Manuel de Travaux Pratiques Administration Système en réseau

Philippe Latu
philippe.latu(at)inetdoc.net

https://www.inetdoc.net

Ce manuel regroupe les supports du cycle de travaux pratiques sur le thème de l'administration système en réseau.

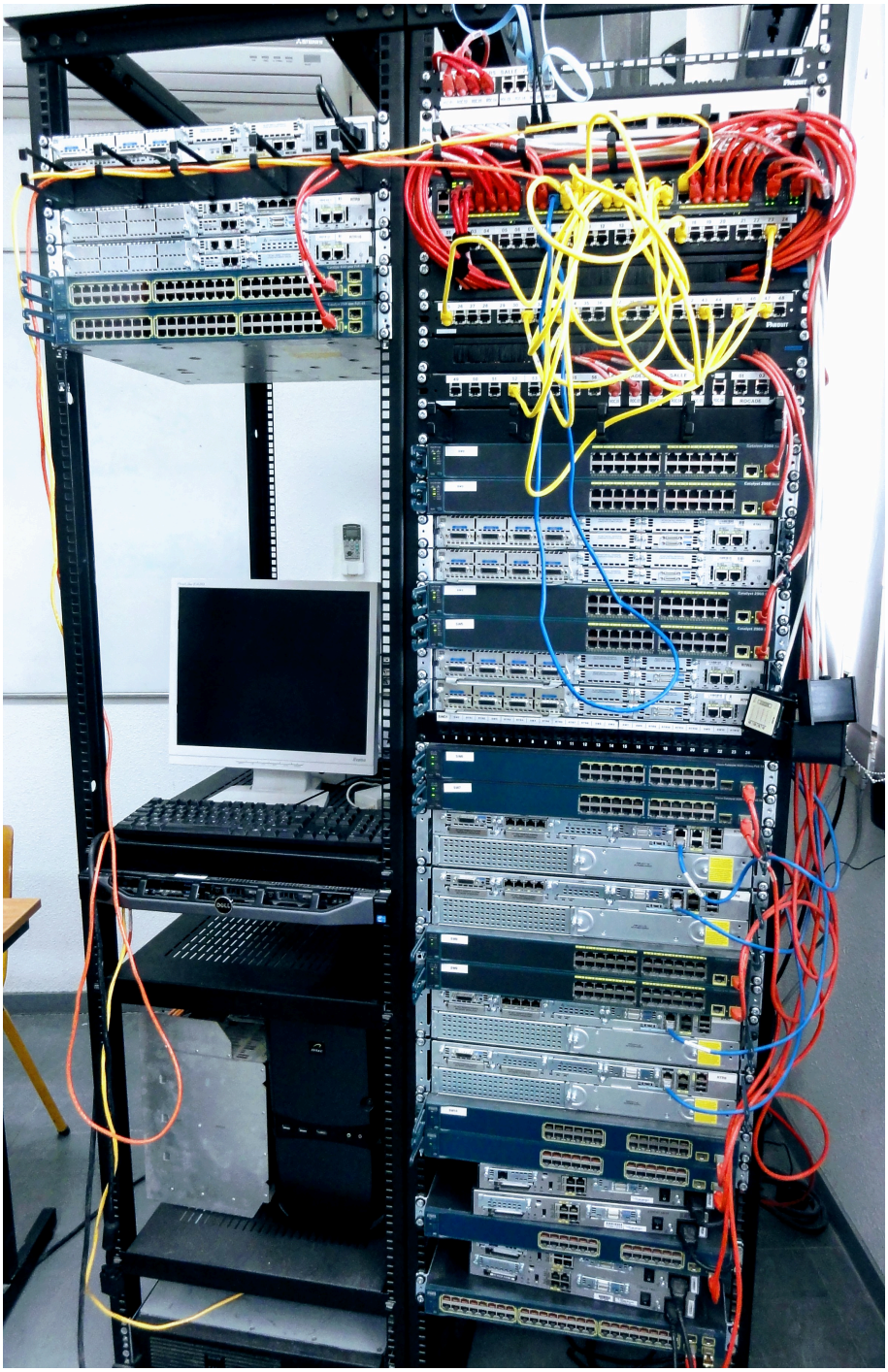


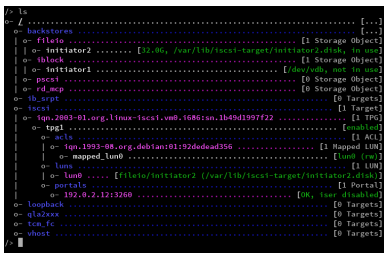
Table des matières

CHAPITRE 1

Introduction au réseau de stockage iSCSI

Résumé

Ce support de travaux pratiques est consacré à l'étude d'une infrastructure de stockage illustrant les technologies DAS (*Direct Attached Storage*), SAN (*Storage Area Network*) et la redondance de niveau 1 (RAID1). Le protocole iSCSI est utilisé pour la partie SAN comme exemple d'accès «en mode bloc» aux unités de stockage réseau. La redondance de niveau 1 utilise les fonctions intégrées au noyau Linux. L'infrastructure proposée montre comment les différentes technologies élémentaires peuvent être combinées pour atteindre les objectifs de haute disponibilité et de sauvegarde.



1.1. Adressage IP des postes de travail

À partir de l'infrastructure proposée dans la section suivante, on constitue des couples de postes de travail qui vont partager le même domaine de diffusion durant la séance de travaux pratiques.

Ces opérations de réaffectation du plan d'adressage IP sont répétées à chaque début de séance de travaux pratiques. Elles s'appuient sur les indications données dans le document *Architecture réseau des travaux pratiques*.

Tableau 1.1. Affectation des adresses et des réseaux IP de la salle 211

Poste 1	Poste 2	Passerelle par défaut
christophsis	corellia	`192.168.143.129/25`
delaya	kasyyyk	`10.30.5.129/26`
korriban	kessel	`192.168.6.17/29`
mygeeto	nelvaan	`10.0.10.33/27`
rattatak	saleucami	`10.3.20.161/27`
taris	teth	`172.20.12.17/29`
utapau	yavin	`10.8.10.65/26`

Tableau 1.2. Affectation des adresses et des réseaux IP de la salle 213

Poste 1	Poste 2	Passerelle par défaut
alderaan	bespin	`10.4.4.1/23`
centares	coruscant	`192.168.109.1/25`
dagobah	endor	`10.0.117.1/27`
felucia	geonosis	`10.7.10.1/23`
hoth	mustafar	`172.19.112.1/26`
naboo	tatooine	`192.168.111.1/25`

Une fois la passerelle du réseau IP affecté à chaque paire de postes de travaux pratiques, il faut rechercher dans le document *Architecture réseau des travaux pratiques* les éléments nécessaires à la connexion physique de ces postes. Les étapes usuelles sont les suivantes :

1. Attribuer une adresse IP à chacun des postes en fonction de l'espace d'adressage du réseau défini.
2. Rechercher le numéro de VLAN correspondant au réseau IP attribué.

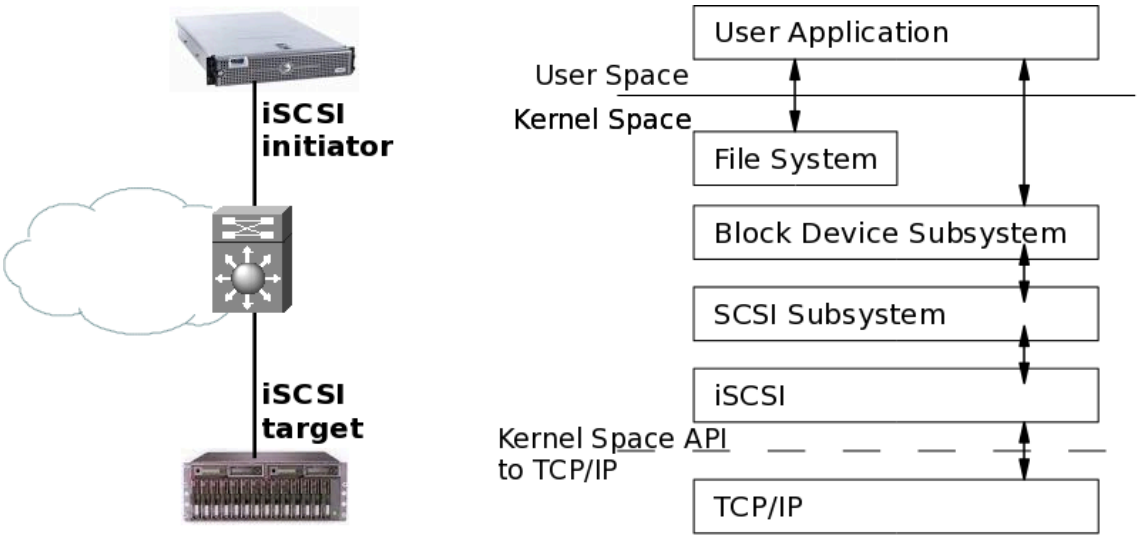
3. Repérer le commutateur sur lequel des ports ont été affectés au VLAN recherché. Connecter les deux postes de travaux pratiques sur les ports identifiés.

4. Configurer les interfaces réseau de chaque poste : adresse, masque et passerelle par défaut. Valider la connectivité IP entre les deux postes puis avec les autres réseaux de l'infrastructure de travaux pratiques.

1.2. Technologie iSCSI et topologie de travaux pratiques

Cette section présente sommairement le protocole iSCSI puis attribue les rôles et les tâches de chacun des postes de travaux pratiques en fonction de la topologie mise en œuvre. Ce support fait suite à la présentation sur le *Stockage Réseau* utilisée en cours.

1.2.1. Bases de la technologie iSCSI



Topologie iSCSI basique - vue complète

La technologie iSCSI dont l'acronyme reprend la définition historique *Internet Small Computer System Interface* est un protocole réseau de stockage basé sur le modèle TCP/IP. Le principe de base consiste à encapsuler des commandes SCSI dans des paquets IP transmis entre un hôte et une unité de disque. Comme les paquets IP peuvent être perdus (et|ou) retransmis, ils peuvent très bien ne pas arriver dans l'ordre d'émission. Le protocole iSCSI doit donc conserver une trace de la séquence de transmission de commandes SCSI. Les commandes sont placées dans une file d'attente dans l'ordre d'émission.

Le protocole iSCSI a initialement été développée par *IBM* et a ensuite été soumise à l'IETF (*Internet Engineering Task Force*). Le standard a été publié par le comité *IP Storage Working Group* en août 2002.

On peut identifier deux fonctions principales dans la technologie iSCSI. La première est la fonction *target*. C'est un système simple qui possède le volume de stockage à publier sur le réseau IP. Ce système peut être matériel ou logiciel. Dans le cas de ces travaux pratiques, il s'agit d'un poste de travail utilisant son second disque dur ou bien un fichier comme unité de stockage SAN. La seconde fonction est baptisée *initiator*. Elle correspond au «client» qui utilise le volume de stockage réseau. Dans le contexte de ce document, l'autre poste de travaux pratiques joue ce rôle de client.

Fondamentalement, iSCSI est un protocole de la famille *Storage Area Network* (SAN). Le client ou *initiator* accède à une unité de stockage en *mode bloc*. Ce mode de fonctionnement est quasi identique à la technologie *Fibre Channel*. Le type de réseau constitue la principale différence entre ces deux technologies. La technologie iSCSI s'appuie sur TCP/IP alors que *Fibre Channel* comprend une définition de réseau propre (FC) qui nécessite des équipements spécifiques.

Ces dernières années, la technologie iSCSI gagne en popularité relativement à son ainée pour plusieurs raisons.

- Le prix des configurations iSCSI peut être bien meilleur marché qu'avec la technologie *Fibre Channel*. Si l'architecture du réseau de de stockage est adaptée, iSCSI devient très attractif.

  Il est important de bien identifier les fonctionnalités réseau que l'on associe à iSCSI pour accroître les performances du stockage. Dans ces fonctions complémentaires on trouve l'agrégation de canaux qui recouvre plusieurs dénominations et plusieurs standards de l'IEEE. Par exemple, elle est baptisée *bonding*

sur les systèmes GNU/Linux et *etherchannel* sur les équipements *Cisco*. Côté standard, le *Link Aggregation Control Protocol* (LACP) pour Ethernet est couvert par les versions *IEEE 802.3ad*, *IEEE 802.1aq* et *IEEE 802.1AX*. L'utilisation de ces techniques est totalement transparente entre équipements hétérogènes. En appliquant ce principe d'agrégation de canaux, on peut pratiquement assimiler les performances de quatre liens Gigabit Ethernet à celles d'un lien *Fibre Channel*. Une autre technique consiste à utiliser aussi plusieurs liens dans une optique et redondance et de balance de charge. Elle est appelée *multipath*.

- L'utilisation d'une technologie réseau unique est nettement moins complexe à administrer. En effet, on optimise les coûts, les temps de formation et d'exploitation en utilisant une architecture de commutation homogène. C'est un des avantages majeurs de la technologie Ethernet sur ses concurrentes.
- Au début de son exploitation, le coût d'un réseau 10 Gigabit Ethernet est prohibitif relativement à toutes les autres solutions. Du point de vue hôte, le point déterminant est l'uniformisation de l'interface réseau. En effet, avec une interface 10GigE on ne devrait plus avoir de distinction entre NIC et HBA.

Aujourd'hui la technologie iSCSI est supportée par tous les systèmes d'exploitation communs. Côté GNU/Linux, plusieurs projets ont vu le jour dans les années qui ont suivi la publication du standard en 2002. Pour la partie *initiator* les développements des deux projets phares ont fusionné pour ne plus fournir qu'un seul code source ; celui disponible à l'adresse *Open-iSCSI*. La partie *target* a suivi un processus analogue et le code source est disponible à l'adresse *Linux-IO : the Linux SCSI Target wiki*.La partie *Kernelspace* de ce dernier code est maintenant directement intégrée dans le noyau Linux. La mise en œuvre du rôle *target* ne nécessite donc que l'installation de la partie utilisateur pour paramétrer le sous-système de stockage du noyau.

À partir des informations fournies à l'adresse *Linux-IO : the Linux SCSI Target wiki*, on recherche le paquet utile de la distribution pour la configuration du rôle *target* :

```
$ aptitude search targetcli
p   targetcli-fb     - Command shell for managing the Linux LIO kernel target
```

Le choix du paquet pour le rôle *initiator* à l'aide de la liste ci-dessous est plus facile en combinant les deux critères de recherche. C'est le paquet open-iscsi qui convient.

```
$ aptitude search "?description(scsi)?description(initiator)"
p   open-iscsi            - iSCSI initiator tools
p   open-isns-discoveryd  - Internet Storage Name Service - iSNS discovery daemon
p   resource-agents       - Cluster Resource Agents
```

1.2.2. Infrastructure de stockage étudiée

Le séquencement des opérations à réaliser lors de la séance de travaux pratiques est décrit dans le tableau ci-dessous. Les deux postes occupent chacun un rôle distinct. Comme le rôle *initiator* demande moins de travail de préparation, c'est à ce poste que l'on confie les essais des outils de *micro-benchmark*.

Tableau 1.3. Attribution des rôles

Rôle *initiator*	**Rôle *target***
Préparation d'une unité de stockage locale pour évaluer les différences entre les accès DAS et SAN	Préparation d'une unité de stockage iSCSI
Recherche et installation du ou des paquet(s) pour le rôle *initiator*	Recherche et installation du ou des paquet(s) pour le rôle *target*
Étude des outils de configuration du service	Étude des outils de configuration du service
Validation manuelle de la configuration SAN iSCSI	
Validation de la configuration système	
Validation de l'authentification mutuelle entre les rôles *initiator* et *target*	
Mise en place de la réplication synchrone avec un tableau RAID1 entre unité de disque locale et le volume iSCSI	Mise en place de la réplication asynchrone avec un volume logique de type *snapshot* de sauvegarde des fichiers images de volume de stockage
Étude comparative des performances d'accès	

1.3. Préparation d'une unité de stockage

Dans cette section on présente les manipulations à effectuer pour préparer une unité de stockage à son utilisation dans une configuration DAS (et|ou) SAN.

> **Avertissement**
>
> Les copies d'écran utilisées dans les réponses correspondent à l'utilisation de machines virtuelles. Les unités de disques apparaissent donc sous le nom `/dev/vd[a-z]`. Les unités de disques physiques d'un système réel apparaissent sous le nom `/dev/sd[a-z]`.

1.3.1. Destruction de la table des partitions

Sachant que les disques des postes de travaux pratiques sont utilisés régulièrement, il est préférable de se placer dans le contexte d'utilisation d'une unité de disque vierge de tout système de fichiers.

Q1. Quelle est la syntaxe d'appel de l'outil parted qui permet de visualiser la table de partition d'une unité de disque ?

Consulter la documentation de parted à l'adresse *Using Parted*.

```
# parted /dev/vda print
Model: Virtio Block Device (virtblk)
Disk /dev/vda: 34,4GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos

Number  Start   End     Size    Type      File system  Flags
 1      1049kB  256MB   255MB   primary   ext2         boot
 2      257MB   34,4GB  34,1GB  extended
 5      257MB   34,4GB  34,1GB  logical               lvm
```

Q2. Quelle est la syntaxe de la commande **dd** qui permet d'effacer complètement la table des partitions d'une unité de disque ?

Utiliser l'aide en ligne de la commande : **`dd --help`**.

La commande suivante écrit des 0 dans les 4 premiers blocs de 512 octets de l'unité de disque.

```
# dd if=/dev/zero of=/dev/vdb bs=512 count=4
4+0 enregistrements lus
4+0 enregistrements écrits
2048 octets (2,0 kB) copiés, 0,00135867 s, 1,5 MB/s

# parted  /dev/vdb print
Error: /dev/vdb: unrecognised disk label
```

1.3.2. Création de la table des partitions et formatage

Une fois que l'on dispose d'une unité de disque vierge, on peut passer à l'étape de création de la table des partitions. Dans le contexte de ces travaux pratiques, cette opération doit être effectuée deux fois sur les postes de travail pour les deux types d'unité de stockage utilisées.

1. Le second disque physique des postes de travail est destiné à intégrer l'unité logique RAID1.

2. Le disque réseau iSCSI est disponible une fois que la configuration du rôle *initiator* est active.

Cette manipulation est l'opération de plus bas niveau qui caractérise un accès réseau au stockage en *mode bloc* et non en mode fichier.

Q3. Quelles sont les instructions de l'outil parted qui permettent de créer une partition primaire unique couvrant la totalité de l'espace de stockage de l'unité de disque ?

Consulter la documentation de parted à l'adresse *Using Parted*.

```
# parted /dev/vdb
GNU Parted 2.3
Using /dev/vdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) print
Error: /dev/vdb: unrecognised disk label
(parted) mklabel gpt
(parted) print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 85,9GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt

Number  Start  End  Size  File system  Name  Flags

(parted) mkpart ext4 1 -1
(parted) print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 85,9GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt

Number  Start   End     Size    File system  Name  Flags
 1      1049kB  85,9GB  85,9GB  ext4         ext4

(parted) quit
```

Q4. Quelle est la commande à utiliser pour les opérations de formatage ? Quel est le rôle de l'option `-T` de cette commande ?

Les informations utiles sont disponibles à la page *Ext4 Howto*. Les pages de manuels détaillent les fonctions des options.

La commande utilisée pour le formatage d'un système de fichiers `ext4`.

```
# dpkg -S `which mkfs.ext4`
e2fsprogs: /sbin/mkfs.ext4
```

L'option `-T` définit le type d'utilisation du système de fichiers à formater suivant sa taille. Les paramètres par défaut sont les suivants :

- `floppy` : 0 < taille < 3Mo
- `small` : 3Mo < taille < 512Mo
- `default` : 512Mo < taille < 4To
- `big` : 4To < taille < 16To
- `huge` : 16To < taille

Q5. Quelle est la syntaxe de la commande de formatage de la partition créée lors de l'étape précédente ?

Des exemples de syntaxe sont disponibles à la page *Ext4 Howto*.

```
# mkfs.ext4 /dev/vdb1
mke2fs 1.42.5 (29-Jul-2012)
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
5242880 i-noeuds, 20971008 blocs
1048550 blocs (5.00%) réservés pour le super utilisateur
Premier bloc de données=0
Nombre maximum de blocs du système de fichiers=4294967296
640 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, 294912, 819200, 884736, 1605632, 2654208,
        4096000, 7962624, 11239424, 20480000

Allocation des tables de groupe : complété
Écriture des tables d'i-noeuds : complété
Création du journal (32768 blocs) : complété
Écriture des superblocs et de l'information de comptabilité du système de
fichiers : complété
```

Q6. Quelle est la syntaxe de la commande de visualisation des attributs du système de fichiers créé lors du formatage ?

Les informations utiles sur les attributs sont fournies à la page *Ext4 Howto*.

```
# tune2fs -l /dev/vdb1
tune2fs 1.42.13 (17-May-2015)
Filesystem volume name:   <none>
Last mounted on:          /var/lib/iscsi-target
Filesystem UUID:          3ccc9115-f206-4a05-86f3-5902ac49b82d
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal ext_attr resize_inode dir_index \
                          filetype needs_recovery extent flex_bg sparse_super \
     large_file huge_file uninit_bg dir_nlink extra_isize
Filesystem flags:         signed_directory_hash
Default mount options:    user_xattr acl
Filesystem state:         clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              4718592
Block count:              18873856
Reserved block count:     943692
Free blocks:              18531693
Free inodes:              4718581
First block:              0
Block size:               4096
Fragment size:            4096
Reserved GDT blocks:      1019
Blocks per group:         32768
Fragments per group:      32768
Inodes per group:         8192
Inode blocks per group:   512
Flex block group size:    16
Filesystem created:       Sun Aug 23 17:20:15 2015
Last mount time:          Sun Aug 23 17:36:50 2015
Last write time:          Sun Aug 23 17:36:50 2015
Mount count:              2
Maximum mount count:      -1
Last checked:             Sun Aug 23 17:20:15 2015
Check interval:           0 (<none>)
Lifetime writes:          1284 MB
Reserved blocks uid:      0 (user root)
Reserved blocks gid:      0 (group root)
First inode:              11
Inode size:               256
Required extra isize:     28
Desired extra isize:      28
Journal inode:            8
Default directory hash:   half_md4
Directory Hash Seed:      8edb7b6a-5586-4d32-a02a-de231242f353
Journal backup:           inode blocks
```

1.3.3. Montage manuel d'un volume de stockage

Une fois qu'un volume de stockage a été partitionné et formaté, il faut le *monter* dans l'arborescence du système de fichiers de la machine de façon à lire et écrire des données dedans.

Q7. Comment obtenir l'identifiant du volume de stockage à ajouter au système de fichiers ?

Consulter la liste des utilitaires fournis avec le paquet util-linux. Il faut se rappeler que la représentation fichier d'un périphérique de stockage se distingue par son mode d'accès : le mode bloc.

La commande à utiliser est **blkid**. Dans l'exemple de la partition `/dev/vdb1`, on obtient le résultat suivant.

```
# blkid /dev/vdb1
/dev/vdb1: UUID="224f6aad-16c0-4923-8949-0628a8e10228" TYPE="ext4"
```

C'est cet identifiant que l'on doit utiliser pour compléter la configuration système et rendre le montage du périphérique de stockage permanent.

Q8. Dans quel fichier de configuration trouve-t-on la liste des périphériques montés lors de l'initialisation du système ?

Consulter la liste des fichiers du paquet util-linux.

Le fichier recherché est `/etc/fstab`. Il contient bien la liste des points de montage. Dans l'exemple ci-dessous, la racine et la partition d'échange utilisée en cas de saturation des ressources RAM du système.

```
# grep -v '^#' /etc/fstab
UUID=78cc7982-fa2e-4498-af98-45272c7726c9 /    ext4    errors=remount-ro 0       1
UUID=56c21189-6cb3-4a6e-a6f6-ccf3e28db8b0 none swap    sw              0       0
```

Q9. Quelle est la commande qui donne la liste des montages en cours d'utilisation sur le système ? Quelle est l'option qui permet de scruter les entrées du fichier recherché dans la question précédente et de monter tous les points non encore utilisés ?

La commande est fournie par le paquet du même nom.

Le paquet mount fournit la commande du même nom. Cette commande liste tous les montages actifs du système. La liste comprend les systèmes de fichiers virtuels qui représentent l'état courant des paramètres du noyau ainsi que les systèmes de fichiers physiques qui correspondent aux volumes de stockage effectifs. En reprenant l'exemple utilisé auparavant et en filtrant les systèmes de fichiers virtuels, on obtient :

```
# mount | grep "/dev/vd"
/dev/vda1 on / type ext4 (rw,relatime,errors=remount-ro,data=ordered)
```

L'option de montage des entrées inutilisées du fichier `/etc/fstab` est `-a`. Elle doit être utilisée dans la question suivante.

Q10. Comment compléter la configuration système pour assurer le montage du nouveau périphérique ?

Il faut utiliser les réponses aux questions précédentes pour valider le nouveau point de montage.

- Création du point de montage dans l'arborescence du système

```
# mkdir /var/lib/iscsi-target
```

- Ajout d'une ligne dans le fichier `/etc/fstab` avec l'identifiant du périphérique à ajouter

```
# echo UUID=3ccc9115-f206-4a05-86f3-5902ac49b82d \
  /var/lib/iscsi-target/ \
  ext4 \
  defaults \
  0      2 >>/etc/fstab
```

- Montage du nouveau périphérique

```
# mount -a
```

- Nouvelle liste des montages actifs

```
# mount | grep "/dev/vd"
/dev/vda1 on / type ext4 (rw,relatime,errors=remount-ro,data=ordered)
/dev/vdb1 on /var/lib/iscsi-target type ext4 (rw,relatime,data=ordered)
```

1.4. Configuration du système initiator

Dans cette partie, on prépare le système auquel on a attribué le rôle *initiator*. Ce système est celui qui utilise le volume de stockage mis à disposition sur le réseau par le rôle *target*.

1.4.1. Sélection du paquet et lancement du service

Q11. Comment identifier et installer le paquet correspondant au rôle *initiator* ?

En effectuant une recherche simple dans le catalogue des paquets disponibles, on obtient la liste des paquets dont le nom contient la chaîne de caractères `iscsi`.

```
# aptitude search iscsi
p   iscsitarget         - iSCSI Enterprise Target userland tools
p   iscsitarget-dkms    - iSCSI Enterprise Target kernel module source - dkms version
p   iscsitarget-source  - iSCSI Enterprise Target kernel module source
p   open-iscsi          - High performance, transport independent iSCSI implementation
```

On remarque que le paquet `open-iscsi` est le seul qui ne soit pas identifié comme appartenant à la catégorie *target*.

```
# aptitude install open-iscsi
```

Q12. Comment lancer le service *initiator* et valider son fonctionnement ?

À partir de la liste des fichiers du paquet on peut identifier les éléments de démarrage et de configuration du service.

```
# dpkg -L open-iscsi
/.
/etc
/etc/init.d
/etc/init.d/open-iscsi
/etc/init.d/umountiscsi.sh
/etc/network
/etc/network/if-up.d
/etc/default
/etc/default/open-iscsi
/etc/iscsi
/etc/iscsi/iscsid.conf
/etc/iscsi/initiatorname.iscsi
/usr
/usr/share
/usr/share/man
/usr/share/man/man8
/usr/share/man/man8/iscsiadm.8.gz
/usr/share/man/man8/iscsid.8.gz
/usr/share/man/man8/iscsi_discovery.8.gz
/usr/share/man/man8/iscsistart.8.gz
/usr/share/man/man8/iscsi-iname.8.gz
/usr/share/initramfs-tools
/usr/share/initramfs-tools/hooks
/usr/share/initramfs-tools/hooks/iscsi
/usr/share/initramfs-tools/scripts
/usr/share/initramfs-tools/scripts/local-top
/usr/share/initramfs-tools/scripts/local-top/iscsi
/usr/share/doc
/usr/share/doc/open-iscsi
/usr/share/doc/open-iscsi/copyright
/usr/share/doc/open-iscsi/README.Debian.gz
/usr/share/doc/open-iscsi/README.gz
/usr/share/doc/open-iscsi/changelog.Debian.gz
/usr/share/doc/open-iscsi/changelog.gz
/usr/sbin
/usr/sbin/iscsi-iname
/usr/sbin/iscsid
/usr/sbin/iscsi_discovery
/usr/sbin/iscsistart
/usr/bin
/usr/bin/iscsiadm
/var
/var/lib
/var/lib/open-iscsi
```

Le lancement du service se fait de façon classique à partir de l'arborescence des scripts des niveaux de démarrage (*runlevels*).

```
# /etc/init.d/open-iscsi restart
```

La même opération avec systemd utilise la syntaxe suivante :

```
# systemctl restart open-iscsi
```

On peut ensuite consulter les journaux système pour valider l'initialisation du ou des démons.

```
# grep -i iscsi /var/log/syslog
Loading iSCSI transport class v2.0-870.
iscsi: registered transport (tcp)
iscsi: registered transport (iser)
```

On retrouve les informations correspondantes aux messages ci-dessus dans la liste des processus actifs.

```
# ps aux | grep -i iscsi
root      3479  0.0  0.0      0     0 ?        S<   16:28   0:00 [iscsi_eh]
root      3487  0.0  0.0   4980   472 ?        Ss   16:28   0:00 /usr/sbin/iscsid
root      3488  0.0  0.6   5480  3280 ?        S<Ls 16:28   0:00 /usr/sbin/iscsid
```

1.4.2. Tests de fonctionnement du service

Q13. Quelle est la commande principale du rôle *initiator* qui permet de tester la connectivité iSCSI ?

Consulter la liste des fichiers du paquet `open-iscsi`.

En consultant la liste donnée ci-dessus, on ne relève qu'un seul outil exécutable : la commande **iscsiadm**.

Q14. Quelles sont les options de découverte proposées avec cette commande ? Donner un exemple fournissant l'identifiant de l'unité de stockage réseau visible.

Consulter les pages de manuels de la commande identifiée dans la question précédente.

À partir du poste *initator* numéro 1, le seul volume de stockage visible est :

```
# iscsiadm -m discovery --type sendtargets --portal=192.0.2.12:3260
192.0.2.12:3260,1 iqn.2003-01.org.linux-iscsi.vm0.i686:sn.1b49d1997f22
[2001:db8:feb2:2:b8ad:ff:feca:fe00]:3260,1 iqn.2003-01.org.linux-iscsi.vm0.i686:sn.1b49d1997f22
192.0.2.12:3260,1 iqn.2003-01.org.linux-iscsi.target.i686:sn.637018090566
[2001:db8:feb2:2:b8ad:ff:feca:fe00]:3260,1 iqn.2003-01.org.linux-iscsi.target.i686:sn.63701809056
```

Q15. Quel est l'identifiant à communiquer ou à paramétrer pour le système *initiator* soit reconnu côté système *target* ?

Rechercher les informations relatives au nommage iSCSI dans les outils et les fichiers fournis avec le paquet de gestion du rôle *initiator*.

Le filtrage de la liste des fichiers fournis avec le paquet open-iscsi donne le résultat suivant.

```
# dpkg -L open-iscsi | grep name
/etc/iscsi/initiatorname.iscsi
/usr/share/man/man8/iscsi-iname.8.gz
/usr/sbin/iscsi-iname
```

- Le fichier `/etc/iscsi/initiatorname.iscsi` contient l'identifiant du système à communiquer au système *target* pour que celui-ci l'associe dans la rubrique des listes de contrôle d'accès : `acls`.

  Dans le contexte de ces travaux pratiques, on se contente de relever l'identifiant généré automatiquement lors de l'installation du paquet et de l'implanter dans la liste de contrôle d'accès créée avec targetcli sur le système *target*.

  Côté *initiator*, on lit l'identifiant iqn

  ```
  # grep -v ^# /etc/iscsi/initiatorname.iscsi
  InitiatorName=iqn.1993-08.org.debian:01:9d11913c78ac
  ```

  Côté *target*, on créé la liste de contrôle d'accès qui associe l'unité logique SCSI au système *initiator*.

  ```
  /iscsi/iqn.20...18090566/tpg1> ls
  o- tpg1 ..................................................................... [enabled]
    o- acls ....................................................................... [1 ACL]
    | o- iqn.1993-08.org.debian:01:9d11913c78ac ..................... [1 Mapped LUN]
    |   o- mapped_lun0 ............................................... [lun0 (rw)]
    o- luns ....................................................................... [1 LUN]
    | o- lun0 ..................................... [iblock/initiator1 (/dev/vdb)]
    o- portals ................................................... [2 Portals]
      o- 192.0.2.12:3260 ..................................... [OK, iser disabled]
      o- 2001:db8:feb2:2:b8ad:ff:feca:fe00:3260 ............... [OK, iser disabled]
  ```

- La commande **iscsi-iname** sert à générer un nouvel identifiant conforme au forat iqn. Elle permet de fournir un nouvel identifiant compatible avec la nomenclature de l'infrastructure de stockage d'un opérateur.

Q16. Quelles sont les options de connexion proposées avec cette même commande ? Donner un exemple illustrant l'établissement d'une connexion.

Consulter les pages de manuels de la commande identifiée précédemment.

```
# iscsiadm -m node -T iqn.2003-01.org.linux-iscsi.target.i686:sn.1b49d1997f22 -p 192.0.2.12 -l
Logging in to [iface: default, target: iqn.2003-01.org.linux-iscsi.target.i686:sn.1b49d1997f22, p
iscsiadm: Could not login to [iface: default, target: iqn.2003-01.org.linux-iscsi.target.i686:sn.
iscsiadm: initiator reported error (24 - iSCSI login failed due to authorization failure)
iscsiadm: Could not log into all portals
```

Dans l'exemple ci-dessus, la connexion sans authentification a échoué faute d'autorisations côté rôle *target*. Comme nous sommes dans un contexte de travaux pratiques, il faut paramétrer deux attributs spécifiques : `authentication=0` et `demo_mode_write_protect=0`.

```
# iscsiadm -m node -T iqn.2003-01.org.linux-iscsi.target.i686:sn.637018090566 -p 192.0.2.12 -l
Logging in to [iface: default, target: iqn.2003-01.org.linux-iscsi.target.i686:sn.637018090566, p
Login to [iface: default, target: iqn.2003-01.org.linux-iscsi.target.i686:sn.637018090566, portal
```

La connexion est maintenant établie et le volume de stockage réseau est disponible sous forme d'unité logique SCSI.

Q17. Comment obtenir les caractéristiques de l'unité de stockage iSCSI utilisée ?

Consulter les journaux système.

Voici un extrait du fichier `/var/log/syslog`.

```
# egrep '(sd|scsi)' /var/log/syslog
initiator1 kernel: [18614.778860] scsi host8: iSCSI Initiator over TCP/IP
initiator1 kernel: [18615.035238] scsi 8:0:0:0: Direct-Access     LIO-ORG  IBLOCK          4.0
initiator1 kernel: [18615.038358] scsi 8:0:0:0: Attached scsi generic sg1 type 0
initiator1 kernel: [18615.067546] sd 8:0:0:0: [sda] 150994944 512-byte logical blocks: (77.3 GB/7
initiator1 kernel: [18615.070885] sd 8:0:0:0: [sda] Write Protect is off
initiator1 kernel: [18615.070893] sd 8:0:0:0: [sda] Mode Sense: 43 00 10 08
initiator1 kernel: [18615.071854] sd 8:0:0:0: [sda] Write cache: enabled, read cache: enabled, su
initiator1 kernel: [18615.117019]  sda:
initiator1 kernel: [18615.120840] sd 8:0:0:0: [sda] Attached SCSI disk
initiator1 iscsid: Connection3:0 to [target: iqn.2003-01.org.linux-iscsi.target.i686:sn.637018090
                                 portal: 192.0.2.12,3260] through [iface: default] is operational
```

Q18. Donner la liste des entrées de périphériques de stockage créées par le démon `udev` ?

Lister les entrées de périphériques mode bloc de l'arborescence système.

Les fichiers de description des périphériques mode bloc sont tous situés dans le répertoire `/dev/`. En reprenant l'exemple ci-dessus, on obtient :

```
# ls -lA /dev/[v,s]d*
brw-rw---- 1 root disk   8,  0 sept.  1 14:38 /dev/sda
brw-rw---- 1 root disk 254,  0 sept.  1 11:28 /dev/vda
brw-rw---- 1 root disk 254,  1 sept.  1 11:28 /dev/vda1
brw-rw---- 1 root disk 254,  2 sept.  1 11:28 /dev/vda2
brw-rw---- 1 root disk 254,  5 sept.  1 11:28 /dev/vda5
brw-rw---- 1 root disk 254, 16 sept.  1 11:28 /dev/vdb
```

L'entrée `/dev/sda` correspond à l'unité de disque iSCSI. Le volume de stockage est donc bien vu de façon transparente comme un périphérique local au système accessible en mode bloc. Il entre bien dans la catégorie SAN ou *Storage Area Network*.

1.4.3. Réinitialisation de session iSCSI

Dans le cas d'une reconfiguration avec un autre hôte *target* ou dans le cas d'un dépannage, il est utile de pouvoir reprendre les paramètres du rôle *initiator*.

Q19. Comment obtenir la liste des sessions actives avec le système *target* ?

Consulter les pages de manuels de la commande de configuration du rôle *initiator* : **iscsiadm**.

C'est le mode `session`, documenté dans les pages de manuels de la commande **iscsiadm**, qui permet de répondre à la question.

```
# iscsiadm -m session
tcp: [3] 192.0.2.12:3260,1 iqn.2003-01.org.linux-iscsi.target.i686:sn.637018090566 (non-flash)
tcp: [4] [2001:db8:feb2:2:b8ad:ff:feca:fe00]:3260,1 iqn.2003-01.org.linux-iscsi.target.i686:sn.63
```

Q20. Comment libérer toutes les sessions actives depuis le système *initiator* ?

Consulter les pages de manuels de la commande de configuration du rôle *initiator* : **iscsiadm**.

Pour cette question, c'est le mode `node` qui nous intéresse.

```
# iscsiadm -m node -U all
Logging out of session [sid: 3, target: iqn.2003-01.org.linux-iscsi.target.i686:sn.637018090566,
Logout of [sid: 3, target: iqn.2003-01.org.linux-iscsi.target.i686:sn.637018090566, portal: 2001:
```

Q21. Comment effacer les informations sur les systèmes *target* déjà découverts en cas de problème de configuration ?

Consulter les pages de manuels de la commande de configuration du rôle *initiator* : **iscsiadm**.

Toutes les manipulations sur les systèmes *target* découverts dépendent du mode `discovery` et l'opération à utiliser est `delete`.

```
# iscsiadm -m discovery -p 192.0.2.12 -o delete
```

Il suffit ensuite de reprendre la découverte décrite à la question Q : Q14.

1.4.4. Configuration système permanente

Une fois la connexion à la ressource iSCSI testée, on peut passer à la configuration système de façon à retrouver le volume de stockage après une réinitialisation du système *initiator*.

Q22. Comment rendre la connexion à l'unité de stockage automatique lors de l'initialisation du système *initiator* ?

Rechercher dans la liste des fichiers du paquet open-iscsi les éléments relatifs à la configuration système. Éditer le fichier de configuration principal de façon à rendre automatique le lancement du service.

Au niveau système, les fichiers de configuration sont nécessairement dans le répertoire `/etc/`.

```
# dpkg -L open-iscsi | grep '/etc/'
/etc/default
/etc/default/open-iscsi
/etc/init.d
/etc/init.d/open-iscsi
/etc/init.d/umountiscsi.sh
/etc/iscsi
/etc/iscsi/iscsid.conf
/etc/iscsi/initiatorname.iscsi
/etc/network
/etc/network/if-up.d
```

Le fichier `/etc/iscsi/iscsid.conf` contient une directive dans la section *Startup settings* qui rend automatique l'accès à une ressource déjà enregistrée. Voici le contenu de cette section extraite du fichier de configuration.

```
#*****************
# Startup settings
#*****************

# To request that the iscsi initd scripts startup a session set to "automatic".
node.startup = automatic
```

Q23. Comment connaître l'état et la liste d'une session iSCSI active ?

Consulter les pages de manuels de la commande de configuration du rôle *initiator* : **iscsiadm**.

Il existe un mode `session` dédié aux manipulations sur les sessions. La commande de test la plus simple est la suivante.

```
# iscsiadm -m session
tcp: [3] 192.0.2.12:3260,1 iqn.2003-01.org.linux-iscsi.target.i686:sn.637018090566 (non-flash)
tcp: [4] [2001:db8:feb2:2:b8ad:ff:feca:fe00]:3260,1 iqn.2003-01.org.linux-iscsi.target.i686:sn.63
```

La copie d'écran ci-dessus indique deux sessions actives. Si la liste est vide, il n'y a pas de session iSCSI en cours.

Il est possible d'obtenir davantage d'informations sur les paramètres de session en cours à l'aide de l'option `-P` suivie d'un numéro désignant le niveau de détail attendu. Par exemple, la commande `iscsiadm -m session -P 3` affiche les paramètres sur les interfaces réseau utilisées, etc.

Q24. Comment retrouver un point de montage unique du volume de stockage iSCSI après réinitialisation du système *initiator* ?

Créer un répertoire de montage et rechercher les options utiles dans les pages de manuels des commandes **mount**, **systemd.mount** et **blkid**. Éditer le fichier `/etc/fstab` en utilisant les options sélectionnées. Noter que le fichier `fstab` possède ses propres pages de manuels.

La création du répertoire destiné au montage du volume de stockage iSCSI ne pose pas de problème.

```
# mkdir /var/cache/iscsi-storage
```

La commande **blkid** permet d'obtenir l'identifiant unique du volume de stockage. Dans la copie d'écran ci-dessous, la partition `/dev/sda1` correspond au résultat de l'établissement de la session iSCSI et l'identification du système de fichiers utilisé (`btrfs`) correspond au résultat du formatage de la partition. Ce ne sont que des exemples particuliers au contexte de la maquette utilisée.

```
# blkid /dev/sda1
/dev/sda1: UUID="11924824-00f1-4735-bd30-4bacaa3cbde0" UUID_SUB="09505fb1-d90c-4d05-b9e7-b4a0454a
           TYPE="btrfs" PARTLABEL="iSCSI-LUN0" PARTUUID="0da2be8f-ff7b-40d1-a720-15f08c456351"
```

Le choix des options à utiliser lors de l'édition du fichier `/etc/fstab` constitue le point délicat de cette question.

```
UUID=11924824-00f1-4735-bd30-4bacaa3cbde0       /var/cache/iscsi-storage        btrfs   noauto,x-
```

- Le choix de la valeur `UUID` se fait à partir du résultat de la commande **blkid** donné ci-dessus.
- Le point de montage `/var/cache/iscsi-storage` a lui aussi été défini ci-dessus.
- Le système de fichiers utilisé est, là encore, connu : `btrfs`.

Les trois paramètres suivants sont spécifiques au contexte iSCSI.

- L'option `noauto` empêche le déclenchement du montage lors de la scrutation du fichier `/etc/fstab`. Les entrées présentes dans ce fichier doivent être disponibles très tôt dans le processus d'initialisation des services du système. Or, un volume de stockage réseau iSCSI n'est pas nécessairement disponible au moment du parcours des entrées en question.
- L'option `x-systemd.automount` provoque la création d'une unité d'automontage (au sens systemd). Le principe de l'automontage veut que l'opération de montage soit effective au moment du parcours de l'arborescence `/var/cache/iscsi-storage` par un utilisateur ou une application. Autrement dit, tant que l'arborescence n'est pas utilisée, le montage n'est pas réalisé et l'initialisation du système *initiator* se poursuit normalement.
- L'option `_netdev` spécifie que le système de fichiers réside sur un périphérique nécessitant des accès réseau. Il est donc inutile d'y accéder tant qu'aucune interface réseau n'est active.

1.5. Configuration du système target

Dans cette partie, on prépare le système auquel on a attribué le rôle *target* à l'aide de l'outil targetcli-fb.

1.5.1. Installation de l'outil de paramétrage du rôle target

Q25. Quel est le paquet qui contient les éléments de configuration du service dans l'espace utilisateur ?

On consulte le site de référence à l'adresse *Linux-IO : the Linux SCSI Target wiki* pour identifier l'outil principal et on effectue ensuite une recherche dans la liste des paquets.

```
# aptitude search targetcli
p   targetcli-fb   - Command shell for managing the Linux LIO kernel target
```

Q26. Comment installer le paquet identifié à la question précédente ?

Consulter les pages de manuels de la commande **aptitude**.

```
# aptitude install targetcli-fb
Les NOUVEAUX paquets suivants vont être installés :
gir1.2-glib-2.0{a} libdbus-glib-1-2{a} libgirepository-1.0-1{a}
python-rtslib-fb{a} python-six{a} python3-configshell-fb{a} python3-dbus{a}
python3-gi{a} python3-pyparsing{a} python3-rtslib-fb{a} python3-six{a}
python3-urwid{a} targetcli-fb
0 paquets mis à jour, 13 nouvellement installés, 0 à enlever et 0 non mis à
jour.
Il est nécessaire de télécharger 1 524 ko d'archives. Après dépaquetage, 5 615
ko seront utilisés.
Voulez-vous continuer ? [Y/n/?]
```

1.5.2. Configuration du rôle target

La technologie iSCSI dispose d'un schéma de nommage propre défini dans le document standard RFC3721 *Internet Small Computer Systems Interface (iSCSI) Naming and Discovery*. Le format retenu ici est baptisé iqn (*iSCSI Qualified Name*). Il s'agit d'une chaîne qui débute par "iqn." suivie d'une date au format AAAA-MM, du nom de l'autorité qui a attribué le nom (le nom de domaine à l'envers), puis une autre chaîne unique qui identifie le nœud de stockage.

On a choisi de n'utiliser aucun mécanisme d'authentification sachant que la configuration initiale se fait dans un contexte de travaux pratiques et non d'exploitation sur un réseau réel.

Q27. Quelles sont les étapes à suivre pour publier un volume de stockage sur le réseau à partir de l'interface de l'outil targetcli ?

Ici aussi, il faut consulter le site de référence à l'adresse *Linux-IO : the Linux SCSI Target wiki* pour identifier les différentes étapes.

La copie d'écran ci-dessous liste les familles de stockage disponibles.

```
# targetcli
targetcli shell version 2.1.fb43
Copyright 2011-2013 by Datera, Inc and others.
For help on commands, type 'help'.

/> ls
o- / ..................................................... [...]
  o- backstores .......................................... [...]
  | o- block .............................. [Storage Objects: 0]
  | o- fileio ............................. [Storage Objects: 0]
  | o- pscsi .............................. [Storage Objects: 0]
  | o- ramdisk ............................ [Storage Objects: 0]
  o- iscsi ........................................ [Targets: 0]
  o- loopback ..................................... [Targets: 0]
  o- vhost ........................................ [Targets: 0]
/>
```

- La section *backstores* désigne les volumes de stockage à publier sur le réseau. Ici, les deux items intéressants sont *fileio* et *iblock*. Le premier fait correspondre un fichier du système local au volume à publier. Le second fait correspondre une unité de disque physique au volume à publier.
- La section *iscsi* sert à définir une «cible» (*target*) qui comprend au moins une unité logique (LUN en vocabulaire SCSI) et un point de contact réseau.

Partie stockage local : *backstores*

Q28. Quelles sont les opérations à effectuer définir un disque physique comme volume de stockage ?

Il faut consulter le site de référence et repérer les options du menu `block`.

La création du volume se fait à l'aide de la commande ci-dessous.

```
/> cd backstores/block
/backstores/block> ls
o- block ..................................................... [Storage Objects: 0]
/backstores/block> create initiator1 /dev/vdb
Created block storage object initiator1 using /dev/vdb.
/backstores/block> ls
o- block ..................................................... [Storage Objects: 1]
  o- initiator1 ................... [/dev/vdb (72.0GiB) write-thru deactivated]
```

Q29. Quelles sont les opérations à effectuer pour définir un fichier comme volume de stockage ?

Il faut consulter le site de référence et repérer les options du menu `fileio`.

La création du volume se fait à l'aide de la commande ci-dessous.

```
/backstores/fileio> create storage_file /var/lib/iscsi-target/storage-for-myinitiator 32G
Using buffered mode.
Created fileio storage_file.
/backstores/fileio> ls
o- fileio ..................................................... [2 Storage Objects]
  o- initiator2 ................. [32.0G, /var/lib/iscsi-target/initiator2.disk, not in use]
  o- storage_file ....... [32.0G, /var/lib/iscsi-target/storage-for-myinitiator, not in use]
```

Dans l'exemple ci-dessus on a créé un nouvel objet dans le dépôt des volumes de stockage appelé `storage_file`. Dans la même commande on lui a attribué une capacité de 32Go. On a aussi précisé le chemin d'accès à ce fichier.

Il faut noter que l'espace effectivement occupé par le fichier `/var/lib/iscsi-target/storage-for-myinitiator` correspond à celui utilisé côté *initiator*. La commande de l'exemple ci-dessus a provoqué la création d'un fichier vide.

Partie iSCSI

Q30. Quelles sont les opérations à effectuer pour définir une nouvelle cible iSCSI ?

Il faut consulter le site de référence et repérer les options du menu `iscsi`. Attention ! Une cible iSCSI comprend plusieurs attributs.

1. **Nommage de la cible au format iqn.**
   Si le nom de la cible n'est pas fourni avec la commande **create**, il est généré automatiquement.

   ```
   /> cd iscsi/
   /iscsi> create
   Created target iqn.2003-01.org.linux-iscsi.target.i686:sn.bf156efd0f2e.
   Selected TPG Tag 1.
   Created TPG 1.
   ```

   C'est après cette première opération que les attributs apparaissent pour la nouvelle cible.

   ```
   /iscsi> ls
   o- iscsi ................................................... [1 Target]
     o- iqn.2003-01.org.linux-iscsi.target.i686:sn.bf156efd0f2e ...... [1 TPG]
       o- tpg1 ..................................................... [enabled]
         o- acls ................................................... [0 ACLs]
         o- luns ................................................... [0 LUNs]
         o- portals ................................................ [0 Portals]
   ```

2. **Affectation de l'unité logique à la cible iSCSI.**
   Les numéros d'unités logiques SCSI ou LUNs sont affectés automatiquement. Ici, l'unité `lun0` correspond à la première association faite depuis le dépôt des volumes de stockage.

```
/iscsi> cd iqn.2003-01.org.linux-iscsi.target.i686:sn.bf156efd0f2e/tpg1/
/iscsi/iqn.20...6efd0f2e/tpg1> luns/ create /backstores/iblock/initiator1
Selected LUN 0.
Created LUN 0.
/iscsi/iqn.20...6efd0f2e/tpg1> ls
o- tpg1 ..................................... [enabled]
  o- acls ..................................... [0 ACLs]
  o- luns ..................................... [1 LUN]
  | o- lun0 ................. [iblock/initiator1 (/dev/vdb)]
  o- portals ................................... [0 Portals]
```

3. **Ouverture du point de contact réseau pour cette cible iSCSI.**
Un même point de contact peut être en écoute sur plusieurs adresses IP. Dans l'exemple ci-dessous on ouvre une configuration double pile IPv4 et IPv6.

```
/iscsi/iqn.20...6efd0f2e/tpg1> portals/ create 192.0.2.12 3260
Created network portal 192.0.2.12:3260.
/iscsi/iqn.20...6efd0f2e/tpg1> portals/ create 2001:db8:feb2:2:b8ad:ff:feca:fe00 3260
Created network portal 2001:db8:feb2:2:b8ad:ff:feca:fe00:3260.
/iscsi/iqn.20.../tpg1/portals> ls
o- portals ..................................... [2 Portals]
  o- 192.0.2.12:3260 ............................ [OK, iser disabled]
  o- 2001:db8:feb2:2:b8ad:ff:feca:fe00:3260 ..... [OK, iser disabled]
```

On peut sortir de l'outil targetcli pour vérifier que le service réseau est bien accessible.

```
# ss -tan
State       Recv-Q Send-Q                          Local Address:Port  Peer Address:Port
LISTEN      0      128                                         *:22               *:*
LISTEN      0      20                                      127.0.0.1:25               *:*
LISTEN      0      256                                  192.0.2.12:3260            *:*
LISTEN      0      128                                         :::22              :::*
LISTEN      0      20                                        ::1:25               :::*
LISTEN      0      256       2001:db8:feb2:2:b8ad:ff:feca:fe00:3260              :::*
```

Enfin, on peut aussi vérifier que le service est ouvert *côté initiator* à l'aide de la fonction de découverte.

```
root@initiator1:~# iscsiadm -m discovery --type sendtargets --portal=192.0.2.12:3260
192.0.2.12:3260,1 iqn.2003-01.org.linux-iscsi.target.i686:sn.bf156efd0f2e
[2001:db8:feb2:2:b8ad:ff:feca:fe00]:3260,1 iqn.2003-01.org.linux-iscsi.target.i686:sn.bf156ef
```

4. **Création d'une liste de contrôle d'accès.**
Même si le service réseau et la fonction découverte sont ouverts, le volume de stockage réseau n'est pas encore accessible. La connexion depuis l'hôte *initiator* échoue et on obtient le message suivant.

```
root@initiator1:~# iscsiadm -m node -T iqn.2003-01.org.linux-iscsi.target.i686:sn.bf156efd0f2
Logging in to [iface: default, target: iqn.2003-01.org.linux-iscsi.target.i686:sn.bf156efd0f2
iscsiadm: Could not login to [iface: default, target: iqn.2003-01.org.linux-iscsi.target.i686
iscsiadm: initiator reported error (24 - iSCSI login failed due to authorization failure)
iscsiadm: Could not log into all portals
```

Côté hôte *target*, les journaux système font apparaître un message du type suivant.

```
iSCSI Initiator Node: iqn.1993-08.org.debian:01:9d11913c78ac is not authorized to access iSCS
iSCSI Login negotiation failed.
```

Il est donc necessaire d'autoriser l'accès depuis l'hôte *initiator*. Dans l'outil targetcli, on configure l'attribut `acls` de la cible iSCSI.

```
/iscsi/iqn.20...6efd0f2e/tpg1> acls/ create iqn.1993-08.org.debian:01:9d11913c78ac
Created Node ACL for iqn.1993-08.org.debian:01:9d11913c78ac
Created mapped LUN 0.
/iscsi/iqn.20...6efd0f2e/tpg1> ls
o- tpg1 ..................................................... [enabled]
  o- acls ....................................................... [1 ACL]
  | o- iqn.1993-08.org.debian:01:9d11913c78ac ........... [1 Mapped LUN]
  |   o- mapped_lun0 ..................................... [lun0 (rw)]
  o- luns ....................................................... [1 LUN]
  | o- lun0 ........................... [iblock/initiator1 (/dev/vdb)]
  o- portals ................................................. [2 Portals]
    o- 192.0.2.12:3260 ............................ [OK, iser disabled]
    o- 2001:db8:feb2:2:b8ad:ff:feca:fe00:3260 ...... [OK, iser disabled]
```

Ce n'est pas terminé ! Par défaut, une cible iSCSI n'est accessible qu'après authentification. Il est donc nécessaire de désactiver cette authentification pour tester l'accès depuis l'hôte *initiator*.

```
/iscsi/iqn.20...6efd0f2e/tpg1> set attribute authentication=0 demo_mode_write_protect=0
Parameter authentication is now '0'.
Parameter demo_mode_write_protect is now '0'.
```

Finalement, le volume de stockage est mis à disposition de l'hôte *initiator*.

```
root@initiator1:~# iscsiadm -m node -T iqn.2003-01.org.linux-iscsi.target.i686:sn.bf156efd0f2
Logging in to [iface: default, target: iqn.2003-01.org.linux-iscsi.target.i686:sn.bf156efd0f2
Login to [iface: default, target: iqn.2003-01.org.linux-iscsi.target.i686:sn.bf156efd0f2e, po
```

1.6. Configuration de l'authentification CHAP

Dans cette partie, on suppose que tous les tests précédents ont été effectués avec succès et que les échanges entre les systèmes *target* et *initiator* sont validés.

On s'intéresse maintenant à l'authentification entre ces mêmes systèmes. Pour traiter les questions suivantes, une nouvelle entrée a été utilisée pour le rôle *target*.

Le mécanisme d'authentification le plus communément utilisé dans le déploiement des connexions iSCSI s'appuie sur CHAP (*Challenge-Handshake Authentication Protocol*). Il s'agit d'une méthode d'authentification entre deux hôtes pairs sans échange de mot de passe en clair sur le réseau. Cette méthode suppose que les deux hôtes utilisent le même mot de passe.

Q31. Comment régler les paramètres d'authentification CHAP sur le système *target* ?

Comme pour les étapes précédentes, toutes les manipulations se font à partir de l'outil targetcli. Il faut donc consulter la documentation de cet outil à l'adresse *Linux-IO : the Linux SCSI Target wiki*. Il existe une section *Mutual CHAP authentication*.

Partant d'une nouvelle configuration, on obtient la liste de paramètres suivante dans laquelle aucun contrôle d'accès n'a été défini.

```
/iscsi/iqn.20...57c35b07/tpg1> ls
o- tpg1 ..................................................... [enabled]
  o- acls ....................................................... [0 ACLs]
  o- luns ....................................................... [1 LUN]
  | o- lun0 ..................................... [iblock/initiator1 (/dev/vdb)]
  o- portals ................................................. [1 Portal]
    o- 2001:db8:feb2:2:b8ad:ff:feca:fe00:3260 ............... [OK, iser disabled]
```

On passe à la création d'une entrée de contrôle d'accès basée sur l'identifiant `iqn` unique du système *initiator*.

```
/iscsi/iqn.20...57c35b07/tpg1> acls/ create iqn.2015-09.org.debian:01:9d11913c78ac
Created Node ACL for iqn.2015-09.org.debian:01:9d11913c78ac
Created mapped LUN 0.
/iscsi/iqn.20...57c35b07/tpg1> ls
o- tpg1 ..................................................................... [enabled]
  o- acls ....................................................................... [1 ACL]
  | o- iqn.2015-09.org.debian:01:9d11913c78ac .................. [1 Mapped LUN]
  |   o- mapped_lun0 ........................................... [lun0 (rw)]
  o- luns ....................................................................... [1 LUN]
  | o- lun0 ..................................... [iblock/initiator1 (/dev/vdb)]
  o- portals ................................................... [1 Portal]
    o- 2001:db8:feb2:2:b8ad:ff:feca:fe00:3260 ............. [OK, iser disabled]
```

On définit ensuite les paramètres d'authentification pour cette entrée. Comme la méthode CHAP est symétrique, on doit déposer de part et d'autre le secret. On fixe ici les paramètres `userid` et `password`.

```
/iscsi/iqn.20...57c35b07/tpg1> acls/iqn.2015-09.org.debian:01:9d11913c78ac/ set auth userid=initi
Parameter userid is now 'initiator-username'.
/iscsi/iqn.20...57c35b07/tpg1> acls/iqn.2015-09.org.debian:01:9d11913c78ac/ set auth password=ini
Parameter password is now 'initiator-53cr3t-p455w0rd'.
```

Q32. Comment régler les paramètres d'authentification CHAP sur le système *initiator* ?

Rechercher dans le fichier de configuration principal du rôle *initiator* les paramètres relatifs à l'authentification.

Le nom d'utilisateur et le mot de passe sont définis dans le fichier `/etc/iscsi/iscsid.conf` du système *initiator*.

```
# *************
# CHAP Settings
# *************

# To enable CHAP authentication set node.session.auth.authmethod
# to CHAP. The default is None.
node.session.auth.authmethod = CHAP

# To set a CHAP username and password for initiator
# authentication by the target(s), uncomment the following lines:
node.session.auth.username = SAN-lab-1stInitiator
node.session.auth.password = MyS4N-1stInitiator-53cr3t
```

Le même principe peut être appliqué au mécanisme de découverte en appliquant un couple *login/password* identique ou non à la suite de ce fichier de configuration.

Une fois la configuration en place, on obtient les résultats suivants lors de la validation.

- Découverte du nouveau volume réseau :

```
# iscsiadm -m discovery --type sendtargets --portal=[2001:db8:feb2:2:b8ad:ff:feca:fe00]:3260
[2001:db8:feb2:2:b8ad:ff:feca:fe00]:3260,1 iqn.2003-01.org.linux-iscsi.target.i686:sn.f58f71d5ba26
192.0.2.12:3260,1 iqn.2003-01.org.linux-iscsi.target.i686:sn.f58f71d5ba26
[2001:db8:feb2:2:b8ad:ff:feca:fe00]:3260,1 iqn.2003-01.org.linux-iscsi.target.i686:sn.8b7457c35b07
```

- Connexion avec authentification CHAP :

```
# iscsiadm -m node -T iqn.2003-01.org.linux-iscsi.target.i686:sn.8b7457c35b07 -p 2001:db8:feb2:2:b8ad
Logging in to [iface: default, target: iqn.2003-01.org.linux-iscsi.target.i686:sn.8b7457c35b07, porta
Login to [iface: default, target: iqn.2003-01.org.linux-iscsi.target.i686:sn.8b7457c35b07, portal: 20
```

- Affichage de la session active :

```
# iscsiadm -m session
tcp: [4] [2001:db8:feb2:2:b8ad:ff:feca:fe00]:3260,1 iqn.2003-01.org.linux-iscsi.target.i686:sn.8b7457
```

1.7. Configuration d'une unité logique RAID1

Dans cette partie, on crée une unité logique RAID1 composée d'une unité de disque locale et d'une unité de disque iSCSI dans le but d'illustrer une solution de réplication synchrone. En effet, dans un volume RAID1

chaque disque contient à tout moment exactement les mêmes données. Ici, le contenu de l'unité de disque locale est identique à celui de l'unité de disque réseau. La réplication ainsi réalisée est dite synchrone puisque toute écriture locale est dupliquée sur le réseau de stockage iSCSI.

1.7.1. Sélection du paquet et création de l'unité de stockage

Q33. Quel est le paquet qui contient les outils de configuration et de gestion des différents types d'unités RAID logicielles ? Installer ce paquet et identifier l'outil d'administration de tableau RAID logiciel.

Effectuer une recherche dans les descriptions de paquets avec l'acronyme clé RAID.

```
# aptitude search ~draid | grep administration
p   mdadm                          - outil d'administration d'ensembles RAID
```

Une fois le paquet identifié et installé, on peut lister son contenu et isoler les commandes utilisateur.

```
# dpkg -L mdadm | grep bin
/sbin
/sbin/mdmon
/sbin/mdadm-startall
/sbin/mdadm
```

Q34. Rechercher la syntaxe d'appel à l'outil identifié dans la question précédente pour créer l'unité logique RAID1 ? Exécuter cette commande.

Après s'être assuré qu'aucune table de partition n'existe sur les deux unités constituant le tableau, on obtient le résultat suivant.

```
# mdadm --create /dev/md0 --level=raid1 --raid-devices=2 /dev/sda /dev/vdb
mdadm: Note: this array has metadata at the start and
    may not be suitable as a boot device.  If you plan to
    store '/boot' on this device please ensure that
    your boot-loader understands md/v1.x metadata, or use
    --metadata=0.90
Continue creating array? y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
```

1.7.2. Manipulations sur l'unité de stockage RAID1

Q35. Comment connaître l'état de l'unité logique RAID1 ?

Effectuer une recherche dans le système de fichiers virtuel `/proc/`.

Exemple du tableau créé lors l'exécution de la commande de la question précédente.

```
# cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 vdb[1] sda[0]
      33537920 blocks super 1.2 [2/2] [UU]

unused devices: <none>
```

Q36. Comment afficher la liste des propriétés de l'unité logique RAID1 ?

Effectuer une recherche dans les options de la commande d'administration.

```
# mdadm --detail /dev/md0
/dev/md0:
        Version : 1.2
  Creation Time : Sun Sep  9 17:06:34 2012
     Raid Level : raid1
     Array Size : 33537920 (31.98 GiB 34.34 GB)
  Used Dev Size : 33537920 (31.98 GiB 34.34 GB)
   Raid Devices : 2
  Total Devices : 2
    Persistence : Superblock is persistent

    Update Time : Sun Sep  9 17:13:18 2012
          State : clean
 Active Devices : 2
Working Devices : 2
 Failed Devices : 0
  Spare Devices : 0

           Name : iSCSI-1StInitiator:0  (local to host iSCSI-1StInitiator)
           UUID : 01749969:3a764b9f:2749b4c4:5953b282
         Events : 17

    Number   Major   Minor   RaidDevice State
       0       8        0        0      active sync   /dev/sda
       1     254       16        1      active sync   /dev/vdb
```

Q37. Comment rendre la configuration du tableau RAID1 permanente au niveau système ?

Effectuer une recherche dans les options de la commande d'administration.

C'est le fichier `/etc/mdadm/mdadm.conf` qui contient les directives de configuration. On ajoute en fin de ce fichier la définition du tableau créé plus haut.

```
# mdadm --detail --scan >> /etc/mdadm/mdadm.conf
```

1.8. Configuration d'un volume logique de sauvegarde

L'objectif de cette partie est de créer un mécanisme de sauvegarde réseau automatisé en s'appuyant sur la notion de «prise de vue» ou *snapshot* proposée par le gestionnaire de volume logique LVM. Dans une prise de vue, on ne stocke que les différences relativement au volume logique original.

```
# pvcreate /dev/md0
  Writing physical volume data to disk "/dev/md0"
  Physical volume "/dev/md0" successfully created
```

```
# pvdisplay
  --- Physical volume ---
  PV Name               /dev/vda5
  VG Name               vm0
  PV Size               31,76 GiB / not usable 2,00 MiB
  Allocatable           yes (but full)
  PE Size               4,00 MiB
  Total PE              8130
  Free PE               0
  Allocated PE          8130
  PV UUID               CpaZ5D-vbVS-32w3-QLnk-GVAd-06pB-y2Iw8Y

  "/dev/md0" is a new physical volume of "31,98 GiB"
  --- NEW Physical volume ---
  PV Name               /dev/md0
  VG Name
  PV Size               31,98 GiB
  Allocatable           NO
  PE Size               0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               KAmRl0-ugMa-0eE3-ZJCc-Q2t0-lqeM-RB8Qxn
```

```
# vgextend vm0 /dev/md0
  Volume group "vm0" successfully extended
```

```
# vgdisplay
  --- Volume group ---
  VG Name               vm0
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  4
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                2
  Open LV               2
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               63,74 GiB
  PE Size               4,00 MiB
  Total PE              16317
  Alloc PE / Size       8130 / 31,76 GiB
  Free  PE / Size       8187 / 31,98 GiB
  VG UUID               dnw5zr-hPPU-L1FZ-P6Be-HL7E-FUNu-0OuosE
```

```
# lvcreate --name backup -L12G vm0
```

```
# lvcreate --snapshot --name LVM-snapshot-lab --extents +100%FREE /dev/vm0/root
  Logical volume "LVM-snapshot-lab" created
```

```
# lvdisplay /dev/vm0/LVM-snapshot-lab
  --- Logical volume ---
  LV Path                /dev/vm0/LVM-snapshot-lab
  LV Name                LVM-snapshot-lab
  VG Name                vm0
  LV UUID                md1QF6-NI2p-tmxB-9IeO-mlBi-Xbi6-IUB3xE
  LV Write Access        read/write
  LV Creation host, time iSCSI-1StInitiator, 2012-09-09 21:49:31 +0200
  LV snapshot status     active destination for root
  LV Status              available
  # open                 0
  LV Size                30,41 GiB
  Current LE             7784
  COW-table size         19,98 GiB
  COW-table LE           5115
  Allocated to snapshot  0,00%
  Snapshot chunk size    4,00 KiB
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           252:3
```

```
# mkdir /mnt/LVM-snapshot-lab
# mount /dev/vm0/LVM-snapshot-lab /mnt/LVM-snapshot-lab/
```

```
# ll /mnt/LVM-snapshot-lab/
total 112K
drwxr-xr-x  2 root root 4,0K sept.  5 11:36 bin
drwxr-xr-x  2 root root 4,0K oct.  25  2010 boot
drwxr-xr-x  5 root root 4,0K oct.  25  2010 dev
drwxr-xr-x 79 root root 4,0K sept.  9 18:17 etc
drwxr-xr-x  3 root root 4,0K oct.  25  2010 home
lrwxrwxrwx  1 root root   30 sept.  5 11:36 initrd.img -> /boot/initrd.img-3.2.0-3-amd64
drwxr-xr-x 14 root root  12K sept.  5 11:36 lib
drwxr-xr-x  2 root root 4,0K sept.  5 11:33 lib64
drwx------  2 root root  16K oct.  25  2010 lost+found
drwxr-xr-x  3 root root 4,0K oct.  25  2010 media
drwxr-xr-x  2 root root 4,0K août   6  2010 mnt
drwxr-xr-x  2 root root 4,0K oct.  25  2010 opt
drwxr-xr-x  2 root root 4,0K août   6  2010 proc
drwx------  4 root root 4,0K sept.  7 17:18 root
drwxr-xr-x  2 root root 4,0K déc.  23  2011 run
drwxr-xr-x  2 root root  12K sept.  9 17:05 sbin
drwxr-xr-x  2 root root 4,0K juil. 21  2010 selinux
drwxr-xr-x  2 root root 4,0K oct.  25  2010 srv
drwxr-xr-x  2 root root 4,0K août  15  2010 sys
drwxrwxrwt  2 root root 4,0K sept.  9 18:17 tmp
drwxr-xr-x 10 root root 4,0K janv. 29  2012 usr
drwxr-xr-x 11 root root 4,0K janv. 29  2012 var
```

```
# mkfs.ext4 /dev/vm0/backup
mke2fs 1.42.5 (29-Jul-2012)
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
786432 i-noeuds, 3145728 blocs
157286 blocs (5.00%) réservés pour le super utilisateur
Premier bloc de données=0
Nombre maximum de blocs du système de fichiers=3221225472
96 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, 294912, 819200, 884736, 1605632, 2654208

Allocation des tables de groupe : complété
Écriture des tables d'i-noeuds : complété
Création du journal (32768 blocs) : complété
Écriture des superblocs et de l'information de comptabilité du système de
fichiers : complété

# mkdir /backup
# mount /dev/vm0/backup /backup/
```

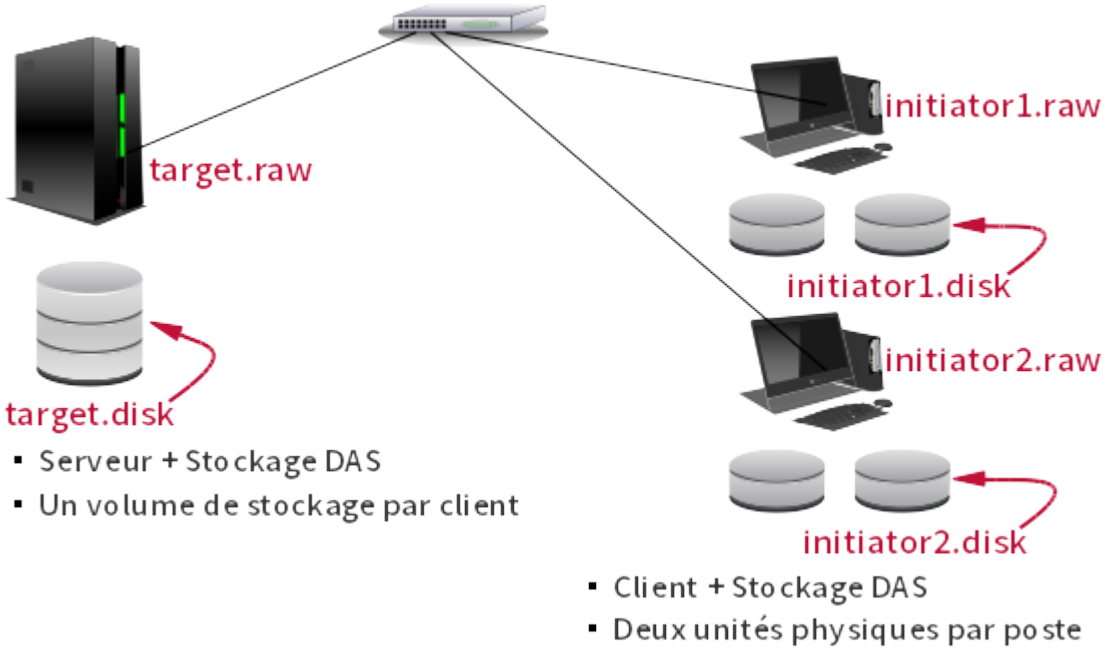
```
# tar --exclude-from backup-exclude.list -cvjf /backup/actually.tar.bz2 /

# /backup
/lib/init/rw
/proc
/sys
/dev
/run
/mnt
/selinux
/media
/var/lib/nfs
/etc/lvm
```

1.9. Manipulations sur machines virtuelles

Il est possible de réaliser l'ensemble des manipulations de ce support à l'aide de deux ou trois instances de machines virtuelles appartenant un même réseau de diffusion (LAN).

L'infrastructure à implanter sur le système hôte est la suivante.



Topologie virtualisation iSCSI - vue complète

Le script `install.sh` donné en copie ci-dessous effectue deux tâches distinctes.

Il commence par la copie des fichiers image des trois systèmes virtuels à partir du fichier maître. Les fichiers au format `.qed` sont des images compressées faciles à transférer.

Ensuite, il créé trois fichiers vides qui serviront de volume de stockage à chaque machine virtuelle.

```
#!/bin/bash

# Création des 3 machines virtuelles
ionice -c 3 cp ../vm0-debian-testing-i386-base.qed target.qed
ionice -c 3 cp ../vm0-debian-testing-i386-base.qed initiator1.qed
ionice -c 3 cp ../vm0-debian-testing-i386-base.qed initiator2.qed

# Création des 3 volumes de stockage
dd if=/dev/null of=target.disk bs=1 seek=72G
dd if=/dev/null of=initiator1.disk bs=1 seek=32G
dd if=/dev/null of=initiator2.disk bs=1 seek=32G
```

Le script `startup.sh` donné en copie ci-dessous sert au lancement des trois systèmes virtuels avec chacun une unité de stockage supplémentaire.

```
#!/bin/bash

../scripts/ovs-startup.sh target.qed 4096 100 \
-drive if=none,id=storagevol0,aio=native,cache=directsync,\
format=raw,media=disk,file=target.disk \
-device virtio-blk,drive=storagevol0,scsi=off,config-wce=off

../scripts/ovs-startup.sh initiator1.qed 1024 101 \
-drive if=none,id=initiator1addon,aio=native,cache=directsync,\
format=raw,media=disk,file=initiator1.disk \
-device virtio-blk,drive=initiator1addon,scsi=off,config-wce=off

../scripts/ovs-startup.sh initiator2.qed 1024 102 \
-drive if=none,id=initiator2addon,aio=native,cache=directsync,\
format=raw,media=disk,file=initiator2.disk \
-device virtio-blk,drive=initiator2addon,scsi=off,config-wce=off
```

Ce script fait lui-même appel au script commun `ovs-startup.sh` qui sert à initialiser une instance de machine virtuelle en utilisant comme paramètres le nom du fichier image, la quantité de RAM et le cordon de brassage virtuel tap. Le guide *Virtualisation système et enseignement* fournit le code source du script de lancement d'une machine virtuelle raccordée à un commutateur Open vSwitch.

1.10. Évaluation des performances

Voici quelques exemples de mesures de performances d'accès aux volumes de stockage. L'objectif est de présenter quelques outils qui produisent des résultats dans un laps de temps relativement court.

Note

La pertinence ou la validité des résultats dépendent énormément du facteur temps. Une mesure valide suppose un temps d'exécution de quelques heures au moins pour chaque outil. Les résultats donnés ici ne peuvent donc pas être considérés comme représentatif des performances de chaque technologie de stockage.

Il convient de décrire de façon très détaillée les conditions dans lesquelles ces tests sont réalisés. En effet, d'une plateforme matérielle à l'autre la distorsion des performances est considérable.

Tous les résultats ci-dessous sont obtenus avec l'outil bonnie++ et une taille de fichier de 8Go.

Unité de disque locale

Système de fichiers `ext3` avec gestion de volume logique LVM.

```
# time bonnie++ -u 1000 -s 8000 -d /var/tmp >result.txt
<snipped>
# cat result.txt
Version  1.96       ------Sequential Output------ --Sequential Input- --Random-
Concurrency   1     -Per Chr- --Block-- -Rewrite- -Per Chr- --Block-- --Seeks--
Machine        Size K/sec %CP K/sec %CP K/sec %CP K/sec %CP K/sec %CP  /sec %CP
iSCSI-1StInit 8000M   511  99 234868  55 180260  30  2985  99 615617  49 15925 260
Latency             26238us     535ms     545ms    4181us    8362us   63959us
Version  1.96       ------Sequential Create------ --------Random Create--------
iSCSI-1StInitiator  -Create-- --Read--- -Delete-- -Create-- --Read--- -Delete--
              files  /sec %CP  /sec %CP  /sec %CP  /sec %CP  /sec %CP  /sec %CP
                 16 +++++ +++ +++++ +++ +++++ +++ +++++ +++ +++++ +++ +++++ +++
Latency               411us     779us    1413us     265us      28us     699us
```

Unité de disque iSCSI

Système de fichiers `ext4`.

```
# time bonnie++ -u 1000 -s 8000 -d /mnt/tmp >result.txt
<snipped>
# cat result.txt
Version  1.96       ------Sequential Output------ --Sequential Input- --Random-
Concurrency   1     -Per Chr- --Block-- -Rewrite- -Per Chr- --Block-- --Seeks--
Machine        Size K/sec %CP K/sec %CP K/sec %CP K/sec %CP K/sec %CP  /sec %CP
iSCSI-1StInit 8000M   534  99 96128  15 44584  11  2761  98 109216  16  3112  96
Latency             17770us     961ms     333ms    6060us    7910us   76502us
Version  1.96       ------Sequential Create------ --------Random Create--------
iSCSI-1StInitiator  -Create-- --Read--- -Delete-- -Create-- --Read--- -Delete--
              files  /sec %CP  /sec %CP  /sec %CP  /sec %CP  /sec %CP  /sec %CP
                 16 27168  50 +++++ +++ +++++ +++ +++++ +++ +++++ +++ +++++ +++
Latency              1228us     762us     820us     262us      34us     749us
```

Tableau RAID1 constitué d'une unité de disque locale et d'une unité de disque iSCSI

Système de fichiers `ext4`.

```
# time bonnie++ -u 1000 -s 8000 -d /mnt/tmp >result.txt
<snipped>
# cat result.txt
Version  1.96       ------Sequential Output------ --Sequential Input- --Random-
Concurrency   1     -Per Chr- --Block-- -Rewrite- -Per Chr- --Block-- --Seeks--
Machine        Size K/sec %CP K/sec %CP K/sec %CP K/sec %CP K/sec %CP  /sec %CP
iSCSI-1StInit 8000M   525  99 93851  15 60117  12  2795  95 177757  19  3707  99
Latency             25078us     729ms     194ms   45986us     343ms    1055ms
Version  1.96       ------Sequential Create------ --------Random Create--------
iSCSI-1StInitiator  -Create-- --Read--- -Delete-- -Create-- --Read--- -Delete--
              files  /sec %CP  /sec %CP  /sec %CP  /sec %CP  /sec %CP  /sec %CP
                 16 26606  51 +++++ +++ +++++ +++ +++++ +++ +++++ +++ 30648  41
Latency               195us     791us     823us     351us      47us     745us
```

1.11. Documents de référence

Architecture réseau des travaux pratiques

Architecture réseau des travaux pratiques : présentation de l'implantation des équipements d'interconnexion réseau dans l'armoire de brassage et du plan d'adressage IP prédéfini pour l'ensemble des séances de travaux pratiques. Ce document est utilisé dans la Section 1.1, « Adressage IP des postes de travail ».

Configuration d'une interface réseau

Configuration d'une interface de réseau local : tout sur la configuration des interfaces réseau de réseau local. Comme dans le cas précédent, ce document est utile pour effectuer les opérations demandées dans la Section 1.1, « Adressage IP des postes de travail ».

Introduction to iSCSI

L'article intitulé *Introduction to iSCSI* du site Linux Magazine présente les points clés de la technologie iSCSI. Il complète la Section 1.2, « Technologie iSCSI et topologie de travaux pratiques ».

iSCSI - Debian Wiki

La page *iSCSI and Debian* contient deux sous-rubriques sur les rôles *initiator* et *target*. Pour le rôle *target*, la section relative à l'utilisation du sous système *Linux-IO : the Linux SCSI Target wiki* n'a pas encore été documentée.

CHAPITRE 2

Introduction au système de fichiers réseau NFSv4

Résumé

L'objectif de ce support de travaux pratiques est l'étude du système de fichiers réseau NFS. Il illustre les accès en «mode fichier» à une unité de stockage réseau. Ce mode d'accès correspond à un stockage de type NAS ou *Network Attached Storage*. Le document débute avec l'étude du principe de fonctionnement des appels de fonctions RPC (*Remotre Procedure Call*) puis il poursuit avec la configuration d'un serveur NFS qui exporte une arborescence de comptes utilisateurs. Côté client, on étudie les accès au système de fichiers réseau NFS suivant deux modes distincts : le montage manuel puis l'automontage.

2.1. Adressage IP des postes de travail

Tableau 2.1. Affectation des adresses et des réseaux IP de la salle 211

Poste 1	Poste 2	Passerelle par défaut
christophsis	corellia	`10.30.6.65/30`
delaya	kasyyyk	`192.168.6.17/29`
korriban	kessel	`192.168.7.33/28`
mygeeto	nelvaan	`172.20.11.9/29`
rattatak	saleucami	`172.20.12.17/29`
taris	teth	`192.168.37.9/29`
utapau	yavin	`10.8.11.9/29`

Tableau 2.2. Affectation des adresses et des réseaux IP de la salle 213

Poste 1	Poste 2	Passerelle par défaut
alderaan	bespin	`172.19.116.1/26`
centares	coruscant	`10.0.119.65/27`
dagobah	endor	`10.0.121.129/27`
felucia	geonosis	`172.19.114.129/26`
hoth	mustafar	`192.168.108.129/25`
naboo	tatooine	`10.5.6.1/23`

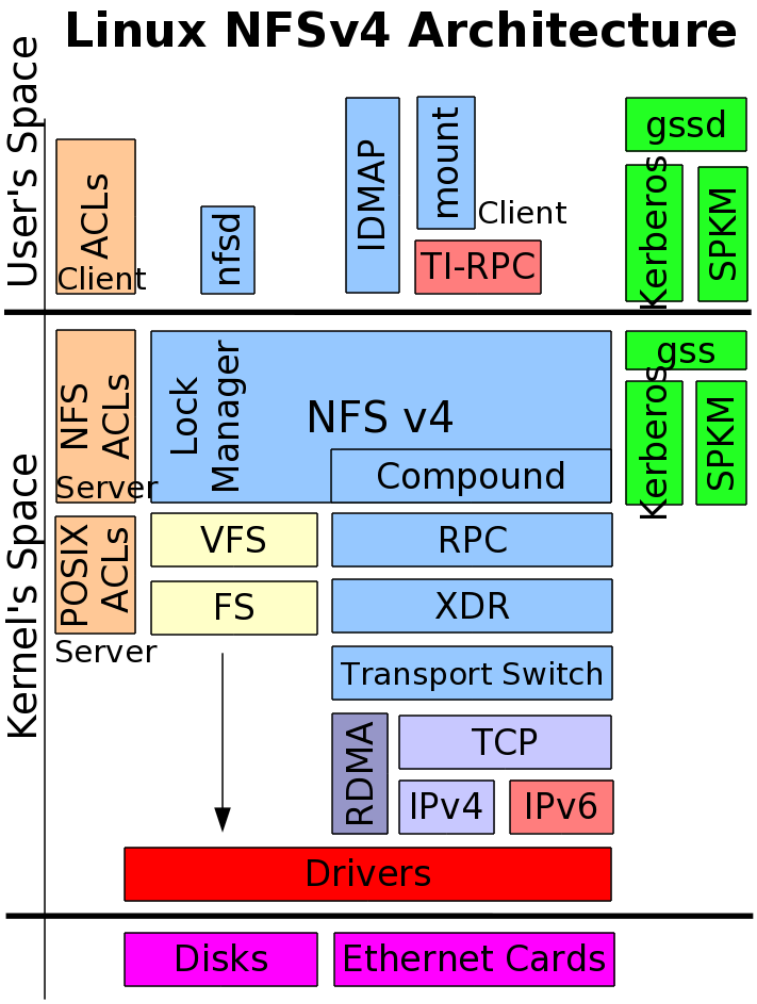
Pour ces travaux pratiques, de nombreuses questions peuvent être traitées à l'aide du document de référence : *Nfsv4 configuration*. Il faut cependant faire correspondre les configurations décrites dans ce document avec les configurations proposées avec les paquets de la distribution *Debian GNU/Linux*.

Pour chaque paire de postes de travaux pratiques, il faut attribuer les rôles serveur et client. Le serveur doit exporter une partie de son arborescence locale de système de fichiers et le client doit pouvoir y accéder de façon transparente via un montage du système de fichiers distant. Ce support de travaux pratiques fait suite à la présentation : *Systèmes de fichiers réseau*.

2.2. Protocole NFS et topologie de travaux pratiques

Cette section reprend les éléments spécifiques au protocole NFS introduits lors de la présentation *Systèmes de fichiers réseau*.

Plusieurs versions du protocole de système de fichiers réseau NFS sont disponibles. Chacune correspond à une «époque» ou à un mode d'exploitation. La vue ci-dessous illustre la distribution des fonctionnalités de la version 4 entre les espaces noyau et utilisateur.



La version 2 du protocole NFS a été la première à être largement adoptée à la fin des années 80. Elle a été conçue pour fournir un service de partage de fichiers entre les hôtes d'un même réseau local. Elle s'appuie sur le protocole UDP au niveau transport et sur le mécanisme d'appel de procédure distant (RPC) aux niveaux supérieurs.

La version 3 du protocole, introduite au milieu des années 90, a apporté de nombreuses améliorations en termes de fiabilité et de performances relativement à la précédente. Avec la version 3 du protocole :

- La taille maximum de fichier n'est plus limitée à 2Go.
- Les écritures asynchrones sur le serveur sont possibles ; ce qui améliore beaucoup les performances. Les requêtes en écriture des clients sont gérées en mémoire cache. Le client n'a plus à attendre que les demandes d'écritures soient effectivement appliquées sur les disques ce qui améliore les temps de réponse.
- Les contrôles d'accès sont effectués avant les manipulations sur les fichiers.
- La taille des données transférées n'est plus limitée à 8Ko.
- Il est possible d'utiliser le protocole TCP au niveau transport.

La version 4 du protocole apporte de nouvelles fonctionnalités relativement aux précédentes.

Les identifiants d'utilisateur et de groupe (`uid`/`gid`) sont représentés par des chaînes de caractères. Un service, baptisé `idmapd`, est utilisé sur le serveur pour faire les correspondances entre les valeurs numériques locales et les chaînes de caractères. Ces correspondances permettent d'utiliser de nouveaux contrôles d'accès indépendants entre clients et serveurs.

Les serveurs maintiennent un pseudo système de fichiers qui assure la cohérence du système de nommage avec les clients. Ainsi, un objet est nommé de façon identique entre le serveur et ses clients. Pour respecter les spécifications POSIX, un client qui a accès à un niveau d'arborescence peut parcourir tous les niveaux inférieurs. Il n'est pas nécessaire d'exporter les sous arborescences.

Les appels de procédures distants n'utilisent plus le multiplexage de ports. Un numéro de port unique a été attribué à la version 4 du protocole NFS : tcp/2049. La version 3 doit utiliser plusieurs ports pour les traitements de ses protocoles complémentaires ; ce qui donne un assemblage plutôt complexe de ports et de couches avec des problèmes de sécurité propres. Aujourd'hui, ce mode de fonctionnement est abandonné et toutes les opérations de mise en œuvre de protocole complémentaire précédemment exécutées via des ports individuels sont maintenant traitées directement à partir d'un port unique connu.

Désormais, le mécanisme d'appel RPC n'est plus aussi important et sert essentiellement d'enveloppe pour les opérations encapsulées dans la pile NFSv4. Ce changement rend le protocole beaucoup moins dépendant de la sémantique du système de fichiers sous-jacent. Pour autant, les opérations de système de fichiers d'autres systèmes d'exploitation n'ont pas été négligées. Par exemple, les systèmes Microsoft™ exigent des appels *stateful* ouverts. Le mécanisme de suivi d'état de communication (*statefulness*) facilite l'analyse de trafic et rend les opérations de système de fichiers beaucoup plus simples à interpréter. Ce même mécanisme permet aux clients de gérer les données «en l'état» en mémoire cache.

La version 4 simplifie les requêtes en utilisant des opérations composées ou groupées (*compound*) qui englobent un grand nombre de traitements sur les objets du système de fichiers. L'effet immédiat est, bien sûr, une diminution très importante des appels RPC et des données qui doivent parcourir le réseau. Bien que chaque appel RPC transporte beaucoup plus de données en accomplit beaucoup plus de traitements, on considère qu'une requête composée de la version 4 du protocole exige cinq fois moins d'interactions client serveur qu'avec la version 3.

L'objectif des manipulations qui sont demandées dans ce document est d'illustrer les nouvelles fonctionnalités apportées par la dernière version du protocole NFS. Le séquencement des opérations à réaliser lors de la séance de travaux pratiques est décrit dans le tableau ci-dessous. Après le traitement de la première partie commune, les deux postes occupent chacun un rôle distinct.

Tableau 2.3. Attribution des rôles

Client	**Serveur**
Identification du mécanisme des appels RPC. Installation et configuration des paquets communs.	
Identification des services disponibles sur le serveur. Création d'un compte local sans répertoire utilisateur.	Installation du paquet spécifique au serveur et configuration du service en fonction de l'arborescence à exporter.
validation de l'accès au système de fichiers réseau avec capture de trafic.	
Installation du paquet spécifique et configuration du service d'automontage des répertoires utilisateurs.	

2.3. Configuration commune au client et au serveur NFS

Plusieurs services communs doivent être actifs pour que les accès au système de fichiers réseau NFS soient utilisables. Le mécanisme de gestion des appels de procédures distants appelé RPC ou *Remote Procedure Call* constitue le point de départ dans la mise œuvre de ces services communs.

Le logiciel de gestion des appels de procédures distants a évolué avec les différentes versions du système de fichiers NFS et l'arrivée du protocole réseau IPv6. La configuration étudiée ici doit permettre de fonctionner de la façon la plus transparente possible avec les versions 3 et 4 du système de fichiers NFS.

Note

Les manipulations présentées ici ne traitent pas le volet authentification et chiffrement des échanges sur le réseau. On considère que les services *Kerberos*, SPKM-3 et LIPKEY ne sont pas actifs sur les systèmes étudiés.

2.3.1. Gestion des appels RPC

Q38. Quels sont les deux logiciels disponibles chargés de la gestion des appels RPC ? Qu'est-ce qui les distinguent ?

La présentation *Systèmes de fichiers réseau* introduit les principes de fonctionnement des appels de procédures distants.

Dans un premier temps, rechercher dans le support *Linux NFS-HOWTO* le service «historique» utilisé par NFS pour le multiplexage des appels de procédures distants. Dans un second temps, consulter la page *TI-RPC / rpcbind support* pour identifier les évolutions apportées.

Le support *Linux NFS-HOWTO* présente le service «historique» utilisé par NFS pour le multiplexage des appels de procédures distants : `portmap`. Ce service est fourni par le paquet du même nom et est limité au protocole réseau IPv4.

La page *TI-RPC / rpcbind support* présente un nouveau logiciel de multiplexage des mêmes appels de procédures distants : `rpcbind`. Ce nouveau démon est aussi fourni par le paquet du même nom. Il se veut plus évolutif que le précédent et supporte le protocole réseau IPv6.

Q39. Quels sont les paquets qui correspondent à ces logiciels ? Installer le paquet ouvrant les services de transport universels.

Utiliser les outils de recherche dans les répertoires de noms de paquets et dans leurs descriptions : **apt-cache**, **dpkg**, **aptitude**.

Comme indiqué dans la documentation, on recherche un paquet portant le nom `rpcbind`.

```
# aptitude search rpcbind
p   rpcbind   - conversion de numéros de programmes RPC en adresses universelles
```

Une fois l'existence du paquet confirmée, on l'installe. Il est possible que ce nouveau paquet entraîne la suppression de l'ancien paquet `portmap` qui est en conflit avec cette nouvelle version du même service.

```
# aptitude install rpcbind
Les NOUVEAUX paquets suivants vont être installés :
  libgssglue1{a} libtirpc1{a} rpcbind
  0 paquets mis à jour, 3 nouvellement installés, 0 à enlever et 0 non mis à jour.
  Il est nécessaire de télécharger 161 ko d'archives. Après dépaquetage, 458 ko seront utilisés.
  Voulez-vous continuer ? [Y/n/?]
```

Q40. Quel est le numéro de port utilisé par le service ? Quel est le principe de fonctionnement du service pour le traitement des appels de procédures distants ?

Utiliser les commandes qui permettent d'obtenir les informations sur :

- La liste des processus actifs sur le système,
- Les numéros de ports en écoute sur les interfaces réseau,
- Les pages de manuels des applications utilisées.
- La liste des processus actifs sur le système,

```
# ps aux | grep rpc[b]ind
root      2963  0.0  0.0  18956   724 ?        Ss   14:01   0:00 /sbin/rpcbind -w
```

- Les numéros de ports en écoute sur les interfaces réseau,

```
# lsof -i | grep rpc[b]ind
rpcbind  2963        root    6u  IPv4   6670      0t0  UDP *:sunrpc
rpcbind  2963        root    7u  IPv4   6673      0t0  UDP *:1018
rpcbind  2963        root    8u  IPv4   6674      0t0  TCP *:sunrpc (LISTEN)
rpcbind  2963        root    9u  IPv6   6677      0t0  UDP *:sunrpc
rpcbind  2963        root   10u  IPv6   6680      0t0  UDP *:1018
rpcbind  2963        root   11u  IPv6   6681      0t0  TCP *:sunrpc (LISTEN)
```

On obtient la correspondance entre numéro de port et nom de service en consultant le fichier `/etc/services`.

```
# grep sunrpc /etc/services
sunrpc          111/tcp         portmapper      # RPC 4.0 portmapper
sunrpc          111/udp         portmapper
```

Le principe de fonctionnement des appels de procédures distants veux que tous ces appels soient reçus sur un numéro de port unique ; `sunrpc/111` dans le cas présent. Ces appels, une fois identifiés, sont transmis aux programmes concernés pour être traités.

- Les pages de manuels des applications utilisées.

```
# man rpcbind
```

Q41. Quelle est a commande qui permet de lister les services accessibles via un appel RPC ? À quel paquet appartient cette commande ?

Rechercher dans le support *Linux NFS-HOWTO* et dans la liste des fichiers du paquet sélectionné pour la gestion des appels RPC.

La commande présentée dans le support *Linux NFS-HOWTO* est appelée **rpcinfo**. On vérifie sa présence sur le système étudié de la façon suivante.

```
# dpkg -S `which rpcinfo`
rpcbind: /usr/sbin/rpcinfo
```

Dans la version la plus récente du programme, c'est l'option `-s` qui permet d'obtenir la présentation la plus synthétique des services accessibles par appel RPC.

```
# rpcinfo -s
program version(s) netid(s)                         service     owner
 100000  2,3,4     local,udp,tcp,udp6,tcp6          portmapper  superuser
```

La copie d'écran ci-dessus montre que le gestionnaire d'appel `portmapper` est le seul service ouvert. On relève l'ordre de priorité des différentes versions du service supportées par le système ainsi que les versions des protocoles de couche transport.

Q42. Donner deux exemples d'exécution : un en local et un sur le poste de travaux pratiques voisin.

Reprendre la commande utilisée dans la question précédente en indiquant l'adresse IP du poste voisin.

L'exemple d'exécution de la commande en local est donné dans la copie d'écran de la question précédente. Pour connaître les services accessibles sur un autre poste, on utilise la même commande suivie de l'adresse IP de cet hôte.

```
# rpcinfo -s 198.51.100.2
program version(s) netid(s)                         service     owner
 100000  2,3,4     local,udp,tcp,udp6,tcp6          portmapper  superuser
```

Cette copie d'écran montre la même liste de paramètres que lors de l'exécution de la commande en local. Les configurations sur les deux hôtes sont donc identiques.

Q43. Réaliser une capture à l'aide de l'analyseur réseau lors de l'exécution de la commande et relever : le protocole de transport utilisé, les numéros de ports caractéristiques de cette transaction ainsi que le nom de la procédure RPC utilisée.

```
 Poste 1                                        Poste 2
-------------------------------------------------------
 <commande>       --- requête --->            <processus>

                  <-- réponse ----
```

Voici un exemple de capture en mode console qui donne les éléments demandés.

```
$ tshark -i eth0 -f "ip and ! port 22"
Capturing on 'eth0'
198.51.100.2 → 198.51.100.3  TCP 74 758→111 [SYN] Seq=0
198.51.100.3 → 198.51.100.2  TCP 74 111→758 [SYN, ACK] Seq=0 Ack=1
198.51.100.2 → 198.51.100.3  TCP 66 758→111 [ACK] Seq=1 Ack=1
198.51.100.2 → 198.51.100.3  Portmap 110 V3 DUMP Call
198.51.100.3 → 198.51.100.2  TCP 66 111→758 [ACK] Seq=1 Ack=45
198.51.100.3 → 198.51.100.2  Portmap 3038 V3 DUMP Reply (Call In 4)
198.51.100.2 → 198.51.100.3  TCP 66 758→111 [ACK] Seq=45 Ack=2973
198.51.100.2 → 198.51.100.3  TCP 66 758→111 [FIN, ACK] Seq=45 Ack=2973
198.51.100.3 → 198.51.100.2  TCP 66 111→758 [FIN, ACK] Seq=2973 Ack=46
198.51.100.2 → 198.51.100.3  TCP 66 758→111 [ACK] Seq=46 Ack=2974
```

- Le protocole de couche transport utilisé est TCP.
- Le numéro de port utilisé correspond bien au service enregistré `sunrpc/111`.
- Le sous-programme distant appelé est : `Portmap V3 DUMP Call`.

2.3.2. Gestion des paquets NFS

Q44. Quel est le paquet commun au client et au serveur ? Identifier le jeu de commandes fournies par ce paquet.

Rechercher dans la liste des paquets disponibles, ceux dont le nom débute par `nfs`.

```
# aptitude search ?name"(^nfs)"
v   nfs-client         -
p   nfs-common         - NFS support files common to client and server
p   nfs-kernel-server  - support for NFS kernel server
v   nfs-server         -
p   nfs4-acl-tools     - Commandline and GUI ACL utilities for the NFSv4 client
p   nfswatch           - Program to monitor NFS traffic for the console
```

Dans la liste ci-dessus, on identifie le paquet `nfs-common` qui correspond bien aux fonctions communes au client et au serveur NFS.

Une fois le paquet installé, la liste des programmes fournis par ce paquet est extraite de la liste de ses fichiers à l'aide de la commande suivante.

```
# dpkg -L nfs-common | grep bin
/sbin
/sbin/mount.nfs
/sbin/osd_login
/sbin/rpc.statd
/sbin/showmount
/sbin/sm-notify
/usr/sbin
/usr/sbin/blkmapd
/usr/sbin/gss_clnt_send_err
/usr/sbin/gss_destroy_creds
/usr/sbin/mountstats
/usr/sbin/nfsidmap
/usr/sbin/nfsiostat
/usr/sbin/nfsstat
/usr/sbin/rpc.gssd
/usr/sbin/rpc.idmapd
/usr/sbin/rpc.svcgssd
/usr/sbin/rpcdebug
/usr/sbin/start-statd
/sbin/mount.nfs4
/sbin/umount.nfs
/sbin/umount.nfs4
```

Dans cette liste on trouve les commandes de montage, de démontage et de suivi d'état du système de fichiers réseau.

Q45. Est-ce que la liste des services accessibles via le mécanisme d'appel de procédure distant (RPC) a évolué ?

Réaliser une capture réseau lors de l'exécution des commandes et relever les protocoles et les numéros de ports caractéristiques de ces transactions.

```
 Poste 1                                        Poste 2
-------------------------------------------------------
 <commande>       --- requête --->            <processus>

                  <-- réponse ----
```

La capture réseau en mode console telle qu'elle est pratiquée dans la question ci-dessus ne montre aucune différence quant à l'utilisation du protocole de couche transport et des numéros de ports utilisés. La différence se situe dans le contenu au niveau de la couche application. La réponse à l'appel de sous-programme distant `Portmap V3 DUMP Call` contient des éléments supplémentaires relatifs aux services ouverts `idmapd` et `statd`.

Pour visualiser la liste des services accessibles via RPC avec l'analyseur réseau, il est préférable de passer en mode graphique. On peut réaliser la capture en mode console en stockant les résultats dans un fichier de capture et procéder à l'analyse en mode graphique à partir de ce fichier.

```
$ tshark -i eth0 -w /var/tmp/rpcinfo.pcap not port 22
10 ^C
$ chmod 644 /var/tmp/rpcinfo.pcap
```

2.4. Configuration du client NFS

Le rôle du client est d'intégrer un accès au système de fichiers d'un hôte distant dans son arborescence locale. On parle de «montage NFS». Dans un premier temps, on teste les opérations de montage manuel. Bien sûr, ces tests ne peuvent aboutir que si une arborescence à été exportée par un serveur.

Ensuite, on teste les opérations de montage automatisées ou «automontage». Si le serveur NFS n'est pas encore disponible au moment des tests de montage manuel, il faut préparer les fichiers de configuration du service d'automontage.

2.4.1. Opérations manuelles de (montage|démontage) NFS

Q46. Quelle est la commande qui permet de tester la disponibilité du service de montage NFS sur un hôte distant ?

Reprendre l'utilisation de la commande identifiée dans la section précédente.

Relativement aux résultats de la section précédente, la liste des services accessibles via RPC s'est étoffée et le service NFS apparaît cliarement.

```
#rpcinfo -s 198.51.100.2
program version(s) netid(s)                         service     owner
 100000  2,3,4     local,udp,tcp,udp6,tcp6          portmapper  superuser
 100003  4,3,2     udp6,tcp6,udp,tcp                nfs         superuser
 100227  3,2       udp6,tcp6,udp,tcp                -           superuser
 100021  4,3,1     tcp6,udp6,tcp,udp                nlockmgr    superuser
 100005  3,2,1     tcp6,udp6,tcp,udp                mountd      superuser
```

Q47. Quelle est la commande qui permet d'identifier l'arborescence disponible à l'exportation sur le serveur NFS ?

Rechercher dans la liste des fichiers du paquet de service commun NFS.

Dans la liste des commandes fournies avec le paquet `nfs-common`, on trouve un programme appelé **showmount**. Après consultation des pages de manuels, on relève l'option `-e` qui permet de consulter l'arborescence exportée par un serveur depuis un client. Voici un exemple d'exécution.

```
# showmount -e 198.51.100.2
Export list for 198.51.100.2:
/home/exports/home 198.51.100.0/24
/home/exports      198.51.100.0/24
```

Q48. Quelle est la commande à utiliser pour les opérations de montage manuel ? À quel paquet appartient cette commande ? Cette commande est-elle exclusivement liée au protocole NFS ?

Après avoir consulté le support *Linux NFS-HOWTO*, interroger la base de données des paquets, rechercher dans le contenus des paquets et consulter les pages de manuels.

La documentation indique que c'est la commande **mount** qui nous intéresse. On effectue ensuite les recherches avec le gestionnaire de paquets.

```
# dpkg -S `which mount`
mount: /bin/mount
```

La commande appartient au paquet du même nom. La consultation des pages de manuels **`# man mount`** montre que cette commande n'est pas réservée au seul protocole NFS mais à l'ensemble des opérations de montage pour tous les systèmes de fichiers utilisables.

Q49. Créer le répertoire `/ahome` destiné à «recevoir» le contenu répertoires utilisateurs exportés depuis le serveur NFS. Quelle est la syntaxe de la commande permettant de *monter* le répertoire exporté par le serveur NFS sur ce nouveau répertoire ?

Rechercher dans le support *Linux NFS-HOWTO*.

Dans le contexte de ces manipulations, il est important de préciser la version du protocole NFS lors du montage manuel.

```
# mkdir /ahome
# mount -t nfs4 198.51.100.2:/home /ahome/
# mount | grep nfs
rpc_pipefs on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
198.51.100.2:/home on /ahome type nfs4 (rw,relatime,vers=4,rsize=262144,wsize=262144, \
     namlen=255,hard,proto=tcp,timeo=600,retrans=2, \
     sec=sys,clientaddr=198.51.100.3,minorversion=0, \
     local_lock=none,addr=198.51.100.2)
```

Q50. Quelles sont les options de montage disponibles avec le protocole NFS ? Relever la signification des options principales ?

Consulter la documentation *Linux NFS-HOWTO*.

Les options caractéristiques sont : choix du protocole de transport, taille des blocs de données et version NFS. On peut aussi consulter les pages de manuels de la catégorie 5 concernant les formats de fichiers à l'aide de la commande **`man 5 nfs`**.

Q51. Réaliser une capture lors de l'exécution des commandes et relever les numéros de ports caractéristiques de ces transactions. Est-il possible de retrouver les informations échangées dans les données de capture ?

```
 Client                                                    Serveur
-------------------------------------------------------------------
 mount                  --- requête RPC --->              portmapper

 mount                  <--- numéro port ---              portmapper

 mount                  --- requête RPC --->              mountd

 mount                  <-- réponse --------              mountd

 lecture/écriture       ---- I/O ------------>            nfsd

 lecture/écriture       <- ACK fin opération -            nfsd
```

La marche à suivre est identique à celle de la même question côté serveur NFS.

Q52. Quelles seraient les opérations à effectuer pour configurer le système et rendre un montage NFS statique permanent ?

Rechercher le fichier de configuration système responsable des montages statiques des partitions.

Il est inutile de modifier les fichiers de configuration du système sachant que l'on change de méthode de montage dans la section suivante.

Il faudrait éditer le fichier `/etc/fstab` pour effectuer un montage statique à chaque initialisation du système. On pourrait par exemple insérer une ligne du type suivant à la fin du fichier.

```
198.51.100.2:/home    /ahome    nfs4     0    0
```

2.4.2. Opérations automatisées de (montage|démontage) NFS

Note

Il existe plusieurs implémentations libres pour le service d'automontage. On se limite ici au logiciel lié au noyau Linux.

Avertissement

Les montages manuels et le service d'automontage ne font pas bon ménage ! Il faut absolument démonter tous les systèmes de fichiers NFS avant d'aborder cette partie.

Dans cette section, on reprend le processus de montage précédent en utilisant le service d'automontage. L'objectif étant de rendre les opérations d'accès au système de fichiers réseau totalement transparentes pour l'utilisateur, le recours au montage manuel doit être évité le plus possible.

Q53. Quel est le paquet qui contient les outils nécessaires au fonctionnement de l'automontage ?

Interroger les méta données dans le cache du gestionnaire de paquets en cherchant le mot clé **automount**.

La recherche dans le champ description du catalogue des paquets disponibles donne les résultats suivants.

```
# aptitude search "?description(automount)"
p   autodir         - Automatically creates home and group directories for LDAP/NIS/SQL/local acco
p   autofs          - kernel-based automounter for Linux
p   autofs-hesiod   - Hesiod map support for autofs
p   autofs-ldap     - LDAP map support for autofs
p   halevt          - generic handler for HAL events
p   libamu-dev      - Support library for amd the 4.4BSD automounter (development)
p   libamu4         - Support library for amd the 4.4BSD automounter (runtime)
p   libnss-cache    - NSS module for using nsscache-generated files
p   ltspfsd         - Fuse based remote filesystem hooks for LTSP thin clients
p   nsscache        - asynchronously synchronise local NSS databases with remote directory service
p   udisks-glue     - simple automount daemon with support for user-defined actions
```

Dans le contexte de ces manipulations, c'est le paquet `autofs` qui nous intéresse.

Q54. Comment créer un compte utilisateur local baptisé `etu-nfs` avec un répertoire utilisateur situé sous la racine `/ahome` dont les fichiers ét répertoires sont placés sur le serveur NFS ?

Après consultation des pages de manuels de la commande **adduser**, on dispose des options de création de compte respectant les deux critères énoncés. L'option `--home` permet de désigner le répertoire utilisateur dans l'arborescence système et l'option `--no-create-home` évite la création de ce répertoire sur le système local.

```
# adduser --no-create-home --home /ahome/etu-nfs etu-nfs
# id etu-nfs
uid=1001(etu-nfs) gid=1001(etu-nfs) groupes=1001(etu-nfs)
```

Les identifiants numériques `uid/gid` jouent un rôle important dans la suite des manipulations. Voir Section 2.6, « Gestion des droits sur le système de fichiers NFS ».

Q55. Quels sont les fichiers de configuration du service d'automontage à éditer ou créer pour que l'utilisateur `etu-nfs` ait accès à ses données personnelles ?

Utiliser les fichiers exemples fournis avec le paquet, les pages de manuels associées et créer un fichier spécifique pour la gestion des comptes utilisateurs.

La liste des fichiers du paquet `autofs` montre qu'il existe une page de manuel consacrée au fichier principal de configuration du service : `/etc/auto.master`. Ces informations permettent de configurer

un point de montage au dessous duquel doivent se trouver les répertoires utilisateurs. Ces derniers utilisent un fichier de configuration propre : `/etc/auto.home`.

1. On définit la racine de montage `/ahome` dans le fichier de configuration principal `/etc/auto.master`. Cette racine de montage pointe vers le fichier de configuration dédié au montage automatique des répertoires des utilisateurs.

```
# grep -v ^# /etc/auto.master
/ahome  /etc/auto.home
```

2. Le fichier `/etc/auto.home` utilise une syntaxe particulière pour que le montage du système de fichiers du serveur soit générique et indépendant du nombre des comptes utilisateurs.

```
# cat /etc/auto.home
*       -fstype=nfs4    198.51.100.2:/home/&
```

   - Le premier paramètre est le symbole ***** qui se substitue au nom d'utilisateur : `etu-nfs` dans notre exemple.
   - Le deuxième paramètre `-fstype=nfs4` correspond à une option de montage qui privilégie la version 4 du protocole NFS. Le jeu des options de montage est le même que pour un montage statique.
   - Le troisième paramètre est l'adresse IP du serveur. Comme on ne dispose pas d'un service DNS à ce stade de la progression des travaux pratiques, on utilise directement les adresses IP.
   - Le répertoire `/home/` correspond à la configuration de l'exportation NFS sur le serveur. Le répertoire `/home/` est situé sous la racine d'exportation qui est uniquement connue du serveur.
   - Le symbole **&** indique la répétition du premier paramètre : le nom d'utilisateur.

Q56. Quelles sont les conditions à respecter sur le client et le serveur NFS pour que l'utilisateur `etu-nfs` ait la capacité à écrire dans son répertoire personnel ?

Rechercher les attributs d'un compte utilisateur qui correspondent aux propriétés des objets d'un système de fichiers au sens général.

Les identifiants numériques `uid/gid` doivent nécessairement être identiques sur le client et le serveur NFS. Toute la gestion des droits sur le système de fichiers est conditionnée par ces valeurs.

Q57. Comment prendre l'identité de l'utilisateur `etu-nfs` pour tester la validité du montage ?

Cette validation suppose que l'utilisateur puisse atteindre son répertoire et que l'on visualise l'automontage avec les commandes **mount** et **df**.

C'est la commande **su** qui permet de «changer d'identité» sur le système. On l'utilise donc pour prendre l'identité de l'utilisateur dont le répertoire est situé sur le serveur NFS. Pour que l'opération de montage automatique ait lieu, il suffit de se placer dans ce répertoire.

```
root@vm-nfs-client:/home/etu# su etu-nfs
etu-nfs@vm-nfs-client:/home/etu$ cd
etu-nfs@vm-nfs-client:~$ pwd
/ahome/etu-nfs
etu-nfs@vm-nfs-client:~$ df -h
Sys. fich.                 Taille Util. Dispo Uti% Monté sur
rootfs                        30G  908M   28G   4% /
udev                          10M     0   10M   0% /dev
tmpfs                         50M  264K   50M   1% /run
/dev/mapper/vm0-root          30G  908M   28G   4% /
tmpfs                        5,0M     0  5,0M   0% /run/lock
tmpfs                        100M     0  100M   0% /run/shm
/dev/vda1                    228M   20M  196M  10% /boot
198.51.100.2:/home/etu-nfs    30G  1,1G   28G   4% /ahome/etu-nfs
etu-nfs@vm-nfs-client:~$ mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,relatime,size=10240k,nr_inodes=62070,mode=755)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,noexec,relatime,size=50896k,mode=755)
/dev/mapper/vm0-root on / type ext3 (rw,relatime,errors=remount-ro,barrier=1,data=ordered)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k)
tmpfs on /run/shm type tmpfs (rw,nosuid,nodev,noexec,relatime,size=101780k)
/dev/vda1 on /boot type ext2 (rw,relatime,errors=continue)
rpc_pipefs on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
/etc/auto.home on /ahome type autofs (rw,relatime,fd=6,pgrp=4475, \
                                                timeout=300,minproto=5,maxproto=5,indirect)
198.51.100.2:/home/etu-nfs on /ahome/etu-nfs type nfs4 (rw,relatime,vers=4, \
                                                        rsize=262144,wsize=262144,namlen=255, \
       hard,proto=tcp,timeo=600,retrans=2, \
       sec=sys,clientaddr=198.51.100.3, \
       minorversion=0,local_lock=none, \
       addr=198.51.100.2)
```

Bien sûr, ces manipulations ne sont possibles que si la configuration du serveur est effective.

Q58. Réaliser une capture réseau lors de l'exécution des commandes et relever les numéros de ports caractéristiques de ces transactions. Est-il possible de retrouver les informations échangées dans les données de capture ?

La marche à suivre est identique à celle de la même question côté serveur NFS.

2.5. Configuration du serveur NFS

Le rôle du serveur NFS est de mettre à disposition sur le réseau une partie de son arborescence locale de système de fichiers. On parle d'«exportation».

> **Note**
>
> Il existe plusieurs implémentations libres de serveur NFS. On se limite ici à l'utilisation du logiciel lié au noyau Linux.

Cette section traite de l'installation d'un serveur NFS en version 4 dont le but est d'exporter le contenu des répertoires utilisateurs vers les clients.

Q59. Quel est le paquet qui contient les outils nécessaires au fonctionnement du serveur NFS ? Installez ce paquet.

Interroger les méta données du gestionnaire de paquets pour identifier le nom du paquet à installer.

La recherche des mots clés `nfs` et `server` donne les résultats suivants.

```
# aptitude search '?and(nfs, server)'
p   nfs-kernel-server   - gestion du serveur NFS du noyau
v   nfs-server
```

Les informations données par la commande **`# aptitude show nfs-kernel-server`** permettent de confirmer qu'il s'agit bien du paquet à installer.

```
# aptitude install nfs-kernel-server
```

Q60. Quel est le fichier de configuration principal de gestion des exportations NFS ?

Rechercher dans le support *Linux NFS-HOWTO*.

Quelles que soient les versions du protocole, c'est toujours le fichier `/etc/exports` qui est utilisé. Ce fichier est présenté dans le support *Linux NFS-HOWTO*. Le fichier livré avec le paquet contient, en commentaires, deux exemples complets de configuration NFSv3 et NFSv4. C'est ce dernier exemple que l'on adapte pour traiter les questions suivantes.

Q61. Créer le répertoire `/home/exports/home`. Quelle est la syntaxe à utiliser dans le fichier de configuration pour «exporter» ce répertoire ?

Rechercher dans les supports *Linux NFS-HOWTO* et *Nfsv4 configuration*. On peut aussi utiliser les pages de manuels fournies avec le paquet du serveur NFS.

En exploitant la documentation *Nfsv4 configuration* et l'exemple donné dans le fichier de configuration, on applique la configuration suivante.

```
# mkdir -p /home/exports/home
# grep -v ^# /etc/exports
/home/exports           198.51.100.0/24(rw,sync,fsid=0,crossmnt,no_subtree_check)
/home/exports/home      198.51.100.0/24(rw,sync,no_subtree_check)
```

Pour les besoins de ces travaux pratiques, les fonctions de sécurité *Kerberos* ne sont pas utilisées. On utilise l'appartenance au réseau IP comme critère de contrôle d'accès ; ce qui correspond à un niveau de sécurité faible.

Note

Du point de vue pédagogique, le choix d'une progression en séances de travaux pratiques autonomes et indépendantes implique que l'étude de la configuration *Kerberos* soit repoussée en dernière étape. En effet, le service *Kerberos* intervient à tous les niveaux : LDAP, NFS et authentification. Il peut faire l'objet d'une étude de synthèse supplémentaire une fois que les configurations des différentes fonctions ont été validées l'une après l'autre.

En ce qui concerne les options entre parenthèses, elles sont documentées dans les pages de manuels `exports` : **`# man 5 exports`**. Les éléments suivants en sont extraits.

- `rw` : autoriser les requêtes en lecture et en écriture sur le volume NFS. Le comportement par défaut est d'interdire toute requête qui modifierait le système de fichiers.
- `sync` : ne répondre aux requêtes qu'après l'exécution de tous les changements sur le support réel.
- `fsid=0` : avec NFSv4, un système de fichiers particulier est la racine de tous les systèmes de fichiers partagés. Il est défini par fsid=root ou fsid=0, qui veulent tous deux dire exactement la même chose.
- `crossmnt` : cette option permet aux clients de se déplacer du système de fichiers marqué `crossmnt` aux systèmes de fichiers partagés montés dessus. Voir l'option nohide.
- `no_subtree_check` : cette option neutralise la vérification de sous-répertoires, ce qui a des subtiles implications au niveau de la sécurité, mais peut améliorer la fiabilité dans certains cas. Si un sous-répertoire dans un système de fichiers est partagé, mais que le système de fichiers ne l'est pas, alors chaque fois qu'une requête NFS arrive, le serveur doit non seulement vérifier que le fichier accédé est dans le système de fichiers approprié (ce qui est facile), mais aussi qu'il est dans l'arborescence partagée (ce qui est plus compliqué). Cette vérification s'appelle subtree_check.

Q62. Qu'est-ce qui distingue l'exportation d'une arborescence entre les versions 3 et 4 du protocole NFS ?

Rechercher dans les différences relatives à la notion de nommage dans les manipulations proposées dans les supports *Linux NFS-HOWTO* et *Nfsv4 configuration*.

Donner la signification du paramètre `fsid=0` dans la documentation relative à la version 4. Proposer une analogie avec le fonctionnement d'un serveur Web.

Au delà des évolutions du protocole, c'est la cohérence du système de nommage qui distingue la version 4 du système de fichiers réseau. Il s'agit de garantir qu'un objet (fichier ou répertoire) soit représenté de la même manière sur un serveur et sur ses clients.

Dans le contexte de ces travaux pratiques les répertoires utilisateurs doivent être référencés à partir d'une racine nommée `/ahome/`.

Du point de vue infrastructure, l'utilisation de cette référence de nommage unique présente un avantage non négligeable. En effet, les répertoires d'exportation tels qu'ils ont été définis dans le fichier `/etc/exports` donné ci-dessus désignent un espace de stockage physique. La racine `/ahome/` désigne un espace de stockage logique. Ce schéma de nommage logique doit rester constant alors que les volumes de stockages physique peuvent migrer et se déplacer, être étendus, etc. sans qu'il soit nécessaire de remettre en question la configuration des clients.

Les différences entre les manipulations proposées dans les supports *Linux NFS-HOWTO* et *Nfsv4 configuration* traduisent les différences de conception entre les deux générations du protocole NFS. On peut relever deux paramètres importants sur le serveur.

- L'option `fsid=0`, présente dans le fichier `/etc/exports/`, permet de définir une *racine de montage* tout comme on le verrait sur un serveur Web. Le paramètre de configuration `DocumentRoot /var/www` du serveur *apache2* désigne la racine à partir de laquelle les pages Web publiées sont référencées. Cette racine est indépendante de l'arborescence du système de fichier local du serveur.

- L'utilisation d'un montage local avec l'option `bind` de la commande **mount** permet de mettre en cohérence l'arborescence du serveur et de ses clients. Ainsi, le répertoire `/ahome/` présente les mêmes objets que l'on soit connecté sur le serveur ou sur un client. Le schéma de nommage est donc cohérent.

  Le montage local peut se faire manuellement sur le serveur avec la syntaxe suivante.

```
# mkdir /ahome
# mount --bind /home/exports/home /ahome
```

  Une fois la configuration validée, on peut intégrer ce montage local dans la configuration système pour que l'opération soit effectuée à chaque initialisation. Il faut alors éditer le fichier de configuration dédié aux montages des volumes locaux du système : `/etc/fstab`. Voici un exemple donnant les dernières lignes d'un fichier `/etc/fstab` de serveur.

```
# tail -4 /etc/fstab
UUID=15fb1316-1260-44bf-8931-ff052d99d315 /boot           ext2    defaults        0       2
/dev/mapper/vm0-root   /       ext3    errors=remount-ro 0       1
/dev/mapper/vm0-swap_1 none    swap    sw                0       0
/home/exports/home     /ahome  none    defaults,bind     0       0
```

Q63. Quelle est la commande qui permet de visualiser l'état courant de l'arborescence exportée ?

Rechercher dans la liste des fichiers du paquet relatif au serveur NFS.

La liste des commandes fournies avec le paquet `nfs-kernel-server` est la suivante.

```
# dpkg -L nfs-kernel-server | grep bin
/usr/sbin
/usr/sbin/exportfs
/usr/sbin/rpc.mountd
/usr/sbin/rpc.nfsd
/usr/sbin/rpc.svcgssd
```

Chacune de ces commandes dispose de pages de manuels. En consultant ces pages, on relève que la commande **exportfs** est chargée de la gestion de la liste des systèmes de fichiers partagés par NFS. L'exécution de cette commande sans argument affiche la liste des répertoires exportés. Dans notre cas, on obtient le résultat suivant.

```
# exportfs
/home/exports   198.51.100.0/24
/home/exports/home
                198.51.100.0/24
```

On peut ainsi vérifier que les directives données dans le fichier `/etc/exports` sont effectivement appliquées.

Q64. Quelles sont les principales options disponibles pour l'exportation d'une arborescence ? Relever la signification des paramètres.

Rechercher dans le support *Linux NFS-HOWTO*. On doit s'intéresser plus particulièrement aux options : `(rw|ro)`, `(sync|async)` et `*squash`.

Voici quelques éléments de réponse issus des pages de manuels : **`# man 5 exports`**

- L'option `rw` autorise les requêtes en lecture et en écriture sur le volume NFS alors que l'option `ro` interdit toute requête qui modifierait le système de fichiers.

- L'option `async` permet au serveur de transgresser le protocole NFS en répondant aux requêtes avant que tous les changements impliqués par la requête en cours n'aient été effectués sur le support réel (par exemple, le disque dur). L'utilisation de cette option améliore généralement les performances, mais au risque de perdre ou de corrompre des données en cas de redémarrage brutal du serveur. À l'opposé, l'option `sync` impose de ne répondre aux requêtes qu'après l'exécution de tous les changements sur le support réel.

- Les options `*_squash` sont relatives aux transformations des identifiants `uid` et `gid` entre le serveur NFS et ses clients. Par exemple, l'option `root_squash` transforme les requêtes avec un couple `uid/gid` à 0 (ie. le super-utilisateur) en un couple `uid/gid` anonyme.

Q65. Comment créer un compte utilisateur local baptisé `etu-nfs` avec un répertoire utilisateur situé sous la racine /ahome ?

Après consultation des pages de manuels de la commande **adduser**, on dispose des options de création de compte respectant le critère énoncé. L'option `--home` permet de désigner le répertoire utilisateur dans l'arborescence système.

```
# adduser --home /ahome/etu-nfs etu-nfs
# id etu-nfs
uid=1001(etu-nfs) gid=1001(etu-nfs) groupes=1001(etu-nfs)
```

Les identifiants numériques `uid/gid` jouent un rôle important dans la suite des manipulations. Voir Section 2.6, « Gestion des droits sur le système de fichiers NFS ».

Q66. Réaliser une capture et relever les numéros de ports caractéristiques de des transactions de montage. Est-il possible de retrouver les informations échangées dans les données de capture ?

Pour réaliser cette capture, il faut synchroniser les opérations entre les postes client et serveur. On commence par le lancement du l'analyseur réseau puis on effectue un montage manuel par exemple pour caractériser les transactions réseau.

Voici un extrait de capture en mode console qui illustre la séquence de commande suivante exécutée sur le poste client.

```
# showmount -e 198.51.100.2
Export list for 198.51.100.2:
/home/exports/home 198.51.100.0/24
/home/exports      198.51.100.0/24
# mount -t nfs4 198.51.100.2:/home /ahome
# ls -lAh /ahome
# umount /ahome/
```

Côté serveur, la capture réseau donne les résultats suivants.

```
Source  Destination    Protocol Length Info
198.51.100.3    198.51.100.2   Portmap  98     V2 GETPORT Call (Reply In 2) MOUNT(100005) V:3 TCP
198.51.100.2    198.51.100.3   Portmap  70     V2 GETPORT Reply (Call In 1) Port:43090
198.51.100.3    198.51.100.2   TCP      74     lanserver > 43090 [SYN] Seq=0
198.51.100.2    198.51.100.3   TCP      74     43090 > lanserver [SYN, ACK] Seq=0 Ack=1
198.51.100.3    198.51.100.2   TCP      66     lanserver > 43090 [ACK] Seq=1 Ack=1
198.51.100.3    198.51.100.2   MOUNT    150    V3 EXPORT Call (Reply In 8)
198.51.100.2    198.51.100.3   TCP      66     43090 > lanserver [ACK] Seq=1 Ack=85
198.51.100.2    198.51.100.3   MOUNT    206    V3 EXPORT Reply (Call In 6)
198.51.100.3    198.51.100.2   TCP      66     lanserver > 43090 [ACK] Seq=85 Ack=141
198.51.100.3    198.51.100.2   TCP      66     lanserver > 43090 [FIN, ACK] Seq=85 Ack=141
198.51.100.2    198.51.100.3   TCP      66     43090 > lanserver [FIN, ACK] Seq=141 Ack=86
198.51.100.3    198.51.100.2   TCP      66     lanserver > 43090 [ACK] Seq=86 Ack=142
198.51.100.3    198.51.100.2   TCP      74     755 > nfs [SYN] Seq=0
198.51.100.2    198.51.100.3   TCP      74     nfs > 755 [SYN, ACK] Seq=0 Ack=1
198.51.100.3    198.51.100.2   TCP      66     755 > nfs [ACK] Seq=1 Ack=1
198.51.100.3    198.51.100.2   NFS      110    V4 NULL Call (Reply In 20)
198.51.100.2    198.51.100.3   TCP      66     nfs > 755 [ACK] Seq=1 Ack=45
198.51.100.2    198.51.100.3   NFS      94     V4 NULL Reply (Call In 18)
198.51.100.3    198.51.100.2   TCP      66     755 > nfs [ACK] Seq=45 Ack=29
198.51.100.3    198.51.100.2   NFS      186    V4 Call (Reply In 23) PUTROOTFH | GETATTR
198.51.100.2    198.51.100.3   NFS      302    V4 Reply (Call In 22) PUTROOTFH | GETATTR
198.51.100.3    198.51.100.2   NFS      190    V4 Call (Reply In 25) GETATTR FH:0x62d40c52
198.51.100.2    198.51.100.3   NFS      158    V4 Reply (Call In 24) GETATTR
198.51.100.3    198.51.100.2   NFS      194    V4 Call (Reply In 27) GETATTR FH:0x62d40c52
198.51.100.2    198.51.100.3   NFS      178    V4 Reply (Call In 26) GETATTR
198.51.100.3    198.51.100.2   NFS      190    V4 Call (Reply In 29) GETATTR FH:0x62d40c52
198.51.100.2    198.51.100.3   NFS      158    V4 Reply (Call In 28) GETATTR
198.51.100.3    198.51.100.2   NFS      194    V4 Call (Reply In 31) GETATTR FH:0x62d40c52
198.51.100.2    198.51.100.3   NFS      178    V4 Reply (Call In 30) GETATTR
198.51.100.3    198.51.100.2   NFS      190    V4 Call (Reply In 33) GETATTR FH:0x62d40c52
198.51.100.2    198.51.100.3   NFS      142    V4 Reply (Call In 32) GETATTR
198.51.100.3    198.51.100.2   NFS      190    V4 Call (Reply In 35) GETATTR FH:0x62d40c52
198.51.100.2    198.51.100.3   NFS      158    V4 Reply (Call In 34) GETATTR
198.51.100.3    198.51.100.2   NFS      194    V4 Call (Reply In 37) GETATTR FH:0x62d40c52
198.51.100.2    198.51.100.3   NFS      282    V4 Reply (Call In 36) GETATTR
198.51.100.3    198.51.100.2   NFS      202    V4 Call (Reply In 39) ACCESS FH:0x62d40c52,
                                               [Check: RD LU MD XT DL]
198.51.100.2    198.51.100.3   NFS      298    V4 Reply (Call In 38) ACCESS,
                                               [Access Denied: MD XT DL], [Allowed: RD LU]
198.51.100.3    198.51.100.2   NFS      210    V4 Call (Reply In 41) LOOKUP DH:0x62d40c52/home
198.51.100.2    198.51.100.3   NFS      318    V4 Reply (Call In 40) LOOKUP
198.51.100.3    198.51.100.2   TCP      66     755 > nfs [ACK] Seq=1325 Ack=1541
198.51.100.3    198.51.100.2   NFS      202    V4 Call (Reply In 44) GETATTR FH:0x8834bc40
198.51.100.2    198.51.100.3   NFS      282    V4 Reply (Call In 43) GETATTR
198.51.100.3    198.51.100.2   TCP      66     755 > nfs [ACK] Seq=1461 Ack=1757
198.51.100.3    198.51.100.2   NFS      210    V4 Call (Reply In 47) ACCESS FH:0x8834bc40,
                                               [Check: RD LU MD XT DL]
198.51.100.2    198.51.100.3   NFS      298    V4 Reply (Call In 46) ACCESS,
                                               [Access Denied: MD XT DL], [Allowed: RD LU]
198.51.100.3    198.51.100.2   NFS      202    V4 Call (Reply In 49) GETATTR FH:0x8834bc40
198.51.100.2    198.51.100.3   NFS      282    V4 Reply (Call In 48) GETATTR
198.51.100.3    198.51.100.2   NFS      226    V4 Call (Reply In 51) READDIR FH:0x8834bc40
198.51.100.2    198.51.100.3   NFS      362    V4 Reply (Call In 50) READDIR
198.51.100.3    198.51.100.2   TCP      66     755 > nfs [ACK] Seq=1901 Ack=2501
198.51.100.3    198.51.100.2   Portmap  98     V2 GETPORT Call (Reply In 58) MOUNT(100005) V:3 UD
198.51.100.2    198.51.100.3   Portmap  70     V2 GETPORT Reply (Call In 57) Port:39073
198.51.100.3    198.51.100.2   MOUNT    82     V3 NULL Call (Reply In 60)
198.51.100.2    198.51.100.3   MOUNT    66     V3 NULL Reply (Call In 59)
198.51.100.3    198.51.100.2   MOUNT    134    V3 UMNT Call (Reply In 62) /home
198.51.100.2    198.51.100.3   MOUNT    66     V3 UMNT Reply (Call In 61)
198.51.100.3    198.51.100.2   TCP      66     755 > nfs [FIN, ACK] Seq=1901 Ack=2501
198.51.100.2    198.51.100.3   TCP      66     nfs > 755 [FIN, ACK] Seq=2501 Ack=1902
198.51.100.3    198.51.100.2   TCP      66     755 > nfs [ACK] Seq=1902 Ack=2502
```

Les points caractéristiques illustrés par cette capture sont : l'utilisation du protocole TCP, l'utilisation du port enregistré `2049/nfs`, les appels de sous-programmes par lots.

2.6. Gestion des droits sur le système de fichiers NFS

Le contrôle les droits sur les objets de l'arborescence exportée par le serveur NFS est limité au masque de permissions de ces objets. Il est donc important de faire correspondre les identifiants `uid` et `gid` entre le client et le serveur.

Les manipulations suivantes sont à réaliser en «concertation» entre les administrateurs des postes client et serveur. Le compte utilisateur `etu-nfs` doit avoir été créé sur le serveur et sur le client.

> **Note**
>
> Ces manipulations se font sans système de gestion centralisé de l'authentification. L'utilisation d'un annuaire LDAP pour fournir une base de comptes utilisateurs fait l'objet d'un support de travaux pratiques qui vient après celui-ci. Ce support se concentre sur le volet système de fichiers réseau.

Q67. Quelles sont les valeurs numériques des identifiants `uid` et `gid` du compte utilisateur `etu-nfs` sur le client et sur le serveur NFS ?

Si les valeurs différent entre le client et le serveur, il faut détruire ces comptes utilisateurs et reprendre les options de la commande **adduser** pour fournir ces valeurs de façon explicite.

L'extrait du résultat de l'instruction **`# adduser --help`** ci-dessous montre les options utiles.

```
adduser [--home DIR] [--shell SHELL] [--no-create-home] [--uid ID]
[--firstuid ID] [--lastuid ID] [--gecos GECOS] [--ingroup GROUP | --gid ID]
[--disabled-password] [--disabled-login] USER
   Ajoute un utilisateur normal
```

Reprendre la question sur la création d'un compte utilisateur local dont le répertoire est situé sur le serveur NFS.

Q68. Sur quel poste peut-on créer des fichiers et des répertoires avec des masques de permissions ayant d'autres valeurs `uid` et `gid` que celles de l'utilisateur `etu-nfs` ? Quelles sont les options des commandes **chmod** et **chown** à utiliser pour réaliser ces opérations ?

Utiliser les pages de manuels des commandes.

C'est sur le serveur que le super-utilisateur a la possibilité de créer n'importe quel objet avec n'importe quel propriétaire dans la mesure où le système de fichiers est local et non réseau.

```
# cd /ahome/etu-nfs/
root@srvr:/ahome/etu-nfs# touch ThisOneIsMine
root@srvr:/ahome/etu-nfs# chown etu-nfs.etu-nfs ThisOneIsMine
root@srvr:/ahome/etu-nfs# touch ThisOneIsNotMine
root@srvr:/ahome/etu-nfs# chown 2000.2000 ThisOneIsNotMine
root@srvr:/ahome/etu-nfs# ll This*
-rw-r--r-- 1 etu-nfs etu-nfs 0 21 avril 00:36 ThisOneIsMine
-rw-r--r-- 1    2000    2000 0 21 avril 00:37 ThisOneIsNotMine
```

Côté client, les objets créés sont biens visibles mais la vue réseau du système de fichiers NFS passe par une correspondance des propriétaires.

```
root@clnt:/home/etu# su etu-nfs
etu-nfs@clnt:/home/etu$ cd
etu-nfs@clnt:~$ ll This*
-rw-r--r-- 1 etu-nfs etu-nfs 0 21 avril 00:36 ThisOneIsMine
-rw-r--r-- 1 nobody  nogroup 0 21 avril 00:37 ThisOneIsNotMine
etu-nfs@clnt:~$ touch ThisOneIsMine
etu-nfs@clnt:~$ touch ThisOneIsNotMine
touch: impossible de faire un touch « ThisOneIsNotMine »: Permission non accordée
```

Q69. Quel est le service qui assure la conformité des identifiants entre serveur et client NFS ?

Reprendre la liste des service RPC actifs sur les deux systèmes.

Voir Section 2.2, « Protocole NFS et topologie de travaux pratiques ». Le démon `rpc.idmapd` est fourni avec le paquet `nfs-common`.

2.7. Système de fichiers NFS & sécurité

Lors de leur conception, au début des années 80, la sécurité des mécanismes RPC la sécurité n'était pas une préoccupation. Il a donc fallu appliquer des fonctions de sécurité sur des protocoles qui n'étaient pas prévus pour.

Avec les versions 2 et 3 du protocole NFS, le service `portmap` ne dispose d'aucun mécanisme interne de sécurité. C'est la raison pour laquelle on lui associe les utilitaires *TCP wrapper* qui «encadrent» les accès aux appels RPC. Cette sécurisation à minima est très limitée puisqu'elle se limite à définir les adresses IP des hôtes qui peuvent accéder au service. Il n'est donc pas réaliste d'utiliser les versions 2 et 3 du protocole NFS sur un réseau étendu sans passer par des tunnels. De plus, l'affectation dynamique de numéro de port pour les montages avec ces versions du protocole ne facilite pas la configuration des pare feux.

Avec le développement de la version 4 du protocole NFS, des fonctions d'authentification basées sur la technologie *Kerberos* ont été introduites. L'affectation dynamique est abandonnée au profit d'un numéro de port unique : tcp/2049.

2.8. Documents de référence

Systèmes de fichiers réseau : NFS & CIFS

Systèmes de fichiers réseau : présentation des modes de fonctionnement des systèmes de fichiers réseau NFS & CIFS. Cette présentation est a consulter avant d'aborder la Section 2.2, « Protocole NFS et topologie de travaux pratiques ».

Linux NFS-HOWTO

Linux NFS-HOWTO : documentation historique complète sur la configuration d'un serveur et d'un client NFS jusqu'à la version 3 inclue.

Nfsv4 configuration

Nfsv4 configuration : traduction française extraite des pages du projet CITI de l'université du Michigan.

Autres liens

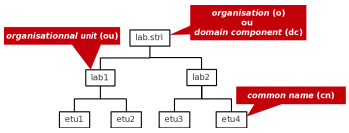
Marque-pages Delicious sur NFSv4

CHAPITRE 3

Introduction aux annuaires LDAP avec OpenLDAP

Résumé

Dans ce support de travaux pratiques, on explore le service d'annuaire LDAP. On présente succinctement les éléments constitutifs d'un annuaire puis on étudie la configuration d'un service d'annuaire basé sur le logiciel OpenLDAP. Ensuite, on étudie la configuration de l'accès aux entrées de l'annuaire depuis un poste client. Les informations délivrées par l'annuaire sont les propriétés de comptes utilisateurs stockées dans la classe d'objet `posixAccount`.



3.1. Adressage IP des postes de travail

Tableau 3.1. Affectation des adresses et des réseaux IP de la salle 211

Poste 1	Poste 2	Passerelle par défaut	Organisation
christophsis	corellia	`172.20.12.17/29`	o: zone1.lan-211.stri
delaya	kasyyyk	`10.3.20.161/27`	o: zone2.lan-211.stri
korriban	kessel	`192.168.143.129/25`	o: zone3.lan-211.stri
mygeeto	nelvaan	`10.141.0.161/27`	o: zone4.lan-211.stri
rattatak	saleucami	`192.168.10.81/28`	o: zone5.lan-211.stri
taris	teth	`10.31.0.193/26`	o: zone6.lan-211.stri
utapau	yavin	`192.168.142.65/26`	o: zone7.lan-211.stri

Tableau 3.2. Affectation des adresses et des réseaux IP de la salle 213

Poste 1	Poste 2	Passerelle par défaut	Organisation
alderaan	bespin	`10.7.10.1/23`	o: zone1.lan-213.stri
centares	coruscant	`192.168.110.129/25`	o: zone2.lan-213.stri
dagobah	endor	`172.19.113.65/26`	o: zone3.lan-213.stri
felucia	geonosis	`10.3.2.1/23`	o: zone4.lan-213.stri
hoth	mustafar	`172.20.130.25/29`	o: zone5.lan-213.stri
naboo	tatooine	`172.19.115.193/26`	o: zone6.lan-213.stri

Toutes les questions de ce support peuvent être traitées avec le document de référence : *OpenLDAP Software 2.4 Administrator's Guide*. Il est cependant nécessaire de faire la correspondance entre les services décrits dans le document et les paquets de la distribution *Debian GNU/Linux*.

Pour chaque paire de postes de travaux pratiques, il faut attribuer les rôles de serveur et de client. Le serveur doit mettre à disposition de son poste client une base de données utilisateurs. L'objectif en fin de séance de travaux pratiques est de pouvoir se connecter sur un poste client avec ses authentifiants `login/password`.

Relativement au précédent support sur l'*Introduction au système de fichiers réseau NFSv4*, on ne dispose pas d'un système de fichiers réseau. Ici, seule l'authentification est fonctionnelle et il n'est pas possible d'accéder au répertoire utilisateur stocké sur le serveur NFS.

3.2. Principes d'un annuaire LDAP

Dans l'histoire des systèmes Unix, les services de *nommage* ont connu de nombreuses évolutions avec le développement de l'Internet et des volumes d'informations à partager.

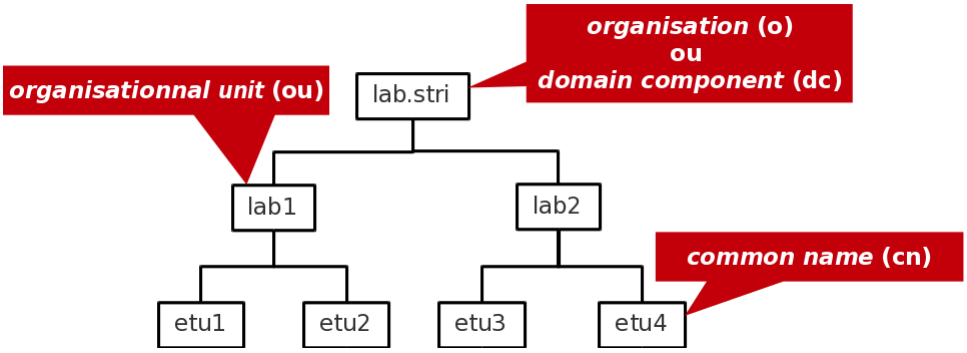
Au début des années 80, un premier service baptisé *Network Information Service* (NIS) a vu le jour. Ce service est une méthode de distribution de la base de données des utilisateurs, de fichiers de configuration, d'authentification et d'autres données entre les hôtes d'un réseau local. Le logiciel NIS développé par Sun Microsystems™ fonctionne sur le mode Client/Serveur à partir d'une base de données «à plat» (*flat bindery base*). Son utilisation est étudiée dans le support de travaux pratiques *Introduction au service NIS*. Avec un service NIS, il n'est pas possible de constituer des groupes logiques ayant des attributs propres. Cette limitation est rapidement devenue critique avec l'augmentation du nombres des utilisateurs et des clients.

D'autres services plus complets tels que NIS+ ou *kerberos* qui n'assure que la partie authentification ont été développés par la suite. Depuis quelques années, les annuaires LDAP ou *Lightweight Directory Access Protocol* se sont imposés comme étant l'outil d'échange universel des paramètres utilisateurs.

Pour définir ce qu'est le service LDAP, on peut retenir les caractéristiques suivantes.

- Un service de publication d'annuaire
- Un protocole d'accès aux annuaires de type X.500 ou *Lightweight Directory Access Protocol*
- Un dépôt de données basées sur des attributs ou un «genre» de base de données
- Un logiciel optimisé pour les recherches avancées et les lectures
- Une implémentation client/serveur
- Un mécanisme extensible de schémas de description de classes d'objets

Les entrées (*Directory Service Entry*) d'un annuaire LDAP sont distribuées suivant une arborescence (*Directory Information Tree*) hiérarchisée que l'on peut voir comme un système de fichiers avec ses répertoires et ses fichiers. Au sommet de l'arborescence on trouve un nom de racine (*Domain Component*) ou suffixe.



Arborescence LDAP élémentaire - vue complète

L'adresse d'une entrée de l'annuaire LDAP est appelée : *distinguished name* ou dn. En reprenant l'exemple d'arborescence ci-dessus, les adresses des différentes entrées sont notées comme suit.

- `dn: dc=lab,dc=stri`
- `dn: ou=lab1,dc=lab,dc=stri`

  `dn: ou=lab2,dc=lab,dc=stri`
- `dn: cn=etu1,ou=lab1,dc=lab,dc=stri`

  `dn: cn=etu2,ou=lab1,dc=lab,dc=stri`

  `dn: cn=etu3,ou=lab2,dc=lab,dc=stri`

  `dn: cn=etu4,ou=lab2,dc=lab,dc=stri`

L'adresse de chaque entrée appartient à une classe d'objet (*ObjectClass*) spécifiée dans un schéma (*schema*). En reprenant les mêmes exemples d'entrées, on peut associer les classes d'objets correspondantes.

entry	***objectclass***

`o: lab.stri`
`dc: lab`
`dc: stri`
	`organisation`
	`dcObject`
	`dcObject`

`ou: lab1`	`organisationalUnit`
`cn: etu1` `sn: etu1`	`inetOrgPerson`

Un schéma peut être vu comme un ensemble de règles qui décrivent la nature des données stockées. C'est un outil qui aide à maintenir la cohérence, la qualité et qui évite la duplication des données dans l'annuaire. Les attributs des classes d'objets déterminent les règles qui doivent être appliquées à une entrée. Un schéma contient les éléments suivants.

- Les attributs requis
- Les attributs autorisés
- Les règles de comparaison des attributs
- Les valeurs limites qu'un attribut peut recevoir
- Les restrictions sur les informations qui peuvent être enregistrées

3.3. Configuration du serveur LDAP

Avant d'aborder la configuration du service LDAP, il faut passer par les étapes rituelles de sélection et d'installation des paquets contenant les outils logiciels du service. Ensuite, il faut identifier les processus, les numéros de ports ouverts et les fichiers de configuration à éditer.

3.3.1. Installation du serveur LDAP

Q70. Quels sont les paquets Debian relatifs au service LDAP ?

Interroger la base de données des paquets pour obtenir les informations demandées.

Dans la requête ci-dessous, on privilégie la recherche dans les champs de description des paquets.

```
$ aptitude search '?description(OpenLDAP)'
p   collectd-core               - démon de statistiques et surveillance - système central
p   golang-openldap-dev         - OpenLDAP client integration for Go, using cgo
p   ldap-git-backup             - Back up LDAP database in an Git repository
p   ldap-utils                  - OpenLDAP utilities
p   ldapscripts                 - Add and remove users and groups (stored in a LDAP directory)
p   libdbd-ldap-perl            - Perl extension for LDAP access via an SQL/Perl DBI interface
i A libldap-2.4-2               - Bibliothèques OpenLDAP
i A libldap-common              - fichiers communs OpenLDAP pour les bibliothèques
p   libldap2-dev                - bibliothèques de développement pour OpenLDAP
p   liblmdb-dev                 - Lightning Memory-Mapped Database development files
p   liblmdb0                    - Lightning Memory-Mapped Database shared library
p   libmozilla-ldap-perl        - LDAP Perl module for the OpenLDAP C SDK
p   libnet-ldapapi-perl         - Perl bindings for OpenLDAP C API
p   libsasl2-modules-ldap       - Cyrus SASL - pluggable authentication modules (LDAP)
p   lmdb-doc                    - Lightning Memory-Mapped Database doxygen documentation
p   lmdb-utils                  - Lightning Memory-Mapped Database Utilities
p   lua-ldap                    - LDAP library for the Lua language
p   python-django-python3-ldap  - Django LDAP user authentication backend (Python2 version)
p   python-ldap                 - LDAP interface module for Python
p   python-ldap-dbg             - LDAP interface module for Python (debug extension)
p   python-lmdb                 - Python binding for LMDB Lightning Memory-Mapped Database
p   python-pyldap               - implements an LDAP client - Python 2.7
p   python-pyldap-doc           - implements an LDAP client - doc
p   python3-django-python3-ldap - Django LDAP user authentication backend (Python3 version)
p   python3-lmdb                - Python 3 binding for LMDB Lightning Memory-Mapped Database
p   python3-pyldap              - implements an LDAP client - Python 3.x
p   ruby-ldap                   - OpenLDAP library binding for Ruby
p   schema2ldif                 - Tool for converting OpenLDAP-style schemas to the LDIF format
p   slapd                       - OpenLDAP server (slapd)
p   smbldap-tools               - Scripts to manage Unix and Samba accounts stored on LDAP
p   vbackup                     - utilitaire modulaire de sauvegarde
```

Q71. Quels sont les paquets Debian à installer pour mettre en œuvre un serveur LDAP ?

Dans liste obtenue en réponse à la question précédente, rechercher les paquets relatifs aux utilitaires et au serveur.

Dans la liste ci-dessus, on retient deux paquets : `ldap-utils` et `slapd`.

```
# aptitude install slapd ldap-utils
Les NOUVEAUX paquets suivants vont être installés :
  ldap-utils libltdl7{a} libodbc1{a} slapd
0 paquets mis à jour, 4 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de télécharger 2 314 ko d'archives. Après dépaquetage, 19,0 Mo
seront utilisés.
Voulez-vous continuer ? [Y/n/?]
```

Lors de l'installation, deux écrans `debconf` demandent la saisie du mot de passe administrateur du service LDAP.

Q72. Comment identifier le ou les processus correspondant au service installé ?

Utiliser une commande d'affichage de la liste des processus actifs sur le système pour identifier le démon correspondant au serveur LDAP.

```
$ ps aux | grep  l[d]ap
openldap  1303  0.0  0.1 1078732 5936 ?        Ssl  19:22   0:00 \
 /usr/sbin/slapd -h ldap:/// ldapi:/// -g openldap -u openldap -F /etc/ldap/slapd.d
```

À partir de ces informations, on identifie le démon serveur `slapd`, le compte utilisateur et le groupe système propriétaires du processus (`openldap`) et enfin le répertoire contenant les fichiers de configuration `/etc/ldap/slapd.d`.

Q73. Comment identifier le ou les numéros de ports ouverts par le service installé ?

Utiliser une commande d'affichage de la liste des ports ouverts sur le système.

Voici deux exemples usuels.

```
# lsof -i | grep l[d]ap
slapd    1303    openldap    8u  IPv4  22394      0t0  TCP *:ldap (LISTEN)
slapd    1303    openldap    9u  IPv6  22395      0t0  TCP *:ldap (LISTEN)
```

```
# ss -tau | grep l[d]ap
tcp    LISTEN     0      128     *:ldap                 *:*
tcp    LISTEN     0      128    :::ldap                :::*
```

Les numéros de port enregistrés pour le service LDAP sont disponibles dans le fichier `/etc/services`.

```
$ grep ldap /etc/services
ldap            389/tcp         # Lightweight Directory Access Protocol
ldap            389/udp
ldaps           636/tcp         # LDAP over SSL
ldaps           636/udp
```

Relativement au indications données par les commandes **lsof** et **ss**, c'est le numéro de port 389 qui est ouvert en écoute lors de l'installation du paquet `slapd`.

3.3.2. Analyse de la configuration du service LDAP

Les versions récentes du logiciel *OpenLDAP* utilisent un mode de configuration basé sur un *Directory Information Tree* ou DIT propre. Cette arborescence de configuration est pointée par le nom `cn=config`. Elle est utilisée pour configurer dynamiquement le démon `slapd`, modifier les définitions de schéma, les index, les listes de contrôle d'accès ACLs, etc. Ce mode de configuration présente un avantage déterminant lorsque l'on exploite des annuaires volumineux : toutes les opérations se font sans interruption de service.

Les documents fournis avec le paquet `slapd` contiennent des informations indispensables à l'analyse du fonctionnement du service.

Q74. Quel est le mode de gestion de la configuration du service adopté depuis la version `2.4.23-3` du paquet de la distribution Debian GNU/Linux ?

Consulter les fichiers de documentation fournis avec le paquet `slapd`.

Les documents relatifs au paquet `slapd` sont situés dans le répertoire `/usr/share/doc/slapd/`. Le fichier `README.Debian.gz` contient un exemple d'instruction de consultation de la configuration du service.

```
# ldapsearch -Y EXTERNAL -H ldapi:/// -b "cn=config"
```

Q75. Quel est le gestionnaire de base de données (*backend*) proposé dans l'annuaire de configuration ?

Reprendre la commande préconisée en réponse à la question précédente en utilisant le type de base de donnée comme filtre.

```
# ldapsearch -Y EXTERNAL -H ldapi:/// -b "cn=config" olcDatabase={1}mdb
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
# extended LDIF
#
# LDAPv3
# base <cn=config> with scope subtree
# filter: olcDatabase={1}mdb
# requesting: ALL
#

# {1}mdb, config
dn: olcDatabase={1}mdb,cn=config
objectClass: olcDatabaseConfig
objectClass: olcMdbConfig
olcDatabase: {1}mdb
olcDbDirectory: /var/lib/ldap
olcSuffix: dc=nodomain
olcAccess: {0}to attrs=userPassword by self write by anonymous auth by * none
olcAccess: {1}to attrs=shadowLastChange by self write by * read
olcAccess: {2}to * by * read
olcLastMod: TRUE
olcRootDN: cn=admin,dc=nodomain
olcRootPW: {SSHA}Hsonccb6iwsCLyvV5Qa8SNbwO9vNVVej
olcDbCheckpoint: 512 30
olcDbIndex: objectClass eq
olcDbIndex: cn,uid eq
olcDbIndex: uidNumber,gidNumber eq
olcDbIndex: member,memberUid eq
olcDbMaxSize: 1073741824

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

Par définition, un annuaire LDAP est une base de données optimisée en lecture. Du point de vue implémentation, les entrées sont stockées sous forme «binaire» et indexées à l'aide d'un gestionnaire de base de données. Le gestionnaire d'arrière plan proposé par défaut est `mdb`. Il s'agit d'une variante récente du gestionnaire *Berkeley DB transactional backend*.

Q76. Comment identifier le nom de l'annuaire fourni par défaut avec le paquet `slapd` ?

Rechercher la clé `olcSuffix` dans la configuration de l'annuaire.

```
# ldapsearch -LLL -Y EXTERNAL -H ldapi:/// -b "cn=config" olcSuffix | grep ^olcSuffix
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
olcSuffix: dc=nodomain
```

Q77. Quels sont les *schemas* actifs avec la configuration courante du paquet `slapd` ?

Rechercher la clé `olcSchemaConfig` dans la configuration de l'annuaire.

```
# ldapsearch -LLL -Y EXTERNAL -H ldapi:/// -b "cn=config" olcSchemaConfig | grep ^cn
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
cn: config
cn: module{0}
cn: schema
cn: {0}core
cn: {1}cosine
cn: {2}nis
cn: {3}inetorgperson
```

Q78. Où sont stockées les bases définies par défaut lors de l'installation du paquet `slapd` ?

Rechercher la clé `olcDbDirectory` dans la configuration de l'annuaire.

```
# ldapsearch -LLL -Y EXTERNAL -H ldapi:/// -b "cn=config" olcDbDirectory |\
 grep ^olcDbDirectory
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
olcDbDirectory: /var/lib/ldap
```

C'est dans le répertoire `/var/lib/ldap` que sont stockées les fichiers des bases *Berkeley DB*.

```
# ls -lAh /var/lib/ldap/
total 68K
-rw------- 1 openldap openldap  64K sept. 12 17:21 data.mdb
-rw------- 1 openldap openldap 8,0K sept. 12 17:21 lock.mdb
```

3.3.3. Réinitialisation de la base de l'annuaire LDAP

L'installation du paquet `slapd` implique l'installation d'un annuaire minimal avec une base associée. Ce mode opératoire est nécessaire, ne serait-ce que pour accéder à la configuration du service et tester la validité de l'installation. Après avoir traité les questions ci-dessus, on sait que l'installation est fonctionnelle. On peut donc passer à l'initialisation de notre propre annuaire.

> **Note**
>
> Les manipulations proposées dans cette section permettent de reprendre à zéro la configuration d'un annuaire LDAP. Il peut être utile de revenir à cette étape en cas de «doute» sur l'intégrité de l'annuaire lors du traitement des questions des sections suivantes.

Q79. Comment arrêter le service LDAP ?

Utiliser les scripts fournis avec le gestionnaire de lancement des processus système.

Chaque processus système dispose d'un script de gestion de son lancement, arrêt (et|ou) redémarrage. Avec le gestionnaire systemd, il faut faire une recherche dans la liste des services. Une fois le service identifié, on l'arrête avec la commande **systemctl**.

```
# systemctl list-units | grep slapd
slapd.service       loaded active running   LSB: OpenLDAP standalone server

# systemctl stop slapd
```

Q80. Quels sont les éléments à supprimer pour pouvoir installer une nouvelle configuration et une nouvelle base LDAP ?

Utiliser le résultat de la question sur la localisation des bases et la documentation fournie avec le paquet `slapd`.

À partir des réponses aux questions ci-dessus, on sait que c'est le répertoire `/var/lib/ldap/` qui contient les bases. La lecture du fichier de documentation du paquet avec la commande **`# zless /usr/share/doc/slapd/README.Debian.gz`** indique que les fichiers de configuration sont situés dans le répertoire `/etc/ldap/slapd.d/`.

On supprime donc tous ces fichiers et répertoires.

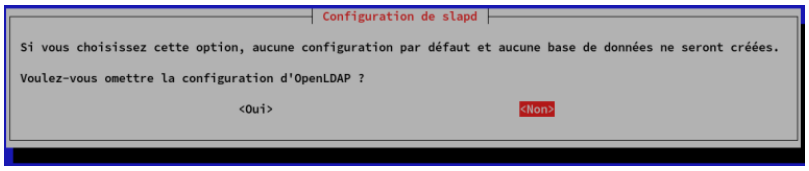
```
# rm /var/lib/ldap/*
# rm -rf /etc/ldap/slapd.d
```

Q81. Comment reprendre à zéro la configuration du paquet `slapd` ?

Utiliser l'outil du gestionnaire de paquets *Debian GNU/Linux* qui permet la modification des paramètres de configuration d'un service à l'aide de menus `debconf`.

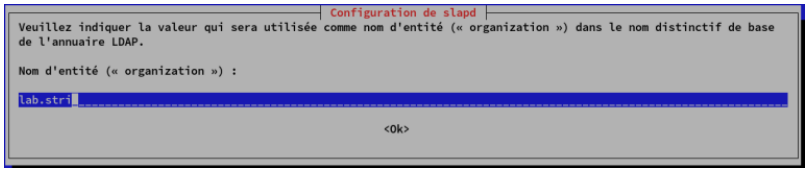
C'est la commande **dpkg-reconfigure** qui sert à réviser les paramètres de configuration d'un paquet. Voici une copie des écrans proposés avec le paquet `slapd`.

```
# dpkg-reconfigure slapd
No configuration file was found for slapd at /etc/ldap/slapd.conf. ... (warning).
  Creating initial configuration... done.
  Creating LDAP directory... done.
Starting OpenLDAP: slapd.
```

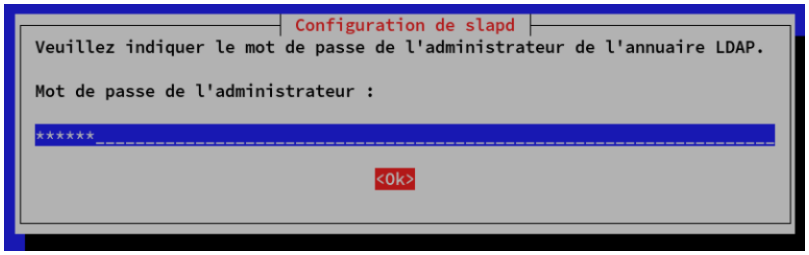


Copie d'écran dpkg-reconfigure slapd - vue complète

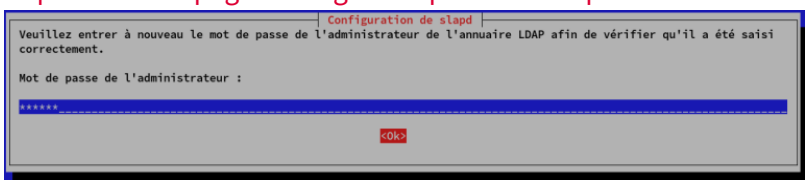
Copie d'écran dpkg-reconfigure slapd - vue complète



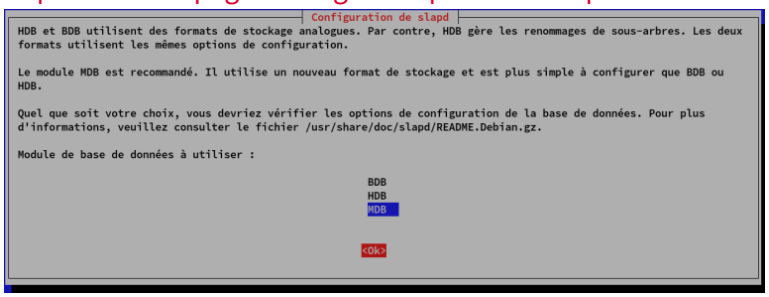
Copie d'écran dpkg-reconfigure slapd - vue complète



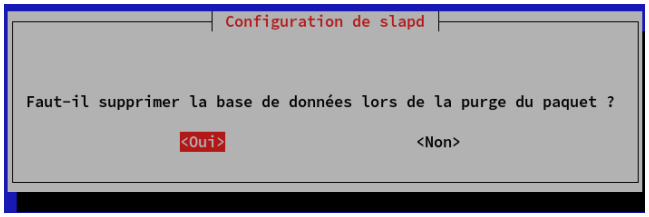
Copie d'écran dpkg-reconfigure slapd - vue complète



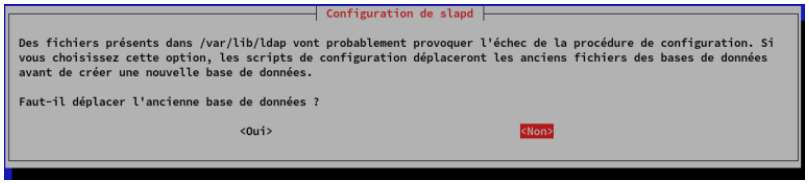
Copie d'écran dpkg-reconfigure slapd - vue complète



Copie d'écran dpkg-reconfigure slapd - vue complète



Copie d'écran dpkg-reconfigure slapd - vue complète



Copie d'écran dpkg-reconfigure slapd - vue complète

Q82. Comment valider la nouvelle configuration du paquet `slapd` ?

Reprendre la question sur le nom distinctif de l'annuaire.

```
# ldapsearch -LLL -Y EXTERNAL -H ldapi:/// -b "cn=config" olcSuffix | grep ^olcSuffix
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
olcSuffix: dc=lab,dc=stri
```

3.3.4. Composition d'un nouvel annuaire LDAP

Une fois que les fichiers de configuration et de base de données du nouvel annuaire sont en place, on peut passer à l'ajout de nouvelles entrées dans cet annuaire. Comme le fil conducteur de cette série de travaux pratiques est la gestion d'une base de comptes utilisateurs, on doit ajouter les objets suivants.

- Deux unités organisationnelles : `people` et `groups`.
- Quatre compte utilisateurs : papa et maman Skywalker ainsi que leurs deux enfants

Toutes les manipulations sur les objets de l'annuaire utilisent un format de fichier texte particulier baptisé LDIF pour *LDAP Data Interchange Format*. C'est un format de représentation des données contenues dans un annuaire particulièrement utile pour les opérations de sauvegarde et de restauration en volume.

Du point de vue formatage, chaque enregistrement doit être séparé du suivant par une ligne vide et chaque attribut d'un enregistrement apparaît sur une ligne sous la forme «nomAttribut: valeur».

Q83. Comment visualiser la liste des entrées contenues dans l'annuaire LDAP ?

Utiliser les pages de manuels de la commande **ldapsearch** et rechercher les informations sur les méthodes d'authentification, la désignation de la base dans laquelle on effectue la recherche et le nom distinctif utilisé pour se connecter à l'annuaire.

La commande **ldapsearch** propose plusieurs modes d'authentification qui influent sur la liste des attributs affichés pour une même entrée. Dans notre exemple, ce sont les mots de passes qui peuvent ne pas apparaître ou apparaître sous différentes formes.

- L'option `-x` évite le recours à la méthode SASL pour l'authentification.

```
# ldapsearch -LLL -x -H ldap:/// -b "dc=lab,dc=stri" -D cn=admin,dc=lab,dc=stri -W
Enter LDAP Password:
dn: dc=lab,dc=stri
objectClass: top
objectClass: dcObject
objectClass: organization
o: lab.stri
dc: lab

dn: cn=admin,dc=lab,dc=stri
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
userPassword:: e1NTSEF9Ykd1QVJVWi82UWt1WXhpd1QvS0ZVUHM5dkFpNVdwVU4=
```

- L'option `-Y EXTERNAL` utilise la méthode SASL du même nom.

```
# ldapsearch -LLL -Y EXTERNAL -H ldapi:/// -b "dc=lab,dc=stri" \
 -D cn=admin,dc=lab,dc=stri -W
Enter LDAP Password:
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
dn: dc=lab,dc=stri
objectClass: top
objectClass: dcObject
objectClass: organization
o: lab.stri
dc: lab

dn: cn=admin,dc=lab,dc=stri
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
```

- L'option `-LLL` désactive l'affichage des commentaires et de la version LDIF utilisée dans la réponse.
- L'option `-b` désigne le point de départ de la recherche.
- L'option `-D` désigne le nom distinctif de connexion à l'annuaire.
- L'option `-W` provoque l'affichage de l'invite de demande du mot passe correspondant au nom distinctif utilisé.

Q84. Comment activer la journalisation des manipulations sur les entrées de l'annuaire LDAP ?

Rechercher l'entrée relative au niveau de journalisation dans le DIT et modifier sa valeur de façon à obtenir un état dans les journaux système à chaque opération sur l'annuaire.

La modification de l'entrée du DIT doit se faire à l'aide d'un fichier LDIF approprié.

L'entrée à rechercher dans le DIT est baptisée `olcLogLevel`.

```
# ldapsearch -LLL -Y EXTERNAL -H ldapi:/// -b "cn=config" olcLogLevel |\
    grep ^olcLogLevel
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
olcLogLevel: none
```

on se propose de modifier la valeur `none` par `stats` de façon à journaliser les connexions, les opérations et les résultats. Voici une copie du fichier LDIF permettant de réaliser cette modification.

```
# cat setolcLogLevel2stats.ldif
# Set olcLogLevel 2 stats
dn: cn=config
changetype: modify
replace: olcLogLevel
olcLogLevel: stats
```

On applique ce changement de valeur avec la commande **ldapmodify** puis on vérifie que l'attribut a bien reçu le paramètre.

```
# ldapmodify -Y EXTERNAL -H ldapi:/// -f setolcLogLevel2stats.ldif
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
modifying entry "cn=config"

# ldapsearch -LLL -Y EXTERNAL -H ldapi:/// -b "cn=config" olcLogLevel |\
 grep ^olcLogLevel
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
olcLogLevel: stats
```

Enfin, on relève les traces de la dernière opération dans les journaux système.

```
# grep -5 olcLogLevel /var/log/syslog
slapd[4867]: conn=1004 op=0 BIND dn="" method=163
slapd[4867]: conn=1004 op=0 BIND authcid="gidNumber=0+uidNumber=0,\
 cn=peercred,cn=external,cn=auth" \
 authzid="gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth"
slapd[4867]: conn=1004 op=0 BIND dn="gidNumber=0+uidNumber=0,\
 cn=peercred,cn=external,cn=auth" \
 mech=EXTERNAL sasl_ssf=0 ssf=71
slapd[4867]: conn=1004 op=0 RESULT tag=97 err=0 text=
slapd[4867]: conn=1004 op=1 SRCH base="cn=config" scope=2 deref=0 \
 filter="(objectClass=*)"
slapd[4867]: conn=1004 op=1 SRCH attr=olcLogLevel
slapd[4867]: conn=1004 op=1 SEARCH RESULT tag=101 err=0 nentries=11 text=
slapd[4867]: conn=1004 op=2 UNBIND
slapd[4867]: conn=1004 fd=14 closed
```

Note

Dans le contexte des travaux pratiques, le nombre d'entrées de l'annuaire reste très limité et la journalisation n'a pas d'impact mesurable sur les performances du système. Dans un contexte d'exploitation réelle avec un annuaire comprenant au moins une dizaine de milliers d'entrées, la situation est très différente et il faut limiter au maximum le recours à la journalisation des transactions sur l'annuaire.

Pour ramener la valeur de l'attribut `olcLogLevel` à `none`, il suffit de créer un fichier LDIF avec la directive correspondante.

```
# Set olcLogLevel 2 none
dn: cn=config
changetype: modify
replace: olcLogLevel
olcLogLevel: none
```

Q85. Quelle est la syntaxe du fichier LDIF qui permet d'ajouter les deux unités organisationnelles (*organisational unit*) ?

Rechercher un tutoriel LDIF en ligne donnant un exemple de fichier LDIF avec une ou plusieurs entrées `ou:`.

Voici un exemple de fichier LDIF contenant les déclarations des deux unités organisationnelles à ajouter.

```
# cat ou.ldif
dn: ou=people,dc=lab,dc=stri
objectClass: organizationalUnit
ou: people

dn: ou=groups,dc=lab,dc=stri
objectClass: organizationalUnit
ou: groups
```

Q86. Quelle est la commande à utiliser pour ajouter une ou plusieurs entrées dans l'annuaire ?

Rechercher dans la liste des programmes fournis avec le paquet des outils LDAP.

C'est la commande **ldapadd** qui est utile dans notre contexte. On l'utilise en mode d'authentification simple avec le fichier LDIF ci-dessus pour compléter l'annuaire.

```
# ldapadd -cxWD cn=admin,dc=lab,dc=stri -f ou.ldif
Enter LDAP Password:
adding new entry "ou=people,dc=lab,dc=stri"

adding new entry "ou=groups,dc=lab,dc=stri"
```

On vérifie ensuite que les deux nouvelles entrées sont bien présentes dans l'annuaire.

```
# ldapsearch -LLL -x -H ldap:/// -b "dc=lab,dc=stri" -D cn=admin,dc=lab,dc=stri -W
Enter LDAP Password:
dn: dc=lab,dc=stri
objectClass: top
objectClass: dcObject
objectClass: organization
o: lab.stri
dc: lab

dn: cn=admin,dc=lab,dc=stri
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
userPassword:: e1NTSEF9Ykd1QVJVWi82UWt1WXhpd1QvS0ZVUHM5dkFpNVdwVU4=

dn: ou=people,dc=lab,dc=stri
objectClass: organizationalUnit
ou: people

dn: ou=groups,dc=lab,dc=stri
objectClass: organizationalUnit
ou: groups
```

Q87. Quelle est la commande à utiliser pour saisir manuellement un mot de passe et obtenir la chaîne chiffrée correspondante ?

Rechercher dans la liste des programmes fournis avec les paquets de la distribution puis consulter les pages de manuels correspondantes.

En effectuant une recherche par mot clé dans les pages de manuels du système, on peut identifier l'outil recherché.

```
# man -k passwd | grep -i ldap
ldappasswd (1)       - change the password of an LDAP entry
slappasswd (8)       - OpenLDAP password utility
```

On utilise la commande **slappasswd** pour générer une chaîne chiffrée que l'on insère dans le fichier LDIF des comptes utilisateurs.

```
# slappasswd
New password:
Re-enter new password:
{SSHA}LrPFvc6YekTGSEYiMezxYxcmE/0ZE/9L
```

Dans le contexte de ces travaux pratiques, on attribue le même mot de passe aux quatre comptes utilisateurs.

Il existe une technique simple pour la génération de mots de passe utilisateurs aléatoires. Une fois le mot de passe généré, il peut être transmis à l'utilisateur final par un «canal de confiance» et implanté dans les attributs de l'annuaire relatifs au compte utilisateur.

```
# head -c 9 /dev/urandom | base64
piupfsRIU23u
# slappasswd -v -h "{SSHA}" -s piupfsRIU23u
{SSHA}PxK09I2Oi6ZA2bPP55Eptm9JRkJeP6oV
```

Q88. Quelle est la syntaxe du fichier LDIF qui permet d'ajouter les quatre utilisateurs avec leurs attributs système : identifiants `uid/gid`, authentifiants `login/passwd`, etc ?

Rechercher un tutoriel LDIF en ligne donnant un exemple de fichier LDIF avec un exemple de description des attributs d'un compte utilisateur.

Voici un exemple de fichier LDIF contenant les déclarations des quatre comptes utilisateurs à ajouter.

```
# cat users.ldif
# Padmé Amidala
dn: uid=padme,ou=people,dc=lab,dc=stri
objectClass: inetOrgPerson
objectClass: shadowAccount
objectClass: posixAccount
cn: Padme
sn: Padmé Amidala Skywalker
uid: padme
uidNumber: 10000
gidNumber: 10000
loginShell: /bin/bash
homeDirectory: /ahome/padme
userPassword: {SSHA}LrPFvc6YekTGSEYiMezxYxcmE/0ZE/9L
gecos: Padme Amidala Skywalker

# Anakin Skywalker
dn: uid=anakin,ou=people,dc=lab,dc=stri
objectClass: inetOrgPerson
objectClass: shadowAccount
objectClass: posixAccount
cn: Anakin
sn: Anakin Skywalker
uid: anakin
uidNumber: 10001
gidNumber: 10001
loginShell: /bin/bash
homeDirectory: /ahome/anakin
userPassword: {SSHA}LrPFvc6YekTGSEYiMezxYxcmE/0ZE/9L
gecos: Anakin Skywalker

# Leia Organa Skywalker
dn: uid=leia,ou=people,dc=lab,dc=stri
objectClass: inetOrgPerson
objectClass: shadowAccount
objectClass: posixAccount
cn: Leia
sn: Leia Organa
uid: leia
uidNumber: 10002
gidNumber: 10002
loginShell: /bin/bash
homeDirectory: /ahome/leia
userPassword: {SSHA}LrPFvc6YekTGSEYiMezxYxcmE/0ZE/9L
gecos: Leia Organa Skywalker

# Luke Skywalker
dn: uid=luke,ou=people,dc=lab,dc=stri
objectClass: inetOrgPerson
objectClass: shadowAccount
objectClass: posixAccount
cn: Luke
sn: Luke Skywalker
uid: luke
uidNumber: 10003
gidNumber: 10003
loginShell: /bin/bash
homeDirectory: /ahome/luke
userPassword: {SSHA}LrPFvc6YekTGSEYiMezxYxcmE/0ZE/9L
gecos: Luke Skywalker
```

Comme dans le cas précédent, on utilise la commande **ldapadd** en mode d'authentification simple pour insérer les comptes dans l'annuaire.

```
# ldapadd -cxWD cn=admin,dc=lab,dc=stri -f users.ldif
Enter LDAP Password:
adding new entry "uid=padme,ou=people,dc=lab,dc=stri"

adding new entry "uid=anakin,ou=people,dc=lab,dc=stri"

adding new entry "uid=leia,ou=people,dc=lab,dc=stri"

adding new entry "uid=luke,ou=people,dc=lab,dc=stri"
```

Le résultat de la commande **`# ldapsearch -LLL -x -H ldap:/// -b "dc=lab,dc=stri" -D cn=admin,dc=lab,dc=stri -W`** doit faire apparaître les nouvelles entrées de l'annuaire.

3.4. Configuration de l'accès client au serveur LDAP

Dans cette section, on suppose qu'un annuaire LDAP existe et qu'il contient des comptes utilisateurs. On se propose de configurer un poste client pour qu'il obtienne de façon transparente les informations sur les comptes utilisateurs desservis par l'annuaire.

3.4.1. Interrogation à distance de l'annuaire LDAP

On reprend ici les requêtes de consultation des entrées de l'annuaire vues dans la Section 3.3.4, « Composition d'un nouvel annuaire LDAP ». Cette fois-ci les requêtes sont émises depuis un hôte réseau différent du serveur LDAP.

Q89. Quel est le paquet qui fournit, entre autres, la commande de consultation des entrées de l'annuaire ?

Interroger la base de données des paquets pour obtenir les informations demandées.

```
# aptitude install ldap-utils
```

Le paquet `ldap-utils` apparaît à la question sur la liste des paquets relatifs au service LDAP. Si on recherche les commandes présentes dans la liste des fichiers de ce paquet, on obtient les informations suivantes.

```
$ dpkg -L ldap-utils | grep bin
/usr/bin
/usr/bin/ldapmodrdn
/usr/bin/ldappasswd
/usr/bin/ldapdelete
/usr/bin/ldapsearch
/usr/bin/ldapmodify
/usr/bin/ldapexop
/usr/bin/ldapurl
/usr/bin/ldapcompare
/usr/bin/ldapwhoami
/usr/bin/ldapadd
```

Une fois ce paquet installé, il est possible d'utiliser toutes les commandes disponibles pour manipuler les enregistrements de l'annuaire.

Q90. Quelle est la syntaxe d'interrogation de l'annuaire qui permet d'obtenir tous les attributs de l'enregistrement correspondant à un utilisateur particulier ?

On reprend la commande **ldapsearch** en spécifiant un attribut `uid` particulier.

```
# ldapsearch -LLL -H ldap://192.0.2.12 \
 -b dc=lab,dc=stri -D cn=admin,dc=lab,dc=stri -W uid=padme
Enter LDAP Password:
dn: uid=padme,ou=people,dc=lab,dc=stri
objectClass: inetOrgPerson
objectClass: shadowAccount
objectClass: posixAccount
cn: Padme
sn:: UGFkbcOpIEFtaWRhbGEgU2t5d2Fsa2Vy
uid: padme
uidNumber: 10000
gidNumber: 10000
loginShell: /bin/bash
homeDirectory: /ahome/padme
userPassword:: e1NTSEF9THJQRnZjNllla1RHU0VZaU1lenhZeGNtRS8wWkUvOUw=
gecos: Padme Amidala Skywalker
```

Q91. Quelle est la syntaxe de la commande permettant de changer le mot de passe de l'utilisateur dont on an affiché les attributs à la question précédente ?

On utilise la commande **ldappasswd** fournie par le paquet `ldap-utils` comme dans le cas de la commande de recherche. Après consultation des pages de manuels, on obtient la syntaxe suivante.

```
# ldappasswd -x -H ldap://192.0.2.12 \
 -D cn=admin,dc=lab,dc=stri -W -S uid=padme,ou=people,dc=lab,dc=stri
New password:
Re-enter new password:
Enter LDAP Password:
```

En posant exactement la même requête que dans la question précédente, on peut vérifier que le mot de passe utilisateur a bien été modifié.

```
# ldapsearch -LLL -H ldap://192.0.2.12 \
 -b dc=lab,dc=stri -D cn=admin,dc=lab,dc=stri -W uid=padme
Enter LDAP Password:
dn: uid=padme,ou=people,dc=lab,dc=stri
objectClass: inetOrgPerson
objectClass: shadowAccount
objectClass: posixAccount
cn: Padme
sn:: UGFkbcOpIEFtaWRhbGEgU2t5d2Fsa2Vy
uid: padme
uidNumber: 10000
gidNumber: 10000
loginShell: /bin/bash
homeDirectory: /ahome/padme
gecos: Padme Amidala Skywalker
userPassword:: e1NTSEF9blZYUG9NTTNOdVdoZWFSOTBMZFBkY1lGZGRvSEVlTEY=
```

3.4.2. Configuration *Name Service Switch*

Les manipulations présentées ici ont pour but de rendre transparent l'accès aux attributs des comptes utilisateurs. Le mécanisme *Name Service Switch* assure un aiguillage de l'accès à ces attributs entre les fichiers locaux et les différents services réseau disponibles. Ici, l'annuaire LDAP constitue un dépôt de référence pour le stockage des attributs des comptes utilisateurs.

Q92. Quel est le nom du paquet relatif au mécanisme *Name Service Switch* permettant d'accéder aux ressources de l'annuaire LDAP ?

Rechercher dans les bases du gestionnaire de paquets un paquet dont le nom débute par la chaîne `libnss`.

La liste ci-dessous permet d'identifier le paquet `libnss-ldapd`.

```
# aptitude search ^libnss-
p   libnss-cache      - NSS module for using nsscache-generated files
p   libnss-db         - Module NSS pour utiliser des bases de données Berkeley
      comme service de noms
p   libnss-docker     - nss module for finding Docker containers
p   libnss-extrausers - nss module to have an additional passwd, shadow and
      group file
p   libnss-gw-name    - nss module that names the current gateway's IP address
p   libnss-ldap       - NSS module for using LDAP as a naming service
p   libnss-ldapd      - NSS module for using LDAP as a naming service
p   libnss-libvirt    - nss plugin providing IP add ress resolution for virtual
      machines
p   libnss-lwres      - NSS module for using bind9's lwres as a naming service
i A libnss-mdns       - module NSS pour la résolution de nom Multicast DNS
p   libnss-myhostname - nss module providing fallback resolution for the
      current hostname
p   libnss-mymachines - nss module to resolve hostnames for local container
      instances
v   libnss-pgsql1     -
p   libnss-pgsql2     - NSS module for using PostgreSQL as a naming service
p   libnss-rainbow2   - nss library for rainbow
p   libnss-resolve    - module NSS pour la résolution de nom avec
      systemd-resolved
p   libnss-securepass - NSS (Name Service Switch) module for Securepass
p   libnss-sss        - Nss library for the System Security Services Daemon
p   libnss-systemd    - nss module providing dynamic user and group name
      resolution
p   libnss-winbind    - greffons d'intégration de service de nom pour Samba
p   libnss-wrapper    - NSS wrapper library
```

Avertissement

Relativement au paquet `libnss-ldap`, le paquet `libnss-ldapd` modifie directement les paramètres des fichiers `/etc/pam.d/common-*`.

Q93. Quels sont les paquets supplémentaires qui sont ajoutés lors de l'installation des bibliothèques LDAP pour le mécanisme *Name Service Switch* ?

Utiliser les informations fournies par le gestionnaire de paquets pour chaque ajout.

Le lancement de l'installation du paquet `libnss-ldapd` donne la liste suivante.

```
# aptitude install libnss-ldapd
Les NOUVEAUX paquets suivants vont être installés :
  libnss-ldapd libpam-ldapd{a} nscd{a} nslcd{a} nslcd-utils{a}
0 paquets mis à jour, 5 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de télécharger 669 ko d'archives. Après dépaquetage,
1 286 ko seront utilisés.
Voulez-vous continuer ? [Y/n/?]
```

Plusieurs paquets supplémentaires apparaissent :

- `libpam-ldapd` fournit les fonctions PAM nécessaires à l'authentification, aux autorisations et à la gestion de session via un annuaire LDAP.
- `nscd` (*Name Service Cache Daemon*) est un démon qui gère la recherche des mots de passe, des groupes et hôtes des programmes en cours d'exécution, et met en cache le résultat pour une prochaine recherche.
- `nslcd` fournit una autre démon pour la collecte des informations sur les comptes utilisateurs depuis un serveur LDAP.
- `nslcd-utils` fournit des outils pour l'interrogation et la mise à jour des entrées d'annuaire LDAP.

Avertissement

Pour les besoins des travaux pratiques ou de la mise au point de l'authentification via LDAP, il est utile de désactiver les services de cache qui ne sont utiles qu'en exploitation avec un grand nombre d'entrées dans l'annuaire.

```
# systemctl stop nscd
```

Q94. Quel est le rôle de l'interface entre les fonctions PAM (*Pluggable Authentication Modules*) et l'annuaire LDAP ?

Par définition, PAM est un mécanisme qui permet d'intégrer différents modes d'authentification en les rendant transparents vis à vis de l'utilisateur et des logiciels qui accèdent aux ressources du système. Dans le contexte de ces travaux pratiques, il s'agit de permettre à l'utilisateur de se connecter, d'accéder au système de fichiers, de changer son mot de passe, etc sans avoir à lancer des commandes spécifiques.

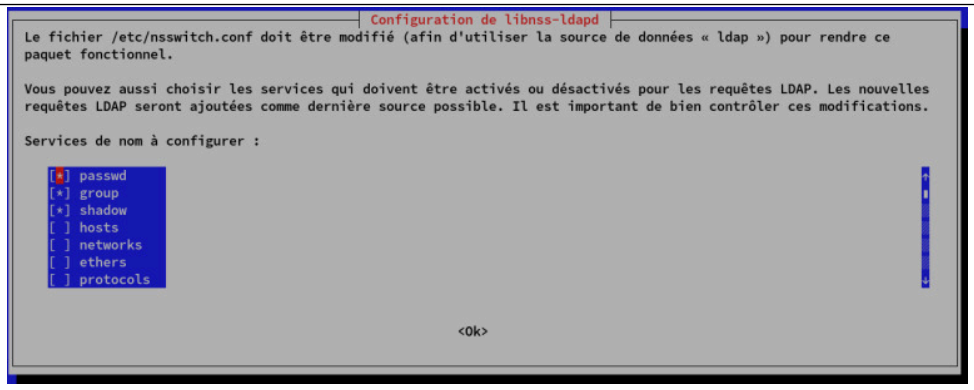
Q95. Quelles sont les principales étapes de la configuration des paquets de bibliothèques NSS et PAM ?

Lors de l'installation des principaux paquets de bibliothèques LDAP, on passe par une série de menus `debconf` qu'il faut renseigner correctement pour accéder au serveur LDAP de façon transparente.

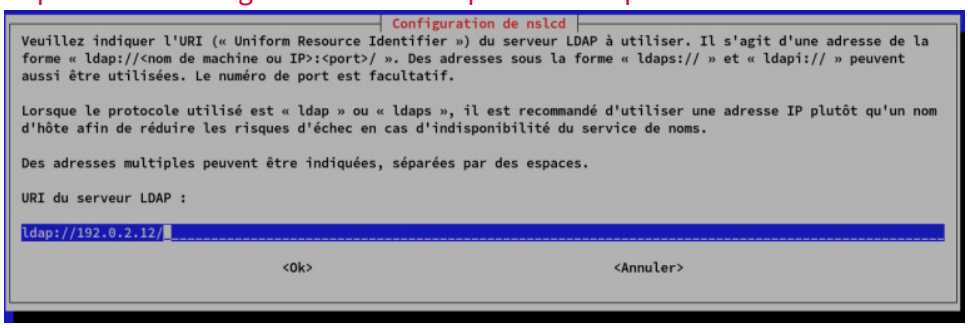
Avertissement

En cas d'erreur de saisie dans la série de menus ci-dessous, il faut reprendre la configuration de chacun des deux paquets individuellement. Classiquement, on passe par la commande **dpkg-reconfigure**.

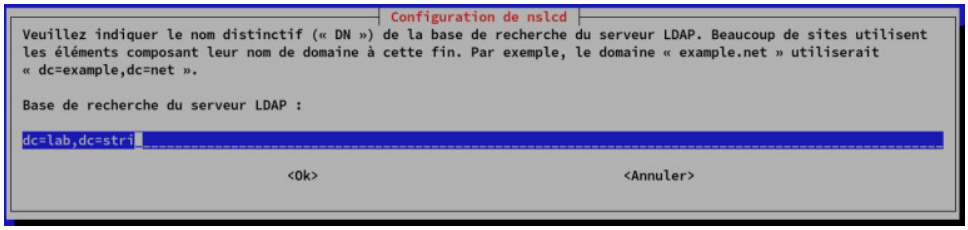
```
# dpkg-reconfigure libnss-ldapd
# dpkg-reconfigure libpam-ldapd
# dpkg-reconfigure nslcd
```



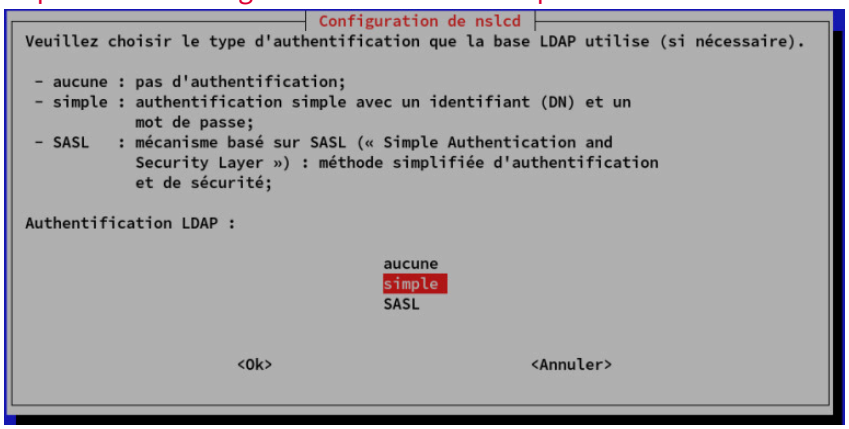
Copie d'écran configuration libnss-ldapd - vue complète



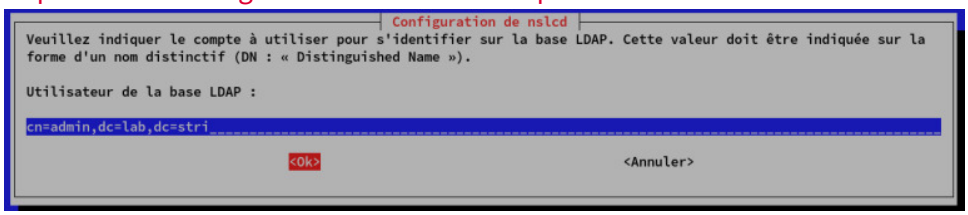
Copie d'écran configuration nslcd - vue complète



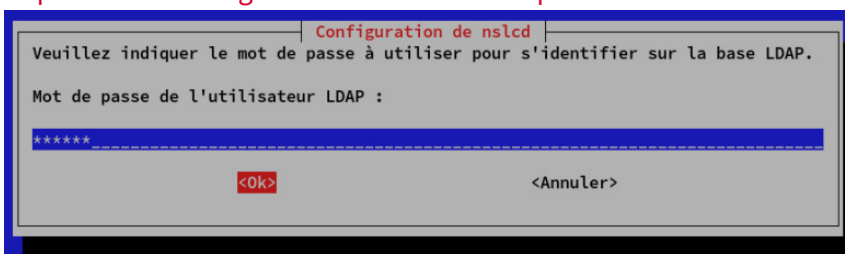
Copie d'écran configuration nslcd - vue complète



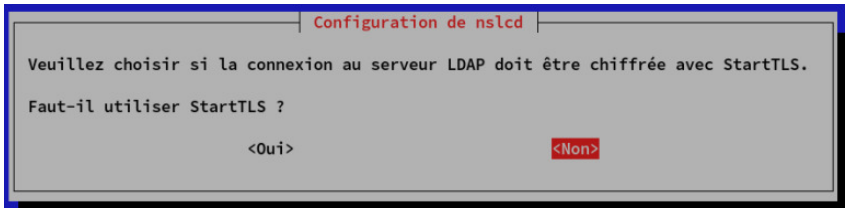
Copie d'écran configuration nslcd - vue complète



Copie d'écran configuration nslcd - vue complète



Copie d'écran configuration nslcd - vue complète



Copie d'écran configuration nslcd - vue complète

Q96. Quelles sont les modifications apportées au fichier de configuration `/etc/nsswitch.conf` pour activer l'accès aux ressources de l'annuaire LDAP ?

Lors de l'installation des paquets à l'étape précédente, le fichier `/etc/nsswitch.conf` a été modifié.

```
# grep ldap /etc/nsswitch.conf
passwd:         compat ldap
group:          compat ldap
shadow:         compat ldap
```

Q97. Comment illustrer simplement le fonctionnement du mécanisme *Name Service Switch* intégrant l'utilisation de l'annuaire LDAP ?

Rechercher la commande de récupération des entrées depuis les bases de données d'administration dans les outils fournis avec les bibliothèques standard (paquet `libc-bin`).

La commande **getent** fournie avec le paquet `libc-bin` donne la liste des entrées accessibles pour chaque catégorie du fichier de configuration. Voici un exemple pour la catégorie `passwd` qui fait apparaître les entrées de l'annuaire LDAP à la suite des comptes utilisateurs système issus des fichiers locaux.

```
# getent passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
Debian-exim:x:101:103::/var/spool/exim4:/bin/false
statd:x:102:65534::/var/lib/nfs:/bin/false
sshd:x:103:65534::/var/run/sshd:/usr/sbin/nologin
etu:x:1000:1000:Etudiant,,,:/home/etu:/bin/bash
rdnssd:x:104:65534::/var/run/rdnssd:/bin/false
systemd-timesync:x:105:108:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:106:109:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:107:110:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:108:111:systemd Bus Proxy,,,:/run/systemd:/bin/false
messagebus:x:109:113::/var/run/dbus:/bin/false
uuidd:x:100:101::/run/uuidd:/bin/false
_apt:x:110:65534::/nonexistent:/bin/false
avahi:x:111:116:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
nslcd:x:112:117:nslcd name service LDAP connection daemon,,,:/var/run/nslcd/:/usr/sbin/nologin
padme:x:10000:10000:Padme Amidala Skywalker:/ahome/padme:/bin/bash
anakin:x:10001:10001:Anakin Skywalker:/ahome/anakin:/bin/bash
leia:x:10002:10002:Leia Organa:/ahome/leia:/bin/bash
luke:x:10003:10003:Luke Skywalker:/ahome/luke:/bin/bash
```

Q98. Comment valider l'authentification d'un utilisateur déclaré dans l'annuaire LDAP ?

Choisir un service qui nécessite une authentification sur le système et qui utilise une entrée de l'annuaire LDAP.

Avertissement

Après chaque manipulation sur la configuration des paquets `libnss-ldapd` et `nslcd`, il faut impérativement relancer le démon de gestion du cache des services de noms : **`# systemctl restart nscd`**.

Sans le redémarrage de ce démon, il est fréquent que les tests de connexion échouent alors que la configuration système est correcte.

Les exemples de services nécessitant une authentification ne manquent pas. La commande **su** qui permet de changer d'identité est le plus immédiat.

```
etu@LDAP-Client:~$ su luke
Mot de passe :
luke@LDAP-Client:/home/etu$ cd
bash: cd: /ahome/luke: Aucun fichier ou dossier de ce type
```

Dans les journaux du système, on retrouve les mêmes éléments.

```
# grep luke /var/log/auth.log
LDAP-Client su[1676]: pam_unix(su:auth): authentication failure; logname=etu
  uid=1000 euid=0 tty=/dev/pts/0 ruser=etu rhost=  user=luke
LDAP-Client su[1676]: Successful su for luke by etu
LDAP-Client su[1676]: + /dev/pts/0 etu:luke
LDAP-Client su[1676]: pam_unix(su:session): session opened for user luke by
  etu(uid=1000)
LDAP-Client su[1676]: pam_unix(su:session): session closed for user luke
```

Voici un autre exemple d'accès avec SSH.

```
$ ssh anakin@fe80::b8ad:caff:fefe:65%vlan10
Warning: Permanently added 'fe80::b8ad:caff:fefe:65%vlan10' (ECDSA) to the list
of known hosts.
anakin@fe80::b8ad:caff:fefe:65%vlan10's password:
Linux LDAP-Client 4.12.0-1-686-pae #1 SMP Debian 4.12.6-1 (2017-08-12) i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Could not chdir to home directory /ahome/anakin: No such file or directory
```

Il ne manque que l'accès au système de fichiers pour que la configuration soit vraiment complète.

3.5. Accès à l'annuaire LDAP depuis un service web

Du point de vue métier, les manipulations à base de fichiers LDIF sont réservées aux traitements en volume réalisés par les administrateurs système. Les développeurs disposent de bibliothèques fournies avec les langages de programmation. Dans la plupart des cas, les développements ont pour but de fournir une interface Web.

3.5.1. Gestion de l'annuaire avec phpLDAPadmin

phpLDAPadmin est un excellent exemple de service Web utile pour les manupulations sur les objets d'un annuaire LDAP. Malheureusement, son code est basé sur le langage PHP en version 5 et n'est plus maintenu au moment de la rédaction de ces lignes. Il est donc utilisable avec la version *jessie* de la distribution Debian GNU/Linux mais plus avec les versions récentes.

Dans cette section, on commence par installer l'outil avec le serveur Web *apache2* et on configure un accès sécurisé SSL. On ajoute un groupe d'utilisateurs baptisé `StarWars` dans l'unité organisationnelle `groups` et on visualise le schéma d'une entrée du type `posixAccount`.

Q99. Quel est le paquet à installer pour mettre en place le client Web *phpLDAPadmin* ?

Rechercher le nom `phpldapadmin` dans la liste des paquets de la distribution et installer ce paquet.

Le résultat de la recherche est immédiat puisque le paquet du même nom que celui de l'outil existe. On passe donc directement à l'installation.

```
# aptitude install phpldapadmin apache2
```

Q100. Comment activer l'accès SSL au service Web ?

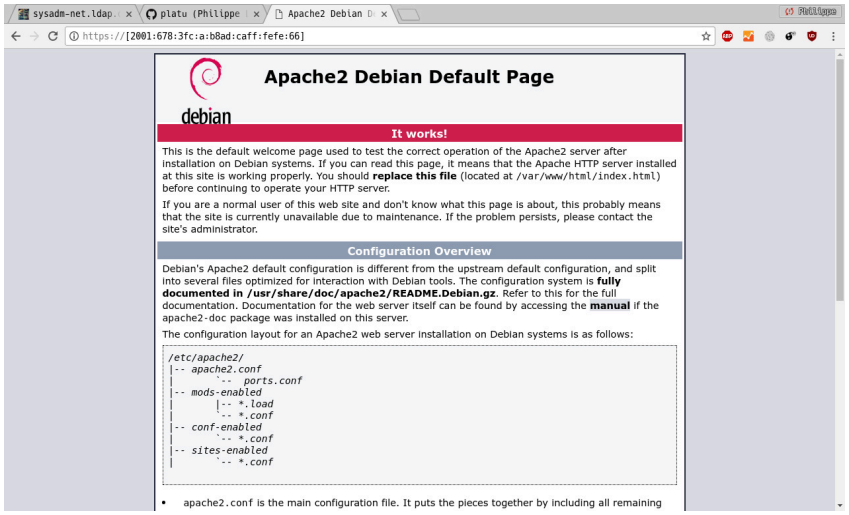
Consulter les fichiers de documentation fournis avec le paquet *apache2* et repérer les instructions d'activation du service SSL.

L'activation du module `ssl` informe directement sur le fichier à consulter. On le visualise avec la commande **`# view /usr/share/doc/apache2/README.Debian.gz`**.

```
# a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create
self-signed certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
```

```
# a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
  systemctl reload apache2
```

Après avoir accepté l'exception de sécurité relative à l'utilisation d'un certificat auto signé, on accède à une page du type suivant.



Copie d'écran accès Web SSL - vue complète

Q101. Quel est le fichier de configuration du paquet `phpldapadmin` qui contient la définition du contexte de nommage (suffixe) ?

Rechercher le répertoire contenant les fichiers de configuration du paquet. Repérer le fichier contenant la définition du suffixe de l'annuaire.

Le répertoire qui contient les éléments de configuration du paquet est nécessairement baptisé `/etc/phpldapadmin`. On recherche ensuite le fichier contenant la définition de l'entrée `dc=`.

```
# grep -rl 'dc=' /etc/phpldapadmin/*
/etc/phpldapadmin/config.php
/etc/phpldapadmin/templates/creation/sambaGroupMapping.xml
```

Q102. Quelles modifications apporter à ce fichier de configuration pour utiliser le suffixe de travaux pratiques ?

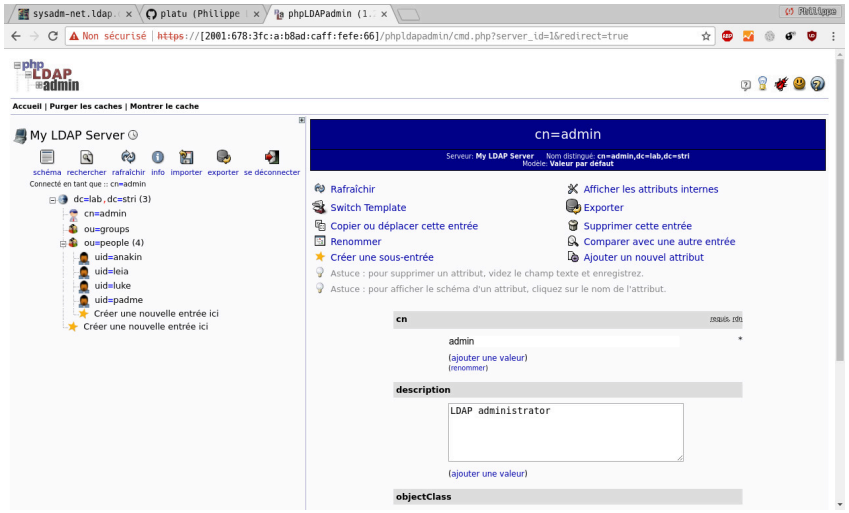
Rechercher les options de la commande **sed** qui permettent de substituer `dc=example,dc=com` dans le fichier de configuration du paquet `phpldapadmin`.

```
# sed -i 's/dc=example,dc=com/dc=lab,dc=stri/g' /etc/phpldapadmin/config.php

# sed -i 's/127\.0\.0\.1/192\.0\.2\.12/g' /etc/phpldapadmin/config.php

# systemctl reload apache2
```

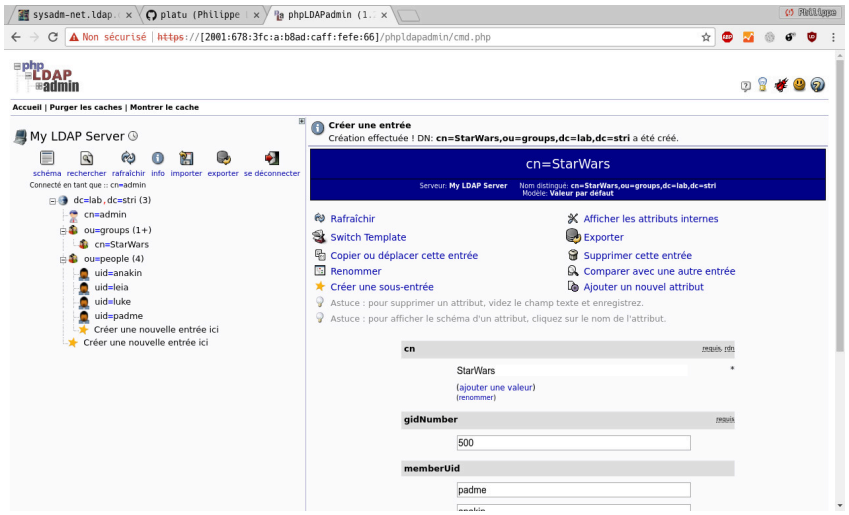
Une fois le service Web redémarré, on peut se connecter à l'annuaire avec le bon suffixe et visualiser les entrées de l'annuaire.



Copie d'écran suffixe et entrées - vue complète

Q103. Comment ajouter un groupe `StarWars` dans l'unité organisationnelle `groups` ?

On sélectionne l'unité organisationnelle `groups` et suit le lien Créer une sous-entrée pour ajouter le groupe supplémentaire.



Copie d'écran ajout d'un groupe - vue complète

Q104. Comment visualiser les attributs d'une entrée de type `posixAccount` ?

On sélectionne la catégorie schéma puis Sauter vers un objectClass: avec l'option `posixAccount`.

Copie d'écran schéma de l'entrée posixAccount - vue complète

3.6. Analyse de la configuration

Dans cette partie, on considère que les services élémentaires sont en place. Côté serveur, on dispose de l'unité organisationnelle `people` qui contient quatre entrées de comptes utilisateurs. Côté client, les outils d'accès à l'annuaire LDAP ont été installés et l'authentification sur la base des attributs des entrées de l'annuaire fonctionne.

Les manipulations suivantes sont à réaliser en concertation entre les deux postes de travaux pratiques client et serveur.

3.6.1. Indexation des entrées de l'annuaire LDAP

Comme la journalisation des transactions sur l'annuaire a été activée sur le serveur, toutes les authentifications réalisées par le client apparaissent dans ces journaux.

Q105. Quelles sont les informations relatives à l'indexation des entrées de l'annuaire qui apparaissent dans les journaux système du serveur lorsqu'une transaction est initiée par le client ?

Que peut-on constater ?

Rechercher le mot clé `index` dans le pricncipal fichier de journalisation système du serveur.

On constate que de nombreux attributs utilisés ne sont pas indexés en consultant les journaux système.

```
# grep -i index /var/log/syslog
slapd[1161]: <= bdb_equality_candidates: (uid) not indexed
slapd[1161]: <= bdb_equality_candidates: (memberUid) not indexed
slapd[1161]: <= bdb_equality_candidates: (uid) not indexed
slapd[1161]: <= bdb_equality_candidates: (memberUid) not indexed
slapd[1161]: <= bdb_equality_candidates: (uniqueMember) not indexed
slapd[1161]: <= bdb_equality_candidates: (uid) not indexed
slapd[1161]: <= bdb_equality_candidates: (uid) not indexed
```

Q106. Quelle est la syntaxe du fichier LDIF permettant d'ajouter les index identifiés dans la configuration du service LDAP ?

Rechercher un tutoriel LDIF en ligne donnant un exemple de fichier LDIF avec une ou plusieurs entrées `index`.

Voici un exemple de fichier LDIF dédié à l'ajout d'index sur les principales entrées de l'annuaire.

```
# cat olcDbIndex.ldif
dn: olcDatabase={1}hdb,cn=config
changetype: modify
add: olcDbIndex
olcDbIndex: uid pres,eq,sub
-
add: olcDbIndex
olcDbIndex: sn eq,sub
-
add: olcDbIndex
olcDbIndex: uidNumber eq
-
add: olcDbIndex
olcDbIndex: gidNumber eq
-
add: olcDbIndex
olcDbIndex: memberUid pres,eq,sub
-
add: olcDbIndex
olcDbIndex: uniqueMember pres,eq
-
add: olcDbIndex
olcDbIndex: cn pres,eq,sub
-
add: olcDbIndex
olcDbIndex: ou eq
-
add: olcDbIndex
olcDbIndex: dc eq
```

Dans cet exemple plusieurs types d'index ont été spécifiés.

- Le type `pres` est la contraction de `presence` et correspond à des requêtes comme `objectclass=inetOrgPerson` ou `attribute=mail`.
- Le type `eq` est la contraction de `equality` et correspond à des requêtes comme `sn=dupond`.
- Le type `sub` est la contraction de `substring` et correspond à des requêtes comme `sn=du*`.

Note

D'après la spécification du format LDIF, les lignes qui ne contiennent qu'un caractère '**-**' sont des séparateurs entre des modifications apportées à une même entrée tandis que les lignes vides séparent des traitements sur des entrées différentes.

Q107. Comment mettre en place les nouveaux index et valider leur présence dans la configuration du service LDAP ?

Reprendre la démarche suivie lors de l'activation des fonctions de journalisation.

On utilise la commande **ldapmodify** pour appliquer les instructions contenues dans le fichier LDIF.

```
# ldapmodify -Y EXTERNAL -H ldapi:/// -f olcDbIndex.ldif
```

On valide la présence des index dans la configuration courante avec une requête sur les index.

```
# ldapsearch -Y EXTERNAL -H ldapi:/// -b "cn=config" | grep ^olcDbIndex
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
olcDbIndex: objectClass eq
olcDbIndex: uid pres,eq,sub
olcDbIndex: sn eq,sub
olcDbIndex: memberUid pres,eq,sub
olcDbIndex: uniqueMember pres,eq
olcDbIndex: cn pres,eq,sub
olcDbIndex: ou eq
olcDbIndex: dc eq
```

Q108. Comment créer les index dans la base de données du démon `slapd` ?

Rechercher dans la liste des fichiers du paquet `slapd` la commande relative à l'indexation des entrées d'un annuaire.

```
# dpkg -L slapd | grep index
/usr/sbin/slapindex
/usr/share/man/man8/slapindex.8.gz
```

La création des fichiers de bases d'index nécessite un arrêt du service avant l'appel à la commande **slapindex**. Il est nécessaire de prendre l'identité `openldap` pour exécuter cette commande. Tous les fichiers de bases de données (*backend*) doivent avoir le même propriétaire que le processus `slapd`.

```
# /etc/init.d/slapd stop
Stopping OpenLDAP: slapd.
# su openldap -c "slapindex"
# /etc/init.d/slapd start
Starting OpenLDAP: slapd.
```

```
# ll /var/lib/ldap/*.bdb
-rw------- 1 openldap openldap 8,0K 25 avril 19:37 /var/lib/ldap/cn.bdb
-rw------- 1 openldap openldap 8,0K 25 avril 19:37 /var/lib/ldap/dc.bdb
-rw------- 1 openldap openldap 8,0K 24 avril 17:37 /var/lib/ldap/dn2id.bdb
-rw------- 1 openldap openldap  32K 24 avril 19:23 /var/lib/ldap/id2entry.bdb
-rw------- 1 openldap openldap 8,0K 25 avril 19:37 /var/lib/ldap/memberUid.bdb
-rw------- 1 openldap openldap 8,0K 24 avril 17:37 /var/lib/ldap/objectClass.bdb
-rw------- 1 openldap openldap 8,0K 25 avril 19:37 /var/lib/ldap/ou.bdb
-rw------- 1 openldap openldap 8,0K 25 avril 21:07 /var/lib/ldap/sn.bdb
-rw------- 1 openldap openldap 8,0K 25 avril 19:31 /var/lib/ldap/uid.bdb
-rw------- 1 openldap openldap 8,0K 25 avril 20:33 /var/lib/ldap/uniqueMember.bdb
```

3.6.2. Analyse réseau des transactions LDAP

Comme dans le cas du support sur l'*Introduction au système de fichiers réseau NFSv4*, la compréhension des mécanismes d'accès à un annuaire passe par l'analyse réseau. Les opérations de capture de trafic peuvent être réalisées aussi bien sur le poste client que sur le poste serveur.

Q109. Quelles sont les étapes de l'accès aux ressources de l'annuaire LDAP dans les trois cas de figure ci-dessous ?

Exécuter les instructions suivantes depuis le poste client.

- **`# getent passwd`**
- **`$ su anakin`**
- **`$ passwd`**

3.7. Documents de référence

OpenLDAP Software 2.4 Administrator's Guide

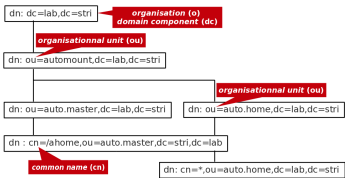
La documentation officielle : *OpenLDAP Software 2.4 Administrator's Guide* constitue le point d'entrée essentiel pour la mise en œuvre du service LDAP.

CHAPITRE 4

Association LDAP, NFSv4 et autofs

Résumé

Ce support reprend les deux précédents sur NFSv4 et LDAP en associant les services. Le système de fichiers réseau NFSv4 sert au partage des répertoires utilisateur tandis que l'annuaire LDAP sert au partage des attributs des comptes utilisateur et de la configuration du service d'automontage. Une fois que les deux services associés sont en place, les comptes utilisateurs peuvent être utilisés de façon transparente depuis n'importe quel poste client faisant appel à ces services.



4.1. Adressage IP des postes de travail

Tableau 4.1. Affectation des adresses et des réseaux IP de la salle 211

Poste 1	Poste 2	Passerelle par défaut	Organisation
christophsis	corellia	`10.31.1.145/28`	o: zone1.lan-211.stri
delaya	kasyyyk	`172.19.9.65/26`	o: zone2.lan-211.stri
korriban	kessel	`10.0.11.65/27`	o: zone3.lan-211.stri
mygeeto	nelvaan	`192.168.6.17/28`	o: zone4.lan-211.stri
rattatak	saleucami	`172.21.12.17/29`	o: zone5.lan-211.stri
taris	teth	`192.168.7.33/28`	o: zone6.lan-211.stri
utapau	yavin	`172.20.11.9/29`	o: zone7.lan-211.stri

Tableau 4.2. Affectation des adresses et des réseaux IP de la salle 213

Poste 1	Poste 2	Passerelle par défaut	Organisation
alderaan	bespin	`172.24.132.17/28`	o: zone1.lan-213.stri
centares	coruscant	`172.20.129.17/29`	o: zone2.lan-213.stri
dagobah	endor	`192.168.123.17/28`	o: zone3.lan-213.stri
felucia	geonosis	`192.168.125.49/28`	o: zone4.lan-213.stri
hoth	mustafar	`10.5.6.1/23`	o: zone5.lan-213.stri
naboo	tatooine	`172.20.136.81/28`	o: zone6.lan-213.stri

Pour chaque paire de postes de travaux pratiques, il faut attribuer les rôles de serveur et de client. Le serveur doit mettre en œuvre le service d'annuaire LDAP comprenant les propriétés des comptes utilisateurs et exporter l'arborescence du système de fichiers de ces mêmes comptes utilisateurs avec NFS. Le client doit accéder à ces ressources. Il doit permettre l'authentification auprès du serveur LDAP pour les comptes utilisateurs concernés et pouvoir monter dynamiquement à la demande le système de fichiers de ces comptes utilisateurs.

L'objectif en fin de séance de travaux pratiques est de pouvoir se connecter sur un poste client avec ses identifiants `login/password` et d'accéder à son répertoire utilisateur stocké sur le serveur de façon totalement transparente.

4.2. Mise en œuvre de l'annuaire LDAP

Cette partie reprend les étapes décrites dans le support *Introduction aux annuaires LDAP avec OpenLDAP*. Il s'agit d'installer les paquets correspondants au logiciel *OpenLDAP*, d'initialiser une base avec le bon contexte de nommage puis d'implanter les deux unités organisationnelles et les entrées des comptes utilisateurs.

Q110. Comment installer le service d'annuaire LDAP sur le poste serveur ?

Choisir les paquets à installer et valider le bon fonctionnement du service en contrôlant la liste des processus et des numéros de ports ouverts.

Reprendre les questions des parties Installation du serveur LDAP et Analyse de la configuration du service LDAP

Q111. Comment initialiser une nouvelle base et un nouveau contexte de nommage pour ce service d'annuaire ?

Réinitialiser la configuration du démon `slapd` avec le contexte de nommage défini dans la Section 4.1, « Adressage IP des postes de travail ».

Reprendre les questions de la partie Réinitialisation de la base de l'annuaire LDAP

Q112. Comment activer la journalisation des transactions sur le service d'annuaire ?

Créer un fichier LDIF qui remplace la valeur par défaut de l'attribut `olcLogLevel` par `stats`.

Reprendre la question Comment activer la journalisation des manipulations sur les entrées de l'annuaire LDAP ?

Q113. Comment implanter les deux unités organisationnelles `people` et `groups` dans le nouvel annuaire ?

Créer un fichier LDIF qui décrit la création des deux unités organisationnelles dans le bon contexte. Ajouter ces deux unités organisationnelles dans l'annuaire.

Reprendre les questions Quelle est la syntaxe du fichier LDIF qui permet d'ajouter les deux unités organisationnelles (*organisational unit*) ? et Quelle est la commande à utiliser pour ajouter une ou plusieurs entrées dans l'annuaire ?

Q114. Comment implanter les quatre comptes utilisateurs dans cet annuaire ?

Créer un fichier LDIF qui décrit la création des des quatre comptes utilisateurs dans le bon contexte avec un jeu d'attributs complet pour l'authentification et le système de fichiers. Ajouter ces comptes dans l'annuaire.

Reprendre la question Quelle est la syntaxe du fichier LDIF qui permet d'ajouter les quatre utilisateurs avec leurs attributs système ?

4.3. Mise en œuvre de l'exportation NFS

Cette partie reprend les étapes décrites dans le support *Introduction au système de fichiers réseau NFSv4*. Après avoir traité la partie commune de la configuration NFS, il s'agit d'installer le paquet correspondant au serveur NFS et de créer l'arborescence des comptes utilisateurs à exporter avec le bon contexte de nommage.

Q115. Comment installer et valider les services commun au client et au serveur NFS ?

Rechercher et installer le paquet puis contrôler la liste des processus et des numéros de port ouverts.

On reprend ici les questions de la partie Gestion des paquets NFS

- Identification du paquet à installer.

```
# aptitude search ^nfs
v   nfs-client        -
p   nfs-common        - fichiers de prise en charge NFS communs au client et au serveur
p   nfs-ganesha       - NFS server in User Space
p   nfs-ganesha-doc   - Documentation for nfs-ganesha
p   nfs-ganesha-fsal  - nfs-ganesha fsal libraries
p   nfs-kernel-server - gestion du serveur NFS du noyau
v   nfs-server        -
p   nfs4-acl-tools    - Commandline and GUI ACL utilities for the NFSv4 client
p   nfstrace          - NFS tracing/monitoring/capturing/analyzing tool
p   nfstrace-doc      - NFS tracing/monitoring/capturing/analyzing tool (documentation)
p   nfswatch          - Program to monitor NFS traffic for the console
```

- Identification des processus actifs après installation du paquet.

```
# ps aux | grep [r]pc
root      4876  0.0  0.0   6872  3180 ?        Ss   20:18   0:00 /sbin/rpcbind -f -w
```

Q116. Comment installer et configurer le paquet relatif à l'exportation d'une arborescence avec le protocole NFS ?

On reprend ici les questions de la partie Configuration du serveur NFS

- Identification du paquet à installer.

```
# aptitude search '?and(nfs, server)'
p   nfs-kernel-server - gestion du serveur NFS du noyau
v   nfs-server
```

- Création de l'arborescence d'exportationNFS.

```
# mkdir -p /home/exports/home
```

- Ajout des instructions d'exportation dans le fichier de configuration du serveur NFS : `/etc/exports`.

```
# grep -v ^# /etc/exports
/home/exports           192.0.2.0/24(rw,sync,fsid=0,crossmnt,no_subtree_check)
/home/exports/home      192.0.2.0/24(rw,sync,no_subtree_check)
```

Q117. Comment valider la configuration de l'exportation réalisée par le serveur NFS ?

On reprend la question sur la la commande qui permet d'identifier l'arborescence disponible à l'exportation.

- Côté client, on utilise la commande **showmount**suivie de l'option `-e` et de l'adresse IP du serveur à interroger.
- Côté serveur, on utilise la commande **exportfs**.

Q118. Quel est le montage local à mettre en place pour garantir la cohérence du schéma de nommage entre les postes serveur et client ?

On reprend ici la question sur la distinction entre les versions 3 et 4 du protocole NFS et sur le contexte de nommage.

- Création de la racine commune entre client et serveur.

```
# mkdir /ahome
```

- Montage local entre racine commune et arborescence exportée.

```
# mount --bind /home/exports/home /ahome
```

Q119. Comment créer automatiquement l'arborescence d'un utilisateur qui n'existe que dans l'annuaire LDAP ?

Rechercher les fonctions de création automatique de répertoire utilisateur dans la liste des paquets de la distribution.

Avertissement

Cette opération se déroule en plusieurs étapes dans la mesure où il est impossible de créer un répertoire utilisateur sur le serveur directement depuis le client.

1. Activer sur le serveur NFSv4 l'appel au module de création de répertoire utilisateur.
2. Toute nouvelle connexion depuis un client NFSv4 utilise l'arborescence utilisateur créée lors de la première connexion.

1. Sur le serveur, on ajoute le paquet oddjob-mkhomedir puis on complète le fichier commun de gestion de session : `/etc/pam.d/common-session`.

```
$ aptitude search mkhomedir
v   libpam-mkhomedir   -
p   oddjob-mkhomedir   - Oddjob helper which creates and populates home directories

<snip>
# aptitude install oddjob-mkhomedir
<snip>

# systemctl status oddjobd
<snip>
```

```
# grep -v ^# /etc/pam.d/common-session
session [default=1]                     pam_permit.so
session requisite                       pam_deny.so
session required                        pam_permit.so
session optional        pam_oddjob_mkhomedir.so skel=/etc/skel/ umask=0022
session required        pam_unix.so
session optional        pam_systemd.so
```

2. Depuis un autre hôte, on provoque la création du répertoire utilisateur sur le serveur.

```
$ ssh padme@fe80::b8ad:caff:fefe:0%vlan2
Warning: Permanently added 'fe80::b8ad:caff:fefe:0%vlan2' (ECDSA) to the list of known hosts.
padme@fe80::b8ad:caff:fefe:0%vlan2's password:
Creating home directory for padme.

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
padme@server:~$ pwd
/ahome/padme
padme@server:~$
```

Enfin, on lance une nouvelle connexion sur un client NFS de façon à tester l'automontage du répertoire utilisateur.

```
$ ssh padme@fe80::b8ad:caff:fefe:1%vlan2
Warning: Permanently added 'fe80::b8ad:caff:fefe:1%vlan2' (RSA) to the list of known hosts.
padme@fe80::b8ad:caff:fefe:1%vlan2's password:
Last login: Sun Sep 18 20:33:38 2016 from fe80::3413:59ff:fe55:23e4%eth0
padme@client1:~$ mount | grep nfs
192.0.2.12://home/padme on /ahome/padme type nfs4 <snip>
padme@client1:~$ pwd
/ahome/padme
```

4.4. Configuration de l'automontage avec le service LDAP

Le principe de l'automontage veut que le montage d'une arborescence de système de fichiers réseau se fasse automatiquement et uniquement à l'utilisation. En effet, il n'est pas nécessaire de mobiliser les ressources du protocole NFS tant qu'une arborescence n'est pas effectivement parcourue. Dans le contexte de ce support, il n'est pas nécessaire de monter l'arborescence d'un répertoire utilisateur si celui-ci n'est pas connecté sur le poste client. On optimise ainsi les ressources du système et du réseau.

Du point de vue administration système, il est essentiel que la configuration des postes clients ne soit pas remise en question à chaque évolution du serveur ou à chaque ajout de nouveau compte utilisateur. C'est ici que le service LDAP intervient. Ce service sert à publier la configuration de l'automontage en direction des clients.

Pour appliquer ces principes, cette section doit couvrir les étapes suivantes.

- Pour compléter les informations publiées par le service LDAP, il faut ajouter un schéma spécifique à la fonction d'automontage et ensuite importer le contenu d'un fichier de description LDIF contenant les paramètres de configuration à diffuser vers les clients.
- Pour que le montage des arborescences soit automatique, il faut ajouter un paquet spécifique sur les systèmes clients et désigner le service LDAP comme fournisseur de la configuration. Cette désignation se fait à l'aide du *Name Service Switch*.

La principale difficulté dans le traitement des questions suivantes vient du fait qu'il est nécessaire d'échanger des informations entre le client et le serveur.

Dans le contexte de ce support, le service LDAP et le serveur NFS sont implantés sur le même système.

Q120. Quel est le paquet de la distribution Debian GNU/Linux qui fournit le service d'automontage via LDAP ?

Rechercher le mot clé *automount* dans le champ description du catalogue des paquets disponibles.

```
# aptitude search "?description(automount)"
p   autodir                 - crée automatiquement les répertoires home et
  group pour les comptes LDAP/NIS/SQL et locaux
p   autofs                  - montage automatique pour Linux basé sur le noyau
p   autofs-hesiod           - gestion de la carte Hesiod pour autofs
p   autofs-ldap             - gestion des schémas LDAP pour autofs
p   libnss-cache            - NSS module for using nsscache-generated files
p   libunix-configfile-perl - Perl interface to various Unix configuration files
p   ltspfsd                 - Fuse based remote filesystem hooks for LTSP thin
  clients
p   nsscache                - asynchronously synchronise local NSS databases
  with remote directory services
p   vfu                     - Versatile text-based filemanager
```

Le paquet `autofs-ldap` correspond au besoin. On peut obtenir des informations supplémentaires en consultant sa description complète à l'aide de la commande **`# aptitude show autofs-ldap`**.

Q121. Sur quel type de poste ce paquet doit il être installé ?

Le service d'automontage est a exécuter sur le poste qui ne détient pas le système de fichiers dans lequel se trouvent les répertoires utilisateur.

Ce paquet doit être installé sur le poste client puisque le processus `automount` doit être exécuté sur ce même client. Son installation se fait simplement avec la commande usuelle **`# aptitude install autofs-ldap`**.

Q122. Quelles sont les informations relatives au service LDAP à transférer entre client et serveur ?

Pour publier la configuration de l'automontage via le service LDAP, il est nécessaire de disposer du schéma de définition des attributs dans l'annuaire. Ce schéma est fourni avec le paquet `autofs-ldap` et doit être transféré vers le serveur LDAP pour compléter le catalogue des objets qu'il peut contenir.

```
# dpkg -L autofs-ldap | grep schema
/etc/ldap/schema
/etc/ldap/schema/autofs.schema

# scp /etc/ldap/schema/autofs.schema etu@192.0.2.12:~
```

L'adresse IP utilisée dans la copie d'écran ci-dessus correspond au serveur LDAP et NFS.

Q123. Dans quel répertoire les informations transférées doivent elles être placées ?

Rechercher le répertoire de stockage des fichiers de schémas dans l'arborescence du serveur LDAP.

Une fois le fichier de schéma de transféré du client vers le serveur, celui-ci doit être placé dans l'arborescence du service LDAP avec les autres fichiers du même type.

```
# ls -lAh /etc/ldap/schema/autofs.schema
-rw-r--r-- 1 etu etu 830 sept. 27 10:29 /etc/ldap/schema/autofs.schema
```

Q124. Comment intégrer ces nouvelles informations d'automontage dans la configuration du service LDAP ?

L'intégration du nouveau schéma dans la configuration du serveur se fait en plusieurs étapes. Le fichier délivré avec le paquet `autofs-ldap` doit être converti en fichier LDIF avant d'être ajouté au DIT de configuration du démon `slapd`.

La conversion en fichier LDIF se fait à l'aide de la commande **slaptest** fournie avec le paquet `slapd`.

1. Création du répertoire de stockage du résultat de la conversion.

   ```
   # mkdir schema-convert
   ```

2. Création du fichier de traitement des schémas. Comme de schéma `autofs` utilise des définitions issues des schémas de rang supérieur, il est nécessaire d'inclure les autres fichiers de schémas fournis avec le paquet.

   ```
   # cat schema-convert.conf
   include /etc/ldap/schema/core.schema
   include /etc/ldap/schema/cosine.schema
   include /etc/ldap/schema/inetorgperson.schema
   include /etc/ldap/schema/autofs.schema
   ```

3. Conversion des fichiers de schémas au format LDIF.

   ```
   # slaptest -f schema-convert.conf -F schema-convert
   config file testing succeeded
   ```

4. Extraction des définitions utiles et formatage du résultat de la conversion. La commande ci-dessous élimine toutes les informations relatives à l'horodatage et à l'identification de l'utilisateur.

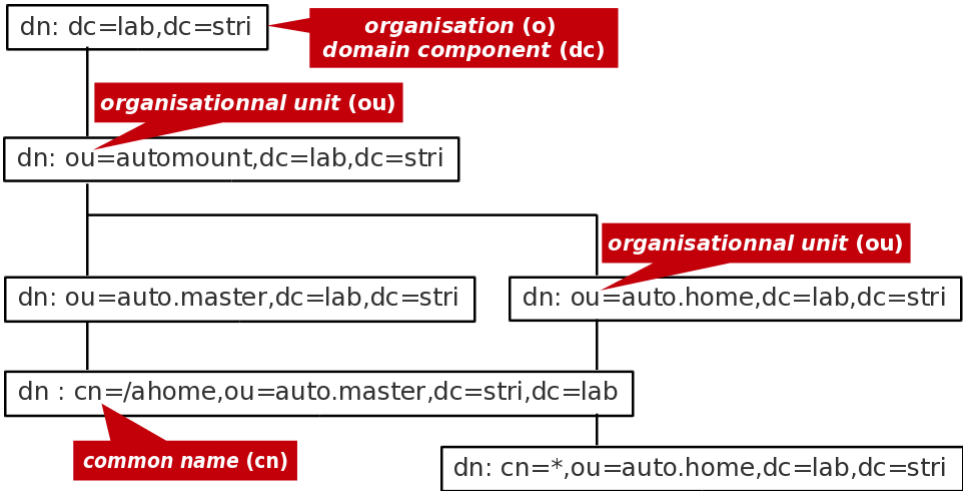
   ```
   # cat schema-convert/cn\=config/cn\=schema/cn\=\{3\}autofs.ldif | \
   egrep -v structuralObjectClass\|entryUUID\|creatorsName | \
   egrep -v createTimestamp\|entryCSN\|modifiersName\|modifyTimestamp | \
   sed 's/dn: cn={.}autofs/dn: cn=autofs,cn=schema,cn=config/g' | \
   sed 's/{.}autofs/autofs/' > autofs.ldif
   ```

5. Ajout du schéma `autofs` dans la configuration du service.

   ```
   # ldapadd -Y EXTERNAL -H ldapi:/// -f autofs.ldif
   SASL/EXTERNAL authentication started
   SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
   SASL SSF: 0
   adding new entry "cn=autofs,cn=schema,cn=config"
   ```

Q125. Quelle est la syntaxe du fichier de description LDIF contenant la configuration de l'automontage ?

Le fichier de description ci-dessus correspond à l'arborescence suivante.



Arborescence LDAP de l'automontage - vue complète

```
# cat ou-autofs.ldif
dn: ou=automount,dc=lab,dc=stri
ou: automount
objectClass: top
objectClass: organizationalUnit

dn: ou=auto.master,ou=automount,dc=lab,dc=stri
ou: auto.master
objectClass: top
objectClass: automountMap

dn: cn=/ahome,ou=auto.master,ou=automount,dc=lab,dc=stri
cn: /ahome
objectClass: top
objectClass: automount
automountInformation: ldap:ou=auto.home,ou=automount,dc=lab,dc=stri

dn: ou=auto.home,ou=automount,dc=lab,dc=stri
ou: auto.home
objectClass: top
objectClass: automountMap

dn: cn=*,ou=auto.home,ou=automount,dc=lab,dc=stri
cn: *
objectClass: top
objectClass: automount
automountInformation: -fstype=nfs4 192.0.2.12:/home/&
```

Q126. Comment intégrer ces définitions dans l'annuaire LDAP ?

Retrouver la syntaxe de la commande **ldapadd** qui permet d'insérer de nouvelles entrées dans l'annuaire.

On suit la même démarche que pour les comptes utilisateurs.

```
# ldapadd -cxWD cn=admin,dc=lab,dc=stri -f ou-autofs.ldif
Enter LDAP Password:
adding new entry "ou=automount,dc=lab,dc=stri"

adding new entry "ou=auto.master,ou=automount,dc=lab,dc=stri"

adding new entry "cn=/ahome,ou=auto.master,ou=automount,dc=lab,dc=stri"

adding new entry "ou=auto.home,ou=automount,dc=lab,dc=stri"

adding new entry "cn=*,ou=auto.home,ou=automount,dc=lab,dc=stri"
```

4.5. Accès aux ressources LDAP & NFS depuis le client

Dans cette section, on suppose que l'annuaire LDAP du poste serveur est complet et accessible. Dans un premier temps, on configure le poste client pour qu'il obtienne de façon transparente les informations sur les comptes utilisateurs desservis par l'annuaire. Dans un second temps, on complète sa configuration pour qu'il obtienne, toujours de façon transparente les informations sur le système de fichiers réseau.

Cette partie reprend les étapes décrites dans la section Configuration *Name Service Switch* du support *Introduction aux annuaires LDAP avec OpenLDAP*.

4.5.1. Configuration LDAP

Q127. Quels sont les paquets de bibliothèques LDAP relatifs au mécanisme *Name Service Switch* et au gestionnaire d'authentification PAM ?

Rechercher la liste des paquets dont le nom débute par `libnss`.

Les deux paquets utiles sont : `libnss-ldapd` et `libpam-ldap`

Q128. Quelles sont les étapes de la configuration des paquets de bibliothèques NSS et PAM ?

Lors de l'installation des deux paquets, on passe par une série de menus debconf.

Voici un récapitulatif des réponses.

Pour le paquet `libnss-ldapd`, on donne la liste des services de nom à configurer :

- `passwd`
- `group`
- `shadow`

Pour le paquet `nslcd`, on donne les paramètres pour contacter le serveur LDAP.

- URI du serveur LDAP : ldap://192.0.2.12
- Base de recherche du serveur LDAP : dc=lab,dc=stri
- Authentification LDAP : aucune
- La base LDAP demande-t-elle une identification ? non
- Faut-il utiliser StartTLS ? non

Q129. Comment valider la configuration de l'accès à l'annuaire LDAP ?

Rechercher une commande permettant d'effectuer un appel système au bibliothèques standard `libc`.

On qualifie le mécanisme *Name Service Switch* à l'aide de la commande **getent**.

```
$ getent passwd
root:x:0:0:root:/root:/bin/bash
<snip>
nslcd:x:111:117:nslcd name service LDAP connection daemon,,,:/var/run/nslcd/:/bin/false
padme:x:10000:10000:Padme Amidala Skywalker:/ahome/padme:/bin/bash
anakin:x:10001:10001:Anakin Skywalker:/ahome/anakin:/bin/bash
leia:x:10002:10002:Leia Organa:/ahome/leia:/bin/bash
luke:x:10003:10003:Luke Skywalker:/ahome/luke:/bin/bash
```

On qualifie l'authentication PAM à l'aide de la commande **su**.

```
$ su luke
Mot de passe :
:/home/etu$
```

4.5.2. Configuration NFS avec automontage

On considère que le paquet `autofs-ldap` a déjà été installé pour fournir le schéma de la partie automontage au serveur LDAP. Voir Section 4.4, « Configuration de l'automontage avec le service LDAP ».

Q130. Quelle est la modification à apporter au fichier de configuration /etc/nsswitch.conf pour que le démon automount accède aux ressources de l'annuaire LDAP ?

Il faut ajouter une directive supplémentaire qui spécifie l'ordre de recherche des informations pour le démon `automount`.

La syntaxe est la suivante.

```
# echo -e "\nautomount:      files ldap" >> /etc/nsswitch.conf
```

Q131. Quel est le fichier de configuration du service d'automontage dans lequel sont définis ses paramètres globaux ?

Rechercher le répertoire dans lequel sont placés les fichiers de paramétrage de tous les services.

Il s'agit du fichier `/etc/default/autofs`.

Q132. Quelles sont les modifications à apporter à ce fichier pour que le démon accède à l'annuaire LDAP et que la journalisation soit active ?

Il faut éditer le fichier avec les éléments suivants.

- Désigner l'unité organisationnelle qui contient les entrées de configuration de l'automontage
- Faire apparaître les évènements du service d'automontage dans les journaux système
- Désigner le serveur LDAP à contacter
- Spécifier le point d'entrée pour les recherches dans l'annuaire

```
# grep -v ^# /etc/default/autofs
MASTER_MAP_NAME="ou=auto.master,ou=automount,dc=lab,dc=stri"
TIMEOUT=300
BROWSE_MODE="no"
LOGGING="verbose"
LDAP_URI="ldap://192.0.2.12"
SEARCH_BASE="ou=automount,dc=lab,dc=stri"
```

Q133. Quelles sont les méthodes qui permettent de valider le fonctionnement du service d'automontage ?

Donner deux moyens d'acquérir l'identité d'un utilisateur ou d'une utilisatrice défini(e) dans l'annuaire LDAP uniquement.

ne pas oublier le consulter les journaux système pour observer les étapes de ces connexions utilisateur.

- Connexion SSH depuis un autre hôte
- Changement d'identité sur le même hôte avec la commande **su**
- Utilisation du gestionnaire de connexion graphique

4.6. Documents de référence

OpenLDAP Software 2.4 Administrator's Guide

Le guide *OpenLDAP Software 2.4 Administrator's Guide* est la référence essentielle sur le service LDAP.

Systèmes de fichiers réseau : NFS & CIFS

Systèmes de fichiers réseau : présentation des modes de fonctionnement des systèmes de fichiers réseau NFS & CIFS.

Linux NFS-HOWTO

Linux NFS-HOWTO : documentation historique complète sur la configuration d'un serveur et d'un client NFS jusqu'à la version 3 inclue.

Nfsv4 configuration

Nfsv4 configuration : traduction française extraite des pages du projet CITI de l'université du Michigan.

CHAPITRE 5

Introduction au service DNS

Résumé

Ce support de travaux pratiques sur le service *Domain Name System* s'appuie sur le logiciel BIND. Côté client ou *resolver*, il illustre les différents tests de fonctionnement du service à l'aide de la dig. Côté serveur, il présente l'utilisation du service suivant 3 modes : cache seulement (*cache-only*), maître (*primary*|*master*) et esclave (*secondary*|*slave*).

5.1. Architecture type de travaux pratiques

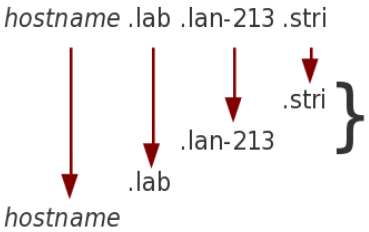
Comme indiqué dans le support *Architecture réseau des travaux pratiques*, on part d'une configuration type avec deux de postes de travail qui partagent le même domaine de diffusion. Le schéma d'une maquette utilisant deux instances de machines virtuelles et un système hôte est le suivant :

Tableau 5.1. Adressage IP des postes et attribution des zones DNS

Poste 1 : serveur primaire	Adresse IP	Poste 2 : serveur secondaire	Passerelle par défaut	zone DNS
Alderaan	`192.168.126.66/28`	Bespin	`192.168.126.65/28`	`zone1.lan-213.stri`
Centares	`172.19.115.194/26`	Coruscant	`172.19.115.193/26`	`zone2.lan-213.stri`
Dagobah	`192.168.109.2/25`	Endor	`192.168.109.1/25`	`zone3.lan-213.stri`
Felucia	`10.7.10.2/23`	Geonosis	`10.7.10.1/23`	`zone4.lan-213.stri`
Hoth	`10.5.6.2/23`	Mustafar	`10.5.6.1/23`	`zone5.lan-213.stri`
Naboo	`172.19.114.130/26`	Tatooine	`172.19.114.129/26`	`zone6.lan-213.stri`

Q134. Quelle est la représentation graphique de l'arborescence DNS corresopondant aux affectations données ci-dessus ?

À partir des information du document *Architecture réseau des travaux pratiques*, compléter la chaîne des serveurs DNS permettant la résolution des noms de domaines jusqu'à la racine.



Serveur primaire	Serveur secondaire
casper	cooper
casper	cooper
primary-srvr	secondary-srvr

Dans la suite de ce document, on utilise le nom de domaine `lab.lan-213.stri` auquel correspond le réseau `198.51.100.0/24`.

Les affectations d'adresses IP sont :

- `primary-srvr.lab.lan-213.stri` : `198.51.100.2`
- `secondary-srvr.lab.lan-213.stri` : `198.51.100.3`

5.2. Installation du service DNS cache-only

Avant d'aborder la configuration du service DNS, il faut passer par l'étape rituelle de sélection et d'installation des paquets contenant les outils logiciels de ce service.

Q135. Quels sont les paquets Debian correspondant au service DNS ?

Reprendre les différentes possibilités d'interrogation de la base de données des paquets vues lors des travaux pratiques précédents. On ne retient que les paquets relatifs à la version 9.x du logiciel BIND (*Berkeley Internet Name Domain*).

On oriente la recherche dans la base de données des paquets de la distribution vers la chaîne de caractères qui débute par `bind`.

```
# aptitude search ^bind
p   bind9      - Serveur de noms de domaines internet
p   bind9-doc  - documentation de BIND
i   bind9-host - Version de « host » intégrée avec BIND 9.X
p   bind9utils - Utilitaires pour BIND
p   bindfs     - mirrors or overlays a local directory with altered permissions
p   bindgraph  - DNS statistics RRDtool frontend for BIND9
```

Les paquets à installer à partir de la liste ci-dessus sont : `bind9` et `bind9-doc`. Une fois l'opération **`# aptitude install bind9 bind9-doc`** effectuée, on vérifie le résultat.

```
# aptitude search ~ibind9
i   bind9       - Serveur de noms de domaines internet
i   bind9-doc   - documentation de BIND
i   bind9-host  - Version de « host » intégrée avec BIND 9.X
i A bind9utils  - Utilitaires pour BIND
i A libbind9-80 - Bibliothèque partagée BIND9 utilisée par BIND
```

Q136. Quelles sont les manipulations à effectuer pour valider le fonctionnement du service DNS ?

Contrôler la liste des processus actifs sur le système, la liste des ports réseau ouverts ainsi que les journaux système.

La «singularité» du service DNS provient du nom du processus exécuté : `named`.

Liste des processus actifs

```
# ps aux | grep na[m]ed
bind      2863  0.0  1.2 170168 13224 ?        Ssl  21:05   0:00 /usr/sbin/named -u bind
```

Ports réseau ouverts

En utilisant la commande **lsof**, on obtient la liste ports ouverts en fonction du processus.

```
# lsof -i | grep na[m]ed
named   2863        bind   20u  IPv6   6733      0t0  TCP *:domain (LISTEN)
named   2863        bind   21u  IPv4   6738      0t0  TCP localhost:domain (LISTEN)
named   2863        bind   22u  IPv4   6740      0t0  TCP 198.51.100.2:domain (LISTEN)
named   2863        bind   23u  IPv4   6743      0t0  TCP localhost:953 (LISTEN)
named   2863        bind   24u  IPv6   6744      0t0  TCP localhost:953 (LISTEN)
named   2863        bind  512u  IPv6   6732      0t0  UDP *:domain
named   2863        bind  513u  IPv4   6737      0t0  UDP localhost:domain
named   2863        bind  514u  IPv4   6739      0t0  UDP 198.51.100.2:domain
```

En utilisant la commande **netstat**, on obtient les mêmes informations en partant des ports réseau ouverts.

```
# netstat -autp | grep na[m]ed
tcp   0      0 198.51.100.2:domain     *:*         LISTEN      2863/named
tcp   0      0 localhost:domain        *:*         LISTEN      2863/named
tcp   0      0 localhost:953           *:*         LISTEN      2863/named
tcp6  0      0 [::]:domain             [::]:*      LISTEN      2863/named
tcp6  0      0 localhost:953           [::]:*      LISTEN      2863/named
udp   0      0 198.51.100.2:domain     *:*                     2863/named
udp   0      0 localhost:domain        *:*                     2863/named
udp6  0      0 [::]:domain             [::]:*                  2863/named
```

Journaux systèmes

```
# grep na[m]ed /var/log/syslog
named[2863]: starting BIND 9.8.1-P1 -u bind
named[2863]: built with '--prefix=/usr' '--mandir=/usr/share/man'
'--infodir=/usr/share/info' '--sysconfdir=/etc/bind' '--localstatedir=/var'
'--enable-threads' '--enable-largefile' '--with-libtool' '--enable-shared'
'--enable-static' '--with-openssl=/usr' '--with-gssapi=/usr' '--with-gnu-ld'
'--with-geoip=/usr' '--enable-ipv6' 'CFLAGS=-fno-strict-aliasing -DDIG_SIGCHASE
-O2'
named[2863]: adjusted limit on open files from 4096 to 1048576
named[2863]: found 2 CPUs, using 2 worker threads
named[2863]: using up to 4096 sockets
named[2863]: loading configuration from '/etc/bind/named.conf'
named[2863]: reading built-in trusted keys from file '/etc/bind/bind.keys'
named[2863]: using default UDP/IPv4 port range: [1024, 65535]
named[2863]: using default UDP/IPv6 port range: [1024, 65535]
named[2863]: listening on IPv6 interfaces, port 53
named[2863]: listening on IPv4 interface lo, 127.0.0.1#53
named[2863]: listening on IPv4 interface eth0, 198.51.100.2#53
named[2863]: generating session key for dynamic DNS
named[2863]: sizing zone task pool based on 5 zones
named[2863]: using built-in root key for view _default
named[2863]: set up managed keys zone for view _default, file 'managed-keys.bind'
named[2863]: Warning: 'empty-zones-enable/disable-empty-zone' not set: disabling RFC 1918 empt
named[2863]: automatic empty zone: 254.169.IN-ADDR.ARPA
named[2863]: automatic empty zone: 2.0.192.IN-ADDR.ARPA
named[2863]: automatic empty zone: 100.51.198.IN-ADDR.ARPA
named[2863]: automatic empty zone: 113.0.203.IN-ADDR.ARPA
named[2863]: automatic empty zone: 255.255.255.255.IN-ADDR.ARPA
named[2863]: automatic empty zone: 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0
named[2863]: automatic empty zone: 1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0
named[2863]: automatic empty zone: D.F.IP6.ARPA
named[2863]: automatic empty zone: 8.E.F.IP6.ARPA
named[2863]: automatic empty zone: 9.E.F.IP6.ARPA
named[2863]: automatic empty zone: A.E.F.IP6.ARPA
named[2863]: automatic empty zone: B.E.F.IP6.ARPA
named[2863]: automatic empty zone: 8.B.D.0.1.0.0.2.IP6.ARPA
named[2863]: command channel listening on 127.0.0.1#953
named[2863]: command channel listening on ::1#953
named[2863]: zone 0.in-addr.arpa/IN: loaded serial 1
named[2863]: zone 127.in-addr.arpa/IN: loaded serial 1
named[2863]: zone 255.in-addr.arpa/IN: loaded serial 1
named[2863]: zone localhost/IN: loaded serial 2
named[2863]: managed-keys-zone ./IN: loading from master file managed-keys.bind failed: file n
named[2863]: managed-keys-zone ./IN: loaded serial 0
named[2863]: running
```

Q137. Quels sont les répertoires contenant les fichiers de configuration du service DNS ?

Identifier les répertoires à partir de la liste des fichiers du paquet bind9.

Comme pour tout service implanté sur un système GNU/Linux, les fichiers de configuration sont placés dans le répertoire `/etc/`.

```
# dpkg -L bind9 |grep etc
/etc
/etc/bind
/etc/bind/named.conf.default-zones
/etc/bind/named.conf
/etc/bind/db.empty
/etc/bind/db.255
/etc/bind/db.127
/etc/bind/db.local
/etc/bind/db.root
/etc/bind/db.0
/etc/bind/named.conf.local
/etc/bind/zones.rfc1918
/etc/bind/bind.keys
/etc/init.d
/etc/init.d/bind9
/etc/ppp
/etc/ppp/ip-down.d
/etc/ppp/ip-down.d/bind9
/etc/ppp/ip-up.d
/etc/ppp/ip-up.d/bind9
/etc/apparmor.d
/etc/apparmor.d/force-complain
/etc/apparmor.d/usr.sbin.named
/etc/apparmor.d/local
/etc/apparmor.d/local/usr.sbin.named
/etc/network
/etc/network/if-down.d
/etc/network/if-down.d/bind9
/etc/network/if-up.d
/etc/network/if-up.d/bind9
/etc/ufw
/etc/ufw/applications.d
/etc/ufw/applications.d/bind9
```

De la même façon, les données du service doivent être placées dans le répertoire `/var/`.

```
# dpkg -L bind9 |grep var
/var
/var/cache
/var/cache/bind
/var/run
```

Q138. Qu'est ce qui distingue le répertoire général de configuration du répertoire de stockage des fichiers de zone ?

Consulter la documentation *BIND 9 Administrator Reference Manual*.

C'est dans le répertoire `/var/cache/bind/` que l'on place les fichiers contenant les enregistrements ou *Resource Records* (RRs). Ces enregistrements correspondent aux zones sur lesquelles le serveur a autorité. Ce choix de répertoire fait partie des options du service. Voir l'option `directory` dans le fichier `/etc/bind/named.conf.options`.

Q139. Pourquoi l'installation du paquet bind9 correspond à un service DNS de type *cache-only* ?

Identifier la ou les zones sur lesquelles le services a autorités à partir des informations contenues dans les journaux système et les fichiers de configuration `named.conf.*`.

Consulter la section relative au service de type *cache-only* dans le document *BIND 9 Administrator Reference Manual*.

- La configuration livrée avec le paquet ne contient aucune déclaration de zone spécifique. Le fichier `/etc/bind/named.conf.local` ne contient que des commentaires.
- Le répertoire `/var/cache/bind/` est vide.
- Le service peut contacter les serveurs racine. La liste de ces serveurs est donnée dans le fichier `db.root`.

- Le service étant actif, il peut prendre en charge les requêtes et mémoriser dans son cache les résultats.

Q140. Comment appelle-t-on le logiciel client chargé d'interroger le service de noms de domaines ?

Rechercher le mot clé *resolver* dans les pages de manuels.

C'est le fichier `/etc/resolv.conf` qui sert à configurer la partie cliente du service de résolution des noms de domaines ; le *resolver*. Dans le cas des postes de travaux pratiques, la configuration initiale du *resolver* est prise en charge par le service DHCP.

Q141. Quelle est l'opération à effectuer pour le service DNS installé plus tôt soit effectivement utilisé ?

Rechercher la syntaxe à utiliser pour éditer le fichier `/etc/resolv.conf`.

Il est possible de créer un nouveau fichier simplement en désignant l'interface de boucle locale.

```
# echo nameserver 127.0.0.1 >/etc/resolv.conf
```

Vu du système sur lequel le service est exécuté, on optimise le traitement des requêtes en alimentant puis en utilisant le cache mémoire. Vu de l'Internet, on sollicite directement les serveurs racines à chaque nouvelle requête.

Q142. À quel paquet appartient la commande **dig** ? Quelle est sa fonction ?

Utiliser le gestionnaire de paquets local **dpkg**.

La commande **dig** est le «couteau suisse» qui va permettre d'effectuer tous les tests de requêtes DNS. On obtient le nom du paquet auquel elle appartient à partir d'une recherche du type :

```
# dpkg -S `which dig`
dnsutils: /usr/bin/dig
```

Le paquet `dsnutils` fait partie de l'installation de base. Il est donc présent sur tous les systèmes.

5.3. Requêtes DNS sur les différents types d'enregistrements (*Resource Records*)

Avant d'aborder la déclaration de nouvelles zones, il faut installer et valider le fonctionnement du service. La phase de validation passe par une batterie de tests d'interrogation des différents champs du service DNS.

Cette section est basée sur la commande **dig**. Les pages de manuels de cette commande doivent servir de base de réponse aux questions suivantes.

Pourquoi abandonner nslookup ?

La commande **nslookup** est la commande historique liée aux requêtes du service DNS. Le principal reproche fait à cette commande vient de ses réponses inadéquates en cas d'erreurs. Malheureusement, ce comportement non conforme a été utilisé dans de très nombreux développements de *shell scripts*. Pour ne pas entraîner des problèmes en cascade, les développeurs ont décidé d'initier un nouveau développement avec les versions 8.x puis 9.x de BIND : la commande **dig**. Comme ces travaux pratiques utilisent une version 9.x de BIND, il est logique de s'appuyer sur cette nouvelle commande **dig**.

Q143. Comment reconnaître le serveur DNS utilisé lors d'une requête avec la commande **dig** ? Comment peut on visualiser l'utilisation du cache du service DNS ?

Lire attentivement les résultats d'une exécution de la commande **dig** sur un nom de domaine quelconque.

L'utilisation du cache du serveur DNS est identifiable à partir du temps de traitement d'une requête. Ce temps de traitement apparaît dans le champ `Query time` des résultats affichés à la suite d'un appel à la commande **dig**.

Dans les deux exemples ci-dessous, le serveur interrogé est bien le service local avec l'adresse IP `127.0.0.1`. La première requête a un temps de traitement de 1301ms tandis que la seconde a un temps de traitement de 0ms. Cette seconde réponse est fournie par le cache du serveur DNS.

```
# dig www.iana.org

; <<>> DiG 9.8.1-P1 <<>> www.iana.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61419
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 4

;; QUESTION SECTION:
;www.iana.org.                  IN      A

;; ANSWER SECTION:
www.iana.org.           600     IN      CNAME   ianawww.vip.icann.org.
ianawww.vip.icann.org.  30      IN      A       192.0.32.8

;; AUTHORITY SECTION:
vip.icann.org.          3600    IN      NS      gtm1.lax.icann.org.
vip.icann.org.          3600    IN      NS      gtm1.dc.icann.org.

;; ADDITIONAL SECTION:
gtm1.dc.icann.org.      21600   IN      A       192.0.47.252
gtm1.dc.icann.org.      21600   IN      AAAA    2620:0:2830:296::252
gtm1.lax.icann.org.     21600   IN      A       192.0.32.252
gtm1.lax.icann.org.     21600   IN      AAAA    2620:0:2d0:296::252

;; Query time: 1301 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Mon Oct  8 00:28:32 2012
;; MSG SIZE  rcvd: 211
```

```
# dig www.iana.org

; <<>> DiG 9.8.1-P1 <<>> www.iana.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61419
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 4

;; QUESTION SECTION:
;www.iana.org.                  IN      A

;; ANSWER SECTION:
www.iana.org.           600     IN      CNAME   ianawww.vip.icann.org.
ianawww.vip.icann.org.  30      IN      A       192.0.32.8

;; AUTHORITY SECTION:
vip.icann.org.          3600    IN      NS      gtm1.lax.icann.org.
vip.icann.org.          3600    IN      NS      gtm1.dc.icann.org.

;; ADDITIONAL SECTION:
gtm1.dc.icann.org.      21600   IN      A       192.0.47.252
gtm1.dc.icann.org.      21600   IN      AAAA    2620:0:2830:296::252
gtm1.lax.icann.org.     21600   IN      A       192.0.32.252
gtm1.lax.icann.org.     21600   IN      AAAA    2620:0:2d0:296::252

;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Mon Oct  8 00:28:40 2012
;; MSG SIZE  rcvd: 211
```

Q144. Quelles sont les options de la commande **dig** à utiliser pour émettre des requêtes des types suivants : `NS`, `A`, `PTR`, et `MX` ? Donner un exemple de chaque type.

Les différents enregistrements ou *Resource Records* d'une zone sont accessibles à partir de requêtes individuelles. Les options de la commande **dig**, documentées dans les pages de manuels (**`man dig`**), permettent d'indiquer le type d'enregistrement demandé (RR) après le nom de domaine.

Les réponses aux requêtes suivantes apparaissent après la mention `ANSWER SECTION:`.

Requête sur un serveur de noms, NS

```
$ dig ns iana.org

; <<>> DiG 9.8.1-P1 <<>> ns iana.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 25044
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;iana.org.                      IN      NS

;; ANSWER SECTION:
iana.org.               86400   IN      NS      d.iana-servers.net.
iana.org.               86400   IN      NS      ns.icann.org.
iana.org.               86400   IN      NS      c.iana-servers.net.
iana.org.               86400   IN      NS      a.iana-servers.net.
iana.org.               86400   IN      NS      b.iana-servers.net.

;; Query time: 313 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun Oct  7 22:41:52 2012
;; MSG SIZE  rcvd: 129
```

Requête sur un nom d'hôte, A

```
$ dig a iana.org

; <<>> DiG 9.8.1-P1 <<>> a iana.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 56033
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 5, ADDITIONAL: 0

;; QUESTION SECTION:
;iana.org.                      IN      A

;; ANSWER SECTION:
iana.org.               600     IN      A       192.0.43.8

;; AUTHORITY SECTION:
iana.org.               86293   IN      NS      a.iana-servers.net.
iana.org.               86293   IN      NS      ns.icann.org.
iana.org.               86293   IN      NS      c.iana-servers.net.
iana.org.               86293   IN      NS      b.iana-servers.net.
iana.org.               86293   IN      NS      d.iana-servers.net.

;; Query time: 190 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun Oct  7 22:43:39 2012
;; MSG SIZE  rcvd: 145
```

Requête sur une adresse IP, PTR

```
$ dig -x 192.0.32.9

; <<>> DiG 9.8.1-P1 <<>> -x 192.0.32.9
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 16786
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 5, ADDITIONAL: 0

;; QUESTION SECTION:
;9.32.0.192.in-addr.arpa.       IN      PTR

;; ANSWER SECTION:
9.32.0.192.in-addr.arpa. 21600  IN      PTR     www.internic.net.

;; AUTHORITY SECTION:
32.0.192.in-addr.arpa.  86400   IN      NS      b.iana-servers.net.
32.0.192.in-addr.arpa.  86400   IN      NS      a.iana-servers.net.
32.0.192.in-addr.arpa.  86400   IN      NS      c.iana-servers.net.
32.0.192.in-addr.arpa.  86400   IN      NS      ns.icann.org.
32.0.192.in-addr.arpa.  86400   IN      NS      d.iana-servers.net.

;; Query time: 426 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun Oct  7 22:46:44 2012
;; MSG SIZE  rcvd: 174
```

Requête sur un agent de transfert de courrier électronique, MX

```
$ dig mx internic.net

; <<>> DiG 9.8.1-P1 <<>> mx internic.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 45729
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;internic.net.                  IN      MX

;; ANSWER SECTION:
internic.net.           600     IN      MX      10 pechorax.dc.icann.org.
internic.net.           600     IN      MX      10 pechorax.lax.icann.org.

;; Query time: 112 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun Oct  7 22:48:27 2012
;; MSG SIZE  rcvd: 96
```

Q145. Quelle est l'option de la commande **dig** à utiliser pour émettre des requêtes itératives ? Donner un exemple

Consulter les pages de manuels de la commande **dig** à la recherche du traçage des étapes d'une requête.

Pour émettre une requête itérative (ou non récursive), il faut utiliser l'option `+trace`.

```
$ dig +trace ns iana.org

; <<>> DiG 9.8.1-P1 <<>> +trace ns iana.org
;; global options: +cmd
.                       511837  IN      NS      i.root-servers.net.
.                       511837  IN      NS      j.root-servers.net.
.                       511837  IN      NS      c.root-servers.net.
.                       511837  IN      NS      h.root-servers.net.
.                       511837  IN      NS      a.root-servers.net.
.                       511837  IN      NS      l.root-servers.net.
.                       511837  IN      NS      d.root-servers.net.
.                       511837  IN      NS      e.root-servers.net.
.                       511837  IN      NS      g.root-servers.net.
.                       511837  IN      NS      m.root-servers.net.
.                       511837  IN      NS      f.root-servers.net.
.                       511837  IN      NS      b.root-servers.net.
.                       511837  IN      NS      k.root-servers.net.
;; Received 512 bytes from 127.0.0.1#53(127.0.0.1) in 8 ms

org.                    172800  IN      NS      a0.org.afilias-nst.info.
org.                    172800  IN      NS      c0.org.afilias-nst.info.
org.                    172800  IN      NS      d0.org.afilias-nst.org.
org.                    172800  IN      NS      b2.org.afilias-nst.org.
org.                    172800  IN      NS      b0.org.afilias-nst.org.
org.                    172800  IN      NS      a2.org.afilias-nst.info.
;; Received 428 bytes from 128.8.10.90#53(128.8.10.90) in 1705 ms

iana.org.               86400   IN      NS      a.iana-servers.net.
iana.org.               86400   IN      NS      b.iana-servers.net.
iana.org.               86400   IN      NS      c.iana-servers.net.
iana.org.               86400   IN      NS      d.iana-servers.net.
iana.org.               86400   IN      NS      ns.icann.org.
;; Received 173 bytes from 2001:500:48::1#53(2001:500:48::1) in 1101 ms

iana.org.               86400   IN      NS      c.iana-servers.net.
iana.org.               86400   IN      NS      a.iana-servers.net.
iana.org.               86400   IN      NS      d.iana-servers.net.
iana.org.               86400   IN      NS      b.iana-servers.net.
iana.org.               86400   IN      NS      ns.icann.org.
;; Received 129 bytes from 199.43.132.53#53(199.43.132.53) in 18 ms
```

Note

Après tous ces exemples de requêtes, on voit clairement que le fonctionnement par défaut du logiciel BIND est récursif. Cette prise en charge «ouverte» des requêtes peut poser quelques soucis de sécurité. Si il est légitime de prendre complètement en charge les interrogations DNS émises par les hôtes du réseau administré de façon à alimenter le cache et optimiser le fonctionnement du service, il n'en va pas de même pour les hôtes du réseau public. Il est donc important de configurer le service en conséquence. Les contrôles d'accès qui permettent de ne satisfaire que les requêtes émises par les hôtes appartenant aux «réseaux de confiance» sont présentées dans la Section 5.8, « Sécurisation de premier niveau ».

Q146. Quelle est la syntaxe de la commande **dig** à utiliser pour interroger la classe *CHAOS* ? Donner deux exemples de requêtes sur les champs `version.bind` et `authors.bind`.

Consulter les pages de manuels de la commande **dig** à la recherche des définitions de classes.

Tous les exemples de requêtes donnés ci-avant utilisent la classe Internet (IN) de façon implicite. Pour interroger un type de la classe CHAOS, il est nécessaire d'indiquer cette classe dans la commande d'interrogation du service DNS. Voici deux exemples de requêtes sur les deux types les plus souvent recherchés : la version du logiciel et la liste de ses auteurs.

```
$ dig @localhost. version.bind txt chaos +novc

; <<>> DiG 9.8.1-P1 <<>> @localhost. version.bind txt chaos +novc
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39711
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;version.bind.                  CH      TXT

;; ANSWER SECTION:
version.bind.           0       CH      TXT     "9.8.1-P1"

;; AUTHORITY SECTION:
version.bind.           0       CH      NS      version.bind.

;; Query time: 0 msec
;; SERVER: ::1#53(::1)
;; WHEN: Sun Oct  7 23:01:44 2012
;; MSG SIZE  rcvd: 65
```

```
$ dig @localhost. authors.bind txt chaos +novc

; <<>> DiG 9.8.1-P1 <<>> @localhost. authors.bind txt chaos +novc
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 36899
;; flags: qr aa rd; QUERY: 1, ANSWER: 15, AUTHORITY: 1, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;authors.bind.                  CH      TXT

;; ANSWER SECTION:
authors.bind.           0       CH      TXT     "Matt Nelson"
authors.bind.           0       CH      TXT     "Jeremy C. Reed"
authors.bind.           0       CH      TXT     "Michael Sawyer"
authors.bind.           0       CH      TXT     "Brian Wellington"
authors.bind.           0       CH      TXT     "Mark Andrews"
authors.bind.           0       CH      TXT     "James Brister"
authors.bind.           0       CH      TXT     "Ben Cottrell"
authors.bind.           0       CH      TXT     "Michael Graff"
authors.bind.           0       CH      TXT     "Andreas Gustafsson"
authors.bind.           0       CH      TXT     "Bob Halley"
authors.bind.           0       CH      TXT     "Evan Hunt"
authors.bind.           0       CH      TXT     "JINMEI Tatuya"
authors.bind.           0       CH      TXT     "David Lawrence"
authors.bind.           0       CH      TXT     "Danny Mayer"
authors.bind.           0       CH      TXT     "Damien Neil"

;; AUTHORITY SECTION:
authors.bind.           0       CH      NS      authors.bind.

;; Query time: 0 msec
;; SERVER: ::1#53(::1)
;; WHEN: Sun Oct  7 23:03:43 2012
;; MSG SIZE  rcvd: 430
```

5.4. Validation ou dépannage d'une configuration

Les sections précédentes sur les types de requêtes fournissent déjà quelques éléments sur la validation ou le dépannage du service DNS.

- Le temps de réponse à une requête (*Query time:*) renseigne sur l'utilisation ou non du cache mémoire.
- En cas de panne, une requête itérative permet d'identifier le point de rupture dans la chaîne de résolution des noms.

Il reste deux options particulièrement utiles à la mise au point d'une configuration correcte.

Il est possible de désigner explicitement le serveur DNS qui doit prendre en charge la requête à l'aide de son adresse IP. Cette opération est très utile pour vérifier qu'un serveur primaire répond correctement aux demandes sur les enregistrements qu'il détient. Dans le contexte de la sécurisation du service, cette même opération sert à contrôler qu'un serveur ne répond qu'au requêtes qu'il est sensé traiter. Voici deux exemples utilisant respectivement la désignation du serveur interrogé par son adresse IP et la requête directe de transfert de zone.

Pour vérifier que le service DNS de la zone `nic.fr` fournit l'adresse du serveur Web ayant le nom `www.nic.fr`, on peut procéder comme suit.

- On identifie un serveur de nom pour la zone.

```
$ dig ns nic.fr

; <<>> DiG 9.8.1-P1 <<>> ns nic.fr
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 23937
;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 11

;; QUESTION SECTION:
;nic.fr.                                IN      NS

;; ANSWER SECTION:
nic.fr.                 176789  IN      NS      ns1.ext.nic.fr.
nic.fr.                 176789  IN      NS      ns3.nic.fr.
nic.fr.                 176789  IN      NS      ns1.nic.fr.
nic.fr.                 176789  IN      NS      ns4.ext.nic.fr.
nic.fr.                 176789  IN      NS      ns2.nic.fr.
nic.fr.                 176789  IN      NS      ns6.ext.nic.fr.

;; ADDITIONAL SECTION:
ns1.ext.nic.fr.         176789  IN      A       193.51.208.13
ns1.nic.fr.             176789  IN      A       192.134.4.1
ns1.nic.fr.             176789  IN      AAAA    2001:660:3003:2::4:1
ns2.nic.fr.             176789  IN      A       192.93.0.4
ns2.nic.fr.             176789  IN      AAAA    2001:660:3005:1::1:2
ns3.nic.fr.             176789  IN      A       192.134.0.49
ns3.nic.fr.             176789  IN      AAAA    2001:660:3006:1::1:1
ns4.ext.nic.fr.         176789  IN      A       193.0.9.4
ns4.ext.nic.fr.         176789  IN      AAAA    2001:67c:e0::4
ns6.ext.nic.fr.         176789  IN      A       130.59.138.49
ns6.ext.nic.fr.         176789  IN      AAAA    2001:620:0:1b:5054:ff:fe74:8780

;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun Oct  7 23:09:40 2012
;; MSG SIZE  rcvd: 372
```

- On interroge directement le serveur primaire de la zone.

```
$ dig @ns1.nic.fr www.nic.fr

; <<>> DiG 9.8.1-P1 <<>> @ns1.nic.fr www.nic.fr
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 33946
;; flags: qr aa rd; QUERY: 1, ANSWER: 2, AUTHORITY: 6, ADDITIONAL: 11
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;www.nic.fr.                    IN      A

;; ANSWER SECTION:
www.nic.fr.             172800  IN      CNAME   web.nic.fr.
web.nic.fr.             172800  IN      A       192.134.4.20

;; AUTHORITY SECTION:
nic.fr.                 172800  IN      NS      ns3.nic.fr.
nic.fr.                 172800  IN      NS      ns6.ext.nic.fr.
nic.fr.                 172800  IN      NS      ns4.ext.nic.fr.
nic.fr.                 172800  IN      NS      ns1.nic.fr.
nic.fr.                 172800  IN      NS      ns1.ext.nic.fr.
nic.fr.                 172800  IN      NS      ns2.nic.fr.

;; ADDITIONAL SECTION:
ns1.ext.nic.fr.         172800  IN      A       193.51.208.13
ns1.nic.fr.             172800  IN      A       192.134.4.1
ns1.nic.fr.             172800  IN      AAAA    2001:660:3003:2::4:1
ns2.nic.fr.             172800  IN      A       192.93.0.4
ns2.nic.fr.             172800  IN      AAAA    2001:660:3005:1::1:2
ns3.nic.fr.             172800  IN      A       192.134.0.49
ns3.nic.fr.             172800  IN      AAAA    2001:660:3006:1::1:1
ns4.ext.nic.fr.         172800  IN      A       193.0.9.4
ns4.ext.nic.fr.         172800  IN      AAAA    2001:67c:e0::4
ns6.ext.nic.fr.         172800  IN      A       130.59.138.49
ns6.ext.nic.fr.         172800  IN      AAAA    2001:620:0:1b:5054:ff:fe74:8780

;; Query time: 40 msec
;; SERVER: 2001:660:3003:2::4:1#53(2001:660:3003:2::4:1)
;; WHEN: Sun Oct  7 23:11:33 2012
;; MSG SIZE  rcvd: 410
```

On voit apparaître une indication selon laquelle le serveur interrogé ne prendra pas en charge les requêtes récursives pour le client utilisé. C'est tout à fait normal dans la mesure où ces tests de requêtes ne sont pas effectués depuis un poste client appartenant au domaine `nic.fr`.

Pour autant, on obtient bien la réponse à la requête posée puisque l'enregistrement demandé appartient bien à la zone sur laquelle le serveur a autorité.

- On interroge directement le même serveur avec une requête portant sur une autre zone.

```
$ dig @ns1.nic.fr www.phrack.org

; <<>> DiG 9.8.1-P1 <<>> @ns1.nic.fr www.phrack.org
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: REFUSED, id: 16990
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;www.phrack.org.                IN      A

;; Query time: 39 msec
;; SERVER: 2001:660:3003:2::4:1#53(2001:660:3003:2::4:1)
;; WHEN: Sun Oct  7 23:14:58 2012
;; MSG SIZE  rcvd: 32
```

Cette fois-ci la requête est refusée. Le serveur primaire ne veut pas prendre en charge la requête posée. C'est encore tout à fait normal dans la mesure le client n'appartient pas aux réseaux de la zone `nic.fr`.

- Certains services sont très «ouverts» et acceptent de prendre en charge les requêtes de n'importe quel client. La même requête posée à un de ces services est traitée normalement.

```
$ dig @dns1.gaoland.net www.phrack.org

; <<>> DiG 9.8.1-P1 <<>> @dns1.gaoland.net www.phrack.org
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 19478
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;www.phrack.org.                        IN      A

;; ANSWER SECTION:
www.phrack.org.         86400   IN      A       120.138.19.103

;; AUTHORITY SECTION:
phrack.org.             86400   IN      NS      ns1.register-it.net.
phrack.org.             86400   IN      NS      ns2.register-it.net.

;; ADDITIONAL SECTION:
ns1.register-it.net.    86395   IN      A       83.246.76.254
ns2.register-it.net.    86395   IN      A       83.246.77.10

;; Query time: 48 msec
;; SERVER: 212.94.162.1#53(212.94.162.1)
;; WHEN: Sun Oct  7 23:17:38 2012
;; MSG SIZE  rcvd: 145
```

  Sous toute réserve, il semble bien que le fait de répondre aux requêtes de n'importe quel client ne corresponde pas aux bonnes pratiques sur la configuration du service DNS de nos jours.

  Dans le cadre de ces travaux pratiques, on veillera donc à n'autoriser les requêtes récursives qu'aux clients appartenant aux réseaux définis dans le plan d'adressage IP de l'énoncé.

La requête directe de transfert de zone permet de valider les autorisations d'échanges entre le serveur primaire et les autres serveurs ayant autorité sur la même zone.

Dans l'exemple de requête ci-dessous on interroge le serveur primaire à partir du serveur secondaire.

```
$ dig @172.16.80.1 axfr lan-213.stri

; <<>> DiG 9.8.1-P1 <<>> @172.16.80.1 axfr lan-213.stri
; (1 server found)
;; global options: +cmd
lan-213.stri.           86400   IN      SOA     casper.infra.stri. root.casper.infra.stri. 2012090701 2
lan-213.stri.           86400   IN      MX      0 mail.stri.
lan-213.stri.           86400   IN      NS      casper.infra.stri.
lan-213.stri.           86400   IN      NS      cooper.lan-213.stri.
alderaan.lan-213.stri.  86400   IN      A       172.16.80.10
amethyste.lan-213.stri. 86400   IN      A       172.16.80.5
anison.lan-213.stri.    86400   IN      A       172.16.80.23
bespin.lan-213.stri.    86400   IN      A       172.16.80.11
casper.lan-213.stri.    86400   IN      A       172.16.80.2
centares.lan-213.stri.  86400   IN      A       172.16.80.12
cooper.lan-213.stri.    86400   IN      A       172.16.80.1
coruscant.lan-213.stri. 86400   IN      A       172.16.80.13
dagobah.lan-213.stri.   86400   IN      A       172.16.80.14
endor.lan-213.stri.     86400   IN      A       172.16.80.15
felucia.lan-213.stri.   86400   IN      A       172.16.80.16
geonosis.lan-213.stri.  86400   IN      A       172.16.80.17
hoth.lan-213.stri.      86400   IN      A       172.16.80.18
kamino.lan-213.stri.    86400   IN      A       172.16.80.19
mustafar.lan-213.stri.  86400   IN      A       172.16.80.20
naboo.lan-213.stri.     86400   IN      A       172.16.80.21
perle.lan-213.stri.     86400   IN      A       172.16.80.6
tatooine.lan-213.stri.  86400   IN      A       172.16.80.22
topaze.lan-213.stri.    86400   IN      A       172.16.80.4
lan-213.stri.           86400   IN      SOA     casper.infra.stri. root.casper.infra.stri. 2012090701 2
;; Query time: 1 msec
;; SERVER: 172.16.80.1#53(172.16.80.1)
;; WHEN: Sun Oct  7 23:24:57 2012
;; XFR size: 24 records (messages 1, bytes 619)
```

Pour éviter une «recensement trop facile» de l'identité des hôtes d'une zone, il est essentiel de n'autoriser ces requêtes de transfert qu'entre serveurs DNS. Cette configuration du contrôle d'accès est présentée dans la Section 5.8, « Sécurisation de premier niveau ».

5.5. Serveur primaire de la zone zone(i).lan-213.stri

Il s'agit ici de configurer un serveur maître pour une nouvelle branche ou zone de l'arborescence DNS de travaux pratiques. On part de l'installation du service *cache-only* et on complète les fichiers de configuration.

La syntaxe des fichiers de zone n'est pas facile à maîtriser au premier abord. Il est donc nécessaire de faire appel à des patrons de fichiers de configuration. Un premier jeu de ces fichiers est disponible dans la documentation *BIND 9 Administrator Reference Manual*. Un second jeu, pour une configuration sécurisée, est disponible à partir du site *Secure BIND Template*.

Le fichier `/usr/share/doc/bind9/README.Debian.gz` contient des informations importantes sur l'organisation des fichiers de configuration du service. Il faut retenir les éléments suivants :

- Les fichiers `db.*` qui contiennent les enregistrements sur les serveurs racine et l'interface de boucle locale sont fournis directement avec le paquet Debian. Ils sont donc susceptibles d'être mis à jour à chaque nouvelle version du paquet.
- Le fichier de configuration principal `named.conf` a été éclaté en trois parties.

  `named.conf`

  Déclarations d'autorité sur le `localhost` et la diffusion en résolution directe et inverse. Liste des fichiers `db.*`.

  Ce fichier *appartient* au paquet bind9 et est susceptible d'être mis à jour. Il ne faut donc pas éditer ce fichier ou y insérer des informations de définitions de zones contrôlées par le service DNS.

  `named.conf.local`

  Déclarations d'autorité sur les zones administrées par le serveur ; qu'il s'agisse d'un serveur primaire ou secondaire. Ce fichier n'est pas modifié lors d'une mise à jour du paquet Debian.

C'est donc le fichier qui doit être édité pour déclarer les zones sous le contrôle du serveur DNS.

`named.conf.options`

Paramétrage des options du service notamment du répertoire contenant les fichiers de déclaration des zones administrées `/var/cache/bind/`. Voir le *BIND 9 Administrator Reference Manual* pour obtenir la liste de ces options.

C'est le fichier qui doit être édité pour sécuriser les accès aux enregistrements des zones sous le contrôle du serveur DNS..

Q147. Quel est le fichier de configuration à éditer pour que le service DNS installé ait autorité sur la zone `zone(i).lan-213.stri` ?

Établir la correspondance entre l'organisation des fichiers de configuration du paquet Debian et les indications des documents de référence.

Le fichier `/etc/bind/named.conf.local` du nouveau serveur DNS doit être édité de façon à faire apparaître les zones directes et inverses sur lesquelles il a autorité. Une fois l'opération effectuée, on recharge la configuration du serveur et on consulte les journaux système. Voici une copie du fichier correspondant à la zone `lab.lan-213.stri`.

```
# cat /etc/bind/named.conf.local
//
// Do any local configuration here
//

zone "lab.lan-213.stri" {
        type master;
        file "lab.lan-213.stri";
        };

zone "100.51.198.in-addr.arpa" {
        type master;
        file "100.51.198";
        };

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";
```

Q148. Quel est le fichier de configuration qui désigne le répertoire de stockage des fichiers de déclaration de zone ? Quel est ce répertoire ? Quelle est la particularité de son masque de permissions ?

Établir la correspondance entre l'organisation des fichiers de configuration du paquet Debian et les indications de la documentation de référence. Repérer le propriétaire du processus `named` et relever ses caractéristiques : `uid`, `gid`, répertoire utilisateur, etc.

- C'est le fichier `named.conf.options` qui désigne le répertoire de travail du service de noms de domaines : `/var/cache/bind/`.
- On retrouve la même information au niveau des paramètres du compte utilisateur système dédié au service.

```
$ grep bind /etc/passwd
bind:x:105:107::/var/cache/bind:/bin/false
```

- Le masque de permissions donne les droits d'écriture aux membres du groupe système `bind`.

```
$ ll /var/cache/ | grep bind
drwxrwxr-x  2 root bind 4,0K oct.   7 21:05 bind
```

Q149. À l'aide de l'exemple donné dans le document *DNS HOWTO : A real domain example*, créer un fichier de déclaration de la zone directe `zone(i).lan-213.stri` dans le répertoire adéquat.

Le fichier de zone doit comprendre :

- Deux serveurs de noms : un primaire et un secondaire.

- Un *Mail Exchanger*.
- Trois hôtes avec des adresses IP différentes et quelques *Canonical Names*.

Avertissement

Pour les besoins des travaux pratiques, les temps définis dans l'enregistrement SOA ont été considérablement réduits pour caractériser l'effet des notifications et des durées de maintien en cache mémoire. Ces temps permettent aussi de propager les modifications sur les enregistrements plus rapidement en incrémentant les numéros de version.

En respectant les options de configuration du paquet Debian, on créé le fichier `lab.lan-213.stri` dans le répertoire `/var/cache/bind/`.

```
# cat /var/cache/bind/lab.lan-213.stri
$TTL 60
@       IN      SOA     lab.lan-213.stri. postmaster.lab.lan-213.stri. (
                        2012100801      ; serial, yearmonthdayserial#
                        20              ; refresh, seconds
                        5               ; retry, seconds
                        420             ; expire, seconds
                        60 )            ; minimum, seconds
                        NS      primary-srvr.lab.lan-213.stri.
                        NS      secondary-srvr.lab.lan-213.stri.
                        MX      10 smtp.lab.lan-213.stri.  ; Primary Mail Exchanger
                        TXT     "DNS training Lab"

rtr             A       198.51.100.1
primary-srvr    A       198.51.100.2
ns1             CNAME   primary-srvr.lab.lan-213.stri.
secondary-srvr  A       198.51.100.3
ns2             CNAME   secondary.lab.lan-213.stri.
file-srvr       A       198.51.100.5
nfs             CNAME   file-srvr.lab.lan-213.stri.
ldap            CNAME   file-srvr.lab.lan-213.stri.
smtp            A       198.51.100.10
```

Q150. À l'aide de l'exemple donné dans le document *DNS HOWTO : A real domain example*, créer un fichier de déclaration de la zone inverse `100.51.198` dans le répertoire adéquat.

Les enregistrements (RRs) utilisés pour la résolution inverse des adresses IP doivent correspondre exactement aux décarations de la zone directe.

```
# cat /var/cache/bind/100.51.198
$TTL 60
@       IN      SOA     lab.lan-213.stri. postmaster.lab.lan-213.stri. (
                        2012100801      ; serial, yearmonthdayserial#
                        20              ; refresh, seconds
                        5               ; retry, seconds
                        420             ; expire, seconds
                        60 )            ; minimum, seconds
                        NS      primary-srvr.lab.lan-213.stri.
                        NS      secondary-srvr.lab.lan-213.stri.

1               PTR     rtr.lab.lan-213.stri.
2               PTR     primary-srvr.lab.lan-213.stri.
3               PTR     secondary-srvr.lab.lan-213.stri.
;
5               PTR     file-srvr.lab.lan-213.stri.
10              PTR     smtp.lab.lan-213.stri.
```

Q151. Quel est l'outil à utiliser pour valider la syntaxe des déclarations d'enregistrement avant d'activer la nouvelle zone ?

Consulter la liste des outils fournis avec les paquets relatifs au logiciel bind9.

Le paquet `bind9utils` fournit plusieurs outils dont le programme named-checkzone qui permet de valider la syntaxe des fichiers de déclaration de zone.

Dans le cas des deux exemples ci-dessus, on obtient les résultats suivants.

```
# named-checkzone lab.lan-213.stri. /var/cache/bind/lab.lan-213.stri
zone lab.lan-213.stri/IN: loaded serial 2012100801
OK
```

```
# named-checkzone  100.51.198.in-addr.arpa. /var/cache/bind/100.51.198
zone 100.51.198.in-addr.arpa/IN: loaded serial 2012100801
OK
```

Q152. Comment activer les nouveaux enregistrements de zone ? Valider la prise en charge de ces enregistrements

Recharger la configuration du service DNS et consulter les journaux système correspondant

Le rechargement de la configuration du service ne se distingue pas des autres services Internet.

```
# service bind9 reload
[ ok ] Reloading domain name service...: bind9.
```

Voici un extrait de journal système.

```
# tail -100 /var/log/syslog
named[2863]: received control channel command 'reload'
named[2863]: loading configuration from '/etc/bind/named.conf'
named[2863]: reading built-in trusted keys from file '/etc/bind/bind.keys'
named[2863]: using default UDP/IPv4 port range: [1024, 65535]
named[2863]: using default UDP/IPv6 port range: [1024, 65535]
named[2863]: sizing zone task pool based on 7 zones
named[2863]: using built-in root key for view _default
named[2863]: Warning: 'empty-zones-enable/disable-empty-zone' not set: disabling RFC 1918 empty z
named[2863]: reloading configuration succeeded
named[2863]: reloading zones succeeded
named[2863]: zone 100.51.198.in-addr.arpa/IN: zone serial (2012100801) unchanged. zone may fail t
named[2863]: zone 100.51.198.in-addr.arpa/IN: loaded serial 2012100801
named[2863]: zone 100.51.198.in-addr.arpa/IN: sending notifies (serial 2012100801)
named[2863]: zone lab.lan-213.stri/IN: zone serial (2012100801) unchanged. zone may fail to trans
named[2863]: zone lab.lan-213.stri/IN: loaded serial 2012100801
named[2863]: zone lab.lan-213.stri/IN: sending notifies (serial 2012100801)
```

Q153. Comment valide-t-on individuellement chacun des enregistrements déclarés ?

Reprendre la séquence des tests donnés dans la Section 5.3, « Requêtes DNS sur les différents types d'enregistrements (*Resource Records*) ».

5.6. Configuration du serveur secondaire de la zone zone(i).lan-213.stri

Il s'agit ici de configurer un serveur secondaire pour la zone de l'arborescence DNS de travaux pratiques mise en place dans la section précédente. Comme dans le cas du serveur primaire, on part de l'installation du service *cache-only* fournie par le paquet Debian et on complète les fichiers de configuration.

Pour distinguer un serveur primaire d'un serveur secondaire, il faut savoir que le serveur primaire détient effectivement les fichiers de déclaration des enregistrements. Un serveur secondaire, en revanche, obtient les copies des déclarations des enregistrements par transfert réseau.

Q154. Quel est le fichier de configuration à éditer pour que le service DNS installé ait autorité sur la zone `zone(i).lan-213.stri` ?

Établir la correspondance entre l'organisation des fichiers de configuration du paquet Debian et les indications des documents de référence.

Le fichier `/etc/bind/named.conf.local` du serveur DNS secondaire doit être édité. Bien sûr, les noms de zone doivent correspondre à ceux du serveur primaire. Voici une copie de la configuration globale du service.

```
# cat /etc/bind/named.conf.local
//
// Do any local configuration here
//

zone "lab.lan-213.stri." {
        type slave;
        masters {
                198.51.100.2;
        };
        file "backup.lab.lan-213.stri";
};

zone "100.51.198.in-addr.arpa" {
        type slave;
        masters {
                198.51.100.2;
        };
        file "backup.100.51.198";
};

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";
```

Q155. Quel est le fichier de configuration qui désigne le répertoire de stockage des fichiers de déclaration de zone ? Quel est ce répertoire ? Quelle est la particularité de son masque de permissions ?

Établir la correspondance entre l'organisation des fichiers de configuration du paquet Debian et les indications de la documentation de référence. Repérer le propriétaire du processus `named` et relever ses caractéristiques : `uid`, `gid`, répertoire utilisateur, etc.

- C'est le fichier `named.conf.options` qui désigne le répertoire de travail du service de noms de domaines : `/var/cache/bind/`.

- On retrouve la même information au niveau des paramètres du compte utilisateur système dédié au service.

```
$ grep bind /etc/passwd
bind:x:105:107::/var/cache/bind:/bin/false
```

- Le masque de permissions donne les droits d'écriture aux membres du groupe système `bind`.

```
$ ll /var/cache/ | grep bind
drwxrwxr-x  2 root bind 4,0K oct.   7 21:05 bind
```

Q156. Quel est l'outil à utiliser pour valider la syntaxe des déclarations d'enregistrement avant d'activer la nouvelle zone ?

Consulter la liste des outils fournis avec les paquets relatifs au logiciel bind9.

Le paquet `bind9utils` fournit plusieurs outils dont le programme named-checkconf qui permet de valider la syntaxe des fichiers de configuration.

Dans le cas de notre exemple, on obtient les résultats suivants.

```
# named-checkconf -p /etc/bind/named.conf
options {
        directory "/var/cache/bind";
        listen-on-v6 {
                "any";
        };
        auth-nxdomain no;
        dnssec-validation auto;
};
zone "lab.lan-213.stri." {
        type slave;
        file "backup.lab.lan-213.stri";
        masters {
                198.51.100.2 ;
        };
};
zone "100.51.198.in-addr.arpa" {
        type slave;
        file "backup.100.51.198";
        masters {
                198.51.100.2 ;
        };
};
zone "." {
        type hint;
        file "/etc/bind/db.root";
};
zone "localhost" {
        type master;
        file "/etc/bind/db.local";
};
zone "127.in-addr.arpa" {
        type master;
        file "/etc/bind/db.127";
};
zone "0.in-addr.arpa" {
        type master;
        file "/etc/bind/db.0";
};
zone "255.in-addr.arpa" {
        type master;
        file "/etc/bind/db.255";
};
```

Q157. Comment les enregistrements (*Resource Records*) d'un serveur DNS secondaire sont ils obtenus ? Quel est le type de requête qui permet de valider la disponibilité des nouveaux enregistrements ?

Rechercher dans la liste des requêtes utilisables avec la commande **dig**.

Les enregistrements d'un serveur secondaire sont obtenus par *transfert réseau*.

Le type d'une requête de transfert de zone est : AXFR. Voici deux exemples de résultats.

```
# dig axfr @198.51.100.2 lab.lan-213.stri

; <<>> DiG 9.8.1-P1 <<>> axfr @198.51.100.2 lab.lan-213.stri
; (1 server found)
;; global options: +cmd
lab.lan-213.stri.       60      IN      SOA     lab.lan-213.stri. postmaster.lab.lan-213.stri. 20
lab.lan-213.stri.       60      IN      NS      primary-srvr.lab.lan-213.stri.
lab.lan-213.stri.       60      IN      NS      secondary-srvr.lab.lan-213.stri.
lab.lan-213.stri.       60      IN      MX      10 smtp.lab.lan-213.stri.
lab.lan-213.stri.       60      IN      TXT     "DNS training Lab"
file-srvr.lab.lan-213.stri. 60  IN      A       198.51.100.5
ldap.lab.lan-213.stri.  60      IN      CNAME   file-srvr.lab.lan-213.stri.
nfs.lab.lan-213.stri.   60      IN      CNAME   file-srvr.lab.lan-213.stri.
ns1.lab.lan-213.stri.   60      IN      CNAME   primary-srvr.lab.lan-213.stri.
ns2.lab.lan-213.stri.   60      IN      CNAME   secondary.lab.lan-213.stri.
primary-srvr.lab.lan-213.stri. 60 IN    A       198.51.100.2
rtr.lab.lan-213.stri.   60      IN      A       198.51.100.1
secondary-srvr.lab.lan-213.stri. 60 IN  A       198.51.100.3
smtp.lab.lan-213.stri.  60      IN      A       198.51.100.10
lab.lan-213.stri.       60      IN      SOA     lab.lan-213.stri. postmaster.lab.lan-213.stri. 20
;; Query time: 1 msec
;; SERVER: 198.51.100.2#53(198.51.100.2)
;; WHEN: Mon Oct  8 17:20:52 2012
;; XFR size: 15 records (messages 1, bytes 400)
```

```
# dig axfr @198.51.100.2 100.51.198.in-addr.arpa.

; <<>> DiG 9.8.1-P1 <<>> axfr @198.51.100.2 100.51.198.in-addr.arpa.
; (1 server found)
;; global options: +cmd
100.51.198.in-addr.arpa. 60     IN      SOA     lab.lan-213.stri. postmaster.lab.lan-213.stri. 20
100.51.198.in-addr.arpa. 60     IN      NS      primary-srvr.lab.lan-213.stri.
100.51.198.in-addr.arpa. 60     IN      NS      secondary-srvr.lab.lan-213.stri.
1.100.51.198.in-addr.arpa. 60   IN      PTR     rtr.lab.lan-213.stri.
10.100.51.198.in-addr.arpa. 60  IN      PTR     smtp.lab.lan-213.stri.
2.100.51.198.in-addr.arpa. 60   IN      PTR     primary-srvr.lab.lan-213.stri.
3.100.51.198.in-addr.arpa. 60   IN      PTR     secondary-srvr.lab.lan-213.stri.
5.100.51.198.in-addr.arpa. 60   IN      PTR     file-srvr.lab.lan-213.stri.
100.51.198.in-addr.arpa. 60     IN      SOA     lab.lan-213.stri. postmaster.lab.lan-213.stri. 20
;; Query time: 1 msec
;; SERVER: 198.51.100.2#53(198.51.100.2)
;; WHEN: Mon Oct  8 17:22:34 2012
;; XFR size: 9 records (messages 1, bytes 296)
```

Q158. Comment activer les nouveaux enregistrements de zone ? Valider la prise en charge de ces enregistrements

Recharger la configuration du service DNS et consulter les journaux système correspondant

Le rechargement de la configuration du service ne se distingue pas des autres services Internet.

```
# service bind9 reload
[ ok ] Reloading domain name service...: bind9.
```

Voici un extrait de journal système.

```
# tail -100 /var/log/syslog
named[3188]: received control channel command 'reload'
named[3188]: loading configuration from '/etc/bind/named.conf'
named[3188]: reading built-in trusted keys from file '/etc/bind/bind.keys'
named[3188]: using default UDP/IPv4 port range: [1024, 65535]
named[3188]: using default UDP/IPv6 port range: [1024, 65535]
named[3188]: sizing zone task pool based on 7 zones
named[3188]: using built-in root key for view _default
named[3188]: Warning: 'empty-zones-enable/disable-empty-zone' not set: disabling RFC 1918 empty z
named[3188]: zone 100.51.198.IN-ADDR.ARPA/IN: (master) removed
named[3188]: reloading configuration succeeded
named[3188]: reloading zones succeeded
named[3188]: zone 100.51.198.in-addr.arpa/IN: Transfer started.
named[3188]: transfer of '100.51.198.in-addr.arpa/IN' from 198.51.100.2#53: connected using 198.5
named[3188]: zone 100.51.198.in-addr.arpa/IN: transferred serial 2012100801
named[3188]: transfer of '100.51.198.in-addr.arpa/IN' from 198.51.100.2#53: \
  Transfer completed: 1 messages, 9 records, 296 bytes, 0.001 secs (296000 bytes/sec)
named[3188]: zone 100.51.198.in-addr.arpa/IN: sending notifies (serial 2012100801)
named[3188]: zone lab.lan-213.stri/IN: Transfer started.
named[3188]: transfer of 'lab.lan-213.stri/IN' from 198.51.100.2#53: connected using 198.51.100.3
named[3188]: zone lab.lan-213.stri/IN: transferred serial 2012100801
named[3188]: transfer of 'lab.lan-213.stri/IN' from 198.51.100.2#53: \
  Transfer completed: 1 messages, 15 records, 400 bytes, 0.001 secs (400000 bytes/sec)
named[3188]: zone lab.lan-213.stri/IN: sending notifies (serial 2012100801)
```

Lors d'une modification de la liste des enregistrements, il est important d'incrémenter correctement le numéro de série de façon à notifier l'ensemble des serveurs ayant autorité sur une zone. Dans l'extrait du fichier `/var/log/syslog/` du serveur primaire donné ci-dessous, on voit bien apparaître ces notifications.

```
named[2863]: client 198.51.100.3#54299: transfer of 'lab.lan-213.stri/IN': AXFR started
named[2863]: client 198.51.100.3#54299: transfer of 'lab.lan-213.stri/IN': AXFR ended
named[2863]: client 198.51.100.3#57978: transfer of '100.51.198.in-addr.arpa/IN': AXFR started
named[2863]: client 198.51.100.3#57978: transfer of '100.51.198.in-addr.arpa/IN': AXFR ended
```

5.7. Délégation de la zone lab depuis le niveau lan-213.stri

5.7.1. Échange du niveau supérieur vers le niveau inférieur

Avertissement

Cette partie est complétée par l'enseignant sur le serveur DNS de travaux pratiques ayant autorité au niveau supérieur. Ce niveau supérieur correspond à un *Top Level Domain* (TLD) factice.

Le serveur maître de la zone `lan-213.stri` doit *déléguer* le domaine `lab.lan-213.stri` aux postes de travaux pratiques qui détiennent les enregistrements (RRs) du sous-domaine.

Dans le contexte de la maquette utilisée pour ce document, le système hôte doit déléguer le sous-domaine aux deux instances de machines virtuelles.

Les fichiers de configuration donnés dans cette section sont surtout utiles pour les communications inter-zones lors des travaux pratiques. En effet, pour que les services internet qui s'appuient sur la résolution des noms puissent fonctionner normalement, il est essentiel que les branches de cette arborescence DNS factice soient toutes reliées les unes aux autres.

Le fichier de configuration du service sur le système hôte comprend les éléments suiavnts.

```
zone "lab.lan-213.stri" {
        type slave;
        file "lab.lan-213.stri.bak";
        masters { 198.51.100.2; };
};

zone "100.51.198.in-addr.arpa" {
        type slave;
        file "100.51.198.bak";
        masters { 198.51.100.2; };
};
```

Avertissement

Le fonctionnement de la résolution inverse s'avère délicat lorsque l'on utilise des sous-réseaux. Dans le cas de ces travaux pratiques, il est essentiel que les déclarations de zones inverses soient *identiques* entre les différents niveaux.

Après rechargement de la configuration du service DNS sur le système hôte, les journaux système montrent que les transferts de zone se sont déroulés correctement.

```
# grep 'lab.lan-213.stri' /var/log/named/named.log
transfer of 'lab.lan-213.stri/IN/standard' from 198.51.100.2#53: \
  connected using 198.51.100.1#35001
createfetch: primary-srvr.lab.lan-213.stri A
createfetch: primary-srvr.lab.lan-213.stri AAAA
transfer of 'lab.lan-213.stri/IN/standard' from 198.51.100.2#53: \
  Transfer completed: 1 messages, 15 records, 400 bytes, 0.001 secs (400000 bytes/sec)
zone lab.lan-213.stri/IN/standard: sending notifies (serial 2012100801)

# grep '100.51.198' /var/log/named/named.log
transfer of '100.51.198.in-addr.arpa/IN/standard' from 198.51.100.2#53: \
  connected using 198.51.100.1#44547
transfer of '100.51.198.in-addr.arpa/IN/standard' from 198.51.100.2#53: \
  Transfer completed: 1 messages, 9 records, 296 bytes, 0.001 secs (296000 bytes/sec)
zone 100.51.198.in-addr.arpa/IN/standard: sending notifies (serial 2012100801)
```

On peut vérifier que les numéros de série des notifications correspondent bien aux enregistrements publiés au niveau inférieur.

5.7.2. Échange du niveau inférieur vers le niveau supérieur

Pour que les enregistrements déclarés dans les différentes zones de travaux pratiques soient visibles les uns des autres, il est nécessaire de faire appel à la notion de *forwarder*.

Q159. Est-ce que les enregistrements de l'arborescence factice sont accessibles depuis les serveurs du niveau `zone(i).lan-213.stri` ? Quelle requête faut-il utiliser pour accéder à ces enregistrements ?

Rechercher l'adresse IP correspondant au nom `cooper.lan-213.stri`.

La requête directe n'aboutit pas puisque les serveurs racines n'ont aucune connaissance de l'arborescence factice.

```
# dig cooper.lan-213.stri

; <<>> DiG 9.8.1-P1 <<>> cooper.lan-213.stri
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 61354
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;cooper.lan-213.stri.           IN      A

;; AUTHORITY SECTION:
.                       10800   IN      SOA     a.root-servers.net. nstld.verisign-grs.com. 20121

;; Query time: 184 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Mon Oct  8 23:02:29 2012
;; MSG SIZE  rcvd: 112
```

En interrogeant directement le niveau supérieur, on obtient l'information demandée.

```
# dig @198.51.100.1 cooper.lan-213.stri

; <<>> DiG 9.8.1-P1 <<>> @198.51.100.1 cooper.lan-213.stri
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 2583
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 1

;; QUESTION SECTION:
;cooper.lan-213.stri.           IN      A

;; ANSWER SECTION:
cooper.lan-213.stri.    86400   IN      A       172.16.80.1

;; AUTHORITY SECTION:
lan-213.stri.           86400   IN      NS      cooper.lan-213.stri.
lan-213.stri.           86400   IN      NS      casper.infra.stri.

;; ADDITIONAL SECTION:
casper.infra.stri.      86400   IN      A       172.16.0.2

;; Query time: 1 msec
;; SERVER: 198.51.100.1#53(198.51.100.1)
;; WHEN: Mon Oct  8 23:08:06 2012
;; MSG SIZE  rcvd: 110
```

Q160. Comment diriger toutes les requêtes du niveau `zone(i).lan-213.stri` vers le niveau `lan-213.stri` ?

Rechercher l'option `forwarder` dans le document *BIND 9 Administrator Reference Manual*.

On édite le fichier `/etc/bind/named.conf.options` de façon à compléter la section `forwarders`.

```
forwarders {
        198.51.100.1;
};
```

5.8. Sécurisation de premier niveau

L'objectif de cette section est de présenter les mécanismes de contrôle d'accès offerts par le service *Berkeley Internet Name Domain* à un niveau très modeste. On se contente ici de définir les adresses IP ou les réseaux qui sont autorisés à émettre des requêtes récursives sur le service DNS ainsi que les adresses IP ou les réseaux qui sont autorisés à émettre des requêtes de transfert de zone.

Les éléments de configuration présentés ci-après sont à appliquer sur tous les serveurs DNS quel que soit leur rôle.

On commence par la définition des listes de contrôle d'accès dans le fichier `/etc/bind/named.conf.options`. Ces listes permettent de définir des groupes d'adresses IP ou de réseaux. Ces groupes peuvent ensuite être réutilisés autant de fois que nécessaire au niveau global de la configuration du service ou bien dans les déclarations de zones.

Ici, on se limite à la définition de deux groupes.

- Le groupe `xfer` donne la liste des adresses IP à partir desquelles les opérations de transfert de zone sont possibles.
- Le groupe `trusted` donne la liste des réseaux de confiance qui sont habilités à utiliser le service.

Ces définitions se retrouvent au début du fichier de configuration global du service DNS.

```
# cat /etc/bind/named.conf.options
acl "xfer" {
        localhost;
        198.51.100.1;
 198.51.100.3;
 198.51.100.4;
 };

acl "trusted" {
 localhost;
 198.51.100.0/27;
 };

options {
 directory "/var/cache/bind";

 // If there is a firewall between you and nameservers you want
 // to talk to, you may need to fix the firewall to allow multiple
 // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

 // If your ISP provided one or more IP addresses for stable
 // nameservers, you probably want to use them as forwarders.
 // Uncomment the following block, and insert the addresses replacing
 // the all-0's placeholder.

 forwarders {
  198.51.100.1;
 };

 auth-nxdomain no;    # conform to RFC1035
 listen-on-v6 { any; };

 allow-transfer {
  none;
  };

 allow-query {
  trusted;
  };

 allow-query-cache {
  trusted;
  };
 };
```

C'est dans la section `options` que l'on trouve la première utilisation des listes de contrôle d'accès. Ce niveau est dit global puisqu'il est examiné avant les déclarations de zone qui sont effectuées dans le fichier `/etc/bind/named.conf.local`. Dans l'exemple donné ci-dessus, les opérations de transfert sont interdites au niveau global et les requêtes récursives pour des enregistrements sur lesquels le serveur n'a pas autorité ne sont autorisées que pour les réseaux de confiance.

Il faut noter que la section `forwarders` a été décomentée et configurée avec l'adresse IP du serveur de niveau supérieur dans l'arborescence DNS. Cette configuration est nécessaire pour garantir la «continuité» de l'arborescence factice de travaux pratiques. Il faut que les communications inter zones soient effectives pour mettre en œuvre les autres services internet qui s'appuient sur la résolution des noms.

On retrouve les listes de contrôle d'accès dans le fichier de déclaration de zone.

```
# cat /etc/bind/named.conf.local
//
// Do any local configuration here
//

zone "0.200.192.in-addr.arpa" {
        type master;
        file "198.51.100";

 allow-query {
  any;
  };

 allow-transfer {
  xfer;
  };
 };

zone "stri.lab" {
 type master;
 file "stri.lab";

 allow-query {
  any;
  };

 allow-transfer {
  xfer;
  };
 };

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";
```

Les choix effectués ici reviennent à autoriser sans restriction les requêtes directes et inverses sur les enregistrements de la zone `stri.lab`. Les transferts sur les mêmes enregistrements ne sont autorisés que pour les serveurs dont les adresses IP figurent dans la liste `xfer`.

Comme dans les sections précédentes, ces options de configuration sont à valider avec la suite des tests utilisant les différents types de requêtes à l'aide de la commande **dig**. À titre d'exemple, voici ce que l'on peut lire dans les journaux système lors d'une requête de transfert de zone non autorisée.

```
named[1524]: client 198.51.100.4#58025: zone transfer 'stri.lab/AXFR/IN' denied
```

Pour être plus complète, la sécurisation de la configuration devrait utiliser la notion de vue interne et externe du service de résolution des noms. Ce niveau de configuration dépasse «quelque peu» le cadre de ces travaux pratiques d'introduction. Le contenu de cette section n'est qu'une première prise de contact avec les fonctionnalités de sécurité d'un serveur DNS.

5.9. Documents de référence

BIND 9 Administrator Reference Manual

BIND 9 Administrator Reference Manual : documentation complète la plus récente sur la syntaxe de configuration du service DNS. Si le paquet `bind9-doc` est installé, ce manuel est placé dans le répertoire `/usr/share/doc/bind9-doc/arm/`.

Secure BIND Template

Secure BIND Template : patrons de fichiers de configuration d'un service DNS.

root-servers.org

root-servers.org : informations sur les serveurs racines du service de noms de domaines.

Administration système en réseau : architecture réseau

Architecture réseau des travaux pratiques : présentation de l'infrastructure des travaux pratiques.

Configuration d'une interface de réseau local

Configuration d'une interface de réseau local : tout sur la configuration des interfaces réseau ; notamment les explications sur les opérations «rituelles» de début de travaux pratiques.