

# README

joan carles llompart i seguí

April 14, 2017

## Contents

<b>1</b>	<b>About</b>	<b>2</b>
<b>2</b>	<b>Taula de continguts</b>	<b>TOC 2</b>
<b>3</b>	<b>Introducció</b>	<b>3</b>
3.1	<b>DONE</b> Context i justificació del treball . . . . .	3
3.2	<b>DONE</b> Objectius del treball . . . . .	3
3.3	<b>DONE</b> Enfocament i mètode seguit . . . . .	4
3.4	<b>DONE</b> Planificació del treball . . . . .	4
3.5	<b>TODO</b> Sumari de productes obtinguts . . . . .	4
3.6	<b>TODO</b> Descripció dels capítols de la memòria . . . . .	4
<b>4</b>	<b>Capítols</b>	<b>4</b>
4.1	Eines . . . . .	4
4.1.1	<b>DONE</b> GNU Emacs . . . . .	5
4.1.2	<b>DONE</b> Vi . . . . .	6
4.1.3	<b>DONE</b> VirtualBox . . . . .	6
4.1.4	<b>DONE</b> Scripts . . . . .	6
4.2	Xarxa . . . . .	7
4.2.1	<b>DONE</b> DNS . . . . .	7
4.2.2	<b>DONE</b> DHCP . . . . .	11
4.2.3	<b>TODO</b> NTP . . . . .	14
4.3	<b>TODO</b> Autenticació . . . . .	17
4.4	<b>TODO</b> Sistema de fitxers . . . . .	17
4.5	<b>TODO</b> Programari . . . . .	17
<b>5</b>	<b>Conclusions</b>	<b>17</b>

<b>6</b>	<b>Glossari</b>	<b>17</b>
<b>7</b>	<b>Bibliografia</b>	<b>17</b>
<b>8</b>	<b>Annexos</b>	<b>17</b>

## 1 About

Petit projecte que explica com implementar un sistema informàtic GNU/Linux dins una petita empresa. En concret, el projecte mostra com crear un servidor i implementar-hi els serveis de xarxa, autenticació, sistema de fitxers i distribució de programari. Es proveeix també d'eines per simular el projecte en màquines virtuals abans de posar-lo en producció.

## 2 Taula de continguts TOC

- About
- Introducció
  - Context i justificació del treball
  - Objectius del treball
  - Enfocament i mètode seguit
  - Planificació del treball
  - Sumari de productes obtinguts
  - Descripció dels capítols de la memòria
- Capítols
  - Eines
  - Xarxa
  - Autenticació
  - Sistema de fitxers
  - Programari
- Conclusions
- Glossari
- Bibliografia
- Annexos

## 3 Introducció

### 3.1 DONE Context i justificació del treball

Se suposa el següent escenari fictici. Una empresa X de nova constitució es veu en la necessitat d'instal·lar un sistema informàtic. Després de valorar les diferents opcions, decideix optar per un entorn linux.

Pel que fa al maquinari inicial, l'empresa haurà de fer una despesa inicial per a la compra de 20 ordinadors i 1 servidor, a part de la infraestructura pròpia de telecomunicacions.

Quant a l'entorn linux, s'han valorat les distribucions més importants que hi ha actualment al mercat, s'ha optat per Debian en el servidor i Ubuntu als escriptoris.

En la tria d'aquests sistemes operatius s'han tengut en compte aspectes com la fiabilitat, interoperabilitat i facilitat de maneig.

Debian i Ubuntu formen un bon conjunt per als requeriments de l'empresa. Ubuntu és la distribució linux molt extesa, àmpliament usada per als ordinadors tipus escriptori, de fàcil instal·lació, i amb una interfície d'usuari amigable. Per ser un sistema operatiu derivat de Debian, usa el sistema de paquets *dpkg*, la qual cosa aportarà una interoperabilitat completa entre el servidor i els ordinadors de sobretaula. Aquesta característica, en principi, no afecta el sistema en conjunt, però sí és cert que fa l'administració del sistema més senzilla. En canvi, es tria Debian per al servidor perquè és una distribució molt testejada i fiable, i no instal·la paqueteria extra com és el cas d'Ubuntu.

S'han descartat altres distribucions, com ara CentOS o ArchLinux, perquè si bé són distribucions amb característiques que els fan potents, no són tan fàcils de manejar a nivell d'usuari com les distribucions Ubuntu.

### 3.2 DONE Objectius del treball

Per a què aquest objectiu principal s'assoleixi, s'hauran d'aconseguir els objectius parcials següents:

- Creació d'un servidor GNU/Linux amb serveis de:
  - Xarxa: DNS (Domain Name System), DHCP (Dynamic Host Configuration Protocol) i NTP (Network Time Protocol).
  - Directori: OpenLDAP (Open LightWeight Directory Access Protocol).
  - Autenticació: PAM (Pluggable Authentication Modules).

- Arxius: NFS (Network File System).
- Distribució de Programari: Puppet.
- Creació de diverses estacions de treball GNU/Linux amb:
  - Login remot: ssh, vinagre.
  - Client de programari: puppet.

### **3.3 DONE Enfocament i mètode seguit**

A l'hora de crear la infraestructura, s'empraran màquines virtuals que emularan dita infraestructura dins l'empresa. Per a la creació de les màquines s'utilitzarà un programari -scripts- especial ja es va implementar en el **tfg** del passat semestre, i que afavoreix en gran mesura la creació i/o manipulació de màquines virtuals. De fet, quan es crea una màquina virtual, el que es fa realment és clonar-ne una pre-concebuda amb un determinat sistema operatiu, aleshores aconseguir una màquina virtual amb un sistema operatiu determinat resulta extramadament senzill. El llistat d'aquests scripts està disponible en els annexos.

La feina més important és la que es durà a terme en el servidor. Per tant, primer es començarà creant una màquina virtual amb Debian per a després passar a instal·lar els serveis necessaris. En molts casos, per provar que els serveis funcionen, serà necessària la creació d'una màquina client; aquesta màquina client anirà provista d'un sistema operatiu Ubuntu.

### **3.4 DONE Planificació del treball**

### **3.5 TODO Sumari de productes obtinguts**

### **3.6 TODO Descripció dels capítols de la memòria**

## **4 Capítols**

### **4.1 Eines**

Aquest capítol es dedicarà a la descripció de les eines que s'utilitzaran per a l'elaboració del projecte. S'usaran altres eines, com kdenlive per editar vídeo, o vokoscreen per crear-ne, però el seu ús serà més específic.

#### 4.1.1 DONE GNU Emacs

GNU Emacs <sup>1</sup>, com enuncia la pàgina oficial, és un editor de text lliure <sup>2</sup>, extensible, personalitzable, i més. En la comunitat d'usuaris de Unix/Linux se'l califica com un petit sistema operatiu al que li manca un bon editor de textos -en clara referència a Vi-. S'han intentat fer branques d'Emacs per emular Vi en forma nativa, però o bé no són estables o bé els projectes estan aturats. Actualment existeix el paquet Evil que emula Vi dins Emacs. Aquesta és una molt bona solució per als usuaris que tenen memoritzada la musculatura de Vi. Un servidor definiria Emacs com l'aplicació que s'obre quan s'obre l'ordinador, que l'ajuda en totes les tasques, i es tanca quan es tanca l'ordinador. En concret, dins Emacs s'usaran les eines següents:

- **evil**: per a edició en mode vi. Eina bàsica per als usuaris que vénen de Vi.
- **org**: per a edició, que-fers, agenda, projectes, etc. Eina bàsica dins Emacs.
- **babel**: per extreure el codi font dels fitxers org i ficar-lo als corresponents fitxers marcats pel tangle. Característica interessant del *literate programming*.
- **dired**: per a navegació, execució de programes, edició del sistema de fitxers, creació directoris, moviments, etc. Eina molt potent que s'empra constantment.
- **eshell**: per a ús de shells emacs. Emacs també porta shells natives, però eshell aporta comandes atractives. Es poden llançar tantes eshells com es vulgui dins emacs.
- **latex**: per a publicacions en html, pdf, etc.
- **beamer**: per fer presentacions dins emacs.
- **tramp**: per a connexions remotes. Fa servir ssh, amb l'afegit que es té emacs per treballar en remot.
- **undo-tree**: arbre de canvis. El més potent arbre de canvi que s'hagi vist mai. Fa i desfa qualsevol buffer -bé sigui una shell, un fitxer, etc.- en el temps, amb totes les branques possibles de canvis que s'hagin pogut fer en cada buffer.

---

<sup>1</sup>GNU Emacs té llicència GNU General Public License versió 2.

<sup>2</sup>Software lliure no és el mateix que software *open source*. Vegeu *gnu.org*.

- **magit**: embolcall git per a emacs. El més potent embolcall de git. Es poden executar fins al 90% de les comandes natives de git.
- **google-it**: implementació d'una funció per googlejar tires de text dins emacs.

Dins emacs s'empraran altres eines, però s'enumeren aquestes per la seva rellevància en l'elaboració del projecte.

#### 4.1.2 DONE Vi

Vi <sup>3</sup>, com diu la Wikipèdia, és un editor de textos lliure que es troba en gairebé tots els sistemes Unix. Per això és important conèixer Vi, perquè sempre estarà disponible en qualsevol distribució Unix/Linux; apart de ser, probablement, el millor editor de textos que existeixi.

Vi pràcticament no s'usarà, atès que s'usarà l'eina tramp, abans mencionada, per a connexions remotes, i així poder usar emacs dins sistemes remots. Així i tot, no es descarta el seu ús específic en moments concrets, o quan no sigui possible una connexió a una ip determinada.

#### 4.1.3 DONE VirtualBox

Segons la Wikipèdia, VirtualBox és un hipervisor per a ordinadors x86/AMD64 d'Oracle Corporation... amb el qual és possible instal·lar sistemes operatius addicionals, coneguts com a "sistemes convidats", dins d'un altre sistema operatiu "amfitrió".

Segons VirtualBox <sup>4</sup>, és l'única solució professional que està lliurement disponible com a programari de codi obert.

#### 4.1.4 DONE Scripts

S'empraran una sèrie d'scripts per afavorir la creació de màquines virtuals amb sistemes operatius predeterminats, per a connexions amb ssh -en cas que no es faci mitjançant tramp-, per a transferència de fitxers de configuració, etc.

En concret, els scripts que s'utilitzaran són els següents:

- **clonevm**: crea una màquina virtual amb un sistema operatiu predeterminat. La sintaxi és `[\s|\./]clonevm [nom_mv_a_clonar] [nom_nova_mv]`.

---

<sup>3</sup>Vi té llicència Berkeley Software Distribution.

<sup>4</sup>VirtualBox es troba sota una llicència privativa d'Oracle, o bé sota una llicència lliure GNU GPLv2 de VirtualBox OSE.

- **closevm**: apaga la màquina virtual. La sintaxi és `[\s|\./]closevm [nom_mv]`.
- **connectvm**: es crea una connexió via ssh a una màquina virtual. La sintaxi és `[\s|\./]connectvm [nom_mv]`.
- **network**: script necessari per realitzar el binding entre el nom de la màquina virtual i la corresponent ip. La sintaxi és `[python\s|\./]network [nom_mv]`.
- **registervm**: registra una màquina virtual a l'hipervisor de Virtual-Box. La sintaxi és `[\s|\./]registervm [nom_mv]`.
- **startvm**: encén una màquina virtual. La sintaxi és `[\s|\./]startvm [nom_mv]`.
- **transfer**: transfereix fitxers de la màquina local a la màquina remota via ssh, gràcies a la utilitat scp. La sintaxi és `[python\s|\./]transfer [configs|installs|modules]connectvm [nom_mv]`.
- **unregistervm**: anul·la la inscripció a l'hipervisor. La sintaxi és `[\s|\./]connectvm [nom_mv]`.

## 4.2 Xarxa

Aquí comença el desenvolupament del projecte. Un cop creada la màquina Debian que farà el rol de servidor, el primer que caldrà fer serà la instal·lació d'uns serveis mínims de xarxa. En concret, s'instal·larà **dns**, **dhcp** i **ntp**.

### 4.2.1 DONE DNS

El *Domain Name Server*, per les seves sigles en anglès, és un sistema que assigna noms a adreces ip. Cal destacar, però, la jerarquia en forma d'arbre amb la que s'ha construït dit servei -el mateix patró s'utilitza al sistema de fitxers de Unix-. Cada nom -domini per ser més acurat- es troba dins una branca de la jerarquia. Així, per a què un domini es pugui trobar cal baixar per l'arbre des de l'arrel fins arribar al domini pròpiament dit -lloc on habita el servidor autoritari per a aquest domini-. Per exemple, el nom de domini **es.wikipedia.org** es correspon amb **es.wikipedia.org**. a l'arbre dns.

Bind <sup>5</sup> és la implementació més usada del protocol estàndard DNS, mantinguda per l'Internet Systems Consortium. Actualment s'utilitza la versió Bind9, que és una reescriptura nova del codi amb millores en temes de seguretat. Les passes que cal fer per oferir el servei de dns són:

1. INSTALAR BIND9
2. CONFIGURAR BIND9
3. LLANÇAR BIND9

Aquestes 3 passes formen un patró típic en les tasques d'instal·lació d'un servei per part de l'administrador de sistemes. Per tant, en els pròxims capítols, quan es parli d'instal·lar un servei, es donarà per entès que se seguiran dites passes.

#### 1- INSTALAR BIND9

L'script d'instal·lació diu així:

```
/tmp/install-bind.sh

#!/usr/bin/env python
import os
command1 = 'apt-get update'
command2 = 'apt-get install -y bind9'
commands = command1+'\n'+command2
os.system(commands)
```

Per llançar l'script d'instal·lació cal obrir un terminal i executar el codi següent:

```
$ chmod +r /tmp/install-bind.py && /tmp/install-bind.py
```

#### 2- CONFIGURAR BIND9

Per configurar el servidor dns cal fer les configuracions següents:

1. Servidor cache.
2. Servidor primari.
3. Zona directa.
4. Zona inversa.

---

<sup>5</sup>Bind es troba sota llicència ISC License. Les versions iguals o posteriors a Bind 9.11.0b es troben baix llicència Mozilla Public License 2.0.



#### 1- Servidor cache:

Servidor que guarda les consultes a altres servidors dns dins un fitxer cau. Aquest fitxer cau s'actualitza de forma periòdica. Per configurar la cache cal editar el fitxer `/etc/bind/named.conf.options/`:

```
# /etc/bind/named.conf.options

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
root@server:/etc/bind
options {
directory "/var/cache/bind";
# google dns servers
forwarders {
    8.8.8.8;
    8.8.4.4;
};
dnssec-enable no;
dnssec-validation no;
auth-nxdomain no;
listen-on-v6 { any; };
listen-on port 53 { 127.0.0.1; 192.168.1.2; };
};
```

En aquest fitxer només cal descomentar l'apartat *forwarders* i inserir el servidor dns del proveïdor d'Internet, o qualsevol altre servidor de dns públic. En aquest cas, s'han inserit els de google.

#### 2- Servidor de noms primari:

El pas següent és configurar bind9 per a què sigui el servidor primari per al domini/zona *domain.ext*. Cal afegir, doncs, les resolucions directa i inversa per a la zona *domain.ext*:

```
# /etc/bind/named.conf.local

zone "domain.ext" {
    type master;
    file "/etc/bind/db.domain.ext";
    #allow-update { key "rndc.key"; };
    notify yes;
}
```

```

zone "1.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/db.1.168.192";
    #allow-update { key "rndc.key"; };
    notify yes;
}

```

En aquest fitxer s'especifiquen les resolucions directa i inversa per al domini *domain.ext*, i els dos fitxers on es troben tots els bindings -directe i invers- per a cada màquina que operi dins aquest domini.

### 3- Zona directa:

El fitxer de resolució de zona directa és allà on se citaran les màquines que hi hagi dins el domini, amb el corresponent binding d'ip:

```

# /etc/bind/db.domain.ext

$TTL      86400
@          IN      SOA      server.domain.ext. admin.domain.ext. (
    20170325      ; Serial Number
    3600          ; Refresh 1h
    3600          ; Retry 1h
    86400         ; Expire 1d
    3600 )        ; Negative Caching TTL 1h
;
@          IN      NS       server.domain.ext.
@          IN      MX       10  server.domain.ext.

server     IN      A        192.168.1.11

```

### 1. Zona inversa:

El fitxer de zona inversa ha de tenir tantes entrades d'apuntadors -PTR- com entrades d'adreces -A- tenguí el fitxer de zona directa:

```

/etc/bind/db.1.168.192

$TTL      86400
@          IN      SOA      server.domain.ext. admin.domain.ext. (
    20170325      ; Serial Number
    3600          ; Refresh 1h

```

```

3600          ; Retry 1h
          86400          ; Expire 1d
3600 )          ; Negative Caching TTL 1h
;
@              IN          NS          server.domain.ext.

11              IN          PTR         server.domain.ext.

```

### 3- LLANÇAR BIND9

Abans de llançar, cal comentar que bind9 proveeix dos programes per avaluar la correcció en la sintaxi en els fitxers *named.conf* i els corresponents fitxers de zona. Així doncs, cal obrir un terminal i executar les ordres següents:

```

$ named-checkconf /etc/bind/named.conf
$ named-checkzone domain.ext /etc/bin/db.domain.exten
$ named-checkzone 1.168.192 /etc/bind/db.1.168.192

```

Un cop comprovat que no hi ha erros, és l'hora d'engegar el dimoni, des d'un terminal o des d'emacs -amb emacs existeixen altres possibilitats d'engegar el dimoni sense una shell-:

```
$ systemctl restart bind9.service
```

Una petita comprovació de què el sistema funciona correctament és fent:

```
$ dig archlinux.com
```

### 4.2.2 DONE DHCP

El Dynamic Host Control Protocol, o DHCP <sup>6</sup>, és un protocol estàndard, la qual funció principal és la de designar de forma dinàmica adrecess IP als ordinadors de la xarxa. Si no opera un servidor de DHCP dins la xarxa, llavors les configuracions de xarxa s'han de realitzar manualment. La implementació del protocol DHCP, mantinguda per l'Internet Systems Consortium, porta el mateix nom que el protocol. ISC ha desenvolupant una nova implementació del protocol, anomenada Kea, que no serà la que s'instal·larà, atès que no es troba en tots els repositoris debian i no és prou estable. El paquet Debian que porta DHCP duu per nom *isc-dhcp-server*, nom que s'usarà a partir d'ara.

---

<sup>6</sup>DHCP es troba sota llicència ISC License. Les versions de Kea es troben baix llicència Mozilla Public License 2.0.

## 1- INSTAL·LACIÓ D'ISC-DHCP-SERVER

Script d'instal·lació d'isc-dhcp-server:

```
/tmp/install-isc-dhcp-server.py
```

```
#!/usr/bin/env python
import os
command = 'apt-get install -y isc-dhcp-server'
os.system(command)
```

Llançament de l'script:

```
$ chmod +r /tmp/install-isc-dhcp-server.py && /tmp/install-isc-dhcp-server.py
```

## 2- CONFIGURACIÓ D'ISC-DHCP-SERVER

Si es té més d'una targeta de xarxa, s'ha d'especificar en quina interfície el servidor ha de gestionar el servei -bàsicament, respostes d'ips-. En qualsevol cas, sempre és recomanable especificar la interfíciei:

```
/etc/default/isc-dhcp-server
```

```
# Defaults for isc-dhcp-server initscript
# sourced by /etc/init.d/isc-dhcp-server
# installed at /etc/default/isc-dhcp-server by the maintainer scripts

#
# This is a POSIX shell fragment
#

# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
#DHCPD_CONF=/etc/dhcp/dhcpd.conf

# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
#DHCPD_PID=/var/run/dhcpd.pid

# Additional options to start dhcpd with.
# Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
# Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACES="eth0"
```

El següent pas és configurar el servidor de dhcp pròpiament dit:

```
/etc/dhcp/dhcpd.conf
```

```
# General configuration:
default-lease-time 3600;
max-lease-time 86400;
option domain-name "domain.com";
option domain-name-servers server.domain.com;
authoritative;
log-facility local7;
# specify network address and subnet-mask
subnet 192.168.1.0 netmask 255.255.255.0 {
    # specify the range of leased IP addresses
    range 192.168.1.100 192.168.1.150;
    # specify subnet-mask
    option subnet-mask 255.255.255.0;
    # specify broadcast address
    option broadcast-address 192.168.1.255;
    # specify default gateway
    option-routers gateway.domain.com;
}
# dns updates:
server-identifier server;
ddns-updates on;
ddns-update-style interim;
ddns-domainname "domain.com.";
ddns-rev-domainname "in-addr.arpa.";
deny client-updates;
include "/etc/bind/rndc.key";
zone domain.com. {
    primary 127.0.0.1;
    key rndc-key;
}
zone 1.168.192. in-addr.arpa. {
    primary 127.0.0.1;
    key rndc-key;
}
```

### 3- LLANÇAR ISC-DHCP-SERVER

Per llançar el dimoni, s'obre una consola i s'insereix el codi:

```
$ systemctl restart isc-dhcp-server
```

### 4.2.3 TODO NTP

NTP <sup>7</sup>, acrònim de *Network Time Protocol*, és un protocol que sincronitza els ordinadors d'una xarxa amb l'UTC, *Coordinated Universal Time*, amb un marge d'error d'alguns mil·lisegons. NTP usa un sistema jeràrquic de servidors, En el nivell més alt es troben els Stratum0, equipats amb unitats de temps atòmiques, etc. Aquests donen servei als Stratum1, i així successivament. NTP fou dissenyat per un enginyer informàtic l'any 1983, i actualment és mantingut com un projecte open source.

#### 1- INSTAL·LACIÓ D'NTP

Script d'instal·lació:

```
/tmp/install-ntp.py
```

```
#!/usr/bin/env python
import os
command = 'apt-get install -y ntp'
os.system(command)
```

Obrir un terminal i teclejar:

```
$ chmod +r /tmp/install-ntp.py && /tmp/install-ntp.py
```

#### 2- COFIGURACIÓ D'NTP

El més important és la configuració dels servidors d'ntp; en aquest cas s'han triat els que té disponibles Debian:

```
/etc/ntp.conf
```

```
# /etc/ntp.conf, configuration for ntpd; see ntp.conf(5) for help
driftfile /var/lib/ntp/ntp.drift
# Enable this if you want statistics to be logged.
#statsdir /var/log/ntpstats/
statistics loopstats peerstats clockstats
filegen loopstats file loopstats type day enable
filegen peerstats file peerstats type day enable
filegen clockstats file clockstats type day enable
# You do need to talk to an NTP server or two (or three).
#server ntp.your-provider.example
```

---

<sup>7</sup>NTP té llicència NTP License.

```

# pool.ntp.org maps to about 1000 low-stratum NTP servers. Your server will
# pick a different set every time it starts up. Please consider joining the
# pool: <http://www.pool.ntp.org/join.html>
#server 0.debian.pool.ntp.org iburst
#server 1.debian.pool.ntp.org iburst
#server 2.debian.pool.ntp.org iburst
#server 3.debian.pool.ntp.org iburst
# Access control configuration; see /usr/share/doc/ntp-doc/html/acconf.html for
# details. The web page <http://support.ntp.org/bin/view/Support/AccessRestrictions>
# might also be helpful.
#
# Note that "restrict" applies to both servers and clients, so a configuration
# that might be intended to block requests from certain clients could also end
# up blocking replies from your own upstream servers.
# By default, exchange time with everybody, but don't allow configuration.
restrict -4 default kod notrap nomodify nopeer noquery
#restrict -6 default kod notrap nomodify nopeer noquery
# Local users may interrogate the ntp server more closely.
restrict 127.0.0.1
restrict ::1
# Clients from this (example!) subnet have unlimited access, but only if
# cryptographically authenticated.
#restrict 192.168.123.0 mask 255.255.255.0 notrust
# If you want to provide time to your local subnet, change the next line.
# (Again, the address is an example only.)
#broadcast 192.168.123.255
# If you want to listen to time broadcasts on your local subnet, de-comment the
# next lines. Please do this only if you trust everybody on the network!
#disable auth
#broadcastclient

# CUSTOMIZATION
# NIST SERVERS
server time-a.nist.gov iburst
server time-b.nist.gov iburst
server time-c.nist.gov iburst
server time-d.nist.gov iburst

# By default, exchange time with everybody, but don't allow configuration.

```

```

restrict -6 default kod notrap nomodify nopeer noquery

# By default don't answer anything
restrict default ignore

# Allow hosts
restrict 192.168.1.0 mask 255.255.255.0 limited kod modify notrap nopeer noquery

# These lines allow this NTP server to query level 1 servers for time
#server time-a.nist.gov
server 129.6.15.28
restrict 129.6.15.28 mask 255.255.255.255 nomodify notrap nopeer noquery
#server time-b.nist.gov
server 129.6.15.29
restrict 129.6.15.29 mask 255.255.255.255 nomodify notrap nopeer noquery
#server time-c.nist.gov
server 129.6.15.30
restrict 129.6.15.30 mask 255.255.255.255 nomodify notrap nopeer noquery
#server time-d.nist.gov
server 129.6.15.27
restrict 129.6.15.27 mask 255.255.255.255 nomodify notrap nopeer noquery

```

Si només es vol sincronitzar amb servidors IPv4:

```
/etc/default/ntp
```

```

# Accept only ipv4
NTPD_OPTS='-4 -g'

```

### 3- LLANÇAR NTP

Com sempre, obrir un terminal i inserir la instrucció:

```
$ systemctl restart ntp.service
```

Amb això s'acabaria el capítol relacionat amb la instal·lació dels serveis de xarxa mínims que requereix un entorn linux en una empresa.



- 4.3 TODO Autenticació
- 4.4 TODO Sistema de fitxers
- 4.5 TODO Programari
- 5 Conclusions
- 6 Glossari
- 7 Bibliografia
- 8 Annexos