

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
- *: *: shadowbyname: port
- *: *: password.adjunctbyname: port

/etc/sysconfig/network

ASORC

Administración de sistemas operativos y
redes de computadores

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

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

/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
```

Configuración de la red

Objetivo: Asignar una dirección IP estática/dinámica al sistema

Fichero: /etc/sysconfig/network-scripts/ifcfg-enp0s8

Caso práctico:

IP estática: BOOTPROTO=none
 IPADDR=192.168.56.12
 NETMASK=255.255.255.0
 #GATEWAY=192.168.2.1
 #DNS1=8.8.8.8

IP dinámica: BOOTPROTO=dhcp

Comprobación: ifconfig / ip a

Rocky 9: Red

Opción 1: Nmcli: NetworkManager

```
nmcli con mod ens8 ipv4.addresses 192.168.1.10/24  
nmcli con mod ens8 ipv4.gateway 192.168.1.1  
nmcli con mod ens8 ipv4.method manual  
nmcli con mod ens8 ipv4.dns "192.168.1.1"  
nmcli con up ens8
```

```
nmcli con mod enp0s8 ipv4.addresses 192.168.56.31/24  
nmcli con mod enp0s8 ipv4.method manual  
nmcli con down enp0s8; nmcli con up enp0s8
```

DHCP: nmcli con mod enp0s8 ipv4.method auto

```
nmcli con mod enp0s8 ipv4.method auto  
nmcli con reload  
nmcli con down enp0s8 ; nmcli con up enp0s8
```

Opción 2: Fichero:

/etc/sysconfig/network-scripts/enp0s8

nmcli connection reload

nmcli con up ens8

```
HWADDR=08:00:27:98:06:76  
TYPE=Ethernet  
PROXY_METHOD=none  
BROWSER_ONLY=no  
BOOTPROTO=none # Static IP Address  
DEFROUTE=yes  
IPV4_FAILURE_FATAL=no  
IPV6INIT=yes  
IPV6_AUTOCONF=yes  
IPV6_DEFROUTE=yes  
IPV6_FAILURE_FATAL=no  
IPV6_ADDR_GEN_MODE=stable-privacy  
NAME=enp0s3 # Connection Name  
DEVICE=enp0s3 # Device Name  
ONBOOT=yes # Activate on Boot  
IPV6_PRIVACY=no  
PEERDNS=no  
UUID=3c36b8c2-334b-57c7-91b6-4401f3489c69  
IPADDR=192.168.1.10 # IP Address  
NETMASK=255.255.255.0 # NetMask  
GATEWAY=192.168.1.1 # Gateway / Router  
DNS1=192.168.1.1 # DNS Server 1  
DNS2=8.8.8.8 # DNS Server 2  
DOMAIN=itzgeek.local # Default Domain Search
```

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

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
```

Debian 10: Red

NetworkManager

Fichero: /etc/network/interfaces

DHCP:

```
iface enp0s8 inet dhcp
```

Static IP:

```
iface enp0s8 inet static
address 192.168.100.12
netmask 255.255.255.0
network 192.168.100.0
broadcast 192.168.100.255
gateway 192.168.100.1
```

```
# The primary network interface
auto enp0s3
allow-hotplug enp0s3
iface enp0s3 inet static
    address 192.168.250.99
    netmask 255.255.255.0
    network 192.168.1.1
    broadcast 192.168.255.255
    gateway 192.168.1.1
```

Restart service:

```
systemctl restart networking
```

FreeBSD 12: Red

Static IP address on FreeBSD 12.

```
# vi /etc/rc.conf
```

Add:

```
ifconfig_em1="inet 192.168.13.4 netmask 255.255.255.0"  
defaultrouter="192.168.13.1"
```

Restart FreeBSD network service and routing table:

```
# /etc/rc.d/netif restart && /etc/rc.d/routing restart
```

To configure an interface for DHCP, reset the settings to:

```
ifconfig_em1="DHCP"
```

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:

```
mdamad -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

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”.

Gestión de repositorios

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

Instalación: dnf -y install 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`

Fichero de configuración

Caso práctico:

- `systemctl restart vsftpd`
- `systemctl restart dhcpcd`
- `systemctl restart xinetd`

Comprobación:

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

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 dhcpcd:

- Igual que para LTSP

Servicio tftp:

- Igual que para LTSP

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
- Usuarios autorizados: AllowUsers marc
- Acceso ROOT: PermitRootLogin no

Caso práctico:

- systemctl restart sshd
- netstat -tunlp

Comprobación: ssh -p 1234 marc@192.168.2.2

Rocky 9 (Debian 10): SSHD

Install the SSH server package openssh by using the dnf command:

```
# dnf install openssh-server      (apt install openssh-server)
```

Start the sshd daemon and set to start after reboot:

```
# systemctl start sshd  
# systemctl enable sshd
```

Confirm that the sshd daemon is up and running:

```
# systemctl status sshd
```

Open the SSH port 22 to allow incoming traffic:

```
# firewall-cmd --zone=public --permanent --add-service=ssh  
# firewall-cmd --reload
```

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 Rocky 7-Minimal 64-bit  
kernel Rocky/vmlinuz  
append initrd=Rocky/initrd.img method=ftp://192.168.2.2/Rocky devfs=nomount
```

Servidor PXE

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

Instalación: yum -y install dhcpcd tftp-server vsftpd syslinux

Configuración previa:

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

FreeBSD: SSHD

OpenSSH Server is Installed by default but not enabled

```
# vi /etc/rc.conf
```

Add the following line at the end:

```
sshd_enable="YES"
```

```
# /etc/rc.d/sshd start
```

```
# sockstat -4l ó sockstat -6l
```

SSHD Configuration:

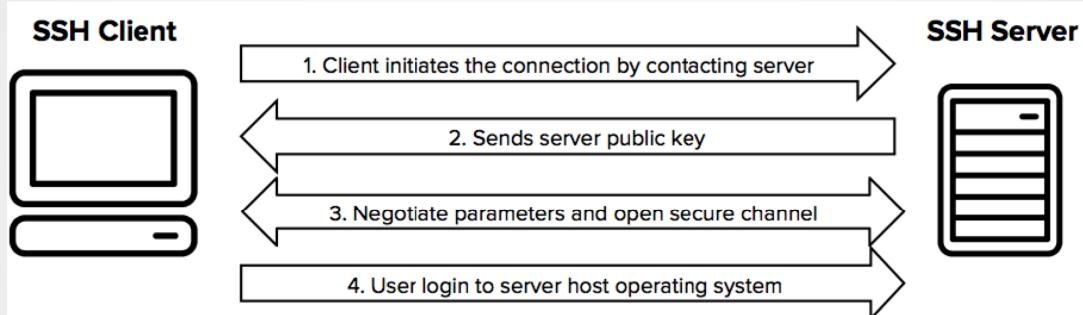
```
# vi /etc/ssh/sshd_config
```

```
PermitRootLogin no
```

```
AllowUsers bernardo
```

Servidor SSH

Objetivo: Public key Authentication



```
# ssh-keygen -t rsa  
# ssh-copy-id bernardo@192.168.56.11
```

Fichero de configuración

Caso práctico:

```
systemctl restart nfs-server
```

```
systemctl restart dhcpcd
```

```
systemctl restart xinetd
```

Comprobación:

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

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
}
```

Servidor RDP

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

Instalación: dnf install xrdp

Puerto: 3389

Caso práctico:

- systemctl start xrdp

Comprobación: xfreerdp marc@192.168.2.2:3389

Servidor VNC

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

Instalación: dnf -y install tigervnc-server tigervnc-server-module

Puerto: 5901

User side:

1. Crear password para acceso remoto
2. Crear fichero de configuración para el

usuario

Root side:

- 3.- Indicar qué usuarios inician sesión y puerto
- 4.- Arrancar servicios
- 5.- Validar servicio activo
- 6.- Firewall roules

Fichero de configuración

Configuración:

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

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

/etc(exports

```
 /opt/ltspl/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 "pixelinux.0";
```

```
    option root-path "192.168.2.2:/opt/ltspl/amd64";
```

```
}
```

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 dhcpcd tftp-server syslinux

Servidor VNC

Caso práctico: 1.- Crear password para acceso remoto

```
# su – Bernardo
$ vncpasswd
Password: / Verify:
Would you like to enter a view-only password (y/n)? n
```

Caso práctico: 2.- Crear fichero de configuración para el usuario

```
$ vi .vnc/config
session=gnome
securitytypes=vncauth,tlsVNC
desktop=sandbox
geometry=1024x768
alwaysShared
```

Servidor VNC

Caso práctico: 3.- Indicar qué usuarios inician sesión y puerto

```
# vi /etc/tigervnc/vncserver.users  
:1=bernardo
```

Caso práctico: 4.- Arrancar servicios

```
# systemctl start vncserver@:1  
# systemctl enable vncserver@:1
```

Caso práctico: 5.- Validar servicio activo

```
# netstat -tunlp | grep vnc  
tcp      0      0 0.0.0.0:5901          0.0.0.0:*      LISTEN    8640/Xvnc  
tcp6     0      0 :::5901              :::*        LISTEN    8640/Xvnc
```

```
# systemctl status vncserver@:1
```

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
```

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

Servidor VNC

Caso práctico: 6.- Firewall

```
# firewall-cmd --permanent --add-port=5901/tcp  
success  
# firewall-cmd --reload  
success  
#
```

Comprobación: vncviewer bernardo@192.168.2.2:5901

Servidor VNC on FreeBSD

```
# pkg install x11vnc  
  
$ x11vnc -storepasswd  
$ x11vnc -xkb -forever -rfbauth /home/bernardo/.vnc/passwd -ncache 10 -  
geometry 1024x768
```

Comprobación:

Server: sockstat
\$ sockstat -4
Cliente: vncviewer
bernardo@192.168.56.136:5900

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/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 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";
};
```

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

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

Caso práctico:

- tar zxfv 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>

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
```

Fichero de configuración

/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.
```

```
#!/bin/bash
# This script is prepared to backup data in two ways: weekly or daily
# Add to crontab with -d to have a daily backup respect to the saturday.
# else the file will be uploaded to Mega
#
# Variables to fill
#
# backup file name of the absolute copy:
BACKUP_FILE_NAME_ABS='_Abs'
# backup file name of the differential copy:
BACKUP_FILE_NAME_DIFF='_Diff'
# Name of the directory to have the backup
DIR='/r1data'
# Begin the script
#
. ~/.bash_profile
if [ $1 = "-d" ]
then
  echo 'copia diferencial'
  DATE=`date -d "last sat" +%Y%m%d`
  ADD_TO_TAR='-N '$DATE
  TYPE=$BACKUP_FILE_NAME_DIFF
else
  echo 'copia absoluta'
  DATE=""
  ADD_TO_TAR=""
  TYPE=$BACKUP_FILE_NAME_ABS
```

Backup: rsync

Local to remote

```
$ rsync -a ~/dir1 username@remote_host:destination_directory
```

remote to Local

```
$ rsync -a username@remote_host:/home/username/dir1  
place_to_sync_on_local_machine
```

-z data travels compressed

-P shows progress bar and allow partial transfers

Servidor DHCP

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

Instalación: dnf install dhcp-server

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

DHCP: 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;
}
```

Backup: tar

Crear la copia full (la del día 9 de Febrero):

```
$ tar -cpvzf "fullbackup_`date +%d%m%Y`.tgz"
/home/nebul4ck/recursos/*
```

Crear copia diferencial con los cambios ocurridos desde el *Lunes 9 al Miércoles 11 del 2015*

```
$ tar -cpvzf "dif1_backup_`date +%d%m%Y`.tar.gz"
/home/nebul4ck/recursos/* -N 09-feb-15
```

Crontab: crontab -e

```
# crontab -e
SHELL=/bin/bash
MAILTO=root
HOME=/root
# For details see man 4 crontabs
# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# || .----- day of month (1 - 31)
# ||| .---- month (1 - 12) OR jan,feb,mar,apr ...
# |||| .--- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |||||
# * * * * * username command to be executed

0 1 * * 6 /bin/sh /root/backup.sh -a >/dev/null
0 1 * * 0,1,2,3,4,5 /bin/sh /root/backup.sh -d >/dev/null
```

DHCP

dhclient enp0s8

cat /var/lib/dhclient/dhclient.leases

Servidor NFS

Objetivo: Permitir el acceso remoto a un sistema de archivos através 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

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 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

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  
# Ficheos ocultos  
hide dot file = Yes  
# Papelera de reciclaje  
vfs objects = recycle  
recycle:repository = Recycle Bin
```

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
```

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
```

Servidor MYSQL-MARIADB

Objetivo: Gestionar un sistema de bases de datos.

Instalación: dnf -y install mariadb-server

Puerto: 3306

Caso práctico:

- systemctl start mariadb
- mysql_secure_installation
- mysql -u root -p
 - create database asorc
 - grant all privileges on asorc.* to 'bernardo'@'%' identified by 'passwd'

Comprobación:

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

Servidor MYSQL-MARIADB

Objetivo: Instalar versión 10.6

Instalación: Crear fichero repo: /etc/yum.repos.d/mariadb.repo

```
[mariadb]
name = MariaDB
baseurl = http://yum.mariadb.org/10.6/rhel8-amd64
module_hotfixes=1
gpgkey=https://yum.mariadb.org/RPM-GPG-KEY-MariaDB
gpgcheck=1
```

```
# dnf install mariadb-server mariadb
# mysql_install_db
# mysql_upgrade
# systemctl start mariadb
# mariadb-secure-installation
```

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

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:

Servidor MYSQL-MARIADB

```
# mysql -u root -p
> use asorc
> create table users(user_name varchar(255), user_address varchar(255),
details varchar(255));
> insert into users(user_name,user_address,details) values ('Bernardo
Ledesma', 'Mi casa', 'Mis Detalles');
> select * from users;
```

PHP installation

```
# dnf install php php-{common,gmp,fpm,curl,intl,pdo,mbstring,gd,xml,cli,zip,mysqli}
```

```
# php -version
```

To create a php program:

```
<?php
```

```
phpinfo();
```

```
?>
```

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
```

Fichero de configuración

Insertar datos en el directorio:

Crea archivos standard

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

Crear subconjunto de archivos ldif

```
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"
```

Servidor HTTP

Objetivo: Servir contenido web

Instalación: dnf -y install httpd

Puerto: 80

Fichero: /etc/httpd/conf/httpd.conf

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

Caso práctico: apachectl configtest prueba la configuración antes de arrancar.

```
# echo Apache Rocky 9 - Bernardo / Rocky 9 >
/var/www/html/index.html
# apachectl configtest
# systemctl start httpd
Incluir nombre de dominio en el DNS (opcional)
```

Comprobación:

- http://192.168.56.11

Fichero de configuración

/etc/httpd/conf/httpd.conf

```
ServerName www.myserver.name:80
```

/etc/httpd/conf.d/embedidosgutierrez.conf

```
<VirtualHost *:80>
    DocumentRoot /var/www/html/embedidosgutierrez
    ServerName www.embedidosgutierrez.net
</VirtualHost>

/var/www/html/embedidosgutierrez/index.html
<html>
    <head></head>
    <body> Web de Embutidos Gutierrez </body>
</html>
```

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

/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

/etc/httpd/conf.d/virtualdomains.conf

<virtualhost *:80>

Servername www.**emburtidosgutierrez.com**

Serveralias **emburtidosgutierrez.com**

Documentroot /var/www/**emburtidosgutierrez.com**/html

<directory /var/www/**emburtidosgutierrez.com**/html>

Options -Indexes +FollowSymLinks

Allowoverride All

</directory>

Errorlog /var/log/httpd/**emburtidosgutierrez.com**-error.log

Customlog /var/log/httpd/**emburtidosgutierrez.com**-access.log combined

</virtualhost>

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
```

Fichero de configuración

Creación de la autoridad certificadora:

```
cd /etc/openldap/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
```

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'

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  
# chroot users (jail)  
chroot_local_user=YES  
chroot_list_enable=YES  
chroot_list_file=/etc/vsftpd/chroot_list  
allow_writeable_chroot=YES
```

Añadir /sbin/nologin a /etc/shells

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 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

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
```