



## **Presentación**

### **NOMBRE DEL ESTUDIANTE:**

Jamiel Santana Pérez.

### **Matricula:**

20198095

### **Materia:**

SO3.

### **INSTITUTO:**

Instituto tecnológico de las Américas

- **Importancia de los backups:**

- Sin lugar a duda, poder respaldar datos y archivos de suma importancia es esencial para una empresa, sobre todo si esos archivos son críticos para el desarrollo normal de la organización. Los días en los cuales los archivos físicos eran la única opción ya acabaron. Ahora tenemos a oportunidad de hacer esto posible con el uso de tecnologías cloud y con los sistemas de almacenamiento externo local.

- **¿Qué es Báculo?**

- Báculo es una colección de herramientas de respaldo capaz de cubrir las necesidades de respaldo de equipos bajo redes IP. Se basa en una arquitectura Cliente-servidor que resulta eficaz y fácil de manejar, dada la amplia gama de funciones y características que brinda; copiar y restaurar ficheros dañados o perdidos. Además, debido a su desarrollo y estructura modular, Báculo se adapta tanto al uso personal como profesional, desde un equipo hasta grandes parques de servidores.

- **Medios de almacenamiento recomendados para Backups:**

- Los backups son recomendables de guardar en medios externos al dispositivo original, ya sea dispositivos externos como discos duros externos o podemos optar por almacenar nuestro respaldo en la nube, como azure o Google drive, o almacenamiento por la red como puede ser montar un servidor NAS.

- **Proceso de instalación y configuración de báculo como servidor de backups:**

- Antes de iniciar la instalación y configuración, cabe recalcar que el proyecto se estará realizando mediante escritorio remoto, pero esto no afecta en nada en los comandos mostrados en las imágenes.
- Como primer paso empezaremos desactivando Selinux que es, es un módulo de seguridad para el kernel Linux que proporciona el mecanismo para soportar políticas de seguridad para el control de acceso, incluyendo controles de acceso obligatorios como los del Departamento de Defensa de Estados Unidos. Lo desactivaremos dirigiéndonos al siguiente archivo /etc/selinux/config en mi caso lo abriré con el editor de archivos nano.

```
[jam@localhost ~]$ cd /etc/selinux/  
[jam@localhost selinux]$ nano config
```

- En la sección donde nos dice selinux=enable, la cambiaremos a selinux=disabled, y con esto desactivamos selinux, finalmente guardamos los

cambios hechos en el archivo.

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of three values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

- Ahora pasaremos a permitir los puertos en el firewall con los que trabaja b cula estos puertos son el, 9101, 9102, 9103, y lo habilitaremos con el comando **firewall-cmd --add-port={puerto}/protocolo (TCP) --permanent**, el par metro “permanent” se utiliza para que los puertos est n h biles a n se haya reiniciado el sistema, luego para que se efect en los cambios en el firewall insertamos el comando **firewall-cmd --reload**.

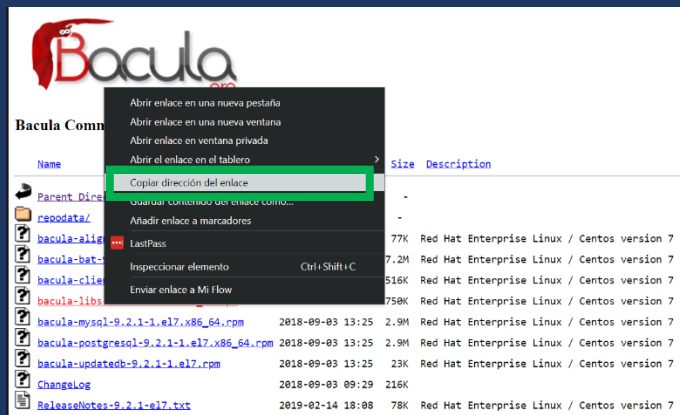
```
[root@localhost selinux]# firewall-cmd --add-port=9101/tcp --permanent
success
[root@localhost selinux]# firewall-cmd --add-port=9102/tcp --permanent
success
[root@localhost selinux]# firewall-cmd --add-port=9103/tcp --permanent
success
[root@localhost selinux]# firewall-cmd --reload
success
```

- Para asegurarnos de que los puertos se agregaron correctamente al firewall utilizamos el comando **firewall-cmd --list-all**, y nos mostrara la lista de puertos.

```
[root@localhost jam]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens33
  sources:
  services: dhcpv6-client ssh
  ports: 9101/tcp 9102/tcp 9103/tcp
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

[root@localhost jam]#
```

- Para que se puedan realizar los cambios que aplicamos para montar debidamente la b cula, debemos reiniciar nuestro servidor esto lo haremos mediante CLI, mediante el comando **systemctl reboot**.
- Una vez reiniciado nuestro servidor, nos vamos a dirigir a los repositorios de b cula que podemos acceder a ellos mediante un peque o registro en su p gina, donde nos enviar n v  correo el link de la web de repositorios, donde podremos escoger los RPM correctos para nuestro SO, en mi caso estar  utilizando los repositorios de Red Hat qu son totalmente compatibles con CentOS 7, dicho esto el primer RPM que descargaremos es el bacula libs, y esto lo realizaremos con el comando **wget “link de enlace del repositorio”**.



Cabe recalcar que la instalación de repositorios es recomendable hacerla en el directorio /root

```
jam@localhost:~
[root@localhost ~]# pwd
/root
[root@localhost ~]# wget https://bacula.org/packages/5ec297c8c3665/rpms/9.2.1/el7/x86_64/bacula-libs-9.2.1-1.el7.x86_64.rpm
```

- Ahora pasaremos con la instalación de bacula mysql, que es la base de datos del servidor bacula

```
root@localhost:~
[root@localhost ~]# wget https://bacula.org/packages/5ec297c8c3665/rpms/9.2.1/el7/x86_64/bacula-mysql-9.2.1-1.el7.x86_64.rpm
--2020-06-09 06:40:58-- https://bacula.org/packages/5ec297c8c3665/rpms/9.2.1/el7/x86_64/bacula-mysql-9.2.1-1.el7.x86_64.rpm
Resolviendo bacula.org (bacula.org)... 80.244.178.6
Conectando con bacula.org (bacula.org)[80.244.178.6]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 3617372 (2,9M) [application/x-redhat-package-manager]
Grabando a: "bacula-mysql-9.2.1-1.el7.x86_64.rpm"

1% [>] 40.960 121KB/s
```

- Luego seguiremos con la instalación del bacula client, que es el repositorio que se instala en los equipos clientes a los que se le hará backup:

```
jam@localhost:~
[root@localhost ~]# pwd
/root
[root@localhost ~]# wget https://bacula.org/packages/5ec297c8c3665/rpms/9.2.1/el7/x86_64/bacula-client-9.2.1-1.el7.x86_64.rpm
--2020-06-09 06:59:34-- https://bacula.org/packages/5ec297c8c3665/rpms/9.2.1/el7/x86_64/bacula-client-9.2.1-1.el7.x86_64.rpm
Resolviendo bacula.org (bacula.org)... 80.244.178.6
Conectando con bacula.org (bacula.org)[80.244.178.6]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 528304 (516K) [application/x-redhat-package-manager]
Grabando a: "bacula-client-9.2.1-1.el7.x86_64.rpm"

100%[=====] 528.304 285KB/s en 1,8s

2020-06-09 06:59:36 (285 KB/s) - "bacula-client-9.2.1-1.el7.x86_64.rpm" guardado [528304/528304]
[root@localhost ~]#
```

- Después de descargar los RPM, ahora tenemos que descomprimir cada uno de los archivos, esto lo haremos con el **comando rpm -Uvh (nombre del RPM)**. El único archivo que descomprimiremos de otra forma es el bacula mysql y esto lo haremos con el **comando yum install (nombre del repositorio)**, debido a que si lo instalamos con rpm, nos pedirá un módulo de perl que no tenemos instalado para evitarnos la búsqueda lo haremos con el comando yum, este a resolver la dependencia por nosotros.

```
[root@localhost ~]# ls
anaconda-ks.cfg      bacula-libs-9.2.1-1.el7.x86_64.rpm  Descargas  Escritorio  initial-setup-ks.cfg  Plantillas  Videos  wget-log.1
bacula-client-9.2.1-1.el7.x86_64.rpm  bacula-mysql-9.2.1-1.el7.x86_64.rpm  Documentos  Imágenes    Música            Publico     wget-log  wget-log.2
[root@localhost ~]# rpm -Uvh Bacula-libs-9.2.1-1.el7.x86_64.rpm
advertencia: bacula-libs-9.2.1-1.el7.x86_64.rpm: EncabezadoV4 RSA/SHA1 Signature, ID de clave e9df3643: NOKEY
Preparando...
Actualizando / instalando...
1: bacula-libs-9.2.1-1.el7 [100%]
```

- Ahora procederemos a instalar el motor de base de datos mariadb, esto lo haremos con el comando **yum install mariadb-server -y**.

```
[root@localhost bacula]# yum install mariadb-server -y
Complementos cargados:fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirror.ufscar.br
 * extras: mirror.ufscar.br
 * updates: mirror.ufscar.br
Resolviendo dependencias
--> Ejecutando prueba de transacción
--> Paquete mariadb-server.x86_64 1:5.5.65-1.el7 debe ser instalado
--> Procesando dependencias: perl-DBD-MySQL para el paquete: 1:mariadb-server-5.5.65-1.el7.x86_64
--> Ejecutando prueba de transacción
--> Paquete perl-DBD-MySQL.x86_64 4:4.023-6.el7 debe ser instalado
--> Resolución de dependencias finalizada

Dependencias resueltas

=====
Package                                Arquitectura Versión                                Repositorio
=====
Instalando:
mariadb-server                        x86_64      1:5.5.65-1.el7                          base
Instalando para las dependencias:
perl-DBD-MySQL                        x86_64      4.023-6.el7                              base

Resumen de la transacción
=====
Instalar: 1 Paquete (+1 Paquete dependiente)

Tamaño total de la descarga: 11 M
Tamaño instalado: 59 M
Downloading packages:
(1/2): perl-DBD-MySQL-4.023-6.el7.x86_64.rpm          140 kB 00:00
(2/2): mariadb-server-5.5.65-1.el7.x86_64.rpm        83 kB/s | 4.5 MB 00:01
39% [=====]
```

- Vamos a activar el motor de base de datos mariadb con el comando **systemctl enable mariadb** y posteriormente **systemctl start mariadb**.

```
[root@localhost bacula]# systemctl enable mariadb
```

```
[[root@localhost bacula]# systemctl start mariadb
```

- Como siguiente paso ejecutaremos el script de acondicionamiento, **mysql\_secure\_installation**, donde una vez insertado el comando nos pedirá una password y como no tenemos solo le daremos a ENTER, luego nos preguntara para sestionarla pass y crear una y le diremos que sí, así podremos asignarle una password al mysql, nos pregunta igualmente si queremos remover el user Anónimo por razones de seguridad le diremos que sí, aunque no se muestre en la imagen nos preguntara que si queremos deshabilitar el acceso remoto le diremos que no, cuando nos pregunte sobre remover la base de datos de prueba le diremos que sí, luego le diremos que si para que haga una recarga de los privilegios.
- Ahora nos iremos al directorio donde se encuentran los scripts de creación de base de datos, tablas y acondicionamiento de permisos de bacula. Esto lo haremos dirigiéndonos a la ruta **cd /opt/bacula/scripts/**, pero antes de utilizar los scripts de este directorio, crearemos la base dedatos y las tablas de mysql y esto se hace con los comando **./create\_mysql\_database -u root -p** y **./make\_mysql\_tables -u root -p** donde los parámetros -u es para el user que creamos anteriormente en el reacondicionamiento de

la base de datos y -p para que nos pida la pass del user recién creado.

```
[root@localhost ~]# cd /opt/bacula/scripts/
```

```
[root@localhost scripts]# ./make_mysql_tables -u root -p
```

```
[root@bacula scripts]# ./create_mysql_database -u root -p
```

- Y por último ejecutaremos el **./grant\_mysql\_privileges -u root -p**

```
[root@bacula scripts]# ./grant_mysql_privileges -u root -p
```

- Al ver la siguiente salida significa que salió todo a la perfección >

```
Created MySQL database user: bacula
Enter password:
Host: User Password Select_priv Insert_priv Update_priv Delete_priv Create_priv Drop_priv Reload_priv Shutdown_priv Pro
File_priv Grant_priv References_priv Index_priv Alter_priv Show_db_priv Super_priv Create_tmp_table_priv Lock_tables_priv Exe
Repl_slave_priv Repl_client_priv Create_view_priv Show_view_priv Create_routine_priv Alter_routine_priv Create_user_priv Event_priv
ger_priv Create_tablespace_priv ssl_type ssl_cipher x509_issuer x509_subject max_questions max_updates max_connections max_user_co
plugin authentication_string
localhost root *23AE809DDACAF96AF0FD78ED04B6A265E05AA257 Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y
Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y
localhost.localdomain root *23AE809DDACAF96AF0FD78ED04B6A265E05AA257 Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y
Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y
0
127.0.0.1 root *23AE809DDACAF96AF0FD78ED04B6A265E05AA257 Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y
Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y
:::1 root *23AE809DDACAF96AF0FD78ED04B6A265E05AA257 Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y
Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y
localhost bacula N N N N N N N N N N N N N N N N N N N N N N N N N N N N
N N N N N N N N N N N N N N N N N N N N N N N N N N N N
% bacula N N N N N N N N N N N N N N N N N N N N N N N N N N N N
N N N N N N N N N N N N N N N N N N N N N N N N N N N N
Privileges for user bacula granted on database bacula.
```

- Ahora crearemos un link simbólico para poder ejecutar los comandos de báculo esto lo haremos con el comando **ln -s /opt/bacula/bin/\* /usr/bin**, hecho esto podremos llamar a báculo desde cualquier parte de nuestro servidor.

```
[root@bacula bin]# ln -s /opt/bacula/bin/* /usr/bin/
```

```
[root@bacula bin]# cd
```

```
[root@bacula ~]# bacula
```

- Ahora podremos ejecutar el motor de backups bacula y esto se hace con el comando **bacula start**. Esto hará que enciendas los tres motores, el storage daemon, el file Daemon y el Dir Daemon, para verificar si todo ha funcionado insertamos el comando **bacula status** para ver cómo van los procesos

```
root@localhost:~# cd /opt/bacula/scripts
```

```
[root@localhost scripts]# bacula start
Starting the Bacula Storage daemon
Starting the Bacula File daemon
Starting the Bacula Director daemon
[root@localhost scripts]# bacula status
bacula-sd (pid 4923) is running...
bacula-fd (pid 4932) is running...
bacula-dir (pid 4940) is running...
[root@localhost scripts]#
```

```
[root@localhost scripts]# bacula status
bacula-sd (pid 4923) is running...
bacula-fd (pid 4932) is running...
bacula-dir (pid 4940) is running...
[root@localhost scripts]#
```

- Vamos a dirigirnos al directorio **/opt/bacula/etc/** nos encontraremos con cuatro archivos de configuración de bacula los cuales editaremos, con el objetivo de en cada parámetro de password asignarle una contraseña apropiada a nuestro servidor, empezaremos editando el archivo **bacula-dir**, luego seguiremos con el **bacula-sd.conf**, luego el **bacula-fd** y por



último el bconsole.conf e iremos buscando apartados como el resaltado en la imagen.

```
root@bacula:/opt/bacula/etc
#
# Default Bacula Director Configuration file
#
# The only thing that MUST be changed is to add one or more
# file or directory names in the Include directive of the
# FileSet resource.
#
# For Bacula release 9.2.1 (12 August 2018) -- redhat Enterprise release
#
# You might also want to change the default email address
# from root to your address. See the "mail" and "operator"
# directives in the Messages resource.
#
# Copyright (C) 2000-2017 Kern Sibbald
# License: BSD 2-Clause; see file LICENSE-FOSS
#
Director {
    # define myself
    Name = bacula-dir
    DIRport = 9101 # where we listen for UA connections
    QueryFile = "/opt/bacula/scripts/query.sql"
    WorkingDirectory = "/opt/bacula/working"
    PidDirectory = "/opt/bacula/working"
    MaximumConcurrentJobs = 20
    Password = "8/dsVnUcpgm0d5d12af1WxL5ppFZlFe9r5hi/ALFH4t" # Console password
    Messages = Daemon
}

JobDefs {
    Name = "DefaultJob"
    Type = Backup
    Level = Incremental
    Client = bacula-fd
    FileSet = "Full Set"
    Schedule = "WeeklyCycle"
}
```

- Como siguiente paso nos vamos a dirigir al archivo /etc/hosts y agregaremos la ip de lookup de nuestro servidor la cual es 127.0.0.1, y agregaremos los nombres por los cuales podremos llamar al servidor en este caso "bacula y bacula.example.com".

```
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
127.0.0.1    bacula bacula.example.com
```

- Ahora si podremos hacer ping al servidor con el nombre que le asignamos.

```
[root@bacula etc]# ping bacula
ping: bacula: Nombre o servicio desconocido
[root@bacula etc]# ping bacula.example.com
PING bacula.example.com (10.20.30.200) 56(84) bytes of data:
64 bytes from bacula.example.com (10.20.30.200): icmp_seq=1 ttl=64 time=0.129 ms
64 bytes from bacula.example.com (10.20.30.200): icmp_seq=2 ttl=64 time=0.154 ms
64 bytes from bacula.example.com (10.20.30.200): icmp_seq=3 ttl=64 time=0.092 ms
64 bytes from bacula.example.com (10.20.30.200): icmp_seq=4 ttl=64 time=0.098 ms
^C
--- bacula.example.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 0.092/0.118/0.154/0.026 ms
[root@bacula etc]# vim /etc/hosts
[root@bacula etc]# ping bacula
PING bacula (127.0.0.1) 56(84) bytes of data:
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.111 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.146 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.104 ms
64 bytes from localhost (127.0.0.1): icmp_seq=4 ttl=64 time=0.081 ms
^C
--- bacula ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3021ms
rtt min/avg/max/mdev = 0.081/0.110/0.146/0.025 ms
```

- El paso siguiente sería asignarle un passWord al usuario de base de datos bacula, por defecto no trae, pero nosotros como buena práctica le asignaremos una. Para ello entraremos a la base de datos con el comando mysql -u root -p, una vez allí adentro crearemos el user con la password esto se hace con el comando **update mysql.user set password=password('la pass') where user= 'nombre del user'**. **Para que aplique todos los cambios**

## insertamos el comando flush privileges.

```
root@localhost:~# cd /opt/bacula/etc
root@localhost:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 16
Server version: 5.5.65-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> update mysql.user password=password('bacula') where user='bacula';
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near '=password('bacula')' at line 1
MariaDB [(none)]> update mysql.user password=password('bacula') where user='bacula';
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near '=password('bacula')' at line 1
MariaDB [(none)]> update mysql.user set password=password('bacula') where user='bacula';
Query OK, 2 rows affected (0.00 sec)
Rows matched: 2  Changed: 2  Warnings: 0

MariaDB [(none)]>
```

- Tendremos que ir a editar el archivo bacula-dir.conf, y buscamos el apartado de configuración del catálogo, que está es la base de datos donde báculo va guardando todos los jobs que hizo de backups y de restore, para que ella sepa donde guarda las cosas, editaremos la parte de dbpassword y entre comillas dobles le pondremos la pass que le asignamos al user que acabamos de crear.

```
# Generic catalog service
Catalog {
    Name = MyCatalog
    dbname = "bacula"; dbuser = "bacula"; dbpassword = "bacula"
}
```

- Posterior a esto daremos una báculo restart para reiniciar los servicios de báculo.
- Ahora haremos la primera conexión al motor de backups mediante el comando bconsole, si todo está bien configurado, nos saldrá una salida que dice 10000 ok que significa que nos conectamos, si tenemos el prompt del \* significa que estamos conectados, si damos un help, podremos ver la cantidad de opciones que podemos hacer desde el bconsole, desde agregar, listar...
- Ahora vamos a instalar una interfaz grafica de bacula con webmin para esto emepzaremos descargando el servidor apache esto se hace con el comando yum install httpd -y.



```

root@localhost/opt/bacula/etc
Loading mirror speeds from cached hostfile
* base: mirrors.upr.edu
* extras: mirrors.upr.edu
* updates: mirrors.upr.edu
Resolviendo dependencias
--> Ejecutando prueba de transacción
--> Paquete httpd.x86_64 0:2.4.6-93.el7.centos debe ser instalado
--> Procesando dependencias: httpd-tools = 2.4.6-93.el7.centos para el paquete: httpd-2.4.6-93.el7.centos.x86_64
--> Procesando dependencias: /etc/mime.types para el paquete: httpd-2.4.6-93.el7.centos.x86_64
--> Procesando dependencias: libaprutil-1.so.0()(64bit) para el paquete: httpd-2.4.6-93.el7.centos.x86_64
--> Ejecutando prueba de transacción
--> Paquete apr.x86_64 0:1.4.8-5.el7 debe ser instalado
--> Paquete apr-util.x86_64 0:1.5.2-6.el7 debe ser instalado
--> Paquete httpd-tools.x86_64 0:2.4.6-93.el7.centos debe ser instalado
--> Paquete mailcap.noarch 0:2.1.41-2.el7 debe ser instalado
--> Resolución de dependencias finalizada

Dependencias resueltas

=====
Package                Arquitectura      Versión              Repositorio          Tamaño
=====
Instalando:
httpd                   x86_64            2.4.6-93.el7.centos  base                  2.7 M
Instalando para las dependencias:
apr                     x86_64            1.4.8-5.el7          base                  103 k
apr-util                x86_64            1.5.2-6.el7          base                   92 k
httpd-tools             x86_64            2.4.6-93.el7.centos base                   92 k
mailcap                 noarch            2.1.41-2.el7         base                   31 k

Resumen de la transacción
=====
Instalar 1 Paquete (+4 Paquetes dependientes)

Tamaño total de la descarga: 3.0 M
Tamaño instalado: 10 M
Downloading packages:
(1/5): apr-1.4.8-5.el7.x86_64.rpm | 103 kB 00:00:00
(2/5): mailcap-2.1.41-2.el7.noarch.rpm | 31 kB 00:00:00
(3/5): httpd-tools-2.4.6-93.el7.centos.x86_64.rpm | 92 kB 00:00:01
(4/5): apr-util-1.5.2-6.el7.x86_64.rpm | 110 kB/s 307 kB 00:00:24 ETA
12% [=====]

```

- Una vez instalado lo habilitaremos con el comando `systemctl enable httpd` y para iniciarlo `systemctl start httpd`.

```

[root@localhost etc]#
[root@localhost etc]# systemctl enable httpd
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
[root@localhost etc]# systemctl start httpd
[root@localhost etc]#

```

- Ahora vamos a habilitar el puerto del firewall por donde se maneja webmin, con el comando `firewall-cmd --add-service=http --permanent`, aunque podríamos agregar el puerto tal y como lo hicimos con bacula preferimos no apegarnos a un solo método, ejecutaremos el mismo comando nuevamente, pero en vez de `http` pondremos `https` ya que webmin trabaja sobre este último. Por último reiniciamos el firewall para aplicar cambios

```

root@localhost/opt/bacula/etc

[root@localhost etc]# firewall-cmd --add-service=http --permanent
success
[root@localhost etc]# firewall-cmd --add-service=https --permanent
success
[root@localhost etc]# firewall-cmd --reload
success
[root@localhost etc]#

```

- Ahora procederemos a la descarga del webmin, escribiendo en el buscador de preferencia `webmin.com`, una vez allí buscamos la opción de RPM y descargamos el repositorio igual como lo hicimos con los rpm de bacula

```
[root@localhost etc]# pwd
/opt/bacula/etc
[root@localhost etc]# cd /root/
[root@localhost ~]# wget https://downloads.sourceforge.net/project/webadmin/webmin/1.941/webmin-1.941-1.noarch.rpm?r=https%3A%2F%2Fsourceforge.net%2Fprojects%2Fwebadmin%2Ffiles%2Fwebmin%2F1.941%2Fwebmin-1.941-1.noarch.rpm%2Fdownload%3Fuse_mirror%3Diweb&ts=1591726500
[1] 6147
[root@localhost ~]# --2020-06-09 20:16:03-- https://downloads.sourceforge.net/project/webadmin/webmin/1.941/webmin-1.941-1.noarch.rpm?r=https%3A%2F%2Fsourceforge.net%2Fprojects%2Fwebadmin%2Ffiles%2Fwebmin%2F1.941%2Fwebmin-1.941-1.noarch.rpm%2Fdownload%3Fuse_mirror%3Diweb
Resolviendo downloads.sourceforge.net (downloads.sourceforge.net)... 216.105.38.13
Conectando con downloads.sourceforge.net (downloads.sourceforge.net)[216.105.38.13]:443... conectado.
Petición HTTP enviada, esperando respuesta... 302 Found
Localización: https://cfhcable.dl.sourceforge.net/project/webadmin/webmin/1.941/webmin-1.941-1.noarch.rpm [siguiendo]
--2020-06-09 20:16:05-- https://cfhcable.dl.sourceforge.net/project/webadmin/webmin/1.941/webmin-1.941-1.noarch.rpm
Resolviendo cfhcable.dl.sourceforge.net (cfhcable.dl.sourceforge.net)... 146.71.73.6
Conectando con cfhcable.dl.sourceforge.net (cfhcable.dl.sourceforge.net)[146.71.73.6]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 23204616 (22M) [application/octet-stream]
Grabando a: "webmin-1.941-1.noarch.rpm?r=https%2F%2Fsourceforge.net%2Fprojects%2Fwebadmin%2Ffiles%2Fwebmin%2F1.941%2Fwebmin-1.941-1.noarch.rpm%2Fdownload%3fuse_mirror=iweb"
100%[=====] 23.204.616 1,16MB/s en 54s

2020-06-09 20:17:00 (416 KB/s) - "webmin-1.941-1.noarch.rpm?r=https%2F%2Fsourceforge.net%2Fprojects%2Fwebadmin%2Ffiles%2Fwebmin%2F1.941%2Fwebmin-1.941-1.noarch.rpm%2Fdownload%3fuse_mirror=iweb" guardado [23204616/23204616]

[1]+  Hecho                  wget https://downloads.sourceforge.net/project/webadmin/webmin/1.941/webmin-1.941-1.noarch.rpm?r=https%3A%2F%2Fsourceforge.net%2Fprojects%2Fwebadmin%2Ffiles%2Fwebmin%2F1.941%2Fwebmin-1.941-1.noarch.rpm%2Fdownload%3Fuse_mirror%3Diweb
[root@localhost ~]#
```

- Si damos un ls donde se descargo el webmin veremos que el archivo que se descargo contiene un nombre muy largo, le cambiaremos ese nombre con el comando **mv “nombre del archivo” “nombre que le queremos dar”**

```
[root@localhost ~]# ls
anaconda-ks.cfg
bacula-client-9.2.1-1.el7.x86_64.rpm
bacula-lb-9.2.1-1.el7.x86_64.rpm
bacula-mysql-9.2.1-1.el7.x86_64.rpm
Descargas
Documentos
Escritorio
Imágenes
initial-setup-ks.cfg
Música
Plantillas
Público
webmin-1.941-1.noarch.rpm?r=https%2F%2Fsourceforge.net%2Fprojects%2Fwebadmin%2Ffiles%2Fwebmin%2F1.941%2Fwebmin-1.941-1.noarch.rpm%2Fdownload%3fuse_mirror=iweb
wget-log
wget-log.1
wget-log.2
[root@localhost ~]# mv webmin-1.941-1.noarch.rpm?r=https%2F%2Fsourceforge.net%2Fprojects%2Fwebadmin%2Ffiles%2Fwebmin%2F1.941%2Fwebmin-1.941-1.noarch.rpm%2Fdownload%3fuse_mirror=iweb webmin-1.941-1.noarch.rpm
[root@localhost ~]# ls
anaconda-ks.cfg      bacula-mysql-9.2.1-1.el7.x86_64.rpm  Escritorio  Música  Plantillas  webmin-1.941-1.noarch.rpm  wget-log.1
bacula-client-9.2.1-1.el7.x86_64.rpm  Descargas  Imágenes  Plantillas  webmin-1.941-1.noarch.rpm  wget-log.2
bacula-lb-9.2.1-1.el7.x86_64.rpm      Documentos  initial-setup-ks.cfg  Público  wget-log
[root@localhost ~]#
```

- Después de esto insertaremos webmin con el comando: **yum install “el nombre del archivo webmin.”**

```
webmin                               noarch                               1.941-1                               /webmin-1.941-1.noarch
Instalando para las dependencias:
perl-Encode-Detect                   x86_64                               1.01-13.el7                          base
perl-Net-SSLeay                      x86_64                               1.55-6.el7                            base

Resumen de la transacción
=====
Instalar 1 Paquete (+2 Paquetes dependientes)

Tamaño total: 79 M
Tamaño total de la descarga: 368 k
Tamaño instalado: 80 M
Is this ok [y/d/N]: y
Downloading packages:
(1/2): perl-Encode-Detect-1.01-13.el7.x86_64.rpm | 82 kB 0
(2/2): perl-Net-SSLeay-1.55-6.el7.x86_64.rpm | 285 kB 0
-----
Total                                                                    200 kB/s | 368 kB 0
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Instalando : perl-Net-SSLeay-1.55-6.el7.x86_64
  Instalando : perl-Encode-Detect-1.01-13.el7.x86_64
Operating system is CentOS Linux
  Instalando : webmin-1.941-1.noarch

Webmin install complete. You can now login to https://localhost.localdomain:10000/
as root with your root password.
Comprobando : webmin-1.941-1.noarch
Comprobando : perl-Encode-Detect-1.01-13.el7.x86_64
Comprobando : perl-Net-SSLeay-1.55-6.el7.x86_64

Instalado:
webmin.noarch 0:1.941-1

Dependencia(s) instalada(s):
perl-Encode-Detect.x86_64 0:1.01-13.el7                                perl-Net-SSLeay.x86_64 0:1.55-6.el7

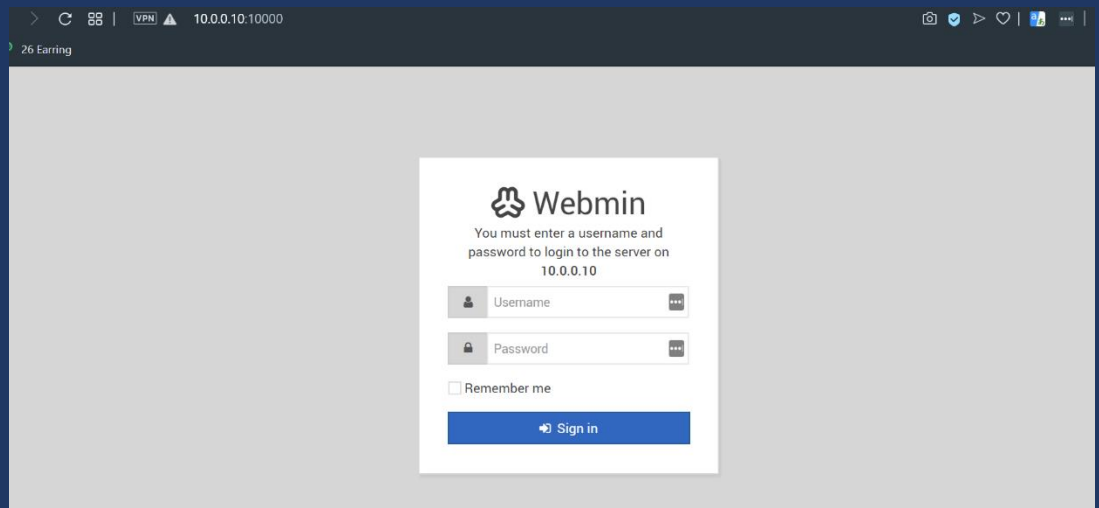
¡Listo!
[root@localhost ~]#
[root@localhost ~]#
```

- Ahora habilitaremos webmin para que inicie con el sistema con el comando **systemctl enable webmin**, pero no podremos insertar el comando **systemctl start webmin** sin antes reiniciar el sistema ya que es un requisito de webmin.

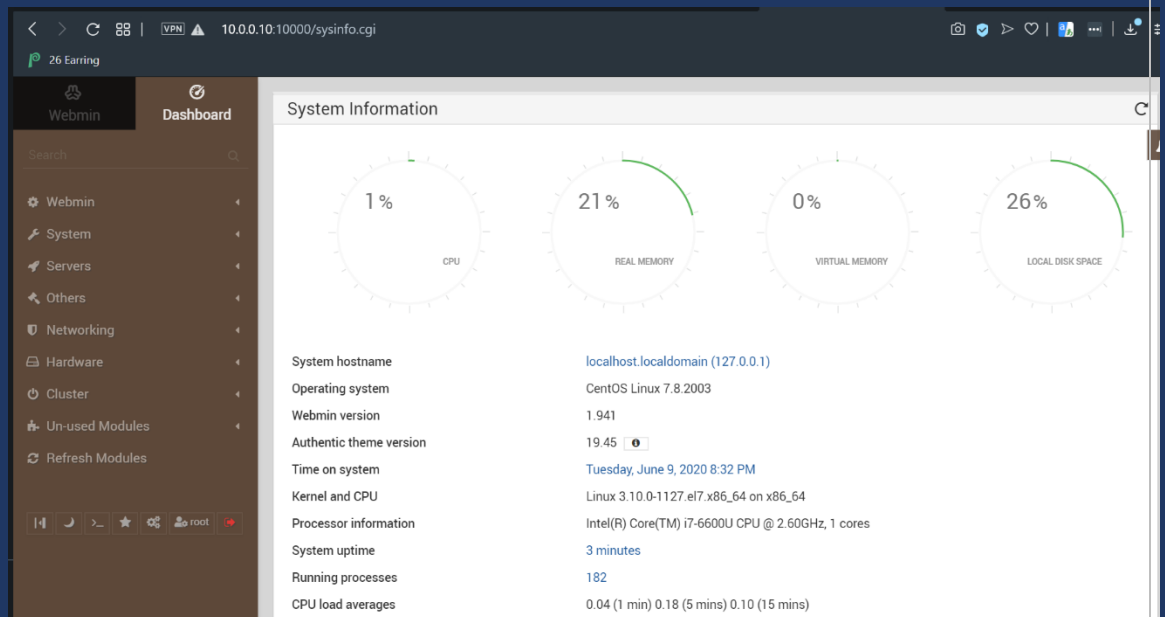
- Pero antes de reiniciar podemos habilitar el puerto por donde trabaja webmin que es el 10000 y esto se hace con el comando **firewall-cmd --add-port=10000/tcp --permanent** y reiniciamos el firewall con el comando **firewall-cmd --reload**. Ahora si procederemos a reiniciar el sistema con el comando **systemctl reboot**

```
[root@localhost ~]# systemctl enable webmin
webmin.service is not a native service, redirecting to /sbin/chkconfig.
Executing /sbin/chkconfig webmin on
[root@localhost ~]# systemctl start webmin
Job for webmin.service failed because the control process exited with error code. See "systemctl status webmin.service" and "journalctl -xe" for details.
[root@localhost ~]# firewall-cmd --add-port=10000/tcp --permanent
success
[root@localhost ~]# firewall-cmd --reload
success
[root@localhost ~]# systemctl reboot
bash: systemctl: no se encontró la orden...
[root@localhost ~]# systemctl reboot
```

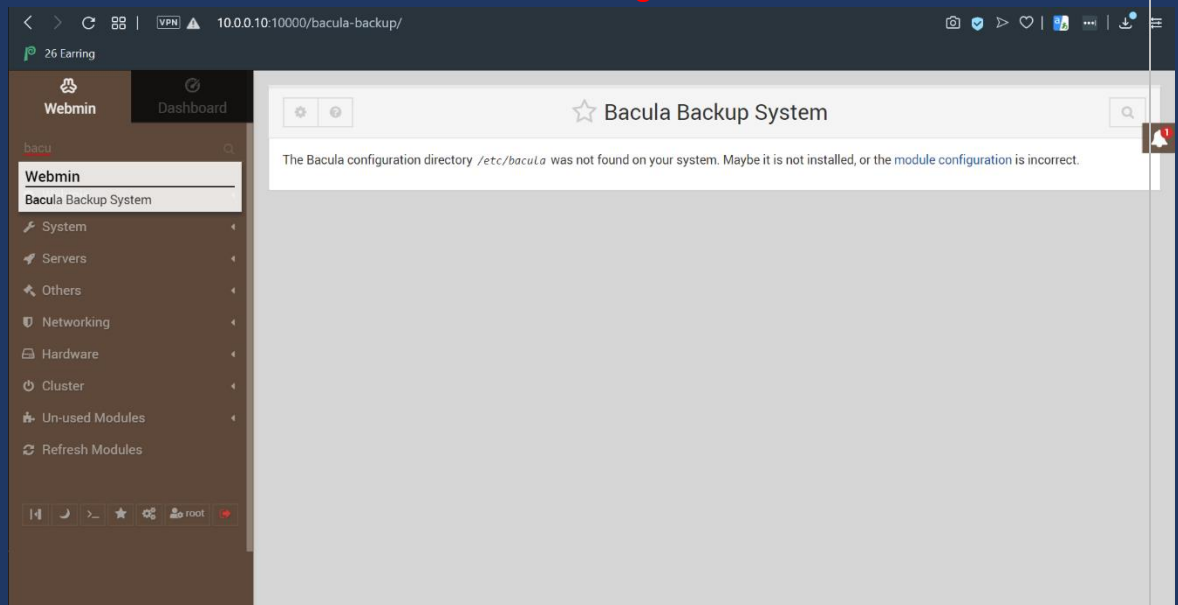
- Una vez reiniciada la maquina sin darle un systemctl start a webmin si insertamos el comando systemctl status webmin veremos que el webmin esta corriendo.
- Hecho todo esto nos podemos dirigir a la consola de administración grafica, para ello vamos a un buscador de preferencia y escribimos <http://direccion-ip-del-servidor:10000>



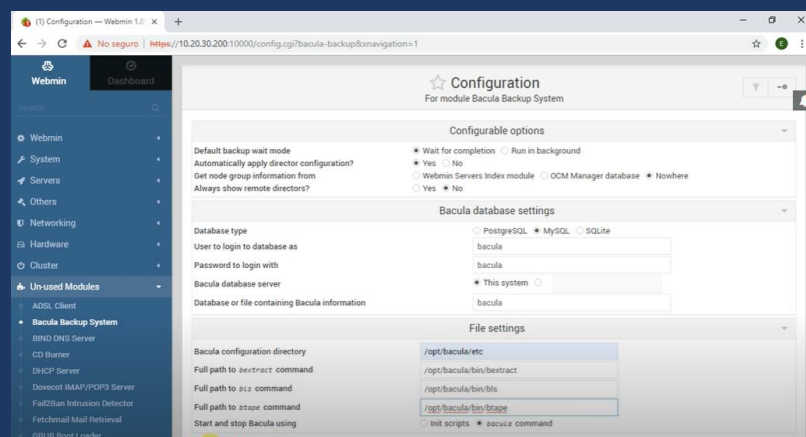
- Para ver nuestro módulo de bacula tendremos que ingresarlo manualmente ya que si webmin no lo detecta, no lo mostrara, y para ello iremos al buscador y pondremos bacula backup system, que viene con webmin



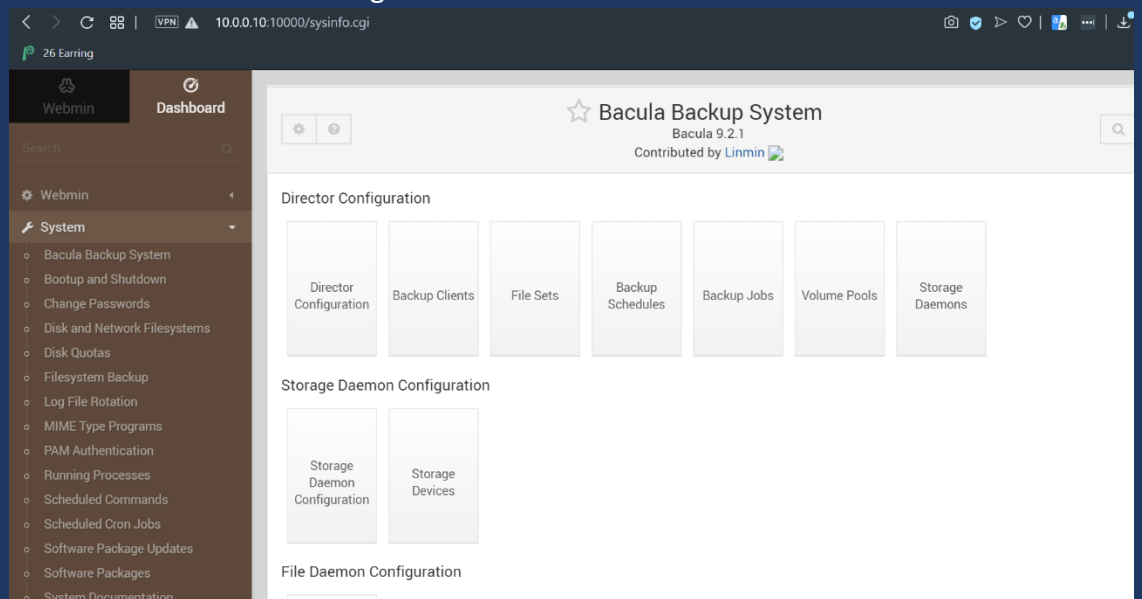
- Ahora le daremos click donde dice **module configuration**.



- Dode ese apartado lo configuraremos de la siguiente manera:



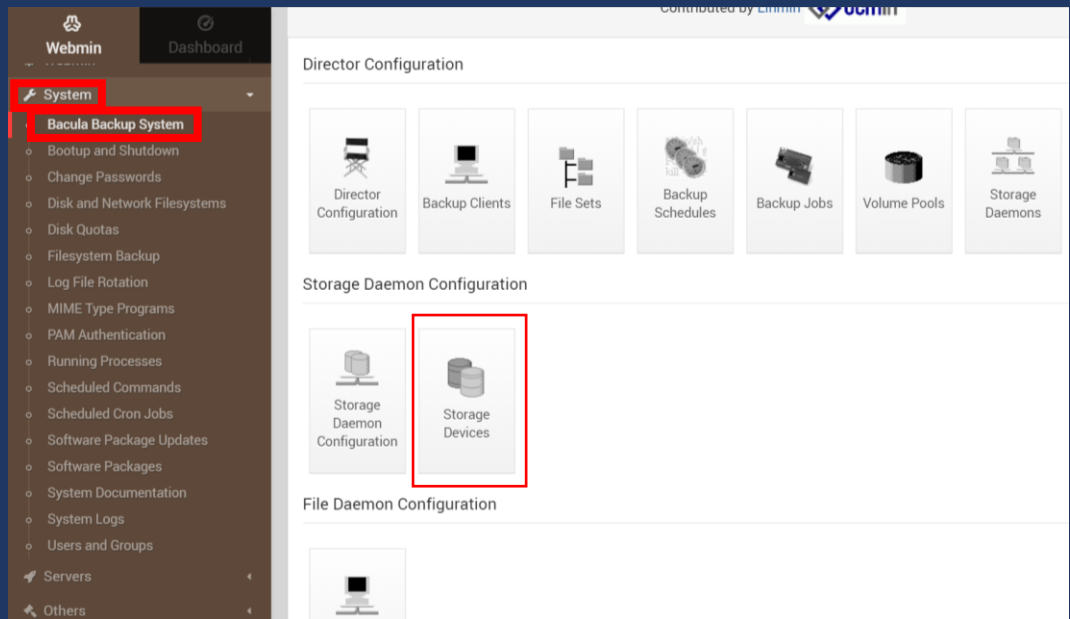
- Una vez insertados los parámetros, le daremos a guardar y veremos como nos trae la ventana de configuración bacula



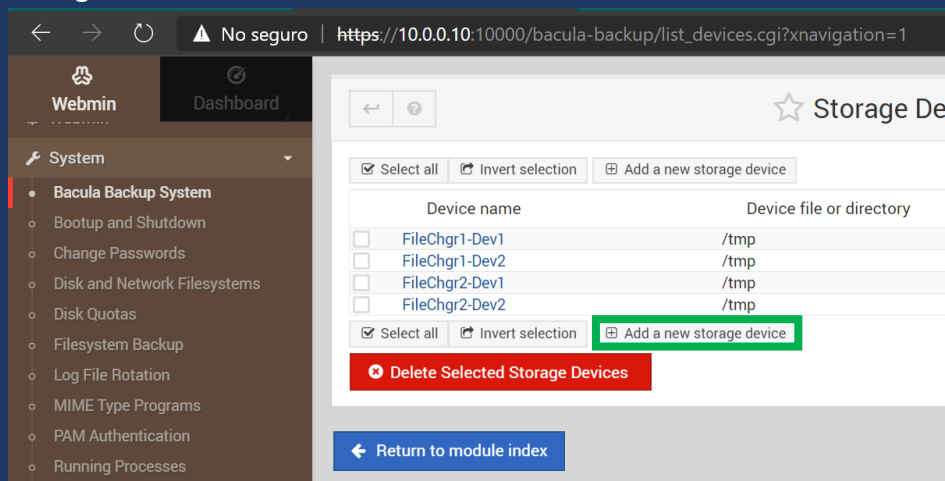
- Como realizar backups a clientes Windows:
  - Como primer paso debes crear y configurar el espacio de almacenamiento local, que es la parte donde se irán guardando los backups que se les realizarán a las máquinas cliente. Para esto deberemos crear en el servidor báculo, la carpeta donde se guardarán los backups, la cual tendrá como ruta /opt/bacula/ (NOMBRE DEL DIRECTORIO), posterior a esto le daremos permisos de privilegios al usuario bacula que creamos en pasos anteriores.

```
[jam@localhost ~]$ cd /opt/bacula/
[jam@localhost bacula]$ ls
bin  etc  lib64  plugins  scripts  share  working
[jam@localhost bacula]$ su
Contraseña:
[root@localhost bacula]# mkdir backups
[root@localhost bacula]# ls
backups  bin  etc  lib64  plugins  scripts  share  working
[root@localhost bacula]# chown bacula:bacula backups/
[root@localhost bacula]#
```

- Después de esto vamos a la web de administración de bacula, y en la parte izquierda en el menú, nos dirigimos a **system > Bacula backup system**, que es prácticamente la consola donde nos manejaremos mayormente con báculo, una vez allí vamos a crear un **storage device**.



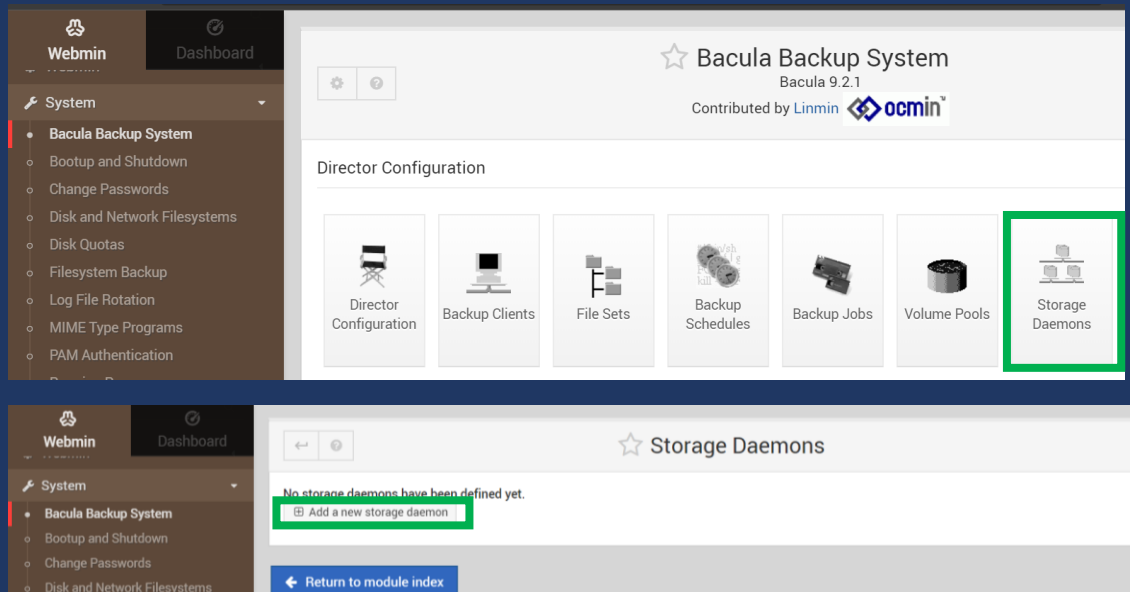
- Cuando nos encontremos en storage device, veremos varios storage que vienen preconfigurados en la b cula, los cuales no utilizaremos, mejor crearemos el propio nuestro, esto se hace dando click en la opci n add a new storage device.



- Hecho click en la opci n marcada en la imagen anterior, una vez dentro del apartado, le daremos un nombre al **storage device**, y en la casilla de **archive device or directory** pondremos la ruta donde creamos el Directorio que ser  **/opt/bacula/backups** y en la casilla de media type name pondremos el mismo nombre de storage device. Hecho est de damos a create y podemos retornar a la paina anterior.

| Details of file storage device                                   |   |                            |   |
|--|---|----------------------------|---|
| Storage device name  | file-back   |                            |   |
| Archive device or directory                                      | /opt/bacula/backups   |                            |   |
| Media type name  | File  | Automatically label media? | <input checked="" type="radio"/> Yes <input type="radio"/> No <input type="radio"/> Default |
| Random access medium?  | <input checked="" type="radio"/> Yes <input type="radio"/> No <input type="radio"/> Default | Mount automatically?       | <input checked="" type="radio"/> Yes <input type="radio"/> No <input type="radio"/> Default |
| Removable media?   | <input type="radio"/> Yes <input checked="" type="radio"/> No <input type="radio"/> Default | Always keep open?          | <input type="radio"/> Yes <input checked="" type="radio"/> No <input type="radio"/> Default |
| <input type="button" value="Create"/>                            |   |                            |   |
| <input type="button" value="Return to list of storage devices"/> |   |                            |   |

- Ahora procederemos a crearnos un storage daemons, para ello vamos a la opción mencionada y marcamos la opción de agregar un storage daemons.



- Una vez allí crearemos configuraremos el nombre del Daemon, su clave, su dirección ip, que en caso de no tener un servidor DNS, tendremos que poner la dirección ip de nuestro servidor bácula, en el apartado de storage device vamos a elegir el que acabamos de crear y en la media type pondremos el mismo nombre que el storage device. Luego de esto hacemos click en crear, si queremos podemos mejorar la seguridad de nuestro backups modificando las opciones aparecidas en la parte baja de la imagen, pero para fines de este Lab dejaremos todo como default.

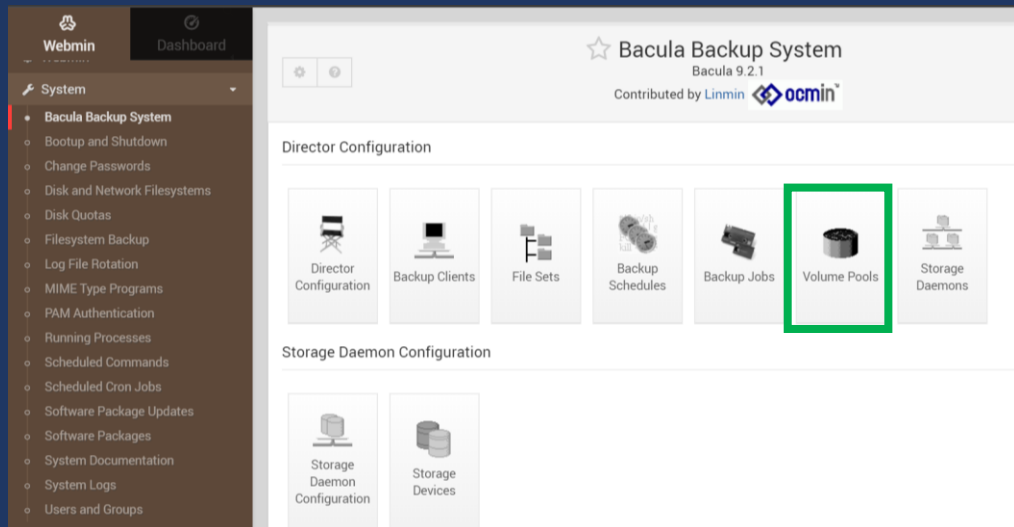
The screenshot shows the 'Create Storage Daemon' form. The title is 'Details of remote storage daemon'. The form fields are:

- Storage daemon name: File-sd
- Bacula SD password: [masked]
- Hostname or IP address: 10.0.0.10
- Bacula SD port: 9103
- Storage device name: file-back
- Media type name: file-back
- Maximum concurrent jobs: 20

Below the form, there are checkboxes for TLS encryption and TLS certificates, and a 'Create' button.



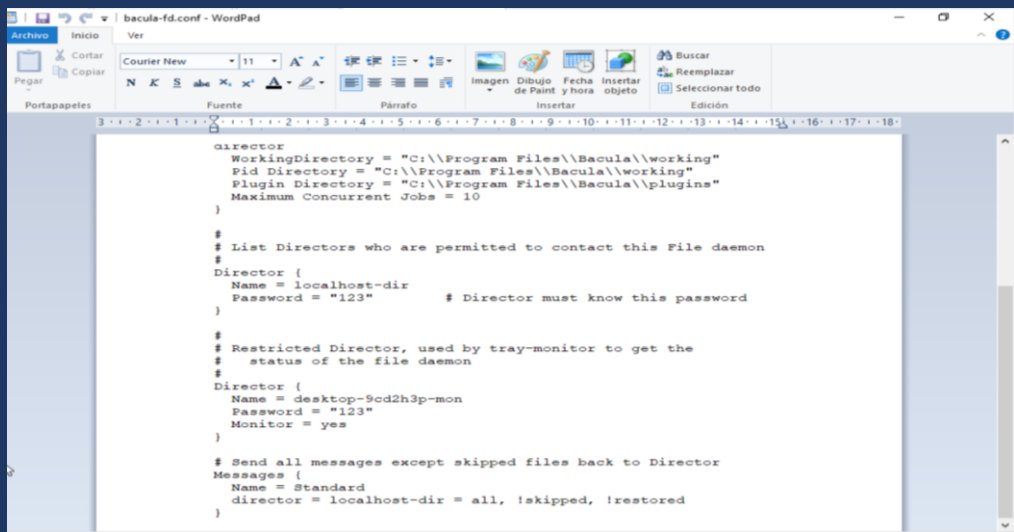
- Ahora nos dirigimos a la ventana principal de nuestra b cula web e iremos al apartado de **volumen pools**.



- Crearemos un nuevo pool:
  - Volume pool name: nombre que le daremos al pool.
  - Volume pool type: para que servir  el pool.
  - Volume retention period: periodo de retencion de los archivos.
  - Prune expired volumes: habilitada la opcion de eliminar volúmenes expirados.
  - Automatically recycle volumes: automatizar el reciclaje de volúmenes.
  - Maximum volume size: cantidad de gb maximos que puede almacenar un archivo backup.
- Hecho todo esto le damos a crear y tendremos nuestro servidor listo, para almacenar los backups.

- Ahora vamos a dirigirnos al lado cliente para instalar los binarios necesarios, ya que el backup se har  en un cliente Windows, los binarios se descargaron de este link <https://www.bacula.org/binary-download-center/> donde escogeremos el binario seg n nuestro SO y la versi n del b cula.
- Una vez instalado, tendremos que dirigirnos a la carpeta de b cula y abrir el archivo que dice edit cliente configuration, donde en el apartado de name pondremos el nombre del director de nuestro bacula esto lo encontraremos

en la sección de director configuration, ya obtenido y puesto el nombre le pondremos una pass de nuestro gusto. y así lo haremos el otro apartado que se ubica un poco más abajo.



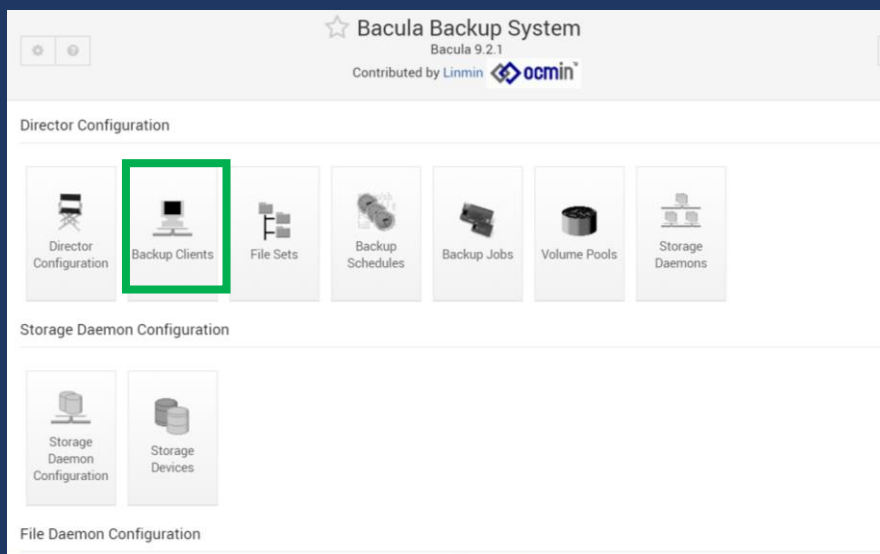
```
director
WorkingDirectory = "C:\\Program Files\\Bacula\\working"
Pid Directory = "C:\\Program Files\\Bacula\\working"
Plugin Directory = "C:\\Program Files\\Bacula\\plugins"
Maximum Concurrent Jobs = 10
}

# List Directors who are permitted to contact this File daemon
#
Director {
  Name = localhost-dir
  Password = "123"      # Director must know this password
}

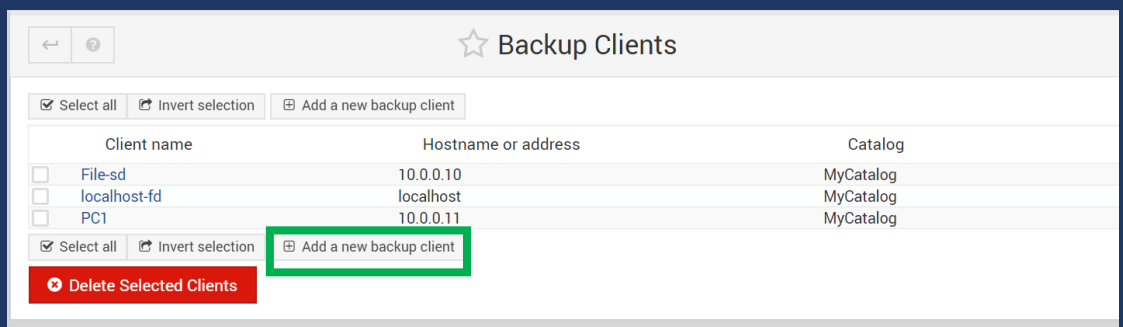
# Restricted Director, used by tray-monitor to get the
# status of the file daemon
#
Director {
  Name = desktop-9cd2h3p-mon
  Password = "123"
  Monitor = yes
}

# Send all messages except skipped files back to Director
Messages {
  Name = Standard
  director = localhost-dir = all, !skipped, !restored
}
```

- Realizado esto, podremos dirigirnos a nuestro servidor, para agregar a nuestro cliente, y esto lo haremos en la sección de **backups client**.



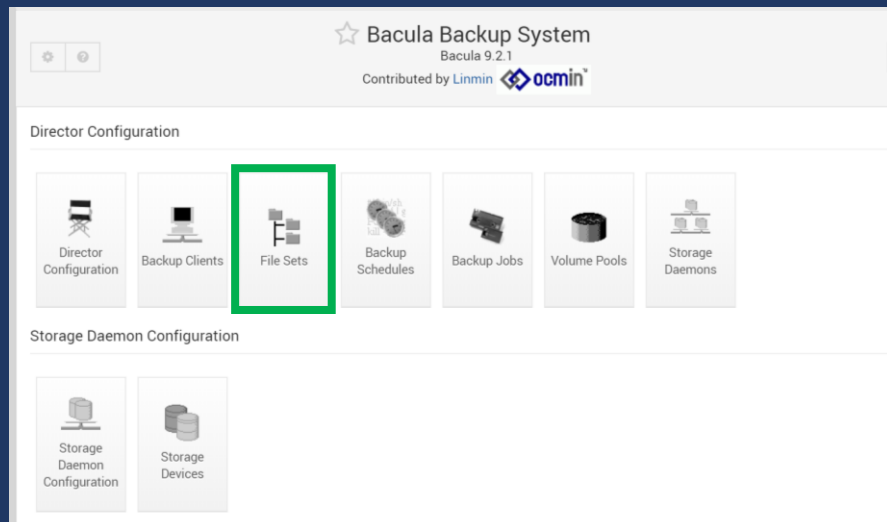
- Una vez allí le damos a **add a new backup client**, para poder agregar al nuevo cliente.



- Cuando nos encontremos en ese apartado pondremos el nombre que le daremos al cliente dentro de bacula, la password que le asignamos en el archivo de lado del cliente, su dirección ip, el puerto del fileD que es el 9102, casi siempre aparece por default, pero tenemos que asegurarnos y con esto tenemos nuestro cliente listo, configurado los apartados ya mencionados le daremos a crear.

- Para asegurarnos que el cliente tiene comunicación correcta con báculo solo tenemos que seleccionar el cliente creado anteriormente en la sección de backups client y dar click en una de las casillas de abajo nombrada show status. En caso que la comunicación salga exitosa, nos aparecerá el nombre de la maquina cliente, la versión de bacula client que utiliza entre otras cosas, en dado caso que no la comunicación no sea lograda nos notificará que no se pudo obtener la comunicación.

- Ahora vamos a definir que es lo que vamos a respaldar, esto lo haremos dirigiéndonos a la sección de **file sets**, una vez dentro veremos algunos files predeterminados, pero estos no los tocamos crearemos uno nuevo.



- Estando allí le daremos a crear uno nuevo, y comenzaremos a configurar nuestro file set, comenzaremos poniéndole un nombre, y en la segunda opción, nombrada **file and directories to backup**, definiremos las rutas que vamos a respaldar, en las opciones de compresión la dejaremos por default, ya que es una de las mejores, y por ultimo por configurar en esta sección el **file signature type**, este comparará hashes para asegurarse que la información que está respaldando es verídica, en mi caso escogí el **MD5**. Hecho esto ya podemos darle a crear.

- Ahora vamos a definir cuando lo vamos a respaldar, para eso nos vamos a la sección **backup schedules**, donde veremos igualmente varios archivos creados, pero nosotros crearemos el nuestro, una vez en el apartado le asignaremos un nombre, luego en run levels and times, escogeremos la primera línea para configurar, empezaremos por darle full en backup level, en el volume, seleccionaremos que que hemos creado y en run at times, le daremos la configuración de días deseada, en mi caso los hará de Lunes a Viernes, todo el año, a las 22:00. Hecho esto la podremos crear

← Create Backup Schedule

Backup schedule details

Backup schedule name: windows horario

| Backup level | Volume    | Run at times     |
|--------------|-----------|------------------|
| Full         | File-pool | mon-fri at 22:00 |
| <Default>    | <Default> |                  |
| <Default>    | <Default> |                  |

Run levels and times

Create

← Return to list of schedules

- Ahora vamos a ejecutar nuestro primer job, esto lo haremos dirigiéndonos, a la sección de backups jobs, donde veremos una serie de Jobs predeterminados, pero como siempre nosotros crearemos nuestro propio Job, por eso le daremos a la pestaña de crear, una vez allí le asignamos un nombre al Job, luego en la opción de backup job enable la dejaremos en yes, en default type la dejaremos con la opción stand-alone-job, el job type le decimos que es un backup, en backup level lo pondremos en full, cliente to backup: elegimos nuestro cliente creado, backup on Schedule: elegimos nuestro horario creado, volumen pool: seccionamos nuestro pool creado, file set backup: escogemos nuestro file set creado, el destination storage device: se escogerá automatico ya que solo tenemos un solo storage y por ultimo el destination messages es el estándar, de esta manera ya tendremos nuestro primer job creado, ya podemos darle a crear.

← Edit Backup Job

Backup job details

Backup job name: Cliente Windows

Backup job enabled? ☒ Yes ☐ No

Default type: ☐ Default definition ☒ Stand-alone job ☐ Inherit defaults from: DefaultJob

Job type: Backup

Client to backup: PC1

Backup on schedule: windows horario

Volume pool: File-pool

Backup priority: ☒ Default ☐

Backup level: Full

File set to backup: windows backups

Destination storage device: File-sd

Destination for messages: Standard

Command before job: ☒ Default ☐

Command after job: ☒ Default ☐

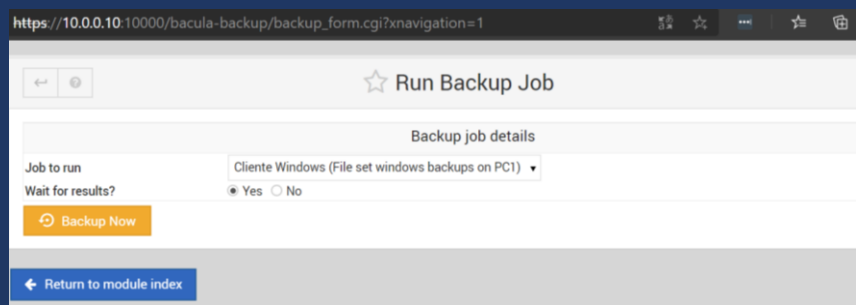
Command before job (on client): ☒ Default ☐

Command after job (on client): ☒ Default ☐

Save Run Now Delete

- Realizado esto, podemos poner a correr el job, es cierto que hay diferentes formas de hacerlo, pero nosotros lo haremos de la forma más sencilla, vamos a dirigirnos al índice de módulos y buscaremos la sección que se nombra por run backup Jobs, una vez allí seleccionamos nuestro Jobs creado en el paso

anterior, dejamos la opción de esperar resultados encendida y le podemos dar a backup now.



Si todo ha sido realizado correctamente nos saldrá una info completa de como fue el proceso del backup al sistema, en caso de tener algún fallo nos lo notificará de igual forma.

```
04-nov 19:58 bacula-dir JobId 2: Using Device "Local-File" to write.
04-nov 19:58 bacula-sd JobId 2: Labeled new Volume "Backup-0001" on File device "Local-File" (/opt/bacula/backups).
04-nov 19:58 bacula-sd JobId 2: Wrote label to pre-labeled Volume "Backup-0001" on File device "Local-File" (/opt/bacula/backups)
04-nov 19:58 bacula-sd JobId 2: Elapsed time=00:00:08, Transfer rate=3.513 M Bytes/second
04-nov 19:58 bacula-sd JobId 2: Sending spooled attrs to the Director. Despooling 541,860 bytes ...
04-nov 19:58 bacula-dir JobId 2: Bacula bacula-dir 9.2.1 (12Aug18):
Build OS:      x86_64-redhat-linux-gnu-bacula redhat Enterprise release
JobId:         2
Job:           Cliente_Linux.2018-11-04_19.58.31_05
Backup Level:  Full
Client:        "cliente-fd" 9.2.1 (12Aug18) x86_64-redhat-linux-gnu-bacula,redhat,Enterprise release
FileSet:       "Archivos cliente linux" 2018-11-04 19:52:01
Pool:         "File-Pool" (From Job resource)
Catalog:      "MyCatalog" (From Client resource)
Storage:      "File-sd" (From Job resource)
Scheduled time: 04-nov-2018 19:58:31
Start time:    04-nov-2018 19:58:33
End time:      04-nov-2018 19:58:42
Elapsed time:  9 secs
Priority:      10
FD Files Written: 2,492
SD Files Written: 2,492
FD Bytes Written: 27,796,005 (27.79 MB)
SD Bytes Written: 28,109,003 (28.10 MB)
Rate:         3088.5 KB/s
Software Compression: None
Comm Line Compression: 48.7% 1.9:1
Snapshot/VSS: no
Encryption:   no
Accurate:     no
Volume name(s): Backup-0001
```