

Queste de savoir

Zest of dev 18

19 mars 2022

Table des matières

	Introduction	1
1.	Évolution de l'infrastructure	1
1.1.	Sauvegardes	2
1.2.	Passage à Debian 11	2
1.3.	Rapatricier Munin	3
1.4.	Mise à jour de Matomo	3
2.	Organisation du projet et de l'équipe	3
2.1.	État de l'équipe	3
2.2.	Organisation du projet	4
3.	En avant pour la v30.2!	5
3.1.	Résumé des principaux changements depuis la v30.1	5
3.2.	Petits préparatifs finaux	5
3.3.	Planning de release	5
4.	Travaux pour la nouvelle organisation des publications	6

Introduction

Le jeudi 17 mars 2022, l'équipe de développement de Zeste de Savoir s'est réunie pour une réunion de coordination. Cela faisait plusieurs mois qu'une telle réunion n'avait pas eu lieu et il s'est dit *vraiment* beaucoup de choses!

Ce billet rend compte de notre réunion et vise à donner à la communauté quelques nouvelles ainsi que transmettre notre vision pour les mois à venir. Il y a notamment du travail prévu pour mettre en place les évolutions concernant l'organisation des publications sur le site.

Étaient présents à cette réunion les développeurs suivants, dans l'ordre alphabétique: @Aabu, @Amaury, @artagis, @philippemilink, @Situphen, @Stalone, ainsi que quelques spectateurs, dont @sgble.

1. Évolution de l'infrastructure

Sans infrastructure pour le faire tourner, le code de Zeste de Savoir ne servirait pas à grand-chose. En parallèle du développement du site, l'infrastructure du site évolue aussi afin de rester à un niveau satisfaisant de résilience et ne pas tomber dans l'obsolescence.

1. Évolution de l'infrastructure

1.1. Sauvegardes

La base de données du site est très précieuse, car ce qui fait Zeste de Savoir, c'est son contenu, que ce soit les forums ou les publications.

Anciennement, le système de sauvegarde copiait les données (chiffrées) du serveur de production vers le serveur de bêta pour y être stockées. On gardait 60 jours de sauvegarde, à raison d'une sauvegarde toutes les 2h pour tous les contenus (dépôts Git et base de donnée). Le problème qui s'est posé est le suivant: comment éviter de stocker trop de données, quand l'ensemble des sauvegardes pèse 50 Go? (le revers de la médaille de notre popularité :D)!

Des évolutions ont été étudiées puis mise en place. Les outils de sauvegarde et de compression ont été mis à jours. Borg a été généralisé pour toutes les sauvegardes, ce qui rend le système plus uniforme et plus paramétrable. @philippemilink stocke désormais une copie des sauvegardes (chiffrées) pour encore plus de résilience¹footnote:1. Nota: les sauvegardes sont utilisées pour que la bêta ait des données de prod pour nos tests, ce qui est documenté et automatisé grâce à un script.

À l'avenir, on pourrait optimiser à la marge le système. Par exemple, il est possible de changer la durée des sauvegardes en passant à une complète tous les deux jours puis 23 incrémentales au lieu que 1 tous les jours puis 11 incrémentales. Il faudrait également sauvegarder Matomo, pour mitiger le risque de perdre l'historique des statistiques de nos publications.

Le sujet des sauvegardes a également été discuté [sur le forum](#) .

1.2. Passage à Debian 11

Actuellement, la bêta et la prod utilisent Debian 10.

Debian 11 est sorti entre temps et il est temps de songer à mettre à jour les serveurs pour utiliser cette version. Il n'y a pas d'urgence, mais il est de bon goût de suivre les versions pour éviter de se retrouver dans l'urgence lors de l'arrivée de la fin du support. L'objectif est de mettre à jour fin avril si possible, mais pas de pression particulière sur la date.

Cette mise à jour lève quelques questions sur les versions des logiciels utilisés par le serveur, entre celles proposées dans Debian 10, celles qu'on utilise nous (potentiellement ultérieures) et celles distribuées avec Debian 11. Par exemple, Node.js 12 arrive en EOL, et on est en 12.22, alors que Debian propose la 12.21...

Ce sujet a également été discuté [sur le forum](#)).

1. ²footnote:1 Cette sauvegarde externe permet d'éviter de perdre les données au cas où les accès au serveur de production et de bêta seraient compromis ou corrompus, ce que ne permet pas la sauvegarde simplement sur la bêta.

2. Organisation du projet et de l'équipe

1.3. Rapatrier Munin

Munin est un logiciel qui permet de superviser nos serveurs et d'avertir en cas de soucis.

Actuellement, notre Munin est hébergé par un ancien développeur de Zeste de Savoir sur un de ses serveurs.

L'objectif est de déployer Munin sur nos serveurs à nous. On a un serveur pour la bêta et un pour la prod, qui pourront se surveiller l'un l'autre.

Il serait mieux de le faire avant le passage à Debian 11 afin d'observer les éventuelles pertes de performance.

1.4. Mise à jour de Matomo

Matomo est le logiciel utilisé pour collecter et traiter les statistiques d'utilisation de Zeste de Savoir. Il permet de garder toutes nos données en interne et de ne pas dépendre de services tiers avec toutes les questions qui se posent en termes traçage.

Il faudrait le mettre à jour pour suivre son évolution et ne pas se retrouver à la traîne.

2. Organisation du projet et de l'équipe

2.1. État de l'équipe

2.1.1. Bilan des contributions depuis 2021

Github permet facilement de lister les [contributeurs depuis janvier 2021](#) ↗ .

On compte:

- 3 contributeurs réguliers (> 5 commits et réguliers sur l'année): @Situphen, @Aabu, @philippemilink. Ils forment *de facto* la *core team*.
- 4 contributeurs périodiques (> 5 commits et irréguliers sur l'année): @Amaury, @firm1, @viki53, @artraxis.
- 10 contributeurs ponctuels: @entwanne, @Migwel, @Spacefox, @Moté, ksauvee, @Deuch-nord, @Taguan, @Rowin, @Luuka, @Melcore.

Il faut faire attention à cette métrique: les commits ne sont pas forcément de nouvelles fonctionnalités significatives apportées aux utilisateurs sur le site, mais plus souvent de la maintenance: mise à jour des dépendances, de la doc, etc.

2.1.2. @philippemilink prend (encore) du galon

@artraxis quitte son rôle de responsable du projet et prend sa relève. @artraxis garde cependant son accès au serveur en cas de pépin, mieux vaut être assuré en cas de gros soucis.

2. Organisation du projet et de l'équipe

2.1.3. Perspectives sur l'évolution de l'équipe

État d'esprit majoritaire des développeurs actuels: motivés, mais peu disponibles. Le temps disponible et la motivation des développeurs sont plus ou moins variables au cours du temps en fonction des contributeurs.

Pour l'instant il y a assez de contributeurs pour maintenir le site à son niveau actuel, mais on en manquerait pour se lancer dans de gros projets. Pas besoin de faire un appel à contributeurs *pour le moment*.

Est-ce qu'il y a un retour sur investissement à lancer des appels à contributeurs? Difficile de le dire. Peut-être que ce temps là serait plus efficace si on le consacrait à améliorer la documentation ou les tickets faciles par exemple. Cela dit on a un exemple qui a fonctionné un peu: [Hacktoberfest](#) [↗](#).

Comment augmenter l'attractivité du projet et augmenter la motivation?

- Faire les QA le plus rapidement possible pour garder le développement, même ponctuel motivant pour les contributeurs.
- Communiquer plus vers l'extérieur de manière générale, comme l'article sur la faille, des billets techniques, etc.
- Organiser un Hackathon sur une journée ou demi-journée comme il y en a eu pour l'écriture de contenu ?

2.2. Organisation du projet

Comment éviter une aussi grande période sans nouvelle version telle qu'on l'a connu ces derniers mois?

Le consensus qui a émergé de la discussion est le suivant:

- voir pour déployer la bêta en continu
- rester sur des nouvelles versions manuelles sur la prod, mais plus petites et plus régulières (par exemple tous les mois).

Déployer en continu en prod a été jugé trop risqué pour le moment par rapport à la maturité technique et la fiabilité du processus de développement.

La nouvelle stratégie de *release* et de déploiement permettra tout de même de réduire le temps entre la fusion de la PR et le moment où elle est déployée en production.

Reste-t-on sur le modèle «chacun fait ce qu'il veut sans direction particulière» ou est-ce que l'on définit des grands axes de développement? On restera probablement sur un modèle assez libre, comme on est tous bénévoles et que la motivation est variable selon les tâches. Cependant, on souhaite aussi mieux partager la vision et le travail à faire sur les gros chantiers, afin d'aider les contributeurs ponctuels à travailler sur les tâches prioritaires pour le site et s'intégrer aux plus grosses évolutions du site.

3. *En avant pour la v30.2!*

3. En avant pour la v30.2!

Après une très longue période sans nouvelle version, il est temps de préparer la v30.2!

3.1. Résumé des principaux changements depuis la v30.1

Changements visibles par les utilisateurs:

- Mise à jour possible d'images avec un SVG (le téléversement de SVG était déjà possible, mais pas la mise à jour).
- Affichage des comptes sur un même réseau IPV6 dans la page de multi-comptes (visible par les modos).
- Importe l'information 'ready_to_publish' depuis les archives (visible par @entwanne).
- Mise à jour de EasyMDE (corrige des bugs sur les boutons).

Changements invisibles pour les utilisateurs, mais pas pour les développeurs:

- Moins de jQuery
- tests plus nombreux et de meilleure qualité
- nombreuses mises à jour de dépendances
- nouvelle version de Django (3.2), qui vient remplacer celle actuelle (2.2), en fin de vie

3.2. Petits préparatifs finaux

À faire avant la v30.2: fusionner la PR «Mise à jour triviale des dépendances Python»

Et si l'éventualité se présente (i.e. QA faite à temps, on attendra pas ces PR), fusion de:

- Mise à jour des dépendances front
- Améliore la connexion aux réseaux sociaux
- Journal des événements sur un contenu

Au passage, nous avons fait un petit rangement des PR en cours, avec le dépoussiérage des PR récentes, et la mise au cimetière des PR en décomposition (marquées «zombie» ou essentiellement abandonnées par leur auteur).

3.3. Planning de release

On envisage peut-être une mise en bêta le WE du 27 mars, ou alors début avril selon dispo de @philippemilink.

4. Travaux pour la nouvelle organisation des publications

Récemment, un groupe de travail a eu lieu autour de l'organisation des publications. Les résultats des travaux sont disponibles [sur le forum](#) . Ces travaux ont des implications techniques sur le site, qui subira des évolutions avec un travail de développement significatif.

Pour que ces développements avancent, il est intéressant de les structurer, pour rendre le chemin praticable, partager une vision et permettre une avancée progressive sans cahots.

Un [document de travail collaboratif](#) a été créé à cette fin. Il donnera lieu à des tickets sur GitHub, ainsi qu'un projet GitHub pour le suivi et garder le lien entre les tickets.