Bacula Enterprise User Interfaces

Bacula Systems Documentation

Contents

Contents

The following chapter aims at presenting information on Bacula Enterprise user interfaces.

1 Console

To allow interaction from administrators or users, Bacula uses *Consoles*. The Bacula Console is a program that allows the user or the System Administrator to interact with the Bacula Director Daemon while the daemon is running. Note that, even when managing storage or checking client status, the Console interacts with the Director only, which in turn contacts the other daemons as needed.

Since the Console program interacts with the Director through the network, the Console and Director programs do not necessarily need to run on the same machine. In fact, in an installation containing a single tape drive, a certain minimal knowledge of the Console program may be needed in order for Bacula to be able to write on more than one Volume, because when Bacula requests a new one, it waits until the user, via the Console program, indicates that the new Volume is mounted or labeled to be used.

1.1 Console Installation

The Console is installed by default with the Director or the File Daemon installation.

The Console can be installed also on its own with the package named *bacula-enterprise-console*.

Console Configuration

When the *Console* starts, it reads a standard **Bacula** configuration file named **bconsole.conf** or **bat.conf** in the case of the *BAT* QT *Console* version from the current directory unless you specify the **-c** command line option (see below). This file allows default configuration of the *Console*, and at the current time, the only Resource Record defined is the **Director**, which gives the *Console* the name and address of the *Director*.

Console Management

Running the Console Program

Plese type *sudo -u bacula /opt/bacula/bin/bconsole* in order to run the console.

The Bacula **Console** program can be run with the following options:

```
Usage: bconsole [-s] [-c config-file] [-d debug-level]
       -D <dir>    select a Director
       -l          list Directors defined
       -L          list Consoles defined
       -C <cons>   select a console
       -c <file>   set configuration file to file
       -d <nn>     set debug level to <nn>
       -dt         print timestamp in debug output
       -n          no conio
       -s          no signals
       -u <nn>     set command execution timeout to <nn> seconds
       -t          test - read configuration and exit
       -?          print this message.
```

After launching the *Console* program (**bconsole**), it will prompt you for the next command with an asterisk (*). Generally, for all commands, you can simply enter the command name and the Console

program will prompt you for the necessary arguments. Alternatively, in most cases, you may enter the command followed by arguments. The general format is:

```
<command> <keyword1>[=<argument1>] <keyword2>[=<argument2>] ...
```

where **command** is one of the commands listed below; **keyword** is one of the keywords listed below (usually followed by an argument); and **argument** is the value. The command may be abbreviated to the shortest unique form. If two commands have the same starting letters, the one that will be selected is the one that appears first in the **help** listing. If you want the second command, simply spell out the full command. None of the keywords following the command may be abbreviated.

For example:

```
list files jobid=23
```

will list all files saved for JobId 23. Or:

```
show pools
```

will display all the **Pool** resource records.

An abbreviated command would be similar to

```
s dir
```

which will show the overall status of the Director currently connected to.

In many cases, **bconsole** will provide tab completion of commands, options names and arguments. Also, command line history and editing may be available.

Depending on how the program is built, the regular **readline** key bindings will be used. If the readline functionality is used, and not the alternative Bacula native one, the full functionality of **readline**, including command history and configuration through the **~/.inputrc** file, is available. We recommend to look at the available configuration as provided by the operating system distribution.

Tab completion is a particularly useful tool, but depends on availability of information that is gathered from the Director at run time, using some dot commands behind the scenes. Thus, it depends on the availability of the commands needed, which may be restricted in some cases. Please see *Special dot Commands* for more details.

Command line completion can simplify interactive activity a lot:

```
~$ bconsole -c .config/bacula.org/k8s-bconsole.conf
Connecting to Director am-d11-director-12-8-3-tst.supportlab.lan:9101
1000 OK: 10002 am-d11-director-12-8-3-tst-dir Version: 12.8.3 (28 September␣
→2021)
Enter a period to cancel a command.
*st<tab><tab>
statistics  status      stop
*status cli<tab>ent=<tab>am-u20-k8s-master01-bck-fd<enter>
Connecting to Client am-u20-k8s-master01-bck-fd at am-u20-k8s-master01-bck.
→supportlab.lan:9102

am-u20-k8s-master01-bck-fd Version: 12.8.3 (28 September 2021) ...
```

The maximum command line length is limited to **511** characters, so if you are scripting the console, you may need to take some care to limit the line length.

Stopping the Console Program

Normally, you simply enter **quit** or **exit** and the *Console* program will terminate. However, it waits until the *Director* acknowledges the command. If the *Director* is already doing a lengthy command (e.g. **prune**), it may take some time. If you want to immediately terminate the *Console* program, enter the **.quit** command.

There is currently no way to interrupt a *Console* command once issued (i.e. `Ctrl-C` does not work). However, if you are at a prompt that is asking you to select one of several possibilities and you would like to abort the command, you can enter a period (**.**), and in most cases, you will either be returned to the main command prompt or if appropriate the previous prompt (in the case of nested prompts). In a few places such as where it is asking for a Volume name, the period will be taken to be the Volume name. In that case, you will most likely be able to cancel at the next prompt.

Alphabetic List of Console Keywords

Unless otherwise specified, each of the following keywords takes an argument, which is specified after the keyword following an equal sign. For example:

```
jobid=536
```

all
> Permitted on the **status** and **show** commands to specify all components or resources respectively. Takes no argument.

frompool
> Permitted on the **update volume** command to specify that the volume specified on the command line should be updated with pool parameters.

allfrompool=<pool>
> Permitted on the **update** command to specify that all Volumes in the pool (specified on the command line) should be updated.

fromallpools
> Permitted on the **update** command to specify that all Volumes in all pools should be updated.

before
> Used in the **restore** command.

bootstrap
> Used in the **restore** command.

catalog
> Allowed in the **use** command to specify the catalog name to be used.

catalogs
> Used in the **show** command. Takes no arguments.

client | fd

clients
> Used in the **show**, **list**, and **llist** commands. Takes no arguments.

counters
> Used in the **show** command. Takes no arguments.

current
> Used in the **restore** command. Takes no argument.

days
Used to define the number of days the **list nextvol** command should consider when looking for jobs to be run. The **days** keyword can also be used on the **status dir** command so that it will display jobs scheduled for the number of days you want.

devices
Used in the **show** command. Takes no arguments.

dir | director

directors
Used in the **show** command. Takes no arguments.

directory
Used in the **restore** command. Its argument specifies the directory to be restored.

enabled
This keyword can appear on the **update volume** as well as the **update slots** commands, and can allows one of the following arguments: **yes**, **true**, **no**, **false**, **archived**, **0**, **1**, **2**. Where **0** corresponds to no or false, **1** corresponds to **yes** or **true**, and **2** corresponds to **archived**. Archived volumes will not be used, nor will the Media record in the catalog be pruned. Volumes that are not enabled, will not be used for backup or restore.

done
Used in the **restore** command. Takes no argument.

file
Used in the **restore** command.

files
Used in the **list** and **llist** commands. Takes no arguments.

fileset

filesets
Used in the **show** command. Takes no arguments.

help
Used in the **show** command. Takes no arguments.

jobs
Used in the **show**, **list** and **llist** commands. Takes no arguments.

jobmedia
Used in the **list** and **llist** commands. Takes no arguments.

jobtotals
Used in the **list** and **llist** commands. Takes no arguments.

jobid
The JobId is the numeric jobid that is printed in the Job Report output. It is the index of the database record for the given job. While it is unique for all the existing Job records in the catalog database, the same JobId can be reused once a Job is removed from the catalog. Probably you will refer specific Jobs that ran using their numeric JobId.

job | jobname
The **job** or **jobname** keyword refers to the name you specified in the **Job**, and hence it refers to any number of Jobs that ran. It is typically useful if you want to list all jobs of a particular name.

level

listing

> Permitted on the **estimate** command. Takes no argument.

limit

messages

> Used in the **show** command. Takes no arguments.

media

> Used in the **list** and **llist** commands. Takes no arguments.

name

> Used in the **list object** commands. Can specify an Object name.

nextvol | nextvolume

> Used in the **list** and **llist** commands. Takes no arguments.

noautoparent

> Used with the **restore** command. Takes no arguments.

object

> Used in the **list** commands. Takes no arguments.

objectid

> Used in the **list** and **restore** commands. Takes an ObjectId as argument.

order

> Used in the **list** to sort records. Can take "ASC" or "DESC" as argument.

on

> Takes no keyword.

off

> Takes no keyword.

pool

pools

> Used in the **show**, **list**, and **llist** commands. Takes no arguments.

select

> Used in the **restore** command. Takes no argument.

limit

> Used in the **setbandwidth** command. Takes integer in `KB/s` unit.

storages

> Used in the **show** command. Takes no arguments.

schedules

> Used in the **show** command. Takes no arguments.

sd | store | storage

tag

> Used in the **list** command. Takes no arguments.

ujobid

> The **ujobid** is a unique job identification that is printed in the Job Report output. At the current time, it consists of the Job name (from the **Name** for the job) appended with the date and time the job was run. This keyword is useful if you want to completely identify the Job instance run.

volume

volumes

Used in the **list** and **llist** commands. Takes no arguments.

where

Used in the **restore** command.

yes

Used in the **restore** command. Takes no argument.

Alphabetic List of Console Commands

The following commands are currently implemented:

add [pool=<pool-name> storage=<storage> jobid=<JobId>]

This command is used to add Volumes to an existing Pool. That is, it creates the Volume name in the catalog and inserts into the Pool in the catalog, but does not attempt to access the physical Volume. Once added, **Bacula** expects that Volume to exist and to be labeled. This command is not normally used since **Bacula** will automatically do the equivalent when Volumes are labeled. However, there may be times when you have removed a Volume from the catalog and want to later add it back.

Normally, the **label** command is used rather than this command because the **label** command labels the physical media (tape, disk, DVD, …) and does the equivalent of the **add** command. The **add** command affects only the Catalog and not the physical media (data on Volumes). The physical media must exist and be labeled before use (usually with the **label** command). This command can, however, be useful if you wish to add a number of Volumes to the Pool that will be physically labeled at a later time. It can also be useful if you are importing a tape from another site. Please see the **label** command below for the list of legal characters in a Volume name.

autodisplay on/of

This command accepts **on** or **off** as an argument, and turns auto-display of messages on or off respectively. The default for the console program is **off**, which means that you will be notified when there are console messages pending, but they will not automatically be displayed.

When autodisplay is turned off, you must explicitly retrieve the messages with the **messages** command. When **autodisplay** is turned on, the messages will be displayed on the console as they are received.

automount on/of

This command accepts **on** or **off** as the argument, and turns auto-mounting of the Volume after a **label** command on or off respectively. The default is **on**. If **automount** is turned off, you must explicitly **mount** tape Volumes after a **label** command to use it.

cancel [jobid=<number> job=<job-name> ujobid=<unique-jobid>] | [inactive
client=<client-name> storage=<storage-name>]

This command is used to cancel a job and accepts **jobid=nnn** or **job=xxx** as an argument where nnn is replaced by the JobId and xxx is replaced by the job name. If you do not specify a keyword, the *Console* program will prompt you with the names of all the active jobs allowing you to choose one.

Once a Job is marked to be canceled, it may take a bit of time (generally within a minute but up to two hours) before the Job actually terminates, depending on what operations it is doing. Don't be surprised that you receive a Job not found message. That just means that one of the three daemons had already canceled the job. Messages numbered in the 1000's are from the Director, 2000's are from the File daemon and 3000's from the Storage daemon.

If the Job is still active on the Storage Daemon and/or the File Daemon, but not on the Director, it is possible to stop the Job with the **inactive** option of the **cancel** command.

cloud [storage=<sd-name> volume=<vol-name> allpools allfrompool pool=<p-name> mediatype=<type-name> drive=<num> slots=<num>] [truncate | prune | list | upload]
>The **cloud bconsole** command allows you to do a number of things with cloud volumes.
>
>The **truncate** option of the cloud command will attempt to truncate the local cache for the specified Volume.
>
>The **prune** option of the cloud command will attempt to prune the local cache for the specified Volume. **Bacula** will respect the **CacheRetention** volume attribute to determine if the cache can be truncated or not. Only parts that are uploaded to the cloud will be deleted from the cache.
>
>The **upload** option of the cloud command will attempt to upload the specified Volumes.
>
>The **list** option of the cloud command will list volumes stored in the Cloud. If a volume name is specified, the command will list all parts for the given volume.

create [pool=<pool-name>]
>This command is not normally used as the Pool records are automatically created by the *Director* when it starts based on what it finds in the conf file. If needed, this command can be to create a *Pool* record in the database using the **Pool** record defined in the *Director*'s configuration file. So in a sense, this command simply transfers the information from the **Pool** in the configuration file into the *Catalog*. Normally this command is done automatically for you when the *Director* starts providing the *Pool* is referenced within a **Job**. If you use this command on an existing *Pool*, it will automatically update the *Catalog* to have the same information as the **Pool**. After creating a *Pool*, you will most likely use the **label** command to label one or more volumes and add their names to the Media database.
>
>When starting a *Job*, if **Bacula** determines that there is no *Pool* record in the database, but there is a **Pool** of the appropriate name, it will create it for you. If you want the *Pool* record to appear in the database immediately, simply use this command to force it to be created.

dedup
>This command is used to manage Global Deduplication Engine.
>
>>**[storage=<storage-name> vacuum [forceoptimize] [holepunching] [check-miss] [checkindex] usage scrub [run | stop | suspend | resume | status] [worker=<nn>] [reset] tune indexmemory bnum_min=<nn> bnum_max=<nn> mlock_strategy=<nn> mlock_max=<nn> hole_size=<nn> rehydra_check_hash=<nn> scrub_bwlimit=<nn> qindex [[:NN:] [:XXXXXXXX] [:X{32}] [0xXX] [all]] qchunk [[size] [check-hash] [:NN:] [0xXX] [all]] qextent [[:NN:] [0xXX] all] qcontainer]**

delete [volume=<vol-name> pool=<pool-name> job jobid=<id> object [objectid=id]]
>The **delete** command is used to delete a *Volume*, *Pool* , Object, Snapshot or *Client* record from the *Catalog* as well as all associated catalog Volume records that were created. This command operates only on the *Catalog* database and has no effect on the actual data written to a *Volume*. This command can be dangerous and we strongly recommend that you do not use it unless you know what you are doing.

If the keyword **Volume** appears on the command line, the named *Volume* will be deleted from the *Catalog*, if the keyword **Pool** appears on the command line, a *Pool* will be deleted, if the keyword **Object** appears on the command line, an Object and all of its associated records (File) will be deleted from the catalog, and if the keyword **Job** appears on the command line, a *Job* and all of its associated records (File and JobMedia) will be deleted from the catalog. The full form of this command is:

```
delete pool=<pool-name>
```

or

```
delete volume=<volume-name> pool=<pool-name>
```

or

```
delete JobId=<job-id> JobId=<job-id2> ...
```

or

```
delete JobId=n,m,o-r,t ...
```

or

```
delete client=<client-name>
```

or

```
delete snapshot
```

or

```
delete object [objectid=id [all]] [category=obj_cat] [type=obj_type]⏎
→[name=obj_name] [uuid=obj_uuid] [source=obj_source]
```

The first form deletes a *Pool* record from the catalog database. The second form deletes a *Volume* record from the specified pool in the *Catalog* database. The third form deletes the specified record from the catalog database. The fourth form deletes records for JobIds **n, m, o, p, q, r** and **t**. Where each one of the **n,m,...** is, of course, a number. That is a **delete jobid** accepts lists and ranges of jobids.

The deletion of a *Client* record will prune all Job records associated with the Client. This command is possible only once the *Client* resource of the given name is no longer defined in the Director configuration file.

The deletion of a Snapshot record can be done with the sixth command.

The deletion of an Object records can be done with the last command. Different filters can be used along with 'delete object' command: category, type, name, uuid or source. Passing specific object id can be also used as a filter to delete many objects at once. Following command will delete all objects records with the same category, type, name, uuid and source as object with given id.

```
delete object objectid=1 all
```

disable job<job-name>
> This command permits you to disable a for automatic scheduling. The job may have been previously enabled with the **Job Enabled** or using the console **enable** command. The next time the *Director* is restarted, the **Enable/Disable** state will be set to the value in the **Job** (default **enabled**) as defined in the **bacula-dir.conf** file.

disable jobs all
> This command permits you to disable all Jobs for automatic scheduling. The next time the *Director* is restarted, the **Enable/Disable** state will be set to the value in the *Job* resource (default **enabled**) as defined in the **bacula-dir.conf** file.

enable job<job-name>
> This command permits you to enable a for automatic scheduling. The job may have been previously disabled with the *Job* resource **Enabled** or using the console **disable** command. The next time the *Director* is restarted, the **Enable/Disable** state will be set to the value in the **Job** (default **enabled**) as defined in the **bacula-dir.conf** file.

enable jobs all

> This command permits you to enable all Jobs for automatic scheduling. It does not enable jobs which have **Disabled** directive. The next time the *Director* is restarted, the **Enable/Disable** state will be set to the value in the **Job** (default **enabled**) as defined in the **bacula-dir.conf** file.

estimate

> Using this command, you can get an idea how many files will be backed up, or if you are unsure about your Include statements in your *Fileset*, you can test them without doing an actual backup. The default is to assume a Full backup. However, you can override this by specifying a **level=Incremental** or **level=Differential** on the command line. A name must be specified or you will be prompted for one, and optionally a *Client* and *Fileset* may be specified on the command line. It then contacts the client which computes the number of files and bytes that would be backed up. Please note that this is an estimate calculated from the number of blocks in the file rather than by reading the actual bytes. As such, the estimated backup size will generally be larger than an actual backup.
>
> The **estimate** command can use the accurate code to detect changes and give a better estimation. You can set the accurate behavior on command line using **accurate=yes/no** or use the setting as default value.
>
> Optionally you may specify the keyword **listing** in which case, all the files to be backed up will be listed. Note, it could take quite some time to display them if the backup is large. The full form is:

```
estimate job=<job-name> listing client=<client-name> accurate=<yes/no>␣
↪fileset=<fileset-name> level=<level-name>
```

> Specification of the **job** is sufficient, but you can also override the **client**, **fileset**, **accurate** and/or **level** by specifying them on the **estimate** command line.
>
> As an example, you might do:

```
@tall /tmp/listing
estimate job=NightlySave listing level=Incremental
@tall
```

> which will do a full listing of all files to be backed up for the Job **NightlySave** during an Incremental save and put it in the file **/tmp/listing**. Note, the byte estimate provided by this command is based on the file size contained in the directory item. This can give wildly incorrect estimates of the actual storage used if there are sparse files on your systems. Sparse files are often found on 64 bit systems for certain system files. The size that is returned is the size **Bacula** will backup if the sparse option is not specified in the *Fileset*. There is currently no way to get an estimate of the real file size that would be found should the sparse option be enabled.

exit

> This command terminates the *Console* program.

gui

> Invoke the non-interactive gui mode.

```
gui [on|off]
```

help

> This command displays the list of commands available.

label

> This command is used to label physical volumes. The full form of this command is:

```
label storage=<storage-name> volume=<volume-name>
slot=<slot>
```

If you leave out any part, you will be prompted for it. The media type is automatically taken from the **Storage** definition that you supply. Once the necessary information is obtained, the *Console* program contacts the specified daemon and requests that the *Volume* be labeled. If the Volume labeling is successful, the Console program will create a *Volume* record in the appropriate *Pool*.

The Volume name is restricted to letters, numbers, and the special characters hyphen (-), underscore (_), colon (:), and period (.). All other characters including a space are invalid. This restriction is to ensure good readability of Volume names to reduce operator errors.

Please note, when labeling a blank tape, **Bacula** will get **read I/O error** when it attempts to ensure that the tape is not already labeled. If you wish to avoid getting these messages, please write an EOF mark on your tape before attempting to label it:

```
mt rewind
mt weof
```

The **label** command can fail for a number of reasons:

1. The *Volume* name you specify is already in the Volume database.

2. The daemon has a tape or other Volume already mounted on the device, in which case you must **unmount** the device, insert a blank tape, then do the **label** command.

3. The Volume in the device is already a **Bacula** labeled Volume. (**Bacula** will never relabel a **Bacula** labeled Volume unless it is recycled and you use the **relabel** command).

4. There is no Volume in the drive.

There are two ways to relabel a volume that already has a **Bacula** label. The brute force method is to write an end of file mark on the tape using the system **mt** program, something like the following:

```
mt -f /dev/st0 rewind
mt -f /dev/st0 weof
```

For a disk volume, you would manually delete the Volume.

Then you use the **label** command to add a new label. However, this could leave traces of the old volume in the catalog.

The preferable method to relabel a Volume is to first **purge** the volume, either automatically, or explicitly with the **purge** command, then use the **relabel** command described below.

If your autochanger has barcode labels, you can label all the Volumes in your autochanger one after another by using the **label barcodes** command. For each tape in the changer containing a barcode, **Bacula** will mount the tape and then label it with the same name as the barcode. An appropriate Media record will also be created in the catalog. Any barcode that begins with the same characters as specified on the " **Cleanprefix = xxx** " directive in the *Director*'s *Pool* resource, will be treated as a cleaning tape, and will not be labeled. However, an entry for the cleaning tape will be created in the catalog. For example with:

```
Pool {
  Name ...
  Cleaning Prefix = "CLN"
}
```

Copyright © 2025 Bacula Systems. All trademarks are the property of their respective owners.

Any slot containing a barcode of `CLNxxxx` will be treated as a cleaning tape and will not be mounted. Note, the full form of the command is:

```
label storage=xxx pool=yyy slots=1-5,10 barcodes
```

list

The **list** command lists the requested contents of the Catalog. The various fields of each record are listed on a single line. To see the complete list of options, use the `help list` command. The various forms of the list command are:

```
list events

list jobs

list jobid=<id>     (list jobid id)

list jobs joberrors    (list jobs with errors)

list jobs jobstatus=f     (list jobs with jobstatus f)

list jobs limit=10 order=desc   (list the last 10 jobs)

list jobs limit=10 order=asc    (list the first 10 jobs)

list jobs client=xxx     (list jobs for client xxx)

list copies    (list copy jobs)

list ujobid=<unique job name>   (list job with unique name)

list job=<job-name>   (list all jobs with "job-name")

list jobname=<job-name>   (same as above)

  In the above, you can add "limit=nn" to limit the output to nn jobs.

list joblog jobid=<id>     (list job output if recorded in the catalog)

list jobmedia

list jobmedia jobid=<id>

list jobmedia job=<job-name>

list fileevents jobid=<xx>   (list errors associated with a specific␣
↪JobId)

list files jobid=<id>

list files job=<job-name>

  In the above, you can add type=<all|deleted> to see all file records␣
↪or only deleted records.
```

```
list files type=<deleted|all> jobid=<id>

list pools

list clients

list jobtotals

list metadata type=email from=<str> cc=<str> subject=<str> client=<cli>␣
↪limit=<nb>

list metadata type=attachment id=<str>

list objects (list plugin objects)

list objects jobid=<id>

list object client=<client> (list plugin objects of specified client)

list object type=<type> (list plugin objects of specified type)

list objects category=<category> (list plugin objects of specified␣
↪category)

list objects limit=10 order=desc (list the last 10 objects)

list objects limit=10 order=asc (list the first 10 objects)

list volumes

list volumes jobid=<id>

list volumes pool=<pool-name>

list volumes job=<job-name>

list volume=<volume-name>

list nextvolume job=<job-name>

list nextvol job=<job-name>

list nextvol job=<job-name> days=nnn
```

What most of the above commands do should be more or less obvious. In general if you do not specify all the command line arguments, the command will prompt you for what is needed.

The **list nextvol** command will print the Volume name to be used by the specified job. You should be aware that exactly what Volume will be used depends on a lot of factors including the time and what a prior job will do. It may fill a tape that is not full when you issue this command. As a consequence, this command will give you a good estimate of what Volume will be used but not a definitive answer. In addition, this command may have certain side effect because it runs through

the same algorithm as a job, which means it may automatically purge or recycle a Volume. By default, the job specified must run within the next two days or no volume will be found. You can, however, use the **days=nnn** specification to specify up to 50 days. For example, if on Friday, you want to see what Volume will be needed on Monday, for job MyJob, you would use **list nextvol job=MyJob days=3**.

If you wish to add specialized commands that list the contents of the catalog, you can do so by adding them to the **query.sql** file. However, this takes some knowledge of programming *SQL*. Please see the **query** command below for additional information. See below for listing the full contents of a catalog record with the **llist** command.

As an example, the command **list pools** might produce the following output:

```
+------+---------+---------+---------+----------+------------+
| PoId | Name    | NumVols | MaxVols | PoolType | LabelFormat |
+------+---------+---------+---------+----------+------------+
| 1    | Default | 0       | 0       | Backup   | *          |
| 2    | Recycle | 0       | 8       | Backup   | File       |
+------+---------+---------+---------+----------+------------+
```

As mentioned above, the **list** command lists what is in the database. Some things are put into the database immediately when **Bacula** starts up, but in general, most things are put in only when they are first used, which is the case for a Client as with Job records, etc.

Bacula should create a client record in the database the first time you run a job for that client. Doing a **status** will not cause a database record to be created. The client database record will be created whether or not the job fails, but it must at least start. When the Client is actually contacted, additional info from the client will be added to the client record (a **uname -a** output).

If you want to see what *Client* resource you have available in your conf file, you use the Console command **show clients**.

llist

The **llist** or "long list" command takes all the same arguments that the **list** command described above does. The difference is that the **llist** command **list** the full contents of each database record selected. It does so by listing the various fields of the record vertically, with one field per line. It is possible to produce a very large number of output lines with this command.

If instead of the **list pools** as in the example above, you enter **llist pools** you might get the following output:

```
PoolId: 1
Name: Default
NumVols: 0
MaxVols: 0
UseOnce: 0
UseCatalog: 1
AcceptAnyVolume: 1
VolRetention: 1,296,000
VolUseDuration: 86,400
MaxVolJobs: 0
MaxVolBytes: 0
AutoPrune: 0
Recycle: 1
PoolType: Backup
LabelFormat: *
```

15

```
PoolId: 2
Name: Recycle
NumVols: 0
MaxVols: 8
UseOnce: 0
UseCatalog: 1
AcceptAnyVolume: 1
VolRetention: 3,600
VolUseDuration: 3,600
MaxVolJobs: 1
MaxVolBytes: 0
AutoPrune: 0
Recycle: 1
PoolType: Backup
LabelFormat: File
```

messages

> This command causes any pending console messages to be immediately displayed.

memory

> Print current memory usage.

mount

> The **mount** command is used to get **Bacula** to read a volume on a physical device. It is a way to tell **Bacula** that you have mounted a tape and that **Bacula** should examine the tape. This command is normally used only after there was no Volume in a drive and **Bacula** requests you to mount a new Volume or when you have specifically unmounted a Volume with the **unmount** console command, which causes **Bacula** to close the drive. If you have an autoloader, the **mount** command will not cause **Bacula** to operate the autoloader unless you specify a **slot** and possibly a **drive**. The various forms of the mount command are:

```
mount storage=<storage-name> [ slot=<num> ] [ drive=<num> ]
```

```
mount [ jobid=<id> | job=<job-name> ]
```

> If you have specified **Automatic Mount = yes** in the Storage daemon's **Device**, under most circumstances, **Bacula** will automatically access the Volume unless you have explicitly **unmount**ed it in the Console program.

prune

> The **prune** command allows you to safely remove expired Jobs, Files and Statistics database records from the Catalog. This command works only on the Catalog database and does not affect data written to Volumes. In all cases, the **prune** command adheres to retention times of the specified records. You can prune expired File entries, Job records or statistics from the database. Additionally, when you prune volumes, you prune both expired Job and File records from Volumes, once all the records pertinent to the volume have been pruned, the volume status will change to **Purged**.

```
prune files|stats client=<client-name> [yes]

prune jobs (all|client=<client-name>) [yes]
```

```
prune volume(=<volume-name>| allfrompool (pool=<pool-name>|allpools))␣
↪[yes]

prune all
```

For a Volume to be pruned, the **VolStatus** must be *Full*, *Used*, or *Append*, otherwise the pruning will not take place.

When pruning Jobs With the **all** keyword, all combinations of **Client/Pool** present in the *Job* table will be pruned. **prune jobs all yes**

purge

The **purge** command will delete associated Catalog database records from Jobs and Volumes without considering the retention period. **purge** works only on the Catalog database and does not affect data written to Volumes. This command can be dangerous because you can delete catalog records associated with current backups of files, and we recommend that you do not use it unless you know what you are doing. The permitted forms of **purge** are:

```
purge files jobid=<jobid>|job=<job-name>|client=<client-name>
```

```
purge jobs client=<client-name> (of all jobs)
```

```
purge volume|volume=<vol-name> (of all jobs)
```

For the **purge** command to work on Volume Catalog database records the **VolStatus** must be **Append**, **Full**, **Used**, or **Error.**

The actual data written to the Volume will be unaffected by this command unless you are using the **ActionOnPurge=Truncate** option on those Media.

To ask **Bacula** to truncate your Purged volumes, you need to use the following command in interactive mode or in a **RunScript**:

```
* purge volume action=truncate storage=File allpools

# or by default, action=all
* purge volume action storage=File pool=Default
```

This is possible to specify the volume name, the media type, the pool, the storage, etc... (see **help purge**) Be sure that your storage device is idle when you decide to run this command.

query

This command reads a predefined SQL query from the query file (the name and location of the query file is defined with the **QueryFile** record in the Director's configuration file). You are prompted to select a query from the file, and possibly enter one or more parameters, then the command is submitted to the Catalog database SQL engine.

The following queries are currently available (version 2.2.7):

```
Available queries:
1: List up to 20 places where a File is saved regardless of the directory
2: List where the most recent copies of a file are saved
3: List last 20 Full Backups for a Client
4: List all backups for a Client after a specified time
5: List all backups for a Client
```

```
6: List Volume Attributes for a selected Volume
7: List Volumes used by selected JobId
8: List Volumes to Restore All Files
9: List Pool Attributes for a selected Pool
10: List total files/bytes by Job
11: List total files/bytes by Volume
12: List Files for a selected JobId
13: List Jobs stored on a selected MediaId
14: List Jobs stored for a given Volume name
15: List Volumes Bacula thinks are in changer
16: List Volumes likely to need replacement from age or errors
Choose a query (1-16):
```

quit

This command terminates the console program. The console program sends the **quit** request to the Director and waits for acknowledgment. If the Director is busy doing a previous command for you that has not terminated, it may take some time. You may quit immediately by issuing the **.quit** command (i.e. quit preceded by a period).

relabel

This command is used to label physical volumes. The full form of this command is:

```
relabel storage=<storage-name> oldvolume=<old-volume-name> volume=
↪<newvolume-name> pool=<pool>
```

If you leave out any part, you will be prompted for it. In order for the Volume (old-volume-name) to be relabeled, it must be in the catalog, and the volume status must be marked **Purged** or **Recycle**. This happens automatically as a result of applying retention periods, or you may explicitly purge the volume using the **purge** command.

Once the volume is physically relabeled, the old data previously written on the Volume is lost and cannot be recovered.

release

This command is used to cause the Storage daemon to rewind (release) the current tape in the drive, and to re-read the Volume label the next time the tape is used.

```
release storage=<storage-name>
```

After a **release** command, the device is still kept open by **Bacula** (unless **Always Open** is set to **No** in the Storage Daemon's configuration) so it cannot be used by another program. However, with some tape drives, the operator can remove the current tape and to insert a different one, and when the next Job starts, **Bacula** will know to re-read the tape label to find out what tape is mounted. If you want to be able to use the drive with another program (e.g. **mt**), you must use the **unmount** command to cause **Bacula** to completely release (close) the device.

reload

The **reload** command causes the Director to re-read its configuration file and apply the new values. The new values will take effect immediately for all new jobs. However, if you change schedules, be aware that the scheduler pre-schedules jobs up to two hours in advance, so any changes that are to take place during the next two hours may be delayed. Jobs that have already been scheduled to run (i.e. surpassed their requested start time) will continue with the old values. New jobs will use the new values. Each time you issue a **reload** command while jobs are running, the old config values will keptf until all jobs that were running before issuing the reload terminate, at which time the old config values will be released from memory. As a default a maximum number of 32 reload

requests that can be made, which is generally sufficient. In the case that you make a very large number of reload requests, you may use the **Maximum Reload Requests** directive in the Director resource of **bacula-dir.conf** to set a larger maximum to that value you wish.

restart

The **restart** command allows console users to restart a canceled, failed, or incomplete Job. For canceled and failed Jobs, the Job will restart from the beginning. For incomplete Jobs the Job will restart at the point that it was stopped either by a stop command or by some recoverable failure.

The **restart** command, when entered in bconsole in this plain way, will create the following prompt:

```
*restart
You have the following choices:
    1: Incomplete
    2: Canceled
    3: Failed
    4: All
Select termination code: (1-4):
```

If you select the **All** option, you may see something like:

Select termination code: (1-4): 4

Table 1: bVerbatim

jo-bid	name	starttime	type	level	job-files	jobbytes	jobsta-tus
1	Incremen-tal	2012-03-26 12:15:21	B	F	0	0	A
2	Incremen-tal	2012-03-26 12:18:14	B	F	350	4,013,397	I
3	Incremen-tal	2012-03-26 12:18:30	B	F	0	0	A
4	Incremen-tal	2012-03-26 12:18:38	B	F	331	3,548,058	I

Enter the JobId list to select:

Then you may enter one or more JobIds to be restarted, which may take the form of a list of JobIds separated by commas, and/or JobId ranges such as **1-4**, which indicates you want to restart JobIds 1 through 4, inclusive.

The **restart** command accepts some parameters to simplify selection of the Job to restart, and to modify the behaviour of the restarted Job instance.

To Filter, you can use

- the keywords *failed*, *canceled* or *incomplete*

- the option *client=<name>*

- the option *job=<name>*

Note: The restart command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the restart

command will result in a new Job.

---

resume

The **resume** command does exactly the same thing as a **restart** command, but for some users the name may be more logical because in general the **restart** command is used to resume running a Job that was incomplete.

restore

The **restore** command allows you to select one or more Jobs (JobIds) to be restored using various methods. Once the JobIds are selected, the File records for those Jobs are placed in an internal **Bacula** directory tree, and the restore enters a file selection mode that allows you to interactively walk up and down the file tree selecting individual files to be restored.

```
restore storage=<storage-name> client=<backup-client-name> where=<path>
pool=<pool-name> fileset=<fileset-name> comment=<comment-specification>
restoreclient=<restore-client-name> objectid=<id> restorejob=<job-name>
jobuser=<user-name> jobgroup=<group-name>
noautoparent select current all done
```

Where **current**, if specified, tells the **restore** command to automatically select a restore to the most current backup. If not specified, you will be prompted. The **all** specification tells the **restore** command to restore all files. If it is not specified, you will be prompted for the files to restore.

The client keyword initially specifies the client from which the backup was made and the client to which the restore will be make. However, if the **restoreclient** keyword is specified, then the restore is written to that client.

The comment keyword can be used to store information related to the job in the catalog, like who has requested the job. It can have up to 4096 characters and cannot use the following characters : ' < > & \ "

The **noautoparent** keyword advises the restore preparation to not automatically select directory entries leading up to the explicitly marked files for restoration. Normally, such entries are selected and restored to ensure that proper permissions are recreated. However, in some cases this may not be needed or desirable, and thus, the functionality can be disabled here. Directories will still be created, but they will receive the default permissons; on Unix / Linux hosts, this is usually resulting in them being owned by root and having permissions set as defined by system defaults and `umask` modifier. This keyword is available as of Bacula Enterprise version 8.10.

The restore Job rarely needs to be specified, as bacula installations commonly only have a single restore job configured. However, for certain cases, such as a varying list of **RunScript** specifications, multiple restore jobs may be configured. The **restorejob** argument allows the selection of one of these jobs.

run

This command allows you to schedule jobs to be run immediately. The full form of the command is:

```
run job=<job-name> client=<client-name>
  fileset=<Fileset-name> level=<level-keyword>
  storage=<storage-name> where=<directory-prefix> comment=<comment-
↪specification>
  when=<universal-time-specification> spooldata=yes|no yes
```

The comment keyword can be used to store information related to the job in the catalog, like who has requested the job. It can have up to 4096 characters and cannot use the following characters : ' < > & \ "

Any information that is needed but not specified will be listed for selection, and before starting the job, you will be prompted to accept, reject, or modify the parameters of the job to be run, unless you have specified **yes**, in which case the job will be immediately sent to the scheduler.

On my system, when I enter a run command, I get the following prompt:

```
A job name must be specified.
The defined Job resources are:
  1: Matou
  2: Polymatou
  3: Rufus
  4: Minimatou
  5: Minou
  6: PmatouVerify
  7: MatouVerify
  8: RufusVerify
  9: Watchdog
Select Job resource (1-9):
```

If I then select number 5, I am prompted with:

```
Run Backup job
JobName: Minou
Fileset: Minou Full Set
Level: Incremental
Client: Minou
Storage: DLTDrive
Pool: Default
When: 2003-04-23 17:08:18
OK to run? (yes/mod/no):
```

If I now enter **yes**, the Job will be run. If I enter **mod**, I will be presented with the following prompt:

```
Parameters to modify:
  1: Level
  2: Storage
  3: Job
  4: Fileset
  5: Client
  6: When
  7: Pool
Select parameter to modify (1-7):
```

If you wish to start a job at a later time, you can do so by setting the When time. Use the **mod** option and select **When** (no. 6). Then enter the desired start time in `YYYY-MM-DD HH:MM:SS` format.

The **spooldata** argument of the run command cannot be modified through the menu and is only accessible by setting its value on the intial command line. If no spooldata flag is set, the job, storage or schedule flag is used.

setbandwidth

This command is used to limit the bandwidth of a running job or a client.

```
setbandwidth limit=<nb> [ jobid=<id> | client=<cli> ]
```

The limit can be provided as a plain number, which would then mean the number of bytes per second, or a multiplier from among (case-insensitive) k/s (1,024), kb/s (1,000), m/s (1,048,576) or mb/s (1,000,000) can be appended.

This bandwidth control is applied to data transfers from File Daemon to Storage Daemon, and it may not result in as smooth a network bandwidth usage as traffic shaping at the network layer, but it requires no external facilities.

setdebug

This command is used to set the debug level in each daemon. The form of this command is:

```
setdebug level=nn [trace=0/1 client=<client-name> | dir | director |
storage=<storage-name> | all | options=0cidtTlp | tags=<tags>]
```

If **trace=1** is set, then tracing will be enabled, and the daemon will be placed in trace mode, which means that all debug output as set by the debug level will be directed to the file **bacula.trace** in the current directory of the daemon. Normally, tracing is needed only for Win32 clients where the debug output cannot be written to a terminal or redirected to a file. When tracing, each debug output message is appended to the trace file. You must explicitly delete the file when you are done.

If **options** parameter is set, the following arguments can be used to control debug functions.

0 clear debug flags

i Turn off, ignore bwrite() errors on restore on File Daemon

d Turn off decomp of BackupRead() streams on File Daemon

t Turn on timestamp in traces

T Turn off timestamp in traces

c Truncate trace file if trace file is activated

l Turn on recoding events on P() and V()

p Turn on the display of the event ring when doing a lockdump

It is now possible to use *class* of debug messages called tags to control the debug output of **Bacula** daemons.

all Display all debug messages

bvfs Display BVFS debug messages

sql Display SQL related debug messages

memory Display memory and poolmem allocation messages

scheduler Display scheduler related debug messages

```
* setdebug level=10 tags=bvfs,sql,memory
* setdebug level=10 tags=!bvfs
```

The tags option is composed of a list of tags, tags are separated by "," or "+" or "-" or "!". To disable a specific tag, use "-" or "!" in front of the tag.

setip

Sets new client address — if authorized.

Copyright © 2025 Bacula Systems. All trademarks are the property of their respective owners.

A console is authorized to use the **SetIP** command only if it has a **Console** definition in both the Director and the Console. In addition, if the console name, provided on the **Name**, must be the same as a Client name, the user of that console is permitted to use the **SetIP** command to change the **Address** in the Director's *Client* resource to the IP address of the Console. This permits portables or other machines using DHCP (non-fixed IP addresses) to "notify" the Director of their current IP address.

show

The **show** command will list the records as defined in the Director's configuration file (normally **bacula-dir.conf**). This command is used mainly for debugging purposes by developers. The following keywords are accepted on the **show** command line: **catalogs**, **clients**, **counters**, **devices**, **directors filesets**, **jobs**, **messages**, **pools**, **schedules**, **storages**, **all**, **help**. Please don't confuse this command with the **list**, which displays the contents of the catalog.

sqlquery

The **sqlquery** command puts the Console program into SQL query mode where each line you enter is concatenated to the previous line until a semicolon (;) is seen. The semicolon terminates the command, which is then passed directly to the SQL database engine. When the output from the SQL engine is displayed, the formation of a new SQL command begins. To terminate SQL query mode and return to the Console command prompt, you enter a period (.) in column 1.

Using this command, you can query the SQL catalog database directly. Note you should really know what you are doing otherwise you could damage the catalog database. See the **query** command above for simpler and safer way of entering SQL queries.

Depending on what database engine you are using (*MySQL* or *PostgreSQL*), you will have somewhat different SQL commands available. For more detailed information, please refer to the *MySQL*, *PostgreSQL* documentation.

status

This command will display the status of all components. For the director, it will display the next jobs that are scheduled during the next 24 hours as well as the status of currently running jobs. For the Storage Daemon, you will have drive status or autochanger content. The File Daemon will give you information about current jobs like average speed or file accounting. The full form of this command is:

```
status [all | dir=<dir-name> | director [days=nnn] |
schedule [client=<client-name> | job=<job-name>] [days=nnn | limit=nnn]
client=<client-name> | [slots] storage=<storage-name>] | network [bytes=
↪<nb>]
```

If you do a **status dir**, the console will list any currently running jobs, a summary of all jobs scheduled to be run in the next 24 hours, and a listing of the last ten terminated jobs with their statuses. The scheduled jobs summary will include the Volume name to be used. You should be aware of two things: 1. to obtain the volume name, the code goes through the same code that will be used when the job runs, but it does not do pruning nor recycling of Volumes; 2. The Volume listed is at best a guess. The Volume actually used may be different because of the time difference (more durations may expire when the job runs) and another job could completely fill the Volume requiring a new one.

In the Running Jobs listing, you may find the following types of information:

```
2507 Catalog MatouVerify.2004-03-13-05.05.02 is waiting execution
5349 Full    CatalogBackup.2004-03-13-01.10.00 is waiting for higher↪
↪priority jobs to finish
5348 Differe Minou.2004-03-13-01.05.09 is waiting on max Storage jobs
```

(continues on next page)

```
5343 Full    Rufus.2004-03-13-01.05.04 is running
```

Looking at the above listing from bottom to top, obviously JobId 5343 (Rufus) is running. JobId 5348 (Minou) is waiting for JobId 5343 to finish because it is using the *Storage* resource, hence the "waiting on max Storage jobs". JobId 5349 has a lower priority than all the other jobs so it is waiting for higher priority jobs to finish, and finally, JobId 2507 (MatouVerify) is waiting because only one job can run at a time, hence it is simply "waiting execution"

If you do a **status dir**, it will by default list the first occurrence of all jobs that are scheduled today and tomorrow. If you wish to see the jobs that are scheduled in the next three days (e.g. on Friday you want to see the first occurrence of what tapes are scheduled to be used on Friday, the weekend, and Monday), you can add the **days=3** option. Note, a **days=0** shows the first occurrence of jobs scheduled today only. If you have multiple run statements, the first occurrence of each run statement for the job will be displayed for the period specified.

If your job seems to be blocked, you can get a general idea of the problem by doing a **status dir**, but you can most often get a much more specific indication of the problem looking into the File and Storage Daemon status.

The **status schedule** provides more insight into the scheduler's world view. It will present a list of scheduled jobs, starting with the current day, and including past jobs of the current day.

By default, the Jobs for 10 days will be presented. Also, by default, the output will be limited to 100 items. Using the **days** and **limit** named options, which each take a numerical value, the output can be shortened. The **client** and **job** keywords take Client and Job names, respectively, and can be used to filter the Jobs to list. As of Bacula Enterprise 8.10 these keywords can appear multiple times, but that filters of the same type will be or'ed together, whereas Client and Job filters would both have to match. In other words, as each Job runs by definition on exactly one Client, combining client and job filters is pointless.

An example:

```
status schedule client=vanyar-fd client=GolgiApparat-fd limit=8

Scheduled Jobs (30/8):
Level          Type    Pri  Scheduled          Job Name           ␣
↪Schedule
====================================================================================
Incremental    Backup   10  Tue 23-May 22:05   vanyar-psql        ␣
↪DailyPG
Incremental    Backup   10  Tue 23-May 23:05   GolgiApparat-all   Daily
Incremental    Backup   10  Tue 23-May 23:05   vanyar-all         ␣
↪DailyLC
Incremental    Backup   10  Wed 24-May 22:05   vanyar-psql        ␣
↪DailyPG
Incremental    Backup   10  Wed 24-May 23:05   GolgiApparat-all   Daily
Incremental    Backup   10  Wed 24-May 23:05   vanyar-all         ␣
↪DailyLC
Incremental    Backup   10  Thu 25-May 22:05   vanyar-psql        ␣
↪DailyPG
Incremental    Backup   10  Thu 25-May 23:05   GolgiApparat-all   Daily
====
```

If you enter **status storage**, **Bacula** will prompt you with a list of the storage resources. When you select one, the Storage daemon will be requested to do a **status**. However, note that the Storage

daemon will do a status of all the devices it has, and not just of the one you requested. In the current version of **Bacula**, when you enter the **status storage** command, it prompts you only with a subset of all the storage resources that the Director considers to be in different Storage daemons. In other words, it attempts to remove duplicate storage definitions. This can be a bit confusing at first, but can vastly simplify the promt listing if you have defined a large number of storage resources.

If you prefer to see the full list of all storage resources, simply add the keyword **select** to the command such as: **status select storage** and you will get a prompt that includes all storage resources even if they reference the same storage daemon.

If you enter **status network**, **Bacula** will prompt you with a list of the storage resources and a list of the client resources. Then, **Bacula** will test the network throughput between the two selected daemons.

```
*status network
The defined Client resources are:
  1: zog82-fd
  2: zog8-fd
Select Client (File daemon) resource (1-2): 2
Automatically selected Storage: File1
Connecting to Storage File1 at zog8:8103
Connecting to Client zog8-fd at zog8:8102
Running network test between Client=zog8-fd and Storage=File1 with 52.42␣
↪MB ...
2000 OK bytes=52428800 duration=48ms write-speed=1.088 GB/s 2000 OK␣
↪bytes=52428800 duration=56ms read-speed=933.8 MB/s
```

stop

The **stop** command is very similar to the **cancel** command with the main difference that the Job that is stopped is marked as Incomplete so that it can be restarted later by the *restart* command where it left off.

The JobId of the job to be stopped can be passed as a named parameter, such as `stop jobid=12345`.

The **stop** command with no arguments will prompt you with the list of running jobs allowing you to select one, which might look like the following:

```
*stop
Select Job:
    1: JobId=3 Job=Incremental.2012-03-26_12.04.26_07
    2: JobId=4 Job=Incremental.2012-03-26_12.04.30_08
    3: JobId=5 Job=Incremental.2012-03-26_12.04.36_09
Choose Job to stop (1-3): 2
2001 Job "Incremental.2012-03-26_12.04.30_08" marked to be stopped.
3000 JobId=4 Job="Incremental.2012-03-26_12.04.30_08" marked to be stopped.
```

tag

The **tag** command will add, delete or list the tags associated with catalog records such as Clients, Jobs, Volumes or Objects. The command accepts all parameters in command line.

```
*tag add name="#test1" client=zog8-fd
1000 Tag added

*tag
```

(continues on next page)

```
Available Tag operations:
  1: Add
  2: Delete
  3: List
Select Tag operation (1-3): 1
Available Tag target:
  1: Client
  2: Job
  3: Volume
Select Tag target (1-3): 1
Automatically selected Client: zog8-fd
Enter the Tag value: test1
1000 Tag added
*tag
Available Tag operations:
  1: Add
  2: Delete
  3: List
Select Tag operation (1-3): 3
Available Tag target:
  1: Client
  2: Job
  3: Volume
Select Tag target (1-3): 1
Automatically selected Client: zog8-fd
+--------+----------+---------+
| tag    | clientid | client  |
+--------+----------+---------+
| test1  |        1 | zog8-fd |
| #test1 |        1 | zog8-fd |
+--------+----------+---------+
```

time

Prints the current time.

trace

Turn on/off trace to file.

umount

For old-time Unix guys. See the **unmount** command for full details.

unmount

This command causes the indicated **Bacula** Storage daemon to unmount the specified device. The forms of the command are the same as the **mount** command:

```
unmount storage=<storage-name> [ drive=<num> ]
```

```
unmount [ jobid=<id> | job=<job-name> ]
```

Once you unmount a storage device, **Bacula** will no longer be able to use it until you issue a mount command for that device. If **Bacula** needs to access that device, it will block and issue mount requests periodically to the operator.

If the device you are unmounting is an autochanger, it will unload the drive you have specified on

the command line. If no drive is specified, it will assume drive 1.

update

> This command will update the catalog for either a specific Pool record, a Volume record, or the Slots in an autochanger with barcode capability. In the case of updating a Pool record, the new information will be automatically taken from the corresponding Director's configuration resource record. It can be used to increase the maximum number of volumes permitted or to set a maximum number of volumes. The following main keywords may be specified: **media**, **volume**, **pool**, **slots**, **stats**, **jobid**.
>
> In the case of updating a Job record, you can modify the **starttime**, the **client** and the **prune** attributes. **starttime** and **client** can be modified when doing a migration from an other backup software to Bacula for example. The **prune** attribute of the Job record is checked when trying to purge job or prune files from the catalog. If the **prune** Job's catalog attribute is 1 (Job record) or 2 (Files record), Bacula will not be able to purge the job record (or the file records) and recycle the associated volumes.
>
> In the case of updating a Volume, you will be prompted for which value you wish to change. The following Volume parameters may be changed:

```
Volume Status
Volume Retention Period
Volume Use Duration
Maximum Volume Jobs
Maximum Volume Files
Maximum Volume Bytes
Recycle Flag
Recycle Pool
Slot
InChanger Flag
Pool
Volume Files
Volume from Pool
All Volumes from
Pool All Volumes from all Pools
```

> For slots **update slots**, **Bacula** will obtain a list of slots and their barcodes from the Storage daemon, and for each barcode found, it will automatically update the slot in the catalog Media record to correspond to the new value. This is very useful if you have moved cassettes in the magazine, or if you have removed the magazine and inserted a different one. As the slot of each Volume is updated, the InChanger flag for that Volume will also be set, and any other Volumes in the Pool that were last mounted on the same Storage device will have their InChanger flag turned off. This permits **Bacula** to know what magazine (tape holder) is currently in the autochanger.
>
> If you do not have barcodes, you can accomplish the same thing in version 1.33 and later by using the **update slots scan** command. The **scan** keyword tells **Bacula** to physically mount each tape and to read its VolumeName.
>
> For Pool **update pool**, **Bacula** will move the Volume record from its existing pool to the pool specified.
>
> For **Volume from Pool**, **All Volumes from Pool** and **All Volumes from all Pools**, the following values are updated from the Pool record: Recycle, RecyclePool, VolRetention, VolUseDuration, MaxVolJobs, MaxVolFiles, and MaxVolBytes. (RecyclePool feature is available with **Bacula** 2.1.4 or higher.)
>
> The full form of the **update** command with all command line arguments is:

```
update volume=xxx pool=yyy slots volstatus=xxx VolRetention=ddd
VolUse=ddd MaxVolJobs=nnn MaxVolBytes=nnn Recycle=yes|no slot=nnn
enabled=n recyclepool=zzz actiononpurge=xxx
```

```
update volume=xxx frompool
```

```
update volume allfrompool=xxx
```

```
update volume fromallpools
```

use

> This command allows you to specify which Catalog database to use. Normally, you will be using only one database so this will be done automatically. In the case that you are using more than one database, you can use this command to switch from one to another.

```
use [catalog=name-of-catalog]
```

var

> This command takes a string or quoted string and does variable expansion on it the same way variable expansion is done on the **LabelFormat** string. Thus, for the most part, you can test your LabelFormat strings. The difference between the **var** command and the actual LabelFormat process is that during the var command, no job is running so "dummy" values are used in place of Job specific variables. Generally, however, you will get a good idea of what is going to happen in the real case.

version

> The command prints the Director's version.

wait

> The **wait** command causes the Director to pause until there are no jobs running. This command is useful in a batch situation such as regression testing where you wish to start a job and wait until that job completes before continuing. This command now has the following options:

```
wait [jobid=nn] [jobuid=unique id] [job=job name]
```

> If specified with a specific JobId, the **wait** command will wait for that particular job to terminate before continuing.

Special dot Commands

There is a list of commands that are prefixed with a period (.). These commands are intended to be used either by batch programs or graphical user interface front-ends. They are not normally used by interactive users. Once GUI development begins, this list will be considerably expanded. More information about these commands can be found in src/dird/ua_dotcmds.c of the Bacula Community Project (www.bacula.org). The following is the list of dot commands:

```
.api
.backups job=xxx                   list backups for specified job
.clients                           list all client names
.catalogs                          list all catalog defined
.defaults client=xxx fileset=yyy   list defaults for specified client
.die                               cause the Director to segfault (for␣
```

(continues on next page)

```
→debugging)
.dir                            when in tree mode prints the equivalent to␣
→the dir
                                command, but with fields separated by␣
→commas rather
                                than spaces.
.dump
.exit                           quit
.events                         list record custom events
.filesets                       list all fileset names
.help                           help command output
.jobs                           list all job names
.estimate                       estimate the size of the next job
.jlist                          list catalog objects in JSON format (see␣
→list command)
.levels                         list all levels
.messages                       get quick messages
.msgs                           return any queued messages
.pools                          list all pool names
.quit                           quit
.putfile                        upload a PluginRestore object to the␣
→director
.schedule                       list all schedule resources
.sql
.status                         get status output
.status dir header              get header status output
.status dir running             get running jobs status output
.status dir scheduled           get scheduled jobs status output
.status dir terminated          get terminated jobs status output
.storage                        return storage resource names
.volstatus                      list all possible volume statuses
.media                          list all media
.mediatypes                     list all defined mediatype
.locations
.actiononpurge                  list all possible values for ActionOnPurge␣
→setting
.bvfs_lsdirs                    list directories in a directory for a␣
→given set of jobs
.bvfs_lsfiles                   list files in a directory for a given set␣
→of jobs
.bvfs_get_volumes               list volumes needed for a restore
.bvfs_update                    update the bvfs cache tables
.bvfs_get_jobids                get jobids needed to restore a given job
.bvfs_get_jobs                  get job information
.bvfs_get_bootstrap             generate a bootstrap from a bvfs restore
.bvfs_get_fileindex             get the fileindex content for a given file
.bvfs_versions                  get all version of a file
.bvfs_get_delta                 get all delta parts of a file
.bvfs_restore                   generate a SQL table with all information␣
→needed for a
                                restore
.bvfs_cleanup                   cleanup the SQL restore table
```

```
.bvfs_decode_lstat              decode the LSTAT field
.bvfs_clear_cache               clear the BVFS cache of the catalog
.bvfs_update_fv                 compute extra statistics in the BVFS␣
↪tables (number of

                                files and size)
.bvfs_delete_fileid             delete a given file
.setuid                         restrict BVFS queries with UID/GID
.ls                             list files on a client

.types                          list job types
.query
.tags                           list tags
```

Special At (@) Commands

Normally, all commands entered to the Console program are immediately forwarded to the Director, which may be on another machine, to be executed. However, there is a small list of **at** commands, all beginning with an at character (@), that will not be sent to the Director, but rather interpreted by the Console program directly. Note, these commands are implemented only in the tty console program and not in the *BAT* Console. These commands are:

- **@input <filename>** Read and execute the commands contained in the file specified.

- **@output <filename> w/a** Send all following output to the filename specified either overwriting the file (w) or appending to the file (a). To redirect the output to the terminal, simply enter **@output** without a filename specification. WARNING: be careful not to overwrite a valid file. A typical example during a regression test might be:

```
@output /dev/null
commands ...
@output
```

- **@tee <filename>w/a** Send all subsequent output to both the specified file and the terminal. It is turned off by specifying **@tall**, **@tee** or **@output** without a filename.

- **@tall <filename>w/a** Send all subsequent input and output to both the specified file and the terminal. It is turned off by specifying **@tall**, **@tee** or **@output** without a filename.

- **@sleep <seconds>** Sleep the specified number of seconds.

- **@time** Print the current time and date.

- **@version** Print the console's version.

- **@quit** quit

- **@exit** quit

- **@# anything** Comment

- **@help** Get the list of all special @ commands.

- **@separator <char>** When using bconsole with readline, you can set the command separator to one of those characters to write commands who require multiple input on one line, or to put multiple commands on a single line.

```
!$%&'()*+,-/:;<>?[]^`{|}~
```

Note, if you use a semicolon (;) as a separator character, which is common, you will not be able to use the **sql** command, which requires each command to be terminated by a semicolon.

Running the Console from a Shell Script

You can automate many Console tasks by running the console program from a shell script. For example, if you have created a file containing the following commands:

```
./bconsole -c ./bconsole.conf <<END_OF_DATA
unmount storage=DDS-4
quit
END_OF_DATA
```

when that file is executed, it will unmount the current DDS-4 storage device. You might want to run this command during a Job by using the `RunBeforeJob` or `RunAfterJob` records.

It is also possible to run the Console program from file input where the file contains the commands as follows:

```
./bconsole -c ./bconsole.conf <filename
```

where the file named `filename` contains any set of console commands.

As a real example, the following script is part of the **Bacula** regression tests. It labels a volume (a disk volume), runs a backup, then does a restore of the files saved.

```
bin/bconsole -c bin/bconsole.conf <<END_OF_DATA
@output /dev/null
messages
@output /tmp/log1.out
label volume=TestVolume001
run job=Client1 yes
wait
messages
@#
@# now do a restore
@#
@output /tmp/log2.out
restore current all
yes
wait
messages
@output
quit
END_OF_DATA
```

The output from the backup is directed to `/tmp/log1.out` and the output from the restore is directed to `/tmp/log2.out`. To ensure that the backup and restore ran correctly, the output files are checked with:

```
grep "^ *Termination: *Backup OK" /tmp/log1.out
backupstat=$?
```

```
grep "^ *Termination: *Restore OK" /tmp/log2.out
restorestat=$?
```

Adding Volumes to a Pool

If you have used the **label** command to label a Volume, it will be automatically added to the Pool, and you will not need to add any media to the pool.

Alternatively, you may choose to add a number of Volumes to the pool without labeling them. At a later time when the Volume is requested by **Bacula** you will need to label it.

Before adding a volume, you must know the following information:

1. The name of the Pool (normally "Default")

2. The Media Type as specified in the `Storage` in the Director's configuration file (e.g. "DLT8000")

3. The number and names of the Volumes you wish to create.

For example, to add media to a Pool, you would issue the following commands to the console program:

```
*add
Enter name of Pool to add Volumes to: Default
Enter the Media Type: DLT8000
Enter number of Media volumes to create. Max=1000: 10
Enter base volume name: Save
Enter the starting number: 1
10 Volumes created in pool Default
*
```

To see what you have added, enter:

```
*list media pool=Default
+-------+----------+---------+---------+-------+-----------------+
| MedId | VolumeNa | MediaTyp| VolStat | Bytes | LastWritten     |
+-------+----------+---------+---------+-------+-----------------+
|    11 | Save0001 | DLT8000 | Append  |     0 | 0000-00-00 00:00 |
|    12 | Save0002 | DLT8000 | Append  |     0 | 0000-00-00 00:00 |
|    13 | Save0003 | DLT8000 | Append  |     0 | 0000-00-00 00:00 |
|    14 | Save0004 | DLT8000 | Append  |     0 | 0000-00-00 00:00 |
|    15 | Save0005 | DLT8000 | Append  |     0 | 0000-00-00 00:00 |
|    16 | Save0006 | DLT8000 | Append  |     0 | 0000-00-00 00:00 |
|    17 | Save0007 | DLT8000 | Append  |     0 | 0000-00-00 00:00 |
|    18 | Save0008 | DLT8000 | Append  |     0 | 0000-00-00 00:00 |
|    19 | Save0009 | DLT8000 | Append  |     0 | 0000-00-00 00:00 |
|    20 | Save0010 | DLT8000 | Append  |     0 | 0000-00-00 00:00 |
+-------+----------+---------+---------+-------+-----------------+
*
```

Notice that the console program automatically appended a number to the base Volume name that you specify (Save in this case). If you don't want it to append a number, you can simply answer 0 (zero) to the question "Enter number of Media volumes to create. Max=1000:", and in this case, it will create a single Volume with the exact name you specify.

2 Bweb Management Suite

Note: If you have upgraded to Bacula Enterprise 14 or later from a previous version, read the Introduction To BWeb New Layout with Bacula 14.0.

Note: To learn about BWeb authentication methods, click here.

Note: If you wish to set up a tape autochanger using Bweb, click here.

This document is intended to provide insight into the configurations and processes required to install Bweb Management Suite.

2.1 Introduction

BWeb™ Management Suite is a web application that permits simple and rapid monitoring and administration of Bacula Jobs in a multi-client production environment.

If your production involves a dozen machines, `bconsole` and/or `BAT` will be sufficient to administer your backups. A tool such as BWeb™ becomes essential in bigger environments.

Features

BMS implements all basic operational functionality for Bacula, and allows to configure all Director, Storage, FileDaemon and Console resources. The main features are:

- Run a new job with the possibility to change pre-defined values

- Follow Job progression in real-time (backup and restore — see figure 1). You have access to information like the current filename, the number of files seen, the backup size, logs and an estimation of the progression percent completion based on previous backup sizes

- Restarting a Job by clicking on a failed Job using all the previous Job values

- Possibility to interface with your wiki to simplify access to documentation and to recovery hints

- A restore screen with on screen navigation, which groups multiple different Filesets of a Client in a single tree structure, as shown in figure 2

- Summary view of failure and detection of missing Jobs to simplify daily monitoring of Jobs

- Schedule analysis as a calendar

- Possibility to create groups of Clients

- Visualization of pool usage (occupation) and volumes based on estimations (cf. figure 3 below and figure 4)

- Modification of Volume attributes (values)

- Automatic management of media locations (autochanger, off-site vault, . . .) including management of importing and exporting cartridges.

- Module for graphical statistics to optimize production management as presented in figure 6.
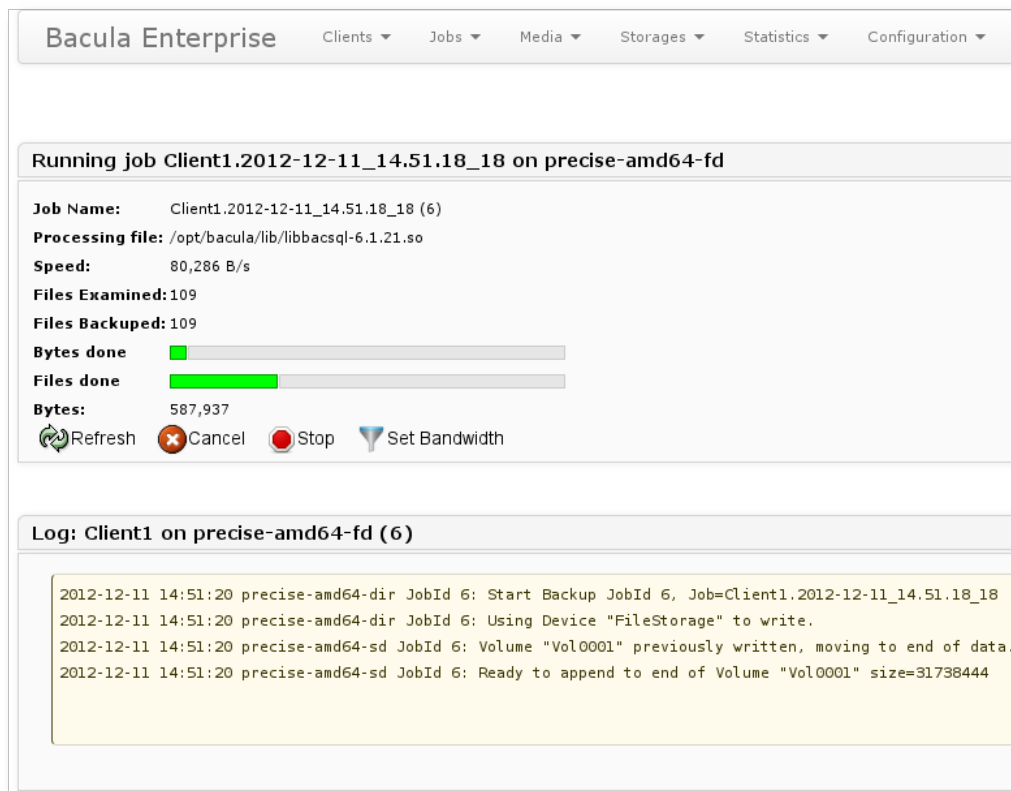
Fig. 1: Figure 1: Job progression in real time
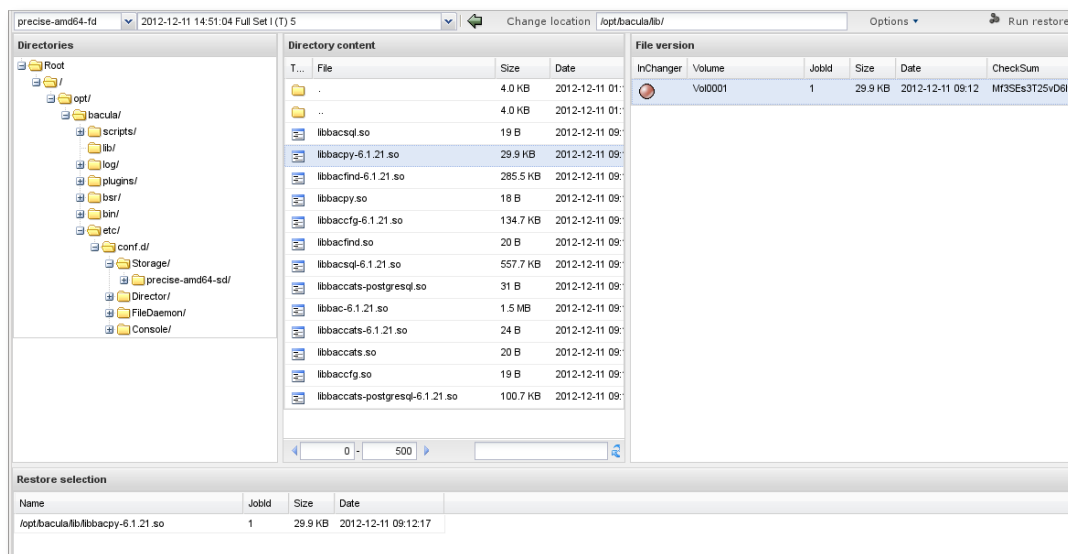
Fig. 2: Figure 2: Web Restore Interface
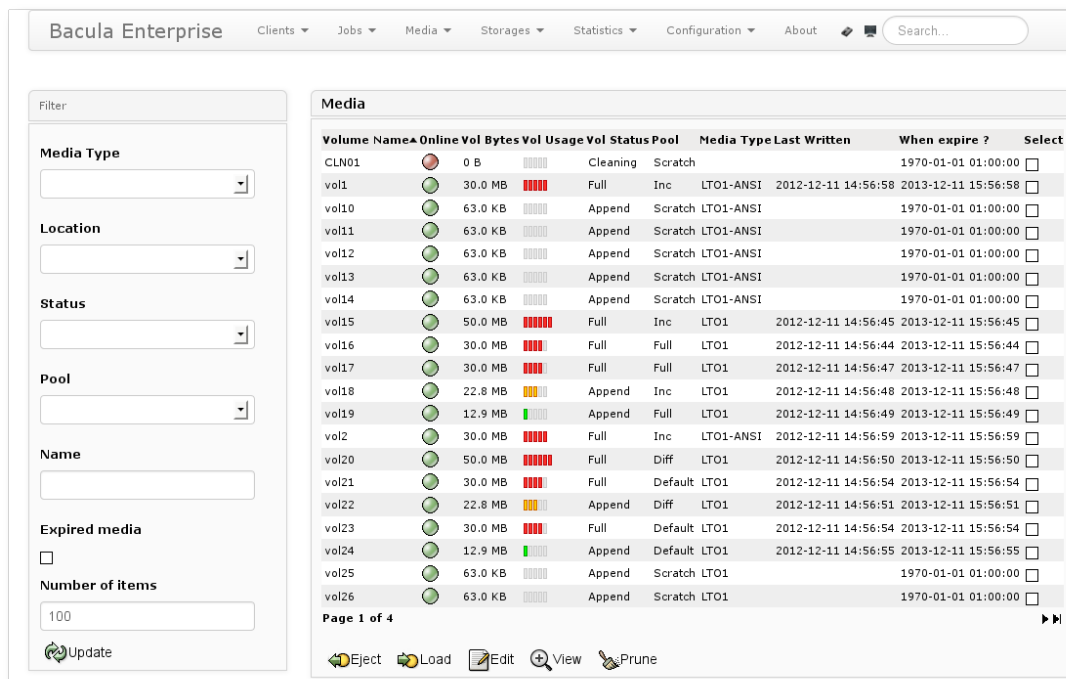
Fig. 3: Figure 3: Volume Information

- Visualization of your backup contents to help optimization of their content. You can rapidly navigate in backed up directories without affecting your production (See figure 7);

- Export of statistics to a spreadsheet for management review and invoicing by server groups.

- Library management (robot, autoloader, autochanger, etc.) with cartridge selection for import and export, visualization of the robot contents, management of the import/export tray, etc.

- Ability to create roles and access rights for groups of machines and for different users

- Advanced supervision module for monitoring backups

- BWeb core libraries for rapid development of Bacula scripts

- Works with both `PostgreSQL` and `MySQL`

BWeb Limitations

BWeb works with the latest version of Firefox and Chrome. In addition, you must be respect the following items:

- Media, Storage and Pools must have simple names that contain letters, numbers "-", "_", and "." (but no spaces or letters with accents)

- Your autochangers must have the same name in BWeb as in the Director's configuration file, and have an associated location.

- When managing multiple Directors, the media for each Director must have distinct (different) names.

- BWeb does not support multiple Catalogs in the same Director.

Fig. 4: Figure 4: Modification of a Volume

Fig. 5: Figure 5: Job log output
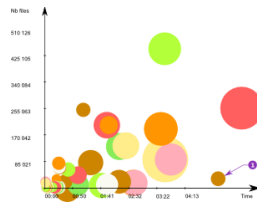
Fig. 6: Figure 6: Statistics Module

BWeb Configuration Module Limitations

If you do not intend to use the configuration module, nothing will be changed on your configuration files and you can skip this section.

In version 6.4 of the Bweb Configuration Module module, some advanced Bacula configurations are not supported.

- Dir Addresses
- Sd Addresses
- Fd Addresses
- Counter Resource

To edit the configuration with BWeb, BWeb should be installed on the same machine as the Director.

Comments

During the initialization of the BWeb configuration module, comments present in your configuration files will be striped out. You may keep some comments by using the `Description` directive.

```
# Job for John's Laptop, available from 6pm to 8am
Job {
  Name = Laptop-001
  ...
}
```

Can be adapted to:

```
Job {
  Name = Laptop-001
  Description = "John's laptop, available from 6pm to 8am"
  ...
}
```

Special Configuration Layout

If you are using @ in your configuration files to include sub-files, Bweb Configuration Module module will not use them anymore.

Once initialized, the configuration layout will have the following structure:

```
# find /opt/bacula/etc
bacula-dir.conf
conf.d/Director/
conf.d/Director/your-dir
conf.d/Director/your-dir/bacula-dir.conf
conf.d/Director/your-dir/JobDefs
conf.d/Director/your-dir/JobDefs/DefaultJob.cfg
conf.d/Director/your-dir/Messages
conf.d/Director/your-dir/Messages/Daemon.cfg
conf.d/Director/your-dir/Messages/Standard.cfg
...
```

This special configuration layout will imply some changes on how resources will be displayed in `bconsole` menus. Resources will be displayed in alphanumeric order instead of configuration file definition order. For more information, see Changes in bconsole.

Alternatives

Several other Open Source interfaces exist for Bacula

- `bat` is a graphical administration interface that gives access to most of Bacula's functionality. However, it does not contain the reporting modules found in BWeb.

- `bacula-web` is a simple PHP interface that gives a static view of terminated Jobs.

2.2 Installation

In the following chapter, we are presenting a few ways to install Bweb. It is possible to install Bweb using Bacula Installation Manager, which is recommended.

The advantages of using BIM are the following:

- resolves all dependencies

- enables SSL by default

- enables user login by default.

The alternative way is to install with packages.

Bweb Installation with Bacula Installation Manager

For Bweb to be successfully installed, the installation of the Director is needed.

See the article on Bacula Enterprise installation with BIM for Linux or Bacula Enterprise installation with BIM for Windows.

Installation with Packages

The following chapter presents the information on how to install Bweb with specific packages.

Ubuntu/Debian

Bacula Systems provides BWeb packages for Debian and Ubuntu. You can use dpkg/apt to install them:

```
# dpkg -i bacula-enterprise-bweb-extrajs_6.4-1_all.deb
# dpkg -i bacula-enterprise-bweb_8.8.4-1_all.deb
#
# apt-get install -f      # fix dependencies
```

Or you can use `apt-get` (in this example, you need to replace @CUST@ with your private FRS URL, the @PLATFORM@ with your platform and the version with the one matching your core **Bacula Enterprise Edition** product):

```
deb https://www.baculasystems.com/dl/@CUST@/debs/bin/8.8.4/@PLATFORM@-64
↪@PLATFORM@ main
deb https://www.baculasystems.com/dl/@CUST@/debs/bweb/8.8.4/@PLATFORM@-64
↪@PLATFORM@ bweb
```

```
# apt-get install bacula-enterprise-bweb
```

Load BWeb Specific Functions to Bacula Catalog

On BWeb 6.4 and later, you can load BWeb specific SQL functions using the following script:

```
root# /opt/bweb/bin/install_bweb.sh
```

If your catalog is already configured, the script will print a lot of messages indicating that some "Objects already exist". Everything should be OK.

If the script is unable to load BWeb specific functions to the catalog with a message such as "Permission denied" or "Unable to contact database server", contact Bacula Systems support team to get help.

If your BWeb version doesn't have the `install_bweb.sh` script, you can use the following procedure.

In some versions of PostgreSQL, you need to add the `plpgsql` language to the `bacula` database using the `postgres` super-user.

```
# su - postgres
postgres# createlang plpgsql -Upostgres -d bacula
postgres# exit
```

Then, you need to add BWeb specific tables to your Bacula catalog.

```
# su – bacula
bacula# psql -U bacula -f /opt/bweb/etc/bweb-postgresql.sql bacula
```

Or with MySQL

```
# mysql -u bacula bacula < /opt/bweb/etc/bweb-mysql.sql
```

Note that the suggested steps may not be applicable to all environments; for example, the `bacula` user may not be allowed to log in – knowledge of system administration may help, and **Bacula Systems** support is ready to assist if needed.

Start BWeb Service

By default, BWeb is using a dedicated service that you can start with

```
# /etc/init.d/bweb start
```

We strongly advise you to secure the default configuration file `/opt/bweb/etc/httpd.conf` by setting credentials and/or `https` navigation.

BWeb should be accessible through http://yourhost:9180/bweb/. The next step is to configure the DBI SQL Connection to match the `Catalog` resource of your `bacula-dir.conf` file.

```
DBI:      DBI:Pg:database=bacula;host=127.0.0.1
user:     bacula
password: xxxxxxxxxx
```

Or

```
DBI:      DBI:mysql:database=bacula;host=127.0.0.1
user:     bacula
password: xxxxxxxxxx
```

Naturally, if the catalog database is hosted on a different machine, the host will have to be changed accordingly.

SuSE Enterprise Linux 11/12

You can install Perl dependencies through the SuSE channel, Bacula Systems repositories or via CPAN (Cf. section Dependencies).

You can enable the Bacula Systems Perl dependencies repository by configuring the following YaST configuration file on SuSE:

```
# cat /etc/zypp/repos.d/Bacula_Systems_DAG.repo
[Bacula_Systems_DAG]
name=Bacula Systems DAG
enabled=1
autorefresh=1
baseurl=https://www.baculasystems.com/dl/DAG/@PLATFORM@
path=/
type=rpm-md
keeppackages=0
```

Where @PLATFORM@ can be:

- sles112

You can use `yum` (in this example, you need to replace @CUSTOMER@ with your private FRS url, @PLATFORM@ with your platform and the version with the one matching your core **Bacula Enterprise Edition** product):

```
[root@sles112-64 ~]# more /etc/zypp/repos.d/Bacula_Systems_bin.repo
[Bacula_Systems_bin]
name=Bacula Systems bin
enabled=1
autorefresh=1
baseurl=https://www.baculasystems.com/dl/@CUSTOMER@/rpms/bin/@PLATFORM@/6.6.6
path=/
type=rpm-md
keeppackages=0

[root@sles112-64 ~]# more /etc/zypp/repos.d/Bacula_Systems_bweb.repo
[Bacula_Systems_bweb]
name=Bacula Systems bweb
enabled=1
autorefresh=1
baseurl=https://www.baculasystems.com/dl/@CUSTOMER@/rpms/bweb/@PLATFORM@/6.6.6
path=/
type=rpm-md
keeppackages=0

[root@sles112-64 ~]# more /etc/zypp/repos.d/Bacula_Systems_perl.repo
[Bacula_Systems_perl]
name=Bacula Systems perl
enabled=1
autorefresh=1
baseurl=https://www.baculasystems.com/dl/DAG/@PLATFORM@/
path=/
type=rpm-md
keeppackages=0
```

```
# yast -i bacula-enterprise-bweb
```

Or you can install all dependencies manually using:

```
# yast -i perl-DBI perl-DBD-mysql perl-DBD-Pg perl-GDGraph \
    perl-HTML-Template perl-Time-modules gd perl-Date-Calc      \
    perl-Bit-Vector perl-Expect perl-GD  \
    perl-GD-Text-Util perl-IO-Tty perl-CGI-Session perl-JSON   \
    perl-JSON-XS lighttpd
```

Bacula Systems provides BWeb packages for SuSE Enterprise Linux. You can download them and use `rpm` to install them:

```
# rpm -ivh bacula-enterprise-bweb_8.8.4.rpm
```

Depending on your installation, you may need to change the default `bconsole` program and `bconsole.conf` file permissions to let the `bacula` user account access them. (For a different `bconsole` setup, see
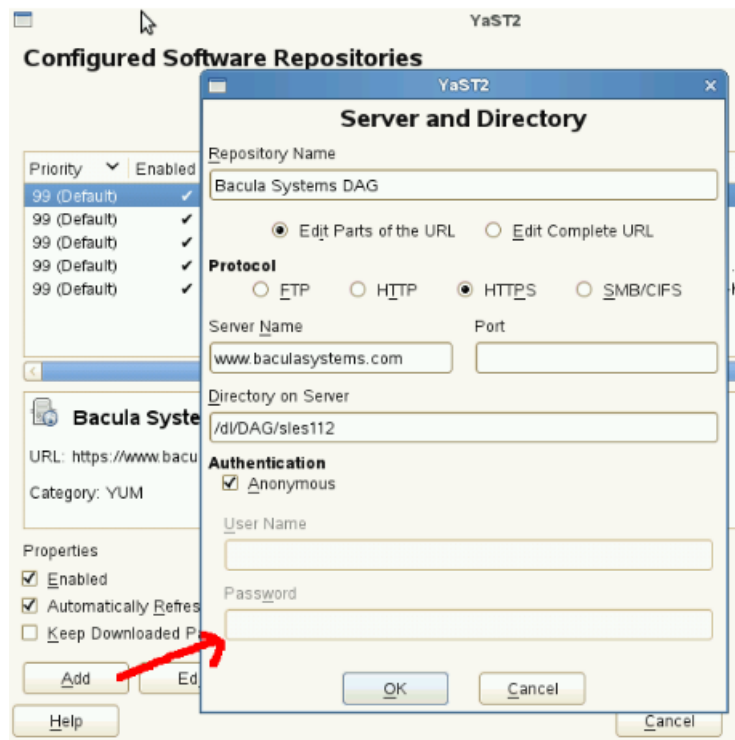
Fig. 7: Figure 9: YaST Repository for Perl

bconsole Configuration).

On SLES, it is required to add the `/etc/sudoers.d` to the `sudo` configuration to restart bacula services through BWeb.

```
# mkdir /etc/sudoers.d
# chown 750 /etc/sudoers.d
# echo '#includedir /etc/sudoers.d' >> /etc/sudoers
```

Configure PostgreSQL Authentication

The default PostgreSQL authentication method on SuSE Enterprise Linux is `ident`. This method will **perfectly work** with the `bacula-dir` and the BWeb daemon running as the `bacula` user. Then you can skip this section and come back later if you want to configure password authentication.

In the file pg_hba.conf, the first record with a matching connection type, client address, requested database, and user name is used to perform authentication. There is no "fall-through" or "backup": if one record is chosen and the authentication fails, subsequent records are not considered. If no record matches, access is denied.

However, you will need to add a password authentication to let another user connect to the catalog or connect to the catalog via TCP/IP.

Be careful, the order of each line is important in pg_hba.conf, make sure you read the documentation included on the top of the file before making any change.

Fig. 8: Figure 10: YaST Repository Example

```
postgres# cat /var/lib/pgsql/data/pg_hba.conf
# TYPE    DATABASE    USER    ADDRESS METHOD

# Let postgres user connect everywhere
local    all            postgres        ident

local    bacula        bacula          md5
local    all            all             ident

host    bacula        bacula  127.0.0.1/32 md5
...

postgres# psql
# ALTER USER bacula PASSWORD 'thisisapassword';


# using ident
postgres# psql -U bacula bacula
Password: ******

bacula=>

# using host
postgres# psql -U bacula -h localhost bacula
Password: ******

bacula=>
```

Load BWeb Specific Functions to Bacula Catalog

In BWeb 6.4 and later, you can load BWeb specific SQL functions using the following script:

```
root# /opt/bweb/bin/install_bweb.sh
```

If your BWeb version doesn't have the `install_bweb.sh` script, you can use the following procedure.

In some versions of PostgreSQL, you need to add the `plpgsql` language to the `bacula` database using the `postgres` super-user.

```
# su - postgres
postgres# createlang plpgsql -Upostgres -d bacula
```

You then need to add BWeb specific tables to your **Bacula Enterprise Edition** catalog.

```
# su - postgres
postgres# psql -U bacula -f /opt/bweb/etc/bweb-postgresql.sql bacula
```

With a MySQL catalog, you need to update the catalog with the following command:

```
# mysql -u bacula bacula < /opt/bweb/etc/bweb-mysql.sql
```

Copyright © 2025 Bacula Systems. All trademarks are the property of their respective owners.

IPtables and SELinux

SELinux and/or IPtables might need some configuration to run on your system, if you want to quickly see if BWeb is properly installed, you need to disable SELinux and IPtables.

To disable IPtables up to the next reboot:

```
# /etc/init.d/iptables stop
```

To configure IPtables permanently, you need to configure your system with a command such as:

```
# iptables -A INPUT -p tcp --tcp-port 9180 -j ACCEPT
# /etc/init.d/iptables save
```

To disable SELinux up to the next reboot:

```
echo 0 >/selinux/enforce
```

To disable SELinux permanently, you need to configure the SELINUX parameter in `/etc/selinux/config`.

Start BWeb Service

By default, BWeb is using a dedicated service that you can start with

```
# /etc/init.d/bweb start
```

We strongly advise you to secure the default configuration file `/opt/bweb/etc/httpd.conf` by setting credentials and/or `https` navigation. Detailed instructions are available at the bottom of the configuration file.

BWeb should be accessible through http://yourhost:9180/bweb/

If you want to install BWeb from sources, read Installation from Source.

Online Documentation

BWeb can display documentation about directives and resources, the documentation package can be found on your download area in the `docs` directory. Just untar it in the `html` directory, `/opt/bweb/html` by default.

RHEL

To install Perl dependencies, you can install them through RHEL Server Optional Chanel or via CPAN (Cf. Dependencies) or you can use RPMForge source and add that repository to your system by installing the rpmforge-release package following these instructions http://dag.wieers.com/rpm/FAQ.php#B.

If you don't want to enable the entire RPMForge repository, for your convenience, you can also refer to the **Bacula Systems** DAG repository by configuring the following yum configuration file on RHEL:

```
# cat /etc/yum.repos.d/baculasystems-dag.repo
[dag]
name = Bacula Enterprise Bweb - dag
baseurl = https://www.baculasystems.com/dl/DAG/@PLATFORM@
enabled = 1
protect = 0
gpgcheck = 0
```

Where @PLATFORM@ can be:

- rhel8-64

- rhel9-64

You can use yum (in this example, you need to replace @CUSTOMER@ with your private FRS url, @PLATFORM@ with your platform and the version with the one matching your core **Bacula Enterprise Edition** product):

```
[root@redhat6-64 ~]# more /etc/yum.repos.d/baculasystems.repo
[baculasystems]
name = Red Hat Enterprise  - baculasystems
baseurl = https://www.baculasystems.com/dl/@CUSTOMER@/rpms/bin/8.8.4/
↪@PLATFORM@
enabled = 1
protect = 0
gpgcheck = 0

[baculasystems-bweb]
name = Red Hat Enterprise  - baculasystems bweb plugin
baseurl = https://www.baculasystems.com/dl/@CUSTOMER@/rpms/bweb/8.8.4/
↪@PLATFORM@
enabled = 1
protect = 0
gpgcheck = 0

[baculasystems-dag]
name = Red Hat Enterprise  - baculasystems DAG
baseurl = https://www.baculasystems.com/dl/DAG/@PLATFORM@
enabled = 1
protect = 0
gpgcheck = 0
```

```
# yum install bacula-enterprise-bweb
```

Or you can install all dependencies manually using:

```
# yum install perl-DBI perl-DBD-mysql perl-DBD-Pg perl-GDGraph \
   perl-HTML-Template perl-Time-modules gd perl-Date-Calc      \
   perl-Bit-Vector perl-Expect perl-GD  \
   perl-GD-Text-Util perl-IO-Tty perl-CGI-Session perl-JSON    \
   perl-JSON-XS lighttpd
```

Bacula Systems provides BWeb packages for RHEL. You can download them and use rpm to install them:

```
# rpm -ivh bacula-enterprise-bweb_8.8.4.rpm
```

Depending on your installation, you may need to change the default `bconsole` program and `bconsole.conf` file permissions to let the `bacula` user account access them. (For a different `bconsole` setup, see bconsole Configuration). The RHEL package is using a copy of `bconsole`'s files in `/opt/bweb/bin`.

Configure PostgreSQL Authentication

The default PostgreSQL authentication method on RHEL is `ident`. This method will **perfectly work** with the `bacula-dir` and the BWeb daemon running as the `bacula` user. Then you can skip this section and come back later if you want to configure password authentication.

In the file pg_hba.conf, the first record with a matching connection type, client address, requested database, and user name is used to perform authentication. There is no "fall-through" or "backup": if one record is chosen and the authentication fails, subsequent records are not considered. If no record matches, access is denied.

However, you will need to add a password authentication to let another user connect to the catalog or connect to the catalog via TCP/IP.

Be careful, the order of each line is important in pg_hba.conf, make sure you read the documentation included on the top of the file before making any change.

```
postgres# cat /var/lib/pgsql/data/pg_hba.conf
# TYPE    DATABASE    USER    ADDRESS METHOD

# Let postgres user connect everywhere
local    all          postgres        ident

local    bacula       bacula          md5
local    all          all             ident

host     bacula       bacula   127.0.0.1/32 md5
...

postgres# psql
# ALTER USER bacula PASSWORD 'thisisapassword';


# using ident
postgres# psql -U bacula bacula
Password: ******

bacula=>

# using host
postgres# psql -U bacula -h localhost bacula
Password: ******

bacula=>
```

Load BWeb Specific Functions to Bacula Catalog

In BWeb 6.4 and later, you can load BWeb specific SQL functions using the following script:

```
root# /opt/bweb/bin/install_bweb.sh
```

If your BWeb version doesn't have the `install_bweb.sh` script, you can use the following procedure.

In some versions of PostgreSQL, you need to add the `plpgsql` language to the `bacula` database using the `postgres` super-user.

```
# su - postgres
postgres# createlang plpgsql -Upostgres -d bacula
```

You then need to add BWeb specific tables to your **Bacula Enterprise Edition** catalog.

```
 # su - postgres
 postgres# psql -U bacula -f /opt/bweb/etc/bweb-postgresql.sql bacula

With a MySQL catalog, you need to update the catalog with the
```

following command:

```
# mysql -u bacula bacula < /opt/bweb/etc/bweb-mysql.sql
```

IPtables and SELinux

SELinux and/or IPtables might need some configuration to run on your system, if you want to quickly see if BWeb is properly installed, you need to disable SELinux and IPtables.

To disable IPtables up to the next reboot:

```
# /etc/init.d/iptables stop
```

To configure IPtables permanently, you need to configure your system with a command such as:

```
# iptables -A INPUT -p tcp --tcp-port 9180 -j ACCEPT
# /etc/init.d/iptables save
```

To disable SELinux up to the next reboot:

```
echo 0 >/selinux/enforce
```

To disable SELinux permanently, you need to configure the SELINUX parameter in `/etc/selinux/config`.

Start BWeb Service

By default, BWeb is using a dedicated service that you can start with

```
# /etc/init.d/bweb start
```

We strongly advise you to secure the default configuration file `/opt/bweb/etc/httpd.conf` by setting credentials and/or `https` navigation. Detailed instructions are available at the bottom of the configuration file.

BWeb should be accessible through http://yourhost:9180/bweb/

If you want to install BWeb from sources, please read the appendix **app:install-from-sources**.

Online Documentation

BWeb can display documentation about directives and resources, the documentation package can be found on your download area in the `docs` directory. Just untar it in the `html` directory, `/opt/bweb/html` by default.

Link with the Director

Network Access

The following ports should be accessible between the server that hosts BWeb and other Bacula Enterprise components. All protocols can use SSL/TLS or, for `ssh` access, are encrypted anyway.

Table 2: Table 1: Network access

Component	Source	Destination	Default port
SQL (PostgreSQL)	BWeb	Catalog	5432/tcp
SQL (MySQL)	BWeb	Catalog	3306/tcp
Command Link	BWeb	Director	9101/tcp
Command Link (optional)	BWeb	Storage	22/tcp
Admin Web Access	Admin Workstation	BWeb	80,443/tcp

SQL Catalog Access

[sec:dbi] BWeb uses the Perl DBI layer to query **Bacula**'s catalog. You should configure the connection string in the `bweb.conf` file or in the web **configuration** panel (If your web server can modify the configuration file).

There is no standard for the text following the driver name. Each driver is free to use whatever syntax it wants. You must consult the documentation for the drivers you are using for a description of the syntax they require.

For example, if your PostgreSQL catalog `baculaprod` is accessible through the network on `192.168.1.2` port 5432/tcp, you will use:

```
DBI:Pg:database=baculaprod;host=192.168.1.2;port=5432
```

For a local MySQL catalog `baculaprod`, you will use:

Fig. 9: Figure 11: Connection parameters

```
DBI::mysql:database=baculaprod
```

Table 3: Table 2: Typical DBI/DSN parameters

Parameters	Example	Informations
database	database=bacula	Database name
host	host=bacula-dir	SQL server name
port	port=5432	Port of the connection to the server
sslmode	sslmode=require	Use SSL for Postgres SQL connection
mysql_ssl	mysql_ssl=1	Use SSL for MySQL SQL connection

bconsole Configuration

BWeb communicates with the Director through `bconsole`. It requires that your web server has permission to execute the command and read the associated configuration file `bconsole.conf`. When using Bacula Systems default configuration, the web server is running as "bacula", so everything should work out of the box.

When running BWeb through a custom apache configuration, the web server is running as "www-data" or "apache". It means that **in this configuration, `Apache` will not allow to communicate with Bacula without manual changes**.

```
# cd /opt/bacula
# ls -l bin/bconsole etc/bconsole.conf
-rwxr-xr-x 1 root root 843227 2008-11-01 16:18 bin/bconsole
-rw-r----- 1 root bacula 171 2008-11-04 18:48 etc/bconsole.conf
```

If you don't want to give this kind of permission, you can add the web server user to the `bacula` group or use a copy of the `bconsole.conf` file with restricted permissions.

```
# cp /opt/bacula/etc/bconsole.conf /opt/bacula/etc/bconsole-web.conf
# ls -l etc/bconsole.conf etc/bconsole-web.conf
-rw-rw---- 1 root bacula 171 2008-11-04 18:48 etc/bconsole.conf
-rw-r----- 1 root www-data 171 2008-11-04 18:48 etc/bconsole-web.conf
```

Note that we provide configuration files to run `Apache` or `Lighttpd` with the `bacula` Unix user. So, if the `bacula` user can read configuration files and execute binaries, BWeb should read them too.

Director Configuration

To be able to see Job logs in BWeb, you must add the catalog as a destination in all **Messages** resources. See the "catalog =" line below:

```
Messages {
 Name = Standard

 operator = root@localhost = mount
 console = all, !skipped, !saved
 append = "/opt/bacula/working/log" = all, !skipped

 catalog = all, !skipped, !saved
}
```

bweb.conf Configuration

By default, BWeb assumes its configuration file is `/opt/bweb/etc/bweb.conf` like described on figure 12.

If your web server doesn't have access to write to the configuration file, it will display the configuration in your browser and you may manually copy and paste what is displayed to `/opt/bweb/etc/bweb.conf`. To use another location for this file, you must to edit the `$config_file` variable in `Bweb.pm`.

```
# cat /opt/bweb/etc/bweb.conf
$VAR1 = bless( {
 'bconsole' => '/opt/bacula/bin/bconsole -n -c /opt/bacula/etc/bconsole-web.
↪conf',
 'wiki_url' => 'wiki.localdomain',
 'stat_job_table' => 'JobHisto',

 'template_dir' => '/opt/bweb/tpl',
 'lang' => 'en',

 'dbi' => 'DBI:Pg:database=bacula',
 'password' => 'xxx',
 'user' => 'bacula',

 'debug' => 0,
 'email_media' => 'admin@localhost'
}, 'Bweb::Config' );
```

For more information on BWeb options, read Configuration Options.

Fig. 10: Figure 12: BWeb parameters

Catalog Update

BWeb defines several internal tables that are used to create groups of machines, or provide instant navigation during restorations. An SQL script is used to declare these tables in the schema catalog.

PostgreSQL

The PL procedure language must be available in your Postgres cluster. You can install it with the administrator account `postgres`:

```
postgres:~$ createlang -d bacula plpgsql
```

After this operation, you can load the BWeb tables. To avoid permission problems, you should use the `bacula` account as defined in your database.

```
$ psql -U bacula -f script/bweb-postgresql.sql bacula
```

Mysql

The following command permits loading BWeb tables and functions into your catalog:

```
$ mysql -u bacula bacula < script/bweb-mysql.sql
```

Using an Autochanger

BWeb permits graphically administering your autochangers using `mtx`.

Bacula Configuration

Your autochanger as well as its drives, if there is more than one, must be defined in the Director's configuration file. Our example will use an LTO3 changer with three drives: `LTO3_0`, `LTO3_1`, `LTO3_2`. All these three drives will be defined in three different Storage resources in the Director configuration file.

```
Storage {
 Name = LTO3
 ...
 Autochanger = LTO3
}

Storage {
 Name = LTO3_0
 ...
 Autochanger = LTO3
}

Storage {
 Name = LTO3_1
 ...
```

(continues on next page)

```
 Autochanger = LTO3
}

Storage {
 Name = LTO3_2
 ...
 Autochanger = LTO3
}
```

Hardware Access

BWeb doesn't use **Bacula** to do physical operations on media, you have to configure BWeb to be able to use system commands. Under normal circumstances, the web users can't access drives or an autochanger directly. It is possible to configure `sudo` with `visudo` for that purpose.

```
# grep mtx /etc/sudoers
www-data ALL = (root) NOPASSWD: /usr/sbin/mtx -f /dev/changer transfer *
www-data ALL = (root) NOPASSWD: /usr/sbin/mtx -f /dev/changer status
www-data ALL = (root) NOPASSWD: /usr/sbin/mtx -f /dev/changer load *
www-data ALL = (root) NOPASSWD: /usr/sbin/mtx -f /dev/changer unload *
```

If you are using `bacula` account with `lighttpd` to run the web server, `bacula` should be a member of the `tape` group, and you don't need sudo configuration.

Remote Autochanger

If you want to use a remote Storage Daemon, you can use `ssh` to remotely access your autochanger from your web server. You will need to configure an ssh key exchange between your web user and an authorized user that can use the hardware (directly or with the `sudo` command). You will have to configure `ssh` as pre-command under BWeb (see figure 14). An example could be to use `ssh bacula@storage1`.

BWeb Configuration

Before you begin using BWeb, you must define a location with the same name as your autochanger (Cf. Managing Volume Locations). Through the Setup menu, you will be able to declare your autochangers in BWeb as shown in figure 13.
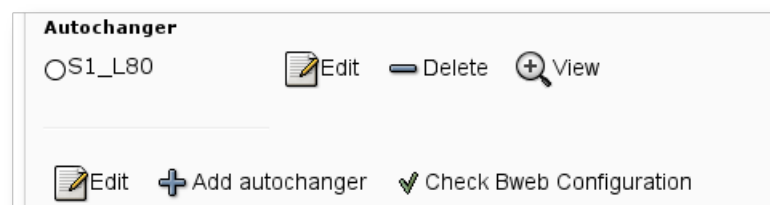
Fig. 11: Figure 13: Add/Modify an autochanger

To be able to work, BWeb has to know your autochanger configuration, especially the drive order — **Drive Index** in the Storage Daemon configuration — see figure 14.

Fig. 12: Figure 14: Definition of an autochanger

Analyze Backup Content (bfileview)

The **bfileview** module uses temporary space to store files used in its screens. This directory must be accessible for writing by your web user and must be mapped through `/bweb/fv/` in the web server configuration. The directory size should not grow to much, but you can configure automatic purging of its elements after few days.

```
# mkdir -p -m 750 /opt/bweb/spool
# chown bacula /opt/bweb/spool
```

Security Access

The web server is responsible for the user authentication, it is possible to use all the features of `lighttpd` (LDAP, PAM, etc.) to identify users and secure access to BWeb. It is necessary to edit the web server configuration to enable the authentication method wanted. The examples in this manual use a simple authentication via `htaccess` file. For the previous example, you could use the following command to configure `/opt/bweb/etc/htpasswd.bweb`.

```
# htpasswd -c /opt/bweb/etc/htpasswd.bweb admin
New password:
Re-type new password:

# htpasswd /opt/bweb/etc/htpasswd.bweb scott
New password:
Re-type new password:
```

If your system doesn't provide the `htpasswd` tool (included in `apache2-utils` package for example), you can also use `openssl` to generate the password file.

```
# echo admin:`openssl passwd -crypt myPass` | tee -a htpasswd.bweb
admin:X1w5cpQOvhfWo
```

Note that the user `admin` is defined by default. This user ID has full privileges and cannot be disabled.

VMware Integration

See the following article to enable VMware integration.

Maintenance Tasks

Statistics

The BWeb statistics module permits using the long term history feature of **Bacula**. This functionality allows to keep track of all jobs run even after them being purged from the catalog. You should schedule the `update stats days=3` command everyday to get that information. For example, you can add a **RunScript** to the Backup Catalog job.

```
Job {
 Name = "BackupCatalog"
```

```
...
 RunScript {
 Console = "update stats days=3"
 Runs on Client = no
 Runs when = After
 }
}
```

BWeb Folder Cache

To get instant restore navigation in all folders from your backups, BWeb uses a cache that has to be computed for each new Job. You can schedule this process to a calm period outside of your backup window, or do them automatically when you decide to restore something (the computing time depends on the number of files).

```
Job {
 name = "BWEB_CACHE"
 type = Admin
 RunScript {
 RunsWhen = Before
 RunsOnClient = no
 Command = "sh -c \"echo .bvfs_update | /opt/bacula/bin/bconsole\""
 }
 JobDefs = DefaultJob
 Schedule = BWEB_CACHE
}
```

Language Selection

The interface language can be configured globally or per user. Note that **Bacula**'s logs stay in their original language.

Verification

To verify that all dependencies are satisfied, you can execute `bweb.pl` manually.

```
$ cd /opt/bweb/cgi
$ ./bweb.pl
<html>
<body>
...
```

To verify all steps of your installation, you can click on the "Check Bweb Configuration" link provided with the BWeb package under "Configure Bweb" menu (see figure 15).

Checking Catalog Configuration

Status	Parameter	Current Setting	Test	Description
⚠	log_checkpoints	off ()	ne 'on'	Logs each checkpoint.
⚠	checkpoint_segments	3 ()	< 16	Sets the maximum distance in log segments between automatic WAL checkpoints.
⚠	checkpoint_timeout	300 (s)	< 1200	Sets the maximum time between automatic WAL checkpoints.
⚠	effective_cache_size	16384 (8kB)	< 131072	Sets the planner's assumption about the size of the disk cache.
⚠	work_mem	1024 (kB)	< 32768	Sets the maximum memory to be used for query workspaces.
⚠	maintenance_work_mem	16384 (kB)	< 102400	Sets the maximum memory to be used for maintenance operations.
⚠	shared_buffers	4096 (8kB)	< 131072	Sets the number of shared memory buffers used by the server.

Testing bweb configuration : bweb_testconfig

▼ Command output

```
Content-Type: text/plain

1..36
ok 1 - require CGI;
ok 2 - require Bweb;
ok 3 - require Bconsole;
ok 4 - require CCircle;
ok 5 - require GBalloon;
ok 6 - require GTime;
ok 7 - not root account
```
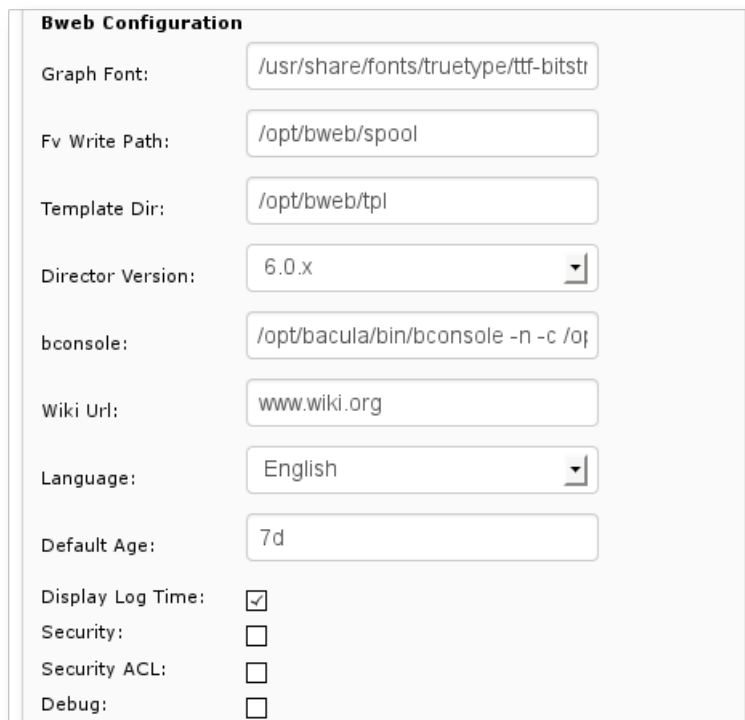
Fig. 13: Figure 15: Configuration Checks
root# mysql -u bacula bacula < upgrade-4.0_6.0_mysql.sql

2.3 BWeb Configuration

The following document aims at presenting the information on how to configure BWeb.

Configuration Options

Fig. 14: Figure 16: BWeb Parameters

Table 4: Table 3: Configuration Option Descriptions

Option	Description
email_me	This is the default email address used when modifying Volume locations
graph_fo	Allows setting the type face used in the graphics. Note that some GD libraries do not support ttf fonts.
bconsole	Full path and arguments for the bconsole program
fv_write	Temp directory for results from the **bfileview**. This directory must be writable by the web user and placed under /bweb/fv relative to the web server.
stat_job	This option (default JobHisto) specifies the name of the history table
wiki_url	The URL of your internal corporate documentation Wiki, if any (look here)
template	Path specifying the BWeb templates directory (look here)
language	The default language used in BWeb (the exact keyword in the configuration file is lang). See template_dir
default_	Default time period used by BWeb modules. E.g: 12h, 1d, 7d,…
default_	Default number of items displayed when displaying a list (ex: Job History, Media list,…)
rows_per	Default number of items displayed in a list (default to 20) (Available in version 6.6.6 and later)
display_	Determines whether or not the dates are shown when displaying logs
security	Turns on multi-user access. Note, you must login with the web server to access BWeb once this function is turned on. The admin userid is defined by default (look here)
security	Permits allowing access to machine groups when multi-user mode is enabled. This option permits restricting what view and what modules can be used by users depending on their defined profile (look here)
config_d	Path specifying where BWeb will store the Bacula configuration. This directory should be owned by the Bacula unix user.
ssl_dir	Path specifying where BWeb will store the Bacula SSL certificate. This directory should be owned by the Bacula unix user.
html_dir	Unix path specifying where BWeb can find html elements (/bweb).
workset_	Unix path specifying where BWeb can store Bacula configuration during edition.

Internal Backup Configuration Documentation

BWeb is designed to be configurable for optimum integration into your particular backup environment. Many companies internally use a documentation portal (web pages) to maintain the documentation and configuration of their servers. If the wiki_url option is defined, a link to the given URL will be displayed on the backup report pages. For example by using:
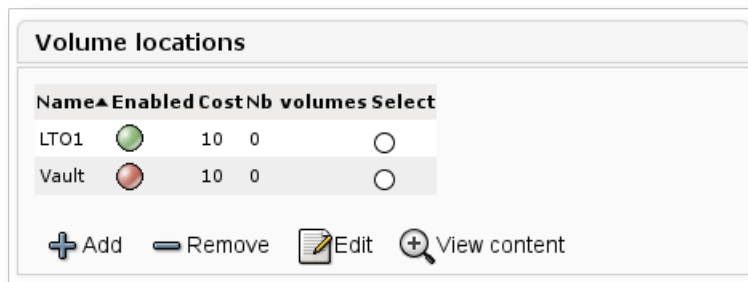
```
wiki_url=http://fr.wikipedia.org/wiki/
```

a page concerning a Job on server client1 will display a link to http://fr.wikipedia.org/wiki/client1 and thus allow accessing this page directly from the reports.

It is possible to modify the BWeb template for more complex links.

Managing Volume Locations
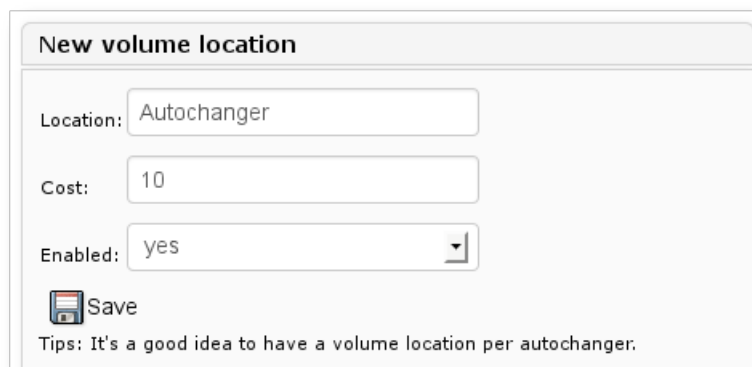
Management of Volume locations is an optional feature of BWeb. It permits you to specify Volume locations and track Volume movements. In the Media menu → Location, BWeb permits management of the list of Volume locations where Volumes can be found. When using the BWeb Volume import/export screens (Cf. Exporting Volumes and Multiple Sites), BWeb automatically updates the Volume location.

Fig. 15: Figure 17: Volume Locations

Fig. 16: Figure 18: Add a Volume Location

You should create Volume Locations for each Autochanger.

BWeb Groups

BWeb enables you to organize your Bacula clients into administrative groups. For example, you could define groups of geographic Volume locations, by business divisions, by the server owners, by application type, etc.

Thus it is simpler to supervise the proper execution of your Job. BWeb has several screens particularly adapted to sites with a very large number of File daemons (see following screenshots). Using Jobs → Job Overview (see figure 20 and figure 21), it is easy to view backup problems organized on a time scale. When was the last time the backup on this server failed, what are the variables referenced in this incident, "*"

Groups are also useful in statistical calculations, for example you can create a dashboard or allocate infrastructure costs by group and by resources used.

Group summary (last 24 hours limited to 100)								
Group▼	**Nb Jobs**	**Nb Ok**	**Nb Err**	**Nb Files**	**Size**	**Duration**	**Errors**	**Status**
Windows	108	108	0	675648	7.1 GB	18:00:00	1	⚠
Test	216	216	0	1238814	13.0 GB	36:00:00	1	⚠
Prod	90	90	0	563040	5.9 GB	15:00:00	0	✓
Linux	234	234	0	1351422	14.2 GB	39:00:00	0	✓
Japan	108	108	0	563814	5.9 GB	18:00:00	0	✓
Europe	54	54	0	337824	3.5 GB	09:00:00	0	✓
America	72	72	0	450432	4.7 GB	12:00:00	0	✓
Admin	90	90	0	563040	5.9 GB	15:00:00	0	✓

Fig. 17: Figure 19: Group View

Jobs overview (since 2009-01-15 and during 7 days)							
	15	**16**	**17**	**18**	**19**	**20**	**21**
Admin	✓	✓	✓	✗	✓	✓	⚠
America	✓	✓	✓	✓	✓	✓	✗
Europe	✓	✓	✓	✗	✓	✓	⚠
Japan	✓	✓	✓	✓	✓	✓	✗
Linux	✓	✓	✓	✓	✓	✓	⚠
Prod	✓	✓	✓	✓	✓	✓	⚠
Test	✓	✓	✓	✓	✓	✓	⚠
Windows	✓	✓	✓	✓	✓	✓	⚠

Fig. 18: Figure 20: Daily recap by group

Fig. 19: Figure 21: Daily recap by client

It is also possible to associate groups with user profiles to spread BWeb responsibilities within your organization (Cf. Management of Profiles and Users)

Adding, modification, and deletion of groups is done with the Clients screen → Groups (see figures 22 and 23). The **View others** button permits listing the clients that are not in any group.

Fig. 20: Figure 22: Groups configuration

Management of Profiles and Users

To access user management found in Configuration → Users (see figures 24 and 25), you must active the security function in the general BWeb configuration (see the section *Self User Restore*). Note, if you activate the security function without having configured `Lighttpd` authentication, you will not be able to access BWeb. In such a case, you must manually deactivate the security feature by modifying the file `/opt/bweb/etc/bweb.conf`.

To activate access control by user profile, check the `security_acl` option in the general BWeb configuration.

By default in BWeb the user `admin` has full privileges that cannot be disabled. If you experience configuration problems, you can use the `admin` account to reconfigure BWeb.
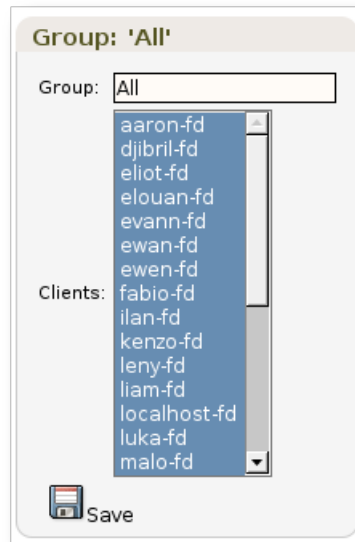
Fig. 21: Figure 23: "All members" group

Fig. 22: Figure 24: Users management

Fig. 23: Figure 25: User configuration

Restrict Bacula Resources

The `bconsole` BWeb configuration option can use `%u` to specify a specific bconsole configuration file. Then, it is possible to use restricted Consoles and fine tune permissions for BWeb. Note that in this case, it is mandatory to create a `bconsole` configuration file for the default `admin` user.

```
bconsole = /opt/bacula/bin/bconsole -n -c /opt/bacula/etc/bconsole-%u.conf
```

```
# cat /opt/bacula/etc/bconsole-admin.conf
Director {
  Name = "ks82-dir"
  Address = "ks82"
  DirPort = 9101
  Password = "bacula"
}

 # cat /opt/bacula/etc/bconsole-user1.conf
Director {
  Name = "ks82-dir"
  Address = "ks82"
  DirPort = 9101
  Password = "bacula"
}
Console {
 Name = user1
 Password = "bacula"
 Director = "ks82-dir"
}

 # cat /opt/bacula/etc/bacula-dir.conf
Console {
  Name = "user1"
  CatalogAcl = *all*
  ClientAcl = aclient-fd, otherclient-fd
  CommandAcl = *all*
  FilesetAcl = *all*
  JobAcl = home,usr
  Password = "bacula"
  PoolAcl = *all*
  ScheduleAcl = *all*
  StorageAcl = *all*
  WhereAcl = *all*
  DirectoryAcl = *all*
  UserIdAcl = *all*
}
```

Copyright © 2025 Bacula Systems. All trademarks are the property of their respective owners.

User Password Management

The following `lighttpd` configuration permits limiting access with basic authentication. After changing the lighttpd config file, you need to restart the bweb service:

```
# more /opt/bweb/etc/httpd.conf
...
server.modules += ( "mod_auth")       # if not already done

auth.backend = "htpasswd"
auth.backend.htpasswd.userfile =  "/opt/bweb/etc/htpasswd.bweb"
auth.require = ( "/" =>
        (
        "method" => "basic",
        "realm" => "Password protected area",
        "require" => "valid-user"
        )
   )
...
```

The command `htpasswd` enables you to manage the list of users that can authenticate.

```
# htpasswd -c /opt/bweb/etc/htpasswd.bweb admin
New password:
Re-type new password:
Adding password for user admin
```

Note: To add more users using the htpasswd command, you should remove the `-c` option:

```
# htpasswd /opt/bweb/etc/htpasswd.bweb demouser
```

When creating `Lighttpd` authentication, the user names given must correspond to the user names configured in BWeb. For example if your LDAP user name is `admin-windows`, the BWeb user name you configured must also be `admin-windows`.

See Security Access for more information about user password management.

Note, if you use the basic authentication and you are going to register new Bacula Clients by using QR code or registration link, you need to uncomment in `lighttpd` configuration the following two lines:

```
## Uncomment this line if you use client registration by QR code or
↪registration link
#$HTTP["url"] !~ "^/cgi-bin/bweb/register.pl$" {
```

```
## Uncomment this line if you use client registration by QR code or
↪registration link
#}
```

This way users, who register clients by QR code or registration link, will not be asked about login/password.

System Auth

The option `System authentication`[1] enables BWeb to authenticate users with the PAM subsystem via the `su` command. With the `System Authentication` option, only users defined on the Bacula Linux/Unix server will be allowed to use BWeb. With this option, if a user "admin" is defined on the system, it will be mapped automatically to the BWeb "admin" special account.

Permissions

Table 5: Table 4: User permissions

Permission	Description
`r_user_mgnt`	This permission is required to use the user management screen to view or modify users permissions.
`r_group_mgnt`	The permission is required for management of Bacula client groups.
`r_configure`	All configure screens require this permission. The Bacula configuration part of BWeb requires this permission.
`r_autochanger_m`	Management of autochangers requires this permission.
`r_location_mgnt`	Screens managing adding or modification of Volume locations require this permission.
`r_media_mgnt`	Media management is required to update the media properties and load/unload media.
`r_storage_mgnt`	Screens managing adding or modification of storage and devices require this permission.
`r_delete_job`	If this permission is set, you can prohibit or authorize Job and Job history deletion.
`r_prune`	Required to achieve the `prune` command.
`r_purge`	Required to achieve the `purge` command.
`r_view_job`	This permission allows the user to view jobs.
`r_view_log`	This permission allows the user to view the job logs.
`r_view_media`	Required to view the media.
`r_view_stat`	Required to display the statistics.
`r_view_group`	Required to view the groups.
`r_view_running_`	This permission is required to be able to follow real-time Job execution.
`r_run_job`	This permission is required to start Jobs via BWeb.
`r_restore_job`	This permission is required to start restore Jobs via BWeb.
`r_restart_job`	This permission is required to restart Jobs that were stopped or failed into an incomplete state.
`r_cancel_job`	This permission is required to prohibit or authorize Job cancellation.
`r_client_status`	This permission allows the user to obtain the client status.
`r_stop_job`	This permission enables to stop the Job.
`r_cancel_job`	This permission enables to cancel the Job.
`r_client_setbwl`	This permission allows the client bandwidth limit setting.
`r_bconsole`	Required to run the bconsole commands via the web bconsole.

[1] Available in version 8.8

Self User Restore

The option `Self User Restore`[2], used in conjunction with the `Security` or the `System authentication` options, enables BWeb to display a restricted view to users who are not defined in the BWeb system.

For example, If a Linux/Unix user called "john" is logged on his Linux/Unix workstation "john-laptop" and he points his FireFox web browser to BWeb, once logged into the BWeb system as "john", BWeb will search in the Bacula Director Client list for a system that matches the IP address of the workstation connected (probably "john-laptop-fd" in this case) and will restrict the restore process to this particular client automatically. If the Unix user information for "john" is available via the getpwnam() system interface, the restore process will automatically limit the files that can be viewed and restored to the files that the Unix user "john" can normally see or access.

The IP of the workstation can be found in the BWeb access log configured in lighttpd (`/opt/bacula/working/bweb-access.log`).

In addition, it is possible to use other Console ACLs such as `DirectoryAcl` or `WhereAcl` to apply even more access restrictions.

On Windows clients, it is not possible to automatically limit what a user can see. Here, the `DirectoryACL` directive must be used to limit the view and the restore capabilities of the user.

2.4 Configuring Bacula with BMS

This section describes the Bacula Configuration GUI module that is integrated in BMS 6.2.x and later. It allows you to configure all the Bacula configuration files (`bacula-dir.conf`, `bacula-sd.conf`, `bacula-fd.conf`, and `bconsole.conf`).
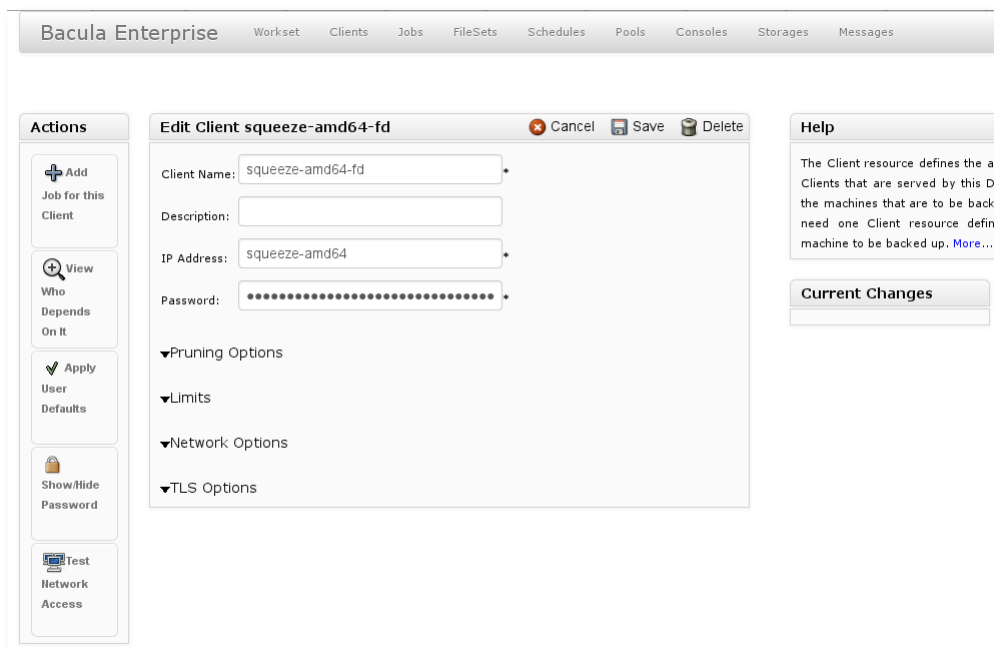
Fig. 24: Figure 26: BWeb configurator: Resource edition

---

[2] Available in version 8.8

General Considerations

Before using the Bacula Configuration GUI module, there are a few things that you should take into consideration.

- The configuration module is capable of creating, modifying, and saving the above mentioned Bacula .conf files.

- You may either start from scratch — with the default configuration files — or start with existing configuration files.

- If you start with existing configuration files, the configuration module will read them in, and keep them in its own internal format. This format is probably different from the original configuration file layout that you have. When the files are written out, they will be rearranged into a format that is compatible with the Bacula daemons but slightly different from any prior configuration files. For example, each resource is kept in a separate file in a directory of resources, and any Bacula configuration comments are not kept in the process. Depending on how your Fileset contents are organized, Bacula may force a new full backup after the BMS initialization. See Important Note About Fileset for more details.

- Once the new configuration files are written, you may modify them either with the configuration GUI or manually or using any scripts you may have developed.

Getting Started

If you are upgrading from BWeb 6.0.x or 4.0.x, you need to configure the new BWeb configuration items (see figure 27).

Fig. 25: Figure 27: BWeb configurator: Configuration parameters

To configure Bacula Enterprise using BWeb, click on the top menu Configuration → Configure Bacula.

The first time you will use the configuration module of BWeb, BWeb will present you a wizard that will guide you during the initial configuration phase. This wizard will display all Bacula Enterprise component detected on the local system.
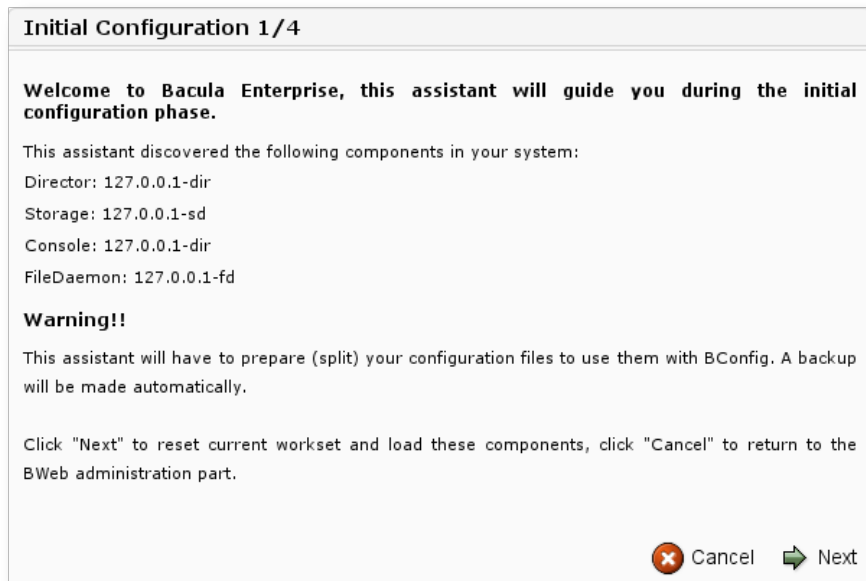
Fig. 26: Figure 28: BWeb configurator: Configuration wizard, step 1

Configuration File Layout

The BWeb configuration module needs to use a specific Bacula Enterprise configuration file layout. Instead of using a single file containing all resources — which is the default—, BWeb puts each resource in a single file, and groups resources by type in subdirectories. Note that all files included — with @ — will be automatically included in this conversion, and will not be used by Bacula afterward.

```
/opt/bacula/etc/bacula-dir.conf
```

Will be reorganized as:

```
  /opt/bacula/etc/bacula-dir.conf
  /opt/bacula/etc/conf.d/Director/host-dir/
  /opt/bacula/etc/conf.d/Director/host-dir/Client/
  /opt/bacula/etc/conf.d/Director/host-dir/Job/
  /opt/bacula/etc/conf.d/Director/host-dir/Pool/
...
```

A backup will be made as "`bacula-dir.conf.old.xxx`" and you can revert to your previous configuration by renaming this file to "`bacula-dir.conf`".

Important Note About Comments

Important, when converting your configuration file to BWeb specific layout, the process will strip out all your comments from the configuration files. If you want to keep some of your comments, feel free to use the `Description` before doing the conversion.

```
Job {
  Name = VeryImportant    # Job for the CEO
  Client = ceo-fd
...
}
```

Can be converted to

```
Job {
  Name = VeryImportant
  Description = "Job for the CEO"
  Client = ceo-fd
...
}
```

Important Note About Fileset

When BMS will prepare, i.e. "split", your configuration files to use the new configuration layout format, `Filesets` will be rewritten, and depending on the original organization, it may lead to some layout changes.

```
Fileset {
  Name = xxx
  Exclude {
   File = /tmp
   File = /mnt
  }
  Include {
   File = /etc
   File = /home
  }
}
```

Will be converted to:

```
Fileset {
  Name = xxx
  Include {
   File = /etc
   File = /home
  }
  Exclude {
   File = /tmp
   File = /mnt
  }
}
```

Note the inversion between Exclude and Include blocks.

Even if this layout change will not change the backup content, the Director may detect a change in the Fileset and may upgrade Incremntal or Differential jobs to Full jobs.

To avoid this problem, you can use the "IgnoreFilesetChanges = yes" option in your Filesets and remove it after a full backup cycle. If you want to remove this option as soon as possible, you can update Job records in the catalog with the help of the Bacula Systems support team.

Understanding Workset

When using BWeb, the configuration layout permits to locate a resource very efficiently, for example, the definition of a Pool "Default" will be stored in the following file inside the config_dir:

```
/opt/bacula/etc/conf.d/Director/yourdir-dir/Pool/Default
```

Where yourdir-dir is the name of your Director. The **Workset** uses the same tree structure, and all your modifications are made in the worket_dir instead of the config_dir. If you decide to change the volume retention of the "Default" Pool, BWeb will create a new file in the **Workset** as:

```
/opt/bacula/working/conf.d/Director/yourdir-dir/Pool/Default
```

Even with a reload or a restart, Bacula will still use the configuration in the config_dir. When modifying the configuration with BWeb, all files in the workset_dir will take precedence over production files in config_dir.

Fig. 27: Figure 29: BWeb configurator: Workset

Once your modifications are made, click on "Commit" to update files from config_dir with the content of workset_dir. To delete all modifications, click on "Clear Workset".

To reload the Director, click on "Directors" then on "Reload" on the left panel. To restart a local component — Director, Storage or File Daemon —, it's possible to configure "sudo" on the system to allow BWeb to restart the appropriate service.

```
% grep bacula /etc/sudoers
Defaults:bacula !requiretty
bacula ALL = (root) NOPASSWD: /etc/init.d/bacula-fd restart
bacula ALL = (root) NOPASSWD: /etc/init.d/bacula-sd restart
bacula ALL = (root) NOPASSWD: /etc/init.d/bacula-fd restart
```
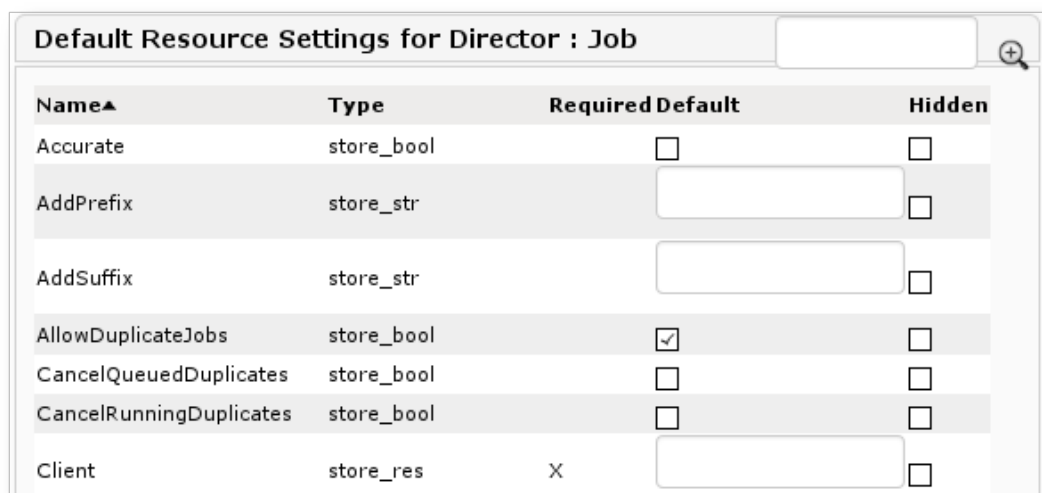
Or if the system is not using `/etc/init.d` anymore:

```
% grep bacula /etc/sudoers
Defaults:bacula !requiretty
bacula ALL = (root) NOPASSWD: /sbin/service bacula-fd restart
bacula ALL = (root) NOPASSWD: /sbin/service bacula-sd restart
bacula ALL = (root) NOPASSWD: /sbin/service bacula-fd restart
```

You may have to adapt the path to the service program. On RHEL, SELinux must be disabled to restart daemons from BWeb.

Working with User Defaults

When dealing with Jobs, you can define templates — JobDefs — to standardize your configuration. When using BWeb, you can use the "User Defaults" feature to define default values on almost all directives. When creating a new resource, BWeb will automatically fill directives with your defaults. This is also possible to hide resources that you do not use on your site (see figure 30) in order to simplify resource edition and/or avoid mistakes.
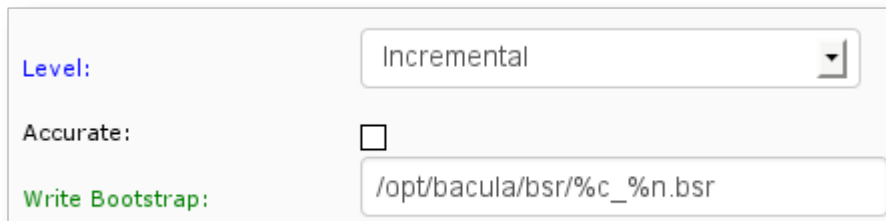
Fig. 28: Figure 30: BWeb configurator: Setup default values

"User Defaults" are applied when you create a new Resource, or when you click on "Apply User Defaults" button on the left toolbar. When applying "User Defaults" to an existing Resource, BWeb will first display all hidden fields, then after your confirmation will apply "User Defaults" to all unconfigured directives.

Color Legend

The following color legend is used in BWeb to identify from where a Directive is set.

- (blue) Directives imported from JobDefs
- (green) Directives imported from User Defaults
- Directives imported from the current object
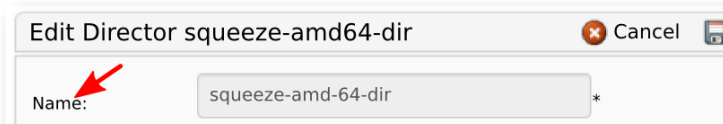
Fig. 29: Figure 31: BWeb configurator: Directive colors
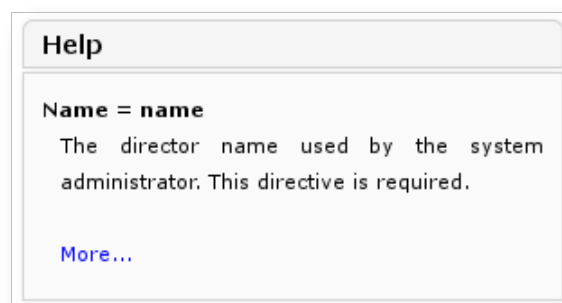
Online Documentation

When clicking on a Directive name, such as "Name", "Address", "Maximum Concurrent Jobs", …
BWeb will update the content of the "Help" box on the right side of the screen with the documenta-
tion related to the current Directive (see figures 32 and 33). When clicking on "More…" BWeb will
open a new window pointing to right section in the manual.

Fig. 30: Figure 32: BWeb configurator: Click on directive for online documentation

Fig. 31: Figure 33: BWeb configurator: Online documentation displayed

Defining New Jobs

In general, when you want to create a new backup Job in Bacula, you need to create other resources such as Client, Schedule or Pool first. If you decide to create the new Job directly from the "Jobs" menu, all resources that you plan to use should be created first.

We advise you to use the "Add a new Backup" wizard available on the left toolbox of the main screen. This wizard will guide you and will allow you to create new resources in a logical order.

Define Retention Policy

The wizard called "Add a new Backup Policy" is designed to help you defining a backup strategy around Pool retention and Schedule. In the 6.2.0 version, the Pool retention creator will use only Full and Incremental backup level. Scheduling Differential level is planed for a future version.

Configure Remote Clients

BWeb configuration module includes a wizard that allows you to configure remote clients. Bacula Systems provides a new tool to allow remote installation, configuration and upgrade of Clients. BWeb creates a shell script in your workset that may help you to push the configuration to your Linux/Unix/Windows clients.

- Linux
- Windows

This shell script is not designed to handle all possible configurations, especially on Windows environments. Feel free to modify the script and report your feedbacks and ideas to the Bacula Systems team through the support channel.

If Something Goes Wrong

If after some work on your workset BWeb is displaying a message such as:

```
13-Dec 15:03 bacula-dir: ERROR TERMINATION at lex.c:340
Config error: expected a positive integer number, got: a
  : line 10, col 15 of file /opt/bacula/working/conf.d/Director/z/Job/j1.cfg
  Priority = a
```

You can still edit with your editor the file that causing the error, or you can clear the workset on the main page. This kind of problem should not happen, and you can contact Bacula Systems support team to resolve the issue.

Current Limitations

The Configuration module is suitable for small, medium, and large installations varying from one client to thousands.

Unsupported Configuration Items

In version 8.2 of the BWeb configuration module, the following directives and resources are not supported.

- `Dir Addresses` directive

- `Sd Addresses` directive

- `Fd Addresses` directive

If you are using these Directives, feel free to contact the Bacula Systems support team to analyse the impact on your situation.

Changes in bconsole

Once the configuration is handled by BWeb, resources will no longer be sorted by configuration definition order, but by alphanumeric order. It means that when `bconsole` will display a prompt, the list may change from what you had in the past.

```
*run
The defined Job resources are:
     1: BackupClient1
     2: BackupCatalog
     3: RestoreFiles
```

Will become:

```
*run
The defined Job resources are:
     1: BackupCatalog
     2: BackupClient1
     3: RestoreFiles
```

2.5 Administration Functions

The following chapter presents the information on how to perform administration functions.

Managing Backups

Monitoring Backups

The list of running Jobs is shown on the main screen and in the menu item Jobs → Running jobs.

Jobs shown on this screen can be running or waiting execution . Clicking on the the Status icon permits monitoring the Job in real-time.

On the real-time monitoring screen, you have access to the logs of running Jobs, and other information such as the name of the current file being backed up, the backup rate, the number of files examined, the number of files backed up, etc.

This screen appears only when the file backup is completed (the despooling to tape and insertion of metadata in the catalog can still be in progress).

Fig. 32: Figure 34: List of running Jobs

Fig. 33: Figure 35: Monitoring Backups in real-time

The estimation of the backup progress is calculated from previous jobs of the type found in the catalog. A tooltip on the progression bar will show the estimation reliability (covariance).

This screen is regularly refreshed. You can cancel the current Job by clicking on ⊗ (all *critical* operations require a confirmation).

Viewing Job history and logs

Connect to BWeb and go to the Jobs → Job History menu. On this screen, you can select the Job that interests you by name, Fileset, Client, Type, etc.

Fig. 34: Figure 36: Job History

It is possible to save a bookmark or send the page address by email after having filtered the information on this page.

To do so, click on the Status icon shown on the left, OK ✔ , OK with Errors ⚠ , Terminated in Error ⊗ in order to access Job information and the logs.

On this screen (Cf. figure *37*), it is possible to view information such as Volumes used (2), statistics (4), suggestions and documentation (7). You may also delete this Job from the catalog (1) or restart the Job (8) with the same parameters (Pool, level, Fileset, Client, etc) that were previously used.

If the statistics on the growth of the backup size (4) interests you, you can view the backup contents with the **bfileview** module (6) in order to rapidly find the largest files in the Fileset.

In this example, Job 10 has problems. Once in the detailed screen (Cf. figure *37*), you can filter the logs to display only the error messages (9). Verify the history of this Job (3) to see if the error is recurring or not. If the error is critical, and it seems reasonable to restart the Job, it can be restarted directly (8). When restarted, the original Job 10 can be deleted from the catalog. However, we advise not to delete Jobs from the catalog that contain backup data as they may still be useful for recovery later.
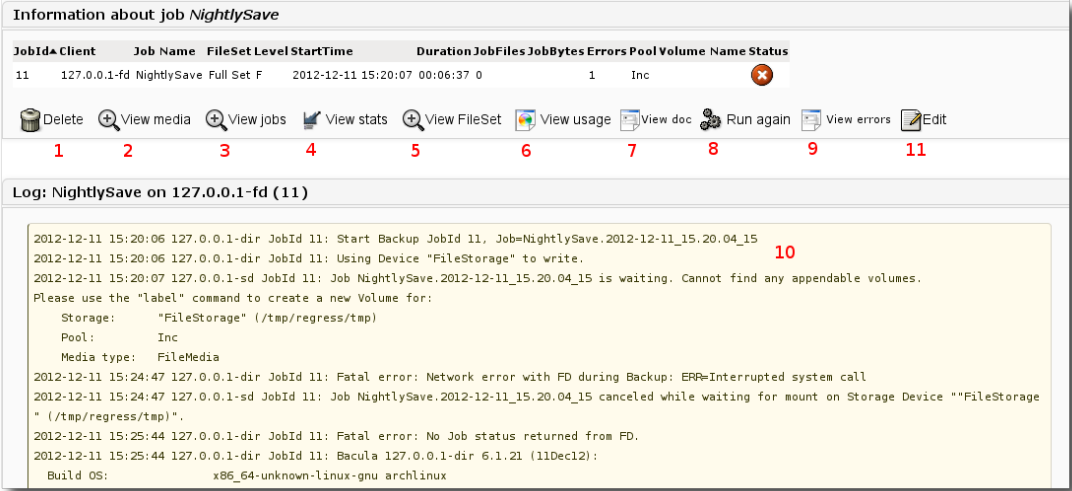
Fig. 35: Figure 37: Job Information

Problem and Missing Jobs Detection

BWeb has a screen, reached by Jobs → Missing Jobs, which shows a list of failed Jobs or Jobs that are not found in the the catalog (due to a purge or a Director crash for example). BWeb makes a calculation based on the Job schedules that should execute during the period and looks for them in the catalog. If the Job is not present or its termination status is not OK, it will be shown in the list (Cf. figure *38*).
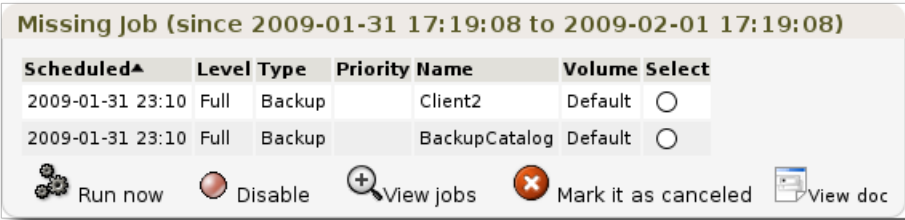
Fig. 36: Figure 38: Monitoring problems

From this screen (Cf. figure *38*), it is possible to see from the Job history if there was a recurrent problem, to check the information concerning restarting a job, to start the Job, or to mark this Job in error in the catalog history table if restarting is not possible. (The Job *absent* will therefore be included in the statistics)

It is possible to modify the frequency of looking for failed Jobs by using the `age` or `begin` and `end` parameters.

Copyright © 2025 Bacula Systems. All trademarks are the property of their respective owners.

Manually Starting a Backup Job

The list of defined Jobs is available in the menu Jobs → Defined jobs. (Cf. figure *39*)
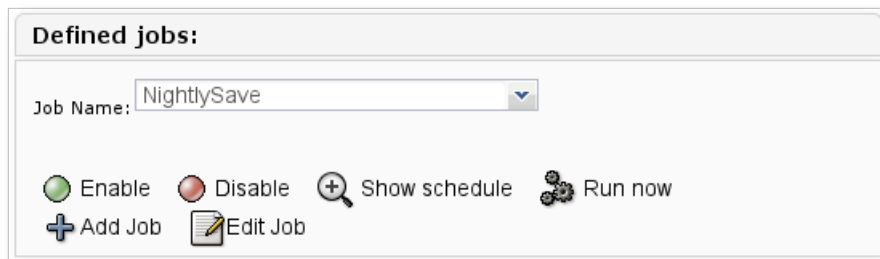
Fig. 37: Figure 39: List of defined Jobs

It is possible to see the Jobs that are scheduled, to deactivate a Job or to manually start one. After submitting a Job to the Director, BWeb automatically changes the current screen to show the Job Monitoring screen for the started Job.
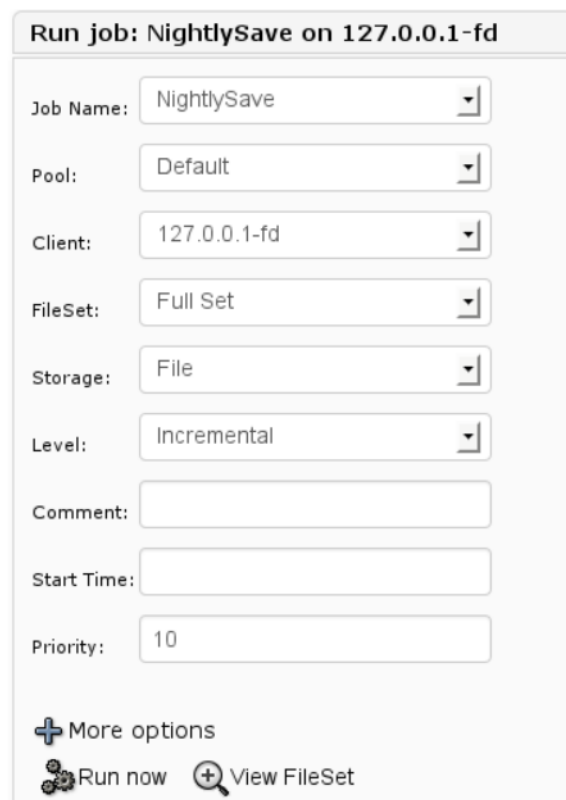
Fig. 38: Figure 40: Modification Job parameters before running

Restarting a Backup

When starting a backup, the Director uses the various values specified by you. For example, you can select the Pool or the backup Level from the **Schedule** resource. These values can depend on the scheduled time and thus can be a source of error for manually restarted Jobs. BWeb permits to automatically set these values for each job on the details screen for the Job (Cf. figure *37* option 8)

Stopping a Backup

Canceling a running Job is possible from several screens using the option . All *critical* options require a confirmation.

Managing Restores

BWeb has a dynamic interface that permits a natural navigation in the directory structure of the Job. By default the list of directories and files are automatically presented considering all the different Filesets and the date specified. For example if you have divided your backup in several Jobs to improve backup speed or to simplify restores as follows:

```
Fileset {
 Name = "FS1"
 Include { File = /usr }
}
Fileset {
 Name = "FS2"
 Include { File = /home }
}
```

In the above case, both `/usr` and `/home` will be displayed on the same screen. This feature can be modified using the options menu (4).

Starting a restore

The restore interface can be accessed from Jobs → Web Restore.

Table 6: Table 5: BWeb restore interface: layout information (the zones are highlighted in figure 41).

Zone	Information
1	List of Clients defined by the Director. The list utilizes an automatic filter when you begin entering the name
2	For each Client, this zone gives the list of Backups recorded in the catalog. If you click on a Differential or Incremental Job, you will see all the files since the last Full backup.
3	The current directory and file. Clicking on "Change location", will cause BWeb to show the selected directory.
4	Options for navigation. By default, BWeb will only show the most recent version of your selection. You can can choose to show all versions of your files in the "File Version" panel. By default, BWeb will combine all the Filesets in the current view taking into account the backup level (Full, Differential, Incremental). It is also possible to show only the Job selected.
5	When your selection is made, this button will start the restore of the selected files in zone (9)
6	List of the backed up directories. It is possible to go directly to a path in zone (3)
7	List of files present in the chosen directory. The panel has a *regex* filter option and supports directories with a large number of files. In the filter zone, pressing on "Enter" applies the filter to the elements shown in the list. Clicking on the reload button applies the filter to the catalog.
8	Listing of available versions of a file. The panel permits rapid viewing of the different versions of a file as well as the name of the Volume and the on-line availability(only available for tape media). To list all the versions of a file, you must select the option: "See all file versions" in the menu (4).
9	Selection zone of files to restore. You should drop and drag directories (6) and files (7 or 8) to be saved into this zone. It is possible to delete one or more entries with the "Del" key.

After selecting your files, the start restore screen will permit you to modify the restore options such as the destination client, or to relocate the restored files (Cf. figure 42). The list of Volumes needed will permit mounting them if necessary. **Note**, the current list of Volumes does not take into account the directories selected.

Finding a version of a File

Once you have found the file to restore, the panel (8) should present a list of all the versions found in the catalog. By default only the latest versions (having a different checksum) are displayed. The options menu permits you to display all available versions. **Note**, you can only restore one version of a file in any given restore Job.
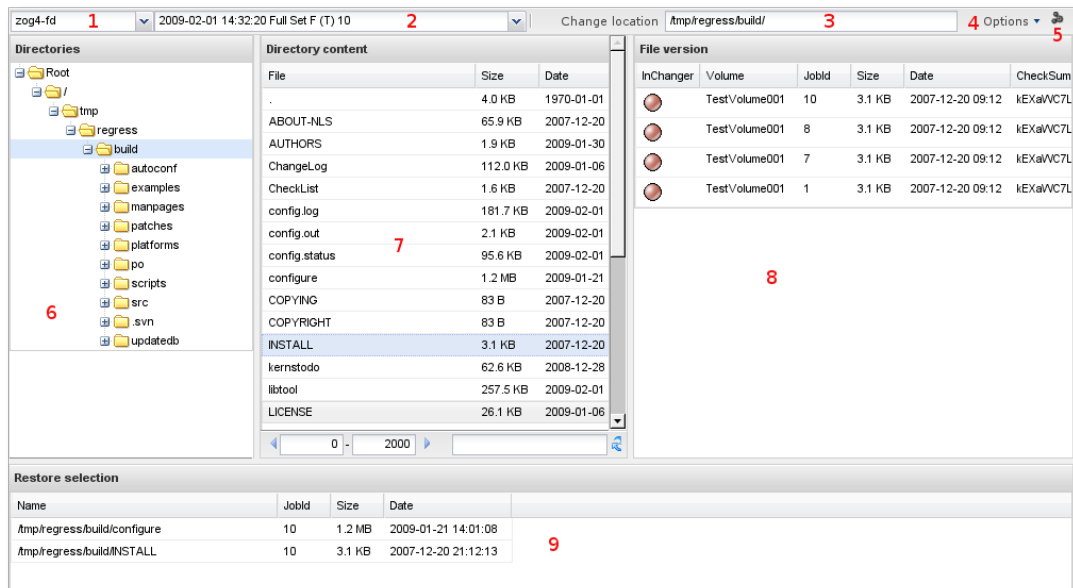
Fig. 39: Figure 41: Web Restore Interface
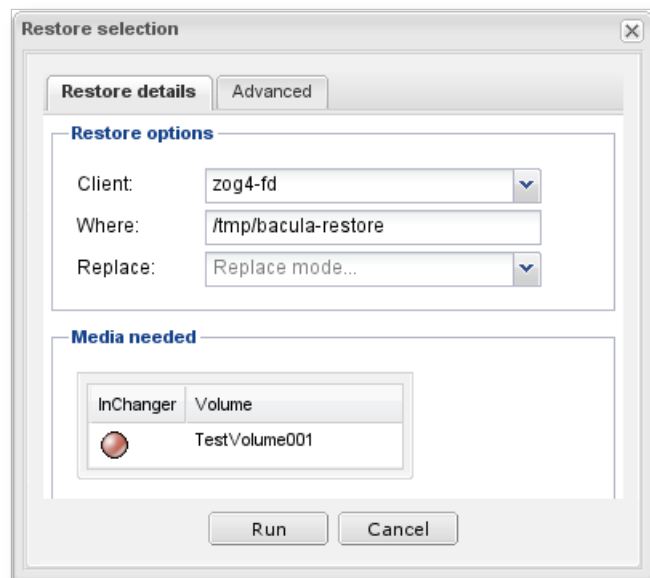
Fig. 40: Figure 42: Restore Options

Optimization of Restored

BWeb tries to select the best version possible for a file. However, it can sometimes be useful to specifically select the files by their location in order to minimize the number of Volumes necessary to effect the restore.

Relocating Files

The tab "Advanced" on the start screen for a restore permits you to manage the relocation of files (where the files are restored). The main options permit you to remove a part of the path of a file (**Strip prefix**), add a prefix or a suffix to each file (**Add prefix** et **Add suffix**), or to construct more complicated expressions using the **Where regexp** fields (see the Bacula manual for more information). The following example (Cf. figure 43), permits to restore a file originally on drive C: to drive D: on a Windows machine.
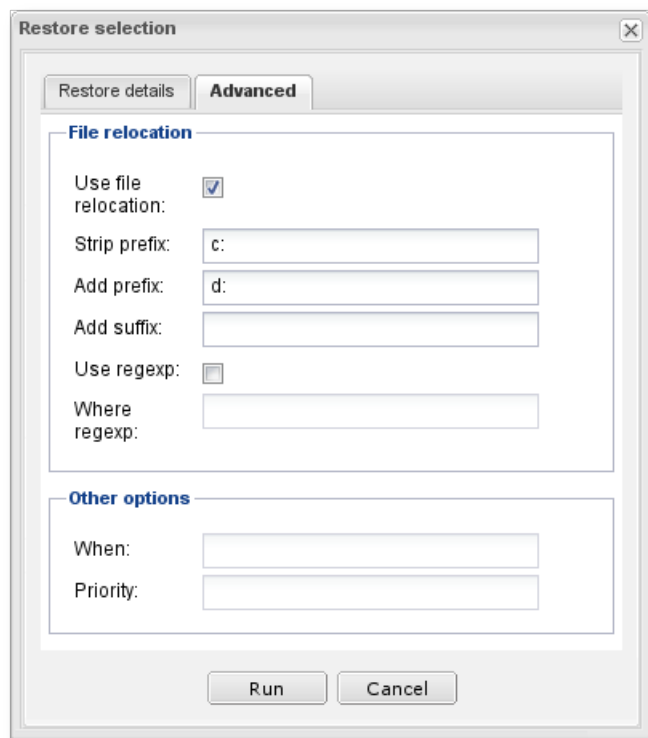
Fig. 41: Figure 43: Relocalization of files

Note, on Windows you must use "/" and and not "\" in the destination field.

Consulting the restore log

Accessing the restore logs is done in the same way as accessing other logs: by the history menu (See Viewing Job history and logs).
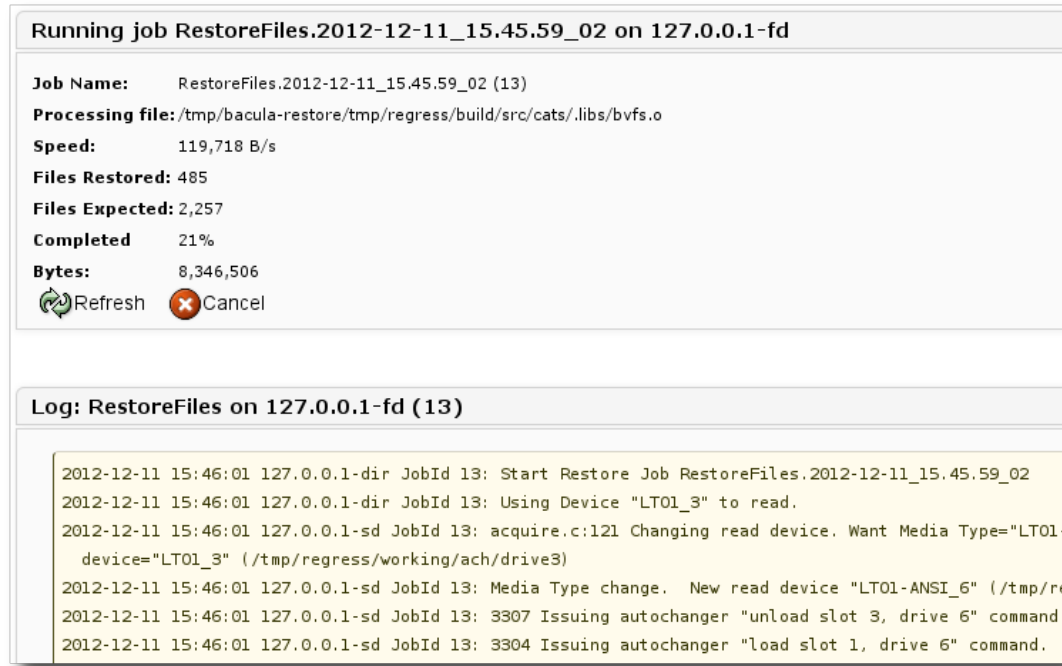
Fig. 42: Figure 44: Restore

Viewing the restored files list

To view the list of files actually restored, you must configure the in your client to send the messages of type `restored` to the Director (see below). Be aware that the list of restored files can be very long and thus slow down your restore job.

```
# bacula-fd.conf
Messages {
  Name = Standard
-  director = zog4-dir = all, !skipped, !restored
+  director = zog4-dir = all, !skipped
}
```

Copying the log to a file

The text can be copied into a file and used in an email.

Managing Volumes

The following section provides information about managing tape media with Bacula and BWeb.

Volume Life Cycle

Figure 45 describes the lifecycle of media.

Status	Information
Unknown	The Volume is not known to Bacula. You must label it using BWeb or `bconsole` with `label barcodes` command
Append	The medium is directly usable by Bacula.
Used	The Volume is not usable by Bacula. A limit set by the administrator has been exceeded (maximum size, utilization period, number of jobs, number of files, etc.) The Volume can be removed.
Full	The Volume is full and can be removed.
Purge	The data is still on the Volume, but the retention period has expired. The Volume can be put back into the autochanger to be reused (overwritten). When Purged is set, the Volume will be automatically moved to the Pool specified by the **Recycle-Pool** directive.
Recycled	If a Volume is marked to be recycled, it will be placed in the Recycle state before being once again placed in Append mode.
Error	The Volume is in error. Error does not always indicate that it was a hardware error. If you want to reuse the Volume after verification that it is OK, you can reset it to Append status.

Pruning deletes records from the catalog when the retention period has expired. It is a safe form of Purging.

Viewing cassettes by Pool

With BWeb, under Media → Pools, you can see the list of Pools defined in the Director (see figure 46). There is a different entry for each Media Type.

You can easily estimate the capacity and the occupation percentage of your Pool with **Usage**. It calculates the estimated Volume capacity using the average size of all Volumes of the same Media Type marked as *Full*.

The **Vol Status** section gives you a listing by status of the Volumes in the Pool. A tooltip will show you the exact count (e.g. 1 Full, 2 Append, 1 Purged).

Fig. 43: Figure 45: Life Cycle of a Volume

Fig. 44: Figure 46: Listing Pools

Volume List

The list of Volumes is available in several places including menu: Media → All Media. This list permits a rapid overview of the Volumes known to the Director. It is possible to filter the display via the **Filter** menu on the left, where you can specify the media type, its physical location, its status, ... It is also possible the select several Volumes at the same time for *pruning*.
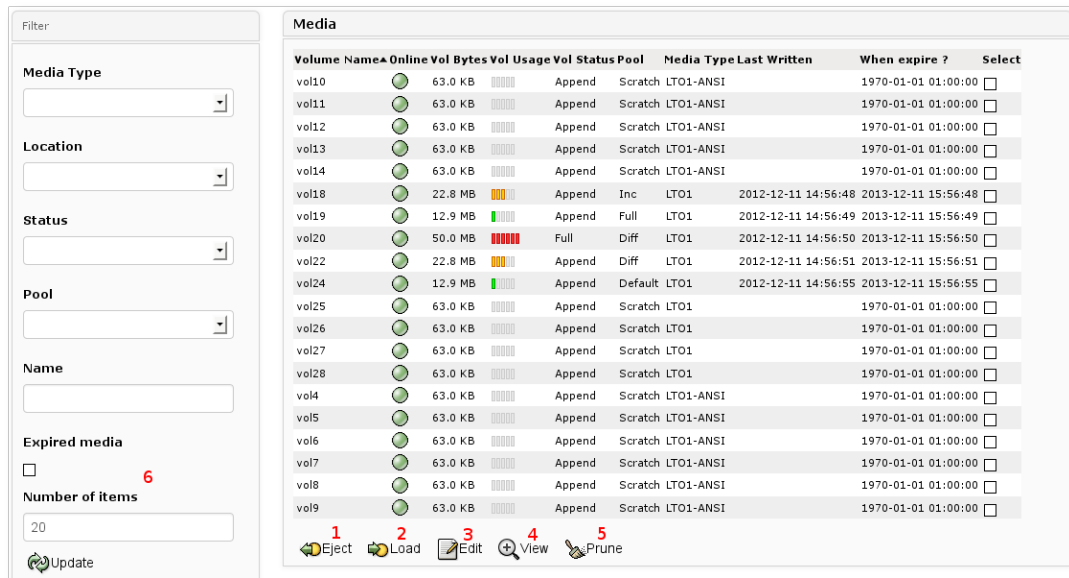
Fig. 45: Figure 47: Volume List

The table 7 below describes the several labels displayed on figure 47.

Table 7: Table 7: BWeb Volume list: layout informations

Zone	Information
1	Ejection of Volumes to the import/export tray
2	Start introduction of selected Volumes
3	Edit properties of selected Volumes
4	Display information about selected Volume
5	Start a prune of selected Volumes (respects retention periods)
6	Display only Volumes marked **Full** or **Used** having an expired retention period

Viewing Volume expiration dates

The expiration date is show on all the Volume displays. You can sort the column entitled **When expire ?** to know when the Volumes will be available for reuse.

You can also filter the Volume display to show only Volumes that are expired (Cf. figure 47, field 6).

Viewing Volume properties

To view the Volume properties, you can:

- use the search box at the top left
- use the "Show Volumes" found in the backup log
- select the Volume on the Pools screen
- select the Volume on the location screen

The figure 48 shows the volume properties interface while the table 8 explains its content.

Table 8: Table 8: BWeb volume properties: layout informations

Zone	Information
1	Eject the Volume (put it in the import/export tray) if the Volume is online
2	Edit the Volume properties
3	Purge the Volume (disregards retention periods). You must confirm this request.
4	Start pruning a Volume (respecting retention periods)
5	Show the log of Volume movements from place to place
6	Show the backup log
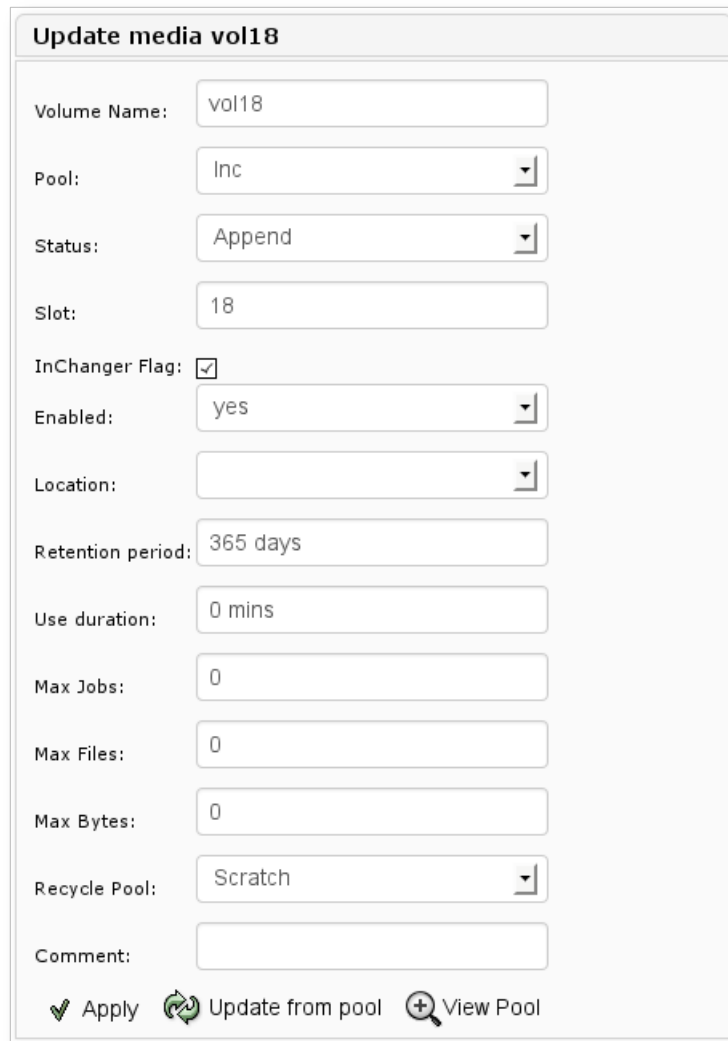
Fig. 46: Figure 48: Volume Properties

Copyright © 2025 Bacula Systems. All trademarks are the property of their respective owners.

Modify the Volume properties

Certain properties of a Volume are not shown on the properties screen (cf. figure 49), for example, the number of files on a Volume. These *advanced* properties can be modified with caution using `bconsole` and the `update volume` command.

Fig. 47: Figure 49: Modify Volume properties

Managing the autochanger

BWeb permits you to manage an autochanger with a barcode reader, multiple drives, and an import/export tray. The screen (See figure 50), found under Autochanger → YourAutoChanger, permits you to:

- show the list of slots and their contents (9) ;

- move Volumes from the import/export tray to free slots (3) ;

- detect synchronization problems with the catalog (7) ;

- synchronize the catalog with the contents of the autochanger (`update slots`) (4) ;

- view which drives are occupied (8) ;

- move selected Volumes from their slots to the import/export tray (2).

Fig. 48: Figure 50: Managing an autochanger

Labeling Volumes in an autochanger

BWeb facilitates adding new Volumes to Bacula's catalog. The new Volumes will by default be added to the Scratch pool.

1. Use the BWeb's eject screen (Cf. Exporting Volumes) to make room in your autochanger, if necessary.

2. Place fresh Volumes in the import/export tray of your autochanger (if the autochanger has one).

3. Move the new Volumes into the free slots with the "Clear I/O" button on the Autochanger → YourAutoChanger screen.

4. The new Volumes should be shown as "Unknown"

5. Select the new Volumes.

6. Select a free drive.

7. Click on "Label"

The labeling operation can be rather long depending on your configuration, and it can even block the display until the end of labeling all volumes. If you have a lot of volumes to label, we suggest you label in groups of five at a time. If you label a large number, you may want to modify your Web server configuration to increase the "Timeout" setting for running scripts.

Deleting Volumes

This function is not available with BWeb. You must use the command `delete volume=R30001` in `bconsole` to delete a Volume from the catalog. It is not necessary to have the Volume in the autochanger.

Managing Scratch Volumes

The Scratch pool permits simplifying management of tape Volumes. Using the Scratch pool is not mandatory and must be setup by the Bacula administrator.

To make a backup using the Scratch pool, if there is not a Volume in the current Pool with status Append, Purged, or Recycled, Bacula can select a Volume from the Scratch pool. When you label new Volumes using BWeb, they will automatically be placed in the Scratch pool. If you use the **RecyclePool** option, you can force a purged Volume to be automatically returned to the Scratch pool.

BWeb automatically manages Scratch pool Volumes. A screen allows you to make room in your autochanger by automatically selecting Volumes to eject and will propose the best Volumes to import (Cf. Exporting Volumes and Multiple Sites).

Exporting Volumes and Multiple Sites

If you use an autochanger and tape Volumes, you may want to export Volumes to another site to make room in your autochanger or for security of your Volumes. BWeb has several screens to help manage multiple sites. It is possible to let BWeb chose the best Volume export candidates (e.g. Volumes marked *Full* and long retention periods), and BWeb can also chose the best candidates for importing into your autochanger (e.g. expired media or very old Volumes).

Also available is a simple workflow that permits one to track and record each Volume movement.

Exporting Volumes

You must first select the Volumes to export by one of the following:

• location

• Pool

• the Export screen

Then you must click on the "Eject" button . If your Autochanger has an import/export tray, the Volume or Volumes selected will be automatically put there.

Use the Volume movement management screen (Cf. figure 52)

Fig. 49: Figure 51: Moving Volumes to the import/export tray

Fig. 50: Figure 52: Moving a Volume

You must specify the Volume destination (new location). When clicking on the "Move", you can send an email with important information (Cf. figure 53), to the people in charge of physically moving Volumes, or you can confirm that the move was actually made ("Update now").

Once the Volume or Volumes are out of the autochanger, you should synchronize the autochanger and the catalog as documented in Managing the autochanger. See figure 55 for a picture of how the workflow for moving media works.

Importing Volumes

Importing Volumes follows the same workflow diagram as exporting (Cf. *Exporting Volumes* and figure 55). Once the Volumes are in the import tray of your autochanger, you can click the button "Clear I/O" on the Autochanger → YourAutoChanger screen to ask BWeb to put

Help managing the autochanger contents

BWeb can help you manage the contents of your autochanger by selecting the best volumes to import for use with your future backups.

Fig. 51: Figure 53: Sending an Email for moving a Volume

Fig. 52: Figure 54: Confirmation of Volume movement
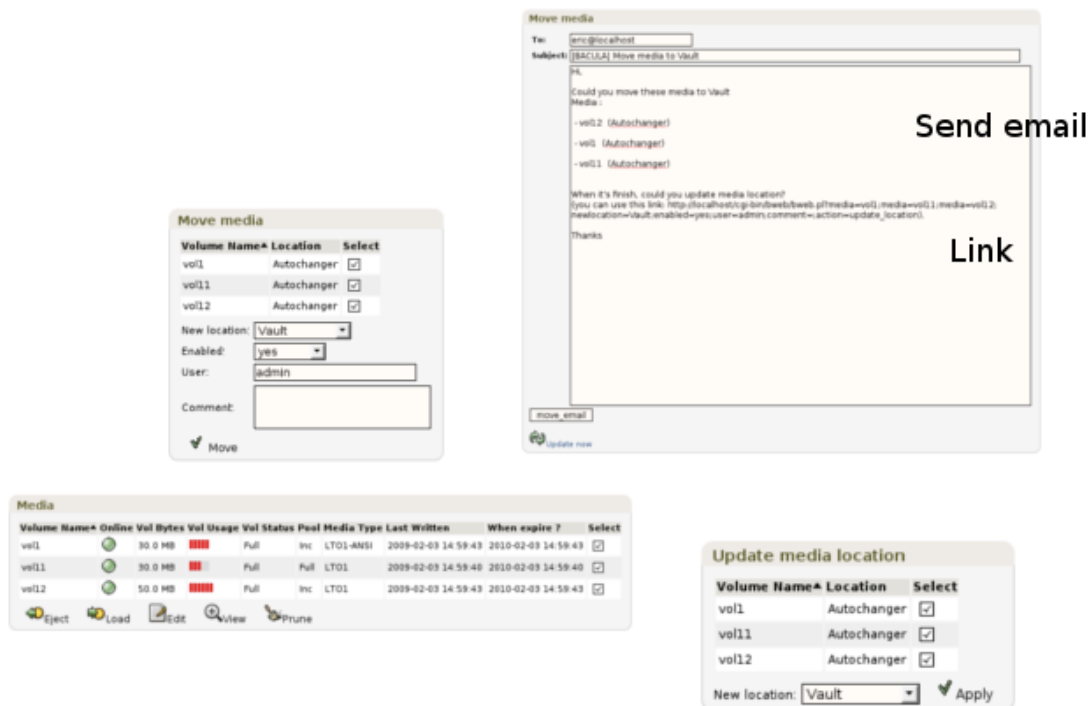
Fig. 53: Figure 55: Workflow of moving Volumes

Automatic exporting

In the Media menu → Eject Media, BWeb can select a list of Volumes for you that can possibly be moved out of your autochanger to make room for other Volumes. To do so, you must chose a Pool, a Media Type, and the Volume's current location (Cf. figure 56).

The candidate Volumes are sorted by their retention period, and their use (the number of times mounted in a drive) (Cf. fig 57).

Clicking on the export button will start the process of question and answers for exporting Volumes according to the workflow defined above (Cf. *Exporting Volumes*). Note: BWeb will not let you export more Volumes than your export tray will hold.

Automatic Importation

In the Media menu → Load Media, BWeb will select for you a list of Volumes possible to import back into your autochanger for use in future backups. You must choose a Pool, a Media Type, the current Volume location, and whether or not you want to include Volumes that will expire in the near future (Cf. figure 58).

The candidate Volumes will be presented sorted by their usage (number of hours writing and number of times mounted in a drive) (fig 59).

Clicking on the import button will start the workflow question and answers described above (cf. *Exporting Volumes*)

Fig. 54: Figure 56: Selection criteria for exporting Volumes

Fig. 55: Figure 57: List of media to Export

Fig. 56: Figure 58: Selection criteria for importing Volumes

Fig. 57: Figure 59: Volumes selected for importing

Monitoring

The Bacula Systems Monitoring package uses BWeb libraries to communicate with the Director, so you must install this package on a server that has a functioning BWeb. Once installed, you can monitor the following elements:

- Storage Daemons
- Scratch Volumes
- The number of failed Jobs during a specified period
- The number of Jobs running or blocked during a specified period
- The evolution of backup statistics (for example: find an error after a big change in backup size).

For more information, see Administration Functions.

Fields Validation

In the table below, you are presented with a list of allowed characters in particular fields in BWeb.

Table 9: BWeb fields validation: allowed characters

Field	Allowed characters
System authentication username	`A-Za-z0-9.-]+[@`
System authentication password	`A-Za-z0-9_./ :@-+!~#$%^&()?`
Cloud Storage access key	`A-Za-z0-9/+=`
Cloud Storage secret key	`A-Za-z0-9/+=`
Cloud Storage bucket name	`A-Za-z0-9.-`
Vmware Center username	`A-Za-z0-9.-@~^+`
Vmware Center password	`A-Za-z0-9.-/+@#()[]{}|=*%$ :;!,?^<>&~`
Vmware Center vSphere server name	`A-Za-z0-9-`
Password in Bacula resources	`A-Za-z0-9.-/+@#()[]{}|=*%$ :;!,?^<>&~`

Interface Customization

Custom CSS overwrite can be done by adding CSS classes to `/opt/bweb/html/user.css` file.

2.6 Activity Analysis

This section of the BWeb manual presents a methodology for you to analyze and optimize the different parts of your Bacula installation.

Planning Help

The BWeb `btime` module, which is accessible in the Statistics → Timing screen, permits visualization of Job progress during a time period. Each significant step of the backup is shown (waiting for a resource, *spooling*, *despooling*, attribute insertion, . . .). The analysis is made from backup logs kept in the catalog. Logging the data must be enabled (Cf. section Director Configuration).
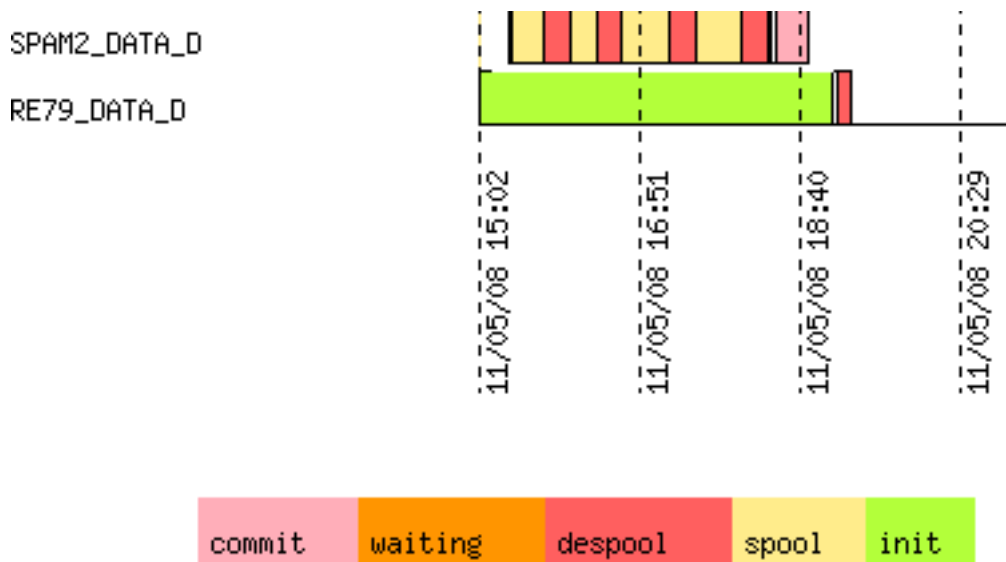
Fig. 58: Figure 60: Analysis of Job scheduling

This module permits optimization of your autochanger drives by their utilization in your backup window. You can rapidly determine if any given drive is underutilized or if an additional Job will work within your current scheduling. If you have multiple drives, this screen will permit you to plan the use of several Pools to run Jobs in parallel to maximize drive usage.
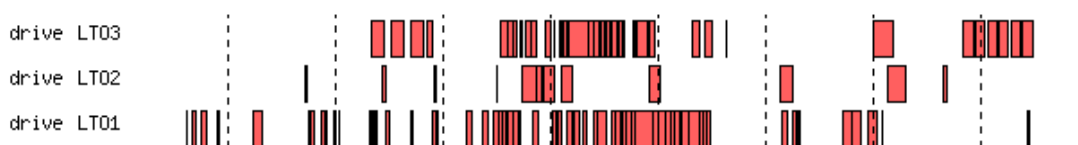
Fig. 59: Figure 61: Time Analysis of Drives

In general, you should avoid periods of underutilization (waiting for resources, run before script, . . .) and underutilization before *despooling* by modifying the **SpoolSize** value and by adjusting scheduling.

You can also detect throughput problems and overuse of the catalog by varying the number of Jobs that run simultaneous.

Rapid Detection of Problems

BWeb has several tools to help you quickly find backup problems by using the 80/20 rule. Certain servers could have configuration problems (switch port in half-duplex, saturation of E/S, Firewalls, . . .) which cause problems in proper ordering of your Jobs. In Statistics → Type → "Time,Size,Files" and "Time,Files,Size" you can graphically see the classification of your Jobs by backup time, number of files backed, and file size (Cf. figure 62).

Each circle — which can be selected with the mouse — represents a backup whose name can be seen in a tool-tip. This view permits you to quickly see what Jobs are fast, slow, have a large number of files, a large backup volume, . . .

You can see in the example figure 62 that Job (1) is rather small and yet took a lot of time to run. Other Jobs with similar sizes finish much faster. By clicking on the circle you will get the backup report for the Job in question, and you will be able to start your investigation.
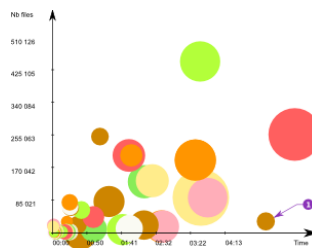
Fig. 60: Figure 62: Break down of Jobs by run time, number of files, and file size

Statistics

Prerequisites

The BWeb statistics module uses the *Long term statistics* feature of Bacula available since version 3.0. With older Bacula versions you must manually create the table and fill it with statistics.

In order to be able to study the evolution of your backups, analyze the tendencies and make reliable reports, you should actively archive your Jobs (Cf. Maintanence Tasks/Statistics).

Job Errors

In order to have a measure of success rate that is closer to reality in the case that you successfully restarted a failed job, you should delete the failed job from your catalog.

After having viewed the logs of your failed Job in BWeb, (Cf. Viewing Job history and logs), you can directly restart the Job after having verified the parameters (Cf. section Restarting a Backup), then remove the failed Job from the catalog. In the worst case the newly started Job will also fail, and your statistics will take it into account.

Detection of missing Jobs

The missing job detection feature (Cf. Problem and Missing Jobs Detection) also allow an easy way to restart missing jobs during the period your server is down for maintenance or there was a server outage.

If restarting a job is not possible clicking the button ⊗ "Mark this job as canceled" will add a special record in the history table to consider this Job as failed.

Proper management requires you to regularly monitor your Jobs with BWeb.

Exporting data to a spreadsheet

With the Statistics menu → Groups, BWeb presents a series of data by group of machines which can be cut and copied into a spreadsheet.

Backup Contents Analysis

BWeb's `bfileview` available in the detailed description of a Job with 🖼 "View file usage" permits viewing utilization of disk space by a series of clickable circles. This analysis uses the Bacula catalog and doesn't involve your production servers.

On figure 63, you can easily see that the directory `examples/database` occupies 70% of the total space. A tool-tip shows that the file `backup.old.tgz` is responsible. This analysis permits you to rapidly focus on large resource users.
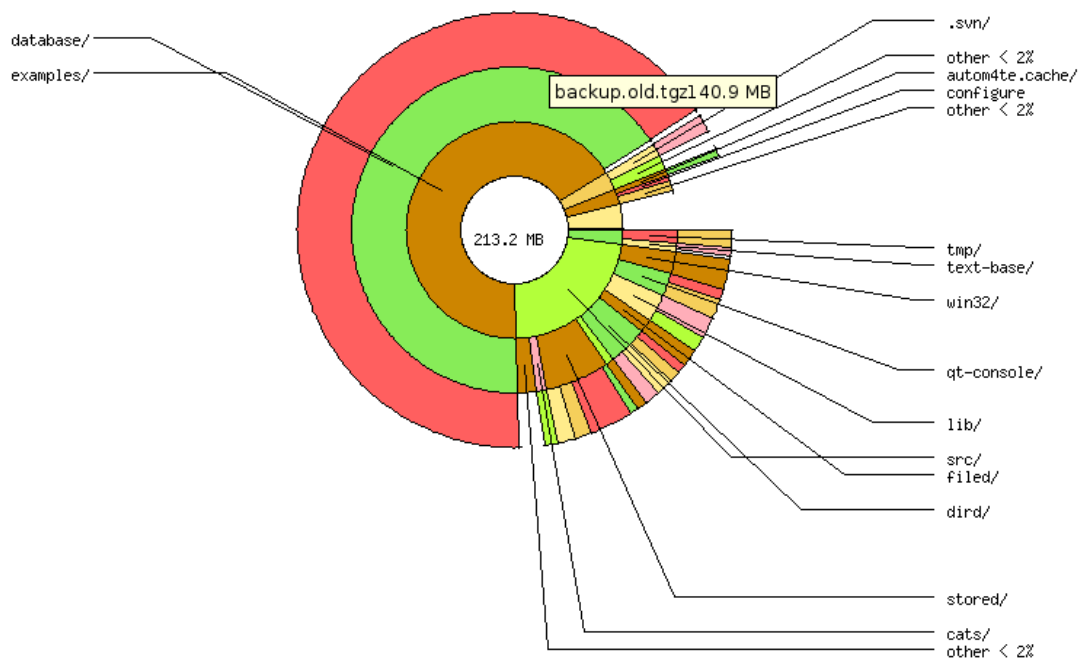
Fig. 61: Figure 63: Analysis of the contents of a backup

In order to rapidly navigate, `bfileview` uses a cache maintained in the catalog. The updating of this cache can take some time depending on how may files were backed up. To avoid doing this on-line, you can automatically update the cache using a script.

```
$ /opt/bweb/cgi/bfileview.pl jobid=xxx where=/ mode=batch$
```

When browsing to restore a Windows machine, you must first enter the disk drive name such as: `c:` or `d:` in order to see the file view.

2.7 Problem Resolution

If you have some problems with BWeb, run the "Check Bweb Configuration" tool available in the "Configure BWeb" screen.

I don't see the defined job list

If you use Apache, then you should look at the Apache `error_log` file to check for Director connection errors. You should also verify that the Director is running and that the Apache user is able to execute `bconsole` and read `bconsole.conf` (see bconsole Configuration).

I don't see job logs

You should verify that `Messages` resources of your Director uses the `catalog` as backend (Cf. Director Configuration) and that BWeb SQL functions are loaded in your catalog (See Catalog Update).

I don't see anything in tables

You should verify that you loaded Bweb SQL scripts to your catalog (see Catalog Update).

I don't see any locations when "Adding an Autochanger"

You should check your web server log (Apache `error_log`) file. BWeb is unable to run the `bconsole` program. Possible causes are usually:

- `/opt/bacula/bin/bconsole` is used instead of `/opt/bweb/bin/bconsole`

- `/opt/bacula/etc/bconsole.conf` is used instead of `/opt/bweb/etc/bconsole.conf`

- the Apache user does not have the rights on `/opt/bweb/bin` and/or `/opt/bweb/etc` directories.

2.8 Adding Remote Storage Daemon to Bacula using BWeb

The following article presents how to add a remote Storage Daemon to existing Bacula Enterprise infrastructure using BWeb. It consists of the following parts: configuring the new SD in BWeb, pushing the configuration using the BWeb push wizard.

New SD Configuration in BWeb

1. Navigate to http://xxx.xxx.xxx.xxx:9180/cgi-bin/bweb/bweb.pl (xxx.xxx.xxx.xxx is the remote Storage server IP address).

2. Click "Configuration" –> "Director" –> "Storage".


3. Click "Add".

4. Click "Add New Storage".
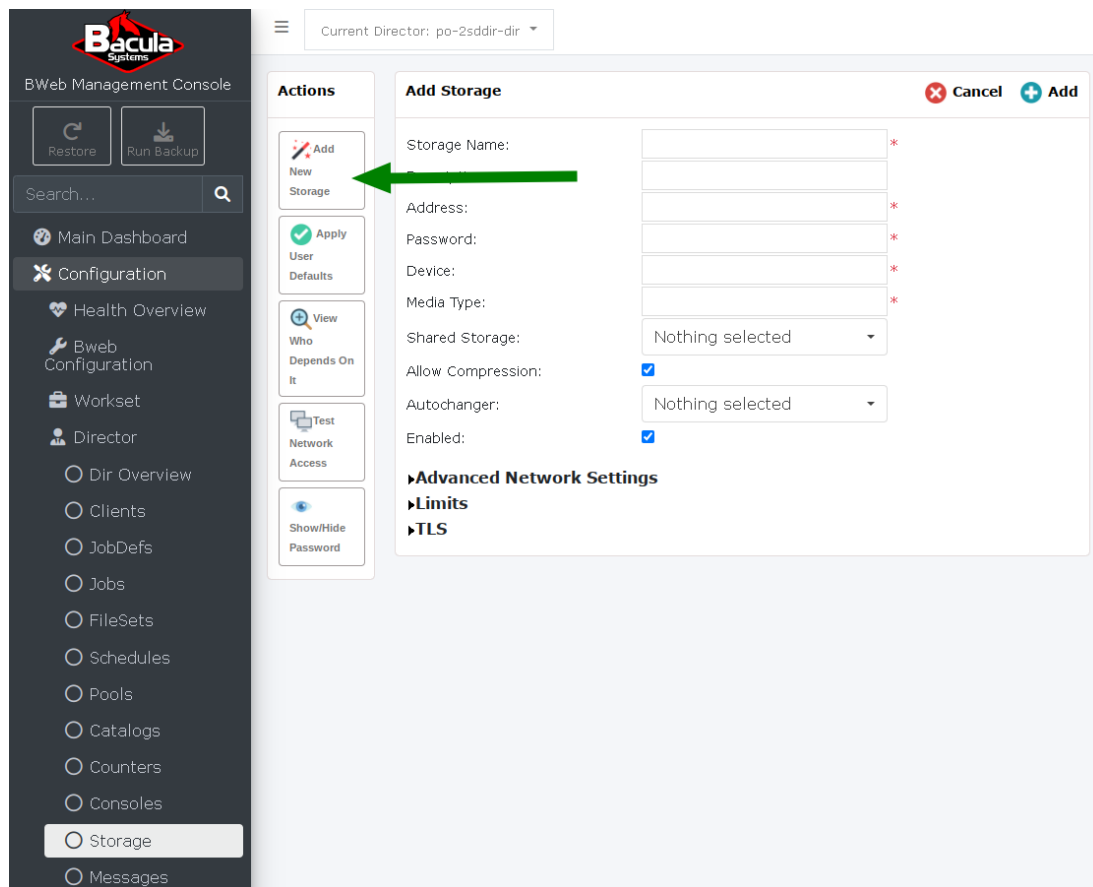
5. In Step 1/4, choose the 2nd option: "Create a new Storage Daemon and add newly created Devices into it".

6. Fill in the text fields with "RemoteStorage" for Storage Daemon Name, and "20" for Maximum Concurrent Jobs.

7. Click "Next".

8. In Step 2/4, expand "OS Type" dropdown and choose "Linux/Unix".

9. Expand "Copy from existing Storage" dropdown and choose "DiskAutochanger".

10. Type the the IP address of the Storage Daemon of the remote server (DNS or IP number)in the Address field.

11. Click "Next".

12. In Step 3/4, type "DiskType" in the "Media Type" field.

13. Click "Next".

14. In the Step 4/4, type "RemoteStorage" in the "Name" field.

15. Click "Save".

16. In the workset screen, click "Commit" to commit your changes to production.

SD Configuration Push

1. Click "Configuration" –> "Storage Daemon" –> "SDs Overview".

2. Click the "RemoteStorage" radio button.

3. Click "Push".

4. Click the "Perform the deployment directly from BWeb Manager Suite" field.
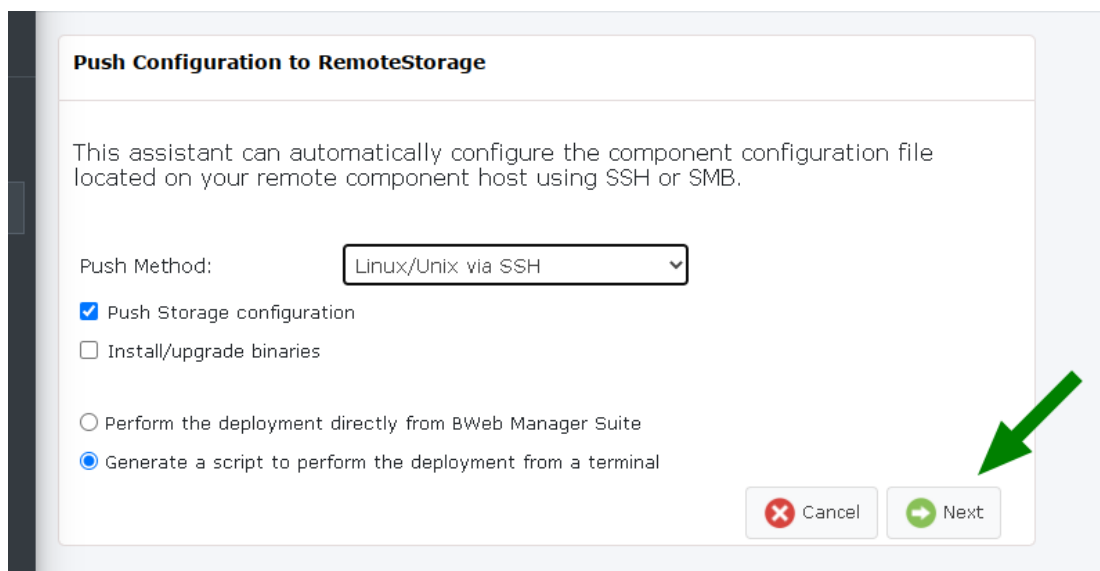
5. Click "Next".

6. Type in the password in the "Password" field.

7. Click "Next".

8. Click "OK".

Configure Pool and Job to Use the Remote Storage

The following article presents how to configure Pool and Job to use the remote Storage Daemon.

Configure Pool to Use the Remote Storage

The following article presents how to configure Pool to use the remote Storage Daemon.

1. Navigate to http://xxx.xxx.xxx.xxx:9180/cgi-bin/bweb/bweb.pl (xxx.xxx.xxx.xxx is the remote Storage server IP address).

2. Click "Configuration" -> "Director" -> "Pools".
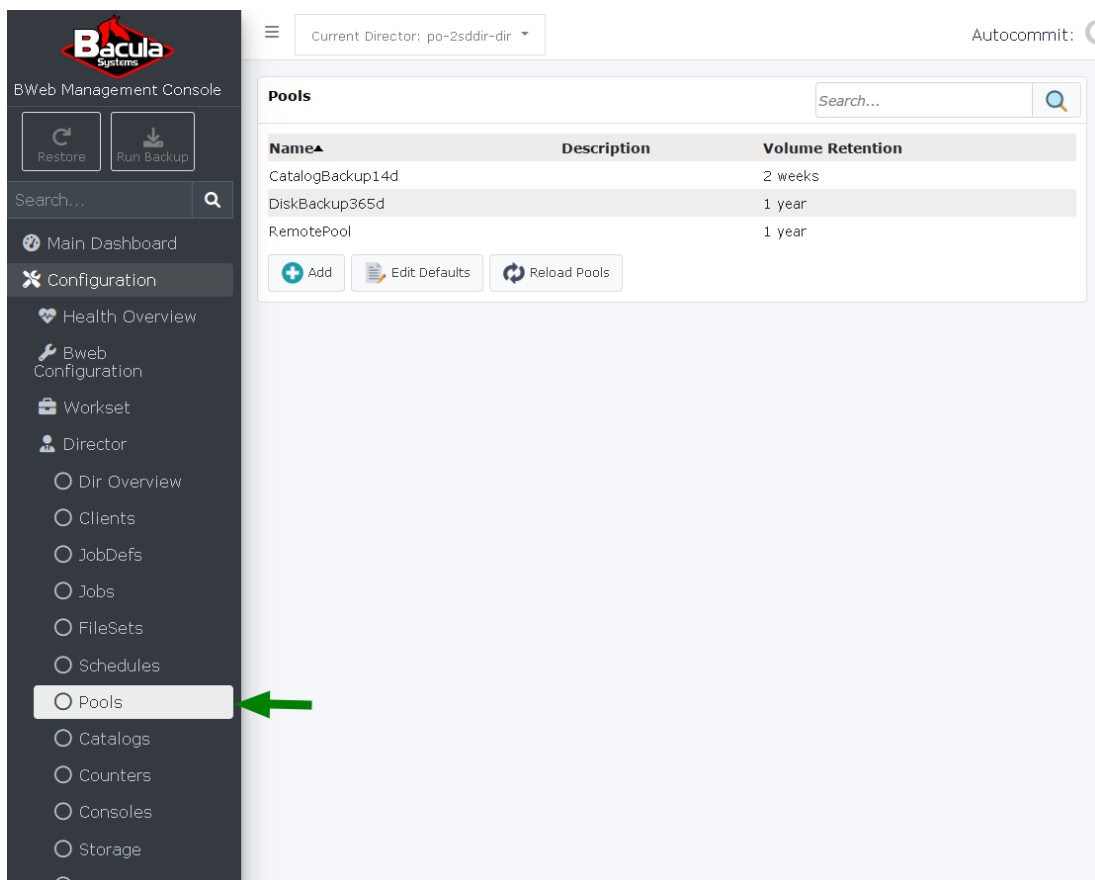
3. Click "Add".

4. Expand the "Storage" dropdown and choose "Remote Storage".

5. Type "RemotePool" in the "Pool Name" field.

6. Click "Add".
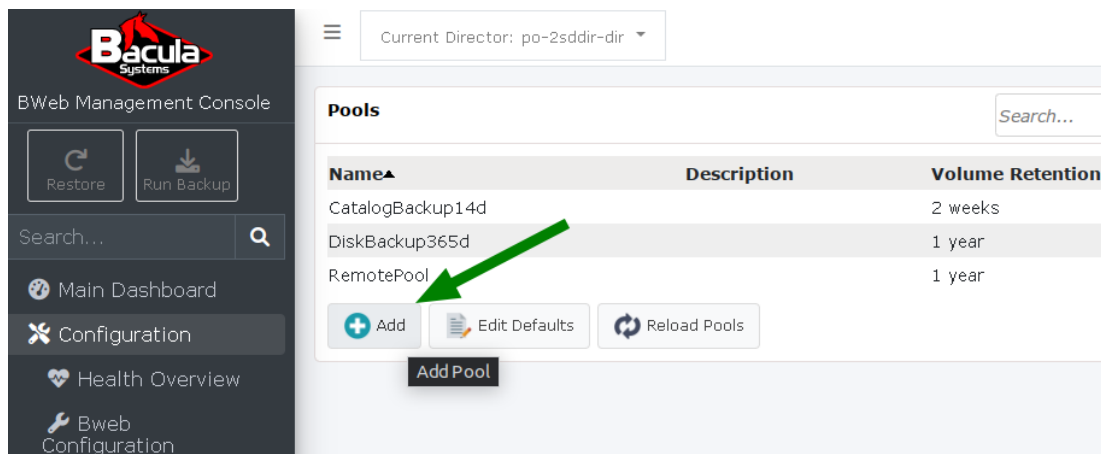
7. Click "Reload Pools".
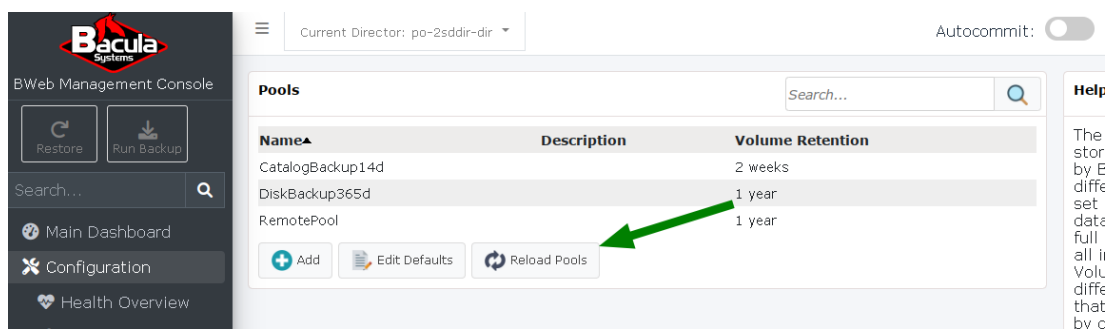
Configure Job to Use the Remote Storage

The following article presents how to configure Job to use the remote Storage Daemon.

1. Navigate to http://xxx.xxx.xxx.xxx:9180/cgi-bin/bweb/bweb.pl (xxx.xxx.xxx.xxx is the remote Storage server IP address).

2. Click "Configuration" -> "Director" -> "Jobs".
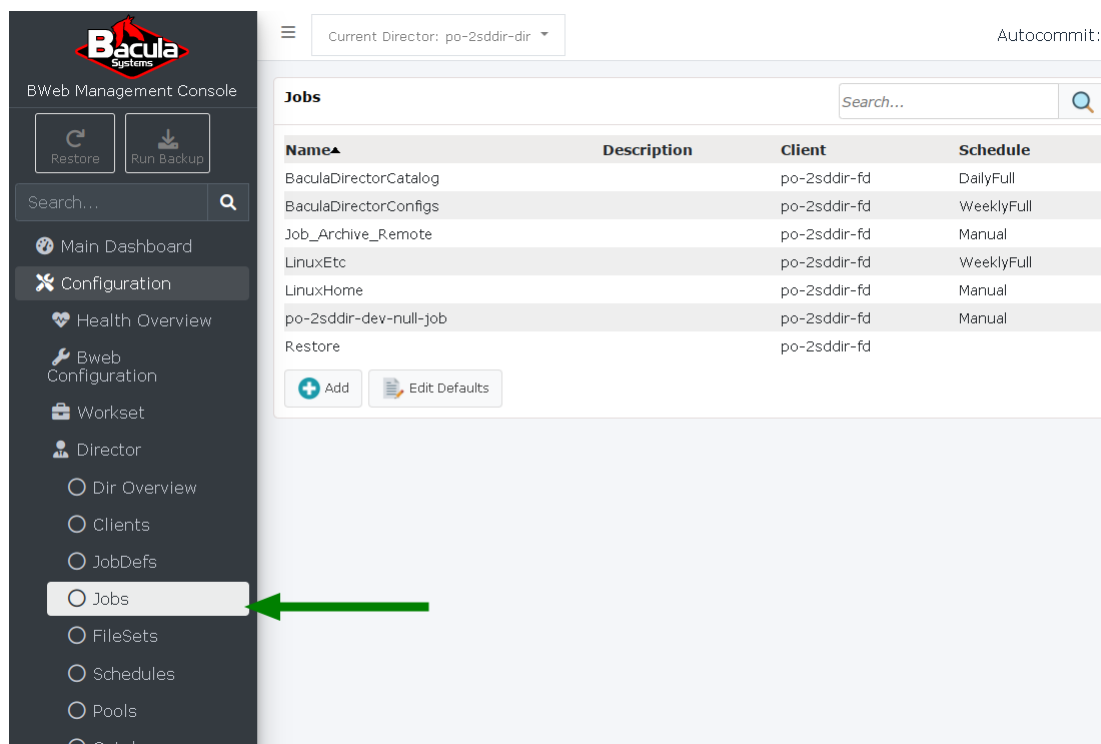
3. Click "Add".

4. In the Job Name field, type: "Job_Archive_Remote".

5. Expand the Fileset field and choose "LinuxHome".

6. Expand the Client field and choose "po-2sddir-fd".

7. Expand the Job Defs field and choose "RemoteStorage".

8. Click "Add".
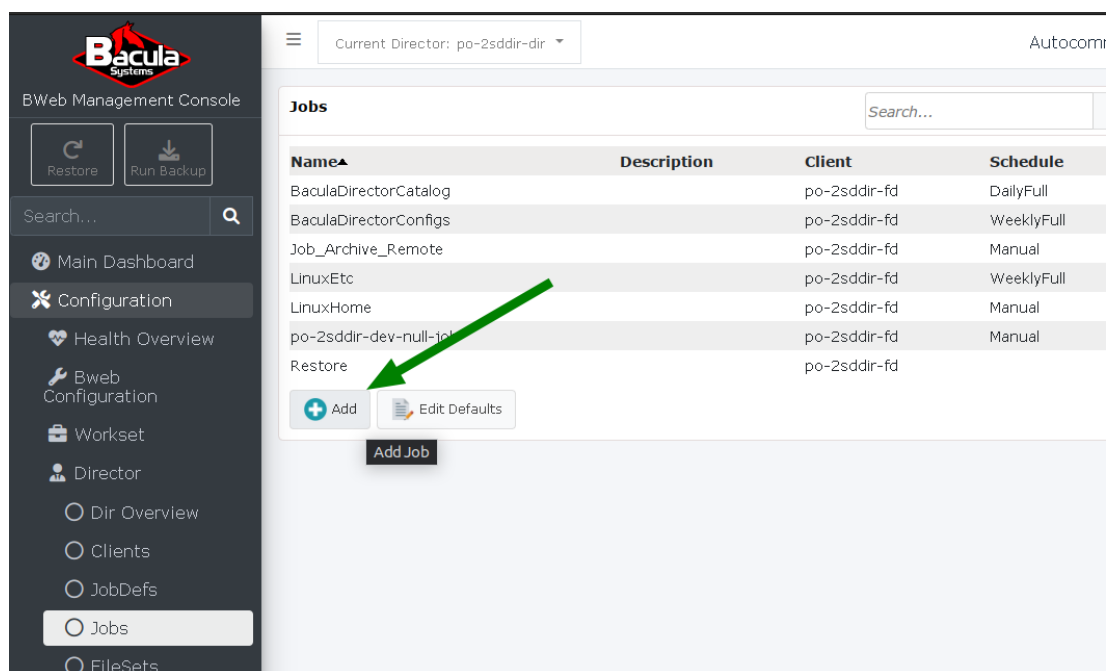
9. Click the bell.
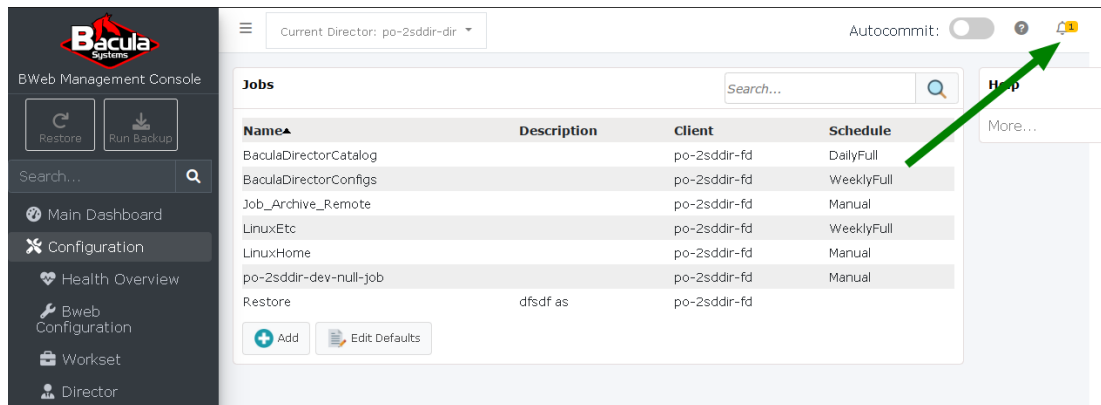
10. Click "Commit & Reload".

Example Backup Job on Remote Storage Daemon

1. Click "Run Backup".
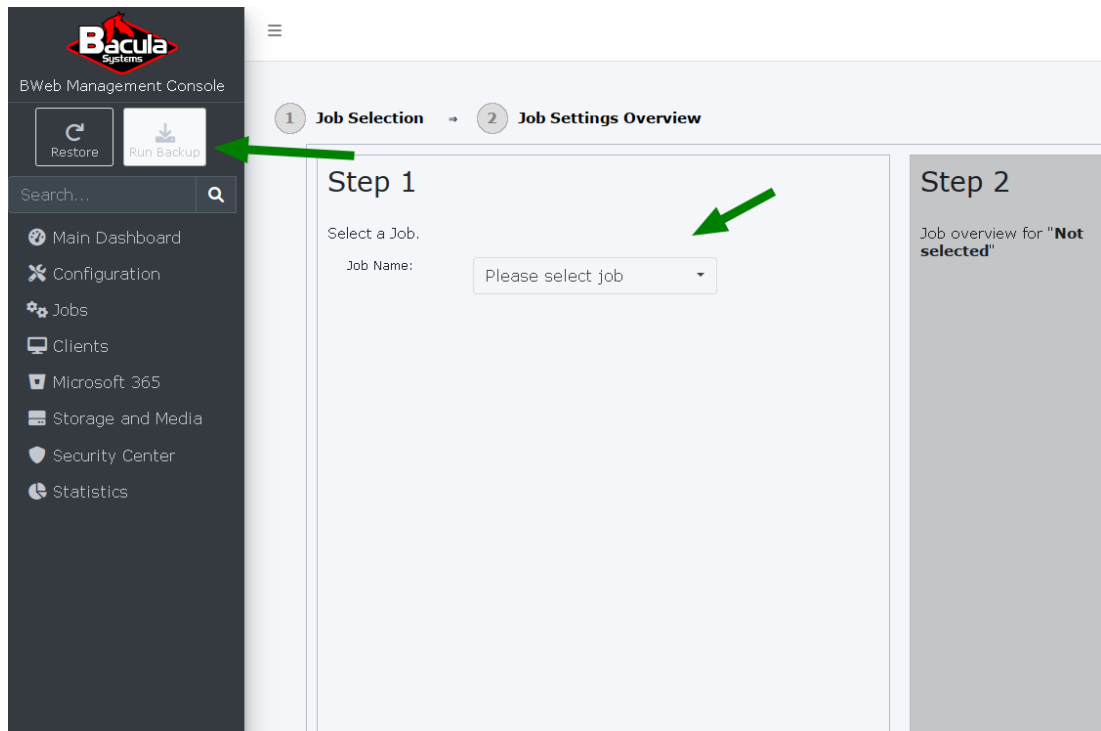
2. Click "Please select job".

3. Choose: Job_Archive_Remote.
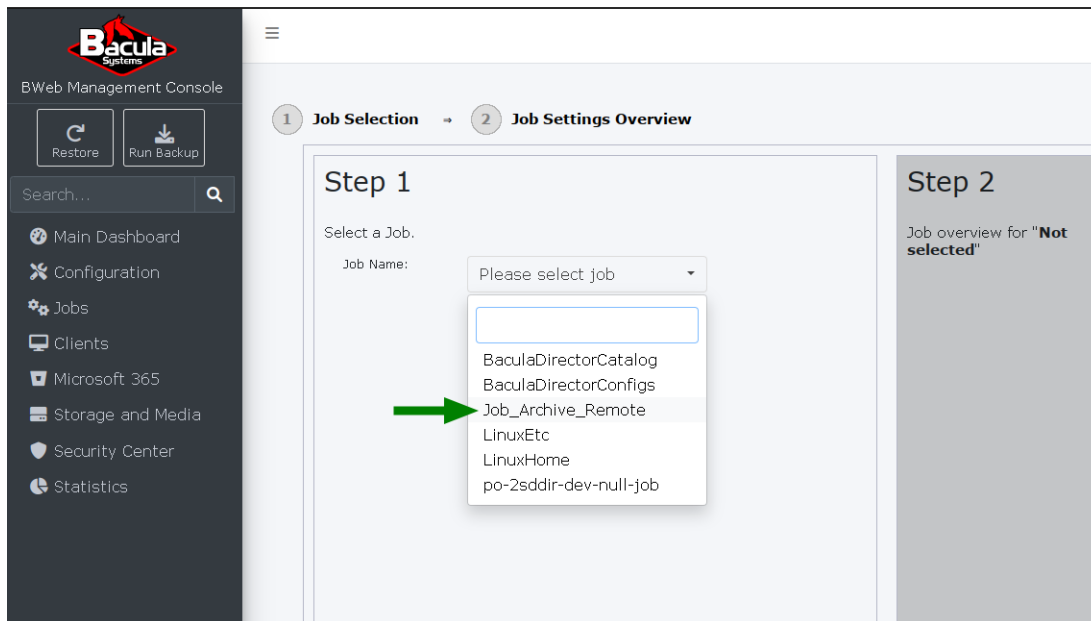
4. Click the Where image.
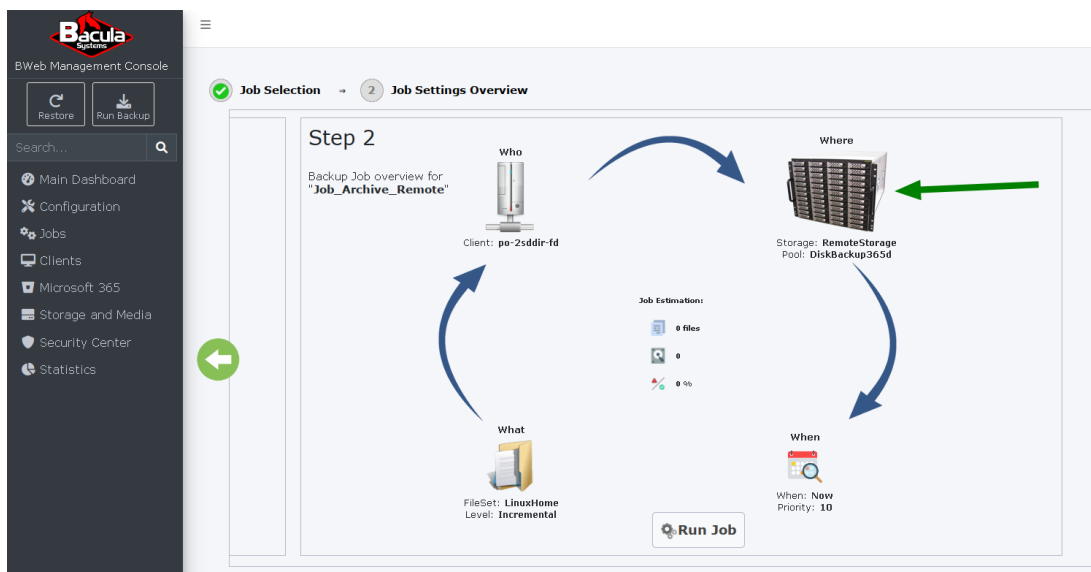
5. Click RemotePool.
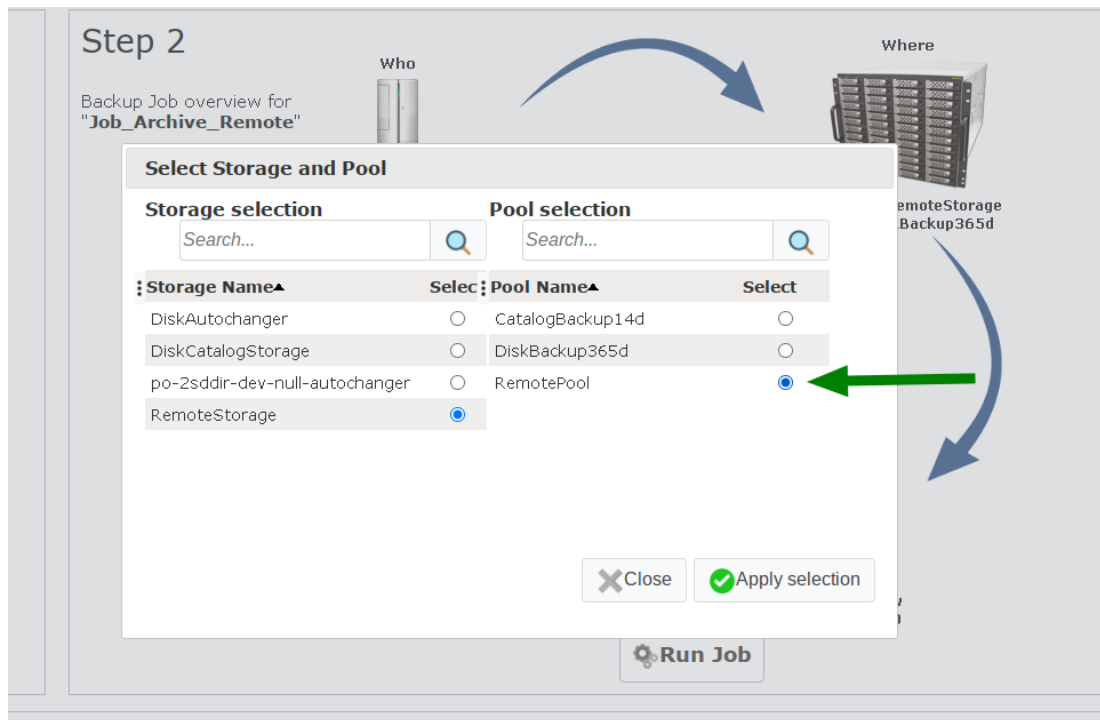
6. Click "Apply selection".

7. Click "Run Job".

2.9 Introduction To BWeb New Layout with Bacula 14.0

- *Key Changes*
- *Menu Description*
    - *Main Dashboard*
    - *Configuration*
    - *Jobs*
    - *Clients*
    - *The Client Overview*
    - *Virtual Machines*
    - *Microsoft 365*
    - *Databases*
    - *Storage and Media*
    - *Security Center*
    - *Statistics*

Bacula Enterprise 14.0 brings a new BWeb layout that simplifies its usage. All features from previous BWeb versions are kept and new ones were added. We can achieve the same tasks with the new BWeb in a more structured way.

This document describes the key changes, the new layout, its menu and also describes the main dashboard and the different overview pages.

Key Changes

BWeb 14 offers quick access to the key Backup and Restore tasks from its main screen. It offers a search tool that provides quick access to all BWeb elements. Resource name or configuration searches are not yet implemented.
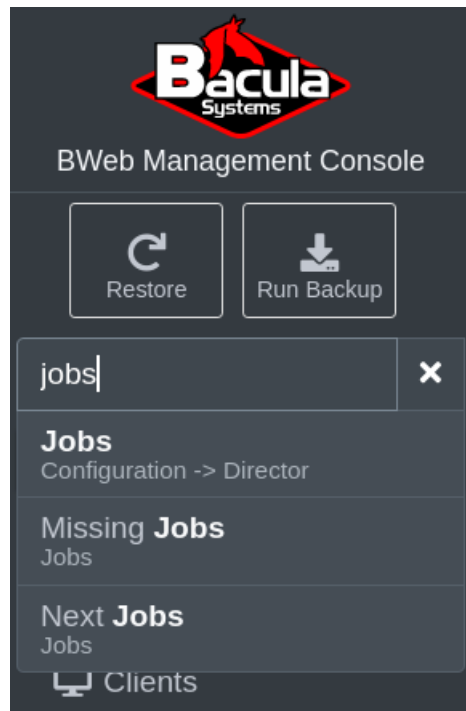

The workset page that permits you to see staged Bacula configuration changes is also easily accessible from the upper right corner via a new bell icon. There is still a `Workset` menu entry under `Configuration`
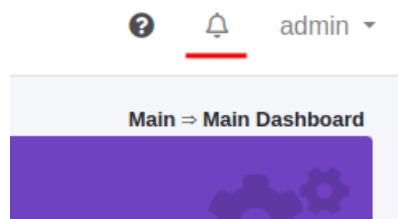
The bell can indicate that Bacula configuration changes within the workset have not yet been applied by displaying a number.

The useful wizards for setting up a new client, a new backup policy, a new storage, and more are now available from the bottom right corner with the blue (+) sign. They were available from the `Workset` page in previous BWeb versions.

The menu on the left offers all options from a single menu. Monitoring and configuration tasks are grouped under this menu. For example the jobs history page can be reached from the entry `Jobs` and `Job Overview`. Please find more information about the menu below in the *Menu Description* section.
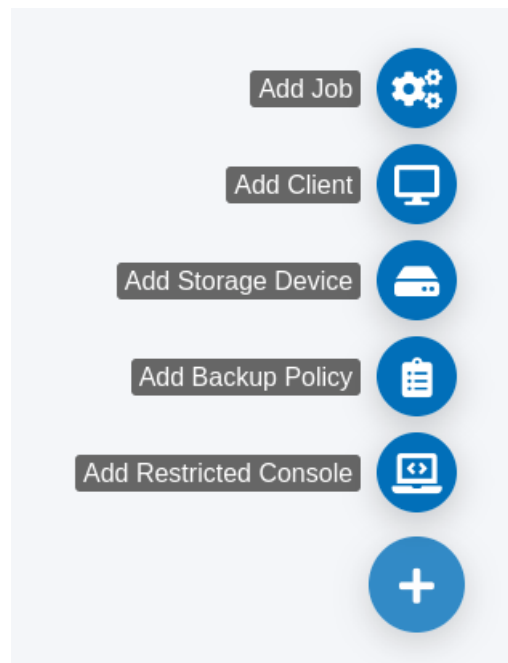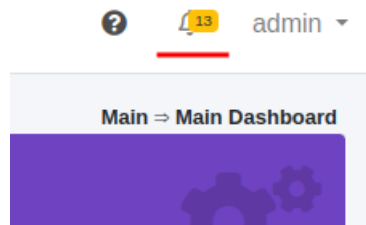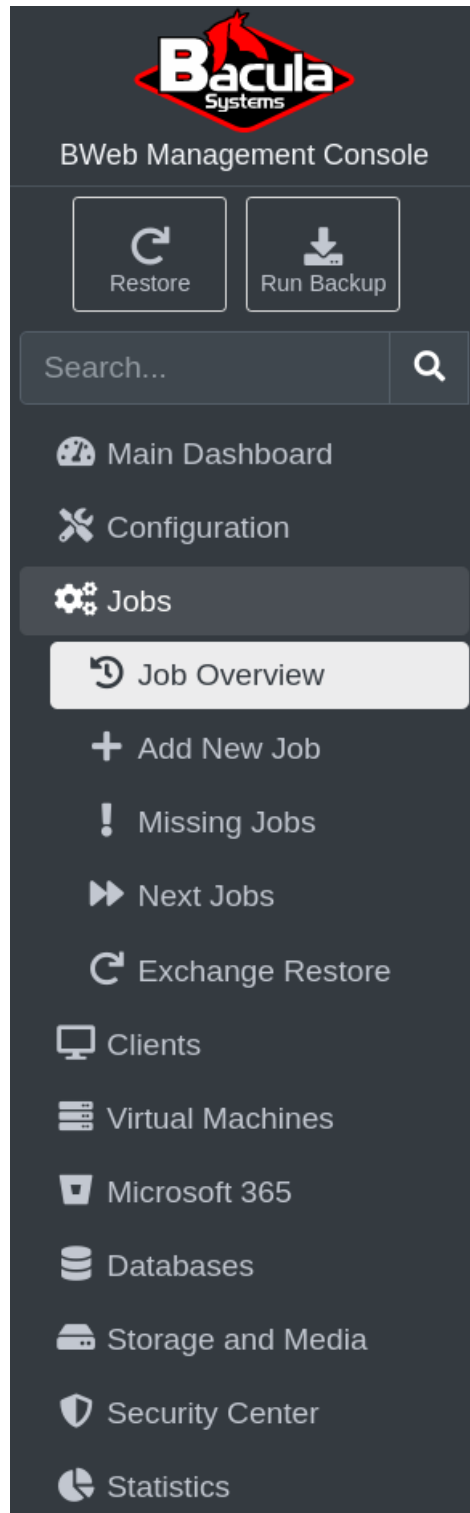
The `Configuration` menu allows you to configure Bacula deamons running locally on the server and remote ones as well. Every time a `Director`, a `File Daemon`, a `Storage Daemon` or a `Console` daemon is selected, the upper left selector allows you to verify that the changes will be applied to the correct resource. For example in the screenshot below, the `Messages` resource of the File Daemon "wa-centos7-gui-fd" will be changed.
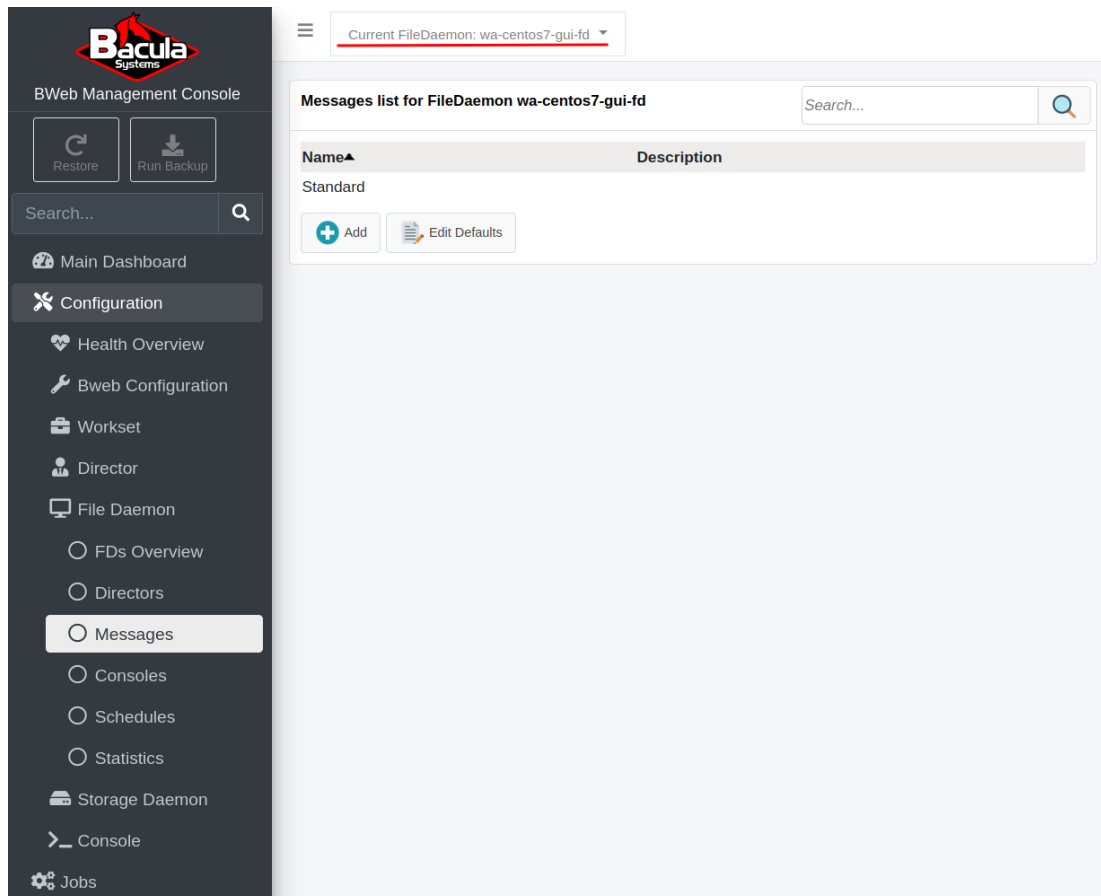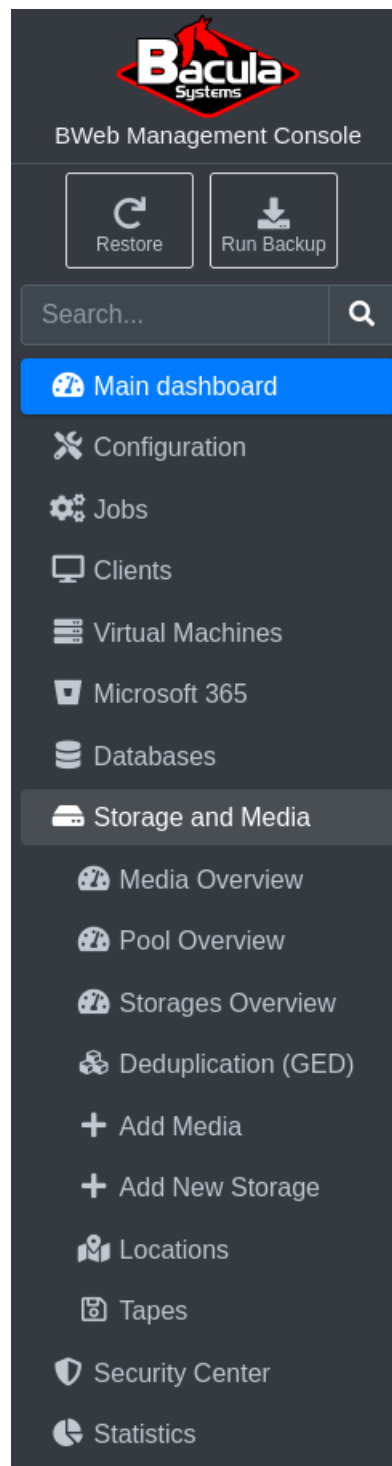

Storages, Media, and Deduplication storages are all grouped under a single main menu named `Storage and Media`. More information can be found below in the *Menu Description* section about the different menu entries.

The `VMware Center` wizard allows you to configure the automation of VMware VM backups. It is accessible from every client configuration if no VMware backup job has been run. To reach it, the menu entry `Configuration` should be chosen, then `Director` and `Clients`. Editing the client will make it available.

If VMware plugin backup jobs exist in the catalog, then a menu entry named `Virtual Machines` will be present. It offers a VM overview page, Single Item Restore and Instant Recovery for VMs. Please find more information in the *Virtual Machines* menu description below.

If database plugin backup jobs exist in the catalog, a menu entry named `Databases` will be present. It offers a Database overview page and a filter to render a table which allows you to perform some actions, like a Database backup or restore.

BWeb Management Console

Restore   Run Backup

Search...

Main dashboard

Configuration

Jobs

Clients

Virtual Machines

Microsoft 365

Databases

Storage and Media

　　Media Overview

　　Pool Overview

　　Storages Overview

　　Deduplication (GED)

　　Add Media

　　Add New Storage

　　Locations

　　Tapes

Security Center

Statistics

Menu Description

Main Dashboard

The menu that leads to different parts of BWeb is now entirely on the left side. Two buttons and one search field offer the ability to restore data, backup data and search for any BWeb element keyword. The `Main Dashboard` gives a broad overview of your backup environment and is conveniently available at the top of the menu. It shows the number of clients, the size of the Catalog, the total backed up data size and the number of Volumes and finally the number of Jobs recorded in the catalog. It also shows a list of running jobs that can be quickly accessed, canceled or stopped. The `Job Size Statistics` graph can help you to notice large increases in Job sizes, which should be investigated. The `Last Jobs` list offers quick access to recent Jobs.

Configuration

All configuration tasks are under the `Configuration` menu entry. For example, in order to edit a Fileset configuration, the `Configuration` entry will be selected, then the `Director` entry as a Fileset is defined in the Director resource, then `FileSets`. Please check the upper left selector to verify that the changes will be applied to the correct resource, as illustrated above.

The `Health Overview` page is available from the menu under `Configuration`. This is a health overview of the internal tools used for Bacula or BWeb. It gives a broad overview of BWeb capabilities and setup. Its content is key for performance fine tuning and capabilities of BWeb to manage the plugins installed.

It checks the configuration of the PostgreSQL or MySQL Catalog database based on best practices available in the psql-tuning documentation. It also gives information about your browser compatiblity with BWeb as well as the version of the Bacula components installed locally on the system running BWeb.

It also checks the available features for plugins and additional features. For example, in the section `Checking Push Bacula Windows component configuration to remote hosts`, it will check if the required Windows binaries and other modules are installed so that Windows Client configurations may be pushed from BWeb. The `Checking Available Features` section checks which tools are installed to offer wider usage of BWeb in regards to plugins, among others.

Finally, the `Testing BWeb configuration : bweb_testconfig` lists if any settings in BWeb or Bacula are not correct. Usually, any errors in this list will need to be corrected.

The `BWeb Configuration` menu allows you to review the BWeb configuration. These are the options that were setup when BWeb was first started.

The `Workset` page lists all changes made in the configuration that are not yet applied to production. The `Autocommit` slider in the top right, when enabled, automatically commits any changes you make as soon as you click 'save'. When enabled, the `Workset` page is no longer necessary and will always be blank.

The `Director` entry lists all resources that can be configured, added, or modified in the Director configuration.

The `File Daemon` (FD) allows you to manage the Clients or File Daemon parts of the backup infrastructure. From here, a new FD can be pushed with its configuration and more.

The `Storage Deamon` (SD) allows you to manage the locations where backed up data is sent. It comes with a lot of options like Deduplication, Cloud Storage, Tape Libraries, Disk Autochangers, and more.

The `Console` entry manages all the console resources configured in the Bacula infrastructure.
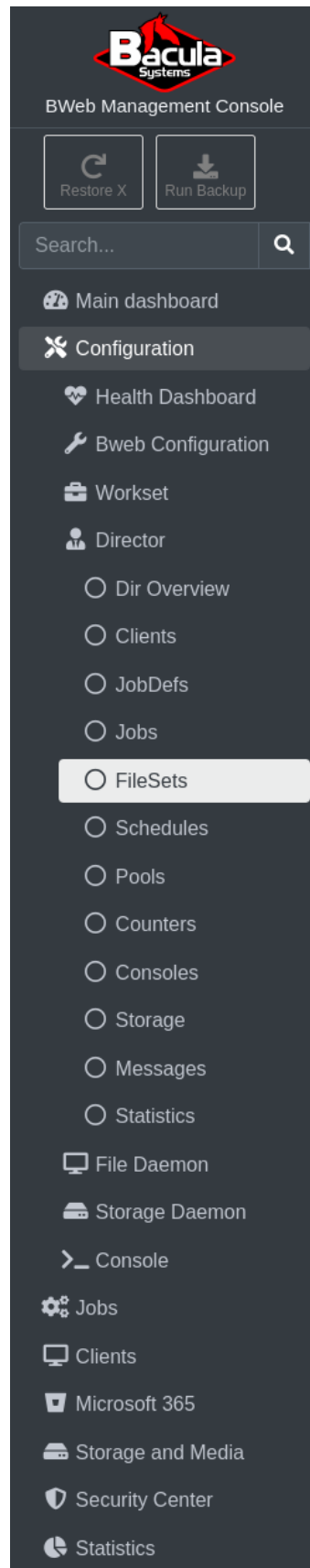
BWeb Management Console

Restore X | Run Backup

Search... 🔍

🎛 Main dashboard

🔧 Configuration

💗 Health Dashboard

🔧 Bweb Configuration

💼 Workset

👥 Director

   ○ Dir Overview

   ○ Clients

   ○ JobDefs

   ○ Jobs

   ○ FileSets

   ○ Schedules

   ○ Pools

   ○ Counters

   ○ Consoles

   ○ Storage

   ○ Messages

   ○ Statistics

🖥 File Daemon

🖴 Storage Daemon

>_ Console

⚙ Jobs

🖥 Clients

🗄 Microsoft 365

🖴 Storage and Media

🛡 Security Center

📊 Statistics

Checking Catalog Configuration

Status	Parameter	Current Setting	Test	Description
⚠	log_checkpoints	off	ne 'on'	Logs each checkpoint.
⚠	checkpoint_timeout	300 (s)	< 1200 (s)	Sets the maximum time between automatic WAL checkpoints.
⚠	work_mem	4096 (kB)	< 32768 (kB)	Sets the maximum memory to be used for query workspaces.
⚠	maintenance_work_mem	65536 (kB)	< 102400 (kB)	Sets the maximum memory to be used for maintenance operations.
⚠	shared_buffers	16384 (8kB)	< 131072 (8kB)	Sets the number of shared memory buffers used by the server.

If the Catalog configuration table is not empty, you will need to correct the MySQL/PostgreSQL configuration in order to have good performance. Please review the Catalog tuning white paper in your download area or contact Bacula Systems support for more information.

Checking Browser Compatibility

Status	Current	Description
✅	Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:96.0) Gecko/20100101 Firefox/96.0	BWeb is supported on Firefox.

Status	Time shift detection between BWeb and web browser	Server time	Client time	Time difference
✅	Time difference between BWeb and web browser is correct.	2022-01-14 14:56:19	2022-01-14 14:56:30	11 sec

Checking Local Bacula and BWeb Versions

Component Name	Version	Release Date
Director	14.0.0	22 December 2021
FileDaemon	14.0.0	22 December 2021
Storage	14.0.0	22 December 2021
BWeb	14.0.0	N/A

Checking Push Bacula Windows component configuration to remote hosts

```
INFO: Python 3.6 detected
INFO: Found python module pypsexec
INFO: Found python module winrm
INFO: Found python module pypsrp
INFO: Found command pwsh
INFO: Found NTLM Libraries
INFO: Found 4/4 modules or commands to deploy Windows binaries
```

Checking Available Features

Available	Feature Name	Description
✅	SSH Communication Keys	SSH Keys are used to run commands on remote Linux/Unix hosts.
✅	Samba Common Tools	Samba Tools are used to run commands on remote Windows hosts.
✅	Advanced Virtual Machine Restore Support	Allows restoration of selected files directly from a virtual machine backup.
✅	Exchange Mailbox Restore Support	Allows restoration of Exchange mailboxes from a VSS Exchange backup.
✅	VMware Integration Support	Enables the VMWare Integration features (VMware Center, Datacenter integration)
✖	Proxmox SDK Support	Enables the Proxmox SDK Support
✖	NDMP Backup Support	Enables the NDMP backup feature (see installation instructions)

Testing bweb configuration : bweb_testconfig

▶ Command output ✅

Main Configuration

SQL Connection

DBI:	DBI:Pg:database=bacula
User:	bacula
Password:	xxxxx
Stat Job Table:	JobHisto

General Options

Email Media:	root@localhost
Download Area ID:	xxxxxxxxxxxxxxxxxxxxx

BWeb Configuration

Fv Write Path:	/opt/bweb/spool
Template Dir:	/opt/bweb/tpl
Bconsole:	/opt/bacula/bin/bconsole -n -c /opt/bacula/etc/bconsole.conf
Wiki Url:	
Language:	en
Default Age:	7d
Default Limit:	100
Rows Per Page:	20
Display Log Time:	on
Default Size Unit:	Bi (binary bytes)
Use Number Grouping Separator:	on
Security:	0
Security ACL:	0
System Authentication:	0
Self User Restore:	0
Web Bconsole:	on
Debug:	on
Description:	Demo - Bacula Enterprise Edition 12.8.3 on CentOS 8

Bacula Configuration

Configuration Dir:	/opt/bacula/etc/conf.d
HTML Dir:	/opt/bweb/html
Workset Dir:	/opt/bacula/working/conf.d
Certs and Keys Dir:	/opt/bacula/etc/conf.d/ssl
Disable Bacula Configuration Menu:	0

Session Configuration

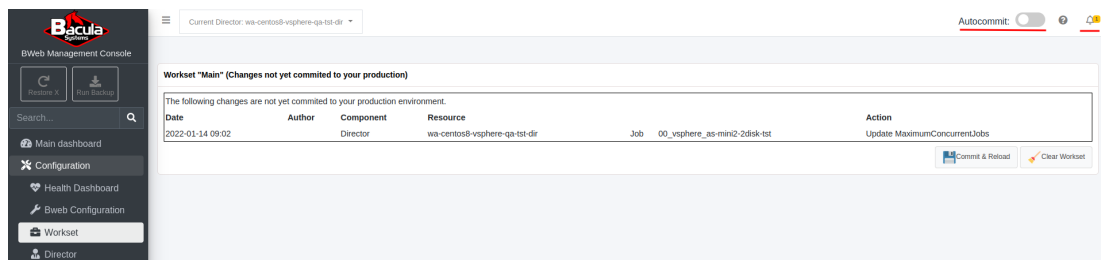|	
Always Display Main Menu	off
Display Graph Grid Lines	on
Table Pagination Pager Position	bottom
Allow page layout modification	off

Actions

- Graphite integration wizard
- Edit
- Add Autochanger
- Generate Bsys Report

Jobs

The `Jobs` menu provides the overview of all backup, Verify, Copy, and Restore jobs.

The `Job Overview` is available from the menu under `Jobs`. It offers a broad overview about backup jobs termination status at the top. The number of running Jobs, Jobs terminated with errors or warnings, and OK Jobs are shown at the top.

The table at the bottom can be filtered in order to fine tune the list of the jobs.

You may quickly add a new Job with the `Add New Job` wizard. It offers you the possibility to create backup, copy, migration or verify jobs. You may also create Linux BMR (Bare Metal Recovery) jobs (available with a LinuxBMR plugin subscription).

The `Missing Jobs` menu item will share the backup jobs that have not run for the last 7 days. It lists an overview of jobs that may need to be rescheduled manually so a consistent backup is created.

The `Next Jobs` menu item lists the next jobs planned to run based on their schedule.

If the Single Item Restore plugin is installed and the VSS plugin that can backup Exchange is used, an `Exchange Restore` menu entry will be displayed which will allow you to start an Exchange restore.

Copyright © 2025 Bacula Systems. All trademarks are the property of their respective owners.

Clients

The Client Overview

The `Client Overview` is available from the menu under `Clients`. It lists all Clients or File Daemons configured in the Director. The pie charts at the top show the versions of the configured clients and the operating systems on which they run.

A set of actions are proposed below the client table for the client selected.

The `Add New Client` menu item allows you to setup a new client easily using the `Configure a new Client resource` wizard and permits you to push the binaries and configuration to the remote client.

Copyright © 2025 Bacula Systems. All trademarks are the property of their respective owners.

The `Groups` menu item allows you to group clients in order to gather statistics about a group of client, list their jobs, allow specific BWeb users access to certain groups of clients, and more.

Virtual Machines

The `VM Overview` item under the `Virtual Machines` menu item shows the VM overview page. It lists all VMs backed up from any hypervisor plugin (vSphere, Hyper-V, Proxmox, etc). The graphics and the table filter help to monitor VM backup jobs. The action button for every `ObjectId` allows you to run a backup, a VM restore, a Single Item Restore or perform the Instant Recovery of the selected ObjectId. You may also jump directly to an individual VM backup joblog by clicking on the Job's `Job Status` icon.
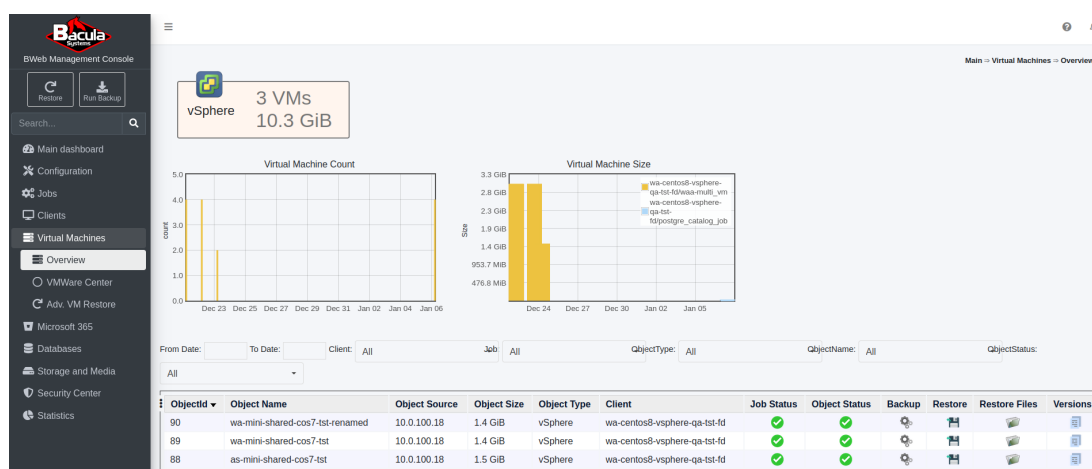
The `VMware Center` menu item allows you to manage the vCenter and ESXi host connection configurations. The different VMware wizards can also be run from here, for example the VMware datacenter autodiscovery and backup automation wizard.
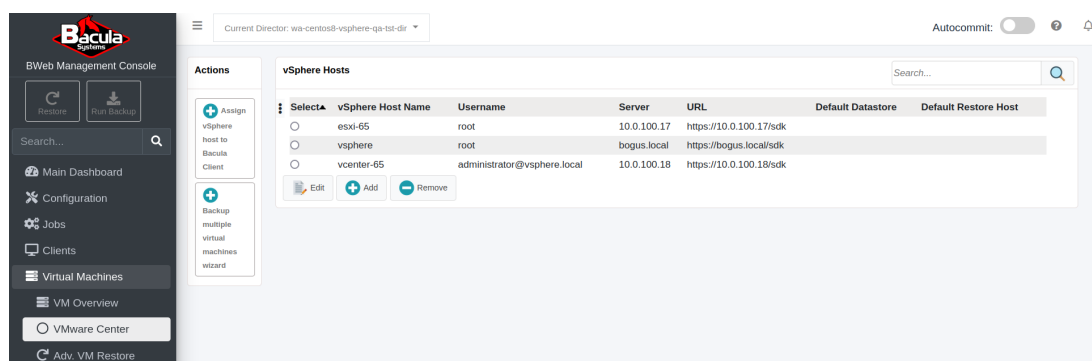
The `Adv. VM Restore` menu item gives access to the Single Item Restore and the Instant Recovery tools. Note that accessing these tools can be also achieved from the `VM Overview` of the `Virtual Machines` menu which is usually simpler as only one VM can be selected for restore purpose.
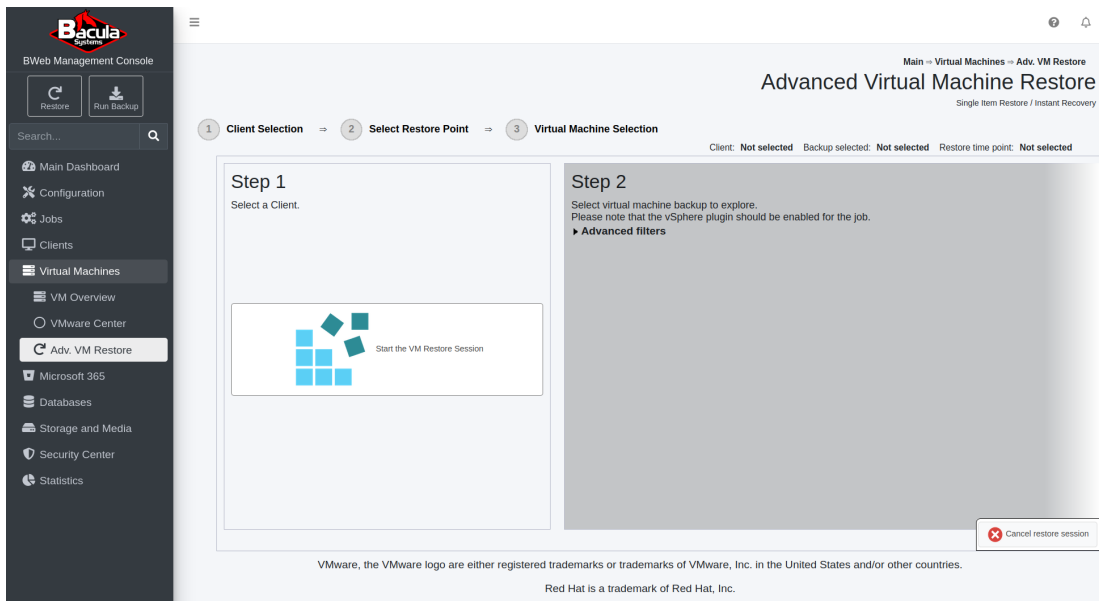
Microsoft 365

The `Microsoft 365` menu entry allows you to configure the Microsoft 365 plugin if subscribed.

Databases

The `Databases` menu entry is visible if you have any databases backed up using one of the database plugins. This `Databases` page gives a detailed overview of backed up databases. A filter can be applied to the table. New backups or restores can be triggered from the table. Specific joblogs can also be accessed from the table.

Copyright © 2025 Bacula Systems. All trademarks are the property of their respective owners.
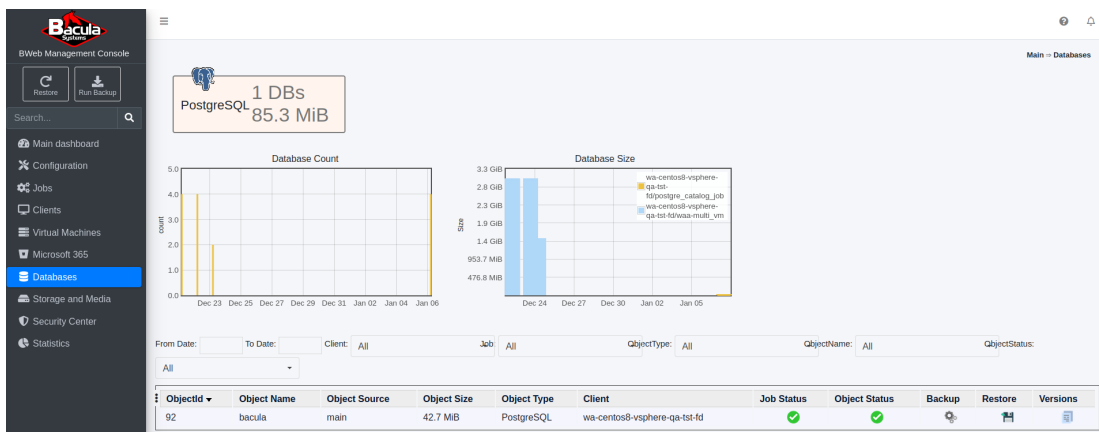
Storage and Media

This menu entry allows you to manage and configure your Storage Daemons (SD). This is here that you can verify disk volumes or tapes are well recycled, that some are free for next backup jobs and more.

The `Media Overview` page is available from the menu under `Storage and Media`. It offers a broad overview of the media status on top. The filter permits to list media by selecting different options. An action offered by the buttons at the bottom of the table can be applied to selected media.

The `Pool Overview` page is available from the menu under `Storage and Media`. It offers a Pool view, for example every pool configuration in regards to retention and number volumes used. It also displays some graphics about pool occupancy.

The `Storages Overview` is available from the menu under `Storage and Media`. It gives a broad

overview of a the selected storage's status. The jobs running, their speed, the tape or disk volumes loaded, the storage capacity of the storage and more is displayed on this page. A network test can be executed as well.

The `Deduplicaton (GED)` overview is available from the menu under `Storage and Media`. It produces detailed information about any deduplication storage setup. Information about dedup ratio, the disk occupancy and any issue with the deduplication engine are shown.

The + `Add Media` menu item allows you to label a tape or add new disk volumes.

The + `Add New Storage` menu item starts the new Storage resource wizard to add Device resources into an existing storage, create a new Storage Daemon (SD) Device resources or create a new Director Storage resource.
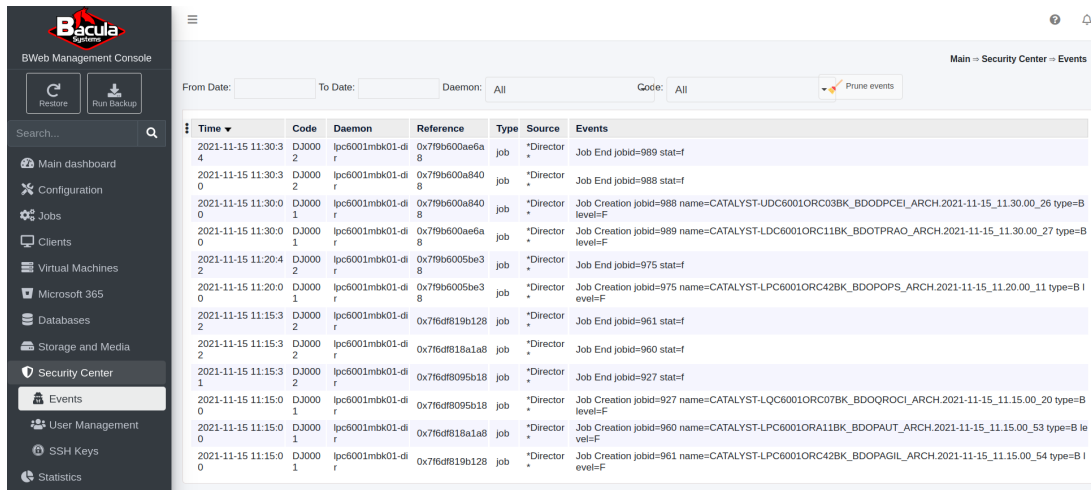
The `Locations` menu item allows you to manage media location, usually tapes.

The `Tapes` menu item offers two tape-only actions: `Load Media` and `Eject Media`.

Security Center

The `Security Center` is the place where security features like events, users, SSL certificates and ssh keys can be managed.

The `Events` page allows you to filter and review all events recorded. It allows you to see reloads of the Director configuration, connections to the console, and much more. Please see events for more information.

The `User Management` menu item allows you to manage BWeb user access and select the permissions allowed for each configured user.

The `SSH Keys` menu item sllows you to set up SSL certificates and SSH keys.

Statistics

This page offers plenty of options in regards to Bacula activities and statistics.

The `Stats Dashboard` is available from the menu under `Statistics`. Different kinds of statistics can be generated to monitor the Bacula infrastructure, see if any unusual activities in regards to data volume backed up or number of files changes, generate reports for decision makers and more.

For example, statistics can be generated about:

- The longest running jobs
- The slowest or fastest jobs
- The job sizes
- The files backed up per day
- The size of backup jobs by time
- The number of files per backup job by time

The `Jobs Timeline` gives the opportunity to generate graphics about different parameters with jobs.
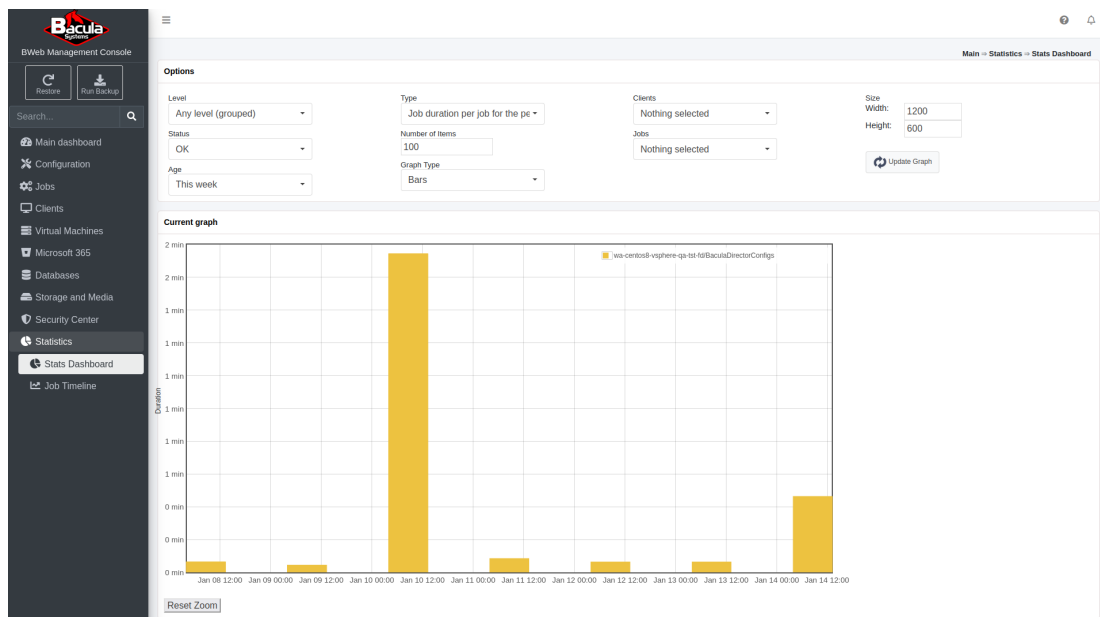
2.10 BWeb Authentication Methods

The following article presents information on how to integrate Bweb authentication with LDAP and ACTIVE DIRECTORY.

Requirements

LDAP SERVER

- user "admin" created in the Ldap directory tree
- user "bwebadmin" created in the Ldap directory tree, this user will be used has admin in BWEB.

BWEB SERVER

- auth.backend = "htpasswd" method should be working before activating the LDAP authentication <— if anything goes wrong you can revert to this stage.
- create "bwebadmin" user and give administration permissions.

Configure the HTTPD

To configure the HTTPD, follow the steps below:

1. Create an admin user password.

   Use the following command to configure `/opt/bweb/etc/htpasswd.bweb` - create a password for the user: admin:

   ```
   [root@nasabacula etc]# htpasswd -c /opt/bweb/etc/htpasswd.bweb admin
   New password:
   ```

(continues on next page)

```
Re-type new password:
Adding password for user admin
```

2. Configure Lighttpd configuration.

```
# vi /opt/bweb/etc/httpd.conf
...
server.modules += ( "mod_auth") # if not already done
auth.backend = "htpasswd"
auth.backend.htpasswd.userfile = "/opt/bweb/etc/htpasswd.bweb"
auth.require = ( "/" =>
(
"method" => "basic",
"realm" => "Password protected area",
"require" => "valid-user"
)
)
```

3. Restart Bweb and check if it runs.

```
[root@nasabacula etc]# service bweb restart
Redirecting to /bin/systemctl restart bweb.service

[root@nasabacula etc]# service bweb status

Redirecting to /bin/systemctl status bweb.service
* bweb.service - Bacula Enterprise Bweb
Loaded: loaded (/usr/lib/systemd/system/bweb.service; enabled; vendor␣
↪preset: disabled)
Active: active (running) since Fri 2020-10-30 13:31:59 WET; 3s ago
Process: 12826 ExecStart=/usr/sbin/lighttpd -f /opt/bweb/etc/httpd.conf␣
↪(code=exited, status=0/SUCCESS)
Main PID: 12828 (lighttpd)
Tasks: 1 (limit: 49777)
Memory: 800.0K
CGroup: /system.slice/bweb.service
└─12828 /usr/sbin/lighttpd -f /opt/bweb/etc/httpd.conf

out 30 13:31:59 nasabacula systemd[1]: Stopped Bacula Enterprise Bweb.
out 30 13:31:59 nasabacula systemd[1]: Starting Bacula Enterprise Bweb...
out 30 13:31:59 nasabacula lighttpd[12826]: 2020-10-30 13:31:59: (server.
↪c.1521) server started (lighttpd/1.4.54)
out 30 13:31:59 nasabacula systemd[1]: Started Bacula Enterprise Bweb.
```

4. Log in to Bweb.
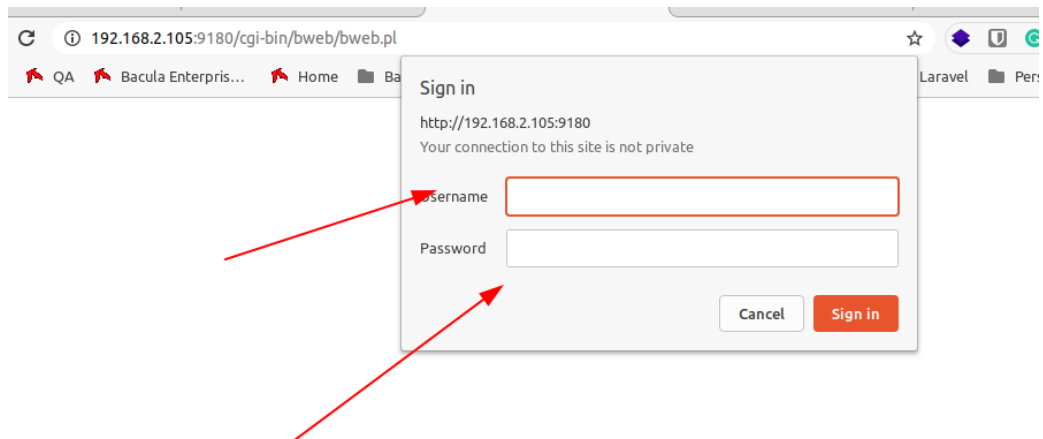
5. Click Bweb –> Main –> Configuration –> Configure BWeb.
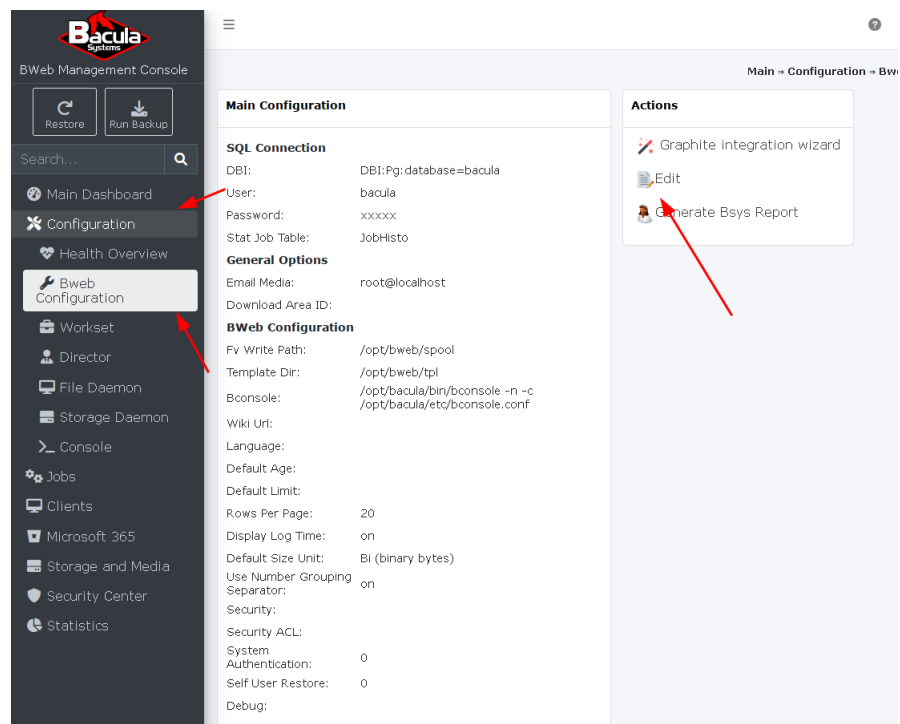
6. Choose the following options.

Copyright © 2025 Bacula Systems. All trademarks are the property of their respective owners.
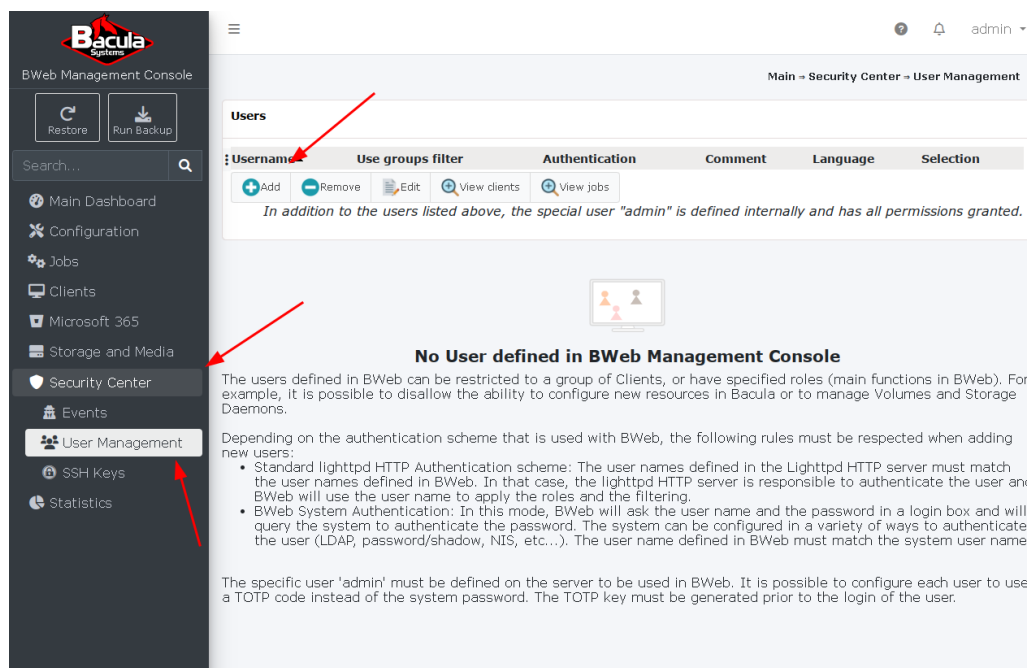
7. Log out and log in again as admin user.
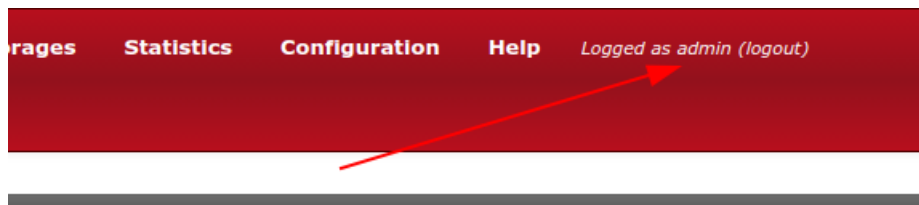
8. Check if "User Management" button appears.

9. Add a new User.

10. Set administrator permissions.

11. Log out and log in with the other administrator user.

> **Note:** Remember to create an htpasswd password for this user.

```
# htpasswd /opt/bweb/etc/htpasswd.bweb pedro
New password:
Re-type new password:
Updating password for user pedro
```

Important: To test the login and logout, you need to refresh the page or clean cache files.

LDAP/Active Directory - Authentication

Note: If you are experiencing any troubles with this method, it is recommended to use the Configure the HTTPD method.

---

1. Edit the `/opt/bweb/etc/httpd.conf` file.

   Comment

```
#auth.backend = "htpasswd"
#auth.backend.htpasswd.userfile = "/opt/bweb/etc/htpasswd.bweb"
```

2. Add the following for LDAP server (for Linux):

```
server.modules += ( "mod_authn_ldap" )
auth.backend = "ldap"
auth.backend.ldap.hostname = "192.168.1.8" <--- change to you Ldap Server
auth.backend.ldap.base-dn = "ou=people,dc=my-domain,dc=com" <-- change
→to you Ldap Users Directory tree
auth.backend.ldap.filter = "(uid=$)"
auth.backend.ldap.bind-dn = "cn=admin,dc=my-domain,dc=com" <----- Ldap
→Authentication user to check users
auth.backend.ldap.bind-pw = "xxxxxx" <----- Password User

# "disable" for requiring passwords, "enable" for allowing empty
→passwords
auth.backend.ldap.allow-empty-pw = "disable"
```

or

Add the following for ACTIVE DIRECTORY server (for Windows):

```
server.modules += ( "mod_authn_ldap" )
auth.backend = "ldap"
auth.backend.ldap.hostname = "192.168.1.8" <--- change to you Ldap Server

auth.backend.ldap.base-dn = "ou=Users,dc=supportlab,dc=baculasystems,
→dc=com" <--- Change to you Ldap Users Directory tree

auth.backend.ldap.filter = "(cn=?)"                                    ↵
→      <--- Change to Username Active Directory Field confrotation

auth.backend.ldap.bind-dn = "CN=ad-admin,CN=Users,DC=supportlab,
→DC=baculasystems,DC=com"      <--- AD Administration username to check
→users

auth.backend.ldap.bind-pw = "xxxxxx"                                   ↵
→      <--- Password User

# "disable" for requiring passwords, "enable" for allowing empty
→passwords
auth.backend.ldap.allow-empty-pw = "disable"


##### YOU CAN ALSO DO THIS ########

auth.backend.ldap.bind-dn ="SUPPORTLAB\ad-admin"
auth.backend.ldap.bind-pw = "xxxxxx"
```

Note: To Activate LDAPS, copy the CA certificate to Bweb Server and add the following Configuration to HTTPD. Conf:

```
auth.backend.ldap.hostname = "192.168.1.8" <--- change to you Ldap Server | DO
NOT SPECIFY SSL PORT

auth.backend.ldap.starttls = "enable"        auth.backend.ldap.ca-file = "/etc/
```

```
openldap/certs/ca.cert.pem" auth.backend.ldap.allow-empty-pw = "disable"
```

Example configuration for LDAP/Active directory:

```
Set 09 10:26:41 po-ldap slapd[10200]: conn=1039 op=4 SRCH base="ou=People,
↪dc=supportlab,dc=lan" scope=2 deref=0 filter="(uid=user1)"
Set 09 10:26:41 po-ldap slapd[10200]: conn=1039 op=4 SRCH attr=1.1
Set 09 10:26:41 po-ldap slapd[10200]: <= bdb_equality_candidates: (uid) not
↪indexed
Set 09 10:26:41 po-ldap slapd[10200]: conn=1039 op=4 SEARCH RESULT tag=101
↪err=0 nentries=1 text=
Set 09 10:26:41 po-ldap slapd[10200]: conn=1043 fd=22 ACCEPT from IP=10.0.110.
↪174:58650 (IP=0.0.0.0:389)
Set 09 10:26:41 po-ldap slapd[10200]: conn=1043 op=0 EXT oid=1.3.6.1.4.1.1466.
↪20037
Set 09 10:26:41 po-ldap slapd[10200]: conn=1043 op=0 STARTTLS
Set 09 10:26:41 po-ldap slapd[10200]: conn=1043 op=0 RESULT oid= err=0 text=
Set 09 10:26:41 po-ldap slapd[10200]: conn=1043 fd=22 TLS established tls_
↪ssf=256 ssf=256   <--------------
Set 09 10:26:41 po-ldap slapd[10200]: conn=1043 op=1 BIND dn="uid=user1,
↪ou=People,dc=supportlab,dc=lan" method=128
Set 09 10:26:41 po-ldap slapd[10200]: conn=1043 op=1 BIND dn="uid=user1,
↪ou=People,dc=supportlab,dc=lan" mech=SIMPLE ssf=0
Set 09 10:26:41 po-ldap slapd[10200]: conn=1043 op=1 RESULT tag=97 err=0 text=
Set 09 10:26:41 po-ldap slapd[10200]: conn=1043 op=2 UNBIND
Set 09 10:26:41 po-ldap slapd[10200]: conn=1043 fd=22 closed
```

Note: See Tips and Tricks for LDAP/Active Directory.

3. Restart Bweb service.

4. Try to Login with user "admin" and "bwebadmin" and check if exists any error on the logs.

   The logs:

```
tail -f /var/log/messages

Mar 16 12:15:11 bee lighttpd: 2018-03-16 12:15:10: (mod_auth.c.525)
↪password doesn't match for /cgi-bin/bweb/bweb.pl username: user1, IP:
↪192.168.1.11
Mar 16 12:15:18 bee lighttpd: 2018-03-16 12:15:18: (mod_auth.c.525)
↪password doesn't match for /bweb/natcompare.js username: user1, IP:
↪192.168.1.11
Mar 16 12:15:52 bee lighttpd: 2018-03-16 12:15:52: (mod_auth.c.525)
↪password doesn't match for /bweb/remove.png username: user1, IP: 192.
↪168.1.11
Mar 16 12:21:49 bee systemd: Stopping Bacula Enterprise Bweb...
Mar 16 12:21:49 bee lighttpd: 2018-03-16 12:21:49: (server.c.2005)
↪server stopped by UID = 0 PID = 1
Mar 16 12:21:49 bee systemd: Starting Bacula Enterprise Bweb...
Mar 16 12:21:49 bee lighttpd: 2018-03-16 12:21:49: (server.c.1412)
↪server started (lighttpd/1.4.48)
```

(continues on next page)

```
Mar 16 12:21:49 bee systemd: Started Bacula Enterprise Bweb.
Mar 16 12:32:36 bee dhclient[672]: DHCPREQUEST on enp0s3 to 192.168.1.1␣
→port 67 (xid=0x3bc9f94d)
```

Note: Users

The Users that need authentication via Ldap must exist in Bweb with the same login name. The following script copies the LDAP users to the BWEB database users table. Adapt this script to customer LDAP infrastructure.

```perl
#!/usr/bin/perl
use Net::LDAP;
use Data::Dumper;
use DBI;

my $driver = "Pg";
my $database = "bacula";
my $dsn = "DBI:$driver:dbname = $database;host = 192.168.1.28;␣
→port = 5432";
my $userid = "postgres";
my $password = "";
my $dbh = DBI->connect($dsn, $userid, $password, { RaiseError => 1␣
→})
   or die $DBI::errstr;

print "Opened database successfully\n";

my $uid = "cn=admin,dc=my-domain,dc=com";
my $bindPass = "password";
my $ldapServer = "192.168.1.8";
my $userLogin = "displayName";
my $userPass = "sambaLMPassword";

# connect to ldap server
$ldap = Net::LDAP -> new ($ldapServer) || die "Could not connect to␣
→server\n";
 # bind to ldap server
 $ldap -> bind($uid, password => $bindPass);

$result = $ldap->search( # perform a search
 base => "ou=people,dc=my-domain,dc=com",
 filter => "(objectClass=*)"
 );

die $result->error if $result->code;

printf "COUNT: %s\n", $result->count;
foreach my $entry ($result->entries) {
 # $entry->dump;
 my $l = $entry->get_value($userLogin);
 my $p = $entry->get_value($userPass);
```

```
 my $stmt = qq(INSERT INTO bweb_user (username,passwd) VALUES ('$l','
→$p'));
 my $rv = $dbh->do($stmt) or die $DBI::errstr;


}
print "===============================================\n";


$ldap->unbind;
```

Tips and Tricks for LDAP/Active Directory

TRY TO USE LDAPSEARCH to find your LDAP directory Tree and Authentication Test

You can simulate the Bweb authentication method with the Ldapsearch tool, please check the following example:

```
ldapsearch -x -b "CN=Users,DC=supportlab,DC=baculasystems,DC=com" -H ldap://
→XXX.XXX.XXX.XXX -D
"CN=ad-admin,CN=users,DC=supportlab,DC=baculasystems,DC=com" -w "password"␣
→sAMAccountName=*
```

The above command will list all users under "CN=Users,DC=supportlab,DC=baculasystems, DC=com", if you set to "sAMAccountName=user", you will see the ldif card of the user.

With the ldapsearch tool, you can check if the base DN is correctly set and if the user can access the AD Directory and read the directory tree.

Check the following pictures and URLs to help you get better acquainted with the Ldapsearch tool: https://linux.die.net/man/1/ldapsearch

```
# extended LDIF
#
# LDAPv3
# base <CN=Users,DC=supportlab,DC=baculasystems,DC=com> with scope subtree
# filter: cn=pedro
# requesting: ALL
#

# pedro, Users, supportlab.baculasystems.com
dn: CN=pedro,CN=Users,DC=supportlab,DC=baculasystems,DC=com
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: user
cn: pedro
sn: Oliveira
givenName: Pedro
distinguishedName: CN=pedro,CN=Users,DC=supportlab,DC=baculasystems,DC=com
instanceType: 4
whenCreated: 20180403151958.0Z
whenChanged: 20201102162912.0Z
```

```
displayName: Pedro Oliveira
uSNCreated: 53322
uSNChanged: 2789194
name: pedro
objectGUID:: zp8x/XPKw06N5KqzVbLSXQ==
userAccountControl: 66048
badPwdCount: 0
codePage: 0
countryCode: 0
badPasswordTime: 131673917475251005
lastLogoff: 0
lastLogon: 131673917773373354
pwdLastSet: 132488081524100730
primaryGroupID: 513
objectSid:: AQUAAAAAAUVAAAARJDCrCTkmXj8FpvsXQQAAA==
accountExpires: 9223372036854775807
logonCount: 0
sAMAccountName: pedro
sAMAccountType: 805306368
userPrincipalName: pedro@supportlab.baculasystems.com
lockoutTime: 0
objectCategory: CN=Person,CN=Schema,CN=Configuration,DC=supportlab,DC=baculasy
stems,DC=com
dSCorePropagationData: 20200824131529.0Z
dSCorePropagationData: 20181127035418.0Z
dSCorePropagationData: 16010101000417.0Z
lastLogonTimestamp: 131673917773373354

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```
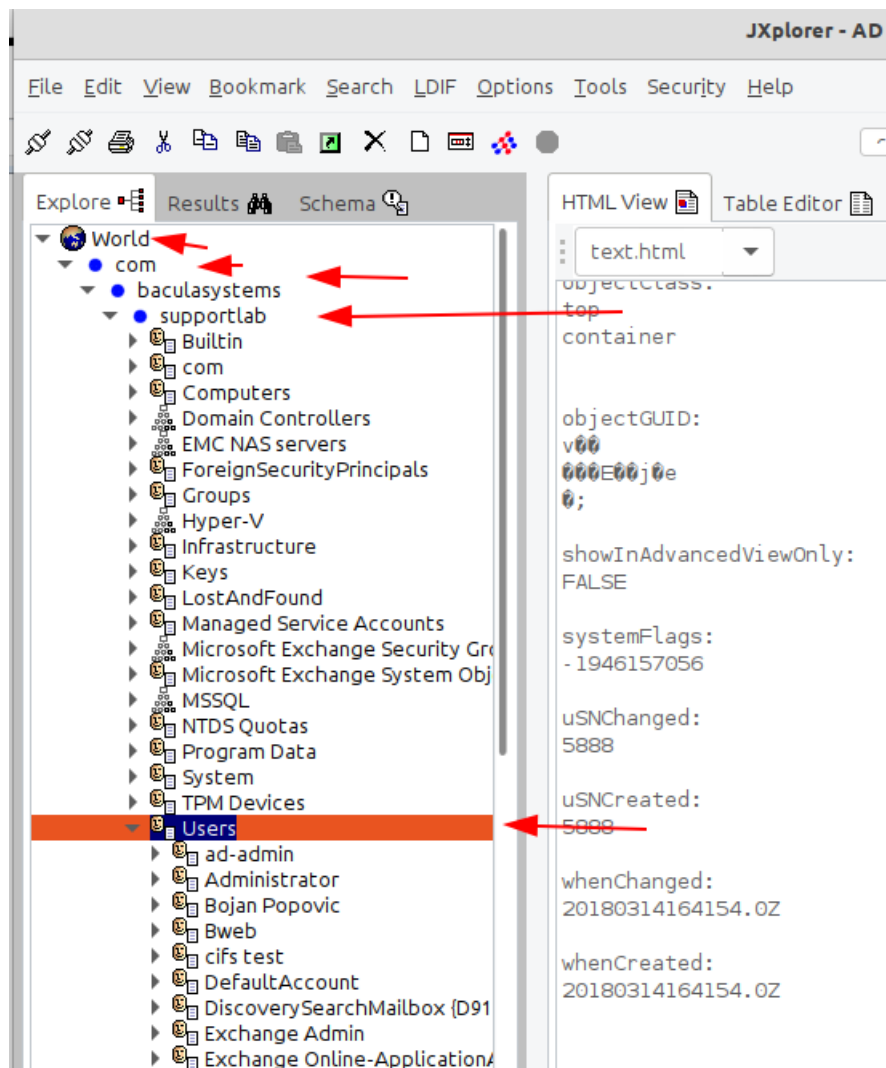
- `auth.backend.ldap.hostname`: server address

- `auth.backend.ldap.base-dn`: tree were are the user saved

- `auth.backend.ldap.filter`: filter to apply to obtain the users

- `auth.backend.ldap.bind-dn`: login to use to bind to LDAP server

- `auth.backend.ldap.bind-pw`: associated password

`auth.backend.ldap.bind-dn` and `auth.backend.ldap.bind-pw` parameters are only necessary if the LDAP server require a specific account to be able to access the different information.

BWeb Authentication Active Directory Groups

The following article is an extension to the LDAP/Active Directory with the option to authenticate users belonging to a certain group.

BWeb Installation with Apache

The following chapter presents an extension of the LDAP/Active Directory with the option to authenticate users belonging to a certain group.

BWeb Installation with Apache and LDAP Authentication Centos 7

Note: If you are using RHEL 8 or Oracle Linux 8 or any derivatives, check those pages in regards to bweb-apache.conf and httpd.conf:

- Bweb Installation with Apache 2.4 and Ldap Authentication Redhat Oracle 8
- BWeb Installation with Apache 2.4 and Ldap with Group Authentication Redhat Oracle 8

1. Install Apache.

```
# Install Apache
yum install httpd -y

# Open Firewall
firewall-cmd --permanent --add-service=http
firewall-cmd --permanent --add-service=https
firewall-cmd --reload

#Start Apache Service

 root@po-bweb-apache ~]# systemctl start httpd
 [root@po-bweb-apache ~]# systemctl status httpd
 * httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled;␣
↪vendor preset: disabled)
   Active: active (running) since Qua 2022-04-06 12:26:20 EDT; 7s ago
     Docs: man:httpd(8)
           man:apachectl(8)
 Main PID: 10915 (httpd)
   Status: "Processing requests..."
   CGroup: /system.slice/httpd.service
           ├─10915 /usr/sbin/httpd -DFOREGROUND
           ├─10916 /usr/sbin/httpd -DFOREGROUND
           ├─10917 /usr/sbin/httpd -DFOREGROUND
           ├─10918 /usr/sbin/httpd -DFOREGROUND
           ├─10919 /usr/sbin/httpd -DFOREGROUND
           └─10920 /usr/sbin/httpd -DFOREGROUND

Abr 06 12:26:20 po-bweb-apache systemd[1]: Starting The Apache HTTP␣
```

```
→Server...
Abr 06 12:26:20 po-bweb-apache systemd[1]: Started The Apache HTTP␣
→Server.
```

2. Configure Bweb.

```
# Install Bweb Repo
vi /etc/yum.repos.d/baculasystems.repo

[Bacula-Enterprise-BWeb]
name = BWeb Management Suite
baseurl = https://www.baculasystems.com/dl/Customer-123456/rpms/bweb/12.
→2.2/rhel7-64/
enabled = 1
protect = 0
gpgcheck = 1

[Bacula-Enterprise-DAG]
name = Bacula Systems DAG for BWeb
baseurl = https://www.baculasystems.com/dl/DAG/rhel7-64/
enabled = 1
protect = 0
gpgcheck = 0
```

3. Install Bweb and Bweb Dependencies.

```
# Install Bweb Dependecies

 yum install perl-DBI perl-DBD-mysql perl-DBD-Pg perl-GDGraph \
 perl-HTML-Template perl-Time-modules gd perl-Date-Calc \
 perl-Bit-Vector perl-Expect perl-GD \
 perl-GD-Text-Util perl-IO-Tty perl-CGI-Session perl-JSON \
 perl-JSON-XS \program{lighttpd}

# Install Bweb

yum install bacula-enterprise-bweb

/opt/bweb/bin/install_bweb.sh
```

4. Configure Apache

```
# The apache.conf.sample file available on /opt/bweb/etc/ does not work␣
→for Centos 7 Apache 2.4. Created a new one with the following settings:

vi /etc/httpd/httpd.conf

####################

ServerRoot "/etc/httpd"

Listen 9180
```

```
Include conf.modules.d/*.conf

User bacula
Group bacula

ServerAdmin root@localhost

<Directory />
    Options FollowSymLinks
    AllowOverride None
    Order deny,allow
    Deny from all
</Directory>

DocumentRoot "/opt/bweb/html"

<Directory "/opt/bweb/html">
    Options Indexes FollowSymLinks
    AllowOverride All
    Order allow,deny
    Allow from all
</Directory>

<IfModule dir_module>
    DirectoryIndex index.html
</IfModule>

<FilesMatch "^\.ht">
    Order allow,deny
    Deny from all
    Satisfy All
</FilesMatch>

ErrorLog "logs/error_log"

LogLevel warn

<IfModule log_config_module>
    LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}
↪i\"" combined
    LogFormat "%h %l %u %t \"%r\" %>s %b" common

    <IfModule logio_module>
      LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-
↪Agent}i\" %I %O" combinedio
    </IfModule>

    CustomLog "logs/access_log" combined
</IfModule>

<IfModule alias_module>
```

```
    ScriptAlias /cgi-bin/bweb/ "/opt/bweb/cgi/"
</IfModule>

Alias /bweb/fv "/tmp"
<Location /bweb/fv>
    Options None
    Order allow,deny
    Allow from all
</Location>

Alias /bweb /opt/bweb/html

<Directory "/opt/bweb/cgi">
    AllowOverride None
    Options None
    Require all granted
</Directory>

<Directory "/opt/bweb/cgi">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
    PassEnv BWEBBASE
    PassEnv BWEBLOG
    PassEnv BWEBCONF
    SetENV PERLLIB "/opt/bweb/lib"
</Directory>

<IfModule mime_module>
    TypesConfig /etc/mime.types
    AddType application/x-compress .Z
    AddType application/x-gzip .gz .tgz
    AddHandler cgi-script .cgi
    AddType text/html .shtml
    AddOutputFilter INCLUDES .shtml

    AddType application/javascript                          js
    AddType application/json                                json
    AddType image/png                                       png
    AddType image/jpeg                                      jpeg jpg jpe
    AddType image/gif                                       gif
    AddType text/html                                       html htm
    AddType text/css                                        css

</IfModule>

AddDefaultCharset UTF-8

<IfModule mime_magic_module>
    MIMEMagicFile conf/magic
</IfModule>
```

```
EnableSendfile on

Timeout 300
KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 5
UseCanonicalName Off
AccessFileName .htaccess
ServerTokens Minimal
ServerSignature On
HostnameLookups Off

<IfModule ssl_module>
    SSLRandomSeed startup builtin
    SSLRandomSeed connect builtin
</IfModule>

IncludeOptional conf.d/*.conf

################################
```

5. Restart Apache

```
[root@po-bweb-apache ~]# systemctl restart httpd
[root@po-bweb-apache ~]# systemctl status httpd

* httpd.service - The Apache HTTP Server
    Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled;␣
→vendor preset: disabled)
    Active: active (running) since Qui 2022-04-07 12:00:30 EDT; 6min ago
      Docs: man:httpd(8)
            man:apachectl(8)
  Process: 9118 ExecStop=/bin/kill -WINCH ${MAINPID} (code=exited,␣
→status=0/SUCCESS)
 Main PID: 9123 (httpd)
    Status: "Total requests: 34; Current requests/sec: 0; Current␣
→traffic:   0 B/sec"
    CGroup: /system.slice/httpd.service
            ├─9123 /usr/sbin/httpd -DFOREGROUND
            ├─9124 /usr/sbin/httpd -DFOREGROUND
            ├─9125 /usr/sbin/httpd -DFOREGROUND
            ├─9127 /usr/sbin/httpd -DFOREGROUND
            ├─9128 /usr/sbin/httpd -DFOREGROUND
            ├─9131 /usr/sbin/httpd -DFOREGROUND
            ├─9133 /usr/sbin/httpd -DFOREGROUND
            ├─9134 /usr/sbin/httpd -DFOREGROUND
            ├─9157 /usr/sbin/httpd -DFOREGROUND
            ├─9158 /usr/sbin/httpd -DFOREGROUND
            └─9159 /usr/sbin/httpd -DFOREGROUND

   Abr 07 12:00:30 po-bweb-apache systemd[1]: Stopped The Apache HTTP␣
```

```
↪Server.
  Abr 07 12:00:30 po-bweb-apache systemd[1]: Starting The Apache HTTP␣
↪Server...
  Abr 07 12:00:30 po-bweb-apache httpd[9123]: [Thu Apr 07 12:00:30.
↪402034 2022] [env:warn] [pid 9123] AH01506: PassEnv variable BWEBBASE␣
↪was undefined
  Abr 07 12:00:30 po-bweb-apache httpd[9123]: [Thu Apr 07 12:00:30.
↪402206 2022] [env:warn] [pid 9123] AH01506: PassEnv variable BWEBLOG␣
↪was undefined
  Abr 07 12:00:30 po-bweb-apache httpd[9123]: [Thu Apr 07 12:00:30.
↪402218 2022] [env:warn] [pid 9123] AH01506: PassEnv variable BWEBCONF␣
↪was undefined
  Abr 07 12:00:30 po-bweb-apache httpd[9123]: [Thu Apr 07 12:00:30.
↪402343 2022] [alias:warn] [pid 9123] AH00671: The Alias directive in /
↪etc/httpd/conf.d/bweb.conf at line 14 will probably never match␣
↪because it overlaps an earlier Alias.
  Abr 07 12:00:30 po-bweb-apache httpd[9123]: [Thu Apr 07 12:00:30.
↪402374 2022] [alias:warn] [pid 9123] AH00671: The Alias directive in /
↪etc/httpd/conf.d/bweb.conf at line 22 will probably never match␣
↪because it overlaps an earlier Alias.
  Abr 07 12:00:30 po-bweb-apache systemd[1]: Started The Apache HTTP␣
↪Server.
```

6. Check the Browser Page.

7. Configure Apache Htpasswd Authentication.

```
[root@po-bweb-apache ~]# htpasswd -c /opt/bweb/etc/bweb.htpasswd admin
```

```
New password:
Re-type new password:
Adding password for user admin



vi /etc/httpd/conf.d/bweb.conf


<Directory /opt/bweb/cgi>
        Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch

##### Uncomment this Lines for Htpasswd Authentication ####

        AuthType Basic
        AuthName Bweb
        AuthUserFile /opt/bweb/etc/bweb.htpasswd
        Require valid-user
########################

        AllowOverride None
</Directory>
ScriptAlias /cgi-bin/bweb /opt/bweb/cgi

Alias /bweb/fv /opt/bweb/spool
<Directory "/var/spool/bweb">
    Options None
    AllowOverride AuthConfig
    Order allow,deny
    Allow from all
</Directory>

Alias /bweb /opt/bweb/html
<Directory "/opt/bweb/html">
    Options None
    AllowOverride AuthConfig
    # For Apache 2.4
    Require all granted
    # For Apache 2.2
    #Order allow,deny
    #Allow from all
</Directory>
```

8. Restart Apache.

```
[root@po-bweb-apache ~]# systemctl restart httpd
[root@po-bweb-apache ~]# systemctl status httpd
* httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled;␣
↪vendor preset: disabled)
   Active: active (running) since Qui 2022-04-07 12:17:34 EDT; 3s ago
     Docs: man:httpd(8)
           man:apachectl(8)
  Process: 9309 ExecStop=/bin/kill -WINCH ${MAINPID} (code=exited,␣
```

```
↪status=0/SUCCESS)
Main PID: 9314 (httpd)
  Status: "Processing requests..."
  CGroup: /system.slice/httpd.service
           ├─9314 /usr/sbin/httpd -DFOREGROUND
           ├─9315 /usr/sbin/httpd -DFOREGROUND
           ├─9316 /usr/sbin/httpd -DFOREGROUND
           ├─9317 /usr/sbin/httpd -DFOREGROUND
           ├─9318 /usr/sbin/httpd -DFOREGROUND
           └─9319 /usr/sbin/httpd -DFOREGROUND

Abr 07 12:17:34 po-bweb-apache systemd[1]: Starting The Apache HTTP␣
↪Server...
Abr 07 12:17:34 po-bweb-apache httpd[9314]: [Thu Apr 07 12:17:34.386406␣
↪2022] [env:warn] [pid 9314] AH01506: PassEnv variable BWEBBASE was␣
↪undefined
Abr 07 12:17:34 po-bweb-apache httpd[9314]: [Thu Apr 07 12:17:34.386604␣
↪2022] [env:warn] [pid 9314] AH01506: PassEnv variable BWEBLOG was␣
↪undefined
Abr 07 12:17:34 po-bweb-apache httpd[9314]: [Thu Apr 07 12:17:34.386616␣
↪2022] [env:warn] [pid 9314] AH01506: PassEnv variable BWEBCONF was␣
↪undefined
Abr 07 12:17:34 po-bweb-apache httpd[9314]: [Thu Apr 07 12:17:34.386805␣
↪2022] [alias:warn] [pid 9314] AH00671: The Alias directive in /etc/
↪httpd/conf.d/bweb.conf at line 14 will probably never match because it␣
↪overlaps an earlier Alias.
Abr 07 12:17:34 po-bweb-apache httpd[9314]: [Thu Apr 07 12:17:34.386839␣
↪2022] [alias:warn] [pid 9314] AH00671: The Alias directive in /etc/
↪httpd/conf.d/bweb.conf at line 22 will probably never match because it␣
↪overlaps an earlier Alias.
Abr 07 12:17:34 po-bweb-apache systemd[1]: Started The Apache HTTP␣
↪Server.
```

9. Test Authentication on Browser.
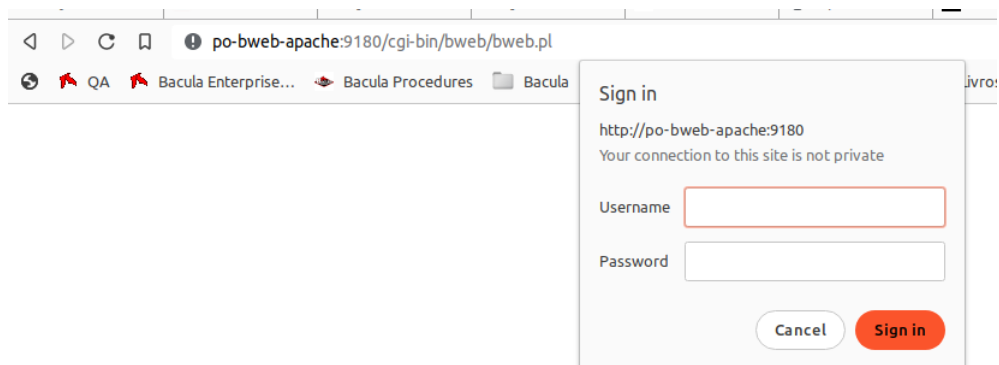
10. Configure Apache Ldap Authentication.

```
In the file vi /etc/httpd/conf.d/bweb.conf
```

```
Change the Following Htpasswd

########################
        AuthType Basic
        AuthName Bweb
        AuthUserFile /opt/bweb/etc/bweb.htpasswd
        Require valid-user
########################

To AD/LDAP Authenctication

########################
AuthType Basic
AuthName "Apache LDAP authentication"
AuthBasicAuthoritative Off
AuthBasicProvider ldap
AuthLDAPURL "ldap://SERVER_IP/CN=Users,DC=tech,DC=local?sAMAccountName?
↪sub?(objectClass=*)"
AuthLDAPBindDN "CN=ad-admin,CN=Users,DC=supportlab,DC=baculasystems,
↪DC=com"
AuthLDAPBindPassword "xxxxxx"
Require valid-user
########################
```

11. Enable User Authentication.

12. Restart Apache and Test Authentication.

```
[root@po-bweb-apache ~]# systemctl restart httpd
[root@po-bweb-apache ~]# systemctl status httpd
* httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled;␣
↪vendor preset: disabled)
   Active: active (running) since Qui 2022-04-07 12:17:34 EDT; 3s ago
     Docs: man:httpd(8)
           man:apachectl(8)
  Process: 9309 ExecStop=/bin/kill -WINCH ${MAINPID} (code=exited,␣
↪status=0/SUCCESS)
 Main PID: 9314 (httpd)
   Status: "Processing requests..."
   CGroup: /system.slice/httpd.service
           ├─9314 /usr/sbin/httpd -DFOREGROUND
           ├─9315 /usr/sbin/httpd -DFOREGROUND
           ├─9316 /usr/sbin/httpd -DFOREGROUND
           ├─9317 /usr/sbin/httpd -DFOREGROUND
           ├─9318 /usr/sbin/httpd -DFOREGROUND
           └─9319 /usr/sbin/httpd -DFOREGROUND

Abr 07 12:17:34 po-bweb-apache systemd[1]: Starting The Apache HTTP␣
↪Server...
Abr 07 12:17:34 po-bweb-apache httpd[9314]: [Thu Apr 07 12:17:34.386406␣
↪2022] [env:warn] [pid 9314] AH01506: PassEnv variable BWEBBASE was␣
↪undefined
Abr 07 12:17:34 po-bweb-apache httpd[9314]: [Thu Apr 07 12:17:34.386604␣
↪2022] [env:warn] [pid 9314] AH01506: PassEnv variable BWEBLOG was␣
↪undefined
Abr 07 12:17:34 po-bweb-apache httpd[9314]: [Thu Apr 07 12:17:34.386616␣
↪2022] [env:warn] [pid 9314] AH01506: PassEnv variable BWEBCONF was␣
↪undefined
Abr 07 12:17:34 po-bweb-apache httpd[9314]: [Thu Apr 07 12:17:34.386805␣
↪2022] [alias:warn] [pid 9314] AH00671: The Alias directive in /etc/
↪httpd/conf.d/bweb.conf at line 14 will probably never match because it␣
↪overlaps an earlier Alias.
Abr 07 12:17:34 po-bweb-apache httpd[9314]: [Thu Apr 07 12:17:34.386839␣
↪2022] [alias:warn] [pid 9314] AH00671: The Alias directive in /etc/
↪httpd/conf.d/bweb.conf at line 22 will probably never match because it␣
↪overlaps an earlier Alias.
Abr 07 12:17:34 po-bweb-apache systemd[1]: Started The Apache HTTP␣
↪Server.
```

Bweb Installation with Apache 2.4 and Ldap Authentication RHEL 8

This article gives specific insight on how to set up ldap authentication on RHEL 8 and any derivatives. See for the whole procedure to enable it in the main article.

Configuration Settings

vi /etc/httpd/conf.d/bweb-apache.conf

```
Listen 9180
Define BWEBBASE "/opt/bweb"
Define BWEBLOG  "/opt/bacula/working"
Define BWEBCONF "/opt/bweb/etc/bweb.conf"

LDAPVerifyServerCert Off

### Apache Bweb With Without SSL
<VirtualHost *:9180>

    ServerName po-oracle8

        DocumentRoot "${BWEBBASE}/html"
        CustomLog "${BWEBLOG}/bweb-access.log" combined
        ErrorLog "${BWEBLOG}/bweb-error.log"
        <Directory "${BWEBBASE}/html">
                # Apache 2.2
                <IfModule !mod_authz_core.c>
                        Order Allow,Deny
                        Allow from all
                </IfModule>
                # Apache 2.4
                <IfModule mod_authz_core.c>
                        Require all granted
                </IfModule>
        </Directory>
        <IfModule alias_module>
                Alias /bweb/fv "/tmp"
                Alias /bweb ${BWEBBASE}/html
                ScriptAlias /cgi-bin/bweb/ "${BWEBBASE}/cgi/"
        </IfModule>
        <Directory "${BWEBBASE}/cgi">
                AllowOverride None
                SetEnv BWEBBASE $BWEBBASE
                SetEnv BWEBLOG $BWEBLOG
                SetEnv BWEBCONF $BWEBCONF
                SetENV PERLLIB "${BWEBBASE}/lib"


        AllowOverride AuthConfig

            #Bweb Configuration htpasswd
        ##############################
```

```
            # Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
            # AuthType Basic
            # AuthName Bweb
            # AuthUserFile /opt/bweb/etc/bweb.htpasswd
            # Require valid-user

            ###############################

            #Bweb Configuration htpasswd
            ###############################

        AuthType Basic
            AuthName "Auth"
            AuthBasicProvider ldap
            AuthLDAPURL ldap://10.0.200.21:389/ou=Users,dc=supportlab,
→dc=baculasystems,dc=com?sAMAccountName?sub?(objectClass=*)
            AuthLDAPBindDN "CN=ad-admin,CN=Users,DC=supportlab,
→DC=baculasystems,DC=com"
            AuthLDAPBindPassword "Bacula18"
            Require valid-user

            ###############################



    </Directory>
    <Location /bweb/fv>
            # Apache 2.2
            <IfModule !mod_authz_core.c>
                    Order Allow,Deny
                    Allow from all
            </IfModule>
            # Apache 2.4
            <IfModule mod_authz_core.c>
                    Require all granted
            </IfModule>
    </Location>
</VirtualHost>


### Bweb With SSL Certificates
Listen 9443
<VirtualHost *:9443>

    ServerName po-oracle8.supportlablan.baculasytems.com
    SSLEngine on
    SSLCertificateFile "/home/sslkey/po-oracle8.cert"
    SSLCertificateKeyFile "/home/sslkey/po-oracle8.key"

    DocumentRoot "${BWEBBASE}/html"
        ServerName po-oracle8.supportlablan.baculasystems.com
        CustomLog "${BWEBLOG}/bweb-access.log" combined
```

(continues on next page)

```
        ErrorLog "${BWEBLOG}/bweb-error.log"
        <Directory "${BWEBBASE}/html">
                # Apache 2.2
                <IfModule !mod_authz_core.c>
                        Order Allow,Deny
                        Allow from all
                </IfModule>
                # Apache 2.4
                <IfModule mod_authz_core.c>
                        Require all granted
                </IfModule>
        </Directory>
        <IfModule alias_module>
                Alias /bweb/fv "/tmp"
                Alias /bweb ${BWEBBASE}/html
                ScriptAlias /cgi-bin/bweb/ "${BWEBBASE}/cgi/"
        </IfModule>
        <Directory "${BWEBBASE}/cgi">
                AllowOverride None
                SetEnv BWEBBASE $BWEBBASE
                SetEnv BWEBLOG $BWEBLOG
                SetEnv BWEBCONF $BWEBCONF
                SetENV PERLLIB "${BWEBBASE}/lib"


                AllowOverride AuthConfig

                #Bweb Configuration htpasswd
                ##############################

                # Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
                # AuthType Basic
                # AuthName Bweb
                # AuthUserFile /opt/bweb/etc/bweb.htpasswd
                # Require valid-user

                ##############################

                #Bweb Configuration htpasswd
                ##############################

                AuthType Basic
                AuthName "Auth"
                AuthBasicProvider ldap
                AuthLDAPURL ldap://10.0.200.21:389/ou=Users,dc=supportlab,
→dc=baculasystems,dc=com?sAMAccountName?sub?(objectClass=*)
                AuthLDAPBindDN "CN=ad-admin,CN=Users,DC=supportlab,
→DC=baculasystems,DC=com"
                AuthLDAPBindPassword "Bacula18"
                Require valid-user

                ##############################
```

```
        </Directory>
        <Location /bweb/fv>
                # Apache 2.2
                <IfModule !mod_authz_core.c>
                        Order Allow,Deny
                        Allow from all
                </IfModule>
                # Apache 2.4
                <IfModule mod_authz_core.c>
                        Require all granted
                </IfModule>
        </Location>
</VirtualHost>
```

[root@po-oracle8 ~]# vi /etc/httpd/conf/httpd.conf

```
ServerRoot "/etc/httpd"
Listen 80
Include conf.modules.d/*.conf

User bacula
Group bacula

ServerAdmin root@localhost

<Directory />
    AllowOverride none
    Require all denied
</Directory>

DocumentRoot "/var/www/html"

<Directory "/var/www">
    AllowOverride None
    # Allow open access:
    Require all granted
</Directory>

<Directory "/var/www/html">
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

<IfModule dir_module>
    DirectoryIndex index.html
</IfModule>

<Files ".ht*">
    Require all denied
```

```
</Files>

ErrorLog "logs/error_log"

LogLevel warn

<IfModule log_config_module>
    LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\""␣
→combined
    LogFormat "%h %l %u %t \"%r\" %>s %b" common

    <IfModule logio_module>
      LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"␣
→%I %O" combinedio
    </IfModule>
    CustomLog "logs/access_log" combined
</IfModule>

<IfModule alias_module>
    ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"
</IfModule>

<Directory "/var/www/cgi-bin">
    AllowOverride None
    Options None
    Require all granted
</Directory>


<IfModule mime_module>
    TypesConfig /etc/mime.types
    AddType application/x-compress .Z
    AddType application/x-gzip .gz .tgz
    AddType text/html .shtml
    AddOutputFilter INCLUDES .shtml
</IfModule>

AddDefaultCharset UTF-8

<IfModule mime_magic_module>
    MIMEMagicFile conf/magic
</IfModule>

EnableSendfile on

# Load config files in the "/etc/httpd/conf.d" directory, if any.
IncludeOptional conf.d/*.conf
Include conf.modules.d/00-systemd.conf
```

BWeb Installation with Apache 2.4 and Ldap with Group Authentication RHEL

This article gives specific insight on how to set up group ldap authentication on RHEL 8 and any deriva-
tives. See for the whole procedure to enable it in the main article.

Configuration Settings

The content presented below was tested successfully on a rhel8 VM enabling authentication for users
in a certain group. In the example below, the only users that should be able to authenticate are in the
"bacadmin" group.

Openldap structure on the tested openldap server:

```
dn: dc=u,dc=l3
objectClass: top
objectClass: dcObject
objectClass: organization
o: u
dc: u

dn: cn=bacula_binding.service,dc=u,dc=l3
objectClass: organizationalRole
cn: admin
cn: bacula_binding.service
description: Ldap admin

dn: ou=People,dc=u,dc=l3
objectClass: organizationalUnit
ou: People

dn: ou=Groups,dc=u,dc=l3
objectClass: organizationalUnit
ou: Groups

dn: cn=bacadmin,ou=Groups,dc=u,dc=l3
objectClass: top
objectClass: posixGroup
gidNumber: 678
cn: bacadmin
memberUid: charles
memberUid: foo

dn: ou=users,dc=u,dc=l3
objectClass: top
objectClass: organizationalUnit
ou: users

dn: uid=charles,ou=users,dc=u,dc=l3
objectClass: top
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount
```

(continues on next page)

```
cn: charles
uid: charles
uidNumber: 16859
gidNumber: 678
homeDirectory: /home/charles
loginShell: /bin/bash
gecos: charles
shadowMax: 0
shadowWarning: 0

dn: uid=foo,ou=users,dc=u,dc=l3
objectClass: top
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount
cn: foo
uid: foo
uidNumber: 16860
gidNumber: 678
homeDirectory: /home/foo
loginShell: /bin/bash
gecos: foo
shadowMax: 0
shadowWarning: 0

dn: uid=bweb,ou=users,dc=u,dc=l3
objectClass: top
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount
cn: bweb
uid: bweb
uidNumber: 16869
gidNumber: 679
homeDirectory: /home/bweb
loginShell: /bin/bash
gecos: bweb
shadowMax: 0
shadowWarning: 0

dn: uid=bwebadmin,ou=users,dc=u,dc=l3
objectClass: top
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount
cn: bwebadmin
uid: bwebadmin
uidNumber: 16879
gidNumber: 679
homeDirectory: /home/bwebadmin
loginShell: /bin/bash
gecos: bwebadmin
```

```
shadowMax: 0
shadowWarning: 0

dn: cn=dev,ou=Groups,dc=u,dc=l3
objectClass: top
objectClass: posixGroup
gidNumber: 679
cn: dev
memberUid: bweb
memberUid: bwebadmin
```

vi /etc/httpd/conf.d/bweb-apache.conf

```
############################### Authentication with groups

            AuthType Basic
            AuthName "Auth"
            AuthBasicProvider ldap
            AuthLDAPURL ldap://$your_ldap_server:389/dc=u,dc=l3?uid?sub?
↪(objectClass=*)
            AuthLDAPBindDN "cn=bacula_binding.service,dc=u,dc=l3"
            AuthLDAPBindPassword "$your_password"
            AuthLDAPGroupAttribute memberUid
            AuthLDAPGroupAttributeIsDN off
            Require ldap-group cn=bacadmin,ou=Groups,dc=u,dc=l3
#######################################
```

Authentication via Pam LDAP

Bweb Authentication system users + Linux System Authentication via Pam LDAP

Its possible to have the following authentication workflow with Bweb: Ldap Authentication + Linux Server Pam + Bweb System Authentication

Linux Server Pam LDAP Authentication

https://wiki.debian.org/LDAP/PAM

https://www.howtoforge.com/linux_ldap_authentication

On Bweb Side Configure Bweb System Users, just check the checkbox of Systems Users Authentication and add bacula to the System Wheel Group.

How to add a wheel group in Linux Server

When a server had to be maintained at a higher level than the day-to-day system administrator, root rights were often required. The 'wheel' group was used to create a pool of user accounts that were allowed to get that level of access to the server. If you weren't in the 'wheel' group, you were denied access to root.

Simple speaking, the wheel group is a group of users allowed to access root user.

Let us take an example of RHEL server. First, make sure there is a wheel group in the /etc/group file.

`wheel:x:10:root`

If the above line does not exist, add it.

You must put users that you want to be admins into the wheel group. To do this, add the user to the end of the wheel group line. This will make the wheel group a secondary group.

Second, change into the /etc/pam.d directory, and edit the file su.

This file controls the access to the program su and modifies its behaviors during the authentication process. The change will modify the access so that only those in the wheel group have access to the program su.

Find these lines in /etc/pam.d/su:

```
# Uncomment the following line to require a user to be in the ?wheel? group.
#auth required pam_wheel.so use_uid
```

And change them (as suggested) to this:

```
# Uncomment the following line to require a user to be in the ?wheel? group.
auth required pam_wheel.so use_uid
```

Then, expand the permissions in sudo to account for those with wheel permissions. Edit the configuration file with visudo and change these lines:

```
# Uncomment to allow people in group wheel to run all commands        # %wheel
ALL=(ALL) ALL
```

To this

```
# Uncomment to allow people in group wheel to run all commands %wheel ALL=(ALL)
ALL
```

This will allow anyone in the wheel group to execute commands using sudo. It would also allow anyone this sort of access on any machine that they have wheel group membership.

3 BCloud Service

- *Executive summary*
- *Features Summary*
- *Architecture*
- *Installation*
- *Packages and Programs*
- *Initial Configuration*
- *Configure BCloud Service*
- *BCloud Service Customization*
- *Network Configuration*
- *VPN Configuration*
- *Upgrade Notes*
- *10.2.3 to 12.2.0*
- *10.2.1 to 10.2.2*

3.1 Executive summary

IT organizations are constantly being challenged to deliver high quality solutions with reduced total cost of ownership.

This document is intended to provide insight into the considerations and processes required to implement BCloud Service with your Bacula Enterprise installation.

Note: The below sections describe how to install and configure BCloud by the backup administrator. The BCloud User Manual is available as a PDF file: PDF

3.2 Features Summary

BCloud Service is a user oriented web frontend (sometimes also called *panel*) that is designed to delegate backup administrator tasks to the end user. BCloud Service offers the following features:

- Based on Bacula Enterprise REST API

- The web interface can be translated into different languages

- Simple user oriented web interface (compatible with Chrome and Firefox)

- System administrator can customize BCloud using php hook scripts

- System administrator can limit the number of Client, Job, Schedule and Pool size per user

- All communications are encrypted by default

- Ordinary users are authenticated via LDAP

- Ordinary users can start, monitor or abort backup/restore jobs

- Ordinary users can register their own clients

- Ordinary users can add and schedule their own backup jobs

- Ordinary users can select files to restore

- Ordinary users can choose the restore client

- Ordinary users can see a dashboard with useful information about previous and currently running jobs

- Ordinary users can download installation programs and documentation

3.3 Architecture
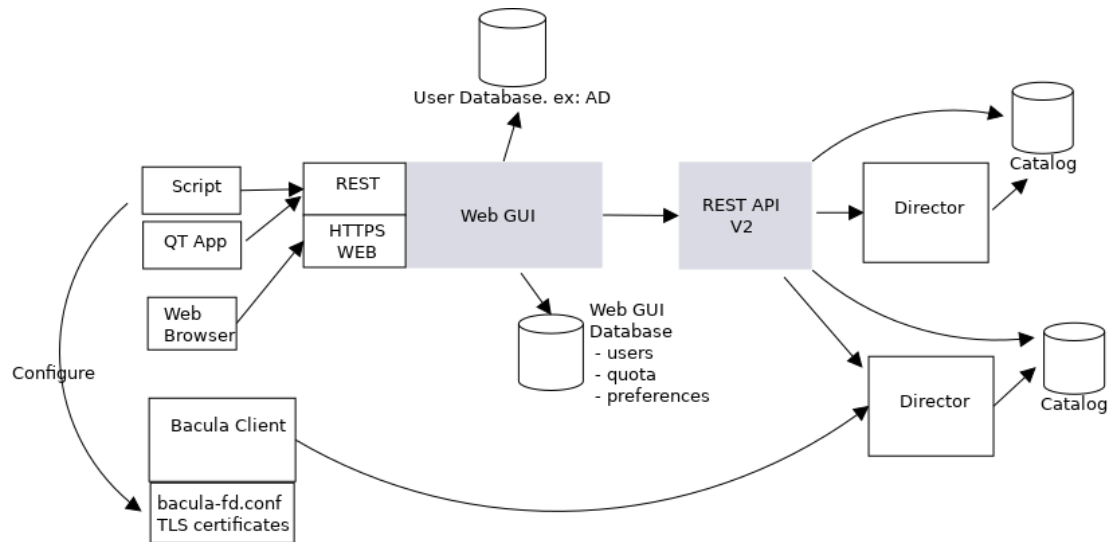
Fig. 62: BCloud Service Architecture
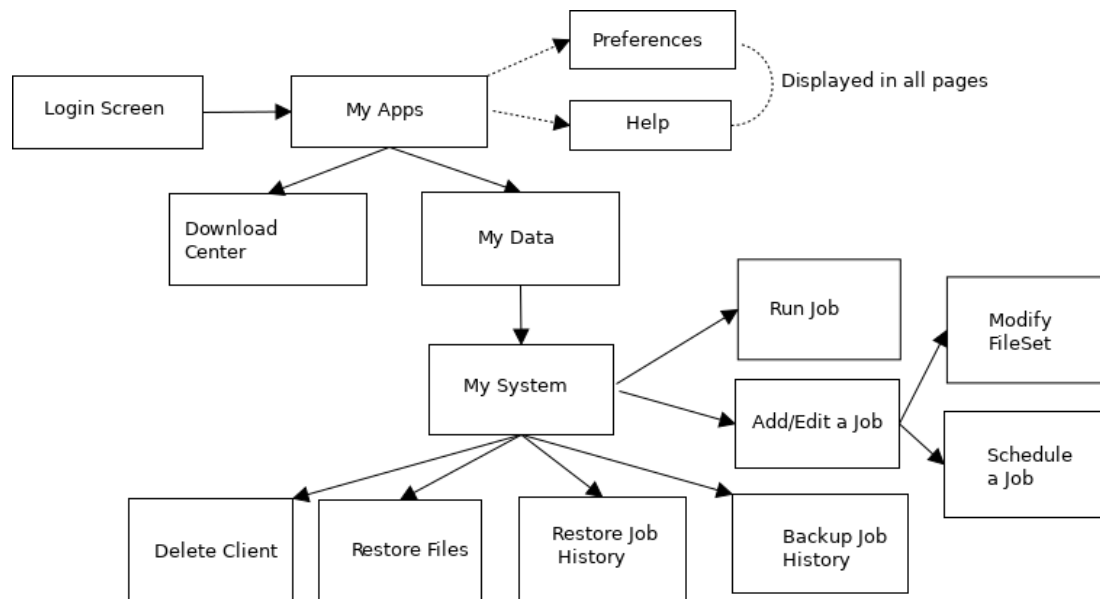
Fig. 63: Screen Organization

3.4 Installation

3.5 Packages and Programs

You can install the Bacula Enterprise BCloud Service by extending the repository file for your package manager to contain a section for the programs. For example, `/etc/yum.repos.d/bacula.repo`:

```
[baculasystems]
name = Red Hat Enterprise  - baculasystems
baseurl = https://www.baculasystems.com/dl/@customer@/rpms/bin/@version@/
→rhel7-64
enabled = 1
protect = 0
gpgcheck = 0

[baculasystems-bweb]
name = Red Hat Enterprise  - baculasystems
baseurl = https://www.baculasystems.com/dl/@customer@/rpms/bweb/@version@/
→rhel7-64
enabled = 1
protect = 0
gpgcheck = 0

[baculasystems-bcloud]
name = Red Hat Enterprise  - baculasystems
baseurl = https://www.baculasystems.com/dl/@customer@/rpms/bcloud/@version@/
→rhel7-64
enabled = 1
protect = 0
gpgcheck = 0
```

```
yum install bacula-enterprise-postgresql
```

```
yum install bacula-enterprise-bcloud-service
```

If you experience dependency issues while installing (like missing lighttpd packages) you could try adding the EPEL repository configuration, if not already present, to your system by issuing the following command:

```
yum install epel-release
```

YUM will install the packages and the various dependencies that are needed such as:

- bacula-enteprise-bweb
- bacula-enteprise-rest-api
- php, php-bcmath, php-json, php-ldap, …

To install php-zip, please refer to Section *install-php-zip*.

If you prefer to manually install the packages, you may also download them from your download area and use one of the low level package manager tools such as `rpm` to do the plugin installation.

3.6 Initial Configuration

The following steps are required to setup PostgreSQL, Bacula Enterprise, Bacula Enterprise REST API, and Bacula Enterprise BCloud Service. The following scenario assumes that the commands and the Internet browser are started from the system that hosts Director, PosgreSQL, BWeb, RESTAPI and BCloud.

1. Stop or configure the Firewall (*network-setup*)

```
service firewalld stop
```

2. Stop or configure selinux

```
setenforce 0
```

3. Install and configure Bacula Enterprise PostgreSQL

```
yum install postgresql-server
```

4. Install and configure PostgreSQL

   - Configure `postgresql.conf`

   - Create (`initdb`) the PostgreSQL cluster and start the service

     ```
     postgresql-setup initdb
     systemctl start postgresql
     ```

   - Execute Bacula catalog scripts located under `/opt/bacula/scripts`:

     ```
     # su - postgres
     postgres $ /opt/bacula/scripts/create_bacula_database
     postgres $ /opt/bacula/scripts/make_bacula_tables
     postgres $ /opt/bacula/scripts/grant_bacula_privileges
     ```

   - Change bacula database ownership and assign a password to the bacula user:

     ```
     # su - postgres
     postgres $ psql
     postgres> ALTER DATABASE bacula OWNER TO bacula;
     postgres> ALTER USER bacula WITH PASSWORD '<SOME_PASSWORD>';
     ```

   - Edit `/var/lib/pgsql/data/pg_hba.conf` to use **md5** instead of **ident** for local users

   - Restart the PostgreSQL service

     ```
     systemctl restart postgresql
     ```

   - Set the `bacula` SQL user password in the `/opt/bacula/etc/bacula-dir.conf`

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

- Try to connect the PostgreSQL service with `psql -U bacula -d bacula -W -h localhost`

5. Install php-zip extension

```
yum install php-pear php-devel zlib-devel libzip-devel gcc-c++ wget
```

```
wget https://pecl.php.net/get/zip-1.12.5.tgz
```

```
pecl install zip-1.12.5.tgz
```

```
echo 'extension=zip.so' > /etc/php.d/zip.ini
```

6. Configure the PHP Timezone.

Edit `/etc/php.ini` to configure the `date.timezone` parameter according to your physical location and the table http://php.net/manual/en/timezones.php.

```
# grep timezone /etc/php.ini
date.timezone = Europe/Paris
```

7. Configure REST API, edit the `/opt/bacula/rest-api/etc/lighttpd.conf` and change the port from 443 to 4443. Optionally, it is possible to change limit the connectivity of the REST API service to the `localhost` interface once everything is setup correctly. (It simplifies the setup if the `bind` parameter is changed at the end of the installation process).

```
server.bind = "localhost"
```

8. Start all services

```
systemctl start bacula-fd
systemctl start bacula-dir
systemctl start bacula-sd
systemctl start bweb
/opt/bacula/rest-api/bin/lighttpd.redhat start
/opt/bacula/bcloud/bin/bcloud start
```

9. Verify that the Director can access the Catalog

```
/opt/bacula/bin/bconsole
* status dir
```

10. Enable services at boot

```
systemctl enable bacula-fd
systemctl enable bacula-dir
systemctl enable bacula-sd
systemctl enable bweb
systemctl enable postgresql
echo /opt/bacula/rest-api/bin/lighttpd.redhat start >> /etc/rc.local
echo /opt/bacula/bcloud/bin/bcloud start >> /etc/rc.local
chmod a+x /etc/rc.d/rc.local
```

11. Initialize BWeb Mangement Suite SQL tables

```
/opt/bweb/bin/install_bweb.sh
```

12. Configure BWeb http://localhost:9180, click **Configure** and **Bacula Configuration** to initialize the BConfig tool

13. In BWeb, click on the Security Center and setup master CA certificate and generate certificates for each of daemons. Once done, commit the workset and restart all daemons.

14. Configure REST API via https://localhost:4443/panel/

15. Add a REST API User via the REST API Panel

16. Keep the User ID and User Secret for Bcloud configuration

17. Set the Redirect URI to https://localhost/?page=Callback&directorid=1

18. Install and configure LDAP server or use the file driver auth backend (See *file-auth*).

19. Configure the BCloud Service (See section *configure-bcloud*).

3.7 Configure BCloud Service

You must edit the BCloud Service configuration file:

```
cat  /opt/bacula/bcloud/www/protected/conf/settings.conf
[db]
type = "pgsql"
dbname = "bacula"
address = "127.0.0.1"
user = "bacula"
password = "<SOME_PASSWORD>"

[api]
auto_commit = "1"

[app]
log_file = "/opt/bacula/bcloud/log/web_app.log"
log_level = "1"
auth_module = "ldap"
ldap_server = "ldap://localhost:389/"
ldap_base_dn = "dc=ok"
ldap_additional_user_dn = "ou=People"

[frs]
download_dir = "/opt/bacula/bcloud/downloads"

[notification]
admin_email = root@localhost
mail_headers = "From: BCloud Service <root@localhost>\r\n"
```

165

Database Connection

db section of the configuration file. The database can be specific to BCloud or be shared with the Bacula catalog.

- **type** The database type is `pgsql`

- **address** Address of the database

- **dbname** Name of the database

- **user** Database user name

- **password** Password for the database

Application Settings

app section of the configuration file.

- **log_level** is the bcloud log level. Can be 1 to 10

- **log_file** is the full path of the bcloud log file on disk

---

New in Bacula Enterprise 12.8.1

- **auth_module** specifies the authentication module. Possible values are: "ldap", "ldap2", or "file"

- **ldap_server** a full LDAP URI of the form ldap://hostname:port

- **ldap_base_dn** a base LDAP distinguished name, ex. dc=lan. Please note, that at least one part of the domain must be used in the username field. For example for a domain defined as: dc=mydomain,dc=lan, the 'ldap_base_dn' should define 'dc=lan' and the username should be 'my-domainusername'.

- **ldap_additional_user_dn** additional LDAP user distinguished name. ex. ou=People

---

New in Bacula Enterprise 12.8.1

- **ldap2_server** a full LDAP URI in the form ldap://hostname:port.

- **ldap2_auth_method** supported are two methods: `anonymous` and `simple`. The `anonymous` method does not require providing LDAP administrator parameters (`ldap2_bind_dn` and `ldap2_bind_pwd`) to search for LDAP entries. The LDAP server must support the anonymous method. The `simple` method uses credentials defined in the `ldap2_bind_dn` and `ldap2_bind_pwd` options.

- **ldap2_bind_dn** defines the administrator account distinguished name. This option is mandatory for the `simple` authentication method and it is not used for the `anonymous` method.

- **ldap2_bind_pwd** stores the administrator account password. This option is mandatory for the `simple` authentication method and it is not used for the `anonymous` method.

- **ldap2_base_dn** determines the starting point the LDAP server uses when searching for users to authenticate within the directory.

- **ldap2_filter** is the filter when searching for users. It uses the `%user` keyword that is replaced by the username that was entered during log in. The filter can be simple such as: `(uid=%user)`, or more advanced such as: `(&(uid=%user)(|(ou:dn:=People)(ou:dn:=OtherPeople)))`. More information about filters can be found in RFC 4515 (http://www.faqs.org/rfcs/rfc4515.html)

- **user_registration** can hide/show the Client registration button on MyData page. Can be 0 or 1.

- **generate_console** determines what kind of console configuration is generated for the registration. If the value is set to 0 (default value), then a Console resource configuration will be generated with full access. If the value is set to 1, a check is performed to see if this is the first registered File Daemon. If yes, BCloud will create a new console in the Director configuration automatically and BCloud will prepare the appropriate Console resource with restricted access for the Jobs, Filesets, Client, and Pools that correspond to the BCloud user and the Client values. If a Console resource already exists for the user, the existing Console configuration will be used for the restricted Console settings without creating new one. If set to 2, the BCloud Service will check the Director configuration to see if there is a Console resource related to the currently logged in user, and if one exists, the prepared console configuration will be generated with the console access.

Download Center Settings

`frs` section of the configuration file.

- **download_dir** is the full path to the directory of the download area

Notification Settings

`notification` section of the configuration file.

- **admin_email** is the email of the bcloud administrator
- **mail_headers** is the headers of the email that is sent

REST API Settings

`api` section of the configuration file.

- **auto_commit** will control if BCloud can automatically commit changes made from the user interface. If **auto_commit** is set to 0, the BWeb administrator will recieve email notifications and will have to commit changes in the BWeb configuration interface. If the **auto_commit** is set to 1, end user modifications will be automatically applied to the Director.

BCloud Service SQL Schema

The BCloud Service SQL schema must be loaded into the database described in the `settings.conf` file.

```
psql -U bacula -d bacula -h localhost -W \
  -f /opt/bacula/bcloud/www/protected/conf/bcloud_service-pgsql.sql
```

BCloud Service REST API Connection

A Director must be registered within the BCloud Service. The command `bcloud-add-director.php` will populate the `bcloud_director` table in the catalog. The name of the Director must exactly match the name of the Director defined in `bacula-dir.conf`.

```
php   /opt/bacula/bcloud/www/protected/tools/bcloud-add-director.php
  /opt/bacula/bcloud/www/protected/tools/bcloud-add-director.php:
                --name
                --api_server_uri
                --authorization_uri
                --access_token_uri
                --refresh_token_uri
                --client_id
                --client_secret
                --redirect_uri
                --scope
                --restricted
                --admin_email
                --confirm_registration
                --confirm_new_jobs
                --confirm_deletion
                --template_dir

php   /opt/bacula/bcloud/www/protected/tools/bcloud-add-director.php \
 --name=centos7-64-dir                                        \
 --api_server_uri=https://localhost:4443/                     \
 --authorization_uri=https://localhost:4443/v2/authorize/     \
 --access_token_uri=https://localhost:4443/v2/request_token/  \
 --refresh_token_uri=https://localhost:4443/v2/refresh_token/ \
 --client_id=XXXXXXXXXXXXXXXX                                 \
 --client_secret=XXXXXXXXXXXXXXXXXXXX                         \
 --redirect_uri="https://localhost/?page=Callback&directorid=1" \
 --scope="/*"                  \
 --restricted=0                \
 --admin_email=root@localhost \
 --confirm_registration=0      \
 --confirm_new_jobs=0          \
 --confirm_deletion=0          \
 --template_dir=/opt/bacula/bcloud/www/protected/conf

          ====     Director added successfully.    ====
```

File Based Authentication

For test purposes, it is possible to use file based authentication instead of LDAP authentication. To use file based authentication, the following file must be created:

```
/opt/bacula/bcloud/www/protected/conf/auth
```

Inside this file there must be one row per user with the following format:

```
userid1,username1,password_hash1
userid2,username2,password_hash2
userid3,username3,password_hash3
```

Example:

```
1,john,a51dda7c7ff50b61eaea0444371f4a6a9301e501
2,paul,a027184a55211cd23e3f3094f1fdc728df5e0500
3,admin,d033e22ae348aeb5660fc2140aec35850c4da997
```

The password hashes must be generated with the following command:

```
echo -n 'yourpasss' | sha1sum
```

In the `settings.conf` configuration file, the `auth_module` must be set to `file`.

```
[app]
auth_module = "file"
```

BCloud Service Administrator

It is possible to promote a particular user as BCloud Service Administrator.

```
# psql -U bacula -d bacula -h localhost -W
bacula> SELECT name,userid from bcloud_user;
bacula> UPDATE bcloud_roles SET roles = 'user,admin' WHERE userid=3;
```

If the user is not listed in the `bcloud_users` table, try to first login to the BCloud Service with the user and then try the SQL queries again.

The BCloud Service Administrator will have access to advanced options to edit user properties like:

- Max Clients (default *1*, SQL field `max_clients`)
- Max Jobs (default *3*, SQL field `max_jobs`)

It is possible to edit Bcloud Service users properties via a SQL query, and a set of user definable hooks will be available in a next version.

For the moment, it is possible to change default user attributes by modifying the file:

```
/opt/bacula/bcloud/www/protected/class/User.php
```

The user quota cannot be set via a SQL. The initial value can be set via the Pool.tpl template file described Sec **PoolTPL**, or via the admin page.

3.8 BCloud Service Customization

To change the company logo, replace the current logo file in the location below with a custom logo.

```
/opt/bacula/bcloud/www/images/logo.png
```

The logo, buttons and link style settings are can be customized in the following file:

```
/opt/bacula/bcloud/www/css/custom.css
```

Additional CSS with detailed interface styles can be found in the following files:

```
/opt/bacula/bcloud/www/css/style.css
/opt/bacula/bcloud/www/css/restore.css
```

It is not recommended to modify these files directly.

To customize the BCloud Service CSS files, the firefox plugin "Live editor for CSS, Less / Sass - Magic CSS" can be used. It is available on https://addons.mozilla.org/en-US/firefox/addon/live-editor-for-css-and-less/

Once the Firefox plugin is installed, it is possible via the HTML inspector to select the BCloud element and to change and define attributes in a special window CSS styles.

At the end of session with the BCloud Service customization, you can copy and paste the CSS result into the `custom.css` file located in the BCloud Service web directory.
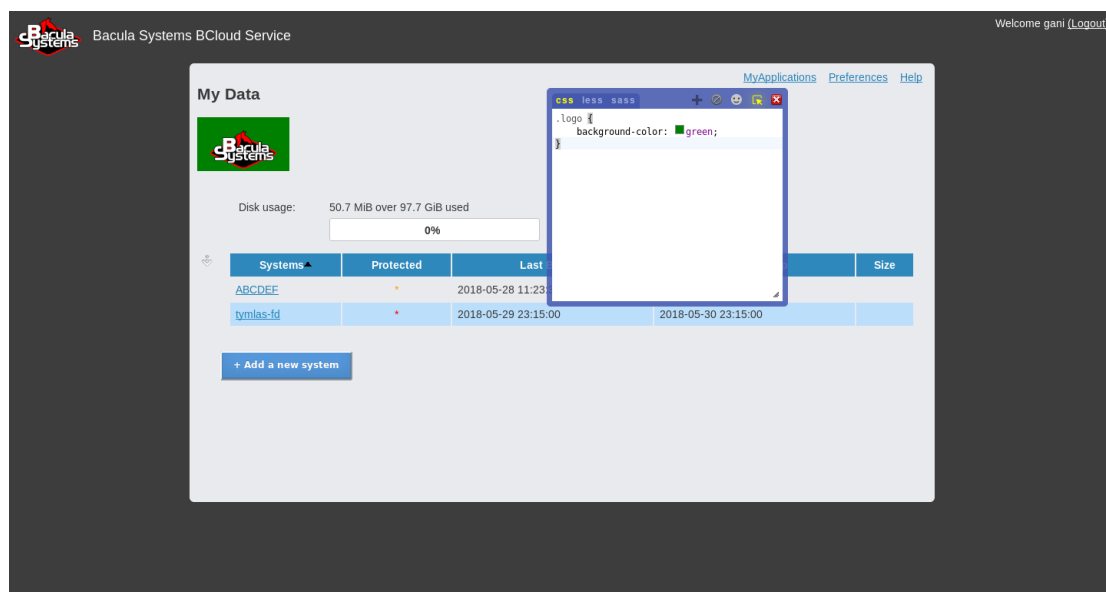
Fig. 64: Screen Customization

Copyright © 2025 Bacula Systems. All trademarks are the property of their respective owners.

3.9 Network Configuration

The following TCP network ports should be accessible to ordinary users:

- `https/443` BCloud Service interface
- `9103` Bacula Enterprise Storage Daemon

The following TCP network ports can be optionally accessible to ordinary users:

- `9101` Bacula Enterprise Director Daemon

The following TCP network ports should be accessible to system administrators

- `9102` Bacula Enterprise File Daemon
- `4443` Bacula Enterprise REST API
- `9180` Bacula Enterprise BWeb Management Suite

The following UDP network port is optional and should be accessible to ordinary users if the VPN is used:

- `1194 udp` OpenVPN

To properly confure the firewall on RHEL 7, please check the following documentation: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/security_guide/sec-configuring_firewalld

3.10 VPN Configuration

OpenVPN Server Confguration

- Install the `openvpn` package

```
yum install openvpn
```

```
yum install ca-certificates
```

- Indentify the Root CA certificate and root key from the Security Center in the BWeb.

```
/opt/bacula/etc/conf.d/ssl/keys/root_key.pem
/opt/bacula/etc/conf.d/ssl/certs/root_cert.pem
```

- Prepare server certificate

  1. Set the current directory to openvpn

     ```
     cd /etc/openvpn/server
     ```

  2. Generate key:

     ```
     openssl genrsa -out ukey.pem 4096
     ```

  3. Generate certificate request:

```
openssl req -new -batch \
  -subj "/C=<COUNTRY>/ST=<STATE>/O=<COMPANY>/CN=<ADDRESS>/
↪emailAddress=<EMAIL>" \
  -key ukey.pem -out ucert.csr

COUNTRY - two letter country name such as 'EN'
STATE - region in COUNTRY
COMPANY - company name
ADDRESS - hostname or address for the certificate (Common Name)
EMAIL - e-mail address
```

4. Generate certificate using certificate request:

```
# cat > extra.cnf << EOF
[KUE]
extendedKeyUsage = serverAuth
keyUsage = digitalSignature,keyEncipherment
EOF

# openssl ca -keyfile /opt/bacula/etc/conf.d/ssl/keys/root_key.pem \
  -batch -policy policy_anything \
  -enddate YYYYMMDDHHIISSZ \
  -config <(cat /etc/pki/tls/openssl.cnf extra.cnf) \
  -extensions KUE -out ucert.pem \
  -cert /opt/bacula/etc/conf.d/ssl/certs/root_cert.pem \
  -infiles ucert.csr

YYYYMMDDHHIISSZ - date and time after which certificate expires (ex.␣
↪20180813162412Z)
```

For this step, it is possible that some files will be needed, such as:

```
touch /etc/pki/CA/index.txt /etc/pki/CA/index.txt.attr
echo "01" > /etc/pki/CA/serial
```

- Use `root_cert.pem`, `ucert.pem` and `ukey.pem` in the OpenVPN server configuration as below:

```
# cat /etc/openvpn/server/server.conf
# OpenVPN Server Config
port 1194
proto udp
dev tun
sndbuf 0
rcvbuf 0
dh dh.pem
tls-auth ta.key 0
#crl-verify crl.pem
auth SHA512
topology subnet
server 10.8.0.0 255.255.0.0
ifconfig-pool-persist ipp.txt
keepalive 10 120
cipher AES-256-CBC
```

(continues on next page)

```
comp-lzo
user nobody
group nobody
persist-key
persist-tun
client-config-dir /etc/openvpn/ccd
status /var/log/openvpn-status.log
log /var/log/openvpn.log
log-append /var/log/openvpn.log
verb 3
# custom files
ca /opt/bacula/etc/conf.d/ssl/certs/root_cert.pem
cert ucert.pem
key ukey.pem
```

The OpenVPN configuration might require some files such as `ta.key` or `dh.pem`.

```
openvpn --genkey --secret /etc/openvpn/server/ta.key
openssl dhparam -out /etc/openvpn/server/dh.pem 1024
```

To register clients, the BCloud service process executed under the `bcloud` account must be able to excute a script via sudo.

```
# cat /etc/sudoers.d/bcloud
Defaults:lighttpd !requiretty
lighttpd ALL= NOPASSWD: /opt/bacula/bcloud/www/protected/conf/vpn-post-config-
↪script.sh
```

On RHEL, the service can be started with `service openvpn-server@server start` where `server` is the name of the configuration file.

The default subnet `10.8.0.0/16` can be configured in the script

```
# vi /opt/bacula/bcloud/www/protected/conf/vpn-post-config-script.sh
...
# VPN server address
readonly SERVER_ADDRESS='10.8.0.1'

# OpenVPN server start/stop base command
readonly OPENVPN_START_STOP_CMD='service openvpn-server@server'
...
```

To start the openvpn service, the following command can be used:

```
systemctl enable openvpn-server@server
```

```
systemctl start openvpn-server@server
```

If the server configuration is `server2.conf`, then the command should be adapted to `openvpn-server@server2`.

With the default vpn configuration script, only one bacula client can be configured for each external IP address. Some more advanced OpenVPN configuration can be done when configuring the `vpn-post-config-script.sh` script.

In order for the FD (located behind NAT) to be able to reach the SD on a different subnet than the OPenVPN server itself, it is needed to insert a `push route` rule in the OpenVPN server configuration file in order to instruct the client to include the target network in the routing table as reachable through the VPN. Example

```
push "route 10.254.45.0 255.255.255.0"
```

where 10.254.45.0 is the remote network address and the 255.255.255.0 is the remote network netmask. This modification should be done in the `vpn-post-config-script.sh` script.

OpenVPN Client Template

To prepare the BCloud OpenVPN client template, it is needed to update the following file:

```
/opt/bacula/bcloud/www/protected/conf/openvpn-client.template
```

and add between <tls-auth></tls-auth> a static key shared with the VPN server. It is file that can be generated with the command described here *generate-ta*.

```
cat /etc/openvpn/server/ta.key
```

The `HOST_ADDRESS` keyword must be replaced with the external host address in the "remote" option as showed below:

```
remote HOST_ADDRESS 1194
```

for example:

```
remote some.good.company.us 1194
```

OpenVPN Client Configuration

The Registration Wizard can request an OpenVPN configuration file from the server and use it locally to configure OpenVPN on the client side. OpenVPN must be installed on the computer and discoverable in the PATH.

To use this feature, simply check the Configure VPN box before pressing Register.

To allow customization, a script is executed to perform the actual VPN configuration. Under Unix-like platform `open_vpn.sh` is called while on Windows its `ovpn_config.bat`. Two parameters are passed to the script:

1. location of the .ovpn config file (can be copied or moved by the script)

2. OpenVPN executable file location, detected by the Registration Wizard ("None" is no executable could be found).

`ovpn_config.sh` exemple:

```
#!/bin/sh
ovpn_config_file_path="$1"
ovpn_executable_path="$2"

if [ "$2" == "None" ]
then
```
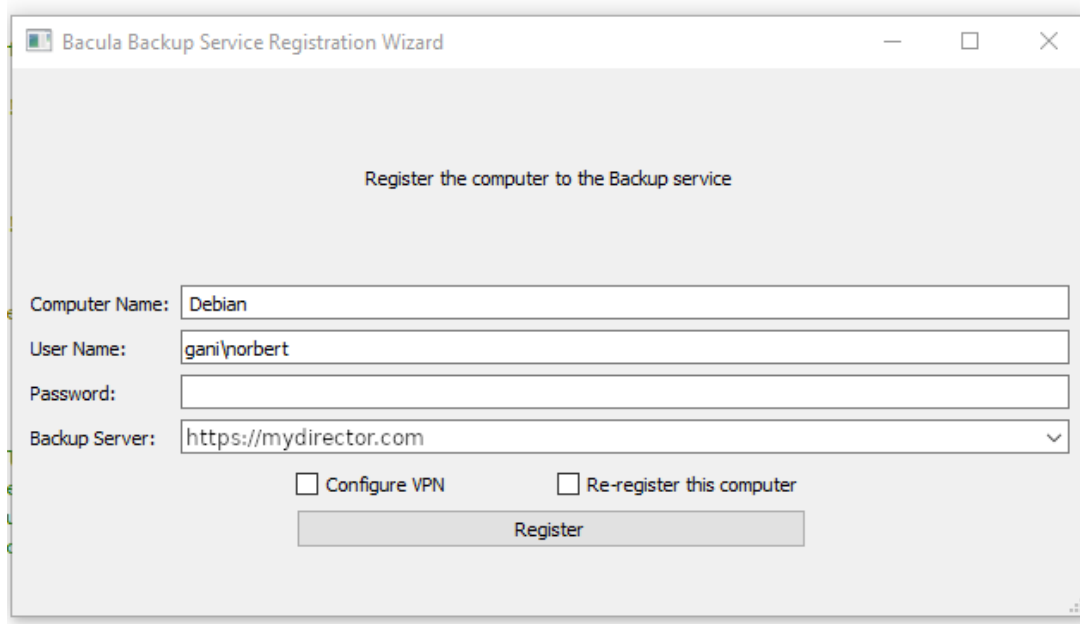
Fig. 65: BCloud Registration Wizard

```
    /usr/sbin/openvpn --config ${ovpn_config_file_path}
else
    ${ovpn_executable_path} --config ${ovpn_config_file_path}
fi
```

The script can be customized as needed.

3.11 Upgrade Notes

3.12 10.2.3 to 12.2.0

The major release 12.2 of BCloud Service uses a new catalog format. We provide a SQL script that converts 12.0.x and earlier catalog versions (2) to the new 12.2 format (3). Please apply it to the database using by BCloud Service.

```
/opt/bacula/bcloud/www/protected/conf/upgrade-10.0_12.2_postgresql.sql
```

After applying the SQL script, please go to Admin => Directors page on the BCloud Service interface and please fill for each defined Director Address and Port fields.

3.13 10.2.1 to 10.2.2

The BCloud version 10.2.2 can now handle multiple clients from the same external IP address. If you have a customized version of the `vpn-post-config-script.sh` script, please revert to the official version of the script and re-apply your modifications.

3.14 10.2.1 and 10.0.8

The BCloud version 10.2.1 and 10.0.8 are now using the default OpenVPN client directory (`/etc/openvpn/ccd`) instead of `/etc/openvpn/client`.

It is recommended to rename the client directory (or move the existing configuration files), adapt your custom `vpn-post-config-script.sh` script (if any), adapt your OpenVPN server configuration file (`client-config-dir /etc/openvpn/ccd`) and restart the OpenVPN server.

The directory must be accessible to the OpenVPN user of your system.

3.15 10.0.7

The BCloud version 10.0.7 includes a minor modification of the BCloud catalog. To apply the upgrade, do the following command:

```
$ psql -U bacula bacula < /opt/bacula/bcloud/www/protected/conf/upgrade-8.10_
→10.0_postgresql.sql
```

3.16 Bacula Enterprise Configuration

3.17 Resource Templates

BCloud Service will automatically generate the necessary Bacula configuration files at the ordinary user request. The following resources are automatically generated when a new system is registered by an ordinary user:

- Client

- Pool (generated only during the first client registration. The pool is common to all clients)

The following resources are automatically generated when the ordinary user creates new jobs:

- Job

- Schedule

- Fileset

When a new resource is created, BCloud Service will use templates to initialize resources with custom directives.

```
# ls /opt/bacula/bcloud/www/protected/conf/*.tpl
Pool.tpl
Client.tpl
Job.tpl
Fileset.tpl
Schedule.tpl
Console.tpl
```

These template files are JSON files with resource names and default values.

```
# cat /opt/bacula/bcloud/www/protected/conf/Pool.tpl
{
    "PoolType": "Backup",
    "RecyclePool": "Scratch",
    "LabelFormat": "Vol",
    "MaximumVolumeBytes": "50GB",
    "VolumeRetention": "3months",
    "AutoPrune": 0,
    "Recycle": 1
}
```

We recommend validating the JSON file after any change with the json_pp command from the `perl-JSON-PP` package.

```
# cat protected/conf/Client.tpl | json_pp
```

Please note, starting from version 12.2 the Fileset.tpl template can have complex structure as show two examples below:

```
{
    "Include": [{
        "Options": [{
                "Dedup": "BothSides"
        }]
    }]
}
```

```
{
    "Include": [{
        "Options": [{
                "Dedup": "BothSides",
                "ReadFifo": 1,
                "Signature": "Sha1"
        },{
                "Compression": "Gzip8"
        }]
    },{
        "Options": [{
                "Sparse": 1,
                "Signature": "Md5"
        },{
                "Compression": "Gzip2"
        }],
        "File": ["/etc/one", "/etc/two"]
    }],
    "Exclude": {
        "File": ["/exclude12", "/exclude34"]
    }
}
```

Please also note, that in case Fileset.tpl template the first Include section is merged with that Include section created on the BCloud Service interface. Each next Include section is just added to configuration

as new Include block.

Additional directives defined in Fileset.tpl template are not visible on the BCloud Service interface except include paths.

3.18 Recommended Configuration

Bacula Systems advises the use of the following configuration:

3.19 Autopruning

The Auto Pruning process is suitable for small to medium sites. Directors with more than 50 jobs per day should disable Autopruning and schedule a pruning task with an Admin Job once or twice a day.

```
Job {
  Name = "ADM_TASKS"
  Type = Admin
  Schedule = DailyAt1800
  ...
  ConsoleRunBeforeJob = "prune expired volume yes"
  ConsoleRunBeforeJob = "prune allfrompool allpools volume yes"
  ConsoleRunBeforeJob = ".bvfs_update"
}
```

3.20 Quota and Retention Mangement

The ordinary user storage usage can be limited with a quota. This quota can be implemented with the `MaximumPoolBytes` Pool directive. When the size of the Pool reaches the given limit, jobs will stop and request that a new volume be created, or that old volumes be manually pruned. By default, Purged or Recycled Volumes will be counted in the total Pool size. To avoid that situation, the "RecyclePool" directive must be set.

It is important to configure the original quota to a size that will enable users to rely on the backup system. Various recycling techniques can be configured for Bacula. The Storage directive in the Pool resource is mandatory to use the Quota feature.

[PoolTPL]

```
# cat /opt/bacula/bcloud/www/protected/conf/Pool.tpl
{
  "VolumeUseDuration": "23h",
  "VolumeRetention": "1month",
  "MaximumVolumeBytes": "50GB",
  "AutoPrune": "no",
  "Storage": "File1",
  "LabelFormat": "Vol",
  "RecyclePool": "Scratch",
  "PoolType": "Backup",
  "AutoPrune": 0,
  "Recycle": 1,
  "MaximumPoolBytes": "50GB",
  "MaximumVolumeJobs": "1"
```

```
}

# cat /opt/bacula/bcloud/www/protected/conf/Job.tpl
{
  "Type": "Backup",
  "Storage": "File1",
  "JobDefs": "JobTemplate",
  "MaximumWaitTime": 60
}

# cat /opt/bacula/bcloud/www/protected/conf/Client.tpl
{
  "JobRetention": "5years",
  "FileRetention": "5years",
  "AutoPrune": "no",
  "Catalog": "MyCatalog",
  "TlsCaCertificateFile": "/opt/bacula/etc/conf.d/ssl/certs/root_cert.pem",
  "TlsCertificate": "/opt/bacula/etc/conf.d/ssl/certs/centos7-64-dir_cert.pem
↪",
  "TlsEnable": "yes",
  "TlsKey": "/opt/bacula/etc/conf.d/ssl/keys/centos7-64-dir_key.pem",
  "TlsRequire": "yes"
}

# cat /opt/bacula/etc/conf.d/Director/centos7-64/JobDefs/JobTemplate.cfg
JobDefs {
  Name = "JobTemplate.cfg"
  Type = Backup
  Level = Incremental
  Storage = File1
  Messages = Standard
  Priority = 10
  Accurate = yes
  Write Bootstrap = "/opt/bacula/working/%c.bsr"
}
```

Purge Oldest Volume

It is possible to configure Bacula to automatically purge the oldest Volumes in the Pool. The Job definition must use the directive `Maximum Full Interval` to regularly schedule new Full backups and avoid a situation where the oldest Volume contains the latest Full backup.

Client Job Retention

Bacula can be configured to prune Jobs and recycle Volumes automatically. As with the "Purge Oldest Volume" technique, the Job definition must use the directive `Maximum Full Interval` to regulary schedule new Full backups and avoid pruning the last Full Job.

Progressive Virtual Full, Incremental Forever

With this technique, the last Full job will be merged with subsequent Incrementals into a new Full job. At the end of the Virtual Full Job, old Volumes can be recycled. The downside of this technique is that the Storage Daemon has to read/write the old data into new Volumes frequently. Bacula Enterprise Editiion's Global Endpoint Deduplication (GED) technology can void this downside, since only the references are copied from one volume to an other when GED is used.

During the Virtual Full Job, the quota of the user might exceed the configured value. To avoid this problem, the copy can be done into a temporary pool, and at the end of the Job, old Volumes will be purged and new Volumes can be moved to the user's Pool. This technique requires the use of a small script to update the Pool at the end of the VirtualFull Job.

For example:

```perl
#!/usr/bin/perl -w
#  Copyright (C) 2000-2018 Bacula Systems SA
#  All rights reserved.
use strict;

use IPC::Open2;
my $jobid = shift or die "Usage: $0 jobid level client";
my $level = shift or die "Usage: $0 jobid level client";
if ($level ne 'Virtual') {
    exit 0;
}
shift;                          # Eat the "Full" keyword

my $client = join(" ", @ARGV) or die "Usage $0 jobid level client";
my $pool;

# Adapt to the client format here, C_<ID>_<NAME>
if ($client =~ /^C_(\d+)_/) {
    $pool = "P_$1";            # The Pool format is P_<ID>
} else {
    print "Client format not recognized\n";
    exit 0;                    # Incorrect format for the pool
}

open2(\*OUT, \*IN, "/opt/bacula/bin/bconsole") or die "Unable to open␣
↪bconsole $@";
print IN "update job jobid=$jobid pool=\"$pool\"\n";
print IN "quit\n";
close(IN);
while (<OUT>) {
    print;
}
close(OUT);
```

The Jobs that are migrated by the VirtualFull Job may be automatically deleted and the number of valid backups to keep can be configured in the Job template or in the JobDefs resource.

```
# cat Job.tpl
{
  "DeleteConsolidatedJobs": "yes",
  "BackupsToKeep": "7",
  "RunsAfterJob": "/opt/bacula/scripts/vbackup.pl %i %l %c"
}
```

The VirtualFull function can be enabled with the NextPool directive in all new pools.

```
# cat Pool.tpl
{
  "NextPool": "CommonPool"
}
```

The CommonPool resource should look like:

```
Pool {
  Name = CommonPool
  PoolType = Backup
  LabelFormat = Vol
  Storage = File1
  MaximumVolumeJobs = 1
  MaximumVolumeBytes = 50G
  AutoPrune = no
  Recycle = yes
}
```

The Storage resource should be a Virtual Changer configured with multiple devices.

The VirtualFull Job can be scheduled automatically once a month:

```
# cat Schedule.tpl
{
  "Run": [
    [
      "WeekOfMonth": [1, 2, 3, 4],
      "DayOfWeek": [1],
      "Hour": [10],
      "Minute": 15,
      "Level": "VirtualFull"
    ]
  ]
}
```

3.21 Limitation

- SELinux is not supported and should be disabled.

- Clients must have a direct connection to the Director. Clients behind NAT are not directly supported. An optional VPN tunnel can be configured automatically during the registration process.

- The underscore character _ is not allowed in Client resource name. It will be replaced by - automatically.

- The use of the Registration Wizard cannot be used with the option TLS Verify Peer. The directive TLS Verify Peer should be set to no.