



# ASORC

Administración de sistemas operativos  
en redes de computadores

Familia de sistemas LINUX

Basado en CentOS 7

# INDICE

Indice:

# Particionado del disco

Objetivo: Crear/añadir particiones en el disco duro.

Instalación: yum -y install e2fsprogs

Caso práctico:

- fdisk /dev/sdb
  - m: menu
  - n: nueva partición ( p: partición primaria )
  - t: type file system ( L: opciones disponibles )
  - p: previsualizar la nueva tabla
  - w: escribir la tabla de particiones en disco
- mount /dev/sdb1 /disco2

Comprobación: fdisk -l

# Configuración de la red

Objetivo: Asignar una dirección IP estática/dinámica al sistema

Fichero: /etc/sysconfig/network-scripts/enp0s3

Caso práctico:

- IP estática:           BOOTPROTO=static  
                                  IPADDR=192.168.2.2  
                                  NETMASK=255.255.255.0  
                                  GATEWAY=192.168.2.1  
                                  DNS1=8.8.8.8
- IP dinámica:        BOOTPROTO=dhcp
- Recomendación: NM\_CONTROLLED=no

Comprobación: ifconfig

# Gestión de repositorios

Objetivo: Establecer los repositorios del sistema para la instalación de software

Instalación: yum -y epel-release

Fichero: /etc/yum.repos.d/epel.repo

- enabled = 1

Caso práctico:

Actualización: yum repolist

Otros repositorios: nux-dextop, rpmforge...

# Inicio y parada de servicios

Objetivo: Poner en marcha o detener los servicios configurados en el sistema. Generalmente se aplica al 'demonio' correspondiente: servicio acabado en 'd'.

Caso práctico: servicio 'sshd'

- Inicio: `systemctl start sshd`
- Parada: `systemctl stop sshd`
- Activo al inicio: `systemctl enable sshd`
- Estado actual: `systemctl status sshd`

Comprobación: `netstat -lp`

# Servidor SSH

Objetivo: Acceso remoto al sistema de forma segura.

Instalación: Por defecto.

Puerto: 22 ( recomendable modificarlo )

Fichero: /etc/ssh/sshd\_config

- Cambio del puerto del servicio: Port 1234
- Protocolo actualizado: Protocol 2
- Usuarios autorizados: AllowUsers marc
- Acceso ROOT: PermitRootLogin no

Caso práctico:

- systemctl restart sshd

Comprobación: ssh -p 1234 marc@192.168.2.2

# Servidor VNC

Objetivo: Acceso remoto al sistema mediante un entorno gráfico. Acceso NO seguro.

Instalación: `yum -y install tigervnc-server`

Puerto: 5901

Caso práctico:

- `cp /lib/systemd/system/vncserver@.service \`  
`/etc/systemd/system/vncserver@:1.service`
- `/etc/systemd/system/vncserver@\:1.service`
  - Modificar <USER>: marc
- `systemctl start vncserver@:1.service`

Comprobación: `vncview marc@192.168.2.2:5901`



# Servidor RDP

Objetivo: Acceso remoto al sistema mediante un entorno gráfico. Acceso NO seguro.

Instalación: yum install xrdp

Puerto: 3389

Caso práctico:

- systemctl start xrdp

Comprobación: xfreerdp marc@192.168.2.2:3389

# Servidor NX

Objetivo: Acceso remoto al sistema mediante un entorno gráfico. Acceso seguro mediante ssh.

Instalación: yum -y install x2goserver

Puerto: el mismo de ssh

Caso práctico:

- X2godbadmin --createdb
- x2gocleansession

Comprobación: x2goclient

# Servidor DNS

Objetivo: Identificar una máquina conectada a la red mediante un nombre de dominio.

Instalación: yum install bind

Puerto: 53

Fichero: /etc/named.conf

Caso práctico:

- Nombre de la red: network.com
- Dirección de red: 192.168.2.0/24
- Nodos:
  - servidor: 192.168.2.100
  - nodoA: 192.168.2.101

Comprobación: nslookup nodoA

# Fichero de configuración

/etc/named.conf

```
options {
```

```
    listen-on port 53 { 127.0.0.1; 192.168.2.0/24 };
```

```
    directory "/var/named";
```

```
    forwarders { 193.145.233.5; 8.8.8.8; };
```

```
};
```

```
zone "network.net" IN {
```

```
    type master;
```

```
    file "network.zone"; };
```

```
zone "2.168.192.in-addr.arpa" IN {
```

```
    type master;
```

```
    file "reverse.zone"; };
```

# Fichero de configuración

## /var/named/network.zone

```
$TTL 86400
@ IN SOA network.net root.network.net.
(150115 28800 7200 604800 86400)
    IN NS      servidor.network.net.
    IN MX     10 servidor.network.net.
servidor.network.net. IN A 192.168.2.100
nodoA.network.net.  IN A 192.168.2.101
```

## /var/named/reverse.zone

```
$TTL 86400
@ IN SOA network.net. root.network.net.
(150115 28800 7200 604800 86400)
    IN NS      servidor.network.net.
100.2.168.192.in-addr.arpa. IN PTR servidor.network.net.
101.2.168.192.in-addr.arpa. IN PTR nodoA.network.net.
```

# Servidor DHCP

Objetivo: Permitir que un equipo conectado a una red pueda obtener su configuración de red de forma dinámica.

Instalación: yum install dhcp

Puerto: 67-68 UDP

Fichero: /etc/dhcp/dhcpd.conf

Caso práctico:

- Nombre de la red: network.com
- Dirección de red: 192.168.2.0/24
- Nodos:
  - servidor dhcp: 192.168.2.100
  - servidor dns: 192.168.2.100
  - gateway: 192.168.2.1
  - nodoA: 192.168.2.101

# Fichero de configuración

/etc/dhcp/dhcpd.conf

```
shared-network network.net {  
    subnet 192.168.2.0 netmask 255.255.255.0 {  
        option routers 192.168.2.1;  
        option subnet-mask 255.255.255.0;  
        option broadcast-address 192.168.2.255;  
        option domain-name-servers 192.168.2.100;  
        range 192.168.2.201 192.168.2.209;  
    }  
}  
  
host learn {  
    option host-name "nodoA.network.net";  
    hardware ethernet 00:25:d3:66:63:b3;  
    fixed-address 192.168.2.101;  
}
```

# Servidor NFS

Objetivo: Permitir el acceso remoto a un sistema de archivos a través de la red.

Instalación: `yum -y install nfs-utils`

Puerto: 2049

Fichero: `/etc/exports`

`/directorio_a_compartir 192.168.2.0/24(rw,no_root_squash)`

Caso práctico:

- `systemctl start nfs-server`

Comprobación:

- `showmount -e 192.168.2.2`
- `mount -t nfs 192.168.2.2:/directorio_a_compartir /mi_directorio_local`



# Servidor SAMBA

Objetivo: Similar a NFS. Permite el acceso remoto a un sistema de archivos cuando se involucran sistemas Windows.

Instalación: `yum -y install samba samba-client samba-common`

Puerto: 137-139

Fichero: `/etc/samba/smb.conf`

Caso práctico:

- usuario del servicio: `smbpasswd -a marc`
- `systemctl start nmb`
- `systemctl start smb`

Comprobación:

- `smbclient //192.168.2.2:/samba -U marc`
- `mount -t cifs -o username=marc //192.168.2.2/samba /mi_dir_local`

# Fichero de configuración

/etc/samba/smb.conf:

workgroup = administracion

netbios name = admin

server string = Mi servidor SAMBA

hosts allow = 192.168.2.

[samba]

comment = Directorio compartido

path = /samba

# Ficheros ocultos

hide dot file = Yes

# Papelera de reciclaje

vfs objects = recycle

recycle:repository = Recycle Bin

# Servidor FTP

Objetivo: Permitir la transferencia de archivos entre el cliente y el servidor.

Instalación: `yum -y install vsftpd`

Puerto: 20-21

Fichero: `/etc/vsftpd/vsftpd.conf`

Caso práctico:

- `touch /etc/vsftpd/chroot_list`
- `systemctl start vsftpd`

Comprobación: filezilla

# Fichero de configuración

/etc/vsftpd/vsftpd.conf:

# Acceso usuario anonimo

anonymous\_enable=NO

# Acceso usuarios local

local\_enable=YES

# SSL/TLS

ssl\_enable=NO

# Compatibilidad con filezilla

ssl\_ciphers=HIGH

require\_ssl\_reuse=NO

# Servidor SENDMAIL

Objetivo: Transferir correo de forma segura entre hosts usando el protocolo SMTP.

Instalación: `yum -y install sendmail sendmail-cf m4 cyrus-sasl cyrus-sasl-plain`

Puerto: 25

Fichero: `/etc/mail/sendmail.mc`

Configuración previa:

- `alternatives --config mta`
- `systemctl stop postfix`

Caso práctico:

- `newaliases`
- `systemctl start saslauthd`
- `systemctl start sendmail`

Comprobación: `echo `date` | mail to user@domain`

# Servidor SENDMAIL

## Certificados SSL/TSL:

```
openssl req -sha256 -new -x509 -nodes -newkey rsa:4096 -days 1825 -out  
/etc/pki/tls/certs/sendmail.pem -keyout /etc/pki/tls/certs/sendmail.pem
```

```
openssl x509 -subject -fingerprint -noout -in /etc/pki/tls/certs/sendmail.pem
```

## /etc/sysconfig/saslauthd

- FLAGS=-r

## /etc/mail/local-host-names

- domain.com

## /etc/mail/access

- Connect:192.168.2.0/24 RELAY

## /etc/aliases

- root: marc

# Servidor SENDMAIL

## /etc/mail/sendmail.mc

```
define(`confAUTH_OPTIONS', `A p')dnl
```

```
TRUST_AUTH_MECH(`EXTERNAL LOGIN PLAIN')dnl
```

```
define(`confAUTH_MECHANISMS', `EXTERNAL LOGIN PLAIN')dnl
```

```
define(`confCACERT_PATH', `/etc/pki/tls/certs')dnl
```

```
define(`confCACERT', `/etc/pki/tls/certs/ca-bundle.crt')dnl
```

```
define(`confSERVER_CERT', `/etc/pki/tls/certs/sendmail.pem')dnl
```

```
define(`confSERVER_KEY', `/etc/pki/tls/certs/sendmail.pem')dnl
```

```
DAEMON_OPTIONS(`Port=smtp, Name=MTA')dnl
```

```
LOCAL_DOMAIN(`localhost.localdomain')dnl
```

```
MASQUERADE_AS(`asorc.net')dnl
```

# Servidor CUPS

Objetivo: Permitir que el sistema actúe como servidor de impresión. Acepta tareas desde los clientes, las procesa y las envía al medio de impresión apropiado.

Instalación: `yum -y install cups cups-pdf`

Puerto: 631

Fichero: `/etc/cups/cupsd.conf`

Caso práctico:

- Impresión en fichero \*.pdf
- `systemctl start cups`

Administración web: `http://localhost:631`



# Fichero de configuración

## /etc/cups/cupsd.conf:

Listen localhost:631 Port 631

Browsing On

BrowseOrder allow,deny

BrowseAllow all

BrowseRemoteProtocols CUPS

BrowseAddress @LOCAL

BrowseLocalProtocols CUPS dnssd

<Location />

Order allow,deny

Allow all

</Location>

## /etc/cups/cups-pdf.conf

Out \${HOME}

# Servidor LDAP

Objetivo: Permite el acceso a un servicio de directorio ordenado y distribuido para buscar información en un entorno de red. Se puede considerar una base de datos.

Instalación: `yum -y install openldap-clients openldap-servers authconfig authconfig-gtk migrationtools`

Puerto: 389

Fichero: `/etc/openldap/slapd.conf`

Comprobación:

- `systemctl start slapd`
- `ldapsearch -x -b 'uid=marc,ou=People,dc=net,dc=dominio'`

# Fichero de configuración

## Creación de la autoridad certificadora:

```
cd /etc/openssl/cacerts
```

```
echo "01" > ca.srl
```

```
openssl genrsa -aes128 2048 > cacert.key
```

```
openssl req -utf8 -new -key cacert.key -out cacert.csr
```

```
openssl x509 -req -in cacert.csr -out cacert.pem -signkey cacert.key -days 3650
```

## Certificado y firma digital para el servidor:

```
openssl genrsa -aes128 2048 > key.pem
```

```
openssl req -utf8 -new -key key.pem -out slapd.csr
```

```
openssl x509 -req -in slapd.csr -out cert.pem -CA cacert.pem -CAkey  
cacert.key -days 3650 -CAcreateserial -CAserial ca.seq
```

```
openssl rsa -in key.pem -out key.pem
```

# Fichero de configuración

/etc/sysconfig/ldap

SLAPD\_LDAPS=yes

Configuración:

cp /usr/share/openldap-servers/DB\_CONFIG.example  
/var/lib/ldap/autenticar/DB\_CONFIG

slappasswd (copiar salida)

/etc/openldap/slapd.conf

rootpw (copiar la salida de slappasswd)

# Fichero de configuración

## Configuración de la seguridad:

```
cacertdir_rehash /etc/openldap/cacerts
```

```
chown -R root:ldap /etc/openldap/cacerts
```

```
chmod -R 750 /etc/openldap/cacerts
```

```
chown -R ldap:ldap /var/lib/ldap/autenticar
```

```
chmod 700 /var/lib/ldap/autenticar
```

```
chown ldap:ldap /etc/openldap/slapd.conf
```

```
chmod 600 /etc/openldap/slapd.conf
```

```
rm -rf /etc/openldap/slapd.d/*
```

# Fichero de configuración

## Insertar datos en el directorio:

Crea archivos standard

```
echo "" | slapadd -f /etc/openldap/slapd.conf
```

Crear subconjunto de archivos Idif

```
slaptest -f /etc/openldap/slapd.conf -F /etc/openldap/slapd.d/
```

Configuración para la migración de cuentas

/usr/share/migrationtools/migrate\_common.ph

```
$DEFAULT_MAIL_DOMAIN = "domain.net";
```

```
$DEFAULT_BASE = "dc=domain,dc=net"
```

# Fichero de configuración

## Insertar datos en el directorio:

Crea el objeto base

```
/usr/share/migrationtools/migrate_base.pl > base.ldif
```

Importar usuarios y grupos

```
/usr/share/migrationtools/migrate_group.pl /etc/group > group.ldif
```

```
/usr/share/migrationtools/migrate_passwd.pl /etc/passwd > passwd.ldif
```

Insertar todo en LDAP

```
ldapadd -x -W -D 'cn=Manager,dc=domino,dc=net' -h 127.0.0.1 -f base.ldif
```

```
ldapadd -x -W -D 'cn=Manager,dc=dominio,dc=net' -h 127.0.0.1 -f group.ldif
```

```
ldapadd -x -W -D 'cn=Manager,dc=dominio,dc=net' -h 127.0.0.1 -f passwd.ldif
```

# Servidor MYSQL

Objetivo: Gestionar un sistema de bases de datos.

Instalación: `yum -y install mariadb mariadb-server`

Puerto: 3306

Caso práctico:

- `systemctl start mariadb`
- `mysql_secure_installation`
- `mysql -u root -p`
  - create database music
  - grant all on music.\* to 'marc'@'%' identified by 'passwd'

Comprobación:

- workbenck
- `mysql -u marc -p music -h 192.168.2.2`



# Servidor HTTP

Objetivo: Servir contenido web

Instalación: `yum -y install httpd`

Puerto: 80

Fichero: `/etc/httpd/conf.d/*.conf`

Caso práctico:

- `systemctl start httpd`
- Incluir nombre de dominio en el DNS (opcional)

Comprobación:

- `http://192.168.2.2`
- `http://www.embutidosgutierrez.com`

# Fichero de configuración

/etc/httpd/conf/httpd.conf

ServerName www.myserver.name:80

/etc/httpd/conf.d/embutidosgutierrez.conf

<VirtualHost \*:80>

DocumentRoot /var/www/html/embutidosgutierrez

ServerName www.embutidosgutierrez.net

</VirtualHost>

/var/www/html/embutidosgutierrez/index.html

<html>

<head></head>

<body> Web de Embutidos Gutierrez </body>

</html>

# Fichero de configuración

## /etc/named.conf

```
zone "embutidosgutierrez.com" in {  
    type master;  
    file "embutidos.zone";  
}
```

## /var/named/embutidos.zone

```
$TTL 86400  
@ IN SOA     network.net root.network.net.  
(150115 28800 7200 604800 86400)  
IN NS                               servidor.network.net.  
IN MX 10                               servidor.network.net.  
www.embutidosgutierrez.com. IN A 192.168.2.100
```

# Servidor VPN

Objetivo: Crear una conexión segura entre dos red a través de Internet. Todo el tráfico que viaja está asegurado y protegido.

Instalación: `yum -y install openvpn easy-rsa openssl`

Puerto: 1194

Fichero:

- Servidor: `/etc/openvpn/servidor.conf`
- Cliente: `/etc/openvpn/cliente.conf`

Caso práctico:

- Creación de la red VPN 192.168.37.0
- `systemctl --config servidor.conf`

Comprobación:

- `ifconfig`
- `ping 192.168.37.1`

# Servidor VPN

## Configuración previa:

- `cp -r /usr/share/easy-rsa/2.0/* /etc/openvpn`

## Creación de certificados:

- `mkdir /etc/openvpn/keys/`
- `/usr/share/easy-rsa/2.0/build-ca`
- `/usr/share/easy-rsa/2.0/build-dh`
- `/usr/share/easy-rsa/2.0/build-key-server servidor`
- `/usr/share/easy-rsa/2.0/build-key cliente`

# Servidor VPN

/etc/openvpn/servidor.conf

port 1194

proto udp

dev tun

### Sección de firma y certificados

ca /etc/openvpn/keys/ca.crt

cert /etc/openvpn/keys/servidor.crt

key /etc/openvpn/keys/servidor.key

dh keys/dh2048.pem

###

server 192.168.37.0 255.255.255.0

ifconfig-pool-persist ipp.txt

keepalive 10 120

comp-lzo

persist-key

persist-tun

status status-openvpn.log

verb 3

# Servidor VPN

/etc/openvpn/cliente.conf

client

dev tun

proto udp

remote 192.168.2.2 1194

float

resolv-retry infinite

nobind

persist-key

persist-tun

### Sección de firma y certificados

ca /etc/openvpn/keys/ca.crt

cert /etc/openvpn/keys/cliente.crt

key /etc/openvpn/keys/cliente.key

ns-cert-type server

###

comp-lzo

verb 3

# Servidor JABBER

Objetivo: (XMPP) Protocolo extensible de mensajería y comunicación de presencia basado en XML, originalmente ideado para mensajería instantánea.

Instalación:

- Necesario Java (JRE), Mysql
- Openfire:

<http://www.igniterealtime.org/downloads/index.jsp#openfire>

- `systemctl start openfire`

Configuración web: <http://192.168.2.2:9090>

Comprobación: pidgin



# Fichero de configuración

/etc/sysconfig/openfire

- OPENFIRE\_OPTS="-Xmx1024m"
- JAVA\_HOME=/usr/java/latest

mysql -u root -p

create database openfire;

create user openfire identified by 'passwd';

grant all on openfire.\* to 'openfire'@'%';

# Servidor ZIMBRA

Objetivo: Programa informático colaborativo con un cliente/servidor de correo, calendario, etc...

Instalación:

[http://files2.zimbra.com/downloads/8.5.0\\_GA/zcs-8.5.0\\_GA\\_3042.RHEL6\\_64.20140828192005.tgz](http://files2.zimbra.com/downloads/8.5.0_GA/zcs-8.5.0_GA_3042.RHEL6_64.20140828192005.tgz)

Caso práctico:

- tar xfv zcs-8.5.0\_GA\_3042.RHEL6\_64.20140828192005.tgz
- cd zcs-8.5.0\_GA\_3042.RHEL6\_64.20140828192005
- ./install.sh
- service start zimbra

Comprobación: <https://192.168.2.2:7071>

# Servidor NAGIOS

Objetivo: Monitor de red que vigila equipos (hardware) y servicios (software) definidos, alertando cuando su comportamiento no es el deseado.

Instalación: `yum -y install nagios nagios-plugins-all`

Fichero: `/etc/httpd/conf.d/nagios.conf`

Caso práctico:

- `htpasswd /etc/nagios/passwd nagiosadmin`
- `systemctl start nagios`

Comprobación: `http://192.168.2.2/nagios`

# Fichero de configuración

/etc/httpd/conf.d/nagios

```
<IfModule !mod_authz_core.c>
```

```
    # Order allow,deny
```

```
    # Allow from all
```

```
    Order deny,allow
```

```
    Deny from all
```

```
    Allow from 127.0.0.1 192.168.2.0/24
```

# Servidor SQUID

Objetivo: Mejorar el rendimiento de las conexiones web guardando en caché peticiones recurrentes, acelerar el acceso al servidor web y añadir seguridad filtrando tráfico.

Instalación: `yum -y install squid`

Puerto: 3128

Fichero: `/etc/squid/squid.conf`

Caso práctico:

- redirección del tráfico (script: `/etc/squid/redirect.sh`)
- `systemctl start squid`

Comprobación: `http://www.elpais.es`

# Fichero de configuración

/etc/squid/squid.conf

- acl network src 192.168.2.0/24
- acl web\_deny url\_regex “/etc/squid/web\_deny.acl”
- http\_access allow list\_deny !web\_deny
- http\_port 3128

/etc/squid/web-deny.acl

- www.elpais.es

# Fichero de configuración

Redirección del tráfico hacia el proxy-cache ( scripting ):

- Interfaz enp0s3: WAN
- Interfaz enp0s8: LAN

/etc/squid/redirect.sh

# Permite el paso de una red a la otra

```
iptables -A FORWARD -i enp0s8 -o enp0s3 -j ACCEPT
```

# Envío del tráfico entrante por enp0s8 hacia el proxy

```
iptables -t nat -A PREROUTING -i enp0s8 -p tcp -m tcp --dport 80 -j  
DNAT --to-destination 192.168.2.2:3128
```

# Envío del tráfico saliente a la red externa

```
iptables -t nat -A POSTROUTING -o enp0s3 -j MASQUERADE
```

# BIT DE FORWARDING

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

# Servidor LTSP

Objetivo: Proporcionar la capacidad de ejecutar Linux en computadores de pocas prestaciones. El sistema consiste en distribuir a los clientes, por medio de la red, el núcleo Linux que se está ejecutando en el servidor.

Previo:

Obtener un thin-client: <http://distrowatch.com/>

Volcar la distro en /opt/ltsp/amd64

Instalación: `yum -y install nfs-utils dhcpd tftp-server syslinux`



# Fichero de configuración

## Configuración:

```
cp -r /usr/share/syslinux/* /var/lib/tftpboot/
```

```
mkdir /var/lib/tftpboot/pxelinux.cfg;
```

## /etc/exports

```
/opt/ltsp/amd64 *(ro,async,no_root_squash)
```

## /etc/dhcpd/dhcpd.conf

```
class "pxeclients" {  
    match if substring(option vendor-class-identifier, 0, 9) = "PXEClient";  
    next-server 192.168.2.2;  
    filename "pxelinux.0";  
    option root-path "192.168.2.2:/opt/ltsp/amd64";  
}
```

# Fichero de configuración

/etc/xinetd.d/tftp

service tftp

{

    socket\_type = dgram

    protocol = udp

    wait = yes

    user = root

    server = /usr/sbin/in.tftpd

    server\_args = -s /var/lib/tftpboot

    disable = no

    per\_source = 11

    cps = 100 2

    flags = IPv4

}

# Fichero de configuración

## Caso práctico:

`systemctl restart nfs-server`

`systemctl restart dhcpd`

`systemctl restart xinetd`

## Comprobación:

Arrancar el cliente con la opción “Arranque por red”.

# Servidor PXE

Objetivo: Crear un entorno para arrancar e instalar el sistema operativo en computadoras a través de una red.

Instalación: `yum -y install dhcpd tftp-server vsftpd syslinux`

Configuración previa:

- `mount -t iso9660 -o loop CentOS-7-x86_64-Minimal.iso /var/ftp`
- `cp /var/ftp/images/pxeboot/vmlinuz /var/lib/tftpboot/centos/`
- `cp /var/ftp/images/pxeboot/initrd.img /var/lib/tftpboot/centos/`
- `cp -r /usr/share/syslinux/* /var/lib/tftpboot/`
- `mkdir /var/lib/tftpboot/pxelinux.cfg`

# Fichero de configuración

/var/lib/tftpboot/pxelinux.cfg/default

default menu.c32

prompt 0

timeout 300

ONTIMEOUT local

menu title ##### PXE Boot Menu #####

label 1

menu label ^1) Install Centos 7-Minimal 64-bit

kernel centos/vmlinuz

append initrd=centos/initrd.img method=ftp://192.168.2.2/centos devfs=nomount

# Fichero de configuración

## Servicio FTP /etc/vsftpd/vsftpd.conf

anonymous\_enable=YES

no\_anon\_password=YES

anon\_root=/var/ftp/

anon\_upload\_enable=NO

anon\_mkdir\_write\_enable=NO

## Servicio dhcpd:

- Igual que para LTSP

## Servicio tftp:

- Igual que para LTSP

# Fichero de configuración

## Caso práctico:

- `systemctl restart vsftpd`
- `systemctl restart dhcpd`
- `systemctl restart xinetd`

## Comprobación:

Arrancar el cliente con la opción “Arranque por red”.

# Servidor DRBL

Objetivo: Permite tener un S.O. corriendo en varias máquinas sin necesidad de que tengan un disco duro conectado. También permite clonar o restaurar varios equipos a la vez mediante paquetes Multicast.

Instalación: No requiere instalación.

Caso práctico:

- <http://drbl.org/download>

Comprobación:

- Arrancar el servidor.
- Arrancar el cliente con la opción “Arranque por red”.



# RAID

Objetivo: Permite implementar un volumen de almacenamiento de datos formado por varios discos duros con el objetivo de proteger la información y conseguir mayor tolerancia a fallos si el disco duro sufre una avería.

Instalación: `yum -y install mdadm`

Caso práctico:

Creación del raid con cuatro discos duros:

```
mdadm --create /dev/md1 --level=raid10 --raid-device=4 /dev/sdb /dev/sdc /dev/sdd /dev/sde
```

Configuración:

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

Simular fallo de disco: `mdamnd -f /dev/md1 /dev/sdb`

Extraer disco del RAID: `mdadm -r /dev/md1 /dev/sdb`

Añadir disco al RAID: `mdadm -a /dev/md1 /dev/sdb`

Comprobación: `mdadm --detail /dev/md1`

# RAID

Objetivo: Creación de un volumen lógico. Configurar el RAID para usarlo como un directorio del sistema de ficheros.

```
pvcreate /dev/md1
```

```
vgcreate VGDatos /dev/md1
```

```
lvcreate -l 90%FREE VGDatos -n LVDatos
```

```
mkfs.ext4 /dev/mapper/VGDatos-LVDatos
```

```
mkdir -p /datos
```

```
mount /dev/mapper/VGDatos /datos
```

# IPTABLES

Objetivo: Es un firewall integrado en el kernel. Permite interceptar y manipular paquetes que circulan por la red.

Instalación: `yum -y install iptables`

Caso práctico:

- Entrada por interfaz `enp3s0` (LAN 192.168.2.0/24)
- Salida por interfaz `enp4s0` (WAN internet)

Comprobación:

- `iptables -L -n --lines-numbers`
- `iptables -L -n --lines-numbers -t nat`

# Fichero de configuración

/root/iptables.sh ( scripting )

# Eliminar reglas anteriores

iptables -F; iptables -X

iptables -Z; iptables -t nat -F

# Establecer politicas por defecto

iptables -P INPUT ACCEPT

iptables -P OUTPUT ACCEPT

iptables -P FORWARD ACCEPT

# Paso de paquetes entre interfaces

iptables -A FORWARD -i enp3s0 -o enp4s0 -s 192.168.2.0/24 -m conntrack --ctstate NEW -j ACCEPT

iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

iptables -t nat -F POSTROUTING

iptables -t nat -A POSTROUTING -o enp4s0 -j MASQUERADE

# Activación del paso de paquetes

echo 1 > /proc/sys/net/ipv4/ip\_forward

# Servidor NIS

Objetivo: Permitir el envío de datos de configuración tales como nombres de usuarios y hosts dentro de una red.

Instalación: `yum -y install ypbind yp-tools ypserv`

Caso práctico:

- `domainname ypdomain.net`
- `systemctl start ypserv`

Comprobación: `rpcinfo -u localhost ypserv`

# Fichero de configuración

Fichero:

/etc/yp.conf

- domain ypdomain.net server 192.168.2.2

/etc/hosts

- 192.168.2.2 server

/etc/yp.serv.conf

- dns: no
- files: 30
- xfr\_check\_port: yes
- \*: \*: shadow.byname: port
- \*: \*: password.adjunct.byname: port

/etc/sysconfig/network

- NISDOMAIN:"ypdomain.net"

/var/yp/securenets

- host 127.0.0.1
- 255.255.255.0 192.168.2.0