



HAL
open science

Framaspace, 10 000 instances Nextcloud pour les associations

Pierre-Yves Gosset, Luc Didry

► To cite this version:

Pierre-Yves Gosset, Luc Didry. Framaspace, 10 000 instances Nextcloud pour les associations. JRES (Journées réseaux de l'enseignement et de la recherche) 2024, Renater, Dec 2024, Rennes, France. <hal-04893867>

HAL Id: hal-04893867

<https://hal.science/hal-04893867v1>

Submitted on 17 Jan 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY-NC 4.0 - Attribution - Non-commercial use - International License

Framaspace, 10 000 instances Nextcloud pour les associations

Luc Didry

Framasoft
10 bis rue Jangot
69 007 Lyon

Pierre-Yves Gosset

Framasoft
10 bis rue Jangot
69 007 Lyon

Résumé

En novembre 2022, l'association Framasoft annonçait un projet techniquement et politiquement ambitieux : Framaspace.

En effet, Framasoft souhaite mettre à disposition, d'ici 3 ans, jusqu'à 10 000 espaces « cloud » collaboratifs, gratuitement, à destination des associations et collectifs militants. 40 Gio et de nombreuses fonctionnalités à se partager entre 50 comptes sur un espace de type <https://monasso.frama.space>.

Framaspace, c'est un projet avec de multiples ambitions :

- *accroître la notoriété du logiciel libre Nextcloud (solution de cloud collaboratif : fichiers, agendas, contacts, etc.) en France et dans les territoires francophones ;*
- *fournir jusqu'à 10 000 instances Nextcloud, gratuitement, aux petites associations et collectifs militants, afin de leur permettre de « sortir » de Google Apps ou Office 365 ;*
- *faciliter les pratiques collaboratives libres, à la fois intra-associations et inter-associations, afin de renforcer la capacité et le pouvoir d'agir de ces structures pour mieux faire face au monde actuel et préparer celui qui vient.*

À ce jour, plus de 1 280 instances Nextcloud ont été déployées avec succès.

Pour cela, nous utilisons une infrastructure relativement spécifique à ce projet, mais qui peut être répliquée dans d'autres contextes : stockage objet, réplication de nombreux services sous-jacents, développement de logiciels spécifiques...

L'objectif de cette conférence sera donc de présenter l'infrastructure technique du projet et faire un retour d'expérience après plus d'un an, en détaillant les soucis rencontrés et les leçons apprises.

Mots-clefs

Nextcloud, Automatisation, PostgreSQL, MinIO, HAProxy, Framasoft, Framaspace

1 Choix des morceaux et morceaux de choix

1.1 Identification des services nécessaires

Nextcloud, pour l'usage que nous souhaitons en faire avec Framaspace, requiert plusieurs éléments, certains classiques, certains plus spécifiques à nos besoins :

- une base de données ;
- un serveur web ;
- un interpréteur PHP ;
- du stockage de fichiers ;
- du cache mémoire ;
- un serveur **Imaginary** (pour la génération d'aperçus d'images) ;
- une suite bureautique (traitement de texte, tableur, diaporama, etc.) ;
- une solution de centralisation de logs ;
- un répartiteur de charge HTTP.

Les quatre ou cinq premiers services nécessaires font partie de la stack classique d'une installation de Nextcloud, les autres répondent à des besoins divers :

- le serveur **Imaginary** permet de reporter la charge de la génération des aperçus d'images sur une machine différente de celle qui interprète le code PHP ;
- la suite bureautique est là pour répondre aux besoins des usagers ;
- la centralisation des logs permet à un·e administrateur·ice de ne pas avoir à courir après les logs des différentes composantes de Framaspace pour régler un problème ;
- le répartiteur de charge sert à anticiper la croissance de Framaspace : une intégration précoce aide à l'élasticité de l'infrastructure.

1.2 Choix des logiciels

Pour chaque service identifié à la section précédente, plusieurs logiciels sont généralement susceptibles de répondre au besoin. Il a donc été nécessaire de faire des choix.

Il faut aussi noter que nous souhaitons, dès la création du projet, avoir une infrastructure la plus évolutive et redondante possible, de façon à pouvoir accompagner la croissance de Framaspace et éviter les points de défaillance unique.

PostgreSQL a été choisi pour la base de données pour sa robustesse, le peu de modifications de configuration nécessaires à de bonnes performances pour notre usage (aucune modification n'est nécessaire pour un usage « classique » de Nextcloud).

Nginx est notre serveur web de prédilection pour sa vélocité et... pour les plus grandes expériences et affinités que nous avons avec celui-ci.

PHP-FPM est un choix classique pour servir du PHP avec Nginx, et nous trouvons intéressant qu'il soit possible de le séparer du serveur web (même si cette idée fut abandonnée en cours de route).

Pour le stockage des fichiers, nous sommes partis sur du stockage objet avec MinIO. Celui-ci est très simple à prendre en main et tester et se révèle très agréable à l'usage (un seul binaire à déployer, interface web de gestion fournie nativement, client bien pensé, etc.). Le seul inconvénient

que l'on puisse lui trouver est l'impossibilité d'ajouter un serveur au *pool* de serveurs en cas d'augmentation de l'espace disque nécessaire. Pour ajouter de l'espace disque, il faut ajouter un nouveau *pool* au *cluster* MinIO.

Redis est notre choix pour le cache mémoire. Il possède en effet des avantages certains sur ses concurrents : possibilité d'être interrogé à distance au contraire d'APCu et possibilité de haute disponibilité au contraire de Memcached. Redis servira aussi pour le stockage des sessions PHP.

Au contraire des autres services pour lesquels nous avons dû trancher entre plusieurs logiciels, nous avons choisi les deux solutions disponibles pour Nextcloud, à savoir Collabora et OnlyOffice. Les administrateur·ices d'espaces Framaspace peuvent ainsi choisir l'une ou l'autre suite et modifier ce choix à tout moment.

N'ayant pas de volonté d'exploitation des logs mais plutôt de se simplifier la vie en ayant tous les logs disponibles à un seul endroit, nous avons rapidement évacué l'utilisation d'une stack complexe type ELK et choisi d'utiliser rsyslog : simple, efficace et nous en avons l'expérience.

Enfin, HAProxy est notre répartiteur de charge HTTP. Sa documentation est bien fournie et très accessible et sa réputation est excellente.

Nous n'avions pas précédemment listé le système d'exploitation dans les besoins de Nextcloud, mais il faut bien évidemment en choisir un. Nous avons choisi Debian pour une simple raison : pourquoi choisir autre chose que ce qui fonctionne parfaitement ?

2 Chaque chose à sa place et une place pour chaque chose

Le cahier des charges du projet demandait un système évolutif avec une bonne tolérance aux pannes, le tout avec une contrainte financière, le budget de Framasoft n'étant pas illimité.

L'évolutivité de l'infrastructure est d'ailleurs liée aux limitations budgétaires : il était impensable de monter dès le départ une infrastructure pour 10 000 instances Nextcloud si cet objectif était pensé pour être atteint plusieurs années dans le futur.

Si certains services ont immédiatement été mis en place avec de la redondance, d'autres voient leur non-redondance compensée par la capacité d'évolution et l'automatisation du système.

2.1 Les services redondés

Le *cluster* PostgreSQL de bases de données étant un service des plus critiques, celui-ci est redondé. Nous avons limité la redondance à deux serveurs pour commencer, mais d'autres serveurs pourront facilement être ajoutés. À noter : la répllication n'est pas multi-directionnelle mais en mode *primary/replica*, afin d'éviter une dégradation des performances.

MinIO nécessite un minimum de quatre serveurs pour proposer un service résistant aux pannes. Le site officiel offre un [service appelé Erasure Code Calculator](#), qui permet de calculer quel espace de stockage sera disponible selon le nombre de serveurs et la configuration du *cluster*, ainsi que la tolérance à des pannes de disques ou de serveurs. Cet outil nous a permis de voir qu'un cinquième serveur nous permettrait, pour un coût convenable, de repousser de manière sensible le moment où nous devrions ajouter un *pool* de serveurs au *cluster* MinIO et d'augmenter la tolérance aux pannes. C'est pourquoi nous avons choisi un *cluster* de cinq serveurs. La volumétrie prévue étant trop importante pour être sauvegardée suffisamment vite pour avoir des sauvegardes pertinentes, nous avons doublé le nombre de serveurs pour créer un deuxième *cluster* MinIO, grâce à la fonctionnalité de répllication présente dans le logiciel.

Redis peut être redondé via deux modes : *sentinel* ou *cluster*. Le mode *sentinel* a l'avantage de nécessiter un nombre minimum (trois) de serveurs inférieur au mode *cluster* (qui en demande au

moins six) mais a le désavantage de faire transiter toutes les écritures par un seul serveur primaire, ne permettant ainsi pas de répartition de charge, au contraire du mode *cluster* qui va répartir l'écriture des données sur plusieurs serveurs, eux-mêmes recopiant leurs données vers des serveurs secondaires. Le mode *cluster* a donc été choisi.

2.2 Les services non redondés

Si on regarde attentivement la liste des services redondés, on constate que ceux-ci contiennent tous des données : base de données, fichiers et mémoire cache / sessions PHP.

En effet, il est généralement long de remonter un service de données depuis une sauvegarde, aussi avons-nous voulu nous assurer de pouvoir simplement basculer sur un réplikat en cas de souci.

Mais pour les autres services évoqués, c'est autre chose : ils sont « vides » et, par le biais de la forte automatisation de l'infrastructure, facilement remplaçables.

Nginx, associé à PHP-FPM, n'est dépendant que de la base de code de Nextcloud et de quelques fichiers de configuration. Tout ceci étant géré par l'automatisation, le service Nginx/PHP-FPM n'a initialement pas été redondé. Seule la hausse de la charge de l'infrastructure a nécessité l'ajout d'un nouveau serveur dédié à ce service, accessible via le répartiteur de charge HAProxy.

Imaginary est un service non essentiel au bon fonctionnement de Nextcloud (même s'il permet de réduire la charge sur les serveurs Nginx/PHP-FPM). L'installation et la configuration d'un serveur Imaginary est encore plus aisée que pour Nginx/PHP-FPM, encore une fois grâce à l'automatisation du système. Si jamais nous avons besoin d'un service en plus en cas de trop forte utilisation de ce service, HAProxy pourra encore une fois jouer son rôle de répartiteur de charge.

Les services des suites bureautiques sont lancés via la solution de conteneurisation Podman et l'automatisation permettra là aussi, soit la reprise d'activité, soit l'augmentation de la charge supportable. Contrairement à Nginx et Imaginary, HAProxy ne fera pas de répartition de charge entre des services de suite bureautique mais se contentera de rediriger les requêtes vers le serveur idoine. L'augmentation du nombre d'espaces a récemment nécessité l'ajout d'un nouveau serveur de suites bureautiques, non pas à cause de la charge serveur, mais de l'espace disque qui venait à manquer.

En ce qui concerne le service rsyslog, étant tout à fait non essentiel à la fourniture du service Framaspace, il n'est pas prévu de redondance ni de répartition de charge. L'automatisation permettra de remonter aisément le serveur pour reprendre une activité interrompue. Les journaux sont bien évidemment sauvegardés et pourraient être remis en place.

Le seul point de défaillance unique du système est HAProxy, qui ne nécessite que de la configuration, encore et toujours gérée de façon automatisée : cela nous permet de remonter un frontal HAProxy très rapidement sur n'importe quel serveur, ce qui explique que nous soyons confiants au point de ne pas redonder ce service.

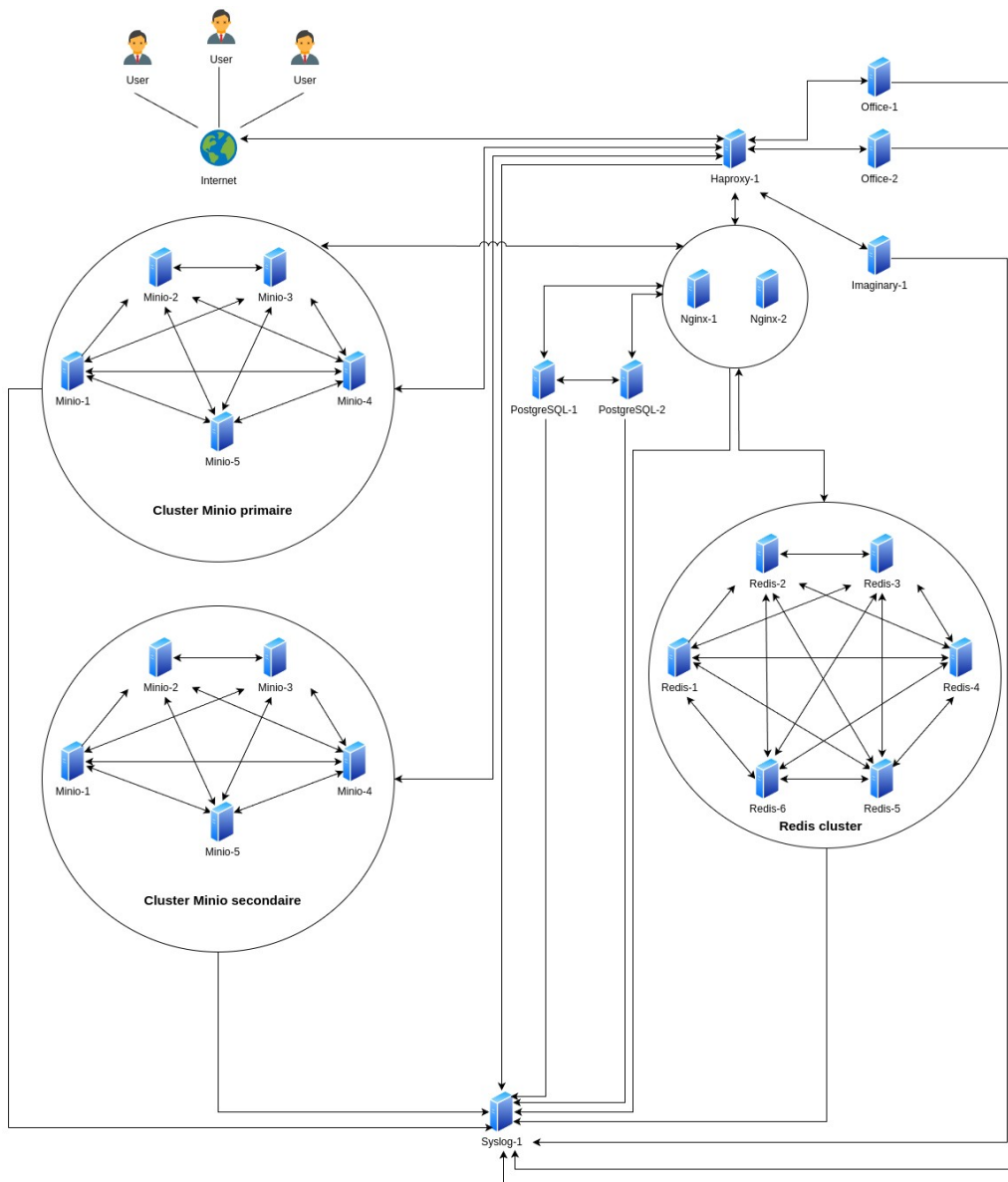


Figure 1 — Schéma de l'infrastructure de Framaspace au 16 septembre 2024

3 Encore des logiciels !

3.1 Les liaisons (pas) dangereuses

Si nous avons jusque là décrit les différents logiciels composant le service Framaspace, nous n'avons pas évoqué un point important : comment ceux-ci communiquent-ils ?

S'il eut été possible de faire communiquer les logiciels via des liaisons sécurisées avec SSL/TLS, le coût en ressources d'établissement des connexions n'aurait pas été négligeable. C'est pourquoi nous avons utilisé Wireguard pour monter des tunnels VPN entre les différents serveurs ayant besoin de communiquer.

Là encore, l'automatisation vient à notre rescousse pour la configuration des tunnels, de façon à pouvoir ajouter ou supprimer des serveurs aisément.

3.2 Des répartiteurs de charge, encore

Si HAProxy fait office de répartiteur de charge au niveau frontal, d'autres connexions ont aussi besoin de tels logiciels.

Pgpool-II viendra permettre la répartition des requêtes vers les serveurs PostgreSQL, en discriminant entre autres sur les requêtes en écriture, qui iront forcément vers le serveur primaire et les requêtes en lecture, qui pourront être effectuées sur tous les serveurs. Il est installé sur les serveurs Nginx/PHP-FPM.

Sidekick est un répartiteur de charge pour MinIO. Il est officiellement recommandé d'utiliser Sidekick au plus près des services utilisant le cluster de stockage, c'est pourquoi celui-ci est lui aussi installé sur les serveurs Nginx/PHP-FPM, ainsi que sur le serveur HAProxy pour les accès externes qui nous permettent de gérer les *clusters*.

3.3 De la virtualisation

À part les serveurs MinIO et un couple Nginx/PHP-FPM sur les deux, tous les services sont hébergés sur des machines virtuelles KVM, gérées par Ganeti.

3.4 De la sauvegarde !

Une infrastructure redondée n'est pas une infrastructure sauvegardée. À part le cluster MinIO, impossible à sauvegarder de par sa volumétrie, nous sauvegardons bien évidemment nos données.

Barman est la solution de sauvegarde des bases de données PostgreSQL de Framasoft depuis maintenant plusieurs années. Ses sauvegardes « au fil de l'eau », sa facilité de configuration et d'utilisation ainsi que sa fiabilité nous séduisant toujours, nous n'avons aucune raison d'en changer.

Pour la sauvegarde des fichiers, BorgBackup et son compare borgmatic qui en facilite l'utilisation sont nos outils de sauvegarde préférés depuis longtemps. Pourquoi changer une équipe qui gagne ? Les sauvegardes sont surtout importantes pour Redis et les fichiers journaux mais il est à noter que si nous pouvons remonter n'importe quel service en peu de temps via l'automatisation, il est toujours bon d'avoir des sauvegardes, ne serait-ce que pour pouvoir comparer une configuration actuelle avec une plus ancienne version.

3.5 Et bien sûr, de l'automatisation

L'automatisation est omniprésente au sein de Framaspace : de l'infrastructure à la création des instances Nextcloud en passant par le choix de la suite bureautique.

Nous nous appuyons sur [Salt](#), logiciel que nous utilisons au sein de Framasoft depuis plus de 10 ans.

3.6 Enfin, des notifications

Nos différents outils envoient des notifications à diverses occasions (demande d'espace, mauvais fonctionnement, etc.).

Nous utilisons généralement le mail pour de telles notifications mais encore faut-il que sa boîte mail soit relevée fréquemment et la regarder. C'est pourquoi nous utilisons en plus un canal dont le but est de fournir des notifications en temps réel : [Gotify](#).

4 Les logiciels que nous avons développés

Entre l'interface web pour que les associations puissent demander une instance Nextcloud et la création automatique de celle-ci, en passant par l'automatisation de l'installation et de la configuration de l'infrastructure, nous avons développé un certain nombre de logiciels et de recettes :

- **Charon**, qui est l'interface web au projet, écrit en PHP.
C'est sur Charon que s'effectuent les demandes d'espaces collaboratifs et leur gestion. Une interface d'administration nous permet d'accepter ou de rejeter les demandes, ainsi que d'effectuer des opérations sur les espaces existants ;
- **Hermès** est un démon en Python qui interroge régulièrement Charon sur son API afin de connaître les tâches à exécuter : création ou suppression d'espace ou modification de suite bureautique.
Hermès utilise Salt pour effectuer les tâches envoyées par Charon ;
- **Poséidon** contient toutes nos recettes Salt (Poséidon, roi des mers, l'eau salée, Salt... vous l'avez ?). Ces recettes servent à la configuration des serveurs, la gestion des espaces et à la mise à jour de Nextcloud ;
- **Chronos** est un démon, encore une fois en Python, qui se charge d'exécuter régulièrement les tâches cron des espaces Nextcloud.
Nous ne pouvions pas mettre des centaines de tâches cron en parallèle, c'est pourquoi Chronos fut développé.
Il utilise pour cela une base de données PostgreSQL pour gérer la file d'attente et exécute les tâches cron en parallèle par paquets de 10 (par défaut) ;
- **Argos Panoptès** est un logiciel de supervision de sites web. Notre Shinken ayant déjà du mal à superviser nos ± 200 sites suffisamment rapidement (le délai de 5 minutes prévu entre chaque vérification pouvait être largement supérieur dans les faits, ce qui s'explique aisément par le nombre de vérifications par site — au minimum 3 — et le nombre important des autres sondes — systèmes, applicatives, etc.), nous n'envisagions pas de pouvoir l'utiliser pour les 10 000 instances Nextcloud prévues à terme. Suivant la principe « un outil qui fait une seule chose mais qui le fait bien », nous avons cherché un outil de supervision dédié aux sites web, mais aucun ne tenait la charge de nos 200 sites. C'est pourquoi nous avons développé notre propre solution, en faisant appel à un prestataire pour démarrer le projet, que nous avons mené à terme et continuons à améliorer.

Nous avons aussi des projets pour les [patches Nextcloud](#) à déployer (nous reportons les patches au projet Nextcloud mais nous n'attendons pas leur intégration pour en bénéficier), les [fichiers de ressources](#) (fonds d'écran, fichiers CSS, modèles, etc), la [page d'accueil](#) de Framaspace et enfin [une recette de conteneur Docker](#) pour Collabora.

Tous nos développements sont disponibles sur [la page Framagit du projet](#).

5 Problèmes rencontrés

5.1 Services de suite bureautique

Nous avons décidé initialement de dédier un service de suite bureautique à chaque instance Nextcloud mais la consommation de ressources d'un tel déploiement était trop élevée pour la machine virtuelle contenant les suites bureautiques, sans parler du pic de charge crevant le plafond au démarrage de la machine.

Nous avons contourné le problème en mutualisant des conteneurs Collabora entre plusieurs instances Nextcloud (les conteneurs OnlyOffice sont encore dédiés à une instance, de par leurs limitations d'usage).

Collabora et OnlyOffice ont chacun leurs avantages et leurs inconvénients :

- Collabora sera meilleur qu'OnlyOffice sur des documents au format OpenDocument ;
- A contrario, OnlyOffice saura mieux se débrouiller avec des documents au format Office Open XML, ce qui peut être un avantage si on utilise plutôt ce format ;
- Le conteneur OnlyOffice est plus gourmand en ressources système que celui de Collabora ;
- Il est possible de faire de l'over-provisioning avec Collabora en utilisant des conteneurs recompilés avec des limites d'utilisation simultanées plus grandes que celles par défaut (pour le conteneur Collabora Online Development Edition). Ces limites concernent le nombre maximum de documents ouverts et de connexions simultanées.

5.2 Gestion des versions sur MinIO

La réplication des données vers le deuxième cluster implique l'activation du versionnement. Nous avons voulu profiter de cela pour la gestion des versions des fichiers dans Nextcloud.

Problème : cela versionnait les chunks des fichiers téléversés de grande taille, empêchant leur suppression et saturant vite le quota du bucket S3.

Nous avons rebasculé sur la gestion native de Nextcloud des versions des fichiers et avons configuré un délai rapide de suppression réelle des fichiers supprimés sur MinIO (la suppression d'un fichier dans MinIO n'étant en fait que l'addition d'un marqueur de suppression au fichier).

5.3 Absence occasionnelle de transactions SQL dans Nextcloud

Certaines opérations SQL dans Nextcloud font une écriture puis une lecture immédiatement après... sans utiliser de transaction, ce qui posait souci car nous avons initialement choisi une réplication asynchrone pour PostgreSQL.

Sans transaction explicite, Pgpool-II n'est pas capable d'orienter forcément la lecture suivante vers le serveur primaire, ce qui aboutissait parfois à des échecs de l'opération.

Nous avons ajouté (et remonté) des transactions dans Nextcloud pour tenter de régler le problème mais la tâche est immense et notre temps disponible limité.

Nous avons contourné le souci en passant la réplication PostgreSQL en synchrone, au prix d'un petit délai supplémentaire dans les écritures.

6 Nos recettes Salt au centre de Framaspace

Le rôle des recettes Salt est essentiel au fonctionnement de Framaspace car elles permettent un grand nombre de choses :

- configurer les machines pour les intégrer à l'infrastructure ;
- mettre à jour la configuration des éléments de l'infrastructure ;
- déployer les espaces Nextcloud (base de données, bucket S3, bootstrap de l'espace, ajout dans Chronos) ;
- changer de suite bureautique ;
- mettre à jour Nextcloud et ses applications ;

- mettre à jour les espaces.

7 Une ferme de Nextcloud

7.1 Une seule base de code Nextcloud

Toutes les instances partagent la même base de code de Nextcloud. Seuls diffèrent leurs fichiers de configuration.

La raison est fort simple : Nextcloud, avec juste les applications recommandées, pèse 519Mio. Il était impensable de devoir utiliser près de 5Tio de disque quand nous atteindrions les 10 000 instances !

Le fichier de configuration de Nextcloud (config/config.php) fait une inclusion selon le nom d'hôte de la requête ou la variable NC_INSTANCE (pour la CLI).

Nginx utilise comme racine du site web un dossier au nom de l'hôte de la requête, dossier qui est un lien symbolique vers la base de code Nextcloud.

7.2 Les mises à jour

Les mises à jour de Nextcloud se passent ainsi :

1. téléchargement de la nouvelle version de Nextcloud (nous contrôlons la version téléchargée ainsi que les versions des applications mises à jour, nous ne nous contentons pas de prendre la dernière version disponible) ;
2. copie de cette version dans deux dossiers :
 - un dossier qui sera la cible des liens symboliques des dossiers de Nginx (fonctionnement de production) ;
 - un dossier qui sera utilisé de façon transitoire lors des mises à jour des espaces (voir plus bas).
3. mise à jour d'un espace de test :
 1. passage en mode maintenance ;
 2. configuration complète de l'instance dans une base de code qui ne sert qu'aux mises à jour (le dossier transitoire évoqué plus haut) ;
 3. php occ upgrade et éventuellement d'autres commandes telles l'ajout d'index de bdd ;
 4. modification du lien symbolique du dossier utilisé par Nginx pour utiliser la base de code de la nouvelle version ;
 5. sortie du mode maintenance.
4. lorsqu'on est sûr qu'il n'y a pas de problème, on met à jour un deuxième espace de test, avant de mettre à jour tous les espaces.

L'ensemble de ces opérations est lancé via Salt. La mise à jour d'un espace vers une nouvelle version mineure prend généralement 50 secondes et plus ou moins 2 minutes pour une nouvelle version majeure.

8 Give me some (data)love !

Au 16 septembre 2024, voici quelques chiffres concernant Framaspace :

- 15 serveurs physiques (voir tableau 1, page 12), dont 4 servant à la virtualisation ;
- 14 machines virtuelles (voir tableau 2, page 12), dont deux sur notre infrastructure habituelle (c-à-d pas sur les serveurs de virtualisation dédiés à Framaspace) ;
- 40Gio de quota pour chaque instance Nextcloud ;
- 224Tio de stockage provisionné ;
- 1 281 espaces déployés (sur 10 000 prévus) ;
- Les serveurs sont dans 2 centres de données différents :
 - 9 machines utilisent le même, à Falkenstein en Allemagne : les 4 serveurs de virtualisation et 5 serveurs MinIO ;
 - 5 machines sont à Helsinki en Finlande : les 5 serveurs de réplication MinIO ;
 - NB : les serveurs de virtualisation de notre infrastructure habituelle sont encore dans deux autres centres de données distincts, à Falkenstein en Allemagne.

9 Et... ça marche ou pas ?

9.1 Un succès « politique »

Un des principaux objectifs de Framaspace est de permettre aux associations et collectifs militants de retrouver de la cohérence entre leurs valeurs et les outils numériques qu’elles utilisent. En fournissant un tel service, qui plus est gratuitement, Framasoft souhaite redonner du « pouvoir d’agir » à ces organisations, mais aussi leur permettre de leur redonner de la confiance et de la dignité en montrant qu’elles peuvent s’appuyer sur une solidarité inter-associative plutôt que de dépendre de solutions capitalistes ayant un respect plus que discutable concernant la vie privée.

Sur les structures disposant d’un Framaspace : 71 % sont des associations loi 1901, 24 % sont des associations de fait, et 5 % des syndicats. Les secteurs de l’éducation/formation, de l’environnement, de la culture et du social représentent l’immense majorité des espaces. Plus de 70 % des espaces sont d’ailleurs utilisés par des structures sans salarié·e, et la médiane du budget déclaré n’excède pas 4 000 € annuel. Nous validons et déployons plus d’une centaine d’espaces par mois. Ces chiffres, ainsi que les objets sociaux des structures candidates, nous renforcent dans le sentiment d’atteindre les objectifs définis en amont du projet : outiller les structures qui changent le monde (ou lui évite de s’effondrer). D’autant que Framaspace en a encore « sous le capot » et pourra aisément accueillir le double ou le triple de structures, bien sûr en envisageant la location potentielle de quelques serveurs de plus, mais sans pour autant doubler ou tripler le coût. Politiquement, la prochaine étape importante sera de favoriser le dialogue et la coopération entre les structures hébergées, notamment en mettant à profit les fonctionnalités de fédération entre les espaces Nextcloud.

9.2 Un succès technique

Déployer près de 1 400 Nextcloud, et envisager entre 5 000 et 10 000 déploiements à terme, est un pari ambitieux, notamment pour une petite structure associative comme la nôtre.

Comparé à d’autres types d’infrastructures techniques déployant du Nextcloud en **multitenant**, celle de Framasoft permet notamment d’envisager une infogérance simplifiée. Ainsi, les mises-à-jour depuis la première version (Nextcloud 24) à la version actuelle (version 29) se passent de façon fluide et ne nécessitent qu’environ deux minutes de mise en maintenance par espace. Elle nous permet aussi de déployer facilement de nouvelles applications (externes, ou développées par nous)

en évitant largement les problèmes de régression ou de conflit entre applications que rencontrent nombre d'autres hébergeurs.

Cette architecture nous permet aussi de bien maîtriser nos coûts, car notre association ne vit que des dons. Ces derniers s'élèvent, grossièrement, à 30 000 € en 2022 (essentiellement en temps de travail salarié, et un peu en infrastructure puisque le service a été annoncé fin 2022), et environ 30 000 € par an en 2023 et 2024, réparti là encore grossièrement en 15 000 € d'infrastructure technique et 15 000 € de temps de travail (salarié, ou prestataire). Pour 1 280 espaces, cela représente un coût d'environ 2 € mensuel par instance, ce qui est déjà extrêmement faible. Mais si on se projette à 5 000 instances et un coût de 45 000 € par an, nous pourrions alors atteindre un coût de 0,75 €/mois (9 € / an) par instance déployée.

Le seul choix technique que nous questionnons aujourd'hui est celui de permettre aux administrateurs d'instances de choisir leur suite bureautique. En effet, les conteneurs OnlyOffice étant dédiés par espace (une instance = un conteneur OnlyOffice), cela peut entraîner à terme un coût d'infrastructure important si une forte proportion d'administrateurs choisissaient cette solution plutôt que Collabora (proposé par défaut, heureusement).

Pour le reste, nous sommes très satisfaits de nos choix techniques, qui font que la maintenance et le support de l'ensemble de l'infrastructure sont tout à fait raisonnables malgré nos moyens humains très limités puisque l'équipe opérationnelle est principalement composée de trois personnes.

Annexe

Tableau 1: Précisions sur les serveurs physiques

Machines physiques	CPU (cœurs/threads)	RAM (Gio)	Stockage
3 serveurs de virtualisation « rapides »	AMD Ryzen 5 3600 (6c/12t)	64	2x512Gio NVMe
1 serveur de virtualisation à grosse capacité de stockage	AMD Ryzen 5 3600 (6c/12t)	64	2x2Tio
Serveurs MinIO	AMD Ryzen 5 3600 (6c/12t)	64	4x16Tio
Nginx-2	AMD Ryzen 5 3600 (6c/12t)	64	2x512Gio NVMe

Tableau 2: Précisions sur les machines virtuelles

Machines virtuelles	VCPUs	RAM (Gio)	Serveur de virtualisation rapide
Haproxy-1	3	4	Oui
Nginx-1	6	16	Oui
Postgresql-1	7	16	Oui
Postgresql-2	3	16	Oui
Imaginary-1	2	4	Oui
Office-1	6	30	Oui
Office-2	4	30	Non
Syslog-1	4	10	Non
Redis 1 à 3	2	4	Oui
Redis 4 à 6	2	4	Non