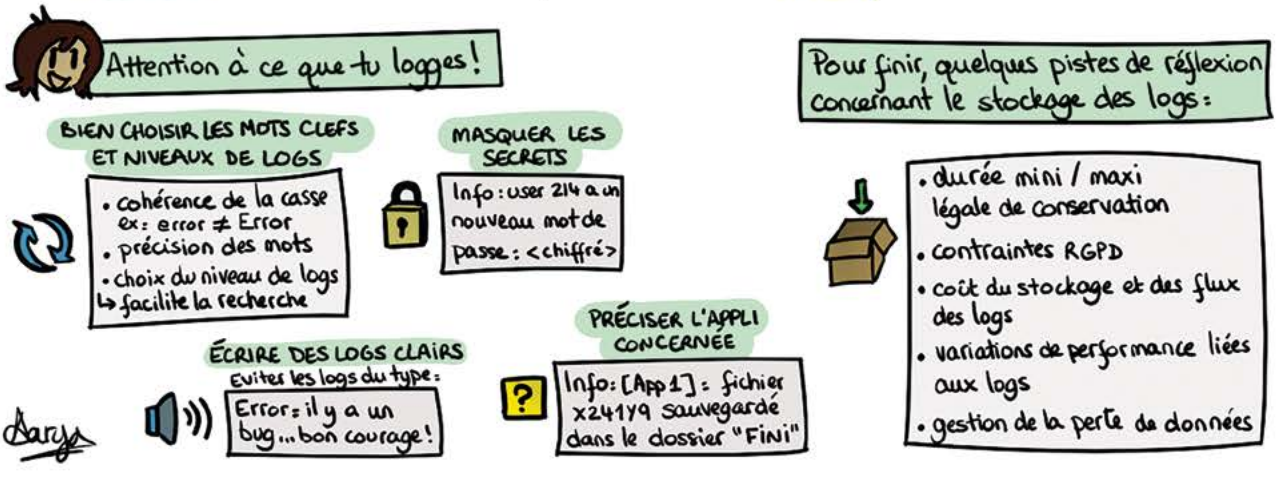
Les Bases de Données



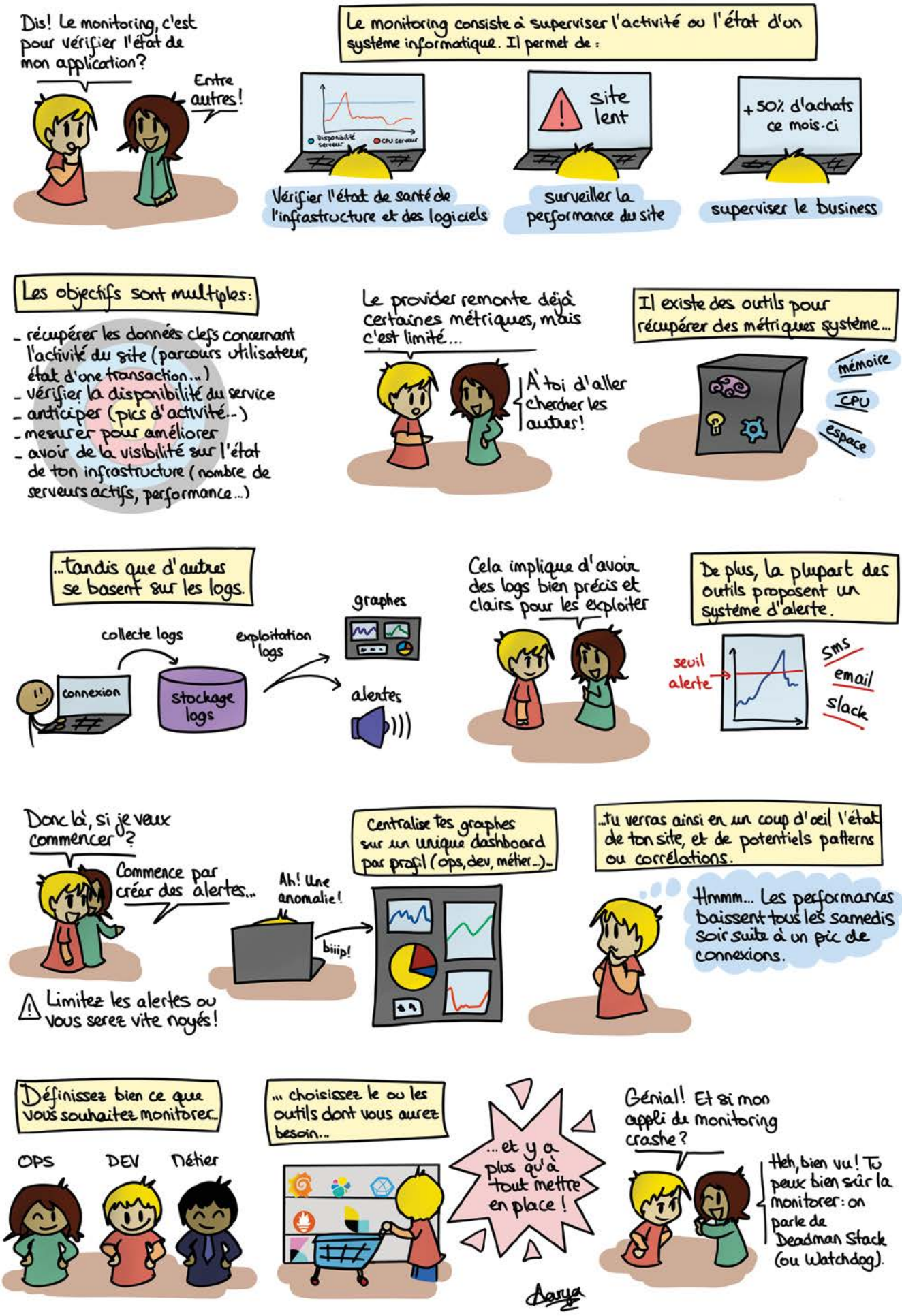
Les Logs





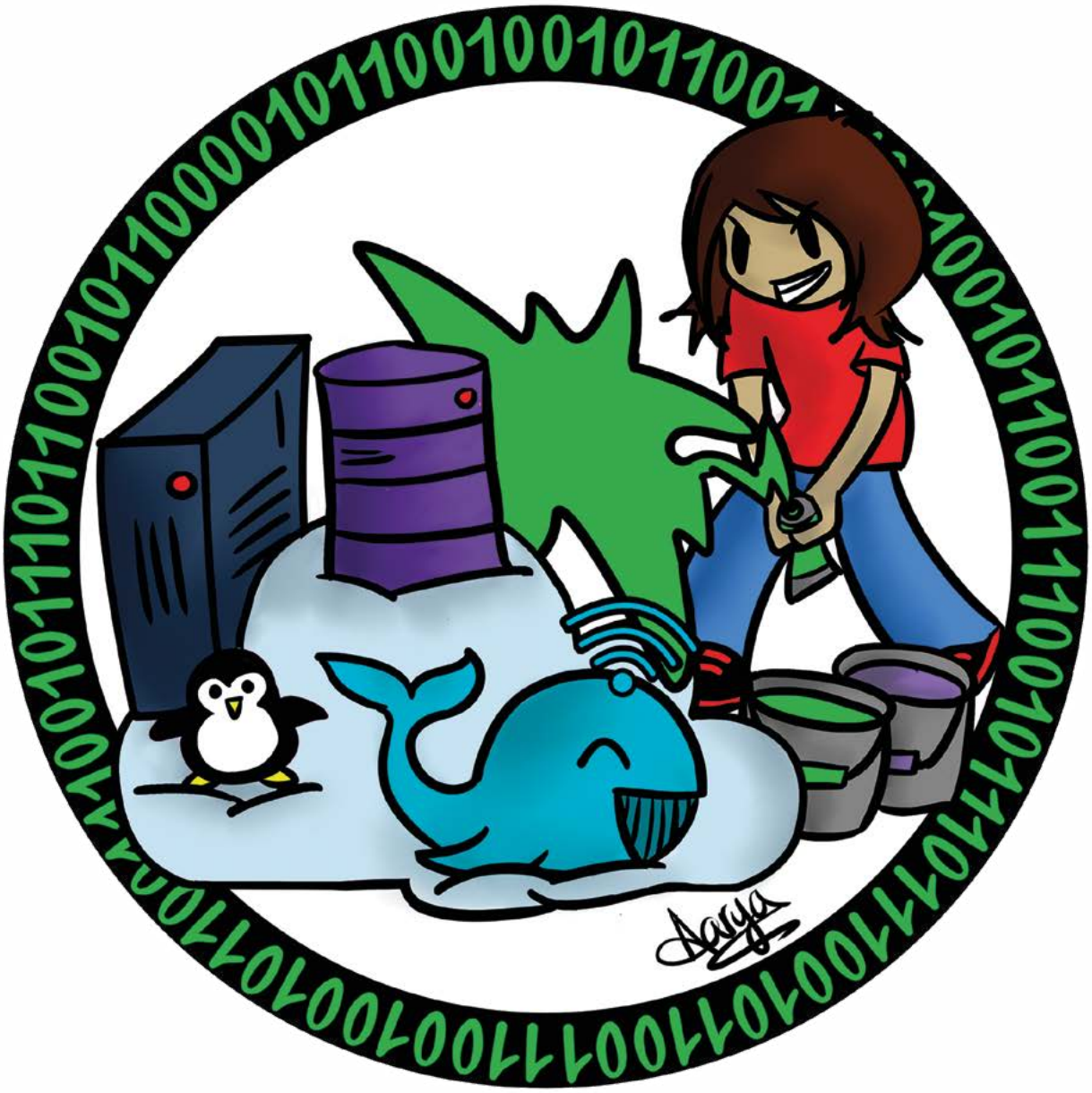


Le Monitoring



Partie 2

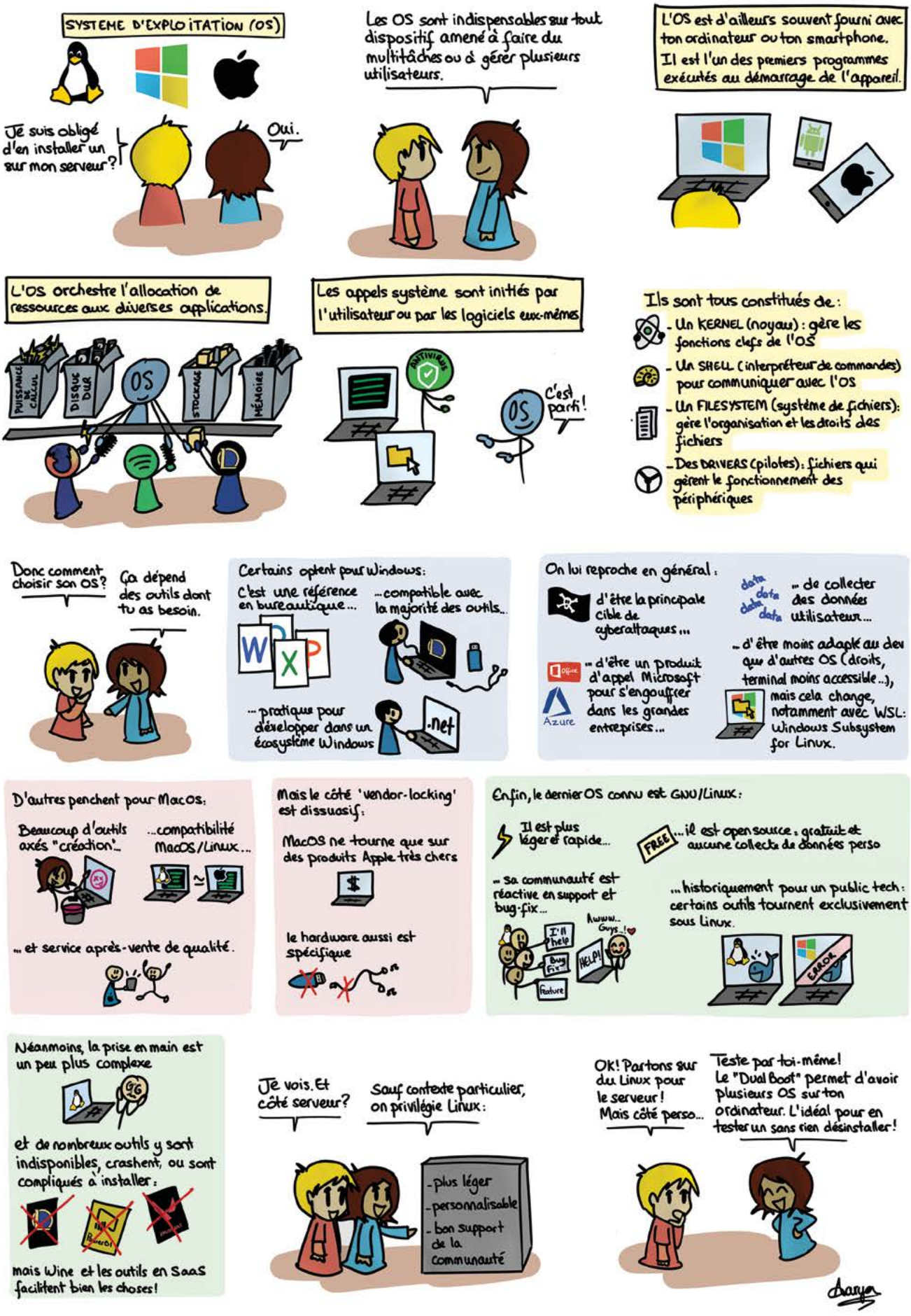
Pour aller plus loin 26 - 43



L'informatique fourmille d'outils et de termes auxquels nous sommes régulièrement confrontés, aussi bien en tant que professionnels qu'en tant que particuliers, sans pour autant que l'on sache exactement comment ils fonctionnent.

Le moment est enfin venu de creuser ces concepts !

L'OS



Le Terminal

Le Terminal (aussi dit "Console" ou "Interface de ligne de commande") permet de donner des instructions à l'ordinateur.

Comme ce n'était pas simple, les Interfaces Utilisateur sont apparues fin 1970's pour rendre les ordinateurs plus accessibles...

...Donc pourquoi t'obstiner à tout faire en ligne de commande... ?!

C'est plus pratique !

...ben voyons...

Hmm... Quelle est la commande...?

Fichtre ! J'comprends pas !

C'est parti mon kiki !

Déjà, je n'ai souvent pas le choix lorsque je travaille sur un serveur distant:

SERVEUR APPLICATIF

Il y a rarement une interface, je dois écrire mes commandes ! *

* ou les automatiser, c'est encore mieux !

J'envoie directement mes commandes au système d'exploitation (OS) via un interpréteur de commandes.

Ouvre le fichier

INTERPRÉTEUR DE COMMANDES

- sh
- bash
- zsh
- ...

OS

- Linux
- MacOS
- Windows
- ...

Contrairement à l'interface, les commandes varient très peu d'un OS ou d'un interpréteur à l'autre.

UBUNTU
sh
. cd
. grep
. apt install

MacOS
sh
. cd
. grep
. brew install

Cela facilite donc le partage d'informations : c'est plus pratique de recopier une commande que d'expliquer où cliquer.

Ouvre le dossier. C'est bon ?

Yes ! Après ?

Donc si tu cherches une commande, Internet est ton ami ! Quelqu'un se sera toujours posé la question avant toi..!

De rien.

commande renommer ses photos

Et enfin, en cas d'erreur, j'ai un message d'erreur détaillé.

$ emoc :

Parfois même un peu trop..

Après, chacun son terminal et chacun ses habitudes : shell, couleurs, raccourcis...

"...jusqu'à obtenir ton terminal rêvé !

Elle est folle...

Et toi ? Tu utilises quelles commandes ?

Hehe ! Il existe même une commande pour répondre à cette question ! Tadaam !

GESTION DE FICHIERS

- . cd : changer de dossier
- . chmod : changer les droits d'accès à un fichier ou dossier
- . cp : copier un fichier ou dossier
- . grep : rechercher une expression dans un fichier ou dossier
- . gzip : compresser un fichier en .gz
- . ls : lister les fichiers ou dossiers présents
- . open : ouvrir un fichier
- . rm : supprimer un fichier ou dossier

RÉSEAU

- . dig : obtenir l'IP d'un domaine
- . ping : vérifier la disponibilité d'un serveur
- . ssh : se connecter à un serveur distant
- . tcpdump : écouter le trafic sur une interface réseau
- . wget : télécharger un fichier

GESTION DES PROCESSUS

- . kill : envoyer un signal à un processus, souvent pour l'arrêter.
- . ps : afficher les processus actifs

DONNÉES SYSTÈME

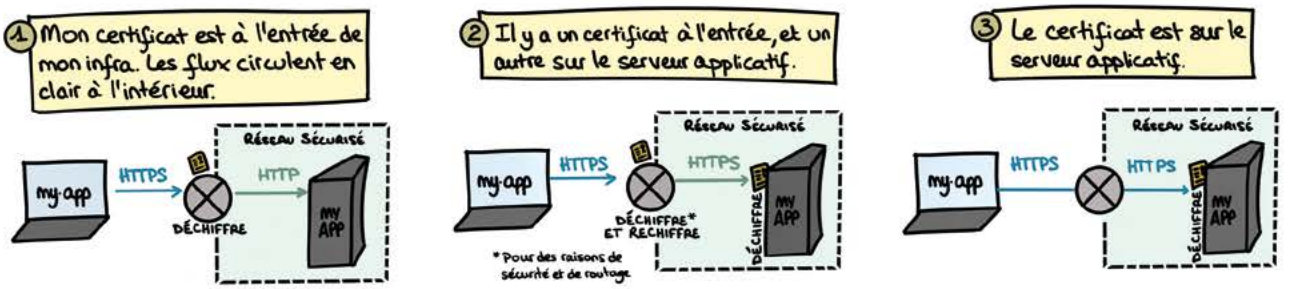
- . date : afficher la date et l'heure
- . du : afficher l'espace dossier utilisé
- . which : localiser une commande

COMMANDES OUTIL

- . { brew / apt / yum / ... } install <tool>
- . <tool> : lancer une commande liée à l'outil

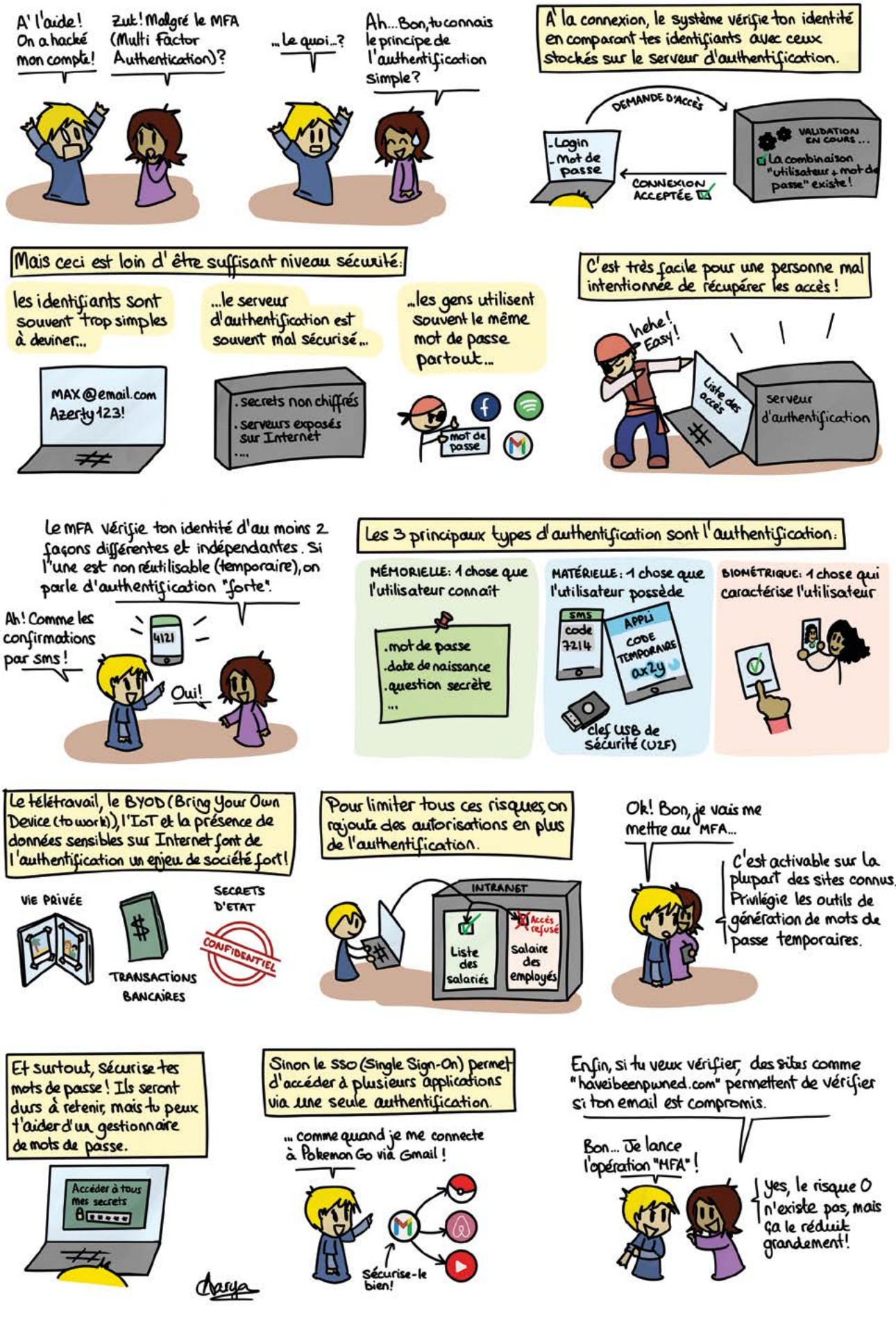
Et pour plus d'infos sur une commande, tapez "man" ou "tldr" + <commande> dans le terminal !

Le HTTPS

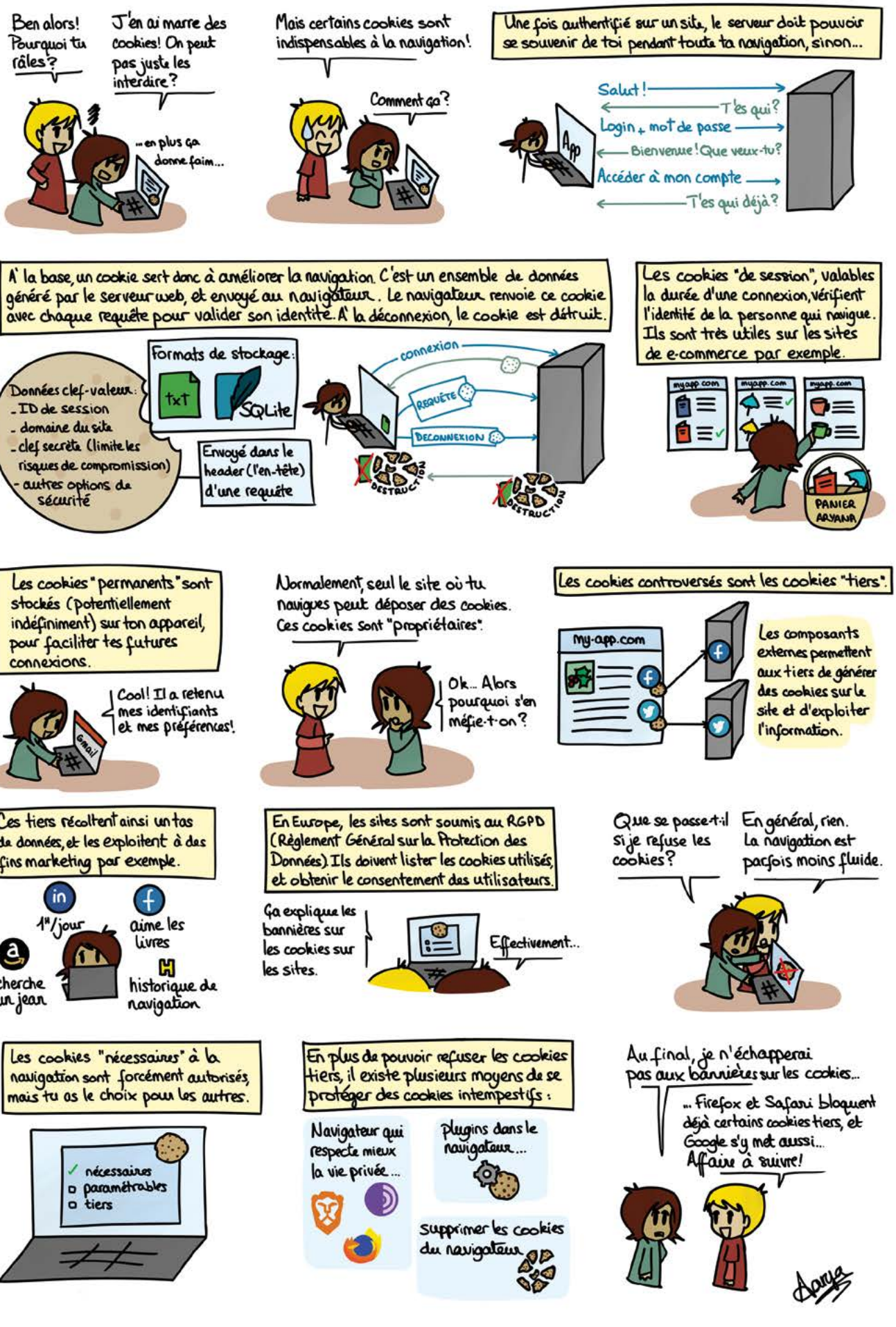


Haute Dispo à tout prix !

L'Authentification



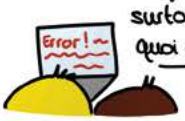
Les Cookies



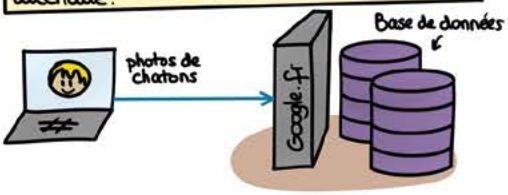
Le Cache

Hmm... C'est un soucis de cache... Tu t'y connais en gestion de cache?

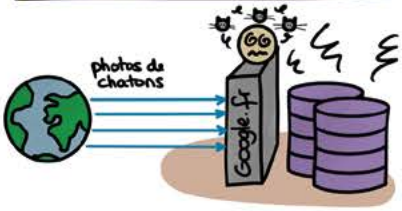
Baf... Je sais surtout à quoi ça sert.



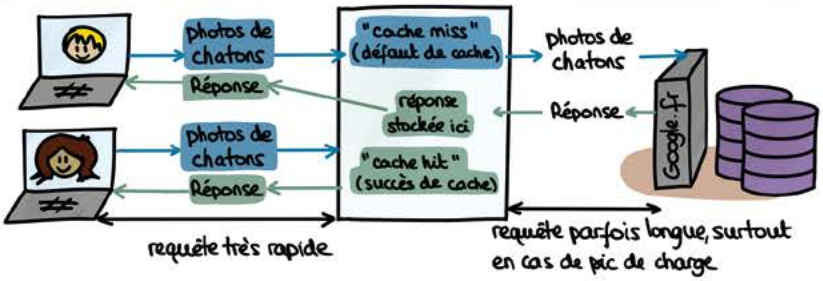
A chaque clic ou recherche, une requête est envoyée à un serveur pour afficher la page attendue.



Sauf que ce sont souvent les mêmes requêtes qui reviennent...



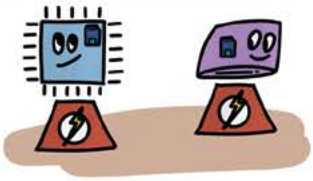
Le cache est un système de mémoire intermédiaire, qui conserve les données dans le but d'une utilisation ultérieure. L'objectif est d'accéder plus rapidement aux données souvent utilisées ou générées par des processus complexes. Il y a des caches à divers stades de la requête.



Un cache a une taille limitée. Une fois plein, il écrase les anciennes données par des données plus pertinentes.

S'il est trop grand, il perd en performance.

Il existe des caches hardware, pour le processeur et les disques durs, qui ont besoin d'être extrêmement rapides...



... et une floppée de caches software :

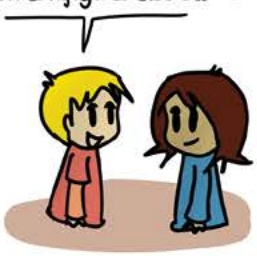
Enfin, les CDN (Content Delivery Network) peuvent stocker en cache une version de ton site, à divers endroits du monde.

Ok c'est pas mal!

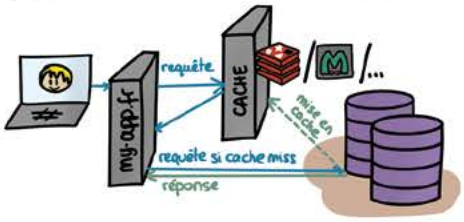
Mais je ne sais pas comment le configurer...

Le navigateur gère lui-même le stockage en cache sur le disque dur des utilisateurs. L'utilisateur a la main seulement pour le vider.

Côté base de données, un cache est souvent déjà mis en place et on configure surtout sa taille.



Côté serveur web, on peut configurer un CDN ou des outils comme Redis ou Memcached.



On peut donc choisir la taille du cache, et ce qu'on y met!

Donc les requêtes complexes ou récurrentes?

Voilà!

C'est aussi un indicateur à surveiller : s'il se remplit trop vite, ta requête est peut-être trop complexe...

Ma requête...?

Dans ton code j'ai trouvé une requête de plusieurs lignes... Je pense que c'est ça qui a cramé le cache de la base de données..

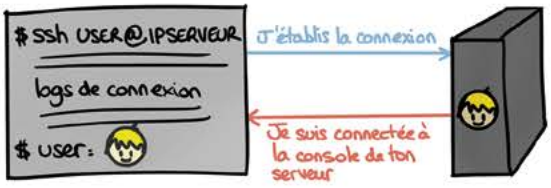
Le SSH

Dis, tu peux te connecter à mon serveur applicatif ?

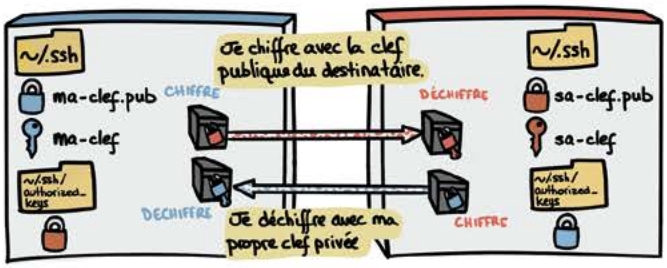
Oui !

On utilise le protocole SSH (Secure SHell) pour se connecter sur un serveur distant.

On indique l'utilisateur et l'adresse IP du serveur à contacter :



La sécurité des flux est assurée par un système de cryptographie asymétrique. Chaque machine a une paire de clefs privée ("sa-clef") et publique ("sa-clef.pub"). Je déchiffre le message chiffré par ma clef publique avec ma clef privée.



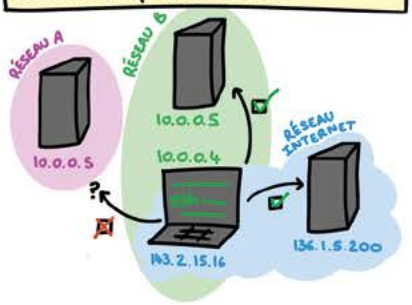
Donc quand tu te connectes, il chiffre tous les flux entre un client et un serveur. Le processus est similaire au https.

Attention ! Ne communique jamais ta clef privée. Si elle est compromise, re-génère une nouvelle paire de clefs.

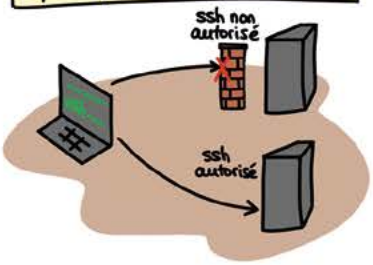
Avec ça, tu peux te connecter à n'importe quel serveur ?

Non, heureusement ! Les serveurs doivent remplir certains critères.

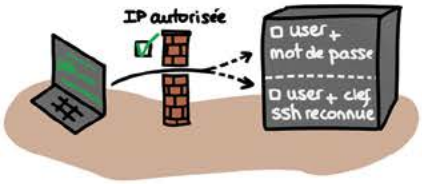
Premièrement, le serveur appelé doit être accessible depuis ton réseau



Ensuite, le serveur doit accepter les flux SSH entrants (par défaut sur le port 22).

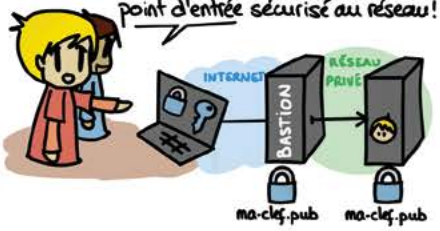


Enfin, il faut avoir le droit d'accéder au serveur : IP autorisée + User du serveur + mot de passe ou clef ssh reconnue



Mais mon serveur n'est pas exposé sur Internet, et tu n'es pas dans son réseau...

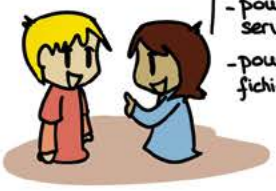
Je "rebondis" sur le "bastion", un point d'entrée sécurisé au réseau !



Dans quels cas fais-tu du SSH ?

Les principales raisons sont :

- pour configurer des serveurs (via ansible par exemple)
- pour débugger des serveurs
- pour transférer des fichiers sur un serveur



Et si je veux faire du SSH ?

Voici quelques petites choses à savoir :

- ssh-keygen : génère une paire de clefs privée-publique
- ~/.ssh/config : fichier pour configurer sa commande ssh
- bien restreindre les droits de la clef privée, car sinon la clef publique est inutilisable.
- il existe des outils pour empêcher le brute force ssh (ssh-guard, fail2ban...)
- Windows : on utilise "putty" pour le ssh.

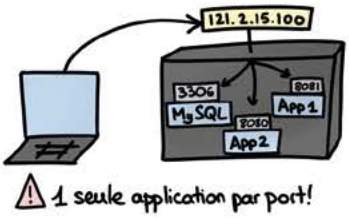
Manue

Les Ports et les Protocoles

C'est quoi un port ?

C'est un point d'entrée sur ton serveur. Ça lui permet de distinguer les applications sur le serveur.

Le port est nécessaire pour établir une connexion :



⚠ 1 seule application par port !

Comment savoir quel port contacter ?

Les ports vont de 0 à 65535.

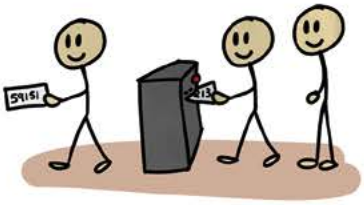
C'est beaucoup !

Les ports 0 à 1023 sont "reconnus" ou "réservés", et enregistrés auprès de l'IANA (Internet Assigned Numbers Authority). Ce sont surtout des processus système très utilisés.

STARS DE L'IANA !

Les ports de 1024 à 49151 sont "enregistrés" ou "semi-réservés".

Les ports 49152 à 65535 sont "dynamiques" ou "éphémères", valables juste sur la durée de la connexion.



On se connecte à un port via un protocole. C'est une convention - un ensemble de règles - qui permet à plusieurs appareils ou logiciels de communiquer.

Protocole : English (UK)

Les modèles "OSI" et "TCP/IP" séparent ces protocoles par couches.

TCP/IP	OSI
APPLICATION	
	PRÉSENTATION
	SESSION
TRANSPORT	
INTERNET	RÉSEAU
HÔTE-RÉSEAU	LIAISON
	PHYSIQUE

Il existe un tas de protocoles, certains désuets, et d'autres tout récents !

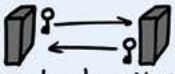
Les protocoles les plus connus sont :

HTTP : HyperText Transfer Protocol

HTTPS : HTTP + couche de chiffrement SSL ou TLS

Protocole de communication client-serveur développé par le World Wide Web.

SSH : Secure Shell



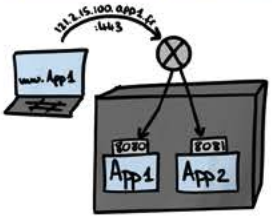
protocole qui impose un échange de clefs de chiffrement en début de connexion.

SMS : Short Message Service

Hmm... J'ai compris que certains processus ont des ports définis... Mais comment connaît-on les autres ?

Certaines commandes permettent de voir les ports actifs sur ton serveur.

Dans notre réseau, c'est à nous de configurer les ports et rediriger les flux.



On peut aussi changer les ports par défaut pour brouiller les pistes...

... au risque d'embrouiller l'équipe technique aussi...

Donc si quelqu'un connaît l'adresse du serveur et le port d'un service, il est vulnérable !

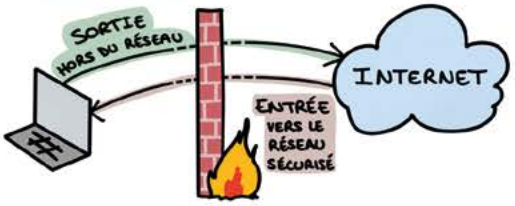
D'où les systèmes d'authentification (comme on l'a vu pour le ssh), et... l'utilisation de firewalls !

Le Firewall

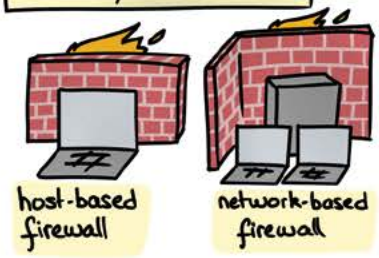
Tu vois ? J'ai la bonne IP, le bon port, mais la connexion au serveur échoue...

Ah ! C'est à cause du firewall (pare-feu) qui sécurise ton serveur !

Un firewall contrôle, analyse et autorise ou non la communication entre plusieurs réseaux, en fonction de règles prédéfinies. Il peut y en avoir plusieurs entre 2 réseaux.

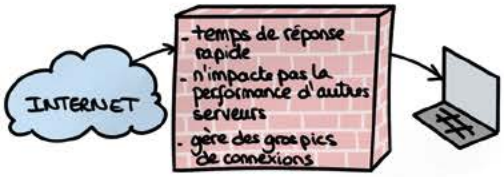


Il s'applique soit à un seul ordinateur, soit à un réseau.

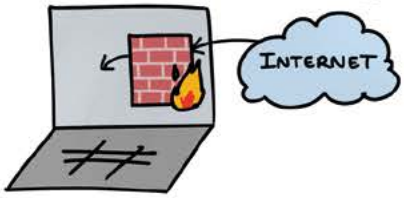


Il peut être physique ou virtuel.

Les firewalls matériels sont des machines dédiées à ce filtrage. Certains routeurs embarquent déjà un firewall.



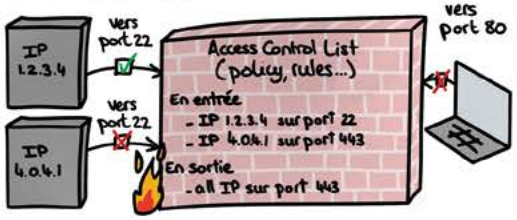
Les firewalls virtuels peuvent être plus ou moins complexes. Ils sont souvent utilisés pour protéger une infrastructure cloud ou des postes individuels.



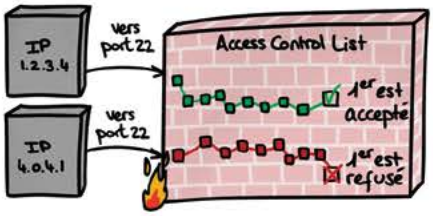
Et il filtre sur quoi?

Ça dépend des firewalls

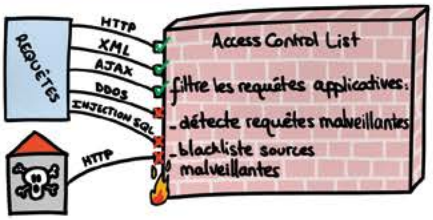
Le firewall sans état (stateless packet inspection firewall) analyse chaque paquet entrant en fonction d'une checklist configurée.



Le firewall à état (stateful packet inspection firewall) analyse le 1er paquet entrant et vérifie que chaque paquet est bien la suite du précédent.

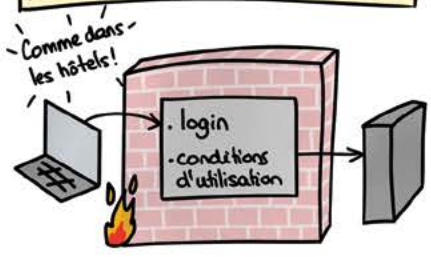


Le firewall applicatif (Web Application Firewall ou WAF) vérifie que les requêtes sont conformes au protocole attendu (HTTP, HTTPS), et protège des principales vulnérabilités applicatives.



Le firewall identifiant, filtre les connexions par utilisateur.

Le firewall captif intercepte les usagers pour leur présenter une page spéciale.



Du coup, un firewall ça suffit niveau sécurité ?

C'est un début. Il y a beaucoup à dire sur le sujet !

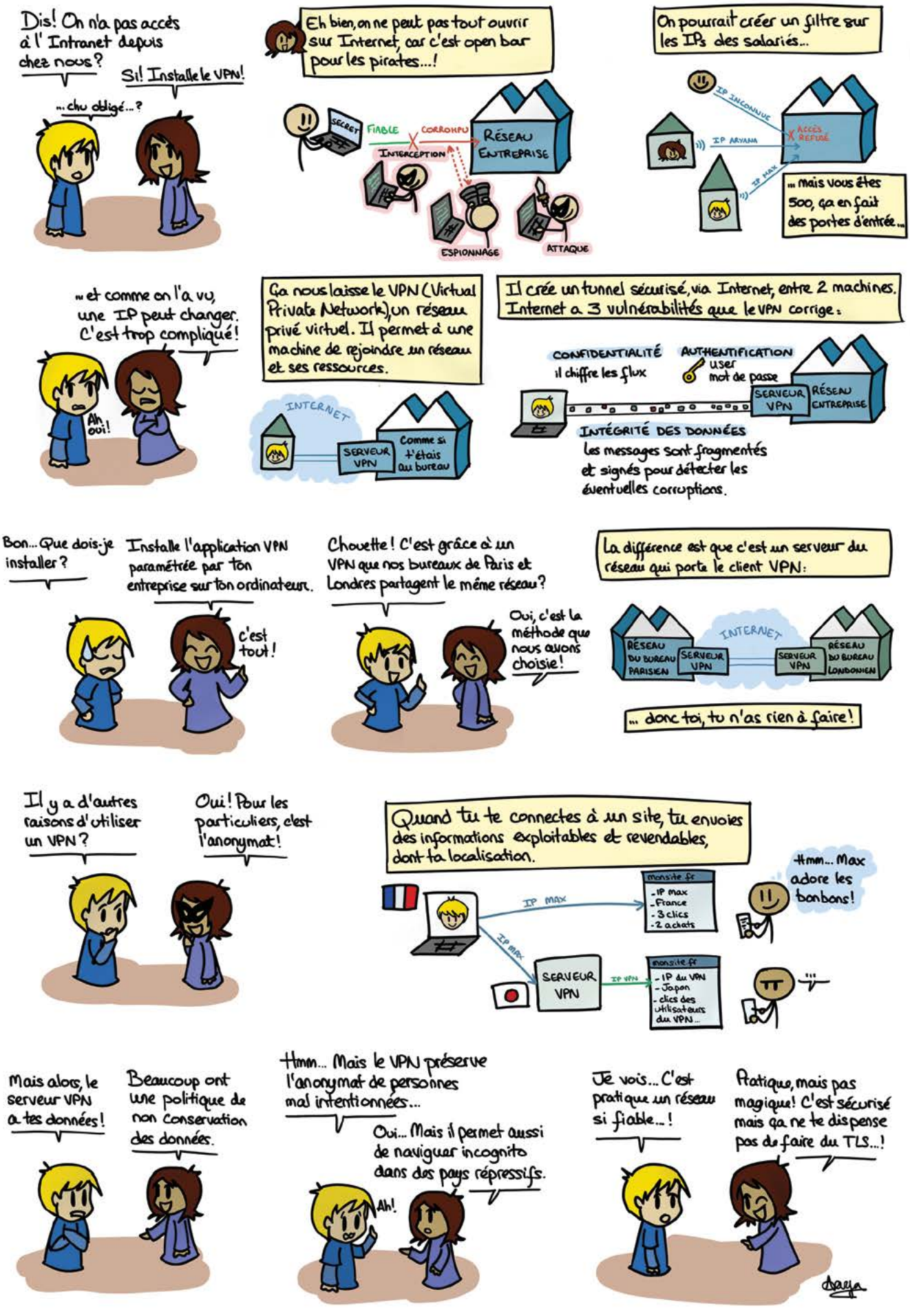
...Ok... Bon, et en attendant je fais comment ?

Heh ! Viens, on va whitelister ton IP dans le firewall !

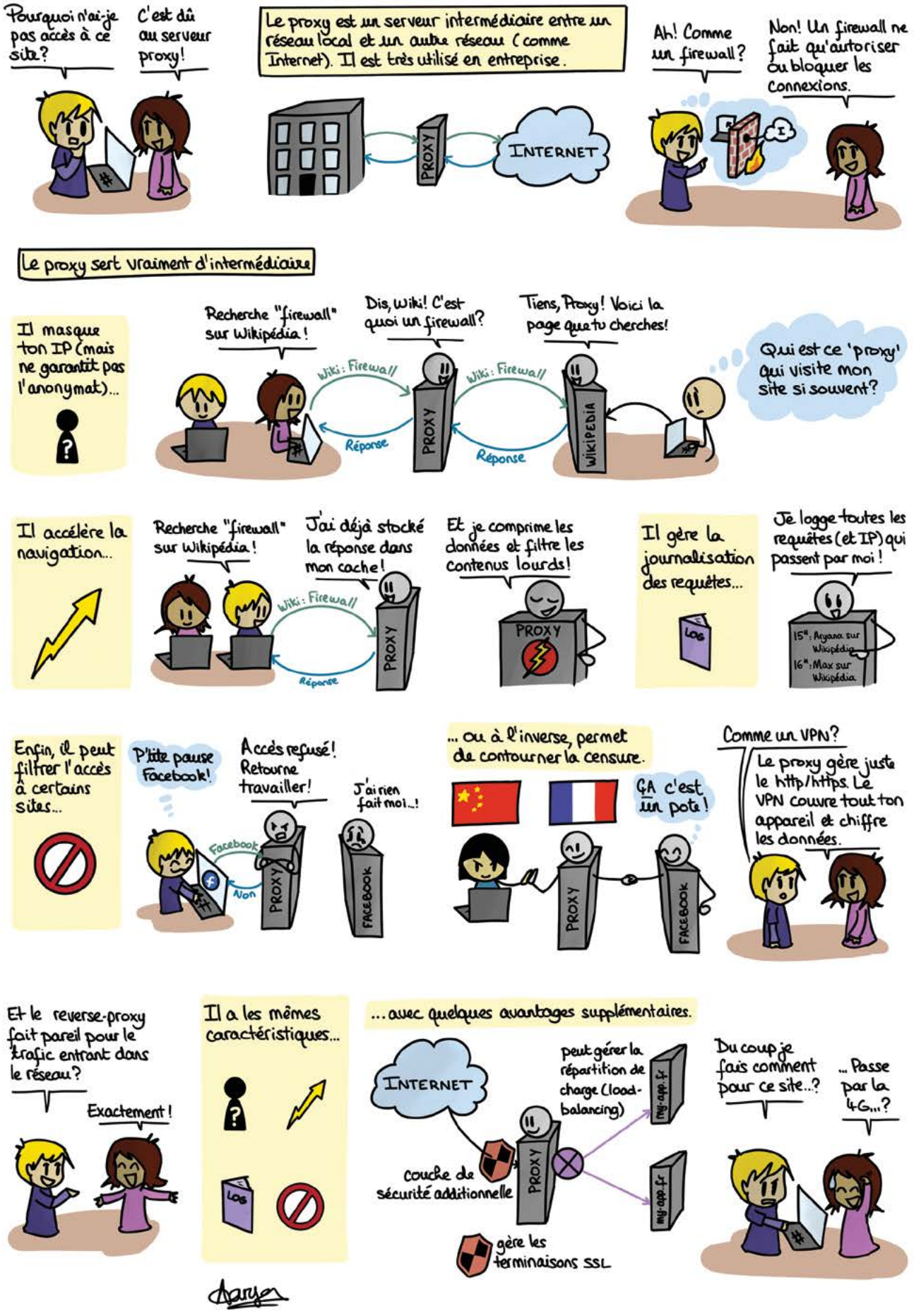
Aanya

L'informatique, ça creuse !

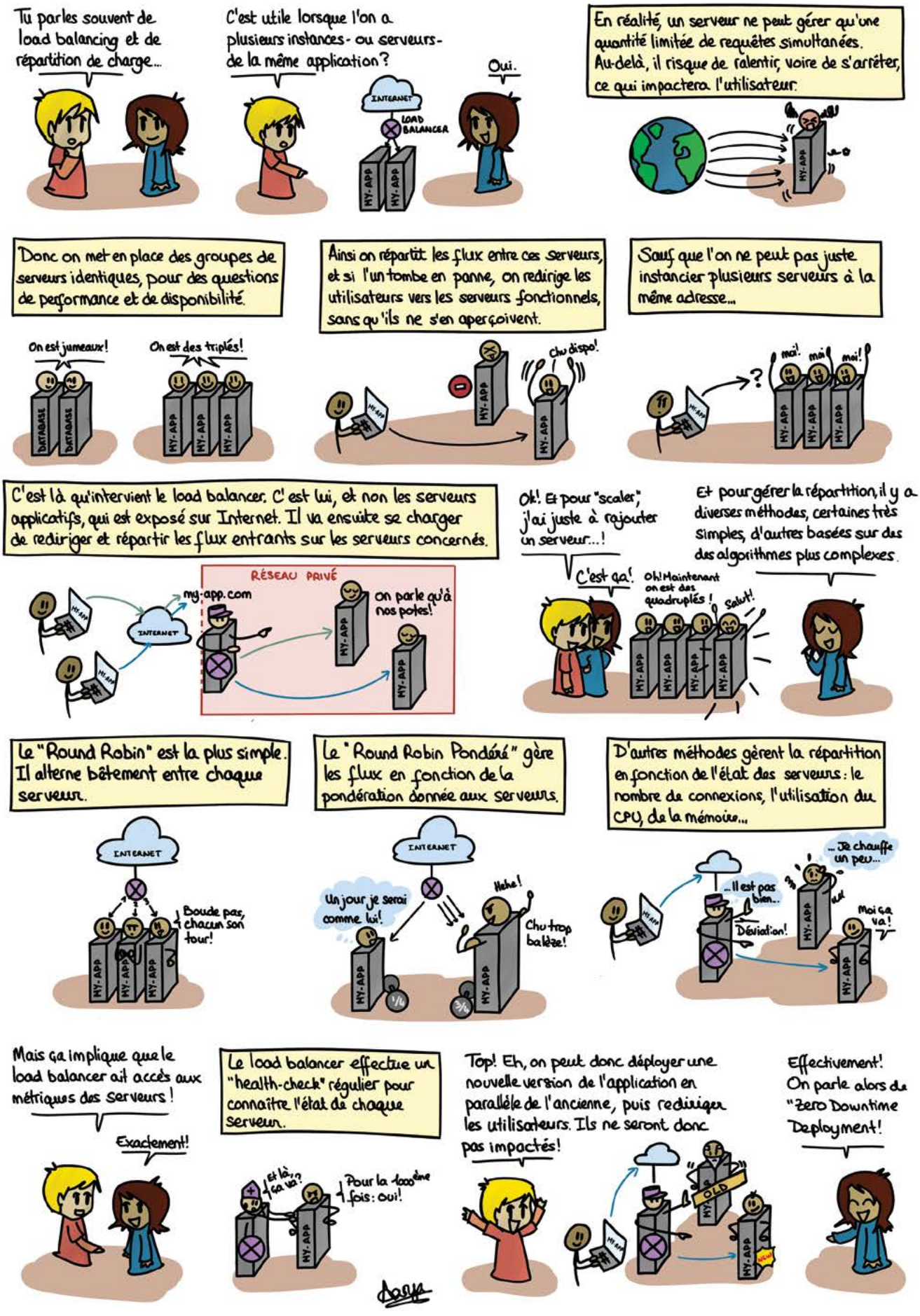
Le VPN



Le Proxy et le Reverse Proxy

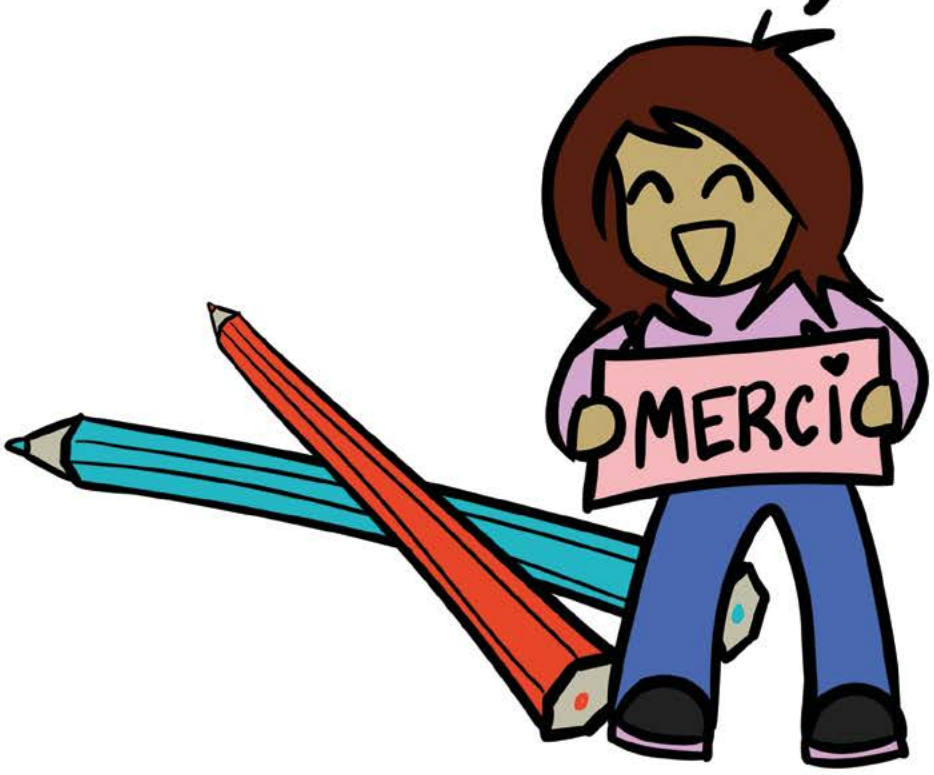


Le Load Balancer



J'adore mon métier !

MERCI à Edouard qui a lancé le projet, à la tribu Ops pour m'avoir donné les moyens de travailler dessus, à OCTO et à la Comm' pour la publication de l'ouvrage et à mes managers pour leur soutien. MERCI à tous les Octos et à tous ceux et celles qui m'ont relue, challengée et encouragée. C'est grâce à vous que mes planches sont ce qu'elles sont.



There is a Better Way

OCTO Technology

CABINET DE CONSEIL ET DE RÉALISATION IT

« Nous croyons que l'informatique transforme nos sociétés. Nous savons que les réalisations marquantes sont le fruit du partage des savoirs et du plaisir à travailler ensemble. Nous recherchons en permanence de meilleures façons de faire. THERE IS A BETTER WAY ! »

– Manifeste OCTO Technology

Part of Accenture

Dépot légal : Octobre 2021

Conçu, réalisé et édité par OCTO Technology.