**Administración de Sistemas Operativos y Redes de Computadores**

**2021-22**

**Práctica 3**

*Nikita Polyanskiy*

*Y4441167L*

**Índice**

Contenido

[Rocky Linux 3](#_Toc90220897)

[VSFTPD (FTP): 3](#_Toc90220898)

[Openfire + Spark 6](#_Toc90220899)

[Proxy(squid): 11](#_Toc90220900)

[RAID: 13](#_Toc90220901)

[Nagios: 17](#_Toc90220902)

[FreeBSD (Unix) 20](#_Toc90220903)

[FTP 20](#_Toc90220904)

[Openfire + Spark 22](#_Toc90220905)

[Nagios 23](#_Toc90220906)

[Proxy (Squid) 24](#_Toc90220907)

[Raidz 26](#_Toc90220908)

[Windows Server 2022 28](#_Toc90220909)

[FTP 28](#_Toc90220910)

[Openfire 30](#_Toc90220911)

[Raid 5: 31](#_Toc90220912)

[PRTG (monitor de red) 34](#_Toc90220913)

[Backup (Cobian) 35](#_Toc90220914)

# Rocky Linux

## VSFTPD (FTP):

Primero instalamos el paquete:

dnf -y install vsftpd

dnf -y install ftp (para el cliente)

Luego crearemos los 2 usuarios que deseamos usar, el primero estará enjaulado, el segundo no:

adduser ftp1

passwd ftp1

(ftp1, ASORC123)

adduser ftp2

passwd ftp2

([ftp2,ASORC123](ftp://ftp2,ASORC123))

Luego en la carpeta de cada usuario crearemos una carpeta ftp\_folder, y quitaremos los permisos de escritura :

mkdir ftp\_folder

chown [ftp2:ftp2](ftp://ftp2:ftp2) ftp\_folder

chmod a-w ftp\_folder

comprobamos que los permisos estén bien:

Texto

Descripción generada automáticamente con confianza baja

Luego crearemos una carpeta donde el usuario podrá escribir:

mkdir /ftp\_folder/files

chown [ftp2:ftp2](ftp://ftp2:ftp2) /ftp\_folder/files

Comprobamos que los permisos estén bien:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente con confianza media

Luego para configurar el acceso deberemos modificar el arhcivo /etc/vsftpd.conf, y asegurarnos de tener las siguientes líneas:

anonymous\_enable=NO

local\_enable=YES

write\_enable=YES

chroot\_local\_user=YES

chroot\_list\_enable=YES

chroot\_list\_file=/etc/vsftpd/nonchroot\_list

user\_sub\_token=$USER

local\_root=/home/$USER/ftp\_folder

userlist\_enable=YES

userlist\_file=/etc/vsftpd/user\_list

userlist\_deny=NO

Luego crearemos/modificaremos 2 archivos, /etc/vsftpd/user\_list, donde agregaremos los usuarios ftp1 y ftp2, y /etc/vsftpd/nonchroot\_list, donde agregaremos al usuario que deseamos no enjaular, en este caso ftp2.

Una vez hecho esto reiniciamos el servicio

systemctl restart vsftpd

Y podemos comprobar el funcionamiento del servicio:

ftp -p 192.168.137.222

Texto

Descripción generada automáticamente`

El usuario ftp1 no puede salir del directorio:

Texto

Descripción generada automáticamente

Pero el usuario ftp2 si:

Texto

Descripción generada automáticamente

## Openfire + Spark

wget <http://download.igniterealtime.org/openfire/openfire-4.6.4-1.i686.rpm>

yum install -y openfire-4.6.4-1.i686.rpm

yum install -y glibc.i686

chkconfig openfire on

systemctl start openfire

yum install -y postgresql postgresql-server postgresql-devel postgresql-libs

postgresql-setup initdb

systemctl start postgresql

systemctl enable postgresql

systemctl status postgresql

Texto

Descripción generada automáticamente

su postgres

createdb openfire

createuser -P openfire

Password: 123456

psql -U postgres -d postgres -c “ALTER USER postgres WITH PASSWORD ‘123456”;”

nano /var/lib/pgsql/data/pg\_hba.conf

Captura de pantalla con letras blancas

Descripción generada automáticamente

systemctl restart postgresql

Seguimos los pasos de instalación:

Domain name: openfire-nik

server host name :openfire-nik

openfire

123456

[admin@example.com](mailto:admin@example.com)

ASORC123

Interfaz de usuario gráfica

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Creamos nuevos usuarios

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

En las maquinas clientes instalamos Spark:

(Linux):

yum install java

wget <https://www.igniterealtime.org/downloadServlet?filename=spark/spark-2.9.4.rpm>

yum install -y downloadServlet?filename=spark/spark-2.9.4.rpm

(Windows) <https://www.igniterealtime.org/downloads/index.jsp>

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Word

Descripción generada automáticamente

## Proxy(squid):

Instalamos el paquete:

dnf install -y squid

Modificamos el archive de configuración:

nano /etc/squid/squid.conf

Agregamos las siguientes líneas:

acl denegar dstdomain “/etc/squid/bad-sites.acl”

http\_access deny denegar

Crear fichero bad-sites.acl en la carpeta squid:

.facebook.com

En el navegador:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

systemctl restart squid

Y comprobamos que funciona:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

## RAID:

Creamos 4 nuevas unidades de almacenamiento:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Comprobamos que tenemos los 4 discos asignados (sde,sdb,sdc,sdd):

Texto

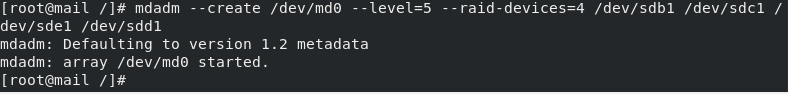
Descripción generada automáticamente

Creamos partición en los 4 discos:

Texto, Calendario

Descripción generada automáticamente

Configuramos RAID 5:



Comprobamos que RAID 5 esté activo en los 4 discos:

Texto

Descripción generada automáticamente

Modificamos el siguiente fichero:

nano /etc/sysconfig/raid-check

CHECK\_DEVS="md0"

Lo montamos en nuestro Sistema:Texto

Descripción generada automáticamente con confianza media

Desmontamos un disco:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Comprobamos que hay fallo en el disco “sde”:

Texto

Descripción generada automáticamente

Añadimos un nuevo disco:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Arreglamos el fallo reemplazando el disco:

Texto

Descripción generada automáticamente

## Nagios:

Instalamos todas las dependencias:

dnf update

dnf install -y php perl @httpd wget unzip glibc automake glibc-common gettext autoconf php php-cli gcc gd gd-devel net-snmp openssl-devel unzip net-snmp postfix net-snmp-utils

dnf -y groupinstall "Development Tools"

systemctl enable --now httpd php-fpm

systemctl start httpd

systemctl start php-fpm

systemctl status httpd

systemctl status php-fpm

Instalamos nagios:

wget <https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz>

tar -xzf nagios-4.4.6.tar.gz

cd nagios-\*/

./configure

make all

make install-groups-users

usermod -aG nagios apache

make install

make install-init

make install-daemoninit

make install-commandmode

make install-config

make install-webconf

htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin

chown apache:apache /usr/local/nagios/etc/htpasswd.users

chmod 640 /usr/local/nagios/etc/htpasswd.users

systemctl restart httpd

systemctl enable nagios --now

Texto

Descripción generada automáticamente

nagiosadmin

password: 123

Instalamos los plugins:

wget <https://github.com/nagios-plugins/nagios-plugins/releases/download/release-2.4.0/nagios-plugins-2.4.0.tar.gz>

tar xzf nagios-plugins-2.4.0.tar.gz

cd nagios-plugins-2.4.0.tar.gz

./configure

make

make install

Comprobamos su funcionamiento:  
Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

# FreeBSD (Unix)

## FTP

Creamos 2 nuevos usuarios:  
ftp1 | pass:123

ftp2 | pass:123

service ftpd enable

service ftpd start

nano etc/ftpchroot (introducimos el usuario enjaulado):

ftp1

Desde Windows comprobamos que funciona:

(ftp1 enjaulado)

Texto

Descripción generada automáticamente

(ftp2 no enjaulado):

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

## Openfire + Spark

Instalamos los paquetes:

pkg install openfire

pkg install postgresql11-server postgresql11-client

Añadimos las siguientes lineas a /etc/rc.conf:

openfire\_enable=”YES”

postgresql\_enable=”YES”

Iniciamos el servicio openfire

service openfire start

service openfire status

Iniciamos la bd de postgre:

/usr/local/etc/rc.d/postgresql initdb

service postgresql start

su – postgres

createdb openfire;

createuser -P openfire

pass: 123

psql -U postgres -d postgres -c “ALTER USER postgres WITH PASSWORD ‘123’”;

Interfaz de usuario gráfica

Descripción generada automáticamente

Configuramos el servidor openfire de la misma forma que en Rocky Linux.

Instalamos el cliente Spark como lo haríamos para Rocky Linux.

Comprobamos que funciona:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

## Nagios

Se han seguido las instrucciones de esta página:

<https://kifarunix.com/install-nagios-core-on-freebsd-13/>

Comprobamos que funciona:

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente

## Proxy (Squid)

Instalamos el paquete

pkg install squid

squid -f /usr/local/etc/squid/squid.conf -k parse

nano /etc/rc.conf

squid\_enable=”YES”

Modificamos el archivo /usr/local/etc/squid/squid.conf y agregamos las siguientes líneas:

acl denegar dstdomain “/usr/local/etc/squid/bad-sites.acl”

http\_access deny denegar

Creamos el archivo bad-sites.acl en la carpeta de squid:

.facebook.com

En nuestro navegador activamos el proxy manual:

Texto, Tabla

Descripción generada automáticamente con confianza media

Reiniciamos el servicio

service squid restart

Comprobamos que funciona:  
Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

## Raidz

Agregamos 4 unidades de almacenamiento como lo hemos hecho para Rocky Linux.

Luego creamos la unidad raidz:

Texto

Descripción generada automáticamente

Quitamos un disco:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Vemos que hubo un fallo:

Texto

Descripción generada automáticamente

Lo arreglamos de la siguiente forma:Texto

Descripción generada automáticamente

# Windows Server 2022

## FTP

Instalamos FileZilla Server 0.9.60:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

En editar -> usuarios añadiremos 2 usuarios, ftp1 (enjaulado en una carpeta dentro del directorio C:\ftp\ftp1) y otro desenjaulado (con directorio C:\)

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Comprobamos que funciona:

ftp1 (enjaulado):

Texto

Descripción generada automáticamente

ftp2 (desenjaulado):

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

## Openfire

Se instala openfire con JRE incluido, y spark: <https://www.igniterealtime.org/downloads/>

Una vez instalado se abrirá la pagina localhost:9090, los pasos de instalación son iguales que para Rocky Linux y FreeBSD.

Instalamos Spark, entramos con nuestros usuarios, y los añadimos a contactos, luego comprobamos que funciona:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

## Raid 5:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Abrimos Disk Management, seleccionamos uno de los discos que hemos añadido:

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Añadimos los demás discos y seguimos los pasos de instalación y formateo.

Una vez terminado debería verse así:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Añadimos un archivo de prueba al disco:

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Quitamos uno de los discos desde VB, y añadimos uno nuevo:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Primero convertiremos nuestro nuevo disco a dinámico, luego reparamos el volumen D:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Y elegimos el nuevo disco.

Comprobamos que esta reparado:

Tabla

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

## PRTG (monitor de red)

Instalamos el servicio desde la pagina oficial: <https://www.paessler.com/prtg?gclid=CjwKCAiAtdGNBhAmEiwAWxGcUi_weH8qL0dGDOPz2IaPIRh6c_CIq0kn7O27eaiH01jAxlMIxjobAxoC-nAQAvD_BwE>

Una vez instalado entramos al servicio:  
Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Comprobamos que funciona:

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

## Backup (Cobian)

Descargamos Cobian Backup 11: <https://cobiansoft.com>

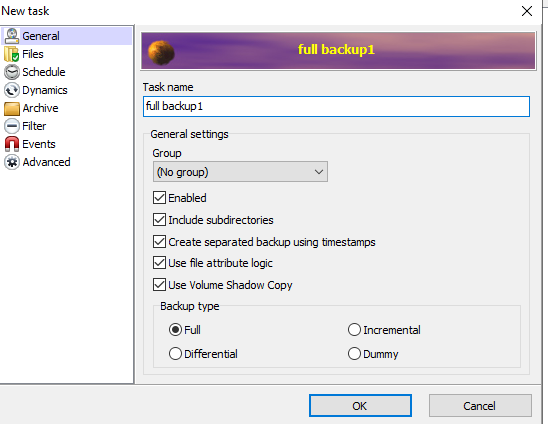
Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Una vez instalado, creamos una nueva tarea:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente



Añadimos el directorio que deseamos hacer backup, y el destino de la copia:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

En la carpeta backup1 comprobamos que se ha realizado con éxito:

