

Bacula®

The Open Source Network Backup Solution

Sistema de Backup



Elaborado por Area Project Solutions

Parque Científico y Tecnológico de Castilla-La Mancha
Paseo de la Innovación, 1
ALBACETE

Autor: Jesús Valero Bleda

Fecha: Abril de 2010



Esta obra está bajo una [licencia de Creative Commons](https://creativecommons.org/licenses/by/4.0/).

ÍNDICE

1. Descripción.....	3
2. Componentes del programa	4
3. Configuración de Bacula	6
4. Interacción entre servicios Bacula	7
5. Nociones básicas sobre Bacula	8
6. Instalación del servidor (Director) Bacula.....	9
7. Conceptos básicos de la definición de los clientes	14
8. Instalación de un cliente Linux	16
9. Instalación de un cliente Windows	18
10. Herramientas administrativas.....	20
11. Adaptación del servidor tras la instalación de clientes	21

1. Descripción

Bacula es una solución basada en Open Source que gestiona la copia de seguridad centralizada en un servidor, de las diferentes máquinas de una red independientemente del sistema operativo.

Bacula actúa como un director de copias, guardando un catálogo (BD Mysql) de todos los ficheros relevantes que deben ser copiados y realizando las copias de forma completa o bien sólo de los cambios ocurridos desde una fecha concreta. Para depositar las copias, este producto puede utilizar diversos dispositivos los cuales pueden ser añadidos sin límite y pueden ser desde unidades de discos, volúmenes NAS/SAN, unidades de cinta, librerías y autocargadores, etc.

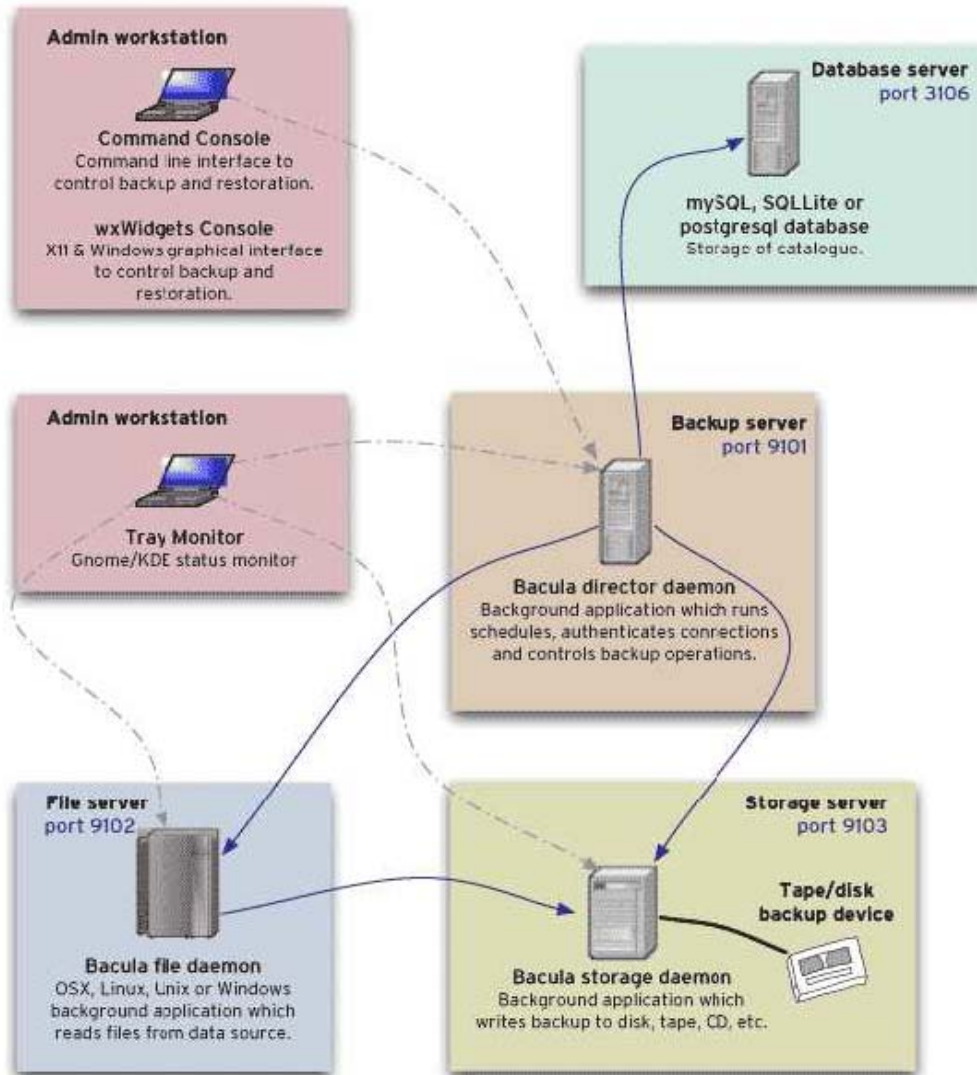
Permite salvaguardar los datos clientes, transmitiéndolos a través de la red local o incluso Internet. La información circula y se almacena de forma totalmente segura gracias a la integración de algoritmos de cifrado y certificados digitales.

Es una herramienta muy sencilla de usar lo que permite que la administración se realice sin necesidad de tener elevados conocimientos técnicos.

El presente manual recoge la documentación necesaria para instalar, configurar y administrar el servidor y los clientes Windows/Linux.

2. Componentes del programa

Bacula está constituido por cinco elementos/servicios principales que son :



DIRECTOR (DIR)

Programa servidor que supervisa todas las funciones necesarias para las operaciones de "backup" y "restore". Es el eje central de la solución y en él se declaran todos los parámetros necesarios. Se ejecuta como un "demonio" en el servidor.

STORAGE (SD)

Programa que gestiona las unidades (discos, bandas magnéticas, etc. donde se almacenarán los datos. Es el responsable de escribir y leer en los medios que utilizaremos para nuestras copias de seguridad. Se ejecuta como un "demonio" en la máquina propietaria de los medios utilizados. En muchos casos será en el propio servidor.

FILE (FD)

Este servicio, conocido como "cliente" está instalado en cada máquina a salvaguardar y es específico al sistema operativo donde se ejecuta. Responsable para enviar al "Director" los datos cuando este lo requiera.

CATALOG

Base de datos (MySQL en nuestro caso) que almacena la información necesaria para localizar donde se encuentran los datos salvaguardados de cada archivo, de cada cliente, etc.

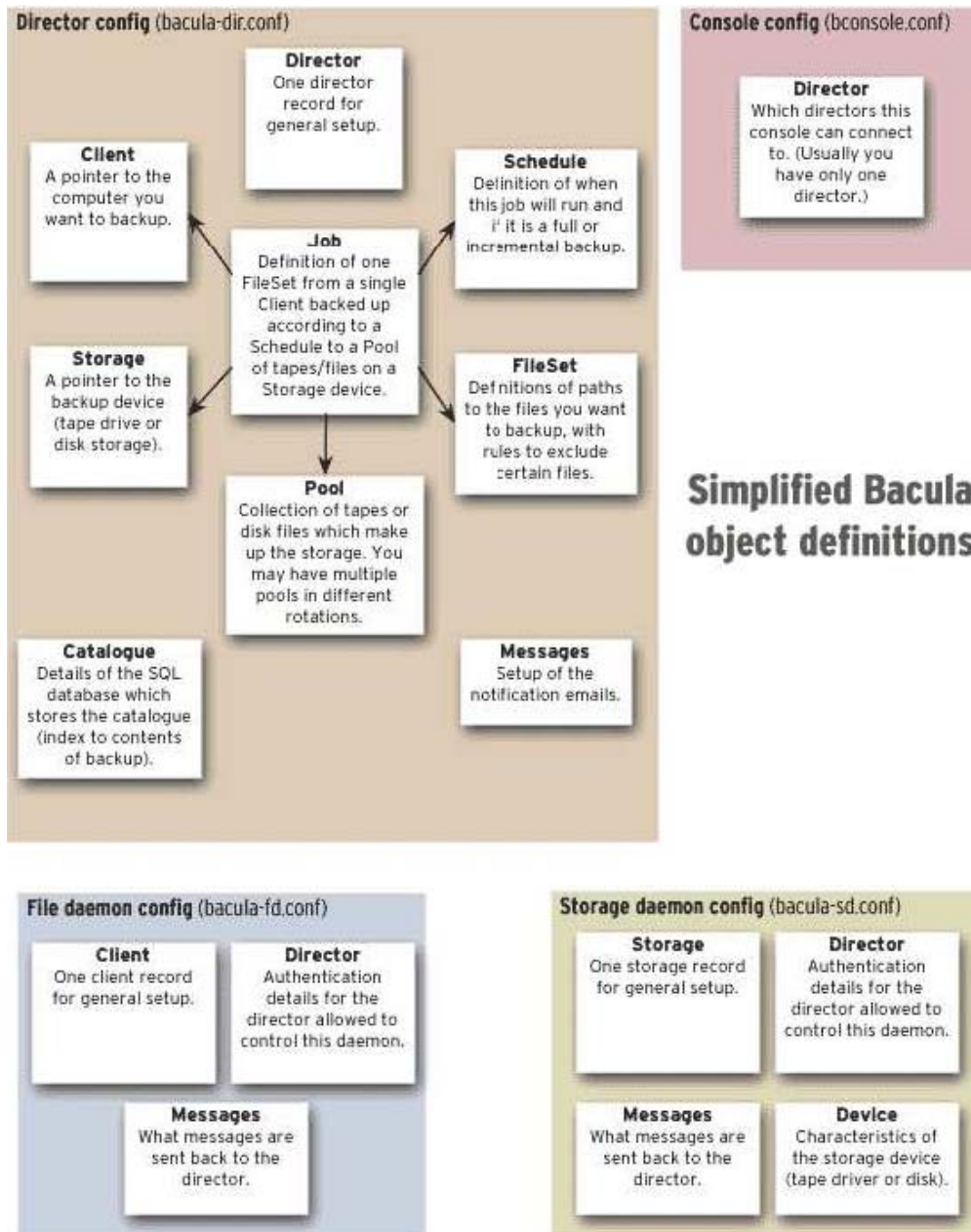
CONSOLE

Programa que permite al administrador la interacción con el "Director" para todas las funciones del servidor. La versión original es una aplicación en modo texto. Existen igualmente aplicaciones GUI para Windows y Gnome (BAT).

3. Configuración de Bacula

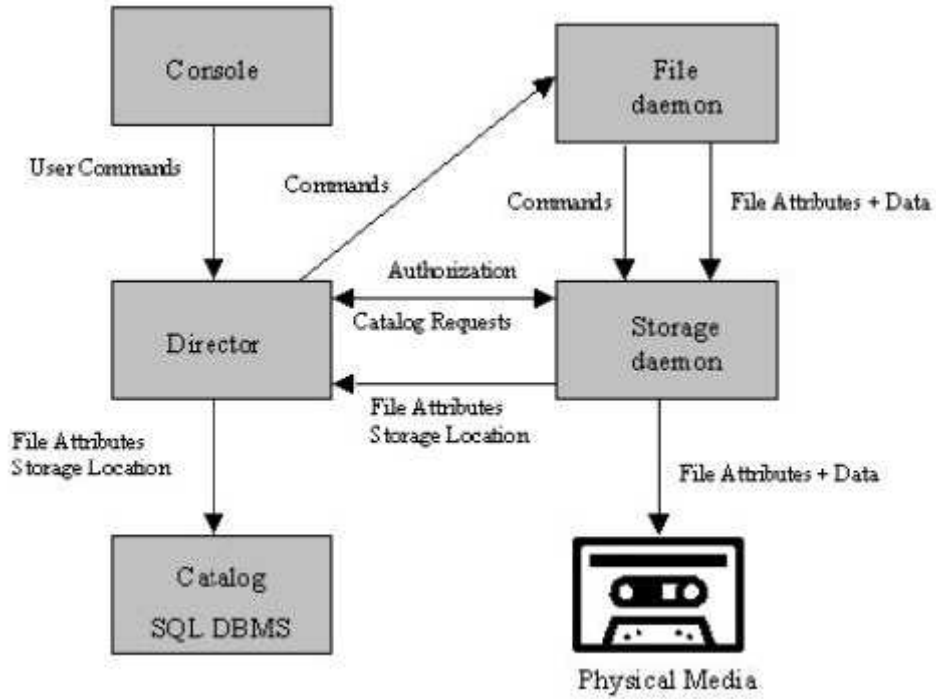
Un conjunto de archivos de configuración identifican todos los componentes necesarios para la buena marcha del sistema.

El más complejo es, sin duda, el del servidor (DIR). En él se declara la mayoría de la información que se utilizará para gestionar toda la actividad de esta solución. En los anexos figura un ejemplo de cada uno de ellos.



4. Interacción entre servicios BACULA

La figura siguiente muestra las típicas interacciones entre los servicios de Bacula para la ejecución de un "backup". El Director inicia y gestiona estas interacciones además de administrar el Catalogo.



5. Nociones básicas sobre BACULA

Sin entrar en los detalles vamos a aclarar algunos conceptos y términos básicos de la solución Bacula.

Un “backup” consiste en una tarea (JOB), un conjunto de directorios/archivos (FILESET), un cliente (CLIENT), un horario (SCHEDULE) y unos recursos (POOL).

En otros términos, el Fileset es lo que vamos a salvaguardar, el Client es la proveniencia de los datos, el Schedule determina cuando lo vamos a ejecutar y el Pool es el destino de la copia de seguridad.

Tipicamente una combinación CLIENT/FILESET generará un determinado JOB. Además de los JOB de “backup”, existirán también JOB de “restore” y otros de control y administración.

Los medios de almacenamiento se definen como POOL. El POOL es un conjunto de VOLUME y este representa por ejemplo una banda magnética o un archivo disco.

Todos estos conceptos se verán detalladamente más adelante.

6. Instalación del servidor (DIRECTOR) Bacula

Antes de todo recordar que Bacula necesita un SGDB (en nuestro caso MySql) para mantener el CATALOGO.

Los paquetes existentes para Debian STABLE son de una versión demasiado antigua y causan problemas de compatibilidad con los clientes Windows por lo que hemos optado por una instalación manual.

Distribución "Source" actual

bacula-5.0.1.tar.gz
bacula-gui-5.0.1.tar.gz
bacula-docs-5.0.1.tar.bz2
win32bacula-5.0.1.exe

Compilación/instalación del servidor (y del cliente de esa máquina)

- Instalar y configurar MySQL

```
apt-get install mysql-server mysql-client
```
- Crear un directorio /bacula donde se descomprime el archivo bacula-5.0.1.tar.gz

```
tar -zxf bacula-5.0.1.tar.gz
```

esto nos creará la carpeta /bacula/bacula-5.0.1
- Antes de empezar con la compilación, verificar si los paquetes siguientes estén instalados, sino instalarlos con apt-get:

```
gcc - g++ - make - libmysqlclient15-dev
```
- Podemos iniciar el proceso de compilación con :

```
cd /bacula/bacula-5.0.1
./configure --with-mysql
```
- Controlar atentamente el resultado con :

```
more configure.out
```

si considera que todos los parametros son correctos y que no hay errores proseguir.
- Ejecutamos la compilación :

```
make
make install
```

si todo se ha desarrollado bien, seguimos con la creación del CATALOGO en Mysql

Creación del catalogo Mysql

- Vamos a utilizar unos scripts SQL para crear la estructura necesaria en el CATALOGO de Bacula. Estos scripts suponen que el usuario root:localhost en Mysql no tiene contraseña. En el caso de que si tenga contraseña vamos a modificar la llamada a **mysql**.

Editamos los tres archivos que vamos a utilizar:

```
/etc/bacula/create-mysql-database  
/etc/bacula/make-mysql-tables  
/etc/bacula/grant-mysql-privileges
```

Modificamos la línea:

```
if $bindir/mysql $* -f <<END-OF-DATA
```

Por la siguiente:

```
if $bindir/mysql -u root -p $* -f <<END-OF-DATA
```

- Creación de la base de datos :

```
/etc/bacula/create-mysql-database
```
- Creación de las tablas :

```
/etc/bacula/make-mysql-tables
```
- Creación del usuario con sus autorizaciones:

```
/etc/bacula/grant-mysql-privileges
```

En este punto el servidor completo (con su catalogo) y el cliente de la máquina están instalados. Seguimos con los **archivos de configuración** antes de proceder al **arranque** de los servicios.

Archivos de configuración

En el servidor tenemos cuatro archivos de configuración. No voy a documentar de manera exhaustiva los diferentes parámetros del archivo.

En los anexos y en la carpeta de instalación figuran los ejemplos. La documentación oficial del producto es muy completa y muy clara.

- **bacula-dir.conf**

Es el más complejo de todos ya que contiene la mayoría de los datos necesarios para la gestión del entorno completo. También es el que más se modifica dado que cada instalación de un nuevo cliente conlleva de agregarle los datos pertinentes como veremos más tarde.

- **bacula-sd.conf**

La configuración del almacenamiento (STORAGE) tiene pocos parámetros para un entorno normal (disco y/o bandas magnéticas como medios de almacenamiento).

- **bacula-fd.conf**

La configuración del cliente es la más simple y raramente se tiene que modificar aparte de que tiene también muy pocos parámetros.

- **bconsole.conf**

Configuración de la consola de administración.

Renombrar los cuatro archivos **.conf** de la carpeta **/etc/bacula** y **copiar** los cuatro que figuran en la carpeta de instalación.

```
mv bacular-dir.conf bacular-dir-old.conf
mv bacular-sd.conf bacular-sd-old.conf
mv bacular-fd.conf bacular-fd-old.conf
mv bconsole.conf bconsole.conf
```

En el **archivo bacular-sd.conf** se define el tipo de soporte que vamos a utilizar para almacenar los backups. En nuestro caso almacenaremos todo en disco y en una carpeta:

```
Device {
  Name = FileStorage
  Media Type = File
  Archive Device = /var/bacula/backups
  ....
}
```

Creamos por tanto este directorio:

```
mkdir /var/bacula/backups
```

En los cuatro archivos de configuración hay que **comprobar que las direcciones/hosts IP** sean correctas.

Arranque del servidor

En la carpeta /etc/bacula se encuentran los scripts de arranque. En nuestro caso nos interesa al archivo bacula que ejecutará otros scripts.

- Para arrancar/detener el servidor BACULA

```
/etc/bacula/bacula start
/etc/bacula/bacula stop
```

- Para comprobar el estado de los "demonios" :

```
/etc/bacula/bacula status
```

Si los tres "demonios" están **"running..."** vamos por el buen camino, proseguimos...

En este punto tenemos que poder conectarnos con la consola de administración:

```
/etc/bacula/bconsole
```

Nos tiene que responder con un prompt: *

Si es el caso, se puede probar el comando: **help**

Más adelante veremos en detalle otros comandos, para salir: **exit**

Conceptos básicos de la definición de los medios de almacenamiento

Esta instalación está prevista para que Bacula utilice el disco como medio de almacenamiento. Vamos a explicar algunos detalles. **No hay que cambiar nada para una instalación normal.** El concepto POOL/VOLUMES nos permite una flexibilidad bastante grande para adaptar el programa a nuestras necesidades.

En el archivo de configuración **bacula-sd** se define el tipo de la "unidad de soporte" tal y como vemos seguidamente.

```
Device {
  Name = FileStorage
  Media Type = File
  Archive Device = /var/bacula/backup
  LabelMedia = yes
  Random Access = Yes;
  AutomaticMount = yes;
  RemovableMedia = no;
```

```
AlwaysOpen = no;  
}
```

En el archivo de configuración **bacula-dir** se definen los apartados **"POOL"** y **"STORAGE"**. El apartado **"STORAGE"** crea la relación con el "demonio" STORAGE que está escuchando en el puerto 9103 de la dirección "Address".

```
Storage {  
  Name = File  
  Address = baculaserver01  
  SDPort = 9103  
  Password = "password"  
  Device = FileStorage  
  Media Type = File  
}
```

El apartado POOL define los **volúmenes** que se van a crear

```
Pool {  
  Name = File  
  Pool Type = Backup  
  Recycle = yes # Bacula can automatically recycle Volumes  
  AutoPrune = yes # Prune expired volumes  
  Volume Retention = 365 days # one year  
  Maximum Volume Bytes = 10G # Limit Volume size to something reasonable  
  Maximum Volumes = 100 # Limit number of Volumes in Pool  
  LabelFormat = "Vol"  
}
```

7. Conceptos básicos de la definición de los clientes

Cada cliente y su respectivo "backup" se gestionan en el archivo de configuración **bacula-dir.conf** a través de los cuatro apartados siguientes:

```
Client {
  Name = baculaserver01-fd # Nombre del cliente, tiene que corresponder con el
                           # nombre en el archivo bacula-fd conf del cliente
  Address = baculaserver01 # Dirección del cliente (IP o Host)
  FDPort = 9102            # Puerto de escucha del cliente
  Catalog = MyCatalog
  Password = "password"
  File Retention = 30 days
  Job Retention = 6 months
  AutoPrune = yes
}
```

```
Job {
  Name = "BackupServer"
  Client = baculaserver01-fd
  Type = Backup
  Level = Incremental
  FileSet = "Full Set Linux"
  Schedule = "WeeklyCycle"
  Storage = File
  JobDefs = "DefaultJob"
}
```

La pareja Client/Job es obligatoria, tiene que ser única y define el backup de cada cliente. Los detalles de los datos a salvaguardar y cuando se van a salvaguardar los definimos en los dos siguientes apartados que pueden ser específicos de un cliente o comunes a varios clientes.

```
FileSet {
  Name = "Full Set Linux"
  Include {
    Options {
      signature = MD5
    }
    File = /root
    File = /home
  }

  Exclude {
    File = /var/bacula/working
    File = /tmp
    File = /proc
    File = /tmp
  }
}
```

```
}  
}
```

En el apartado FileSet, definimos con Include/Exclude que carpetas/archivos vamos a salvaguardar. Puede ser global para varios clientes o personalizado para un cliente.

```
Schedule {  
  Name = "WeeklyCycle"  
  Run = Full 1st sun at 23:05  
  Run = Differential 2nd-5th sun at 23:05  
  Run = Incremental mon-sat at 23:05  
}
```

En el apartado Schedule, definimos cuando se va a ejecutar la salvaguarda. Igual que el FileSet, puede ser global para varios clientes o personalizado para un cliente.

Con la instalación del servidor se instala también el cliente del servidor.

8. INSTALACION DE UN CLIENTE LINUX (FD) Bacula

Distribución "Source" actual

bacula-5.0.1.tar.gz

Compilación/instalación del cliente Linux

- **Crear un directorio /bacula** donde copiamos y descomprimos el archivo bacula-5.0.1.tar.gz

```
tar -zxf bacula-5.0.1.tar.gz
```

Esto nos creará la carpeta **/bacula/bacula-5.0.1**

- Antes de empezar con la compilación, verificar si los paquetes siguientes estén instalados, sino instalarlos con apt-get:

```
gcc - g++ - make
```

- Podemos iniciar el proceso de compilación con :

```
cd /bacula/bacula-5.0.1  
./configure --enable-client-only
```

- Controlar atentamente el resultado con :

```
more configure.out
```

Si considera que todos los parámetros son correctos y que no hay errores proseguir.

- Ejecutamos la compilación :

```
make  
make install
```

Archivos de configuración

La carpeta de instalación se encuentra normalmente en `"/etc/bacula"`. Editamos el archivo de configuración del cliente `bacula-fd.conf`. Adaptamos los parámetros señalados para que coincidan con el archivo `bacula-dir.conf` del servidor.

```
#
# List Directors who are permitted to contact this File daemon
#
Director {
    Name = baculaserver01
    Password = "password"
}

#
# Restricted Director, used by tray-monitor to get the
# status of the file daemon
#

#### Comentar

#Director {
# Name = debian-client-001-mon
# Password = "Gddm8epYBPUi0J4mt2lJSsnsFN6jj50hNX5b"
# Monitor = yes
#}

#
# "Global" File daemon configuration specifications
#
FileDaemon {
    Name = ClientDEB-001-fd # this is me
    FDport = 9102 # where we listen for the director
    WorkingDirectory = /var/bacula/working
    Pid Directory = /var/run
    Maximum Concurrent Jobs = 20
}

# Send all messages except skipped files back to Director
Messages {
    Name = Standard
    director = baculaserver01-dir = all, !skipped, !restored
}

```

Ejecutamos el script `"bacula start"` y nuestro cliente está activo.

[Saltar al capítulo "Adaptación del servidor"](#)

9. INSTALACION DE UN CLIENTE WINDOWS XP/7 (FD) Bacula

Distribución actual

win32bacula-5.0.1.exe

Compilación/instalación del cliente Windows

- Ejecutar el programa win32bacula-5.0.1.exe
- Next en la pantalla :
 - Welcome to the Bacula Setup**
- I agree en la pantalla :
 - License Agreement**
- Next en la pantalla :
 - Installation Type (automatic)**
- Seleccionar todas (3) las opciones y Next en la pantalla :
 - Choose components**
- Pantalla Configuration :

DIR Name	:	baculaserver01-01
DIR Password	:	password
DIR Address	:	baculaserver01

Estos datos tienen que coincidir con la definición del DIRECTOR en el servidor.
Se pueden modificar en archivo de configuración bacula-fd.
- Install, en esta pantalla (Configuration) :
 - Durante la instalación un mensaje "normal" de error aparece, darle a Aceptar.
- Next en la pantalla (Create Templates) :
 - Nos informa de que crea un archivo ejemplo para los datos a agregar en el servidor. Lo veremos más adelante.
- Finish

La instalación se ha completado. Nos ha creado y activado un "servicio" "Bacula File Service" y se queda a la escucha en el **puerto 9102**.

Archivos de configuración

La carpeta de instalación se encuentra normalmente en "**c:\Archivos de Programa\bacula**". Editamos el archivo de configuración del cliente **bacula-fd.conf**.

```
FileDaemon {
  Name = ClientW7-001-fd
  FDport = 9102
  WorkingDirectory = "C:\\Program Files\\Bacula\\working"
  Pid Directory = "C:\\Program Files\\Bacula\\working"
  # Plugin Directory = "C:\\Program Files\\Bacula\\plugins"
  Maximum Concurrent Jobs = 10
}
```

Adaptamos el nombre a nuestros estándares. Este nombre deberá coincidir con el cliente que tendremos que agregar al bacula-dir.conf en el Servidor:

```
Director {
  Name = baculaserver01
  Password = "4hPQJFAxOkG+H5WDfhXeygDT4eyPcw19o6+edrJsJ5hh"
```

Adaptar el password y verificar que el nombre coincida con el bacular-dir.conf del Servidor.

```
Director {
  Name = equipo01-mon
  Password = "2duIMPtTfi1Qm4rZqF+BbW3N1DQOzAZgmmqTKUZ3OcPe"
  Monitor = yes
```

Comentar estas líneas que no se utilizan por ahora.

10. Herramientas administrativas

Las herramientas administrativas se instalan también por defecto, aunque no sean necesarias en todos los casos. Si se van a utilizar, editar los archivos **bat.conf** y **bconsole.conf**.

```
Director {  
  Name = baculaserver01  
  DIRport = 9101  
  address = baculaserver01  
  Password = "password"  
}
```

Los dos son idénticos y tienen que corresponder en los datos al bacula-dir.conf. Lo más importante es que el host o la dirección IP del servidor sean accesibles.

11. Adaptación del servidor después de una instalación cliente

Llegados a este punto nos queda a adaptar el archivo del servidor **bacula-dir.conf**.

Tenemos que **agregar un apartado Client**:

```
Client {
  Name = ClientW7-001-fd
  Address = 192.168.0.50
  FDPort = 9102
  Catalog = MyCatalog
  Password = "password"
  File Retention = 30 days
  Job Retention = 6 months
  AutoPrune = yes
}
```

El nombre tiene que coincidir con el bacula-fd.conf del cliente y la dirección IP o el host del cliente tiene que ser accesible desde el cliente.

Tenemos que **agregar un apartado Job**

```
Job {
  Name = "BackupClientW7-001"
  Client = ClientW7-001-fd
  Type = Backup
  Level = Incremental
  FileSet = "Full Set W7"
  Schedule = "WeeklyCycle"
  Storage = File
  JobDefs = "DefaultJob"
}
```

En **FileSet** y **Schedule** tenemos que darle unos que ya existan o crear nuevos apartados :

```
FileSet {
  Name = "Full Set W7"
  Include {
    Options {
      signature = MD5
    }
    File = "c:/prueba"
  }
  Exclude {
  }
}
```

```
Schedule {  
  Name = "WeeklyCycle"  
  Run = Full 1st sun at 23:05  
  Run = Differential 2nd-5th sun at 23:05  
  Run = Incremental mon-sat at 23:05  
}
```

Una vez que los clientes están instalados y que hemos agregado los apartados necesarios al archivo de configuración del servidor (DIRECTOR) es el momento de analizar los procesos de respaldo y restauración (backup/restore).

Recordamos que hemos instalado tres clientes. Cada uno con su tarea (JOB) respectiva. Esta tarea se va a ejecutar a una hora determinada (SCHEDULE) y va a respaldar una serie de carpetas/archivos que definimos en el FILESET y el servidor las almacenará en el POOL.

En base a estos datos, **el Servidor gestiona todos los contactos** con los clientes para que estos procedan a enviar sus datos.

Tenemos que recordar también un elemento que quizás hemos pasado por alto hasta ahora pero que es muy importante. Se trata del CATALOG, base de datos Mysql en nuestro caso y donde el servidor almacenará la información necesaria sobre los respaldos efectuados y que le permitirán gestionar una restauración.

Como ya hemos mencionado, disponemos de una herramienta de base, en modo carácter con posibilidad de integrarla en scripts, que se llama **bconsole**. Esta herramienta permite todo tipo de interacción con el servidor. En entornos Windows/Gnome existe una interfaz GUI llamada **BAT** que facilita bastante el control de la actividad del servidor y **muy útil para las restauraciones**.

Bacula-web es un programa PHP que presenta datos resumidos sobre las tareas ejecutadas con gráficos. No deja de ser interesante pero no es indispensable.

Copyright © 2010 AREA PROJECT SOLUTIONS



Esta obra está bajo una [licencia de Creative Commons](https://creativecommons.org/licenses/by/4.0/).