

# Sysadmin

The hitchhiker's guide to the gconfs infrastructure

- [Infrastructure](#)
- [Deprecated](#)

# Infrastructure

Gconfs gère des outils pour ses membres. A partir de ce moment, se pose la question de l'administration des systèmes et des outils en question.

## Nom de domaine

## Auto-hébergement

L'hébergement a commencé sur une VM Kimsufi chez OVH. Quelques années après le CRI nous a proposé un hébergement gratuit chez eux que nous avons accepté avec plaisir. Malheureusement l'uptime était pas terrible en partie à cause du Bocal et le CRI nous viré comme des malpropres deux ans plus tard (ah les promesses, ça n'engage que ceux qui y croient...). On a migré dans la douleur sur Vultr et depuis c'est l'amour fou.

## Site web

Voir le [repo GitHub](#).

## Wiki

Le Wiki utilise Bookstack. Tout se fait via l'interface en ligne via votre compte Slack GConfs.

## Vault

### Comment ça marche?

Le [vault](#) sert à conserver les mots de passe des différents comptes utilisés par Gconfs, pour qu'ils soient partagés entre les sysadmins et le bureau et transmis facilement. Le tout est géré par [passbolt](#). Les mots de passe sont stockés chiffrés avec GPG.

## Créer un compte

Pour avoir un compte, il faut demander a quelqu'un qui a la main sur le serveur (pour l'instant). Il lancera alors la commande magique (`sudo -H -u www-data /srv/vault/bin/cake passbolt register_user -u [email] -f [prénom] -l [nom] -r [role(user ou admin)]`) et ça lui donnera une URL, qu'il pourra transmettre a la personne qui veut créer le compte pour qu'elle l'active.

## Comment je l'utilise

Passbolt fonctionne avec une extension navigateur et des applis android/iOS/whatever. Il *faut* avoir l'extension pour consulter les mots de passe et pour créer son compte. Lors de l'activation du compte, une clé GPG est créée (en js dans le browser), et sera utilisée pour déchiffrer les mots de passe présents sur le vault. Ça filera aussi un ensemble (couleur + trois lettres) qui est utilisé pour identifier passbolt quand il demande un mot de passe. Après avoir sauvegardé la clé GPG, la passphrase et cet ensemble (couleur + lettres), bah il suffit d'aller sur [vault](#) et a commencer a stocker/partager des mots de passe.

## Pad

Le [pad](#) sert a, euh, écrire des trucs de manière collaborative. C'est géré par [CodiMD](#).

## Créer un compte

On va sur la page d'accueil, on clique sur log in, on rentre ses infos, et on clique sur register. Done.

## Comment je l'utilise

L'interface de base comprend un panel a gauche éditable en markdown, et un panel a droite qui affiche le résultat HTML. Les couleurs correspondantes aux différentes personnes éditant le document sont tout a gauche du panel d'édition, et euh voilà quoi ☹️

# Les trucs externes

## Slack

Avant, on avait un channel IRC sur rézosup. Depuis quelques années, Gconfs est passé a [Slack](#). Puis maintenant à Discord. Bah c'est Discord quoi. Y'a plus que les vieux cons sur Slack.

## Google

Les comptes email @gconfs.fr sont gérés par gmail, tout comme les mailing lists @gconfs.fr. C'est un compte gsuite associatif, mais on utilise vraiment que Gmail et google drive pour stocker des fichiers partagés.

# Deprecated

Cette page regroupe toute les informations dépréciées concernant la gestion de l'infrastructure de GConfs.

## Autohébergement - Serveur CRI

Une VM est prêtée a Gconfs par le CRI. C'est un petit truc avec 2 CPU et 1Go de RAM, mais c'est déjà pas mal pour ce qu'on veut en faire. Dessus on host a peu près tout ce qui est pas trop chiant a hoster, en configurant le tout avec [Ansible](#)

## Site web Hyde

Pour que le site web ne soit jamais trop en retard, il faut rajouter les nouvelles conférences sur le site. Pour cela, clonez le dépôt git contenant le site (<https://github.com/gconfs/website-hyde>), allez dans `content/confs`, puis créez un nouveau dossier `yyyy-mm-nom_de_la_conférence`. Dans ce dossier, créez un fichier `index.html` en vous inspirant des conférences existantes, puis rajouter un lien vers cette nouvelle conférence sur la page `content/confs/index.html`, dans les conférences a venir. Une fois la conférence terminée, n'oubliez pas de la bouger dans les conférences passées!

Pour mettre a jour le site web lui même, il est nécessaire de lancer le [playbook Ansible](#) `website`

---

## Site web Gconfs

### Comment ça marche?

Le [site de Gconfs](#) est fait avec [Hyde](#), le dépôt correspondant se trouve [ici](#). Y avait un projet de le remplacer par un site en django mais y a pas trop de nouvelles. Le déploiement du site sur le serveur est fait automatiquement avec le playbook ansible `website`, qui git clone et regénère le site automatiquement.

### Comment mettre a jour le contenu

Du coup, pour updater le site il faut modifier le dépôt, **vérifier que ça compile correctement**, push sur le dépôt github, puis puller depuis le serveur en utilisant le playbook (`sudo ansible-playbook`

/etc/ansible/playbooks/website.yml )

# Wiki Gconfs

## Comment ça marche?

Le [wiki de GConfs](#) utilise [gitit](#). Les données du wiki sont donc stockées dans git, et il faudrait setup un push vers github en parallèle pour avoir une version master. Il peut se déployer via le playbook wiki ( `sudo ansible-playbook /etc/ansible/playbooks/wiki.yml` ), mais ne clone pas (encore?) le contenu directement depuis github.

## Comment mettre a jour le contenu

Pour éditer le wiki, il est bien sur possible d'utiliser l'interface web, mais on peut aussi git clone le dépôt qui contient les données ( `git clone gconfs.fr:/srv/gitit/wikidata wiki_gconfs` ) (quand on a un compte sur le serveur, s'entend), éditer le texte, puis le push sur le serveur a nouveau, ce qui mettra a jour la version web dynamiquement. Ça s'écrit en markdown.