Guida a Ubuntu Server

Guida a Ubuntu Server

Sommario

Benvenuti nella *Guida a Ubuntu server*. Questa guida contiene informazioni su come installare e configurare diverse applicazioni server per Ubuntu a seconda delle proprie esigenze. È una guida passo-passo, orientata ai processi per configurare e personalizzare il sistema.

Riconoscimenti e licenza

Contributors to this document are:

- Members of the *Ubuntu Documentation Project*[2]
- Members of the *Ubuntu Server Team*[3]
- Contributors to the *Ubuntu Documentation Wiki*[4]
- Other contributors can be found in the revision history of the *serverguide*[5] and *ubuntu-docs*[6] bzr branches available on Launchpad.

[1] http://creativecommons.org/licenses/by-sa/3.0/
[2] https://launchpad.net/~ubuntu-core-doc
[3] https://launchpad.net/~ubuntu-server
[4] https://help.ubuntu.com/community/
[5] https://code.launchpad.net/serverguide
[6] https://code.launchpad.net/ubuntu-docs

Indice

Lista delle tabelle

Capitolo 1. Introduzione

Benvenuti alla guida a *Ubuntu server*.

In questa guida è possibile trovare informazioni su come installare e configurare diversi applicativi server; è una guida passo-passo, orientata ai processi per configurare e personalizzare il proprio sistema.

This guide assumes you have a basic understanding of your Ubuntu system. Some installation details are covered in *Capitolo 2, Installazione [3]*, but if you need detailed instructions installing Ubuntu please refer to the *Ubuntu Installation Guide*[1].

A HTML version of the manual is available online at *the Ubuntu Documentation website*[2].

[1] https://help.ubuntu.com/12.04/installation-guide/
[2] https://help.ubuntu.com

1. Supporto

Esistono diverse forme di supporto per la Ubuntu Server Edition: supporto commerciale e dalla comunità. Il supporto commerciale è disponibile attraverso Canonical Ltd.: fornisce contratti di supporto a prezzi ragionevoli per postazione desktop o server. Per maggiori informazioni, consultare il *sito web di Canonical*[3].

Il supporto della comunità è fornito grazie all'impegno di singole persone, o di aziende, che desiderano rendere Ubuntu il migliore sistema operativo possibile. Il supporto viene erogato attraverso l'utilizzo di mailing list, canali IRC, forum, blog, wiki e altro. L'enorme quantità di informazioni disponibili può sembrare schiacciante, ma una valida interrogazione con un motore di ricerca può spesso fornire una risposta ai propri dubbi. Per maggiori informazioni, consultare la pagina *Ubuntu Support*[4].

[3] http://www.canonical.com/services/support
[4] http://www.ubuntu.com/support

Capitolo 2. Installazione

This chapter provides a quick overview of installing Ubuntu 12.04 LTS Server Edition. For more detailed instructions, please refer to the *Ubuntu Installation Guide*[1].

[1] https://help.ubuntu.com/12.04/installation-guide/

1. Preparazione dell'installazione

Questa sezione spiega i diversi aspetti da considerare prima di avviare l'installazione.

1.1. Requisiti di sistema

Ubuntu 12.04 LTS Server Edition supports three (3) major architectures: Intel x86, AMD64 and ARM. The table below lists recommended hardware specifications. Depending on your needs, you might manage with less than this. However, most users risk being frustrated if they ignore these suggestions.

Tabella 2.1. Requisiti minimi raccomandati

Tipo di installazione	**CPU**	**RAM**	**Spazio disco fisso**	
			Sistema di base	**Installazione completa**
Server	300 megahertz	128 megabyte	500 megabyte	1 gigabyte

La Server Edition fornisce una base comune per tutte le tipologie di applicazioni server: ha una progettazione minimalista in grado di fornire una piattaforma per qualsiasi servizio desiderato come servizi di file e stampa, host web, email, ecc...

The requirements for UEC are slightly different; for Front End requirements see *Sezione 3.2.1, «Front End Requirements» [325]*, and for UEC Node requirements see *Sezione 3.2.2, «Requisiti del nodo» [326]*.

1.2. Differenze tra Server e Desktop

There are a few differences between the *Ubuntu Server Edition* and the *Ubuntu Desktop Edition*. It should be noted that both editions use the same apt repositories, making it just as easy to install a *server* application on the Desktop Edition as it is on the Server Edition.

Le differenze tra le due edizioni sono la mancanza dell'ambiente X nella Server Edition, il processo di installazione e diverse opzioni per il kernel.

1.2.1. Differenze del kernel

Ubuntu version 10.10 and prior, actually had different kernels for the server and desktop editions. Ubuntu no longer has separate -server and -generic kernel flavors. These have been merged into a single -generic kernel flavor to help reduce the maintenance burden over the life of the release.

Usando una versione a 64-bit di Ubuntu su processori a 64-bit non si è limitati nello spazio di indirizzamento della memoria.

To see all kernel configuration options you can look through `/boot/config-3.2.0-server`. Also, *Linux Kernel in a Nutshell*[2] is a great resource on the options available.

1.3. Effettuare una copia di backup

- Prima di installare Ubuntu Server Edition è utile creare una copia di sicurezza di tutti i dati nel sistema. Per maggiori informazioni sulle opzioni di backup, consultare il *Capitolo 19, Backup [296]*.

  Se non è la prima volta che viene installato un sistema operativo nel computer, potrebbe essere necessario ripartizionare il disco fisso per creare spazio per l'installazione di Ubuntu.

  A ogni partizionamento del disco fisso è necessario essere preparati per eventuali perdite di dati causate da errori o da malfunzionamenti nel sistema di partizionamento. I programmi usati durante l'installazione sono sicuri e usati da molti anni, ma possono anche eseguire azioni distruttive.

[2] http://www.kroah.com/lkn/

2. Installare da CD

The basic steps to install Ubuntu Server Edition from CD are the same as those for installing any operating system from CD. Unlike the *Desktop Edition*, the *Server Edition* does not include a graphical installation program. The Server Edition uses a console menu based process instead.

- First, download and burn the appropriate ISO file from the *Ubuntu web site*[3].
- Avviare il sistema dal CD-ROM.
- At the boot prompt you will be asked to select a language.
- From the main boot menu there are some additional options to install Ubuntu Server Edition. You can install a basic Ubuntu Server, check the CD-ROM for defects, check the system's RAM, boot from first hard disk, or rescue a broken system. The rest of this section will cover the basic Ubuntu Server install.
- The installer asks for which language it should use. Afterwards, you are asked to select your location.
- Next, the installation process begins by asking for your keyboard layout. You can ask the installer to attempt auto-detecting it, or you can select it manually from a list.
- Il programma d'installazione rileva l'hardware e configura le impostazioni di rete utilizzando il servizio DHCP. Per non utilizzare questo servizio, alla schermata successiva scegliere «Indietro» e quindi scegliere l'opzione per configurare la rete manualmente.
- Vengono chiesti il nome host e il fuso orario.
- You can then choose from several options to configure the hard drive layout. Afterwards you are asked for which disk to install to. You may get confirmation prompts before rewriting the partition table or setting up LVM depending on disk layout. If you choose LVM, you will be asked for the size of the root logical volume. For advanced disk options see *Sezione 4, «Installazione avanzata» [10]*.
- Il sistema base Ubuntu viene quindi installato.
- A new user is set up; this user will have *root* access through the sudo utility.
- After the user settings have been completed, you will be asked to encrypt your `home` directory.
- Il passo successivo nel processo di installazione consiste nel decidere come aggiornare il sistema. Sono disponibili tre opzioni:
  - *Nessun aggiornamento automatico*: richiede che un amministratore si colleghi al computer e installi manualmente gli aggiornamenti.
  - *Install security updates automatically*: this will install the unattended-upgrades package, which will install security updates without the intervention of an administrator. For more details see *Sezione 5, «Aggiornamenti automatici» [29]*.

[3] http://www.ubuntu.com/download/server/download

  - *Gestire il sistema con Landscape*: Landscape è un servizio a pagamento fornito da Canonical che consente di gestire diversi computer con Ubuntu installato. Per maggiori informazioni, consultare la *pagina web dedicata a Landscape*[4].
- Ora è possibile scegliere se installare o non installare diversi pacchetti per attività specifiche (tasks) (per maggiori informazioni, consultare *Sezione 2.1, «Pacchetti per attività specifiche» [7]*). È inoltre presente un'opzione per lanciare il programma aptitude per scegliere dei pacchetti specifici da installare. Per maggiori informazioni, consultare *Sezione 4, «Aptitude» [27]*.
- Infine, prima di riavviare, è necessario impostare l'orologio a UTC.

Se durante l'installazione non si è soddisfatti delle impostazioni predefinite, usare la funzione «Indietro» per visualizzare un menù d'installazione dettagliato che consente di modificare le impostazioni.

In qualsiasi momento dell'installazione è possibile leggere la guida fornita dal sistema, basta premere F1.

Once again, for detailed instructions see the *Ubuntu Installation Guide*[5].

2.1. Pacchetti per attività specifiche

Durante l'installazione della Server Edition è possibile installare dei pacchetti aggiuntivi, raggruppati per il tipo di servizio che forniscono.

- Server DNS: seleziona il server DNS BIND e la documentazione.
- Server LAMP: seleziona un server Linux/Apache/MySQL/PHP.
- Mail server: This task selects a variety of packages useful for a general purpose mail server system.
- Server OpenSSH: seleziona i pacchetti necessari per un server OpenSSH.
- Server PostgreSQL: seleziona i pacchetti client e server per il database PostreSQL.
- Server di stampa: configura il sistema come un server di stampa.
- Server file Samba: configura il sistema come server di file Samba, utile particolarmente all'interno di reti eterogenee, con sistemi Windows e Linux.
- Tomcat Java server: Installs Apache Tomcat and needed dependencies.
- Virtual Machine host: Includes packages needed to run KVM virtual machines.
- Manually select packages: Executes aptitude allowing you to individually select packages.

Installing the package groups is accomplished using the tasksel utility. One of the important differences between Ubuntu (or Debian) and other GNU/Linux distribution is that, when installed, a package is also configured to reasonable defaults, eventually prompting you for additional required information. Likewise, when installing a task, the packages are not only installed, but also configured to provided a fully integrated service.

[4] http://www.canonical.com/projects/landscape
[5] https://help.ubuntu.com/12.04/installation-guide/

Una volta completata l'installazione, è possibile vedere un elenco dei "task" disponibili digitando il seguente comando:

```
tasksel --list-tasks
```

L'output elenca i "task" di altre distribuzioni basate su Ubuntu come Kubuntu ed Edubuntu. È comunque possibile invocare il comando **tasksel**, che presenta un menù con i diversi "task" disponibili.

Tramite l'opzione *--task-packages* è possibile visualizzare un elenco dei pacchetti installati con ogni "task". Per esempio, per elencare i pacchetti installati con *DNS Server* digitare:

```
tasksel --task-packages dns-server
```

L'output del comando dovrebbe essere:

```
bind9-doc
bind9utils
bind9
```

If you did not install one of the tasks during the installation process, but for example you decide to make your new LAMP server a DNS server as well, simply insert the installation CD and from a terminal:

```
sudo tasksel install dns-server
```

3. Avanzamento di versione

Ci sono diversi metodi per eseguire un avanzamento da un rilascio di Ubuntu a un altro. In questa sezione vengono presentati i metodi raccomandati.

3.1. do-release-upgrade

Il metodo di avanzamento raccomandato per la Server Edition è l'utilizzo dell'utilità do-release-upgrade, installata in modo predefinito come parte del pacchetto *update-manager-core* e priva di alcuna dipendenza grafica.

I sistemi basati su Debian possono ricorrere anche al comando **apt-get dist-upgrade**. L'uso di do-release-upgrade è comunque raccomandato in quanto è in grado di gestire le modifiche necessarie alla configurazione di sistema tra i rilasci.

Per avanzare a un nuovo rilascio, da un terminale digitare:

```
do-release-upgrade
```

È anche possibile usare do-release-upgrade per avanzare a una versione di sviluppo di Ubuntu. Per fare ciò, usare l'opzione *-d*:

```
do-release-upgrade -d
```

Avanzare a una versione di sviluppo *non* è consigliato in ambienti di produzione.

4. Installazione avanzata

4.1. RAID software

Redundant Array of Independent Disks "RAID" is a method of using multiple disks to provide different balances of increasing data reliability and/or increasing input/output performance, depending on the RAID level being used. RAID is implemented in either software (where the operating system knows about both drives and actively maintains both of them) or hardware (where a special controller makes the OS think there's only one drive and maintains the drives 'invisibly').

Il RAID software incluso nelle attuali versioni di Linux (e Ubuntu) è basato sul driver mdadm e funziona perfettamente, molto meglio di alcuni cosiddetti controller RAID hardware. In questa sezione viene spiegato come installare Ubuntu Server Edition utilizzando due partizioni RAID1 su due dischi fissi, uno utilizzato per */* e l'altro come *swap*.

4.1.1. Partizionamento

Seguire i passi dell'installazione fino a giungere a *Partizionamento dei dischi*, quindi:

1. Selezionare *Manuale* come metodo di partizionamento.
2. Selezionare il primo disco fisso e acconsentire alla domanda *Creare una nuova tabella delle partizioni sul dispositivo*.

   Ripetere questo passo per ogni disco da inserire nell'array RAID.
3. Selezionare lo *spazio libero* sul primo disco e quindi selezionare *Creare una nuova partizione*.
4. Selezionare la *Dimensione* della partizione: questa partizione sarà quella di *swap* e come regola generale, la dimensione della partizione di swap è solitamente il doppio della memoria RAM. Digitare la dimensione della partizione, scegliere *Primaria* e quindi *Inizio*.

   

   A swap partition size of twice the available RAM capacity may not always be desirable, especially on systems with large amounts of RAM. Calculating the swap partition size for servers is highly dependent on how the system is going to be used.
5. Select the *"Use as:"* line at the top. By default this is *"Ext4 journaling file system"*, change that to *"physical volume for RAID"* then *"Done setting up partition"*.
6. Per la partizione */*, selezionare *spazio libero* sul primo drive e quindi *Crea una nuova partizione*.
7. Utilizzare il restante spazio libero sul dispositivo e scegliere *Continua*, quindi *Primaria*.
8. As with the swap partition, select the *"Use as:"* line at the top, changing it to *"physical volume for RAID"*. Also select the *"Bootable flag:"* line to change the value to *"on"*. Then choose *"Done setting up partition"*.
9. Ripetere i passi dal 3 al numero 8 per gli altri dischi e partizioni.

4.1.2. Configurare RAID

Impostate le partizioni è quindi possibile configurare gli array:

1. Nella sezione di partizionamento dei dischi, selezionare *Configurare il software RAID*.
2. Selezione *Sì* per scrivere le modifiche sul disco.
3. Choose *"Create MD device"*.
4. Per questo esempio, selezionare *RAID1*. Nel caso si stia utilizzando una diversa configurazione, scegliere la tipologia adatta (RAID0 RAID1 RAID5).

   

   Per poter usare il *RAID5* sono necessari almeno *tre* dischi. Per RAID0 oppure RAID1 solo *due*.
5. Inserire il numero dei dispositivi attivi (active), *2*, oppure il numero totale dei dischi disponibili per l'array, quindi selezionare *Continua*.
6. Inserire il numero dei dispositivi di scorta (spare), *0* come valore predefinito, quindi selezionare *Continua*.
7. Scegliere la partizione da usare: solitamente sda1, sdb1, sdc1, ecc... I numeri e le lettere solitamente corrispondono a diversi dischi fissi.

   Per la partizione di *swap* scegliere *sda1* e *sdb1*. Selezionare *Continua* per andare al passo successivo.
8. Ripetere i passi dal *tre* al *sette* per la partizione */* scegliendo *sda2* e *sdb2*.
9. Una volta completato tutto, selezionare *Terminare*.

4.1.3. Formattare

Dovrebbe essere visibile un elenco di dischi fissi e dispositivi RAID. Il passo successivo consiste nel formattare e impostare il punto di mount per i dispositivi RAID: tali dispositivi sono da considerare come dei normali dischi locali.

1. Select *"#1"* under the *"RAID1 device #0"* partition.
2. Scegliere *Usato come:*, quindi *area di swap* e infine *Preparazione di questa partizione completata*.
3. Next, select *"#1"* under the *"RAID1 device #1"* partition.
4. Choose *"Use as:"*. Then select *"Ext4 journaling file system"*.
5. Selezionare *Punto di mount* e scegliere */ - il file system root*. Modificare se necessario le altre opzioni e selezionare *Preparazione di questa partizione completata*.
6. Selezionare *Terminare il partizionamento e scrivere i cambiamenti sul disco*.

Se è stato scelto di posizionare la partizione di root nell'array RAID, il programma di installazione chiederà se avviare il sistema in modalità *degraded*. Per maggiori informazioni, consultare *Sezione 4.1.4, «RAID degraded» [12]*.

Il processo di installazione continuerà normalmente.

4.1.4. RAID degraded

Durante l'arco di vita di un computer si potrebbero verificare dei danni ai dischi. Quando si verifica un'eventualità come questa, usando il RAID software, il sistema operativo abilita la modalità *degraded* per l'array.

Se l'array è degradato ("degraded") a causa di dati rovinati, il sistema operativo, in modo predefinito, si avvierà in *initramfs* dopo 30 secondi. Una volta avviato, è possibile, entro 15 secondi, continuare il normale avvio o tentare un ripristino manuale. L'avvio in initramfs potrebbe non essere consigliato, soprattutto se si opera sul computer da remoto. Avviare il sistema in un array "degraded" può essere svolto in diversi modi:

- L'utilità dpkg-reconfigure può essere usata per configurare il comportamento predefinito e durante l'elaborazione verranno poste delle domande relative a impostazioni addizionali per l'array come monitoraggio, avvisi via email, ecc... Per riconfigurare mdadm, digitare il seguente comando:

```
sudo dpkg-reconfigure mdadm
```

- Il processo **dpkg-reconfigure mdadm** modificherà il file di configurazione `/etc/initramfs-tools/conf.d/mdadm`. Tale file presenta il vantaggio di pre-configurare il comportamento del sistema e può essere modificato a mano:

```
BOOT_DEGRADED=true
```

Il file di configurazione può essere scavalcato utilizzando un argomento per il kernel.

- È possibile avviare il sistema in un array "degraded" utilizzando anche un argomento per il kernel:
  - When the server is booting press **Shift** to open the Grub menu.
  - Press **e** to edit your kernel command options.
  - Press the **down** arrow to highlight the kernel line.
  - Aggiungere *bootdegraded=true* alla fine della riga.
  - Premere **Ctrl**+**x** per avviare il sistema.

Una volta avviato il sistema, è possibile riparare l'array (consultare *Sezione 4.1.5, «Manutenzione RAID» [12]*) o copiare i dati importanti in un altro computer.

4.1.5. Manutenzione RAID

L'utilità mdadm può essere usata per visualizzare lo stato dell'array, aggiungere un disco all'array, rimuovere dischi, ecc...

- Per visualizzare lo stato di un array, da un terminale digitare:

```
sudo mdadm -D /dev/md0
```

L'opzione *-D* indica a mdadm di stampare informazioni *dettagliate* riguardo il device `/dev/md0`. Sostituire `/dev/md0` con il device RAID appropriato.

- Per visualizzare lo stato di un disco in un array:

```
sudo mdadm -E /dev/sda1
```

L'output è molto simile al comando **mdadm -D**, regolare `/dev/sda1` per ogni disco.

- Se un disco si rompe e deve essere rimosso da un array:

```
sudo mdadm --remove /dev/md0 /dev/sda1
```

Modificare `/dev/md0` e `/dev/sda1` con il device e il disco RAID appropriati.

- Per aggiungere un nuovo disco:

```
sudo mdadm --add /dev/md0 /dev/sda1
```

Qualche volta può succedere che un disco imposti il suo stato come *difettoso* ("faulty"), anche se non presenta alcun malfunzionamento hardware. Può essere utile in questi casi rimuovere e aggiungere il disco all'array: verrà così nuovamente sincronizzato con l'array. Se il disco non riesce a sincronizzarsi con l'array, può indicare che il dispositivo sia effettivamente difettoso.

Il file `/proc/mdstat` contiene anche informazioni utili riguardo i device RAID del sistema:

```
cat /proc/mdstat
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sda1[0] sdb1[1]
      10016384 blocks [2/2] [UU]

unused devices: <none>
```

Il seguente comando è utile per controllare lo stato di un drive di sincronizzazione:

```
watch -n1 cat /proc/mdstat
```

Premere *Ctrl+C* per fermare il comando watch.

Se è necessario sostituire un disco difettoso, una volta sostituito e sincronizzato, è necessario reinstallare grub. Per installare grub nel nuovo disco, procedere come segue:

```
sudo grub-install /dev/md0
```

Sostituire `/dev/md0` con il nome dell'array appropriato.

4.1.6. Risorse

L'argomento degli array RAID è molto complesso e vasto poiché sono disponibili molti modi diversi di configurare un array RAID. Per maggiori informazioni, consultare i seguenti collegamenti:

- *Documentazione online riguardo RAID*[6].
- *Software RAID HOWTO*[7]
- *Managing RAID on Linux*[8]

4.2. Logical Volume Manager (LVM)

Logical Volume Manger, o *LVM*, consente agli amministratori di creare volumi *logici* da uno o più dischi fissi. I volumi LVM possono essere creati sia sulle partizioni RAID software sia sulle partizioni normali presenti su un singolo disco. I volumi possono essere estesi, garantendo un'alta flessibilità al sistema nel caso cambino le necessità.

4.2.1. Panoramica

Purtroppo, la potenza e la flessibilità di LVM, comportano maggiori complicazioni. Prima di tutto è quindi necessario introdurre la terminologia adatta.

- *Physical Volume (PV):* physical hard disk, disk partition or software RAID partition formatted as LVM PV.
- *Volume Group (VG):* is made from one or more physical volumes. A VG can can be extended by adding more PVs. A VG is like a virtual disk drive, from which one or more logical volumes are carved.
- *Logical Volume (LV):* is similar to a partition in a non-LVM system. A LV is formatted with the desired file system (EXT3, XFS, JFS, etc), it is then available for mounting and data storage.

4.2.2. Installazione

Come esempio, in questa sezione, viene descritto come installare Ubuntu Server Edition con `/srv` montato come volume LVM. Durante l'installazione un solo volume fisico (PV) farà parte del gruppo di volumi (VG). Un altro PV verrà aggiunto dopo l'installazione come dimostrazione delle funzionalità di estensione di un VG.

Sono disponibili diverse opzioni per l'installazione LVM, *Guidato - usare l'intero disco e impostare LVM* consente di assegnare una parte dello spazio disponibile a LVM, *Guidato - usare l'intero disco e impostare LVM cifrato* o *manuale*. Attualmente l'unico metodo per configurare un sistema affinché utilizzi sia partizioni LVM che normali durante l'installazione è quello manuale.

1. Seguire i passi dell'installazione fino a giungere a *Partizionamento dei dischi*, quindi:

[6] https://help.ubuntu.com/community/Installation#raid
[7] http://www.faqs.org/docs/Linux-HOWTO/Software-RAID-HOWTO.html
[8] http://oreilly.com/catalog/9781565927308/

2. Alla finestra *Partizionamento dei dischi* scegliere *Manuale*.
3. Selezionare il disco fisso e nella schermata successiva scegliere confermare *Creare una nuova tabella delle partizioni sul dispositivo*.
4. Creare le partizioni */boot*, *swap* e */* con il file system di propria scelta.
5. Per la partizione */srv* LVM, creare una nuova partizione *Logica* e modificare *Usato come* in *volume fisico per LVM*, quindi selezionare *Preparazione di questa partizione completata*.
6. Selezionare *Configurare il Logical Volume Manager* in alto e scegliere *Sì* per scrivere le modifiche sul disco.
7. Per il *Passo di configurazione di LVM* nella schermata successiva, scegliere *Creare gruppi di volumi*. Inserire un nome per il VG come *vg01* o qualche cosa più descrittivo. Fatto ciò, selezionare la partizione configurata per LVM e scegliere *Continua*.
8. Sempre nella schermata *Passo di configurazione di LVM*, selezionare *Creare volume logico*, selezionare il gruppo di volumi appena creato e inserire un nome per il nuovo LV, per esempio *srv* dato che verrà utilizzato come punto di mount per quella partizione. Scegliere la dimensione, che in questo caso può essere l'intera partizione dato che è possibile estenderla o ridurla, scegliere *Termina* per tornare alla schermata *Partizionamento dei dischi*.
9. Ora aggiungere il file system al nuovo LVM. Selezionare la partizione *LVM VG vg01, LV srv*, o in base al nome inserito, e scegliere *Usato come*. Impostare un file system selezionando */srv* come punto di mount e una volta completato, selezionare *Preparazione di questa partizione completata*.
10. Infine, selezionare *Terminare il partizionamento e scrivere i cambiamenti sul disco*, confermare le modifiche e continuare l'installazione.

Per visualizzare informazioni riguardo LVM sono disponibili diverse utilità:

- *pvdisplay:* shows information about Physical Volumes.
- *vgdisplay*: visualizza informazioni riguardo i gruppi di volumi.
- *lvdisplay:* shows information about Logical Volumes.

4.2.3. Estendere i gruppi di volumi

Continuing with *srv* as an LVM volume example, this section covers adding a second hard disk, creating a Physical Volume (PV), adding it to the volume group (VG), extending the logical volume `srv` and finally extending the filesystem. This example assumes a second hard disk has been added to the system. In this example, this hard disk will be named `/dev/sdb` and we will use the entire disk as a physical volume (you could choose to create partitions and use them as different physical volumes)

Make sure you don't already have an existing `/dev/sdb` before issuing the commands below. You could lose some data if you issue those commands on a non-empty disk.

1. Creare il volume fisico. In un terminale digitare:

```
sudo pvcreate /dev/sdb
```

2. Estendere il gruppo di volumi (VG):

```
sudo vgextend vg01 /dev/sdb
```

3. Usare vgdisplay per trovare gli extent fisici (PE) liberi (PE/dimensione = dimensione da allocare). In questo esempio viene considerata una dimensione di 511 PE (equivalenti a 2GB con una dimensione di PE di 4MB) e viene utilizzato tutto lo spazio libero. Utilizzare i PE in base alle proprie disponibilità.

   Il volume logico (LV) può essere esteso in diversi modi. In questo esempio viene considerato il caso di utilizzo del PE per estendere il LV:

```
sudo lvextend /dev/vg01/srv -l +511
```

   L'opzione *-l* consente di estendere il LV attraverso l'uso di PE. L'opzione *-L* invece, consente di estendere il LV utilizzando megabyte, gigabyte, terabyte, ecc...

4. Even though you are supposed to be able to *expand* an ext3 or ext4 filesystem without unmounting it first, it may be a good practice to unmount it anyway and check the filesystem, so that you don't mess up the day you want to reduce a logical volume (in that case unmounting first is compulsory).

   I seguenti comandi sono pensati per un file system *ext3* o *ext4*. Se si sta utilizzando un altro file system potrebbero essere disponibili altri programmi.

```
sudo umount /srv
sudo e2fsck -f /dev/vg01/srv
```

   L'opzione *-f* di e2fsck forza il controllo anche se il file system sembra non avere problemi.

5. Infine, ridimensionare il file system:

```
sudo resize2fs /dev/vg01/srv
```

6. Montare la partizione e controllarne la dimensione.

```
mount /dev/vg01/srv /srv && df -h /srv
```

4.2.4. Risorse

- Consultare la *documentazione online riguardo LVM*[9].
- Per maggiori informazioni, consultare *LVM HOWTO*[10].
- Un ottimo articolo presente su linuxdevcenter.com è *Managing Disk Space with LVM*[11].

[9] https://help.ubuntu.com/community/Installation#lvm
[10] http://tldp.org/HOWTO/LVM-HOWTO/index.html
[11] http://www.linuxdevcenter.com/pub/a/linux/2006/04/27/managing-disk-space-with-lvm.html

- For more information on fdisk see the *fdisk man page*[12].

[12] http://manpages.ubuntu.com/manpages/precise/en/man8/fdisk.8.html

5. Kernel Crash Dump

5.1. Introduzione

A Kernel Crash Dump refers to a portion of the contents of volatile memory (RAM) that is copied to disk whenever the execution of the kernel is disrupted. The following events can cause a kernel disruption :

- Kernel Panic
- Non Maskable Interrupts (NMI)
- Machine Check Exceptions (MCE)
- Hardware failure
- Manual intervention

For some of those events (panic, NMI) the kernel will react automatically and trigger the crash dump mechanism through *kexec*. In other situations a manual intervention is required in order to capture the memory. Whenever one of the above events occurs, it is important to find out the root cause in order to prevent it from happening again. The cause can be determined by inspecting the copied memory contents.

5.2. Kernel Crash Dump Mechanism

When a kernel panic occurs, the kernel relies on the *kexec* mechanism to quickly reboot a new instance of the kernel in a pre-reserved section of memory that had been allocated when the system booted (see below). This permits the existing memory area to remain untouched in order to safely copy its contents to storage.

5.3. Installazione

The kernel crash dump utility is installed with the following command:

```
sudo apt-get install linux-crashdump
```

A reboot is then needed.

5.4. Configurazione

No further configuration is required in order to have the kernel dump mechanism enabled.

5.5. Verifica

To confirm that the kernel dump mechanism is enabled, there are a few things to verify. First, confirm that the *crashkernel* boot parameter is present (note: The following line has been split into two to fit the format of this document:

```
cat /proc/cmdline
```

```
BOOT_IMAGE=/vmlinuz-3.2.0-17-server root=/dev/mapper/PreciseS-root ro
 crashkernel=384M-2G:64M,2G-:128M
```

The *crashkernel* parameter has the following syntax:

```
crashkernel=<range1>:<size1>[,<range2>:<size2>,...][@offset]
    range=start-[end] 'start' is inclusive and 'end' is exclusive.
```

So for the crashkernel parameter found in `/proc/cmdline` we would have :

```
crashkernel=384M-2G:64M,2G-:128M
```

The above value means:

- if the RAM is smaller than 384M, then don't reserve anything (this is the "rescue" case)
- if the RAM size is between 386M and 2G (exclusive), then reserve 64M
- if the RAM size is larger than 2G, then reserve 128M

Second, verify that the kernel has reserved the requested memory area for the kdump kernel by doing:

```
dmesg | grep -i crash
```

```
...
[    0.000000] Reserving 64MB of memory at 800MB for crashkernel (System RAM: 1023MB)
```

5.6. Testing the Crash Dump Mechanism

Testing the Crash Dump Mechanism will cause *a system reboot*. In certain situations, this can cause data loss if the system is under heavy load. If you want to test the mechanism, make sure that the system is idle or under very light load.

Verify that the *SysRQ* mechanism is enabled by looking at the value of the `/proc/sys/kernel/sysrq` kernel parameter :

```
cat /proc/sys/kernel/sysrq
```

If a value of *0* is returned the feature is disabled. Enable it with the following command :

```
sudo sysctl -w kernel.sysrq=1
```

Once this is done, you must become root, as just using **sudo** will not be sufficient. As the *root* user, you will have to issue the command **echo c > /proc/sysrq-trigger**. If you are using a network

connection, you will lose contact with the system. This is why it is better to do the test while being connected to the system console. This has the advantage of making the kernel dump process visible.

A typical test output should look like the following :

```
sudo -s
[sudo] password for ubuntu:
# echo c > /proc/sysrq-trigger
[   31.659002] SysRq : Trigger a crash
[   31.659749] BUG: unable to handle kernel NULL pointer dereference at           (null)
[   31.662668] IP: [<ffffffff8139f166>] sysrq_handle_crash+0x16/0x20
[   31.662668] PGD 3bfb9067 PUD 368a7067 PMD 0
[   31.662668] Oops: 0002 [#1] SMP
[   31.662668] CPU 1
....
```

The rest of the output is truncated, but you should see the system rebooting and somewhere in the log, you will see the following line :

```
Begin: Saving vmcore from kernel crash ...
```

Once completed, the system will reboot to its normal operational mode. You will then find Kernel Crash Dump file in the `/var/crash` directory :

```
ls /var/crash
linux-image-3.0.0-12-server.0.crash
```

5.7. Risorse

Kernel Crash Dump is a vast topic that requires good knowledge of the linux kernel. You can find more information on the topic here :

- *Kdump kernel documentation*[13].
- *The crash tool*[14]
- *Analyzing Linux Kernel Crash*[15] (Based on Fedora, it still gives a good walkthrough of kernel dump analysis)

[13] http://www.kernel.org/doc/Documentation/kdump/kdump.txt
[14] http://people.redhat.com/~anderson/
[15] http://www.dedoimedo.com/computers/crash-analyze.html

Capitolo 3. Gestione dei pacchetti

Ubuntu features a comprehensive package management system for installing, upgrading, configuring, and removing software. In addition to providing access to an organized base of over 35,000 software packages for your Ubuntu computer, the package management facilities also feature dependency resolution capabilities and software update checking.

Per l'interazione con il sistema di gestione dei pacchetti di Ubuntu sono disponibili diversi strumenti, a partire da semplici utilità a riga di comando che possono essere usate con facilità da amministratori di sistema per attività automatizzate, fino a interfacce grafiche semplici da usare per chi si è avvicinato da poco a Ubuntu.

1. Introduzione

Il sistema di gestione dei pacchetti di Ubuntu è derivato dallo stesso sistema usato dalla distribuzione Debian GNU/Linux. I file di pacchetto contengono tutti i file, i meta-dati e le istruzioni necessari per implementare sui sistemi Ubuntu una particolare funzionalità o un'applicazione software.

Debian package files typically have the extension '.deb', and usually exist in *repositories* which are collections of packages found on various media, such as CD-ROM discs, or online. Packages are normally in a pre-compiled binary format; thus installation is quick, and requires no compiling of software.

Many complex packages use the concept of *dependencies*. Dependencies are additional packages required by the principal package in order to function properly. For example, the speech synthesis package festival depends upon the package libasound2, which is a package supplying the ALSA sound library needed for audio playback. In order for festival to function, it and all of its dependencies must be installed. The software management tools in Ubuntu will do this automatically.

2. dpkg

dpkg is a package manager for *Debian*-based systems. It can install, remove, and build packages, but unlike other package management systems, it cannot automatically download and install packages or their dependencies. This section covers using dpkg to manage locally installed packages:

- To list all packages installed on the system, from a terminal prompt type:

```
dpkg -l
```

- In base a quanti pacchetto sono installati nel sistema, questo comando può generare molto output. È comunque possibile passare l'output attraverso una pipe all'applicazione grep per vedere se un particolare pacchetto è installato o meno:

```
dpkg -l | grep apache2
```

  Sostituire *apache2* con il nome di un qualsiasi altro pacchetto, parte del nome o qualsiasi altra espressione regolare.

- Per elencare i file installati da un pacchetto, in questo caso ufw, digitare:

```
dpkg -L ufw
```

- Se non si è sicuri di quale pacchetto abbia installato un file, usare il comando dpkg -S. Per esempio:

```
dpkg -S /etc/host.conf
base-files: /etc/host.conf
```

  L'output mostra che `/etc/host.conf` appartiene al pacchetto base-files.

  Many files are automatically generated during the package install process, and even though they are on the filesystem, **dpkg -S** may not know which package they belong to.

- Per installare un file *.deb* locale, digitare:

```
sudo dpkg -i zip_3.0-4_i386.deb
```

  Change `zip_3.0-4_i386.deb` to the actual file name of the local .deb file you wish to install.

- Per disinstallare un pacchetto:

```
sudo dpkg -r zip
```

  Uninstalling packages using dpkg, in most cases, is *NOT* recommended. It is better to use a package manager that handles dependencies to ensure that the system is in a consistent state. For example using **dpkg -r zip** will remove the zip package, but any packages that depend on it will still be installed and may no longer function correctly.

Per le ulteriori opzioni di dpkg, consultare la pagina di manuale: **man dpkg**.

3. Apt-Get

The apt-get command is a powerful command-line tool, which works with Ubuntu's *Advanced Packaging Tool* (APT) performing such functions as installation of new software packages, upgrade of existing software packages, updating of the package list index, and even upgrading the entire Ubuntu system.

Being a simple command-line tool, apt-get has numerous advantages over other package management tools available in Ubuntu for server administrators. Some of these advantages include ease of use over simple terminal connections (SSH), and the ability to be used in system administration scripts, which can in turn be automated by the cron scheduling utility.

Alcuni esempi di utilizzo tipico dell'utilità apt-get:

- **Install a Package**: Installation of packages using the apt-get tool is quite simple. For example, to install the network scanner nmap, type the following:

  ```
  sudo apt-get install nmap
  ```

- **Remove a Package**: Removal of a package (or packages) is also straightforward. To remove the package installed in the previous example, type the following:

  ```
  sudo apt-get remove nmap
  ```

  

  Pacchetti multipli: è possibile specificare più di un pacchetto da installare o rimuovere, separati da spazi.

  Also, adding the *--purge* option to **apt-get remove** will remove the package configuration files as well. This may or may not be the desired effect, so use with caution.

- **Update the Package Index**: The APT package index is essentially a database of available packages from the repositories defined in the `/etc/apt/sources.list` file and in the `/etc/apt/sources.list.d` directory. To update the local package index with the latest changes made in the repositories, type the following:

  ```
  sudo apt-get update
  ```

- **Aggiornare i pacchetti**: versioni aggiornate dei pacchetti installati possono essere disponibili attraverso i repository dei pacchetti (per esempio per aggiornamenti di sicurezza). Per aggiornare il proprio sistema è necessario, prima di tutto, aggiornare l'indice dei pacchetti come spiegato sopra, quindi digitare:

  ```
  sudo apt-get upgrade
  ```

  Per informazioni sull'avanzamento a un nuovo rilascio di Ubuntu, consultare la *Sezione 3, «Avanzamento di versione» [9]*.

Le azioni del comando apt-get, come l'installazione o la rimozione di pacchetti, vengono registrate nel file di registro /var/log/dpkg.log.

For further information about the use of APT, read the comprehensive *Debian APT User Manual*[1] or type:

```
apt-get help
```

[1] http://www.debian.org/doc/user-manuals#apt-howto

4. Aptitude

Launching Aptitude with no command-line options, will give you a menu-driven, text-based front-end to the *Advanced Packaging Tool* (APT) system. Many of the common package management functions, such as installation, removal, and upgrade, can be performed in Aptitude with single-key commands, which are typically lowercase letters.

Aptitude is best suited for use in a non-graphical terminal environment to ensure proper functioning of the command keys. You may start the menu-driven interface of Aptitude as a normal user by typing the following command at a terminal prompt:

```
sudo aptitude
```

When Aptitude starts, you will see a menu bar at the top of the screen and two panes below the menu bar. The top pane contains package categories, such as *New Packages* and *Not Installed Packages*. The bottom pane contains information related to the packages and package categories.

Usare Aptitude per la gestione dei pacchetti è relativamente chiaro e l'interfaccia utente rende le operazioni comuni semplici da eseguire. Di seguito vengono presentati alcuni esempi di funzioni comuni della gestione dei pacchetti con Aptitude:

- **Install Packages**: To install a package, locate the package via the *Not Installed Packages* package category, by using the keyboard arrow keys and the **ENTER** key. Highlight the desired package, then press the **+** key. The package entry should turn *green*, indicating that it has been marked for installation. Now press **g** to be presented with a summary of package actions. Press **g** again, and you will be prompted to become root to complete the installation. Press **ENTER** which will result in a *Password:* prompt. Enter your user password to become root. Finally, press **g** once more and you'll be prompted to download the package. Press **ENTER** on the *Continue* prompt, and downloading and installation of the package will commence.
- **Remove Packages**: To remove a package, locate the package via the *Installed Packages* package category, by using the keyboard arrow keys and the **ENTER** key. Highlight the desired package you wish to remove, then press the **-** key. The package entry should turn *pink*, indicating it has been marked for removal. Now press **g** to be presented with a summary of package actions. Press **g** again, and you will be prompted to become root to complete the removal. Press **ENTER** which will result in a *Password:* prompt. Enter your user password to become root. Finally, press **g** once more, then press **ENTER** on the *Continue* prompt, and removal of the package will commence.
- **Update Package Index**: To update the package index, simply press the **u** key and you will be prompted to become root to complete the update. Press **ENTER** which will result in a *Password:* prompt. Enter your user password to become root. Updating of the package index will commence. Press **ENTER** on the *OK* prompt when the download dialog is presented to complete the process.
- **Upgrade Packages**: To upgrade packages, perform the update of the package index as detailed above, and then press the **U** key to mark all packages with updates. Now press **g** whereby you'll be presented with a summary of package actions. Press **g** again, and you will be prompted to

become root to complete the installation. Press **ENTER** which will result in a *Password:* prompt. Enter your user password to become root. Finally, press **g** once more, and you'll be prompted to download the packages. Press **ENTER** on the *Continue* prompt, and upgrade of the packages will commence.

La prima colonna delle informazioni mostrate nell'elenco dei pacchetti nel riquadro superiore, indica l'attuale stato del pacchetto, utilizzando le seguenti chiavi per descrivere lo stato del pacchetto:

- **i**: pacchetto installato
- **c**: pacchetto non installato, ma nel sistema è rimasta traccia della configurazione del pacchetto
- **p**: rimosso completamente dal sistema
- **v**: pacchetto virtuale
- **B**: pacchetto non integro
- **u**: file decompressi, ma pacchetto non ancora configurato
- **C**: configurato in parte. La configurazione è fallita e necessita di essere corretta
- **H**: installato parzialmente. La rimozione è fallita e necessita di essere sistemata

Per chiudere Aptitude, è sufficiente premere il tasto **q** e confermare l'uscita. Sono disponibili molte altre funzioni dal menù di Aptitude, premendo il tasto **F10**.

4.1. Command Line Aptitude

You can also use Aptitude as a command-line tool, similar to apt-get. To install the nmap package with all necessary dependencies, as in the apt-get example, you would use the following command:

```
sudo aptitude install nmap
```

To remove the same package, you would use the command:

```
sudo aptitude remove nmap
```

Consult the man pages for more details of command line options for Aptitude.

5. Aggiornamenti automatici

Il pacchetto unattended-upgrades può essere usato per installare automaticamente gli aggiornamenti e può essere configurato per aggiornare tutti i pacchetti o installare solamente gli aggiornamenti di sicurezza. Per prima cosa, installare il pacchetto digitando:

```
sudo apt-get install unattended-upgrades
```

Per configurare unattended-upgrades, aprire il file `/etc/apt/apt.conf.d/50unattended-upgrades` e modificare quanto segue secondo le proprie esigenze:

```
Unattended-Upgrade::Allowed-Origins {
        "Ubuntu precise-security";
//      "Ubuntu precise-updates";
};
```

Alcuni pacchetti possono essere inseriti nella *blacklist* per non aggiornarli mai. Per inserire un pacchetto nella blacklist, aggiungerlo all'elenco:

```
Unattended-Upgrade::Package-Blacklist {
//      "vim";
//      "libc6";
//      "libc6-dev";
//      "libc6-i686";
};
```

I doppi slash («//») servono come commento; tutto quello che segue "//" non verrà valutato.

To enable automatic updates, edit `/etc/apt/apt.conf.d/10periodic` and set the appropriate apt configuration options:

```
APT::Periodic::Update-Package-Lists "1";
APT::Periodic::Download-Upgradeable-Packages "1";
APT::Periodic::AutocleanInterval "7";
APT::Periodic::Unattended-Upgrade "1";
```

The above configuration updates the package list, downloads, and installs available upgrades every day. The local download archive is cleaned every week.

You can read more about apt Periodic configuration options in the `/etc/cron.daily/apt` script header.

I risultati di unattended-upgrades vengono registrati in `/var/log/unattended-upgrades`.

5.1. Notifiche

Impostando *Unattended-Upgrade::Mail* nel file `/etc/apt/apt.conf.d/50unattended-upgrades`, si abilita unattended-upgrades all'invio di email all'amministratore indicando i pacchetti da aggiornare o con problemi.

Another useful package is apticron. apticron will configure a cron job to email an administrator information about any packages on the system that have updates available, as well as a summary of changes in each package.

Per installare apticron, digitare:

```
sudo apt-get install apticron
```

Una volta installato, aprire il file `/etc/apticron/apticron.conf` e impostare l'indirizzo email e altre opzioni:

```
EMAIL="root@example.it"
```

6. Configurazione

Configuration of the *Advanced Packaging Tool* (APT) system repositories is stored in the `/etc/apt/sources.list` file and the `/etc/apt/sources.list.d` directory. An example of this file is referenced here, along with information on adding or removing repository references from the file.

Questo[2] è un semplice esempio di un tipico file `/etc/apt/sources.list`.

È possibile modificare il file per abilitare o disabilitare i repository. Per esempio, per disabilitare la necessità di inserire il CD-ROM di Ubuntu ogni volta che viene effettuata un'operazione sui pacchetti, è sufficiente commenentare la riga relativa al CD-ROM, che si trova all'inizio del file:

```
# no more prompting for CD-ROM please
# deb cdrom:[Ubuntu 12.04 _Precise Pangolin_ - Release i386 (20111013.1)]/ precise main restricted
```

6.1. Repository aggiuntivi

In addition to the officially supported package repositories available for Ubuntu, there exist additional community-maintained repositories which add thousands more packages for potential installation. Two of the most popular are the *Universe* and *Multiverse* repositories. These repositories are not officially supported by Ubuntu, but because they are maintained by the community they generally provide packages which are safe for use with your Ubuntu computer.

I pacchetti nel repository *multiverse* presentano spesso problemi di licenza che non gli permettono di essere distribuiti con un sistema operativo gratuito e potrebbero essere illegali in alcuni paesi.

Né il repository *universe* né quello *multiverse* contengono pacchetti supportati ufficialmente. In particolare, potrebbero non esserci aggiornamenti di sicurezza per tali pacchetti.

Sono disponibili molte altre sorgenti di pacchetti, alcune delle quali offrono solo un pacchetto, come nel caso di sorgenti di pacchetto fornite dallo sviluppatore di una singola applicazione. L'utilizzo di sorgenti di pacchetto non standard è rischioso, pertanto è necessario prestare la massima attenzione. È opportuno controllare la sorgente e i pacchetti in modo accurato prima di effettuare una qualsiasi installazione, poiché alcune sorgenti di pacchetto, e i rispettivi pacchetti, potrebbero rendere il sistema instabile e non funzionante sotto certi aspetti.

I repository *universe* e *multiverse*, in modo predefinito, sono abilitati, ma se si desidera disabilitarli è possibile modificare il file `/etc/apt/sources.list` e commentare le seguenti righe:

```
deb http://archive.ubuntu.com/ubuntu precise universe multiverse
deb-src http://archive.ubuntu.com/ubuntu precise universe multiverse
```

[2] ../sample/sources.list

```
deb http://us.archive.ubuntu.com/ubuntu/ precise universe
deb-src http://us.archive.ubuntu.com/ubuntu/ precise universe
deb http://us.archive.ubuntu.com/ubuntu/ precise-updates universe
deb-src http://us.archive.ubuntu.com/ubuntu/ precise-updates universe

deb http://us.archive.ubuntu.com/ubuntu/ precise multiverse
deb-src http://us.archive.ubuntu.com/ubuntu/ precise multiverse
deb http://us.archive.ubuntu.com/ubuntu/ precise-updates multiverse
deb-src http://us.archive.ubuntu.com/ubuntu/ precise-updates multiverse

deb http://security.ubuntu.com/ubuntu precise-security universe
deb-src http://security.ubuntu.com/ubuntu precise-security universe
deb http://security.ubuntu.com/ubuntu precise-security multiverse
deb-src http://security.ubuntu.com/ubuntu precise-security multiverse
```

7. Riferimenti

La maggior parte di quanto discusso in questo capitolo è disponibile nella pagine man, molte delle quali sono reperibili anche in rete.

- The *InstallingSoftware*[3] Ubuntu wiki page has more information.
- For more dpkg details see the *dpkg man page*[4].
- The *APT HOWTO*[5] and *apt-get man page*[6] contain useful information regarding apt-get usage.
- See the *aptitude man page*[7] for more aptitude options.
- La pagina *riguardo i repository*[8] della documentazione italiana, contiene maggiori informazioni su come aggiungere repository.

[3] https://help.ubuntu.com/community/InstallingSoftware
[4] http://manpages.ubuntu.com/manpages/precise/en/man1/dpkg.1.html
[5] http://www.debian.org/doc/manuals/apt-howto/
[6] http://manpages.ubuntu.com/manpages/precise/en/man8/apt-get.8.html
[7] http://manpages.ubuntu.com/manpages/precise/man8/aptitude.8.html
[8] http://wiki.ubuntu-it.org/Repository

Capitolo 4. Rete

Le reti consistono in due o più dispositivi, come computer, stampanti e altri equipaggiamenti correlati, connessi tramite un cavo fisico oppure tramite collegamenti senza fili, con lo scopo di condividere e distribuire informazioni tra di loro.

Questa sezione fornisce informazioni generali e specifiche sulle reti (creare, modificare e gestire reti), compresa una panoramica sui concetti delle reti e discussioni dettagliate dei più comuni protocolli di rete.

1. Configurare la rete

Ubuntu è corredato da una serie di utilità grafiche per la configurazione dei dispositivi di rete. Questa sezione è diretta agli amministratori di server e si focalizza sulla gestione della rete da riga di comando.

1.1. Ethernet Interfaces

Ethernet interfaces are identified by the system using the naming convention of *ethX*, where *X* represents a numeric value. The first Ethernet interface is typically identified as *eth0*, the second as *eth1*, and all others should move up in numerical order.

1.1.1. Identify Ethernet Interfaces

To quickly identify all available Ethernet interfaces, you can use the ifconfig command as shown below.

```
ifconfig -a | grep eth
eth0      Link encap:Ethernet  HWaddr 00:15:c5:4a:16:5a
```

Another application that can help identify all network interfaces available to your system is the lshw command. In the example below, lshw shows a single Ethernet interface with the logical name of *eth0* along with bus information, driver details and all supported capabilities.

```
sudo lshw -class network
  *-network
       description: Ethernet interface
       product: BCM4401-B0 100Base-TX
       vendor: Broadcom Corporation
       physical id: 0
       bus info: pci@0000:03:00.0
       logical name: eth0
       version: 02
       serial: 00:15:c5:4a:16:5a
       size: 10MB/s
       capacity: 100MB/s
       width: 32 bits
       clock: 33MHz
       capabilities: (snipped for brevity)
       configuration: (snipped for brevity)
       resources: irq:17 memory:ef9fe000-ef9fffff
```

1.1.2. Ethernet Interface Logical Names

Interface logical names are configured in the file `/etc/udev/rules.d/70-persistent-net.rules`. If you would like control which interface receives a particular logical name, find the line matching the interfaces physical MAC address and modify the value of *NAME=ethX* to the desired logical name. Reboot the system to commit your changes.

1.1.3. Ethernet Interface Settings

ethtool is a program that displays and changes Ethernet card settings such as auto-negotiation, port speed, duplex mode, and Wake-on-LAN. It is not installed by default, but is available for installation in the repositories.

```
sudo apt-get install ethtool
```

The following is an example of how to view supported features and configured settings of an Ethernet interface.

```
sudo ethtool eth0
Settings for eth0:
        Supported ports: [ TP ]
        Supported link modes:   10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Half 1000baseT/Full
        Supports auto-negotiation: Yes
        Advertised link modes:  10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Half 1000baseT/Full
        Advertised auto-negotiation: Yes
        Speed: 1000Mb/s
        Duplex: Full
        Port: Twisted Pair
        PHYAD: 1
        Transceiver: internal
        Auto-negotiation: on
        Supports Wake-on: g
        Wake-on: d
        Current message level: 0x000000ff (255)
        Link detected: yes
```

Changes made with the ethtool command are temporary and will be lost after a reboot. If you would like to retain settings, simply add the desired ethtool command to a *pre-up* statement in the interface configuration file `/etc/network/interfaces`.

The following is an example of how the interface identified as *eth0* could be permanently configured with a port speed of 1000Mb/s running in full duplex mode.

```
auto eth0
iface eth0 inet static
pre-up /sbin/ethtool -s eth0 speed 1000 duplex full
```

Although the example above shows the interface configured to use the *static* method, it actually works with other methods as well, such as DHCP. The example is meant to demonstrate only proper placement of the *pre-up* statement in relation to the rest of the interface configuration.

1.2. IP Addressing

The following section describes the process of configuring your systems IP address and default gateway needed for communicating on a local area network and the Internet.

1.2.1. Temporary IP Address Assignment

For temporary network configurations, you can use standard commands such as ip, ifconfig and route, which are also found on most other GNU/Linux operating systems. These commands allow you to configure settings which take effect immediately, however they are not persistent and will be lost after a reboot.

To temporarily configure an IP address, you can use the ifconfig command in the following manner. Just modify the IP address and subnet mask to match your network requirements.

```
sudo ifconfig eth0 10.0.0.100 netmask 255.255.255.0
```

To verify the IP address configuration of eth0, you can use the ifconfig command in the following manner.

```
ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:15:c5:4a:16:5a
          inet addr:10.0.0.100  Bcast:10.0.0.255  Mask:255.255.255.0
          inet6 addr: fe80::215:c5ff:fe4a:165a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:466475604 errors:0 dropped:0 overruns:0 frame:0
          TX packets:403172654 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2574778386 (2.5 GB)  TX bytes:1618367329 (1.6 GB)
          Interrupt:16
```

To configure a default gateway, you can use the route command in the following manner. Modify the default gateway address to match your network requirements.

```
sudo route add default gw 10.0.0.1 eth0
```

To verify your default gateway configuration, you can use the route command in the following manner.

```
route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        0.0.0.0         255.255.255.0   U     1      0        0 eth0
0.0.0.0         10.0.0.1        0.0.0.0         UG    0      0        0 eth0
```

If you require DNS for your temporary network configuration, you can add DNS server IP addresses in the file `/etc/resolv.conf`. The example below shows how to enter two DNS servers to `/etc/`

`resolv.conf`, which should be changed to servers appropriate for your network. A more lengthy description of DNS client configuration is in a following section.

```
nameserver 8.8.8.8
nameserver 8.8.4.4
```

If you no longer need this configuration and wish to purge all IP configuration from an interface, you can use the ip command with the flush option as shown below.

```
ip addr flush eth0
```

Flushing the IP configuration using the ip command does not clear the contents of `/etc/resolv.conf`. You must remove or modify those entries manually.

1.2.2. Dynamic IP Address Assignment (DHCP Client)

To configure your server to use DHCP for dynamic address assignment, add the *dhcp* method to the inet address family statement for the appropriate interface in the file `/etc/network/interfaces`. The example below assumes you are configuring your first Ethernet interface identified as *eth0*.

```
auto eth0
iface eth0 inet dhcp
```

By adding an interface configuration as shown above, you can manually enable the interface through the ifup command which initiates the DHCP process via dhclient.

```
sudo ifup eth0
```

To manually disable the interface, you can use the ifdown command, which in turn will initiate the DHCP release process and shut down the interface.

```
sudo ifdown eth0
```

1.2.3. Static IP Address Assignment

To configure your system to use a static IP address assignment, add the *static* method to the inet address family statement for the appropriate interface in the file `/etc/network/interfaces`. The example below assumes you are configuring your first Ethernet interface identified as *eth0*. Change the *address*, *netmask*, and *gateway* values to meet the requirements of your network.

```
auto eth0
iface eth0 inet static
address 10.0.0.100
netmask 255.255.255.0
gateway 10.0.0.1
```

By adding an interface configuration as shown above, you can manually enable the interface through the ifup command.

```
sudo ifup eth0
```

To manually disable the interface, you can use the ifdown command.

```
sudo ifdown eth0
```

1.2.4. Loopback Interface

The loopback interface is identified by the system as *lo* and has a default IP address of 127.0.0.1. It can be viewed using the ifconfig command.

```
ifconfig lo
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:2718 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2718 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:183308 (183.3 KB)  TX bytes:183308 (183.3 KB)
```

By default, there should be two lines in `/etc/network/interfaces` responsible for automatically configuring your loopback interface. It is recommended that you keep the default settings unless you have a specific purpose for changing them. An example of the two default lines are shown below.

```
auto lo
iface lo inet loopback
```

1.3. Name Resolution

Name resolution as it relates to IP networking is the process of mapping IP addresses to hostnames, making it easier to identify resources on a network. The following section will explain how to properly configure your system for name resolution using DNS and static hostname records.

1.3.1. DNS Client Configuration

Traditionally, the file `/etc/resolv.conf` was a static configuration file that rarely needed to be changed or automatically changed via DCHP client hooks. Nowadays, a computer can switch from one network to another quite often and the *resolvconf* framework is now being used to track these changes and update the resolver's configuration automatically. It acts as an intermediary between programs that supply nameserver information and applications that need nameserver information. Resolvconf gets populated with information by a set of hook scripts related to network interface configuration. The most notable difference for the user is that any change manually done to /etc/resolv.conf will be lost as it gets overwritten each time something triggers resolvconf. Instead,

resolvconf uses DHCP client hooks, and `/etc/network/interfaces` to generate a list of nameservers and domains to put in /etc/resolv.conf, which is now a symlink:

```
/etc/resolv.conf -> ../run/resolvconf/resolv.conf
```

To configure the resolver, add the IP addresses of the nameservers that are appropriate for your network in the file `/etc/network/interfaces`. You can also add an optional DNS suffix search-lists to match your network domain names. For each other valid resolv.conf configuration option, you can include, in the stanza, one line beginning with that option name with a **dns-** prefix. The resulting file might look like the following:

```
iface eth0 inet static
    address 192.168.3.3
    netmask 255.255.255.0
    gateway 192.168.3.1
    dns-search example.com
    dns-nameservers 192.168.3.45 192.168.8.10
```

The *search* option can also be used with multiple domain names so that DNS queries will be appended in the order in which they are entered. For example, your network may have multiple sub-domains to search; a parent domain of *example.com*, and two sub-domains, *sales.example.com* and *dev.example.com*.

If you have multiple domains you wish to search, your configuration might look like the following:

```
iface eth0 inet static
    address 192.168.3.3
    netmask 255.255.255.0
    gateway 192.168.3.1
    dns-search example.com sales.example.com dev.example.com
    dns-nameservers 192.168.3.45 192.168.8.10
```

If you try to ping a host with the name of *server1*, your system will automatically query DNS for its Fully Qualified Domain Name (FQDN) in the following order:

1. server1**.example.com**
2. server1**.sales.example.com**
3. server1**.dev.example.com**

If no matches are found, the DNS server will provide a result of *notfound* and the DNS query will fail.

1.3.2. Static Hostnames

Static hostnames are locally defined hostname-to-IP mappings located in the file `/etc/hosts`. Entries in the `hosts` file will have precedence over DNS by default. This means that if your system tries to resolve a hostname and it matches an entry in /etc/hosts, it will not attempt to look up the record in DNS. In some configurations, especially when Internet access is not required, servers that

communicate with a limited number of resources can be conveniently set to use static hostnames instead of DNS.

The following is an example of a `hosts` file where a number of local servers have been identified by simple hostnames, aliases and their equivalent Fully Qualified Domain Names (FQDN's).

```
127.0.0.1 localhost
127.0.1.1 ubuntu-server
10.0.0.11 server1 vpn server1.example.com
10.0.0.12 server2 mail server2.example.com
10.0.0.13 server3 www server3.example.com
10.0.0.14 server4 file server4.example.com
```

In the above example, notice that each of the servers have been given aliases in addition to their proper names and FQDN's. *Server1* has been mapped to the name *vpn*, *server2* is referred to as *mail*, *server3* as *www*, and *server4* as *file*.

1.3.3. Name Service Switch Configuration

The order in which your system selects a method of resolving hostnames to IP addresses is controlled by the Name Service Switch (NSS) configuration file `/etc/nsswitch.conf`. As mentioned in the previous section, typically static hostnames defined in the systems `/etc/hosts` file have precedence over names resolved from DNS. The following is an example of the line responsible for this order of hostname lookups in the file `/etc/nsswitch.conf`.

```
hosts:          files mdns4_minimal [NOTFOUND=return] dns mdns4
```

- **files** first tries to resolve static hostnames located in `/etc/hosts`.
- **mdns4_minimal** attempts to resolve the name using Multicast DNS.
- **[NOTFOUND=return]** means that any response of *notfound* by the preceding *mdns4_minimal* process should be treated as authoritative and that the system should not try to continue hunting for an answer.
- **dns** represents a legacy unicast DNS query.
- **mdns4** represents a Multicast DNS query.

To modify the order of the above mentioned name resolution methods, you can simply change the *hosts:* string to the value of your choosing. For example, if you prefer to use legacy Unicast DNS versus Multicast DNS, you can change the string in `/etc/nsswitch.conf` as shown below.

```
hosts:          files dns [NOTFOUND=return] mdns4_minimal mdns4
```

1.4. Bridging

Il "bridging" di molteplici interfacce è una configurazione avanzata, ma utile in diversi scenari. Uno di questi scenari può consistere nel configurare un bridge con molteplici interfacce di rete e usare

un firewall per filtrare il traffico tra due segmenti della rete. Un altro scenario consiste nell'usare un bridge su un sistema con una sola interfaccia per permettere alle macchine virtuali accesso diretto alla rete esterna. L'esempio che segue prende in considerazione quest'ultimo scenario.

Prima di configurare un bridge è necessario installare il pacchetto bridge-utils. In un terminale digitare:

```
sudo apt-get install bridge-utils
```

Configurare il bridge modificando il file `/etc/network/interfaces`:

```
auto lo
iface lo inet loopback

auto br0
iface br0 inet static
        address 192.168.0.10
        network 192.168.0.0
        netmask 255.255.255.0
        broadcast 192.168.0.255
        gateway 192.168.0.1
        bridge_ports eth0
        bridge_fd 9
        bridge_hello 2
        bridge_maxage 12
        bridge_stp off
```

Inserire i valori appropriati per la propria interfaccia di rete.

Riavviare la rete per abilitare il bridge sull'interfaccia:

```
sudo /etc/init.d/networking restart
```

La nuova interfaccia dovrebbe ora essere funzionante. L'applicazione brctl fornisce utili informazioni riguardo lo stato del bridge, controlla le interfacce che compongono il bridge, ecc... Per maggiori informazioni, consultare la pagina di manuale: **man brctl**.

1.5. Risorse

- The *Ubuntu Wiki Network page*[1] has links to articles covering more advanced network configuration.
- The *resolvconf man page*[2] has more information on resolvconf.

[1] https://help.ubuntu.com/community/Network
[2] http://manpages.ubuntu.com/manpages/man8/resolvconf.8.html

- The *interfaces man page*[3] has details on more options for `/etc/network/interfaces`.
- The *dhclient man page*[4] has details on more options for configuring DHCP client settings.
- For more information on DNS client configuration see the *resolver man page*[5]. Also, Chapter 6 of O'Reilly's *Linux Network Administrator's Guide*[6] is a good source of resolver and name service configuration information.
- For more information on *bridging* see the *brctl man page*[7] and the Linux Foundation's *Net:Bridge*[8] page.

[3] http://manpages.ubuntu.com/manpages/man5/interfaces.5.html
[4] http://manpages.ubuntu.com/manpages/man8/dhclient.8.html
[5] http://manpages.ubuntu.com/manpages/man5/resolver.5.html
[6] http://oreilly.com/catalog/linag2/book/ch06.html
[7] http://manpages.ubuntu.com/manpages/man8/brctl.8.html
[8] http://www.linuxfoundation.org/en/Net:Bridge

2. TCP/IP

Il protocollo TCP/IP (Transmission Control Protocol e Internet Protocol) è un insieme standard di protocolli sviluppato nella seconda metà degli anni '70 dalla DARPA (Defence Advanced Research Project Agency) con lo scopo di permettere la comunicazione tra diversi tipi di computer e di reti di computer. TCP/IP è il motore di Internet, ecco perché è l'insieme di protocolli di rete più diffuso al mondo.

2.1. Introduzione a TCP/IP

I due protocolli che compongono TCP/IP, interagiscono con differenti aspetti di una rete. L'*Internet Protocol*, la parte "IP" di TCP/IP, è un protocollo privo di connessione che interagisce solamente con il routing dei pacchetti attraverso la rete, usando l'*IP Datagram* come unità di base delle informazioni che consiste in un'intestazione seguita da un messaggio. Il *Transmission Control Protocol*, la parte "TCP" di TCP/IP, consente agli host della rete di stabilire le connessioni che possono essere usate per scambiare dati. Inoltre, garantisce che i dati tra le connessioni siano consegnati correttamente e nello stesso ordine in cui sono stati inviati.

2.2. Configurazione di TCP/IP

La configurazione del protocollo TCP/IP è composta da vari elementi che debbono essere impostati modificando gli appropriati file di configurazione oppure adottando soluzioni quali un server DHCP (Dynamic Host Configuration Protocol); tale server provvede ad assegnare automaticamente le corrette impostazioni di configurazione TCP/IP ai client della rete. Questi valori di configurazione debbono essere impostati correttamente per consentire al sistema Ubuntu di operare adeguatamente in rete.

I tipici elementi di configurazione di TCP/IP e i loro scopi sono i seguenti:

- **Indirizzo IP**: l'indirizzo IP è una stringa d'identificazione unica, espressa da quattro numeri decimali compresi tra zero (0) e duecentocinquantacinque (255), separati da punti; ciascuno dei quattro numeri rappresenta otto (8) bit dell'indirizzo per una lunghezza totale di trentadue (32) bit per l'indirizzo completo. Questo formato è detto *notazione decimale a punti*.
- **Maschera di rete**: la maschera di rete (o semplicemente *netmask*) è una maschera locale di bit, ovvero un insieme di indicatori che separano la porzione di un indirizzo IP che indica la rete dai bit che indicano la *sotto-rete*. Ad esempio, in una rete di classe C, la maschera di rete standard è 255.255.255.0 che serve a mascherare i primi tre byte dell'indirizzo IP, consentendo all'ultimo byte dell'indirizzo IP di essere disponibile per specificare gli host della sotto-rete.
- **Indirizzo di rete**: l'indirizzo di rete è dato dai byte che comprendono la parte di rete di un indirizzo IP. Per esempio, l'host 12.128.1.2 in una rete di classe A deve usare 12.0.0.0 come indirizzo di rete, dove dodici (12) è il primo byte dell'indirizzo IP (la parte di rete), e gli zeri (0) nei rimanenti tre byte indicano tutti i possibili valori degli host. Un host di rete che ha un indirizzo IP 192.168.1.100 deve invece usare un indirizzo di rete di 192.168.1.0, nel quale i primi tre byte specificano la rete di classe C 192.168.2 e lo zero (0) per tutti i possibili valori degli host nella rete.

- **Indirizzo di broadcast**: l'indirizzo di broadcast è un indirizzo IP che permette di inviare dei dati di rete simultaneamente a tutti gli host di una data sotto-rete piuttosto che a uno specifico host. L'indirizzo broadcast generale di base per una rete IP è 255.255.255.255, ma questo indirizzo broadcast non può essere usato per inviare un messaggio broadcast a tutti gli host presenti in internet perché i router lo bloccherebbero. Un indirizzo di broadcast appropriato è quello che indica una specifica sotto-rete. Per esempio, in una rete privata di classe C, 192.168.1.0, l'indirizzo broadcast è 192.168.1.255. I messaggi broadcast sono di norma prodotti dai protocolli di rete come il protocollo per la risoluzione degli indirizzi (ARP, Address Resolution Protocol) e il protocollo delle informazioni di instradamento (RIP, Routing Information Protocol).
- **Indirizzo del gateway**: l'indirizzo del gateway è l'indirizzo IP attraverso il quale una particolare rete, o un host su una rete, può essere raggiunta. Se un host di rete desidera comunicare con un altro host di rete, senza essere localizzato nelle stessa rete, allora deve essere usato un *gateway*. In molti casi l'indirizzo del gateway coincide con quello di un router della medesima rete che ha il compito di far transitare il traffico ad altre reti o host, come Internet. L'impostazione del valore dell'indirizzo del gateway deve essere corretta, altrimenti il sistema non è in grado di raggiungere gli host che non si trovano sulla rete cui appartiene.
- **Indirizzo del server dei nomi**: l'indirizzo del server dei nomi rappresenta l'indirizzo IP del sistema DNS (Domain Name Service) che traduce il nome host della rete in un indirizzo IP reale. Esistono tre livelli di indirizzo del server dei nomi che possono essere specificati in ordine di precedenza: il server dei nomi *primario*, quello *secondario* e il *terziario*. Affinché il sistema possa tradurre i nomi host in indirizzi IP, è necessario specificare degli indirizzi validi per i server dei nomi che è possibile utilizzare all'interno della configurazione TCP/IP del sistema. Nella maggior parte dei casi, questi indirizzi vengono forniti dal proprio fornitore di servizio Internet, ma ne sono disponibili anche di gratuiti e liberamente utilizzabili, come i server di terzo livello di Verizon con indirizzi IP da 4.2.2.1 a 4.2.2.6.

Gli indirizzi IP, le maschere di rete, gli indirizzi di rete, gli indirizzi di broadcast e gli indirizzi di gateway sono tipicamente determinati attraverso appropriate direttive nel file `/etc/network/interfaces`. Gli indirizzi di server dei nomi sono tipicamente specificati attraverso le direttive *nameserver* nel file `/etc/resolv.conf`. Per maggiori informazioni, consultare rispettivamente le pagine di manuale di sistema per `interfaces` e `resolv.conf`, usando i seguenti comandi da digitare al prompt di un terminale:

Accedere alla pagina di manuale di sistema per `interfaces` con il seguente comando:

```
man interfaces
```

Accedere alla pagina di manuale di sistema per `resolv.conf` con il seguente comando:

```
man resolv.conf
```

2.3. Instradamento IP

L'instradamento IP è un modo per indicare e scoprire percorsi in una rete TCP/IP attraverso i quali inviare dati. L'instradamento utilizza un insieme di *tabelle di instradamento (routing)* per dirigere i pacchetti di dati in una rete dalla loro sorgente avanti fino alla destinazione, spesso attraverso molti nodi di rete intermediari chiamati *router*. Esistono due forme primarie di instradamento IP: l'*instradamento statico* e l'*instradamento dinamico*.

L'instradamento statico comporta l'aggiunta manuale di rotte IP nella tabella di instradamento del sistema, attività che viene fatta modificando la tabella di instradamento con il comando route. L'instradamento statico presenta molti vantaggi rispetto quello dinamico, come la semplicità di implementazione per piccole reti, la predicibilità (la tabella di instradamento è scritta a priori, quindi la rotta è sempre la stessa ogni volta che viene utilizzata) e il basso carico di lavoro sugli altri router e nodi di rete dovuto all'assenza di un protocollo di instradamento dinamico. In ogni caso, l'instradamento statico presenta anche degli svantaggi. Per esempio, è limitato a piccole reti e non è facilmente espandibile. L'instradamento statico fallisce completamente se si prova ad adattarlo ai ritardi della rete e le perdite lungo la rotta per la natura statica della rotta stessa.

L'instradamento dinamico serve nelle grandi reti con molte possibili rotte IP tra una sorgente e una destinazione. Fa uso di protocolli di instradamento speciali, come il protocollo di informazione dell'instradamento (RIP, Router Information Protocol) che gestisce le correzioni automatiche nella tabella di instradamento rendendo possibile l'instradamento dinamico. Ci sono molti vantaggi rispetto l'instradamento statico, come l'adattamento alle dimensioni superiori e l'abilità di adattarsi agli errori e alle perdite lungo le rotte della rete. Inoltre, necessita di una minore configurazione manuale delle tabelle di instradamento, dato che i router comunicano tra di loro la relativa esistenza e le possibili rotte. Questo tratto caratteristico elimina anche la possibilità di introdurre inesattezze nelle tabelle di instradamento causate da errori umani. In ogni caso, l'instradamento dinamico non è perfetto e presenta alcuni svantaggi come, l'aumento della complessità e del carico di lavoro dovuto alle comunicazioni dei router della rete, dei quali non può beneficiare subito l'utente finale che comunque consuma banda di rete.

2.4. TCP e UDP

TCP è un protocollo basato sulla connessione, che offre correzione d'errore e che garantisce la consegna dei dati attraverso ciò che è conosciuto come *controllo di flusso*. Il controllo di flusso determina quando il flusso di uno stream di dati debba essere fermato e i pacchetti di dati inviati in precedenza debbano essere reinviati a causa di problemi come *collisioni*, assicurando quindi la completa e accurata consegna dei dati. TCP è tipicamente usato nello scambio di informazioni importanti come transazioni di database.

UDP (User Datagram Protocol), al contrario, è un protocollo *senza connessione* che raramente tratta della trasmissione dei dati importanti a causa della mancanza del controllo di flusso o di un altro metodo che garantisca la consegna affidabile dei dati. UDP è normalmente usato in applicazioni come lo streaming audio e video, in cui risulta considerevolmente più veloce del protocollo TCP, data la

mancanza di correzione d'errore e del controllo di flusso, e in cui la perdita di alcuni pacchetti non è generalmente un evento catastrofico.

2.5. ICPM

ICMP (Internet Control Messaging Protocol) è un'estensione di IP (Internet Protocol), come definito nell'RFC (Request For Comments) numero 792; ICPM supporta pacchetti di rete contenenti messaggi di controllo, di errore e di informazione. ICMP è usato da applicazioni di rete come l'utilità ping, che consente di determinare la disponibilità di un host o un'interfaccia di rete. Esempi di alcuni dei messaggi di errore restituiti da ICMP utili sia agli host e interfacce di rete che ai router sono *Destination Unreachable* e *Time Exceeded*.

2.6. Demoni

I demoni sono speciali applicazioni di sistema che, tipicamente, sono in continua esecuzione sullo sfondo, attendendo dagli altri programmi richieste relative a funzioni da essi fornite. Molti demoni hanno a che fare con la rete e molti di questi in esecuzione sullo sfondo nei sistemi Ubuntu forniscono delle funzionalità legate alla rete. Alcuni esempi di questi demoni di rete includono *httpd* (Hyper Text Transport Protocol Daemon), che fornisce funzionalità di server web; *sshd* (Secure SHell Daemon), che fornisce funzionalità di login e trasferimento file sicuro da remoto; *imapd* (Internet Message Access Protocol Daemon), che fornisce servizi di email.

2.7. Risorse

- There are man pages for *TCP*[9] and *IP*[10] that contain more useful information.
- Inoltre, consultare il RedBook di IBM: *TCP/IP Tutorial and Technical Overview*[11].
- Un'altra utile risorsa è il libro *TCP/IP Network Administration*[12].

[9] http://manpages.ubuntu.com/manpages/precise/en/man7/tcp.7.html
[10] http://manpages.ubuntu.com/manpages/precise/man7/ip.7.html
[11] http://www.redbooks.ibm.com/abstracts/gg243376.html
[12] http://oreilly.com/catalog/9780596002978/

3. DHCP (Dynamic Host Configuration Protocol)

Il DHCP (Dynamic Host Configuration Protocol) è un servizio di rete che consente di assegnare automaticamente le impostazioni per agli host da un server, senza la necessità di dover configurare manualmente ogni singolo host nella rete. I computer configurati per essere client DHCP non hanno alcun controllo sulle impostazioni che ricevono dal server DHCP e la configurazione è trasparente all'utente del computer.

Le impostazioni comuni fornite da un server DHCP a un client includono:

- IP address and netmask
- IP address of the default-gateway to use
- IP adresses of the DNS servers to use

Un server DHCP può fornire anche altre proprietà di configurazione come:

- Nome dell'host
- Nome del dominio
- Server NTP (Network Time Protocol)
- Server di stampa

Il vantaggio di utilizzare DHCP è che i cambiamenti apportati alla rete, per esempio una modifica dell'indirizzo del server DNS, devono essere apportati solamente al server DHCP, mentre tutti gli host della rete vengono riconfigurati quando i client DHCP interrogano il server DHCP. Come ulteriore vantaggio, risulta anche molto semplice integrare nuovi computer nella rete, senza la necessità di controllare la disponibilità di un indirizzo IP. I conflitti nell'allocazione degli indirizzi IP sono quindi notevolmente ridotti.

A DHCP server can provide configuration settings using the following methods:

Manual allocation (MAC address)
: This method entails using DHCP to identify the unique hardware address of each network card connected to the network and then continually supplying a constant configuration each time the DHCP client makes a request to the DHCP server using that network device. This ensures that a particular address is assigned automatically to that network card, based on it's MAC address.

Dynamic allocation (address pool)
: In this method, the DHCP server will assign an IP address from a pool of addresses (sometimes also called a range or scope) for a period of time or lease, that is configured on the server or until the client informs the server that it doesn't need the address anymore. This way, the clients will be receiving their configuration properties dynamically and on a "first come, first served" basis. When a DHCP client is no longer on the network for a specified period, the configuration is expired and released back to the address pool for use by other DHCP Clients. This way, an address cand be leased or used for a period of time. After this period, the client has to renegociate the lease with the server to maintain use of the address.

Automatic allocation
: Using this method, the DHCP automatically assigns an IP address permanently to a device, selecting it from a pool of available addresses. Usually DHCP is used to assign a temporary address to a client, but a DHCP server can allow an infinite lease time.

The last two methods can be considered “automatic” because in each case the DHCP server assigns an address with no extra intervention needed. The only difference between them is in how long the IP address is leased, in other words whether a client's address varies over time. Ubuntu is shipped with both DHCP server and client. The server is dhcpd (dynamic host configuration protocol daemon). The client provided with Ubuntu is dhclient and should be installed on all computers required to be automatically configured. Both programs are easy to install and configure and will be automatically started at system boot.

3.1. Installazione

A un prompt di terminale, inserire il seguente comando per installare dhcpd:

```
sudo apt-get install isc-dhcp-server
```

You will probably need to change the default configuration by editing /etc/dhcp/dhcpd.conf to suit your needs and particular configuration.

You also may need to edit /etc/default/isc-dhcp-server to specify the interfaces dhcpd should listen to.

I messaggi di dhcpd vengono inviati nel syslog, consultare quindi i relativi messaggi per quelli di diagnostica.

3.2. Configurazione

Il messaggio di errore con cui si conclude l'installazione potrebbe essere fuorviante, ma i passi seguenti consentono di configurare il servizio.

Nella maggior parte dei casi si vuole assegnare un indirizzo IP in modo casuale. Questo può essere ottenuto con impostazioni come le seguenti:

```
# minimal sample /etc/dhcp/dhcpd.conf
default-lease-time 600;
max-lease-time 7200;

subnet 192.168.1.0 netmask 255.255.255.0 {
 range 192.168.1.150 192.168.1.200;
 option routers 192.168.1.254;
 option domain-name-servers 192.168.1.1, 192.168.1.2;
 option domain-name "mydomain.example";
}
```

This will result in the DHCP server giving clients an IP address from the range 192.168.1.150-192.168.1.200. It will lease an IP address for 600 seconds if the client doesn't ask for a specific time frame. Otherwise the maximum (allowed) lease will be 7200 seconds. The server will also "advise" the client to use 192.168.1.254 as the default-gateway and 192.168.1.1 and 192.168.1.2 as its DNS servers.

After changing the config file you have to restart the dhcpd:

```
sudo /etc/init.d/isc-dhcp-server restart
```

3.3. Riferimenti

- The *dhcp3-server Ubuntu Wiki*[13] page has more information.
- For more `/etc/dhcp/dhcpd.conf` options see the *dhcpd.conf man page*[14].
- *ISC dhcp-server*[15]

[13] https://help.ubuntu.com/community/dhcp3-server

[14] http://manpages.ubuntu.com/manpages/precise/en/man5/dhcpd.conf.5.html

[15] http://www.isc.org/software/dhcp

4. Sincronizzazione del tempo con NTP

NTP è un protocollo TCP/IP per sincronizzare l'ora attraverso la rete: un client richiede l'ora corrente a un server e usa questa per impostare il proprio orologio.

Behind this simple description, there is a lot of complexity - there are tiers of NTP servers, with the tier one NTP servers connected to atomic clocks, and tier two and three servers spreading the load of actually handling requests across the Internet. Also the client software is a lot more complex than you might think - it has to factor out communication delays, and adjust the time in a way that does not upset all the other processes that run on the server. But luckily all that complexity is hidden from you!

Ubuntu uses ntpdate and ntpd.

4.1. ntpdate

Ubuntu comes with ntpdate as standard, and will run it once at boot time to set up your time according to Ubuntu's NTP server.

```
ntpdate -s ntp.ubuntu.com
```

4.2. ntpd

The ntp daemon ntpd calculates the drift of your system clock and continuously adjusts it, so there are no large corrections that could lead to inconsistent logs for instance. The cost is a little processing power and memory, but for a modern server this is negligible.

4.3. Installazione

To install ntpd, from a terminal prompt enter:

```
sudo apt-get install ntp
```

4.4. Configurazione

Edit `/etc/ntp.conf` to add/remove server lines. By default these servers are configured:

```
# Use servers from the NTP Pool Project. Approved by Ubuntu Technical Board
# on 2011-02-08 (LP: #104525). See http://www.pool.ntp.org/join.html for
# more information.
server 0.ubuntu.pool.ntp.org
server 1.ubuntu.pool.ntp.org
server 2.ubuntu.pool.ntp.org
server 3.ubuntu.pool.ntp.org
```

After changing the config file you have to reload the ntpd:

```
sudo /etc/init.d/ntp reload
```

4.5. View status

Use ntpq to see to see more info:

```
# sudo ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
+stratum2-2.NTP. 129.70.130.70    2 u    5   64  377   68.461  -44.274 110.334
+ntp2.m-online.n 212.18.1.106     2 u    5   64  377   54.629  -27.318  78.882
*145.253.66.170  .DCFa.           1 u   10   64  377   83.607  -30.159  68.343
+stratum2-3.NTP. 129.70.130.70    2 u    5   64  357   68.795  -68.168 104.612
+europium.canoni 193.79.237.14    2 u   63   64  337   81.534  -67.968  92.792
```

4.6. Riferimenti

- See the *Ubuntu Time*[16] wiki page for more information.
- *ntp.org, home of the Network Time Protocol project*[17]

[16] https://help.ubuntu.com/community/UbuntuTime

[17] http://www.ntp.org/

Capitolo 5. DM-Multipath

1. Device Mapper Multipathing

Device mapper multipathing (DM-Multipath) allows you to configure multiple I/O paths between server nodes and storage arrays into a single device. These I/O paths are physical SAN connections that can include separate cables, switches, and controllers. Multipathing aggregates the I/O paths, creating a new device that consists of the aggregated paths. This chapter provides a summary of the features of DM-Multipath that are new for the initial release of Ubuntu Server 12.04. Following that, this chapter provides a high-level overview of DM Multipath and its components, as well as an overview of DM-Multipath setup.

1.1. New and Changed Features for Ubuntu Server 12.04

Migrated from multipath-0.4.8 to multipath-0.4.9

1.1.1. Migration from 0.4.8

The priority checkers are no longer run as standalone binaries, but as shared libraries. The key value name for this feature has also slightly changed. Copy the attribute named **prio_callout** to **prio**, also modify the argument the name of the priority checker, a system path is no longer necessary. Example conversion:

```
device {
        vendor "NEC"
        product "DISK ARRAY"
        prio_callout mpath_prio_alua /dev/%n
        prio    alua
}
```

See Table *Priority Checker Conversion [54]* for a complete listing

Tabella 5.1. Priority Checker Conversion

v0.4.8	v0.4.9
prio_callout mpath_prio_emc /dev/%n	**prio emc**
prio_callout mpath_prio_alua /dev/%n	**prio alua**
prio_callout mpath_prio_netapp /dev/%n	**prio netapp**
prio_callout mpath_prio_rdac /dev/%n	**prio rdac**
prio_callout mpath_prio_hp_sw /dev/%n	**prio hp_sw**
prio_callout mpath_prio_hds_modular %b	**prio hds**

Since the multipath config file parser essentially parses all key/value pairs it finds and then makes use of them, it is safe for both **prio_callout** and **prio** to coexist and is recommended that the **prio** attribute be inserted before beginning migration. After which you can safely delete the legacy **prio_calliout** attribute without interrupting service.

1.2. Panoramica

DM-Multipath can be used to provide:

- *Redundancy* DM-Multipath can provide failover in an active/passive configuration. In an active/passive configuration, only half the paths are used at any time for I/O. If any element of an I/O path (the cable, switch, or controller) fails, DM-Multipath switches to an alternate path.
- *Improved Performance* Performance DM-Multipath can be configured in active/active mode, where I/O is spread over the paths in a round-robin fashion. In some configurations, DM-Multipath can detect loading on the I/O paths and dynamically re-balance the load.

1.3. Storage Array Overview

By default, DM-Multipath includes support for the most common storage arrays that support DM-Multipath. The supported devices can be found in the multipath.conf.defaults file. If your storage array supports DM-Multipath and is not configured by default in this file, you may need to add them to the DM-Multipath configuration file, multipath.conf. For information on the DM-Multipath configuration file, see Section, *The DM-Multipath Configuration File*. Some storage arrays require special handling of I/O errors and path switching. These require separate hardware handler kernel modules.

1.4. DM-Multipath components

Table *"DM-Multipath Components"* describes the components of the DM-Multipath package.
<include></include>

1.5. DM-Multipath Setup Overview

DM-Multipath includes compiled-in default settings that are suitable for common multipath configurations. Setting up DM-multipath is often a simple procedure. The basic procedure for configuring your system with DM-Multipath is as follows:

1. Install the **multipath-tools** and **multipath-tools-boot** packages
2. Create an empty config file, `/etc/multipath.conf`, that re-defines the *following*
3. If necessary, edit the **multipath.conf** configuration file to modify default values and save the updated file.
4. Start the multipath daemon
5. Update initial ramdisk

For detailed setup instructions for multipath configuration see Section, *Setting Up DM-Multipath*.

2. Multipath Devices

Without DM-Multipath, each path from a server node to a storage controller is treated by the system as a separate device, even when the I/O path connects the same server node to the same storage controller. DM-Multipath provides a way of organizing the I/O paths logically, by creating a single multipath device on top of the underlying devices.

2.1. Multipath Device Identifiers

Each multipath device has a World Wide Identifier (WWID), which is guaranteed to be globally unique and unchanging. By default, the name of a multipath device is set to its WWID. Alternately, you can set the ***user_friendly_names*** option in the multipath configuration file, which causes DM-Multipath to use a node-unique alias of the form **mpathn** as the name. For example, a node with two HBAs attached to a storage controller with two ports via a single unzoned FC switch sees four devices: **/dev/sda**, **/dev/sdb**, **/dev/sdc**, and **/dev/sdd**. DM-Multipath creates a single device with a unique WWID that reroutes I/O to those four underlying devices according to the multipath configuration. When the ***user_friendly_names*** configuration option is set to **yes**, the name of the multipath device is set to **mpathn**. When new devices are brought under the control of DM-Multipath, the new devices may be seen in two different places under the **/dev** directory: **/dev/mapper/mpathn** and **/dev/dm-n**.

- The devices in **/dev/mapper** are created early in the boot process. Use these devices to access the multipathed devices, for example when creating logical volumes.
- Any devices of the form **/dev/dm-n** are for internal use only and should never be used.

For information on the multipath configuration defaults, including the ***user_friendly_names*** configuration option, see Section , *"Configuration File Defaults"*. You can also set the name of a multipath device to a name of your choosing by using the ***alias*** option in the **multipaths** section of the multipath configuration file. For information on the **multipaths** section of the multipath configuration file, see Section, *"Multipaths Device Configuration Attributes"*.

2.2. Consistent Multipath Device Names in a Cluster

When the **user_friendly_names** configuration option is set to yes, the name of the multipath device is unique to a node, but it is not guaranteed to be the same on all nodes using the multipath device. Similarly, if you set the **alias** option for a device in the **multipaths** section of the `multipath.conf` configuration file, the name is not automatically consistent across all nodes in the cluster. This should not cause any difficulties if you use LVM to create logical devices from the multipath device, but if you require that your multipath device names be consistent in every node it is recommended that you leave the **user_friendly_names** option set to **no** and that you not configure aliases for the devices. By default, if you do not set **user_friendly_names** to yes or configure an alias for a device, a device name will be the WWID for the device, which is always the same. If you want the system-defined user-friendly names to be consistent across all nodes in the cluster, however, you can follow this procedure:

1. Set up all of the multipath devices on one machine.
2. Disable all of your multipath devices on your other machines by running the following commands:

```
# service multipath-tools stop
# multipath -F
```

3. Copy the `/etc/multipath/bindings` file from the first machine to all the other machines in the cluster.
4. Re-enable the multipathd daemon on all the other machines in the cluster by running the following command:

```
# service multipath-tools start
```

If you add a new device, you will need to repeat this process.

Similarly, if you configure an alias for a device that you would like to be consistent across the nodes in the cluster, you should ensure that the `/etc/multipath.conf` file is the same for each node in the cluster by following the same procedure:

1. Configure the aliases for the multipath devices in the in the `multipath.conf` file on one machine.
2. Disable all of your multipath devices on your other machines by running the following commands:

```
# service multipath-tools stop
# multipath -F
```

3. Copy the `multipath.conf` file from the first machine to all the other machines in the cluster.
4. Re-enable the multipathd daemon on all the other machines in the cluster by running the following command:

```
# service multipath-tools start
```

When you add a new device you will need to repeat this process.

2.3. Multipath Device attributes

In addition to the **user_friendly_names** and **alias** options, a multipath device has numerous attributes. You can modify these attributes for a specific multipath device by creating an entry for that device in the **multipaths** section of the **multipath** configuration file. For information on the **multipaths** section of the multipath configuration file, see Section, "*Configuration File Multipath Attributes*".

2.4. Multipath Devices in Logical Volumes

After creating multipath devices, you can use the multipath device names just as you would use a physical device name when creating an LVM physical volume. For example, if /dev/mapper/mpatha is the name of a multipath device, the following command will mark /dev/mapper/mpatha as a physical volume.

```
# pvcreate /dev/mapper/mpatha
```

You can use the resulting LVM physical device when you create an LVM volume group just as you would use any other LVM physical device.

If you attempt to create an LVM physical volume on a whole device on which you have configured partitions, the pvcreate command will fail.

When you create an LVM logical volume that uses active/passive multipath arrays as the underlying physical devices, you should include filters in the **lvm.conf** to exclude the disks that underlie the multipath devices. This is because if the array automatically changes the active path to the passive path when it receives I/O, multipath will failover and failback whenever LVM scans the passive path if these devices are not filtered. For active/passive arrays that require a command to make the passive path active, LVM prints a warning message when this occurs. To filter all SCSI devices in the LVM configuration file (lvm.conf), include the following filter in the devices section of the file.

```
filter = [ "r/block/", "r/disk/", "r/sd.*/", "a/.*/" ]
```

After updating `/etc/lvm.conf`, it's necessary to update the **initrd** so that this file will be copied there, where the filter matters the most, during boot. Perform:

```
update-initramfs -u -k all
```

Every time either `/etc/lvm.conf` or `/etc/multipath.conf` is updated, the initrd should be rebuilt to reflect these changes. This is imperative when blacklists and filters are necessary to maintain a stable storage configuration.

3. Setting up DM-Multipath Overview

This section provides step-by-step example procedures for configuring DM-Multipath. It includes the following procedures:

- Basic DM-Multipath setup
- Ignoring local disks
- Adding more devices to the configuration file

3.1. Setting Up DM-Multipath

Before setting up DM-Multipath on your system, ensure that your system has been updated and includes the **multipath-tools** package. If boot from SAN is desired, then the **multipath-tools-boot** package is also required.

A basic **/etc/multipath.conf** need not even exist, when **multpath** is run without an accompanying `/etc/multipath.conf`, it draws from it's internal database to find a suitable configuration, it also draws from it's internal blacklist. If after running **multipath -ll** without a config file, no multipaths are discovered. One must proceed to increase the verbosity to discover why a multipath was not created. Consider referencing the SAN vendor's documentation, the multipath example config files found in `/usr/share/doc/multipath-tools/examples`, and the live multipathd database:

```
# echo 'show config' | multipathd -k > multipath.conf-live
```

To work around a quirk in multipathd, when an `/etc/multipath.conf` doesn't exist, the previous command will return nothing, as it is the result of a *merge* between the `/etc/multipath.conf` and the database in memory. To remedy this, either define an empty `/etc/multipath.conf`, by using **touch**, or create one that redefines a default value like:

```
defaults {
        user_friendly_names no
}
```

and restart multipathd:

```
# service multipath-tools restart
```

Now the "show config" command will return the live database.

3.2. Installing with Multipath Support

To enable *multipath support during installation*[1] use

```
install disk-detect/multipath/enable=true
```

at the installer prompt. If multipath devices are found these will show up as **/dev/mapper/mpath<X>** during installation.

[1] http://wiki.debian.org/DebianInstaller/MultipathSupport

3.3. Ignoring Local Disks When Generating Multipath Devices

Some machines have local SCSI cards for their internal disks. DM-Multipath is not recommended for these devices. The following procedure shows how to modify the multipath configuration file to ignore the local disks when configuring multipath.

1. Determine which disks are the internal disks and mark them as the ones to blacklist. In this example, **/dev/sda** is the internal disk. Note that as originally configured in the default multipath configuration file, executing the **multipath -v2** shows the local disk, **/dev/sda**, in the multipath map. For further information on the **multipath** command output, see Section *"Multipath Command Output"*.

```
# multipath -v2
create: SIBM-ESXSST336732LC____F3ET0EP0Q000072428BX1 undef WINSYS,SF2372
size=33 GB features="0" hwhandler="0" wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 0:0:0:0 sda 8:0  [---------

device-mapper ioctl cmd 9 failed: Invalid argument
device-mapper ioctl cmd 14 failed: No such device or address
create: 3600a0b80001327d80000006d43621677 undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:0 sdb 8:16  undef ready  running
    `- 3:0:0:0 sdf 8:80 undef ready  running

create: 3600a0b800013275100000009a436215ec undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:1 sdc 8:32 undef ready  running
    `- 3:0:0:1 sdg 8:96 undef ready  running

create: 3600a0b80001327d800000070436216b3 undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:2 sdd 8:48 undef ready  running
    `- 3:0:0:2 sdg 8:112 undef ready  running

create: 3600a0b800013275100000009b4362163e undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:3 sdd 8:64 undef ready  running
    `- 3:0:0:3 sdg 8:128 undef ready  running
```

2. In order to prevent the device mapper from mapping **/dev/sda** in its multipath maps, edit the blacklist section of the `/etc/multipath.conf` file to include this device. Although you could blacklist the **sda** device using a **devnode** type, that would not be safe procedure since **/dev/sda** is not guaranteed to be the same on reboot. To blacklist individual devices, you can blacklist using the WWID of that device. Note that in the output to the **multipath -v2** command, the WWID of

the `/dev/sda` device is SIBM-ESXSST336732LC____F3ET0EP0Q000072428BX1. To blacklist this device, include the following in the `/etc/multipath.conf` file.

```
blacklist {
      wwid SIBM-ESXSST336732LC____F3ET0EP0Q000072428BX1
}
```

3. After you have updated the `/etc/multipath.conf` file, you must manually tell the **multipathd** daemon to reload the file. The following command reloads the updated `/etc/multipath.conf` file.

```
# service multipath-tools reload
```

4. Run the following command to remove the multipath device:

```
# multipath -f SIBM-ESXSST336732LC____F3ET0EP0Q000072428BX1
```

5. To check whether the device removal worked, you can run the **multipath -ll** command to display the current multipath configuration. For information on the **multipath -ll** command, see Section *"Multipath Queries with multipath Command"*. To check that the blacklisted device was not added back, you can run the multipath command, as in the following example. The multipath command defaults to a verbosity level of **v2** if you do not specify a **-v** option.

```
# multipath

create: 3600a0b80001327d80000006d43621677 undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:0 sdb 8:16  undef ready  running
    `- 3:0:0:0 sdf 8:80 undef ready  running

create: 3600a0b800013275100000009a436215ec undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:1 sdc 8:32 undef ready  running
    `- 3:0:0:1 sdg 8:96 undef ready  running

create: 3600a0b80001327d800000070436216b3 undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:2 sdd 8:48 undef ready  running
    `- 3:0:0:2 sdg 8:112 undef ready  running

create: 3600a0b800013275100000009b4362163e undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:3 sdd 8:64 undef ready  running
    `- 3:0:0:3 sdg 8:128 undef ready  running
```

3.4. Configuring Storage Devices

By default, DM-Multipath includes support for the most common storage arrays that support DM-Multipath. The default configuration values, including supported devices, can be found in the `multipath.conf.defaults` file.

If you need to add a storage device that is not supported by default as a known multipath device, edit the `/etc/multipath.conf` file and insert the appropriate device information.

For example, to add information about the HP Open-V series the entry looks like this, where **%n** is the device name:

```
devices {
     device {
            vendor "HP"
            product "OPEN-V."
            getuid_callout "/lib/udev/scsi_id --whitelisted --device=/dev/%n"
     }
}
```

For more information on the devices section of the configuration file, see Section *Configuration File Devices [71]*.

4. The DM-Multipath Configuration File

By default, DM-Multipath provides configuration values for the most common uses of multipathing. In addition, DM-Multipath includes support for the most common storage arrays that support DM-Multipath. The default configuration values and the supported devices can be found in the `multipath.conf.defaults` file.

You can override the default configuration values for DM-Multipath by editing the `/etc/multipath.conf` configuration file. If necessary, you can also add a storage array that is not supported by default to the configuration file. This chapter provides information on parsing and modifying the `multipath.conf` file. It contains sections on the following topics:

- *Configuration File Overview [63]*
- *Configuration File Blacklist [64]*
- *Configuration File Defaults [66]*
- *Configuration File Multipath Attributes [70]*
- *Configuration File Devices [71]*

In the multipath configuration file, you need to specify only the sections that you need for your configuration, or that you wish to change from the default values specified in the `multipath.conf.defaults` file. If there are sections of the file that are not relevant to your environment or for which you do not need to override the default values, you can leave them commented out, as they are in the initial file.

The configuration file allows regular expression description syntax.

An annotated version of the configuration file can be found in `/usr/share/doc/multipath-tools/examples/multipath.conf.annotated.gz`.

4.1. Configuration File Overview

The multipath configuration file is divided into the following sections:

blacklist
: Listing of specific devices that will not be considered for multipath.

blacklist_exceptions
: Listing of multipath candidates that would otherwise be blacklisted according to the parameters of the blacklist section.

defaults
: General default settings for DM-Multipath.

multipath
: Settings for the characteristics of individual multipath devices. These values overwrite what is specified in the **defaults** and **devices** sections of the configuration file.

devices

Settings for the individual storage controllers. These values overwrite what is specified in the **defaults** section of the configuration file. If you are using a storage array that is not supported by default, you may need to create a devices subsection for your array.

When the system determines the attributes of a multipath device, first it checks the multipath settings, then the per devices settings, then the multipath system defaults.

4.2. Configuration File Blacklist

The blacklist section of the multipath configuration file specifies the devices that will not be used when the system configures multipath devices. Devices that are blacklisted will not be grouped into a multipath device.

- If you do need to blacklist devices, you can do so according to the following criteria:
  - By WWID, as described *Blacklisting By WWID [64]*
  - By device name, as described in *Blacklisting By Device Name [64]*
  - By device type, as described in *Blacklisting By Device Type [65]*

  By default, a variety of device types are blacklisted, even after you comment out the initial blacklist section of the configuration file. For information, see *Blacklisting By Device Name [64]*

4.2.1. Blacklisting By WWID

You can specify individual devices to blacklist by their World-Wide IDentification with a **wwid** entry in the **blacklist** section of the configuration file.

The following example shows the lines in the configuration file that would blacklist a device with a WWID of 26353900f02796769.

```
blacklist {
       wwid 26353900f02796769
}
```

4.2.2. Blacklisting By Device Name

You can blacklist device types by device name so that they will not be grouped into a multipath device by specifying a **devnode** entry in the **blacklist** section of the configuration file.

The following example shows the lines in the configuration file that would blacklist all SCSI devices, since it blacklists all sd* devices.

```
blacklist {
       devnode "^sd[a-z]"
}
```

You can use a **devnode** entry in the **blacklist** section of the configuration file to specify individual devices to blacklist rather than all devices of a specific type. This is not recommended, however, since unless it is statically mapped by udev rules, there is no guarantee that a specific device will have the same name on reboot. For example, a device name could change from `/dev/sda` to `/dev/sdb` on reboot.

By default, the following **devnode** entries are compiled in the default blacklist; the devices that these entries blacklist do not generally support DM-Multipath. To enable multipathing on any of these devices, you would need to specify them in the **blacklist_exceptions** section of the configuration file, as described in *Blacklist Exceptions [65]*

```
blacklist {
       devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
       devnode "^hd[a-z]"
}
```

4.2.3. Blacklisting By Device Type

You can specify specific device types in the **blacklist** section of the configuration file with a device section. The following example blacklists all IBM DS4200 and HP devices.

```
blacklist {
       device {
               vendor  "IBM"
               product "3S42"       #DS4200 Product 10
       }
       device {
               vendor  "HP"
               product "*"
       }
}
```

4.2.4. Blacklist Exceptions

You can use the **blacklist_exceptions** section of the configuration file to enable multipathing on devices that have been blacklisted by default.

For example, if you have a large number of devices and want to multipath only one of them (with the WWID of 3600d0230000000000e13955cc3757803), instead of individually blacklisting each of the devices except the one you want, you could instead blacklist all of them, and then allow only the one you want by adding the following lines to the `/etc/multipath.conf` file.

```
blacklist {
        wwid "*"
}
```

```
blacklist_exceptions {
        wwid "3600d0230000000000e13955cc3757803"
}
```

When specifying devices in the **blacklist_exceptions** section of the configuration file, you must specify the exceptions in the same way they were specified in the **blacklist**. For example, a WWID exception will not apply to devices specified by a **devnode** blacklist entry, even if the blacklisted device is associated with that WWID. Similarly, devnode exceptions apply only to devnode entries, and device exceptions apply only to device entries.

4.3. Configuration File Defaults

The `/etc/multipath.conf` configuration file includes a **defaults** section that sets the **user_friendly_names** parameter to **yes**, as follows.

```
defaults {
        user_friendly_names yes
}
```

This overwrites the default value of the **user_friendly_names** parameter.

The configuration file includes a template of configuration defaults. This section is commented out, as follows.

```
#defaults {
#       udev_dir                /dev
#       polling_interval        5
#       selector                "round-robin 0"
#       path_grouping_policy    failover
#       getuid_callout          "/lib/dev/scsi_id --whitelisted --device=/dev/%n"
# prio   const
# path_checker  directio
# rr_min_io  1000
# rr_weight  uniform
# failback  manual
# no_path_retry  fail
# user_friendly_names no
#}
```

To overwrite the default value for any of the configuration parameters, you can copy the relevant line from this template into the **defaults** section and uncomment it. For example, to overwrite the **path_grouping_policy** parameter so that it is **multibus** rather than the default value of **failover**, copy the appropriate line from the template to the initial **defaults** section of the configuration file, and uncomment it, as follows.

```
defaults {
        user_friendly_names     yes
```

```
        path_grouping_policy    multibus
}
```

Table *Multipath Configuration Defaults [67]* describes the attributes that are set in the **defaults** section of the `multipath.conf` configuration file. These values are used by DM-Multipath unless they are overwritten by the attributes specified in the **devices** and **multipaths** sections of the `multipath.conf` file.

Tabella 5.2. Multipath Configuration Defaults

Attribute	Description
polling_interval	Specifies the interval between two path checks in seconds. For properly functioning paths, the interval between checks will gradually increase to (4 * **polling_interval**). The default value is **5**.
udev_dir	The directory where udev device nodes are created. The default value is /dev.
multipath_dir	The directory where the dynamic shared objects are stored. The default value is system dependent, commonly `/lib/multipath`.
verbosity	The default verbosity. Higher values increase the verbosity level. Valid levels are between 0 and 6. The default value is 2.
path_selector	Specifies the default algorithm to use in determining what path to use for the next I/O operation. Possible values include: • **round-robin 0**: Loop through every path in the path group, sending the same amount of I/O to each. • **queue-length 0**: Send the next bunch of I/O down the path with the least number of outstanding I/O requests. • **service-time 0**: Send the next bunch of I/O down the path with the shortest estimated service time, which is determined by dividing the total size of the outstanding I/O to each path by its relative throughput. The default value is **round-robin 0**.
path_grouping_policy	Specifies the default path grouping policy to apply to unspecified multipaths. Possible values include: • **failover** = 1 path per priority group • **multibus** = all valid paths in 1 priority group • **group_by_serial** = 1 priority group per detected serial number • **group_by_prio** = 1 priority group per path priority value • **group_by_node_name** = 1 priority group per target node name. The default value is **failover.**

Attribute	Description
getuid_callout	Specifies the default program and arguments to call out to obtain a unique path identifier. An absolute path is required. The default value is **/lib/udev/scsi_id --whitelisted --device=/dev/%n.**
prio	Specifies the default function to call to obtain a path priority value. For example, the ALUA bits in SPC-3 provide an exploitable prio value. Possible values include: • **const**: Set a priority of 1 to all paths. • **emc**: Generate the path priority for EMC arrays. • **alua**: Generate the path priority based on the SCSI-3 ALUA settings. • **netapp**: Generate the path priority for NetApp arrays. • **rdac**: Generate the path priority for LSI/Engenio RDAC controller. • **hp_sw**: Generate the path priority for Compaq/HP controller in active/standby mode. • **hds**: Generate the path priority for Hitachi HDS Modular storage arrays. The default value is **const**.
prio_args	The arguments string passed to the prio function Most prio functions do not need arguments. The datacore prioritizer need one. Example, "**timeout=1000 preferredsds=foo**". The default value is (null) "".
features	The extra features of multipath devices. The only existing feature is **queue_if_no_path**, which is the same as setting **no_path_retry** to **queue**. For information on issues that may arise when using this feature, see Section, *"Issues with queue_if_no_path feature"*.
path_checker	Specifies the default method used to determine the state of the paths. Possible values include: • **readsector0**: Read the first sector of the device. • **tur**: Issue a TEST UNIT READY to the device. • **emc_clariion**: Query the EMC Clariion specific EVPD page 0xC0 to determine the path. • **hp_sw**: Check the path state for HP storage arrays with Active/Standby firmware. • **rdac**: Check the path stat for LSI/Engenio RDAC storage controller. • **directio**: Read the first sector with direct I/O. The default value is **directio**.

Attribute	Description
failback	Manages path group failback. • A value of **immediate** specifies immediate failback to the highest priority path group that contains active paths. • A value of **manual** specifies that there should not be immediate failback but that failback can happen only with operator intervention. • A numeric value greater than zero specifies deferred failback, expressed in seconds. The default value is **manual**.
rr_min_io	Specifies the number of I/O requests to route to a path before switching to the next path in the current path group. The default value is `1000`.
rr_weight	If set to **priorities**, then instead of sending **rr_min_io** requests to a path before calling **path_selector** to choose the next path, the number of requests to send is determined by **rr_min_io** times the path's priority, as determined by the prio function. If set to **uniform**, all path weights are equal. The default value is **uniform**.
no_path_retry	A numeric value for this attribute specifies the number of times the system should attempt to use a failed path before disabling queueing. A value of fail indicates **immediate** failure, without queueing. A value of **queue** indicates that queueing should not stop until the path is fixed. The default value is `0`.
user_friendly_names	If set to yes, specifies that the system should use the `/etc/multipath/bindings` file to assign a persistent and unique **alias** to the **multipath**, in the form of mpathn. If set to no, specifies that the system should use the WWID as the **alias** for the **multipath**. In either case, what is specified here will be overridden by any device-specific aliases you specify in the multipaths section of the configuration file. The default value is **no**.
queue_without_daemon	If set to no, the **multipathd** daemon will disable queueing for all devices when it is shut down. The default value is **yes**.
flush_on_last_del	If set to yes, then **multipath** will disable queueing when the last path to a device has been deleted.

Attribute	Description
	The default value is **no**.
max_fds	Sets the maximum number of open file descriptors that can be opened by **multipath** and the **multipathd** daemon. This is equivalent to the ulimit -n command. A value of max will set this to the system limit from `/proc/sys/fs/nr_open`. If this is not set, the maximum number of open file descriptors is taken from the calling process; it is usually 1024. To be safe, this should be set to the maximum number of paths plus 32, if that number is greater than 1024.
checker_timer	The timeout to use for path checkers that issue SCSI commands with an explicit timeout, in seconds. The default value is taken from `/sys/block/sdx/device/timeout`, which is `30` seconds as of 12.04 LTS
fast_io_fail_tmo	The number of seconds the SCSI layer will wait after a problem has been detected on an FC remote port before failing I/O to devices on that remote port. This value should be smaller than the value of dev_loss_tmo. Setting this to off will disable the timeout. The default value is determined by the OS.
dev_loss_tmo	The number of seconds the SCSI layer will wait after a problem has been detected on an FC remote port before removing it from the system. Setting this to infinity will set this to 2147483647 seconds, or 68 years. The default value is determined by the OS.

4.4. Configuration File Multipath Attributes

Table *Multipath Attributes [70]* shows the attributes that you can set in the **multipaths** section of the `multipath.conf` configuration file for each specific multipath device. These attributes apply only to the one specified multipath. These defaults are used by DM-Multipath and override attributes set in the **defaults** and **devices** sections of the multipath.conf file.

Tabella 5.3. Multipath Attributes

Attribute	Description
wwid	Specifies the WWID of the **multipath** device to which the **multipath** attributes apply. This parameter is mandatory for this section of the `multipath.conf` file.
alias	Specifies the symbolic name for the **multipath** device to which the **multipath** attributes apply. If you are using **user_friendly_names**, do not set this value to mpathn; this may conflict with an automatically assigned user friendly name and give you incorrect device node names.

In addition, the following parameters may be overridden in this **multipath** section

- *path_grouping_policy*
- *path_selector*
- *failback*
- *prio*
- *prio_args*
- *no_path_retry*
- *rr_min_io*
- *rr_weight*
- *flush_on_last_del*

The following example shows multipath attributes specified in the configuration file for two specific multipath devices. The first device has a WWID of 3600508b4000156d70001200000b0000 and a symbolic name of yellow.

The second multipath device in the example has a WWID of 1DEC_____321816758474 and a symbolic name of red. In this example, the *rr_weight* attributes is set to priorities.

```
multipaths {
       multipath {
              wwid                   3600508b4000156d70001200000b0000
              alias                  yellow
              path_grouping_policy   multibus
              path_selector          "round-robin 0"
              failback               manual
              rr_weight              priorities
              no_path_retry          5
       }
       multipath {
              wwid                   1DEC_____321816758474
              alias                  red
              rr_weight              priorities
        }
}
```

4.5. Configuration File Devices

Table *Device Attributes [72]* shows the attributes that you can set for each individual storage device in the devices section of the multipath.conf configuration file. These attributes are used by DM-Multipath unless they are overwritten by the attributes specified in the **multipaths** section of the `multipath.conf` file for paths that contain the device. These attributes override the attributes set in the **defaults** section of the `multipath.conf` file.

Many devices that support multipathing are included by default in a multipath configuration. The values for the devices that are supported by default are listed in the `multipath.conf.defaults` file.

You probably will not need to modify the values for these devices, but if you do you can overwrite the default values by including an entry in the configuration file for the device that overwrites those values. You can copy the device configuration defaults from the `multipath.conf.annotated.gz` or if you wish to have a brief config file, `multipath.conf.synthetic` file for the device and override the values that you want to change.

To add a device to this section of the configuration file that is not configured automatically by default, you must set the **vendor** and **product** parameters. You can find these values by looking at **/sys/block/device_name/device/vendor** and **/sys/block/device_name/device/model** where device_name is the device to be multipathed, as in the following example:

```
# cat /sys/block/sda/device/vendor
WINSYS
# cat /sys/block/sda/device/model
SF2372
```

The additional parameters to specify depend on your specific device. If the device is active/active, you will usually not need to set additional parameters. You may want to set *path_grouping_policy* to **multibus**. Other parameters you may need to set are *no_path_retry* and *rr_min_io*, as described in Table *Multipath Attributes [70]*.

If the device is active/passive, but it automatically switches paths with I/O to the passive path, you need to change the checker function to one that does not send I/O to the path to test if it is working (otherwise, your device will keep failing over). This almost always means that you set the *path_checker* to **tur**; this works for all SCSI devices that support the Test Unit Ready command, which most do.

If the device needs a special command to switch paths, then configuring this device for multipath requires a hardware handler kernel module. The current available hardware handler is emc. If this is not sufficient for your device, you may not be able to configure the device for multipath.

Tabella 5.4. Device Attributes

Attribute	Description
vendor	Specifies the vendor name of the storage device to which the device attributes apply, for example **COMPAQ**.
product	Specifies the product name of the storage device to which the device attributes apply, for example **HSV110 (C)COMPAQ**.
revision	Specifies the product revision identifier of the storage device.
product_blacklist	Specifies a regular expression used to blacklist devices by product.
hardware_handler	Specifies a module that will be used to perform hardware specific actions when switching path groups or handling I/O errors. Possible values include:

Attribute	Description
	• **1 emc**: hardware handler for EMC storage arrays • **1 alua**: hardware handler for SCSI-3 ALUA arrays. • **1 hp_sw**: hardware handler for Compaq/HP controllers. • **1 rdac**: hardware handler for the LSI/Engenio RDAC controllers.

In addition, the following parameters may be overridden in this **device** section

- *path_grouping_policy*
- *getuid_callout*
- *path_selector*
- *path_checker*
- *features*
- *failback*
- *prio*
- *prio_args*
- *no_path_retry*
- *rr_min_io*
- *rr_weight*
- *fast_io_fail_tmo*
- *dev_loss_tmo*
- *flush_on_last_del*

Whenever a hardware_handler is specified, it is your responsibility to ensure that the appropriate kernel module is loaded to support the specified interface. These modules can be found in **/lib/modules/`uname -r`/kernel/drivers/scsi/device_handler/** . The requisite module should be integrated into the initrd to ensure the necessary discovery and failover-failback capacity is available during boot time. Example,

```
# cat scsi_dh_alua >> /etc/initramfs-tools/modules  ## append module to file
# update-initramfs -u -k all
```

The following example shows a device entry in the multipath configuration file.

```
#devices {
# device {
#  vendor   "COMPAQ  "
#  product   "MSA1000         "
#  path_grouping_policy multibus
#  path_checker  tur
#  rr_weight  priorities
# }
#}
```

The spacing reserved in the **vendor**, **product**, and **revision** fields are significant as multipath is performing a direct match against these attributes, whose format is defined by the SCSI specification, specifically the *Standard INQUIRY*[2] command. When quotes are used, the vendor, product, and revision fields will be interpreted strictly according to the spec. Regular expressions may be integrated into the quoted strings. Should a field be defined without the requisite spacing, multipath will copy the string into the properly sized buffer and pad with the appropriate number of spaces. The specification expects the entire field to be populated by printable characters or spaces, as seen in the example above

- vendor: 8 characters
- product: 16 characters
- revision: 4 characters

To create a more robust configuration file, regular expressions can also be used. Operators include **^ $ [ ] . * ? +**. Examples of functional regular expressions can be found by examining the live multipath database and `multipath.conf` example files found in `/usr/share/doc/multipath-tools/examples`:

```
# echo 'show config' | multipathd -k
```

[2] http://en.wikipedia.org/wiki/SCSI_Inquiry_Command

5. DM-Multipath Administration and Troubleshooting

5.1. Resizing an Online Multipath Device

If you need to resize an online multipath device, use the following procedure

1. Resize your physical device. This is storage platform specific.
2. Use the following command to find the paths to the LUN:

   ```
   # multipath -l
   ```

3. Resize your paths. For SCSI devices, writing 1 to the `rescan` file for the device causes the SCSI driver to rescan, as in the following command:

   ```
   # echo 1 > /sys/block/device_name/device/rescan
   ```

4. Resize your multipath device by running the multipathd resize command:

   ```
   # multipathd -k 'resize map mpatha'
   ```

5. Resize the file system (assuming no LVM or DOS partitions are used):

   ```
   # resize2fs /dev/mapper/mpatha
   ```

5.2. Moving root File Systems from a Single Path Device to a Multipath Device

This is dramatically simplified by the use of UUIDs to identify devices as an intrinsic label. Simply install **multipath-tools-boot** and reboot. This will rebuild the initial ramdisk and afford multipath the opportunity to build it's paths before the root file system is mounted by UUID.

Whenever `multipath.conf` is updated, so should the initrd by executing **update-initramfs -u -k all**. The reason being is `multipath.conf` is copied to the ramdisk and is integral to determining the available devices for grouping via it's blacklist and device sections.

5.3. Moving swap File Systems from a Single Path Device to a Multipath Device

The procedure is exactly the same as illustrated in the previous section called *Moving root File Systems from a Single Path to a Multipath Device*.

5.4. The Multipath Daemon

If you find you have trouble implementing a multipath configuration, you should ensure the multipath daemon is running as described in *"Setting up DM-Multipath"*. The **multipathd** daemon must be running in order to use multipathd devices. Also see section *Troubleshooting with the multipathd interactive console* concerning interacting with **multipathd** as a debugging aid.

5.5. Issues with queue_if_no_path

If **features "1 queue_if_no_path"** is specified in the `/etc/multipath.conf` file, then any process that uses I/O will hang until one or more paths are restored. To avoid this, set the ***no_path_retry*** **N** parameter in the `/etc/multipath.conf`.

When you set the **no_path_retry** parameter, remove the **features "1 queue_if_no_path"** option from the `/etc/multipath.conf` file as well. If, however, you are using a multipathed device for which the `features "1 queue_if_no_path"` option is set as a compiled in default, as it is for many SAN devices, you must add `features "0"` to override this default. You can do this by copying the existing **devices** section, and just that section (not the entire file), from `/usr/share/doc/multipath-tools/examples/multipath.conf.annotated.gz` into `/etc/multipath.conf` and editing to suit your needs.

If you need to use the `features "1 queue_if_no_path"` option and you experience the issue noted here, use the **dmsetup** command to edit the policy at runtime for a particular LUN (that is, for which all the paths are unavailable). For example, if you want to change the policy on the multipath device `mpathc` from `"queue_if_no_path"` to `"fail_if_no_path"`, execute the following command.

```
# dmsetup message mpathc 0 "fail_if_no_path"
```

You must specify the `mpathN` alias rather than the path

5.6. Multipath Command Output

When you create, modify, or list a multipath device, you get a printout of the current device setup. The format is as follows. For each multipath device:

```
action_if_any: alias (wwid_if_different_from_alias) dm_device_name_if_known vendor,product
size=size features='features' hwhandler='hardware_handler' wp=write_permission_if_known
```

For each path group:

```
-+- policy='scheduling_policy' prio=prio_if_known
status=path_group_status_if_known
```

For each path:

```
 `- host:channel:id:lun devnode major:minor dm_status_if_known path_status
online_status
```

For example, the output of a multipath command might appear as follows:

```
3600d0230000000000e13955cc3757800 dm-1 WINSYS,SF2372
size=269G features='0' hwhandler='0' wp=rw
|-+- policy='round-robin 0' prio=1 status=active
| `- 6:0:0:0 sdb 8:16  active ready  running
`-+- policy='round-robin 0' prio=1 status=enabled
```

```
    `- 7:0:0:0 sdf 8:80  active ready  running
```

If the path is up and ready for I/O, the status of the path is **ready** or *ghost*. If the path is down, the status is **faulty** or **shaky**. The path status is updated periodically by the **multipathd** daemon based on the polling interval defined in the `/etc/multipath.conf` file.

The dm status is similar to the path status, but from the kernel's point of view. The dm status has two states: **failed**, which is analogous to **faulty**, and **active** which covers all other path states. Occasionally, the path state and the dm state of a device will temporarily not agree.

The possible values for **online_status** are **running** and **offline**. A status of *offline* means that the SCSI device has been disabled.

When a multipath device is being created or modified , the path group status, the dm device name, the write permissions, and the dm status are not known. Also, the features are not always correct

5.7. Multipath Queries with multipath Command

You can use the **-l** and **-ll** options of the **multipath** command to display the current multipath configuration. The **-l** option displays multipath topology gathered from information in sysfs and the device mapper. The **-ll** option displays the information the **-l** displays in addition to all other available components of the system.

When displaying the multipath configuration, there are three verbosity levels you can specify with the **-v** option of the multipath command. Specifying **-v0** yields no output. Specifying **-v1** outputs the created or updated multipath names only, which you can then feed to other tools such as kpartx. Specifying **-v2** prints all detected paths, multipaths, and device maps.

The default **verbosity** level of multipath is **2** and can be globally modified by defining the *verbosity attribute* in the **defaults** section of `multipath.conf`.

The following example shows the output of a **multipath -l** command.

```
# multipath -l
  3600d0230000000000e13955cc3757800 dm-1 WINSYS,SF2372
  size=269G features='0' hwhandler='0' wp=rw
  |-+- policy='round-robin 0' prio=1 status=active
  | `- 6:0:0:0 sdb 8:16  active ready  running
  `-+- policy='round-robin 0' prio=1 status=enabled
    `- 7:0:0:0 sdf 8:80  active ready  running
```

The following example shows the output of a **multipath -ll** command.

```
# multipath -ll
  3600d0230000000000e13955cc3757801 dm-10 WINSYS,SF2372
  size=269G features='0' hwhandler='0' wp=rw
  |-+- policy='round-robin 0' prio=1 status=enabled
  | `- 19:0:0:1 sdc 8:32  active ready  running
```

```
  `-+- policy='round-robin 0' prio=1 status=enabled
    `- 18:0:0:1 sdh 8:112 active ready  running
    3600d0230000000000e13955cc3757803 dm-2 WINSYS,SF2372
    size=125G features='0' hwhandler='0' wp=rw
    `-+- policy='round-robin 0' prio=1 status=active
      |- 19:0:0:3 sde 8:64  active ready  running
        `- 18:0:0:3 sdj 8:144 active ready  running
```

5.8. Multipath Command Options

Table *Useful multipath Command Options [78]* describes some options of the **multipath** command that you might find useful.

Tabella 5.5. Useful multipath Command Options

Option	Description
-l	Display the current multipath configuration gathered from **sysfs** and the device mapper.
-ll	Display the current multipath configuration gathered from **sysfs**, the device mapper, and all other available components on the system.
-f device	Remove the named multipath device.
-F	Remove all unused multipath devices.

5.9. Determining Device Mapper Entries with dmsetup Command

You can use the **dmsetup** command to find out which device mapper entries match the **multipathed** devices.

The following command displays all the device mapper devices and their major and minor numbers. The minor numbers determine the name of the dm device. For example, a minor number of **3** corresponds to the multipathed device **/dev/dm-3**.

```
# dmsetup ls
mpathd  (253, 4)
mpathep1        (253, 12)
mpathfp1        (253, 11)
mpathb  (253, 3)
mpathgp1        (253, 14)
mpathhp1        (253, 13)
mpatha  (253, 2)
mpathh  (253, 9)
mpathg  (253, 8)
VolGroup00-LogVol01     (253, 1)
mpathf  (253, 7)
VolGroup00-LogVol00     (253, 0)
mpathe  (253, 6)
```

```
mpathbp1        (253, 10)
mpathd  (253, 5)
```

5.10. Troubleshooting with the multipathd interactive console

The **multipathd -k** command is an interactive interface to the **multipathd** daemon. Entering this command brings up an interactive multipath console. After entering this command, you can enter help to get a list of available commands, you can enter a interactive command, or you can enter **CTRL-D** to quit.

The multipathd interactive console can be used to troubleshoot problems you may be having with your system. For example, the following command sequence displays the multipath configuration, including the defaults, before exiting the console. See the IBM article *"Tricks with Multipathd"*[3] for more examples.

```
# multipathd -k
  > > show config
  > > CTRL-D
```

The following command sequence ensures that multipath has picked up any changes to the multipath.conf,

```
# multipathd -k
> > reconfigure
> > CTRL-D
```

Use the following command sequence to ensure that the path checker is working properly.

```
# multipathd -k
> > show paths
> > CTRL-D
```

Commands can also be streamed into multipathd using stdin like so:

```
# echo 'show config' | multipathd -k
```

[3] http://www-01.ibm.com/support/docview.wss?uid=isg3T1011985

Capitolo 6. Amministrazione remota

There are many ways to remotely administer a Linux server. This chapter will cover two of the most popular applications OpenSSH, and Puppet.

1. Server OpenSSH

1.1. Introduzione

This section of the Ubuntu Server Guide introduces a powerful collection of tools for the remote control of, and transfer of data between, networked computers called *OpenSSH*. You will also learn about some of the configuration settings possible with the OpenSSH server application and how to change them on your Ubuntu system.

OpenSSH is a freely available version of the Secure Shell (SSH) protocol family of tools for remotely controlling, or transferring files between, computers. Traditional tools used to accomplish these functions, such as telnet or rcp, are insecure and transmit the user's password in cleartext when used. OpenSSH provides a server daemon and client tools to facilitate secure, encrypted remote control and file transfer operations, effectively replacing the legacy tools.

Il componente server di OpenSSH, sshd, è in ascolto continuo per le connessioni in arrivo dei client, qualunque sia lo strumento usato sui client. Quando avviene una richiesta di connessione, per mezzo di sshd viene impostata la corretta connessione in base allo strumento utilizzato dal client. Per esempio, se il computer remoto sta effettuando una connessione con l'applicazione client ssh, il server OpenSSH imposta, dopo l'autenticazione, una sessione di controllo remoto. Se un utente remoto si connette a un server OpenSSH con scp, il demone server OpenSSH inizializza, dopo l'autenticazione, una procedura di copia sicura di file tra il server e il client. OpenSSH permette l'utilizzo di diversi metodi di autenticazione, inclusi password semplice, chiave pubblica e ticket Kerberos.

1.2. Installazione

L'installazione delle applicazioni server e client di OpenSSH è semplice. Per installare l'applicazione client OpenSSH sui sistemi Ubuntu, usare questo comando al prompt di un terminale:

```
sudo apt-get install openssh-client
```

Per installare l'applicazione server di OpenSSH e i relativi file di supporto, usare questo comando al prompt di un terminale:

```
sudo apt-get install openssh-server
```

È possibile scegliere di installare il pacchetto openssh-server durante il processo di installazione della Server Edition.

1.3. Configurazione

È possibile configurare il comportamento predefinito dell'applicazione server di OpenSSH, sshd, modificando il file `/etc/ssh/sshd_config`. Per maggiori informazioni riguardo le direttive di

configurazione usate in questo file, consultare l'appropriata pagina di manuale inserendo, a un prompt di terminale, il seguente comando:

```
man sshd_config
```

There are many directives in the sshd configuration file controlling such things as communication settings, and authentication modes. The following are examples of configuration directives that can be changed by editing the `/etc/ssh/sshd_config` file.

Prima di modificare il file di configurazione, è consigliato fare una copia del file originale e proteggerla dalla scrittura, così da avere le impostazioni originali come riferimento ed eventualmente riusarle se necessario.

Copiare il file `/etc/ssh/sshd_config` e proteggerlo da scrittura, con il seguente comando, digitando a un prompt di terminale:

```
sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.original
sudo chmod a-w /etc/ssh/sshd_config.original
```

Quelli che seguono sono esempi delle direttive di configurazione che è possibile cambiare:

- Per impostare OpenSSH in modo da restare in ascolto sulla porta TCP 2222 invece che sulla predefinita porta TCP 22, cambiare la direttiva Port come segue:

  Port 2222

- Per consentire l'utilizzo in sshd di credenziali di accesso basate su chiave pubblica, aggiungere o modificare la riga:

  PubkeyAuthentication yes

  If the line is already present, then ensure it is not commented out.

- Per far sì che il server OpenSSH mostri il contenuto del file `/etc/issue.net` come un banner di pre-accesso, aggiungere o modificare la riga:

  Banner /etc/issue.net

  Nel file `/etc/ssh/sshd_config`.

Dopo aver apportato dei cambiamenti al file `/etc/ssh/sshd_config`, salvarlo e, per rendere effettivi i cambiamenti, riavviare il demone sshd usando il seguente comando:

```
sudo /etc/init.d/ssh restart
```

Many other configuration directives for sshd are available to change the server application's behavior to fit your needs. Be advised, however, if your only method of access to a server is ssh, and you make a mistake in configuring sshd via the `/etc/ssh/sshd_config` file, you

may find you are locked out of the server upon restarting it. Additionally, if an incorrect configuration directive is supplied, the sshd server may refuse to start, so be extra careful when editing this file on a remote server.

1.4. Chiavi SSH

SSH *keys* allow authentication between two hosts without the need of a password. SSH key authentication uses two keys, a *private* key and a *public* key.

Per generare le chiavi, in un terminale, digitare:

```
ssh-keygen -t dsa
```

This will generate the keys using the *Digital Signature Algorithm (DSA)* method. During the process you will be prompted for a password. Simply hit *Enter* when prompted to create the key.

La chiave *pubblica* viene salvata, in modo predefinito, nel file `~/.ssh/id_dsa.pub`, mentre quella *privata* in `~/.ssh/id_dsa`. Ora, copiare il file `id_dsa.pub` nell'host remoto e aggiungere il suo contenuto al file `~/.ssh/authorized_keys` digitando:

```
ssh-copy-id NOME_UTENTE@HOST_REMOTO
```

Infine, controllare i permessi del file `authorized_keys`: solo l'utente autenticato dovrebbe avere i permessi di lettura e scrittura. Nel caso non fossero corretti, modificarli:

```
chmod 600 .ssh/authorized_keys
```

Dovrebbe essere possibile ora collegarsi via SSH all'host senza l'utilizzo di una password.

1.5. Riferimenti

- *Ubuntu Wiki SSH*[1] page.
- *Sito web di OpenSSH*[2]
- *Pagina wiki di OpenSSH avanzato*[3]

[1] https://help.ubuntu.com/community/SSH
[2] http://www.openssh.org/
[3] https://wiki.ubuntu.com/AdvancedOpenSSH

2. Puppet

Puppet is a cross platform framework enabling system administrators to perform common tasks using code. The code can do a variety of tasks from installing new software, to checking file permissions, or updating user accounts. Puppet is great not only during the initial installation of a system, but also throughout the system's entire life cycle. In most circumstances puppet will be used in a client/server configuration.

This section will cover installing and configuring Puppet in a client/server configuration. This simple example will demonstrate how to install Apache using Puppet.

2.1. Installazione

To install Puppet, in a terminal on the *server* enter:

```
sudo apt-get install puppetmaster
```

On the *client* machine, or machines, enter:

```
sudo apt-get install puppet
```

2.2. Configurazione

Prior to configuring puppet you may want to add a DNS *CNAME* record for *puppet.example.com*, where *example.com* is your domain. By default Puppet clients check DNS for puppet.example.com as the puppet server name, or *Puppet Master*. See *Capitolo 8, DNS (Domain Name Service) [140]* for more DNS details.

If you do not wish to use DNS, you can add entries to the server and client `/etc/hosts` file. For example, in the Puppet server's `/etc/hosts` file add:

```
127.0.0.1 localhost.localdomain localhost puppet
192.168.1.17 meercat02.example.com meercat02
```

On each Puppet client, add an entry for the server:

```
192.168.1.16 meercat.example.com meercat puppet
```

Replace the example IP addresses and domain names above with your actual server and client addresses and domain names.

Now setup some resources for apache2. Create a file `/etc/puppet/manifests/site.pp` containing the following:

```
package {
```

```
    'apache2':
        ensure => installed
}

service {
    'apache2':
        ensure => true,
        enable => true,
        require => Package['apache2']
}
```

Next, create a node file `/etc/puppet/manifests/nodes.pp` with:

```
node 'meercat02.example.com' {
   include apache2
}
```

Replace *meercat02.example.com* with your actual Puppet client's host name.

The final step for this simple Puppet server is to restart the daemon:

```
sudo /etc/init.d/puppetmaster restart
```

Now everything is configured on the Puppet server, it is time to configure the client.

First, configure the Puppetagent daemon to start. Edit `/etc/default/puppet`, changing *START* to yes:

```
START=yes
```

Then start the service:

```
sudo /etc/init.d/puppet start
```

Back on the Puppet server sign the client certificate by entering:

```
sudo puppetca --sign meercat02.example.com
```

Check `/var/log/syslog` for any errors with the configuration. If all goes well the apache2 package and it's dependencies will be installed on the Puppet client.

This example is *very* simple, and does not highlight many of Puppet's features and benefits. For more information see *Sezione 2.3, «Risorse» [85]*.

2.3. Risorse

- See the *Official Puppet Documentation*[4] web site.

[4] http://docs.puppetlabs.com/

- Also see *Pro Puppet*[5].
- Another source of additional information is the *Ubuntu Wiki Puppet Page*[6].

[5] http://www.apress.com/9781430230571

[6] https://help.ubuntu.com/community/Puppet

3. Zentyal

Zentyal is a Linux small business server, that can be configured as a Gateway, Infrastructure Manager, Unified Threat Manager, Office Server, Unified Communication Server or a combination of them. All network services managed by Zentyal are tightly integrated, automating most tasks. This helps to avoid errors in the network configuration and administration and allows to save time. Zentyal is open source, released under the GNU General Public License (GPL) and runs on top of Ubuntu GNU/Linux.

Zentyal consists of a serie of packages (usually one for each module) that provide a web interface to configure the different servers or services. The configuration is stored on a key-value Redis database but users, groups and domains related configuration is on OpenLDAP . When you configure any of the available parameters through the web interface, final configuration files are overwritten using the configuration templates provided by the modules. The main advantages of using Zentyal are: unified, graphical user interface to configure all network services and high, out-of-the-box integration between them.

3.1. Installazione

Zentyal 2.3 is available on Ubuntu 12.04 Universe repository. The modules available are:

- zentyal-core & zentyal-common: the core of the Zentyal interface and the common libraries of the framework. Also include the logs and events modules that give the administrator an interface to view the logs and generate events from them.
- zentyal-network: manages the configuration of the network. From the interfaces (supporting static IP, DHCP, VLAN, bridges or PPPoE), to multiple gateways when having more than one Internet connection, load balancing and advanced routing, static routes or dynamic DNS.
- zentyal-objects & zentyal-services: provide an abstration level for network addresses (e.g. LAN instead of 192.168.1.0/24) and ports named as services (e.g. HTTP instead of 80/TCP).
- zentyal-firewall: configures the iptables rules to block forbiden connections, NAT and port redirections.
- zentyal-ntp: installs the NTP daemon to keep server on time and allow network clients to synchronize their clocks against the server.
- zentyal-dhcp: configures ISC DHCP server supporting network ranges, static leases and other advanced options like NTP, WINS, dynamic DNS updates and network boot with PXE.
- zentyal-dns: brings ISC Bind9 DNS server into your server for caching local queries as a forwarder or as an authoritative server for the configured domains. Allows to configure A, CNAME, MX, NS, TXT and SRV records.
- zentyal-ca: integrates the management of a Certification Authority within Zentyal so users can use certificates to authenticate against the services, like with OpenVPN.
- zentyal-openvpn: allows to configure multiple VPN servers and clients using OpenVPN with dynamic routing configuration using Quagga.

- zentyal-users: provides an interface to configure and manage users and groups on OpenLDAP. Other services on Zentyal are authenticated against LDAP having a centralized users and groups management. It is also possible to synchronize users, passwords and groups from a Microsoft Active Directory domain.
- zentyal-squid: configures Squid and Dansguardian for speeding up browsing thanks to the caching capabilities and content filtering.
- zentyal-samba: allows Samba configuration and integration with existing LDAP. From the same interface you can define password policies, create shared resources and assign permissions.
- zentyal-printers: integrates CUPS with Samba and allows not only to configure the printers but also give them permissions based on LDAP users and groups.

To install Zentyal, in a terminal on the *server* enter (where <zentyal-module> is any of the modules from the previous list):

```
sudo apt-get install <zentyal-module>
```

Zentyal publishes one major stable release once a year (in September) based on latest Ubuntu LTS release. Stable releases always have even minor numbers (e.g. 2.2, 3.0) and beta releases have odd minor numbers (e.g. 2.1, 2.3). Ubuntu 12.04 comes with Zentyal 2.3 packages. If you want to upgrade to a new stable release published after the release of Ubuntu 12.04 you can use *Zentyal Team PPA*[7]. Upgrading to newer stable releases can provide you minor bugfixes not backported to 2.3 in Precise and newer features.

If you need more information on how to add packages from a PPA see *Add a Personal Package Archive (PPA)*[8].

Not present on Ubuntu Universe repositories, but on *Zentyal Team PPA*[9] you will find these other modules:

- zentyal-antivirus: integrates ClamAV antivirus with other modules like the proxy, file sharing or mailfilter.
- zentyal-asterisk: configures Asterisk to provide a simple PBX with LDAP based authentication.
- zentyal-bwmonitor: allows to monitor bandwith usage of your LAN clients.
- zentyal-captiveportal: integrates a captive portal with the firewall and LDAP users and groups.
- zentyal-ebackup: allows to make scheduled backups of your server using the popular duplicity backup tool.
- zentyal-ftp: configures a FTP server with LDAP based authentication.

[7] https://launchpad.net/~zentyal/
[8] https://help.ubuntu.com/12.04/ubuntu-help/addremove-ppa.html
[9] https://launchpad.net/~zentyal/

- zentyal-ids: integrates a network intrusion detection system.
- zentyal-ipsec: allows to configure IPsec tunnels using OpenSwan.
- zentyal-jabber: integrates ejabberd XMPP server with LDAP users and groups.
- zentyal-thinclients: a LTSP based thin clients solution.
- zentyal-mail: a full mail stack including Postfix and Dovecot with LDAP backend.
- zentyal-mailfilter: configures amavisd with mail stack to filter spam and attached virus.
- zentyal-monitor: integrates collectd to monitor server performance and running services.
- zentyal-pptp: configures a PPTP VPN server.
- zentyal-radius: integrates FreeRADIUS with LDAP users and groups.
- zentyal-software: simple interface to manage installed Zentyal modules and system updates.
- zentyal-trafficshaping: configures traffic limiting rules to do bandwidth throttling and improve latency.
- zentyal-usercorner: allows users to edit their own LDAP attributes using a web browser.
- zentyal-virt: simple interface to create and manage virtual machines based on libvirt.
- zentyal-webmail: allows to access your mail using the popular Roundcube webmail.
- zentyal-webserver: configures Apache webserver to host different sites on your machine.
- zentyal-zarafa: integrates Zarafa groupware suite with Zentyal mail stack and LDAP.

3.2. First steps

Any system account belonging to the sudo group is allowed to log into Zentyal web interface. If you are using the user created during the installation, this should be in the sudo group by default.

If you need to add another user to the sudo group, just execute:

```
sudo adduser username sudo
```

To access Zentyal web interface, browse into https://localhost/ (or the IP of your remote server). As Zentyal creates its own self-signed SSL certificate, you will have to accept a security exception on your browser.

Once logged in you will see the dashboard with an overview of your server. To configure any of the features of your installed modules, go to the different sections on the left menu. When you make any changes, on the upper right corner appears a red *Save changes* button that you must click to save all configuration changes. To apply these configuration changes in your server, the module needs to be enabled first, you can do so from the *Module Status* entry on the left menu. Every time you enable a module, a pop-up will appear asking for a confirmation to perform the necessary actions and changes on your server and configuration files.

If you need to customize any configuration file or run certain actions (scripts or commands) to configure features not available on Zentyal place the custom configuration file templates on /etc/zentyal/stubs/<module>/ and the hooks on /etc/zentyal/hooks/<module>.<action>.

3.3. Riferimenti

Zentyal Official Documentation [10] page.

See also *Zentyal Community Documentation*[11] page.

And don't forget to visit the *forum* [12] for community support, feedback, feature requests, etc.

[10] http://doc.zentyal.org/
[11] http://trac.zentyal.org/wiki/Documentation
[12] http://forum.zentyal.org/

Capitolo 7. Autenticazione di rete

This section applies LDAP to network authentication and authorization.

1. Server OpenLDAP

The Lightweight Directory Access Protocol, or LDAP, is a protocol for querying and modifying a X.500-based directory service running over TCP/IP. The current LDAP version is LDAPv3, as defined in *RFC4510*[1], and the LDAP implementation used in Ubuntu is OpenLDAP, currently at version 2.4.25 (Oneiric).

So this protocol accesses LDAP directories. Here are some key concepts and terms:

- A LDAP directory is a tree of data *entries* that is hierarchical in nature and is called the Directory Information Tree (DIT).
- An entry consists of a set of *attributes*.
- An attribute has a *type* (a name/description) and one or more *values*.
- Every attribute must be defined in at least one *objectClass*.
- Attributes and objectclasses are defined in *schemas* (an objectclass is actually considered as a special kind of attribute).
- Each entry has a unique identifier: it's *Distinguished Name* (DN or dn). This consists of it's *Relative Distinguished Name* (RDN) followed by the parent entry's DN.
- The entry's DN is not an attribute. It is not considered part of the entry itself.

The terms *object*, *container*, and *node* have certain connotations but they all essentially mean the same thing as *entry*, the technically correct term.

For example, below we have a single entry consisting of 11 attributes. It's DN is "cn=John Doe,dc=example,dc=com"; it's RDN is "cn=John Doe"; and it's parent DN is "dc=example,dc=com".

```
dn: cn=John Doe,dc=example,dc=com
cn: John Doe
givenName: John
sn: Doe
telephoneNumber: +1 888 555 6789
telephoneNumber: +1 888 555 1232
mail: john@example.com
manager: cn=Larry Smith,dc=example,dc=com
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: top
```

The above entry is in *LDIF* format (LDAP Data Interchange Format). Any information that you feed into your DIT must also be in such a format. It is defined in *RFC2849*[2].

Although this guide will describe how to use it for central authentication, LDAP is good for anything that involves a large number of access requests to a mostly-read, attribute-based (name:value)

[1] http://tools.ietf.org/html/rfc4510
[2] http://tools.ietf.org/html/rfc2849

backend. Examples include an address book, a list of email addresses, and a mail server's configuration.

1.1. Installazione

Install the OpenLDAP server daemon and the traditional LDAP management utilities. These are found in packages slapd and ldap-utils respectively.

The installation of slapd will create a working configuration. In particular, it will create a database instance that you can use to store your data. However, the suffix (or base DN) of this instance will be determined from the domain name of the localhost. If you want something different, edit `/etc/hosts` and replace the domain name with one that will give you the suffix you desire. For instance, if you want a suffix of *dc=example,dc=com* then your file would have a line similar to this:

```
127.0.1.1       hostname.example.com hostname
```

You can revert the change after package installation.

This guide will use a database suffix of *dc=example,dc=com*.

Proceed with the install:

```
sudo apt-get install slapd ldap-utils
```

Since Ubuntu 8.10 slapd is designed to be configured within slapd itself by dedicating a separate DIT for that purpose. This allows one to dynamically configure slapd without the need to restart the service. This configuration database consists of a collection of text-based LDIF files located under `/etc/ldap/slapd.d`. This way of working is known by several names: the slapd-config method, the RTC method (Real Time Configuration), or the cn=config method. You can still use the traditional flat-file method (slapd.conf) but it's not recommended; the functionality will be eventually phased out.

Ubuntu now uses the *slapd-config* method for slapd configuration and this guide reflects that.

During the install you were prompted to define administrative credentials. These are LDAP-based credentials for the *rootDN* of your database instance. By default, this user's DN is *cn=admin,dc=example,dc=com*. Also by default, there is no administrative account created for the slapd-config database and you will therefore need to authenticate externally to LDAP in order to access it. We will see how to do this later on.

Some classical schemas (cosine, nis, inetorgperson) come built-in with slapd nowadays. There is also an included "core" schema, a pre-requisite for any schema to work.

1.2. Post-install Inspection

The installation process set up 2 DITs. One for slapd-config and one for your own data (dc=example,dc=com). Let's take a look.

- This is what the slapd-config database/DIT looks like. Recall that this database is LDIF-based and lives under `/etc/ldap/slapd.d`:

```
   /etc/ldap/slapd.d/

### cn=config
#   ### cn=module{0}.ldif
#   ### cn=schema
#   #   ### cn={0}core.ldif
#   #   ### cn={1}cosine.ldif
#   #   ### cn={2}nis.ldif
#   #   ### cn={3}inetorgperson.ldif
#   ### cn=schema.ldif
#   ### olcBackend={0}hdb.ldif
#   ### olcDatabase={0}config.ldif
#   ### olcDatabase={-1}frontend.ldif
#   ### olcDatabase={1}hdb.ldif
### cn=config.ldif
```

Do not edit the slapd-config database directly. Make changes via the LDAP protocol (utilities).

- This is what the slapd-config DIT looks like via the LDAP protocol:

```
sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b cn=config dn

dn: cn=config

dn: cn=module{0},cn=config

dn: cn=schema,cn=config

dn: cn={0}core,cn=schema,cn=config

dn: cn={1}cosine,cn=schema,cn=config

dn: cn={2}nis,cn=schema,cn=config

dn: cn={3}inetorgperson,cn=schema,cn=config

dn: olcBackend={0}hdb,cn=config

dn: olcDatabase={-1}frontend,cn=config
```

```
dn: olcDatabase={0}config,cn=config

dn: olcDatabase={1}hdb,cn=config
```

Explanation of entries:

- *cn=config*: global settings
- *cn=module{0},cn=config*: a dynamically loaded module
- *cn=schema,cn=config*: contains hard-coded system-level schema
- *cn={0}core,cn=schema,cn=config*: the hard-coded core schema
- *cn={1}cosine,cn=schema,cn=config*: the cosine schema
- *cn={2}nis,cn=schema,cn=config*: the nis schema
- *cn={3}inetorgperson,cn=schema,cn=config*: the inetorgperson schema
- *olcBackend={0}hdb,cn=config*: the 'hdb' backend storage type
- *olcDatabase={-1}frontend,cn=config*: frontend database, default settings for other databases
- *olcDatabase={0}config,cn=config*: slapd configuration database (cn=config)
- *olcDatabase={1}hdb,cn=config*: your database instance (dc=examle,dc=com)

- This is what the dc=example,dc=com DIT looks like:

```
ldapsearch -x -LLL -H ldap:/// -b dc=example,dc=com dn

dn: dc=example,dc=com

dn: cn=admin,dc=example,dc=com
```

Explanation of entries:

- *dc=example,dc=com*: base of the DIT
- *cn=admin,dc=example,dc=com*: administrator (rootDN) for this DIT (set up during package install)

1.3. Modifying/Populating your Database

Let's introduce some content to our database. We will add the following:

- a node called *People* (to store users)
- a node called *Groups* (to store groups)
- a group called *miners*
- a user called *john*

Create the following LDIF file and call it `add_content.ldif`:

```
dn: ou=People,dc=example,dc=com
```

```
objectClass: organizationalUnit
ou: People

dn: ou=Groups,dc=example,dc=com
objectClass: organizationalUnit
ou: Groups

dn: cn=miners,ou=Groups,dc=example,dc=com
objectClass: posixGroup
cn: miners
gidNumber: 5000

dn: uid=john,ou=People,dc=example,dc=com
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: john
sn: Doe
givenName: John
cn: John Doe
displayName: John Doe
uidNumber: 10000
gidNumber: 5000
userPassword: johnldap
gecos: John Doe
loginShell: /bin/bash
homeDirectory: /home/john
```

It's important that uid and gid values in your directory do not collide with local values. Use high number ranges, such as starting at 5000. By setting the uid and gid values in ldap high, you also allow for easier control of what can be done with a local user vs a ldap one. More on that later.

Add the content:

```
ldapadd -x -D cn=admin,dc=example,dc=com -W -f add_content.ldif

Enter LDAP Password: ********
adding new entry "ou=People,dc=example,dc=com"

adding new entry "ou=Groups,dc=example,dc=com"

adding new entry "cn=miners,ou=Groups,dc=example,dc=com"

adding new entry "uid=john,ou=People,dc=example,dc=com"
```

We can check that the information has been correctly added with the ldapsearch utility:

```
ldapsearch -x -LLL -b dc=example,dc=com 'uid=john' cn gidNumber
```

```
dn: uid=john,ou=People,dc=example,dc=com
cn: John Doe
gidNumber: 5000
```

Explanation of switches:

- *-x:* "simple" binding; will not use the default SASL method
- *-LLL:* disable printing extraneous information
- *uid=john:* a "filter" to find the john user
- *cn gidNumber:* requests certain attributes to be displayed (the default is to show all attributes)

1.4. Modifying the slapd Configuration Database

The slapd-config DIT can also be queried and modified. Here are a few examples.

- Use ldapmodify to add an "Index" (DbIndex attribute) to your {1}hdb,cn=config database (dc=example,dc=com). Create a file, call it `uid_index.ldif`, with the following contents:

  ```
  dn: olcDatabase={1}hdb,cn=config
  add: olcDbIndex
  olcDbIndex: uid eq,pres,sub
  ```

  Then issue the command:

  ```
  sudo ldapmodify -Q -Y EXTERNAL -H ldapi:/// -f uid_index.ldif

  modifying entry "olcDatabase={1}hdb,cn=config"
  ```

  You can confirm the change in this way:

  ```
  sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b \
  cn=config '(olcDatabase={1}hdb)' olcDbIndex

  dn: olcDatabase={1}hdb,cn=config
  olcDbIndex: objectClass eq
  olcDbIndex: uid eq,pres,sub
  ```

- Let's add a schema. It will first need to be converted to LDIF format. You can find unconverted schemas in addition to converted ones in the `/etc/ldap/schema` directory.

  

  - It is not trivial to remove a schema from the slapd-config database. Practice adding schemas on a test system.
  - Before adding any schema, you should check which schemas are already installed (shown is a default, out-of-the-box output):

```
sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b \
cn=schema,cn=config dn

dn: cn=schema,cn=config

dn: cn={0}core,cn=schema,cn=config

dn: cn={1}cosine,cn=schema,cn=config

dn: cn={2}nis,cn=schema,cn=config

dn: cn={3}inetorgperson,cn=schema,cn=config
```

In the following example we'll add the CORBA schema.

1. Create the conversion configuration file `schema_convert.conf` containing the following lines:

```
include /etc/ldap/schema/core.schema
include /etc/ldap/schema/collective.schema
include /etc/ldap/schema/corba.schema
include /etc/ldap/schema/cosine.schema
include /etc/ldap/schema/duaconf.schema
include /etc/ldap/schema/dyngroup.schema
include /etc/ldap/schema/inetorgperson.schema
include /etc/ldap/schema/java.schema
include /etc/ldap/schema/misc.schema
include /etc/ldap/schema/nis.schema
include /etc/ldap/schema/openldap.schema
include /etc/ldap/schema/ppolicy.schema
include /etc/ldap/schema/ldapns.schema
include /etc/ldap/schema/pmi.schema
```

2. Create the output directory `ldif_output`.
3. Determine the index of the schema:

```
slapcat -f schema_convert.conf -F ldif_output -n 0 | grep corba,cn=schema

cn={1}corba,cn=schema,cn=config
```

When slapd injests objects with the same parent DN it will create an *index* for that object. An index is contained within braces: {X}.

4. Use slapcat to perform the conversion:

```
slapcat -f schema_convert.conf -F ldif_output -n0 -H \
ldap:///cn={1}corba,cn=schema,cn=config -l cn=corba.ldif
```

The converted schema is now in `cn=corba.ldif`

5. Edit `cn=corba.ldif` to arrive at the following attributes:

```
dn: cn=corba,cn=schema,cn=config
...
cn: corba
```

Also remove the following lines from the bottom:

```
structuralObjectClass: olcSchemaConfig
entryUUID: 52109a02-66ab-1030-8be2-bbf166230478
creatorsName: cn=config
createTimestamp: 20110829165435Z
entryCSN: 20110829165435.935248Z#000000#000#000000
modifiersName: cn=config
modifyTimestamp: 20110829165435Z
```

Your attribute values will vary.

6. Finally, use ldapadd to add the new schema to the slapd-config DIT:

```
sudo ldapadd -Q -Y EXTERNAL -H ldapi:/// -f cn\=corba.ldif

adding new entry "cn=corba,cn=schema,cn=config"
```

7. Confirm currently loaded schemas:

```
sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b cn=schema,cn=config dn

dn: cn=schema,cn=config

dn: cn={0}core,cn=schema,cn=config

dn: cn={1}cosine,cn=schema,cn=config

dn: cn={2}nis,cn=schema,cn=config

dn: cn={3}inetorgperson,cn=schema,cn=config

dn: cn={4}corba,cn=schema,cn=config
```

For external applications and clients to authenticate using LDAP they will each need to be specifically configured to do so. Refer to the appropriate client-side documentation for details.

1.5. Registrazione

Activity logging for slapd is indispensible when implementing an OpenLDAP-based solution yet it must be manually enabled after software installation. Otherwise, only rudimentary messages will appear in the logs. Logging, like any other slapd configuration, is enabled via the slapd-config database.

OpenLDAP comes with multiple logging subsystems (levels) with each one containing the lower one (additive). A good level to try is *stats*. The *slapd-config*[3] man page has more to say on the different subsystems.

Create the file `logging.ldif` with the following contents:

```
dn: cn=config
changetype: modify
add: olcLogLevel
olcLogLevel: stats
```

Implement the change:

```
sudo ldapmodify -Q -Y EXTERNAL -H ldapi:/// -f logging.ldif
```

This will produce a significant amount of logging and you will want to throttle back to a less verbose level once your system is in production. While in this verbose mode your host's syslog engine (rsyslog) may have a hard time keeping up and may drop messages:

```
rsyslogd-2177: imuxsock lost 228 messages from pid 2547 due to rate-limiting
```

You may consider a change to rsyslog's configuration. In `/etc/rsyslog.conf`, put:

```
# Disable rate limiting
# (default is 200 messages in 5 seconds; below we make the 5 become 0)
$SystemLogRateLimitInterval 0
```

And then restart the rsyslog daemon:

```
sudo service rsyslog restart
```

1.6. Replication

The LDAP service becomes increasingly important as more networked systems begin to depend on it. In such an environment, it is standard practice to build redundancy (high availability) into LDAP to prevent havoc should the LDAP server become unresponsive. This is done through *LDAP replication*.

[3] http://manpages.ubuntu.com/manpages/en/man5/slapd-config.5.html

Replication is achieved via the *Syncrepl* engine. This allows changes to be synchronized using a *Consumer - Provider* model. The specific kind of replication we will implement in this guide is a combination of the following modes: *refreshAndPersist* and *delta-syncrepl*. This has the Provider push changed entries to the Consumer as soon as they're made but, in addition, only actual changes will be sent, not entire entries.

1.6.1. Provider Configuration

Begin by configuring the *Provider*.

1. Create an LDIF file with the following contents and name it `provider_sync.ldif`:

```
# Add indexes to the frontend db.
dn: olcDatabase={1}hdb,cn=config
changetype: modify
add: olcDbIndex
olcDbIndex: entryCSN eq
-
add: olcDbIndex
olcDbIndex: entryUUID eq

#Load the syncprov and accesslog modules.
dn: cn=module{0},cn=config
changetype: modify
add: olcModuleLoad
olcModuleLoad: syncprov
-
add: olcModuleLoad
olcModuleLoad: accesslog

# Accesslog database definitions
dn: olcDatabase={2}hdb,cn=config
objectClass: olcDatabaseConfig
objectClass: olcHdbConfig
olcDatabase: {2}hdb
olcDbDirectory: /var/lib/ldap/accesslog
olcSuffix: cn=accesslog
olcRootDN: cn=admin,dc=example,dc=com
olcDbIndex: default eq
olcDbIndex: entryCSN,objectClass,reqEnd,reqResult,reqStart

# Accesslog db syncprov.
dn: olcOverlay=syncprov,olcDatabase={2}hdb,cn=config
changetype: add
objectClass: olcOverlayConfig
objectClass: olcSyncProvConfig
olcOverlay: syncprov
olcSpNoPresent: TRUE
olcSpReloadHint: TRUE
```

```
# syncrepl Provider for primary db
dn: olcOverlay=syncprov,olcDatabase={1}hdb,cn=config
changetype: add
objectClass: olcOverlayConfig
objectClass: olcSyncProvConfig
olcOverlay: syncprov
olcSpNoPresent: TRUE

# accesslog overlay definitions for primary db
dn: olcOverlay=accesslog,olcDatabase={1}hdb,cn=config
objectClass: olcOverlayConfig
objectClass: olcAccessLogConfig
olcOverlay: accesslog
olcAccessLogDB: cn=accesslog
olcAccessLogOps: writes
olcAccessLogSuccess: TRUE
# scan the accesslog DB every day, and purge entries older than 7 days
olcAccessLogPurge: 07+00:00 01+00:00
```

Change the rootDN in the LDIF file to match the one you have for your directory.

2. The apparmor profile for slapd will need to be adjusted for the accesslog database location. Edit `/etc/apparmor.d/local/usr.sbin.slapd` by adding the following:

```
/var/lib/ldap/accesslog/ r,
/var/lib/ldap/accesslog/** rwk,
```

Create a directory, set up a databse config file, and reload the apparmor profile:

```
sudo -u openldap mkdir /var/lib/ldap/accesslog
sudo -u openldap cp /var/lib/ldap/DB_CONFIG /var/lib/ldap/accesslog
sudo service apparmor reload
```

3. Add the new content and, due to the apparmor change, restart the daemon:

```
sudo ldapadd -Q -Y EXTERNAL -H ldapi:/// -f provider_sync.ldif
sudo service slapd restart
```

The Provider is now configured.

1.6.2. Consumer Configuration

And now configure the *Consumer*.

1. Install the software by going through *Sezione 1.1, «Installazione» [93]*. Make sure the slapd-config databse is identical to the Provider's. In particular, make sure schemas and the databse suffix are the same.
2. Create an LDIF file with the following contents and name it `consumer_sync.ldif`:

```
dn: cn=module{0},cn=config
changetype: modify
add: olcModuleLoad
olcModuleLoad: syncprov

dn: olcDatabase={1}hdb,cn=config
changetype: modify
add: olcDbIndex
olcDbIndex: entryUUID eq
-
add: olcSyncRepl
olcSyncRepl: rid=0 provider=ldap://ldap01.example.com bindmethod=simple binddn="cn=admin,dc=exa
 credentials=secret searchbase="dc=example,dc=com" logbase="cn=accesslog"
 logfilter="(&(objectClass=auditWriteObject)(reqResult=0))" schemachecking=on
 type=refreshAndPersist retry="60 +" syncdata=accesslog
-
add: olcUpdateRef
olcUpdateRef: ldap://ldap01.example.com
```

Ensure the following attributes have the correct values:

- *provider* (Provider server's hostname -- ldap01.example.com in this example -- or IP address)
- *binddn* (the admin DN you're using)
- *credentials* (the admin DN password you're using)
- *searchbase* (the database suffix you're using)
- *olcUpdateRef* (Provider server's hostname or IP address)
- *rid* (Replica ID, an unique 3-digit that identifies the replica. Each consumer should have at least one rid)

3. Add the new content:

```
sudo ldapadd -Q -Y EXTERNAL -H ldapi:/// -f consumer_sync.ldif
```

You're done. The two databases (suffix: dc=example,dc=com) should now be synchronizing.

1.6.3. Test

Once replication starts, you can monitor it by running

```
ldapsearch -z1 -LLLQY EXTERNAL -H ldapi:/// -s base contextCSN

dn: dc=example,dc=com
contextCSN: 20120201193408.178454Z#000000#000#000000
```

on both the provider and the consumer. Once the output (`20120201193408.178454Z#000000#000#000000` in the above example) for both machines match, you have replication. Every time a change is done in the provider, this value will change and so should the one in the consumer(s).

If your connection is slow and/or your ldap database large, it might take a while for the consumer's *contextCSN* match the provider's. But, you will know it is progressing since the consumer's *contextCSN* will be steadly increasing.

If the consumer's *contextCSN* is missing or does not match the provider, you should stop and figure out the issue before continuing. Try checking the slapd (syslog) and the auth log files in the provider to see if the consumer's authentication requests were successful or its requests to retrieve data (they look like a lot of ldapsearch statements) return no errors.

To test if it worked simply query, on the Consumer, the DNs in the database:

```
sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b dc=example,dc=com dn
```

You should see the user 'john' and the group 'miners' as well as the nodes 'People' and 'Groups'.

1.7. Access Control

The management of what type of access (read, write, etc) users should be granted to resources is known as *access control*. The configuration directives involved are called *access control lists* or ACL.

When we installed the slapd package various ACL were set up automatically. We will look at a few important consequences of those defaults and, in so doing, we'll get an idea of how ACLs work and how they're configured.

To get the effective ACL for an LDAP query we need to look at the ACL entries of the database being queried as well as those of the special frontend database instance. The ACLs belonging to the latter act as defaults in case those of the former do not match. The frontend database is the second to be consulted and the ACL to be applied is the first to match ("first match wins") among these 2 ACL sources. The following commands will give, respectively, the ACLs of the hdb database ("dc=example,dc=com") and those of the frontend database:

```
sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b \
cn=config '(olcDatabase={1}hdb)' olcAccess
```

```
dn: olcDatabase={1}hdb,cn=config
olcAccess: {0}to attrs=userPassword,shadowLastChange by self write by anonymous
              auth by dn="cn=admin,dc=example,dc=com" write by * none
olcAccess: {1}to dn.base="" by * read
olcAccess: {2}to * by self write by dn="cn=admin,dc=example,dc=com" write by *
  read
```

The rootDN always has full rights to it's database. Including it in an ACL does provide an explicit configuration but it also causes slapd to incur a performance penalty.

```
sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b \
cn=config '(olcDatabase={-1}frontend)' olcAccess
```

```
dn: olcDatabase={-1}frontend,cn=config
olcAccess: {0}to * by dn.exact=gidNumber=0+uidNumber=0,cn=peercred,
              cn=external,cn=auth manage by * break
olcAccess: {1}to dn.exact="" by * read
olcAccess: {2}to dn.base="cn=Subschema" by * read
```

The very first ACL is crucial:

```
olcAccess: {0}to attrs=userPassword,shadowLastChange by self write by anonymous
              auth by dn="cn=admin,dc=example,dc=com" write by * none
```

This can be represented differently for easier digestion:

```
to attrs=userPassword
 by self write
 by anonymous auth
 by dn="cn=admin,dc=example,dc=com" write
 by * none

to attrs=shadowLastChange
 by self write
 by anonymous auth
 by dn="cn=admin,dc=example,dc=com" write
 by * none
```

This compound ACL (there are 2) enforces the following:

- Anonymous 'auth' access is provided to the *userPassword* attribute for the initial connection to occur. Perhaps counter-intuitively, 'by anonymous auth' is needed even when anonymous access to the DIT is unwanted. Once the remote end is connected, howerver, authentication can occur (see next point).
- Authentication can happen because all users have 'read' (due to 'by self write') access to the *userPassword* attribute.
- The *userPassword* attribute is otherwise unaccessible by all other users, with the exception of the rootDN, who has complete access to it.
- In order for users to change their own password, using **passwd** or other utilities, the *shadowLastChange* attribute needs to be accessible once a user has authenticated.

This DIT can be searched anonymously because of 'by * read' in this ACL:

```
to *
 by self write
 by dn="cn=admin,dc=example,dc=com" write
 by * read
```

If this is unwanted then you need to change the ACLs. To force authentication during a bind request you can alternatively (or in combination with the modified ACL) use the 'olcRequire: authc' directive.

As previously mentioned, there is no administrative account created for the slapd-config database. There is, however, a SASL identity that is granted full access to it. It represents the localhost's superuser (root/sudo). Here it is:

```
dn.exact=gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
```

The following command will display the ACLs of the slapd-config database:

```
sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b \
cn=config '(olcDatabase={0}config)' olcAccess
```

```
dn: olcDatabase={0}config,cn=config
olcAccess: {0}to * by dn.exact=gidNumber=0+uidNumber=0,cn=peercred,
              cn=external,cn=auth manage by * break
```

Since this is a SASL identity we need to use a SASL *mechanism* when invoking the LDAP utility in question and and we have seen it plenty of times in this guide. It is the EXTERNAL mechanism. See the previous command for an example. Note that:

1. You must use *sudo* to become the root identity in order for the ACL to match.
2. The EXTERNAL mechanism works via *IPC* (UNIX domain sockets). This means you must use the *ldapi* URI format.

A succinct way to get all the ACLs is like this:

```
sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b \
cn=config '(olcAccess=*)' olcAccess olcSuffix
```

There is much to say on the topic of access control. See the man page for *slapd.access*[4].

1.8. TLS

When authenticating to an OpenLDAP server it is best to do so using an encrypted session. This can be accomplished using Transport Layer Security (TLS).

Here, we will be our own *Certificate Authority* and then create and sign our LDAP server certificate as that CA. Since slapd is compiled using the gnutls library, we will use the certtool utility to complete these tasks.

1. Install the gnutls-bin and ssl-cert packages:

   ```
   sudo apt-get install gnutls-bin ssl-cert
   ```

2. Create a private key for the Certificate Authority:

[4] http://manpages.ubuntu.com/manpages/en/man5/slapd.access.5.html

```
sudo sh -c "certtool --generate-privkey > /etc/ssl/private/cakey.pem"
```

3. Create the template/file `/etc/ssl/ca.info` to define the CA:

```
cn = Example Company
ca
cert_signing_key
```

4. Create the self-signed CA certificate:

```
sudo certtool --generate-self-signed \
--load-privkey /etc/ssl/private/cakey.pem \
--template /etc/ssl/ca.info \
--outfile /etc/ssl/certs/cacert.pem
```

5. Make a private key for the server:

```
sudo certtool --generate-privkey \
--bits 1024 \
--outfile /etc/ssl/private/ldap01_slapd_key.pem
```

Replace *ldap01* in the filename with your server's hostname. Naming the certificate and key for the host and service that will be using them will help keep things clear.

6. Create the `/etc/ssl/ldap01.info` info file containing:

```
organization = Example Company
cn = ldap01.example.com
tls_www_server
encryption_key
signing_key
expiration_days = 3650
```

The above certificate is good for 10 years. Adjust accordingly.

7. Create the server's certificate:

```
sudo certtool --generate-certificate \
--load-privkey /etc/ssl/private/ldap01_slapd_key.pem \
--load-ca-certificate /etc/ssl/certs/cacert.pem \
--load-ca-privkey /etc/ssl/private/cakey.pem \
--template /etc/ssl/ldap01.info \
--outfile /etc/ssl/certs/ldap01_slapd_cert.pem
```

Create the file `certinfo.ldif` with the following contents (adjust accordingly, our example assumes we created certs using https://www.cacert.org):

```
dn: cn=config
add: olcTLSCACertificateFile
olcTLSCACertificateFile: /etc/ssl/certs/cacert.pem
-
```

```
add: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ssl/certs/ldap01_slapd_cert.pem
-
add: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/ssl/private/ldap01_slapd_key.pem
```

Use the ldapmodify command to tell slapd about our TLS work via the slapd-config database:

```
sudo ldapmodify -Y EXTERNAL -H ldapi:/// -f /etc/ssl/certinfo.ldif
```

Contratry to popular belief, you do not need *ldaps://* in `/etc/default/slapd` in order to use encryption. You should have just:

```
SLAPD_SERVICES="ldap:/// ldapi:///"
```

LDAP over TLS/SSL (ldaps://) is deprecated in favour of *StartTLS*. The latter refers to an existing LDAP session (listening on TCP port 389) becoming protected by TLS/SSL whereas LDAPS, like HTTPS, is a distinct encrypted-from-the-start protocol that operates over TCP port 636.

Tighten up ownership and permissions:

```
sudo adduser openldap ssl-cert
sudo chgrp ssl-cert /etc/ssl/private/ldap01_slapd_key.pem
sudo chmod g+r /etc/ssl/private/ldap01_slapd_key.pem
sudo chmod o-r /etc/ssl/private/ldap01_slapd_key.pem
```

Restart OpenLDAP:

```
sudo service slapd restart
```

Check your host's logs (/var/log/syslog) to see if the server has started properly.

1.9. Replication and TLS

If you have set up replication between servers, it is common practice to encrypt (StartTLS) the replication traffic to prevent evesdropping. This is distinct from using encryption with authentication as we did above. In this section we will build on that TLS-authentication work.

The assumption here is that you have set up replication between Provider and Consumer according to *Sezione 1.6, «Replication» [100]* and have configured TLS for authentication on the Provider by following *Sezione 1.8, «TLS» [106]*.

As previously stated, the objective (for us) with replication is high availablity for the LDAP service. Since we have TLS for authentication on the Provider we will require the same on the Consumer. In addition to this, however, we want to encrypt replication traffic. What remains to be done is to create a key and certificate for the Consumer and then configure accordingly. We will generate the

key/certificate on the Provider, to avoid having to create another CA certificate, and then transfer the necessary material over to the Consumer.

1. On the Provider,

   Create a holding directory (which will be used for the eventual transfer) and then the Consumer's private key:

   ```
   mkdir ldap02-ssl
   cd ldap02-ssl
   sudo certtool --generate-privkey \
   --bits 1024 \
   --outfile ldap02_slapd_key.pem
   ```

   Create an info file, `ldap02.info`, for the Consumer server, adjusting it's values accordingly:

   ```
   organization = Example Company
   cn = ldap02.example.com
   tls_www_server
   encryption_key
   signing_key
   expiration_days = 3650
   ```

   Create the Consumer's certificate:

   ```
   sudo certtool --generate-certificate \
   --load-privkey ldap02_slapd_key.pem \
   --load-ca-certificate /etc/ssl/certs/cacert.pem \
   --load-ca-privkey /etc/ssl/private/cakey.pem \
   --template ldap02.info \
   --outfile ldap02_slapd_cert.pem
   ```

   Get a copy of the CA certificate:

   ```
   cp /etc/ssl/certs/cacert.pem .
   ```

   We're done. Now transfer the `ldap02-ssl` directory to the Consumer. Here we use scp (adjust accordingly):

   ```
   cd ..
   scp -r ldap02-ssl user@consumer:
   ```

2. On the Consumer,

   Configure TLS authentication:

   ```
   sudo apt-get install ssl-cert
   sudo adduser openldap ssl-cert
   ```

```
sudo cp ldap02_slapd_cert.pem cacert.pem /etc/ssl/certs
sudo cp ldap02_slapd_key.pem /etc/ssl/private
sudo chgrp ssl-cert /etc/ssl/private/ldap02_slapd_key.pem
sudo chmod g+r /etc/ssl/private/ldap02_slapd_key.pem
sudo chmod o-r /etc/ssl/private/ldap02_slapd_key.pem
```

Create the file `/etc/ssl/certinfo.ldif` with the following contents (adjust accordingly):

```
dn: cn=config
add: olcTLSCACertificateFile
olcTLSCACertificateFile: /etc/ssl/certs/cacert.pem
-
add: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ssl/certs/ldap02_slapd_cert.pem
-
add: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/ssl/private/ldap02_slapd_key.pem
```

Configure the slapd-config database:

```
sudo ldapmodify -Y EXTERNAL -H ldapi:/// -f certinfo.ldif
```

Configure `/etc/default/slapd` as on the Provider (SLAPD_SERVICES).

3. On the Consumer,

Configure TLS for Consumer-side replication. Modify the existing *olcSyncrepl* attribute by tacking on some TLS options. In so doing, we will see, for the first time, how to change an attribute's value(s).

Create the file `consumer_sync_tls.ldif` with the following contents:

```
dn: olcDatabase={1}hdb,cn=config
replace: olcSyncRepl
olcSyncRepl: rid=0 provider=ldap://ldap01.example.com bindmethod=simple
 binddn="cn=admin,dc=example,dc=com" credentials=secret searchbase="dc=example,dc=com"
 logbase="cn=accesslog" logfilter="(&(objectClass=auditWriteObject)(reqResult=0))"
 schemachecking=on type=refreshAndPersist retry="60 +" syncdata=accesslog
 starttls=critical tls_reqcert=demand
```

The extra options specify, respectively, that the consumer must use StartTLS and that the CA certificate is required to verify the Provider's identity. Also note the LDIF syntax for changing the values of an attribute ('replace').

Implement these changes:

```
sudo ldapmodify -Y EXTERNAL -H ldapi:/// -f consumer_sync_tls.ldif
```

And restart slapd:

```
sudo service slapd restart
```

4. On the Provider,

   Check to see that a TLS session has been established. In `/var/log/syslog`, providing you have 'conns'-level logging set up, you should see messages similar to:

```
slapd[3620]: conn=1047 fd=20 ACCEPT from IP=10.153.107.229:57922 (IP=0.0.0.0:389)
slapd[3620]: conn=1047 op=0 EXT oid=1.3.6.1.4.1.1466.20037
slapd[3620]: conn=1047 op=0 STARTTLS
slapd[3620]: conn=1047 op=0 RESULT oid= err=0 text=
slapd[3620]: conn=1047 fd=20 TLS established tls_ssf=128 ssf=128
slapd[3620]: conn=1047 op=1 BIND dn="cn=admin,dc=example,dc=com" method=128
slapd[3620]: conn=1047 op=1 BIND dn="cn=admin,dc=example,dc=com" mech=SIMPLE ssf=0
slapd[3620]: conn=1047 op=1 RESULT tag=97 err=0 text
```

1.10. Autenticazione LDAP

Once you have a working LDAP server, you will need to install libraries on the client that will know how and when to contact it. On Ubuntu, this has been traditionally accomplishd by installing the libnss-ldap package. This package will bring in other tools that will assist you in the configuration step. Install this package now:

```
sudo apt-get install libnss-ldap
```

You will be prompted for details of your LDAP server. If you make a mistake you can try again using:

```
sudo dpkg-reconfigure ldap-auth-config
```

I risultati della configurazione possono essere visualizzati nel file `/etc/ldap.conf`. Se il server richiede delle opzioni non contemplate durante la fase di configurazione. modificare il file secondo le proprie esigenze.

Now configure the LDAP profile for NSS:

```
sudo auth-client-config -t nss -p lac_ldap
```

Configure the system to use LDAP for authentication:

```
sudo pam-auth-update
```

From the menu, choose LDAP and any other authentication mechanisms you need.

You should now be able to log in using LDAP-based credentials.

LDAP clients will need to refer to multiple servers if replication is in use. In `/etc/ldap.conf` you would have something like:

```
uri ldap://ldap01.example.com ldap://ldap02.example.com
```

The request will time out and the Consumer (ldap02) will attempt to be reached if the Provider (ldap01) becomes unresponsive.

If you are going to use LDAP to store Samba users you will need to configure the Samba server to authenticate using LDAP. See *Sezione 2, «Samba e LDAP» [118]* for details.

An alternative to the libnss-ldap package is the libnss-ldapd package. This, however, will bring in the nscd package which is problably not wanted. Simply remove it afterwards.

1.11. Gestire utenti e gruppi

The ldap-utils package comes with enough utilities to manage the directory but the long string of options needed can make them a burden to use. The ldapscripts package contains wrapper scripts to these utilities that some people find easier to use.

Install the package:

```
sudo apt-get install ldapscripts
```

Then edit the file `/etc/ldapscripts/ldapscripts.conf` to arrive at something similar to the following:

```
SERVER=localhost
BINDDN='cn=admin,dc=example,dc=com'
BINDPWDFILE="/etc/ldapscripts/ldapscripts.passwd"
SUFFIX='dc=example,dc=com'
GSUFFIX='ou=Groups'
USUFFIX='ou=People'
MSUFFIX='ou=Computers'
GIDSTART=10000
UIDSTART=10000
MIDSTART=10000
```

Now, create the `ldapscripts.passwd` file to allow rootDN access to the directory:

```
sudo sh -c "echo -n 'secret' > /etc/ldapscripts/ldapscripts.passwd"
sudo chmod 400 /etc/ldapscripts/ldapscripts.passwd
```

Replace «secret» with the actual password for your database's rootDN user.

The scripts are now ready to help manage your directory. Here are some examples of how to use them:

- Creare un nuovo utente:

```
sudo ldapadduser mario example
```

  Viene creato un utente con UID *mario* e imposta il gruppo primario (GID) dell'utente a *example*

- Cambiare la password di un utente:

```
sudo ldapsetpasswd mario
Changing password for user uid=mario,ou=People,dc=example,dc=com
New Password:
New Password (verify):
```

- Eliminare un utente:

```
sudo ldapdeleteuser mario
```

- Aggiungere un gruppo:

```
sudo ldapaddgroup qa
```

- Eliminare un gruppo:

```
sudo ldapdeletegroup qa
```

- Aggiungere un utente a un gruppo:

```
sudo ldapaddusertogroup george qa
```

  Dovrebbe essere possibile visualizzare un attributo *memberUid* per il gruppo *qa* con un valore di *mario*.

- Rimuovere un utente da un gruppo:

```
sudo ldapdeleteuserfromgroup george qa
```

  L'attributo *memberUid* dovrebbe ora essere rimosso dal gruppo *qa*.

- Lo script ldapmodifyuser consente di aggiungere, rimuovere o replicare gli attributi di un utente. Lo script utilizza la stessa sintassi dell'utilità ldapmodify. Per esempio:

```
sudo ldapmodifyuser george
# About to modify the following entry :
dn: uid=george,ou=People,dc=example,dc=com
objectClass: account
objectClass: posixAccount
cn: george
uid: george
uidNumber: 1001
gidNumber: 1001
```

```
homeDirectory: /home/george
loginShell: /bin/bash
gecos: george
description: User account
userPassword:: e1NTSEF9eXFsTFcyWlhwWkF1eGUybVdFWHZKRzJVMjFTSG9vcHk=

# Enter your modifications here, end with CTRL-D.
dn: uid=george,ou=People,dc=example,dc=com
replace: gecos
gecos: Mario Rossi
```

L'utente *gecos* dovrebbe ora essere «Mario Rossi».

- A nice feature of ldapscripts is the template system. Templates allow you to customize the attributes of user, group, and machine objectes. For example, to enable the *user* template edit `/etc/ldapscripts/ldapscripts.conf` changing:

```
UTEMPLATE="/etc/ldapscripts/ldapadduser.template"
```

Diversi *esempi* sono disponibili nella directory `/etc/ldapscripts`. Copiare o rinominare il file `ldapadduser.template.sample` in `/etc/ldapscripts/ldapadduser.template`:

```
sudo cp /usr/share/doc/ldapscripts/examples/ldapadduser.template.sample \
/etc/ldapscripts/ldapadduser.template
```

Edit the new template to add the desired attributes. The following will create new users with an objectClass of inetOrgPerson:

```
dn: uid=<user>,<usuffix>,<suffix>
objectClass: inetOrgPerson
objectClass: posixAccount
cn: <user>
sn: <ask>
uid: <user>
uidNumber: <uid>
gidNumber: <gid>
homeDirectory: <home>
loginShell: <shell>
gecos: <user>
description: User account
title: Employee
```

Notice the *<ask>* option used for the *sn* attribute. This will make ldapadduser prompt you for it's value.

There are utilities in the package that were not covered here. Here is a complete list:

[5] http://manpages.ubuntu.com/manpages/en/man1/ldaprenamemachine.1.html

ldaprenamemachine[5]
ldapadduser[6]
ldapdeleteuserfromgroup[7]
ldapfinger[8]
ldapid[9]
ldapgid[10]
ldapmodifyuser[11]
ldaprenameuser[12]
lsldap[13]
ldapaddusertogroup[14]
ldapsetpasswd[15]
ldapinit[16]
ldapaddgroup[17]
ldapdeletegroup[18]
ldapmodifygroup[19]
ldapdeletemachine[20]
ldaprenamegroup[21]
ldapaddmachine[22]
ldapmodifymachine[23]
ldapsetprimarygroup[24]
ldapdeleteuser[25]

1.12. Backup and Restore

Now we have ldap running just the way we want, it is time to ensure we can save all of our work and restore it as needed.

What we need is a way to backup the ldap database(s), specifically the backend (cn=config) and frontend (dc=example,dc=com). If we are going to backup those databases into, say, `/export/backup`, we could use slapcat as shown in the following script, called `/usr/local/bin/ldapbackup`:

```
#!/bin/bash
```

[6] http://manpages.ubuntu.com/manpages/en/man1/ldapadduser.1.html
[7] http://manpages.ubuntu.com/manpages/en/man1/ldapdeleteuserfromgroup.1.html
[8] http://manpages.ubuntu.com/manpages/en/man1/ldapfinger.1.html
[9] http://manpages.ubuntu.com/manpages/en/man1/ldapid.1.html
[10] http://manpages.ubuntu.com/manpages/en/man1/ldapgid.1.html
[11] http://manpages.ubuntu.com/manpages/en/man1/ldapmodifyuser.1.html
[12] http://manpages.ubuntu.com/manpages/en/man1/ldaprenameuser.1.html
[13] http://manpages.ubuntu.com/manpages/en/man1/lsldap.1.html
[14] http://manpages.ubuntu.com/manpages/en/man1/ldapaddusertogroup.1.html
[15] http://manpages.ubuntu.com/manpages/en/man1/ldapsetpasswd.1.html
[16] http://manpages.ubuntu.com/manpages/en/man1/ldapinit.1.html
[17] http://manpages.ubuntu.com/manpages/en/man1/ldapaddgroup.1.html
[18] http://manpages.ubuntu.com/manpages/en/man1/ldapdeletegroup.1.html
[19] http://manpages.ubuntu.com/manpages/en/man1/ldapmodifygroup.1.html
[20] http://manpages.ubuntu.com/manpages/en/man1/ldapdeletemachine.1.html
[21] http://manpages.ubuntu.com/manpages/en/man1/ldaprenamegroup.1.html
[22] http://manpages.ubuntu.com/manpages/en/man1/ldapaddmachine.1.html
[23] http://manpages.ubuntu.com/manpages/en/man1/ldapmodifymachine.1.html
[24] http://manpages.ubuntu.com/manpages/en/man1/ldapsetprimarygroup.1.html
[25] http://manpages.ubuntu.com/manpages/en/man1/ldapdeleteuser.1.html

```
BACKUP_PATH=/export/backup
SLAPCAT=/usr/sbin/slapcat

nice ${SLAPCAT} -n 0 > ${BACKUP_PATH}/config.ldif
nice ${SLAPCAT} -n 1 > ${BACKUP_PATH}/example.com.ldif
nice ${SLAPCAT} -n 2 > ${BACKUP_PATH}/access.ldif
chmod 640 ${BACKUP_PATH}/*.ldif
```

These files are uncompressed text files containing everything in your ldap databases including the tree layout, usernames, and every password. So, you might want to consider making `/export/backup` an encrypted partition and even having the script encrypt those files as it creates them. Ideally you should do both, but that depends on your security requirements.

Then, it is just a matter of having a cron script to run this program as often as we feel comfortable with. For many, once a day suffices. For others, more often is required. Here is an example of a cron script called `/etc/cron.d/ldapbackup` that is run every night at 22:45h:

```
MAILTO=backup-emails@domain.com
45 22 * * *  root    /usr/local/bin/ldapbackup
```

Now the files are created, they should be copied to a backup server.

Assuming we did a fresh reinstall of ldap, the restore process could be something like this:

```
sudo service slapd stop
sudo mkdir /var/lib/ldap/accesslog
sudo slapadd -F /etc/ldap/slapd.d -n 0 -l /export/backup/config.ldif
sudo slapadd -F /etc/ldap/slapd.d -n 1 -l /export/backup/domain.com.ldif
sudo slapadd -F /etc/ldap/slapd.d -n 2 -l /export/backup/access.ldif
sudo chown -R openldap:openldap /etc/ldap/slapd.d/
sudo chown -R openldap:openldap /var/lib/ldap/
sudo service slapd start
```

1.13. Risorse

- The primary resource is the upstream documentation: *www.openldap.org*[26]
- There are many man pages that come with the slapd package. Here are some important ones, especially considering the material presented in this guide:

  `slapd`[27]
  `slapd-config`[28]
  `slapd.access`[29]

[26] http://www.openldap.org/
[27] http://manpages.ubuntu.com/manpages/en/man8/slapd.8.html
[28] http://manpages.ubuntu.com/manpages/en/man5/slapd-config.5.html
[29] http://manpages.ubuntu.com/manpages/en/man5/slapd.access.5.html

`slapo-syncprov`[30]

- Other man pages:

`auth-client-config`[31]
`pam-auth-update`[32]

- Zytrax's *LDAP for Rocket Scientists*[33]; a less pedantic but comprehensive treatment of LDAP
- A Ubuntu community *OpenLDAP wiki*[34] page has a collection of notes
- O'Reilly's *LDAP System Administration*[35] (textbook; 2003)
- Packt's *Mastering OpenLDAP*[36] (textbook; 2007)

[30] http://manpages.ubuntu.com/manpages/en/man5/slapo-syncprov.5.html
[31] http://manpages.ubuntu.com/manpages/en/man8/auth-client-config.8.html
[32] http://manpages.ubuntu.com/manpages/en/man8/pam-auth-update.8.html
[33] http://www.zytrax.com/books/ldap/
[34] https://help.ubuntu.com/community/OpenLDAPServer
[35] http://www.oreilly.com/catalog/ldapsa/
[36] http://www.packtpub.com/OpenLDAP-Developers-Server-Open-Source-Linux/book

2. Samba e LDAP

This section covers the integration of Samba with LDAP. The Samba server's role will be that of a "standalone" server and the LDAP directory will provide the authentication layer in addition to containing the user, group, and machine account information that Samba requires in order to function (in any of it's 3 possible roles). The pre-requisite is an OpenLDAP server configured with a directory that can accept authentication requests. See *Sezione 1, «Server OpenLDAP» [92]* for details on fulfilling this requirement. Once this section is completed, you will need to decide what specifically you want Samba to do for you and then configure it accordingly.

2.1. Software Installation

There are three packages needed when integrating Samba with LDAP: samba, samba-doc, and smbldap-tools packages.

Strictly speaking, the smbldap-tools package isn't needed, but unless you have some other way to manage the various Samaba entities (users, groups, computers) in an LDAP context then you should install it.

Install these packages now:

```
sudo apt-get install samba samba-doc smbldap-tools
```

2.2. LDAP Configuration

We will now configure the LDAP server so that it can accomodate Samba data. We will perform three tasks in this section:

1. Import a schema
2. Index some entries
3. Add objects

2.2.1. Samba schema

In order for OpenLDAP to be used as a backend for Samba, logically, the DIT will need to use attributes that can properly describe Samba data. Such attributes can be obtained by introducing a Samba LDAP schema. Let's do this now.

For more information on schemas and their installation see *Sezione 1.4, «Modifying the slapd Configuration Database» [97]*.

1. The schema is found in the now-installed samba-doc package. It needs to be unzipped and copied to the `/etc/ldap/schema` directory:

```
sudo cp /usr/share/doc/samba-doc/examples/LDAP/samba.schema.gz /etc/ldap/schema
sudo gzip -d /etc/ldap/schema/samba.schema.gz
```

2. Have the configuration file `schema_convert.conf` that contains the following lines:

```
include /etc/ldap/schema/core.schema
include /etc/ldap/schema/collective.schema
include /etc/ldap/schema/corba.schema
include /etc/ldap/schema/cosine.schema
include /etc/ldap/schema/duaconf.schema
include /etc/ldap/schema/dyngroup.schema
include /etc/ldap/schema/inetorgperson.schema
include /etc/ldap/schema/java.schema
include /etc/ldap/schema/misc.schema
include /etc/ldap/schema/nis.schema
include /etc/ldap/schema/openldap.schema
include /etc/ldap/schema/ppolicy.schema
include /etc/ldap/schema/ldapns.schema
include /etc/ldap/schema/pmi.schema
include /etc/ldap/schema/samba.schema
```

3. Have the directory `ldif_output` hold output.

4. Determine the index of the schema:

```
slapcat -f schema_convert.conf -F ldif_output -n 0 | grep samba,cn=schema

dn: cn={14}samba,cn=schema,cn=config
```

5. Convert the schema to LDIF format:

```
slapcat -f schema_convert.conf -F ldif_output -n0 -H \
ldap:///cn={14}samba,cn=schema,cn=config -l cn=samba.ldif
```

6. Edit the generated `cn=samba.ldif` file by removing index information to arrive at:

```
dn: cn=samba,cn=schema,cn=config
...
cn: samba
```

Remove the bottom lines:

```
structuralObjectClass: olcSchemaConfig
entryUUID: b53b75ca-083f-102d-9fff-2f64fd123c95
creatorsName: cn=config
createTimestamp: 20080827045234Z
entryCSN: 20080827045234.341425Z#000000#000#000000
modifiersName: cn=config
modifyTimestamp: 20080827045234Z
```

Your attribute values will vary.

7. Add the new schema:

```
sudo ldapadd -Q -Y EXTERNAL -H ldapi:/// -f cn\=samba.ldif
```

To query and view this new schema:

```
sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b cn=schema,cn=config 'cn=*samba*'
```

2.2.2. Samba indices

Now that slapd knows about the Samba attributes, we can set up some indices based on them. Indexing entries is a way to improve performance when a client performs a filtered search on the DIT.

Create the file `samba_indices.ldif` with the following contents:

```
dn: olcDatabase={1}hdb,cn=config
changetype: modify
add: olcDbIndex
olcDbIndex: uidNumber eq
olcDbIndex: gidNumber eq
olcDbIndex: loginShell eq
olcDbIndex: uid eq,pres,sub
olcDbIndex: memberUid eq,pres,sub
olcDbIndex: uniqueMember eq,pres
olcDbIndex: sambaSID eq
olcDbIndex: sambaPrimaryGroupSID eq
olcDbIndex: sambaGroupType eq
olcDbIndex: sambaSIDList eq
olcDbIndex: sambaDomainName eq
olcDbIndex: default sub
```

Using the ldapmodify utility load the new indices:

```
sudo ldapmodify -Q -Y EXTERNAL -H ldapi:/// -f samba_indices.ldif
```

If all went well you should see the new indices using ldapsearch:

```
sudo ldapsearch -Q -LLL -Y EXTERNAL -H \
ldapi:/// -b cn=config olcDatabase={1}hdb olcDbIndex
```

2.2.3. Adding Samba LDAP objects

Ora, configurare il pacchetto smbldap-tools in base al proprio ambiente di lavoro. Il pacchetto è dotato di uno script di configurazione che richiede l'impostazione delle opzioni necessarie. Per eseguire lo script:

```
sudo gzip -d /usr/share/doc/smbldap-tools/configure.pl.gz
sudo perl /usr/share/doc/smbldap-tools/configure.pl
```

You may need to comment out the strict pragma in the `configure.pl` file.

Once you have answered the questions, the files `/etc/smbldap-tools/smbldap.conf` and `/etc/smbldap-tools/smbldap_bind.conf` should be generated. If you made any mistakes while executing the script you can always edit the files afterwards.

The smbldap-populate script will add the LDAP objects required for Samba. It is a good idea to first make a backup of your entire directory using slapcat:

```
sudo slapcat -l backup.ldif
```

Once you have a backup proceed to populate your directory:

```
sudo smbldap-populate
```

You can create a LDIF file containing the new Samba objects by executing **sudo smbldap-populate -e samba.ldif**. This allows you to look over the changes making sure everything is correct. If it is, rerun the script without the '-e' switch. Alternatively, you can take the LDIF file and import it's data per usual.

Your LDAP directory now has the necessary information to authenticate Samba users.

2.3. Configurare Samba

There are multiple ways to configure Samba. For details on some common configurations see *Capitolo 18, Reti Windows [277]*. To configure Samba to use LDAP, edit it's configuration file `/etc/samba/smb.conf` commenting out the default *passdb backend* parameter and adding some ldap-related ones:

```
#   passdb backend = tdbsam

# LDAP Settings
   passdb backend = ldapsam:ldap://hostname
   ldap suffix = dc=example,dc=com
   ldap user suffix = ou=People
   ldap group suffix = ou=Groups
   ldap machine suffix = ou=Computers
   ldap idmap suffix = ou=Idmap
   ldap admin dn = cn=admin,dc=example,dc=com
   ldap ssl = start tls
   ldap passwd sync = yes
...
   add machine script = sudo /usr/sbin/smbldap-useradd -t 0 -w "%u"
```

Change the values to match your environment.

Riavviare samba per abilitare le nuove impostazioni:

```
sudo restart smbd
sudo restart nmbd
```

Now inform Samba about the rootDN user's password (the one set during the installation of the slapd package):

```
sudo smbpasswd -w password
```

If you have existing LDAP users that you want to include in your new LDAP-backed Samba they will, of course, also need to be given some of the extra attributes. The smbpasswd utility can do this as well (your host will need to be able to see (enumerate) those users via NSS; install and configure either libnss-ldapd or libnss-ldap):

```
sudo smbpasswd -a NOME_UTENTE
```

You will prompted to enter a password. It will be considered as the new password for that user. Making it the same as before is reasonable.

To manage user, group, and machine accounts use the utilities provided by the smbldap-tools package. Here are some examples:

- To add a new user:

  ```
  sudo smbldap-useradd -a -P NOME_UTENTE
  ```

  The *-a* option adds the Samba attributes, and the *-P* option calls the smbldap-passwd utility after the user is created allowing you to enter a password for the user.

- To remove a user:

  ```
  sudo smbldap-userdel NOME_UTENTE
  ```

  In the above command, use the *-r* option to remove the user's home directory.

- To add a group:

  ```
  sudo smbldap-groupadd -a NOME_GRUPPO
  ```

  As for smbldap-useradd, the *-a* adds the Samba attributes.

- To make an existing user a member of a group:

  ```
  sudo smbldap-groupmod -m NOME_UTENTE NOME_GRUPPO
  ```

  The *-m* option can add more than one user at a time by listing them in comma-separated format.

- To remove a user from a group:

```
sudo smbldap-groupmod -x NOME_UTENTE NOME_GRPPO
```

- To add a Samba machine account:

```
sudo smbldap-useradd -t 0 -w NOME_UTENTE
```

Replace *username* with the name of the workstation. The *-t 0* option creates the machine account without a delay, while the *-w* option specifies the user as a machine account. Also, note the *add machine script* parameter in `/etc/samba/smb.conf` was changed to use smbldap-useradd.

There are utilities in the smbldap-tools package that were not covered here. Here is a complete list:

`smbldap-groupadd`[37]
`smbldap-groupdel`[38]
`smbldap-groupmod`[39]
`smbldap-groupshow`[40]
`smbldap-passwd`[41]
`smbldap-populate`[42]
`smbldap-useradd`[43]
`smbldap-userdel`[44]
`smbldap-userinfo`[45]
`smbldap-userlist`[46]
`smbldap-usermod`[47]
`smbldap-usershow`[48]

2.4. Risorse

- For more information on installing and configuring Samba see *Capitolo 18, Reti Windows [277]* of this Ubuntu Server Guide.
- There are multiple places where LDAP and Samba is documented in the upstream *Samba HOWTO Collection*[49].
- Regarding the above, see specifically the *passdb section*[50].
- Although dated (2007), the *Linux Samba-OpenLDAP HOWTO*[51] contains valuable notes.

[37] http://manpages.ubuntu.com/manpages/en/man8/smbldap-groupadd.8.html
[38] http://manpages.ubuntu.com/manpages/en/man8/smbldap-groupdel.8.html
[39] http://manpages.ubuntu.com/manpages/en/man8/smbldap-groupmod.8.html
[40] http://manpages.ubuntu.com/manpages/en/man8/smbldap-groupshow.8.html
[41] http://manpages.ubuntu.com/manpages/en/man8/smbldap-passwd.8.html
[42] http://manpages.ubuntu.com/manpages/en/man8/smbldap-populate.8.html
[43] http://manpages.ubuntu.com/manpages/en/man8/smbldap-useradd.8.html
[44] http://manpages.ubuntu.com/manpages/en/man8/smbldap-userdel.8.html
[45] http://manpages.ubuntu.com/manpages/en/man8/smbldap-userinfo.8.html
[46] http://manpages.ubuntu.com/manpages/en/man8/smbldap-userlist.8.html
[47] http://manpages.ubuntu.com/manpages/en/man8/smbldap-usermod.8.html
[48] http://manpages.ubuntu.com/manpages/en/man8/smbldap-usershow.8.html
[49] http://samba.org/samba/docs/man/Samba-HOWTO-Collection/
[50] http://samba.org/samba/docs/man/Samba-HOWTO-Collection/passdb.html
[51] http://download.gna.org/smbldap-tools/docs/samba-ldap-howto/

- The main page of the *Samba Ubuntu community documentation*[52] has a plethora of links to articles that may prove useful.

[52] https://help.ubuntu.com/community/Samba#samba-ldap

3. Kerberos

Kerberos è un sistema di autenticazione di rete basato sul principio di un "agente" terzo fidato. Le altre due parti sono l'utente e il servizio a cui l'utente vuole autenticarsi. Non tutti i serivizi e le applicazioni possono usare Kerberos, ma quelle che ne sono in grado, consentono di portare la rete a essere un SSO (Single Sign On).

Questa sezione spiega come installare e configurare un server Kerberos, fornendo alcuni esempi di configurazione.

3.1. Panoramica

Se si è nuovi di Kerberos, ci sono alcuni termini che è bene comprendere prima di procedere. Molti di questi termini potrebbero essere simili ad altri concetti di altri ambienti più familiari.

- *Principal*: qualsiasi utente, computer e servizio fornito da server deve essere difinito come "Kerberos Principal".
- *Istanze*: usate dai principal di servizio e da quelli amministrativi.
- *Realms:* the unique realm of control provided by the Kerberos installation. Think of it as the domain or group your hosts and users belong to. Convention dictates the realm should be in uppercase. By default, ubuntu will use the DNS domain converted to uppercase (EXAMPLE.COM) as the realm.
- *Key Distribution Center* (KDC): consiste di tre parti, un database di tutti i principal, il server di autenticazione e il server che garantisce i ticket. Per ogni reame deve esserci almeno un KDC.
- *Ticket Granting Ticket* (TGT): emesso dallo "Authentication Server" (AS), il "Ticket Granting Ticket" è cifrato con la password dell'utente ed è quindi noto solo all'utente e al KDC.
- *Ticket Granting Server* (TGS): emette i ticket su richiesta dei client.
- *Ticket:*: conferma l'identità dei due principal. Uno è l'utente e l'altro il servizio richiesto. Il ticket stabilisce una chiave di cifratura usata per garantire la sicurezza della comunicazione durante la fase di autenticazione.
- *File keytab*: sono file estratti dal KDC e contengono le chiavi di cifratura per un servizio o un host.

To put the pieces together, a Realm has at least one KDC, preferably more for redundancy, which contains a database of Principals. When a user principal logs into a workstation that is configured for Kerberos authentication, the KDC issues a Ticket Granting Ticket (TGT). If the user supplied credentials match, the user is authenticated and can then request tickets for Kerberized services from the Ticket Granting Server (TGS). The service tickets allow the user to authenticate to the service without entering another username and password.

3.2. Server Kerberos

3.2.1. Installazione

For this discussion, we will create a MIT Kerberos domain with the following features (edit them to fit your needs):

- *Realm:* EXAMPLE.COM
- *Primary KDC:* kdc01.example.com (192.168.0.1)
- *Secondary KDC:* kdc02.example.com (192.168.0.2)
- *User principal:* steve
- *Admin principal:* steve/admin

It is *strongly* recommended that your network-authenticated users have their uid in a different range (say, starting at 5000) than that of your local users.

Before installing the Kerberos server a properly configured DNS server is needed for your domain. Since the Kerberos Realm by convention matches the domain name, this section uses the *EXAMPLE.COM* domain configured in *Sezione 2.3, «Server primario» [143]* of the DNS documentation.

Also, Kerberos is a time sensitive protocol. So if the local system time between a client machine and the server differs by more than five minutes (by default), the workstation will not be able to authenticate. To correct the problem all hosts should have their time synchronized using the same *Network Time Protocol (NTP)* server. For details on setting up NTP see *Sezione 4, «Sincronizzazione del tempo con NTP» [51]*.

The first step in creating a Kerberos Realm is to install the krb5-kdc and krb5-admin-server packages. From a terminal enter:

```
sudo apt-get install krb5-kdc krb5-admin-server
```

You will be asked at the end of the install to supply the hostname for the Kerberos and Admin servers, which may or may not be the same server, for the realm.

By default the realm is created from the KDC's domain name.

Creare il reame con l'utilità kdb5_newrealm:

```
sudo krb5_newrealm
```

3.2.2. Configurazione

The questions asked during installation are used to configure the `/etc/krb5.conf` file. If you need to adjust the Key Distribution Center (KDC) settings simply edit the file and restart the krb5-kdc

daemon. If you need to reconfigure Kerberos from scratch, perhaps to change the realm name, you can do so by typing

```
sudo dpkg-reconfigure krb5-kdc
```

1. Once the KDC is properly running, an admin user -- the *admin principal* -- is needed. It is recommended to use a different username from your everyday username. Using the kadmin.local utility in a terminal prompt enter:

   ```
   sudo kadmin.local
   Authenticating as principal root/admin@EXAMPLE.COM with password.
   kadmin.local: addprinc steve/admin
   WARNING: no policy specified for steve/admin@EXAMPLE.COM; defaulting to no policy
   Enter password for principal "steve/admin@EXAMPLE.COM":
   Re-enter password for principal "steve/admin@EXAMPLE.COM":
   Principal "steve/admin@EXAMPLE.COM" created.
   kadmin.local: quit
   ```

   In the above example *steve* is the *Principal*, */admin* is an *Instance*, and *@EXAMPLE.COM* signifies the realm. The *"every day"* Principal, a.k.a. the *user principal*, would be *steve@EXAMPLE.COM*, and should have only normal user rights.

   

   Sostituire *EXAMPLE.COM* e *steve* con il proprio reame e il nome utente dell'amministratore.

2. Il nuovo utente amministratore necessita dei permessi ACL (Access Control List) corretti, configurati tramite il file `/etc/krb5kdc/kadm5.acl`:

   ```
   steve/admin@EXAMPLE.COM        *
   ```

   This entry grants *steve/admin* the ability to perform any operation on all principals in the realm. You can configure principals with more restrictive privileges, which is convenient if you need an admin principal that junior staff can use in Kerberos clients. Please see the *kadm5.acl* man page for details.

3. Riavviare krb5-admin-server affinché le nuove ACL abbiano effetto:

   ```
   sudo /etc/init.d/krb5-admin-server restart
   ```

4. Il nuovo utente può essere provato con l'utilità kinit:

   ```
   kinit steve/admin
   steve/admin@EXAMPLE.COM's Password:
   ```

   Una volta inserita la password, usare l'utilità klist per visualizzare le informazioni riguardo il TGT (Ticket Granting Ticket):

```
klist
Credentials cache: FILE:/tmp/krb5cc_1000
        Principal: steve/admin@EXAMPLE.COM

  Issued           Expires          Principal
Jul 13 17:53:34  Jul 14 03:53:34  krbtgt/EXAMPLE.COM@EXAMPLE.COM
```

Where the cache filename `krb5cc_1000` is composed of the prefix `krb5cc_` and the user id (uid), which in this case is `1000`. You may need to add an entry into the `/etc/hosts` for the KDC so the client can find the KDC. For example:

```
192.168.0.1   kdc01.example.com       kdc01
```

Replacing *192.168.0.1* with the IP address of your KDC. This usually happens when you have a Kerberos realm encompassing different networks separated by routers.

5. The best way to allow clients to automatically determine the KDC for the Realm is using DNS SRV records. Add the following to `/etc/named/db.example.com`:

```
_kerberos._udp.EXAMPLE.COM.     IN SRV 1  0 88  kdc01.example.com.
_kerberos._tcp.EXAMPLE.COM.     IN SRV 1  0 88  kdc01.example.com.
_kerberos._udp.EXAMPLE.COM.     IN SRV 10 0 88  kdc02.example.com.
_kerberos._tcp.EXAMPLE.COM.     IN SRV 10 0 88  kdc02.example.com.
_kerberos-adm._tcp.EXAMPLE.COM. IN SRV 1  0 749 kdc01.example.com.
_kpasswd._udp.EXAMPLE.COM.      IN SRV 1  0 464 kdc01.example.com.
```

Sostituire *EXAMPLE.COM*, *kdc01* e *kdc02* con il nome del proprio dominio, il KDC primario e quello secondario.

Consultare *Capitolo 8, DNS (Domain Name Service) [140]* per le istruzioni sulla configurazione di DNS.

Il reame Kerberos è ora pronto per autenticare i client.

3.3. KDC secondario

Once you have one Key Distribution Center (KDC) on your network, it is good practice to have a Secondary KDC in case the primary becomes unavailable. Also, if you have Kerberos clients that are in different networks (possibly separated by routers using NAT), it is wise to place a secondary KDC in each of those networks.

1. Per prima cosa installare il pacchetto e quando vengono chiesti i nomi di Kerberos e Admin, inserire il nome del KDC primario:

```
sudo apt-get install krb5-kdc krb5-admin-server
```

2. Una volta installato il pacchetto, creare il KDC secondario. Da un terminale, digitare:

```
kadmin -q "addprinc -randkey host/kdc02.example.com"
```

Una volta eseguiti i comandi kadmin viene chiesto la propria password *NOME_UTENTE/ADMIN@EXAMPLE.COM*.

3. Estrarre il file *keytab_*

```
kadmin -q "ktadd -norandkey -k keytab.kdc02 host/kdc02.example.com"
```

4. Dovrebbe esserci un file `keytab.kdc02` nella directory corrente, spostare il file in `/etc/krb5.keytab`:

```
sudo mv keytab.kdc02 /etc/krb5.keytab
```

Se il percorso a `keytab.kdc02` è diverso, modificarlo in base al proprio caso.

È possibile elencare tutti i principal presenti in un file Keytab, utile durante la risoluzione dei problemi, con l'utilità klist:

```
sudo klist -k /etc/krb5.keytab
```

The -k option indicates the file is a keytab file.

5. Dovrebbe esserci un file `kpropd.acl` in ogni KDC che presenti tutti i KDC del reame. Per esempio, sia sul KDC primario che secondario, creare un file `/etc/krb5kdc/kpropd.acl`:

```
host/kdc01.example.com@EXAMPLE.COM
host/kdc02.example.com@EXAMPLE.COM
```

6. Creare un database vuoto nel *KDC secondario*:

```
sudo kdb5_util -s create
```

7. Avviare il demone kpropd che resterà in ascolto per le connessioni dall'utilità kprop. kprop è usato per trasferire i file di dump:

```
sudo kpropd -S
```

8. Da un terminale dal *KDC primario*, creare un file di dump del database principale:

```
sudo kdb5_util dump /var/lib/krb5kdc/dump
```

9. Estrarre il *keytab* del KDC primario e copiarlo in `/etc/krb5.keytab`:

```
kadmin -q "ktadd -k keytab.kdc01 host/kdc01.example.com"
sudo mv keytab.kdc01 /etc/krb5.keytab
```

Assicurarsi che ci sia un *host* per *kdc01.example.com* prima di estrarre il keytab.

10. Usando l'utilità kprop eseguire il push del database sul KDC secondario:

```
sudo kprop -r EXAMPLE.COM -f /var/lib/krb5kdc/dump kdc02.example.com
```

Dovrebbe essere visualizzato un messaggio di *SUCCEEDED* se la propagazione è andata a buon fine. Se si è verificato un errore, per maggiori informazioni, controllare `/var/log/syslog` sul KDC secondario.

You may also want to create a cron job to periodically update the database on the Secondary KDC. For example, the following will push the database every hour (note the long line has been split to fit the format of this document):

```
# m h  dom mon dow   command
0 * * * * /usr/sbin/kdb5_util dump /var/lib/krb5kdc/dump &&
/usr/sbin/kprop -r EXAMPLE.COM -f /var/lib/krb5kdc/dump kdc02.example.com
```

11. Sempre nel *KDC secondario*, creare un file *stash* in cui salvare la chiave principale di Kerberos:

```
sudo kdb5_util stash
```

12. Avviare il demone krb5-kdc sul KDC secondario:

```
sudo /etc/init.d/krb5-kdc start
```

The *Secondary KDC* should now be able to issue tickets for the Realm. You can test this by stopping the krb5-kdc daemon on the Primary KDC, then by using kinit to request a ticket. If all goes well you should receive a ticket from the Secondary KDC. Otherwise, check `/var/log/syslog` and `/var/log/auth.log` in the Secondary KDC.

3.4. Client Kerberos Linux

Questa sezione spiega come configurare un sistema Linux come un client Kerberos consentendo l'accesso a qualsiasi servizio Kerberos ad accesso effettuato correttamente da parte degli utenti.

3.4.1. Installazione

Per autenticarsi in un reame Kerberos sono necessari i pacchetti krb5-user e libpam-krb5 oltre ad altri non strettamente necessari, ma che semplificano molto la gestione. Per installare questi pacchetti, digitare:

```
sudo apt-get install krb5-user libpam-krb5 libpam-ccreds auth-client-config
```

Il pacchetto auth-client-config consente una semplice configurazione dell'autenticazione PAM per diverse sorgenti e libpam-ccreds memorizza le credenziali di autenticazione consentendo di effettuare

l'accesso anche se il KDC non è disponibile. Questo pacchetto è utile anche per i computer portatili che possono autenticarsi su reti aziendali, ma devono essere in grado di farlo anche al di fuori della rete.

3.4.2. Configurazione

Per configurare il client, in un terminale digitare:

```
sudo dpkg-reconfigure krb5-config
```

Viene quindi chiesto di inserire il nome del reame Kerberos. Inoltre, se non si dispone di un DNS configurato con i record *SRV* di Kerberos, viene richiesto il nome dell'host del KDC e del server amministrativo.

Il comando dpkg-reconfigure aggiunge delle voci al file `/etc/krb5.conf` del proprio reame. Dovrebbero essere disponibili delle voci simili alle seguenti:

```
[libdefaults]
        default_realm = EXAMPLE.COM
...
[realms]
        EXAMPLE.COM = }
                kdc = 192.168.0.1
                admin_server = 192.168.0.1
        }
```

If you set the uid of each of your network-authenticated users to start at 5000, as suggested in *Sezione 3.2.1, «Installazione» [126]*, you can then tell pam to only try to authenticate using Kerberos users with uid > 5000:

```
# Kerberos should only be applied to ldap/kerberos users, not local ones.
for i in common-auth common-session common-account common-password; do
 sudo sed -i -r \
 -e 's/pam_krb5.so minimum_uid=1000/pam_krb5.so minimum_uid=5000/' \
 /etc/pam.d/$i
done
```

This will avoid being asked for the (non-existent) Kerberos password of a locally authenticated user when changing its password using **passwd**.

Per avviare la configurazione, richiedere un ticket usando l'utilità kinit. Per esempio:

```
kinit steve@EXAMPLE.COM
Password for steve@EXAMPLE.COM:
```

Una volta ottenuto un ticket, i dettagli possono essere visualizzati usando klist:

```
klist
Ticket cache: FILE:/tmp/krb5cc_1000
Default principal: steve@EXAMPLE.COM

Valid starting     Expires            Service principal
07/24/08 05:18:56  07/24/08 15:18:56  krbtgt/EXAMPLE.COM@EXAMPLE.COM
        renew until 07/25/08 05:18:57


Kerberos 4 ticket cache: /tmp/tkt1000
klist: You have no tickets cached
```

Usare auth-client-config per configurare il modulo libpam-krb5 affinché richieda un ticket durante la fase di accesso:

```
sudo auth-client-config -a -p kerberos_example
```

Una volta autenticati con successo, si dovrebbe ricevere un ticket.

3.5. Risorse

- For more information on MIT's version of Kerberos, see the *MIT Kerberos*[53] site.
- The *Ubuntu Wiki Kerberos*[54] page has more details.
- Il libro *Kerberos: The Definitive Guide*[55] di O'Reilly è un ottimo punto di riferimento per impostare un server Kerberos.
- Also, feel free to stop by the *#ubuntu-server* and *#kerberos* IRC channels on *Freenode*[56] if you have Kerberos questions.

[53] http://web.mit.edu/Kerberos/
[54] https://help.ubuntu.com/community/Kerberos
[55] http://oreilly.com/catalog/9780596004033/
[56] http://freenode.net/

4. Kerberos e LDAP

Most people will not use Kerberos by itself; once an user is authenticated (Kerberos), we need to figure out what this user can do (authorization). And that would be the job of programs such as LDAP.

Sostituire un database principale di Kerberos tra due server può essere complicato e aggiunge un ulteriori database all'interno della rete. Il server Kerberos può comunque essere configurato per utilizzare una directory LDAP come database principale. In questa sezione viene descritto come configurare un server Kerberos, primario e secondario, affinché utilizzi OpenLDAP come database principale.

The examples presented here assume MIT Kerberos and OpenLDAP.

4.1. Configurare OpenLDAP

Per prima cosa è necessario caricare lo *schema* all'interno del server OpenLDAP collegato ai KDC primario e secondario. I successivi passi qui descritti hanno come presupposto la presenza di un server LDAP di replica configurato tra due server. Per maggiori informazioni su come impostare un server OpenLDAP, consultare *Sezione 1, «Server OpenLDAP» [92]*.

È inoltre richiesto per configurare OpenLDAP all'uso di connessioni TLS e SSL, in modo che il traffico tra il KDC e il server LDAP sia cifrato. Per maggiori informazioni, consultare *Sezione 1.8, «TLS» [106]*.

`cn=admin,cn=config` is a user we created with rights to edit the ldap database. Many times it is the RootDN. Change its value to reflect your setup.

- Per caricare lo schema all'interno del server LDAP, installare su tale server il pacchetto krb5-kdc-ldap. Da un terminale, digitare:

  ```
  sudo apt-get install krb5-kdc-ldap
  ```

- Estrarre il file `kerberos.schema.gz`:

  ```
  sudo gzip -d /usr/share/doc/krb5-kdc-ldap/kerberos.schema.gz
  sudo cp /usr/share/doc/krb5-kdc-ldap/kerberos.schema /etc/ldap/schema/
  ```

- Lo schema *kerberos* deve essere aggiunto all'albero *cn=config*. La procedura per aggiungere un nuovo schema a slapd è descritta anche in *Sezione 1.4, «Modifying the slapd Configuration Database» [97]*.

  1. Creare un file di configurazione chiamato `schema_convert.conf`, o simile, contenente quanto segue:

     ```
     include /etc/ldap/schema/core.schema
     ```

```
include /etc/ldap/schema/collective.schema
include /etc/ldap/schema/corba.schema
include /etc/ldap/schema/cosine.schema
include /etc/ldap/schema/duaconf.schema
include /etc/ldap/schema/dyngroup.schema
include /etc/ldap/schema/inetorgperson.schema
include /etc/ldap/schema/java.schema
include /etc/ldap/schema/misc.schema
include /etc/ldap/schema/nis.schema
include /etc/ldap/schema/openldap.schema
include /etc/ldap/schema/ppolicy.schema
include /etc/ldap/schema/kerberos.schema
```

2. Creare una directory temporanea in cui salvare i file LDIF:

```
mkdir /tmp/ldif_output
```

3. Usare quindi slapcat per convertire i file schema:

```
slapcat -f schema_convert.conf -F /tmp/ldif_output -n0 -s \
"cn={12}kerberos,cn=schema,cn=config" > /tmp/cn=kerberos.ldif
```

   Modificare i percorsi e i nomi dei file in base alla proprie esigenze.

4. Modificare il file `/tmp/cn\=kerberos.ldif` generato sistemando i seguenti attributi:

```
dn: cn=kerberos,cn=schema,cn=config
...
cn: kerberos
```

   Rimuovere le seguenti righe dalla fine del file:

```
structuralObjectClass: olcSchemaConfig
entryUUID: 18ccd010-746b-102d-9fbe-3760cca765dc
creatorsName: cn=config
createTimestamp: 20090111203515Z
entryCSN: 20090111203515.326445Z#000000#000#000000
modifiersName: cn=config
modifyTimestamp: 20090111203515Z
```

   I valori degli attributi possono variare, basta solo assicurarsi che gli attributi siano rimossi.

5. Caricare il nuovo schema con ldapadd:

```
ldapadd -x -D cn=admin,cn=config -W -f /tmp/cn\=kerberos.ldif
```

6. Aggiungere un indice per l'attributo *krb5principalname*:

```
ldapmodify -x -D cn=admin,cn=config -W
Enter LDAP Password:
dn: olcDatabase={1}hdb,cn=config
```

```
add: olcDbIndex
olcDbIndex: krbPrincipalName eq,pres,sub

modifying entry "olcDatabase={1}hdb,cn=config"
```

7. Infine, aggiornare le ACL (Access Control Lists):

```
ldapmodify -x -D cn=admin,cn=config -W
Enter LDAP Password:
dn: olcDatabase={1}hdb,cn=config
replace: olcAccess
olcAccess: to attrs=userPassword,shadowLastChange,krbPrincipalKey by
 dn="cn=admin,dc=example,dc=com" write by anonymous auth by self write by * none
-
add: olcAccess
olcAccess: to dn.base="" by * read
-
add: olcAccess
olcAccess: to * by dn="cn=admin,dc=example,dc=com" write by * read

modifying entry "olcDatabase={1}hdb,cn=config"
```

La directory LDAP è ora pronta come database principale per Kerberos.

4.2. Configurazione KDC primario

Configurato OpenLDAP, è necessario configurare KDC.

- Installare i pacchetti necessari. In un terminale, digitare:

```
sudo apt-get install krb5-kdc krb5-admin-server krb5-kdc-ldap
```

- Modificare `/etc/krb5.conf` aggiungendo le seguenti opzioni all'interno delle sezioni appropriate:

```
[libdefaults]
        default_realm = EXAMPLE.COM

...

[realms]
        EXAMPLE.COM = {
                kdc = kdc01.example.com
                kdc = kdc02.example.com
                admin_server = kdc01.example.com
                admin_server = kdc02.example.com
                default_domain = example.com
                database_module = openldap_ldapconf
        }

...
```

```
[domain_realm]
        .example.com = EXAMPLE.COM


...

[dbdefaults]
        ldap_kerberos_container_dn = dc=example,dc=com

[dbmodules]
        openldap_ldapconf = {
                db_library = kldap
                ldap_kdc_dn = "cn=admin,dc=example,dc=com"

                # this object needs to have read rights on
                # the realm container, principal container and realm sub-trees
                ldap_kadmind_dn = "cn=admin,dc=example,dc=com"

                # this object needs to have read and write rights on
                # the realm container, principal container and realm sub-trees
                ldap_service_password_file = /etc/krb5kdc/service.keyfile
                ldap_servers = ldaps://ldap01.example.com ldaps://ldap02.example.com
                ldap_conns_per_server = 5
        }
```

Modificare *example.com*, *dc=example,dc=com*, *cn=admin,dc=example,dc=com* e *ldap01.example.com* con i valori corretti del dominio, dell'oggetto e del server LDAP della propria rete.

- Usare l'utilità kdb5_ldap_util per creare il reame:

```
sudo kdb5_ldap_util -D  cn=admin,dc=example,dc=com create -subtrees \
dc=example,dc=com -r EXAMPLE.COM -s -H ldap://ldap01.example.com
```

- Creare un file stash della password utilizzata per l'associazione al server LDAP. Questa password è usata con le opzioni *ldap_kdc_dn* e *ldap_kadmin_dn* nel file `/etc/krb5.conf`:

```
sudo kdb5_ldap_util -D  cn=admin,dc=example,dc=com stashsrvpw -f \
/etc/krb5kdc/service.keyfile cn=admin,dc=example,dc=com
```

- Copiare il certificato della CA dal server LDAP:

```
scp ldap01:/etc/ssl/certs/cacert.pem .
sudo cp cacert.pem /etc/ssl/certs
```

Modificare il file `/etc/ldap/ldap.conf` affinché utilizzi il certificato:

```
TLS_CACERT /etc/ssl/certs/cacert.pem
```

Il certificato deve anche essere copiato nel KDC secondario per consentire la connessione ai server LDAP utilizzando LDAPS.

Ora è possibile aggiungere i principal Kerberos al database LDAP che verranno copiati su tutti gli altri server LDAP di replica. Per aggiungere un principal utilizzando l'utilità kadmin.local, digitare:

```
sudo kadmin.local
Authenticating as principal root/admin@EXAMPLE.COM with password.
kadmin.local:  addprinc -x dn="uid=steve,ou=people,dc=example,dc=com" steve
WARNING: no policy specified for steve@EXAMPLE.COM; defaulting to no policy
Enter password for principal "steve@EXAMPLE.COM":
Re-enter password for principal "steve@EXAMPLE.COM":
Principal "steve@EXAMPLE.COM" created.
```

Dovrebbero ora essere aggiunti all'oggetto utente *uid=steve,ou=people,dc=example,dc=com* gli attributi krbPrincipalName, krbPrincipalKey, krbLastPwdChange e krbExtraData. Per verificare che all'utente venga emesso un ticket, utilizzare le utilità kinit e klist.

Se l'oggetto utente è già stato creato, è necessario usare l'opzione *-x dn="..."* per aggiungere gli attributi Kerberos, altrimenti verrà creato un nuovo oggetto *principal* nel sottoalbero del reame.

4.3. Configurazione KDC secondario

La configurazione di un KDC secondario utilizzando il backend LDAP è molto simile alla configurazione tramite l'utilizzo del database Kerberos.

1. Installare i pacchetti necessari. In un terminale digitare:

```
sudo apt-get install krb5-kdc krb5-admin-server krb5-kdc-ldap
```

2. Modificare il file `/etc/krb5.conf` affinché utilizzi il backend LDAP:

```
[libdefaults]
        default_realm = EXAMPLE.COM

...

[realms]
        EXAMPLE.COM = {
                kdc = kdc01.example.com
                kdc = kdc02.example.com
                admin_server = kdc01.example.com
                admin_server = kdc02.example.com
                default_domain = example.com
                database_module = openldap_ldapconf
```

```
        }

...

[domain_realm]
        .example.com = EXAMPLE.COM

...

[dbdefaults]
        ldap_kerberos_container_dn = dc=example,dc=com

[dbmodules]
        openldap_ldapconf = {
                db_library = kldap
                ldap_kdc_dn = "cn=admin,dc=example,dc=com"

                # this object needs to have read rights on
                # the realm container, principal container and realm sub-trees
                ldap_kadmind_dn = "cn=admin,dc=example,dc=com"

                # this object needs to have read and write rights on
                # the realm container, principal container and realm sub-trees
                ldap_service_password_file = /etc/krb5kdc/service.keyfile
                ldap_servers = ldaps://ldap01.example.com ldaps://ldap02.example.com
                ldap_conns_per_server = 5
        }
```

3. Creare il file stash per la password di associazione LDAP:

```
sudo kdb5_ldap_util -D  cn=admin,dc=example,dc=com stashsrvpw -f \
/etc/krb5kdc/service.keyfile cn=admin,dc=example,dc=com
```

4. Dal *KDC primario*, copiare il file di stash della *chiave primaria* (`/etc/krb5kdc/.k5.EXAMPLE.COM`) nel KDC secondario. Accertarsi di copiare tale file utilizzando una connessione cifrata come scp o su un supporto fisico.

```
sudo scp /etc/krb5kdc/.k5.EXAMPLE.COM steve@kdc02.example.com:~
sudo mv .k5.EXAMPLE.COM /etc/krb5kdc/
```

Ricordarsi di sostituire *EXAMPLE.COM* con il reame in uso.

5. Back on the *Secondary KDC*, (re)start the ldap server only,

```
sudo service slapd restart
```

6. Infine. avviare il demone krb5-kdc:

```
sudo /etc/init.d/krb5-kdc start
```

7. Verify the two ldap servers (and kerberos by extension) are in sync.

All'interno della propria rete sono quindi disponibili dei KDC ridondanti che assieme ai server LDAP ridondanti permettono l'autenticazione degli utenti anche nel caso in cui un server LDAP, un server Kerberos o uno server LDAP e un server Kerberos non siano più disponibili.

4.4. Risorse

- Maggiori informazioni possono essere trovate nella *Kerberos Admin Guide*[57].
- For more information on kdb5_ldap_util see *Section 5.6*[58] and the *kdb5_ldap_util man page*[59].
- Another useful link is the *krb5.conf man page*[60].
- Also, see the *Kerberos and LDAP*[61] Ubuntu wiki page.

[57] http://web.mit.edu/Kerberos/krb5-1.6/krb5-1.6.3/doc/krb5-admin.html#Configuring-Kerberos-with-OpenLDAP-back_002dend
[58] http://web.mit.edu/Kerberos/krb5-1.6/krb5-1.6.3/doc/krb5-admin.html#Global-Operations-on-the-Kerberos-LDAP-Database
[59] http://manpages.ubuntu.com/manpages/precise/en/man8/kdb5_ldap_util.8.html
[60] http://manpages.ubuntu.com/manpages/precise/en/man5/krb5.conf.5.html
[61] https://help.ubuntu.com/community/Kerberos#kerberos-ldap

Capitolo 8. DNS (Domain Name Service)

Il DNS (Domain Name Service) è un servizio Internet che mappa gli indirizzi IP e i nomi di dominio univoci (FQDN) tra di loro facendo in modo di non dover ricordare gli indirizzi IP. I computer che eseguono DNS sono chiamati *server dei nomi*. Ubuntu è dotato di BIND (Berkley Internet Naming Daemon), il più diffuso programma usato per mantenere un server dei nomi su Linux.

1. Installazione

A un promt di terminale, inserire il seguente comando per installare dns:

```
sudo apt-get install bind9
```

A very useful package for testing and troubleshooting DNS issues is the dnsutils package. Very often these tools will be installed already, but to check and/or install dnsutils enter the following:

```
sudo apt-get install dnsutils
```

2. Configurazione

BIND9 può essere configurato in diversi modi tra cui: come cache per server dei nomi, master principale e master secondario.

- Quando configurato come un server dei nomi cache, BIND9 troverà la risposta alle interrogazioni sui nomi e la archivierà.
- Come server primario, BIND9 legge i dati per una zona da un file ed è autoritativo per quella zona.
- Nella configurazione come server secondario, BIN9 ottiene i dati della zona da un altro server dei nomi per quella zona.

2.1. Panoramica

I file di configurazione di DNS sono archiviati nella directory `/etc/bind`, il file di configurazione principale è `/etc/bind/named.conf`.

La riga *include* specifica il nome del file contenente le opzioni DNS, la riga *directory* nel file `/etc/bind/named.conf.options` indica a DNS dove cercare i file. Tutti i file usati da BIND sono presenti in questa directory.

Il file `/etc/bind/db.root` descrive i server dei nomi "radice" nel mondo. Questi server cambiano col tempo, quindi il file `/etc/bind/db.root` deve essere aggiornato ogni tanto, procedura che viene svolta, solitamente, con gli aggiornamenti al pacchetto bind9. La sezione *zone* definisce un server principale ed è archiviata in un file indicato dall'opzione *file*.

È possibile configurare lo stesso server sia come server dei nomi cache, master primario e secondario. Un server può ricoprire il ruolo di "Start of Authority" (SOA) per una zona, fornendo allo stesso tempo servizi di server secondario per un'altra zona e di cache per gli host della LAN.

2.2. Server dei nomi cache

La configurazione predefinita comporta l'utilizzo come server di cache. È necessario solamente aggiungere gli indirizzi IP dei server DNS del proprio ISP. De-commentare e modificare quanto segue nel file `/etc/bind/named.conf.options`:

```
forwarders {
                1.2.3.4;
                5.6.7.8;
          };
```

Sostituire *1.2.3.4* e *5.6.7.8* con gli indirizzi IP del server di nomi attuale.

Per abilitare la nuova configurazione è necessario riavviare il server DNS. Da un terminale, digitare:

```
sudo service bind9 restart
```

Per maggiori informazioni su come eseguire test su un server cache DNS, consultare *Sezione 3.1.2, «dig» [148]*.

2.3. Server primario

In questa sezione, BIND9 viene configurato come server primario per il dominio *example.com*. Basta sostituire *example.com* con il proprio FQDN (Fully Qualified Domain Name).

2.3.1. File zona forward

Per aggiungere una zona DNS a BIND9, trasformando BIND9 in un server primario, la prima cosa da fare è modificare il file `/etc/bind/named.conf.local`:

```
zone "example.com" {
 type master;
        file "/etc/bind/db.example.com";
};
```

Prendere un file zona esistente come modello per creare il file `/etc/bind/db.example.com`:

```
sudo cp /etc/bind/db.local /etc/bind/db.example.com
```

Edit the new zone file `/etc/bind/db.example.com` change *localhost.* to the FQDN of your server, leaving the additional "." at the end. Change *127.0.0.1* to the nameserver's IP Address and *root.localhost* to a valid email address, but with a "." instead of the usual "@" symbol, again leaving the "." at the end. Change the comment to indicate the domain that this file is for.

Create an *A record* for the base domain, *example.com*. Also, create an *A record* for *ns.example.com*, the name server in this example:

```
;
; BIND data file for example.com
;
$TTL    604800
@       IN      SOA     example.com. root.example.com. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
        IN      A       192.168.1.10
;
@       IN      NS      ns.example.com.
@       IN      A       192.168.1.10
@       IN      AAAA    ::1
ns      IN      A       192.168.1.10
```

È necessario incrementare il numero *Serial* ogni volta che vengono apportate modifiche al file zona. Se vengono eseguite molteplici modifiche prima di riavviare BIND, incrementare il valore solo una volta.

Ora è possibile aggiungere voci DNS alla fine del file zona. Per maggiori informazioni, consultare la *Sezione 4.1, «Tipi di record comuni» [152]*.

Many admins like to use the last date edited as the serial of a zone, such as *2012010100* which is yyyymmddss (where *ss* is the Serial Number)

Once you have made changes to the zone file BIND9 needs to be restarted for the changes to take effect:

```
sudo service bind9 restart
```

2.3.2. File zona reverse

Una volta configurata la zona e la risoluzione dei nomi con un indirizzo IP, è necessaria anche una zona *Reverse*. Una zona "Reverse" consente a DNS di trasformare un indirizzo in un nome.

Modificare il file /etc/bind/named.conf.local aggiungendo quanto segue:

```
zone "1.168.192.in-addr.arpa" {
        type master;
        file "/etc/bind/db.192";
};
```

Sostituire *1.168.192* con i primi tre valori dell'indirizzo della rete che si sta usando. Inoltre, chiamare il file zona `/etc/bind/db.192` in modo appropriato, in modo tale che rispecchi il primo ottetto della propria rete.

Creare il file `/etc/bind/db.192`:

```
sudo cp /etc/bind/db.127 /etc/bind/db.192
```

Quindi modificare `/etc/bind/db.192` cambiando le stesse opzioni di `/etc/bind/db.example.com`:

```
;
; BIND reverse data file for local 192.168.1.XXX net
;
$TTL    604800
@       IN      SOA     ns.example.com. root.example.com. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
```

```
;
@       IN      NS      ns.
10      IN      PTR     ns.example.com.
```

The *Serial Number* in the Reverse zone needs to be incremented on each change as well. For each *A record* you configure in `/etc/bind/db.example.com`, that is for a different address, you need to create a *PTR record* in `/etc/bind/db.192`.

Dopo aver creato il file zona "reverse", riavviare BIND9:

```
sudo service bind9 restart
```

2.4. Server secondario

Una volta configurato un *server primario*, un *server secondario* è necessario per mantenere la disponibilità del dominio nel caso in cui quello primario non fosse più disponibile.

Per prima cosa, nel server primario ("Primary Master"), deve essere consentita la zona "transfer". Aggiungere l'opzione *allow-transfer* alle definizioni delle zone "Forward" e "Reverse" in `/etc/bind/named.conf.local`:

```
zone "example.com" {
        type master;
 file "/etc/bind/db.example.com";
        allow-transfer { 192.168.1.11; };
};

zone "1.168.192.in-addr.arpa" {
        type master;
        file "/etc/bind/db.192";
 allow-transfer { 192.168.1.11; };
};
```

Sostituire *192.168.1.11* con l'indirizzo IP del server di nomi secondario.

Restart BIND9 on the Primary Master:

```
sudo service bind9 restart
```

Quindi, in quello secondario ("Secondary Master"), installare il pacchetto bind9 come fatto per il server primario, quindi modificare il file `/etc/bind/named.conf.local` e aggiungere le seguetni dichiarazioni per le zone "Forward" e "Reverse":

```
zone "example.com" {
 type slave;
        file "db.example.com";
```

```
        masters { 192.168.1.10; };
};

zone "1.168.192.in-addr.arpa" {
 type slave;
        file "db.192";
        masters { 192.168.1.10; };
};
```

Sostituire *192.168.1.10* con l'indirizzo IP del server dei nomi primario.

Riavviare BIND9 nel server secondario:

```
sudo service bind9 restart
```

In `/var/log/syslog` you should see something similar to (some lines have been split to fit the format of this document):

```
client 192.168.1.10#39448: received notify for zone '1.168.192.in-addr.arpa'
zone 1.168.192.in-addr.arpa/IN: Transfer started.
transfer of '100.18.172.in-addr.arpa/IN' from 192.168.1.10#53:
 connected using 192.168.1.11#37531
zone 1.168.192.in-addr.arpa/IN: transferred serial 5
transfer of '100.18.172.in-addr.arpa/IN' from 192.168.1.10#53:
 Transfer completed: 1 messages,
6 records, 212 bytes, 0.002 secs (106000 bytes/sec)
zone 1.168.192.in-addr.arpa/IN: sending notifies (serial 5)

client 192.168.1.10#20329: received notify for zone 'example.com'
zone example.com/IN: Transfer started.
transfer of 'example.com/IN' from 192.168.1.10#53: connected using 192.168.1.11#38577
zone example.com/IN: transferred serial 5
transfer of 'example.com/IN' from 192.168.1.10#53: Transfer completed: 1 messages,
8 records, 225 bytes, 0.002 secs (112500 bytes/sec)
```

Note: A zone is only transferred if the *Serial Number* on the Primary is larger than the one on the Secondary. If you want to have your Primary Master DNS notifying Secondary DNS Servers of zone changes, you can add *also-notify { ipaddress; };* in to `/etc/bind/named.conf.local` as shown in the example below:

```
zone "example.com" {
 type master;
 file "/etc/bind/db.example.com";
 allow-transfer { 192.168.1.11; };
 also-notify { 192.168.1.11; };
 };

zone "1.168.192.in-addr.arpa" {
```

```
type master;
file "/etc/bind/db.192";
allow-transfer { 192.168.1.11; };
also-notify { 192.168.1.11; };
};
```

The default directory for non-authoritative zone files is `/var/cache/bind/`. This directory is also configured in AppArmor to allow the named daemon to write to it. For more information on AppArmor see *Sezione 4, «AppArmor» [168]*.

3. Risoluzione problemi

Questa sezione descrive i metodi per determinare le cause dei problemi che si possono verificare con DNS e BIND9.

3.1. Test

3.1.1. resolv.conf

Il primo passo per verificare BIND9 consiste nell'aggiungere l'indirizzo IP del server di nomi in un risolutore di host. Il server dei nomi primario dovrebbe essere configurato così come un altro host per verificare il tutto. Modificare il file `/etc/resolv.conf` e aggiungere quanto segue:

```
nameserver 192.168.1.10
nameserver 192.168.1.11
```

Potrebbe essere necessario aggiungere anche l'indirizzo IP del server di nomi secondario nel caso in cui il primario non fosse più disponibile.

3.1.2. dig

Se è stato installato il pacchetto dnsutils, è possibile configurare l'utilità di ricerca DNS dig:

- Una volta installato BIND9 usare dig sull'interfaccia di loopback per assicurarsi che sia in ascolto sulla porta 53. Da un terminale digitare:

  ```
  dig -x 127.0.0.1
  ```

  L'output del comando dovrebbe essere simile al seguente:

  ```
  ;; Query time: 1 msec
  ;; SERVER: 192.168.1.10#53(192.168.1.10)
  ```

- Se BIND9 è stato configurato come un server di *cache*, eseguire "dig" su un dominio esterno per verificare il tempo dell'interrogazione:

  ```
  dig ubuntu.com
  ```

  Prestare attenzione al tempo dell'interrogazione verso la fine dell'output:

  ```
  ;; Query time: 49 msec
  ```

  Dopo una seconda esecuzione del comando si dovrebebro vedere dei miglioramenti:

  ```
  ;; Query time: 1 msec
  ```

3.1.3. ping

Per dimostrare come le applicazioni utilizzino i DNS per interpretare un nome host, usare l'utilità ping per inviare una richiesta eco ICMP. Da un terminale digitare:

```
ping example.com
```

In questo modo si verifica che il server dei nome sia in grado di interpretare il nome *ns.example.com* in un indirizzo IP. L'output del comando dovrebbe essere simile a quanto segue:

```
PING ns.example.com (192.168.1.10) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_seq=1 ttl=64 time=0.800 ms
64 bytes from 192.168.1.10: icmp_seq=2 ttl=64 time=0.813 ms
```

3.1.4. named-checkzone

Un ottimo modo per provare i propri file zona consiste nell'usare l'utilità named-checkzone installata con il pacchetto bind9. Questa utilità consente di verificare che la configurazione sia corretta prima di riavviare BIND9 e consentendo di apportare delle modifiche.

- Per provare il file zona "Forward", in un terminale, digitare quanto segue:

  ```
  named-checkzone example.com /etc/bind/db.example.com
  ```

  Se tutto è stato configurato correttamente, si dovrebbe vedere un output simile a questo:

  ```
  zone example.com/IN: loaded serial 6
  OK
  ```

- Analogamente, per verificare il file zona "Reverse", digitare quanto segue:

  ```
  named-checkzone 1.168.192.in-addr.arpa /etc/bind/db.192
  ```

  L'output dovrebbe essere simile a quanto segue:

  ```
  zone 1.168.192.in-addr.arpa/IN: loaded serial 3
  OK
  ```

Il valore *Serial* del proprio file zona probabilmente sarà diverso.

3.2. Registrazione

BIND9 dispone di diverse configurazione per la registrazione degli eventi. Le due opzioni principali sono: *channel* che configura dove vengono salvate le registrazioni e l'opzione *category* che determina quali informazioni registrare.

Se non viene configurata alcuna opzione di registrazione, quella predefinita è:

```
logging {
     category default { default_syslog; default_debug; };
     category unmatched { null; };
};
```

Questa sezione descrive come configurare BIND9 affinché invii i messaggi di *debug* relativi alle interrogazioni DNS in un file diverso.

- Per prima cosa è necessario configurare un canale per specificare quale a quale file inviare i messaggi. Modificare quindi il file `/etc/bind/named.conf.local` e aggiungere quanto segue:

  ```
  logging {
      channel query.log {
          file "/var/log/query.log";
          severity debug 3;
      };
  };
  ```

- Configurare una categoria per inviare tutte le interrogazioni DNS al file:

  ```
  logging {
      channel query.log {
          file "/var/log/query.log";
          severity debug 3;
      };
      category queries { query.log; };
  };
  ```

  

  L'opzione *debug* può essere impostata tra 1 e 3. Se non viene specificato alcun livello, viene considerato quello predefinito, cioè 1.

- Dato che il demone *named* viene eseguito come l'utente *bind*, è necessario creare il file `/var/log/query.log` e modificarne il proprietario:

  ```
  sudo touch /var/log/query.log
  sudo chown bind /var/log/query.log
  ```

- Prima che il demone named possa scrivere nel nuovo file di registrazione, il profilo AppArmor deve esser aggiorato. Per prima cosa modificare `/etc/apparmor.d/usr.sbin.named` e aggiungere:

  ```
  /var/log/query.log w,
  ```

  Quindi ricaricare il profilo:

  ```
  cat /etc/apparmor.d/usr.sbin.named | sudo apparmor_parser -r
  ```

  Per maggiori informazioni riguardo AppArmor, consultare la *Sezione 4, «AppArmor» [168]*.

- Riavviare BIND9 affinché le modifiche abbiano effetto:

```
sudo service bind9 restart
```

Dovrebbe essere possibile vedere il file `/var/log/query.log` riempirsi con le informazioni relative alle interrogazioni. Per maggio informazioni sulle opzioni di registrazione di BIND9, consultare la *Sezione 4.2, «Ulteriori informazioni» [152]*.

4. Riferimenti

4.1. Tipi di record comuni

Questa sezione descrive i più comuni tipi di record DNS.

- Record *A*: mappa un indirizzo IP con un nome host.

```
www      IN    A      192.168.1.12
```

- Record *CNAME*: usato per creare un alias di un record "A" esistente. Non è possibile creare un record CNAME che punti a un altro record CNAME.

```
web      IN    CNAME  www
```

- Record *MX*: usato per definire dove dovrebbero essere inviate le email. Deve puntare a un record A, non a uno CNAME.

```
         IN    MX  1   mail.example.com.
mail     IN    A       192.168.1.13
```

- Record *NS*: usato per definire quali server dispongono di copie di una zona. Deve puntare a un record A, non a un CNAME. Qui vengono definiti i server primario e secondario.

```
         IN    NS     ns.example.com.
         IN    NS     ns2.example.com.
ns       IN    A      192.168.1.10
ns2      IN    A      192.168.1.11
```

4.2. Ulteriori informazioni

- Il *DNS HOWTO*[1] dispone di maggiori informazioni sulla configurazione di BIND9.
- Per un approfondimento di *DNS* e BIND9, consultare *Bind9.net*[2].
- *DNS and BIND*[3] è un libro molto comune giunto ormai alla quinta edizione.
- Un ottimo posto per richiedere assistenza riguardo BIND9, e per partecipare nella comunità di Ubuntu Server, è il canale IRC *#ubuntu-server* su *freenode*[4].
- Consultare anche la *documentazione di bind9 online*[5].

[1] http://www.tldp.org/HOWTO/DNS-HOWTO.html
[2] http://www.bind9.net/
[3] http://www.oreilly.com/catalog/dns5/index.html
[4] http://freenode.net
[5] https://help.ubuntu.com/community/BIND9ServerHowto

Capitolo 9. Sicurezza

La sicurezza deve essere sempre considerata come uno degli aspetti più importanti durante l'installazione, lo sviluppo e l'uso di un sistema. Anche se un'installazione base di Ubuntu offre un livello di sicurezza sufficientemente elevato per l'utilizzo immediato su Internet, è importante avere una buona conoscenza della sicurezza del proprio sistema in base a come verrà usato in produzione.

This chapter provides an overview of security related topics as they pertain to Ubuntu 12.04 LTS Server Edition, and outlines simple measures you may use to protect your server and network from any number of potential security threats.

1. Gestione utenti

L'amministrazione degli utenti è una parte critica per il mantenimento di un sistema sicuro. Utenti poco esperti con privilegi di amministrazione spesso sono la causa della compromissione di sistemi. Pertanto, è importante capire come proteggere il proprio server tramite delle semplici ed efficaci tecniche di gestione degli account utente.

1.1. Dove è l'utente root?

Gli sviluppatori di Ubuntu hanno deciso di disattivare in modo predefinito l'account di amministrazione (root) in tutte le installazioni di Ubuntu. Questo non significa che l'account root sia stato eliminato o che non sia più accessibile, è stata impostata una password che non corrisponde ad alcun possibile valore codificato, pertanto, l'accesso come root non è direttamente possibile.

Gli utenti sono incoraggiati a utilizzare lo strumento sudo per svolgere i compiti di amministrazione di sistema. Lo strumento sudo permette a un utente autorizzato di elevare temporaneamente i propri privilegi usando la propria password, invece di dover conoscere direttamente la password di root. Questo semplice, ma efficace, metodo cerca di fornire responsabilità per tutte le azioni degli utenti e dà all'amministratore un controllo granulare sulle azioni che un utente può eseguire con tali privilegi.

- Se per qualche ragione è necessario abilitare l'account root, basta assegnargli semplicemente una password:

  ```
  sudo passwd
  ```

  Il programma «sudo» chiederà di inserire la propria password e successivamente di inserirne una nuova per l'account root:

  ```
  [sudo] password for username: (inserire la propria password)
  Enter new UNIX password: (inserire una nuova password per root)
  Retype new UNIX password: (reinserire la nuova password per root)
  passwd: password updated successfully
  ```

- Per disabilitare l'account root, utilizzare la seguente sintassi per passwd:

  ```
  sudo passwd -l root
  ```

- Per maggiori informazioni riguardo sudo, consultarne il manuale:

  ```
  man sudo
  ```

In modo predefinito, l'utente iniziale creato dall'installazione di Ubuntu è un membro del gruppo «admin» ed è stato aggiunto al file `/etc/sudoers` come utente autorizzato all'utilizzo di sudo. Per autorizzare altri utenti ai pieni poteri amministrativi di root attraverso l'uso del comando sudo, è sufficiente aggiungerli al gruppo «admin».

1.2. Aggiungere e rimuovere utenti

Il processo per la gestione di utenti e gruppi locali è molto intuitivo e differisce poco dalla maggior parte degli altri sistemi GNU/Linux. Ubuntu e altre distribuzioni basate su Debian, incoraggiano l'utilizzo del pacchetto «adduser» per la gestione degli utenti.

- Per aggiungere un nuovo utente, utilizzare i seguenti comandi e seguire le istruzioni per impostare all'account una password e fornire le caratteristiche identificabili come nome, cognome, numero di telefono, ecc...

  ```
  sudo adduser NOME_UTENTE
  ```

- Per eliminare un utente e il suo gruppo principale, digitare:

  ```
  sudo deluser NOME_UTENTE
  ```

  Quando si elimina un account utente non viene rimossa la sua cartella home. È decisione dell'amministratore se rimuoverla o no in base alle proprie scelte.

  Ricordare che, se non sono state prese le necessarie precauzioni, ogni nuovo utente aggiunto successivamente con gli stessi UID/GID del precedente proprietario della cartella, avrà accesso a tale cartella.

  È possibile modificare questi valori UID/GID con qualcosa di più appropriato, come per esempio l'account root, e spostare la cartella per evitare futuri conflitti:

  ```
  sudo chown -R root:root /home/NOME_UTENTE/
  sudo mkdir /home/archived_users/
  sudo mv /home/NOME_UTENTE /home/archived_users/
  ```

- Per bloccare o sbloccare temporaneamente l'account di un utente, utilizzare, rispettivamente, i seguenti comandi:

  ```
  sudo passwd -l NOME_UTENTE
  sudo passwd -u NOME_UTENTE
  ```

- Per aggiungere o rimuovere un gruppo personalizzato, utilizzare, rispettivamente, i seguenti comandi:

  ```
  sudo addgroup NOME_GRUPPO
  sudo delgroup NOME_GRUPPO
  ```

- Per aggiungere un utente a un gruppo, digitare:

  ```
  sudo adduser NOME_UTENTE NOME_GRUPPO
  ```

1.3. Sicurezza dei profili utente

Quando viene creato un nuovo utente, l'applicazione «adduser» crea una nuova directory chiamata `/home/NOME_UTENTE`. Il profilo predefinito è modellato secondo i contenuti presenti nella directory `/etc/skel` che contiene tutti i profili di base.

Se il proprio server ospiterà più utenti, è necessario prestare la massima attenzione alle autorizzazioni delle home degli utenti, al fine di garantirne la riservatezza. In modo predefinito, in Ubuntu, le home degli utenti sono create con permessi di lettura e di esecuzione per tutti gli utenti. Questo significa che tutti gli utenti possono visualizzare e accedere al contenuto delle home degli altri utenti, cosa che potrebbe non essere soddisfacente per il proprio ambiente.

- Per verificare i permessi attuali della home degli utenti, utilizzare il seguente comando:

  ```
  ls -ld /home/NOME_UTENTE
  ```

  Il seguente output mostra che la directory `/home/NOME_UTENTE` è accessibile in lettura da parte di tutti gli utenti:

  ```
  drwxr-xr-x 2 nomeutente nomeutente 4096 2007-10-02 20:03 nomeutente
  ```

- È possibile rimuovere il permesso in lettura da tutti con il seguente comando:

  ```
  sudo chmod 0750 /home/NOME_UTENTE
  ```

  

  Alcuni amministratori utilizzano anche l'opzione per la modifica ricorsiva (-R) di tutte le sotto-cartelle e file della home, ma questo non è necessario e potrebbe inoltre causare degli effetti indesiderati. Modificare i permessi alla cartella principale è più che sufficiente per prevenire degli accessi non autorizzati.

  Un modo più efficiente potrebbe essere quello di modificare direttamente le impostazioni predefinite dell'applicazione adduser sui permessi da assegnare alle home degli utenti appena creati. È sufficiente modificare la variabile `DIR_MODE`, nel file `/etc/adduser.conf`, secondo le proprie esigenze.

  ```
  DIR_MODE=0750
  ```

- Dopo aver corretto opportunamente i permessi di accesso alle directory home come descritto precedentemente, verificare il risultato con il seguente comando:

  ```
  ls -ld /home/NOME_UTENTE
  ```

  Il risultato qui sotto mostra come i permessi di lettura per tutti gli altri utenti siano stati rimossi:

  ```
  drwxr-x--- 2 nomeutente nomeutente 4096 2007-10-02 20:03 nomeutente
  ```

1.4. Politica delle password

Una severa politica delle password è uno dei più importanti aspetti della sicurezza di un sistema. Le più frequenti violazioni di un sistema avvengono tramite attacchi di forza bruta con degli elenchi di parole che statisticamente possono comprende delle parole chiavi utilizzate come password. Se si vuole di offrire un qualsiasi tipo di accesso remoto utilizzando la propria password locale, assicurarsi che la complessità della stessa superi dei limiti minimi di adeguatezza, di impostare delle password con durate massime e controllare frequentemente i propri sistemi di autenticazione.

1.4.1. Lungezza minima di una password

By default, Ubuntu requires a minimum password length of 6 characters, as well as some basic entropy checks. These values are controlled in the file `/etc/pam.d/common-password`, which is outlined below.

```
password        [success=2 default=ignore]      pam_unix.so obscure sha512
```

If you would like to adjust the minimum length to 8 characters, change the appropriate variable to min=8. The modification is outlined below.

```
password        [success=2 default=ignore]      pam_unix.so obscure sha512 min=8
```

Basic password entropy checks and minimum length rules do not apply to the administrator using sudo level commands to setup a new user.

1.4.2. Scadenza delle password

Quando vengono creati dei nuovi utenti è possibile impostare una durata minima e massima per le loro password, obbligando gli stessi a modificarla alla scadenza.

- Per visualizzare facilmente lo stato attuale di un account utente, utilizzare il seguente comando:

  `sudo chage -l NOME_UTENTE`

  L'output seguente mostra informazioni interessanti sull'account dell'utente, in particolare che non ci sono politiche applicate:

  ```
  Ultimo cambio della password : gen 20, 2008
  Scadenza della password : mai
  Inattività della password : mai
  Scadenza dell'account : mai
  Numero minimo di giorni tra i cambi di password : 0
  Numero massimo di giorni tra i cambi di password : 99999
  Giorni di preavviso prima della scadenza della password : 7
  ```

- Per impostare uno qualsiasi di questi campi, utilizzare il seguente comando e seguire le istruzioni:

sudo chage NOME_UTENTE

Quello che segue è un esempio di come sia possibile modificare manualmente la data di scadenza dell'account (-E) al 31/01/2008 (inserirla nel formato mm/gg/aaaa o nel formato aaaa/mm/gg), l'età minima della password (-m) a 5 giorni, l'età massima (-M) a 90 giorni, il periodo di inattività (-I) a 5 giorni dopo la scadenza della password e un avvertimento (-W) di 14 giorni prima della scadenza delle password.

sudo chage -E 01/31/2011 -m 5 -M 90 -I 30 -W 14 username

- Per verificare le modifiche, utilizzare lo stesso comando di prima:

sudo chage -l NOME_UTENTE

Il seguente output mostra i cambiamenti effettuati sull'account:

```
Ultimo cambio della password : gen 20, 2008
Scadenza della password : apr 19, 2008
Inattività della password : mag 19, 2008
Scadenza dell'account : gen 31, 2008
Numero minimo di giorni tra i cambi di password : 5
Numero massimo di giorni tra i cambi di password : 90
Giorni di preavviso prima della scadenza della password : 14
```

1.5. Ulteriori considerazioni sulla sicurezza

Molte applicazioni usano meccanismi di autenticazione alternativi che possono essere facilmente trascurati anche da esperti amministratori di sistema. Pertanto, è importante comprendere e controllare come avviene l'autenticazione degli utenti e come accedono ai servizi e alle applicazioni sul proprio server.

1.5.1. Accesso SSH per gli utenti disabilitati

Disattivando o bloccando l'account di un utente non impedisce che quest'ultimo riesca a effettuare l'accesso al server se precedentemente utilizzava una chiave pubblica RSA; saranno ancora in grado di ottenere l'accesso al server senza la necessità della password. Controllare sempre se nella directory home degli utenti sono presenti dei file che permettano questo tipo di autenticazione SSH, come per esempio `/home/nomeutente/.ssh/authorized_keys`.

Eliminare o rinominare la directory `.ssh/` nella home degli utenti per prevenire future autenticazioni SSH.

Assicurarsi di controllare qualsiasi connessione SSH stabilita dagli utenti disabilitati, dato che potrebbero esserci connessioni aperti in entrata o in uscita. Terminare tutte quelle che vengono trovate.

Limitare l'accesso SSH solo agli utenti che ne hanno il diritto. Per esempio, è possibile creare un gruppo chiamato «sshlogin» e aggiungere il nome del gruppo alla voce `AllowGroupsvarname> nel file /etc/ssh/sshd_config.`

```
AllowGroups sshlogin
```

Dopo aver aggiunto gli utenti con diritto di accesso SSH al gruppo «sshlogin», riavviare il server SSH.

```
sudo adduser NOME_UTENTE sshlogin
sudo service ssh restart
```

1.5.2. Autenticazione utenti su database esterno

La maggior parte delle reti aziendali richiedono un servizio di autenticazione e di controllo degli accessi centralizzato per tutte le risorse di sistema. Se il server è stato configurato per gestire l'autenticazione attraverso database esterni, assicurarsi di disabilitare gli account utente sia esternamente che internamente, in questo modo l'autenticazione locale di riserva non è più possibile.

2. Sicurezza della console

As with any other security barrier you put in place to protect your server, it is pretty tough to defend against untold damage caused by someone with physical access to your environment, for example, theft of hard drives, power or service disruption, and so on. Therefore, console security should be addressed merely as one component of your overall physical security strategy. A locked "screen door" may deter a casual criminal, or at the very least slow down a determined one, so it is still advisable to perform basic precautions with regard to console security.

Le seguenti istruzioni consentiranno di proteggere il proprio server da problemi che potrebbero portare serie conseguenze.

2.1. Disabilitare il Ctrl+Alt+Canc

Qualsiasi persona con accesso fisico alla tastiera può semplicemente premere **Ctrl**+**Alt**+**Canc** per riavviare il server senza eseguire l'accesso. Qualcuno può sempre scollegare la presa della corrente, ma per lo meno è da evitare l'uso di questa combinazione di tasti su un server in produzione. In questo modo un malintenzionato è costretto a utilizzare altre strategie per riavviare un server e consente di non riavviarlo accidentalmente.

- To disable the reboot action taken by pressing the **Ctrl**+**Alt**+**Delete** key combination, comment out the following line in the file `/etc/init/control-alt-delete.conf`.

```
#exec shutdown -r now "Control-Alt-Delete pressed"
```

3. Firewall

3.1. Introduzione

Il kernel Linux include il sottosistema *Netfilter* usato per manipolare o decidere la sorte del traffico di rete diretto all'interno o attraverso un server. Tutte le moderne soluzioni firewall per Linux si basano su questo sistema di filtraggio dei pacchetti.

Il sistema di filtraggio dei pacchetti del kernel non è di grande utilità per gli amministratori senza un'interfaccia nello spazio utente per gestirlo. Questo è il compito di iptables. Quando un pacchetto raggiunge il proprio server, esso è gestito affidato al sottosistema Netfilter per l'accettazione, la manipolazione oppure il rifiuto secondo quanto stabilito da regole fornite al sottosistema dallo spazio utente attraverso iptables. Quindi, iptables è tutto ciò che è necessario per gestire il proprio firewall, a patto che si abbia la dimestichezza necessaria; sono comunque disponibili molte altre applicazioni per semplificare tale attività.

3.2. ufw - Firewall non complicato

L'applicazione predefinita in Ubuntu per la configurazione di un firewall è ufw. Sviluppato per semplificare la configurazione di iptables, ufw offre un modo semplice per creare un firewall basato su protocolli IPv4 e IPv6.

ufw, in modo predefinito, è inizialmente disabilitato. Dal manuale di ufw si legge:

«ufw is not intended to provide complete firewall functionality via its command interface, but instead provides an easy way to add or remove simple rules. It is currently mainly used for host-based firewalls (ufw non ha lo scopo di implementare tutte le funzionalità di un firewall tramite la sua interfaccia di comandi, ma invece cerca di facilitare l'aggiunta o la rimozione di semplici regole. È usato principalmente per dei firewall host-based)»

Seguono degli esempi sull'uso di ufw:

- Per prima cosa, è necessario abilitare ufw. In un terminale digitare:

```
sudo ufw enable
```

- Per aprire una porta (in questo caso la porta di SSH):

```
sudo ufw allow 22
```

- Le regole possono anche essere aggiunte usando un formato *a numeri*:

```
sudo ufw insert 1 allow 80
```

- Analogamente, per chiudere una porta aperta:

```
sudo ufw deny 22
```

- Per eliminare una regola, usare «delete» seguito dalla regola:

```
sudo ufw delete deny 22
```

- È anche possibile consentire l'accesso da host o da reti specifici a una porta. Il seguente esempio consente accesso SSH dall'host 192.168.0.2 a qualsiasi indirizzo IP su questo host:

```
sudo ufw allow proto tcp from 192.168.0.2 to any port 22
```

  Sostituire 192.168.0.2 con 192.168.0.0/24 per consentire accesso SSH da tutta la sotto-rete.

- Aggiungendo l'opzione *--dry-run* a un comando *ufw* è possibile visualizzare il risultato delle regole, ma senza applicarle. Per esempio, questo è quello che verrebbe applicato nel caso venisse aperta la porta HTTP:

```
sudo ufw --dry-run allow http
```

```
*filter
:ufw-user-input - [0:0]
:ufw-user-output - [0:0]
:ufw-user-forward - [0:0]
:ufw-user-limit - [0:0]
:ufw-user-limit-accept - [0:0]
### RULES ###

### tuple ### allow tcp 80 0.0.0.0/0 any 0.0.0.0/0
-A ufw-user-input -p tcp --dport 80 -j ACCEPT

### END RULES ###
-A ufw-user-input -j RETURN
-A ufw-user-output -j RETURN
-A ufw-user-forward -j RETURN
-A ufw-user-limit -m limit --limit 3/minute -j LOG --log-prefix "[UFW LIMIT]: "
-A ufw-user-limit -j REJECT
-A ufw-user-limit-accept -j ACCEPT
COMMIT
Rules updated
```

- È possibile disabilitare ufw con il comando:

```
sudo ufw disable
```

- Per visualizzare lo stato del firewall usare:

```
sudo ufw status
```

- Per informazioni più dettagliate usare:

```
sudo ufw status verbose
```

- Per visualizzare il formato *a numeri*:

```
sudo ufw status numbered
```

Se la porta che si vuole aprire o chiudere è definita in `/etc/services`, è possibile usare il nome della porta al posto del numero. In questo esempio si sostituisce *22* con *ssh*.

Questa è una breve introduzione all'utilizzo di ufw. Per maggiori informazioni, consultare le pagine man di ufw.

3.2.1. Integrazione delle applicazioni con ufw

Le applicazioni che aprono delle porte possono includere un profilo ufw in cui vengono descritte le porte necessarie all'applicazione per funzionare correttamente. I profili vengono salvati in `/etc/ufw/applications.d` e possono essere modificati se le porte predefinite sono cambiate.

- Per visualizzare quali applicazioni hanno un profilo installato, in un terminale digitare:

```
sudo ufw app list
```

- Usare un profilo di un'applicazione è simile al consentire il traffico attraverso una porta:

```
sudo ufw allow Samba
```

- È disponibile anche una sintassi più estesa:

```
ufw allow from 192.168.0.0/24 to any app Samba
```

Sostituire *Samba* e *192.168.0.0/24* con il profilo dell'applicazione da usare e l'intervallo di indirizzi della propria rete.

Non è necessario specificare il *protocollo* per l'applicazione, dato che queste informazioni sono contenute nel profilo. Notare che il nome dell'*applicazione* sostituisce il numero della *porta*.

- Per visualizzare i dettagli riguardo quali porte, protocolli, ecc... sono definiti per un'applicazione, digitare:

```
sudo ufw app info Samba
```

Not all applications that require opening a network port come with ufw profiles, but if you have profiled an application and want the file to be included with the package, please file a bug against the package in Launchpad.

```
ubuntu-bug nameofpackage
```

3.3. IP masquerading

Il compito dell'IP masquerading è di consentire a quei computer della rete forniti di indirizzi IP privati e non instradabili, di accedere a Internet tramite il computer che opera il masquerading. Il traffico che va dalla rete privata verso Internet deve essere manipolato per ottenere risposte che siano re-instradabili al computer che ne ha fatto richiesta. Per ottenere questo risultato, il kernel deve modificare l'indirizzo IP *sorgente* di ciascun pacchetto affinché tali risposte vengano re-instradate a esso invece che all'indirizzo IP privato che ha fatto la richiesta, procedura impossibile da eseguire su Internet. Linux fa uso del *tracciamento della connessione* (conntrack) per tenere traccia di quale connessione appartenga a quale computer e di conseguenza per re-instradare ciascun pacchetto di risposta. Il traffico in uscita dalla rete privata viene quindi "mascherato" per simulare l'uscita dalla macchina gateway Ubuntu. Nella documentazione Microsoft questo processo è indicato come condivisione delle connessioni internet (Internet Connection Sharing).

3.3.1. Masquerading con ufw

L'IP masquerading può essere ottenuto utilizzando regole ufw personalizzate. Questo è possibile dato che il backend attuale per ufw è iptables-restore con i file delle regole posizionati in `/etc/ufw/*.rules`. Questi file possono essere usati per aggiungere vecchie regole di iptables usate senza ufw e regole maggiormente legate al gateway o al bridge.

Le regole sono divise in due file diversi, regole da eseguire prima delle regole a riga di comando di ufw e regole da eseguire dopo ufw.

- Per prima cosa, è necessario abilitare l'inoltro dei pacchetti modificando due file di configurazione. In `/etc/default/ufw` modificare *DEFAULT_FORWARD_POLICY* in «ACCEPT»:

  ```
  DEFAULT_FORWARD_POLICY="ACCEPT"
  ```

  Quindi modificare il file `/etc/ufw/sysctl.conf` de-commentando:

  ```
  net/ipv4/ip_forward=1
  ```

  Similmente, per abilitare l'inoltro con IPv6 de-commentare:

  ```
  net/ipv6/conf/default/forwarding=1
  ```

- Ora verranno aggiunte delle regole al file `/etc/ufw/before.rules`. Le regole predefinite configurano solamente la tabella *filter* e per abilitare il masquerading è necessario configurare la tabella *nat*. Aggiungere all'inizio del file, subito dopo i commenti dell'intestazione, quanto segue:

  ```
  # regole tabella nat
  *nat
  :POSTROUTING ACCEPT [0:0]
  ```

```
# Inoltro traffico da eth1 attraverso eth0
-A POSTROUTING -s 192.168.0.0/24 -o eth0 -j MASQUERADE

# non cancellare la riga 'COMMIT' o queste tabelle di regole non saranno elaborate
COMMIT
```

I commenti non sono necessari, ma è buona pratica documentare le proprie configurazioni. Inoltre, quando si modificano i file *rules* in `/etc/ufw`, assicurarsi che queste righe siano sempre le ultime in ogni tabella modificata:

```
# non eliminare la riga 'COMMIT' o queste tabelle di regole non saranno elaborate
COMMIT
```

Per ogni *tabella* è necessario un *COMMIT*. In questi esempi sono mostrate solamente le tabelle *nat* e *filter*, ma è possibile aggiungere regole per le tabelle *raw* e *mangle*.

Nell'esempio precedente, sostituire *eth0*, *eth1* e *192.168.0.0/24* con le interfacce appropriate e con l'intervallo di indirizzi corretto.

- Infine, disattivare e riattivare ufw per applicare le modifiche:

```
sudo ufw disable && sudo ufw enable
```

L'IP masquerading ora dovrebbe essere abilitato. È possibile aggiungere regole FORWARD aggiuntive al file `/etc/ufw/before.rules`. È utile che queste regole addizionali vengano aggiunte alla catena *ufw-before-forward*.

3.3.2. Masquerading con iptables

iptables can also be used to enable Masquerading.

- Similmente a ufw, il primo passo per abilitare l'inoltro di pacchetti con IPv4 è quello di modificare il file `/etc/sysctl.conf` e de-commentare la seguente riga:

```
net.ipv4.ip_forward=1
```

Per abilitare l'inoltro con IPv6, de-commetare:

```
net.ipv6.conf.default.forwarding=1
```

- Quindi, eseguire il comando sysctl per abilitare le nuove impostazioni nel file di configurazione:

```
sudo sysctl -p
```

- L'IP masquerading può essere ottenuto con una sola regola di iptables, che può cambiare leggermente in base alla configurazione della propria rete:

```
sudo iptables -t nat -A POSTROUTING -s 192.168.0.0/16 -o ppp0 -j MASQUERADE
```

Il comando precedente assume che lo spazio di indirizzi privato sia 192.168.0.0/16 e che il dispositivo collegato a Internet sia ppp0. La sintassi del comando è la seguente:

- -t nat: regola viene inserita nella tabella nat
- -A POSTROUTING: la regola viene accodata (-A) alla catena POSTROUTING
- -s 192.168.0.0/16: la regola si applica al traffico originato dallo spazio di indirizzi specificato
- -o ppp0: la regola si applica al traffico instradato attraverso l'interfaccia di rete specificata
- -j MASQUERADE: il traffico che soddisfa questa regola viene "saltato" (-j sta per jump) alla destinazione MASQUERADE per essere manipolato come descritto in precedenza

- Inoltre, ogni catena nella tabella "filter" (la tabella predefinita e dove avvengono la maggior parte dei filtri sui pacchetti) ha una *politica* predefinita di ACCEPT, ma se si sta creando un firewall in aggiunta a un dispositivo gateway, è possibile aver impostato le politiche DROP e REJECT, nel cui caso il traffico "masqueraded" deve essere consentito attraverso la catena FORWARD affinché la regola precedente possa funzionare:

```
sudo iptables -A FORWARD -s 192.168.0.0/16 -o ppp0 -j ACCEPT
sudo iptables -A FORWARD -d 192.168.0.0/16 -m state \
--state ESTABLISHED,RELATED -i ppp0 -j ACCEPT
```

I precedenti comandi consentiranno a tutte le connessioni della propria rete locale accesso a Internet e a tutto il traffico relativo a queste connessioni di ritornare ai computer che lo hanno originato.

- Per fare in modo che il masquerading sia abilitato al riavvio, modificare il file `/etc/rc.local` e aggiungere qualsiasi dei comandi utilizzati precedentemente. Per esempio, aggiungere il primo comando senza filtro:

```
iptables -t nat -A POSTROUTING -s 192.168.0.0/16 -o ppp0 -j MASQUERADE
```

3.4. Registri

I registri del firewall sono molto utili per riconoscere gli attacchi, migliorare le regole del firewall e per verificare attività inusuali nella propria rete. È necessario includere regole di registrazione per fare in modo che vengano eseguite le registrazioni e queste devono essere inserite prima di qualsiasi regola terminante applicabile (un regola con un obiettivo che decide il destino di un pacchetto, come ACCEPT, DROP o REJECT).

Se si sta usando ufw è possibile attivare la registrazione con il seguente comando:

```
sudo ufw logging on
```

Per disabilitare la registrazione in ufw, sostituire, nel comando precedente, *on* con *off*.

Se è in uso iptables al posto di ufw, digitare:

```
sudo iptables -A INPUT -m state --state NEW -p tcp --dport 80 \
-j LOG --log-prefix "NEW_HTTP_CONN: "
```

A request on port 80 from the local machine, then, would generate a log in dmesg that looks like this (single line split into 3 to fit this document):

```
[4304885.870000] NEW_HTTP_CONN: IN=lo OUT= MAC=00:00:00:00:00:00:00:00:00:00:00:00:08:00
 SRC=127.0.0.1 DST=127.0.0.1 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=58288 DF PROTO=TCP
 SPT=53981 DPT=80 WINDOW=32767 RES=0x00 SYN URGP=0
```

The above log will also appear in `/var/log/messages`, `/var/log/syslog`, and `/var/log/kern.log`. This behavior can be modified by editing `/etc/syslog.conf` appropriately or by installing and configuring ulogd and using the ULOG target instead of LOG. The ulogd daemon is a userspace server that listens for logging instructions from the kernel specifically for firewalls, and can log to any file you like, or even to a PostgreSQL or MySQL database. Making sense of your firewall logs can be simplified by using a log analyzing tool such as logwatch, fwanalog, fwlogwatch, or lire.

3.5. Altri strumenti

Esistono diversi strumenti per "costruire" un firewall completo senza alcuna conoscenza di iptables. Per chi preferisce un'interfaccia grafica:

- *fwbuilder*[1] è molto potente e ha un aspetto che può risultare familiare agli amministratori che hanno utilizzato un firewall commerciale come Checkpoint FireWall-1.

Per chi preferisce uno strumento a riga di comando con file di configurazione in semplice testo:

- *Shorewall*[2] è una soluzione molto potente per configurare un firewall di livello avanzato per qualsiasi rete.

3.6. Riferimenti

- The *Ubuntu Firewall*[3] wiki page contains information on the development of ufw.
- Inoltre, la pagina di manuale di ufw contiene molte informazioni utili: **man ufw**.
- Per maggiori informazioni sull'uso di iptables, consultare *packet-filtering-HOWTO*[4].
- Il *nat-HOWTO*[5] contiene ulteriori dettagli sul masquerading.
- The *IPTables HowTo*[6] in the Ubuntu wiki is a great resource.

[1] http://www.fwbuilder.org/
[2] http://www.shorewall.net/
[3] https://wiki.ubuntu.com/UncomplicatedFirewall
[4] http://www.netfilter.org/documentation/HOWTO/packet-filtering-HOWTO.html
[5] http://www.netfilter.org/documentation/HOWTO/NAT-HOWTO.html
[6] https://help.ubuntu.com/community/IptablesHowTo

4. AppArmor

AppArmor è un'implementazione del «Linux Security Module» per il controllo degli accessi vincolante basato sul nome. AppArmor racchiude individualmente i programmi in un insieme di file e capacità posix 1003.1e draft.

AppArmor è installato e caricato in modo predefinito e utilizza i *profili* di un'applicazione per determinare quali file e permessi siano necessari all'applicazione. Alcuni pacchetti installano i propri profili e ulteriori profili possono essere trovati nel pacchetto apparmor-profiles.

Per installare il pacchetto apparmor-profiles, in un terminale digitare:

```
sudo apt-get install apparmor-profiles
```

I profili di AppArmor dispongono di due modalità di esecuzione:

- Apprendimento (complaining/learning): le violazioni del profilo sono consentite e vengono registrate. Utile per verificare e sviluppare nuovi profili.
- Esecutiva (enforced/confined): obbliga a rispettare la politica del profilo e registra le violazioni.

4.1. Utilizzare AppArmor

Il pacchetto apparmor-utils contiene utilità a riga di comando che è possibile usare per modificare la modalità di esecuzione di AppArmor, trovare lo stato di un profilo, creare nuovi profili, ecc...

- apparmor_status è utilizzata per visualizzare lo stato attuale dei profili AppArmor.

  ```
  sudo apparmor_status
  ```

- aa-complain posiziona un profilo nella modalità *apprendimento*.

  ```
  sudo aa-complain /percorso/al/binario
  ```

- aa-enforce posiziona un profilo nella modalità *esecutiva*.

  ```
  sudo aa-enforce /percorso/al/bianrio
  ```

- Nella directory `/etc/apparmor.d` sono archiviati tutti i profili di AppArmor ed è possibile, da qui, modificare la *modalità* di tutti i profili.

  Usare il seguente comando per impostare tutti i profili nella modalità apprendimento:

  ```
  sudo aa-complain /etc/apparmor.d/*
  ```

  Per impostare tutti i profili nella modalità esecutiva:

  ```
  sudo aa-enforce /etc/apparmor.d/*
  ```

- apparmor_parser è utilizzata per caricare un profilo all'interno del kernel. Può essere usata anche per ricaricare profili attraverso l'opzione *-r*. Per caricare un profilo:

```
cat /etc/apparmor.d/profile.name | sudo apparmor_parser -a
```

  Per ricaricare un profilo:

```
cat /etc/apparmor.d/profile.name | sudo apparmor_parser -r
```

- `/etc/init.d/apparmor` può essere usato per *ricaricare* tutti i profili:

```
sudo /etc/init.d/apparmor reload
```

- La directory `/etc/apparmor.d/disable` può essere usata con l'opzione apparmor_parser -R per *disabilitare* un profilo.

```
sudo ln -s /etc/apparmor.d/profile.name /etc/apparmor.d/disable/
sudo apparmor_parser -R /etc/apparmor.d/profile.name
```

  Per *riabilitare* un profilo disabilitato, rimuovere il collegamento simbolico al profilo in `/etc/apparmor.d/disable/`, quindi caricare il profilo usando l'opzione *-a*.

```
sudo rm /etc/apparmor.d/disable/profile.name
cat /etc/apparmor.d/profile.name | sudo apparmor_parser -a
```

- È possibile disabilitare AppArmor e scaricare il modulo del kernel attraverso i seguenti comandi:

```
sudo /etc/init.d/apparmor stop
sudo update-rc.d -f apparmor remove
```

- Per riabilitare AppArmor:

```
sudo /etc/init.d/apparmor start
sudo update-rc.d apparmor defaults
```

Sostituire *profile.name* con il nome del profilo da modificare e sostituire anche `/percorso/eseguibile/` con il percorso all'eseguibile. Per esempio, per il comando ping, usare `/bin/ping`

4.2. Profili

I profili di AppArmor sono dei semplici file di testo posizionati in `/etc/apparmor.d/`. Questi file vengono nominati con il percorso completo all'eseguibile del profilo, sostituendo «/» con «.». Per esempio, `/etc/apparmor.d/bin.ping` è il profilo AppArmor del comando `/bin/ping`.

Esistono due principali tipologie di regole usate nei profili:

- *Voci di percorso*: specificano a quali file nel file system un'applicazione può accedere.

- *Voci di capacità*: determinano quali privilegi un processo può utilizzare.

Per un esempio, consultare `/etc/apparmor.d/bin.ping`:

```
#include <tunables/global>
/bin/ping flags=(complain) {
  #include <abstractions/base>
  #include <abstractions/consoles>
  #include <abstractions/nameservice>

  capability net_raw,
  capability setuid,
  network inet raw,

  /bin/ping mixr,
  /etc/modules.conf r,
}
```

- *#include <tunables/global>*: asserzioni di inclusione da altri file. Consente di usare un file comune con le asserzioni di inclusione per molteplici applicazioni.
- */bin/ping flags=(complain)*: percorso al programma con profilo, impostandone la modalità ad *apprendimento*.
- *capability net_raw,*: consente all'applicazione di accedere alla capacità CAP_NET_RAW Posix.1e.
- */bin/ping mixr,*: consente all'applicazione accesso in lettura e in esecuzione al file.

Dopo aver modificato un profilo, è necessario ricaricarlo. Per maggiori informazioni, consultare *Sezione 4.1, «Utilizzare AppArmor» [168]*.

4.2.1. Creare un profilo

- *Progettare un piano di verifica*: cercare di pensare a come l'applicazione dovrebbe essere eseguita. Il piano di verifica dovrebbe essere diviso in tanti piccoli casi d'uso, ognuno dei quali dovrebbe avere una breve descrizione e un elenco dei passi da compiere.

  Alcuni casi standard da verificare sono:

  - Avvio del programma.
  - Arresto del programma.
  - Ricaricamento del programma.
  - Verifica di tutti i comandi supportati dallo script init.
- *Generare il nuovo profilo*: usare aa-genprof per generare un nuovo profilo. Da un terminale:

  ```
  sudo aa-genprof eseguibile
  ```

  Per esempio:

```
sudo aa-genprof slapd
```

- Affinché il proprio nuovo profilo venga incluso nel pacchetto apparmor-profiles, segnalare un bug su *Launchpad* riguardo il pacchetto *AppArmor*[7]:
  - Includere la pianificazione e le casistiche del test.
  - Allegare il nuovo profilo al bug.

4.2.2. Aggiornare i profili

Quando il programma si comporta stranamente, messaggi di audit vengono inviati ai file di registro. Il programma aa-logprof può essere usato per analizzare i file di registro per i messaggi di audit di AppArmor, per controllarli e per aggiornare i profili. Da un terminale:

```
sudo aa-logprof
```

4.3. Riferimenti

- Per le opzioni avanzate di configurazione, consultare la *AppArmor Administration Guide*[8]
- Per maggiori informazioni su come usare AppArmor con altri rilasci di Ubuntu, consultare la *documentazione della comunità italiana*[9].
- The *OpenSUSE AppArmor*[10] page is another introduction to AppArmor.
- Un buon posto per chiedere assistenza riguardo AppArmor, e per partecipare nella comunità di Ubuntu Server, è il canale IRC *#ubuntu-server* su *freenode*[11].

[7] https://bugs.launchpad.net/ubuntu/+source/apparmor/+filebug

[8] http://www.novell.com/documentation/apparmor/apparmor201_sp10_admin/index.html?page=/documentation/apparmor/apparmor201_sp10_admin/data/book_apparmor_admin.html

[9] http://wiki.ubuntu-it.org/Sicurezza/AppArmor

[10] http://en.opensuse.org/SDB:AppArmor_geeks

[11] http://freenode.net

5. Certificati

Una delle più comuni forme di crittografia odierna è la crittografia a *chiave pubblica*. Questo tipo di crittografia utilizza una *chiave pubblica* e una *chiave privata*. Il sistema funziona *cifrando* le informazioni usando la chiave pubblica che possono solo essere *decifrate* con la chiave privata.

L'utilizzo più comune della crittografia a chiave pubblica è nella cifratura del traffico delle applicazioni attraverso una connessione SSL (Secure Socket Layer) o TLS (Transport Layer Security), per esempio configurando Apache affinché fornisca *HTTPS*, il protocollo HTTP via SSL. Questo consente di cifrare il traffico utilizzando un protocollo che non fornisce nativamente una cifratura.

Un *certificato* è un metodo di distribuzione di una *chiave pubblica* e di altre informazioni riguardo un server e l'organizzazione che ne è responsabile. I certificati possono essere firmati digitalmente a un'*Autorità di Certificazione* o CA. Una CA è un'entità fidata che conferma la veridicità delle informazioni contenute nel certificato.

5.1. Tipologie dei certificati

Per configurare un server sicuro affinché usi la crittografia a chiave pubblica, nella maggior parte dei casi, è necessario inviare la richiesta del certificato (compresa la chiave pubblica), una prova di esistenza della propria società e il pagamento a una CA. La CA verifica la richiesta e la propria identità e quindi invia un certificato per il proprio server. In alternativa, è possibile creare il proprio certificato *auto-firmato*.

I certificati auto-firmati non dovrebbero essere usati in ambienti di produzione.

Continuando l'esempio di HTTPS, un certificato CA firmato dispone di caratteristiche che un certificato auto-firmato non ha:

- I browser, solitamente, riconoscono automaticamente il certificato e consentono l'attivazione di una connessione sicura senza chiedere nulla all'utente.
- Quando una CA emette un certificato, garantisce l'identità dell'organizzazione che fornisce la pagina web al browser.

La maggior parte dei browser web, e dei computer che supportano SSL, dispongono di un elenco di CA i cui certificati sono accettati automaticamente. Se un browser incontra un certificato la cui CA non è presente nell'elenco, il browser chiede all'utente di accettare o rifiutare la connessione. Inoltre, altre applicazioni possono generare un messaggio di errore quando viene usato un certificato auto-firmato.

Il processo per ottenere un certificato da una CA è molto semplice. Un piccolo promemoria:

1. Creare un coppia di chiavi pubblica e privata.
2. Creare una richiesta per un certificato basato su chiave pubblica. La richiesta del certificato contiene informazioni riguardo il server a la società che lo ospita.

3. Inviare la richiesta, con una fotocopia di un documento di identità, a una CA. Non è possibile consigliare quale autorità di certificazione scegliere. La decisione potrebbe essere basata su esperienze passate, esperienze di amici o colleghi o per un fattore economico.

   Una volta scelta la CA, è necessario seguire le istruzioni fornite dal CA per ottenere il certificato.
4. Una volta che la CA ha verificato l'identità del richiedente, invierà un certificato digitale.
5. Installare questo certificato sul proprio server sicuro e configurare le applicazioni appropriate affinché usino il certificato.

5.2. Generare una CSR (Certificate Signing Request)

Sia che si stia ottenendo un certificato da una CA sia che si auto-firmi il proprio, il primo passo consiste nel generare una chiave di cifratura.

Se il certificato verrà usato da servizi come Apache, Postfix, Dovect, ecc..., è solitamente indicato usare una chiave priva di passphrase.

Questa sezione indica come generare una chiave dotata di passphrase e una priva di passphrase. La chiave priva di passphrase verrà impiegata per generare un certificato che può essere usato da diversi servizi.

Avere in esecuzione i servizi senza una passphrase è conveniente poiché non vi è la necessità di digitare la passphrase a ogni avvio del servizio, ma non è molto sicuro in quanto se la chiave viene compromessa, verrà compromesso anche il server.

Per generare le *chiavi* per la CSR (Certificate Signing Request), eseguire in un terminale il seguente comando:

```
openssl genrsa -des3 -out server.key 2048
```

```
Generating RSA private key, 2048 bit long modulus
.........................++++++
.......++++++
e is 65537 (0x10001)
Enter pass phrase for server.key:
```

È ora necessario inserire una passphrase. Per una maggiore sicurezza, dovrebbe contenere almeno 8 caratteri. La lunghezza minima con l'opzione «-des3» è di 4 caratteri. Dovrebbe includere numeri o segni di punteggiatura e non dovrebbe essere una parola reperibile in un vocabolario. Ricordarsi che una passphrase differenzia tra minuscole e maiuscole.

Digitare nuovamente la passphrase per la verifica. Una volta digitata correttamente, la chiave per il server viene generata e archiviata nel file `server.key`.

Creare la chiave insicura, quella priva di passphrase, e scambiare i nomi delle chiavi:

```
openssl rsa -in server.key -out server.key.insecure
mv server.key server.key.secure
mv server.key.insecure server.key
```

La chiave insicura è ora chiamata `server.key` ed è possibile usare questo file per generare la CSR senza passphrase.

Per creare il CSR, eseguire il seguente comando:

```
openssl req -new -key server.key -out server.csr
```

It will prompt you enter the passphrase. If you enter the correct passphrase, it will prompt you to enter Company Name, Site Name, Email Id, etc. Once you enter all these details, your CSR will be created and it will be stored in the `server.csr` file.

È ora possibile inviare il file della CSR alla CA che lo utilizzerà per creare il certificato finale. È comunque possibile creare un certificato auto-firmato utilizzando questa CSR.

5.3. Creare un certificato auto-firmato

Per creare un certificato auto-firmato, eseguire da un terminale il seguente comando:

```
openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt
```

Il comando precedente chiederà la passphrase. Una volta digitata correttamente, il certificato viene creato e sarà disponibile nel file `server.crt`.

Se il server deve essere utilizzato in ambito commerciale, è necessario un certificato emesso da una CA. Non è raccomandato utilizzare un certificato auto-firmato.

5.4. Installare il certificato

È possibile installare il file `server.key` e quello del certificato `server.crt`, o il file del certificato fornito dalla CA, eseguendo, in un terminale, i seguenti comandi:

```
sudo cp server.crt /etc/ssl/certs
sudo cp server.key /etc/ssl/private
```

Ora basta configurare le applicazioni che possono usare la crittografia a chiave pubblica affinché utilizzino i file del *certificato* e della *chiave*. Per esempio, Apache può fornire HTTPS, Dovecot può fornire IMAPS e POP3S ecc...

5.5. Autorità di Certificazione

Se i servizi all'interno della propria rete richiedono più di un certificato auto-firmato, potrebbe essere utile impostare una *Autorità di Certificazione* personale. Usando certificati firmati dalla propria CA,

consente ai vari servizi che usano tali certificati di fidarsi di altri servizi che fanno uso di certificati emessi dalla stessa CA.

1. Per prima cosa, creare le directory che conterranno il certificato della CA e i file relativi

   ```
   sudo mkdir /etc/ssl/CA
   sudo mkdir /etc/ssl/newcerts
   ```

2. La CA necessita di alcuni altri file per funzionare correttamente: uno per tenere traccia dell'ultimo numero seriale usato (ogni certificato deve avere un numero univoco) e l'altro per registrare quali certificati sono stati emessi:

   ```
   sudo sh -c "echo '01' > /etc/ssl/CA/serial"
   sudo touch /etc/ssl/CA/index.txt
   ```

3. Il terzo file è il file di configurazione della CA. Benché non strettamente necessario, è molto utile quando vengono emessi certificati multipli. Aprire il file `/etc/ssl/openssl.cnf` e nella sezione *[CA_default]* modificare:

   ```
   dir             = /etc/ssl/             # Dove viene salvato tutto
   database        = $dir/CA/index.txt     # File indice del database
   certificate     = $dir/certs/cacert.pem # Il certificato della CA
   serial          = $dir/CA/serial        # Il numero seriale corrente
   private_key     = $dir/private/cakey.pem# La chiave privata
   ```

4. Creare il certificato auto-firmato principale:

   ```
   openssl req -new -x509 -extensions v3_ca -keyout cakey.pem -out cacert.pem -days 3650
   ```

   Viene chiesto di inserire i dettagli del certificato.

5. Installare il certificato principale e la chiave:

   ```
   sudo mv cakey.pem /etc/ssl/private/
   sudo mv cacert.pem /etc/ssl/certs/
   ```

6. È ora possibile firmare i certificati. La prima cosa necessaria è una CSR (Certificate Signing Request), consultare *Sezione 5.2, «Generare una CSR (Certificate Signing Request)» [173]*. Una volta ottenuta, digitare quanto segue per generare un certificato firmato dalla CA:

   ```
   sudo openssl ca -in server.csr -config /etc/ssl/openssl.cnf
   ```

   Inserita la passoword della chiave CA, viene chiesto di firmare il certificato e di generare quello nuovo. Dovrebbe quindi essere visibile l'output della generazione del certificato stesso.

7. There should now be a new file, `/etc/ssl/newcerts/01.pem`, containing the same output. Copy and paste everything beginning with the line: -----*BEGIN CERTIFICATE*----- and continuing through the line: ----*END CERTIFICATE*----- lines to a file named after the hostname of the

server where the certificate will be installed. For example `mail.example.com.crt`, is a nice descriptive name.

Tutti i certificati successivi saranno chiamati `02.pem`, `03.pem`, ecc...

Sostituire *mail.example.it.crt* con un nome descrittivo appropriato al proprio caso.

8. In fine, copiare il nuovo certificato nell'host e configurare le applicazioni al suo uso. La posizione predefinita per l'installazione dei certificati è `/etc/ssl/certs`, consentendo così a molteplici servizi di usare lo stesso certificato senza complicare inutilmente i permessi.

   Per le applicazioni che possono essere configurate all'uso di un certificato di una CA, è necessario copiare il file `/etc/ssl/certs/cacert.pem` nella directory `/etc/ssl/certs/` di ogni server.

5.6. Riferimenti

- Per ulteriori informazioni sull'utilizzo della crittografia, consultare lo *SSL Certificates HOWTO*[12].
- La pagina Wikipedia *HTTPS*[13] dispone di ulteriori informazioni riguardo HTTPS.
- Per maggiori informazioni riguardo *OpenSSL*, consultare il *sito web di OpenSSL*[14].
- Inoltre, il libro *Network Security with OpenSSL*[15] di O'Reilly è un ottimo punto di riferimento.

[12] http://tldp.org/HOWTO/SSL-Certificates-HOWTO/index.html
[13] http://it.wikipedia.org/wiki/HTTPS
[14] http://www.openssl.org/
[15] http://oreilly.com/catalog/9780596002701/

6. eCryptfs

eCryptfs è un file system crittografico POSIX-conforme per Linux. Disponendosi al di sopra del livello del file system normale, *eCryptfs* è in grado di proteggere i file indipendentemente dal file system sottostante, dal tipo di partizione, ecc...

Durante la fase di installazione è disponibile un'opzione per cifrare l'intera partizione `/home` in grado di configurare tutto il necessario per cifrare e montare la partizione.

As an example, this section will cover configuring `/srv` to be encrypted using *eCryptfs*.

6.1. Usare eCryptfs

Per prima cosa, installare i pacchetti necessari. In un terminale digitare:

```
sudo apt-get install ecryptfs-utils
```

Montare la partizione da cifrare:

```
sudo mount -t ecryptfs /srv /srv
```

Vengono chiesti alcuni dettagli su come ecryptfs dovrebbe cifrare i dati.

Per verificare che i file in `/srv` siano veramente cifrati, copiare la directory `/etc/default` in `/srv`:

```
sudo cp -r /etc/default /srv
```

Smontare `/srv` e cercare di visualizzare un file:

```
sudo umount /srv
cat /srv/default/cron
```

Montare `/srv` utilizzando ecryptfs per poter visualizzare nuovamente i dati.

6.2. Montare automaticamente le partizioni cifrate

È possibile montare un file system ecryptfs in diversi modi all'avvio. Questo esempio fa uso di un file `/root/.ecryptfsrc` contenente le opzioni di mount e un file, salvato su una chiave USB, contenente la passphrase.

Creare il file `/root/.ecryptfsrc` contenente:

```
key=passphrase:passphrase_passwd_file=/mnt/usb/passwd_file.txt
ecryptfs_sig=5826dd62cf81c615
ecryptfs_cipher=aes
ecryptfs_key_bytes=16
```

```
ecryptfs_passthrough=n
ecryptfs_enable_filename_crypto=n
```

Modificare il campo *ecryptfs_sig* con la firma presente in `/root/.ecryptfs/sig-cache.txt`.

Creare il file `/mnt/usb/passwd_file.txt` per la passphrase:

```
passphrase_passwd=[secrets]
```

Aggiungere quanto necessario in `/etc/fstab`:

```
/dev/sdb1       /mnt/usb        ext3    ro      0 0
/srv /srv ecryptfs defaults 0 0
```

Assicurarsi che il dispositivo USB venga montato prima della partizione cifrata.

Finally, reboot and the `/srv` should be mounted using *eCryptfs*.

6.3. Altre utilità

Il pacchetto ecryptfs-utils contiene diverse utilità:

- *ecryptfs-setup-private*: crea una directory `~/Private` per contenere informazioni cifrate. Questa utilità può essere eseguita da utenti senza alcun tipo di privilegio all'interno del sistema per creare una piccola zona privata dove salvare dati.
- *ecryptfs-mount-private* e *ecryptfs-umount-private*: monta e smonta la directory `~/Private` degli utenti.
- *ecryptfs-add-passphrase*: aggiunge una nuova passphrase al portachiavi.
- *ecryptfs-manager*: gestisce gli oggetti eCryptfs come le chiavi.
- *ecryptfs-stat* consente di visualizzare le meta informazioni di ecryptfs relative a un file.

6.4. Riferimenti

- For more information on *eCryptfs* see the *Launchpad project page*[16].
- There is also a *Linux Journal*[17] article covering *eCryptfs*.
- Also, for more ecryptfs options see the *ecryptfs man page*[18].
- The *eCryptfs Ubuntu Wiki*[19] page also has more details.

[16] https://launchpad.net/ecryptfs
[17] http://www.linuxjournal.com/article/9400
[18] http://manpages.ubuntu.com/manpages/precise/en/man7/ecryptfs.7.html
[19] https://help.ubuntu.com/community/eCryptfs

Capitolo 10. Monitoraggio

1. Panoramica

Il monitoraggio di server e servizi essenziali è un aspetto importante dell'amministrazione di sistema. La maggior parte dei servizi di rete vengono monitorati per controllarne prestazioni, disponibilità oppure entrambi. Questa sezione descrive l'installazione e la configurazione di Nagios per il monitoraggio mirato alla disponibilità dei servizi e di Munin per il monitoraggio delle prestazioni.

Gli esempi in questa sezione utilizzano due server con nome host *server01* e *server02*. Il server chiamato *server01* viene configurato con Nagios per monitorare i servizi sul server stesso e su *server02*. Inoltre, viene configurato anche munin per raccogliere informazioni dalla rete. Utilizzando il pacchetto munin-node, *server02* viene configurato per inviare informazioni a *server01*.

Questi semplice esempi dovrebbe permettere di monitorare server aggiuntivi e servizi all'interno della rete.

2. Nagios

2.1. Installazione

Per prima cosa, su *server01* installare il pacchetto nagios. In un terminale digitare:

```
sudo apt-get install nagios3 nagios-nrpe-plugin
```

Viene chiesto di inserire una password per l'utente *nagiosadmin*. Le credenziali vengono salvate nel file `/etc/nagios3/htpasswd.users`. Per modificare la password dell'utente *nagiosadmin* o per aggiungere altri utenti, usare il comando htpasswd, parte del pacchetto apache2-utils.

Per esempio, per modificare la password dell'utente *nagiosadmin* digitare:

```
sudo htpasswd /etc/nagios3/htpasswd.users nagiosadmin
```

Per aggiungere un utente:

```
sudo htpasswd /etc/nagios3/htpasswd.users steve
```

Su *server02* installare il pacchetto nagios-nrpe-server. Da un terminale su server02 inserire:

```
sudo apt-get install nagios-nrpe-server
```

NRPE consente di eseguire controlli locali sugli host remoti. Esistono anche altri metodi per eseguire questo attraverso l'uso di altri plugin o controlli di Nagios.

2.2. Panoramica della configurazione

Esistono diverse directory contenenti file di configurazione e di controllo di Nagios.

- `/etc/nagios3`: contiene i file di configurazione per le operazioni del demone nagios, i file CGI, host, ecc...
- `/etc/nagios-plugins`: contiene i file di configurazione per i controlli del servizio.
- `/etc/nagios`: sull'host remoto contiene i file di configurazione di nagios-nrpe-server.
- `/usr/lib/nagios/plugins/`: contiene i file binari dei controlli. Per visualizzare le opzioni di un controllo, usare l'opzione *-h*.

  Per esempio: **/usr/lib/nagios/plugins/check_dhcp -h**

Esistono moltissimi controlli che è possibile eseguire tramite Nagios su un qualsiasi host. In questo esempio Nagios viene configurato per controlla lo spazio su disco, DNS e un gruppo di host MySQL. Il controllo DNS avviene su *server02* e il gruppo di host MySQL include sia *server01* che *server02*.

Consultare *Sezione 1, «HTTPD - Server web Apache2» [188]* per informazioni su Apache, *Capitolo 8, DNS (Domain Name Service) [140]* su DNS e *Sezione 1, «MySQL» [207]* su MySQL.

Inoltre, vi sono alcuni termini che una volta descritti, aiuteranno a rendere più semplice la comprensione di Nagios:

- *Host*: un server, una workstation o un dispositivo di rete che viene monitorato.
- *Gruppo di host*: un gruppo di host simili. Per esempio potrebbe essere possibile raggruppare tutti i server web, i server di file, ecc...
- *Servizio*: il servizio che viene monitorato sull'host come HTTP, DNS, FTP, ecc...
- *Gruppo di servizi*: consente di raggruppare servizi simili. Utile, per esempio, per raggruppare più servizi HTTP.
- *Contatto*: una persona da notificare quando si verifica un evento. Nagios può essere configurato per inviare email, SMS, ecc...

Come impostazione predefinita, Nagios è configurato per controllare HTTP, spazio su disco, SSH, gli utenti attuali, i processi e il carico sul *localhost*. Inoltre, è in grado di controllare attraverso il comando ping il *gateway*.

Installazioni di Nagios di grosse dimensioni possono essere complesse da configurare ed è quindi utile partire con una configurazione piccola, uno o due host, prima di aumentare le dimensioni.

2.3. Configurazione

- 1. First, create a *host* configuration file for *server02*. Unless otherwise specified, run all these commands on *server01*. In a terminal enter:

```
sudo cp /etc/nagios3/conf.d/localhost_nagios2.cfg \
/etc/nagios3/conf.d/server02.cfg
```

Nei comandi precedenti e in quelli che seguono, sostituire *"server01"*, *"server02"*, *172.18.100.100* e *172.18.100.101* con i nomi host e gli indirizzi IP dei propri server.

2. Modificare il file `/etc/nagios3/conf.d/server02.cfg`:

```
define host{
        use                     generic-host  ; Name of host template to use
        host_name               server02
        alias                   Server 02
        address                 172.18.100.101
}

# check DNS service.
define service {
        use                             generic-service
        host_name                       server02
        service_description             DNS
        check_command                   check_dns!172.18.100.101
}
```

3. Riavviare il demone nagios per abilitare la nuova configurazione:

```
sudo /etc/init.d/nagios3 restart
```

- 1. Aggiungere una definizione di servizio per il controllo MySQL aggiungendo quanto segue al file `/etc/nagios3/conf.d/services_nagios2.cfg`:

```
# check MySQL servers.
define service {
        hostgroup_name        mysql-servers
        service_description   MySQL
        check_command         check_mysql_cmdlinecred!nagios!secret!$HOSTADDRESS
        use                   generic-service
        notification_interval 0 ; set > 0 if you want to be renotified
}
```

2. A *mysql-servers* hostgroup now needs to be defined. Edit `/etc/nagios3/conf.d/hostgroups_nagios2.cfg` adding:

```
# MySQL hostgroup.
define hostgroup {
        hostgroup_name  mysql-servers
                alias           MySQL servers
                members         localhost, server02
        }
```

3. Il controllo di Nagios necessita di autenticarsi con MySQL. Per aggiungere un utente *nagios* a MySQL inserire:

```
mysql -u root -p -e "create user nagios identified by 'secret';"
```

È necessario aggiungere l'utente *nagios* a tutti gli host deò gruppo *mysql-servers* hostgroup.

4. Riavviare nagios per iniziare il controllo dei server MySQL.

```
sudo /etc/init.d/nagios3 restart
```

- 1. Infine configurare NRPE affinché controlli lo spazio su disco su *server02*.

Sul *server01* aggiungere il controllo del servizio al file `/etc/nagios3/conf.d/server02.cfg`:

```
# NRPE disk check.
define service {
        use                     generic-service
        host_name               server02
        service_description     nrpe-disk
        check_command           check_nrpe_1arg!check_all_disks!172.18.100.101
}
```

2. Su *server02* modificare il file `/etc/nagios/nrpe.cfg`:

```
allowed_hosts=172.18.100.100
```

E nella sezione dove sono definiti i comandi, aggiungere:

```
command[check_all_disks]=/usr/lib/nagios/plugins/check_disk -w 20% -c 10% -e
```

3. Infine, riavviare nagios-nrpe-server:

```
sudo /etc/init.d/nagios-nrpe-server restart
```

4. Riavviare, su *server01*, nagios:

```
sudo /etc/init.d/nagios3 restart
```

Dovrebbe essere possibile visualizzare l'host e i controlli nei file CGI di Nagios. Per accedere a questi file, in un browser web inserire l'indirizzo http://server01/nagios3. Vengono richiesti password e nome utente dell'utente *nagiosadmin*.

2.4. Riferimenti

Questa sezione ha fornito una panoramica preliminare delle caratteristiche di Nagios, i pacchetti nagios-plugins-extra e nagios-snmp-plugins contengono molti altri controlli.

- Per maggiori informazioni, consultare il sito web di *Nagios*[1].
- In particolare, consultare la *documentazione in rete*[2].
- Sono disponibili anche molti *libri*[3] riguardo Nagios e il monitoraggio di rete:
- The *Nagios Ubuntu Wiki*[4] page also has more details.

[1] http://www.nagios.org/
[2] http://nagios.sourceforge.net/docs/3_0/
[3] http://www.nagios.org/propaganda/books/
[4] https://help.ubuntu.com/community/Nagios

3. Munin

3.1. Installazione

Prima di installare Munin su *server01*, è necessario installare apache2. La configurazione predefinita è sufficiente per poter eseguire un server munin. Per maggiori informazioni, consultare *Sezione 1, «HTTPD - Server web Apache2» [188]*.

Installare, su *server01*, munin. In un terminale, inserire:

```
sudo apt-get install munin
```

Su *server02*, installare il pacchetto munin-node:

```
sudo apt-get install munin-node
```

3.2. Configurazione

Su *server01* modificare il file `/etc/munin/munin.conf` aggiungendo l'indirizzo IP di *server02*:

```
## First our "normal" host.
[server02]
       address 172.18.100.101
```

Sostituire *server02* e *172.18.100.101* con il nome host e con l'indirizzo IP del proprio server.

Successivamente, configurare munin-node su *server02*. Modificare il file `/etc/munin/munin-node.conf` per consentire l'accesso al *server01*:

```
allow ^172\.18\.100\.100$
```

Sostituire *^172\.18\.100\.100$* con l'indirizzo IP del proprio server munin.

Riavviare munin-node su *server02* per applicare le modifiche:

```
sudo /etc/init.d/munin-node restart
```

Infine, in un browser, inserire l'indirizzo *http://server01/munin* per visualizzare grafici che rappresentano le informazioni dal pacchetto *munin-plugins* standard per disco, rete, processi e sistema.

Poiché è una nuova installazione, potrebbe impiegare un po' di tempo affinché i grafici visualizzino qualche cosa di utile.

3.3. Plugin aggiuntivi

Il pacchetto munin-plugins-extra contiene controlli per le prestazioni e per servizi come DNS, DHCP, Samba e altri. Per installare il pacchetto, in un terminale inserire:

```
sudo apt-get install munin-plugins-extra
```

Assicurarsi di installare il pacchetto sia sul server che su tutti i nodi.

3.4. Riferimenti

- Per maggiori informazioni, consultare il sito web di *Munin*[5].
- In particolare, la pagina relativa *alla documentazione*[6] contiene informazioni su maggiori plugin, sulla scrittura di plugin, ecc..
- È anche disponibile un libro in tedesco da Open Source Press: *Munin Graphisches Netzwerk- und System-Monitoring*[7].
- Another resource is the *Munin Ubuntu Wiki*[8] page.

[5] http://munin.projects.linpro.no/
[6] http://munin.projects.linpro.no/wiki/Documentation
[7] https://www.opensourcepress.de/index.php?26&backPID=178&tt_products=152
[8] https://help.ubuntu.com/community/Munin

Capitolo 11. Server web

Un server web è un programma interattivo che accetta richieste HTTP da client, noti come browser web, e invia loro risposte HTTP insieme ad altri dati opzionali, di solito pagine web come documenti HTML e oggetti collegati (immagini, ecc.).

1. HTTPD - Server web Apache2

Apache is the most commonly used Web Server on Linux systems. Web Servers are used to serve Web Pages requested by client computers. Clients typically request and view Web Pages using Web Browser applications such as Firefox, Opera, Chromium, or Mozilla.

Users enter a Uniform Resource Locator (URL) to point to a Web server by means of its Fully Qualified Domain Name (FQDN) and a path to the required resource. For example, to view the home page of the *Ubuntu Web site*[1] a user will enter only the FQDN:

```
www.ubuntu.com
```

To view the *community*[2] sub-page, a user will enter the FQDN followed by a path:

```
www.ubuntu.com/community
```

Il protocollo più utilizzato per il trasferimento delle pagine web è l'HTTP (Hyper Text Transfer Protocol). Sono anche supportati protocolli come HTTPS (Hyper Text Transfer Protocol over Secure Sockets Layer) e FTP (File Transfer Protocol), un protocollo per caricare e scaricare file dalla rete.

I server web Apache vengono comunemente usati in combinazione con il motore di database MySQL, il linguaggio di script per la pre-elaborazione dell'ipertesto PHP (Pre-processor Hyper Text) e altri noti linguaggi di script come Python e Perl. Questa configurazione viene denominata LAMP (Linux, Apache, MYSQL e Perl/Phyton/PHP) e costituisce una piattaforma robusta e potente per lo sviluppo e l'installazione di applicazioni basate sul web.

1.1. Installazione

Il server web Apache2 è disponibile in Ubuntu 10.04. Per installare Apache2:

- Al prompt di un terminale, eseguire il seguente comando:

```
sudo apt-get install apache2
```

1.2. Configurazione

La configurazione di Apache2 avviene scrivendo delle *direttive* in semplici file di testo. Queste *direttive* sono suddivise tra i seguenti file e directory:

- *apache2.conf*: il principale file di configurazione di Apache2. Contiene impostazioni *globali* per Apache2.

[1] http://www.ubuntu.com
[2] http://www.ubuntu.com/community

- *conf.d*: contiene file di configurazione che si applicano *globalmente* ad Apache2. Altri pacchetti che usano Apache2 per fornire contenuti possono aggiungere file o collegamenti simbolici in questa directory.
- *envvars*: file dove vengono impostate le variabili *d'ambiente* di Apache2.
- *httpd.conf:* historically the main Apache2 configuration file, named after the httpd daemon. Now the file is typically empty, as most configuration options have been moved to the below referenced directories. The file can be used for *user specific* configuration options that globally effect Apache2.
- *mods-available*: questa directory contiene file di configurazione per caricare e configurare *moduli*. Non tutti i moduli hanno file di configurazione specifici.
- *mods-enabled*: contiene *collegamenti simbolici* ai file in `/etc/apache2/mods-available`. Quando viene creato un collegamento simbolico a un modulo di configurazione, viene abilitato al successivo riavvio di apache2.
- *ports.conf*: contiene le direttive che determinano su quali porte TCP Apache2 sta in ascolto.
- *sites-available*: questa directory contiene i file di configurazione per i *Virtual Hosts* di Apache2. Questi consentono di configurare Apache2 affinché venga utilizzato per siti multipli con configurazioni separate.
- *sites-enabled*: come mods-enabled, `sites-enabled` contiene collegamenti simbolici alla directory `/etc/apache2/sites-available`. Quando viene creato un collegamento simbolico di un file di configurazione nella directory sites-available, il sito configurato sarà attivo al riavvio di Apache2.

Altri file di configurazione possono essere aggiunti attraverso la direttiva *Include* e caratteri speciali possono essere usati per aggiungere molti altri file di configurazione. Una qualsiasi direttiva può essere posizionata in uno qualsiasi di questi file di configurazione. Modifiche ai file principali di configurazione vengono riconosciute solo con un riavvio di Apache2.

The server also reads a file containing mime document types; the filename is set by the *TypesConfig* directive, typically via `/etc/apache2/mods-available/mime.conf`, which might also include additions and overrides, and is `/etc/mime.types` by default.

1.2.1. Impostazioni di base

Questa sezione descrive i parametri di configurazione fondamentali del server Apache2. Per maggiori informazioni, consultare la *documentazione di Apache2*[3].

- Apache2 è dotato di una configurazione predefinita adatta agli host virtuali: è configurato con un singolo host virtuale (attraverso l'uso della direttiva *VirtualHost*) che può essere modificato oppure usato così com'è nel caso si disponga di un solo sito web oppure usato come modello per aggiungere altri host virtuali. Se lasciato così, l'host virtuale predefinito verrà usato come sito predefinito o come il sito che gli utenti vedranno se l'URL inserito non corrisponde alla direttiva

[3] http://httpd.apache.org/docs/2.2/

ServerName in uno qualsiasi dei file personalizzati. Per modificare l'host virtuale, modificare il file `/etc/apache2/sites-available/default`.

Le direttive impostate per un host virtuale si applicano solamente a quel particolare host. Se una direttiva è impostata all'interno del server e non è definita nelle impostazioni dell'host virtuale, vengono utilizzate le impostazioni predefinite. Per esempio, è possibile impostare un indirizzo email per il webmaster e non definirne alcuno per per gli host virtuali.

Per configurare un nuovo host virtuale o un nuovo sito, copiare quel file nella stessa directory con un nome a scelta. Per esempio:

```
sudo cp /etc/apache2/sites-available/default /etc/apache2/sites-available/mionuovosito
```

Modificare il file per configurare il nuovo sito usando alcune delle direttive descritte di seguito.

- La direttiva *ServerAdmin* specifica a quale indirizzo email il sistema deve indirizzare la posta destinata agli amministratori. Il valore predefinito è «webmaster@localhost». Quest'impostazione deve essere modificata con l'indirizzo che è stato assegnato all'utente (nel caso sia l'amministratore). Se il sito presenta dei problemi, Apache2 mostrerà un messaggio di errore indicante l'indirizzo a cui deve essere segnalato il problema. Questa direttiva è presente nel file /etc/apache2/sites-available del proprio sito.
- La direttiva *Listen* specifica la porta, e opzionalmente l'indirizzo IP, su cui Apache2 dovrebbe essere in ascolto. Se l'indirizzo IP non è specificato, Apache2 ascolta tutti gli indirizzi IP assegnati alla macchina. Il valore predefinito per la direttiva Listen è 80. Modificare questo valore, in 127.0.0.1:80 per fare in modo che Apache2 ascolti solo l'interfaccia di loopback e non sia disponibile verso internet, in 81 per modificare la porta di ascolto o lasciare il valore predefinito per il normale funzionamento. Questa direttiva può essere trovata e modifica in un file specifico: `/etc/apache2/ports.conf`
- La direttiva *ServerName* è opzionale e specifica il FQDN a cui il proprio sito risponde. L'host virtuale predefinito non ha la direttiva ServerName impostata, cosicché risponderà a tutte le richieste che non corrispondono alla direttiva ServerName in un altro host virtuale. Se si è i proprietari del dominio "ubunturocks.com" e si vuole ospitare tale dominio su un server Ubuntu, il valore della direttiva ServerName nel file di configurazione dell'host virtuale dovrebbe essere "ubunturocks.com". Aggiungere quindi questa direttiva al nuovo file di configurazione creato precedentemente (`/etc/apache2/sites-available/mionuovosito`).

  Potrebbe essere necessario che il proprio sito risponda anche alle richieste per "www.ubunturocks.com", dato che molti utenti ritengono corretto inserire il prefisso "www". Per ottenere questo, usare la direttiva *ServerAlias*: è possibile usare anche caratteri speciali con la direttiva ServerAlias.

  Per esempio, la seguente configurazione farà in modo che il proprio sito risponda a qualsiasi richiesta il cui dominio termina con *.ubunturocks.com*.

```
ServerAlias *.ubunturocks.com
```

- The *DocumentRoot* directive specifies where Apache2 should look for the files that make up the site. The default value is /var/www, as specified in `/etc/apache2/sites-available/default`. If desired, change this value in your site's virtual host file, and remember to create that directory if necessary!

Abilitare il nuovo *VirtualHost* utilizzando l'utilità a2ensite e riavviare Apache2:

```
sudo a2ensite mionuovosito
sudo service apache2 restart
```

Assicurarsi di sostituire *mionuovosito* con una nome più descrittivo per il VirtualHost. Un metodo molto utilizzato consiste nel definire il nome del file secondo la direttiva *ServerName* dell'host virtuale.

Allo stesso modo, usare l'utilità a2dissite per disabilitare i siti. Questo può rivelarsi utile per diagnosticare problemi di configurazione con molteplici host virtuali:

```
sudo a2dissite mionuovosito
sudo service apache2 restart
```

1.2.2. Impostazioni predefinite

Questa sezione si occupa delle impostazioni predefinite del server Apache2. Per esempio, se viene aggiunto un host virtuale, le impostazioni modificate dell'host virtuale hanno precedenza rispetto quelle dell'host. Per una direttiva non definita, viene utilizzato il valore predefinito.

- *DirectoryIndex* è la pagina predefinita proposta dal server alle richieste dell'indice di una directory, specificate attraverso l'uso di una barra (/) come suffisso al nome della directory.

  For example, when a user requests the page http://www.example.com/this_directory/, he or she will get either the DirectoryIndex page if it exists, a server-generated directory list if it does not and the Indexes option is specified, or a Permission Denied page if neither is true. The server will try to find one of the files listed in the DirectoryIndex directive and will return the first one it finds. If it does not find any of these files and if *Options Indexes* is set for that directory, the server will generate and return a list, in HTML format, of the subdirectories and files in the directory. The default value, found in `/etc/apache2/mods-available/dir.conf` is "index.html index.cgi index.pl index.php index.xhtml index.htm". Thus, if Apache2 finds a file in a requested directory matching any of these names, the first will be displayed.
- The *ErrorDocument* directive allows you to specify a file for Apache2 to use for specific error events. For example, if a user requests a resource that does not exist, a 404 error will occur. By default, Apache2 will simply return a HTTP 404 Return code. Read `/etc/apache2/conf.d/localized-error-pages` for detailed instructions for using ErrorDocument, including locations of example files.

- By default, the server writes the transfer log to the file `/var/log/apache2/access.log`. You can change this on a per-site basis in your virtual host configuration files with the *CustomLog* directive, or omit it to accept the default, specified in `/etc/apache2/conf.d/other-vhosts-access-log`. You may also specify the file to which errors are logged, via the *ErrorLog* directive, whose default is `/var/log/apache2/error.log`. These are kept separate from the transfer logs to aid in troubleshooting problems with your Apache2 server. You may also specify the *LogLevel* (the default value is "warn") and the *LogFormat* (see `/etc/apache2/apache2.conf` for the default value).
- Alcune opzioni vengono specificate per directory piuttosto che per server, come la direttiva *Options*. Una stanza "Directory" è racchiusa tra tag in stile XML:

```
<Directory /var/www/mionuovosito>
...
</Directory>
```

La direttiva *Options* all'interno della stanza "Directory" accetta uno o più dei seguenti valori (tra gli altri) separati da spazi:

- **ExecCGI**: consente l'esecuzione di script CGI. Questi script non vengono eseguiti se l'opzione non è selezionata.

La maggior parte dei file non dovrebbe venir eseguita come script CGI, potrebbe essere molto pericoloso. Gli script CGI dovrebbero essere mantenuti in una directory separata, al di fuori della propria DocumentRoot e solo questa directory dovrebbe avere l'opzione ExecCGI impostata. Questo è il comportamento predefinito in Ubuntu e la posizione per gli script CGI è `/usr/lib/cgi-bin`.

- **Includes** - Allow server-side includes. Server-side includes allow an HTML file to *include* other files. See *Apache SSI documentation (Ubuntu community)*[4] for more information.
- **IncludesNOEXEC**: consente inclusioni lato server, ma disabilita i comandi *#exec* e *#include* negli script CGI.
- **Indexes**: visualizza un elenco formattato dei contenuti della directory se non esiste alcun *DirectoryIndex* (come index.html) nella directory richiesta.

Per motivi di sicurezza, quest'opzione non dovrebbe essere impostata e soprattutto non su DocumentRoot. Abilitare questa opzione con molta cautela solo su alcune directory e nel caso in cui si voglia visualizzare l'intero contenuto della directory.

- **Multiview**: supporta visualizzazioni multiple in base al contenuto, quest'opzione è disabilitata in modo predefinito per ragioni di sicurezza. Per maggiori informazioni, consultare *la documentazione di Apache2*[5].
- **SymLinksIfOwnerMatch**: segue i collegamenti simbolici solamente se il file di arrivo o la directory hanno gli stessi proprietari del collegamento.

[4] https://help.ubuntu.com/community/ServerSideIncludes

[5] http://httpd.apache.org/docs/2.2/mod/mod_negotiation.html#multiviews

1.2.3. Impostazioni di httpd

Questa sezione espone alcune delle configurazioni di base del demone httpd.

LockFile: la direttiva LockFile imposta il percorso al file di lock utilizzato quando il server viene compilato con USE_FCNTL_SERIALIZED_ACCEPT o USE_FLOCK_SERIALIZED_ACCEPT. Deve essere conservato nel disco locale. Questo valore dovrebbe essere lasciato invariato a meno che la directory di log non sia localizzata su una condivisione NFS. In questo caso, il valore dovrebbe essere modificato con una posizione sul disco locale e una directory accessibile solamente dall'utente root.

PidFile: la direttiva PidFile imposta il file in cui il server registra il proprio «pid». Questo file dovrebbe essere leggibile solamente dall'utente root. Nella maggior parte dei casi può essere lasciata invariata.

User - The User directive sets the userid used by the server to answer requests. This setting determines the server's access. Any files inaccessible to this user will also be inaccessible to your website's visitors. The default value for User is "www-data".

A meno che non sia estremamente necessario, non impostare mai la direttiva «User» a root. Utilizzare root con «User» può creare una falla nella sicurezza del server Web.

Group - The Group directive is similar to the User directive. Group sets the group under which the server will answer requests. The default group is also "www-data".

1.2.4. Moduli di Apache2

Apache2 è un server modulare: solo le funzionalità basilari sono incluse nel server principale. È possibile estendere le funzionalità del server attraverso dei moduli che vengono caricati all'interno di Apache2. Un piccolo insieme di moduli è incluso nel server durante la compilazione: se il server è compilato per caricare i moduli dinamicamente, gli stessi moduli possono essere compilati separatamente e aggiungi quando necessario utilizzando la direttiva LoadModule; altrimenti è necessario ricompilare Apache2 per aggiungere o rimuovere i moduli.

La versione di Ubuntu consente il caricamento dinamico dei moduli. Le direttive di configurazione possono essere incluse in base alla presenza di un particolare modulo racchiudendole in un blocco tipo: <*IfModulo*> block.

È quindi possibile installare moduli aggiuntivi di Apache2 e usarli con il server web. Per esempio, per installare il modulo *MySQL Authentication*, in un terminale digitare quanto segue:

```
sudo apt-get install libapache2-mod-auth-mysql
```

Per altri moduli, consultare la directory `/etc/apache2/mods-available`.

Usare l'utilità a2enmod per abilitare un modulo:

```
sudo a2enmod auth_mysql
sudo service apache2 restart
```

Allo stesso modo, a2dismod disabiliterà un modulo:

```
sudo a2dismod auth_mysql
sudo service apache2 restart
```

1.3. Configurazione HTTPS

Il modulo mod_ssl aggiunge un'importante caratteristica al server Apache2, l'abilità di cifrare le comunicazioni. In questo modo, quando il browser utilizza la cifratura SSL per le comunicazioni, il prefisso «https://» verrà inserito nell'URL (Uniform Resource Locator) nella barra degli indirizzi.

Il modulo mod_ssl è disponibile nel pacchetto apache2-common. Per abilitare il modulo mod_ssl, eseguire il seguente comando in un terminale:

```
sudo a2enmod ssl
```

Esiste un file di configurazione HTTPS predefinito in `/etc/apache2/sites-available/default-ssl`. Affinché Apache2 possa fornire connessioni HTTPS, sono necessari un *certificato* e una *chiave*. La configurazione HTTPS predefinita utilizza un certificato e una chiave generati attraverso ssl-cert, utili in fase di test, ma da sostituire con con una versione specifica per il sito o il server. Per maggiori informazioni su come generare una chiave e su come procurarsi un certificato, consultare *Sezione 5, «Certificati» [172]*

Per configurare l'HTTPS per Apache2, digitare quanto segue:

```
sudo a2ensite default-ssl
```

Le directory `/etc/ssl/certs` e `/etc/ssl/private` sono le posizioni predefinite. Se si installa il certificato e la chiave in un'altra directory assicurarsi di modificare *SSLCertificateFile* e *SSLCertificateKeyFile* appropriatamente.

Con l'HTTPS configurato, riavviare il servizio per abilitare le nuove impostazioni:

```
sudo service apache2 restart
```

In base a come è stato ottenuto il certificato, potrebbe essere necessario inserire una passphrase quando viene avviato Apache2.

È possibile accedere alle pagine del server sicuro digitando «https://nome_host/url/» nel barra degli indirizzi del proprio browser.

1.4. Sharing Write Permission

For more than one user to be able to write to the same directory it will be necessary to grant write permission to a group they share in common. The following example grants shared write permission to `/var/www` to the group "webmasters".

```
sudo chgrp -R webmasters /var/www
sudo find /var/www -type d -exec chmod g=rwxs "{}" \;
sudo find /var/www -type f -exec chmod g=rws  "{}" \;
```

If access must be granted to more than one group per directory, enable Access Control Lists (ACLs).

1.5. Riferimenti

- La *documentazione di Apache2*[6] contiene informazioni dettagliate riguardo le direttive di configurazione di Apache2. Inoltre, per la documentazione ufficiale di Apache2, consultare il pacchetto apache2-doc.
- Per maggiori informazioni riguardo SSL, consultare la *documentazione di Mod SSL*[7].
- Il libro *Apache Cookbook*[8] di O'Reilly è un'ottima risorsa per informazioni su specifiche configurazioni di Apache2.
- Per domande relative alla versione di Ubuntu di Apache2, chiedere nel canale IRC *#ubuntu-server* sul server *freenode.net*[9].
- Una buona risorsa riguardo PHP e MySQL può essere trovata nella *documentazione online*[10].

6 http://httpd.apache.org/docs/2.2/
7 http://www.modssl.org/docs/
8 http://oreilly.com/catalog/9780596001919/
9 http://freenode.net/
10 https://help.ubuntu.com/community/ApacheMySQLPHP

2. PHP5 - Linguaggio di scripting

PHP è un linguaggio di script universale pensato per lo sviluppo web. Uno script PHP può essere inserito direttamente nel codice HTML. Questa sezione spiega come installare e configurare PHP5 in sistemi Ubuntu con Apache2 e MySQL.

Questa sessione da per scontato che Apache2 e il server MySQL siano installati e configurati. Per maggiori informazioni sull'installazione e sulla configurazione dei due server, consultare la rispettiva documentazione presenti in questo documento.

2.1. Installazione

The PHP5 is available in Ubuntu Linux. Unlike python and perl, which are installed in the base system, PHP must be added.

- Per installare PHP5 è possibile digitare, in un terminale, quanto segue:

  ```
  sudo apt-get install php5 libapache2-mod-php5
  ```

  È possibile eseguire script di PHP5 dalla riga di comando installando il pacchetto php5-cli. Per installare php5-cli è sufficiente eseguire il seguente comando al prompt del terminale:

  ```
  sudo apt-get install php5-cli
  ```

  È possibile inoltre eseguire gli script di PHP5 senza installare il modulo PHP5 di Apache. Per fare ciò, è sufficiente installare il pacchetto php5-cgi digitando il seguente comando al prompt del terminale: **sudo apt-get install php5-cgi**

  Per usare MySQL con PHP5 è necessario installare il pacchetto php5-mysql. Per installare php5-mysql, eseguire il seguente comando al prompt del terminale:

  ```
  sudo apt-get install php5-mysql
  ```

  Allo stesso modo, per usare PostgreSQL con PHP5 è necessario installare il pacchetto php5-pgsql. Per installare php5-pgsql, eseguire il seguente comando al prompt del terminale:

  ```
  sudo apt-get install php5-pgsql
  ```

2.2. Configurazione

Una volta installato PHP5, è possibile eseguire gli script di PHP5 dal browser web. Se il pacchetto php5-cli è installato, è possibile eseguire gli script PHP5 dal prompt dei comandi.

Il server web Apache2 è configurato, in modo predefinito, per eseguire gli script di PHP5. In altre parole, il modulo PHP5 quando viene installato, viene abilitato automaticamente nel server web

Apache2. Verificare che i file `/etc/apache2/mods-enabled/php5.conf` e `/etc/apache2/mods-enabled/php5.load` esistano. Se non dovessero esistere, è possibile abilitare il modulo usando il comando **a2enmod**.

Once you install PHP5 related packages and enabled PHP5 Apache 2 module, you should restart Apache2 Web server to run PHP5 scripts. You can run the following command at a terminal prompt to restart your web server:

```
sudo service apache2 restart
```

2.3. Test

Per verificare l'installazione, è possibile eseguire la funzione «phpinfo» di PHP5 come segue:

```
<?php
  phpinfo();
?>
```

È sufficiente copiare il contenuto precedente in un file, come `phpinfo.php`, e salvarlo nella directory **DocumentRoot** del server web Apache2. Una volta puntato il browser web all'indirizzo `http://hostname/phpinfo.php`, dovrebbero venir visualizzati i valori di molti parametri di configurazione di PHP5.

2.4. Riferimenti

- Per ulteriori informazioni, consultare la documentazione di *php.net*[11].
- Esistono diversi libri su PHP. O'Reilly dispone di due ottimi libri: *Learning PHP 5*[12] e *PHP Cook Book*[13].
- Consultare anche la *documentazione online*[14].

[11] http://www.php.net/docs.php
[12] http://oreilly.com/catalog/9780596005603/
[13] http://oreilly.com/catalog/9781565926813/
[14] https://help.ubuntu.com/community/ApacheMySQLPHP

3. Squid - Server proxy

Squid è un potente proxy cache server che fornisce servizi proxy e cache per HTTP (Hyper Text Transport Protocol), FTP (File Transfer Protocol) e molti altri protocolli di rete. Squid può implementare servizi di caching e proxy anche per richieste SSL (Secure Sockets Layer), caching per ricerche di DNS (Domain Name Server) e fornire un caching trasparente. Squid supporta molti protocolli per il caching come ICP (Internet Cache Protocol), HTCP (Hyper Text Caching Protocol), CARP (Cache Array Routing Protocol) e WCCP (Web Cache Coordination Protocol).

Il server Squid è una valida soluzione per le necessità di caching e proxy, scala dall'utilizzo in un piccolo ufficio fino alla grande impresa, fornendo, attraverso il protocollo SNMP (Simple Network Management Protocol), un meccanismo di controllo e monitoraggio dei parametri critici molto accurato. Nella selezione di un computer da utilizzare come procy Squid dedicato, o come server cache, assicurarsi che il sistema sia equipaggiato con una grande quantità di memoria fisica, dal momento che Squid mantiene un cache in memoria per aumentare le prestazioni.

3.1. Installazione

Per installare il server Squid, da terminale digitare:

```
sudo apt-get install squid
```

3.2. Configurazione

La configurazione di Squid avviene attraverso la modifica di alcune direttive presenti nel file `/etc/squid/squid.conf`. Gli esempi che seguono descrivono alcune delle direttive che possono essere modificate. Per maggiori informazioni sulla configurazione di Squid consultare la sezione «Riferimenti».

Prima di modificare il file di configurazione, è utile farne una copia e proteggerla dalla scrittura così, in caso di necessità, è possibile utilizzare il file originale.

Copiare il file `/etc/squid/squid.conf` e proteggerlo dalla scrittura utilizzando i seguenti comandi:

```
sudo cp /etc/squid/squid.conf /etc/squid/squid.conf.original
sudo chmod a-w /etc/squid/squid.conf.original
```

- Per impostare il server Squid affinché stia in ascolto sulla porta 8888 invece che sulla porta predefinita 3128, modificare la direttiva http_port:

  ```
  http_port 8888
  ```

- Modificare la direttiva visible_hostname per dare a Squid uno specifico hostname. Questo nome non deve essere necessariamente il nome del computer. Nell'esempio seguente è impostato a *weezie*

```
visible_hostname weezie
```

- Using Squid's access control, you may configure use of Internet services proxied by Squid to be available only users with certain Internet Protocol (IP) addresses. For example, we will illustrate access by users of the 192.168.42.0/24 subnetwork only:

  Aggiungere quanto segue alla **fine** della sezione ACL del file `/etc/squid/squid.conf`:

  ```
  acl fortytwo_network src 192.168.42.0/24
  ```

  Quindi aggiungere quanto segue all'**inizio** della sezione http_access del file `/etc/squid/squid.conf`:

  ```
  http_access allow fortytwo_network
  ```

- Utilizzando il sistema di controllo degli accessi di Squid, è possibile configurare l'utilizzo di alcuni servizi internet in proxy con Squid in alcune fasce orarie: L'esempio seguente descrive come consentire agli utenti l'accesso al servizio dalle 9:00 alle 17:00 dal lunedì al venerdì che utilizza la sotto rete 10.1.42.0/42:

  Aggiungere quanto segue alla **fine** della sezione ACL del file `/etc/squid/squid.conf`:

  ```
  acl biz_network src 10.1.42.0/24
  acl biz_hours time M T W T F 9:00-17:00
  ```

  Quindi aggiungere quanto segue all'**inizio** della sezione http_access del file `/etc/squid/squid.conf`:

  ```
  http_access allow biz_network biz_hours
  ```

Una volta apportate le modifiche al file `/etc/squid/squid.conf`, salvarlo e, per rendere effettivi i cambiamenti, riavviare squid utilizzando il comando:

```
sudo /etc/init.d/squid restart
```

3.3. Riferimenti

Sito web di Squid[15]

Ubuntu Wiki Squid[16] page.

[15] http://www.squid-cache.org/

[16] https://help.ubuntu.com/community/Squid

4. Ruby on Rails

Ruby on Rails è un ambiente web open source, per sviluppare applicazioni web che si avvalgono di database. È ottimizzato per la produttività sostenibile del programmatore dato che richiede di scrivere codice favorendo le convenzioni piuttosto che le configurazioni.

4.1. Installazione

Prima di installare Rails è necessario installare Apache e MySQL. Per installare il pacchetto Apache fare riferimento alla *Sezione 1, «HTTPD - Server web Apache2» [188]*, per MySQL fare riferimento alla *Sezione 1, «MySQL» [207]*.

Una volta installati Apache e MySQL, è possibile installare il pacchetto Ruby on Rails.

Per installare i pacchetti base di Ruby, digitare in un terminale il seguente comando:

```
sudo apt-get install rails
```

4.2. Configurazione

Modificare il file di configurazione `/etc/apache2/sites-available/default` per impostare i propri domini.

La prima cosa da cambiare è la direttiva *DocumentRoot*:

```
DocumentRoot /percorso/applicazione/rails/publicc
```

Successivamente, modificare <Directory "/percorso/applicazione/rails/public">:

```
<Directory "/percorso/applicazione/rails/public">
        Options Indexes FollowSymLinks MultiViews ExecCGI
        AllowOverride All
        Order allow,deny
        allow from all
        AddHandler cgi-script .cgi
</Directory>
```

È utile anche abilitare il modulo mod_rewrite di Apache. Per abilitare il modulo mod_rewrite, digitare il seguente comando in un terminale:

```
sudo a2enmod rewrite
```

Infine, è necessario modificare i proprietari delle directory `/percorso/applicazione/rails/public` e `/percorso/applicazione/rails/tmp` con il proprietario usato per eseguire il processo Apache:

```
sudo chown -R www-data:www-data /percorso/applicazione/rails/public
sudo chown -R www-data:www-data /percorso/applicazione/rails/tmp
```

Il server è ora pronto per le applicazioni Ruby on Rails.

4.3. Riferimenti

- Per ulteriori informazioni, consultare il *sito web di Ruby on Rails*[17].
- Anche *Agile Development with Rails*[18] è un'ottima risorsa.
- Another place for more information is the *Ruby on Rails Ubuntu Wiki*[19] page.

[17] http://rubyonrails.org/

[18] http://pragprog.com/titles/rails3/agile-web-development-with-rails-third-edition

[19] https://help.ubuntu.com/community/RubyOnRails

5. Apache Tomcat

Apache Tomcat è un "contenitore" web che consente di servire Java Servlets e applicazioni web JSP (Java Server Pages).

The Tomcat 6.0 packages in Ubuntu support two different ways of running Tomcat. You can install them as a classic unique system-wide instance, that will be started at boot time will run as the tomcat6 unprivileged user. But you can also deploy private instances that will run with your own user rights, and that you should start and stop by yourself. This second way is particularly useful in a development server context where multiple users need to test on their own private Tomcat instances.

5.1. Installazione globale

To install the Tomcat server, you can enter the following command in the terminal prompt:

```
sudo apt-get install tomcat6
```

In questo modo verrà installato il server Tomcat con un'applicazione web predefinita che visualizza una semplice pagina "It works".

5.2. Configurazione

I file di configurazione di Tomcat possono essere trovati in `/etc/tomcat6`. In questa sezione verranno spiegate solo alcune modifiche, per maggiori informazioni, consultare la *documentazione di Tomcat 6.0*[20].

5.2.1. Modificare la porta predefinita

Tomcat 6.0 esegue un connettore HTTP sulla porta 8080 e un connettore AJP sulla porta 8009; potrebbe essere utile modificare queste porte per evitare conflitti con altri server all'interno del sistema. Per fare questo, basta modificare le seguenti righe nel file `/etc/tomcat6/server.xml`:

```
<Connector port="8080" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443" />
...
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
```

5.2.2. Cambiare la JVM usata

Tomcat viene eseguito preferibilmente con OpenJDK-6, quindi con la JVM di Sun e infine con altre JVM. Se sono installate diverse JVM, è possibile impostare quale usare modificando la variabile JAVA_HOME nel file `/etc/default/tomcat6`:

[20] http://tomcat.apache.org/tomcat-6.0-doc/index.html

```
JAVA_HOME=/usr/lib/jvm/java-6-sun
```

5.2.3. Dichiarare utenti e ruoli

Nomi utente, password e ruoli (gruppi) possono essere definiti in un contenitore Servlet. Con Tomcat 6.0 questo è fatto nel file `/etc/tomcat6/tomcat-users.xml`:

```
<role rolename="admin"/>
<user username="tomcat" password="s3cret" roles="admin"/>
```

5.3. Usare le applicazioni web standard di Tomcat

Tomcat dispone di applicazioni web che è possibile installare per documentarsi, per l'amministrazione o solo per dimostrazione.

5.3.1. Documentazione di Tomcat

Il pacchetto tomcat6-docs contiene la documentazione di Tomcat 6.0 sotto forma di applicazione web a cui è possibile accedere all'indirizzo "http://IL_PROPRIO_SERVER:8080/docs". È possibile installare il pacchetto attraverso il seguente comando:

```
sudo apt-get install tomcat6-docs
```

5.3.2. Applicazioni web amministrative di Tomcat

Il pacchetto tomcat6-admin contiene due applicazioni web che possono essere usate per amministrare il server Tomcat attraverso un'interfaccia web. È possibile installarle attraverso il seguente comando:

```
sudo apt-get install tomcat6-admin
```

La prima applicazione è il cosiddetto *manager*, a cui è possibile accedere dall'indirizzo "http://IL_PROPRIO_SERVER:8080/manager/html". È principalmente usata per ottenere informazioni sul server e riavviare le applicazioni web.

L'accesso al *manager* è protetto: è necessario definire un utente con il ruolo di "manager" nel file `/etc/tomcat6/tomcat-users.xml` prima di potervi accedere:

La seconda applicazione è l'*host-manager* a cui è possibile accedere attraverso l'indirizzo "http://IL_PROPRIO_SERVER:8080/host-manager/html". È possibile usarla per creare host virtuali dinamicamente.

Anche l'accesso all'applicazione *host-manager* è protetto: è necessario definire un utente con il ruolo di "admin" nel file `/etc/tomcat6/tomcat-users.xml` prima di potervi accedere.

Per motivi di sicurezza, l'utente tomcat6 non può scrivere nella directory `/etc/tomcat6` e alcune di queste applicazioni di amministrazione (produzione delle applicazioni, creazione di host virtuali)

necessitano di accesso in scrittura in tale directory. Per poter usare queste caratteristiche, eseguire i seguenti comandi per dare agli utenti del gruppo tomcat6 i permessi necessari:

```
sudo chgrp -R tomcat6 /etc/tomcat6
sudo chmod -R g+w /etc/tomcat6
```

5.3.3. Applicazioni web di esempio

Il pacchetto tomcat6-examples contiene due applicazioni web che possono essere usate per verificare o dimostrare le Servlet o le caratteristiche di JSP e sono accessibile dall'indirizzo "http://IL_PROPRIO_SERVER:8080/examples". Per installarle, usare il seguente comando:

```
sudo apt-get install tomcat6-examples
```

5.4. Usare istanze private

Tomcat è spesso usato in ambienti di sviluppo e di test dove usare una singola istanza all'interno del sistema non risulta molto utile ai molteplici utenti che sfruttano il sistema. I pacchetti di Tomcat 6.0 sono dotati di strumenti che facilitano la creazione di istanze dedicate a ogni singolo utente, consentendo, all'interno del sistema, di eseguire (senza i privilegi di root) istanze private e separate usando però sempre le librerie di sistema.

È possibile eseguire le istanze globali e private in parallelo, basta solo che non usino le stesse porte TCP.

5.4.1. Installare il supporto alle istanze private

È possibile installare tutto il necessario per eseguire istanze private attraverso il seguente comando:

```
sudo apt-get install tomcat6-user
```

5.4.2. Creare un'istanza privata

È possibile creare un'istanza privata attraverso il seguente comando:

```
tomcat6-instance-create mia-istanza
```

In questo modo verrà creata una nuova directory `mia-istanza` con tutte le sottodirectory e gli script necessari. Sarà poi possibile installare le librerie comuni nella sottodirectory `lib/` e sviluppare le proprie applicazioni in `webapps/`. Non vi è alcuna applicazione predefinita in questa directory.

5.4.3. Configurare un'istanza privata

I file di configurazione di Tomcat per un'istanza privata sono disponibili nella sottodirectory `conf/`. È necessario modificare, per esempio, il file `conf/server.xml` per modificare le porte predefinite

usate dall'istanza privata di Tomcat per evitare conflitti con altre istanze che potrebbero essere in esecuzione.

5.4.4. Avviare e fermare un'istanza privata

È possibile avviare un'istanza privata utilizzando il seguente comando (si presuppone che l'istanza sia posizionata nella directory `mia-istanza`):

```
mia-istanza/bin/startup.sh
```

Controllare la sottodirectory `logs/` per la presenza di errori. Se si nota un errore del tipo "*java.net.BindException: Address already in use<null>:8080*", significa che la porta in uso è già utilizzata ed è necessario modificarla.

Per fermare un'istanza, usare il seguente comando (si presuppone che l'istanza sia posizionata nella directory `mia-istanza`):

```
mia-istanza/bin/shutdown.sh
```

5.5. Riferimenti

- Per ulteriori informazioni, consultare il *sito web di Apache Tomcat*[21].
- Il libro *Tomcat: The Definitive Guide*[22] è un'ottima risorsa per creare siti web con Tomcat.
- Per ulteriori libri, consultare la pagina *Tomcat Books*[23].
- Also, see the*Ubuntu Wiki Apache Tomcat*[24] page.

[21] http://tomcat.apache.org/
[22] http://oreilly.com/catalog/9780596003180/
[23] http://wiki.apache.org/tomcat/Tomcat/Books
[24] https://help.ubuntu.com/community/ApacheTomcat5

Capitolo 12. Database

Ubuntu fornisce due dei più popolari server database:

- MySQL™
- PostgreSQL

Questi sono disponibili nel repository "main". La seguente sezione descrive come installare e configurare questi database.

1. MySQL

MySQL is a fast, multi-threaded, multi-user, and robust SQL database server. It is intended for mission-critical, heavy-load production systems as well as for embedding into mass-deployed software.

1.1. Installazione

Per installare MySQL, eseguire il seguente comando dal terminale:

```
sudo apt-get install mysql-server
```

As of Ubuntu 12.04, MySQL 5.5 is installed by default. Whilst this is 100% compatible with MySQL 5.1 should you need to install 5.1 (for example to be a slave to other MySQL 5.1 servers) you can install the mysql-server-5.1 package instead.

During the installation process you will be prompted to enter a password for the MySQL root user.

Una volta completata l'installazione, il server MySQL dovrebbe avviarsi automaticamente. È possibile digitare i seguenti comandi in un terminale per controllare se il server è in esecuzione:

```
sudo netstat -tap | grep mysql
```

L'output del comando precedente dovrebbe essere:

```
tcp        0      0 localhost:mysql         *:*                 LISTEN      2556/mysqld
```

Se il server non funziana correttamente, è possibile digitare il seguente comando per avviarlo:

```
sudo service mysql restart
```

1.2. Configurazione

You can edit the `/etc/mysql/my.cnf` file to configure the basic settings -- log file, port number, etc. For example, to configure MySQL to listen for connections from network hosts, change the *bind-address* directive to the server's IP address:

```
bind-address            = 192.168.0.5
```

Sostituire 192.168.0.5 con l'indirizzo appropriato.

After making a change to `/etc/mysql/my.cnf` the MySQL daemon will need to be restarted:

```
sudo service mysql restart
```

If you would like to change the MySQL *root* password, in a terminal enter:

```
sudo dpkg-reconfigure mysql-server-5.5
```

The MySQL daemon will be stopped, and you will be prompted to enter a new password.

1.3. Database Engines

Whilst the default configuration of MySQL provided by the Ubuntu packages is perfectly functional and performs well there are things you may wish to consider before you proceed.

MySQL is designed to allow data to be stored in different ways. These methods are referred to as either database or storage engines. There are two main engines that you'll be interested in: InnoDB and MyISAM. Storage engines are transparent to the end user. MySQL will handle things differently under the surface, but regardless of which storage engine is in use, you will interact with the database in the same way.

Each engine has its own advantages and disadvantages.

While it is possible, and may be advantageous to mix and match database engines on a table level, doing so reduces the effectiveness of the performance tuning you can do as you'll be splitting the resources between two engines instead of dedicating them to one.

- MyISAM is the older of the two. It can be faster than InnoDB under certain circumstances and favours a read only workload. Some web applications have been tuned around MyISAM (though that's not to imply that they will slow under InnoDB). MyISAM also supports the FULLTEXT data type, which allows very fast searches of large quantities of text data. However MyISAM is only capable of locking an entire table for writing. This means only one process can update a table at a time. As any application that uses the table scales this may prove to be a hindrance. It also lacks journaling, which makes it harder for data to be recovered after a crash. The following link provides some points for consideration about using *MyISAM on a production database*[1].
- InnoDB is a more modern database engine, designed to be *ACID compliant*[2] which guarantees database transactions are processed reliably. Write locking can occur on a row level basis within a table. That means multiple updates can occur on a single table simultaneously. Data caching is also handled in memory within the database engine, allowing caching on a more efficient row level basis rather than file block. To meet ACID compliance all transactions are journaled independently of the main tables. This allows for much more reliable data recovery as data consistency can be checked.

As of MySQL 5.5 InnoDB is the default engine, and is highly recommended over MyISAM unless you have specific need for features unique to the engine.

[1] http://www.mysqlperformanceblog.com/2006/06/17/using-myisam-in-production/
[2] http://en.wikipedia.org/wiki/ACID

1.4. Advanced configuration

1.4.1. Creating a tuned my.cnf file

There are a number of parameters that can be adjusted within MySQL's configuration file that will allow you to improve the performance of the server over time. For initial set-up you may find *Percona's my.cnf generating tool*[3] useful. This tool will help generate a my.cnf file that will be much more optimised for your specific server capabilities and your requirements.

Do not replace your existing my.cnf file with Percona's one if you have already loaded data into the database. Some of the changes that will be in the file will be incompatible as they alter how data is stored on the hard disk and you'll be unable to start MySQL. If you do wish to use it and you have existing data, you will need to carry out a mysqldump and reload:

```
mysqldump --all-databases --all-routines -u root -p > ~/fulldump.sql
```

This will then prompt you for the root password before creating a copy of the data. It is advisable to make sure there are no other users or processes using the database whilst this takes place. Depending on how much data you've got in your database, this may take a while. You won't see anything on the screen during this process.

Once the dump has been completed, shut down MySQL:

```
sudo service mysql stop
```

Now backup the original my.cnf file and replace with the new one:

```
sudo cp /etc/my.cnf /etc/my.cnf.backup
sudo cp /path/to/new/my.cnf /etc/my.cnf
```

Then delete and re-initialise the database space and make sure ownership is correct before restarting MySQL:

```
sudo rm -rf /var/lib/mysql/*
sudo mysql_install_db
sudo chown -R mysql: /var/lib/mysql
sudo service start mysql
```

Finally all that's left is to re-import your data. To give us an idea of how far the import process has got you may find the 'Pipe Viewer' utility, pv, useful. The following shows how to install and use pv for this case, but if you'd rather not use it just replace pv with cat in the following command. Ignore any ETA times produced by pv, they're based on the average time taken to handle each row of the file, but the speed of inserting can vary wildly from row to row with mysqldumps:

[3] http://tools.percona.com/members/wizard

```
sudo apt-get install pv
pv ~/fulldump.sql | mysql
```

Once that is complete all is good to go!

This is not necessary for all my.cnf changes. Most of the variables you may wish to change to improve performance are adjustable even whilst the server is running. As with anything, make sure to have a good backup copy of config files and data before making changes.

1.4.2. MySQL Tuner

MySQL Tuner is a useful tool that will connect to a running MySQL instance and offer suggestions for how it can be best configured for your workload. The longer the server has been running for, the better the advice mysqltuner can provide. In a production environment, consider waiting for at least 24 hours before running the tool. You can get install mysqltuner from the Ubuntu repositories:

```
sudo apt-get install mysqltuner
```

Then once its been installed, run it:

```
mysqltuner
```

and wait for its final report. The top section provides general information about the database server, and the bottom section provides tuning suggestions to alter in your my.cnf. Most of these can be altered live on the server without restarting, look through the official MySQL documentation (link in Resources section) for the relevant variables to change in production. The following is part of an example report from a production database which shows there may be some benefit from increasing the amount of query cache:

```
-------- Recommendations -----------------------------------------------------
General recommendations:
    Run OPTIMIZE TABLE to defragment tables for better performance
    Increase table_cache gradually to avoid file descriptor limits
Variables to adjust:
    key_buffer_size (> 1.4G)
    query_cache_size (> 32M)
    table_cache (> 64)
    innodb_buffer_pool_size (>= 22G)
```

One final comment on tuning databases: Whilst we can broadly say that certain settings are the best, performance can vary from application to application. For example, what works best for Wordpress might not be the best for Drupal, Joomla or proprietary applications. Performance is dependent on the types of queries, use of indexes, how efficient the database design is and so on. You may find it useful to spend some time searching for database tuning tips based on what applications you're using it for. Once you get past a certain point any adjustments you make will only result in minor improvements, and you'll be better off either improving the application, or looking at scaling up your database environment through either using more powerful hardware or by adding slave servers.

1.5. Risorse

- Per maggiori informazioni, consultare *il sito web di MySQL*[4].
- Full documentation is available in both online and offline formats from the *MySQL Developers portal*[5]
- For general SQL information see *Using SQL Special Edition*[6] by Rafe Colburn.
- Ulteriori informazioni sono disponibili nella *documentazione online*[7].

[4] http://www.mysql.com/
[5] http://dev.mysql.com/doc/
[6] http://www.informit.com/store/product.aspx?isbn=0768664128
[7] https://help.ubuntu.com/community/ApacheMySQLPHP

2. PostgreSQL

PostgreSQL è un database relazionale orientato agli oggetti che presenta le caratteristiche di un database commerciale tradizionale e anche miglioramenti dei sistemi DBMS di prossima generazione.

2.1. Installazione

Per installare PostgreSQL, eseguire il seguente comando dal terminale:

```
sudo apt-get install postgresql
```

Una volta che l'installazione è completata, è possibile configurare il server PostgreSQL a seconda delle proprie esigenze, sebbene la configurazione predefinita sia abbastanza buona.

2.2. Configurazione

By default, connection via TCP/IP is disabled. PostgreSQL supports multiple client authentication methods. IDENT authentication method is used for postgres and local users, unless otherwise configured. Please refer *the PostgreSQL Administrator's Guide if you would like to configure alternatives like Kerberos*[8].

L'esempio seguente assume che si vogliano abilitare le connessioni TCP/IP e si voglia usare il metodo MD5 per l'autenticazione lato cliet. I file di configurazione di PostgreSQL sono presenti nella directory `/etc/postgresql/<version>/main`: se è installata la versione 8.4 di PostgreSQL, i file di configurazione sono nella directory `/etc/postgresql/8.4/main`.

To configure *ident* authentication, add entries to the `/etc/postgresql/8.4/main/pg_ident.conf` file. There are detailed comments in the file to guide you.

Per abilitare le connessioni TCP/IP, modificare il file `/etc/postgresql/8.4/main/postgresql.conf`

Localizzare la riga *#listen_addresses = 'localhost'* e modificarla in:

```
listen_addresses = 'localhost'
```

To allow other computers to connect to your PostgreSQL server replace 'localhost' with the *IP Address* of your server, or alternatively to '0.0.0.0' to bind to all interfaces.

Tutti gli altri parametri possono essere modificati, ma bisogna sapere cosa si sta facendo. Per maggiori informazioni, consultare la documentazione di PostgresSQL o fare riferimento ai file di configurazione.

Ora che è possibile collegarsi al server PostgreSQL, è necessario impostare una password per l'utente *postgres*. In un terminale, eseguire il seguente comando per connettersi al modello di database predefinito di PostgreSQL:

[8] http://www.postgresql.org/docs/8.4/static/admin.html

```
sudo -u postgres psql template1
```

Il comando precedente connette al database PostreSQL *template1* come l'utente *postgres*. Una volta collegati al server PostgreSQL, si sarà al prompt SQL. È possibile eseguire il seguente comando SQL al prompt psql per configurare la password per l'utente *postgres*.

```
ALTER USER postgres with encrypted password 'TUA_PASSWORD';
```

Configurata la password, modificare il file `/etc/postgresql/8.4/main/pg_hba.conf` affinché venga usata l'autenticazione *MD5* con l'utente *postgres*:

```
local   all         postgres                          md5
```

Infine, riavviare il servizio PostgreSQL per inizializzare la nuova configurazione. In un terminale, digitare quanto segue per riavviare PostgreSQL:

```
sudo /etc/init.d/postgresql-8.4 restart
```

La configurazione precedente non è completa. Per maggiori informazioni sulla configurazione di altri parametri, fare riferimento alla *guida di amministrazione di PostgreSQL*[9].

2.3. Risorse

- Come detto precedentemente, la *guida di amministrazione*[10] è un'ottima risorsa. La guida è anche disponibile nel pacchetto postgresql-doc-8.4. Per installare il pacchetto, eseguire il seguente comando in un terminale:

  ```
  sudo apt-get install postgresql-doc-8.4
  ```

  Per visualizzare la guida, inserire il seguente URI **file:///usr/share/doc/postgresql-doc-8.4/html/index.html** nella barra degli indirizzi del browser web.
- Per informazioni generali riguardo SQL, consultare *Using SQL Special Edition*[11] di Rafe Colburn.
- Per maggiori informazioni, consultare anche la *documentazione online riguardo PostgreSQL*[12].

[9] http://www.postgresql.org/docs/8.4/static/admin.html
[10] http://www.postgresql.org/docs/8.4/static/admin.html
[11] http://www.informit.com/store/product.aspx?isbn=0768664128
[12] https://help.ubuntu.com/community/PostgreSQL

Capitolo 13. Applicazioni LAMP

1. Panoramica

LAMP installations (Linux + Apache + MySQL + PHP/Perl/Python) are a popular setup for Ubuntu servers. There is a plethora of Open Source applications written using the LAMP application stack. Some popular LAMP applications are Wiki's, Content Management Systems, and Management Software such as phpMyAdmin.

One advantage of LAMP is the substantial flexibility for different database, web server, and scripting languages. Popular substitutes for MySQL include PostgreSQL and SQLite. Python, Perl, and Ruby are also frequently used instead of PHP. While Nginx, Cherokee and Lighttpd can replace Apache.

The fastest way to get started is to install LAMP using tasksel. Tasksel is a Debian/Ubuntu tool that installs multiple related packages as a co-ordinated "task" onto your system. To install a LAMP server:

- Al prompt di un terminale, eseguire il seguente comando:

```
sudo tasksel install lamp-server
```

After installing it you'll be able to install most *LAMP* applications in this way:

- Scaricare un archivio contenente il codice sorgente dell'applicazione.
- Estrarre l'archivio in una directory accessibile a un server web.
- Depending on where the source was extracted, configure a web server to serve the files.
- Configurare l'applicazione affinché si colleghi al database.
- Eseguire uno script o spostarsi su una pagina dell'applicazione per installare il database necessario all'applicazione.
- Completati i passi precedenti, o dei passi simili, è possibile utilizzare l'applicazione.

Esistono anche alcuni svantaggi con questo approccio: i file delle applicazioni non sono organizzati all'interno del file system in modo standard causando confusione sul dove è stata installata l'applicazione. Inoltre, l'aggiornamento dell'applicazione risulta essere complicato: quando viene rilasciata una nuova versione, è necessario ripetere gli stessi passi per l'installazione.

Molte applicazioni *LAMP* sono comunque disponibili all'interno dei repository di Ubuntu e si installano come tutte le altre normali applicazioni. In base però all'applicazione, potrebbe essere necessario apportare alcune configurazioni in più una volta installate.

This section covers how to install some *LAMP* applications.

2. Moin Moin

MoinMoin è un motore per wiki scritto in Python, basato sul motore «PikiPiki» e rilasciato sotto licenza GNU GPL.

2.1. Installazione

Per installare MoinMoin, eseguire il seguente comando al prompt:

```
sudo apt-get install python-moinmoin
```

È necessario installare anche il server web apache2. Per installare apache-2, consultare la sottosezione *Sezione 1.1, «Installazione» [188]* della sezione *Sezione 1, «HTTPD - Server web Apache2» [188]* .

2.2. Configurazione

Per configurare per la prima volta un wiki, chiamato per esempio *mywiki*, eseguire i seguenti comandi:

```
cd /usr/share/moin
sudo mkdir mywiki
sudo cp -R data mywiki
sudo cp -R underlay mywiki
sudo cp server/moin.cgi mywiki
sudo chown -R www-data.www-data mywiki
sudo chmod -R ug+rwX mywiki
sudo chmod -R o-rwx mywiki
```

È ora necessario configurare MoinMoin affinché identifichi il nuovo wiki *mywiki*. Per configurare MoinMoin, aprire il file `/etc/moin/mywiki.py` e modificare la riga:

```
data_dir = '/org/mywiki/data'
```

in

```
data_dir = '/usr/share/moin/mywiki/data'
```

Inoltre, al di sotto dell'opzione *data_dir*, aggiungere *data_underlay_dir*:

```
data_underlay_dir='/usr/share/moin/mywiki/underlay'
```

If the `/etc/moin/mywiki.py` file does not exists, you should copy `/usr/share/moin/config/wikifarm/mywiki.py` file to `/etc/moin/mywiki.py` file and do the above mentioned change.

Se il nome del wiki è *my_wiki_name*, è necessario inserire nel file `/etc/moin/farmconfig.py` questa riga «("my_wiki_name", r".*")» subito dopo la riga «("mywiki", r".*")».

Una volta configurato MoinMoin per trovare il wiki chiamato *mywiki*, è necessario configurare apache2 in modo che gestisca anche i wiki.

Aggiungere le seguenti righe nel file `/etc/apache2/sites-available/default` all'interno della sezione «<VirtualHost *>»

```
### moin
  ScriptAlias /mywiki "/usr/share/moin/mywiki/moin.cgi"
  alias /moin_static193 "/usr/share/moin/htdocs"
  <Directory /usr/share/moin/htdocs>
  Order allow,deny
  allow from all
  </Directory>
### end moin
```

Una volta configurato apache2, è necessario riavviarlo. Per riavviare il server web apache2, digitare:

```
sudo service apache2 restart
```

2.3. Verifica

Per verificare se l'applicazione wiki funziona, è sufficiente aprire con un browser web il seguente URL:

```
http://localhost/mywiki
```

Per ulteriori dettagli, consultare il sito web di *MoinMoin*[1].

2.4. Riferimenti

- Per maggiori informazioni, consultare il *wiki di MoinMoin*[2].
- Also, see the *Ubuntu Wiki MoinMoin*[3] page.

[1] http://moinmo.in/
[2] http://moinmo.in/
[3] https://help.ubuntu.com/community/MoinMoin

3. MediaWiki

MediaWiki è un software per wiki scritto con il linguaggio PHP ed è in grado di utilizzare database come MySQL o PostgreSQL per l'archiviazione dei dati.

3.1. Installazione

Prima di installare MediaWiki è necessario installare Apache2, il linguaggio PHP5 e un sistema di database. MySQL o PostgreSQL sono i più comuni, sceglierne uno in base alle proprie necessità. Per le istruzioni su come installarli, fare riferimento alle relative sezioni all'interno di questa guida.

Per installare MediaWiki, eseguire il seguente comando al prompt:

```
sudo apt-get install mediawiki php5-gd
```

Per maggiori informazioni sulle funzionalità di MediaWiki, consultare il pacchetto mediawiki-extensions.

3.2. Configurazione

Il file di configurazione di Apache `mediawiki.conf` per MediaWiki è installato nella directory `/etc/apache2/conf.d/`. Da questo file, per poter accedere all'applicazione MediaWiki, è utile togliere il commento alla seguente riga.

```
# Alias /mediawiki /var/lib/mediawiki
```

Una volta tolto il commento alla riga precedente, riavviare il server Apache e accedere a MediaWiki utilizzando il seguente URL:

```
http://localhost/mediawiki/config/index.php
```

Consultare la sezione «Checking environment...» presente in quella pagina. È possibile risolvere molti problemi leggendola attentamente.

Once the configuration is complete, you should copy the `LocalSettings.php` file to `/etc/mediawiki` directory:

```
sudo mv /var/lib/mediawiki/config/LocalSettings.php /etc/mediawiki/
```

You may also want to edit `/etc/mediawiki/LocalSettings.php` in order to set the memory limit (disabled by default):

```
ini_set( 'memory_limit', '64M' );
```

3.3. Estensioni

Le estensioni aggiungono nuove funzionalità a MediaWiki e forniscono agli amministratori del wiki e agli utenti l'abilità di personalizzare MediaWiki in base alle loro necessità.

È possibile scaricare estensioni per MediaWiki come un archivio o direttamente dal repository Subversion copiandolo nella directory `/var/lib/mediawiki/extensions` directory. Alla fine del file aggiungere la seguente riga: `/etc/mediawiki/LocalSettings.php`.

```
require_once "$IP/extensions/ExtentionName/ExtentionName.php";
```

3.4. Riferimenti

- Per maggiori informazioni, consultare il *sito web di MediaWiki*[4].
- La *MediaWiki Administrators' Tutorial Guide*[5] contiene molte informazioni per i nuovi amministratori di MediaWiki.
- Also, the *Ubuntu Wiki MediaWiki*[6] page is a good resource.

[4] http://www.mediawiki.org
[5] http://www.packtpub.com/Mediawiki/book
[6] https://help.ubuntu.com/community/MediaWiki

4. phpMyAdmin

phpMyAdmin è un'applicazione LAMP sviluppata appositamente per amministrare server MySQL. Scritta in PHP e accessibile attraverso un browser web, fornisce un'interfaccia grafica per svolgere attività di amministrazione su un database.

4.1. Installazione

Prima di poter installare phpMyAdmin, è necessario poter accedere al database MySQL o dallo stesso host in cui phpMyAdmin è installato o da un host accessibile via rete (per maggiori informazioni, consultare *Sezione 1, «MySQL» [207]*). Da un terminale, digitare:

```
sudo apt-get install phpmyadmin
```

Al prompt dei comandi, scegliere quale server web configurare per phpMyAdmin. Nel resto di sezione sezione viene utilizzato Apache2.

All'interno di un browser, nella barra degli indirizzi, scrivere *http://NOMESERVER/phpmyadmin*, sostituendo *NOMESERVER* con il vero nome dell'host. Alla schermata di accesso, scrivere *root* per *username* o un altro utente MySQL e digitare MySQL per la password.

Una volta effettuato l'accesso, è possibile modificare la password di *root*, creare utenti e creare ed eliminare database, tabelle, ecc...

4.2. Configurazione

I file di configurazione di phpMyAdmin sono posizionati in `/etc/phpmyadmin`. Il file principale di configurazione è `/etc/phpmyadmin/config.inc.php` e contiene le opzioni globali di phpMyAdmin.

Per utilizzare phpMyAdmin per l'amministrazione di un database MySQL presente in un altro server, modificare le seguenti opzioni nel file `/etc/phpmyadmin/config.inc.php`:

```
$cfg['Servers'][$i]['host'] = 'db_server';
```

Sostituire *db_server* con il vero nome del server in cui è presente il database remoto oppure con il suo indirizzo IP. Inoltre, assicurarsi che l'host in cui è presente phpMyAdmin possa accedere al database remoto.

Una volta configurato, terminare e ricominciare la sessione di phpMyAdmin per poter accedere al nuovo server.

I file `config.header.inc.php` e `config.footer.inc.php` vengono usati per aggiungere un'intestazione e un pedice HTML a phpMyAdmin.

Un altro importante file di configurazione è `/etc/phpmyadmin/apache.conf`, un collegamento simbolico al file `/etc/apache2/conf.d/phpmyadmin.conf`, usato per configurare Apache2 affinché

visualizzi phpMyAdmin. Nel file sono presenti le direttive per il caricamento di PHP, i permessi per la directory, ecc... Per maggiori informazioni sulla configurazione di Apache2, consultare *Sezione 1, «HTTPD - Server web Apache2» [188]*.

4.3. Riferimenti

- La documentazione di phpMyAdmin è installata automaticamente con il pacchetto ed è possibile accedervi dal collegamento *phpMyAdmin Documentation* (un punto di domanda) al di sotto del logo di phpMyAdmin. La documentazione ufficiale può anche essere visualizzata direttamente dal *sito di phpMyAdmin*[7].
- Inoltre, il libro *Mastering phpMyAdmin*[8] è un'ottima fonte per reperire ulteriori informazioni.
- A third resource is the *phpMyAdmin Ubuntu Wiki*[9] page.

[7] http://www.phpmyadmin.net/home_page/docs.php
[8] http://www.packtpub.com/phpmyadmin-3rd-edition/book
[9] https://help.ubuntu.com/community/phpMyAdmin

Capitolo 14. Server di file

Se si dispone di più di un computer su una singola rete, a un certo punto potrebbe essere necessario condividere dei file tra questi computer. In questa sezione viene spiegato come installare e configurare servizi come FTP, NFS e CUPS.

1. Server FTP

File Transfer Protocol (FTP) is a TCP protocol for downloading files between computers. In the past, it has also been used for uploading but, as that method does not use encryption, user credentials as well as data transferred in the clear and are easily intercepted. So if you are here looking for a way to upload and download files securely, see the section on OpenSSH in *Capitolo 6, Amministrazione remota [80]* instead.

FTP works on a client/server model. The server component is called an *FTP daemon*. It continuously listens for FTP requests from remote clients. When a request is received, it manages the login and sets up the connection. For the duration of the session it executes any of commands sent by the FTP client.

L'accesso a un server FTP può essere gestito in due modi:

- Anonimo
- Con autenticazione

In the Anonymous mode, remote clients can access the FTP server by using the default user account called "anonymous" or "ftp" and sending an email address as the password. In the Authenticated mode a user must have an account and a password. This latter choice is very insecure and should not be used except in special circumstances. If you are looking to transfer files securely see SFTP in the section on OpenSSH-Server. User access to the FTP server directories and files is dependent on the permissions defined for the account used at login. As a general rule, the FTP daemon will hide the root directory of the FTP server and change it to the FTP Home directory. This hides the rest of the file system from remote sessions.

1.1. vsftpd - Installazione del server FTP

vsftpd is an FTP daemon available in Ubuntu. It is easy to install, set up, and maintain. To install vsftpd you can run the following command:

```
sudo apt-get install vsftpd
```

1.2. Configurazione anonima di FTP

By default vsftpd is *not* configured to allow anonymous download. If you wish to enable anonymous download edit `/etc/vsftpd.conf` by changing:

```
anonymous_enable=Yes
```

During installation a *ftp* user is created with a home directory of `/srv/ftp`. This is the default FTP directory.

If you wish to change this location, to `/srv/files/ftp` for example, simply create a directory in another location and change the *ftp* user's home directory:

```
sudo mkdir /srv/files/ftp
sudo usermod -d /srv/files/ftp ftp
```

Applicate le modifiche, riavviare vsftpd:

```
sudo restart vsftpd
```

Finally, copy any files and directories you would like to make available through anonymous FTP to `/srv/files/ftp`, or `/srv/ftp` if you wish to use the default.

1.3. Configurazione FTP per utenti autenticati

By default vsftpd is configured to authenticate system users and allow them to download files. If you want users to be able to upload files, edit `/etc/vsftpd.conf`:

```
write_enable=YES
```

Riavviare vsftpd:

```
sudo restart vsftpd
```

Ora, quando gli utenti accedono via FTP, il loro punto di partenza sarà la propria directory *home*, dove potranno scaricare e caricare file e creare directory.

Similarly, by default, anonymous users are not allowed to upload files to FTP server. To change this setting, you should uncomment the following line, and restart vsftpd:

```
anon_upload_enable=YES
```

Abilitare il caricamento anonimo di file via FTP può compromettere la sicurezza del sistema. È sconsigliato abilitare il caricamento anonimo su server collegati direttamente a Internet.

Il file di configurazione è composto da diversi parametri di configurazione, le cui informazioni sono disponibili nel file stesso. In alternativa, è possibile fare riferimento alla pagina man (**man 5 vsftpd.conf**).

1.4. FTP sicuro

All'interno del file di configurazione `/etc/vsftpd.conf` di vsftpd, sono presenti molte opzioni per rendere il programma più sicuro. Per esempio, togliendo il commento a quanto segue, gli utenti possono essere limitati all'utilizzo solo della propria directory personale:

```
chroot_local_user=YES
```

È anche possibile limitare un particolare gruppo di utenti all'utilizzo delle sole directory personali:

```
chroot_list_enable=YES
chroot_list_file=/etc/vsftpd.chroot_list
```

Tolto il commento alle opzioni precedenti, creare un file `/etc/vsftpd.chroot_list` con l'elenco degli utenti, uno per riga, quindi riavviare vsftpd:

```
sudo restart vsftpd
```

Inoltre, il file `/etc/ftpusers` contiene un elenco di utenti a cui è *negato* l'accesso FTP. L'elenco comprende gli utenti root, daemon, nobody, ecc... Per disabilitare l'accesso FTP ad altri utenti, aggiungerli semplicemente a questo elenco.

FTP can also be encrypted using *FTPS*. Different from *SFTP*, *FTPS* is FTP over Secure Socket Layer (SSL). *SFTP* is a FTP like session over an encrypted *SSH* connection. A major difference is that users of SFTP need to have a *shell* account on the system, instead of a *nologin* shell. Providing all users with a shell may not be ideal for some environments, such as a shared web host. However, it is possible to restrict such accounts to only SFTP and disable shell interaction. See the section on OpenSSH-Server for more.

Per configurare *FTPS*, modificare il file `/etc/vsftpd.conf` aggiungendo:

```
ssl_enable=Yes
```

Inoltre, notare anche le opzioni relative al certificato e alla chiave:

```
rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
```

By default these options are set to the certificate and key provided by the ssl-cert package. In a production environment these should be replaced with a certificate and key generated for the specific host. For more information on certificates see *Sezione 5, «Certificati» [172]*.

Riavviare vsftpd e gli utenti non-anonimi utilizzeranno *FTPS*:

```
sudo restart vsftpd
```

Per consentire accesso FTP agli utenti dotati di una shell `/usr/sbin/nologin`, ma non dispongono di accesso shell, modificare il file `/etc/shells` aggiungendo *nologin*:

```
# /etc/shells: valid login shells
/bin/csh
/bin/sh
/usr/bin/es
```

```
/usr/bin/ksh
/bin/ksh
/usr/bin/rc
/usr/bin/tcsh
/bin/tcsh
/usr/bin/esh
/bin/dash
/bin/bash
/bin/rbash
/usr/bin/screen
/usr/sbin/nologin
```

Questo è necessario poiché, in modo predefinito, vsftpd utilizza PAM per l'autenticazione e i file di configurazione `/etc/pam.d/vsftpd` contiene:

```
auth    required        pam_shells.so
```

Il modulo *shells* di PAM limita l'accesso alle shell indicate nel file `/etc/shells`.

Most popular FTP clients can be configured to connect using FTPS. The lftp command line FTP client has the ability to use FTPS as well.

1.5. Riferimenti

- Per maggiori informazioni, consultare il *sito web di vsftpd*[1].
- For detailed `/etc/vsftpd.conf` options see the *vsftpd.conf man page*[2].

[1] http://vsftpd.beasts.org/vsftpd_conf.html
[2] http://manpages.ubuntu.com/manpages/precise/en/man5/vsftpd.conf.5.html

2. NFS (Network File System)

NSF permette a un sistema di condividere file e directory con altri attraverso una rete. Utilizzando NFS, utenti e programmi possono accedere ai file presenti su sistemi remoti come se fossero dei file locali.

Alcuni dei principali benefici forniti da NFS sono:

- Le workstation locali utilizzano meno spazio su disco perché i dati comuni possono essere memorizzati su una singola macchina, pur rimanendo accessibili agli altri attraverso la rete.
- Gli utenti non devono avere diverse directory home su ciascuna macchina in rete. Le directory home possono risiedere sul server NFS ed essere rese disponibili attraverso la rete.
- I dispositivi di archiviazione come dischi floppy, unità CD-ROM e USB possono essere utilizzate dagli altri computer della rete. Questo può ridurre il numero di unità per supporti rimovibili presenti nella rete.

2.1. Installazione

Per installare il server NFS, inserire il comando seguente a un prompt di terminale:

```
sudo apt-get install nfs-kernel-server
```

2.2. Configurazione

È possibile configurare le directory da esportare aggiungendole al file `/etc/exports`. Per esempio:

```
/ubuntu *(ro,sync,no_root_squash)
/home *(rw,sync,no_root_squash)
```

È possibile sostituire * con uno qualsiasi dei formati per i nomi di host. È necessario rendere la dichiarazione dei nomi di host più specifica possibile per impedire l'accesso di sistemi indesiderati ai mount NFS.

Per avviare il server NFS, è possibile eseguire il seguente comando a un prompt di terminale:

```
sudo /etc/init.d/nfs-kernel-server start
```

2.3. Configurazione client NFS

Utilizzare il comando mount per montare una directory NFS condivisa da un'altra macchina, digitando un comando simile al seguente a un prompt di terminale:

```
sudo mount esempio.nomehost.it:/ubuntu /locale/ubuntu
```

Il punto di mount `/locale/ubuntu` deve esistere. Non ci dovrebbero essere né file, né sottodirectory all'interno di `/locale/ubuntu`.

Un modo alternativo per montare una condivisione NFS da un'altra macchina consiste nell'aggiungere una riga al file `/etc/fstab`. Questa riga deve contenere il nome dell'host del server NFS, la directory esportata dal server e la directory sulla macchina locale dove montare la condivisione NFS.

La sintassi generale per la riga nel file `/etc/fstab` è come segue:

```
esempio.nomehost.it:/ubuntu /locale/ubuntu nfs rsize=8192,wsize=8192,timeo=14,intr
```

Se si hanno problemi nel montare la condivisione NFS, assicurarsi che il pacchetto nfs-common sia installato sul client. Per installare nfs-common, digitare il seguente comando al prompt del terminale:

```
sudo apt-get install nfs-common
```

2.4. Riferimenti

FAQ di NFS per Linux[3]

Documentazione online riguardo NFS[4]

[3] http://nfs.sourceforge.net/

[4] https://help.ubuntu.com/community/NFSv4Howto

3. iSCSI Initiator

iSCSI (Internet Small Computer System Interface) is a protocol that allows SCSI commands to be transmitted over a network. Typically iSCSI is implemented in a SAN (Storage Area Network) to allow servers to access a large store of hard drive space. The iSCSI protocol refers to clients as *initiators* and iSCSI servers as *targets*.

Ubuntu Server can be configured as both an iSCSI iniator and a target. This guide provides commands and configuration options to setup an iSCSI initiator. It is assumed that you already have an iSCSI target on your local network and have the appropriate rights to connect to it. The instructions for setting up a target vary greatly between hardware providers, so consult your vendor documentation to configure your specific iSCSI target.

3.1. iSCSI Initiator Install

To configure Ubuntu Server as an iSCSI initiator install the open-iscsi package. In a terminal enter:

```
sudo apt-get install open-iscsi
```

3.2. iSCSI Initiator Configuration

Once the open-iscsi package is installed, edit `/etc/iscsi/iscsid.conf` changing the following:

```
node.startup = automatic
```

You can check which targets are available by using the iscsiadm utility. Enter the following in a terminal:

```
sudo iscsiadm -m discovery -t st -p 192.168.0.10
```

- *-m:* determines the mode that iscsiadm executes in.
- *-t:* specifies the type of discovery.
- *-p:* option indicates the target IP address.

Change example *192.168.0.10* to the target IP address on your network.

If the target is available you should see output similar to the following:

```
192.168.0.10:3260,1 iqn.1992-05.com.emc:sl7b92030000520000-2
```

The *iqn* number and IP address above will vary depending on your hardware.

You should now be able to connect to the iSCSI target, and depending on your target setup you may have to enter user credentials. Login to the iSCSI node:

```
sudo iscsiadm -m node --login
```

Check to make sure that the new disk has been detected using dmesg:

```
dmesg | grep sd
```

```
[    4.322384] sd 2:0:0:0: Attached scsi generic sg1 type 0
[    4.322797] sd 2:0:0:0: [sda] 41943040 512-byte logical blocks: (21.4 GB/20.0 GiB)
[    4.322843] sd 2:0:0:0: [sda] Write Protect is off
[    4.322846] sd 2:0:0:0: [sda] Mode Sense: 03 00 00 00
[    4.322896] sd 2:0:0:0: [sda] Cache data unavailable
[    4.322899] sd 2:0:0:0: [sda] Assuming drive cache: write through
[    4.323230] sd 2:0:0:0: [sda] Cache data unavailable
[    4.323233] sd 2:0:0:0: [sda] Assuming drive cache: write through
[    4.325312]  sda: sda1 sda2 < sda5 >
[    4.325729] sd 2:0:0:0: [sda] Cache data unavailable
[    4.325732] sd 2:0:0:0: [sda] Assuming drive cache: write through
[    4.325735] sd 2:0:0:0: [sda] Attached SCSI disk
[ 2486.941805] sd 4:0:0:3: Attached scsi generic sg3 type 0
[ 2486.952093] sd 4:0:0:3: [sdb] 1126400000 512-byte logical blocks: (576 GB/537 GiB)
[ 2486.954195] sd 4:0:0:3: [sdb] Write Protect is off
[ 2486.954200] sd 4:0:0:3: [sdb] Mode Sense: 8f 00 00 08
[ 2486.954692] sd 4:0:0:3: [sdb] Write cache: disabled, read cache: enabled, doesn't
 support DPO or FUA
[ 2486.960577]  sdb: sdb1
[ 2486.964862] sd 4:0:0:3: [sdb] Attached SCSI disk
```

In the output above *sdb* is the new iSCSI disk. Remember this is just an example; the output you see on your screen will vary.

Next, create a partition, format the file system, and mount the new iSCSI disk. In a terminal enter:

```
sudo fdisk /dev/sdb
n
p
enter
w
```

The above commands are from inside the fdisk utility; see **man fdisk** for more detailed instructions. Also, the cfdisk utility is sometimes more user friendly.

Now format the file system and mount it to `/srv` as an example:

```
sudo mkfs.ext4 /dev/sdb1
sudo mount /dev/sdb1 /srv
```

Finally, add an entry to `/etc/fstab` to mount the iSCSI drive during boot:

```
/dev/sdb1       /srv        ext4    defaults,auto,_netdev 0 0
```

It is a good idea to make sure everything is working as expected by rebooting the server.

3.3. Riferimenti

Open-iSCSI Website[5]

Debian Open-iSCSI page[6]

[5] http://www.open-iscsi.org/

[6] http://wiki.debian.org/SAN/iSCSI/open-iscsi

4. CUPS - Server di stampa

Il sistema primario e i servizi di stampa di Ubuntu sono gestiti da **Common UNIX Printing System** (CUPS). Questo è un sistema di stampa liberamente disponibile e altamente portabile ed è diventato il nuovo standard per la stampa in molte distribuzioni Linux.

CUPS gestisce lavori e code di stampa, fornisce la stampa in rete tramite l'utilizzo del protocollo IPP (Internet Printing Protocol) e al tempo stesso offre supporto a una nutrita schiera di stampanti, dalle quelle a matrice di punti a quelle al laser (comprese tutte quelle nel mezzo). CUPS supporta anche il PPD (PostScript Printer Detection) e il rilevamento automatico delle stampanti di rete; inoltre fornisce un semplice strumento di amministrazione e configurazione basato sul web.

4.1. Installazione

Per installare CUPS nel proprio computer Ubuntu, basta usare sudo con il comando apt-get e fornire i pacchetti da installare come primo parametro. Un'installazione completa di CUPS ha molte dipendenze di pacchetti, ma possono essere specificati tutti nella stessa riga di comando. Digitare quello che segue al prompt del terminale per installare CUPS:

```
sudo apt-get install cups
```

Dopo essersi autenticati con la propria password utente, i pacchetti dovrebbero essere scaricati e installati. Completato questo processo, il server CUPS viene avviato automaticamente.

Per la risoluzione dei problemi, è possibile accedere alle registrazioni degli errori attraverso il file `/var/log/cups/error_log`. Se non vengono mostrate informazioni sufficienti per risolvere i problemi incontrati, è possibile incrementare la prolissità delle registrazioni del server CUPS modificando la direttiva **LogLevel** nel file di configurazione dal valore predefinito "info" a "debug" oppure "debug2", che registra tutto. Se vengono apportate ulteriori modifiche, ricordarsi di ripristinare i valori iniziali una volta risolto il problema per evitare di ritrovarsi file di registrazione di notevoli dimensioni

4.2. Configurazione

Il comportamento del server CUPS viene configurato attraverso le direttive contenute nel file `/etc/cups/cupsd.conf`. Il file di configurazione di CUPS segue la stessa sintassi del file di configurazione primario del server HTTP Apache. In questo modo, l'utente che ha familiarità con la modifica del file di configurazione di Apache si sentirà a suo agio nella modifica del file di configurazione di CUPS. Di seguito vengono presentati alcuni esempi di impostazioni che potrebbe essere opportuno cambiare fin da subito.

Prima di modificare il file di configurazione, è opportuno creare un copia del file originale e proteggerla da scrittura, in modo da avere le impostazioni originali come riferimento e per riusarle in caso di necessità.

Copiare il file `/etc/cups/cupsd.conf` e proteggerlo dalla scrittura con i seguenti comandi, inseriti a un prompt di terminale.

```
sudo cp /etc/cups/cupsd.conf /etc/cups/cupsd.conf.original
sudo chmod a-w /etc/cups/cupsd.conf.original
```

- **ServerAdmin**: per configurare l'indirizzo email dell'amministratore del server CUPS, modificare il file di configurazione `/etc/cups/cupsd.conf` con un editor di testo e aggiungere o modificare la riga *ServerAdmin*. Per esempio, se si è amministratori del server CUPS e il proprio indirizzo email è "mario@example.net", modificare la riga ServerAdmin in questo modo:

  ```
  ServerAdmin mario@example.net
  ```

- **Listen**: in modo predefinito, su Ubuntu, l'installazione del server CUPS resta in ascolto solamente sull'interfaccia di loopback all'indirizzo IP *127.0.0.1*. Per poter fare in modo che il server CUPS ascolti sull'indirizzo IP della rete, è necessario specificare un nome host, l'indirizzo IP oppure una coppia indirizzo IP/porta con l'aggiunta di una direttiva «Listen». Per esempio, se il server CUPS è all'interno di una rete locale all'indirizzo IP *192.168.10.250* e si desidera renderlo accessibile ad altri sistemi in questa sotto-rete, è necessario modificare il file `/etc/cups/cupsd.conf` e aggiungere una direttiva «Listen» in questo modo:

  ```
  Listen 127.0.0.1:631 # Listen esistente per loopback
  Listen /var/run/cups/cups.sock # socket Listen esistente
  Listen 192.168.10.250:631 # Listen sull'interfaccia LAN, porta 631 (IPP)
  ```

  Nell'esempio precedente, è possibile rendere un commento o rimuovere il riferimento all'indirizzo di loopback (127.0.0.1) se non si desidera che cupsd resti in ascolto su quell'interfaccia, ma che invece resti in ascolto solo sull'interfaccia Ethernet della LAN (Local Area Network). Per abilitare l'ascolto su tutte le interfacce di rete a cui un certo host è collegato, inclusa quella di loopback, è possibile creare una voce Listen per l'host *socrates* come segue:

  ```
  Listen socrates:631 # Listen su tutte le interfacce dell'host "socrates"
  ```

  oppure omettendo la direttiva Listen e utilizzando quella *Port*, come in:

  ```
  Port 631 # Listen sulla porta 631 di tutte le interfacce
  ```

Per ulteriori esempi di direttive di configurazione nel file di configurazione del server CUPS, consultare la pagina manuale associato inserendo il comando seguente a un prompt di terminale:

```
man cupsd.conf
```

Ogni volta che vengono apportati cambiamenti al file di configurazione `/etc/cups/cupsd.conf`, è necessario riavviare il server CUPS digitando il comando seguente a un prompt di terminale:

```
sudo /etc/init.d/cups restart
```

4.3. Interfaccia web

CUPS può essere configurato e monitorato utilizzando un'interfaccia web disponibile all'indirizzo *http://localhost:631/admin*. L'interfaccia web può anche essere usata per svolgere tutte le attività di gestione della stampante.

Per svolgere le attività di amministrazione attraverso l'interfaccia web, è necessario avere l'account root abilitato sul server o aver eseguito l'autenticazione con un utente nel gruppo *lpadmin*. Per motivi di sicurezza, CUPS non autentica gli utenti provi di password.

Per aggiungere un utente al gruppo *lpadmin*, eseguire il seguente comando in un terminale:

```
sudo usermod -aG lpadmin username
```

Maggiore documentazione è disponibile nella scheda *Documentation/Help* dell'interfaccia web.

4.4. Riferimenti

Sito Web di CUPS[7]

Debian Open-iSCSI page[8]

[7] http://www.cups.org/

[8] http://wiki.debian.org/SAN/iSCSI/open-iscsi

Capitolo 15. Servizi email

Il processo per portare una email da una persona a un'altra all'interno di una rete o attraverso internet, comporta l'utilizzo di diversi sistemi che cooperano tra loro. Ognuno di questi sistemi deve essere configurato correttamente. Colui che spedisce una email utilizza un *Mail User Agent* (MUA), o client email, per spedire il messaggio attraverso uno o più *Mail Transfer Agents* (MTA), l'ultimo dei quali lo consegnerà a un *Mail Delivery Agent* (MDA) per la consegna nella casella di posta del destinatario, che la preleverà utilizzando un client email attraverso un server POP3 o IMAP.

1. Postfix

Postfix è il Mail Transfer Agent (MTA) predefinito di Ubuntu. Cerca di essere facile da amministrare e sicuro ed è compatibile con l'MTA sendmail. Questa sezione espone come installare e configurare postfix e anche come configurare un server SMTP utilizzando un collegamento sicuro (per l'invio di email in sicurezza).

Questa guida non spiega come configurare *Virtual Domains* di Postfix. Per informazioni sui Virtual Domains e altre configurazioni avanzate, consultare *Sezione 1.7.3, «Riferimenti» [242]*.

1.1. Installazione

Per installare postfix eseguire il seguente comando:

```
sudo apt-get install postfix
```

Premere "Invio" quando il processo di installazione pone delle domande, la configurazione verrà effettuata in dettaglio al passo successivo.

1.2. Configurazione di base

Per configurare postfix, eseguire il seguente comando:

```
sudo dpkg-reconfigure postfix
```

Viene visualizzata l'interfaccia utente. In ogni schermata selezionare i seguenti valori:

- Internet Site
- mail.example.com
- steve
- mail.example.com, localhost.localdomain, localhost
- No
- 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128 192.168.0.0/24
- 0
- +
- tutti

Replace mail.example.com with the domain for which you'll accept email, 192.168.0.0/24 with the actual network and class range of your mail server, and steve with the appropriate username.

A questo punto è utile decidere quale formato usare per la mailbox. Postfix, come impostazione predefinita, utilizza **mbox** come formato. Invece di modificare il file di configurazione, è possibile usare il comando **postconf** per configurare tutti i parametri di postfix che vengono salvati nel file

`/etc/postfix/main.cf`. Per riconfigurare un particolare parametro, è sempre possibile eseguire il comando precedente o modificare il file.

Per configurare la casella di posta per **Maildir**:

```
sudo postconf -e 'home_mailbox = Maildir/'
```

Questo posizionerà le nuove mail in /home/*NOME_UTENTE*/Maildir e sarà quindi necessario configurare il proprio MDA (Mail Delivery Agent) affinché utilizzi lo stesso percorso.

1.3. Autenticazione SMTP

SMTP-AUTH consente a un client di identificarsi attraverso un meccanismo di autenticazione (SASL). TLS (Transport Layer Security) dovrebbe essere usato per cifrare il processo di autenticazione. Una volta autenticato, il server SMTP consentirà ai client di scaricare le email.

1. Configurare Postfix per SMTP-AUTH usando SASL (Dovecot SASL):

```
sudo postconf -e 'smtpd_sasl_type = dovecot'
sudo postconf -e 'smtpd_sasl_path = private/auth-client'
sudo postconf -e 'smtpd_sasl_local_domain ='
sudo postconf -e 'smtpd_sasl_security_options = noanonymous'
sudo postconf -e 'broken_sasl_auth_clients = yes'
sudo postconf -e 'smtpd_sasl_auth_enable = yes'
sudo postconf -e 'smtpd_recipient_restrictions = \
permit_sasl_authenticated,permit_mynetworks,reject_unauth_destination'
```

La configurazione *smtpd_sasl_path* è un percorso relativo alla directory di Postfix.

2. Next, generate or obtain a digital certificate for TLS. See *Sezione 5, «Certificati» [172]* for details. This example also uses a Certificate Authority (CA). For information on generating a CA certificate see *Sezione 5.5, «Autorità di Certificazione» [174]*.

MUAs connecting to your mail server via TLS will need to recognize the certificate used for TLS. This can either be done using a certificate from a commercial CA or with a self-signed certificate that users manually install/accept. For MTA to MTA TLS certficates are never validated without advance agreement from the affected organizations. For MTA to MTA TLS, unless local policy requires it, there is no reason not to use a self-signed certificate. Refer to *Sezione 5.3, «Creare un certificato auto-firmato» [174]* for more details.

3. Ottenuto un certificato, configurare Postfix affinché fornisca cifratura TLS per le mail in entrate e in uscita:

```
sudo postconf -e 'smtp_tls_security_level = may'
```

```
sudo postconf -e 'smtpd_tls_security_level = may'
sudo postconf -e 'smtp_tls_note_starttls_offer = yes'
sudo postconf -e 'smtpd_tls_key_file = /etc/ssl/private/server.key'
sudo postconf -e 'smtpd_tls_cert_file = /etc/ssl/certs/server.crt'
sudo postconf -e 'smtpd_tls_loglevel = 1'
sudo postconf -e 'smtpd_tls_received_header = yes'
sudo postconf -e 'myhostname = mail.example.com'
```

4. Se si sta usando la propria *Autorità di Certificazione* per firmare il certificato, digitare:

   ```
   sudo postconf -e 'smtpd_tls_CAfile = /etc/ssl/certs/cacert.pem'
   ```

   Again, for more details about certificates see *Sezione 5, «Certificati» [172]*.

Una volta eseguiti tutti i comandi, Postfix è configurato per SMTP-AUTH ed è stato creato un certificato auto-firmato per la cifratura TLS.

Ora, il file `/etc/postfix/main.cf` dovrebbe essere simile a *questo*[1].

La configurazione iniziale di postfix è completa, eseguire il seguente comando per riavviarne il demone:

```
sudo /etc/init.d/postfix restart
```

Postfix supports SMTP-AUTH as defined in *RFC2554*[2]. It is based on *SASL*[3]. However it is still necessary to set up SASL authentication before you can use SMTP-AUTH.

1.4. Configurare SASL

Postfix supporta due implementazioni SASL: Cyrus SASL e Dovecot SASL. Per abilitare Dovecot SASL è necessario installare il pacchetto dovecot-common. In un terminale digitare:

```
sudo apt-get install dovecot-common
```

Modificare quindi il file `/etc/dovecot/dovecot.conf`. Nella sezione *auth default* de-commentare l'opzione *socket listen* e modificare come di seguito:

```
socket listen {
  #master {
    # Master socket provides access to userdb information. It's typically
    # used to give Dovecot's local delivery agent access to userdb so it
    # can find mailbox locations.
    #path = /var/run/dovecot/auth-master
    #mode = 0600
```

[1] ../sample/postfix_configuration
[2] http://www.ietf.org/rfc/rfc2554.txt
[3] http://www.ietf.org/rfc/rfc2222.txt

```
    # Default user/group is the one who started dovecot-auth (root)
    #user =
    #group =
  #}
  client {
    # The client socket is generally safe to export to everyone. Typical use
    # is to export it to your SMTP server so it can do SMTP AUTH lookups
    # using it.
    path = /var/spool/postfix/private/auth-client
    mode = 0660
    user = postfix
    group = postfix
  }
}
```

In order to let Outlook clients use SMTP-AUTH, in the *auth default* section of /etc/dovecot/dovecot.conf add *"login"*:

```
mechanisms = plain login
```

Una volta configurato Dovecot, riavviarlo:

```
sudo /etc/init.d/dovecot restart
```

1.5. Mail-Stack Delivery

Another option for configuring Postfix for SMTP-AUTH is using the mail-stack-delivery package (previously packaged as dovecot-postfix). This package will install Dovecot and configure Postfix to use it for both SASL authentication and as a Mail Delivery Agent (MDA). The package also configures Dovecot for IMAP, IMAPS, POP3, and POP3S.

You may or may not want to run IMAP, IMAPS, POP3, or POP3S on your mail server. For example, if you are configuring your server to be a mail gateway, spam/virus filter, etc. If this is the case it may be easier to use the above commands to configure Postfix for SMTP-AUTH.

Per installare il pacchetto, in un terminale digitare:

```
sudo apt-get install mail-stack-delivery
```

Il server mail dovrebbe essere funzionante, anche se è possibile modificarne ulteriormente la configurazione. Il pacchetto, per esempio, utilizza il certificato e la chiave presi dal pacchetto ssl-cert, ma con un server in produzione dovrebbero essere usati un certificato e una chiave generati appositamente per l'host. Per maggiori informazioni, consultare *Sezione 5, «Certificati» [172]*.

Ottenuto un certificato personalizzato e una chiave per l'host, modificare le seguenti opzioni nel file `/etc/postfix/main.cf`:

```
smtpd_tls_cert_file = /etc/ssl/certs/ssl-mail.pem
smtpd_tls_key_file = /etc/ssl/private/ssl-mail.key
```

Riavviare Postfix:

```
sudo /etc/init.d/postfix restart
```

1.6. Test

La configurazione di SMTP-AUTH è completa ed è ora necessario provarla.

Per verificare se SMTP-AUTH e TLS funzionano correttamente, eseguire il seguente comando:

```
telnet mail.example.com 25
```

Una volta stabilita la connessione al server mail Postfix, digitare:

```
ehlo mail.example.com
```

Se, tra tutte le righe, viene visualizzato anche questo, allora funziona correttamente. Digitare **quit** per uscire.

```
250-STARTTLS
250-AUTH LOGIN PLAIN
250-AUTH=LOGIN PLAIN
250 8BITMIME
```

1.7. Risoluzione problemi

Questa sezione descrive alcuni metodi comuni per determinare la cause dei problemi che potrebbero verificarsi.

1.7.1. Evitare l'uso di chroot

Il pacchetto postfix di Ubuntu viene installato, in modo predefinito e per ragioni di sicurezza, all'interno di un ambiente *chroot*.

Per terminate l'operazione chroot, localizzare la seguente riga nel file `/etc/postfix/master.cf`:

```
smtp      inet  n       -       -       -       -       smtpd
```

e modicarlo come segue:

```
smtp      inet  n       -       n       -       -       smtpd
```

È necessario riavviare Postfix affinché utilizzi la nuova configurazione. In un terminale, digitare:

```
sudo /etc/init.d/postfix restart
```

1.7.2. File di registro

Postfix invia tutti i messaggi di registrazione in `/var/log/mail.log`. I messaggi di errore e gli avvisi possono andar persi nell'output della registrazione normale, per questo vengono anche registrati in `/var/log/mail.err` e `/var/log/mail.warn` rispettivamente.

Per visualizzare in tempo reale i messaggi che vengono registrati, usare il comando tail -f:

```
tail -f /var/log/mail.err
```

Il livello di dettaglio delle registrazioni può essere incrementato. Di seguito vengono riportate alcune opzioni di configurazione per aumentare i dettagli di registrazione in alcune delle aree descritte precedentemente.

- Per aumentare la registrazione delle attività *TLS*, impostare l'opzione *smtpd_tls_loglevel* a un valore compreso tra 1 e 4.

  ```
  sudo postconf -e 'smtpd_tls_loglevel = 4'
  ```
- Se si riscontrano problemi nell'inviare o nel ricevere email da uno specifico dominio, è possibile aggiungere tale dominio al parametro *debug_peer_list*.

  ```
  sudo postconf -e 'debug_peer_list = problem.domain'
  ```
- È possibile incrementare il livello di registrazione di qualsiasi demone Postfix modificando il file `/etc/postfix/master.cf` e aggiungendo *-v* subito dopo la voce. Per esempio, modificare la voce *smtp*:

  ```
  smtp unix  - - - - - smtp -v
  ```

  

  È importante notare che dopo aver apportato una delle modifiche alla registrazione, il processo Postfix deve essere riavviato affinché riconosca la nuova configurazione: **sudo /etc/init.d/postfix reload**
- Per incrementare il livello di informazioni registrate durante la risoluzione di problemi con *SASL*, è possibile impostare le seguenti opzioni nel file `/etc/dovecot/dovecot.conf`

  ```
  auth_debug=yes
  auth_debug_passwords=yes
  ```

  

  Proprio come Postfix, modificando la configurazione di Dovecot il processo deve essere ricaricato: **sudo /etc/init.d/dovecot reload**.

  

  Alcune delle opzioni precedenti possono aumentare drasticamente la quantità di informazioni inviata ai file di registrazione. Ricordarsi di ripristinare il livello di

registrazione al valore predefinito dopo aver corretto il problema, quindi ricaricare il demone appropriato affinché la configurazione abbia effetto.

1.7.3. Riferimenti

Amministrare un server Postfix può essere un compito molto complicato e potrebbe essere necessario richiedere aiuto alla comunità.

Un ottimo punto per richiedere assistenza riguardo Postfix, e per partecipare nella comunità di Ubuntu Server, è il canale IRC *#ubuntu-server* su *freenode*[4]. È anche possibile lasciare un messaggio in uno dei tanti *forum*[5].

Per informazioni dettagliate riguardo Postfix, gli sviluppatori Ubuntu consigliano il libro *The Book of Postfix*[6].

In fine, il *sito web di Postfix*[7] dispone di ottima documentazione riguardo le diverse opzioni di configurazione.

Also, the *Ubuntu Wiki Postifx*[8] page has more information.

[4] http://freenode.net
[5] http://www.ubuntu.com/support/community/webforums
[6] http://www.postfix-book.com/
[7] http://www.postfix.org/documentation.html
[8] https://help.ubuntu.com/community/Postfix

2. Exim4

Exim4 è un MTA (Message Transfer Agent) sviluppato dalla "University of Cambridge" per essere usato sui sistemi Unix collegati a Internet. Exim può essere installato al posto di sendmail, anche se la configurazione di exim è diversa da quella di sendmail.

2.1. Installazione

Per installare exim4, eseguire il seguente comando:

```
sudo apt-get install exim4
```

2.2. Configurazione

Per configurare Exim4, eseguire il seguente comando:

```
sudo dpkg-reconfigure exim4-config
```

Viene visualizzata l'interfaccia che consente di configurare molti dei parametri. Per esempio, in Exim4 i file di configurazione sono divisi in molti piccoli file, per averli tutti raggruppati in un unico file, è possibile farlo attraverso questa interfaccia.

All the parameters you configure in the user interface are stored in `/etc/exim4/update-exim4.conf` file. If you wish to re-configure, either you re-run the configuration wizard or manually edit this file using your favorite editor. Once you configure, you can run the following command to generate the master configuration file:

```
sudo update-exim4.conf
```

Il file di configurazione principale è generato e archiviato in `/var/lib/exim4/config.autogenerated`.

Per nessun motivo modificare il file `/var/lib/exim4/config.autogenerated`. È aggiornato automaticamente ogni volta che viene eseguito il comando **update-exim4.conf**

Per avviare il demone Exim4, eseguire il seguente comando:

```
sudo /etc/init.d/exim4 start
```

2.3. Autenticazione SMTP

Questa sezione descrive come configurare Exim4 affinché usi SMTP-AUTH con TLS e SASL.

Il primo passo è quello di creare un certificato da usare con TLS. In un terminale, digitare quanto segue:

```
sudo /usr/share/doc/exim4-base/examples/exim-gencert
```

Ora è necessario configurare Exim4 per l'utilizzo di TLS modificando il file `/etc/exim4/conf.d/main/03_exim4-config_tlsoptions` e aggiungendo quanto segue:

```
MAIN_TLS_ENABLE = yes
```

È ora necessario configurare Exim4 affinché utilizzi saslauthd per l'autenticazione. Modificare il file `/etc/exim4/conf.d/auth/30_exim4-config_examples` e de-commentare le sezioni *plain_saslauthd_server* e *login_saslauthd_server*:

```
 plain_saslauthd_server:
   driver = plaintext
   public_name = PLAIN
   server_condition = ${if saslauthd{{$auth2}{$auth3}}{1}{0}}
   server_set_id = $auth2
   server_prompts = :
   .ifndef AUTH_SERVER_ALLOW_NOTLS_PASSWORDS
   server_advertise_condition = ${if eq{$tls_cipher}{}{}{*}}
   .endif
#
 login_saslauthd_server:
   driver = plaintext
   public_name = LOGIN
   server_prompts = "Username:: : Password::"
   # don't send system passwords over unencrypted connections
   server_condition = ${if saslauthd{{$auth1}{$auth2}}{1}{0}}
   server_set_id = $auth1
   .ifndef AUTH_SERVER_ALLOW_NOTLS_PASSWORDS
   server_advertise_condition = ${if eq{$tls_cipher}{}{}{*}}
   .endif
```

Additionally, in order for outside mail client to be able to connect to new exim server, new user needs to be added into exim by using the following commands.

```
sudo /usr/share/doc/exim4/examples/exim-adduser
```

Users should protect the new exim password files with the following commands.

```
sudo chown root:Debian-exim /etc/exim4/passwd
sudo chmod 640 /etc/exim4/passwd
```

Infine, aggiornare la configurazione di Exim4 e riavviare il servizio:

```
sudo update-exim4.conf
sudo /etc/init.d/exim4 restart
```

2.4. Configurare SASL

Questa sezione descrive come configurare saslauthd per fornire l'autenticazione per Exim4.

Per prima cosa è necessario installare il pacchetto sasl2-bin. In un terminale, digitare quando segue:

```
sudo apt-get install sasl2-bin
```

Per configurare saslauthd, modificare il file "/etc/default/saslauthd" e impostare START=no a:

```
START=yes
```

Affinché Exim4 possa usare il servizio saslauth, l'utente *Debian-exim* deve far parte del gruppo *sasl*:

```
sudo adduser Debian-exim sasl
```

Ora avviare il servizio saslauthd:

```
sudo /etc/init.d/saslauthd start
```

Exim4 è ora configurato con il supporto a SMTP-AUTH con l'uso dell'autenticazione TLS e SASL.

2.5. Riferimenti

- Per maggiori informazioni, consultare *exim.org*[9].
- È anche disponibile un *libro su Exim4*[10].
- Another resource is the *Exim4 Ubuntu Wiki* [11] page.

[9] http://www.exim.org/
[10] http://www.uit.co.uk/content/exim-smtp-mail-server
[11] https://help.ubuntu.com/community/Exim4

3. Server Dovecot

Dovecot è un Mail Delivery Agent progettato per garantire la sicurezza. Supporta la maggior parte dei formati di caselle di posta: mbox o maildir. Questa sezione espone come configurarlo come server imap o pop3.

3.1. Installazione

Per installare dovecot, in un terminale, digitare:

```
sudo apt-get install dovecot-imapd dovecot-pop3d
```

3.2. Configurazione

Per configurare dovecot è possibile modificare il file `/etc/dovecot/dovecot.conf`. È possibile scegliere il protocollo da usare, che può essere pop3, pop3s (pop3 sicuro), imap and imaps (imap sicuro). Una descrizione di questi protocolli va oltre lo scopo di questa guida. Per maggiori informazioni, fare riferimento agli articoli su Wikipedia relativi a *POP3*[12] e *IMAP*[13].

IMAPS e POP3S sono più sicuri dei semplici IMAP e POP3 poiché utilizzano la cifratura SSL per connettersi. Una volta scelto il protocollo, modificare la seguente riga nel file `/etc/dovecot/dovecot.conf`:

```
protocols = pop3 pop3s imap imaps
```

Quindi, scegliere la mailbox che si desidera usare. Dovecot supporta i formati **maildir** e **mbox**, che sono i formati di mailbox più comunemente usati. Entrambi hanno i propri vantaggi, discussi sul *sito web di Dovecot*[14].

Una volta scelta la tipologia della casella di posta, modificare il file `/etc/dovecot/dovecot.conf` e cambiare la seguente riga:

```
mail_location = maildir:~/Maildir # (per maildir)
oppure
mail_location = mbox:~/mail:INBOX=/var/spool/mail/%u # (per mbox)
```

È necessario configurare l'MTA (Mail Transport Agent) per trasferire le mail ricevute in questo tipo di casella di posta se è differente da quella impostata.

Una volta configurato, riavviare il demone dovecot per provare le impostazioni:

[12] http://en.wikipedia.org/wiki/POP3
[13] http://en.wikipedia.org/wiki/Internet_Message_Access_Protocol
[14] http://wiki.dovecot.org/MailboxFormat

```
sudo /etc/init.d/dovecot restart
```

Se è stato abilitato imap o pop3, è possibile provare a eseguire l'accesso con i comandi **telnet localhost pop3** o **telnet localhost imap2**. Se viene visualizzata una schermata simile alla seguente, l'installazione è stata eseguita con successo:

```
telnet localhost pop3
Trying 127.0.0.1...
Connected to localhost.localdomain.
Escape character is '^]'.
+OK Dovecot ready.
```

3.3. Configurazione di Dovecot SSL

Per configurare dovecot affinché utilizzi SSL, è possibile modificare il file `/etc/dovecot/dovecot.conf` e cambiare le seguenti righe:

```
ssl_cert_file = /etc/ssl/certs/ssl-cert-snakeoil.pem
ssl_key_file = /etc/ssl/private/ssl-cert-snakeoil.key
ssl_disable = no
disable_plaintext_auth = no
```

È possibile ottenere il certificato SSL da un'entità di certificazione oppure è possibile creare il proprio certificato auto-firmato. Quest'ultima opzione è valida per le email, dato che i client SMPT solitamente non danno grossi problemi a riguardo. Per maggiori informazioni su come creare un certificato SSL, consultare *Sezione 5, «Certificati» [172]*. Una volta creato, sono disponibili una chiave e un certificato sotto forma di file, copiarli nella posizione puntata all'interno del file `/etc/dovecot/dovecot.conf`.

3.4. Configurazione del firewall per un server email

Per accedere al server mail da un altro computer, è necessario configurare il firewall affinché consenta i collegamenti al server sulle porte necessarie.

- IMAP - 143
- IMAPS - 993
- POP3 - 110
- POP3S - 995

3.5. Riferimenti

- Per maggiori informazioni, consultare il *sito web di Dovecot*[15].
- Also, the *Dovecot Ubuntu Wiki*[16] page has more details.

[15] http://www.dovecot.org/

[16] https://help.ubuntu.com/community/Dovecot

4. Mailman

Mailman è un programma open source per la gestione di discussioni elettroniche e newsletter. Molte mailing list open source (incluse tutte le mailing list di *Ubuntu*[17]) utilizzano Mailman come software. È molto potente e facile da installare.

4.1. Installazione

Mailman dispone in un'interfaccia web sia per gli amministratori che per gli utenti, utilizzando un server mail esterno per inviare e ricevere le email e si integra perfettamente con i seguenti server mail:

- Postfix
- Exim
- Sendmail
- Qmail

Viene descritto come installare Mailman, il server web Apache e il server mail Postfix o Exim. Per installare Mailman con un server mail diverso, fare riferimento alla sezione «Riferimenti».

È necessario installare solamente un server mail e Postfix è il Mail Transfer Agent predefinito di Ubuntu.

4.1.1. Apache2

To install apache2 you refer to *Sezione 1.1, «Installazione» [188]* for details.

4.1.2. Postfix

Per le istruzioni su come installare e configurare Postfix, consultare *Sezione 1, «Postfix» [236]*

4.1.3. Exim4

Per installare Exim4, consultare refer to *Sezione 2, «Exim4» [243]*.

Once exim4 is installed, the configuration files are stored in the `/etc/exim4` directory. In Ubuntu, by default, the exim4 configuration files are split across different files. You can change this behavior by changing the following variable in the `/etc/exim4/update-exim4.conf` file:

```
dc_use_split_config='true'
```

4.1.4. Mailman

Per installare Mailman, in un terminale, digitare il seguente comando:

[17] http://lists.ubuntu.com

```
sudo apt-get install mailman
```

Questo copia i file di installazione nella directory /var/lib/mailman, gli script CGI nella directory /usr/lib/cgi-bin/mailman, crea l'utente *list* e il gruppo *list*. Il proprietario del processo mailman sarà l'utente creato.

4.2. Configurazione

Questa sezione ha come presupposto l'avvenuta installazione di mailman, apache2 e di postfix o exim4. Ora è solo necessario configurarle.

4.2.1. Apache2

Un file di esempio di Apache è disponibile con Mailman ed è localizzato in `/etc/mailman/apache.conf`. Affinché Apache possa utilizzare il file di configurazione è necessario copiarlo in `/etc/apache2/sites-available`:

```
sudo cp /etc/mailman/apache.conf /etc/apache2/sites-available/mailman.conf
```

In questo modo verrà configurato un nuovo *VirtualHost* per il sito di amministrazione di Mailman. Ora è necessario abilitare la configurazione e riavviare Apache:

```
sudo a2ensite mailman.conf
sudo service apache2 restart
```

Mailman utilizza apache2 per eseguire gli script CGI. Gli script CGI di mailman sono installati all'interno della directory /usr/lib/cgi-bin/mailman e l'URL di mailman risulta quindi "http://hostname/cgi-bin/mailman/". È possibile apportare cambiamenti al file `/etc/apache2/sites-available/mailman.conf` per modificarne il comportamento.

4.2.2. Postfix

Per l'integrazione di Postfix, verrà associato il dominio "lists.example.com" con le seguenti mailing list. Sostituire *lists.example.com* con il proprio dominio.

È possibile usare il comando postconf per aggiungere la configurazione necessaria in `/etc/postfix/main.cf`:

```
sudo postconf -e 'relay_domains = lists.example.com'
sudo postconf -e 'transport_maps = hash:/etc/postfix/transport'
sudo postconf -e 'mailman_destination_recipient_limit = 1'
```

Controllare che in `/etc/postfix/master.cf` sia presente quanto segue:

```
mailman   unix  -       n       n       -       -       pipe
  flags=FR user=list argv=/usr/lib/mailman/bin/postfix-to-mailman.py
```

```
  ${nexthop} ${user}
```

Invoca lo script *postfix-to-mailman.py* quando viene ricevuta una mail in una lista.

Associare il dominio "lists.example.com" a Mailman con la mappa dei metodi "transport". Modificare il file `/etc/postfix/transport`:

```
lists.example.com      mailman:
```

Ora è necessario far generare a Postfix la mappa "transport" digitando, in un terminale:

```
sudo postmap -v /etc/postfix/transport
```

Riavviare Postfix per abilitare le nuove configurazioni:

```
sudo /etc/init.d/postfix restart
```

4.2.3. Exim4

Una volta installato Exim4, è possibile avviare il server Exim digitando, in un terminale, il seguente comando:

```
sudo /etc/init.d/exim4 start
```

Affinché mailman funzioni con Exim4, è necessario configurare Exim4. Come già spiegato, Exim4 utilizza molteplici file di configurazione di diverse tipologia (per maggiori informazioni, fare riferimento al *sito web di Exim*[18]). Per poter eseguire mailman, è necessario aggiungere un nuovo file di configurazione alle seguenti tipologie di configurazione:

- Main
- Transport
- Router

Exim quindi crea un file di configurazione principale ordinando tutti i file di configurazione: l'ordine di questi file è molto importante.

4.2.4. Main

Tutti i file di configurazione appartenenti al tipo main sono archiviati nella directory `/etc/exim4/conf.d/main/`. È possibile aggiungere il seguente contenuto a un nuovo file di configurazione chiamato `04_exim4-config_mailman`:

```
# start
# Home dir for your Mailman installation -- aka Mailman's prefix
# directory.
# On Ubuntu this should be "/var/lib/mailman"
```

[18] http://www.exim.org

```
# This is normally the same as ~mailman
MM_HOME=/var/lib/mailman
#
# User and group for Mailman, should match your --with-mail-gid
# switch to Mailman's configure script. Value is normally "mailman"
MM_UID=list
MM_GID=list
#
# Domains that your lists are in - colon separated list
# you may wish to add these into local_domains as well
domainlist mm_domains=hostname.com
#
# -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
#
# These values are derived from the ones above and should not need
# editing unless you have munged your mailman installation
#
# The path of the Mailman mail wrapper script
MM_WRAP=MM_HOME/mail/mailman
#
# The path of the list config file (used as a required file when
# verifying list addresses)
MM_LISTCHK=MM_HOME/lists/${lc::$local_part}/config.pck
# end
```

4.2.5. Transport

Tutti i file di configurazione appartenenti al tipo transport sono archiviati nella directory `/etc/exim4/conf.d/transport/`. È possibile aggiungere il seguente contenuto a un nuovo file di configurazione chiamato `40_exim4-config_mailman`:

```
  mailman_transport:
   driver = pipe
   command = MM_WRAP \
               '${if def:local_part_suffix \
                    {${sg{$local_part_suffix}{-(\\w+)(\\+.*)?}{\$1}}} \
                    {post}}' \
               $local_part
    current_directory = MM_HOME
    home_directory = MM_HOME
    user = MM_UID
    group = MM_GID
```

4.2.6. Router

Tutti i file di configurazione appartenenti al tipo router sono archiviati nella directory `/etc/exim4/conf.d/router/`. È possibile aggiungere il seguente contenuto a un nuovo file di configurazione chiamato `101_exim4-config_mailman`:

```
  mailman_router:
```

```
driver = accept
require_files = MM_HOME/lists/$local_part/config.pck
local_part_suffix_optional
local_part_suffix = -bounces : -bounces+* : \
                    -confirm+* : -join : -leave : \
                    -owner : -request : -admin
transport = mailman_transport
```

L'ordine dei file di configurazione main e transport può essere qualsiasi. L'ordine dei file di configurazione del tipo router deve essere lo stesso. Questo particolare file deve apparire prima del file `200_exim4-config_primary`. Questi file contengono le stesse informazioni, ma il primo ha la precedenza. Per maggiori informazioni fare riferimento alla sezione «Riferimenti».

4.2.7. Mailman

Una volta installato mailman, è possibile avviarlo usando il seguente comando:

```
sudo /etc/init.d/mailman start
```

Creare quindi la mailing list predefinita. Per crearla, eseguire il seguente comando:

```
sudo /usr/sbin/newlist mailman
```

```
Enter the email address of the person running the list: bhuvan at ubuntu.com
Initial mailman password:
To finish creating your mailing list, you must edit your /etc/aliases (or
equivalent) file by adding the following lines, and possibly running the
`newaliases' program:

## mailman mailing list
mailman: "|/var/lib/mailman/mail/mailman post mailman"
mailman-admin: "|/var/lib/mailman/mail/mailman admin mailman"
mailman-bounces: "|/var/lib/mailman/mail/mailman bounces mailman"
mailman-confirm: "|/var/lib/mailman/mail/mailman confirm mailman"
mailman-join: "|/var/lib/mailman/mail/mailman join mailman"
mailman-leave: "|/var/lib/mailman/mail/mailman leave mailman"
mailman-owner: "|/var/lib/mailman/mail/mailman owner mailman"
mailman-request: "|/var/lib/mailman/mail/mailman request mailman"
mailman-subscribe: "|/var/lib/mailman/mail/mailman subscribe mailman"
mailman-unsubscribe: "|/var/lib/mailman/mail/mailman unsubscribe mailman"

Hit enter to notify mailman owner..

#
```

Postfix o Exim4 sono stati configurati per riconoscere tutte le email di mailman ed è ora obbligatorio creare le nuove voci in `etc/aliases`. Se sono state apportate modifiche ai file di configurazione, assicurarsi di riavviare tali servizi prima di continuare.

Exim4 non utilizza gli alias precedenti per inoltrare le mail a Mailman dato che usa un approccio di tipo *discover*. Per eliminare gli alias quando viene creato l'elenco, è possibile aggiungere la riga *MTA=None* nel file di configurazione di Mailman `/etc/mailman/mm_cfg.py`.

4.3. Amministrazione

Si assume che sia stata fatta un'installazione di base. Gli script cgi di mailman si trovano nella directory /usr/lib/cgi-bin/mailman/. Mailman fornisce uno strumento di amministrazione basato sul web, per accedere alla relativa pagina, aprire con il browser il seguente url:

http://hostname/cgi-bin/mailman/admin

La mailing list di base, *mailman*, comparirà in questa schermata. Facendo clic sul nome della mailing list, verrà richiesta la password di autenticazione. Se viene inserita la password corretta sarà possibile modificare le preferenze di amministrazione di questa mailing list. È possibile creare una nuova mailing list usando l'utilità a riga di comando (**/usr/sbin/newlist**). In alternativa, è possibile creare una nuova mailing list usando l'interfaccia web.

4.4. Utenti

Mailman fornisce un'interfaccia web per gli utenti. Per accedere a questa pagina, indirizzare il browser web al seguente URL:

http://hostname/cgi-bin/mailman/listinfo

La mailing list predefinita, *mailman*, compare a schermo. Facendo clic sul nome, viene presentato il modulo di iscrizione. È possibile inserire il proprio indirizzo email, il nome (opzionale) e la password per completare l'iscrizione. Viene così inviata una email di invito all'indirizzo specificato. È possibile seguire le istruzioni contenute nell'email per completare l'iscrizione.

4.5. Riferimenti

GNU Mailman - Manuale di installazione[19]

HOWTO - Using Exim 4 and Mailman 2.1 together[20]

Also, see the *Mailman Ubuntu Wiki*[21] page.

[19] http://www.list.org/mailman-install/index.html
[20] http://www.exim.org/howto/mailman21.html
[21] https://help.ubuntu.com/community/Mailman

5. Filtrare le email

Uno dei più grandi problemi oggigiorno con le email è lo Unsolicited Bulk Email (UBE). Conosciuto anche come SPAM, questi messaggi possono essere virus e altre forme di malware. Secondi alcuni rapporti, questi messaggi compongono la maggior parte del traffico di email su Internet.

This section will cover integrating Amavisd-new, Spamassassin, and ClamAV with the Postfix Mail Transport Agent (MTA). Postfix can also check email validity by passing it through external content filters. These filters can sometimes determine if a message is spam without needing to process it with more resource intensive applications. Two common filters are opendkim and python-policyd-spf.

- Amavisd-new è un "wrapper" che può chiamare qualsiasi programma di filtraggio per rilevare la posta indesiderata, virus, ecc...
- Spamassassin utilizza moti meccanismi diversi per filtrare le email in base al contenuto del messaggio.
- ClamAV è un antivirus open source.
- opendkim implements a Sendmail Mail Filter (Milter) for the DomainKeys Identified Mail (DKIM) standard.
- python-policyd-spf abilita il controllo Sender Policy Framework (SPF) con Postfix.

Il processo di elaborazione è il seguente:

- Un messaggio email viene accettato da Postfix.
- The message is passed through any external filters opendkim and python-policyd-spf in this case.
- Amavisd-new quindi elabora il messaggio.
- ClamAV analizza il messaggio. Se contiene un virus, Postfix rifiuta il messaggio.
- I messaggi puliti vengono poi analizzati da Spamassassin per verificare che non sia indesiderato. Spamassassin aggiunge quindi una riga X-Header per consentire ad Amavisd-new di analizzare ulteriormente il messaggio.

Per esempio, se un messaggio ha un punteggio spam di oltre 50, questo può essere scartato automaticamente senza nemmeno farlo arrivare al ricevente. Un altro metodo per gestire i messaggi con una segnalazione, è quello di lasciarli arrivare al Mail User Agent (MUA) consentendo all'utente di gestirli come meglio crede.

5.1. Installazione

Per maggiori informazioni sull'installazione e la configurazione di Postfix, consultare *Sezione 1, «Postfix» [236]*.

Per installare le restanti applicazioni, in un terminale, digitare:

```
sudo apt-get install amavisd-new spamassassin clamav-daemon
sudo apt-get install opendkim postfix-policyd-spf-python
```

Esistono dei pacchetti opzionali che si integrano con Spamassassin per rilevare più efficientemente la posta indesiderata:

```
sudo apt-get install pyzor razor
```

Oltre alle applicazioni per il filtraggio, sono necessarie le utilità di compressioni per elaborare alcuni allegati delle email.

```
sudo apt-get install arj cabextract cpio lha nomarch pax rar unrar unzip zip
```

If some packages are not found, check that the *multiverse* repository is enabled in `/etc/apt/sources.list`

If you make changes to the file, be sure to run **sudo apt-get update** before trying to install again.

5.2. Configurazione

Ora è necessario configurare il tutto affinché i programmi funzionino assieme e vengano filtrate le email.

5.2.1. ClamAV

Il comportamento predefinito di ClamAV soddisferà le proprie necessità. Per le altre opzioni di ClamAV, controllare i file di configurazioni presenti in `/etc/clamav`.

Aggiungere l'utente *clamav* al gruppo *amavis* affinché Amavisd-new possa avere accesso per analizzare i file:

```
sudo adduser clamav amavis
sudo adduser amavis clamav
```

5.2.2. Spamassassin

Spamassassin rileva automaticamente i componenti opzionali e ne fa uso se sono presenti. Ciò significa che non c'è alcuna necessità di configurare pyzor e razor.

Modificare `/etc/default/spamassassin` per attivare il demone Spamassassin daemon. Cambiare *ENABLED=0* in:

```
ENABLED=1
```

Ora avviare il demone:

```
sudo /etc/init.d/spamassassin start
```

5.2.3. Amavisd-new

Per prima cosa, attivare il rilevamento spam e antivirus in Amavisd-new modificando `/etc/amavis/conf.d/15-content_filter_mode`:

```
use strict;

# You can modify this file to re-enable SPAM checking through spamassassin
# and to re-enable antivirus checking.

#
# Default antivirus checking mode
# Uncomment the two lines below to enable it
#

@bypass_virus_checks_maps = (
   \%bypass_virus_checks, \@bypass_virus_checks_acl, \$bypass_virus_checks_re);


#
# Default SPAM checking mode
# Uncomment the two lines below to enable it
#

@bypass_spam_checks_maps = (
   \%bypass_spam_checks, \@bypass_spam_checks_acl, \$bypass_spam_checks_re);

1;  # insure a defined return
```

rifiutare lo spam e rinviarlo al mittente può essere una cattiva idea, dato che l'indirizzo solitamente è fasullo. Modificare quindi `/etc/amavis/conf.d/20-debian_defaults` per impostare *$final_spam_destiny* a "D_DISCARD" piuttosto che "D_BOUNCE":

```
$final_spam_destiny       = D_DISCARD;
```

Per indicare più messaggi come indesiderati, è possibile utilizzare anche questa opzione:

```
$sa_tag_level_deflt = -999; # add spam info headers if at, or above that level
$sa_tag2_level_deflt = 6.0; # add 'spam detected' headers at that level
$sa_kill_level_deflt = 21.0; # triggers spam evasive actions
$sa_dsn_cutoff_level = 4; # spam level beyond which a DSN is not sent
```

Se il *nome host* del server è diverso dal record MX del dominio è necessario impostare manualmente l'opzione *$myhostname*. Inoltre, se il server riceve email da diversi domini, è necessario personalizzare l'opzione *@local_domains_acl*. Modificare il file `/etc/amavis/conf.d/50-user`:

```
$myhostname = 'mail.example.com';
@local_domains_acl = ( "example.com", "example.org" );
```

If you want to cover multiple domains you can use the following in the`/etc/amavis/conf.d/50-user`

```
@local_domains_acl = qw(.);
```

Una volta configurato, Amavisd-new deve essere riavviato:

```
sudo /etc/init.d/amavis restart
```

5.2.3.1. Whitelist DKIM

Amavisd-new può essere configurato per inserire automaticamente in una *whitelist* gli indirizzi da domini dotati di "Domain Keys" valide. Nel file `/etc/amavis/conf.d/40-policy_banks` sono disponibili alcuni domini preconfigurati.

L'aggiunta di un dominio nella whitelist è possibile in diversi modi:

- *'example.com' => 'WHITELIST',*: inserisce nella whitelist qualsiasi indirizzo dal dominio "example.com".
- *'.example.com' => 'WHITELIST',*: inserisce nella whitelist qualsiasi indirizzo da qualsiasi *sotto dominio* di "example.com" con una firma valida.
- *'.example.com/@example.com' => 'WHITELIST',*: inserisce nella whitelist i sotto domini di "example.com" che utilizzano una firma del dominio superiore *example.com.*
- *'./@example.com' => 'WHITELIST',*: adds addresses that have a valid signature from "example.com". This is usually used for discussion groups that sign their messages.

A domain can also have multiple Whitelist configurations. After editing the file, restart amavisd-new:

```
sudo /etc/init.d/amavis restart
```

In questo contesto, una volta aggiunto un dominio alla whitelist, il messaggio non verrà più filtrato dall'anti-virus o dal filtro anti-spam. Questo potrebbe essere o meno un comportamento indesiderato per un dominio.

5.2.4. Postfix

Per l'integrazione con Postfix, in un terminale, digitare quanto segue:

```
sudo postconf -e 'content_filter = smtp-amavis:[127.0.0.1]:10024'
```

Ora modificare il file `/etc/postfix/master.cf` e aggiungere quanto segue alla fine:

```
smtp-amavis     unix    -       -       -       -       2       smtp
        -o smtp_data_done_timeout=1200
        -o smtp_send_xforward_command=yes
        -o disable_dns_lookups=yes
```

```
        -o max_use=20

127.0.0.1:10025 inet    n       -       -       -       -       smtpd
        -o content_filter=
        -o local_recipient_maps=
        -o relay_recipient_maps=
        -o smtpd_restriction_classes=
        -o smtpd_delay_reject=no
        -o smtpd_client_restrictions=permit_mynetworks,reject
        -o smtpd_helo_restrictions=
        -o smtpd_sender_restrictions=
        -o smtpd_recipient_restrictions=permit_mynetworks,reject
        -o smtpd_data_restrictions=reject_unauth_pipelining
        -o smtpd_end_of_data_restrictions=
        -o mynetworks=127.0.0.0/8
        -o smtpd_error_sleep_time=0
        -o smtpd_soft_error_limit=1001
        -o smtpd_hard_error_limit=1000
        -o smtpd_client_connection_count_limit=0
        -o smtpd_client_connection_rate_limit=0
        -o receive_override_options=no_header_body_checks,no_unknown_recipient_checks
```

Aggiungere anche le seguenti righe dopo il servizio di trasporto "*pickup*":

```
         -o content_filter=
         -o receive_override_options=no_header_body_checks
```

In questo modo si eviteranno i messaggi generati per segnalare lo spam che viene classificato come spam.

Infine riavviare Postfix:

```
sudo /etc/init.d/postfix restart
```

Il filtraggio sul contenuto per lo spam e il rilevamento di virus sono ora abilitati.

5.2.5. Amavisd-new and Spamassassin

When integrating Amavisd-new with Spamassassin, if you choose to disable the bayes filtering by editing `/etc/spamassassin/local.cf` and use cron to update the nightly rules, the result can cause a situation where a large amount of error messages are sent to the *amavis* user via the amavisd-new cron job.

There are several ways to handle this situation:

- Configure your MDA to filter messages you do not wish to see.
- Change `/usr/sbin/amavisd-new-cronjob` to check for *use_bayes 0*. For example, edit `/usr/sbin/amavisd-new-cronjob` and add the following to the top before the *test* statements:

```
egrep -q "^[ \t]*use_bayes[ \t]*0" /etc/spamassassin/local.cf && exit 0
```

5.3. Test

Per prima cosa, verificare che Amavisd-new SMTP sia in ascolto:

```
telnet localhost 10024
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 [127.0.0.1] ESMTP amavisd-new service ready
^]
```

Nell'intestazione dei messaggi che passano attraverso il filtraggio del contenuto, dovrebbe essere visibile:

```
X-Spam-Level:
X-Virus-Scanned: Debian amavisd-new at example.com
X-Spam-Status: No, hits=-2.3 tagged_above=-1000.0 required=5.0 tests=AWL, BAYES_00
X-Spam-Level:
```

L'output potrebbe variare, ma l'aspetto importante è la presenza delle voci *X-Virus-Scanned* e *X-Spam-Status*.

5.4. Risoluzione problemi

Il miglior metodo per comprendere cosa non funzioni correttamente è controllare i file di registro.

- Per istruzioni sulle registrazioni di Postfix, consultare *Sezione 1.7, «Risoluzione problemi» [240]*.
- Amavisd-new fa uso di Syslog per inviare i messaggi verso `/var/log/mail.log`. Il livello di dettaglio può essere aumentato aggiungendo l'opzione *$log_level* in `/etc/amavis/conf.d/50-user` e impostando il valore da 1 a 5.

  ```
  $log_level = 2;
  ```

  

  Quando il livello dei messaggi di registro di Amavisd-new viene aumentato, viene aumentato automaticamente anche quello di Spamassassin.
- Il livello di messaggi di ClamAV può invece essere aumentato modificando il file `/etc/clamav/clamd.conf` e impostando la seguente opzione:

  ```
  LogVerbose true
  ```

  ClamAV, in modo predefinito, invia i messaggi verso `/var/log/clamav/clamav.log`.

Dopo aver cambiato le impostazioni di registrazione di un'applicazione, ricordarsi di riavviare il servizio. Una volta risolto il problema, è buona norma ripristinare il livello di registrazioni originale.

5.5. Riferimenti

Per maggiori informazioni sul filtraggio mail, consultare i seguenti indirizzi:

- *Documentazione di Amavisd-new*[22]
- *ClamAV Documentation*[23] and *ClamAV Wiki*[24]
- *Wiki di Spamassassin*[25]
- *Sito web di Pyzor*[26]
- *Sito web di Razor*[27]
- *DKIM.org*[28]
- *Postfix Amavis New*[29]

È possibile anche porre le proprie domande nel canale IRC *#ubuntu-server* su *freenode*[30].

[22] http://www.ijs.si/software/amavisd/amavisd-new-docs.html
[23] http://www.clamav.net/doc/latest/html/
[24] http://wiki.clamav.net/Main/WebHome
[25] http://wiki.apache.org/spamassassin/
[26] http://sourceforge.net/apps/trac/pyzor/
[27] http://razor.sourceforge.net/
[28] http://dkim.org/
[29] https://help.ubuntu.com/community/PostfixAmavisNew
[30] http://freenode.net

Capitolo 16. Applicazioni per conversazioni

1. Panoramica

In questa sezione viene discusso come installare e configurare un server IRC (ircd-irc2) e come installare e configurare Jabber, un server di messaggistica istantanea.

2. Server IRC

Nei repository di Ubuntu sono disponibili molti server Internet Relay Chat, ma in questa sezione viene descritto come installare e configurare il server IRC ircd-irc2.

2.1. Installazione

Per installare ircd-irc2, eseguire il seguente comando in un terminale:

```
sudo apt-get install ircd-irc2
```

I file di configurazione sono presenti nella directory `/etc/ircd`, i documenti nella directory `/usr/share/doc/ircd-irc2`.

2.2. Configurazione

Le impostazioni IRC possono essere svolte nel file di configurazione `/etc/ircd/ircd.conf`, dove è possibile impostare il nome host IRC modificando la seguente riga:

```
M:irc.localhost::Debian ircd default configuration::000A
```

Assicurarsi di aggiungere gli alias DNS per il nome host IRC. Per esempio, se il nome host IRC è irc.example.net, assicurarsi che irc.example.net possa essere risolto dal proprio DNS. Il nome host IRC non dovrebbe essere lo stesso del nome host.

I dettagli dell'amministratore possono essere configurati modificando la seguente riga:

```
A:Organization, IRC dept.:Daemon <ircd@example.irc.org>:Client Server::IRCnet:
```

Per configurare le porte IRC da ascoltare, per configurare le credenziali di Operator o l'autenticazione lato client, è necessario aggiungere delle specifiche righe nel file di configurazione. Per maggiori informazioni, fare riferimento al file di configurazione di esempio `/usr/share/doc/ircd-irc2/ircd.conf.example.gz`.

Il messaggio (banner) IRC da visualizzare nei client IRC, quando gli utenti si connettono al server, può essere impostato nel file `/etc/ircd/ircd.motd`.

Una volta apportate le necessarie modifiche al file di configurazione, riavviare il server IRC tramite il seguente comando:

```
sudo /etc/init.d/ircd-irc2 restart
```

2.3. Riferimenti

Potrebbe essere interessante controllare anche altri server IRC disponibili nei repository Ubuntu come ircd-ircu e ircd-hybrid.

- Per maggiori informazioni riguardo il server IRC, consultare le *IRCD FAQ*[1].

[1] http://www.irc.org/tech_docs/ircnet/faq.html

3. Server di messaggistica istantanea Jabber

Jabber è un protocollo di messaggistica molto diffuso, basato su XMPP, uno standard aperto per la messaggistica istantanea e usato da molte applicazioni. Questa sezione espone come configurare un server *Jabberd 2* in una rete locale. La configurazione può anche essere adattata per fornire servizi di messaggistica agli utenti attraverso Internet.

3.1. Installazione

Per installare jabberd2, in un terminale digitare:

```
sudo apt-get install jabberd2
```

3.2. Configurazione

A couple of XML configuration files will be used to configure jabberd2 for *Berkeley DB* user authentication. This is a very simple form of authentication. However, jabberd2 can be configured to use LDAP, MySQL, PostgreSQL, etc for for user authentication.

Aprire il file `/etc/jabberd2/sm.xml` e alla riga:

```
<id>jabber.example.com</id>
```

Sostituire *jabber.example.com* con il nome host, o altro identificativo, del proprio server.

Nella sezione <storage>, modificare <driver> in:

```
<driver>db</driver>
```

Modificare il file `/etc/jabberd2/c2s.xml` e nella sezione <*local*> cambiare:

```
<id>jabber.example.com</id>
```

Nella sezione <authreg> sistemare la sezione <module> in

```
<module>db</module>
```

Riavviare jabberd2 per abilitare le nuove impostazioni:

```
sudo /etc/init.d/jabberd2 restart
```

Dovrebbe quindi essere possibile connettersi al server utilizzando un client Jabber come Empathy.

Il vantaggio nell'uso di Berkeley DB per i dati utenti consiste nella bassa manutenzione necessaria una volta configurato. Per avere un maggiore controllo sugli account utente e le credenziali di autenticazione, è consigliato usare un altro metodo di autenticazione.

3.3. Riferimenti

- Il *sito web di Jabberd2*[2] contiene molte informazioni sulla configurazione di Jabberd2.
- For more authentication options see the *Jabberd2 Install Guide*[3].
- Ulteriori informazioni sono disponibili nella *documentazione online*[4].

[2] http://codex.xiaoka.com/wiki/jabberd2:start
[3] http://www.jabberdoc.org/
[4] https://help.ubuntu.com/community/SettingUpJabberServer

Capitolo 17. Sistemi per il controllo della versione

Il controllo della versione è l'arte della gestione dell'evolversi delle informazioni. È stato a lungo uno strumento critico per i programmatori, che spendono il loro tempo apportando piccole modifiche al software per poi cancellarle il giorno seguente. Ma l'utilità del software per il controllo della versione va oltre il mondo dello sviluppo di programmi. Ovunque si incontrino persone che utilizzino il computer per gestire informazioni in continuo cambiamento c'è posto per il controllo della versione.

1. Bazaar

Bazaar è un nuovo sistema di controllo della versione sponsorizzato da Canonical, la società commerciale dietro Ubuntu. Diversamente da Subversion e CVS che supportano solamente un modello centralizzato di repository, Bazaar supporta anche un *controllo distribuito della versione*, consentendo alle persone di collaborare più efficientemente. In particolare, Bazaar è progettato per massimizzare il livello di partecipazione della comunità nei progetti open source.

1.1. Installazione

Per installare bzr, in un terminale, digitare:

```
sudo apt-get install bzr
```

1.2. Configurazione

Per introdursi a bzr, usare il comandi *whoami*:

```
$ bzr whoami 'Mario Rossi <mario.rossi@ubuntu.com>'
```

1.3. Imparare a usare Bazaar

La documentazione fornita con Bazaar è installata in /usr/share/doc/bzr/html, il tutorial è un buon punto di partenza. Il comando bzr è dotato di un sistema di aiuto integrato:

```
$ bzr help
```

Per avere maggiori informazioni riguardo il comando *foo*:

```
$ bzr help foo
```

1.4. Integrazione con Launchpad

Anche se è altamente utilizzabile come strumento dedicato, Bazaar è dotato di un'ottima integrazione con *Launchpad*[1], il sistema di sviluppo collaborativo utilizzato da Canonical, e altre comunità di progetti open source, per la gestione di Ubuntu. Per informazioni su come Bazaar possa essere usato con Launchpad per la collaborazione nei progetti open source, consultare *http://bazaar-vcs.org/LaunchpadIntegration*[2].

[1] https://launchpad.net/

[2] http://bazaar-vcs.org/LaunchpadIntegration/

2. Subversion

Subversion è un software open source per il controllo della versione. Utilizzando Subversion è possibile registrare la storia del codice sorgente e dei documenti. È in grado di gestire l'evolversi di file e directory nel tempo. Nel repository centrale viene posizionato un albero di tutti i file. Il repository è come un server di file, tranne per il fatto che si ricorda qualsiasi cambiamento apportato.

2.1. Installazione

Per accedere al repository di Subversion utilizzando il protocollo HTTP, è necessario installare e configurare un server web come Apache2, che funziona molto bene con Subversion. Fare riferimento alla sottosezione HTTP della sezione relativa ad Apache2 per installare e configurare un certificato digitale.

Per installare Subversion, in un terminale, digitare:

```
sudo apt-get install subversion libapache2-svn
```

2.2. Configurazione del server

I passi seguenti presumono siano stati installati i pacchetti elencati in precedenza. Questa sezione descrive come creare un repository con Subversion e come accedere al progetto.

2.2.1. Creare un repository con Subversion

Un repository può essere creato con il seguente comando:

```
svnadmin create /posizione/del/repository/project
```

2.2.2. Importare i file

Una volta creato il repository è possibile *import*arvi file. Per importare una directory, digitare ciò che segue al prompt del terminale:

```
svn import /percorso/della/directory/da/importare file:///percorso/del/repository/
```

2.3. Metodi di accesso

È possibile accedere (checkout) ai repository Subversion in diversi modi, sul disco locale o attraverso diversi protocolli di rete. La posizione di un repository, comunque, è sempre un URL. La tabella illustra come i diversi schemi URL vengono mappati ai diversi metodi di accesso.

Tabella 17.1. Metodi di accesso

Schema	Metodo di accesso
file://	Accesso diretto al repository (sul disco locale)

Schema	Metodo di accesso
http://	Accesso attraverso il protocollo WebDAV al server web Apache2 di Subversion
https://	Come http://, ma con cifratura SSL
svn://	Accesso attraverso un protocollo personalizzato a un server svnserve
svn+ssh://	Come svn://, ma attraverso un tunnel SSH

In questa sezione viene descritto come configurare Subversion per tutti questi metodi. Saranno descritti solo gli elementi basilari. Per maggiori informazioni, fare riferimento al *libro di svn*[3].

2.3.1. Accesso diretto al repository (file://)

Questo è il metodo di accesso più semplice. Non necessita di alcun server di Subversion in esecuzione e serve per accedere a Subversion dalla stessa macchina in cui è in esecuzione. La sintassi del comando è la seguente:

```
svn co file:///percorso/del/repository/progetto
```

o

```
svn co file://localhost/percorso/del/repository/progetto
```

Se non viene specificato l'host, è necessario utilizzare tre slash (///), due per il protocollo (in questo caso file) e uno è lo slash iniziale del percorso. Se viene specificato l'host, utilizzare due slash (//).

I permessi di accesso al repository dipendono dai permessi impostati nel file system. Se l'utente possiede i permessi di scrittura e lettura, allora potrà eseguire checkout e commit al repository.

2.3.2. Accesso con il protocollo WebDAV (http://)

To access the Subversion repository via WebDAV protocol, you must configure your Apache 2 web server. Add the following snippet between the *<VirtualHost>* and *</VirtualHost>* elements in `/etc/apache2/sites-available/default`, or another VirtualHost file:

```
<Location /svn>
 DAV svn
 SVNPath /home/svn
 AuthType Basic
 AuthName "Your repository name"
 AuthUserFile /etc/subversion/passwd
 Require valid-user
</Location>
```

[3] http://svnbook.red-bean.com/

The above configuration snippet assumes that Subversion repositories are created under `/home/svn/` directory using **svnadmin** command. They can be accessible using **http://hostname/svn/repos_name** url.

Per importare o eseguire il "commit" di file nel proprio repository Subversion via HTTP, il repository deve essere di proprietà dell'utente del servizio HTTP. Nei sistemi Ubuntu, solitamente, l'utente del servizio HTTP è **www-data**. Per cambiare il proprietario dei file del repository, digitare il comando seguente in un terminale:

```
sudo chown -R www-data:www-data /percorso/al/repository
```

Modificando il proprietario del repository come **www-data** non sarà più possibile importare o eseguire il "commit" di file nel repository attraverso il comando **svn import file:///** come un qualsiasi utente, ma solo come **www-data**.

Creare il file `/etc/subversion/passwd` che conterrà i dettagli di autenticazione utente. Per creare un file, eseguire il seguente comando al prompt dei comandi (viene creato il file e aggiunto il primo utente):

```
sudo htpasswd -c /etc/subversion/passwd nome_utente
```

Per aggiungere ulteriori utenti, omettere l'opzione "*-c*" poiché questa opzione sostituisce i vecchio file. Usare invece questa forma:

```
sudo htpasswd /etc/subversion/passwd user_name
```

Verrà richiesta la password. Una volta inserita, l'utente viene aggiunto al file. Ora, per accedere al repository, digitare:

```
svn co http://servername/svn
```

La password viene trasmessa come testo in chiaro. Per evitare attacchi di tipo "password snooping", è necessario utilizzare la cifratura SSL. Per maggiori informazioni fare riferimento alla sezione successiva.

2.3.3. Accesso con protocollo WebDAV protetto da cifratura SSL (https://)

Accedere a un repository Subversion attraverso il protocollo WebDAV con cifratura SSL (https://) è simile a http://, l'unica differenza sta nel dover installare e configurare il certificato digitale nel server web Apache. Per usare SSL con Subversion, aggiungere la precedente configurazione di Apache2 al file `/etc/apache2/sites-available/default-ssl`. Per maggiori informazioni su come configurare Apache2 con SSL, consultare *Sezione 1.3, «Configurazione HTTPS» [194]*.

È possibile installare un certificato digitale emesso da un'autorità certificante o in alternativa è possibile usare un certificato auto-firmato.

I passi seguenti hanno come presupposto l'installazione di un certificato digitale all'interno del server web Apache2. Per accedere a un repository Subversion, fare riferimento alla sezione precedente. I metodi di accesso sono esattamente gli stessi tranne per il protocollo, in quanto è necessario utilizzare https://.

2.3.4. Accesso con il protocollo personalizzato (svn://)

Una volta creato il repository è possibile configurare il controllo degli accessi modificando il file `/path/to/repos/project/conf/svnserve.conf`. Per esempio, per impostare l'autenticazione, togliere i commenti alle seguenti righe presenti nel file di configurazione:

```
# [general]
# password-db = passwd
```

Dopo aver tolto i commenti alle righe precedenti, è possibile gestire la lista degli utenti nel file passwd. Modificare il file `passwd` presente nella directory e inserire il nuovo utente. La sintassi da usare è la seguente:

```
username = password
```

Per maggiori informazioni fare riferimento al file.

Per accedere a Subversion attraverso il protocollo svn://, sia dalla stessa macchina sia da un'altra macchina, avviare svnserver utilizzando il comando svnserve. La sintassi è la seguente:

```
$ svnserve -d --foreground -r /percorso/al/repository
# -d -- daemon mode
# --foreground -- run in foreground (useful for debugging)
# -r -- root of directory to serve

Per ulteriori dettagli sull'utilizzo fare riferimento a:
$ svnserve --help
```

Una volta eseguito questo comando, Subversion si mette in ascolto sulla porta predefinita (3690). Per accedere al repository del progetto, è necessario eseguire, da un terminale, il seguente comando:

```
svn co svn://hostname/project project --username nome_utente
```

In base alla configurazione del server, verrà richiesta la password. Una volta autenticati, viene eseguito il check out del codice dal repository di Subversion. Per sincronizzare il repository del progetto con la copia locale, è possibile eseguire il comando **update**. La sintassi del comando è la seguente:

```
cd DIRECTORY_DEL_PROGETTO ; svn update
```

Per maggiori informazioni sui sotto comandi di Subversion fare riferimento al manuale. Per esempio, per informazioni sul comando co (checkout), al prompt dei comandi digitare:

```
svn co help
```

2.3.5. Accesso con protocollo personalizzato a cifratura SSL (svn+ssh://)

La configurazione e le procedure sono le medesime del metodo svn:// . Per i dettagli consultare la sezione precedente. Questo passaggio prevede che sia stata seguita la procedura precedente e il server Subversion sia stato avviato con il comando svnserve.

Si suppone che il server ssh sia in esecuzione sulla macchina e che accetti connessioni in entrata. Per una conferma, provare a collegarsi alla macchina attraverso SSH. Se il login viene eseguito, tutto è configurato. In caso contrario configurare SSH.

Il protocollo svn+ssh:// è utilizzato per accedere al repository di Subversion usando la cifratura SSL. I dati che vengono trasmessi sono cifrati con questo metodo. Per accedere al repository del progetto (per esempio attraverso un checkout), utilizzare, con il comando, la sintassi seguente:

```
svn co svn+ssh://hostname/var/svn/repos/project
```

È necessario utilizzare il percorso completo (/percorso/al/repository/progetto) per accedere al repository di Subversion utilizzando questo metodo di accesso.

In base alla configurazione del server, viene richiesta la password. Utilizzare la password per il login con SSH. Una volta autenticati, viene eseguito il checkout del codice dal repository di Subverison.

3. Server CVS

CVS è un sistema di controllo della versione che è possibile utilizzare per registrare i cambiamenti al codice sorgente di un programma.

3.1. Installazione

Per installare CVS, eseguire il seguente comando in un terminale:

```
sudo apt-get install cvs
```

Una volta installato cvs, installare xinetd per avviare/fermare il server CVS. In un terminale, digitare quando segue per installare xinetd:

```
sudo apt-get install xinetd
```

3.2. Configurazione

Once you install cvs, the repository will be automatically initialized. By default, the repository resides under the /srv/cvs directory. You can change this path by running following command:

```
cvs -d /your/new/cvs/repo init
```

Once the initial repository is set up, you can configure xinetd to start the CVS server. You can copy the following lines to the `/etc/xinetd.d/cvspserver` file.

```
service cvspserver
{
     port = 2401
     socket_type = stream
     protocol = tcp
     user = root
     wait = no
     type = UNLISTED
     server = /usr/bin/cvs
     server_args = -f --allow-root /srv/cvs pserver
     disable = no
}
```

Be sure to edit the repository if you have changed the default repository (/srv/cvs) directory.

Once you have configured xinetd you can start the cvs server by running following command:

```
sudo /etc/init.d/xinetd restart
```

Per avere la conferma che il server CVS è in esecuzione, digitare il seguente comando:

```
sudo netstat -tap | grep cvs
```

L'output del comando precedente dovrebbe essere:

```
tcp 0 0 *:cvspserver *:* LISTEN
```

A questo punto è possibile aggiungere altri utenti, nuovi progetti e gestire il server CVS.

CVS consente di aggiungere nuovi utenti indipendentemente dal sistema operativo. Il modo più semplice è utilizzare l'utente Linux per CVS, benché presenti dei problemi di sicurezza. Per maggiori informazioni, consultare il manuale di CVS.

3.3. Aggiungere progetti

This section explains how to add new project to the CVS repository. Create the directory and add necessary document and source files to the directory. Now, run the following command to add this project to CVS repository:

```
cd your/project
cvs -d :pserver:username@hostname.com:/srv/cvs import -m \
"Importing my project to CVS repository" . new_project start
```

È possibile utilizzare la variabile d'ambiente CVSROOT per memorizzare la directory root di CVS. Una volta esportata, si può evitare di utilizzare l'opzione "-d" nel comando precedente.

The string *new_project* is a vendor tag, and *start* is a release tag. They serve no purpose in this context, but since CVS requires them, they must be present.

When you add a new project, the CVS user you use must have write access to the CVS repository (/srv/cvs). By default, the src group has write access to the CVS repository. So, you can add the user to this group, and he can then add and manage projects in the CVS repository.

4. Riferimenti

Sito web di Bazaar[4]

Launchpad[5]

Sito web di Subversion[6]

Libro su Subversion[7]

Manuale CVS[8]

Easy Bazaar Ubuntu Wiki page[9]

Ubuntu Wiki Subversion page[10]

[4] http://bazaar.canonical.com/en/
[5] https://launchpad.net/
[6] http://subversion.tigris.org/
[7] http://svnbook.red-bean.com/
[8] http://ximbiot.com/cvs/manual/cvs-1.11.21/cvs_toc.html
[9] https://help.ubuntu.com/community/EasyBazaar
[10] https://help.ubuntu.com/community/Subversion

Capitolo 18. Reti Windows

Spesso le reti di computer sono costituite da sistemi eterogenei e, sebbene gestire una rete composta interamente da computer con Ubuntu sarebbe certamente divertente, alcuni ambienti di rete debbono essere costituiti da sistemi Ubuntu e Microsoft® Windows® che operano insieme in armonia. Questa sezione della guida di Ubuntu introduce i principi e gli strumenti utilizzati nella configurazione di un server Ubuntu per la condivisione di risorse di rete con computer Windows.

1. Introduzione

Utilizzare Ubuntu in una rete composta da client Windows significa fornire e integrare i servizi tipici degli ambienti Windows. Questi servizi offrono supporto per la condivisione di dati e informazioni riguardo i computer e gli utenti della rete e possono essere classificati, in base alle loro funzionalità, in tre principali categorie:

- **Servizi per la condivisione di file e stampanti**. Utilizzo del protocollo SMB (Server Message Block) per agevolare la condivisione di file, cartelle, volumi e stampanti attraverso la rete.
- **Servizi di directory**. Condivisione di informazioni vitali sui computer e sugli utenti della rete con l'uso di tecnologie come LDAP (Lightweight Directory Access Protocol) e Microsoft Active Directory®.
- **Autenticazione e accesso**. Stabilire l'identità del computer o dell'utente della rete e determinare quali risorse siano accessibili al computer o all'utente tramite i permessi e i privilegi, utilizzando permessi dei file, politiche di gruppo e il servizio di autenticazione Kerberos.

Fortunatamente, i sistemi Ubuntu sono in grado di fornire queste funzionalità ai client Windows, permettendo la condivisione di risorse di rete. Uno dei componenti software principali, incluso nei sistemi Ubuntu per le operazioni di rete con Windows, è la suite SAMBA, che comprende strumenti e applicazioni per server SMB.

Questa sezione della guida server di Ubuntu è un'introduzione all'uso di Samba e a come installare e configurare i pacchetti necessari. Per maggiori informazioni e documentazione su Samba, consultare il *sito web di Samba*[1].

[1] http://www.samba.org

2. Server di file Samba

Una delle opzioni più comuni per mettere in comunicazione computer con Ubuntu e Windows, è quella di configurare Samba come server di file. Questa sezione spiega come configurare un server Samba per la condivisione di file con client Windows.

Il server viene configurato per condividere file con qualsiasi client nella rete senza dover usare una password. Se all'interno del proprio ambiente di lavoro è richiesto un maggior controllo sugli accessi, consultare *Sezione 4, «Sicurezza di un server di file e di stampa Samba» [284]*

2.1. Installazione

Per prima cosa installare il pacchetto samba. Alla riga di comando, digitare:

```
sudo apt-get install samba
```

Questo è quanto. Ora è possibile configurare Samba affinché possa condividere i file.

2.2. Configurazione

Il file principale di configurazione di Samba è localizzato in `/etc/samba/smb.conf` e dispone di molti commenti utili nella configurazione delle varie direttive.

Non tutte le opzioni disponibili sono incluse nel file di configurazione predefinito. Per maggiori informazioni, consultare la pagina man di `smb.conf` oppure «*Samba HOWTO Collection*[2]».

1. Per prima cosa, modificare le seguenti coppie chiave/valore nella sezione *[global]* del file `/etc/samba/smb.conf`:

   ```
   workgroup = ESEMPIO
   ...
   security = user
   ```

   Il parametro *security* è più avanti nella sezione [global] ed è commentato. Inoltre, modificare *ESEMPIO* in modo che rispecchi il proprio ambiente di lavoro.
2. Per la nuova directory da condividere, creare una nuova sezione verso la fine del file oppure togliere il commento a uno degli esempi:

   ```
   [share]
       comment = Condivisione file Ubuntu
       path = /srv/samba/share
       browsable = yes
       guest ok = yes
       read only = no
   ```

[2] http://samba.org/samba/docs/man/Samba-HOWTO-Collection/

```
create mask = 0755
```

- *comment*: una breve descrizione della condivisione. Modificarla in base alle proprie esigenze.
- *path*: il percorso alla directory da condividere.

  Questo esempio utilizza `/srv/samba/sharename` poiché, in base alla *Filesystem Hierarchy Standard (FHS)*, */srv*[3] è la posizione in cui dovrebbero essere tenuti i file relativi ai siti. Tecnicamente, le condivisione Samba possono essere posizionate ovunque all'interno del file system, basta che i permessi siano impostati correttamente. In ogni caso, è raccomandato aderire agli standard.
- *browsable*: abilita i client Windows a esplorare la directory condivisa usando Windows Explorer.
- *guest ok*: consente ai client di connettersi alla condivisione senza dover fornire una password.
- *sola lettura:* determina se la condivisione è di sola lettura o se sono garantiti anche i privilegi di scrittura. I privilegi di scrittura sono consentiti solo quando il valore è *no*, come mostrato nell'esempio. Se il valore è *si*, allora l'accesso alla condivisione è in sola lettura.
- *create mask*: determina i permessi dei nuovi file creati.

3. Ora che Samba è configurato, è necessario creare la directory e modificarne i permessi. Da un terminale digitare:

```
sudo mkdir -p /srv/samba/share
sudo chown nobody.nogroup /srv/samba/share/
```

The *-p* switch tells mkdir to create the entire directory tree if it doesn't exist.

4. Infine, riavviare il servizio samba per abilitare la nuova configurazione:

```
sudo restart smbd
sudo restart nmbd
```

La configurazione precedente fornisce accesso completo a tutti i client nella rete locale. Per una configurazione più sicura, consultare *Sezione 4, «Sicurezza di un server di file e di stampa Samba» [284]*.

From a Windows client you should now be able to browse to the Ubuntu file server and see the shared directory. If your client doesn't show your share automatically, try to access your server by its IP address, e.g. \\192.168.1.1, in a Windows Explorer window. To check that everything is working try creating a directory from Windows.

Per creare ulteriori condivisioni basta creare delle nuove sezioni *[dir]* nel file `/etc/samba/smb.conf` e riavviare *Samba*. Assicurarsi che le directory da condividere esistano e abbiano i permessi impostati correttamente.

[3] http://www.pathname.com/fhs/pub/fhs-2.3.html#SRVDATAFORSERVICESPROVIDEDBYSYSTEM

The file share named *"[share]"* and the path `/srv/samba/share` are just examples. Adjust the share and path names to fit your environment. It is a good idea to name a share after a directory on the file system. Another example would be a share name of *[qa]* with a path of `/srv/samba/qa`.

2.3. Risorse

- Per delle configurazioni più dettagliate riguardo Samba, consultare *Samba HOWTO Collection*[4]
- La guida è disponibile anche in *formato cartaceo*[5].
- Il libro *Using Samba*[6] di O'Reilly è un'altra buona lettura.
- La pagina *su Samba*[7] della documentazione.

[4] http://samba.org/samba/docs/man/Samba-HOWTO-Collection/
[5] http://www.amazon.com/exec/obidos/tg/detail/-/0131882228
[6] http://www.oreilly.com/catalog/9780596007690/
[7] https://help.ubuntu.com/community/Samba

3. Server di stampa Samba

Un' altra configurazione molto comune di Samba è come condivisione di stampanti installate, localmente o in remoto, su un server Ubuntu. Come *Sezione 2, «Server di file Samba» [279]*, questa sezione spiega come configurare Samba affinché qualsiasi client sulla rete locale possa utilizzare le stampanti installate senza la necessità di fornire nome utente o password.

Per una configurazione più sicura, consultare *Sezione 4, «Sicurezza di un server di file e di stampa Samba» [284]*.

3.1. Installazione

Prima di installare e configurare Samba è utile prima di tutto avere un'installazione funzionante di CUPS. Per maggiori informazioni, consultare *Sezione 4, «CUPS - Server di stampa» [232]*.

Per installare il pacchetto samba, da un terminale digitare:

```
sudo apt-get install samba
```

3.2. Configurazione

After installing samba edit `/etc/samba/smb.conf`. Change the *workgroup* attribute to what is appropriate for your network, and change *security* to *user*:

```
workgroup = ESEMPIO
...
security = user
```

Nella sezione *[printers]* modificare l'opzione *guest ok* a *yes*:

```
browsable = yes
guest ok = yes
```

Una volta modificato il file `smb.conf`, riavviare Samba:

```
sudo restart smbd
sudo restart nmbd
```

La configurazione predefinita di Samba condividerà automaticamente qualsiasi stampante installata. Basta installare la stampante localmente sui client Windows.

3.3. Risorse

- Per delle configurazioni più dettagliate riguardo Samba, consultare *Samba HOWTO Collection*[8]

[8] http://samba.org/samba/docs/man/Samba-HOWTO-Collection/

- La guida è disponibile anche in *formato cartaceo*[9].
- Il libro *Using Samba*[10] di O'Reilly è un'altra buona lettura.
- Per maggiori informazioni sulla configurazione di CUPS, consultare il *sito web di CUPS*[11].
- La pagina *su Samba*[12] della documentazione.

[9] http://www.amazon.com/exec/obidos/tg/detail/-/0131882228
[10] http://www.oreilly.com/catalog/9780596007690/
[11] http://www.cups.org/
[12] https://help.ubuntu.com/community/Samba

4. Sicurezza di un server di file e di stampa Samba

4.1. Modalità di sicurezza di Samba

Esistono due livelli di sicurezza disponibili al protocollo CIFS (Common Internet Filesystem): a livello *utente* e a livello *condivisione*. L'implementazione della *modalità di sicurezza* di Samba consente una maggiore flessibilità, fornendo quattro modi per implementare la sicurezza a livello utente e uno per quella a livello condivisione.

- *security = user*: richiede a client di fornire nome utente e password per collegarsi alla condivisione. Gli account di Samba sono separati da quelli di sistema, ma il pacchetto libpam-smbpass consente di sincronizzare utenti e password con il database degli utenti di Samba.
- *security = domain*: questa modalità consente al server Samba di apparire ai client Windows come «Primary Domain Controller» (PDC), «Backup Domain Controller» (BDC) oppure «Domain Member Server» (DMS). Per maggiori informazioni, consultare *Sezione 5, «Samba come controller di dominio» [289]*.
- *security = ADS*: consente al server Samba di unirsi a un dominio «Active Directory» come membro nativo. Per maggiori informazioni, consultare *Sezione 6, «Integrare Samba con Active Directory» [294]*.
- *security = server*: questa modalità non dovrebbe essere usata per motivi di sicurezza. Per maggiori informazioni, consultare la sezione *Server Security*[13] della guida di Samba.
- *security = share*: consente ai client di collegarsi alle condivisioni senza fornire nome utente e password.

La modalità di sicurezza scelta dipende dal proprio ambiente di lavoro e da cosa si vuole ottenere col server Samba.

4.2. Livello di sicurezza utente

Questa sezione spiega come riconfigurare i server di file e di stampa Samba, come spiegato in *Sezione 2, «Server di file Samba» [279]* e *Sezione 3, «Server di stampa Samba» [282]*, affinché richieda l'autenticazione.

Per prima cosa, installare il pacchetto libpam-smbpass che consente di sincronizzare gli utenti di sistema col database degli utenti di Samba:

```
sudo apt-get install libpam-smbpass
```

Se è stato scelto il task *Server Samba* durante l'installazione, il pacchetto libpam-smbpass è già installato.

Aprire il file `/etc/samba/smb.conf` e nella sezione *[share]* modificare:

[13] http://samba.org/samba/docs/man/Samba-HOWTO-Collection/ServerType.html#id349531

```
guest ok = no
```

Riavviare Samba affinché le nuove impostazioni abbiano effetto:

```
sudo restart smbd
sudo restart nmbd
```

Ora, collegandosi alle directory o alle stampanti condivise, verranno richiesti il nome utente e la password.

Se si sceglie di mappare un drive di rete alla condivisione, selezionare la casella di spunta «Reconnect at Logon» affinché sia possibile inserire nome utente e password solo una volta, almeno finché la password non viene cambiata.

4.3. Livello di sicurezza condivisione

Ci sono diverse opzioni disponibili per aumentare la sicurezza di ogni singola directory condivisa. Facendo uso dell'esempio *[share]*, questa sezione illustra alcune di queste opzioni.

4.3.1. Gruppi

I gruppi definiscono un insieme di computer e utenti che godono dei medesimi privilegi di accesso alle risorse condivise, offrendo un alto livello di controllo di questi accessi. Per esempio, se il gruppo *qa* contiene gli utenti *freda*, *danika* e *rob* e viene definito il secondo gruppo *support* che contiene gli utenti *danika*, *jeremy* e *vincent*, allora alcune risorse di rete impostate per concedere l'accesso al gruppo *qa* concedono automaticamente l'accesso anche agli utenti freda, danika e rob, mentre lo negano a jeremy o vincent. Dal momento che l'utente *danika* è membro di entrambi i gruppi *qa* e *support* potrà accedere a tutte le risorse condivise il cui accesso è stato concesso a entrambi i gruppi, gli altri utenti avranno accesso alle risorse esplicitamente assegnate al gruppo di appartenenza.

Samba, in modo predefinito, controlla i gruppi di sistema locali definiti in `/etc/group` per determinare quali utenti appartengono a quali gruppi. Per maggiori informazioni su come aggiungere o rimuovere gruppi, consultare *Sezione 1.2, «Aggiungere e rimuovere utenti» [155]*.

Quando si definiscono i gruppi nel file di configurazione di Samba, `/etc/samba/smb.conf`, la sintassi predefinita è quella di usare il prefisso "@" col nome del gruppo. Per esempio, per definire il gruppo *sysadmin* in una sezione del file `/etc/samba/smb.conf`, bisogna inserire il nome del gruppo come **@sysadmin**.

4.3.2. Permessi dei file

I permessi dei file definiscono i diritti che un computer o un utente ha su una particolare directory, file o insieme di file. Tali permessi possono essere definiti modificando il file `/etc/samba/smb.conf` e specificando i permessi di una condivisione definita.

Per esempio, se è stata definita una condivisione Samba chiamata *share* e si vuole dare il permesso di *sola lettura* al gruppo di utenti conosciuto come *qa*, ma si vuole concedere permesso di scrittura

sulla condivisione al gruppo *sysadmin* e all'utente *vincent*, modificare il file `/etc/samba/smb.conf` e aggiungere quanto segue al di sotto della sezione *[share]*:

```
read list = @qa
write list = @sysadmin, vincent
```

Un altro possibile permesso con Samba consente di usare i permessi *amministrativi* su una particolare risorsa condivisa. Gli utenti con permessi amministrativi possono leggere, scrivere o modificare qualsiasi informazione all'interno della risorsa per cui sono stati abilitati.

Per esempio, per concedere all'utente *melissa* permessi amministrativi all'interno dell'esempio *share*, modificare il file `/etc/samba/smb.conf` e aggiungere quanto segue al di sotto della sezione *[share]*:

```
admin users = melissa
```

Modificato il file `/etc/samba/smb.conf`, riavviare Samba affinché le modifiche abbiano effetto:

```
sudo restart smbd
sudo restart nmbd
```

Affinché *read list* e *write list* funzionino, il modello di sicurezza di Samba *non* deve essere impostato a *security = share*

Ora che Samba è stato configurato per limitare quali gruppi hanno accesso alla directory condivisa, è necessario aggiornare i permessi del file system.

Il sistema dei permessi sui file di Linux non funziona correttamente con le ACL (Access Control List) di Windows NT. In questi casi, nei server Ubuntu, sono disponibili le ACL POSIX che forniscono un controllo più fine. Per esempio, per abilitare le ACL su `/srv` con file system ext3, modificare il file `/etc/fstab` aggiungendo l'opzione *acl*:

```
UUID=66bcdd2e-8861-4fb0-b7e4-e61c569fe17d /srv  ext3    noatime,relatime,acl 0       1
```

Quindi montare nuovamente la partizione:

```
sudo mount -v -o remount /srv
```

L'esempio precedente assume che `/srv` sia in una partizione separata. Se `/srv` o qualsiasi sia il percorso di condivisione, fa parte della partizione `/`, potrebbe essere necessario riavviare il sistema.

Per uguagliare la configurazione precedente di Samba, al gruppo *sysadmin* devono essere dati i permessi di lettura, scrittura e di esecuzione su `/srv/samba/share`, al gruppo *qa* devono essere dati i permessi di lettura ed esecuzione e i file devono essere di proprietà del nome utente *melissa*. In un terminale, digitare quanto segue:

```
sudo chown -R melissa /srv/samba/share/
sudo chgrp -R sysadmin /srv/samba/share/
sudo setfacl -R -m g:qa:rx /srv/samba/share/
```

Il comando setfacl imposta i permessi di *esecuzione* a tutti i file nella directory `/srv/samba/share`. Nel caso non sia desiderato, non eseguire il comando.

Da un client Windows ora dovrebbe essere possibile notare la nuova implementazione dei permessi dei file. Per maggiori informazioni riguardo le ACL POSIX, consultare le pagine di manuale di acl e setfacl.

4.4. Profilo AppArmor Samba

Ubuntu è dotato del modulo di sicurezza AppArmor, che fornisce un controlli di acceso. Il profilo predefinito di AppArmor per Samba deve essere adattato alla propria configurazione. Per maggiori informazioni sull'uso di AppArmor, consultare *Sezione 4, «AppArmor» [168]*.

All'interno del pacchetto apparmor-profiles sono disponibili dei profili predefiniti di AppArmor per `/usr/sbin/smbd` e `/usr/sbin/nmbd`, i binari dei demoni di Samba. Per installare il pacchetto, da un terminale digitare:

```
sudo apt-get install apparmor-profiles apparmor-utils
```

Questo pacchetto contiene profili per molti altri binari.

I profili per smbd e nmbd sono, in modo predefinito, nella modalità *complain*, consentendo a Samba di lavorare senza dover modificare il profilo e registrando solamente gli errori. Per impostare il profilo smbd in modalità *enforce* e per far funzionare Samba come di consueto, il profilo deve essere modificato per rispecchiare le directory da condividere.

Modificare il file `/etc/apparmor.d/usr.sbin.smbd` aggiungendo informazioni alla sezione *[share]* dall'esempio del server di file:

```
  /srv/samba/share/ r,
  /srv/samba/share/** rwkix,
```

Ora impostare il profilo in modalità *enforce* e ricaricarlo:

```
sudo aa-enforce /usr/sbin/smbd
cat /etc/apparmor.d/usr.sbin.smbd | sudo apparmor_parser -r
```

Dovrebbe essere possibile leggere, scrivere ed eseguire i file nella directory condivisa come di consuetudine e il binario smbd dovrebbe avere accesso solo ai file e le directory configurati. Assicurarsi di aggiungere una voce per ogni directory che viene configurata alla condivisione. Tutti gli errori verranno registrati in `/var/log/syslog`.

4.5. Risorse

- Per delle configurazioni più dettagliate riguardo Samba, consultare *Samba HOWTO Collection*[14]
- La guida è disponibile anche in *formato cartaceo*[15].
- Il libro *Using Samba*[16] di O'Reilly è un'altra buona lettura.
- Il *capitolo 18*[17] della «Samba HOWTO Collection» è dedicato alla sicurezza.
- Il libro *Using Samba*[18] di O'Reilly è un'altra buona lettura.
- La pagina *su Samba*[19] della documentazione.

[14] http://samba.org/samba/docs/man/Samba-HOWTO-Collection/
[15] http://www.amazon.com/exec/obidos/tg/detail/-/0131882228
[16] http://www.oreilly.com/catalog/9780596007690/
[17] http://samba.org/samba/docs/man/Samba-HOWTO-Collection/securing-samba.html
[18] http://www.oreilly.com/catalog/9780596007690/
[19] https://help.ubuntu.com/community/Samba

5. Samba come controller di dominio

Benché non possa funzionare come un controller di dominio primario (PDC) Active Directory, un server Samba può essere configurato per apparire come un controller di dominio in stile Windows NT4. Uno dei vantaggi di questa configurazione consiste nell'abilità di centralizzare le credenziali di utenti e computer, inoltre, Samba può utilizzare diversi backend per archiviare le informazioni.

5.1. Controller di dominio primario (PDC)

Questa sezione spiega come configurare Samba come controller di dominio primario (PDC) usando il backend predefinito «smbpasswd».

1. Per prima cosa, installare Samba e libpam-smbpass per sincronizzare gli account utente digitando quanto segue in un terminale:

   ```
   sudo apt-get install samba libpam-smbpass
   ```

2. Configurare Samba modificando il file `/etc/samba/smb.conf`. La variabile *security* dovrebbe essere impostata a *user* e il *workgroup* dovrebbe essere relativo alla propria organizzazione.

   ```
   workgroup = ESEMPIO
   ...
   security = user
   ```

3. In the commented «Domains» section add or uncomment the following (the last line has been split to fit the format of this document):

   ```
   domain logons = yes
   logon path = \\%N\%U\profile
   logon drive = H:
   logon home = \\%N\%U
   logon script = logon.cmd
   add machine script = sudo /usr/sbin/useradd -N -g machines -c Machine -d
         /var/lib/samba -s /bin/false %u
   ```

   

   Per non usare i profili *roaming*, non togliere il commento alle opzioni *logon home* e *logon path*.

   - *domain logons*: fornisce il servizio netlogon facendo in modo che Samba si comporti come un controller di dominio.
   - *logon path*: posiziona il profilo degli utenti Windows all'interno della loro directory home. È possibile anche configurare una condivisione *[profiles]* posizionando tutti i profili all'interno di una sola directory.
   - *logon drive*: specifica il percorso locale della directory home.
   - *logon home*: specifica la posizione della directory home.

- *logon script*: determina quale script eseguire localmente una volta che un utente ha eseguito l'accesso. Lo script deve essere all'interno della condivisione *[netlogon]*.
- *add machine script*: uno script che crea automaticamente lo *Machine Trust Account* necessario per accedere al dominio.

  In questo esempio il gruppo *machines* deve essere creato usando l'utilità addgroup. Per maggiori informazioni, consultare *Sezione 1.2, «Aggiungere e rimuovere utenti» [155]*.

4. Togliere il commento alla condivisione *[homes]* per consentire la mappatura di *logon home*:

```
[homes]
   comment = Home Directories
   browseable = no
   read only = no
   create mask = 0700
   directory mask = 0700
   valid users = %S
```

5. Quando configurato come controller di dominio, è necessario configurare una condivisione *[netlogon]*. Per abilitarla, togliere il commento a:

```
[netlogon]
   comment = Network Logon Service
   path = /srv/samba/netlogon
   guest ok = yes
   read only = yes
   share modes = no
```

Il percorso della condivisione predefinita di *netlogon* è `/home/samba/netlogon`, ma in base allo «Filesystem Hierarchy Standard» (FHS), */srv*[20] è la corretta posizione i cui dovrebbero essere tenuti i file specifici dei siti forniti dal sistema.

6. Creare la directory `netlogon` e un file `logon.cmd` per ora vuoto:

```
sudo mkdir -p /srv/samba/netlogon
sudo touch /srv/samba/netlogon/logon.cmd
```

È possibile inserire qualsiasi comando di logon Windows in `logon.cmd` per personalizzare l'ambiente del client.

7. Restart Samba to enable the new domain controller:

```
sudo restart smbd
sudo restart nmbd
```

8. Lastly, there are a few additional commands needed to setup the appropriate rights.

[20] http://www.pathname.com/fhs/pub/fhs-2.3.html#SRVDATAFORSERVICESPROVIDEDBYSYSTEM

Con l'utente *root* disabilitato in modo predefinito, per poter inserire una workstation nel dominio, un gruppo di sistema deve essere mappato al gruppo Windows *Domain Admins*. Usando l'utilità net, da un terminale digitare:

```
sudo net groupmap add ntgroup="Domain Admins" unixgroup=sysadmin rid=512 type=d
```

Modificare *sysadmin* con un qualsiasi altro gruppo si voglia usare. Inoltre, l'utente usato per unirsi al dominio deve essere membro del gruppo *sysadmin* oltre al gruppo *admin*. Il gruppo *admin* consente l'utilizzo di sudo.

If the user does not have Samba credentials yet, you can add them with the smbpasswd utility, change the *sysadmin* username appropriately:

```
sudo smbpasswd -a sysadmin
```

Inoltre, è necessario fornire i diritti al gruppo *Domain Admins* per consentire ad *add machine script* (e altre funzioni di amministrazione) di funzionare. Per fare ciò:

```
net rpc rights grant -U sysadmin "EXAMPLE\Domain Admins" SeMachineAccountPrivilege \
SePrintOperatorPrivilege SeAddUsersPrivilege SeDiskOperatorPrivilege \
SeRemoteShutdownPrivilege
```

9. Dovrebbe ora essere possibile unire i client Windows al dominio come in un dominio NT4 in esecuzione su un server Windows.

5.2. Controller di dominio di backup

Con la presenza di un controller di dominio primario (PDC) all'interno delle rete è utile avere anche un controller di dominio di backup (BDC). In questo modo i client potranno autenticarsi anche nel caso in cui il PDC non sia più disponibile.

Quando si configura Samba come BDC, è necessario avere un metodo di sincronizzazione delle informazioni sugli account con il PDC. A questo scopo è possibile usare scp, rsync oppure LDAP come backend *passdb*.

Il metodo migliore per sincronizzare le informazioni sugli account consiste nell'usare LDAP, poiché entrambi i controller di dominio possono usare le stesse informazioni in tempo reale. Configurare un server LDAP potrebbe essere troppo complicato per un esiguo numero di utenti e computer. Per maggiori informazioni, consultare *Sezione 2, «Samba e LDAP» [118]*.

1. Installare samba e libpam-smbpass. Da un terminale digitare:

```
sudo apt-get install samba libpam-smbpass
```

2. Modificare il file `/etc/samba/smb.conf` e togliere il commento a quanto segue nella sezione *[global]*:

```
workgroup = ESEMPIO
...
security = user
```

3. Nella sezione *Domains* togliere il commento o aggiungere quanto segue:

```
domain logons = yes
domain master = no
```

4. Assicurarsi che un utente abbia i permessi di lettura sui file in `/var/lib/samba`. Per esempio, per consentire agli utenti del gruppo *admin* di eseguire scp sui file, digitare:

```
sudo chgrp -R admin /var/lib/samba
```

5. Sincronizzare gli account utente usando scp per copiare la directory `/var/lib/samba` dal PDC:

```
sudo scp -r NOME_UTENTE@PDC:/var/lib/samba /var/lib
```

Sostituire *NOME_UTENTE* con un nome utente valido e *PDC* con il nome host o l'indirizzo IP del controller di dominio primario.

6. Riavviare samba:

```
sudo restart smbd
sudo restart nmbd
```

È possibile verificare se il controller di dominio di backup è funzionante fermando il demone Samba sul PDC e quindi cercando di eseguire l'accesso su un client Windows all'interno del dominio.

È utile ricordare anche che se è stata configurata l'opzione *logon home* come directory sul PDC e quest'ultimo non è più disponibile, anche l'accesso al drive *home* degli utenti non lo sarà. Per questo motivo è utile configurare *logon home* affinché sia posizionato in un server di file separato da PDC e BDC.

5.3. Risorse

- Per delle configurazioni più dettagliate riguardo Samba, consultare *Samba HOWTO Collection*[21]
- La guida è disponibile anche in *formato cartaceo*[22].
- Il libro *Using Samba*[23] di O'Reilly è un'altra buona lettura.
- Il *capitolo 4*[24] della «Samba HOWTO Collection» spiega come configurare un controller di dominio primario.

[21] http://samba.org/samba/docs/man/Samba-HOWTO-Collection/
[22] http://www.amazon.com/exec/obidos/tg/detail/-/0131882228
[23] http://www.oreilly.com/catalog/9780596007690/
[24] http://samba.org/samba/docs/man/Samba-HOWTO-Collection/samba-pdc.html

- Il *capitolo 5*[25] della «Samba HOWTO Collection» spiega come configurare un controller di dominio di backup.
- La pagina *su Samba*[26] della documentazione.

[25] http://us3.samba.org/samba/docs/man/Samba-HOWTO-Collection/samba-bdc.html
[26] https://help.ubuntu.com/community/Samba

6. Integrare Samba con Active Directory

6.1. Accedere a una condivisione Samba

Un altro uso di Samba consiste nell'integrarlo all'interno di una rete Windows esistente. Una volta parte di un dominio Active Directory, Samba può fornire servizi di file e stampa agli utenti AD.

The simplest way to join an AD domain is to use Likewise-open. For detailed instructions see the *Likewise Open Installation and Administration Guide*[27].

Once part of the Active Directory domain, enter the following command in the terminal prompt:

```
sudo apt-get install samba smbfs smbclient
```

Aprire il file `/etc/samba/smb.conf` e modificare quanto segue:

```
workgroup = EXAMPLE
...
security = ads
realm = EXAMPLE.IT
...
idmap backend = lwopen
idmap uid = 50-9999999999
idmap gid = 50-9999999999
```

Riavviare samba affinché le nuove impostazioni abbiano effetto:

```
sudo restart smbd
sudo restart nmbd
```

Dovrebbe essere ora possibile accedere qualsiasi condivisione Samba da un client Windows. Assicurarsi comunque di concedere agli utenti o ai gruppi AD accesso alla directory condivisa. Per maggiori informazioni, consultare *Sezione 4, «Sicurezza di un server di file e di stampa Samba» [284]*.

6.2. Accedere a una condivisione Windows

Ora che il server Samba è parte del dominio Active Directory, è possibile accedere a qualsiasi condivisione server di Windows:

- Per montare una condivisione file di Windows, in un terminale digitare quanto segue:

  ```
  mount.cifs //fs01.example.it/share mount_point
  ```

  È possibile accedere alle condivisioni su computer non facenti parte del dominio AD, ma sarà necessario fornire un nome utente e una password.

[27] http://www.likewise.com/resources/documentation_library/manuals/open/likewise-open-guide.html

- Per montare la condivisione durante la fase di avvio, aggiungere una voce al file `/etc/fstab`, per esempio:

```
//192.168.0.5/share /mnt/windows cifs auto,username=steve,password=secret,rw 0        0
```

- Un altro modo per copiare i file da un server Windows consiste nell'usare l'utilità smbclient. Per elencare i file presenti in una condivisione Windows:

```
smbclient //fs01.example.it/share -k -c "ls"
```

- Per copiare un file da una condivisione, digitare:

```
smbclient //fs01.example.com/share -k -c "get file.txt"
```

  In questo modo si copierà il file `file.txt` nella directory corrente.

- Per copiare una file nella condivisione:

```
smbclient //fs01.example.it/share -k -c "put /etc/hosts hosts"
```

  In questo modo il file `/etc/hosts` verrà copiato in `//fs01.example.com/share/hosts`.

- L'opzione *-c* usata nei comandi precedenti consente di eseguire il comando smbclient in una sola volta. Questo è utile all'interno di script e per altre operazioni sui file. Per accedere al prompt *smb: \>*, un prompt simile a quello di FTP dove è possibile svolgere normali operazioni su file e directory, digitare:

```
smbclient //fs01.example.it/share -k
```

Sostituire tutte le occorrenze di *fs01.example.it/share*, *//192.168.0.5/share*, *username=steve,password=secret* e *file.txt* con l'indirizzo IP del proprio server, il nome nome host, il nome della condivisione, il nome del file e il nome utente e la password dell'utente a cui è consentito accedere alla condivisione.

6.3. Risorse

For more smbclient options see the man page: **man smbclient**, also available *online*[28].

The mount.cifs *man page*[29] is also useful for more detailed information.

La pagina *su Samba*[30] della documentazione.

[28] http://manpages.ubuntu.com/manpages/precise/en/man1/smbclient.1.html
[29] http://manpages.ubuntu.com/manpages/precise/en/man8/mount.cifs.8.html
[30] https://help.ubuntu.com/community/Samba

Capitolo 19. Backup

È possibile eseguire dei backup delle installazioni di Ubuntu in molti modi diversi. La fase più importante è comunque quella della *pianificazione*: di cosa eseguire il backup, dove salvarlo e come ripristinarlo.

Questa sezione descrive diversi metodi per compiere queste attività.

1. Script shell

One of the simplest ways to backup a system is using a *shell script*. For example, a script can be used to configure which directories to backup, and pass those directories as arguments to the tar utility, which creates an archive file. The archive file can then be moved or copied to another location. The archive can also be created on a remote file system such as an *NFS* mount.

The tar utility creates one archive file out of many files or directories. tar can also filter the files through compression utilities, thus reducing the size of the archive file.

1.1. Semplice script shell

Il seguente script utilizza tar per creare un archivio su un file system remoto NFS. Il nome dell'archivio è determinato utilizzando delle utilità a riga di comando aggiuntive.

```
#!/bin/sh
####################################
#
# Backup to NFS mount script.
#
####################################

# What to backup.
backup_files="/home /var/spool/mail /etc /root /boot /opt"

# Where to backup to.
dest="/mnt/backup"

# Create archive filename.
day=$(date +%A)
hostname=$(hostname -s)
archive_file="$hostname-$day.tgz"

# Print start status message.
echo "Backing up $backup_files to $dest/$archive_file"
date
echo

# Backup the files using tar.
tar czf $dest/$archive_file $backup_files

# Print end status message.
echo
echo "Backup finished"
date

# Long listing of files in $dest to check file sizes.
ls -lh $dest
```

- *$backup_files*: una variabile con le directory di cui si vuole fare una copia. L'elenco va modificato come desiderato.
- *$day:* a variable holding the day of the week (Monday, Tuesday, Wednesday, etc). This is used to create an archive file for each day of the week, giving a backup history of seven days. There are other ways to accomplish this including using the date utility.
- *$hostname*: variabile contenente il nome host *breve* del sistema. Usare il nome dell'host nel nome dell'archivio, consente di avere backup giornalieri di diversi sistemi in una sola directory.
- *$archive_file*: il nome completo dell'archivio.
- *$dest:* destination of the archive file. The directory needs to be created and in this case *mounted* before executing the backup script. See *Sezione 2, «NFS (Network File System)» [227]* for details of using *NFS*.
- *messaggi:* messaggi opzionali stampati sulla console usando echo.
- *tar czf $dest/$archive_file $backup_files*: il comando tar usato per creare l'archivio.
  - *c*: crea l'archivio.
  - *z*: passa l'archivio attraverso l'utilità di compressione gzip.
  - *f:* output to an archive file. Otherwise the tar output will be sent to STDOUT.
- *ls -lh $dest*: istruzione opzionale che stampa un elenco lungo (-*l*) in un formato leggibile (-*h*) della directory di destinazione. È utile per controllare la dimensione dell'archivo. Questa verifica non dovrebbe sostituire la verifica dell'archivio.

This is a simple example of a backup shell script; however there are many options that can be included in such a script. See *Sezione 1.4, «Riferimenti» [300]* for links to resources providing more in-depth shell scripting information.

1.2. Eseguire lo script

1.2.1. Esecuzione da terminale

Il metodo più facile per eseguire lo script di backup è quello di copiare il contenuto dello script in un file, `backup.sh` per esempio, ed eseguirlo in un terminale:

```
sudo bash backup.sh
```

È un ottimo modo per provare lo script e assicurarsi che funzioni correttamente.

1.2.2. Esecuzione con cron

L'utilità cron può essere usata per automatizzare l'esecuzione dello script. Il demone cron consente l'esecuzione di script o comandi a un determinato orario e data.

L'applicazione cron è configurata attraverso delle voci in un file `crontab` file. I file `crontab` sono separati in campi:

```
# m h dom mon dow   comando
```

- *m:* minute the command executes on, between 0 and 59.
- *h:* hour the command executes on, between 0 and 23.
- *dom*: giorno del mese di esecuzione del comando.
- *mon:* the month the command executes on, between 1 and 12.
- *dow:* the day of the week the command executes on, between 0 and 7. Sunday may be specified by using 0 or 7, both values are valid.
- *comando*: il comando da eseguire.

Per aggiungere o modificare voci in un file `crontab`, dovrebbe essere usato il comando crontab -e, i contenuti di un file `crontab` possono essere visualizzati usando il comando crontab -l.

Per eseguire lo script backup.sh usando cron, in un terminale digitare quanto segue:

```
sudo crontab -e
```

Usare sudo con il comando crontab -e, modifica il crontab dell'utente *root*. Questo è necessario nel caso in cui si stiano eseguendo copie di backup di file accessibili solo dall'utente root.

Aggiungere quanto segue al file `crontab`:

```
# m h dom mon dow   command
0 0 * * * bash /usr/local/bin/backup.sh
```

Lo script backup.sh verrà eseguito ogni giorno alle 12.00 AM.

The backup.sh script will need to be copied to the `/usr/local/bin/` directory in order for this entry to execute properly. The script can reside anywhere on the file system, simply change the script path appropriately.

For more in-depth crontab options see *Sezione 1.4, «Riferimenti» [300]*.

1.3. Ripristinare l'archivio

Una volta creato un archivio, è importante verificarlo, elencandone i contenuti oppure, ed è la scelta migliore, *ripristinare* un file dall'archivio.

- To see a listing of the archive contents. From a terminal prompt type:

  ```
  tar -tzvf /mnt/backup/host-lunedì.tgz
  ```

- Per ripristinare un file dall'archivio in una directory diversa, digitare:

```
tar -xzvf /mnt/backup/host-lunedì.tgz -C /tmp etc/hosts
```

L'opzione *-C* di tar redirige i file estratti nella directory specificata. L'esempio precedente estrarrà il file `/etc/hosts` in `/tmp/etc/hosts`. La struttura della directory viene quindi ricreata da tar.

Notare anche che il simbolo "/" iniziale del percorso in cui ripristinare è stato tralasciato.

- Per ripristinare tutti i file presenti nell'archivio, digitare:

```
cd /
sudo tar -xzvf /mnt/backup/host-lunedì.tgz
```

In questo modo verranno sovrascritti i file attualmente presenti nel file system.

1.4. Riferimenti

- Per maggiori informazioni riguardo lo script da shell, consultare la *Advanced Bash-Scripting Guide*[1]
- Il libro *Teach Yourself Shell Programming in 24 Hours*[2] è disponibile in linea ed è un'ottima risorsa per lo script da shell.
- La pagina della *della documentazione in linea su cron*[3] contiene ulteriori dettagli sulle opzioni avanzate di cron.
- Per maggiori informazioni sulle opzioni del comando tar, consultare il *manuale in linea di tar*[4].
- La pagina inglese di Wikipedia *Backup Rotation Scheme*[5] contiene informazioni sugli schemi di backup.
- Questo script utilizza tar per creare l'archivio, ma esistono diverse altre utilità a riga di comando che possono essere usate, per esempio:
  - *cpio*[6]: usata per copiare file da e verso degli archivi.
  - *dd*[7]: part of the coreutils package. A low level utility that can copy data from one format to another.
  - *rsnapshot*[8]: a file system snapshot utility used to create copies of an entire file system.
  - *rsync*[9]: a flexible utility used to create incremental copies of files.

[1] http://tldp.org/LDP/abs/html/
[2] http://safari.samspublishing.com/0672323583
[3] http://wiki.ubuntu-it.org/AmministrazioneSistema/Cron
[4] http://www.gnu.org/software/tar/manual/index.html
[5] http://en.wikipedia.org/wiki/Backup_rotation_scheme
[6] http://www.gnu.org/software/cpio/
[7] http://www.gnu.org/software/coreutils/
[8] http://www.rsnapshot.org/
[9] http://www.samba.org/ftp/rsync/rsync.html

2. Rotazione degli archivi

The shell script in *Sezione 1, «Script shell» [297]* only allows for seven different archives. For a server whose data doesn't change often, this may be enough. If the server has a large amount of data, a more complex rotation scheme should be used.

2.1. Rotazione degli archivi NFS

In this section, the shell script will be slightly modified to implement a grandfather-father-son rotation scheme (monthly-weekly-daily):

- La rotazione eseguirà un backup *giornaliero* dalla domenica al venerdì.
- Il sabato, viene eseguito un backup *settimanale* consentendo di avere così quattro backup settimanali al mese.
- Il backup *mensile* è eseguito il primo giorno del mese, ruotando due backup mensili se il mese è pari o dispari.

Questo è il nuovo script:

```
#!/bin/bash
####################################
#
# Backup to NFS mount script with
# grandfather-father-son rotation.
#
####################################

# What to backup.
backup_files="/home /var/spool/mail /etc /root /boot /opt"

# Where to backup to.
dest="/mnt/backup"

# Setup variables for the archive filename.
day=$(date +%A)
hostname=$(hostname -s)

# Find which week of the month 1-4 it is.
day_num=$(date +%d)
if (( $day_num <= 7 )); then
        week_file="$hostname-week1.tgz"
elif (( $day_num > 7 && $day_num <= 14 )); then
        week_file="$hostname-week2.tgz"
elif (( $day_num > 14 && $day_num <= 21 )); then
        week_file="$hostname-week3.tgz"
elif (( $day_num > 21 && $day_num < 32 )); then
        week_file="$hostname-week4.tgz"
fi
```

```
# Find if the Month is odd or even.
month_num=$(date +%m)
month=$(expr $month_num % 2)
if [ $month -eq 0 ]; then
        month_file="$hostname-month2.tgz"
else
        month_file="$hostname-month1.tgz"
fi

# Create archive filename.
if [ $day_num == 1 ]; then
 archive_file=$month_file
elif [ $day != "Saturday" ]; then
        archive_file="$hostname-$day.tgz"
else
 archive_file=$week_file
fi

# Print start status message.
echo "Backing up $backup_files to $dest/$archive_file"
date
echo

# Backup the files using tar.
tar czf $dest/$archive_file $backup_files

# Print end status message.
echo
echo "Backup finished"
date

# Long listing of files in $dest to check file sizes.
ls -lh $dest/
```

Lo script può essere eseguito attraverso gli stessi metodi descritti in *Sezione 1.2, «Eseguire lo script» [298]*.

It is good practice to take backup media off-site in case of a disaster. In the shell script example the backup media is another server providing an NFS share. In all likelihood taking the NFS server to another location would not be practical. Depending upon connection speeds it may be an option to copy the archive file over a WAN link to a server in another location.

Another option is to copy the archive file to an external hard drive which can then be taken off-site. Since the price of external hard drives continue to decrease, it may be cost-effective to use two drives for each archive level. This would allow you to have one external drive attached to the backup server and one in another location.

2.2. Dispositivi a nastro

A tape drive attached to the server can be used instead of an NFS share. Using a tape drive simplifies archive rotation, and makes taking the media off-site easier as well.

When using a tape drive, the filename portions of the script aren't needed because the data is sent directly to the tape device. Some commands to manipulate the tape are needed. This is accomplished using mt, a magnetic tape control utility part of the cpio package.

Questo è lo script modificato per l'uso di un dispositivo a nastro:

```
#!/bin/bash
####################################
#
# Backup to tape drive script.
#
####################################

# What to backup.
backup_files="/home /var/spool/mail /etc /root /boot /opt"

# Where to backup to.
dest="/dev/st0"

# Print start status message.
echo "Backing up $backup_files to $dest"
date
echo

# Make sure the tape is rewound.
mt -f $dest rewind

# Backup the files using tar.
tar czf $dest $backup_files

# Rewind and eject the tape.
mt -f $dest rewoffl

# Print end status message.
echo
echo "Backup finished"
date
```

Il nome del device predefinito per un dispositivo a nastro SCSI è `/dev/st0`, utilizzare il percorso al device appropriato per il proprio sistema.

Ripristinare i dati da un dispositivo a nastro funziona allo stesso modo di ripristinare da un file. Riavvolgere il nastro e usare il percorso del dispositivo al posto del percorso al file. Per esempio, per ripristinare il file `/etc/hosts` in `/tmp/etc/hosts`:

```
mt -f /dev/st0 rewind
tar -xzf /dev/st0 -C /tmp etc/hosts
```

3. Bacula

Bacula is a backup program enabling you to backup, restore, and verify data across your network. There are Bacula clients for Linux, Windows, and Mac OS X - making it a cross-platform network wide solution.

3.1. Panoramica

Bacula is made up of several components and services used to manage which files to backup and backup locations:

- Bacula Director: un servizio che controlla tutte le operazioni di backup, ripristino, verifica e di archiviazione.
- Bacula Console: un'applicazione che consente di comunicare con "Director". Sono disponibili tre versioni:
  - Versione testuale per la riga di comando.
  - Versione grafica per GNOME basata su GTK+.
  - Interfaccia wxWidgets.
- Bacula File: conosciuta anche come Bacula Client. Questa applicazione è installata nei computer di cui deve essere fatto il backup ed è responsabile dei dati richiesti dal Director.
- Bacula Storage: il programma che esegue l'archiviazione e il ripristino sul dispositivo fisico.
- Bacula Catalog: responsabile per mantenere l'indice dei file e il database di tutti i file, consentendo una facile localizzazione e ripristino. "Catalog" supporta tre diversi database: MySQL, PostgreSQL e SQLite.
- Bacula Monitor: consente di monitorare i demoni "Director", "File" e "Storage". Attualmente "Monitor" è disponibile solo come applicazione GTK+.

Questi servizi e applicazioni possono essere eseguiti su molteplici server e client oppure possono essere installati su un solo computer se deve essere eseguito il backup di un singolo disco o volume.

3.2. Installazione

If using MySQL or PostgreSQL as your database, you should already have the services available. Bacula will not install them for you.

Ci sono molteplici pacchetti che contengono i diversi componenti di Bacula. Per installare Bacula, in un terminale, digitare:

```
sudo apt-get install bacula
```

In modo predefinito, installando il pacchetto bacula viene usato un database MySQL per "Catalog". Se si vuole usare SQLite oppure PostgreSQL, installare bacula-director-sqlite3 o bacula-director-pgsql rispettivamente.

Durante il processo di installazione viene chiesto di fornire delle credenziali per l'*amministratore* del database e per il *proprietario* del database *bacula*. L'amministratore del database deve avere i diritti apporpriati per poter creare un database. Per maggiori informazioni, consultare la *Sezione 1, «MySQL» [207]*.

3.3. Configurazione

I file di configurazione di Bacula sono formattati in base alle *risorse* composte da *direttive* marcate da parantesi «{}». Ogni compoenente di Bacula dispone di un file nella directory `/etc/bacula`.

I diversi componenti di Bacula devono autorizzarsi tra di loro. Questo è fatto usando la direttiva *password*. Per esempio, la risorsa password di *Storage* nel file `/etc/bacula/bacula-dir.conf` deve corrispondere alla risorsa password di *Director* nel file `/etc/bacula/bacula-sd.conf`.

In modo predefinito, il lavoro di backup chiamato *Client1* è confoigurato per archiviare il "Catalog" di Bacula. Se si intende usare il server per eseguire il backup di più di un client, è necessario modificare il nome del lavoro con qualche cosa di più descrittivo. Per fare questo, modificare il file `/etc/bacula/bacula-dir.conf`:

```
#
# Define the main nightly save backup job
#   By default, this job will back up to disk in
Job {
  Name = "BackupServer"
  JobDefs = "DefaultJob"
  Write Bootstrap = "/var/lib/bacula/Client1.bsr"
}
```

L'esempio precedente modifica il nome del lavoro in *BackupServer*, in corrispondenza del nome host del computer. Sostituire «BackupServer» con il nome host appropriato o un altro nome descrittivo.

Console può essere usato per interrogare *Director* riguardo i lavori, ma per poter usare "Console" con un utente *non-root*, l'utente deve essere nel gruppo *bacula*. Per aggiungere un utente al gruppo "bacula", in un terminale, digitare:

```
sudo adduser NOME_UTENTE bacula
```

Sostituire *NOME_UTENTE* con il vero nome utente. Inoltre, se si sta aggiungendo l'utente corrente al gruppo, è necessario terminare la sessione e rientrarvi affinché le modifiche abbiano effetto.

3.4. Backup locale

Questa sezione descrive come eseguire un backup di specifiche directory di un singolo host in un dispositivo a nastro locale.

- Per prima cosa, *Storage* deve essere configurato. Modificare `/etc/bacula/bacula-sd.conf`:

```
Device {
  Name = "Tape Drive"
  Device Type = tape
  Media Type = DDS-4
  Archive Device = /dev/st0
  Hardware end of medium = No;
  AutomaticMount = yes;               # when device opened, read it
  AlwaysOpen = Yes;
  RemovableMedia = yes;
  RandomAccess = no;
  Alert Command = "sh -c 'tapeinfo -f %c | grep TapeAlert'"
}
```

  The example is for a *DDS-4* tape drive. Adjust the «Media Type» and «Archive Device» to match your hardware.

  È possibile anche de-commentare uno degli altri file di esempio.

- Una volta modificato il file `/etc/bacula/bacula-sd.conf`, il demone Storage deve essere riavviato:

```
sudo /etc/init.d/bacula-sd restart
```

- Ora aggiungere una risorsa *Storage* in `/etc/bacula/bacula-dir.conf` per usare il nuovo "Device":

```
# Definition of "Tape Drive" storage device
Storage {
  Name = TapeDrive
  # Do not use "localhost" here
  Address = backupserver               # N.B. Use a fully qualified name here
  SDPort = 9103
  Password = "Cv70F6pf1t6pBopT4vQOnigDrR0v3LT3Cgkiyjc"
  Device = "Tape Drive"
  Media Type = tape
}
```

  La direttiva *Address* deve essere il "Fully Qualified Domain Name" (FQDN) del server. Modificare quindi *backupserver* col nome host attuale.

  Inoltre, assicurarsi che la direttiva *Password* corrisponda alla stringa in `/etc/bacula/bacula-sd.conf`.

- Creare un nuovo *FileSet*, per determinare di quali directory eseguire il backup:

```
# LocalhostBacup FileSet.
FileSet {
  Name = "LocalhostFiles"
  Include {
```

```
    Options {
      signature = MD5
      compression=GZIP
    }
    File = /etc
    File = /home
  }
}
```

This *FileSet* will backup the `/etc` and `/home` directories. The *Options* resource directives configure the FileSet to create an MD5 signature for each file backed up, and to compress the files using GZIP.

- Creare una nuova sezione *Schedule* per il lavoro di backup:

```
# LocalhostBackup Schedule -- Daily.
Schedule {
  Name = "LocalhostDaily"
  Run = Full daily at 00:01
}
```

Il lavoro verrà eseguito ogni giorno alle 00.01. Sono comunque disponibili molte altre opzioni di schedulatura.

- Infine creare il *Job*:

```
# Localhost backup.
Job {
  Name = "LocalhostBackup"
  JobDefs = "DefaultJob"
  Enabled = yes
  Level = Full
  FileSet = "LocalhostFiles"
  Schedule = "LocalhostDaily"
  Storage = TapeDrive
  Write Bootstrap = "/var/lib/bacula/LocalhostBackup.bsr"
}
```

Questo lavoro crearà un backup *Full* (completo) ogni giorni sul dispositivo a nastro.

- Ogni nastro usato deve avere una *Label*. Se il nastro corrente ne è sprovvisto, Bacula invierà un'email. Per aggiungere un'etichetta a un nastro usando Console, in un terminale, digitare:

```
bconsole
```

- Al prompt di "Console" digitare:

```
label
```

- Viene quindi chiesta la risorsa *Storage*:

```
Automatically selected Catalog: MyCatalog
Using Catalog "MyCatalog"
The defined Storage resources are:
     1: File
     2: TapeDrive
Select Storage resource (1-2):2
```

- Inserire il nome del nuovo *Volume* (volume):

```
Enter new Volume name: Sunday
Defined Pools:
     1: Default
     2: Scratch
```

  Sostituire *Sunday* con l'etichetta desiderata.

- Ora selezionare *Pool*:

```
Select the Pool (1-2): 1
Connecting to Storage daemon TapeDrive at backupserver:9103 ...
Sending label command for Volume "Sunday" Slot 0 ...
```

Bacula è ora configurato per eseguire backup del host locale su un dispositivo a nastro.

3.5. Risorse

- Per maggiori informazioni sulle opzioni di configurazione di *Bacula*, consultare il *manuale di Bacula*[10]
- Il sito *di Bacula*[11] contiene le ultime notizie dello sviluppo di Bacula.
- Consultare anche la *documentazione di bacula online*[12].

[10] http://www.bacula.org/en/rel-manual/index.html
[11] http://www.bacula.org/
[12] https://help.ubuntu.com/community/Bacula

Capitolo 20. Virtualizzazione

La virtualizzazione, al giorno d'oggi, viene utilizzata in diversi ambienti e situazioni. Dal punto di vista dello sviluppatore, la virtualizzazione offre un ambiente sicuro dove poter eseguire qualsiasi tipo di sviluppo, senza compromettere l'ambiente di lavoro. Per l'amministratore di sistema, è possibile usare la virtualizzazione per separare facilmente i propri servizi e spostarli in base alle richieste.

The default virtualization technology supported in Ubuntu is KVM. KVM requires virtualization extensions built into Intel and AMD hardware. Xen is also supported on Ubuntu. Xen can take advantage of virtualization extensions, when available, but can also be used on hardware without virtualization extensions. Qemu is another popular solution for hardware without virtualization extensions.

1. libvirt

La libreria libvirt è utilizzata per interfacciarsi con differenti tecnologie di virtualizzazione. Prima di iniziare a utilizzare libvirt è utile accertarsi che il proprio hardware supporti le estensioni di virtualizzazione necessarie per KVM. In un terminale, digitare quanto segue:

```
kvm-ok
```

Verrà stampato un messaggio che indica se la CPU *supporta* o *non supporta* la virtualizzazione hardware.

Nella maggior parte dei processori che supportano la virtualizzazione è necessario attivarla attraverso un'opzione nel BIOS.

1.1. Rete virtuale

Esistono diversi modi per consentire accesso alla rete esterna a una macchina virtuale. La configurazione di rete predefinita è *usermode*, che utilizza il protocollo SLIRP e il traffico è passato attraverso l'interfaccia dell'host verso la rete esterna.

Affinché gli host esterni possano accedere i servizi su una macchina virtuale, è necessario configurare un *bridge*. Questo consente alle interfacce virtuali di connettersi alla rete esterna attraverso l'interfaccia fisica, facendole apparire come normali host al resto della rete. Per informazioni su come impostare un bridge, consultare *Sezione 1.4, «Bridging» [41]*.

1.2. Installazione

Per installare i pacchetti necessari, da un terminale digitare:

```
sudo apt-get install kvm libvirt-bin
```

Dopo aver installato libvirt-bin, l'utente usato per la gestione delle macchine virtuali deve essere aggiunto al gruppo *libvirtd*. In questo modo, all'utente è garantito accesso alle configurazioni avanzate di rete.

In un terminale digitare:

```
sudo adduser $USER libvirtd
```

Se l'utente scelto è quello corrente, è necessario terminare la sessione e ri-accedervi affinché le modifiche abbiano effetto.

È ora possibile installare un sistema operativo *ospite*. La procedura di installazione di una macchina virtuale è la stessa di un sistema operativo normale ed è quindi necessario automatizzare la procedura oppure avere una tastiera e uno schermo collegati al computer.

Nel caso delle macchine virtuali, un'interfaccia grafica è analoga all'uso di una tastiera e di un mouse. Invece di installare un'interfaccia grafica, è possibile usare virt-viewer per connettersi alla console di una macchina virtuale via VNC. Per maggiori informazioni, consultare la *Sezione 1.6, «Visualizzatore di macchine virtuali» [314]*.

There are several ways to automate the Ubuntu installation process, for example using preseeds, kickstart, etc. Refer to the *Ubuntu Installation Guide*[1] for details.

Un altro metodo per installare una macchina virtuale Ubuntu consiste nell'usare l'applicazione ubuntu-vm-builder. ubuntu-vm-builder consente di impostare partizioni avanzate, eseguire script personalizzati post-installazione, ecc... Per maggiori informazioni, consultare *Sezione 2, «JeOS e vmbuilder» [316]*

Libvirt can also be configured work with Xen. For details, see the Xen Ubuntu community page referenced below.

1.3. virt-install

virt-install is part of the virtinst package. To install it, from a terminal prompt enter:

```
sudo apt-get install virtinst
```

Durante l'uso di virt-install sono disponibili molte azioni, per esempio:

```
sudo virt-install -n web_devel -r 256 \
--disk path=/var/lib/libvirt/images/web_devel.img,bus=virtio,size=4 -c \
jeos.iso --accelerate --network network=default,model=virtio \
--connect=qemu:///system --vnc --noautoconsole -v
```

- *-n web_devel*: il nome della nuova macchine virtuale usato in questo esempio sarà *web_devel*.
- *-r 256:* specifies the amount of memory the virtual machine will use in megabytes.
- *--disk path=/var/lib/libvirt/images/web_devel.img,size=4:* indicates the path to the virtual disk which can be a file, partition, or logical volume. In this example a file named `web_devel.img` in the /var/lib/libvirt/images/ directory, with a size of 4 gigabytes, and using *virtio* for the disk bus.
- *-c jeos.iso*: il file usato come CD-ROM virtuale. Il file può essere un file ISO o il percorso al device del CD-ROM nell'host.
- *--accelerate*: abilita letecnologie di accelerazione nel kernel.
- *--network* provides details related to the VM's network interface. Here the *default* network is used, and the interface model is configured for *virtio*.
- *--vnc*: esporta la console virtuale usando VNC.
- *--noautoconsole*: non si collegherà automaticamente alla console della macchina virtuale.

[1] https://help.ubuntu.com/12.04/installation-guide/

- *-v*: crea un ospite completamente virtualizzato.

Una volta lanciata virt-install è possibile collegarsi alla console della macchina virtuale utilizzando, localmente, un'interfaccia grafica oppure l'utilità virt-viewer.

1.4. virt-clone

L'applicazione virt-clone può essere usata per copiare una macchina virtuale in un'altra, per esempio:

```
sudo virt-clone -o web_devel -n database_devel -f /path/to/database_devel.img \
--connect=qemu:///system
```

- *-o*: macchina virtuale originale.
- *-n*: nome della nuova macchina virtuale.
- *-f*: percorso al file, volume logico o partizione da usare per la nuova macchina virtuale.
- *--connect*: specifica a quale hypervisor collegarsi.

Usare anche le opzioni *-d* o *--debug* per risolvere i problemi che potrebbero verificarsi con virt-clone.

Sostituire *web_devel* e *database_devel* con i nomi delle macchine virtuali appropriati.

1.5. Gestire la macchina virtuale

1.5.1. virsh

Sono disponibili diverse utilità per la gestione delle macchine virtuali e di libvirt. L'utilità virsh può essere utilizzata dalla riga di comando. Alcuni esempi:

- Per elencare le macchine virtuali in esecuzione:

```
virsh -c qemu:///system list
```

- Per avviare una macchina virtuale:

```
virsh -c qemu:///system start web_devel
```

- Similmente, per lanciare una macchina virtuale durante l'avvio del computer:

```
virsh -c qemu:///system autostart web_devel
```

- Riavviare una macchina virtuale con:

```
virsh -c qemu:///system reboot web_devel
```

- Lo *stato* di una macchina virtuale può essere salvato in un file per poterlo ripristinare successivamente. Il seguente comando salva lo stato della macchina virtuale in un file nominato in base alla data.

```
virsh -c qemu:///system save web_devel web_devel-022708.state
```

Una volta salvata, la macchina virtuale non sarà più in esecuzione.

- Per ripristinare una macchina virtuale:

```
virsh -c qemu:///system restore web_devel-022708.state
```

- Per arrestare una macchina virtuale:

```
virsh -c qemu:///system shutdown web_devel
```

- Per montare un CD-ROM in una macchina virtuale, digitare:

```
virsh -c qemu:///system attach-disk web_devel /dev/cdrom /media/cdrom
```

Nell'esempio precedente, sostituire *web_devel* con il nome della macchina virtuale appropriata e `web_devel-022708.state` con un nome file descrittivo.

1.5.2. Gestore macchina virtuale

Il pacchetto virt-manager contiene un'utilità grafica per gestire le macchie virtuali locali e remote. Per installare virt-manager digitare:

```
sudo apt-get install virt-manager
```

Dato che virt-manager richiede un'interfaccia grafica (GUI), è raccomandato installarlo su una workstation o una postazione di prova invece che un server di produzione. Per connettersi al servizio libvirt locale:

```
virt-manager -c qemu:///system
```

È possibile collegarsi al servizio libvirt in esecuzione su un altro host digitando, in un terminale:

```
virt-manager -c qemu+ssh://virtnode1.mydomain.com/system
```

L'esempio precedente assume che la connessione SSH tra il sistema di gestione e virtnode1.mydomain.com sia già configurata e utilizzi le chiavi SSH per l'autenticazione. Le *chiavi* SSH sono necessarie perché libvirt invia il prompt password a un altro processo. Per maggiori informazioni sulla configurazione di SSH, consultare la *Sezione 1, «Server OpenSSH» [81]*

1.6. Visualizzatore di macchine virtuali

L'applicazione virt-viewer consente di collegarsi alla console di una macchina virtuale. virt-viewer non richiede un'interfaccia grafica per interagire con la macchina virtuale.

Per installare virt-viewer, da un terminale digitare:

```
sudo apt-get install virt-viewer
```

Una volta installata e in esecuzione, è possibile connettersi alla console della macchina virtuale digitando:

```
virt-viewer -c qemu:///system web_devel
```

Analogamente a virt-manager, virt-viewer può collegarsi a un host remoto utilizzando *SSH* con chiave di autenticazione:

```
virt-viewer -c qemu+ssh://virtnode1.miodominio.it/system web_devel
```

Assicurarsi di sostituire *web_devel* con il nome corretto della macchina virtuale.

Se configurato per usare un'interfaccia di rete *bridged*, è anche possibile impostare accesso SSH alla macchina virtuale. Per maggiori informazioni, consultare *Sezione 1, «Server OpenSSH» [81]* e *Sezione 1.4, «Bridging» [41]*.

1.7. Risorse

- Per maggiori informazioni, consultare il sito web di *KVM*[2].
- Per maggiori informazioni su libvirt, consultare *il sito web di libvirt*[3]
- Il sito di *Virtual Machine Manager*[4] dispone di ulteriori informazioni riguardo lo sviluppo di virt-manager.
- È anche possibile passare nel canale IRC *#ubuntu-virt* su *freenode*[5] per discutere delle tecnologie di virtualizzazione in Ubuntu.
- Un'altra ottima risorsa è la *documentazione online*[6] riguardo KVM.
- For information on Xen, including using Xen with libvirt, please see the *Ubuntu Wiki Xen*[7] page.

[2] http://kvm.qumranet.com/kvmwiki
[3] http://libvirt.org/
[4] http://virt-manager.et.redhat.com/
[5] http://freenode.net/
[6] https://help.ubuntu.com/community/KVM
[7] https://help.ubuntu.com/community/Xen

2. JeOS e vmbuilder

2.1. Introduzione

2.1.1. Cos'è JeOS

Ubuntu *JeOS* (pronunciato come la parola "juice") è una variante di della versione server di Ubuntu, configurata appositamente per le applicazioni virtuali. Non è disponibile sotto forma di file ISO per CD-ROM, ma solo come opzione:

- durante l'installazione della versione server (premere *F4* alla prima schermata per scegliere l'opzione "Installa un sistema minimale" che equivale a selezionare JeOS).
- oppure può essere generato usando "vmbuilder" come descritto di seguito.

JeOS è un'installazione di Ubuntu Server Edition con un kernel appositamente configurato che contiene gli elementi basilari necessari all'esecuzione di un ambiente virtualizzato.

Ubuntu JeOS è stato progettato per sfruttare tutte quelle tecnologie chiave, relative alle prestazioni, presenti negli ultimi prodotti di virtualizzazione di VMware. La combinazione di una ridotta dimensione e prestazioni ottimizzate, assicurano che Ubuntu JeOS Edition sia in grado di offrire un uso efficiente delle risorse server in grandi produzioni virtuali.

Senza l'utilizzo di driver non necessari e ricorrendo solo ai pacchetti richiesti, gli ISV possono configurare il proprio SO di supporto proprio come desiderano. Inoltre, viene assicurato che gli aggiornamenti, di sicurezza o per miglioramenti, saranno limitati al minimo richiesto dallo specifico ambiente. Gli utenti che sviluppano soluzioni virtuali basate su JeOS, dovranno gestire meno aggiornamenti, e quindi una minor manutenzione, di quanto avrebbero dovuto fare con un'installazione server completa.

2.1.2. Cos'è vmbuilder

Utilizzando vmbuilder non è necessario scaricare un'immagine di JeOS: verranno scaricati i pacchetti necessari per creare una macchina virtuale adatta alle proprie esigenze. vmbuilder è uno script che automatizza la creazione di una macchina virtuale Linux. Gli hypervisor supportati attualmente sono KVM e Xen.

È possibile passare opzioni a riga di comando per aggiungere dei pacchetti, per rimuoverne, per scegliere la versione di Ubuntu, quale mirror, ecc... Su piattaforme hardware recenti dotate di molta memoria RAM, con tmpdir in `/dev/shm` o usando un tmpfs e un mirror locale, è possibile avere una macchina virtuale in meno di un minuto.

Introdotto come semplice script shell in Ubuntu 8.04 LTS, ubuntu-vm-builder era un semplice progetto per aiutare gli sviluppatori nel provare il codice scritto in una virtual machine senza dover ricominciare sempre da capo. Lo script è stato in seguito migliorato e Soren Hansen (l'autore dello script e lo specialista di virtualizzazione in Ubuntu virtualization, non il giocatore di golf) lo ha riscritto da capo per Intrepid in python con i seguenti obiettivi:

- Svilupparlo affinché possa essere usato anche da altre distribuzioni.
- Usare un meccanismo di plugin per tutte le interazioni di virtualizzazione per facilitare l'aggiunta di altri ambienti di virtualizzazione o una logica più complessa.
- Fornire un'interfaccia web facile da usare come opzione alla riga di comando.

I principi generali e i comandi restano sempre gli stessi.

2.2. Configurazione iniziale

Si presuppone che siano già stati installati e configurati libvirt e KVM sul computer che si intende usare. Per maggiori informazioni, consultare:

- *Sezione 1, «libvirt» [311]*
- La pagina relativa a *KVM*[8] nella documentazione (in inglese).

Si dà per assodato che si sappia utilizzare un editor di testo come nano oppure vi. In caso contrario, è possibile avere una panoramica dei vari editor di testo consultando *la documentazione di Ubuntu*[9]. Questa guida è stata scritta basandosi su KVM, ma il principio dovrebbe essere lo stesso anche per altre tecnologie di virtualizzazione.

2.2.1. Installare vmbuilder

Il nome del pacchetto da installare è python-vm-builder. In un terminale digitare:

```
sudo apt-get install python-vm-builder
```

Se si sta eseguendo la versione 8.04 è sempre possibile eseguire queste azioni usando la versione del pacchetto chiamata ubuntu-vm-builder; ci sono solo alcune modifiche nella sintassi da usare con il programma.

2.3. Definire una macchina virtuale

Definire una macchina virtuale con vmbuilder è molto facile, ma è necessario prendere in considerazione alcuni aspetti:

- Se si pianifica di fornire applicativi virtuali, non assumere che l'utente finale sappia come estendere la dimensione del disco secondo le proprie esigenze. Prendere quindi in considerazione l'utilizzo di dischi virtuali di grandi dimensioni per consentire agli applicativi di crescere o spiegare nella documentazione come allocare maggiore spazio. Potrebbe essere una buona idea salvare i dati in un sistema di archiviazione esterno.
- Dato che la memoria RAM è più facile da allocare in una MV, la dimensione della RAM dovrebbe essere impostata a un valore minimo sicuro per la propria applicazione.

[8] https://help.ubuntu.com/community/KVM
[9] http://wiki.ubuntu-it.org/Ufficio/EditorDiTesto#powereditor

Il comando vmbuilder dispone di due parametri principali: la *tecnologia di virtualizzazione (hypervisor)* e la *distribuzione* finale. Sono disponibili molti altri parametri e tutti possono essere visualizzati con il seguente comando:

```
vmbuilder kvm ubuntu --help
```

2.3.1. Parametri base

As this example is based on KVM and Ubuntu 12.04 LTS (Precise Pangolin), and we are likely to rebuild the same virtual machine multiple time, we'll invoke vmbuilder with the following first parameters:

```
sudo vmbuilder kvm ubuntu --suite precise --flavour virtual --arch i386  \
-o --libvirt qemu:///system
```

Il parametro --*suite* definisce il rilascio di Ubuntu, --*flavour* specifica di usare il kernel virtuale (quello usato per generare un'immagine JeOS), --*arch* indica di usare un computer a 32 bit, -*o* indica a vmbuilder di sovrascrivere la versione precedente della macchina virtuale e --*libvirt* aggiunge la macchina virtuale risultante tra quelle disponibili nell'ambiente di virtualizzazione.

Note:

- Data la natura delle operazioni eseguite da vmbuilder, sono necessari i privilegi di root.
- Se la macchina virtuale necessita di usare più di 3GB di RAM, è utile generare una macchina a 64 bit (--arch amd64).
- Fino a Ubuntu 8.10, il kernel virtuale era generato solo per architetture a 32 bit, per definire quindi una macchina amd64 su Hardy, usare --*flavour server*.

2.3.2. Parametri di installazione di JeOS

2.3.2.1. Rete con JeOS

2.3.2.1.1. Assegnare un indirizzo IP fisso

Come applicazione che verrà messa in produzione all'interno di reti diverse, è molto difficile conoscere la struttura attuale della rete. Per semplificare la configurazione è utile procedere come solitamente procedono i venditori di hardware di rete, assegnando un indirizzo IP fisso all'interno di una classe di rete che verrà descritta all'interno della propria documentazione. Un indirizzo nell'intervallo 192.168.0.0/255 è una buona scelta.

Per ottenere questo vengono usati i seguenti parametri:

- --*ip INDIRIZZO*: indirizzo IP (il valore predefinito è dhcp se non viene specificato nulla)
- --*hostname NAME*: Set NAME as the hostname of the guest.
- --*mask VALORE*: maschera di rete (valore predefinito: 255.255.255.0)
- --*net VALORE*: indirizzo IP net (valore predefinito: X.X.X.0)

- *--bcast VALORE*: broadcast (valore predefinito: X.X.X.255)
- *--gw INDIRIZZO*: indirizzo del gateway (valore predefinito: X.X.X.1)
- *--dns INDIRIZZO*: indirizzo server dei nomi (valore predefinito: X.X.X.1)

Si dà per scontato che i valori predefiniti siano sufficiente. Il comando diventa:

```
sudo vmbuilder kvm ubuntu --suite precise --flavour virtual --arch i386 \
-o --libvirt qemu:///system --ip 192.168.0.100 --hostname myvm
```

2.3.2.1.2. Bridging

Because our appliance will be likely to need to be accessed by remote hosts, we need to configure libvirt so that the appliance uses bridge networking. To do this add the *--bridge* option to the command:

```
sudo vmbuilder kvm ubuntu --suite precise --flavour virtual --arch i386 \
-o --libvirt qemu:///system --ip 192.168.0.100 --hostname myvm --bridge br0
```

You will need to have previously setup a bridge interface, see *Sezione 1.4, «Bridging» [41]* for more information. Also, if the interface name is different change *br0* to the actual bridge interface.

2.3.2.2. Partizionamento

Il partizionamento dell'applicativo virtuale deve prendere in considerazione cosa si intende fare. Dato che molti applicativi non avranno un sistema di archiviazione separato per i dati, usare una partizione `/var` separata è una buona idea.

Per ottenere tutto questo, vmbuilder dispone dell'opzione *--part*:

```
--part PATH
  Allows you to specify a partition table in a partition file, located at PATH. Each
  line of the partition file should specify (root first):
      mountpoint size
  where  size  is  in megabytes. You can have up to 4 virtual disks, a new disk starts
  on a line with '---'.  ie :
      root 1000
      /opt 1000
      swap 256
      ---
      /var 2000
      /log 1500
```

In questo caso, creare un file di testo `vmbuilder.partition` contenente quanto segue:

```
root 8000
swap 4000
---
```

```
/var 20000
```

Notare che vengono usate immagini disco virtuali, le dimensioni inserite sono le dimensioni massime dei volumi.

Il comando diventa quindi:

```
sudo vmbuilder kvm ubuntu --suite precise --flavour virtual --arch i386 \
-o --libvirt qemu:///system --ip 192.168.0.100 --hostname myvm --part vmbuilder.partition
```

L'uso di "\" all'interno di un comando consente di scrivere comandi su più righe.

2.3.2.3. Utente e password

È necessario anche impostare un utente e una password predefiniti e generici da poter includere nella documentazione. Successivamente verrà presentato uno script che viene eseguito al primo accesso di un utente che tra le molte cose chiederà di modificare la password. In questo esempio viene usato come nome utente *user* e *default* come password.

Per fare questo vengono usati i seguenti parametri:

- *--user NOME_UTENTE*: imposta il nome utente da aggiungere. Valore predefinito: ubuntu.
- *--name NOME_COMPLETO*: imposta il nome completo dell'utente da aggiungere. Valore predefinito: Ubuntu.
- *--pass PASSWORD*: imposta la password dell'utente: Valore predefinito: ubuntu.

Il comando ora è il seguente:

```
sudo vmbuilder kvm ubuntu --suite precise --flavour virtual --arch i386 \
-o --libvirt qemu:///system --ip 192.168.0.100 --hostname myvm --part \
vmbuilder.partition --user user --name user --pass default
```

2.3.3. Installare i pacchetti richiesti

In questo esempio verrà installato un pacchetto (Limesurvey) che accede a un database MySQL ed è dotato di un'interfaccia web. Il sistema operativo dovrà quindi aver installato:

- Apache
- PHP
- MySQL
- Server OpenSSH
- Limesurvey (un'applicazione di esempio creata appositamente)

This is done using vmbuilder by specifying the --addpkg option multiple times:

```
--addpkg PKG
```

```
  Install PKG into the guest (can be specfied multiple times)
```

Purtroppo, in base al funzionamento di vmbuilder, i pacchetti che devono porre delle domande nella fase di post-installazione non sono supportati e dovrebbero essere installati successivamente quando è possibile interagirvi. Questo è il caso di Limesurvey che verrà installato successivamente, dopo che l'utente ha eseguito l'accesso.

Altri pacchetti che pongono delle semplici domande di debconf, come mysql-server che richiede di impostare una password, possono essere installati, ma dovranno essere riconfigurati una volta eseguito l'accesso.

Se alcuni dei pacchetti che si devono installare non sono presenti nel componente "main", è necessario abilitare dei repository aggiuntivi usando le opzioni "--comp" e "--ppa":

```
--components COMP1,COMP2,...,COMPN
           A comma separated list of distro components to include (e.g. main,universe).
           This defaults to "main"
--ppa=PPA  Add ppa belonging to PPA to the vm's sources.list.
```

Limesurvey non fa parte degli archivi attualmente ed è quindi necessario specificarne l'indirizzo PPA (Personal Package Archive) così da aggiungerlo al file `/etc/apt/source.list` della macchina virtuale. Aggiungere quindi quanto segue al comando:

```
--addpkg apache2 --addpkg apache2-mpm-prefork --addpkg apache2-utils \
         --addpkg apache2.2-common --addpkg dbconfig-common --addpkg libapache2-mod-php5 \
         --addpkg mysql-client --addpkg php5-cli --addpkg php5-gd --addpkg php5-ldap \
         --addpkg php5-mysql --addpkg wwwconfig-common --addpkg mysql-server --ppa nijaba
```

2.3.4. Considerazioni sulla velocità

2.3.4.1. Cache dei pacchetti

When vmbuilder creates builds your system, it has to go fetch each one of the packages that composes it over the network to one of the official repositories, which, depending on your internet connection speed and the load of the mirror, can have a big impact on the actual build time. In order to reduce this, it is recommended to either have a local repository (which can be created using apt-mirror) or using a caching proxy such as apt-proxy. The later option being much simpler to implement and requiring less disk space, it is the one we will pick in this tutorial. To install it, simply type:

```
sudo apt-get install apt-proxy
```

Una volta completata l'installazione, il proxy (vuoto) è pronto all'indirizzo "http://INDIRIZZO_MIRROR:9999" e trovaerà i repository Ubuntu sotto "/ubuntu". Affinché vmbuilder possa usarlo, è necessario usare l'opzione *--mirror*:

```
--mirror=URL  Use Ubuntu mirror at URL instead of the default, which
```

```
            is http://archive.ubuntu.com/ubuntu for official
            arches and http://ports.ubuntu.com/ubuntu-ports
            otherwise
```

Aggiungere quindi al comando:

```
--mirror http://INDIRIZZO_MIRROR:9999/ubuntu
```

The mirror address specified here will also be used in the `/etc/apt/sources.list` of the newly created guest, so it is useful to specify here an address that can be resolved by the guest or to plan on reseting this address later on.

2.3.4.2. Installare un mirror locale

Se si è in un ambiente molto grande, può aver senso creare un mirror locale dei repository di Ubuntu. Il pacchetto "apt-mirror" fornisce uno script per la gestione delle operazioni di mirror. È utile avere almeno 20GB di spazio per ogni rilascio supportato e architettura.

By default, apt-mirror uses the configuration file in `/etc/apt/mirror.list`. As it is set up, it will replicate only the architecture of the local machine. If you would like to support other architectures on your mirror, simply duplicate the lines starting with “deb”, replacing the deb keyword by /deb-{arch} where arch can be i386, amd64, etc... For example, on an amd64 machine, to have the i386 archives as well, you will have (some lines have been split to fit the format of this document):

```
deb  http://archive.ubuntu.com/ubuntu precise main restricted universe multiverse
/deb-i386  http://archive.ubuntu.com/ubuntu precise main restricted universe multiverse

deb  http://archive.ubuntu.com/ubuntu precise-updates main restricted universe multiverse
/deb-i386  http://archive.ubuntu.com/ubuntu precise-updates main
 restricted universe multiverse

deb http://archive.ubuntu.com/ubuntu/ precise-backports main restricted universe multiverse
/deb-i386  http://archive.ubuntu.com/ubuntu precise-backports main
 restricted universe multiverse

deb http://security.ubuntu.com/ubuntu precise-security main restricted universe multiverse
/deb-i386  http://security.ubuntu.com/ubuntu precise-security main
 restricted universe multiverse

deb http://archive.ubuntu.com/ubuntu precise main/debian-installer
 restricted/debian-installer universe/debian-installer multiverse/debian-installer
/deb-i386 http://archive.ubuntu.com/ubuntu precise main/debian-installer
 restricted/debian-installer universe/debian-installer multiverse/debian-installer
```

I pacchetti dei sorgenti non sono stati inclusi nel mirror dato che non sono molto usati quanto i binari e occupano molto spazio. È comunque possibile aggiungerli facilmente all'elenco.

Una volta terminata l'operazione di duplicazione del mirror (può durare molto), è necessario configurare Apache affinché i file del mirror (in `/var/spool/apt-mirror` se non è stato modificato il

valore predefinito) siano pubblicati dal proprio server Apache. Per maggiori informazioni su Apache, consultare *Sezione 1, «HTTPD - Server web Apache2» [188]*.

2.4. Pacchettizzare l'applicativo

Sono disponibili due opzioni:

- Il metodo raccomandato è quello di creare un pacchetto *Debian*. Dato che questo argomento esula da questa guida, non verrà spiegato questo metodo e si rimanda alla *Ubuntu Packaging Guide*[10]. In questo caso è anche utile creare un repository per contenere il pacchetto in modo tale che gli aggiornamenti vengano prelevati da questo. Per ulteriori informazioni, consultare *Debian Administration*[11].
- Installare l'applicativo nella directory `/opt` come raccomandato dalle *linee guida di FHS*[12].

In questo caso viene usato Limesurvey come esempio di applicazione web per cui creare un applicativo virtuale. Come accennato precedentemente. è disponibile un pacchetto di questa applicazione attraverso gli archivi PPA (Personal Package Archive).

2.5. Utili accorgimenti

2.5.1. Configurare gli aggiornamenti automatici

Affinché il sistema sia configurato per aggiornarsi automaticamente a scadenze determinate, basta installare il pacchetto unattended-upgrades. Aggiungere quindi quanto segue al comando:

```
--addpkg unattended-upgrades
```

Dato che il pacchetto dell'applicazione è stata inserito nel PPA, il processo di aggiornamento non aggiornerà solamente il sistema, ma anche l'applicazione ogni qualvolta ci sia una versione aggiornata nel PPA.

2.5.2. Gestire gli eventi ACPI

Affinché la macchina virtuale possa gestire gli eventi come riavvio e arresto che le vengono inviati, è utile installare anche il pacchetto "acpid". Aggiungere quindi quanto segue al comando:

```
--addpkg acpid
```

2.6. Il comando finale

Ecco il comando con tutte le opzioni presentate poco sopra:

[10] https://wiki.ubuntu.com/PackagingGuide
[11] http://www.debian-administration.org/articles/286
[12] http://www.pathname.com/fhs/

```
sudo vmbuilder kvm ubuntu --suite precise --flavour virtual --arch i386 -o \
         --libvirt qemu:///system --ip 192.168.0.100 --hostname myvm \
         --part vmbuilder.partition --user user --name user --pass default \
         --addpkg apache2 --addpkg apache2-mpm-prefork --addpkg apache2-utils \
         --addpkg apache2.2-common --addpkg dbconfig-common \
         --addpkg libapache2-mod-php5 --addpkg mysql-client --addpkg php5-cli \
         --addpkg php5-gd --addpkg php5-ldap --addpkg php5-mysql \
         --addpkg wwwconfig-common --addpkg mysql-server \
         --addpkg unattended-upgrades --addpkg acpid --ppa nijaba \
         --mirror http://mirroraddress:9999/ubuntu
```

2.7. Risorse

Per avere maggiori informaioni, per porre qualche domanda o per lasciare dei suggerimenti, contattare l'«Ubuntu Server Team» presso:

- IRC: #ubuntu-server on freenode
- Mailing list: *ubuntu-server at lists.ubuntu.com*[13]
- Also, see the *JeOSVMBuilder Ubuntu Wiki*[14] page.

[13] https://lists.ubuntu.com/mailman/listinfo/ubuntu-server

[14] https://help.ubuntu.com/community/JeOSVMBuilder

3. UEC

3.1. Panoramica

UEC (Ubuntu Enterprise Cloud) is now depracated in favour of UC (Ubuntu Cloud). The former is based on Eucalyptus and the latter is based on Openstack. This section of the guide will be removed in future iterations.

This tutorial covers UEC installation from the Ubuntu 12.04 LTS Server Edition CD, and assumes a basic network topology, with a single system serving as the *"all-in-one controller"*, and one or more nodes attached.

From this Tutorial you will learn how to install, configure, register and perform several operations on a basic UEC setup that results in a cloud with a one controller *"front-end"* and one or several node(s) for running Virtual Machine (VM) instances. You will also use examples to help get you started using your own private compute cloud.

3.2. Prerequisiti

To deploy a minimal cloud infrastructure, you'll need at least *two* dedicated systems:

- Un'interfaccia.
- Uno o più nodi.

The following are recommendations, rather than fixed requirements. However, our experience in developing this documentation indicated the following suggestions.

3.2.1. Front End Requirements

Use the following table for a system that will run one or more of:

- Cloud Controller (CLC)
- Cluster Controller (CC)
- Walrus (the S3-like storage service)
- Storage Controller (SC)

Tabella 20.1. UEC Front End Requirements

Hardware	Minimo	Suggerito	Note
CPU	1 GHz	2 x 2 GHz	Per un'interfaccia *tutta-in-uno* è utile avere almeno un processore dual core.
Memoria	2 GB	4 GB	Per l'interfaccia Java è utile avere molta memoria disponibile.
Disco	5400 RPM IDE	7200 RPM SATA	È possibile utilizzare anche dischi più lenti, ma i tempi di avvio risulteranno più lenti.
Spazio su disco	40 GB	200 GB	40GB è lo spazio sufficiente per una singola immagine, cache, ecc...
Rete	100 Mbps	1000 Mbps	La dimensione delle immagini è di centinaia di megabyte ed è necessario copiare il tutto attraverso la rete verso i nodi.

3.2.2. Requisiti del nodo

The other system(s) are *nodes*, which will run:

- il Node Controller (NC)

Tabella 20.2. Requisiti nodo UEC

Hardware	Minimo	Suggerito	Note
CPU	Estensioni VT	VT, 64-bit, Multicore	64-bit è in grado di eseguire istanze sia i386 che amd64; Eucalyptus eseguirà solamente 1 VM per core di CPU su un nodo.
Memoria	1 GB	4 GB	Più memoria significa guest più grandi e numerosi.
Disco	5400 RPM IDE	7200 RPM SATA or SCSI	I nodi di Eucalyptus sfruttano molto i dischi, le attese di I/O possono causare cali nelle prestazioni.
Spazio su disco	40 GB	100 GB	Le immagini verranno salvate localmente.
Rete	100 Mbps	1000 Mbps	La dimensione delle immagini è di centinaia di megabyte ed è necessario copiare il tutto attraverso la rete verso i nodi.

3.3. Installare l'interfaccia Server Cloud/Cluster/Storage/Walrus

1. Download the Ubuntu 12.04 LTS Server ISO file, and burn it to a CD.

2. When you boot, select *"Install Ubuntu Enterprise Cloud"*. The installer will detect if any other Eucalyptus components are present.
3. You can then choose which components to install, based on your chosen *topology*[15].
4. When asked whether you want a *"Cluster"* or a *"Node"* install, select *"Cluster"*.
5. It will ask two other cloud-specific questions during the course of the install:
   - Il nome del cluster.
     - per esempio *cluster1*
   - Un insieme di indirizzi IP pubblici sulla rete che il cloud posso allocare.
     - per esempio *192.168.1.200-192.168.1.249*

3.4. Installare i Node Controller

The node controller install is even simpler. Just make sure that you are connected to the network on which the cloud/cluster controller is already running.

1. Boot from the same ISO on the node(s).
2. When you boot, select *"Install Ubuntu Enterprise Cloud"*.
3. Select *"Install Ubuntu Enterprise Cloud"*.
4. It should detect the Cluster and preselect *"Node"* install for you.
5. Confermare lo schema di partizionamento.
6. The rest of the installation should proceed uninterrupted; complete the installation and reboot the node.

3.5. Registrare i nodi

1. Nodes are the physical systems within UEC that actually run the virtual machine instances of the cloud.

   La registrazione dei componenti dovrebbe essere automatica se:

   a. Public SSH keys have been exchanged properly.

   b. The services are configured properly.

   c. The appropriate *uec-component-listener* is running.

   d. Verify Registration.

   Steps a to e should only be required if you're using the *UEC/PackageInstall*[16] method. Otherwise, if you are following this guide, these steps should already be completed automatically for you, and therefore you can skip *"a"* to *"e"*.

[15] https://help.ubuntu.com/community/UEC/Topologies
[16] https://help.ubuntu.com/community/UEC/PackageInstall

2. Exchange Public Keys

   The Cloud Controller's *eucalyptus* user needs to have SSH access to the Walrus Controller, Cluster Controller, and Storage Controller as the eucalyptus user.

   Install the Cloud Controller's *eucalyptus* user's public ssh key by:

   - On the target controller, temporarily set a password for the eucalyptus user:

     ```
     sudo passwd eucalyptus
     ```

   - Then, on the Cloud Controller:

     ```
     sudo -u eucalyptus ssh-copy-id -i ~eucalyptus/.ssh/id_rsa.pub \
     eucalyptus@<IP_OF_NODE>
     ```

   - You can now remove the password of the eucalyptus account on the target controller, if you wish:

     ```
     sudo passwd -d eucalyptus
     ```

3. Configurare i servizi

   Nel *Cloud Controller*:

   - Per la registrazione del *Cluster Controller*:
     - Define the shell variable CC_NAME in `/etc/eucalyptus/eucalyptus-cc.conf`
     - Define the shell variable CC_IP_ADDR in `/etc/eucalyptus/eucalyptus-ipaddr.conf`, as a space separated list of one or more IP addresses.
   - Per la registrazione del *Walrus Controller*:
     - Define the shell variable WALRUS_IP_ADDR in `/etc/eucalyptus/eucalyptus-ipaddr.conf`, as a single IP address.

   Nel *Cluster Controller*:

   - Per la registrazione dello *Storage Controller*:
     - Define the shell variable CC_NAME in `/etc/eucalyptus/eucalyptus-cc.conf`
     - Define the shell variable SC_IP_ADDR in `/etc/eucalyptus/eucalyptus-ipaddr.conf`, as a space separated list of one or more IP addresses.

4. Publish

   Now start the publication services.

   - *Walrus Controller:*

     ```
     sudo start eucalyptus-walrus-publication
     ```

- *Cluster Controller:*

```
sudo start eucalyptus-cc-publication
```

- *Storage Controller:*

```
sudo start eucalyptus-sc-publication
```

- *Node Controller:*

```
sudo start eucalyptus-nc-publication
```

5. Start the Listener

   Nel *Cloud Controller* e nei *Cluster Controller*, eseguire:

```
sudo start uec-component-listener
```

6. Verificare la registrazione

```
cat /var/log/eucalyptus/registration.log
2010-04-08 15:46:36-05:00 | 24243 -> Calling node cluster1 node 10.1.1.75
2010-04-08 15:46:36-05:00 | 24243 -> euca_conf --register-nodes returned 0
2010-04-08 15:48:47-05:00 | 25858 -> Calling walrus Walrus 10.1.1.71
2010-04-08 15:48:51-05:00 | 25858 -> euca_conf --register-walrus returned 0
2010-04-08 15:49:04-05:00 | 26237 -> Calling cluster cluster1 10.1.1.71
2010-04-08 15:49:08-05:00 | 26237 -> euca_conf --register-cluster returned 0
2010-04-08 15:49:17-05:00 | 26644 -> Calling storage cluster1 storage 10.1.1.71
2010-04-08 15:49:18-05:00 | 26644 -> euca_conf --register-sc returned 0
```

L'output sul proprio computer potrebbe essere diverso dall'esempio precedente.

3.6. Ottenere le credenziali

After installing and booting the *Cloud Controller*, users of the cloud will need to retrieve their credentials. This can be done either through a web browser, or at the command line.

3.6.1. Da un browser

1. From your web browser (either remotely or on your Ubuntu server) access the following URL:

```
https://<indirizzo-ip-cloud-controller>:8443/
```

You must use a secure connection, so make sure you use "https" not "http" in your URL. You will get a security certificate warning. You will have to add an exception to view the page. If you do not accept it you will not be able to view the Eucalyptus configuration page.

2. Use username *'admin'* and password *'admin'* for the first time login (you will be prompted to change your password).
3. Then follow the on-screen instructions to update the admin password and email address.
4. Once the first time configuration process is completed, click the *'credentials'* tab located in the top-left portion of the screen.
5. Click the *'Download Credentials'* button to get your certificates.
6. Salvare il tutto in `~/.euca`.
7. Unzip the downloaded zip file into a safe location (`~/.euca`).

```
unzip -d ~/.euca mycreds.zip
```

3.6.2. Dalla riga di comando

- Alternatively, if you are on the command line of the *Cloud Controller*, you can run:

```
mkdir -p ~/.euca
chmod 700 ~/.euca
cd ~/.euca
sudo euca_conf --get-credentials mycreds.zip
unzip mycreds.zip
ln -s ~/.euca/eucarc ~/.eucarc
cd -
```

3.6.3. Estrarre e utilizzare le credenziali

Now you will need to setup EC2 API and AMI tools on your server using X.509 certificates.

1. Installare gli strumenti richiesti:

```
sudo apt-get install euca2ools
```

2. Per verificare che tutto funzioni correttamente, recuperare i dettagli di disponibilità del cluster locale:

```
. ~/.euca/eucarc
euca-describe-availability-zones verbose
AVAILABILITYZONE   myowncloud                192.168.1.1
AVAILABILITYZONE   |- vm types               free / max   cpu   ram  disk
AVAILABILITYZONE   |- m1.small               0004 / 0004   1    128     2
AVAILABILITYZONE   |- c1.medium              0004 / 0004   1    256     5
AVAILABILITYZONE   |- m1.large               0002 / 0002   2    512    10
AVAILABILITYZONE   |- m1.xlarge              0002 / 0002   2   1024    20
AVAILABILITYZONE   |- c1.xlarge              0001 / 0001   4   2048    20
```

L'output del comando precedente potrebbe essere diverso.

3.7. Install an Image from the Store

The following is by far the simplest way to install an image. However, advanced users may be interested in learning how to *Bundle their own image*[17].

The simplest way to add an image to UEC is to install it from the Image Store on the UEC web interface.

1. Access the web interface at the following URL (Make sure you specify https):

   ```
   https://<indirizzo-ip-cloud-controller>:8443/
   ```

2. Enter your login and password (if requested, as you may still be logged in from earlier).
3. Click on the *Store* tab.
4. Browse available images.
5. Click on *install* for the image you want.

Once the image has been downloaded and installed, you can click on *"How to run?"* that will be displayed below the image button to view the command to execute to instantiate (start) this image. The image will also appear on the list given on the *Image* tab.

3.8. Eseguire un'immagine

Ci sono diversi modi per inizializzare un'immagine in UEC:

- Usare la riga di comando.
- Use one of the UEC compatible management tools such as *Landscape*.
- Use the *ElasticFox*[18] extension to Firefox.

Di seguito viene descritta la procedura dalla riga di comando:

1. Before running an instance of your image, you should first create a *keypair* (ssh key) that you can use to log into your instance as root, once it boots. The key is stored, so you will only have to do this once.

   Eseguire il seguente comando:

   ```
   if [ ! -e ~/.euca/mykey.priv ]; then
       mkdir -p -m 700 ~/.euca
       touch ~/.euca/mykey.priv
       chmod 0600 ~/.euca/mykey.priv
       euca-add-keypair mykey > ~/.euca/mykey.priv
   fi
   ```

[17] https://help.ubuntu.com/community/UEC/BundlingImages
[18] https://help.ubuntu.com/community/UEC/ElasticFox

You can call your key whatever you like (in this example, the key is called *'mykey'*), but remember what it is called. If you forget, you can always run **euca-describe-keypairs** to get a list of created keys stored in the system.

2. È necessario consentire accesso alla porta 22 in tutte le istanze:

```
euca-authorize default -P tcp -p 22 -s 0.0.0.0/0
```

3. È quindi possibile creare istanze delle proprie immagini registrate:

```
euca-run-instances $EMI -k mykey -t m1.small
```

If you receive an error regarding *image_id*, you may find it by viewing Images page or click *"How to Run"* on the *Store* page to see the sample command.

4. The first time you run an instance, the system will be setting up caches for the image from which it will be created. This can often take some time the first time an instance is run given that VM images are usually quite large.

   To monitor the state of your instance, run:

```
watch -n5 euca-describe-instances
```

   In the output, you should see information about the instance, including its state. While first-time caching is being performed, the instance's state will be *'pending'*.

5. When the instance is fully started, the above state will become *'running'*. Look at the IP address assigned to your instance in the output, then connect to it:

```
IPADDR=$(euca-describe-instances | grep $EMI | grep running | \
tail -n1 | awk '{print $4}')
ssh -i ~/.euca/mykey.priv ubuntu@$IPADDR
```

6. And when you are done with this instance, exit your SSH connection, then terminate your instance:

```
INSTANCEID=$(euca-describe-instances | grep $EMI | grep running | \
tail -n1 | awk '{print $2}')
euca-terminate-instances $INSTANCEID
```

3.8.1. Primo avvio

The cloud-init package provides "first boot" functionality for the Ubuntu UEC images. It is in charge of taking the generic filesystem image that is booting and customizing it for this particular instance. That includes things like:

- Setting the hostname.
- Putting the provided ssh public keys into `~ubuntu/.ssh/authorized_keys`.

- Running a user provided script, or otherwise modifying the image.

Setting hostname and configuring a system so the person who launched it can actually log into it are not terribly interesting. The interesting things that can be done with cloud-init are made possible by data provided at launch time called *user-data*[19].

First, install the cloud-init package:

```
sudo apt-get install cloud-init
```

If the user-data starts with *'#!'*, then it will be stored and executed as root late in the boot process of the instance's first boot (similar to a traditional 'rc.local' script). Output from the script is directed to the console.

Per esempio, creare un file chiamato `ud.txt` che contiene quanto segue:

```
#!/bin/sh
echo ========== Hello World: $(date) ==========
echo "I have been up for $(cut -d\  -f 1 < /proc/uptime) sec"
```

Ovviare un'istanza con l'opzione *--user-data-file*:

```
euca-run-instances $EMI -k mykey -t m1.small --user-data-file=ud.txt
```

Attendere che il sistema e la console siano disponibili. Per visualizzare i risultati, digitare:

```
euca-get-console-output $EMI | grep --after-context=1 Hello
========== Hello World: Mon Mar 29 18:05:05 UTC 2010 ==========
I have been up for 28.26 sec
```

L'output del proprio comando potrebbe variare.

The simple approach shown above gives a great deal of power. The user-data can contain a script in any language where an interpreter already exists in the image (#!/bin/sh, #!/usr/bin/python, #!/usr/bin/perl, #!/usr/bin/awk ...).

For many cases, the user may not be interested in writing a program. For this case, cloud-init provides *"cloud-config"*, a configuration based approach towards customization. To utilize the cloud-config syntax, the supplied user-data must start with a *'#cloud-config'*.

For example, create a text file named `cloud-config.txt` containing:

```
#cloud-config
```

[19] http://developer.amazonwebservices.com/connect/entry.jspa?externalID=1085

```
apt_upgrade: true
apt_sources:
- source: "ppa:ubuntu-server-edgers/server-edgers-apache "

packages:
- build-essential
- pastebinit

runcmd:
- echo ======= Hello World =====
- echo "I have been up for $(cut -d\  -f 1 < /proc/uptime) sec"
```

Creare una nuova istanza:

```
euca-run-instances $EMI -k mykey -t m1.small --user-data-file=cloud-config.txt
```

Una volta avviato il sistema, dovrebbe avere:

- Added the Apache Edgers PPA.
- Run an upgrade to get all updates available
- Installed the 'build-essential' and 'pastebinit' packages
- Printed a similar message to the script above

The *Apache Edgers PPA*, in the above example, contains the latest version of Apache from upstream source repositories. Package versions in the PPA are unsupported, and depending on your situation, this may or may not be desirable. See the *Ubuntu Server Edgers*[20] web page for more details.

The *'runcmd'* commands are run at the same point in boot that the *'#!'* script would run in the previous example. It is present to allow you to get the full power of a scripting language if you need it without abandoning *cloud-config*.

For more information on what kinds of things can be done with cloud-config, see *doc/examples*[21] in the source.

3.9. Ulteriori informazioni

How to use the *Storage Controller*[22]

Controllare i servizi di Eucalyptus

- sudo service eucalyptus [start|stop|restart] (on the CLC/CC/SC/Walrus side)
- sudo service eucalyptus-nc [start|stop|restart] (on the Node side)

Posizione di alcuni dei file importanti:

[20] https://launchpad.net/~ubuntu-server-edgers
[21] http://bazaar.launchpad.net/~cloud-init-dev/cloud-init/trunk/files/head:/doc/examples/
[22] https://help.ubuntu.com/community/UEC/StorageController

- *File di registro:*
  - /var/log/eucalyptus
- *File di configurazione:*
  - /etc/eucalyptus
- *Database:*
  - /var/lib/eucalyptus/db
- *Chiavi:*
  - /var/lib/eucalyptus
  - /var/lib/eucalyptus/.ssh

Don't forget to source your `~/.euca/eucarc` before running the client tools.

3.10. Riferimenti

- Per informazioni sul caricamento delle istanze, consultare la *documentazione della comunità internazionale*[23].
- *Eucalyptus Project Site (forums, documentation, downloads)*[24].
- *Eucalyptus on Launchpad (bugs, code)*[25].
- *Eucalyptus Troubleshooting (1.5)*[26].
- *Register your cloud with RightScale*[27].
- È anche possibile trovare aiuto nei canali IRC *#ubuntu-virt*, *#eucalyptus*e *#ubuntu-server* sul server *Freenode*[28].

[23] https://help.ubuntu.com/community/Eucalyptus
[24] http://open.eucalyptus.com/
[25] https://launchpad.net/eucalyptus/
[26] http://open.eucalyptus.com/wiki/EucalyptusTroubleshooting_v1.5
[27] http://support.rightscale.com/2._References/02-Cloud_Infrastructures/Eucalyptus/03-Administration_Guide/Register_with_RightScale
[28] http://freenode.net

4. Ubuntu Cloud

Cloud computing is a computing model that allows vast pools of resources to be allocated on-demand. These resources such as storage, computing power, network and software are abstracted and delivered as a service over the Internet anywhere, anytime. These services are billed per time consumed similar to the ones used by public services such as electricity, water and telephony. Ubuntu Cloud Infrastructure uses OpenStack open source software to help build highly scalable, cloud computing for both public and private clouds.

4.1. Panoramica

This tutorial covers the OpenStack installation from the Ubuntu 12.04 LTS Server Edition CD, and assumes a basic network topology, with a single system serving as the "all-in-one cloud infrastructure".Due to the tutorial's simplicity, the instructions as-is are not intended to set up production servers although it allows you to have a POC (proof of concept) of the Ubuntu Cloud using OpenStack.

4.2. Prerequisiti

To deploy a minimal Ubuntu Cloud infrastructure, you'll need at least:

- One dedicated system.
- Two network address ranges (private network and public network).
- Make sure the host in question supports VT (Virtualization Technology) since we will be using KVM as the virtualization technology. Other hypervisors are also supported such as QEMU, UML, Vmware ESX/ESXi and XEN. LXC (Linux Containers) is also supported through libvirt.

  Check if your system supports kvm issuing **sudo kvm-ok** in a linux terminal.

The **"Minimum Topology"** recommended for production use is using three nodes - One master server running nova services (except compute) and two servers running nova-compute. This setup is not redundant and the master server is a SPoF (Single Point of Failure).

4.3. Preconfiguring the network

Before we start installing OpenStack we need to make sure we have bridging support installed, a MySQL database, and a central time server (ntp). This will assure that we have instantiated machines and hosts in sync.

In this example the "private network" will be in the 10.0.0.0/24 range on eth1. All the internal communication between instances will happen there while the "public network" will be in the 10.153.107.0/29 range on eth0.

4.3.1. Install bridging support

```
sudo apt-get install bridge-utils
```

4.3.2. Install and configure NTP

```
sudo apt-get install ntp
```

Add these two lines at the end of the `/etc/ntp.conf` file.

```
server 127.127.1.0
fudge 127.127.1.0 stratum 10
```

Restart ntp service

```
sudo service ntp restart
```

4.3.3. Install and configure MySQL

```
sudo apt-get install mysql-server
```

Create a database and mysql user for OpenStack

```
sudo mysql -uroot -ppassword -e "CREATE DATABASE nova;"
sudo mysql -uroot -ppassword -e "GRANT ALL ON nova.* TO novauser@localhost \
IDENTIFIED BY 'novapassword' ";
```

The line continuation character "\" implies that you must include the subsequent line as part of the current command.

4.4. Install OpenStack Compute (Nova)

OpenStack Compute (Nova) is a cloud computing fabric controller (the main part of an IaaS system). It is written in Python, using the Eventlet and Twisted frameworks, and relies on the standard AMQP messaging protocol, and SQLAlchemy for data store access.

Install OpenStack Nova components

```
sudo apt-get install nova-api nova-network nova-volume nova-objectstore nova-scheduler \
nova-compute euca2ools unzip
```

Restart libvirt-bin just to make sure libvirtd is aware of ebtables.

```
sudo service libvirt-bin restart
```

Install RabbitMQ – Advanced Message Queuing Protocol (AMQP)

```
sudo apt-get install rabbitmq-server
```

Edit `/etc/nova/nova.conf` and add the following:

```
# Nova config FlatDHCPManager
--sql_connection=mysql://novauser:novapassword@localhost/nova
--flat_injected=true
--network_manager=nova.network.manager.FlatDHCPManager
--fixed_range=10.0.0.0/24
--floating_range=10.153.107.72/29
--flat_network_dhcp_start=10.0.0.2
--flat_network_bridge=br100
--flat_interface=eth1
--public_interface=eth0
```

Restart OpenStack services

```
for i in nova-api nova-network nova-objectstore nova-scheduler nova-volume nova-compute; \
do sudo stop $i; sleep 2; done
```

```
for i in nova-api nova-network nova-objectstore nova-scheduler nova-volume nova-compute; \
do sudo start $i; sleep 2; done
```

Migrate Nova database from sqlite db to MySQL db. It may take a while.

```
sudo nova-manage db sync
```

Define a specific private network where all your Instances will run. This will be used in the network of fixed Ips set inside `nova.conf` .

```
sudo nova-manage network create --fixed_range_v4 10.0.0.0/24 --label private \
--bridge_interface br100
```

Define a specific public network and allocate 6 (usable) Floating Public IP addresses for use with the instances starting from 10.153.107.72.

```
sudo nova-manage floating create --ip_range=10.153.107.72/29
```

Create a user (user1), a project (project1), download credentials and source its configuration file.

```
cd ; mkdir nova ; cd nova
sudo nova-manage user admin user1
sudo nova-manage project create project1 user1
sudo nova-manage project zipfile project1 user1
unzip nova.zip
source novarc
```

Verify the OpenStack Compute installation by typing:

```
sudo nova-manage service list
sudo nova-manage version list
```

If nova services don’t show up correctly restart OpenStack services as described previously. For more information please refer to the troubleshooting section on this guide.

4.5. Install Imaging Service (Glance)

Nova uses Glance service to manage Operating System images that it needs for bringing up instances. Glance can use several types of storage backends such as filestore, s3 etc. Glance has two components - *glance-api and glance-registry*. These can be controlled using the concerned upstart service jobs. For this specific case we will be using mysql as a storage backend.

Install Glance

```
sudo apt-get install glance
```

Create a database and user for glance

```
sudo mysql -uroot -ppassword -e "CREATE DATABASE glance;"
sudo mysql -uroot -ppassword -e "GRANT ALL ON glance.* TO glanceuser@localhost \
IDENTIFIED BY 'glancepassword' ";
```

Edit the file /etc/glance/glance-registry.conf and edit the line which contains the option "sql_connection =" to this:

```
sql_connection = mysql://glanceuser:glancepassword@localhost/glance
```

Remove the sqlite database

```
rm -rf /var/lib/glance/glance.sqlite
```

Restart glance-registry after making changes to /etc/glance/glance-registry.conf. The MySQL database will be automatically populated.

```
sudo restart glance-registry
```

If you find issues take a look at the log file in /var/log/glance/api.log and /var/log/glance/registry.log.

4.6. Running Instances

Before you can instantiate images, you first need to setup user credentials. Once this first step is achieved you also need to upload images that you want to run in the cloud. Once you have these images uploaded to the cloud you will be able to run and connect to them. Here are the steps you should follow to get OpenStack Nova running instances:

Download, register and publish an Ubuntu cloud image

```
distro=lucid
wget http://cloud-images.ubuntu.com/$distro/current/$distro-server-cloudimg-amd64.tar.gz
cloud-publish-tarball "$distro"-server-cloudimg-amd64.tar.gz "$distro"_amd64
```

Create a key pair and start an instance

```
cd ~/nova
source novarc
euca-add-keypair user1 > user1.priv
chmod 0600 user1.priv
```

Allow icmp (ping) and ssh access to instances

```
euca-authorize default -P tcp -p 22 -s 0.0.0.0/0
euca-authorize -P icmp -t -1:-1 default
```

Run an instance

```
ami=`euca-describe-images |  awk {'print $2'} | grep -m1 ami`
euca-run-instances $ami -k user1 -t m1.tiny
euca-describe-instances
```

Assign public address to the instance.

```
euca-allocate-address
euca-associate-address -i instance_id public_ip_address
euca-describe-instances
```

You must enter above the instance_id (ami) and public_ip_address shown above by euca-describe-instances and euca-allocate-address commands.

Now you should be able to SSH to the instance

```
ssh -i user1.priv ubuntu@ipaddress
```

To terminate instances

```
euca-terminate-instances instance_id
```

4.7. Install the Storage Infrastructure (Swift)

Swift is a highly available, distributed, eventually consistent object/blob store. It is used by the OpenStack Infrastructure to provide S3 like cloud storage services. It is also S3 api compatible with amazon.

Organizations use Swift to store lots of data efficiently, safely, and cheaply where applications use an special api to interface between the applications and objects stored in Swift.

Although you can install Swift on a single server, a multiple-server installation is required for production environments. If you want to install OpenStack Object Storage (Swift) on a single node for development or testing purposes, use the Swift All In One instructions on Ubuntu.

For more information see: *http://swift.openstack.org/development_saio.html* [29] .

4.8. Support and Troubleshooting

Community Support

- *OpenStack Mailing list*[30]
- *The OpenStack Wiki search*[31]
- *Launchpad bugs area*[32]
- Join the IRC channel #openstack on freenode.

4.9. Risorse

- *Cloud Computing - Service models*[33]
- *OpenStack Compute*[34]
- *OpenStack Image Service*[35]
- *OpenStack Object Storage Administration Guide*
- *Installing OpenStack Object Storage on Ubuntu*[36]
- *http://cloudglossary.com/*

4.10. Glossario

The Ubuntu Cloud documentation uses terminology that might be unfamiliar to some readers. This page is intended to provide a glossary of such terms and acronyms.

- *Cloud* - A federated set of physical machines that offer computing resources through virtual machines, provisioned and recollected dynamically.
- *IaaS* - Infrastructure as a Service — Cloud infrastructure services, whereby a virtualized environment is delivered as a service over the Internet by the provider. The infrastructure can include servers, network equipment, and software.

[29] http://swift.openstack.org/development_saio.html
[30] https://launchpad.net/~openstack
[31] http://wiki.openstack.org
[32] https://bugs.launchpad.net/nova
[33] http://en.wikipedia.org/wiki/Cloud_computing#Service_Models
[34] docs.openstack.org/trunk/openstack-compute/
[35] http://docs.openstack.org/diablo/openstack-compute/starter/content/GlanceMS-d2s21.html
[36] http://docs.openstack.org/trunk/openstack-object-storage/admin/content/installing-openstack-object-storage-on-ubuntu.html

- *EBS* - Elastic Block Storage.
- *EC2* - Elastic Compute Cloud. Amazon's pay-by-the-hour, pay-by-the-gigabyte public cloud computing offering.
- *Node* - A node is a physical machine that's capable of running virtual machines, running a node controller. Within Ubuntu, this generally means that the CPU has VT extensions, and can run the KVM hypervisor.
- *S3* - Simple Storage Service. Amazon's pay-by-the-gigabyte persistent storage solution for EC2.
- *Ubuntu Cloud* - Ubuntu Cloud. Ubuntu's cloud computing solution, based on OpenStack.
- *VM* - Virtual Machine.
- *VT* - Virtualization Technology. An optional feature of some modern CPUs, allowing for accelerated virtual machine hosting.

5. LXC

Containers are a lightweight virtualization technology. They are more akin to an enhanced chroot than to full virtualization like Qemu or VMware, both because they do not emulate hardware and because containers share the same operating system as the host. Therefore containers are better compared to Solaris zones or BSD jails. Linux-vserver and OpenVZ are two pre-existing, independently developed implementations of containers-like functionality for Linux. In fact, containers came about as a result of the work to upstream the vserver and OpenVZ functionality. Some vserver and OpenVZ functionality is still missing in containers, however containers can *boot* many Linux distributions and have the advantage that they can be used with an un-modified upstream kernel.

There are two user-space implementations of containers, each exploiting the same kernel features. Libvirt allows the use of containers through the LXC driver by connecting to 'lxc:///'. This can be very convenient as it supports the same usage as its other drivers. The other implementation, called simply 'LXC', is not compatible with libvirt, but is more flexible with more userspace tools. It is possible to switch between the two, though there are peculiarities which can cause confusion.

In this document we will mainly describe the lxc package. Toward the end, we will describe how to use the libvirt LXC driver.

In this document, a container name will be shown as CN, C1, or C2.

5.1. Installazione

The lxc package can be installed using

```
sudo apt-get install lxc
```

This will pull in the required and recommended dependencies, including cgroup-lite, lvm2, and debootstrap. To use libvirt-lxc, install libvirt-bin. LXC and libvirt-lxc can be installed and used at the same time.

5.2. Host Setup

5.2.1. Basic layout of LXC files

Following is a description of the files and directories which are installed and used by LXC.

- There are two upstart jobs:
  - `/etc/init/lxc-net.conf`: is an optional job which only runs if `/etc/default/lxc` specifies USE_LXC_BRIDGE (true by default). It sets up a NATed bridge for containers to use.
  - `/etc/init/lxc.conf`: runs if LXC_AUTO (true by default) is set to true in `/etc/default/lxc`. It looks for entries under `/etc/lxc/auto/` which are symbolic links to configuration files for the containers which should be started at boot.

- `/etc/lxc/lxc.conf`: There is a default container creation configuration file, `/etc/lxc/lxc.conf`, which directs containers to use the LXC bridge created by the lxc-net upstart job. If no configuration file is specified when creating a container, then this one will be used.
- Examples of other container creation configuration files are found under `/usr/share/doc/lxc/examples`. These show how to create containers without a private network, or using macvlan, vlan, or other network layouts.
- The various container administration tools are found under `/usr/bin`.
- `/usr/lib/lxc/lxc-init` is a very minimal and lightweight init binary which is used by lxc-execute. Rather than `booting' a full container, it manually mounts a few filesystems, especially `/proc`, and executes its arguments. You are not likely to need to manually refer to this file.
- `/usr/lib/lxc/templates/` contains the `templates' which can be used to create new containers of various distributions and flavors. Not all templates are currently supported.
- `/etc/apparmor.d/lxc/lxc-default` contains the default Apparmor MAC policy which works to protect the host from containers. Please see the *Sezione 5.2.6, «Apparmor» [345]* for more information.
- `/etc/apparmor.d/usr.bin.lxc-start` contains a profile to protect the host from **lxc-start** while it is setting up the container.
- `/etc/apparmor.d/lxc-containers` causes all the profiles defined under `/etc/apparmor.d/lxc` to be loaded at boot.
- There are various man pages for the LXC administration tools as well as the `lxc.conf` container configuration file.
- `/var/lib/lxc` is where containers and their configuration information are stored.
- `/var/cache/lxc` is where caches of distribution data are stored to speed up multiple container creations.

5.2.2. lxcbr0

When USE_LXC_BRIDGE is set to true in /etc/default/lxc (as it is by default), a bridge called lxcbr0 is created at startup. This bridge is given the private address 10.0.3.1, and containers using this bridge will have a 10.0.3.0/24 address. A dnsmasq instance is run listening on that bridge, so if another dnsmasq has bound all interfaces before the lxc-net upstart job runs, lxc-net will fail to start and lxcbr0 will not exist.

If you have another bridge - libvirt's default virbr0, or a br0 bridge for your default NIC - you can use that bridge in place of lxcbr0 for your containers.

5.2.3. Using a separate filesystem for the container store

LXC stores container information and (with the default backing store) root filesystems under `/var/lib/lxc`. Container creation templates also tend to store cached distribution information under `/var/cache/lxc`.

If you wish to use another filesystem than `/var`, you can mount a filesystem which has more space into those locations. If you have a disk dedicated for this, you can simply mount it at `/var/lib/lxc`. If you'd like to use another location, like `/srv`, you can bind mount it or use a symbolic link. For instance, if `/srv` is a large mounted filesystem, create and symlink two directories:

```
sudo mkdir /srv/lxclib /srv/lxccache
sudo rm -rf /var/lib/lxc /var/cache/lxc
sudo ln -s /srv/lxclib /var/lib/lxc
sudo ln -s /srv/lxccache /var/cache/lxc
```

or, using bind mounts:

```
sudo mkdir /srv/lxclib /srv/lxccache
sudo sed -i '$a \
/srv/lxclib /var/lib/lxc    none defaults,bind 0 0 \
/srv/lxccache /var/cache/lxc none defaults,bind 0 0' /etc/fstab
sudo mount -a
```

5.2.4. Containers backed by lvm

It is possible to use LVM partitions as the backing stores for containers. Advantages of this include flexibility in storage management and fast container cloning. The tools default to using a VG (volume group) named *lxc*, but another VG can be used through command line options. When a LV is used as a container backing store, the container's configuration file is still `/var/lib/lxc/CN/config`, but the root fs entry in that file (*lxc.rootfs*) will point to the lV block device name, i.e. `/dev/lxc/CN`.

Containers with directory tree and LVM backing stores can co-exist.

5.2.5. Btrfs

If your host has a btrfs `/var`, the LXC administration tools will detect this and automatically exploit it by cloning containers using btrfs snapshots.

5.2.6. Apparmor

LXC ships with an Apparmor profile intended to protect the host from accidental misuses of privilege inside the container. For instance, the container will not be able to write to `/proc/sysrq-trigger` or to most `/sys` files.

The `usr.bin.lxc-start` profile is entered by running **lxc-start**. This profile mainly prevents **lxc-start** from mounting new filesystems outside of the container's root filesystem. Before executing the container's **init**, **LXC** requests a switch to the container's profile. By default, this profile is the `lxc-container-default` policy which is defined in `/etc/apparmor.d/lxc/lxc-default`. This profile prevents the container from accessing many dangerous paths, and from mounting most filesystems.

If you find that **lxc-start** is failing due to a legitimate access which is being denied by its Apparmor policy, you can disable the lxc-start profile by doing:

```
sudo apparmor_parser -R /etc/apparmor.d/usr.bin.lxc-start
sudo ln -s /etc/apparmor.d/usr.bin.lxc-start /etc/apparmor.d/disabled/
```

This will make **lxc-start** run unconfined, but continue to confine the container itself. If you also wish to disable confinement of the container, then in addition to disabling the `usr.bin.lxc-start` profile, you must add:

```
lxc.aa_profile = unconfined
```

to the container's configuration file. If you wish to run a container in a custom profile, you can create a new profile under `/etc/apparmor.d/lxc/`. Its name must start with `lxc-` in order for **lxc-start** to be allowed to transition to that profile. After creating the policy, load it using:

```
sudo apparmor_parser -r /etc/apparmor.d/lxc-containers
```

The profile will automatically be loaded after a reboot, because it is sourced by the file `/etc/apparmor.d/lxc-containers`. Finally, to make container `CN` use this new `lxc-CN-profile`, add the following line to its configuration file:

```
lxc.aa_profile = lxc-CN-profile
```

lxc-execute does not enter an Apparmor profile, but the container it spawns will be confined.

5.2.7. Control Groups

Control groups (cgroups) are a kernel feature providing hierarchical task grouping and per-cgroup resource accounting and limits. They are used in containers to limit block and character device access and to freeze (suspend) containers. They can be further used to limit memory use and block i/o, guarantee minimum cpu shares, and to lock containers to specific cpus. By default, LXC depends on the cgroup-lite package to be installed, which provides the proper cgroup initialization at boot. The cgroup-lite package mounts each cgroup subsystem separately under `/sys/fs/cgroup/SS`, where SS is the subsystem name. For instance the freezer subsystem is mounted under `/sys/fs/cgroup/freezer`. LXC cgroup are kept under `/sys/fs/cgroup/SS/INIT/lxc`, where INIT is the init task's cgroup. This is `/` by default, so in the end the freezer cgroup for container CN would be `/sys/fs/cgroup/freezer/lxc/CN`.

5.2.8. Privilege

The container administration tools must be run with root user privilege. A utility called `lxc-setup` was written with the intention of providing the tools with the needed file capabilities to allow non-root users to run the tools with sufficient privilege. However, as root in a container cannot yet be reliably

contained, this is not worthwhile. It is therefore recommended to not use `lxc-setup`, and to provide the LXC administrators the needed sudo privilege.

The user namespace, which is expected to be available in the next Long Term Support (LTS) release, will allow containment of the container root user, as well as reduce the amount of privilege required for creating and administering containers.

5.2.9. LXC Upstart Jobs

As listed above, the lxc package includes two upstart jobs. The first, `lxc-net`, is always started when the other, `lxc`, is about to begin, and stops when it stops. If the USE_LXC_BRIDGE variable is set to false in `/etc/defaults/lxc`, then it will immediately exit. If it is true, and an error occurs bringing up the LXC bridge, then the `lxc` job will not start. `lxc-net` will bring down the LXC bridge when stopped, unless a container is running which is using that bridge.

The `lxc` job starts on runlevel 2-5. If the LXC_AUTO variable is set to true, then it will look under `/etc/lxc` for containers which should be started automatically. When the `lxc` job is stopped, either manually or by entering runlevel 0, 1, or 6, it will stop those containers.

To register a container to start automatically, create a symbolic link `/etc/default/lxc/name.conf` pointing to the container's config file. For instance, the configuration file for a container `CN` is `/var/lib/lxc/CN/config`. To make that container auto-start, use the command:

```
sudo ln -s /var/lib/lxc/CN/config /etc/lxc/auto/CN.conf
```

5.3. Container Administration

5.3.1. Creating Containers

The easiest way to create containers is using **lxc-create**. This script uses distribution-specific templates under `/usr/lib/lxc/templates/` to set up container-friendly chroots under `/var/lib/lxc/CN/rootfs`, and initialize the configuration in `/var/lib/lxc/CN/fstab` and `/var/lib/lxc/CN/config`, where CN is the container name

The simplest container creation command would look like:

```
sudo lxc-create -t ubuntu -n CN
```

This tells lxc-create to use the ubuntu template (-t ubuntu) and to call the container CN (-n CN). Since no configuration file was specified (which would have been done with `-f file'), it will use the default configuration file under `/etc/lxc/lxc.conf`. This gives the container a single veth network interface attached to the lxcbr0 bridge.

The container creation templates can also accept arguments. These can be listed after --. For instance

```
sudo lxc-create -t ubuntu -n oneiric1 -- -r oneiric
```

passes the arguments '-r oneiric1' to the ubuntu template.

5.3.1.1. Help

Help on the lxc-create command can be seen by using **lxc-create -h**. However, the templates also take their own options. If you do

```
sudo lxc-create -t ubuntu -h
```

then the general **lxc-create** help will be followed by help output specific to the ubuntu template. If no template is specified, then only help for **lxc-create** itself will be shown.

5.3.1.2. Ubuntu template

The ubuntu template can be used to create Ubuntu system containers with any release at least as new as 10.04 LTS. It uses debootstrap to create a cached container filesystem which gets copied into place each time a container is created. The cached image is saved and only re-generated when you create a container using the *-F* (flush) option to the template, i.e.:

```
sudo lxc-create -t ubuntu -n CN -- -F
```

The Ubuntu release installed by the template will be the same as that on the host, unless otherwise specified with the *-r* option, i.e.

```
sudo lxc-create -t ubuntu -n CN -- -r lucid
```

If you want to create a 32-bit container on a 64-bit host, pass *-a i386* to the container. If you have the qemu-user-static package installed, then you can create a container using any architecture supported by qemu-user-static.

The container will have a user named *ubuntu* whose password is *ubuntu* and who is a member of the *sudo* group. If you wish to inject a public ssh key for the *ubuntu* user, you can do so with *-S sshkey.pub*.

You can also *bind* user jdoe from the host into the container using the *-b jdoe* option. This will copy jdoe's password and shadow entries into the container, make sure his default group and shell are

available, add him to the sudo group, and bind-mount his home directory into the container when the container is started.

When a container is created, the `release-updates` archive is added to the container's `sources.list`, and its package archive will be updated. If the container release is older than 12.04 LTS, then the lxcguest package will be automatically installed. Alternatively, if the *--trim* option is specified, then the lxcguest package will not be installed, and many services will be removed from the container. This will result in a faster-booting, but less upgrade-able container.

5.3.1.3. Ubuntu-cloud template

The ubuntu-cloud template creates Ubuntu containers by downloading and extracting the published Ubuntu cloud images. It accepts some of the same options as the ubuntu template, namely *-r release*, *-S sshkey.pub*, *-a arch*, and *-F* to flush the cached image. It also accepts a few extra options. The *-C* option will create a *cloud* container, configured for use with a metadata service. The *-u* option accepts a cloud-init user-data file to configure the container on start. If *-L* is passed, then no locales will be installed. The *-T* option can be used to choose a tarball location to extract in place of the published cloud image tarball. Finally the *-i* option sets a host id for cloud-init, which by default is set to a random string.

5.3.1.4. Other templates

The ubuntu and ubuntu-cloud templates are well supported. Other templates are available however. The debian template creates a Debian based container, using debootstrap much as the ubuntu template does. By default it installs a *debian squeeze* image. An alternate release can be chosen by setting the SUITE environment variable, i.e.:

```
sudo SUITE=sid lxc-create -t debian -n d1
```

Since debian cannot be safely booted inside a container, debian containers will be trimmed as with the *--trim* option to the ubuntu template.

To purge the container image cache, call the template directly and pass it the *--clean* option.

```
sudo SUITE=sid /usr/lib/lxc/templates/lxc-debian --clean
```

A fedora template exists, which creates containers based on fedora releases <= 14. Fedora release 15 and higher are based on systemd, which the template is not yet able to convert into a container-bootable setup. Before the fedora template is able to run, you'll need to make sure that **yum** and **curl** are installed. A fedora 12 container can be created with

```
sudo lxc-create -t fedora -n fedora12 -- -R 12
```

A OpenSuSE template exists, but it requires the **zypper** program, which is not yet packaged. The OpenSuSE template is therefore not supported.

Two more templates exist mainly for experimental purposes. The busybox template creates a very small system container based entirely on busybox. The sshd template creates an application container running sshd in a private network namespace. The host's library and binary directories are bind-mounted into the container, though not its `/home` or `/root`. To create, start, and ssh into an ssh container, you might:

```
sudo lxc-create -t sshd -n ssh1
ssh-keygen -f id
sudo mkdir /var/lib/lxc/ssh1/rootfs/root/.ssh
sudo cp id.pub /var/lib/lxc/ssh1/rootfs/root/.ssh/authorized_keys
sudo lxc-start -n ssh1 -d
ssh -i id root@ssh1.
```

5.3.1.5. Backing Stores

By default, **lxc-create** places the container's root filesystem as a directory tree at `/var/lib/lxc/CN/rootfs`. Another option is to use LVM logical volumes. If a volume group named *lxc* exists, you can create an lvm-backed container called CN using:

```
sudo lxc-create -t ubuntu -n CN -B lvm
```

If you want to use a volume group named schroots, with a 5G xfs filesystem, then you would use

```
sudo lxc-create -t ubuntu -n CN -B lvm --vgname schroots --fssize 5G --fstype xfs
```

5.3.2. Cloning

For rapid provisioning, you may wish to customize a canonical container according to your needs and then make multiple copies of it. This can be done with the **lxc-clone** program. Given an existing container called C1, a new container called C2 can be created using

```
sudo lxc-clone -o C1 -n C2
```

If `/var/lib/lxc` is a btrfs filesystem, then **lxc-clone** will create C2's filesystem as a snapshot of C1's. If the container's root filesystem is lvm backed, then you can specify the *-s* option to create the new rootfs as a lvm snapshot of the original as follows:

```
sudo lxc-clone -s -o C1 -n C2
```

Both lvm and btrfs snapshots will provide fast cloning with very small initial disk usage.

5.3.3. Starting and stopping

To start a container, use **lxc-start -n CN**. By default **lxc-start** will execute `/sbin/init` in the container. You can provide a different program to execute, plus arguments, as further arguments to **lxc-start**:

```
sudo lxc-start -n container /sbin/init loglevel=debug
```

If you do not specify the *-d* (daemon) option, then you will see a console (on the container's `/dev/console`, see *Sezione 5.3.5, «Consoles» [353]* for more information) on the terminal. If you specify the *-d* option, you will not see that console, and lxc-start will immediately exit success - even if a later part of container startup has failed. You can use **lxc-wait** or **lxc-monitor** (see *Sezione 5.3.4, «Monitoring container status » [352]*) to check on the success or failure of the container startup.

To obtain LXC debugging information, use *-o filename -l debuglevel*, for instance:

```
sudo lxc-start -o lxc.debug -l DEBUG -n container
```

Finally, you can specify configuration parameters inline using *-s*. However, it is generally recommended to place them in the container's configuration file instead. Likewise, an entirely alternate config file can be specified with the *-f* option, but this is not generally recommended.

While **lxc-start** runs the container's `/sbin/init`, **lxc-execute** uses a minimal init program called **lxc-init**, which attempts to mount `/proc`, `/dev/mqueue`, and `/dev/shm`, executes the programs specified on the command line, and waits for those to finish executing. **lxc-start** is intended to be used for *system containers*, while **lxc-execute** is intended for *application containers* (see *this article*[37] for more).

You can stop a container several ways. You can use **shutdown**, **poweroff** and **reboot** while logged into the container. To cleanly shut down a container externally (i.e. from the host), you can issue the **sudo lxc-shutdown -n CN** command. This takes an optional timeout value. If not specified,

[37] https://www.ibm.com/developerworks/linux/library/l-lxc-containers/

the command issues a SIGPWR signal to the container and immediately returns. If the option is used, as in **sudo lxc-shutdown -n CN -t 10**, then the command will wait the specified number of seconds for the container to cleanly shut down. Then, if the container is still running, it will kill it (and any running applications). You can also immediately kill the container (without any chance for applications to cleanly shut down) using **sudo lxc-stop -n CN**. Finally, **lxc-kill** can be used more generally to send any signal number to the container's init.

While the container is shutting down, you can expect to see some (harmless) error messages, as follows:

```
$ sudo poweroff
[sudo] password for ubuntu: =

$ =

Broadcast message from ubuntu@cn1
        (/dev/lxc/console) at 18:17 ...

The system is going down for power off NOW!
 * Asking all remaining processes to terminate...
   ...done.
 * All processes ended within 1 seconds....
   ...done.
 * Deconfiguring network interfaces...
   ...done.
 * Deactivating swap...
   ...fail!
umount: /run/lock: not mounted
umount: /dev/shm: not mounted
mount: / is busy
 * Will now halt
```

A container can be frozen with **sudo lxc-freeze -n CN**. This will block all its processes until the container is later unfrozen using **sudo lxc-unfreeze -n CN**.

5.3.4. Monitoring container status

Two commands are available to monitor container state changes. **lxc-monitor** monitors one or more containers for any state changes. It takes a container name as usual with the *-n* option, but in this case the container name can be a posix regular expression to allow monitoring desirable sets of containers. **lxc-monitor** continues running as it prints container changes. **lxc-wait** waits for a specific state change and then exits. For instance,

```
sudo lxc-monitor -n cont[0-5]*
```

would print all state changes to any containers matching the listed regular expression, whereas

```
sudo lxc-wait -n cont1 -s 'STOPPED|FROZEN'
```

will wait until container cont1 enters state STOPPED or state FROZEN and then exit.

5.3.5. Consoles

Containers have a configurable number of consoles. One always exists on the container's `/dev/console`. This is shown on the terminal from which you ran **lxc-start**, unless the *-d* option is specified. The output on `/dev/console` can be redirected to a file using the *-c console-file* option to **lxc-start**. The number of extra consoles is specified by the **lxc.tty** variable, and is usually set to 4. Those consoles are shown on `/dev/ttyN` (for 1 <= N <= 4). To log into console 3 from the host, use

```
sudo lxc-console -n container -t 3
```

or if the *-t N* option is not specified, an unused console will be automatically chosen. To exit the console, use the escape sequence Ctrl-a q. Note that the escape sequence does not work in the console resulting from **lxc-start** without the *-d* option.

Each container console is actually a Unix98 pty in the host's (not the guest's) pty mount, bind-mounted over the guest's `/dev/ttyN` and `/dev/console`. Therefore, if the guest unmounts those or otherwise tries to access the actual character device **4:N**, it will not be serving getty to the LXC consoles. (With the default settings, the container will not be able to access that character device and getty will therefore fail.) This can easily happen when a boot script blindly mounts a new `/dev`.

5.3.6. Container Inspection

Several commands are available to gather information on existing containers. **lxc-ls** will report all existing containers in its first line of output, and all running containers in the second line. **lxc-list** provides the same information in a more verbose format, listing running containers first and stopped containers next. **lxc-ps** will provide lists of processes in containers. To provide **ps** arguments to **lxc-ps**, prepend them with **--**. For instance, for listing of all processes in container plain,

```
sudo lxc-ps -n plain -- -ef
```

lxc-info provides the state of a container and the pid of its init process. **lxc-cgroup** can be used to query or set the values of a container's control group limits and information. This can be more convenient than interacting with the **cgroup** filesystem. For instance, to query the list of devices which a running container is allowed to access, you could use

```
sudo lxc-cgroup -n CN devices.list
```

or to add mknod, read, and write access to `/dev/sda`,

```
sudo lxc-cgroup -n CN devices.allow "b 8:* rwm"
```

and, to limit it to 300M of RAM,

```
lxc-cgroup -n CN memory.limit_in_bytes 300000000
```

lxc-netstat executes **netstat** in the running container, giving you a glimpse of its network state.

lxc-backup will create backups of the root filesystems of all existing containers (except lvm-based ones), using **rsync** to back the contents up under `/var/lib/lxc/CN/rootfs.backup.1`. These backups can be restored using **lxc-restore.** However, **lxc-backup** and **lxc-restore** are fragile with respect to customizations and therefore their use is not recommended.

5.3.7. Destroying containers

Use **lxc-destroy** to destroy an existing container.

```
sudo lxc-destroy -n CN
```

If the container is running, **lxc-destroy** will exit with a message informing you that you can force stopping and destroying the container with

```
sudo lxc-destroy -n CN -f
```

5.3.8. Advanced namespace usage

One of the Linux kernel features used by LXC to create containers is private namespaces. Namespaces allow a set of tasks to have private mappings of names to resources for things like pathnames and process IDs. (See *Sezione 5.9, «Risorse» [363]* for a link to more information). Unlike control groups and other mount features which are also used to create containers, namespaces cannot be manipulated using a filesystem interface. Therefore, LXC ships with the **lxc-unshare** program, which is mainly for testing. It provides the ability to create new tasks in private namespaces. For instance,

```
sudo lxc-unshare -s 'MOUNT|PID' /bin/bash
```

creates a bash shell with private pid and mount namespaces. In this shell, you can do

```
root@ubuntu:~# mount -t proc proc /proc
root@ubuntu:~# ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  6 10:20 pts/9    00:00:00 /bin/bash
root       110     1  0 10:20 pts/9    00:00:00 ps -ef
```

so that **ps** shows only the tasks in your new namespace.

5.3.9. Ephemeral containers

Ephemeral containers are one-time containers. Given an existing container CN, you can run a command in an ephemeral container created based on CN, with the host's jdoe user bound into the container, using:

```
lxc-start-ephemeral -b jdoe -o CN -- /home/jdoe/run_my_job
```

When the job is finished, the container will be discarded.

5.3.10. Container Commands

Following is a table of all container commands:

Tabella 20.3. Container commands

Command	Synopsis
lxc-attach	(NOT SUPPORTED) Run a command in a running container
lxc-backup	Back up the root filesystems for all lvm-backed containers
lxc-cgroup	View and set container control group settings
lxc-checkconfig	Verify host support for containers
lxc-checkpoint	(NOT SUPPORTED) Checkpoint a running container
lxc-clone	Clone a new container from an existing one
lxc-console	Open a console in a running container
lxc-create	Create a new container
lxc-destroy	Destroy an existing container
lxc-execute	Run a command in a (not running) application container

Command	Synopsis
lxc-freeze	Freeze a running container
lxc-info	Print information on the state of a container
lxc-kill	Send a signal to a container's init
lxc-list	List all containers
lxc-ls	List all containers with shorter output than lxc-list
lxc-monitor	Monitor state changes of one or more containers
lxc-netstat	Execute netstat in a running container
lxc-ps	View process info in a running container
lxc-restart	(NOT SUPPORTED) Restart a checkpointed container
lxc-restore	Restore containers from backups made by lxc-backup
lxc-setcap	(NOT RECOMMENDED) Set file capabilities on LXC tools
lxc-setuid	(NOT RECOMMENDED) Set or remove setuid bits on LXC tools
lxc-shutdown	Safely shut down a container
lxc-start	Start a stopped container
lxc-start-ephemeral	Start an ephemeral (one-time) container
lxc-stop	Immediately stop a running container
lxc-unfreeze	Unfreeze a frozen container
lxc-unshare	Testing tool to manually unshare namespaces
lxc-version	Print the version of the LXC tools
lxc-wait	Wait for a container to reach a particular state

5.4. Configuration File

LXC containers are very flexible. The Ubuntu lxc package sets defaults to make creation of Ubuntu system containers as simple as possible. If you need more flexibility, this chapter will show how to fine-tune your containers as you need.

Detailed information is available in the **lxc.conf(5)** man page. Note that the default configurations created by the ubuntu templates are reasonable for a system container and usually do not need customization.

5.4.1. Choosing configuration files and options

The container setup is controlled by the LXC configuration options. Options can be specified at several points:

- During container creation, a configuration file can be specified. However, creation templates often insert their own configuration options, so we usually specify only network configuration options at

this point. For other configuration, it is usually better to edit the configuration file after container creation.

- The file `/var/lib/lxc/CN/config` is used at container startup by default.
- **lxc-start** accepts an alternate configuration file with the *-f filename* option.
- Specific configuration variables can be overridden at **lxc-start** using *-s key=value*. It is generally better to edit the container configuration file.

5.4.2. Configurare la rete

Container networking in LXC is very flexible. It is triggered by the **lxc.network.type** configuration file entries. If no such entries exist, then the container will share the host's networking stack. Services and connections started in the container will be using the host's IP address. If at least one **lxc.network.type** entry is present, then the container will have a private (layer 2) network stack. It will have its own network interfaces and firewall rules. There are several options for **lxc.network.type**:

- **lxc.network.type=empty**: The container will have no network interfaces other than loopback.
- **lxc.network.type=veth**: This is the default when using the ubuntu or ubuntu-cloud templates, and creates a veth network tunnel. One end of this tunnel becomes the network interface inside the container. The other end is attached to a bridged on the host. Any number of such tunnels can be created by adding more **lxc.network.type=veth** entries in the container configuration file. The bridge to which the host end of the tunnel will be attached is specified with **lxc.network.link = lxcbr0**.
- **lxc.network.type=phys** A physical network interface (i.e. eth2) is passed into the container.

Two other options are to use vlan or macvlan, however their use is more complicated and is not described here. A few other networking options exist:

- **lxc.network.flags** can only be set to *up* and ensures that the network interface is up.
- **lxc.network.hwaddr** specifies a mac address to assign the the nic inside the container.
- **lxc.network.ipv4** and **lxc.network.ipv6** set the respective IP addresses, if those should be static.
- **lxc.network.name** specifies a name to assign inside the container. If this is not specified, a good default (i.e. eth0 for the first nic) is chosen.
- **lxc.network.lxcscript.up** specifies a script to be called after the host side of the networking has been set up. See the **lxc.conf(5)** manual page for details.

5.4.3. Control group configuration

Cgroup options can be specified using **lxc.cgroup** entries. **lxc.cgroup.subsystem.item = value** instructs LXC to set cgroup **subsystem**'s **item** to **value**. It is perhaps simpler to realize that this will simply write **value** to the file **item** for the container's control group for subsystem **subsystem**. For instance, to set the memory limit to 320M, you could add

```
lxc.cgroup.memory.limit_in_bytes = 320000000
```

which will cause 320000000 to be written to the file `/sys/fs/cgroup/memory/lxc/CN/limit_in_bytes`.

5.4.4. Rootfs, mounts and fstab

An important part of container setup is the mounting of various filesystems into place. The following is an example configuration file excerpt demonstrating the commonly used configuration options:

```
lxc.rootfs = /var/lib/lxc/CN/rootfs
lxc.mount.entry=proc /var/lib/lxc/CN/rootfs/proc proc nodev,noexec,nosuid 0 0
lxc.mount = /var/lib/lxc/CN/fstab
```

The first line says that the container's root filesystem is already mounted at `/var/lib/lxc/CN/rootfs`. If the filesystem is a block device (such as an LVM logical volume), then the path to the block device must be given instead.

Each **lxc.mount.entry** line should contain an item to mount in valid fstab format. The target directory should be prefixed by `/var/lib/lxc/CN/rootfs`, even if **lxc.rootfs** points to a block device.

Finally, **lxc.mount** points to a file, in fstab format, containing further items to mount. Note that all of these entries will be mounted by the host before the container init is started. In this way it is possible to bind mount various directories from the host into the container.

5.4.5. Other configuration options

- **lxc.cap.drop** can be used to prevent the container from having or ever obtaining the listed capabilities. For instance, including

  ```
  lxc.cap.drop = sys_admin
  ```

  will prevent the container from mounting filesystems, as well as all other actions which require cap_sys_admin. See the **capabilities(7)** manual page for a list of capabilities and their meanings.
- **lxc.aa_profile = lxc-CN-profile** specifies a custom Apparmor profile in which to start the container. See *Sezione 5.2.6, «Apparmor» [345]* for more information.
- **lxc.console=/path/to/consolefile** will cause console messages to be written to the specified file.
- **lxc.arch** specifies the architecture for the container, for instance x86, or x86_64.
- **lxc.tty=5** specifies that 5 consoles (in addition to `/dev/console`) should be created. That is, consoles will be available on `/dev/tty1` through `/dev/tty5`. The ubuntu templates set this value to 4.

- **lxc.pts=1024** specifies that the container should have a private (Unix98) devpts filesystem mount. If this is not specified, then the container will share `/dev/pts` with the host, which is rarely desired. The number 1024 means that 1024 ptys should be allowed in the container, however this number is currently ignored. Before starting the container init, LXC will do (essentially) a

  ```
  sudo mount -t devpts -o newinstance devpts /dev/pts
  ```

  inside the container. It is important to realize that the container should not mount devpts filesystems of its own. It may safely do bind or move mounts of its mounted `/dev/pts`. But if it does

  ```
  sudo mount -t devpts devpts /dev/pts
  ```

  it will remount the host's devpts instance. If it adds the newinstance mount option, then it will mount a new private (empty) instance. In neither case will it remount the instance which was set up by LXC. For this reason, and to prevent the container from using the host's ptys, the default Apparmor policy will not allow containers to mount devpts filesystems after the container's init has been started.
- **lxc.devttydir** specifies a directory under `/dev` in which LXC will create its console devices. If this option is not specified, then the ptys will be bind-mounted over `/dev/console` and `/dev/ttyN`. However, rare package updates may try to blindly *rm -f* and then *mknod* those devices. They will fail (because the file has been bind-mounted), causing the package update to fail. When **lxc.devttydir** is set to LXC, for instance, then LXC will bind-mount the console ptys onto `/dev/lxc/console` and `/dev/lxc/ttyN`, and subsequently symbolically link them to `/dev/console` and `/dev/ttyN`. This allows the package updates to succeed, at the risk of making future gettys on those consoles fail until the next reboot. This problem will be ideally solved with device namespaces.

5.5. Updates in Ubuntu containers

Because of some limitations which are placed on containers, package upgrades at times can fail. For instance, a package install or upgrade might fail if it is not allowed to create or open a block device. This often blocks all future upgrades until the issue is resolved. In some cases, you can work around this by chrooting into the container, to avoid the container restrictions, and completing the upgrade in the chroot.

Some of the specific things known to occasionally impede package upgrades include:

- The container modifications performed when creating containers with the --trim option.
- Actions performed by lxcguest. For instance, because `/lib/init/fstab` is bind-mounted from another file, mountall upgrades which insist on replacing that file can fail.

- The over-mounting of console devices with ptys from the host can cause trouble with udev upgrades.
- Apparmor policy and devices cgroup restrictions can prevent package upgrades from performing certain actions.
- Capabilities dropped by use of **lxc.cap.drop** can likewise stop package upgrades from performing certain actions.

5.6. Libvirt LXC

Libvirt is a powerful hypervisor management solution with which you can administer Qemu, Xen and LXC virtual machines, both locally and remote. The libvirt LXC driver is a separate implementation from what we normally call *LXC*. A few differences include:

- Configuration is stored in xml format
- There no tools to facilitate container creation
- By default there is no console on `/dev/console`
- There is no support (yet) for container reboot or full shutdown

5.6.1. Converting a LXC container to libvirt-lxc

Sezione 5.3.1, «Creating Containers» [347] showed how to create LXC containers. If you've created a valid LXC container in this way, you can manage it with libvirt. Fetch a sample xml file from

```
wget http://people.canonical.com/~serge/o1.xml
```

Edit this file to replace the container name and root filesystem locations. Then you can define the container with:

```
virsh -c lxc:/// define o1.xml
```

5.6.2. Creating a container from cloud image

If you prefer to create a pristine new container just for LXC, you can download an ubuntu cloud image, extract it, and point a libvirt LXC xml file to it. For instance, find the url for a root tarball for the latest daily Ubuntu 12.04 LTS cloud image using

```
url1=`ubuntu-cloudimg-query precise daily $arch --format "%{url}\n"`
url=`echo $url1 | sed -e 's/.tar.gz/-root\0/'`
```

```
wget $url
filename=`basename $url`
```

Extract the downloaded tarball, for instance

```
mkdir $HOME/c1
cd $HOME/c1
sudo tar zxf $filename
```

Download the xml template

```
wget http://people.canonical.com/~serge/o1.xml
```

In the xml template, replace the name o1 with c1 and the source directory `/var/lib/lxc/o1/rootfs` with `$HOME/c1`. Then define the container using

```
virsh define o1.xml
```

5.6.3. Interacting with libvirt containers

As we've seen, you can create a libvirt-lxc container using

```
virsh -c lxc:/// define container.xml
```

To start a container called *container*, use

```
virsh -c lxc:/// start container
```

To stop a running container, use

```
virsh -c lxc:/// destroy container
```

Note that whereas the **lxc-destroy** command deletes the container, the **virsh destroy** command stops a running container. To delete the container definition, use

```
virsh -c lxc:/// undefine container
```

To get a console to a running container, use

```
virsh -c lxc:/// console container
```

Exit the console by simultaneously pressing control and].

5.7. The lxcguest package

In the 11.04 (Natty) and 11.10 (Oneiric) releases of Ubuntu, a package was introduced called *lxcguest*. An unmodified root image could not be safely booted inside a container, but an image with the lxcguest package installed could be booted as a container, on bare hardware, or in a Xen, kvm, or VMware virtual machine.

As of the 12.04 LTS release, the work previously done by the lxcguest package was pushed into the core packages, and the lxcguest package was removed. As a result, an unmodified 12.04 LTS image can be booted as a container, on bare hardware, or in a Xen, kvm, or VMware virtual machine. To use an older release, the lxcguest package should still be used.

5.8. Sicurezza

A namespace maps ids to resources. By not providing a container any id with which to reference a resource, the resource can be protected. This is the basis of some of the security afforded to container users. For instance, IPC namespaces are completely isolated. Other namespaces, however, have various *leaks* which allow privilege to be inappropriately exerted from a container into another container or to the host.

By default, LXC containers are started under a Apparmor policy to restrict some actions. However, while stronger security is a goal for future releases, in 12.04 LTS the goal of the Apparmor policy is not to stop malicious actions but rather to stop accidental harm of the host by the guest.

See the *LXC security*[38] wiki page for more, uptodate information.

5.8.1. Exploitable system calls

It is a core container feature that containers share a kernel with the host. Therefore, if the kernel contains any exploitable system calls, the container can exploit these as well. Once the container controls the kernel it can fully control any resource known to the host.

[38] http://wiki.ubuntu.com/LxcSecurity

5.9. Risorse

- The DeveloperWorks article *LXC: Linux container tools*[39] was an early introduction to the use of containers.
- The *Secure Containers Cookbook*[40] demonstrated the use of security modules to make containers more secure.
- Manual pages referenced above can be found at:

  `capabilities`[41]
  `lxc.conf`[42]
- The upstream LXC project is hosted at *Sourceforge*[43].
- LXC security issues are listed and discussed at *the LXC Security wiki page*[44]
- For more on namespaces in Linux, see: S. Bhattiprolu, E. W. Biederman, S. E. Hallyn, and D. Lezcano. Virtual Servers and Check- point/Restart in Mainstream Linux. SIGOPS Op- erating Systems Review, 42(5), 2008.

[39] https://www.ibm.com/developerworks/linux/library/l-lxc-containers/
[40] http://www.ibm.com/developerworks/linux/library/l-lxc-security/index.html
[41] http://manpages.ubuntu.com/manpages/en/man7/capabilities.7.html
[42] http://manpages.ubuntu.com/manpages/en/man5/lxc.conf.5.html
[43] http://lxc.sf.net
[44] http://wiki.ubuntu.com/LxcSecurity

Capitolo 21. Cluster

1. DRBD

DRDB (Distributed Replicated Block Device) replica i device a blocchi tra diversi host. La replica è trasparente alle applicazioni sul sistema host e qualsiasi device a blocchi (disco fisso, partizione, RAID, volume logico) può essere replicato.

Per utilizzare drbd, per prima cosa è necessario installare i pacchetti necessari. In un terminale digitare:

```
sudo apt-get install drbd8-utils
```

Se si sta usando il *kernel virtuale* come parte di una macchina virtuale, è necessario compilare il modulo drbd. Potrebbe anche essere più semplice installare il pacchetto linux-server nella macchina virtuale.

In questa sezione viene indicato come configurare drbd per replicare tra due host una partizione `/srv` separata con file system ext3. La dimensione della partizione non è rilevante, ma entrambe le partizioni devono avere la stessa dimensione.

1.1. Configurazione

I due host in questo esempio sono chiamati *drbd01* e *drbd02* ed è necessario configurarne la risoluzione del nome attraverso DNS o con il file `/etc/hosts`. Per maggiori informazioni, consultare *Capitolo 8, DNS (Domain Name Service) [140]*.

- Per configurare drbd, sul primo host modificare il file `/etc/drbd.conf`:

```
global { usage-count no; }
common { syncer { rate 100M; } }
resource r0 {
        protocol C;
        startup {
                wfc-timeout  15;
                degr-wfc-timeout 60;
        }
        net {
                cram-hmac-alg sha1;
                shared-secret "secret";
        }
        on drbd01 {
                device /dev/drbd0;
                disk /dev/sdb1;
                address 192.168.0.1:7788;
                meta-disk internal;
        }
        on drbd02 {
                device /dev/drbd0;
                disk /dev/sdb1;
```

```
                address 192.168.0.2:7788;
                meta-disk internal;
        }
}
```

All'interno del file `/etc/drbd.conf` sono disponibili molte opzioni, ma per questo esempio i valori predefinito sono sufficienti.

- Copiare il file `/etc/drbd.conf` sul secondo host:

```
scp /etc/drbd.conf drbd02:~
```

- Sull'host *drbd02*, spostare il file in `/etc`:

```
sudo mv drbd.conf /etc/
```

- Utilizzando l'utilità drbdadm, inizializzare l'archivio dei meta-dati. Su ogni singolo server eseguire il seguente comando:

```
sudo drbdadm create-md r0
```

- Su entrambi gli host, avviare il demone drbd:

```
sudo /etc/init.d/drbd start
```

- Sull'host *drbd01*, o su qualsiasi host primario configurato, digitare:

```
sudo drbdadm -- --overwrite-data-of-peer primary all
```

- Una volta eseguito il comando precedente, inizierà la sincronizzazione dei dati con l'host secondario. Per visualizzare l'avanzamento, su *drbd02*, digitare il seguente comando:

```
watch -n1 cat /proc/drbd
```

  Per fermare l'operazione di controllo, premere *Ctrl+c*.

- Infine, aggiungere un file system a `/dev/drbd0` e montarlo:

```
sudo mkfs.ext3 /dev/drbd0
sudo mount /dev/drbd0 /srv
```

1.2. Test

Per verificare che i dati siano effettivamente sincronizzati tra gli host, copiare alcuni file sull'host primario, *drbd01*, nella directory `/srv`:

```
sudo cp -r /etc/default /srv
```

Smontare `/srv`:

```
sudo umount /srv
```

Retrocedere il server *primario* a ruolo di *secondario*:

```
sudo drbdadm secondary r0
```

Ora, *promuovere* il server *secondario* a *primario*:

```
sudo drbdadm primary r0
```

Per completare, montare la partizione:

```
sudo mount /dev/drbd0 /srv
```

Usando *ls* dovrebbe essere possibile vedere il file `/srv/default` copiato dal precedente host *primario drbd01*.

1.3. Riferimenti

- Per maggiori informazioni riguardo DRBD, consultare il *sito web di DRBD*[1].
- The *drbd.conf man page*[2] contains details on the options not covered in this guide.
- Also, see the *drbdadm man page*[3].
- Ulteriori informazioni sono disponibili nella *documentazione online*[4].

[1] http://www.drbd.org/
[2] http://manpages.ubuntu.com/manpages/precise/en/man5/drbd.conf.5.html
[3] http://manpages.ubuntu.com/manpages/precise/en/man8/drbdadm.8.html
[4] https://help.ubuntu.com/community/DRBD

Capitolo 22. VPN

OpenVPN is a Virtual Private Networking (VPN) solution provided in the Ubuntu Repositories. It is flexible, reliable and secure. It belongs to the family of SSL/TLS VPN stacks (different from IPSec VPNs). This chapter will cover installing and configuring OpenVPN to create a VPN.

1. OpenVPN

If you want more than just pre-shared keys OpenVPN makes it easy to setup and use a Public Key Infrastructure (PKI) to use SSL/TLS certificates for authentication and key exchange between the VPN server and clients. OpenVPN can be used in a routed or bridged VPN mode and can be configured to use either UDP or TCP. The port number can be configured as well, but port 1194 is the official one. And it is only using that single port for all communication. VPN client implementations are available for almost anything including all Linux distributions, OS X, Windows and OpenWRT based WLAN routers.

1.1. Server Installation

Per installare openvpn, in un terminale, digitare:

```
sudo apt-get install openvpn
```

1.2. Public Key Infrastructure Setup

The first step in building an OpenVPN configuration is to establish a PKI (public key infrastructure). The PKI consists of:

- a separate certificate (also known as a public key) and private key for the server and each client, and
- a master Certificate Authority (CA) certificate and key which is used to sign each of the server and client certificates.

OpenVPN supports bidirectional authentication based on certificates, meaning that the client must authenticate the server certificate and the server must authenticate the client certificate before mutual trust is established.

Both server and client will authenticate the other by first verifying that the presented certificate was signed by the master certificate authority (CA), and then by testing information in the now-authenticated certificate header, such as the certificate common name or certificate type (client or server).

1.2.1. Certificate Authority Setup

To setup your own Certificate Authority (CA) and generating certificates and keys for an OpenVPN server and multiple clients first copy the `easy-rsa` directory to `/etc/openvpn`. This will ensure that any changes to the scripts will not be lost when the package is updated. From a terminal change to user root and:

```
mkdir /etc/openvpn/easy-rsa/
cp -r /usr/share/doc/openvpn/examples/easy-rsa/2.0/* /etc/openvpn/easy-rsa/
```

Modificare quindi il file `/etc/openvpn/easy-rsa/vars` sistemando quanto segue al proprio ambiente:

```
export KEY_COUNTRY="IT"
export KEY_PROVINCE="Roma"
export KEY_CITY="Roma"
export KEY_ORG="Società di esempio"
export KEY_EMAIL="mario@example.com"
```

Enter the following to generate the master Certificate Authority (CA) certificate and key:

```
cd /etc/openvpn/easy-rsa/
source vars
./clean-all
./build-ca
```

1.2.2. Certificati server

Next, we will generate a certificate and private key for the server:

```
./build-key-server myservername
```

As in the previous step, most parameters can be defaulted. Two other queries require positive responses, "Sign the certificate? [y/n]" and "1 out of 1 certificate requests certified, commit? [y/n]".

Diffie Hellman parameters must be generated for the OpenVPN server:

```
./build-dh
```

All certificates and keys have been generated in the subdirectory keys/. Common practice is to copy them to /etc/openvpn/:

```
cd keys/
cp myservername.crt myservername.key ca.crt dh1024.pem /etc/openvpn/
```

1.2.3. Certificati client

The VPN client will also need a certificate to authenticate itself to the server. Usually you create a different certificate for each client. To create the certificate, enter the following in a terminal while being user root:

```
cd /etc/openvpn/easy-rsa/
source vars
./build-key client1
```

Copy the following files to the client using a secure method:

- /etc/openvpn/ca.crt
- /etc/openvpn/easy-rsa/keys/client1.crt
- /etc/openvpn/easy-rsa/keys/client1.key

As the client certificates and keys are only required on the client machine, you should remove them from the server.

1.3. Simple Server Configuration

Along with your OpenVPN installation you got these sample config files (and many more if if you check):

```
root@server:/# ls -l /usr/share/doc/openvpn/examples/sample-config-files/
total 68
-rw-r--r-- 1 root root 3427 2011-07-04 15:09 client.conf
-rw-r--r-- 1 root root 4141 2011-07-04 15:09 server.conf.gz
```

Start with copying and unpacking server.conf.gz to /etc/openvpn/server.conf.

```
sudo cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz /etc/openvpn/
sudo gzip -d /etc/openvpn/server.conf.gz
```

Edit `/etc/openvpn/server.conf` to make sure the following lines are pointing to the certificates and keys you created in the section above.

```
ca ca.crt
cert myservername.crt
key myservername.key
dh dh1024.pem
```

That is the minimum you have to configure to get a working OpenVPN server. You can use all the default settings in the sample server.conf file. Now start the server. You will find logging and error messages in your syslog.

```
root@server:/etc/openvpn# /etc/init.d/openvpn start
 * Starting virtual private network daemon(s)...
   *   Autostarting VPN 'server'                       [ OK ]
```

Now check if OpenVPN created a tun0 interface:

```
root@server:/etc/openvpn# ifconfig tun0
tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:10.8.0.1  P-t-P:10.8.0.2  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
[...]
```

1.4. Simple Client Configuration

There are various different OpenVPN client implementations with and without GUIs. You can read more about clients in a later section. For now we use the OpenVPN client for Ubuntu which is the same executable as the server. So you have to install the openvpn package again on the client machine:

```
sudo apt-get install openvpn
```

This time copy the client.conf sample config file to /etc/openvpn/.

```
sudo cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf /etc/openvpn/
```

Copy the client keys and the certificate of the CA you created in the section above to e.g. /etc/openvpn/ and edit `/etc/openvpn/client.conf` to make sure the following lines are pointing to those files. If you have the files in /etc/openvpn/ you can omit the path.

```
ca ca.crt
cert client1.crt
key client1.key
```

And you have to at least specify the OpenVPN server name or address. Make sure the keyword client is in the config. That's what enables client mode.

```
client
remote vpnserver.example.com 1194
```

Now start the OpenVPN client:

```
root@client:/etc/openvpn# /etc/init.d/openvpn start
 * Starting virtual private network daemon(s)...
   *   Autostarting VPN 'client'                          [ OK ]
```

Check if it created a tun0 interface:

```
root@client:/etc/openvpn# ifconfig tun0
tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:10.8.0.6  P-t-P:10.8.0.5  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
```

Check if you can ping the OpenVPN server:

```
root@client:/etc/openvpn# ping 10.8.0.1
PING 10.8.0.1 (10.8.0.1) 56(84) bytes of data.
64 bytes from 10.8.0.1: icmp_req=1 ttl=64 time=0.920 ms
```

The OpenVPN server always uses the first usable IP address in the client network and only that IP is pingable. E.g. if you configured a /24 for the client network mask, the .1 address will be used. The P-t-P address you see in the ifconfig output above is usually not answering ping requests.

Check out your routes:

```
root@client:/etc/openvpn# netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
10.8.0.5        0.0.0.0         255.255.255.255 UH        0 0          0 tun0
10.8.0.1        10.8.0.5        255.255.255.255 UGH       0 0          0 tun0
192.168.42.0    0.0.0.0         255.255.255.0   U         0 0          0 eth0
0.0.0.0         192.168.42.1    0.0.0.0         UG        0 0          0 eth0
```

1.5. First trouble shooting

If the above didn't work for you, check this:

- Check your syslog, e.g. grep -i vpn /var/log/syslog
- Can the client connect to the server machine? Maybe a firewall is blocking access? Check syslog on server.
- Client and server must use same protocol and port, e.g. UDP port 1194, see port and proto config option
- Client and server must use same config regarding compression, see comp-lzo config option
- Client and server must use same config regarding bridged vs routed mode, see server vs server-bridge config option

1.6. Advanved configuration

1.6.1. Advanced routed VPN configuration on server

The above is a very simple working VPN. The client can access services on the VPN server machine through an encrypted tunnel. If you want to reach more servers or anything in other networks, push some routes to the clients. E.g. if your company's network can be summarized to the network 192.168.0.0/16, you could push this route to the clients. But you will also have to change the routing for the way back - your servers need to know a route to the VPN client-network.

Or you might push a default gateway to all the clients to send all their internet traffic to the VPN gateway first and from there via the company firewall into the internet. This section shows you some possible options.

Push routes to the client to allow it to reach other private subnets behind the server. Remember that these private subnets will also need to know to route the OpenVPN client address pool (10.8.0.0/24) back to the OpenVPN server.

```
push "route 10.0.0.0 255.0.0.0"
```

If enabled, this directive will configure all clients to redirect their default network gateway through the VPN, causing all IP traffic such as web browsing and and DNS lookups to go through the VPN (the OpenVPN server machine or your central firewall may need to NAT the TUN/TAP interface to the internet in order for this to work properly).

```
push "redirect-gateway def1 bypass-dhcp"
```

Configure server mode and supply a VPN subnet for OpenVPN to draw client addresses from. The server will take 10.8.0.1 for itself, the rest will be made available to clients. Each client will be able to reach the server on 10.8.0.1. Comment this line out if you are ethernet bridging.

```
server 10.8.0.0 255.255.255.0
```

Maintain a record of client to virtual IP address associations in this file. If OpenVPN goes down or is restarted, reconnecting clients can be assigned the same virtual IP address from the pool that was previously assigned.

```
ifconfig-pool-persist ipp.txt
```

Push DNS servers to the client.

```
push "dhcp-option DNS 10.0.0.2"
push "dhcp-option DNS 10.1.0.2"
```

Allow client to client communication.

```
client-to-client
```

Enable compression on the VPN link.

```
comp-lzo
```

The keepalive directive causes ping-like messages to be sent back and forth over the link so that each side knows when the other side has gone down. Ping every 1 second, assume that remote peer is down if no ping received during a 3 second time period.

```
keepalive 1 3
```

It's a good idea to reduce the OpenVPN daemon's privileges after initialization.

```
user nobody
group nogroup
```

OpenVPN 2.0 includes a feature that allows the OpenVPN server to securely obtain a username and password from a connecting client, and to use that information as a basis for authenticating the client. To use this authentication method, first add the auth-user-pass directive to the client configuration. It will direct the OpenVPN client to query the user for a username/password, passing it on to the server over the secure TLS channel.

```
# client config!
auth-user-pass
```

This will tell the OpenVPN server to validate the username/password entered by clients using the login PAM module. Useful if you have centralized authentication with e.g. Kerberos.

```
plugin /usr/lib/openvpn/openvpn-auth-pam.so login
```

Please read the OpenVPN *hardening security guide*[1] for further security advice.

1.6.2. Advanced bridged VPN configuration on server

OpenVPN can be setup for either a routed or a bridged VPN mode. Sometimes this is also referred to as OSI layer-2 versus layer-3 VPN. In a bridged VPN all layer-2 frames - e.g. all ethernet frames - are sent to the VPN partners and in a routed VPN only layer-3 packets are sent to VPN partners. In bridged mode all traffic including traffic which was traditionally LAN-local like local network broadcasts, DHCP requests, ARP requests etc. are sent to VPN partners whereas in routed mode this would be filtered.

1.6.2.1. Prepare interface config for bridging on server

Make sure you have the bridge-utils package installed:

```
sudo apt-get install bridge-utils
```

Before you setup OpenVPN in bridged mode you need to change your interface configuration. Let's assume your server has an interface eth0 connected to the internet and an interface eth1 connected to the LAN you want to bridge. Your /etc/network/interfaces would like this:

```
auto eth0
iface eth0 inet static
  address 1.2.3.4
  netmask 255.255.255.248
  default 1.2.3.1

auto eth1
iface eth1 inet static
```

[1] http://openvpn.net/index.php/open-source/documentation/howto.html#security

```
  address 10.0.0.4
  netmask 255.255.255.0
```

This straight forward interface config needs to be changed into a bridged mode like where the config of interface eth1 moves to the new br0 interface. Plus we configure that br0 should bridge interface eth1. We also need to make sure that interface eth1 is always in promiscuous mode - this tells the interface to forward all ethernet frames to the IP stack.

```
auto eth0
iface eth0 inet static
  address 1.2.3.4
  netmask 255.255.255.248
  default 1.2.3.1

auto eth1
iface eth1 inet manual
  up ip link set $IFACE up promisc on

auto br0
iface br0 inet static
  address 10.0.0.4
  netmask 255.255.255.0
  bridge_ports eth1
```

At this point you need to restart networking. Be prepared that this might not work as expected and that you will lose remote connectivity. Make sure you can solve problems having local access.

```
sudo /etc/init.d/network restart
```

1.6.2.2. Prepare server config for bridging

Modificare il file `/etc/openvpn/server.conf` cambiando le seguenti opzioni:

```
;dev tun
dev tap
up "/etc/openvpn/up.sh br0 eth1"
;server 10.8.0.0 255.255.255.0
server-bridge 10.0.0.4 255.255.255.0 10.0.0.128 10.0.0.254
```

Next, create a helper script to add the *tap* interface to the bridge and to ensure that eth1 is promiscuous mode. Create `/etc/openvpn/up.sh`:

```
#!/bin/sh

BR=$1
ETHDEV=$2
TAPDEV=$3

/sbin/ip link set "$TAPDEV" up
```

```
/sbin/ip link set "$ETHDEV" promisc on
/sbin/brctl addif $BR $TAPDEV
```

Then make it executable:

```
sudo chmod 755 /etc/openvpn/up.sh
```

Una volta configurato il server, riavviare openvpn digitando:

```
sudo /etc/init.d/openvpn restart
```

1.6.2.3. Configurazione del client

Installare openvpn sul client:

```
sudo apt-get install openvpn
```

Configurato il server e copiati i certificati del client nella directory `/etc/openvpn/`, creare un file di configurazione per il client copiando l'esempio. Nel computer client, da un terminale, digitare:

```
sudo cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf /etc/openvpn
```

Ora modificare `/etc/openvpn/client.conf` sistemando le seguenti opzioni:

```
dev tap
;dev tun
```

Infine, riavviare openvpn:

```
sudo /etc/init.d/openvpn restart
```

Ora dovrebbe essere possibile connettersi alla rete LAN remota attraverso VPN.

1.7. Client software implementations

1.7.1. Linux Network-Manager GUI for OpenVPN

Many Linux distributions including Ubuntu desktop variants come with Network Manager, a nice GUI to configure your network settings. It also can manage your VPN connections. Make sure you have package network-manager-openvpn installed. Here you see that the installation installs all other required packages as well:

```
root@client:~# apt-get install network-manager-openvpn
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

```
The following extra packages will be installed:
  liblzo2-2 libpkcs11-helper1 network-manager-openvpn-gnome openvpn
Suggested packages:
  resolvconf
The following NEW packages will be installed:
  liblzo2-2 libpkcs11-helper1 network-manager-openvpn
  network-manager-openvpn-gnome openvpn
0 upgraded, 5 newly installed, 0 to remove and 631 not upgraded.
Need to get 700 kB of archives.
After this operation, 3,031 kB of additional disk space will be used.
Do you want to continue [Y/n]?
```

To inform network-manager about the new installed packages you will have to restart it:

```
root@client:~# restart network-manager
network-manager start/running, process 3078
```

Open the Network Manager GUI, select the VPN tab and then the 'Add' button. Select OpenVPN as the VPN type in the opening requester and press 'Create'. In the next window add the OpenVPN's server name as the 'Gateway', set 'Type' to 'Certificates (TLS)', point 'User Certificate' to your user certificate, 'CA Certificate' to your CA certificate and 'Private Key' to your private key file. Use the advanced button to enable compression or other special settings you set on the server. Now try to establish your VPN.

1.7.2. OpenVPN with GUI for Mac OS X: Tunnelblick

Tunnelblick is an excellent free, open source implementation of a GUI for OpenVPN for OS X. The project's homepage is at *http://code.google.com/p/tunnelblick/*. Download the latest OS X installer from there and install it. Then put your client.ovpn config file together with the certificates and keys in /Users/username/Library/Application Support/Tunnelblick/Configurations/ and lauch Tunnelblick from your Application folder.

```
# sample client.ovpn for Tunnelblick
client
remote blue.example.com
port 1194
proto udp
dev tun
dev-type tun
ns-cert-type server
reneg-sec 86400
auth-user-pass
auth-nocache
auth-retry interact
comp-lzo yes
verb 3
ca ca.crt
cert client.crt
key client.key
```

1.7.3. OpenVPN with GUI for Win 7

First download and install the latest *OpenVPN Windows Installer*[2]. OpenVPN 2.2.1 was the latest when this was written. Additionally download an alternative Open VPN Windows GUI. The OpenVPN MI GUI from *http://openvpn-mi-gui.inside-security.de* seems to be a nice one for Windows 7. Download the latest version. 20110624 was the latest version when this was written.

You need to start the OpenVPN service. Goto Start > Computer > Manage > Services and Applications > Services. Find the OpenVPN service and start it. Set it's startup type to automatic. When you start the OpenVPN MI GUI the first time you need to run it as an administrator. You have to right click on it and you will see that option.

You will have to write your OpenVPN config in a textfile and place it in C:\Program Files\OpenVPN \config\client.ovpn along with the CA certificate. You could put the user certificate in the user's home directory like in the follwing example.

```
# C:\Program Files\OpenVPN\config\client.ovpn
client
remote server.example.com
port 1194
proto udp
dev tun
dev-type tun
ns-cert-type server
reneg-sec 86400
auth-user-pass
auth-retry interact
comp-lzo yes
verb 3
ca ca.crt
cert "C:\\Users\\username\\My Documents\\openvpn\\client.crt"
key "C:\\Users\\username\\My Documents\\openvpn\\client.key"
management 127.0.0.1 1194
management-hold
management-query-passwords
auth-retry interact
```

1.7.4. OpenVPN for OpenWRT

OpenWRT is described as a Linux distribution for embedded devices like WLAN router. There are certain types of WLAN routers who can be flashed to run OpenWRT. Depending on the available memory on your OpenWRT router you can run software like OpenVPN and you could for example build a small inexpensive branch office router with VPN connectivity to the central office. More info on OpenVPN on OpenWRT is *here*[3]. And here is the OpenWRT project's homepage: *http:// openwrt.org*

[2] http://www.openvpn.net/index.php/open-source/downloads.html
[3] http://wiki.openwrt.org/doc/howto/vpn.overview

Log into your OpenWRT router and install OpenVPN:

```
opkg update
opkg install openvpn
```

Check out /etc/config/openvpn and put you client config in there. Copy certificated and keys to /etc/openvpn/

```
config openvpn client1
        option enable 1
        option client 1
#       option dev tap
        option dev tun
        option proto udp
        option ca /etc/openvpn/ca.crt
        option cert /etc/openvpn/client.crt
        option key /etc/openvpn/client.key
        option comp_lzo 1
```

Restart OpenVPN:

```
/etc/init.d/openvpn restart
```

You will have to see if you need to adjust your router's routing and firewall rules.

1.8. Riferimenti

- Per maggiori informazioni, consultare il sito web di *OpenVPN*[4].
- *OpenVPN hardening security guide*[5]
- Un'ottima risorsa è anche *OpenVPN: Building and Integrating Virtual Private Networks*[6] di Pakt (in inglese).

[4] http://openvpn.net/

[5] http://openvpn.net/index.php/open-source/documentation/howto.html#security

[6] http://www.packtpub.com/openvpn/book

Capitolo 23. Altre utili applicazioni

Esistono molte applicazioni sviluppate dallo Ubuntu Server Team e altre integrate all'interno della Ubuntu Server Edition che non sono molto conosciute. Questo capitolo presenta alcune di queste utili applicazioni che possono rendere l'amministrazione di un server Ubuntu, o di molti server, più facile.

1. pam_motd

Quando si esegue l'accesso con una versione server di Ubuntu, è possibile vedere dei messaggi giornalieri di informazioni (MOTD). Queste informazioni sono ricavate e visualizzate utilizzando diversi pacchetti:

- *landscape-common*: fornisce le librerie principali di landscape-client, che può essere utilizzato per la gestione di sistemi attraverso l'interfaccia web di *Landscape*. Il pacchetto comprende l'utilità / usr/bin/landscape-sysinfo che può essere usata per recuperare informazioni visualizzate attraverso il MOTD.
- *update-notifier-common:* is used to automatically update the MOTD via pam_motd module.

pam_motd executes the scripts in `/etc/update-motd.d` in order based on the number prepended to the script. The output of the scripts is written to `/var/run/motd`, keeping the numerical order, then concatenated with `/etc/motd.tail`.

È possibile aggiungere delle informazioni dinamiche per il messaggio giornaliero. Per esempio, per aggiungere informazioni meteo locali:

- Installare il pacchetto weather-util:

  ```
  sudo apt-get install weather-util
  ```

- L'utilità weather utilizza i dati METAR dalla «National Oceanic and Atmospheric Administration» e le previsioni meteo dal «National Weather Service». Per reperire informazioni locali è necessario il codice a 4 cifre ICAO. Per ottenere questo codice è possibile consultare il sito web del *National Weather Service*[1].

  Benché il «National Weather Service» sia un'agenzia governativa degli Stati Uniti d'America, stazioni meteo sono disponibili in tutti il mondo. Informazioni meteorologiche potrebbero però non essere disponibili per tutte le località al di fuori del territorio americano.

- Creare il file `/usr/local/bin/local-weather`, un semplice script per usare weather con il proprio indicatore ICAO locale:

  ```
  #!/bin/sh
  #
  #
  # Prints the local weather information for the MOTD.
  #
  #

  # Replace KINT with your local weather station.
  # Local stations can be found here: http://www.weather.gov/tg/siteloc.shtml

  echo
  ```

[1] http://www.weather.gov/tg/siteloc.shtml

```
weather -i KINT
echo
```

- Rendere lo script eseguibile:

```
sudo chmod 755 /usr/local/bin/local-weather
```

- Next, create a symlink to `/etc/update-motd.d/98-local-weather`:

```
sudo ln -s /usr/local/bin/local-weather /etc/update-motd.d/98-local-weather
```

- Finally, exit the server and re-login to view the new MOTD.

You should now be greeted with some useful information, and some information about the local weather that may not be quite so useful. Hopefully the local-weather example demonstrates the flexibility of pam_motd.

2. etckeeper

etckeeper consente di archiviare il contenuto della directory `/etc` in un sistema di controllo della versione e si integra con apt per inviare le modifiche apportate a `/etc` quando vengono installati o aggiornati pacchetti. Utilizzare un sistema di controllo della versione per gestire la directory `/etc` è considerata una "best practice" e l'obiettivo di etckeeper è quello di rendere questo processo il più facile possibile.

Installare etckeeper digitando quanto segue in un terminale:

```
sudo apt-get install etckeeper
```

Il file di configurazione principale, `/etc/etckeeper/etckeeper.conf`, è molto semplice. L'opzione principale è quale VCS usare. Come impostazione predefinita, etckeeper utilizza bzr per il controllo della versione. Il repository viene automaticamente inizializzato (e viene eseguito il primo commit) durante l'installazione del pacchetto. È possibile annullare questo inserendo il seguente comando:

```
sudo etckeeper uninit
```

Il programma etckeeper esegue i commint delle modifiche a /etc giornalmente.
Questo comportamento può essere disabilitato usando l'opzione di configurazione AVOID_DAILY_AUTOCOMMITS. Inoltre, esegue i commit delle modifiche prima di ogni installazione di un pacchetto. Per un tracciamento delle modifiche più preciso, è consigliato eseguire i commit manualmente aggiungendovi anche un messaggio di commit:

```
sudo etckeeper commit "..Commento sulle modifiche.."
```

Utilizzando i comandi del sistema di controllo è possibile visualizzare il registro delle informazioni riguardo i file in `/etc`:

```
sudo bzr log /etc/passwd
```

Per una dimostrazione dell'integrazione col sistema di gestione dei pacchetti, installare postfix:

```
sudo apt-get install postfix
```

Completata l'installazione, tutti i file di configurazione di postfix dovrebbero essere inviati al repository:

```
Committing to: /etc/
added aliases.db
modified group
modified group-
modified gshadow
```

```
modified gshadow-
modified passwd
modified passwd-
added postfix
added resolvconf
added rsyslog.d
modified shadow
modified shadow-
added init.d/postfix
added network/if-down.d/postfix
added network/if-up.d/postfix
added postfix/dynamicmaps.cf
added postfix/main.cf
added postfix/master.cf
added postfix/post-install
added postfix/postfix-files
added postfix/postfix-script
added postfix/sasl
added ppp/ip-down.d
added ppp/ip-down.d/postfix
added ppp/ip-up.d/postfix
added rc0.d/K20postfix
added rc1.d/K20postfix
added rc2.d/S20postfix
added rc3.d/S20postfix
added rc4.d/S20postfix
added rc5.d/S20postfix
added rc6.d/K20postfix
added resolvconf/update-libc.d
added resolvconf/update-libc.d/postfix
added rsyslog.d/postfix.conf
added ufw/applications.d/postfix
Committed revision 2.
```

Per un esempio di come etckeeper tiene traccia delle modifiche manuali, aggiungere un nuovo host in `/etc/hosts`. Usando bzr è possibile visualizzare quali file sono stati modificati:

```
sudo bzr status /etc/
modified:
  hosts
```

Ora inviare le modifiche:

```
sudo etckeeper commit "nuovo host"
```

Per maggiori informazioni su bzr consultare *Sezione 1, «Bazaar» [268]*.

3. Byobu

One of the most useful applications for any system administrator is screen. It allows the execution of multiple shells in one terminal. To make some of the advanced screen features more user friendly, and provide some useful information about the system, the byobu package was created.

When executing byobu pressing the *F9* key will bring up the Configuration menu. This menu will allow you to:

- Visualizzare il menù dell'aiuto
- Change Byobu's background color
- Change Byobu's foreground color
- Toggle status notifications
- Modificare le associazioni dei tasti
- Modificare la sequenza di escape
- Create new windows
- Gestire le finestre predefinite
- Byobu currently does not launch at login (toggle on)

The *key bindings* determine such things as the escape sequence, new window, change window, etc. There are two key binding sets to choose from *f-keys* and *screen-escape-keys*. If you wish to use the original key bindings choose the *none* set.

byobu provides a menu which displays the Ubuntu release, processor information, memory information, and the time and date. The effect is similar to a desktop menu.

Using the *"Byobu currently does not launch at login (toggle on)"* option will cause byobu to be executed any time a terminal is opened. Changes made to byobu are on a per user basis, and will not affect other users on the system.

One difference when using byobu is the *scrollback* mode. Press the *F7* key to enter scrollback mode. Scrollback mode allows you to navigate past output using *vi* like commands. Here is a quick list of movement commands:

- *h*: sposta il cursore a sinistra di un carattere
- *j*: sposta il cursore in giù di una riga
- *k*: sposta il cursore in su di una riga
- *l*: sposta il cursore a destra di un carattere
- *0*: va all'inizio della riga attuale
- *$*: va alla fine della riga attuale
- *G*: va alla riga specificata (come valore predefinito va alla fine del buffer)
- */*: cerca in avanti

- *?*: cerca all'indietro
- *n*: si sposta alla corrispondenza successiva, in avanti o all'indietro

4. Riferimenti

- See the *update-motd man page*[2] for more options available to update-motd.
- L'articolo di «The Debian Package of the Day» riguardo *weather*[3], presenta molte altre informazioni.
- Per maggiori informazioni riguardo l'uso di etckeeper, consultare il *sito web di etckeeper*[4].
- The *etckeeper Ubuntu Wiki*[5] page.
- Per maggiori informazioni riguardo bzr, consultare il *sito web di bzr*[6].
- Per maggiori informazioni riguardo screen, consultare il *sito web di screen*[7].
- And the *Ubuntu Wiki screen*[8] page.
- Also, see the byobu *project page*[9] for more information.

[2] http://manpages.ubuntu.com/manpages/precise/en/man1/update-motd.1.html
[3] http://debaday.debian.net/2007/10/04/weather-check-weather-conditions-and-forecasts-on-the-command-line/
[4] http://kitenet.net/~joey/code/etckeeper/
[5] https://help.ubuntu.com/community/etckeeper
[6] http://bazaar-vcs.org/
[7] http://www.gnu.org/software/screen/
[8] https://help.ubuntu.com/community/Screen
[9] https://launchpad.net/byobu

Appendice A. Appendix

1. Reporting Bugs in Ubuntu Server Edition

While the Ubuntu Project attempts to release software with as few bugs as possible, they do occur. You can help fix these bugs by reporting ones that you find to the project. The Ubuntu Project uses *Launchpad*[1] to track its bug reports. In order to file a bug about Ubuntu Server on Launchpad, you will need to *create an account*[2].

1.1. Reporting Bugs With ubuntu-bug

The preferred way to report a bug is with the ubuntu-bug command. The ubuntu-bug tool gathers information about the system useful to developers in diagnosing the reported problem that will then be included in the bug report filed on Launchpad. Bug reports in Ubuntu need to be filed against a specific software package, thus the name of the package that the bug occurs in needs to be given to ubuntu-bug:

```
ubuntu-bug PACKAGENAME
```

For example, to file a bug against the openssh-server package, you would do:

```
ubuntu-bug openssh-server
```

You can specify either a binary package or the source package for ubuntu-bug. Again using openssh-server as an example, you could also generate the report against the source package for openssh-server, openssh:

```
ubuntu-bug openssh
```

See *Capitolo 3, Gestione dei pacchetti [21]* for more information about packages in Ubuntu.

The ubuntu-bug command will gather information about the system in question, possibly including information specific to the specified package, and then ask you what you would like to do with collected information:

```
ubuntu-bug postgresql

*** Collecting problem information

The collected information can be sent to the developers to improve the
application. This might take a few minutes.
..........
```

[1] https://launchpad.net/

[2] https://help.launchpad.net/YourAccount/NewAccount

```
*** Send problem report to the developers?

After the problem report has been sent, please fill out the form in the
automatically opened web browser.

What would you like to do? Your options are:
  S: Send report (1.7 KiB)
  V: View report
  K: Keep report file for sending later or copying to somewhere else
  C: Cancel
Please choose (S/V/K/C):
```

The options available are:

- **Send Report** Selecting Send Report submits the collected information to Launchpad as part of the the process of filing a bug report. You will be given the opportunity to describe the situation that led up to the occurrence of the bug.

  ```
  *** Uploading problem information

  The collected information is being sent to the bug tracking system.
  This might take a few minutes.
  91%

  *** To continue, you must visit the following URL:

  https://bugs.launchpad.net/ubuntu/+source/postgresql-8.4/+filebug/kc6eSnTLnLxF8u0t3e56EukFeqJ?

  You can launch a browser now, or copy this URL into a browser on another
  computer.

  Choices:
    1: Launch a browser now
    C: Cancel
  Please choose (1/C):
  ```

  If you choose to start a browser, by default the text based web browser w3m will be used to finish filing the bug report. Alternately, you can copy the given URL to a currently running web browser.
- **View Report** Selecting View Report causes the collected information to be displayed to the terminal for review.

  ```
  Package: postgresql 8.4.2-2
  PackageArchitecture: all
  Tags: lucid
  ProblemType: Bug
  ProcEnviron:
    LANG=en_US.UTF-8
    SHELL=/bin/bash
  Uname: Linux 2.6.32-16-server x86_64
  Dependencies:
  ```

```
  adduser 3.112ubuntu1
  base-files 5.0.0ubuntu10
  base-passwd 3.5.22
  coreutils 7.4-2ubuntu2
...
```

After viewing the report, you will be brought back to the same menu asking what you would like to do with the report.

- **Keep Report File** Selecting Keep Report File causes the gathered information to be written to a file. This file can then be used to later file a bug report or transferred to a different Ubuntu system for reporting. To submit the report file, simply give it as an argument to the ubuntu-bug command:

```
What would you like to do? Your options are:
  S: Send report (1.7 KiB)
  V: View report
  K: Keep report file for sending later or copying to somewhere else
  C: Cancel
Please choose (S/V/K/C): k
Problem report file: /tmp/apport.postgresql.v4MQas.apport

ubuntu-bug /tmp/apport.postgresql.v4MQas.apport

*** Send problem report to the developers?
...
```

- **Cancel** Selecting Cancel causes the collected information to be discarded.

1.2. Reporting Application Crashes

The software package that provides the ubuntu-bug utility, apport, can be configured to trigger when applications crash. This is disabled by default, as capturing a crash can be resource intensive depending on how much memory the application that crashed was using as apport captures and processes the core dump.

Configuring apport to capture information about crashing applications requires a couple of steps. First, gdb needs to be installed; it is not installed by default in Ubuntu Server Edition.

```
sudo apt-get install gdb
```

See *Capitolo 3, Gestione dei pacchetti [21]* for more information about managing packages in Ubuntu.

Once you have ensured that gdb is installed, open the file `/etc/default/apport` in your text editor, and change the *enabled* setting to be **1** like so:

```
# set this to 0 to disable apport, or to 1 to enable it
# you can temporarily override this with
# sudo service apport start force_start=1
```

```
enabled=1

# set maximum core dump file size (default: 209715200 bytes == 200 MB)
maxsize=209715200
```

Once you have completed editing `/etc/default/apport`, start the apport service:

```
sudo start apport
```

After an application crashes, use the apport-cli command to search for the existing saved crash report information:

```
apport-cli

*** dash closed unexpectedly on 2010-03-11 at 21:40:59.

If you were not doing anything confidential (entering passwords or other
private information), you can help to improve the application by
reporting
the problem.

What would you like to do? Your options are:
  R: Report Problem...
  I: Cancel and ignore future crashes of this program version
  C: Cancel
Please choose (R/I/C):
```

Selecting *Report Problem* will walk you through similar steps as when using ubuntu-bug. One important difference is that a crash report will be marked as private when filed on Launchpad, meaning that it will be visible to only a limited set of bug triagers. These triagers will review the gathered data for private information before making the bug report publicly visible.

1.3. Risorse

- See the *Reporting Bugs*[3] Ubuntu wiki page.
- Also, the *Apport*[4] page has some useful information. Though some of it pertains to using a GUI.

[3] https://help.ubuntu.com/community/ReportingBugs
[4] https://wiki.ubuntu.com/Apport