Administration Systèmes & Réseaux

Administration des services réseaux

May 3, 2015

Houcemeddine HERMASSI

houcemeddine.hermassi@enit.rnu.tn

École Nationale d'Ingénieurs de Carthage ENI-CAR
Université Carthage
Tunisie

Plan de cour

Monter un serveur DHCP

Serveur DNS

Courrier électronique

Service HTTP Apache

Partage de fichiers

Samba

Présentation

Le service **DHCP** (Dynamic Host Configuration Protocol), protocole de configuration dynamique des hôtes, permet aux hôtes d'un réseau de demander et recevoir des informations de configuration (adresse, routage, DNS, etc.). Il y a en général un seul serveur DHCP par segment de réseau même si plusieurs sont possibles. Si le serveur est sur un autre segment, on peut utiliser un agent de retransmission DHCP.
Autrement dit, un client DHCP recherche tout seul un serveur DHCP qui lui communiquera son adresse IP. L'adresse IP est assignée soit dynamiquement à partir de plages d'adresses prédéfinies, soit statiquement en fonction de l'adresse MAC du demandeur. Les informations sont valables un laps de temps donné (un bail) qui peut être renouvelé et configurable.
DHCP est un sur-ensemble de **BOOTP** (Bootstrap Protocol). Quand le client cherche à contacter un serveur, c'est BOOTP qui fournit les informations d'adressage. DHCP gère les renouvellements. BOOTP se base sur le protocole de transport UDP.
Un hôte n'a aucune information réseau disponible au démarrage. Il doit trouver seul un serveur DHCP. Pour cela, BOOTP effectue un broadcast sur l'IP 255.255.255.255 avec une trame contenant ses informations (comme son adresse MAC) et les informations souhaitées (type de requête, ici DHCPDISCOVER, port de connexion, etc.). Le broadcast est envoyé par définition à tous les hôtes du réseau local. Quand le serveur DHCP détecte la trame, il effectue lui aussi un broadcast (l'hôte client n'a pas encore d'IP) avec les informations de base souhaitées par l'hôte (DHCPOFFER, premiers paramètres). L'hôte établit une première configuration puis demande confirmation de l'IP (DHCPREQUEST). Le serveur DHCP confirme (DHCPHACK). Le bail est confirmé et le client dispose dès lors de toutes les informations valides.

Monter un serveur DHCP
Serveur DHCP

Démarrage

Le serveur dhcpd est un service (daemon) lancé à l'aide d'un script (/etc/init.d/dhcpd). Il est configuré à l'aide du fichier /etc/dhcpd.conf. Les adresses IP allouées sont placées dans /var/lib/dhcp/dhcpd.leases.

```
# service dhcpd start "
```

Ou

```
# /etc/init.d/dhcpd start "
```

Coté client

Sous Linux et les distributions de type Red Hat, Fedora, Mandriva, openSUSE, etc., modifiez le fichier `/etc/sysconfig/network-script/ifcfg-xxx` en spécifiant dhcp pour **BOOTPROTO** et relancez la connexion réseau (**ifdown** puis **ifup**).
Le client dhcpcd permet d'activer dhcp sur une interface réseau. La méthode la plus simple est, par exemple pour eth0 :

```
# dhcpcd eth0 &"
```

Vous pouvez transmettre des options à dhcpcd pour prendre en charge diverses possibilités. Parmi ces options :

- `-D` : autorise la modification du nom de domaine
- `-H` : autorise la modification du nom d'hôte
- `-R` : évite l'écrasement du fichier resolv.conf
- `-l` : permet de modifier le leasetime (en secondes).

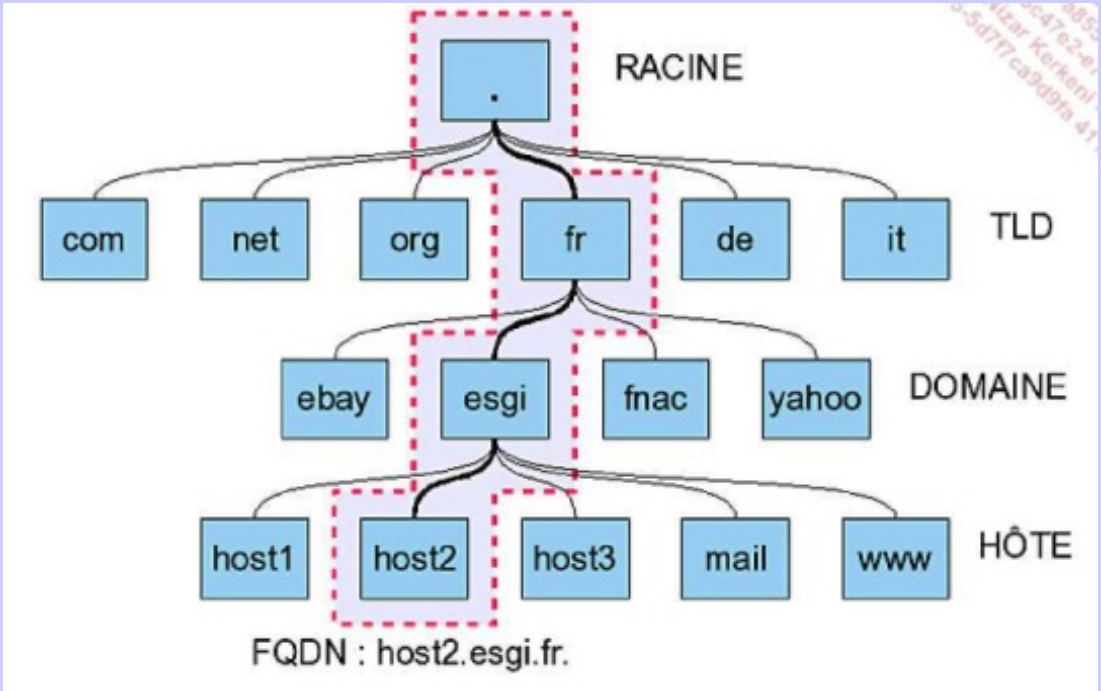
```
# dhcpcd -D -H -l 86400 eth0"
```

Présentation

Le Système de Noms de Domaines **DNS** (Domain Name System) transforme les noms d'hôte en adresses IP : c'est la **résolution de nom**. Il transforme les adresses IP en noms d'hôte : c'est la résolution inverse. Il permet de regrouper les machines par domaines de nom. Il fournit des informations de routage et de courrier électronique.
Le DNS permet de faire référence à des systèmes basés sur IP (les hôtes) à l'aide de noms conviviaux (les noms de domaines). L'intérêt d'un DNS est évident. Les noms de domaine sont plus simples à retenir, et si son adresse IP change l'utilisateur ne s'en rend même pas compte. On comprend que le DNS est un service clé critique pour Internet.
Les noms de domaine sont séparés par des points, chaque élément pouvant être composé de 63 caractères il ne peut y avoir qu'un maximum de 127 éléments et le nom complet ne doit pas dépasser 255 caractères. Le nom complet non abrégé est appelé **FQDN** (Fully Qualified Domain Name). Dans un FQDN, l'élément le plus à droite est appelé **TLD** (Top Level Domain), celui le plus à gauche représente l'hôte et donc l'adresse IP.

Présentation

Le DNS contient une configuration spéciale pour les routeurs de courrier électronique (définitions MX) permettant une résolution inverse, un facteur de priorité et une tolérance de panne.



Une zone est une partie d'un domaine gérée par un serveur particulier. Une zone peut gérer un ou plusieurs sous- domaines, et un sous-domaine peut être réparti en plusieurs zones. Une zone représente l'unité d'administration dont une personne peut être responsable.

Lancement

Le service s'appelle **named**.

```
# service named start"
```

Configuration de Bind

Bind (Berkeley Internet Name Daemon) est le serveur de noms le plus utilisé sur Internet. Bind 9 supporte l'IPv6, les noms de domaine unicode, le multithread et de nombreuses améliorations de sécurité.

Configuration générale

La configuration globale de Bind est placée dans le fichier `/etc/named.conf`. La configuration détaillée des zones est placée dans /var/lib/named. /etc/named.conf est composé de deux parties. La première concerne la configuration globale des options de Bind. La seconde est la déclaration des zones pour les domaines individuels. Les commentaires commencent par un # ou //.

```
# cat /etc/sysconfig/named "
```

Dans ce cas, tous les fichiers de configuration, y compris les zones, sont relatifs à ce chemin (Voir un fichier named.conf de base)

Section globale

La configuration globale est placée dans la section options. Voici un détail de quelques options importantes (le point- virgule doit être précisé) :

- **directory "filename"** : emplacement des fichiers contenant les données des zones.
- **forwarders {adresse-ip }** : si le serveur bind ne peut résoudre lui-même la requête, elle est renvoyée à un serveur DNS extérieur, par exemple celui du fournisseur d'accès.
- **listen-on-port 53 {127.0.0.1 adresse-ip }** : port d'écoute du DNS suivi des adresses d'écoute. On indique ici les adresses IP des interfaces réseau de la machine. Il ne faut pas oublier 127.0.0.1.
- **allow-query { 127.0.0.1 réseau }** : machine(s) ou réseau(x) autorisés à utiliser le service DNS. Par exemple 192.168.1/24. Si la directive est absente, tout est autorisé.
- **allow-transfer { 192.168.1.2 }** : machine(s) ou réseau(x) autorisés à copier la base de données dans le cas d'une relation maître et esclave. Par défaut aucune copie n'est autorisée.
- **notify no** : on notifie ou non les autres serveurs DNS d'un changement dans les zones ou d'un redémarrage du serveur.

Section de zones

Pour chaque domaine ou sous-domaine, on définit deux sections **zone**. La première contient les informations de résolution de nom (nom vers IP) et la seconde les informations de résolution inverse (IP vers Nom). Dans chacun des cas, la zone peut être maître **Master** ou esclave **Slave** :

- **Master** : le serveur contient la totalité des enregistrements de la zone dans ses fichiers de zone. Lorsqu'il reçoit une requête, il cherche dans ses fichiers (ou dans son cache) la résolution de celle-ci.
- **Slave** : le serveur ne contient par défaut aucun enregistrement. Il se synchronise avec un serveur maître duquel il récupère toutes les informations de zone. Ces informations peuvent être placées dans un fichier. Dans ce cas l'esclave stocke une copie locale de la base. Lors de la synchronisation, le numéro de série de cette copie est comparé à celui du maître. Si les numéros sont différents, une nouvelle copie a lieu, sinon la précédente continue à être utilisée.

Zone de résolution

Elle est généralement appelée **zone**. Pour chaque domaine ou sous-domaine, elle indique dans quel fichier sont placées les informations de la zone (c'est-à-dire et entre autres les adresses IP associées à chaque hôte), son type (maître ou esclave), si on autorise ou non la notification, l'adresse IP du serveur DNS maître dans le cas d'un esclave, etc.
Le nom de la zone est très important puisque c'est lui qui détermine le domaine de recherche. Quand le DNS reçoit une requête, il recherche dans toutes les zones une correspondance.

```
zone "domaine.org"
type "master";
file "domaine.org.zone";
;
```

- **type** : master ou slave
- **file** : nom du fichier qui contient les informations de la zone. Il n'y a pas de règles précises de nommage mais pour des raisons de lisibilité il est conseillé de lui donner le même nom que la zone tant pour une zone master que pour une slave. Pour un master, c'est l'original éventuellement rempli par vos soins. Pour un slave, ce n'est pas obligatoire. S'il est présent, ce sera une copie du master, synchronisée.
- Dans le cas d'un Master, on peut rajouter **allow-transfer** (serveurs autorisés à dupliquer la zone) et **notify yes** (indique une mise à jour ou une relance pour les slaves).

En cas de Slave : on rajoute la directive **masters** pour indiquer à partir de quel serveur Master dupliquer.

Zone de résolution inverse

Pour chaque réseau ou sous-réseau IP (ou plage d'adresses) on définit une zone de résolution inverse dont le fichier contient une association IP vers nom de machine. C'est en fait presque la même chose que la zone de résolution sauf que l'on doit respecter une convention de nommage :

- Le nom de la zone se termine toujours par une domaine spécial `.in-addr.arpa`.
- On doit tout d'abord déterminer quel réseau la zone doit couvrir (cas des sous-réseaux). Pour nous : un réseau de classe C 192.168.1.0 soit 192.168.1/24.
- On inverse l'ordre des octets dans l'adresse : 1.168.192.
- On ajoute `in-addr.arpa`. Notre nom de zone sera donc `1.168.192.in-addr.arpa`.

Exemple

Soit un domaine `domaine.org` sur un réseau de classe C 192.168.1.0. Soit deux serveurs DNS 192.168.1.1 Master et 192.168.1.2 Slave.
Sur le Master

```
zone "domaine.org" {
    type    master;
    file    "domaine.org.zone";
    allow-transfer { 192.168.1.2; } ;
    notify yes;
};
zone "1.168.192.in-addr.arpa" {
    type    master;
    file    "192.168.1.zone";
    allow-transfer { 192.168.1.2; } ;
    notify yes;
};
```

Sur le slave

```
zone "domaine.org" {
    type        slave;
    file        "domaine.org.zone";
    masters     { 192.168.1.1; };
};
zone "1.168.192.in-addr.arpa" {
    type        slave;
    file        "192.168.1.zone";
    masters     { 192.168.1.1; } ;
};
```

Zones spéciales

La zone racine « . » permet de spécifier les serveurs racines. Quand aucune des zones n'arrive à résoudre une requête, c'est la zone racine qui est utilisée par défaut et qui renvoie sur les serveurs racines.
La zone de loopback n'est pas nécessaire bien que utile. Elle fait office de cache **DNS**. Quand une requête arrive sur le serveur et qu'il ne possède pas l'information de résolution, il va la demander aux serveurs DNS racines qui redescendront l'information. Celle-ci est alors placée en cache. Du coup les accès suivants seront bien plus rapides !

Serveur DNS
Fichiers de zones

Définitions

Les fichiers de zones utilisent plusieurs termes, caractères et abréviations spécifiques.

- **RR** : Ressource Record. Nom d'un enregistrement DNS (les données du DNS).
- **SOA** : Star Of Authority. Permet de décrire la zone.
- **IN** : the Internet. Définit une classe d'enregistrement qui correspond aux données Internet (IP). C'est celle par défaut si elle n'est pas précisée pour les enregistrements.
- **A** : Address. Permet d'associer une adresse IP à un nom d'hôte. Pour Ipv6 c'est AAAA.
- **NS** : Name Server. Désigne un serveur DNS de la zone.
- **MX** : Mail eXchanger. Désigne un serveur de courrier électronique, avec un indicateur de priorité. Plus la valeur est faible, plus la priorité est élevée.
- **CNAME** : Canonical Name. Permet de rajouter des alias : lier un nom à un autre. On peut créer des alias sur des noms d'hôte et aussi sur des alias.
- **PTR** : Pointer. Dans une zone de résolution inverse, fait pointer une IP sur un nom d'hôte.
- **TTL** : Time To Live. Durée de vie des enregistrements de la zone.
- @ : dans les déclarations de la zone, c'est un alias (caractère de remplacement) pour le nom de la zone déclarée dans /etc/named.conf. Ainsi si la zone s'appelle domaine.org, @ vaut domaine.org. Dans la déclaration de l'administrateur de la SOA, il remplace ponctuellement le point dans l'adresse de courrier électronique.
- Le point « . »: Si l'on omet le point en fin de déclaration d'hôte, le nom de la zone est concaténé à la fin du nom. Par exemple pour la zone domaine.org, si on écrit **poste1**, cela équivaut à **poste1.domaine.org.** Si on écrit **poste1.domaine.org** (sans le point à la fin) alors on obtient comme résultat **poste1.domaine.org.domaine.org** ! Pour éviter cela, vous devez écrire **poste1.domaine.org.** (notez le point à la fin).

Zone

Commencez tout d'abord par une directive **TTL** qui indique le temps de vie de la zone en secondes. Cela signifie que chaque enregistrement de la zone sera valable durant le temps indiqué par **TTL** (note : il est possible de modifier cette valeur pour chaque enregistrement). Durant ce temps, les données peuvent être placées en cache par les autres serveurs de noms distants. Une valeur élevée permet de réduire le nombre de requêtes effectuées et de rallonger les délais entre les synchronisations.

```
$TTL 86400 "
```

Après les directives TTL, placez un enregistrement de ressources **SOA** :

```
<domain> IN SOA <primary-name-server> <hostmaster-email> (
<serial-number>
<time-to-refresh>
<time-to-retry>
<time-to-expire>
<minimum-TTL> )
```

- **domain** : c'est le nom de la zone, le même nom que celui utilisé dans /etc/named.conf. On peut le remplacer par @ sinon il ne faut pas oublier de le terminer par un point (pour éviter une concaténation).
- **primary-name-server** : le nom sur le serveur DNS maître sur cette zone. Il ne faudra pas oublier de le déclarer dans la liste des hôtes (enregistrements PTR ou A).
- **hostmaster-email** : adresse de courrier électronique de l'administrateur du serveur de nom. Le caractère @ étant déjà réservé à un autre usage, on utilise un point pour le remplacer. Ainsi « admin@domaine.org » devra s'écrire « admin.domaine.org. » .
- **serial-number** : c'est un numéro de série que l'on doit incrémenter manuellement à chaque modification du fichier zone pour que le serveur de nom sache qu'il doit recharger cette zone.

Serveur DNS
Fichiers de zones

16

Zone

```
@    IN    SOA    dns1.domaine.org.    hostmaster.domaine.org. (
                  2005122701     ; serial
                  21600          ; refresh de 6 heures
                  3600           ; tenter toutes les 1 heures
                  604800         ; tentatives expirent après une semaine
                  86400    )     ; TTL mini d'un jour
```

Passez ensuite au enregistrements **NS** (Name Server) où vous spécifiez les serveurs de noms de cette zone.

```
IN NS dns1
IN NS dns2
```

Passez ensuite à l'énumération des serveurs de courrier électronique de la zone. La valeur numérique située après MX indique la priorité. Plus la valeur est basse plus le serveur est prioritaire et susceptible d'être contacté en premier. Si les valeurs sont identiques, le courrier est redistribué de manière homogène entre les serveurs. Si un serveur ne répond pas (chargé, en panne) la bascule vers une autre machine est automatique.

```
IN MX 10 mail
IN MX 15 mail2
```

Si vous souhaitez qu'une machine réponde en passant par le FQDN domaine.org sans préciser d'hôte (par exemple http://domaine.org sans utiliser http://www.domaine.org) alors vous pouvez maintenant déclarer une adresse IP pour ce serveur. Ainsi la commande ping domaine.org répondra 192.168.1.3 !

```
IN A 192.168.1.3
```

Vous pouvez maintenant déclarer les autres hôtes dont les serveurs de noms, de mails, les postes, etc.

Zone de résolution inverse

La zone de révolution inverse est presque identique à la précédente, si ce n'est que les enregistrements A sont remplacés par des enregistrements PTR destinés à traduire une IP en hôte. Le TTL et la déclaration SOA doivent être si possible identiques (sauf le nom de la zone). Vous placez aussi les enregistrements NS.

```
IN NS dns1.domaine.org.
IN NS dns2.domaine.org.
```

Vous n'êtes pas obligé de placer dans la zone de résolution inverse la traduction des adresses IP du DNS, étant donné que c'est le DNS lui-même qui résout son propre nom ! Cependant le faire peut accélérer la démarche, le DNS n'ayant pas à exécuter une requête sur lui-même. Passez aux enregistrements PTR traduisant l'adresse IP pour chaque hôte.

```
1     IN    PTR    dns1.domaine.org.
2     IN    PTR    dns2.domaine.org.
3     IN    PTR    server1.domaine.org.
4     IN    PTR    server2.domaine.org.
11    IN    PTR    poste1.domaine.org.
12    IN    PTR    poste2.domaine.org.
13    IN    PTR    poste3.domaine.org.
```

Il est théoriquement possible pour la même IP d'attribuer plusieurs hôtes les RFC ne sont pas très explicites sur cette possibilité qui, au final, peut créer des problèmes.

Diagnostic des problèmes de configuration

La commande **named-checkconf** vérifie la syntaxe du fichier **named.conf**. Vous lui fournissez en paramètre le fichier. La sortie indiquera les lignes posant problème.
La commande **named-checkzone** vérifie la syntaxe d'un fichier de zone (y compris de résolution inverse). Vous lui spécifiez en paramètre le nom du fichier zone.

Interrogation dig et host

Le programme `dig` est un outil d'interrogation avancé de serveur de noms, capable de restituer toutes les informations des zones.

```
> dig free.fr

; <<>> DiG 9.4.1-P1 <<>> free.fr
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63972
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;free.fr.                       IN      A

;; ANSWER SECTION:
free.fr.                86363   IN      A       212.27.48.10

;; Query time: 1 msec
;; SERVER: 10.23.254.240#53(10.23.254.240)
;; WHEN: Wed May 14 09:36:09 2008
;; MSG SIZE  rcvd: 41
```

Par défaut dig ne restitue que l'adresse de l'hôte passé en paramètre. En cas de réussite, le statut vaut **NOERROR**, le nombre de réponses est indiqué par **ANSWER** et la réponse se situe en dessous de la section **ANSWER**. Pour obtenir une résolution inverse il existe deux solutions.

```
$ dig 10.48.27.212.in-addr.arpa ptr "
```

ou plus simplement :

```
$ dig -x 212.27.48.10 "
```

Interrogation dig et host

Dans la première syntaxe, remarquez que vous pouvez rajouter un paramètre d'interrogation. Voici les principaux.

- **a** : uniquement l'adresse
- **any** : toutes les informations concernant le domaine
- **mx** : les serveurs de messagerie
- **ns** : les serveurs de noms
- **soa** : la zone Start of Authority
- **hinfo** : infos sur l'hôte
- **txt** : texte de description
- **ptr** : zone reverse de l'hôte
- **axfr** : liste de tous les hôtes de la zone.

L'outil host fournit le même résultat de manière peut-être un peu plus simple.

```
$ host free.fr "
```

Principe

- Quand un client (un utilisateur) envoie un message, il utilise un **MUA** (Mail User Agent), par exemple Outlook Express, Thunderbird, Evolution, Kmail, Mutt, etc.
- Le **MUA** envoie le message au **MTA** (Mail Transport Agent). Le MTA étudie l'adresse électronique pour isoler l'utilisateur et le domaine de destination. Puis il vérifie les informations DNS de type **MX** (Mail exchanger) pour le domaine choisi, pour savoir à quel serveur transmettre le courrier. Si aucun MTA n'est disponible, le message est placé en file d'attente et relance la distribution plus tard (le délai dépend de la configuration du MTA).
- Le MX peut être soit un autre MTA, qui jouera le rôle de routeur (cas d'une redirection vers un sous-domaine par exemple), soit un MDA (Mail Delivery Agent). Le **MDA** place le message dans un fichier temporaire, peut le filtrer, etc.
- À ce niveau, le destinataire reçoit le message : soit il le récupère en lisant directement le fichier temporaire (cas de la commande mail par exemple) soit il passe par un protocole de type **POP** ou **IMAP**.
- Le protocole de transport de messages est le **SMTP** (Simple Mail Transfer Protocol) sur le port 25.
- Les protocoles de réception de messages soit **POP** (Post Office Protocol) sur le port 110 (POP3), soit **IMAP** (Internet Message Access Protocol).

Deux suites de courrier électronique se partagent l'essentiel du marché sur Unix : **sendmail** et **postfix**.

Principe

La suite libre **sendmail** est la plus connue et la plus utilisée. Sendmail a été créé en 1981 par Eric Allman et a été intégré à BSD 4.2 en 1983. On estimait en 2000 son utilisation à plus de 100 millions de serveurs de courrier électronique. Tant qu'il ne faut pas modifier fortement sa configuration de base, sendmail est idéal. Si vous souhaitez aller plus loin, l'achat d'un livre complet s'avère plus que nécessaire. La configuration de sendmail est si complexe qu'un langage de macros appelé m4 a été inventé rien que pour lui. Aussi, vous n'éditez pas (ou très rarement) le fichier de configuration de sendmail : vous éditez le fichier source des macros et vous le « recompilez » : m4 va créer le fichier de configuration de sendmail. Sendmail est devenu un monstre de puissance et de configuration.
Le produit **postfix** tend à être de plus en plus utilisé non pas par les déçus de sendmail mais par ceux qui craignent de devoir le configurer. C'est une alternative à sendmail. Les buts de ses développeurs (dont certains sont ceux de sendmail) sont :

- la compatibilité avec sendmail
- la rapidité (plus de un million de messages par jour sur un simple Pentium 4)
- la simplicité d'administration (fichier de configuration simple et lisible)
- la sécurité (peut être chrootée)
- la modularité (décomposition des traitements).

On retiendra la simplicité. En effet, on peut configurer postfix avec une seule commande sans avoir besoin d'éditer de fichier. C'est ce serveur que vous allez utiliser.

Configuration simple

La configuration de **postfix** est située dans /etc/postfix/main.cf. On modifie ses valeurs soit à la main, soit à l'aide de la commande **postconf**.
Postfix lance tout d'abord un service maître, **master**, qui sera chargé des processus secondaires **smtpd**, **pickup** et **nqmgr**.
Remarque: Sur certaines distributions il faut modifier la configuration par défaut qui utilise la suite sendmail. Par exemple sur Red Hat vous devez indiquer d'utiliser postfix à la place de sendmail avec la commande **alternatives**.

```
#alternatives -set mta /usr/sbin/sendmail.postfix "
```

Appliquez une configuration de base avec la commande postconf.
1- Domaine d'origine des messages

```
#postconf -e "myorigin = mondomaine.org" "
```

2- De quels domaines recevoir le courrier

```
#postconf -e "mydestination = mondomaine.org" "
```

"3- De quels clients relayer le courrier

```
#postconf -e "mynetworks = 192.168.1.0/24, 127.0.0.1" "
```

4- Sur quelles interfaces écouter

```
#postconf -e "inet_interface = all" "
```

Lancez le service.

```
#service postfix start "
```

Alias d'utilisateurs

Vous pouvez placer dans le fichier `/etc/aliases` des alias pour les utilisateurs locaux. Par exemple, si les messages de webmaster, admin et root doivent être redirigés vers regis

```
regis: webmaster, admin, root "
```

Test

Les traces sont placées dans /var/log/maillog. Testez le serveur de cette manière (par exemple) :

```
mail -s ‘echo $USER‘ root@server1 < /etc/passwd "
```

POP et IMAP

Il existe plusieurs suites pour gérer POP et IMAP. Une suite se nomme **cyrus-imap** et est en principe réservée aux grosses structures et aux serveurs 100% dédiés au courrier, c'est-à-dire où les utilisateurs ne se connectent pas.
Une solution se nomme **dovecot**. Après son installation, il suffit juste de le démarrer en tant que service pour que tout fonctionne, ou presque.

Éditez le fichier `/etc/dovecot.conf` pour vérifier les protocoles supportés.

```
protocols = imap pop3 "
```

Lancez le service :

```
service dovecot start "
```

Testez en envoyant un message. Configurez un client de messagerie pour vérifier si le serveur POP fonctionne :

```
# telnet localhost 110 "
```

Service HTTP Apache
Présentation

Présentation

Apache 2 est le serveur HTTP le plus utilisé actuellement sur les serveurs Web. Sa configuration et sa flexibilité en font un serveur incontournable.
Lorsqu'un serveur Apache reçoit des requêtes, il peut les redistribuer à des processus fils. La configuration permet de lancer des processus de manière anticipée et d'adapter dynamiquement ce nombre en fonction de la charge.
Apache est modulaire. Chaque module permet d'ajouter des fonctionnalités au serveur. Le module le plus connu est probablement celui gérant le langage PHP, « mod_php ». Chaque module s'ajoute via les fichiers de configuration, et il n'y a même pas besoin de relancer le serveur Apache : on lui donne juste l'ordre de relire sa configuration.
Apache peut gérer plusieurs sites Web en même temps, ayant chacun leur nom, à l'aide des hôtes virtuels.

Service HTTP Apache
Arrêt/Relance

Arrêt/Relance

Le nom du service dépend de la distribution. Il est souvent intitulé **apache** ou **httpd**. Suivant la distribution lancez le **service** via la commande service ou directement par son nom **/etc/init.d/apache**.

- **/etc/init.d/httpd start** : démarre
- **/etc/init.d/httpd stop** : stoppe
- **/etc/init.d/httpd restart** : redémarre
- **/etc/init.d/httpd reload** : demande à Apache de relire sa configuration sans redémarrer.

Apache est fourni avec l'outil **apachectl** qui reprend les paramètres (liste non exhaustive) start, stop, status, reload, et surtout configtest qui valide ou non le contenu du fichier de configuration de Apache.

Service HTTP Apache
Configuration

Configuration

La configuration principale est stockée dans /etc/httpd/conf/httpd.conf. Elle contrôle les paramètres généraux du serveur Web, les hôtes virtuels et les accès. La configuration des différents modules est placée dans /etc/httpd/conf.d. Les modules sont présents dans /etc/httpd/modules/. Par défaut la racine du serveur, celle où sont placées les pages du site, est dans /var/www ou /srv/www. Cette position dépend de la directive **DocumentRoot** dans les fichiers de configuration.

Service HTTP Apache
Directives générales

Directives générales

Il n'est pas possible de lister toutes les directives du fichier httpd.conf mais quelques-unes sont importantes.

- **ServerRoot** : répertoire contenant les fichiers du serveur (configuration et modules). C'est généralement /etc/httpd.
- **Listen** : ports sur lesquels le serveur Apache écoute. Par défaut 80 (443 en https). On peut en spécifier plusieurs avec plusieurs directives Listen. Si le serveur dispose de plusieurs adresses IP, on peut rajouter l'IP au port associé : **Listen 192.168.1.3:80**.
- **User** : utilisateur des processus Apache. On n'utilise jamais root, mais un compte créé pour l'occasion, généralement Apache.
- **Group** : idem mais pour le groupe.
- **ServerAdmin** : adresse de courrier électronique de l'administrateur.
- **ServerName** : nom d'hôte (et port) du serveur. Il ne correspond pas forcément au nom d'hôte de la machine. Par contre il doit être valide. ServerName www.mondomaine.org
- **UseCanonicalName** : si elle vaut **on**, Apache va répondre en utilisant les informations de ServerName et Port, et pas les informations envoyées par le client. Par exemple, un http//192.168.1.3 se transforme en http://www.mondomaine.org.
- **UserDir** : nom d'un sous-répertoire ou chaque utilisateur peut placer ses fichiers HTML personnels. Généralement public_html. On y accède avec http://www.mondomaine.org/ login/page.html.
- **ErrorLog** : fichier où sont placées les logs d'erreur du serveur. /var/log/httpd/error_log.
- **CustomLog** : fichier journal de Apache. `/var/log/httpd/access_log`.

Service HTTP Apache
Directives générales

Directives générales

- **Timeout** : durée pendant laquelle le serveur attend des émissions/réceptions au cours d'une communication. Elle est réglée sur 300 secondes.
- **KeepAlive** : définit si le serveur peut exécuter plus d'une requête par connexion. C'est à off par défaut mais si on passe à on Apache peut générer rapidement des processus enfants pour le soulager s'il est très chargé.
- **MaxKeepAliveRequests** : nombre maximum de requêtes par connexion persistante. Une valeur élevée peut augmenter les performances du serveur. 100 par défaut.
- **KeepAliveTimeout** : durée pendant laquelle le serveur (généralement un processus fils) attend après avoir servi une requête. Par défaut 15 secondes. Après les 15 secondes, la demande sera réceptionnée par le serveur avec un Timeout.
- **StartServers** : nombre de serveurs créés au démarrage. Par défaut 8. Apache gérant ensuite dynamiquement le nombre de serveurs fils, ce paramètre a peu d'importance car le nombre va baisser ou augmenter rapidement.

Service HTTP Apache
Gestion des performances

Gestion des performances

- **MaxRequestPerChild** : nombre de requêtes pouvant être exécutées par un processus fils avant de s'arrêter. Par défaut 4000. Ça permet une occupation mémoire plus réduite en la libérant plus rapidement.
- **MaxClients** : limite du nombre total de requêtes pouvant être traitées simultanément. Par défaut 150. La valeur limite le risque de saturation du serveur.
- **MinSpareServers** / **MaxSpareServers** : suivant la charge de la machine, Apache peut lancer d'autres processus serveurs pour s'adapter à la charge actuelle. Par défaut, elles sont de 5 et 20. Ces deux valeurs déterminent les nombres limites autorisés. Si un serveur est peu chargé avec 15 processus, Apache en supprimera mais en gardera toujours au moins 5. Si le serveur devient chargé avec 10 processus, Apache en créera des supplémentaires à concurrence de 20 maximum.
- **MinSpareThreads** / **MaxSpareThreads** : chaque serveur fils peut accepter un certain nombre de requêtes simultanément. Pour cela il utilise les threads. Ces deux valeurs sont fixées par défaut à 20 et 75 et le mécanisme fonctionne comme ci-dessus.
- **ThreadsPerChild** : nombre de threads par défaut au lancement d'un serveur fils. Par défaut 25.

Service HTTP Apache

Les répertoires, alias et emplacements

Directory

Les balises **<Directory chemin>** et **</Directory>** permettent de regrouper des directives qui ne s'appliqueront qu'au chemin (et à ses sous-répertoires) donnés. La directive **Options** est fortement conseillée.

```
<Directory /var/www/html/images>
Options +Indexes +FollowSymLinks
DirectoryIndex index.php index.html
Order allow, deny
Allow from All
</Directory>
<Directory /var/www/html/cgi-bin>
Options +ExecCGI
</Directory>
```

La directive Options accepte les valeurs suivantes précédées de + ou - et séparées par des espaces :

- **All** : toutes les options sauf MultiViews
- **Indexes** : si jamais le répertoire ne contient pas de fichier HTML par défaut (cf DirectoryIndex), le contenu du répertoire est affiché sous forme de listing
- **ExecCGI** : l'exécution de scripts CGI est autorisée
- **FollowSymLinks** : le serveur suit les liens symboliques.

Service HTTP Apache

Les répertoires, alias et emplacements

Directory

La directive **DirectoryIndex** précise les fichiers html ou cgi par défaut lors du chargement d'une URL.

```
DirectoryIndex index.php index.html"
```

Au chargement, sans préciser le nom du fichier html, le serveur tentera de charger index.php et, s'il est absent, index.html. Dans le cas contraire, c'est l'option Indexes qui détermine si le contenu doit être visible sous forme de répertoire.

La directive **Allow** indique quels clients seront autorisés à accéder au répertoire. Ce peut être all, un domaine, une IP, une IP tronquée (sous-réseau), une paire réseau/sous-réseau, etc. La directive `Deny` interdit l'accès et s'utilise de la même manière. L'ordre est déterminé par la directive `Order`.

Service HTTP Apache

Les répertoires, alias et emplacements

Alias

La directive **Alias** permet de créer un raccourci entre l'arborescence logique du site Web et un chemin du système de fichiers.

```
Alias /help "/usr/share/doc/html" "
```

Dans ce cas, l'url http://www.monsite.org/help ne cherchera pas dans le répertoire /var/www/html/help mais dans /usr/share/doc/html.

Contrairement aux balises **<Directory>**, les balises **<Location>** et **</Location>** permettent d'appliquer des directives basées sur l'URL (et pas les répertoires).

```
<Location /help>
Options +All -FollowSymLinks
Order deny, allow
Deny from all
Allow from .mondomaine.org
</Location>
```

Service HTTP Apache

Les répertoires, alias et emplacements

Alias

Un serveur Apache est capable de gérer plusieurs sites Web sur un même serveur. Il existe plusieurs méthodes. La première se base sur les noms (plusieurs sites Web pour un serveur) l'autre sur les adresses ip (une adresse IP pour chaque site Web). Vous allez aborder la première version.
La directive **NameVirtualHost** spécifie l'adresse IP sur laquelle le serveur va recevoir les requêtes d'accès aux hôtes virtuels.
Les balises **<VirtualHost>** et **</Virtualhost>** permettent de définir un hôte virtuel.

```
NameVirtualHost 192.168.1.3

<VirtualHost 192.168.1.3>
    ServerName          www2.mondomaine.org
    ServerAdmin         webmaster@www2.mondomaine.org
    DocumentRoot        /var/www/www2.mondomaine.org/
    ErrorLog        logs/www2 error log
    CustomLog       logs/www2 access log
</VirtualHost>
```

Service HTTP Apache

Les répertoires, alias et emplacements

Alias

Relancez Apache. Avec un navigateur (ex : Firefox) on vérifie si notre hôte virtuel répond avec l'URL http://www2.mondomaine.org (cette adresse doit être déclarée dans /etc/hosts ou connue du serveur de noms et pointer sur le bon serveur).
Remarquez cependant que si vous passez par http://www.mondomaine.org vous n'obtenez plus le site par défaut ! En effet quand on déclare des hôtes virtuels, le premier de la liste devient l'hôte par défaut et prioritaire.
Rajoutez un hôte virtuel pour le site principal.

```
<VirtualHost 192.168.1.3>
    ServerName          www.mondomaine.org
    ServerAdmin         webmaster@www.mondomaine.org
    DocumentRoot        /var/www/html
    ErrorLog        logs/error_log
    CustomLog       logs/access_log
</VirtualHost>
```

Attention, la règle ci-dessus s'applique aussi avec un nom court. Si vous inscrivez http://www ou http://www2 vous tomberez toujours sur l'hôte virtuel par défaut. Il faut demander http://www.mondomaine.org et http://www2.mondomaine.org.

Partage de fichiers
NFS

Lancement

Le partage de fichier **NFS** (Network File System) ou système de fichiers réseau permet de partager tout ou partie de son système de fichiers à destination de clients NFS, bien souvent d'autres Unix. Dans sa version de base c'est un système simple et efficace. Nous allons étudier la version 2.
NFS s'appuie sur le **portmapper (portmap)**, le support nfs du noyau et les services `rpc.nfsd` et `rpc.mountd`.
Pour lancer le service NFS, portmap et nfs doivent être lancés (en vérifier le statut avant).

```
# service portmap status # /etc/init.d/portmap status ou rpcinfo -p "
# service nfs status "
# service portmap start"
# service nfs start "
# service nfslock start "
```

Pour savoir si le service est disponible sur un hôte distant :

```
# rpcinfo -p hote "
```

Partage de fichiers
NFS

Partage côté serveur

La liste des systèmes de fichiers à exporter se trouve dans /etc/exports. Il contient un partage par ligne.

```
# Rep exportes Autorisations d'acces "
/ poste1(rw) poste2(rw,no_root_squash) "
/projects *.mondomaine.org(rw) "
/home/joe poste*.mondomaine.org(rw)"
/pub 192.168.1.0/255.255.255.0(ro) "
```

Chaque ligne est composée de deux parties. La première est le chemin du répertoire exporté. La seconde contient les autorisations d'accès.
L'autorisation d'accès est composée de paires hôtes/permissions selon le format suivant :

```
host(permissions)"
```

Si l'hôte n'est pas défini, c'est tout le réseau (portée dite mondiale) qui sera concerné par les permissions. Si les permissions ne sont pas définies, l'export sera en lecture seule. Il ne faut surtout pas mettre d'espaces entre l'hôte et les permissions. L'hôte peut être :

- un nom d'hôte unique
- un domaine
- un réseau ou un sous-réseau
- une combinaison de l'ensemble avec des caractères de substitution (*, ?).

Partage côté serveur

Les permissions peuvent être :

- **ro** : lecture seule
- **rw** : lecture écriture
- **no_root_squash** : le root distant équivaut au root local
- **root_squash** : si root se connecte au partage, son uid sera remplacé par celui d'un utilisateur anonyme. Ainsi il n'y a pas de risques que l'utilisateur root d'un poste local puisse être root sur un partage distant
- **all_squash** : étend la règle précédente à tous les utilisateurs
- **anonuid** / **anongid** : uid et gid pour l'utilisateur anonyme.

Partage côté serveur

Pour une gestion correcte des droits et des permissions, **les utilisateurs de même nom (login) doivent avoir les mêmes UID et GID sur le serveur et le client**. NFS se base en effet sur ces valeurs pour la sécurité des données du partage. Le nom de login seul ne suffit pas. Dans l'exemple ci-dessus, l'utilisateur joe est autorisé à accéder au partage **/home/joe** (on suppose que c'est son répertoire personnel) sur tous les postes du domaine. L'utilisateur joe doit être déclaré de la même manière (même UID) sur le serveur et sur tous les postes. C'est pour cela que l'on utilise souvent NIS avec NFS.
La commande **exportfs** permet de contrôler les partages.

- `exportfs -r` : rafraîchit la liste des partages après modification de /etc/exports
- `exportfs -v` : liste des partages
- `exportfs -a` : exporte (ou recharge) tous les partages de /etc/exports ou un partage donné
- `exportfs -u` : stoppe le partage donné. -a pour tous.

La commande `showmount` montre les partages d'un hôte donné.

```
showmount -e host "
```

Montage côté client

Le support NFS est inclus sous forme de module du noyau. Il est automatiquement chargé à l'utilisation d'un accès NFS.
Dans `/etc/fstab` notez les modifications :

```
server1:/pub /mnt/pub nfs defaults 0 0 "
```

Le périphérique est remplacé par le chemin du partage sous la forme **serveur:chemin**. Le système de fichiers est **nfs**. C'est identique avec la commande **mount** :

```
mount -t nfs serveur1:/pub /mnt/pub "
```

NFS dispose d'options de montage spécifiques :

- **nolock** : option de compatibilité avec d'anciens serveurs NFS
- **intr** : interrompt une requête NFS si le serveur ne répond pas
- **hard** : bloque les processus qui tentent d'accéder à un partage inaccessible
- **soft** : un processus retournera une erreur en cas d'accès infructueux
- **rsize=8192, wsize=8192** : taille des blocs de lecture/écriture sur le serveur. Une écriture de 8 ko est plus rapide que 8 écritures de 1 ko.

FTP

Le serveur **FTP** (File Transfer Protocol) le plus courant est **vsftpd** (Very Secure FTP Daemon). Il a l'avantage d'être très petit, performant et rapide tout en étant tout de même très configurable (moins toutefois que Proftpd ou d'autres). Il convient dans la quasi-totalité des situations. C'est un service qui peut aussi bien être lancé par **xinetd** que (dans les dernières versions des distributions) en tant que service seul.
Deux niveaux de sécurité sont utilisables :

- **Anonyme** : tout le monde peut se connecter au serveur FTP en tant que utilisateur **ftp** ou **anonymous**. l'environnement FTP est chrooté.
- **Utilisateur** : les utilisateurs qui existent sur le serveur peuvent se connecter avec leur mot de passe et ont un accès complet à leurs données dans leur répertoire personnel.

Les utilisateurs anonymes étant considérés comme l'utilisateur ftp, c'est le répertoire personnel de ce compte qui est la racine du ftp.
Le fichier de configuration est présent dans /etc/vsftpd/vsftpd.conf.
La racine du ftp par défaut est dans /var/ftp.
Le script de lancement est /etc/init.d/vsftpd (service vsftpd start).
Pour activer ou non l'accès anonyme on modifie le fichier de configuration. Dans ce cas, l'utilisateur peut se connecter en tant que anonymous ou ftp. Dans tous les cas, il sera reconnu comme utilisateur « ftp » du serveur une fois connecté :

```
anonymous_enable=YES/NO "
```

FTP

Pour activer ou non l'envoi de fichiers sur le serveur par des anonymes. Dans ce cas, l'autorisation d'écriture dans un répertoire est fonction des droits du répertoire sur le serveur (notamment si l'utilisateur ftp a le droit d'écrire ou non dans un répertoire) :
`anon_upload_enable=YES/NO "`
Vous pouvez interdire à des utilisateurs de se connecter en plaçant leur noms dans /etc/vsftpd.ftpusers.
Vous pouvez ajouter des utilisateurs dans `/etc/vsftpd.user_list` si `userlist_enable=YES`.
Dans ce cas, c'est la valeur de `userlist_deny (YES/NO)` qui déterminera si le fichier contient les utilisateurs interdits ou autorisés.
On peut créer dans chaque répertoire du serveur un fichier .message. Dans ce cas, son contenu sera affiché lors de l'accès au répertoire.

Présentation

Samba est un ensemble de serveurs implémentant les protocoles SMB/CIFS et NetBIOS/WINS pour Unix. Son utilisation la plus connue est le partage de ressources entre Windows et Unix, mais il fonctionne parfaitement bien entre deux Unix. Un **partage** est aussi appelé **service**. Samba est composé de deux services :

- **smbd** : serveur SMB/CIFS.
  - Authentification et autorisation.
  - Partages de fichiers et d'imprimantes.
- **nmbd** : serveur de noms NetBIOS.
  - Parcours des ressources.
  - Serveur WINS.

Un troisième service, **winbindd**, permet d'utiliser les comptes utilisateur d'un domaine Microsoft. Les dernières versions de Samba (3 et suivantes) permettent aussi de se raccorder à Active Directory.
Les fonctions principales de Samba sont :

- Authentification des utilisateurs.
- Partage de fichiers et d'imprimantes.
- Parcours des ressources partagées du réseau.
- Résolution de noms (indépendante de DNS) Nom Netbios IP ou vice versa.

Configuration

La configuration de Samba se trouve dans /etc/samba/smb.conf. Vous pouvez tester sa syntaxe avec l'outil **testparm**.
Le fichier smb.conf reprend la syntaxe des fichiers de configuration de Windows de type **ini** avec des sections délimitées par des crochets [ ].
Par défaut trois sections sont présentes :

- **[global]** : réglages génériques et globaux du serveur, nom, commentaires, méthode d'authentification, réglages par défaut, etc.
- **[homes]** : partage des répertoires personnels des utilisateurs.
- **[printers]** : partage des imprimantes.

Les paramètres sont de la forme :
`nom = valeur "` Les commentaires commencent par un point-virgule ou un dièse #.

```
[global]
workgroup = MYGROUP
netbios name = posteN
security = share
```

- **workgroup** : nom de groupe / domaine de travail
- **netbios name** : nom netbios de la machine
- **security** : méthode d'authentification (cf. plus bas).

Partage de fichiers

Chaque partage doit disposer de sa propre section dans smb.conf. Voici comment partager le répertoire `/opt/partage1` sous le nom de service **partage1** :

```
[partage1]
comment = Répertoire partagé 1
path = /opt/partage1
browseable = yes
public = no
writable = yes
printable = no
group = partage
```

Voici une description des valeurs et de quelques autres possibles:

- **partage1** : le nom du partage tel qu'il apparaît dans le « voisinage ».
- **comment** : le commentaire tel qu'il apparaît à côté du nom de partage.
- **path** : le chemin du partage.
- **public** : le partage est accessible à l'utilisateur par défaut (guest).
- **browseable** : le partage apparaît dans le « voisinage ».
- **writable** : le partage est accessible en lecture et écriture.
- **printable** : le partage est une imprimante.
- **group** : nom du groupe par défaut pour la connexion.
- **valid users** : nom des utilisateurs autorisés à accéder à ce partage.
- **read only** : le partage est en lecture seule pour tout le monde.
- **guest ok** : aucun mot de passe n'est nécessaire pour accéder au partage. Dans ce cas le

Partage des imprimantes

En plus de la section **[printers]**, vous pouvez ajouter des sections pour des imprimantes spécifiques. Le paramètre **printing** de la section **[global]** permet de modifier le sous-système d'impression basé par défaut sur CUPS.

```
[Bureau150]
comment = Laserjet 2100
printer = lj2100
valid users = riri fifi loulou
path = /var/spool/lj2100
public = no
writable = no
PRINTABLE = YES
browseable = yes
```

- **printer** : nom de l'imprimante sous Linux.
- **valid users** : nom des utilisateurs autorisés à accéder à ce partage. Un @ devant le nom indique un groupe d'utilisateurs.
- **path** : chemin du spool d'impression.

Méthodes d'authentification

Samba propose plusieurs méthodes d'authentification définies dans la section **[global]** :

- **user** : méthode par défaut l'accès à l'ensemble des partages d'un serveur se fait par la validation d'un nom d'utilisateur et d'un mot de passe uniques.
- **share** : méthode de validation des identifiants partage par partage. Dans ce cas, tous les accès aux partages, même publics, nécessitent des identifiants.
- **domain** : utilisation d'un groupe de travail avec authentification.
- **ads** : utilisation de Active Directory.

D'autres types d'authentification sont possibles comme un annuaire LDAP.

```
[global]
workgroup = MYGROUP
netbios name = posteN
security = share
```

Correspondance des noms et des mots de passe

Les mots de passe du protocole SMB n'ont pas la même forme que les mots de passe Unix/Linux. Il faut recréer les mots de passe pour chaque utilisateur devant utiliser SMB avec la commande **smbpasswd**. Les utilisateurs doivent déjà exister sous Unix.

```
smbpasswd -a toto "
```

Les utilisateurs SMB sont présents dans `/etc/samba/smbpasswd`. La commande **mksmbpasswd** peut réaliser cela en batch :

```
cat /etc/passwd | mksmbpasswd > /etc/samba/smbpasswd "
```

Vous pouvez établir une table de correspondance entre les noms d'utilisateurs Windows et ceux de Unix dans `/etc/samba/smbusers`.

```
# Unix_name = SMB_name1 SMB_name2 ... "
root = administrator admin administrateur"
```

En ligne

Toute machine sous Microsoft Windows peut accéder aux partages Samba. Les navigateurs des environnements de bureau **KDE** et **GNOME** acceptent la navigation dans les partages grâce au protocole **smb**:/ dans les URL. KDE propose même l'équivalent d'un voisinage réseau.
L'outil **smbclient** est une sorte de client FTP pour le protocole SMB. Les chemins d'accès aux ressources sont de la forme:

```
//machine/partage "
```

Par exemple pour se connecter au service (ici un partage) d'une machine:

```
smbclient //machine/service -U login%passwd"
```

Pour lister les services proposés par une machine:

```
smbclient -L hostname -U login%passwd # liste des ressources "
```

Montage

Montez un système de fichiers SMB avec la commande **smbmount**.

```
smbmount //machine/partage /mnt/mountpoint -o username=login "
```

On peut aussi réaliser le montage dans /etc/fstab. Tout comme avec nfs, c'est un service spécialisé qui montera et démontera les partages. Les derniers noyaux Linux ont remplacé **smbfs** par **cifs** :

```
//machine/partage /mnt/mountpoint cifs defaults,username=nobody 0 0 "
```

Merci pour votre attention!