

ADMINISTRACION DE SISTEMAS OPERATIVOS Y REDES DE COMPUTADORES

PRACTICA 3:

Servicios avanzados sobre servidores

Xavier Siguero Mora 48787124F

Contenido

Windows Server 2016	3
Mensajería instantánea (Jabber)	3
RAID: (en t. instalación, a posteriori. Administración. Reemplazo de un disco).....	3
Servidor Web (dominios virtuales).....	3
Backup (absoluto, incremental y sincronización; AMANDA, RSYNC, etc).....	4
Servidor de BD: Mysql, PostgreSQL, Oracle express. (Obtener datos mediante http)	4
Proxy Cache (Squid): Restricción de contenidos, páginas, usuarios, autenticación LDAP.....	5
Monitorización de servicios (Nagios)	6
CentOS.....	7
RAID: (en t. instalación, a posteriori. Administración. Reemplazo de un disco).....	7
Servidor Web (dominios virtuales): CREAR DOMINIO BD	7
Backup (absoluto, incremental y sincronización; AMANDA, RSYNC, etc).....	9
Servidor de trabajo en grupo (Hula, OpenXchange, Opengroupware, Microsoft Exchange, ...)	
.....	9
Servidor de BD: PostgreSQL.....	9
Mensajería instantánea (Jabber)	11
Proxy Cache (Squid): Restricción de contenidos, páginas, usuarios, autenticación LDAP.....	11
Monitorización de servicios (Nagios)	12
FreeBSD	13
Servidor Web (dominios virtuales) :CREAR DOMINIO BD	13
Backup (absoluto, incremental y sincronización; AMANDA, RSYNC, etc).....	15
Servidor de BD: MYSQL	15
Mensajería instantánea (Jabber)	16
Proxy Cache (Squid): Restricción de contenidos, páginas, usuarios, autenticación LDAP.....	16
Monitorización de servicios (Nagios)	16

Windows Server 2016

Mensajería instantánea (Jabber)

Descargamos java(<https://www.java.com/es/download/manual.jsp>)

Descargamos openfire(https://www.igniterealtime.org/downloads/download-landing.jsp?file=openfire/openfire_4_2_3_x64.exe)

Descargamos spark(<https://github.com/igniterealtime/Spark/releases/2.6.3>)

Lo ejecutamos en el cliente y en el servidor ya funcionaria sin ningún problema.

RAID: (en t. instalación, a posteriori. Administración. Reemplazo de un disco)

Primero insertamos 5 discos, 2 para el RAID 1 y 3 para el RAID 5, luego vamos a Crear y formatear particiones de disco duro. Hacemos click derecho en uno de los discos sin formato y pulsamos Nuevo volumen reflejado en el caso del RAID 1 y Nuevo volumen RAID 5 para el RAID 5. Elegimos los discos a usar y le asignamos una letra.

Servidor Web (dominios virtuales)

Descargamos e instalamos Xampp. El programa aparecerá minimizado en la barra de tareas, lo abrimos y pulsamos Start apache. Una vez iniciado entramos en <http://127.0.0.1/dashboard/> para comprobar que funciona correctamente.

Creamos dos directorios: xampp/htdocs/asorcweb1.com y xampp/htdocs/asorcweb2.com, que es donde se almacenaran los archivos del servidor.

Editamos C:\Windows\System32\drivers\etc\host:

```
127.0.0.1 www.asorcweb1.com
```

```
127.0.0.1 www.asorcweb2.com
```

Y luego C:\xampp\apache\conf\extra\httpd.vhost.conf:

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot "C:/xampp/htdocs/"
    <Directory "C:/xampp/htdocs/">
        Options All
        AllowOverride All
        Require all granted
    </Directory>
    ServerName localhost
    ErrorLog "logs/localhost-error.log"
    CustomLog "logs/localhost-access.log" common
</VirtualHost>

<VirtualHost *:80>
    ServerAdmin webmaster@asorcweb1.com
    DocumentRoot "C:/xampp/htdocs/asorcweb1.com"
    <Directory "C:/xampp/htdocs/asorcweb1.com">
        Options All
```

```

        AllowOverride All
        Require all granted
    </Directory>
    ServerName asorcweb1.com
    ServerAlias www.asorcweb1.com
    ErrorLog "logs/asorcweb1-error.log"
    CustomLog "logs/asorcweb1-access.log" common
</VirtualHost>

<VirtualHost *:80>
    ServerAdmin webmaster@asorcweb2.com
    DocumentRoot "C:/xampp/htdocs/asorcweb2.com"
    <Directory "C:/xampp/htdocs/asorcweb2.com">
        Options All
        AllowOverride All
        Require all granted
    </Directory>
    ServerName asorcweb2.com
    ServerAlias www.asorcweb2.com
    ErrorLog "logs/asorcweb2-error.log"
    CustomLog "logs/asorcweb2-access.log" common
</VirtualHost>

```

Descargamos wordpress(<https://es.wordpress.org/download/>) y lo descomprimos en asorcweb2.com.

Usuario: Root

Contraseña: root

Descargamos joomla(<https://downloads.joomla.org/>) y lo descomprimos en asorcweb1.com

Usuario: xsm2

Contraseña: root

Accedemos a los dos dominios y realizamos la configuración inicial.

[Backup \(absoluto, incremental y sincronización; AMANDA, RSYNC, etc\)](#)

Descargamos e instalamos Cobian Backup. Lo iniciamos y pulsamos Nueva tarea. Primero se creará el backup absoluto. Tendremos que elegir la ruta donde se encuentran los archivos de los que queremos hacer la copia de seguridad y la ruta donde se almacenara esa copia.

Después repetimos el proceso, pero esta vez elegimos backup incremental.

[Servidor de BD: Mysql, PosgreSQL, Oracle express. \(Obtener datos mediante http\)](#)

Para el servidor de bases de datos usaremos xampp. Pulsamos en Start mysql para iniciar mysql y entramos en <http://127.0.0.1/phpmyadmin/>

Creamos una tabla y un usuario usuario.

Creamos un archivo php que lea la base de datos y lo almacenamos en uno de nuestros dominios virtuales:

```
<?php
$servername = "localhost";
$username = "xsm2";
$password = "root";
$dbname = "personas";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT * FROM personas";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        echo "nombre: " . $row["nombre"]. " - apellido: " .
        $row["apellido"]. "<br>";
    }
} else {
    echo "0 results";
}
$conn->close();
?>
```

Proxy Cache (Squid): Restricción de contenidos, páginas, usuarios, autenticación LDAP.

Descargamos squid(<http://www.soluciones.si/blog/2013/04/19/configuracion-de-proxy-squid-sistema-windows>) y lo extraemos en c:/

En la carpeta etc copiamos los archivos y borramos el default del nombre

Abrimos cmd en sbin y hacemos squid -i y squid -z

Abrimos servicios e iniciamos squid

```
user:$apr1$TCVcHj3N$aDdGQabkn76kg1WrjQBtz/
acl localnet src 10.0.2.0/24
acl localnet src 192.168.56.0/24
acl expreg-denegadas url_regex "/etc/squid/listas/expreg-
denegadas"
mkdir /etc/squid/listas
nano /etc/squid/listas/expreg-denegadas
nano /etc/squid/squid.conf
http_access allow localnet localhost !expreg-denegadas
```

Abrir configuración de firefox, proxy manual

En todas las casillas: 10.0.2.7, 3128

```
htpasswd -c /etc/squid/.htpasswd user
```

```
nano /etc/squid/squid.conf
auth_param basic program /usr/lib64/squid/basic_ncsa_auth
/etc/squid/.htpasswd
auth_param basic children 5
auth_param basic realm Squid Basic Authentication
auth_param basic credentialsttl 5 hours
acl password proxy_auth REQUIRED
http_access allow password
```

Monitorización de servicios (Nagios)

Descargamos e instalamos Clouddradar Windows. Creamos una cuenta y nos registramos tanto en el cliente que nos hemos descargado como en el navegador en la página web de clouddradar. Para enlazar el pc a nuestra cuenta nos saldrán unas instrucciones donde deberemos copiar unos códigos del navegador al cliente:

```
hub_url=https://hub.clouddradar.io/ctrapper/
hub_user=3aaec921-f082-489a-8a11-107a418730c9
hub_password=ivMK1gjW2lri
```

frontman

```
hub_url=https://hub.clouddradar.io/checks/
hub_user=0f3f357e-a38d-4bef-881c-23441ff32883
hub_password=juy0HnT3WGYN
```

CentOS

RAID: (en t. instalación, a posteriori. Administración. Reemplazo de un disco)

Insertamos 5 discos.

Para crear el RAID 1:

```
mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sdc
/dev/sdd
cat /proc/mdstat
mkfs.ext4 /dev/md0
mkdir /mnt/var
mount /dev/md0 /mnt/var
rsync -vza /var/ /mnt/var/
```

Para el crear RAID 5:

```
mdadm --create /dev/md1 --level=5 --raid-devices=3 /dev/sde
/dev/sdf /dev/sdg
mdadm --detail /dev/md1
mkfs.ext4 /dev/md1
mkdir /mnt/usr
mount /dev/md1 /mnt/usr
rsync -vza /usr/ /mnt/usr/
```

Si simulamos fallo del disco:

```
mdadm /dev/md0 --fail /dev/sdc --remove /dev/sdc
```

Y lo sustituimos con un sexto disco r:

```
mdadm --zero-superblock /dev/sdc
mdadm --add /dev/md0 /dev/sde
```

Otros comandos interesantes:

```
mdadm -a /dev/nombreRaid /dev/newDisk //añadir
mdadm -r /dev/nombreRaid /dev/newDisk //eliminar
mdadm -f /dev/nombreRaid /dev/newDisk //marcar como faltante
```

Servidor Web (dominios virtuales): CREAR DOMINIO BD

Iniciamos el servicio:

```
service httpd start
yum install php php-mysql php-pgsql
chkconfig httpd on
```

Creamos las carpetas para nuestros dominios:

```
cd /var/www/html
mkdir asorcweb1.com
mkdir asorcweb2.com
```

Y los configuramos:

nano /etc/httpd/conf.d/asorcweb1.conf

```
Alias /asorcweb1 /var/www/html/asorcweb1.com
```

```
Nano /etc/hosts
```

```
10.0.2.7 localhost.localdomain localhost
```

```

127.0.0.1    localhost localhost.localdomain localhost4
localhost4.localhostdomain
::1         localhost localhost.localdomain localhost6
localhost6.localhostdomain
192.168.56.104 www          www.asorcweb1.com
192.168.56.1  www          www.asorcweb1.com
192.168.56.104 www          www.asorcweb2.com
192.168.56.1  www          www.asorcweb2.com

nano /etc/httpd/conf/httpd.conf
NameVirtualHost *:80
<VirtualHost *:80>
ServerName root@asorcweb1.com
DocumentRoot /var/www/html/asorcweb1.com
ServerName www.asorcweb1.com
ServerAlias www.asorcweb1.com
</VirtualHost>

<VirtualHost *:80>
ServerName root@asorcweb2.com
DocumentRoot /var/www/html/wordpress
ServerName www.asorcweb2.com
ServerAlias www.asorcweb2.com
</VirtualHost>

DirectoryIndex index.html index.cgi index.php
ServerTokens Prod
KeepAlive On

```

Instalamos mysql para poder usar wordpress y creamos una base de datos:

```

yum install mariadb mariadb-server
systemctl start mariadb
mysql -u root
CREATE DATABASE wordpressdb;
create user 'wordpress' IDENTIFIED BY 'root';
GRANT ALL PRIVILEGES ON wordpressdb.* TO 'wordpress'@'localhost'
IDENTIFIED BY 'root';
FLUSH PRIVILEGES;
Exit

```

Descargamos wordpress:

```

wget http://wordpress.org/latest.tar.gz
tar -xvf latest.tar.gz
mv -vf wordpress /var/www/html
chown -R apache:apache /var/www/html/*
cd /var/www/html/wordpress
cp wp-config-sample.php wp-config.php
nano wp-config.php

/** The name of the database for WordPress */
define('DB_NAME', 'wordpressdb');
/** MySQL database username */
define('DB_USER', 'wordpress');
/** MySQL database password */
define('DB_PASSWORD', 'root');

```


Accedemos a <http://www.asorcweb2.com/> y realizamos la configuracion inicial.

Descargamos Joomla de su página oficial, lo descomprimos y lo movemos a nuestro directorio:

```
mv joomla/ /var/www/html/
```

Accedemos a <http://www.asorcweb1.com/> y realizamos la configuracion inicial.

Backup (absoluto, incremental y sincronización; AMANDA, RSYNC, etc)

Para el absoluto usamos:

```
rsync -av Documentos/ backups
```

Para el incremental:

```
rsync -avvb --backup-dir=$PWD/backup_$(date +%d%m%y%H%M)
Documentos/ backups
```

Servidor de trabajo en grupo (Hula, OpenXchange, Opengroupware, Microsoft Exchange, ...)

Vamos a usar Zimbra:

```
yum -y install perl-core unzip libaio nmap-ncat sysstat openssh-
clients
nano /etc/hosts
10.0.2.7          localhost.localdomain localhost
wget https://files.zimbra.com/downloads/8.6.0_GA/zcs-
8.6.0_GA_1153.RHEL7_64.20141215151110.tgz
tar zxvf zcs-8.6.0_GA_1153.RHEL7_64.20141215151110.tgz
cd zcs-8.6.0_GA_1153.RHEL7_64.20141215151110
```

Entramos al proceso de instalacion y elegimos las siguientes opciones:

```
./install.sh
7
4
root12
R
A
Y
```

Servidor de BD: PostgreSQL

Instalamos postgres y creamos un usuario y una base de datos:

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

systemctl enable postgresql.service

postgresql-setup initdb

systemctl start postgresql.service

su postgres
createdb asorc
```

```

createuser -P userasorc
psql -U postgres -d postgres -c "ALTER USER postgres WITH
PASSWORD 'root';"
GRANT ALL PRIVILEGES ON DATABASE asorc TO userasorc;

nano /var/lib/pgsql/data/pg_hba.conf
local  all          postgres
md5
local  all userasorc                                md5
host   all  postgres          192.168.56.104/24
md5
host   all  userasorc          192.168.56.104/24
md5
host   all          all          192.168.56.104/24      md5
cambiamos todos los method por md5
su -
nano /etc/php.ini
extension=php_pgsql.so

psql -U userasorc -d asorc
CREATE TABLE persona(nombre varchar(15), apellido varchar(15));
INSERT INTO persona VALUES('Pedro', 'Gimenez');
INSERT INTO persona VALUES('Melania', 'Cruz');
exit

```

Creamos nuestro archivo php y lo guardamos en uno de nuestro dominios virtuales:

```

nano /var/www/html/asorcweb1/tabla.php
<?php
    $host      = "host = 127.0.0.1";
    $port      = "port = 5432";
    $dbname    = "dbname = asorc";
    $credentials = "user = userasorc password=root";

    $db = pg_connect( "$host $port $dbname $credentials" );
    if(!$db) {
        echo "Error : Unable to open database\n";
    } else {
        echo "Opened database successfully\n";
    }

    $sql =<<<EOF
        SELECT * from persona;
EOF;

    $ret = pg_query($db, $sql);
    if(!$ret) {
        echo pg_last_error($db);
        exit;
    }
    while($row = pg_fetch_row($ret)) {
        echo "nombre = ". $row[0] . "\n";
        echo "apellido = ". $row[1] . "\n";
        echo "ADDRESS = ". $row[2] . "\n";
        echo "SALARY = ". $row[4] . "\n\n";
    }

```

```

    }
    echo "Operation done successfully\n";
    pg_close($db);
?>

```

Reiniciamos el servicio:

```
systemctl restart postgresql.service
```

Mensajería instantánea (Jabber)

Instalamos openfire:

```

yum install openfire-4.2.3-1.i686.rpm
yum install glibc.i686

```

Creamos la base de datos y el usuario:

```

su postgres
createdb openfire
createuser -P openfire

```

Iniciamos el servicio y comprobamos que funciona:

```

systemctl start openfire.service
localhost:9090
jdbc:postgresql://localhost:5432/openfire

```

Descargamos Tkabber en el cliente para la comunicación y agregamos contactos con el siguiente formato: root@localhost

Proxy Cache (Squid): Restricción de contenidos, páginas, usuarios, autenticación LDAP.

Instalamos squid y lo configuramos:

```

yum -y install squid
nano /etc/squid/squid.conf
acl localnet src 10.0.2.0/24
acl localnet src 192.168.56.0/24
acl expreg-denegadas url_regex "/etc/squid/listas/expreg-denegadas"
mkdir /etc/squid/listas
nano /etc/squid/listas/expreg-denegadas
nano /etc/squid/squid.conf
http_access allow localnet localhost !expreg-denegadas

```

Abrir configuración de firefox, y seleccionamos proxy manual para poner nuestro proxy. En todas las casillas ponemos nuestra ip y el puerto por defecto: 10.0.2.7, 3128

Para la autenticación creamos un usuario y lo activamos:

```
htpasswd -c /etc/squid/.htpasswd user
```

```

nano /etc/squid/squid.conf
auth_param basic program /usr/lib64/squid/basic_ncsa_auth /etc/squid/.htpasswd
auth_param basic children 5

```

```
auth_param basic realm Squid Basic Authentication
auth_param basic credentialsttl 5 hours
acl password proxy_auth REQUIRED
http_access allow password
```

Monitorización de servicios (Nagios)

```
yum install nagios*
nano /etc/nagios/objects/contacts.cfg
```

```
htpasswd /etc/nagios/passwd nagiosadmin
systemctl start nagios
systemctl restart httpd
firefox localhost/nagios/
ADMIN root12
1 root12
auth_param basic program /usr/lib64/squid/basic_ncsa_auth
/etc/squid/.htpasswd
auth_param basic children 5
auth_param basic realm Squid Basic Authentication
auth_param basic credentialsttl 5 hours
acl password proxy_auth REQUIRED
http_access allow password
```

FreeBSD

Servidor Web (dominios virtuales) :CREAR DOMINIO BD

Instalamos apache:

```
pkg install apache24
nano rc.conf
apache24_enable="YES"
apache_http_accept_enable="YES"
```

Lo iniciamos:

```
service apache24 start
```

Iniciamos php72 ya que es la versión compatible con nagios:

```
pkg install php72-session php72-xml php72-zlib php72-simplexml
php72-mbstring php72-filter php72-ctype php72-json
```

Y configuramos apache:

```
/usr/local/etc/rc.d/apache24 start
nano /usr/local/etc /apache24/httpd.conf
Alias /wordpress /var/www/wordpress
<Directory "/var/www/wordpress">
Options Indexes
Allow from all
Require all granted
</Directory>

Alias /joomla /var/www/joomla
<Directory "/var/www/joomla">
Options Indexes
Allow from all
Require all granted
</Directory>
ServerName localhost:80
<FilesMatch "\.php$">
    SetHandler application/x-httpd-php
</FilesMatch>
<FilesMatch "\.phps$">
    SetHandler application/x-httpd-php-source
</FilesMatch>
cd /usr/local/etc/apache24/Includes
nano vhost
NameVirtualHost *:80
<VirtualHost *:80>
ServerName root@asorcweb1.com
DocumentRoot /var/www/joomla
ServerName www.asorcweb1.com
ServerAlias www.asorcweb1.com
</VirtualHost>

nano vhost2

<VirtualHost *:80>
ServerName root@asorcweb2.com
DocumentRoot /var/www/wordpress
ServerName www.asorcweb2.com
```

```

ServerAlias www.asorcweb2.com
</VirtualHost>

mkdir /var/www
mkdir /var/www/wordpress
mkdir /var/www/joomla
Nano /etc/hosts
10.0.2.29 localhost.localdomain localhost
127.0.0.1 localhost localhost.localdomain localhost4
localhost4.localdomain4
::1 localhost localhost.localdomain localhost6
localhost6.localdomain6
192.168.56.102 www www.asorcweb1.com
192.168.56.1 www www.asorcweb1.com
192.168.56.102 www www.asorcweb2.com
192.168.56.1 www www.asorcweb2.com

```

Lo reiniciamos para aplicar los cambios:

```
apachectl restart
```

Y creamos las bases de datos:

```

mysql -u root
CREATE DATABASE wordpressdb;
create user 'wordpress' IDENTIFIED BY 'root';
GRANT ALL PRIVILEGES ON wordpressdb.* TO 'wordpress'@'localhost'
IDENTIFIED BY 'root';
FLUSH PRIVILEGES;
Exit

```

Descargamos wordpress:

```

fetch http://wordpress.org/latest.tar.gz
tar -xvf latest.tar.gz
mv -vf wordpress /var/www
chown -R www:www /var/www/*
cd /var/www/wordpress
cp wp-config-sample.php wp-config.php
nano wp-config.php

/** The name of the database for WordPress */
define('DB_NAME', 'wordpressdb');
/** MySQL database username */
define('DB_USER', 'wordpress');
/** MySQL database password */
define('DB_PASSWORD', 'root');

```

<http://www.asorcweb2.com/>

Descargamos Joomla, lo descomprimos y lo movemos:

```

mv joomla/ /var/www/
cd /usr/local/etc/
cp php.ini-production php.ini
nano php.ini
memory_limit: 128M
upload_max_filesize: 20M
post_max_size: 20M
max_execution_time: 300

```

<http://www.asorcweb1.com/>

Backup (absoluto, incremental y sincronización; AMANDA, RSYNC, etc)

Instalamos rsync:

```
pkg install rsync
```

Absoluto: rsync -av Documentos/ backups

Incremental: rsync -avvb --backup-dir=\$PWD/backup_`date +%d%m%y%H%M`squi
Documentos/ backups

Servidor de BD: MYSQL

Instalamos y configuramos postgres y la base de datos:

```
/usr/sbin/pwd_mkdb -p /etc/master.passwd
pkg install mysql56-client mysql56-server
nano rc.conf
mysql_enable="YES"
mysql_upgrade
mysql -u root
create database persona;
use persona;
create table persona(nombre varchar(20), apellido varchar(20));
insert into persona values('Pedro','Gial');
pkg install php72-mysqli
```

Creamos el archive php:

Nano table.php

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT * FROM persona";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        echo "id: " . $row["id"]. " - Name: " .
        $row["firstname"]. " " . $row["lastname"]. "<br>";
    }
} else {
    echo "0 results";
}
```

```
$conn->close();  
?>
```

Mensajería instantánea (Jabber)

Instalamos openfire:

```
pkg install openfire  
service openfire start  
localhost:9090  
192.168.56.102
```

Ejecutamos Tkabber desde el cliente.

Proxy Cache (Squid): Restricción de contenidos, páginas, usuarios, autenticación LDAP.

Instalamos squid y los configuramos:

```
pkg install squid  
nano /usr/local/etc/squid/squid.conf  
cache_dir ufs /var/squid/cache 100 16 256  
visible_hostname xavi.bsd  
squid -z  
nano /usr/local/etc/squid/deny.txt  
taringa  
/usr/local/etc/squid/squid.conf  
acl expreg-denegadas url_regex "/usr/local/etc/squid/deny.txt"  
http_access allow localnet !expreg-denegadas  
http_access allow localhost  
  
# And finally deny all other access to this proxy  
http_access deny expreg-denegadas  
service squid restart
```

Abrimos la configuración de firefox, proxy manual y ponemos en todas las casillas: 10.0.2.27, 3128

Monitorización de servicios (Nagios)

Descargamos nagios desde el siguiente enlace y procedemos a instalarlo y configurarlo:

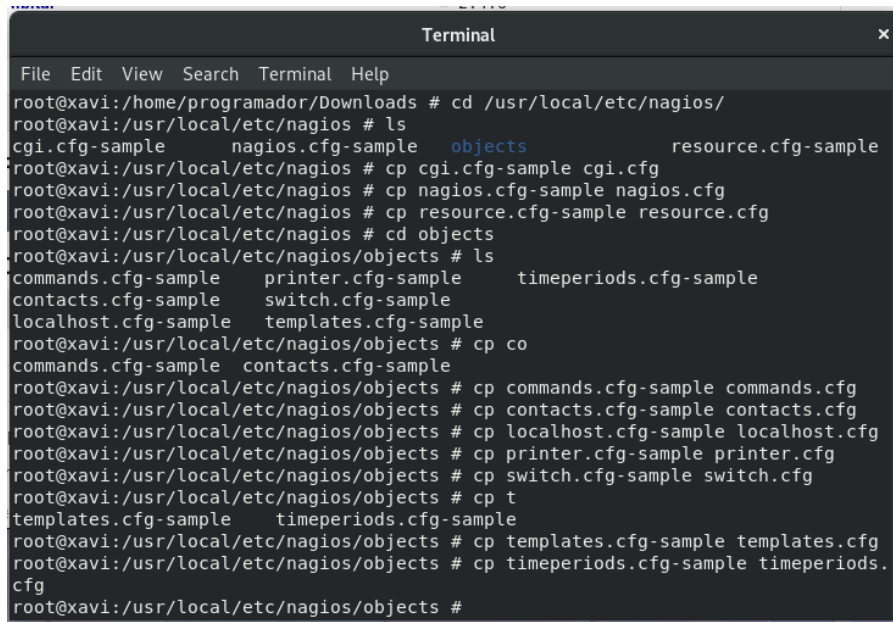
https://freebsd.pkgs.org/12/freebsd-ports-latest-amd64/nagios-3.5.1_11.txz.html

```
nano /usr/local/etc/apache24/httpd.conf  
<Directory /usr/local/www/nagios>  
    Order deny,allow  
    Deny from all  
    Allow from 127.0.0.1  
    php_flag engine on  
    php_admin_value open_basedir  
/usr/local/www/nagios/:/var/spool/nagios/  
</Directory>
```



```
<Directory /usr/local/www/nagios/cgi-bin>
    Options ExecCGI
</Directory>
```

```
ScriptAlias /nagios/cgi-bin/ /usr/local/www/nagios/cgi-bin/
Alias /nagios/ /usr/local/www/nagios/
```



```

Terminal
File Edit View Search Terminal Help
root@xavi:/home/programador/Downloads # cd /usr/local/etc/nagios/
root@xavi:/usr/local/etc/nagios # ls
cgi.cfg-sample      nagios.cfg-sample  objects            resource.cfg-sample
root@xavi:/usr/local/etc/nagios # cp cgi.cfg-sample cgi.cfg
root@xavi:/usr/local/etc/nagios # cp nagios.cfg-sample nagios.cfg
root@xavi:/usr/local/etc/nagios # cp resource.cfg-sample resource.cfg
root@xavi:/usr/local/etc/nagios # cd objects
root@xavi:/usr/local/etc/nagios/objects # ls
commands.cfg-sample  printer.cfg-sample  timeperiods.cfg-sample
contacts.cfg-sample  switch.cfg-sample
localhost.cfg-sample templates.cfg-sample
root@xavi:/usr/local/etc/nagios/objects # cp co
commands.cfg-sample  contacts.cfg-sample
root@xavi:/usr/local/etc/nagios/objects # cp commands.cfg-sample commands.cfg
root@xavi:/usr/local/etc/nagios/objects # cp contacts.cfg-sample contacts.cfg
root@xavi:/usr/local/etc/nagios/objects # cp localhost.cfg-sample localhost.cfg
root@xavi:/usr/local/etc/nagios/objects # cp printer.cfg-sample printer.cfg
root@xavi:/usr/local/etc/nagios/objects # cp switch.cfg-sample switch.cfg
root@xavi:/usr/local/etc/nagios/objects # cp t
templates.cfg-sample  timeperiods.cfg-sample
root@xavi:/usr/local/etc/nagios/objects # cp templates.cfg-sample templates.cfg
root@xavi:/usr/local/etc/nagios/objects # cp timeperiods.cfg-sample timeperiods.
cfg
root@xavi:/usr/local/etc/nagios/objects #

```

```

htpasswd -c /usr/local/etc/nagios/htpasswd.users nagiosadmin
nano /usr/local/etc/apache24/httpd.conf
Directory /usr/local/www/nagios>
    Options None
    AllowOverride None
    Allow from all
    AuthName "Acceso"
    AuthType Basic
    AuthUserFile /usr/local/etc/nagios/htpasswd.users
    Require valid-user
</Directory>

```

```

auth_param basic program
/usr/local/libexec/squid/basic_ncsa_auth/
/usr/local/etc/squid/.htpasswd
auth_param basic children 5
auth_param basic realm Squid Basic Authentication
auth_param basic credentialsttl 5 hours
acl password proxy_auth REQUIRED
http_access allow password

htpasswd -c /usr/local/etc/squid/.htpasswd root
chmod o+r /usr/local/etc/squid/.htpasswd
chmod 777 /usr/local/etc/squid/.htpasswd
chmod o+r /usr/local/libexec/squid/basic_ncsa_auth
chmod 777 /usr/local/libexec/squid/basic_ncsa_auth
service squid restart

```