

# Instalacja i konfiguracja narzędzia Bacula Bweb

4 czerwiec 2011 autor: **gani**

Bweb to jedna z lepszych implementacji interfejsu graficznego do zarządzania serwisami Bacula. Zaopatrzony jest w szereg funkcjonalności używanych w codziennej pracy z Bacula (backup, restore, etykietowanie woluminów, analizy, statystyki i wiele innych). Artykuł przedstawia opis kompletnej instalacji Bweb wraz z jego konfiguracją.

## WSTĘP

Bweb jest jednym z graficznych interfejsów administracyjnych Bacula. Napisany jest w oparciu o język **Perl** i udostępniony jako skrypty CGI. Z tego powodu konieczna jest instalacja **Perl** oraz wybranych jego modułów na komputerze, na którym ma działać **Bweb**. Komunikuje się on z daemon'ami Bacula wykorzystując do tego celu tekstową konsolę **bconsole** oraz bezpośrednio zapytania do bazy danych Bacula. Z tego powodu dobrym rozwiązaniem jest instalacja konsoli **bconsole** na komputerze z Bweb.

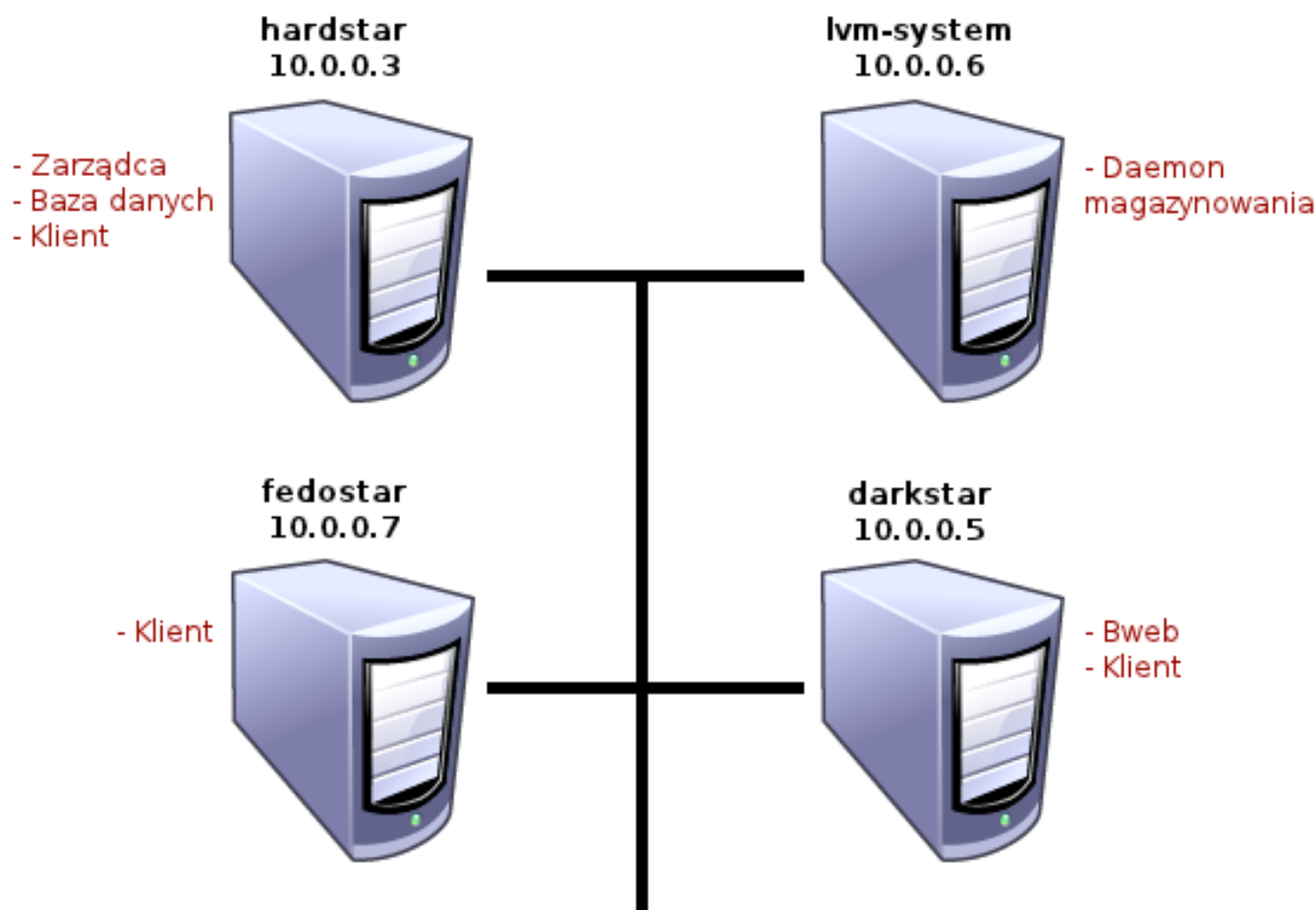
Jak można sugerować się nazwą **Bweb**, jego interfejs dostępny jest poprzez przeglądarkę internetową. Konieczne jest więc również posiadanie **serwera WWW** z obsługą mechanizmu **CGI** do udostępnienia **Bweb**. W niniejszym artykule wykorzystanym serwerem WWW jest serwer **Apache**.

## WYKORZYSTANE ŚRODOWISKO

Poniżej znajdują się informacje o użytym - na potrzeby napisania artykułu - środowisku instalacyjnym (komputer **darkstar**):

- System operacyjny: GNU/Linux
- Dystrybucja GNU/Linux: Debian 6.0.1 stable (Squeeze)
- Serwer WWW: Apache 2.2.16
- Bacula: 5.0.3
- Serwer baz danych: PostgreSQL 8.4.4
- Perl: 5.10.1

## TOPOLOGIA



**Ilustracja 1:** Topologia środowiska instalacyjnego dla Bweb wraz z wyszczególnieniem serwisów dla każdego wykorzystanego komputera.

## PRZYGOTOWANIE ŚRODOWISKA INSTALACYJNEGO

Jak wspomniałem powyżej **Bweb** wykorzystuje w swym działaniu konsolę **bconsole**. Z tego też powodu skompiluję konsolę Bacula na komputerze **darkstar** (patrz Ilustracja 1). W tym celu zaopatrzyłem się w archiwum ze źródłami Bacula. Następnie kompiluję źródła w celu uzyskania programu konsoli bconsole:

```
# tar zxvf bacula-5.0.3.tar.gz
# cd bacula-5.0.3
# ./configure --prefix=/usr/local/bacula --disable-build-dird --disable-
build-stored --enable-client-only --with-scriptdir=/usr/local/bacula/
scripts
# make
# make install
```

Następnym krokiem jest konfiguracja **bconsole**. Edytuję plik konfiguracyjny:

```
/usr/local/bacula/etc/bconsole.conf
```

i ustawiam dyrektywy połączenia z serwisem zarządcy Bacula. U mnie plik konfiguracyjny tekstowej konsoli wygląda następująco:

```
Director {
  Name = hardstar-dir
  DIR Port = 9101
  Address = 10.0.0.3
  Password = "WTWXq2a+ZfEK5IyhFqddagXqdWtbMAMmfzLIGw7FQWoC"
}
```

Uruchamiam konsolę aby sprawdzić czy działa:

```
# /usr/local/bacula/sbin/bconsole -n -c /usr/local/bacula/etc/bconsole.conf
Connecting to Director 10.0.0.3:9101
1000 OK: hardstar-dir Version: 5.0.3 (04 August 2010)
Enter a period to cancel a command.
*
```

## WYMAGANIA DLA BWEB

**Bweb** w swej pracy wykorzystuje poniżej przedstawione moduły **Perl**.

- DBI (wraz z DBD::Pg dla PostgreSQL lub DBD:mysql dla MySQL)
- GD
- GD:Graph
- HTML::Template
- CGI
- Expect
- Time::ParseDate
- Date::Calc

Wszystkie te moduły możliwe są do zainstalowania poprzez powłokę modułu **CPAN**. Alternatywnym rozwiązaniem jest instalacja modułów **Perl** poprzez dystrybucyjny manager pakietów binarnych. W mojej instalacji wszystkie moduły **Perl** zainstalowałem poprzez powłokę modułu **CPAN**. Przed tym jednak potrzebowałem doinstalować pakiet binarny **libgd2-xpm-dev** czyli pakiet z plikami nagłówkowymi niezbędnymi do zbudowania modułu **Perl GD**. Moja sekwencja komend wyglądała jak poniżej.

Instaluję pliki nagłówkowe dla modułu **GD**. Jako, że na maszyna, gdzie ma być zainstalowany **Bweb** pracuje pod kontrolą Debiana, używam do tego celu managera pakietów **apt-get** w następujący sposób:

```
apt-get install libgd2-xpm-dev
```

## UWAGA!

Powyższa komenda pociągnie za sobą kilkanaście dodatkowych pakietów, głównie tzw. "deweloperskich" czyli przydatnych do kompilacji **GD** (mniej lub bardziej ;-)).

## UWAGA!

Przy użyciu instalacji poprzez konsolę modułu **CPAN** do kompilacji modułu **DBD::Pg** potrzebny jest na komputerze z **Bweb** (komputer **darkstar**) zainstalowany serwer baz

danych w takiej samej wersji w jakiej jest serwer baz danych używany przez Bacula. Nie będzie on uruchamiany, lecz wymagany jest do budowy samego modułu **DBD::Pg**.

Poniżej znajduje się komenda do instalacji modułów Perl'a. Użyłem tego typu instalacji zamiast zainstalować moduły **Perl** z pakietów binarnych dystrybucji, gdyż dzięki temu niniejszy artykuł będzie mógł się przydać tam, gdzie nie ma możliwości użycia pakietów **Debiana**.

Wywołuję powłokę modułu **CPAN**:

```
# perl -e shell -MCPAN
Terminal does not support AddHistory.
cpan shell -- CPAN exploration and modules installation (v1.9402)
Enter 'h' for help.
```

i wydaję zbiorczą komendę instalacji wszystkich potrzebnych modułów:

```
cpan[1]> install DBI DBD:Pg GD GD:Graph HTML::Template CGI Expect
Time::ParseDate Date::Calc
```

## UWAGA!

Powyższa instalacja uwzględnia wsparcie dla połączeń z bazą danych **PostgreSQL** (bo taka użyta jest w wykorzystanym na potrzeby artykułu środowisku). W przypadku używania **MySQL** zamiast modułu **DBD::Pg** podaje się **DBD::mysql**. Natomiast połączenia do bazy danych **SQLite** nie są wspierane przez **Bweb**.

## NOTKA AUTORA

Podczas pisania tego artykułu przy instalacji wymienionych wyżej modułów Perl w dniu 4.06.2011 odnotowałem, że nie jest możliwe zainstalowanie modułu **Perl CGI**, gdyż nie przechodzi jeden z testów wykonywanych przed kompilacją. Wyglądało mi to na błąd składni w teście (aczkolwiek nie jestem tego do końca pewien) i zmuszony byłem do ominięcia przeprowadzania testów przy instalacji. W tym celu komendę instalacji trzeba poprzedzić słowem notest, np:

```
cpan[1]> notest install CGI
```

## INSTALACJA BWEB

Aplikacja Bweb znajduje się w pakiecie o nazwie bacula-gui, który można pobrać z oficjalnej strony projektu Bacula pod adresem:

<http://bacula.org/en/?page=downloads>

Po pobraniu archiwum pakietu **bacula-gui-5.0.3** wykonuję komendy:

```
# tar zxvf bacula-gui-5.0.3.tar.gz
# cd bacula-gui-5.0.3/bweb
```

Pozostając w katalogu z rozpakowanym archiwum bacula-gui-5.0.3/bweb instaluję moduły własne **Bweb** do struktury modułów Perl.

Najpierw generuję poprzez skrypt Makefile.PL plik Makefile

```
# perl Makefile.PL
1..8
ok 1 - require DBI;
ok 2 - require GD::Graph;
ok 3 - require GD;
ok 4 - require HTML::Template;
ok 5 - require CGI;
ok 6 - require Expect;
ok 7 - require Time::ParseDate;
ok 8 - require Date::Calc;
Writing Makefile for Bweb
```

następnie instaluję moduły **Bweb**:

```
# make install
cp lib/GTime.pm blib/lib/GTime.pm
cp lib/Bconsole.pm blib/lib/Bconsole.pm
cp lib/Bweb.pm blib/lib/Bweb.pm
cp lib/GBalloon.pm blib/lib/GBalloon.pm
cp lib/CCircle.pm blib/lib/CCircle.pm
Manifying blib/man3/GTime.3pm
Manifying blib/man3/Bconsole.3pm
Manifying blib/man3/Bweb.3pm
Manifying blib/man3/GBalloon.3pm
Manifying blib/man3/CCircle.3pm
Installing /usr/local/share/perl/5.10.1/Bconsole.pm
Installing /usr/local/share/perl/5.10.1/GTime.pm
Installing /usr/local/share/perl/5.10.1/CCircle.pm
Installing /usr/local/share/perl/5.10.1/Bweb.pm
Installing /usr/local/share/perl/5.10.1/GBalloon.pm
Installing /usr/local/man/man3/Bconsole.3pm
Installing /usr/local/man/man3/Bweb.3pm
Installing /usr/local/man/man3/GTime.3pm
Installing /usr/local/man/man3/GBalloon.3pm
Installing /usr/local/man/man3/CCircle.3pm
Appending installation info to /usr/local/lib/perl/5.10.1/perllocal.pod
```

Domyślną lokalizacją pliku konfiguracyjnego **Bweb** jest katalog:

```
/etc/bacula/bweb.conf
```

W opisywanym przypadku chcę umieścić ten plik w lokalizacji:

```
/usr/local/bacula/bweb/bweb.conf
```

Z tego też powodu potrzeba wyedytować perlowy moduł **Bweb**:

```
/usr/local/share/perl/5.10.1/Bweb.pm
```

i podmienić w nim linię:

```
our $config_file= '/etc/bacula/bweb.conf';
```

na linię:

```
our $config_file= '/usr/local/bacula/bweb/bweb.conf';
```

Od tej pory Bweb będzie odwoływał się do swego pliku konfiguracyjnego w nowej lokalizacji.

Kolejnym krokiem jest instalacja skryptów **CGI** do katalogu skryptów CGI serwera **Apache**. Realizują to poprzez komendy:

```
mkdir -m 755 /usr/lib/cgi-bin/bweb  
install -m 755 -o root -g root cgi/*.pl /usr/lib/cgi-bin/bweb
```

Należy pamiętać, że lokalizacja:

```
/usr/lib/cgi-bin/
```

jest domyślną lokalizacją na skrypty **CGI** serwera **WWW Apache**. W przypadku użycia innego serwera WWW potrzeba zmodyfikować powyższe wywołanie do swoich potrzeb.

Tworzę katalog na plik konfiguracyjny **Bweb** oraz sam plik konfiguracyjny zgodnie z określoną powyżej lokalizacją z modułu perla **Bweb.pm**:

```
mkdir -m 755 /usr/local/bacula/bweb  
touch /usr/local/bacula/bweb/bweb.conf
```

Wpisuję do niego początkową konfigurację:

```
$VAR1 = bless( {  
  'enable_security' => 0,  
  'graph_font' => '/usr/share/fonts/truetype/ttf-dejavu/DejaVuSans.ttf',  
  'config_file' => '/usr/local/bacula/bweb/bweb.conf',  
  'password' => '',  
  'dbi' => 'DBI:Pg:database=bacula;host=10.0.0.3',  
  'user' => 'bacula',  
  'debug' => 0,  
  'error' => '',  
  'stat_job_table' => 'Job',  
  'display_log_time' => 0,  
  'lang' => 'en',  
  'wiki_url' => '',  
  'name' => undef,  
  'bconsole' => 'sudo /usr/local/bacula/sbin/bconsole -n -c /usr/local/  
bacula/etc/bconsole.conf',  
  'template_dir' => '/usr/local/bacula/bweb/tpl',  
  'enable_security_acl' => 0,  
  'email_media' => 'gani@localhost',  
  'default_age' => '7d'  
}, 'Bweb::Config' );
```

W wartości **'dbi'** podałem adres maszyny z serwerem **PostgreSQL** jako **10.0.0.3** (adres maszyny **hardstar**), gdyż serwer bazy danych Bacula znajduje się w zdalnej lokalizacji w stosunku do maszyny z instalacją **Bweb**.

Jako że serwer WWW nie może domyślnie posługiwać się wywołaniami konsoli **bconsole**, z tego też powodu przed komendą **bconsole** w pliku konfiguracyjnym **bweb.conf** umieściłem **sudo**. Aby wprowadzona zmiana miała możliwość zadziałać, trzeba dodać do pliku:

```
/etc/sudoers
```

wpis, który nada serwerowi WWW możliwość wykonywania programu konsoli **bconsole**. Ważne jest tu określenie użytkownika, którym posługuje się serwer WWW. W moim przypadku serwer WWW odwołuje się do plików (w tym do programów) z użytkownika o nazwie **www-data**. Dlatego też umieściłem w pliku **sudoers** taki wpis:

```
www-data ALL=NOPASSWD: /usr/local/bacula/sbin/bconsole
```

W kolejnym kroku instaluję szablony HTML aplikacji **Bweb**. Najpierw tworzę katalog na szablony:

```
mkdir -p /usr/local/bacula/bweb/tpl/en
```

a następnie będąc w katalogu **bacula-gui-5.0.3/bweb** instaluję do niego szablony:

```
install -m 644 -o root -g root lang/en/tpl/*.tpl /usr/local/bacula/bweb/tpl/en/
```

**Bweb** w swej pracy wykorzystuje dodatkowe funkcje **SQL** oraz tabele, które nie są tworzone w domyślnej instalacji Bacula. Z tego powodu trzeba zaaplikować do bazy danych Bacula zestaw kilku funkcji i tabel. Plik SQL o nazwie **bweb-postgresql.sql** z zapytaniami tworzącymi to środowisko można znaleźć w katalogu **script** archiwum ze źródłami **bacula-gui**, czyli (dla **PostgreSQL**):

```
bacula-gui-5.0.3/bweb/script/bweb-postgresql.sql
```

Przed jego zaaplikowaniem trzeba skopiować go do komputera z bazą Bacula (w tym wypadku do komputera **hardstar**) oraz dodać do bazy danych obsługę języka **PL/pgSQL** (w innym wypadku uruchomienie skryptu z funkcjami i tabelami zakończy się błędem). Dodanie języka **PL/pgSQL** sprowadza się do wydania dwóch komend:

```
# su postgres
$ echo 'CREATE PROCEDURAL LANGUAGE plpgsql;' | psql bacula
```

Wracam do konta użytkownika **root** i aplikuję do struktury bazy danych tabele i funkcje **SQL** dla **Bweb**, poprzez poniższą komendę:

```
# psql -U bacula bacula < script/bweb-postgresql.sql
```

## UWAGA!

Obydwie powyższe komendy wydajemy na komputerze z zainstalowaną i działającą bazą danych Bacula.

Na koniec pozostało jeszcze skonfigurowanie serwera **WWW Apache** do pracy z **Bweb**. Zgodnie z instrukcją załączoną wraz ze źródłami **Bweb** dodałem następujące wpisy do jednej z moich sekcji **VirtualHost**, a dokonałem tego przez dodanie do pliku:

```
/etc/apache2/sites-enabled/000-default
```

następujących wpisów:

```
Alias /bweb /var/www/bweb
<Directory /var/www/bweb>
    Options FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    allow from all
</Directory>

ScriptAlias /cgi-bin/bweb /usr/lib/cgi-bin/bweb
<Directory /usr/lib/cgi-bin/bweb>
    Options ExecCGI -MultiViews +SymLinksIfOwnerMatch
    AuthType Basic
    AuthName MyPrivateFile
    AuthUserFile /etc/apache2/htpasswd
    AllowOverride None
    Require valid-user
</Directory>
```

Wpisy te uwzględniają autoryzację do interfejsu **Bweb** poprzez konta użytkowników zdefiniowanych w pliku:

```
/etc/apache2/htpasswd
```

Dodaję więc jednego użytkownika, na którego będę się logował do Bweb. Przy pomocy programu **htpasswd** dodaję użytkownika z określonym hasłem:

```
# htpasswd -c /etc/apache2/htpasswd gani
New password:
Re-type new password:
Adding password for user gani
```

Przełącznik **-c** spowoduje stworzenie nowego pliku **htpasswd**. Dodawanie kolejnych użytkowników odbywa się w ten sam sposób, lecz bez przełącznika użycia **-c** (gdyż plik **htpasswd** już będzie istniał).

Powyższe wpisy do pliku konfiguracyjnego serwera **Apache** uwzględniają również lokalizację:

```
/var/www/bweb
```

w której będą znajdować się pliki graficzne, skrypty JavaScript, kaskadowe arkusze stylów oraz pliki HTML. Wszystkie one będą wykorzystywane przy wyświetlaniu podstron **Bweb**. Aby skopiować te pliki ze źródeł **Bweb** do powyższego katalogu pozostają w katalogu z rozpakowanym archiwum *bacula-gui-5.0.3/bweb* i wydaję komendy:

```
# mkdir /var/www/bweb
# install -m 644 -o root -g root html/*.{js,png,css,gif,ico,html} /var/
www/bweb
```



Pozostaje jeszcze zrestartować serwer **Apache**

```
/etc/init.d/apache2 restart
```

i można już cieszyć się dobrodziejstwami **Bweb**, którego interfejs dostępny będzie pod adresem:

```
http://{tutaj-adres-virtual-hosta}/bweb
```

## DODATKI

Skonfigurowałem podstawową funkcjonalność narzędzia **Bweb**. Istnieje natomiast kilka dodatków, które mogą jeszcze bardziej uatrakcyjnić pracę z Bweb. Przedstawiam je poniżej.

## LISTINGI ZE STATUSAMI ZADAŃ

**Bweb** daje możliwość oglądania i analizowania listingów statusów zakończenia zadań (np. backupu) czyli tzw. logów zadań Bacula. Aby móc oglądać je z poziomu **Bweb** trzeba dodać do zasobu **Messages** wykorzystywanego do wykonywanych zadań linii:

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

czyli zasób **Messages** dla zadań może wyglądać jak poniżej (dodałem ostatnią linię):

```
Messages {
  Name = JobMessages
  MailCommand = "mail -u root -s \"Bacula %n %t %e of %c %l\" %r"
  Mail = gani@localhost = all, !skipped
  Console = all, !skipped
  Append = "/usr/local/bacula/var/bacula/working/bacula.log" = all
  catalog = all, !skipped, !saved
}
```

Dodany wpis spowoduje, że logi z zadań będą przetrzymywane również w bazie danych Bacula (dokładniej w tabeli o nazwie **log**).

## WIDOK STRUKTURY PLIKÓW ZADAŃ

Istnieje w **Bweb** możliwość oglądania w postaci graficznej struktury zapisanych przez zadanie (backup) plików. Co więcej, można nawet "spacerować" po tej strukturze i oglądać procentową i bajtową zajętość każdej z warstw struktury katalogów zadania backupu. Od siebie dodam, że to jedna z najprzyjemniejszych funkcjonalności, jaką zaskoczył mnie **Bweb**. Funkcjonalność ta przydać może się m.in. wtedy, gdy chcemy zobaczyć zawartość backupu, gdy np. jest on rozmiarowo niepokojąco duży lub mały.

Aby włączyć widok struktury plików zadań trzeba w pierwszej kolejności stworzyć katalog do przetrzymywania plików widoków struktury plików oraz nadać mu odpowiednie prawa i właściciela (w tym wypadku jest to użytkownik serwera **Apache** czyli **www-data**):

```
mkdir -p /usr/local/bacula/bweb/var/spool/bweb
chmod 700 /usr/local/bacula/bweb/var/spool/bweb
chown www-data /usr/local/bacula/bweb/var/spool/bweb
```

Następnie trzeba zdefiniować tą lokalizację w pliku konfiguracyjnym **Bweb**:

```
/usr/local/bacula/bweb/bweb.conf
```

poprzez dodanie wartości:

```
'fv_write_path' => '/usr/local/bacula/bweb/var/spool/bweb',
```

Mój plik bweb.conf wygląda teraz jak poniżej.

```
$VAR1 = bless( {
  'enable_security' => 0,
  'graph_font' => '/usr/share/fonts/truetype/ttf-dejavu/DejaVuSans.ttf',
  'config_file' => '/usr/local/bacula/bweb/bweb.conf',
  'password' => '',
  'dbi' => 'DBI:Pg:database=bacula;host=10.0.0.3',
  'user' => 'bacula',
  'debug' => 0,
  'error' => '',
  'stat_job_table' => 'Job',
  'display_log_time' => 0,
  'lang' => 'en',
  'wiki_url' => '',
  'name' => undef,
  'fv_write_path' => '/usr/local/bacula/bweb/var/spool/bweb',
  'bconsole' => 'sudo /usr/local/bacula/sbin/bconsole -n -c /usr/local/
bacula/etc/bconsole.conf',
  'template_dir' => '/usr/local/bacula/bweb/tpl',
  'enable_security_acl' => 0,
  'email_media' => 'gani@localhost',
  'default_age' => '7d'
}, 'Bweb::Config' );
```

Potrzeba również dodać odpowiedni wpis w pliku konfiguracyjnym serwera **WWW Apache** w tej samej sekcji **VirtualHost** co poprzednio dodawane wpisy **Bweb**:

```
Alias /bweb/fv /usr/local/bacula/bweb/var/spool/bweb
<Directory "/usr/local/bacula/bweb/var/spool/bweb">
  Options None
  AllowOverride AuthConfig
  Order allow,deny
  Allow from all
</Directory>
```

## UWAGA!

Powyższy wpis dodaje się przed wpisami umieszczonymi wcześniej w konfiguracji VirtualHost'a. Teraz wszystkie wpisy odnośnie Bweb w pliku `/etc/apache2/sites-enabled/000-default` wyglądają jak poniżej:

```
Alias /bweb/fv /usr/local/bacula/bweb/var/spool/bweb
<Directory "/usr/local/bacula/bweb/var/spool/bweb">
    Options None
    AllowOverride AuthConfig
    Order allow,deny
    Allow from all
</Directory>

Alias /bweb /var/www/bweb
<Directory /var/www/bweb>
    Options FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    allow from all
</Directory>

ScriptAlias /cgi-bin/bweb /usr/lib/cgi-bin/bweb
<Directory /usr/lib/cgi-bin/bweb>
    Options ExecCGI -MultiViews +SymLinksIfOwnerMatch
    AuthType Basic
    AuthName MyPrivateFile
    AuthUserFile /etc/apache2/htpasswd
    AllowOverride None
    Require valid-user
</Directory>
```

Restartuję serwer **Apache**.

```
/etc/init.d/apache2 restart
```

Aby móc oglądać graficzną reprezentację struktury zapisanych w backupie katalogów, potrzebne będzie jeszcze dodanie wpisu do zasobu **Job**, który zaktualizuje po każdym backupie odpowiednie dane na potrzeby wyświetlania grafów z katalogami backupu. Wygodnym do tego celu miejscem jest dyrektywa **RunAfterJob**, której zawartość to:

```
RunAfterJob = "/usr/lib/cgi-bin/bweb/bresto.pl action=batch"
```

Oto definicja przykładowego zadania backupu z użytym wpisem do aktualizacji widoków zapisanych plików:

```
Job {
  Name = JakisJob
  Type = Backup
  Level = Incremental
  Pool = JakisPool
  Storage = JakiesUrzadzenie
  Client = klient-fd
  Messages = Messages
  FileSet = JakisFileSet
  RunAfterJob = "/usr/lib/cgi-bin/bweb/bresto.pl action=batch"
}
```

Sama komenda:

```
# /usr/lib/cgi-bin/bweb/bresto.pl action=batch
```

może zostać wywołana ręcznie w celu aktualizacji tabel **Bweb** dla wcześniej wykonanych backupów.

## PRZYWRACANIE DANYCH

Aby mieć możliwość przywracania danych z poziomu **Bweb**, potrzeba zaopatrzyć instalację **Bweb** o dodatkowy zewnętrzny komponent o nazwie **ExtJS**. W skrócie mówiąc jest to swego rodzaju zestaw bibliotek JavaScript, które pomogą w poruszaniu się po drzewie katalogów z backupów. Obecną najnowszą wersją **ExtJS** jest wersja **4.0.1**, jednak twórca **Bweb** zaleca użycie starszej wersji **2.2**. Szczerze mówiąc nie sprawdzałem działania restore z **Bweb** przy użyciu **ExtJS** w wersji **4**. Trzymając się zalecenia twórcy **Bweb**, ściągam wersję **ExtJS 2.2** z archiwum wersji projektu **ExtJS**, który znajduje się pod adresem:

[http://www.sencha.com/learn/Ext\\_Version\\_Archives](http://www.sencha.com/learn/Ext_Version_Archives)

Następnie rozpakowuję archiwum:

```
# unzip ext-2.2.zip
```

oraz przenoszę pliki **ExtJS** do odpowiedniej lokalizacji:

```
# mv ext-2.2 /var/www/bweb/ext
```

I to wszystko.

# SCREENSHOTS



This screenshot displays a table with columns for 'Name', 'Client Name', and 'Job Name / Job Statistics Job Statistics'. The table lists several jobs with their respective client names and statistics. Below the table, there are some graphical icons.

Name	Client Name	Job Name / Job Statistics Job Statistics
Job1	Client1	Job1 / Job Statistics Job Statistics
Job2	Client2	Job2 / Job Statistics Job Statistics
Job3	Client3	Job3 / Job Statistics Job Statistics

This screenshot shows a configuration form for a job. It includes fields for 'Name', 'Client', 'Storage', and 'Retention'. There are also dropdown menus and checkboxes. The interface is clean and organized.

This screenshot displays a table with columns for 'Name', 'Client Name', and 'Job Name / Job Statistics Job Statistics'. The table lists several jobs with their respective client names and statistics. Below the table, there are some graphical icons.

Name	Client Name	Job Name / Job Statistics Job Statistics
Job1	Client1	Job1 / Job Statistics Job Statistics
Job2	Client2	Job2 / Job Statistics Job Statistics
Job3	Client3	Job3 / Job Statistics Job Statistics

This screenshot shows a log file with multiple lines of text. The text is formatted and includes various status messages and error reports. The interface includes a menu bar at the top and a sidebar on the left.

This screenshot shows a configuration form for a job. It includes fields for 'Name', 'Client', 'Storage', and 'Retention'. There are also dropdown menus and checkboxes. The interface is clean and organized.

This screenshot displays a table with columns for 'Name', 'Client Name', and 'Job Name / Job Statistics Job Statistics'. The table lists several jobs with their respective client names and statistics. Below the table, there are some graphical icons.

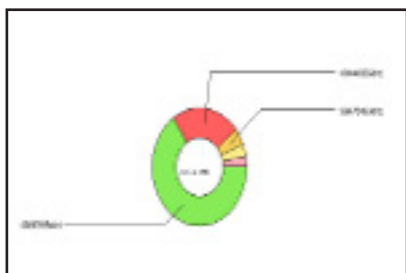
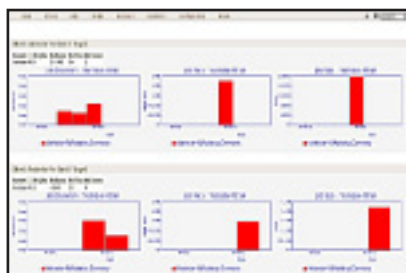
Name	Client Name	Job Name / Job Statistics Job Statistics
Job1	Client1	Job1 / Job Statistics Job Statistics
Job2	Client2	Job2 / Job Statistics Job Statistics
Job3	Client3	Job3 / Job Statistics Job Statistics

This screenshot shows a log file with multiple lines of text. The text is formatted and includes various status messages and error reports. The interface includes a menu bar at the top and a sidebar on the left.

This screenshot shows a configuration form for a job. It includes fields for 'Name', 'Client', 'Storage', and 'Retention'. There are also dropdown menus and checkboxes. The interface is clean and organized.

This screenshot shows a configuration form for a job. It includes fields for 'Name', 'Client', 'Storage', and 'Retention'. There are also dropdown menus and checkboxes. The interface is clean and organized.

This screenshot shows a log file with multiple lines of text. The text is formatted and includes various status messages and error reports. The interface includes a menu bar at the top and a sidebar on the left.



## PODSUMOWANIE

Opisany proces instalacji został w znacznej mierze zautomatyzowany poprzez skrypt powłoki o nazwie **install\_bweb** znajdujący się w archiwum ze źródłami **Bweb**. Przed jego użyciem trzeba wyedytować zawartość skryptu i dopasować lokalizacje na poszczególne komponenty **Bweb**. W artykule zdecydowałem się pójść tzw. drogą "**step by step**" czyli krok po kroku opisać instalację z pominięciem skryptu **install\_bweb**.

Przy tworzeniu niniejszego artykułu wykorzystałem materiały znajdujące się w archiwum ze źródłami **Bweb**, a dokładniej znajdujące się w pliku:

```
bacula-gui-5.0.3/bweb/INSTALL
```

gdzie czytelnik znaleźć może anglojęzyczną instrukcję instalacji **Bweb**.

Na koniec życzę czytelnikom powodzenia w instalacji **Bweb**.