Système d'Information du RISC

Version 0 - 2019-07-12

Vincent Férotin

juil. 12, 2019

Table des matières

CHAPITRE 1

Notes introductives à la présente documentation

Notez que le format d'une telle documentation se voulait au départ sous la forme d'un *wiki* interne au *RISC*, et donc éditable de manière collaborative. Faute de temps (pour installer et configurer un *annuaire LDAP*, permettant du *SSO* vers, entre autres, le *wiki*)…

La présente documentation est, en l'état, encore très parcellaire, et souffre d'un manque flagrant de plusieurs sections critiques. Elle est à considérer, de manière générale, comme un point de départ : peu de sections sont « finalisées ».

Egalement, les tâches et activités d'un(e) administrateur(trice) système ne se limitent aucunement à ce qui est uniquement présenté ici. Pour plagier (et augmenter) le *Unix and Linux system administration handbook*, elles sont notamment de :

- s'assurer et contrôler les accès aux ressources ;
- gérer la maintenance et l'évolution du matériel (*hardware*) ;
- se faciliter au maximum le travail, en automatisant autant que possible les tâches ;
- effectuer et surveiller les sauvegardes ;
- installer, configurer, dimensionner, mettre à jour, changer, etc. les logiciels ;
- monitorer l'ensemble du S.I. ;
- réagir en cas d'imprévu, d'accident, de dysfonctionnement, ce qui implique l'analyse du problème et la mise en place du nécessaire pour sa résolution, tant locale que pérenne ;
- faire évoluer la documentation du S.I. ;
- être pro-actif sur les aspects liés à la sécurité du S.I. ;
- améliorer les performances quand c'est possible ;
- élaborer et partager des politiques de sites ;
- interagir avec les partenaires, tant institutionnels que tiers ;
- aider à la résolution des menus problèmes courants (« fire fighting ») ;
- savoir travailler en équipe, tant au sein des informaticiens qu'avec les non-informaticiens ;
- etc.

CHAPITRE 2

Configurations des serveurs

2.1 Configuration du serveur principal

2.1.1 Historique

2019-01-30

- *chrony* : ajout des configurations suivantes :

```
server 127.127.1.0
bindcmdaddress 127.0.0.1
bindcmdaddress  : :1
noclientlog
logchange 0.5
```

2019-01-29

- mise à jour de la conf. de *chrony*
- ouverture dans le pare-feu du port `123` pour *UDP* pour la zone *public* (pour espérer se connecter aux serveurs *NTP* autres que celui du *SPI*)

CHAPITRE 3

Développements réalisés pour l'administration du SI

3.1 sysutils : utilitaires systèmes

Le « projet » *sysutils* se veut une ombrelle pour regrouper tous les petits développements réalisés pour faciliter l'administation du SI du *RISC*. Il est essentiellement composé de « simples » *scripts shells*.

3.1.1 *upgrade.sh* : mise à jour de l'*O.S.* (CentOS)

projet sur la forge GitLab du CRI https ://forge.cri.ens.fr/risc/sysutils-upgrade

Son utilisation est triviale (pourvu que le chemin du script soit dans le *$PATH*) :

```
$ sudo upgrade.sh
```

3.1.2 *remove_old_kernels.sh* : suppression des kernels trop vieux (CentOS)

projet sur la forge GitLab du CRI https ://forge.cri.ens.fr/risc/sysutils-remove_old_kernels

Son utilisation est triviale (pourvu que le chemin du script soit dans le *$PATH*) :

```
$ sudo remove_old_kernels.sh
```

3.2 fail2ban-utils : utilitaires compléments de *fail2ban*

Le projet *fail2ban-utils* se voulait proposer des outils en ligne de commande suppléant ceux fournis par le projet *fail2ban*, afin de faciliter l'administration d'une installation, en vue notamment de pouvoir facilement réaliser des statistiques sur les *bannis*, afin de les bannir « manuellement » définitivement. Son développement a été arrêté alors qu'à ses tous débuts, faute de temps et vu que ce n'était pas une priorité pour l'unité par rapport à la remise en ligne « au plus tôt » de ses services web.

CHAPITRE 4

Base de connaissances en informatique

4.1 OpenSSH : implémentation libre et répandue du protocole *SSH*

OpenSSH est un logiciel prenant en charge le protocole *SSH*, tant du côté client que du côté serveur. C'est très probablement l'implémentation la plus répandue de ce protocole dans le monde *Linux/*nix*.

Il est utilisé au *RISC* pour par ex. accéder de manière sécurisée et en ligne de commande aux serveurs *serveur principal* et *serveur secondaire*, mais également sur les espaces d'hébergement gérés par *Prestataire*.

4.1.1 Recommandations pour le choix des bi-clefs (aysmétriques) publique/privée

L'utilitaire *ssh-keygen* permet de générer les couples de clefs cryptographiques, utilisés un peu partout. Les formats de ces clefs sont fonction de l'algorithme de chiffrement choisi ; actuellement on note :

- *DSA*,
- *RSA*,
- *ECDSA*,
- *ED25519*.

Le choix de l'algorithme dépend de plusieurs facteurs, essentiellement :

- sa robustesse espérée,
- sa bonne prise en charge par tous les systèmes communiquant.

On exclut d'office *DSA*, unanimement déconsidéré (notamment par l'*ANSSI*). On préférera également *ED25519* à *ECDSA*, qui corrige ses problèmes de génération de nombres aléatoires.

Si les systèmes l'acceptent, on privilégiera a priori le plus récent *ED25519* à *RSA* (rec. *ANSSI*), qui sera utilisé plutôt pour compatibilité. Dans ce dernier cas, la taille des clefs *RSA* devra être d'*au moins* **2048 bits** (rec. *ANSSI*, jusqu'en 2030). Pour *ED25519*, la taille des clefs devra être d'*au moins* **256 bits**.

À faire : OpenSSH 8.0 recommande une longueur de 3072 bits pour *RSA*

Voir l'entrée dans son changelog : https ://www.openssh.com/txt/release-8.0

> « Increase the default RSA key size to 3072 bits, following NIST Special Publication 800-57's guidance for a 128-bit equivalent symmetric security level. »

Voir aussi :

- page dédiée sur le wiki d'Archlinux
- *Référentiel Général de Sécurité* de l'*ANSSI* :
  - Annexe B1 : Mécanismes cryptographiques – Règles et recommandations concernant le choix et le dimensionnement des mécanismes cryptographiques
  - Note technique : Recommandations pour un usage sécurisé d'(Open)SSH
- Question posée sur StackExchange : SSH key-type, rsa, dsa, ecdsa, are there easy answers for which to choose when ?

4.1.2 ssh-agent : utilitaire pour garder en mémoire un jeu spécifique de clefs

Fourni avec *OpenSSH*, l'utilitaire *ssh-agent* permet de conserver en mémoire, durant un temps déterminé, un jeu spécifique de clefs et leurs *passphrases* éventuelles. Il est souvent utilisé conjointement à *ssh-add*.

4.2 chrony : synchronisation de temps via NTP

Chrony est un logiciel faisant office de *démon client NTP*. C'est le logiciel installé par défaut sous *RHEL* / *CentOS* pour cette tâche.

Il est utilisé au *RISC* pour synchroniser l'heure des serveurs *serveur principal* et *serveur secondaire*, via notamment le `ntp.ens.fr` du *SPI*.

L'intérêt d'un tel logiciel, et donc d'avoir un *temps système* fiable, se présente entre autres quand on examine les *logs* : cela permet d'avoir une cohérence dans les heures. Sans un tel logiciel, le sytème dériverait petit à petit, et son *temps système* finirait par ne plus avoir de rapport avec l'extérieur. En plus d'être une mesure recommandée, notamment pour la sécurité, c'est aussi souvent un prérequis à l'utilisation d'autres logiciels.

Le *démon chrony* peut être interrogé par l'utilitaire en ligne de commande *chronyc*.

4.2.1 Ressources

Site web du projet https ://chrony.tuxfamily.org/

Documentation pour RHEL 7 https ://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/system_administrators_guide/ch-configuring_ntp_using_the_chrony_suite

4.3 sudo et sudoers : escalade temporaire et maîtrisée de privilèges utilisateurs pour l'exécution de commandes

sudo est un utilitaire permettant, si un utilisateur y est autorisé, d'exécuter une commande en escaladant temporairement ses privilèges à ceux de `root`. Il est donc utile dans une politique de sécurité, pour déléguer l'appel de commandes bien particulières à des utilisateurs clairement identifiés. Sa configuration peut être très fine ; elle se réalise dans le fichier `/etc/sudoers`.

4.3.1 Edition de la configuration dans `/etc/sudoers`

Comme il y a une vérification additionnelle de bonne conformité à la syntaxe de la configuration – afin de ne pas se retrouver bloqué dans ses actions, par faute de mauvaise configuration –, l'édition ne se fait pas « naïvement » avec l'éditeur de son choix ; on passe par la commande suivante :

```
$ sudo visudo
```

4.3.2 Log des commandes

Les commandes exécutées via *sudo* sont par défaut enregistrées dans `/var/log/secure`.

4.4 etckeeper : versionnement de la configuration d'une machine

etckeeper est une surcouche légère à des logiciels de gestion de version, et *git* en permier lieu, pour versionner une configuration d'une machine de type *nix (Linux, BSD), spécifiquement son répertoire `/etc`.

Il est utilisé au *RISC* pour par ex. sauvegarder et versionner les configurations des machines serveurs (*serveur principal* et *serveur secondaire*).

4.4.1 Ressources

Site web du projet https ://etckeeper.branchable.com/

4.5 git : logiciel de gestion de versions

git est un des logiciels de gestion de versions majeurs courants. Il est libre et gratuit, et s'utilise principalement en ligne de commande.

Il est utilisé au *RISC* pour par ex. le versionnement des développements des projets.

4.5.1 Ressources

Site web du projet https ://git-scm.com/

Bibliographie

- Mémento - Git à 100% de Raphaël Hertzog, Pierre Habouzit (*Eyrolles*, 2018, 2nde éd.)
- Pragmatic Guide to Git de Travis Swicegood (*The Pragmatic Bookshelf*, 2010)

4.5.2 Usage

Compactage d'un dépôt *git* (optionnel)

L'opération de compactage peut être intéressante pour diminuer la taille d'un dépôt :

```
$ cd /chemin/vers/mon/depot
# Afficher les tailles du dépôt
$ du -c -d 1 -h .git
# Réaliser le compactage
$ git gc
# Vérifier la différence des tailles du dépôt
$ du -c -d 1 -h .git
```

4.6 gitolite : gestion fine de droits sur des dépôts git

gitolite est un logiciel pour gérer de manière très fine les accès utilisateurs à un ensemble de dépôts *git*, à travers un dépôt *git*.

Il est utilisé au *RISC* pour par ex. gérer l'accès aux dépôts *git* gérés par *etckeeper*.

4.6.1 Ressources

Ressources en ligne

- Site web du projet : http ://gitolite.com/gitolite/
- CookBook : http ://gitolite.com/gitolite/cookbook

Bibliographie Gitolite Essentials de Sitaram Chamarty (*Packt Pub.*, 2014)

4.6.2 Fonctionnement général

La particularité de *gitolite* consiste à gérer l'administration des utilisateurs et leurs accès aux dépôts *git* à travers un dépôt *git*, dit d'administration (nommé `gitolite-admin`). Egalement, le logiciel va centraliser tous les accès des « utilisateurs », et permettre ainsi de ne pas créer autant d'utilisateurs système que d'utilisateurs des dépôts *git*.

L'authentification (*authentication*) est déléguée à des logiciels tiers : le serveur *SSH* ou le serveur *web* – c'est le premier que nous utiliserons préférentiellement au *RISC*. L'identification des utilisateurs de *gitolite* se fera alors par clef publique uniquement, et tous se connecteront au même utilisateur système, dit « utilisateur hôte » (*hosting user*) et traditionnellement nommé `git`, `gitolite` ou `gitolite3` (suivant le packaging de la distribution).

La configuration de *gitolite* s'effectue, en plus du matériel du dépôt `gitolite-admin`, dans un fichier « rc » (`$HOME/.gitolite.rc`), pour activer ou désactiver des fonctionnalités notamment.

4.6.3 Limitations

Les *dépôts git* gérés par *gitolite* ne peuvent se situer que dans le répertoire `$HOME/repositories` du *host-user* de *gitolite*[1]. On ne pourra donc pas, par exemple, gérer de manière directe le dépôt `/etc/.git` administré par *etckeeper*.

4.7 ReaR : logiciel simple de sauvegarde d'un système complet

ReaR (*Relax-and-Recover*) est un logiciel très simple pour sauvegarder un système d'epxloitation complet.

1. Il semble possible, après l'initialisation de *gitolite*, de déplacer le répertoire `$HOME/repositories` et ensuite le sym-linker depuis ce même emplacement vers le nouveau : http ://gitolite.com/gitolite/odds-and-ends.html#putting-repositories-and-gitolite-elsewhere

Il est utilisé au *RISC* pour avoir une sauvegarde facilement restaurable des serveurs *serveur principal* et *serveur secondaire*. Nous l'avons introduit lors de la montée de version mineure de *CentOS* 7.5 à 7.6 pour *serveur principal* (découvert dans les release notes de *RHEL* 7.6).

Un de ses avantages consiste à ne pas avoir besoin de redémarrer la machine sauvegardée.

4.7.1 Ressources

Site web du projet http ://relax-and-recover.org/

Knowledgebase *RedHat* What is Relax and Recover(ReaR) and how can I use it for disaster recovery ?

4.7.2 Usage

Le logiciel se configure à travers essentiellement le seul fichier `/etc/rear/local.conf`. On lance ensuite une ou plusieurs commandes pour préparer le périphérique auxiliaire, créer le support bootable, et/ou effectuer la sauvegarde proprement dite des données.

4.8 Vérification de bonne santé des disques et partitions : smartmontools et fsck

Les disques durs d'une machine sont un de ses éléments cruciaux, dont il est nécessaire de s'assurer périodiquement de la bonne santé – sous risque de perdre des données et avoir des services inutilisables. On pourra utiliser *smartmontools* pour vérifier l'état des données *S.M.A.R.T.* des disques, et *fsck* pour vérifier la bonne intégrité des données de leurs partitions.

4.8.1 smartmontools : vérification des données *S.M.A.R.T.* de disque

smartmontools est une suite logicielle, présente sur la majorité des *distributions Linux*, permettant de relever et effectuer des tests plus ou moins approfondis sur les données *S.M.A.R.T.* d'un disque dur. Elle s'utilise principalement via son utilitaire en ligne de commande : `smartctl`.

L'interface de *smartmontools*, `smartctl`, est très riche – comme le montre clairement sa page de manuel (`man smartctl`). Nous ne montrerons ici qu'un sous-ensemble très restreint pour débuter son usage.

4.8.2 fsck : vérification des données de partitions

fsck est un utilitaire en ligne de commande permettant de vérifier, de manière plus ou moins approfondie, l'état d'intégrité des données de partitions d'un disque dur. Nous utilisons essentiellement au *RISC* sa déclinaison pour les partitions au format *ext4* : `fsck.ext4`.

4.8.3 badblocks : recherche de blocs défectueux

badblocks est un utilitaire en ligne de commande permettant de rechercher sur un disque dur magnétique, comme son nom le laisse entendre, les blocs défectueux.

4.9 Apache httpd : serveur web (HTTP, HTTPS) courant

Apache httpd est le logiciel phare de la *Fondation Apache*, probablement le *serveur web* le plus connu – et fut un long temps le plus utilisé. Ceci explique qu'on le désigne aussi couramment simplement

par raccourci : « Apache ». Il est généralement exécuté comme *démon* (tournant en tâche de fond), pour répondre aux *requêtes* du protocole *HTTP* (et *HTTPS*) par des *réponses* (*HTTP*/*HTTPS*), généralement sur le port `80` (resp. `443`).

4.9.1 Nom du logiciel sous *CentOS* 7

Sous *CentOS* 7, le logiciel/paquet/service est appelé `httpd`. Son arborescence de logs sera aussi fonction de ce nom : `/var/log/httpd/`.

4.9.2 Versions des dépôts disponibles

Apache httpd est disponible par défaut sous *CentOS*, à une version plutôt ancienne (mais sur la branche `2.4.x` néanmoins), `2.4.8`, dont la sécurité est assurée en amont par *RHEL*. Mais il est également disponible via *IUS*, dans une version, pour respecter la philosophie du projet, proche de l'upstream (e.g. `2.4.38` à l'heure où j'écris ces lignes). La question se pose donc : quelle version préférer et installer ?

Je n'ai pas d'avis tranché ni définitif sur la question. Mais comme nous n'avons (pour le moment) pas *besoin* de fonctionnalités introduites entre ces deux versions, et que *RedHat* a une bonne réputation de stabilité et sécurité, je vois peu d'arguments en faveur d'"*IUS*. J'ai donc laissé pour l'instant la version par défaut de *CentOS*/*RHEL*.

CHAPITRE 5

HowTo.s informatiques

5.1 Mise à jour courante des paquets des serveurs

5.1.1 Prérequis

O.S. *CentOS*

script `upgrade.sh` disponible dans le *$PATH*

5.1.2 Recommandations générales

Vérifier **quotidiennement** (et même les jours chômés) si des mises à jour de *paquets* sont disponibles pour les machines (serveurs compris), et le cas échéant appliquer les mises à jour, fait partie des bonnes pratiques de sécurité notamment, et d'administration de manière plus générale. Ces mises à jour proposées par le projet fournissant l'*O.S.* sont le plus souvent ou des corrections de failles de sécurité connues, ou des corrections de bugs.

Note : On ne traite ici que des mises à jour courantes, en aucun cas de montée de version, même mineure, de l'*O.S.* (e.g. passage de *CentOS* `7.5` à *7.6*).

Le plus souvent, aucune mise à jour n'est proposée, et il n'y a alors rien à faire. Des fois, les mises à jour proposées ne nécessitent que leur simple application. Dans d'autres cas, il conviendra de procéder aux mises à jour en plusieurs temps. Enfin, certaines mises à jour peuvent nécessiter de redémarrer des services, voir la machine complète. Le choix de la procédure sera *in fine* de la prérogative de l'administrateur. Nous détaillons ci-dessous les principaux cas pouvant se présenter.

Pour faciliter les cas les plus fréquents, on pourra utiliser le script `upgrade.sh` du *projet sysutils* :

```
$ sudo upgrade.sh
```

Si des mises à jour de paquets ont lieu, on s'assurera qu'il n'y a pas de changement de configuration dans `/etc`, ou sinon *enregistrera (après review, et adaptation éventuelle) la nouvelle configuration via "etckeeper"*, et ensuite la sauvegardera sur une machine de travail.

5.1.3 Principaux cas de mises à jour

Mise à jour « standard »

Si des mises à jour de *paquets* sont proposées, qui ne sont pas :

- liés aux services principaux (web ou autres) du serveur ;
- liés au système de *paquets* (e.g. *rpm*, *yum*) ;
- liés à des composants bas-niveau comme le *noyau*, *systemd* ou *libc* (par ex.) ;

on procédera directement à leur application, en suivant le déroulé proposé par le *script "upgrade.sh"* :

```
$ sudo upgrade.sh
```

Mise à jour comprenant des paquets liés au système de paquets

Si, dans les mises à jour proposées lors de l'exécution du *script "upgrade.sh"*, des paquets sont présents, qui sont liés au *système de paquets de l'O.S.*, on procédera à leur mise à jour en priorité sur les autres paquets :

1. interruption du script *upgrade.sh* par `Ctrl+C` ;
2. mise à jour des paquets liés au système de paquets par couche, en commençant par les plus « basses » (*RPM*), et terminant par les plus « hautes » (*yum*), e.g. (adaptez au besoin) :

   ```
   $ sudo yum update rpm
   $ sudo yum update yum
   ```

3. une fois ces mises à jour appliquées, on reprendra le processus normal, et agira en fonction des mises à jour proposées :

   ```
   $ sudo upgrade.sh
   ```

Mise à jour de paquets liés aux services courants de la machine

Si des mises à jour proposées concernent un ou des services que la machine fournit, par ex. pour *serveur principal* : *httpd* (serveur web), *php-fpm* (*PHP*), *mariadb* (bases de données), etc., on redémarrera les services concernés après mise à jour des paquets :

1. interruption du script *upgrade.sh* par `Ctrl+C` ;
2. mise à jour des paquets concernés (adaptez au besoin) :

   ```
   $ sudo yum update php72u-fpm-httpd
   $ sudo yum update mariadb101u
   ```

3. redémarrage des services, dans l'ordre suivant : *fail2ban*, *sshd*, *mariadb*, *php-fpm*, *httpd*, e.g. (adaptez au besoin) :

   ```
   $ sudo systemctl restart mariadb
   $ sudo systemctl restart php-fpm
   ```

4. une fois ces mises à jour appliquées, on reprendra le processus normal, et agira en fonction des mises à jour proposées :

   ```
   $ sudo upgrade.sh
   ```

Mises à jour de composants « bas niveau » de l'*O.S.*

Si des mises à jour proposées concernent des composants « bas niveau », comme le *noyau Linux* (*kernel*), le gestionnaire de processus (*systemd*), la *bibliothèque C* (*libc*), etc. un redémarrage sera nécessaire pour que les mises à jour soient prises en compte. Suivant le jugement de l'administrateur, on procédera par précaution à (ou profitera de l'occasion pour faire ;-) une sauvegarde complète du système, par ex. avec *ReaR* :

1. interruption du script *upgrade.sh* par `Ctrl+C` ;
2. au besoin, *sauvegarde complète du système par "ReaR"* ;
3. reprise des mises à jour à appliquer :

```
$ sudo upgrade.sh
```

4. redémarrage du système :

```
$ sudo shutdown -r now
```

5. patienter le temps que la machine redémarre complètement ;
6. une fois *up*, s'assurer que tout semble bien fonctionner ;
7. au besoin, *nouvelle sauvegarde complète du système par "ReaR"*.

5.2 OpenSSH : configuration et usages

5.2.1 Prérequis

O.S. *CentOS* 7

5.2.2 Génération d'un couple de clefs asymétriques publique/privée avec *ssh-keygen*

Voir aussi :

section dédiée à OpenSSH dans la base de connaissance pour les choix des algorithmes et longueurs respectives des clefs.

Avertissement : Sauvegarde des clefs dans un coffre-fort numérique !

A moins que le *$HOME* des utilisateurs ne soit chiffré, on prendra soin d'enregistrer les bi-clefs sur un volume chiffré, par ex. avec *VeraCrypt*, via l'option `-f` de *ssh-keygen*. On les sélectionnera ensuite *en utilisant "ssh-add" pour "ssh-agent"*.

RSA

```
# RSA keys of length 2048 bits with no passphrase
$ ssh-keygen -t rsa -b 2048 -N "" [-f /media/veracrypt1/RISC/secrets/ssh/id_rsa]
Generating public/private rsa key pair.
Enter file in which to save the key (/home/vferotin/.ssh/id_rsa) :
Created directory '/home/vferotin/.ssh'.
Enter passphrase (empty for no passphrase) :
Enter same passphrase again :
Your identification has been saved in /home/vferotin/.ssh/id_rsa.
Your public key has been saved in /home/vferotin/.ssh/id_rsa.pub.
```

(suite sur la page suivante)

(suite de la page précédente)

```
The key fingerprint is :
SHA256 :OPpBtmulguWrek5TgC0vXkT98eFBPSR2Le5UrI8KoGA vferotin@etckeeper
The key's randomart image is :
+---[RSA 2048]----+
|   ..   .+ooo    |
| o.  . ..oo+ +   |
|o o.  . + + =    |
| E.. . o o +     |
|o o.o * S o o    |
|...o.+ +.  o .   |
| .o+. oo. .      |
| .o.o.oo .       |
|.+o..+o          |
+----[SHA256]-----+
```

ED25519

```
# ED25519 keys of length 256 bits with no passphrase
$ ssh-keygen -t ed25519 -b 256 -N "" [-f /media/veracrypt1/RISC/secrets/ssh/id_
↪ed25519]
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/vferotin/.ssh/id_ed25519) :
Your identification has been saved in /home/vferotin/.ssh/id_ed25519.
Your public key has been saved in /home/vferotin/.ssh/id_ed25519.pub.
The key fingerprint is :
SHA256 :ccVG1efDKH+KHG7C4pd7sqp//gxHBbnmn3O9F5Xpeh4 vferotin@etckeeper
The key's randomart image is :
+--[ED25519 256]--+
|           ++... |
|           o+   o|
|        . ....o.+|
|         o +.. =o|
|        S o.o . o|
|          .o . + |
|        ..+.+ =Eo|
|       . B+* * o+|
|     .+++*Oo  =oo|
+----[SHA256]-----+
```

5.2.3 Sécurisation de la configuration du serveur *sshd*

Note : Visualiser les valeurs de configuration courante :

```
$ sudo sshd -T
port 22
addressfamily any
listenaddress [ : :] :22
listenaddress 0.0.0.0 :22
usepam yes
...
```

La configuration se fait dans `/etc/ssh/sshd_config`. Par rapport à la configuration par défaut proposée par *CentOS* :

- Interdire le login en tant que *root* :

```
PermitRootLogin no
```

- N'autoriser qu'une liste explicite d'utilisateurs pouvant se logger :

```
AllowUsers alice bob charles
```

- Autoriser l'authentification par clef publique :

```
PubkeyAuthentication yes
```

- Interdire l'utilisation de variables d'environnement pour l'utilisateur :

```
PermitUserEnvironment no
```

5.2.4 Afficher l'empreinte d'une clef publique

Avant qu'un utilisateur ne se connecte pour la première fois sur le nouveau serveur fraîchement installé, il est bon de lui communiquer les empreintes des différentes clefs publiques du serveur, puisque le client va demander explicitement à l'utilisateur d'en accepter une. Cela permettra à l'utilisateur d'être certain qu'il communique bien avec le serveur, et n'est pas victime d'une attaque de type *Man in the Middle*.

```
$ ssh-keygen -l -f /etc/ssh/ssh_host_rsa_key.pub
2048 SHA256 :OgHj67mSgQ3+Th2/8Ptb5ySfHS3bUzJ5NRIf9SI8Xy8 no comment (RSA)
$ ssh-keygen -l -f /etc/ssh/ssh_host_ecdsa_key.pub
256 SHA256 :TM/dP4sgyXDR2Yl33+eOwzXOXS20rzSQvUjm6FzMWgg no comment (ECDSA)
$ ssh-keygen -l -f /etc/ssh/ssh_host_ed25519_key.pub
256 SHA256 :1G468G2w4Frvcl99ZYZy7f04tPolAw9NuWjMON6ohuc no comment (ED25519)
```

5.2.5 Sélection temporaire d'un jeu de bi-clefs pour ssh-agent

```
# Démarrer une session avec ssh-agent dans un nouveau shell
$ ssh-agent bash
# Ajout d'une bi-clef spécifique, en dehors de votre $HOME, e.g.
$ ssh-add /media/veracrypt1/RISC/secrets/ssh/id_ed25519
Identity added : /media/veracrypt1/RISC/secrets/ssh/id_ed25519 (vferotin@serveur_
↪principal)
# ...do some stuff...
# Fermer la session ssh-agent
$ exit
```

5.3 chrony : installation, configuration et usage

5.3.1 Prérequis

O.S. *CentOS 7*

5.3.2 Installation proprement dite

Note : L'installation de *Chrony* a normalement déjà été réalisée dans le *kickstart*.

```
$ sudo yum install chrony
```

5.3.3 Configuration

La configuration de *Chrony* se fait via le fichier `/etc/chrony.conf`. Les principales options sont les suivantes :

- déclaration des serveurs de temps :

Code source 1: /etc/chrony.conf

```
# ENS
server ntp.ens.fr iburst
# CentOS
server 0.centos.pool.ntp.org iburst
server 1.centos.pool.ntp.org iburst
server 2.centos.pool.ntp.org iburst
server 3.centos.pool.ntp.org iburst
# NTP project (France)
server 0.fr.pool.ntp.org iburst
server 1.fr.pool.ntp.org iburst
server 2.fr.pool.ntp.org iburst
server 3.fr.pool.ntp.org iburst
```

À faire : S'assurer que les serveurs NTP de centos.pool.ntp.org sont bien pris en compte ! Pb. d'ouverture du firewall ? ?

- pour avoir un substitut (soi-même) en cas de problème de connexion aux serveurs ci-dessus[1] :

Code source 2: /etc/chrony.conf

```
# Synchronize on myself (127.127 = myself, 1 = local clock)
server 127.127.1.0
### Conf. suivante : pas valable pour Chrony 3.2 ??
## Stratum 10 (arbitrary)
#fudge 127.127.1.0 stratum 10
```

- ignorer la strate (*stratum*, ?) dans la sélection de sources :

Code source 3: /etc/chrony.conf

```
# Ignore stratum in source selection.
stratumweight 0
```

- enregistrement et mise à dispo. de la variation de temps (*drift*) :

1. Voir par ex. https ://wiki.debian.org/fr/NTP#Astuces

Code source 4: /etc/chrony.conf

```
# Record the rate at which the system clock gains/losses time.
driftfile /var/lib/chrony/drift
```

- ne répondre à des commandes que depuis l'hôte :

Code source 5: /etc/chrony.conf

```
# Listen for commands only on localhost.
bindcmdaddress 127.0.0.1
bindcmdaddress  : :1
```

- spécification des options de *log* :

Code source 6: /etc/chrony.conf

```
# Disable logging of client accesses.
noclientlog

# Specify directory for log files.
logdir /var/log/chrony

# Select which information is logged.
# (disabled by default)
log measurements statistics tracking

# Send a message to syslog if a clock adjustment is larger than 0.5 seconds.
logchange 0.5
```

et création des fichiers de logs correspondant avec les bons droits :

```
$ sudo touch /var/log/chrony/measurements.log
$ sudo touch /var/log/chrony/tracking.log
$ sudo touch /var/log/chrony/statistics.log
$ sudo chown -R  :chrony /var/log/chrony
$ sudo chmod -R g+w /var/log/chrony
```

- ouverture du pare-feu pour se connecter en extérieur sur les serveurs de temps configurés (port `123` en *UDP*) :

```
$ sudo firewall-cmd --zone=public --permanent --add-port=123/udp
$ sudo systemctl reload firewalld
```

- redémarrage du démon *chronyd* :

```
$ sudo systemctl restart chronyd
```

5.3.4 Usage

- Le service *chronyd* doit toujours et fonctionner et être démarré en même temps que le serveur. Au besoin, on le spécifiera à la main :

```
$ sudo systemctl enable chronyd
$ sudo systemctl start chronyd
```

- S'assurer que *chronyd* fonctionne bien :

```
$ sudo systemctl status chronyd
```

- Tester si *chrony* est bien synchronisé :

```
$ sudo chronyc tracking
Reference ID    : 81C76022 (ntp.ens.fr)
Stratum         : 3
Ref time (UTC)  : Tue Jan 29 12:32:56 2019
System time     : 0.000000071 seconds fast of NTP time
Last offset     : -0.000064014 seconds
RMS offset      : 0.000047961 seconds
Frequency       : 53.606 ppm slow
Residual freq   : -0.004 ppm
Skew            : 0.130 ppm
Root delay      : 0.001526146 seconds
Root dispersion : 0.018752204 seconds
Update interval : 129.5 seconds
Leap status     : Normal
```

- Tester la connectivité de *chrony* avec ses *sources* :

```
$ sudo chronyc sources
210 Number of sources = 5
MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================
^* ntp.ens.fr                    2   7   377    23    +32us[  +44us] +/-   35ms
^? 2001:bc8:271b:100::1          0  10     0     -     +0ns[   +0ns] +/-    ␣
↪0ns
```

- Consulter les statistiques relatives aux *sources* de *chrony* :

```
$ sudo chronyc sourcestats
210 Number of sources = 5
Name/IP Address            NP  NR  Span  Frequency  Freq Skew  Offset  Std Dev
==============================================================================
ntp.ens.fr                  9   5  1036     +0.017      0.134   +569ns    26us
2001:bc8:271b:100::1        0   0     0     +0.000   2000.000     +0ns  4000ms
```

5.4 sudo : configuration

5.4.1 Prérequis

O.S. *CentOS* 7

Note : *sudo* est installé par défaut sous *CentOS* 7.

5.4.2 Configuration d'un nouvel alias pour *sudo*

Afin de permettre à *sudo* d'exécuter également des *alias*, on ajoutera le suivant, par exemple dans `/etc/profile.d/common_aliases.sh` :

Code source 7: /etc/profile.d/common_aliases.sh

```
# NB : Bien laisser un espace terminal dans l'alias!
alias sudo='sudo '
```

Voir aussi :

https ://askubuntu.com/questions/22037/aliases-not-available-when-using-sudo

5.4.3 Configuration de `/etc/sudoers`

Le fichier de configuration est divisé en plusieurs sections :

1. les alias de machines hôtes ;
2. les alias d'utilisateurs et groupes ;
3. les alias de commandes et groupes de commandes ;
4. le paramétrage des valeurs par défaut, notamment la sélection des variables d'environnement à préserver ;
5. et enfin les règles d'autorisation : quels utilisateurs depuis quels hôtes sont autorisés à exécuter quelles commandes et sous quelles conditions.

Par la suite, on n'utilisera pas les alias de machines hôtes. Pour les autres sections, en général on utilisera le plus possible l'ensemble des configurations, en voyant ce fichier de configuration comme un mini-programme, où toutes les valeurs doivent être enregistrées dans des variables déclarées explicitement.

Alias d'utilisateurs et groupes

Code source 8: /etc/sudoers

```
# Déclaration d'un groupe d'utilisateur propre à 'sudo' :
User_Alias USERS = vferotin, naima
# Déclaration d'un groupe 'sudo' suivant un groupe système :
User_Alias ADMINS = %wheel
```

Note : En général, la 1ère forme (création d'un groupe d'utilisateurs propre à *sudo*) ne sera pas fréquemment utilisée.

Alias de commandes et groupes de commandes

Un alias de commande peut regrouper plusieurs commandes :

Code source 9: /etc/sudoers

```
# Alias d'une seule commande (notez le chemin complet de l'exécutable) :
Cmnd_Alias SU = /usr/bin/su
# Alias regroupant plusieurs commandes conceptuellement reliées
Cmnd_Alias NETWORKING = /sbin/route, /sbin/ifconfig, /bin/ping, /sbin/dhclient
```

Note :

- Les commandes doivent être spécifiées avec leurs chemins complets depuis la racine.
- Elles peuvent être enregistrées avec leurs options et paramètres respectifs.

Paramétrage des valeurs par défaut

La section suivante traite des valeurs par défaut pour *sudo*. Elle se compose essentiellement de deux parties à distinguer :

- la spécification de valeurs par défaut pour le comportement de *sudo*, qui revient essentiellement à n'être qu'une partie de configuration traditionnelle :

Code source 10: /etc/sudoers

```
# Définition des chemins "sûrs" des exécutables :
Defaults    secure_path = /sbin :/bin :/usr/sbin :/usr/bin :/usr/local/sbin :/usr/
↪local/bin
```

- le choix des variables d'environnement à conserver lors de l'usage de *sudo* :

Code source 11: /etc/sudoers

```
Defaults    always_set_home
Defaults    env_reset
Defaults    env_keep =  "COLORS DISPLAY HOSTNAME HISTSIZE KDEDIR LS_COLORS"
Defaults    env_keep += "MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE"
```

Note : On ferra attention à l'utilisation des guillemets doubles (`"`) et de l'opérateur `+=`.

Rêgles d'autorisation de commandes

Les rêgles d'autorisation de commandes sont les dernières, et s'appuient sur les définitions des variables des sections précédentes, en spécifiant quels utilisateurs, depuis quelles machines, sont autorisés à exécuter quelles commandes. La syntaxe générale pour une rêgle suit l'exemple suivant :

Code source 12: /etc/sudoers

```
USER_ALIAS  HOST_ALIAS=[OPTIONS] COMMAND_ALIAS_1,COMMAND_ALIAS_2[...]
```

où :

- une commande peut être :
  - une « simple » commande (mais nous déconseillons) ;
  - un alias de commande ;
  - `ALL` – pour signifier toutes les commandes ;
  - `!CMD` pour enlever une commande du jeu de celles autorisées – généralement utilisé conjoitement à `ALL`, e.g. :

Code source 13: /etc/sudoers

```
# Allows administrators to run all commands but 'su', providing their␣
↪password
ADMINS  ALL=(ALL)       ALL,!SU
```

- les options sont principalement :
  - un utilisateur, groupe d'utilisateur ou alias, entre parenthèses, pour spécifier l'utilisateur dont on prend les droits, e.g. : `(ALL)` ;

– `NOPASSWD :` pour spécifier que l'utilisateur n'a pas besoin de renseigner son mot de passe pour exécuter la commande, e.g. :

Code source 14: /etc/sudoers

```
# Allow all admins. to never be prompted for their password :
ADMINS   ALL=(ALL) NOPASSWD : ALL
```

5.5 git : installation et configuration

5.5.1 Prérequis

O.S. *CentOS* 7

dépôts RPM alternatifs activés

- *IUS*

Pour avoir une version assez récente de *git*, on installera le paquet `git2` d"*IUS*.

5.5.2 Installation proprement dite

```
$ sudo yum install git2u
```

5.5.3 Configuration minimale de *git* pour l'utilisateur courant

La configuration minimale pour *commiter* consiste à renseigner au minimum nom et adresse e-mail de l'utilisateur courant, e.g. :

```
$ git config --global user.email "vincent.ferotin@risc.cnrs.fr"
$ git config --global user.name "Vincent Férotin"
```

5.6 gitolite : installation, initialisation et usage

5.6.1 Prérequis

O.S. *CentOS* 7

autres logiciels

- *git 2*
- *OpenSSH server*

dépôts RPM alternatifs activés

- *EPEL*

configuration d"*OpenSSHd*

- *autorisation d'authentification par clef publique*

gitolite est fourni par *EPEL* dans le paquet `gitolite3`.

5.6.2 Installation proprement dite

```
$ sudo yum install gitolite3
```

L'installation de ce paquet et ses dépendances va automatiquement créer le nouvel utilisateur système `gitolite3`, qui sera le *host user* de *gitolite*.

5.6.3 Notes sur le nouvel utilisateur système *gitolite3*

- Son *$HOME* est `/var/lib/gitolite3`.
- Pour des raisons de sécurité des accès aux dépôts *git* qu'il gère, on prendra soin de lui adjoindre un mot de passe fort, et connu uniquement des seuls administrateurs.

5.6.4 Initialisation minimale pour configuration pour une utilisation en « mode SSH »

1. Génération, si besoin, d'une paire de clef publique/privée, pour l'utilisateur administrateur des dépôts gérés par *gitolite* :

   ```
   # sur le poste de l'utilisateur (et non sur le serveur)
   $ ssh-keygen -t ed25519 -b 256 -N ""
   ```

   Voir aussi :

   la section relative à la génération de clefs SSH

   puis copier la clef publique (e.g. `/home/vferotin/.ssh/id_ed25519.pub`) sur le serveur, en utilisant le nom de l'utilisateur administrateur de *gitolite* (e.g. `/tmp/vfadmin.pub`).

2. Créer le *HOME* et le mot de passe pour le nouvel utilisateur *gitolite3* :

   ```
   # sur le serveur
   $ sudo passwd gitolite3
   ```

3. Se logger (sur le serveur) en tant que l'utilisateur *gitolite3* :

   ```
   # sur le serveur
   $ su gitolite3 -
   $ cd ~
   ```

   et lancer l'initialisation de *gitolite* :

   ```
   $ gitolite setup -pk /tmp/vfadmin.pub
   Dépôt Git vide initialisé dans /var/lib/gitolite3/repositories/gitolite-admin.
   ↪git/
   Dépôt Git vide initialisé dans /var/lib/gitolite3/repositories/testing.git/
   $ exit
   ```

4. Cloner en local le dépôt d'administration de *gitolite* `gitolite-admin` :

   ```
   # sur le poste de l'utilisateur
   $ cd ~  # par ex.
   $ git clone ssh ://gitolite3@<serveur>/gitolite-admin
   # Vérifier que l'empreinte proposée d'une clef publique correspond
   # bien à une de celles communiquées par l'admin. du serveur.
   ```

   Voir aussi :

   Afficher l'empreinte d'une clef publique

5.6.5 Configuration

La configuration additionnelle de *gitolite* s'effecute dans son fichier « rc » : `/var/lib/gitolite3/.gitolite.rc`. On effectuera les modifications suivantes à la version par défaut fournie par le paquet :

- désactivation des *features* `daemon` et `gitweb` :

Code source 15: .gitolite.rc

```
%RC = (
    # ...
    ENABLE => [
        # ...
        #'daemon',      # <= à commenter
        # ...
        #'gitweb',      # <= à commenter
        # ...
    ],
);
# ...
```

5.6.6 Usage

Note : Cette section n'est qu'un aperçu de haut niveau.

gitolite est bien plus complet et puissant que le bref aperçu donné ci-dessous, qui ne vise qu'à « mettre en selle » l'administrateur avec ce nouvel outil. On n'hésitera pas à se reporter aux *ressources listées dans la base de connaissance*, papier et/ou en ligne, relatives au projet, pour des besoins moins grossiers ou non couverts ci-dessous…

Gestion des « utilisateurs »

Les utilisateurs sont « gérés » par la présence ou l'absence de leurs clefs publiques respectives dans le répertoire `gitolite-admin/keydir`. Le nom de la clef implique le nom de l'utilisateur.

Pour **ajouter** un nouvel utilisateur, on copiera donc sa clef publique dans ce répertoire (local), *commitera* en local le changement, et *pushera* le *commit* sur le dépôt *gitolite-admin* du serveur.

Pour **supprimer** un utilisateur, on procédera de manière symétrique : on supprimera le fichier correspondant à sa clef publique en local, toutes les entrées définissant les droits de cet utilisateur sur les différents dépôts, *commitera* et *pushera* sur le dépôt *gitolite-admin* du serveur.

Gestion des *dépôts git*

L'"**ajout** d'un nouveau dépôt se fait par l'ajout de droits d'au moins un utilisateur sur celui-ci, en modifiant le fichier texte `gitolite-admin/conf/gitolite.conf`, par ex. avec :

Code source 16: gitolite-admin/conf/gitolite.conf

```
repo nouveau_depot
    RW+     =   vferotin
```

Note : On peut très bien préciser un chemin au sein d'une arborescence de répertoires : *gitolite* se chargera de la créer (e.g. `test/arbo/depot` comme « nom » de dépôt) !

La **suppression** d'un dépôt se fait en deux temps :

1. suppression de la section dédiée dans `gitolite-admin/conf/gitolite.conf`, *commit* et *push* sur le serveur ;
2. suppression « à la main » sur le serveur, par ex. pour le dépôt `ancien_depot` :

```
# sur le serveur
$ su gitolite3 -
$ cd ~/repositories
$ rm -Rf ancien_depot.git
$ exit
```

Le **renommage** d'un dépôt existant se fait en deux temps :

1. renommage à la main du dépôt sur le serveur, par ex. `ancien_depot` en `nouveau_depot` :

```
# sur le serveur
$ su gitolite3 -
$ cd ~/repositories
# NE PAS OUBLIER DE SUFFIXER LE NOM PAR ".git"
$ mv ancien_depot.git nouveau_depot.git
$ exit
```

2. changement du nom dans `gitolite-admin/conf/gitolite.conf`, *commit* et *push* sur le serveur.

Gestion simple des droits des « utilisateurs » sur un *dépôt*

Les droits des utilisateurs pour chaque dépôt sont définis dans le fichier texte `gitolite-admin/conf/gitolite.conf`. Ces droits sont basiquement :

`R` *pull* (i.e. « lecture seule »)

`RW` `R` + *fast-forward push* d'une *branche*, ainsi que création de *branches* et *tags*

`RW+` « tous les droits », i.e. `RW` + *rewind push*, ainsi que suppression de *branches* et *tags*

Interaction avec un dépôt directement sur le serveur

Il est toujours possible d'interagir directement, sur le serveur, avec un dépôt *bare*, par ex. ici `nouveau_depot.git`, en spécifiant la variable d'environnement `GIT_DIR` pour exécuter une commande *git* :

```
# sur le serveur
$ su gitolite3 -
$ cd ~
$ GIT_DIR="./repositories/nouveau_depot.git" git remote -v
$ exit
```

5.7 etckeeper : installation et configuration

5.7.1 Prérequis

O.S. *CentOS 7*

autres logiciels

- *git 2*

- *gitolite*
- *sudo*
- *ssh-agent*

dépôts RPM alternatifs activés

- *EPEL* (pour *etckeeper*)

5.7.2 Installation proprement dite

```
$ sudo yum install etckeeper
```

5.7.3 Edition de la configuration du logiciel

On adaptera `/etc/etckeeper/etckeeper.conf` à nos besoins :

Code source 17: /etc/etckeeper/etckeeper.conf

```
VSC="git"
GIT_COMMIT_OPTIONS=""
AVOID_DAILY_AUTOCOMMITS=1
HIGHLEVEL_PACKAGE_MANAGER=yum
LOWLEVEL_PACKAGE_MANAGER=rpm
PUSH_REMOTE=""
```

et désactivera son intégration comme *plugin* à *yum* :

Code source 18: /etc/yum/pluginconf.d/etckeeper.conf

```
[main]
enabled=0
```

en préférant *un usage manuel*.

5.7.4 Configuration minimale de git pour l'utilisateur admin. courant

Cette opération est nécessaire pour pouvoir utiliser `sudo etckeeper commit` : voir *configuration minimale de git*.

5.7.5 Initialisation du dépôt git d"`/etc`

1. Initialisation par *etckeeper* :

   ```
   $ sudo etckeeper init
   ```

   Cette action met plein de fichiers de conf. dans l"*index git* du dépôt, prêts pour versionnement. Mais nous préférrons effectuer, avant ce lourd *commit*, 2 *commits* préalables :

   1. un trivial pour initialiser le dépôt, avec un `.gitignore` vide ;
   2. un spécifique avec une version adaptée de `.gitignore` à partir de celle proposée par *etckeeper*.

2. Initialisation du dépôt *git* avec un `.gitignore` vide :

```
$ cd /etc
$ sudo mv .gitignore .gitignore.save
$ sudo touch .gitignore
```

Comme nous allons utiliser la sous-commande d'*etckeeper* `vcs`, faisant appel à *git*, à travers *sudo*, mais que l'utilisateur *root* n'a pas de configuration minimale réalisée (nous l'avons fait pour l'utilisateur admin.), il nous faut localement surcharger la config. de *git* avec les valeurs qui vont bien.

Enregistrement du 1er commit avec uniquement notre fichier `.gitignore` vide :

```
$ sudo etckeeper vcs -c "user.name=Vincent Férotin" -c "user.email=vincent.
↪ferotin@risc.cnrs.fr" commit -- .gitignore
```

3. Création du 2nd *commit* avec une version modifiée du `.gitignore` proposé par *etckeeper* :

```
$ sudo mv .gitignore.save .gitignore
```

Désactivation de quelques *patterns* à ignorer (on souhaite donc qu'ils soient bien pris en compte par *git*) :

Code source 19: /etc/.gitignore

```
#*.rpmnew
#*.rpmorig
#*.rpmsave
#*.old

# ... (le reste est inchangé)
```

Versionnement de cette nouvelle version (mêmes besoins et remarques que précédemment) :

```
$ sudo git add .gitignore
$ sudo etckeeper vcs -c "user.name=Vincent Férotin" -c "user.email=vincent.
↪ferotin@risc.cnrs.fr" commit -- .gitignore
```

4. Versionnement effectif de la configuration actuelle :

Note : Renseignement au préalable des identifiants de l'utilisateur

Avant de réaliser la commande `sudo etckeeper commit [...]` ci-dessous, on aura pris soin de *renseigner, pour l'utilisateur opérant en tant qu'administrateur, les nom et mail en tant qu'utilisateur de git* que réutilisera automatiquement *etckeeper*, de manière que ce soit ces nouvelles valeurs qui soient utilisées pour l'auteur de ce commit.

```
# Vérification que tous les fichiers à versionner sont bien ajoutés
# à l'index de git
$ sudo git status
# Ajout de fichiers modifiés au besoin, e.g. :
$ sudo etckeeper vcs add .etckeeper firewalld/zones/public.xml.old
$ sudo etckeeper commit "Configuration de ServeurPrincipal au 2019-01-10"
```

5.7.6 Compactage du dépôt *git* (optionnel)

C'est une opération recommandée par les développeurs d'*etckeeper*. Dans nos essais en machines virtuelles, cela permet une diminution environ par deux de l'espace occupé par le dossier `.git/` :

```
$ cd /etc
# Afficher les tailles du dépôt
$ sudo du -c -d 1 -h .git
# Réaliser le compactage
$ sudo etckeeper vcs gc
# Vérifier la différence des tailles du dépôt
$ sudo du -c -d 1 -h .git
```

5.7.7 Configuration pour avoir un clone de `/etc/.git` dans un dépôt géré par gitolite

L'astuce principale consiste ici à configurer le dépôt géré par *gitolite* comme un *remote* de celui géré par *etckeeper*, en passant par le protocole *SSH* et la procédure standard via *gitolite* – même s'il s'agit d'un dépôt sur le filesystem de la même machine.

1. Création d'un nouveau dépôt (futur *clone* de `/etc/.git`) géré par *gitolite* :
    1. création d'une bi-clef pour chaque utilisateur administrateur local, du groupe `wheel`, sur le serveur, qui sera amené à enregistrer de nouveaux commits sur le dépôt géré par *etckeeper* :

       Voir aussi :

       Génération rapide d'une bi-clef pour SSH

       ```
       # bien sur le serveur
       $ whoami
       vfadmin
       $ ssh-keygen -t ed25519 -b 256 -N ""
       # une nouvelle bi-clef est créée dans /home/vfadmin/.ssh/id_ed25519*
       ```

       et :

       ```
       # sur votre machine locale
       $ whoami
       vferotin
       $ ssh-keygen -t ed25519 -b 256 -N ""
       ```

    2. ajouter à *gitolite-admin/keydir* les clefs publiques générées de ces administrateurs :

       ```
       # sur votre machine locale
       $ cd /home/work/RISC/SI/serveur_principal/gitolite-admin
       # chaque clef publique a été copiée sur votre machine depuis le serveur
       # par ex. dans /tmp
       $ cp /tmp/*.pub keydir/
       $ git add keydir/
       ```

    3. ajouter via *gitolite-admin* un nouveau dépôt, auquel : (i) seuls quelques utilisateurs du groupe administrateurs `wheel` auront accès, en lecture et écriture, (ii) seuls quelques utilisateurs extérieurs auront accès, en lecture seulement :

       Code source 20: gitolite-admin/conf/gitolite.conf

       ```
       repo serveur_principal-etc
           RW      =   vfadmin         # admin. local (i.e. sur ServeurPrincipal)
           R       =   vferotin        # utilisateur distant
       ```

       puis *commiter* et *pusher* sur le serveur, afin de créer le dépôt *bare* managé par *gitolite*.
2. Mettre en place la communication entre le dépôt *git* géré par *etckeeper* et son *clone* géré par *gitolite* :

1. ajouter un *remote*, nommé `origin` et pointant vers `/etc`, au dépôt *bare* géré par *gitolite* :

   Note : Bien suffixer le dépôt par `.git` !

```
# sur le serveur
$ su gitolite3 -
$ cd ~
$ GIT_DIR="./repositories/serveur_principal-etc.git" git remote add origin /
↪etc
$ exit
```

2. ajout d'un *remote*, nommé `gitolite` et pointant vers `/var/lib/gitolite3/repositories/serveur_principal-etc.git` **MAIS** via *SSH* :

```
# sur le serveur
$ cd /etc
$ sudo etckeeper vcs remote add gitolite ssh ://gitolite3@localhost/serveur_
↪principal-etc
```

3. Effectuer un premier *push* depuis *etckeeper* vers *gitolite* :
   1. configurer *sudo* pour garder la variable d'environnement `SSH_AUTH_SOCK` initialisée par *ssh-agent* :

```
# sur le serveur
$ sudo visudo
----
Defaults    env_keep += "SSH_AUTH_SOCK"
----
```

   2. *pusher* en *utilisant "ssh-agent"* :

```
# sur le serveur
$ whoami
vfadmin
$ cd /etc
$ ssh-agent bash
$ ssh-add
Identity added : /home/vfadmin/.ssh/id_ed25519 (vfadmin@serveur_principal)
$ sudo etckeeper vcs push --set-upstream gitolite master
# quitter ssh-agent
$ exit
```

4. Configurer *etckeeper* pour automatiquement *pusher* ses nouveaux commits sur son *clone* géré par *gitolite* :

Code source 21: /etc/etckeeper/etckeeper.conf

```
PUSH_REMOTE="gitolite"
```

enregistrer cette modification et les précédentes dans le dépôt géré par *etckeeper*, et *pusher* via *"ssh-agent"* :

```
# sur le serveur
$ whoami
vfadmin
$ cd /etc
# e.g.
```

(suite sur la page suivante)

(suite de la page précédente)

```
$ sudo git add sudoers etckeeper/etckeeper.conf
$ ssh-agent bash
$ ssh-add
Identity added : /home/vfadmin/.ssh/id_ed25519 (vfadmin@serveur_principal)
$ sudo etckeeper commit "[Message de commit]"
$ exit
```

Il ne reste alors plus, sur votre machine, qu'à récupérer l'intégralité du dépôt :

```
# sur votre machine
$ whoami
vferotin
$ cd /home/work/RISC/SI/confs/serveur_principal
$ git clone ssh ://gitolite3@serveur_principal/serveur_principal-etc reference
```

et à *puller* systématiquement lorsque des changements auront enregistrés.

5.7.8 Usage

Une fois configuré, l'utilisation de *etckeeper* via *"ssh-agent"* est simple :

```
# [Des changement surviennent sur /etc
# par exemple suite à mise à jour de paquets...]
# sur le serveur
$ whoami
vfadmin
$ ssh-agent bash
$ ssh-add
Identity added : /home/vfadmin/.ssh/id_ed25519 (vfadmin@serveur_principal)
$ cd /etc
# Lister les modifications de /etc par rapport au dépôt git :
$ sudo git status
# ... liste des fichiers modifiés...
# Préparer le changement, ajouter les fichiers dans leurs nouvelles versions :
$ sudo git add fichier1.conf dir/fichier2.ini
# Enregistrer le changement sous forme d'un nouveau commit :
$ sudo etckeeper commit "Mise à jour de la configuration (paquets : ...)"
# exit from ssh-agent
$ exit
```

et bien penser à récupérer ces nouveaux changements dans votre *clone* local, éventuellement via *"ssh-agent"*, e.g. :

```
# sur votre machine
$ whoami
vferotin
# dans le working directory du clone
$ cd /home/work/RISC/SI/confs/serveur_principal/reference  # sur un volume chiffré,␣
↪évidemment
$ ssh-agent bash
# utilisation d'une bi-clef dédiée, enregistrée elle-aussi sur un volume chiffré
$ ssh-add /media/veracrypt1/RISC/secrets/ssh/serveur_principal/etc/id_ed25519
Identity added : /media/veracrypt1/RISC/secrets/ssh/serveur_principal/etc/id_ed25519␣
↪(vferotin@tryscoped)
$ git pull origin master
```

(suite sur la page suivante)

(suite de la page précédente)

```
# fin de session ssh-agent
$ exit
```

5.8 ReaR : installation, configuration et usage

5.8.1 Prérequis

O.S. *CentOS* 7

version de *ReaR* 2.4

espace libre du support USB supérieur ou égal à (au moins) 4 Go

espace libre sur `/tmp` supérieur ou égal à (au moins) 2 Go

5.8.2 Installation proprement dite

```
$ sudo yum install rear
# Pour créer un périphérique USB, ajouter les dépendances :
$ sudo yum install syslinux syslinux-extlinux
# Pour créer une ISO, ajouter la dépendance :
$ sudo yum install genisoimage
```

5.8.3 Usage et configuration

La configuration de *ReaR* se fait à travers essentiellement le seul fichier `/etc/rear/local.conf`. Nous déclinons ici deux configurations distinctes, pour créer ou une ISO bootable ou un support USB bootable.

En pratique, vu le court temps de création des sauvegardes, on ne se privera pas de faire et l'une et l'autre – dans la foulée – en commençant par créer le périphérique USB bootable, sur lequel ensuite on enregistrera l'ISO.

Avant création des sauvegardes, si la machine est un serveur

On préférera désactiver tous les services web ou assimilés avant de faire la sauvegarde proprement dite (et les remonter à la main après restauration de la sauvegarde). C'est à dire, pour par ex. *serveur principal* :

```
$ sudo systemctl stop httpd
$ sudo systemctl stop php-fpm
$ sudo systemctl stop mariadb
$ sudo systemctl stop sshd
$ sudo systemctl stop fail2ban
```

Création d'un périphérique (e.g. clef) USB bootable

Configuration de *ReaR* pour créer un périphérique USB bootable restaurant complètement le système, par ex. `/dev/sdb` :

1. Branchement physique du périphérique USB (reconnu par ex. comme `/dev/sdb`).
2. Création du système de fichier du périphérique USB :

```
$ sudo rear format -v /dev/sdb
```

3. Création de la configuration de *ReaR* :

Code source 22: /etc/rear/local.conf (USB)

```
# write the rescue initramfs to an USB key
OUTPUT=USB
USB_DEVICE=/dev/disk/by-label/REAR-000
# backup data method
BACKUP=NETFS
# backup data directly into USB (CAUTION : all data will be erased!)
BACKUP_URL=usb :///dev/disk/by-label/REAR-000
```

4. Création de la sauvegarde :

```
$ sudo rear -v mkbackup
```

5. Eventuellement, passer à la section suivante (pour enregistrer une ISO de sauvegarde sur ce même périphérique).
6. Enlever physiquement le périphérique USB.
7. Sauvegarder le contenu du périphérique USB ainsi formatté, sur des autres systèmes de stockage extérieurs (où il sera par ex. reconnu comme `/dev/sdc`), grâce à *fsarchiver* :

```
$ mkdir ~/rear_usb_rescue-$MACHINE-$DATE
$ cd ~/rear_usb_rescue-$MACHINE-$DATE
# fsarchiver options :
#    #1   is new archive
#    #2   is device partition
#    -v   is for more output verbosity
#    -j   is for setting number of threads (jobs) to run, depending
#         on your number of cores
#    -L   is for labelling archive with a short description
#    -Z   is for setting Zstandart compression level (and I have no idea
#         if 16 is a "good" value)
$ sudo fsarchiver savefs ./rear_rescue.fsarchiv /dev/sdc1 -v -Z 16 -j8 -L
↪"Backup ReaR : ServeurPrincipal on CentOS 7.6 - 20190204"
```

Sauvegarde sous forme d'une ISO bootable

Configuration de *ReaR* pour générer une ISO bootable restaurant complètement le système, sauvegardée sur un périphérique USB (par ex. `/dev/sdb`) :

1. Montage de la première partition de `/dev/sdb` sur `/mnt` :

```
$ sudo mount -w /dev/sdb1 /mnt
```

2. Création de la configuration de *ReaR* :

Code source 23: /etc/rear/local.conf (ISO)

```
# write the rescue initramfs to an ISO
OUTPUT=ISO
# write ISO in following directory
OUTPUT_URL=file :///mnt/ServeurPrincipal-CentOS-7.5-ReaRed-20181206-1315
# backup data method
```

(suite sur la page suivante)

(suite de la page précédente)

```
BACKUP=NETFS
# backup data directly into ISO
#   /!\ CAUTION /!\
#   This is not a configuration recommended by ReaR
#   (see e.g. https://github.com/rear/rear/issues/581).
#   Also, this should work for a *small* whole filesystem...
BACKUP_URL=iso:///backup
```

3. Création de la sauvegarde :

```
$ sudo rear -v mkbackup
```

4. Démontage de `/mnt` :

```
$ sudo umount /mnt
```

5. Sauvegarder le fichier ISO ainsi créé sur des espaces de stockage extérieurs.

Restauration de la sauvegarde

Si ISO :

1. La graver (CD-R, DVD-R)
2. Insérer le medium dans le lecteur de la machine à restaurer, et redémarrer
3. Choisir de booter sur le medium
4. Choisir dans le menu de boot du medium l'option « Recover automatically »
5. Relax, and say « yes » to recovering ;-)
6. Rebooter et éjecter le medium ; croiser les doigts

Si image *fsarchiver* :

1. Insérer le périphérique USB sur lequel remettre l'image, par ex. `/dev/sdd`. **Attention !**, cela supprimera tout le contenu de sa première partition `/dev/sdd1`.

2.
```
$ cd  ~/rear_usb_rescue-$MACHINE-$DATE
# fsarchiver options :
#    #1   is archive path
#    #2   is a bag of detailed options :
#         id :   is num. of filesystem in archive to restore
#         dest : is device partition path into which restore filesystem
#    -v  for more output verbosity
$ sudo fsarchiver restfs ./rear_rescue.fsarchiv.fsa id=0,dest=/dev/sdd1 -v
```

Après création des sauvegardes / restauration, si la machine est un serveur

Si la machine est un serveur, et que l'on a précédemment désactivé les principaux services (Web ou associés), on les relancera à la main. C'est à dire, pour par ex. *serveur principal* :

```
$ sudo systemctl start fail2ban
$ sudo systemctl start sshd
$ sudo systemctl start mariadb
$ sudo systemctl start php-fpm
$ sudo systemctl start httpd
```

5.9 smartmontools : Usage basique

Pour un disque `/dev/sdX` donné :

1. Activer *S.M.A.R.T.* pour ce disque :

```
$ sudo smartctl --smart=on --offlineauto=on --saveauto=on /dev/sdX
```

2. Afficher l'état de santé du disque :

```
$ sudo smartctl -H -i /dev/sdX
```

3. Voir le résumé des journaux de tests *S.M.A.R.T.* précédants :

```
$ sudo smartctl -l selftest /dev/sdX
```

Note : Les journaux sont normalement affichés et numérotés de manière anti-chronologique, i.e. le plus récent en premier (`1`).

---

4. Effectuer un test, court ou long, sur le disque :

```
# Test court :
$ sudo smartctl -t short /dev/sdX
# Test long :
$ sudo smartctl -t long /dev/sdX
```

Note : Lancer un test par une telle commande retourne immédiatement l'invite de commande : le test est lancé en arrière-plan. Mais il n'y a alors aucune sortie par défaut vous permettant de voir l'avancement du test et/ou ses résultats. On utilisera alors régulièrement la commande précédante, `-l selftest` pour juger de l'avancement de la tâche demandée.

---

5. Voir le journal d'erreur consécutif d'un test :

```
# Une fois le test terminé
#   Pour voir le journal d'erreurs du dernier test, #1
$ sudo smartctl -l error,1 /dev/sdX
```

À faire : `$ sudo smartctl -l xselftest,1 /dev/sdX` ? ?

5.10 fsck : Usage

5.10.1 Principales options

Les principales options utiles de *fsck* sont à mon avis les suivantes :

`-f` Forcer l'exécution, même si le volue ou la partition semble clean.

`-v` Passer l'utilitaire en mode plus verbeux.

`-C0` Afficher la progression de la vérification sur la sortie standard.

`-y` répondre automatiquement *Yes* à toutes les questions que l'utilitaire poserait, et qui freinerait son application automatique.

`-c` Vérifier l'existence de secteurs défectueux. Option à doubler éventuellement pour plus de rigueur : `-c -c`.

5.10.2 Utilisation basique

Pour une partition *ext4* `/dev/sdaX` donnée :

1. La démonter au préalable :

```
$ sudo umount /dev/sdaX
```

2. Effectuer une première passe « standard » :

```
$ sudo fsck.ext4 -v -f -C0 -y /dev/sdaX
```

3. Au besoin, vérifier les secteurs défectueux :

```
$ sudo fsck.ext4 -v -f -C0 -y -c -c /dev/sdaX
```

5.11 badblocks : Usage

5.11.1 Principales options

Les principales options utiles de *badblocks* sont à mon avis les suivantes :

`-v` Passer l'utilitaire en mode plus verbeux.

`-s` Afficher la progression de la vérification sur la sortie standard.

5.11.2 Utilisation basique

Pour un disque `/dev/sdX` donné, toutes partitions démontées :

1. Effectuer une première passe « standard » :

```
$ sudo badblocks -v -s /dev/sdX
```

5.12 Apache httpd : installation, configuration et usage

5.12.1 Prérequis

O.S. *CentOS* 7

autres logiciels

5.12.2 Installation

```
$ sudo yum install httpd httpd-tools
# Pour installer également le manuel d'Apache :
$ sudo yum install httpd-manual
```

5.12.3 Configurations

Configuration du pare-feu pour autoriser les protocoles *HTTP(S)* sur leurs ports respectifs

```
# Autoriser le traffic HTTP pour la zone 'public'
$ sudo firewall-cmd --zone=public --add-service=http
# Autoriser le traffic HTTPS pour la zone 'public'
$ sudo firewall-cmd --zone=public --add-service=https
# Autoriser le traffic TCP sur le port 80 pour la zone 'public'
$ sudo firewall-cmd --zone=public --add-port=80/tcp
# Autoriser le traffic TCP sur le port 443 pour la zone 'public'
$ sudo firewall-cmd --zone=public --add-port=443/tcp
# Save changes to permanent config
$ sudo firewall-cmd --runtime-to-permanent
```

À faire : *firewalld* : Y a-t-il besoin, et si oui expliquer, d'autoriser et le service HTTP (resp. HTTPS) **ET** le port 80 (resp. 443) ?

Configuration initiale de *systemd*

On souhaite généralement que le démon tourne tout le temps, et dès le démarrage de la machine serveur. Mais pour l'instant, tant qu'on n'a pas un minimum sécurisé sa configuration, on ne souhaite pas qu'il soit démarré :

```
$ sudo systemctl enable httpd
$ sudo systemctl stop httpd
```

Configuration proprement dite de *httpd*

Configuration de `httpd.conf`

Par rapport à la version de base fournie par le paquet, on spécifiera :

- l'adresse e-mail de l'administrateur, e.g. :

Code source 24: /etc/httpd/conf/httpd.conf

```
ServerAdmin webmaster@risc.cnrs.fr
```

- le « nom » de la machine serveur, généralement le nom de domaine du *VirtualHost* par défaut, e.g. :

Code source 25: /etc/httpd/conf/httpd.conf

```
ServerName <NOM.DE.DOMAINE> :80
```

- la désactivation des scripts CGI globaux :

Code source 26: /etc/httpd/conf/httpd.conf

```
<IfModule alias_module>
    #ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"
</IfModule>
```

Désactivation de fichiers entiers de configuration dans `conf.d/`

Pour désactiver des fichiers entiers de configuration dans le répertoire `/etc/httpd/conf.d`, on procédera comme suite pour un fichier donné :

1. renommage du fichier avec un suffixe significatif, e.g. `.disabled` ;
2. création d'un fichier vide portant le nom initial du fichier

e.g. :

```
$ cd /etc/httpd/conf.d
# Désactivation du fichier 'autoindex.conf'
$ sudo mv autoindex.conf autoindex.conf.disabled
$ sudo touch autoindex.conf
```

ceci afin d'empêcher l'installation d'une ancienne version non désirée d'un fichier par la mise à jour d'un paquet, qui ne viendrait pas écraser un fichier absent.

On désactivera de la sorte les fichiers de configuration suivants :

- `autoindex.conf`
- `manual.conf`
- `userdir.conf`
- `welcome.conf`

Activation ou désactivation des modules dans `conf.modules.d/00-base.conf`

Dans le fichier de configuration `/etc/httpd/conf.modules.d/00-base.conf`, on activera / désactivera / gardera les modules suivants, en partant du principe que le moins de modules sont activés le mieux c'est (performances, mémoire, sécurité, cohérence de config., etc.) :

Module	Défaut	Etat désiré	Description
access_compat	activé	**désactivé**	Group authorizations based on host (name or IP address)
actions	activé	**désactivé**	Execute CGI scripts based on media type or request method
alias	activé	activé	Provides for mapping different parts of the host filesystem in the docume
allowmethods	activé	activé	Easily restrict what HTTP methods can be used on the server
auth_basic	activé	activé	Basic HTTP authentication
auth_digest	activé	**désactivé**	User authentication using MD5 Digest Authentication
authn_anon	activé	**désactivé**	Allows « anonymous » user access to authenticated areas
authn_core	activé	activé	Core Authentication
authn_dbd	activé	**désactivé**	User authentication using an SQL database
authn_dbm	activé	**désactivé**	User authentication using DBM files
authn_file	activé	activé	User authentication using text files
authn_socache	activé	activé	Manages a cache of authentication credentials to relieve the load on backe
authz_core	activé	activé	Core Authorization
authz_dbd	activé	**désactivé**	Group Authorization and Login using SQL
authz_dbm	activé	**désactivé**	Group authorization using DBM files
authz_groupfile	activé	activé	Group authorization using plaintext files
authz_host	activé	activé	Group authorizations based on host (name or IP address)
authz_owner	activé	**désactivé**	Authorization based on file ownership
authz_user	activé	activé	User Authorization
autoindex	activé	**désactivé**	Generates directory indexes, automatically, similar to the Unix ls comma
cache	activé	**désactivé**	RFC 2616 compliant HTTP caching filter
cache_disk	activé	**désactivé**	Disk based storage module for the HTTP caching filter.
data	activé	**désactivé**	Convert response body into an RFC2397 data URL

Tableau 1 – suite de la page précédente

Module	Défaut	Etat désiré	Description
dbd	activé	**désactivé**	Manages SQL database connections
deflate	activé	activé	Compress content before it is delivered to the client
dir	activé	activé	Provides for « trailing slash » redirects and serving directory index files
dumpio	activé	**désactivé**	Dumps all I/O to error log as desired.
echo	activé	**désactivé**	A simple echo server to illustrate protocol modules
env	activé	**désactivé**	Modifies the environment which is passed to CGI scripts and SSI pages
expires	activé	activé	Generation of Expires and Cache-Control HTTP headers according to use
ext_filter	activé	**désactivé**	Pass the response body through an external program before delivery to th
filter	activé	**désactivé**	Context-sensitive smart filter configuration module
headers	activé	activé	Customization of HTTP request and response headers
include	activé	**désactivé**	Server-parsed html documents (Server Side Includes)
info	activé	activé	Provides a comprehensive overview of the server configuration
log_config	activé	activé	Logging of the requests made to the server
logio	activé	**désactivé**	Logging of input and output bytes per request
mime_magic	activé	activé	Determines the MIME type of a file by looking at a few bytes of its conte
mime	activé	activé	Associates the requested filename's extensions with the file's behavior (ha
negotiation	activé	activé	Provides for content negotiation
remoteip	activé	**désactivé**	Replaces the original client IP address for the connection with the userag
reqtimeout	activé	activé	Set timeout and minimum data rate for receiving requests
rewrite	activé	activé	Provides a rule-based rewriting engine to rewrite requested URLs on the f
setenvif	activé	activé	Allows the setting of environment variables based on characteristics of the
slotmem_plain	activé	**désactivé**	Slot-based shared memory provider
slotmem_shm	activé	activé	Slot-based shared memory provider
socache_dbm	activé	**désactivé**	DBM based shared object cache provider
socache_memcache	activé	**désactivé**	Memcache based shared object cache provider
socache_shmcb	activé	activé	shmcb based shared object cache provider
status	activé	activé	Provides information on server activity and performance
substitute	activé	**désactivé**	Perform search and replace operations on response bodies
suexec	activé	**désactivé**	Allows CGI scripts to run as a specified user and Group
unique_id	activé	**désactivé**	Provides an environment variable with a unique identifier for each request
unixd	activé	activé	Basic (required) security for Unix-family platforms
userdir	activé	**désactivé**	User-specific directories
version	activé	**désactivé**	Version dependent configuration
vhost_alias	activé	**désactivé**	Provides for dynamically configured mass virtual hosting

Module	Défaut	Etat désiré	Description
buffer	désactivé	désactivé	Support for request buffering
watchdog	désactivé	désactivé	provides infrastructure for other modules to periodically run tasks
heartbeat	désactivé	désactivé	Sends messages with server status to frontend proxy
heartmonitor	désactivé	désactivé	Centralized monitor for mod_heartbeat origin servers
usertrack	désactivé	désactivé	Clickstream logging of user activity on a site
dialup	désactivé	désactivé	Send static content at a bandwidth rate limit, defined by the various old modem standards
charset_lite	désactivé	désactivé	Specify character set translation or recoding
log_debug	désactivé	désactivé	Additional configurable debug logging
ratelimit	désactivé	désactivé	Bandwidth Rate Limiting for Clients
reflector	désactivé	désactivé	Reflect a request body as a response via the output filter stack
request	désactivé	désactivé	Filters to handle and make available HTTP request bodies
sed	désactivé	désactivé	Filter Input (request) and Output (response) content using sed syntax
speling	désactivé	désactivé	Attempts to correct mistaken URLs by ignoring capitalization, or attempting to correct various minor misspellings

Désactivation des modules dans `conf.modules.d/00-dav.conf`

Dans le fichier de configuration `/etc/httpd/conf.modules.d/00-dav.conf`, on désactivera tous les modules (même principes que précédemment) :

Module	Défaut	Etat désiré	Description
dav	activé	**désactivé**	Distributed Authoring and Versioning (WebDAV) functionality
dav_fs	activé	**désactivé**	Filesystem provider for mod_dav
dav_lock	activé	**désactivé**	Generic locking module for mod_dav

Désactivation des modules dans `conf.modules.d/00-lua.conf`

Dans le fichier de configuration `/etc/httpd/conf.modules.d/00-lua.conf`, on désactivera tous les modules (même principes que précédemment) :

Module	Défaut	Etat désiré	Description
lua	activé	**désactivé**	Provides Lua hooks into various portions of the httpd request processing

Configuration de *MPM* dans `conf.modules.d/00-mpm.conf`

Dans le fichier de configuration `/etc/httpd/conf.modules.d/00-mpm.conf`, on sélectionnera le module *event MPM* plutôt que *prefork* ou *worker* :

Module	Défaut	Etat désiré	Description
mpm_prefork	activé	**désactivé**	Implements a non-threaded, pre-forking web server
mpm_worker	désactivé	désactivé	Multi-Processing Module implementing a hybrid multi-threaded multi-process web server
mpm_event	désactivé	**activé**	A variant of the worker MPM with the goal of consuming threads only for connections with active processing

On profitera de l'édition de ce fichier de conf. pour spécifier la configuration de *event MPM* :

Code source 27: /etc/httpd/conf.modules.d/00-mpm.conf

```
LoadModule mpm_event_module modules/mod_mpm_event.so

GracefulShutdownTimeout 10
MaxConnectionsPerChild 10000

ServerLimit 25
#ThreadsPerChild 25
ThreadLimit 25
MaxRequestWorkers 625

MaxSpareThreads 400
MinSpareThreads 200
```

Configuration des *proxy* dans `conf.modules.d/00-proxy.conf`

Dans le fichier de configuration `/etc/httpd/conf.modules.d/00-proxy.conf`, on désactivera tous les modules sauf ceux nécessaires pour *PHP-FPM* (`proxy` et `proxy_fcgi`) :

Module	Défaut	Etat désiré	Description
proxy	activé	désactivé	Multi-protocol proxy/gateway server
lbmethod_bybusyness	activé	**désactivé**	Pending Request Counting load balancer scheduler algorithm for mod_proxy_balancer
lbmethod_byrequests	activé	**désactivé**	Request Counting load balancer scheduler algorithm for mod_proxy_balancer
lbmethod_bytraffic	activé	**désactivé**	Weighted Traffic Counting load balancer scheduler algorithm for mod_proxy_balancer
lbmethod_heartbeat	activé	**désactivé**	Heartbeat Traffic Counting load balancer scheduler algorithm for mod_proxy_balancer
proxy_ajp	activé	**désactivé**	AJP support module for mod_proxy
proxy_balancer	activé	**désactivé**	mod_proxy extension for load balancing
proxy_connect	activé	**désactivé**	mod_proxy extension for CONNECT request handling
proxy_express	activé	**désactivé**	Dynamic mass reverse proxy extension for mod_proxy
proxy_fcgi	activé	activé	FastCGI support module for mod_proxy
proxy_fdpass	activé	**désactivé**	fdpass external process support module for mod_proxy
proxy_ftp	activé	**désactivé**	FTP support module for mod_proxy
proxy_http	activé	**désactivé**	HTTP support module for mod_proxy
proxy_scgi	activé	**désactivé**	SCGI gateway module for mod_proxy
proxy_wstunnel	activé	**désactivé**	Websockets support module for mod_proxy

Activation du module *systemd* dans `conf.modules.d/00-systemd.conf`

Dans le fichier de configuration `/etc/httpd/conf.modules.d/00-systemd.conf`, on laissera activé le module *systemd* :

Module	Défaut	Etat désiré
systemd	activé	activé

Désactivation du module *CGI* dans `conf.modules.d/01-cgi.conf`

On *désactivera complètement* le fichier de config. `/etc/httpd/conf.modules.d/01-cgi.conf`.

Début de configuration pour mitiger les attaques *(D)DOS* dans `conf.d/ddos.conf`

Pour (commencer à) mitiger les attaques par déni de services (*DOS*, distribuées (*DDOS*) ou non), on ajoutera la configuration suivante dans le nouveau fichier `/etc/httpd/conf.d/ddos.conf` :

Code source 28: /etc/httpd/conf.d/ddos.conf

```
# DDOS mitigation
TimeOut 10

MaxKeepAliveRequests 500
KeepAliveTimeout 3

LimitRequestBody 10240
LimitXMLRequestBody 10240
```

À faire : *Apache httpd* : Poursuivre la mitigation des attaques (D)DOS !

À faire : *Apache httpd* : Où en étais-je de l'idée d'utiliser ou s'inspirer de *mod_evasive* ?

De tête (à vérifier !) le projet n'était plus développé. Même plus sûr qu'il fut dans les dépôts officiels ou alternatifs de *CentOS*. Ou alors il ne fonctionnait pas de manière naïve avec *MPM event* ?

Début de sécurisation additionnelle dans `conf.d/security.conf`

Pour (commencer à) augmenter sensiblement la sécurité de la configuration, on ajoutera les directives suivantes dans le fichier `/etc/httpd/conf.d/security.conf` :

Code source 29: /etc/httpd/conf.d/security.conf

```
# Add additional security-related directives
ServerTokens Prod

Header always unset "X-Powered-By"
Header unset "X-Powered-By"
Header always unset "Composed-By"
Header unset "Composed-By"
Header always set "X-Frame-Options" "deny"
Header always set "X-XSS-Protection" "1; mode=block"
Header always set "X-Content-Type-Options" "nosniff"
Header always set "Referrer-Policy" "no-referrer"
Header always set "Content-Security-Policy" "default-src 'none'; script-src 'self'
↪'unsafe-inline'; connect-src 'self'; img-src 'self'; style-src 'self' 'unsafe-inline
↪'; font-src 'self';"
```

et on empêchera de servir des fichiers temporaires et/ou de sauvegardes, qui n'ont normalement rien à faire sur le serveur :

Code source 30: /etc/httpd/conf.d/security.conf

```
# Remove access from (probably) temporary files and saved one,
# not intented to be served.
RedirectMatch 404 ".+(~|swp)$"
RedirectMatch 404 "(?i)/.+(old|orig|save(d)+|sove|nepasutiliser|ne_pas_utiliser|test).
↪*$"
```

Voir aussi :

Documentations externes d'intérêt :

- *OWASP* :
  - Le projet *Secure Headers Project* : https ://www.owasp.org/index.php/OWASP_Secure_Headers_Project
  - La page *Content Security Policy* : https ://www.owasp.org/index.php/Content_Security_Policy
- La documentation *MDN* chez Mozilla :
  - *Content Security Policy (CSP)* : https ://developer.mozilla.org/en-US/docs/Web/HTTP/CSP
  - *X-Content-Type-Options* : https ://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Content-Type-Options
  - *X-Frame-Options* : https ://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
  - *X-XSS-Protection* : https ://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-XSS-Protection
- Le site web *Content Security Policy Reference* : https ://content-security-policy.com/
- La page dédiée chez ZinoUI : https ://zinoui.com/blog/security-http-headers

MISC - TEMP - TODO

À faire : conf.d/vhost.conf

Configuration finale de *systemd*

On souhaite généralement que le démon tourne tout le temps, et dès maintenant :

```
$ sudo systemctl start httpd
```

5.12.4 Usage

Vérification de la bonne conformité syntaxique de tous les fichiers de configuration

```
$ sudo httpd -t
Syntax OK
```

Lister les modules actuellement chargés

```
$ sudo httpd -M  # ou : -t -D DUMP_MODULES
Loaded Modules :
 core_module (static)
 so_module (static)
 http_module (static)
 alias_module (shared)
 allowmethods_module (shared)
 auth_basic_module (shared)
 authn_core_module (shared)
 authn_file_module (shared)
```

(suite sur la page suivante)

(suite de la page précédente)

```
authn_socache_module (shared)
authz_core_module (shared)
authz_groupfile_module (shared)
authz_host_module (shared)
authz_user_module (shared)
deflate_module (shared)
dir_module (shared)
expires_module (shared)
headers_module (shared)
info_module (shared)
log_config_module (shared)
mime_magic_module (shared)
mime_module (shared)
negotiation_module (shared)
reqtimeout_module (shared)
rewrite_module (shared)
setenvif_module (shared)
slotmem_shm_module (shared)
socache_shmcb_module (shared)
status_module (shared)
unixd_module (shared)
mpm_event_module (shared)
proxy_module (shared)
proxy_fcgi_module (shared)
ssl_module (shared)
systemd_module (shared)
```

5.13 Mise à jour de *CentOS* (7.5 -> 7.6) sur serveur principal

5.13.1 Stopper les services

Arrêt et désactivation au démarrage, a priori dans cet ordre, des principaux services :

1. *SSHd*
2. *Apache httpd*
3. *PHP-FPM*
4. *MariaDB*
5. *fail2ban*

```
$ sudo systemctl stop sshd
$ sudo systemctl disable sshd
$ sudo systemctl stop httpd
$ sudo systemctl disable httpd
$ sudo systemctl stop php-fpm
$ sudo systemctl disable php-fpm
$ sudo systemctl stop mariadb
$ sudo systemctl disable mariadb
$ sudo systemctl stop fail2ban
$ sudo systemctl disable fail2ban
```

5.13.2 Supprimer les fichiers inutiles, caches notamment

Pour préparer les sauvegardes à venir, et minimiser leurs tailles, on supprimera les caches suivants :

1. *yum* :

```
# yum
$ sudo yum clean all
#   Eventuellemt, en fonction de la taille donnée par la commande suivante :
$ sudo du -h -s /var/cache/yum
76K    /var/cache/yum
#   supprimer /var/cache/yum :
$ sudo rm -Rf /var/cache/yum
```

2. *MariaDB* :

   Actuellement, *MariaDB* n'a pas son *Query Cache* d'activé (*query_cache_type* à `OFF`) :

```
$ sudo mysql -u root -p
> SHOW VARIABLES LIKE '%query_cache%';
+------------------------------+---------+
| Variable_name                | Value   |
+------------------------------+---------+
| have_query_cache             | YES     |
| query_cache_limit            | 1048576 |
| query_cache_min_res_unit     | 4096    |
| query_cache_size             | 1048576 |
| query_cache_strip_comments   | OFF     |
| query_cache_type             | OFF     |
| query_cache_wlock_invalidate | OFF     |
+------------------------------+---------+
```

3. *Apache httpd* :

   Actuellement, *Apache httpd* n'a pas de cache activé :

```
$ cd /etc/httpd/conf.modules.d
$ grep -r -e "cache" *.conf
00-base.conf :LoadModule authn_socache_module modules/mod_authn_socache.so
00-base.conf :#LoadModule cache_module modules/mod_cache.so
00-base.conf :#LoadModule cache_disk_module modules/mod_cache_disk.so
00-base.conf :#LoadModule socache_dbm_module modules/mod_socache_dbm.so
00-base.conf :#LoadModule socache_memcache_module modules/mod_socache_memcache.so
00-base.conf :LoadModule socache_shmcb_module modules/mod_socache_shmcb.so
```

4. anciens *kernels* :

   Supprimer les kernels inférieurs à n-1 (script maison, voir *projet sysutils*) :

```
$ sudo remove_old_kernels.sh
```

5.13.3 Sauvegarde intégrale de sécurité avec *CloneZilla*

Classique : graver la dernière version *stable*, basée sur Debian, de l'ISO, depuis le site officiel du projet, et opérer, pour bien sauvegarder l'intégralité du « disque » (i.e. toutes les partitions, leur table, MBR & Cie) sur un périphérique USB :

1. Boot par défaut proposée par l'ISO (via touche `F11` au démarrage du BIOS pour booter sur le CD)
2. Choisir sa langue (e.g. « français »)

3. Changer la disposition du clavier si besoin (US par défaut)
4. Démarrer CloneZilla
5. Choisir le mode *device-image*
6. Si sauvegarde sur périphérique USB externe local, choisir de monter *local_dev*
7. Brancher le périphérique USB externe, puis le sélectionner ; cela sera par ex. `/dev/sdb1` monté sous `/home/partimage`
8. Choisir ensuite le *mode expert* pour avoir plus de latitude dans le choix des options :
9. Choisir le mode `save_disk` (sauvegarde intégrale du disque)
10. Choisir le répertoire et le nom de l'image à sauvegarder
11. Choisir `sda` comme disque à sauvegarder
12. Choisir l'option `-q2` pour l'ordre des logiciels de sauvegarde à préférer : *partclone* > *partimage* > *dd*
13. Choisir les options suivantes :
    - `-c` : demande de confirmation
    - `-j2` : cloner les données cachéesentre MBR et 1ère partition
    - `-rescue` : continuer à opérer même en cas de survenue d'erreur
    - `-gS` : calculer les sommes de contrôle SHA1
    - `-z6` : compression par *LZip* parallèle
    - laisser le découpage par défaut en fichiers de taille fixe (4096 Mo)
    - `-fsck` : demander la vérification des systèmes de fichiers
    - vérifier que l'image est bien restaurable
    - `-senc` : ne pas chiffrer l'image de sauvegarde

5.13.4 Installation de la dernière version de ReaR

Voir aussi :

- *présentation de ReaR*
- *documentation d'installation et configuration*

```
$ sudo yum check-update
$ yum info rear
$ sudo yum install rear syslinux syslinux-extlinux genisoimage
```

5.13.5 Sauvegarde complète par ReaR

Méthode par création d'une ISO

Configuration de *ReaR* pour générer une ISO bootable restaurant complètement le système, sauvegardée sur une clef USB montée pour l'occasion par ex. sur `/mnt` :

1. Création de la configuration de *ReaR* :

Code source 31: /etc/rear/local.conf (ISO)

```
# write the rescue initramfs to an ISO
OUTPUT=ISO
# write ISO in following directory
OUTPUT_URL=file :///mnt/ServeurPrincipal-CentOS-7.5-ReaRed-20181206-1315
# backup data method
BACKUP=NETFS
# backup data directly into ISO
#   /!\ CAUTION /!\
#   This is not a configuration recommended by ReaR
#   (see e.g. https ://github.com/rear/rear/issues/581).
#   Also, this should work for a *small* whole filesystem...
BACKUP_URL=iso :///backup
```

2. Création de la sauvegarde :

```
$ sudo rear -v mkbackup
```

3. Sauvegarder l'ISO ainsi créée.

Méthode par création d'une clef USB bootable

Configuration de *ReaR* pour créer une clef USB bootable restaurant complètement le système, par ex. */dev/sdb* :

1. Création du système de fichier de la clef USB :

```
$ sudo rear format -v /dev/sdb
```

2. Création de la configuration de *ReaR* :

Code source 32: /etc/rear/local.conf (USB)

```
# write the rescue initramfs to an USB key
OUTPUT=USB
USB_DEVICE=/dev/disk/by-label/REAR-000
# backup data method
BACKUP=NETFS
# backup data directly into USB (CAUTION : all data will be erased!)
BACKUP_URL=usb :///dev/disk/by-label/REAR-000
```

3. Création de la sauvegarde :

```
$ sudo rear -v mkbackup
```

4. Sauvegarder le contenu de la clef USB ainsi formattée, sur un autre système, où elle sera par ex. reconnue sous le périphérique */dev/sdc* :

```
$ mkdir ~/rear_usb_rescue-$MACHINE-$DATE
$ cd ~/rear_usb_rescue-$MACHINE-$DATE
# fsarchiver options :
#    #1   is new archive
#    #2   is device partition
#    -v   is for more output verbosity
#    -j   is for setting number of threads (jobs) to run, depending
#         on your number of cores
#    -L   is for labelling archive with a short description
```

(suite sur la page suivante)

(suite de la page précédente)

```
#    -Z   is for setting Zstandart compression level (and I have no idea
#         if 16 is a "good" value)
$ sudo fsarchiver savefs ./rear_rescue.fsarchiv /dev/sdc1 -v -Z 16 -j8 -L
↪"Backup ReaR : ServeurPrincipal on CentOS 7.5 - 20181212"
```

5.13.6 Effectuer la mise à jour proprement dite

```
$ sudo yum check-update
$ sudo yum update
```

et redémarrer :

```
$ sudo shutdown -r now
```

S'assurer qu'il n'y a plus de mises à jour à appliquer, dans ce cas recommencer jusqu'à épuisement.

5.13.7 Redémarrer les services principaux

Redémarrage à la main et réactivation au démarrage, a priori dans cet ordre, des principaux services :

1. *fail2ban*
2. *SSHd*
3. *MariaDB*
4. *PHP-FPM*
5. *Apache httpd*

```
$ sudo systemctl start fail2ban
$ sudo systemctl enable fail2ban
$ sudo systemctl start sshd
$ sudo systemctl enable sshd
$ sudo systemctl start mariadb
$ sudo systemctl enable mariadb
$ sudo systemctl start php-fpm
$ sudo systemctl enable php-fpm
$ sudo systemctl start httpd
$ sudo systemctl enable httpd
```

5.13.8 Supprimer les anciens *kernels* (optionnel)

Si tout va bien et que l'on s'en est assuré, et que dans les mises à jour appliquée il y a eu montée de version du *kernel*, on pourra supprimer les anciens, grâce au script maison (voir *projet sysutils*) :

```
$ sudo remove_old_kernels.sh
```

5.13.9 Sauvegarder les sauvegardes

Il ne reste plus qu'à sauvegarder, sur différents supports et localisations, les artefacts de sauvegardes générés par *CloneZilla* et *ReaR*...

5.14 Mise à jour de l'instance Drupal du site de la Fresco par drush

5.14.1 Prérequis

Prestataire

compte *risc*

accès *SSH*

environnement

- *drush* (v.7.4) dans le *$PATH* fonctionnant pour l'installation par défaut de *PHP* sur l'hébergement mutualisé

Note : Versions de *PHP*, *Drupal* et *drush*

Actuellement, nous utilisons la dernière version de la branche 7 de *drush*, la seule compatible avec la version 7.x de *Drupal* et la version 5.3.x de *PHP*, version par défaut présente sur l'hébergement mutualisé proposé par *Prestataire*. Peut-être que l'on pourrait faire une montée de version de *PHP*, et alors utiliser la branche 8 de *drush*…

Par défaut, l'instance est servie par PHP en version 5.3, ce qui est aussi la version par défaut du binaire *php* accessible en ligne de commande. La dernière version de *drush* pour Drupal 7 compatible PHP 5.3 est la 7.4 de 2016-09-22 (https ://github.com/drush-ops/drush/releases/tag/7.4.0), d'après la documentation (http ://docs.drush.org/en/master/install/#drupal-compatibility). Cette version a été installée à la main sur le compte Prestataire *risc* : `/home/risc/drush-7.4.0`, et le binaire `drush` ajouté au *$PATH* (dans `~/.profile`).

À faire : Monter de version *PHP* pour l'instance *Drupal* de la *Fresco*, et utiliser alors *drush* 8 ?

Idéalement, il faudrait avoir la dernière version de la branche 8.x de *drush*, la dernière branche à supporter la branche 7.x de *Drupal*. Idéalement également, il faudrait une version de PHP récente, comme la branche 7.1, mais la comptabilité avec *Drupal* ne semble pas encore atteinte, ne serait-ce que pour la branche 7.0 (voir : https ://www.drupal.org/project/drupal/issues/2656548) ; cela pourrait cependant aussi poser problème pour certains modules… https ://www.drupal.org/docs/7/system-requirements/drupal-7-php-requirements semble indiquer que la valeur sûre pour *Drupal* 7 reste la branche 5.6 de *PHP*.

=> Faire probablement comme ce qui a été fait pour l'instance *Drupal* de la *Fondation Cognition* ?

5.14.2 Usage

Généralités

Note : Méthode lourdement inspirée de : https ://cvuorinen.net/2013/02/updating-a-drupal-multisite-using-drush/

On utilisera *drush* pour mettre à jour en ligne de commande l'instance *Drupal*.

Les opérations sont majoritairement à réaliser en shell distant (*SSH*) avec le compte **** :

```
$ ssh <COMPTE>@<NOM.DE.DOMAINE>
```

Note : Particularités de notre instance *Drupal*

- De tous les « sites » gérés par l'instance, seul celui de la *Fresco* correspond à une installation pleine et entière. Les opérations effectuées par *drush* seront donc majoritairement préfixées de `@fresco`.
- Il y a une typo. dans le nom du répertoire de l'instance *Drupal* : « drupal_asso**ca**tions » (sic !).

Vérifier s'il y a des mises à jour de disponibles

```
$ cd ~/drupal_assocations
$ drush @fresco pm-updatestatus
```

et lire la sortie à l'écran.

Mettre à jour (avec sauvegarde préalable) l'instance *Drupal*

La méthode consiste en :

1. réaliser une sauvegarde complète (base de données, fichiers)
2. mettre l'instance Drupal en « mode maintenance »
3. mettre à jour l'instance (modules, mise à jour de la base de données)
4. désactiver le « mode maintenance »
5. s'assurer que tout semble fonctionner

On enregistrera les données sauvegardées dans un nouveau répertoire, du jour, e.g. :

```
$ cd ~
$ BACKUP_DIR="$HOME/drupal-assos-backup-$(date +%Y%m%d-%H%M)"
$ mkdir $BACKUP_DIR
```

Réaliser une sauvegarde complète du système

La sauvegarde complète de l'instance *Drupal* se réalise en opérant les sauvegardes conjointes des bases de données et des fichiers :

```
$ cd ~/drupal_assocations
$ drush @fresco sql-dump --gzip --result-file=$BACKUP_DIR/database.sql.gz
$ cd $BACKUP_DIR
$ tar zcvf drupal_files.tar.gz ~/drupal_assocations
```

Mettre l'*instance Drupal* en « mode maintenance »

```
$ cd ~/drupal_assocations
$ drush @fresco variable-set maintenance_mode 1
$ drush @fresco cache-clear all
```

Mettre à jour le code de *Drupal* et des modules

```
$ cd ~/drupal_assocations
$ drush @fresco pm-update
```

Mettre à jour la base de données en conformité avec les nouvelles versions des modules

```
$ cd ~/drupal_assocations
$ drush @fresco updatedb
```

Bonus : Mettre à jour les traductions des modules

```
$ drush @fresco l10n-update-refresh
$ drush @fresco l10n-update-status
# au besoin, mettre à jour les traductions françaises
$ drush @fresco l10n-update --languages=fr
```

Désactiver le « mode maintenance »

```
$ cd ~/drupal_assocations
$ drush @fresco variable-set maintenance_mode 0
$ drush @fresco cache-clear all
```

S'assurer que tout semble « ok » à travers l'interface web

Se logguer en tant qu'utilisateur `****` sur `https ://****` puis se rendre sur la page d'administration `https ://****`.

5.15 Mise à jour de l'instance Drupal du clone raté du site de la Fondation Cognition par drush

Prestataire

compte *risc2*

accès *SSH*

environnement

- *drush-fondation* (une version de *drush* en 8.1.x) dans le *$PATH* fonctionnant pour la version de *PHP* utilisée par le site web de l'instance (5.6.x).

À faire : Upgrader la version de *drush* pour la *Fondation Cognition*!

5.15.1 Usage

Généralités

Note : Méthode lourdement inspirée de : https ://cvuorinen.net/2013/02/updating-a-drupal-multisite-using-drush/

On utilisera *drush* pour mettre à jour en ligne de commande l'instance *Drupal*.

Les opérations sont majoritairement à réaliser en shell distant (*SSH*) avec le compte *<COMPTE>* :

```
$ ssh <COMPTE>@<NOM.DE.DOMAINE>
```

Vérifier s'il y a des mises à jour de disponibles

```
$ cd ~/sites/fondation-cognition/public
$ drush-fondation pm-updatestatus
```

et lire la sortie à l'écran.

Mettre à jour (avec sauvegarde préalable) l'instance *Drupal*

La méthode consiste en :

1. réaliser une sauvegarde complète (base de données, fichiers)
2. mettre l'instance Drupal en « mode maintenance »
3. mettre à jour l'instance (modules, mise à jour de la base de données)
4. désactiver le « mode maintenance »
5. s'assurer que tout semble fonctionner

On enregistrera les données sauvegardées dans un nouveau répertoire, du jour, e.g. :

```
$ cd ~
$ FONDATION_DIRPATH="$HOME/sites/fondation-cognition"
$ FONDATION_DRUPAL_DIRPATH="$FONDATION_DIRPATH/public"
$ BACKUP_DIR="$FONDATION_DIRPATH/drupal-backup-$(date +%Y%m%d-%H%M)"
$ mkdir $BACKUP_DIR
```

Réaliser une sauvegarde complète du système

La sauvegarde complète de l'instance *Drupal* se réalise en opérant les sauvegardes conjointes des bases de données et des fichiers :

```
$ cd $FONDATION_DRUPAL_DIRPATH
$ drush-fondation sql-dump --gzip --result-file=$BACKUP_DIR/database.sql.gz
$ cd $BACKUP_DIR
$ tar zcvf drupal_files.tar.gz $FONDATION_DRUPAL_DIRPATH
```

Mettre l'*instance Drupal* en « mode maintenance »

```
$ cd $FONDATION_DRUPAL_DIRPATH
$ drush-fondation variable-set maintenance_mode 1
$ drush-fondation cache-clear all
```

Mettre à jour le code de *Drupal* et des modules

```
$ cd $FONDATION_DRUPAL_DIRPATH
$ drush-fondation pm-update
```

Mettre à jour la base de données en conformité avec les nouvelles versions des modules

```
$ cd $FONDATION_DRUPAL_DIRPATH
$ drush-fondation updatedb
```

Bonus : Mettre à jour les traductions des modules

```
$ cd $FONDATION_DRUPAL_DIRPATH
$ drush-fondation l10n-update-refresh
$ drush-fondation l10n-update-status
# au besoin, mettre à jour les traductions françaises
$ drush-fondation l10n-update --languages=fr
```

Désactiver le « mode maintenance »

```
$ cd $FONDATION_DRUPAL_DIRPATH
$ drush-fondation variable-set maintenance_mode 0
$ drush-fondation cache-clear all
```

À faire : S'assurer que tout semble « ok » à travers l'interface web (après mise àjour de l'instance *Drupal* de la *Fondation Cognition*)

CHAPITRE 6

Annexes

6.1 Aperçu des besoins en infrastructure – document du 2019-03-01 à l'attention du *DAS INSB*

Note :

- Extrait de la version partagée par D.U. dans son mail du 2019-03-01 14 :58.
- Ce document dresse une liste de pistes technologiques à étudier, et n'est pas à voir comme définitive (ni complète).
- Adaptations depuis la version transmise par Vincent.

	pre-2017	état actuel	futurs
O.S.	CentOS 5 ou 6 (x86)	CentOS 7 (amd64)	idem, ou Debian stable (amd64)
Sécurisation	(~ inexistante)	• fail2ban • SELinux	idem, plus : • mod_evasive (ou conf. équiv.) • mod_security (ou conf. équiv.)
Serveur web	Apache httpd 2.2	• Apache httpd 2.4 • HTTPS	• nginx avec HTTPS • si besoin : Apache httpd 2.x • cache (Varnish ?)
SGBD	• MySQL 5 • FileMaker Pro 5	• MariaDB 10.1 • FileMaker Pro 5	• MariaDB 10.x ou PostgreSQL • Redis • si besoin : orienté graphes (OrientDB ?) • moteurs de recherche (ElasticSearch ?)
Languages web	PHP 5.3, via mod_php	PHP 7.2, via PHP-FPM	• PHP 7.x • si besoin : Python 3.x
Sites web	PHP maison old school		CMS (Drupal 8.x ?)
Monitoring et alertes	(inexistante)		• netdata • Sensu go (aka 5.x) • grafana • Prometheus (?)
Virtualisation ou containerisation	(inexistante)	(inexistante)	si encore à notre charge : • LXC / LXD • Docker (?)
Sauvegardes	(manuelle, et script maison régulier)	(manuelle) • ReaR • etckeeper	• etckeeper • borg-backup
Gestion des logs	(inexistante)		[Centralisée, plus alertes et corrélations] • fluentd • graylog (?) • stack E.L.K. (?)

	pre-2017	état actuel	futurs
Annuaires	(inexistants)		• LDAP : 389-ds (?)
Mailing-listes	externalisées : CNRS, SPI-ENS		nos propres instances SYMPA
Forge logicielle	(inexistant)	GitLab externalisé (CRI-ENS), non mis à jour	nos propres forges légères, e.g. : gitea
Gestion collaborative de secrets (mots de passe, clefs SSH, certificats, etc.)	(manuelle, sécurité douteuse)		Outil web commun, e.g. : Vault de HashiCorp ?
Wiki	(inexistant)	(doc. HTML & PDF en attendant mieux)	nos propres instances de DokuWiki
Partage de fichiers	sur serveur de fichiers	idem + dépôts git	• dépôts git • notre propre instance de e.g. NextCloud
Gestion de configuration	(manuelle)	manuelle + • documentation • etckeeper + gitolite	• etckeeper + gitolite • documentation • fully automatisée (recettes Ansible ? Chef ? Salt ? Puppet ? autre ?)
Autres points	architecture monolithique et mono-serveur		• architecture en micro-services (?), permettant l'utilisation conjointe de plusieurs piles technos. différentes (langages, SGBD, versions, etc.) • API JSON (ReSTful ?)

6.2 Bibliographie générale

6.2.1 Livres

- *UNIX and Linux System Administration Handbook* d'Evi Nemeth, Garth Snyder, Trent R. Hein, Ben Whaley et Dan Mackin, *Addison-Wesley* (2017, 5th ed.) – http ://www.informit.com/store/unix-and-linux-system-administration-handbook-9780134277554
- *Pro Linux System Administration* de Dennis Matotek, James Turnbull & Peter Lieverdink, *Apress* (2017, 2nd ed.) – https ://www.apress.com/fr/book/9781484220078

- *The Debian Administrator's Handbook* de Raphaël Hertzog & al. – https ://debian-handbook.info/
  - consultation en ligne de la dernière version stable : https ://debian-handbook.info/browse/stable/
  - consultation des sources à leur dernière version : https ://salsa.debian.org/hertzog/debian-handbook
- *Linux Pocket Guide* de Daniel Barrett, *O'Reilly* (2016, 3rd ed.) – http ://shop.oreilly.com/product/0636920040927.do

6.2.2 Sites web

- *wiki de CentOS* : https ://wiki.centos.org/ notamment :
  - *Tips and Tricks* : https ://wiki.centos.org/TipsAndTricks
  - *HowTos* : https ://wiki.centos.org/HowTos

6.3 Glossaire

6.3.1 Paysage institutionnel

Ecole Normale Supérieure

ENS

Département d'Etudes Cognitives

DEC Etablissement d'enseignement supérieur hébergeant physiquement le *RISC*, au sein du *DEC*, son « bureau » et ses machines. Concernant le *SI* du *RISC*, les services de l'ENS avec lesquels nous avons à faire sont principalement le *CRI* (*Centre de Ressources en Informatique*) et le *SPI*.

Fondation Cognition Fondation pour « favoriser le dialogue et le rapprochement entre tous les acteurs des sciences de la cognition ». *Jean Lorenceau* fait partie du comité de pilotage de la Fondation. Il y a eu envie, durant l'été 2018, que son site web soit hébergé par le RISC, au risque sinon de le voir disparaitre, avec comme résultat un clone parcellaire de l'*instance Drupal*.

Site web : http ://fondation-cognition.fr/

Fresco La *Fédération Française des étudiants et jeunes actifs en Sciences de la Cognition* (*Fresco*) est une association française, dont nous hébergeons le site web.

6.3.2 Machines physiques faisant office de serveurs

serveur de fichiers **** faisant office de *serveur de fichiers* (documents partagés) et *serveur de bureau à distance* (essentiellement pour les *bases FileMaker Pro*). Son système d'exploitation n'est plus mis à jour, pour permettre de continuer à utiliser une vieille version de *FileMaker*.

À faire : Détailler / fiche dédiée sur ****

À faire : Alerte de sécu. sur **** ? (O.S. non mis à jour, accès (à vérifier) à l'extérieur)

---

serveur secondaire Vieux serveur DELL rackable, situé en *salle serveur* du *CRI* au 29, de config. identique à *serveur principal*. Il n'est quasi pas utilisé (précédemment essentiellement pour faire des sauvegardes – ce qui n'est plus le cas actuellement). Il est relié à une baie de stockage de ****

disques (soit **** To utilisables – RAID 1 oblige), dont le contenu ne nous intéresse (en gros) plus trop.

À faire : Détailler / fiche dédiée sur *serveur secondaire*

À faire : *serveur secondaire* a probablement des erreurs sur ses barettes mémoire (à changer) et sur ses disques durs internes (à changer), et (je crois) des batteries de cartes RAID à changer également.

À faire : *serveur secondaire*, une fois les pbs matériels réglés, mériterait le meme traitement que *serveur principal* : mise à jour des firmwares, installation et configuration d'une version récente de *CentOS*, etc.

---

serveur principal Vieux serveur DELL rackable, situé ****, de config. identique à *serveur secondaire* – mais sans baie de stockage dédiée. C'est le serveur principal du *RISC*, qui héberge les sites web et bases de données suivants : **** (dont : import des contenus des bases depuis *FileMaker*/*serveur de fichiers*, etc.).

Configuration *Configuration du serveur principal*

À faire : Détailler / fiche dédiée sur *serveur principal*

NAS *NAS* principalement utilisé pour sauvegarder les ordinateurs *Mac* via *TimeMachine*.

À faire : Alerte de sécu. sur *NAS*? (O.S. non mis à jour, accès (à vérifier) à l'extérieur)

6.3.3 Partenaires

Prestataire Notre principal prestataire non institutionnel, français. Nous avons de nombreux services chez eux : ****, etc. Ils proposent aussi de l'hébergement *virtualisé*, du *dédié*, et peuvent faire office de *Registrar*.

Ressources :

- Site web : https ://www.example.com/
- Blog : https ://blog.example.com/
- Documentation : https ://help.example.com/
- Interface web d'administration : https ://admin.example.com/

Offre de Service CNRS

Offre de Service

ODS CNRS

ODS L'*Offre de Service* (*ODS*) du CNRS, à destination de ses unités, comprend notamment l'hébergement web, que ce soit en mutualisé ou en machines virtuelles.

Site web de l'ODS : https ://www.ods.cnrs.fr/

Service de Prestations Informatiques

SPI

Jacques Beigbeder Le *SPI* est un service de l'*ENS* (i.e. J.B.) qui gère notamment **** pour le *RISC*, etc., mais aussi le *serveur NTP* `ntp.ens.fr`. Jacques Beigbeder est également RSSI de l'ENS, et correspondant logiciel.

6.3.4 Ecosystème CentOS

CentOS

Red Hat Enterprise Linux

RHEL *CentOS* est une distribution Linux, fork communautaire de *Red Hat Entreprise Linux* (*RHEL*), utilisée comme système d'exploitation de nos deux serveurs *serveur principal* et *serveur secondaire*.

dépôt RPM

dépôt de paquets RPM

dépôt alternatif de paquets RPM Liste des dépôts de *paquets RPM* pour *CentOS* : https ://wiki.centos.org/AdditionalResources/Repositories. Pour *serveur principal* et *serveur secondaire*, nous utilisons essentiellement *EPEL* et *IUS*.

EPEL

Extra Packages for Enterprise Linux Probablement le dépôt alternatif de *paquets RPM* le plus connu et utilisé, parmi *ceux listés par le projet CentOS*.

IUS

IUS Community Repo

Inline with Upstream Stable Un *dépôt alternatif de paquets RPM* se voulant plus à jour / récents que ceux de *RHEL* et/ou *CentOS*, et alignés sur les développements « upstream ». Nous l'utilisons particulièrement pour avoir des versions récentes de *PHP*, *MariaDB*, *git*, etc.

Ressources :

- Site web du projet : https ://ius.io/

RPM Package Manager

RPM

paquets RPM

paquet RPM Système de gestion des *paquets*, et format de ceux-ci, pour les distributions *Linux* de la famille *RedHat*. Pour *CentOS* 7, on utilise généralement l'utilitaire *yum* pour leurs installations, suppressions et mises à jour.

6.3.5 Logiciels et protocoles

badblocks *badblocks* est un utilitaire en ligne de commande qui, comme son nom le suggère, permet d'effectuer des recherche de blocs défectueux sur des disques durs (magnétiques).

Base de connaissance *présentation de "badblocks"*

How-to *badblocks : Usage*

Chrony Logiciel faisant office de *démon client NTP*. C'est le logiciel installé par défaut sous *RHEL* / *CentOS* pour cette tâche.

Base de connaissance *présentation de chrony*

How-to *documentation d'installation et configuration*

Drupal *Content Management System* (*CMS*) libre et gratuit, développé en *PHP*, parmi les plus utilisés du marché (avec *Wordpress* et *Joomla*). Au *RISC*, nous l'utilisons pour le site web de la *Fresco*,

ainsi que pour la tentative (inaboutie) d'hébergement du site web de la *Fondation Cognition*. Une *instance Drupal* peut très facilement s'administrer via la ligne de commande par *drush*.

Site web : https ://www.drupal.org/

À faire : Faire probablement une section dédiée à *Drupal*.

---

drush Utilitaire en ligne de commande pour administrer des instances *Drupal*. Nous l'utilisons notamment pour *mettre à jour l'instance de la Fresco*.

Ressources :

- Site web : http ://www.drush.org/
- Liste des commandes : https ://drushcommands.com/

etckeeper Logiciel pour enregistrer la configuration d'un OS Linux, spécifiquement le dossier `/etc`, dans un dépôt de version – usuellement *git*.

Base de connaissance *présentation de etckeeper*

How-to *documentation d'installation et configuration*

fail2ban-utils Le projet *fail2ban-utils* se voulait un complément à *fail2ban*. Son développement, qu'aux débuts, a été arrêté.

Voir *la section dévolue au projet dans la présente documentation*.

fsck

fsck.ext4 *fsck* est un utilitaire en ligne de commande permettant de vérifier, de manière plus ou moins approfondie, l'état d'intégrité des données de partitions d'un disque dur.

Base de connaissance *présentation de "fsck"*

How-to *fsck : Usage*

git Logiciel de gestion de versions, libre et gratuit.

Base de connaissance *présentation de git*

How-to *documentation d'installation et configuration*

gitolite Logiciel pour gérer de manière très fine les accès utilisateurs à un ensemble de dépôts *git*, à travers un dépôt *git*.

Base de connaissance *présentation de gitolite*

How-to *documentation d'installation et configuration*

Apache

Apache httpd

httpd

HTTP

HTTPS *Apache httpd* est LE *serveur web* de la *Fondation Apache*, probablement le plus connu et fut un temps le plus utilisé (ce qui explique que, par raccourci, on l'appelle communément « Apache »). Un *serveur web* est un *démon* (logiciel tournant en tâche de fond) qui répond aux *requêtes* du protocole *HTTP* (et *HTTPS*) par des *réponses* (*HTTP*/*HTTPS*), généralement sur le port `80` (resp. `443`). C'est celui que nous utilisons pour *serveur principal* et *serveur secondaire* afin qu'ils servent leurs différents sites web respectifs.

Base de connaissance *présentation d'Apache "httpd"*

How-to *documentation d'installation et configuration*

OpenSSH

SSH

ssh-agent

ssh-keygen Probablement l'implémentation du protocole *SSH* la plus répandue dans le monde *Linux*. *ssh-keygen* permet de créer des bi-clefs pour l'utilisation de *SSH*. *ssh-agent* permet de choisir et garder en mémoire un jeu de clefs spécifique.

OpenSSH

Base de connaissance *présentation d'OpenSSH*

How-to *documentation de configuration*

ssh-agent

Base de connaissance *présentation de ssh-agent*

How-to *exemple basique d'utilisation de ssh-agent*

Network Time Protocol

NTP *Protocole réseau* permettant à des ordinateurs *clients* de se synchroniser sur une heure de référence, fournie par un ou plusieurs *serveurs*. Sous *CentOS 7*, le *client* est actuellement *Chrony*.

ReaR

Relax-and-Recover Logiciel de sauvegarde de système complet, très simple d'utilisation.

Base de connaissance *présentation de ReaR*

How-to *documentation d'installation et configuration*

smartmontools

smartctl

S.M.A.R.T. *smartmontools* est une suite logicielle permettant de relever et effectuer des tests approfondis sur les données *S.M.A.R.T.* d'un disque dur, via son utilitaire en ligne de commande `smartctl`.

Base de connaissance *présentation de "smartmontools"*

How-to *smartmontools : Usage basique*

sudo

sudoers `sudo` est un utilitaire permettant, si un utilisateur y est autorisé, d'exécuter une commande en escaladant temporairement ses privilèges à ceux de `root`. Sa configuration peut être très fine, et se réalise dans le fichier `sudoers`.

Base de connaissance *présentation de "sudo"*

How-to *documentation de configuration de "sudo"*

sysutils Petit projet de développement à côté, regroupant des *scripts shell* pour faciliter au quotidien quelques tâches d'administration.

Voir *la section dévolue au projet dans la présente documentation*.

6.4 ToDo-list

6.4.1 Rappel automatique de *tous* les items « todo »

À faire : Détailler / fiche dédiée sur ****

(l'entrée originale se trouve dans /vagrant/src/annexes/glossaire.rst, à la ligne 47)

À faire : **Alerte de sécu.** sur ****? (O.S. non mis à jour, accès (à vérifier) à l'extérieur)

(l'entrée originale se trouve dans /vagrant/src/annexes/glossaire.rst, à la ligne 49)

À faire : Détailler / fiche dédiée sur *serveur secondaire*

(l'entrée originale se trouve dans /vagrant/src/annexes/glossaire.rst, à la ligne 60)

À faire : *serveur secondaire* a probablement des erreurs sur ses barettes mémoire (à changer) et sur ses disques durs internes (à changer), et (je crois) des batteries de cartes RAID à changer également.

(l'entrée originale se trouve dans /vagrant/src/annexes/glossaire.rst, à la ligne 62)

À faire : *serveur secondaire*, une fois les pbs matériels réglés, mériterait le meme traitement que *serveur principal* : mise à jour des firmwares, installation et configuration d'une version récente de *CentOS*, etc.

(l'entrée originale se trouve dans /vagrant/src/annexes/glossaire.rst, à la ligne 64)

À faire : Détailler / fiche dédiée sur *serveur principal*

(l'entrée originale se trouve dans /vagrant/src/annexes/glossaire.rst, à la ligne 76)

À faire : **Alerte de sécu.** sur *NAS*? (O.S. non mis à jour, accès (à vérifier) à l'extérieur)

(l'entrée originale se trouve dans /vagrant/src/annexes/glossaire.rst, à la ligne 82)

À faire : Faire probablement une section dédiée à *Drupal*.

(l'entrée originale se trouve dans /vagrant/src/annexes/glossaire.rst, à la ligne 207)

À faire : S'assurer que les serveurs NTP de centos.pool.ntp.org sont bien pris en compte ! Pb. d'ouverture du firewall ? ?

(l'entrée originale se trouve dans /vagrant/src/howto/chrony.rst, à la ligne 50)

À faire : Upgrader la version de *drush* pour la *Fondation Cognition* !

(l'entrée originale se trouve dans /vagrant/src/howto/drupal-Fondation-update.rst, à la ligne 14)

À faire : S'assurer que tout semble « ok » à travers l'interface web (après mise àjour de l'instance *Drupal* de la *Fondation Cognition*)

(l'entrée originale se trouve dans /vagrant/src/howto/drupal-Fondation-update.rst, à la ligne 136)

À faire : Monter de version *PHP* pour l'instance *Drupal* de la *Fresco*, et utiliser alors *drush* 8 ?

Idéalement, il faudrait avoir la dernière version de la branche 8.x de *drush*, la dernière branche à supporter la branche 7.x de *Drupal*. Idéalement également, il faudrait une version de PHP récente, comme la

branche 7.1, mais la comptabilité avec *Drupal* ne semble pas encore atteinte, ne serait-ce que pour la branche 7.0 (voir : https ://www.drupal.org/project/drupal/issues/2656548) ; cela pourrait cependant aussi poser problème pour certains modules… https ://www.drupal.org/docs/7/system-requirements/drupal-7-php-requirements semble indiquer que la valeur sûre pour *Drupal* 7 reste la branche 5.6 de *PHP*.

=> Faire probablement comme ce qui a été fait pour l'instance *Drupal* de la *Fondation Cognition* ?

(l'entrée originale se trouve dans /vagrant/src/howto/drupal-Fresco-update.rst, à la ligne 36)

À faire : *firewalld* : Y a-t-il besoin, et si oui expliquer, d'autoriser et le service HTTP (resp. HTTPS) **ET** le port 80 (resp. 443) ?

(l'entrée originale se trouve dans /vagrant/src/howto/httpd.rst, à la ligne 42)

À faire : *Apache httpd* : Poursuivre la mitigation des attaques (D)DOS !

(l'entrée originale se trouve dans /vagrant/src/howto/httpd.rst, à la ligne 341)

À faire : *Apache httpd* : Où en étais-je de l'idée d'utiliser ou s'inspirer de *mod_evasive* ?

De tête (à vérifier !) le projet n'était plus développé. Même plus sûr qu'il fut dans les dépôts officiels ou alternatifs de *CentOS*. Ou alors il ne fonctionnait pas de manière naïve avec *MPM event* ?

(l'entrée originale se trouve dans /vagrant/src/howto/httpd.rst, à la ligne 343)

À faire : conf.d/vhost.conf

(l'entrée originale se trouve dans /vagrant/src/howto/httpd.rst, à la ligne 415)

À faire : `$ sudo smartctl -l xselftest,1 /dev/sdX` ? ?

(l'entrée originale se trouve dans /vagrant/src/howto/smartmontools.rst, à la ligne 58)

À faire : OpenSSH 8.0 recommande une longueur de 3072 bits pour *RSA*

Voir l'entrée dans son changelog : https ://www.openssh.com/txt/release-8.0

> « Increase the default RSA key size to 3072 bits, following NIST Special Publication 800-57's guidance for a 128-bit equivalent symmetric security level. »

(l'entrée originale se trouve dans /vagrant/src/kb/openssh.rst, à la ligne 44)

CHAPITRE 7

Autres sections

- genindex
- modindex
- search

Index