

Отчеты в Bareos

В данной системе бэкапирования к сожалению нет удобного визуального контроля за тем какие файлы и в какой момент были записаны на ленточное хранилище.

И я решил сам сделать такой функционал т.к. смотреть через **bconsole** не очень удобно из-за привязки записанных файлов к заданию **job**

Я изучил структуру базы данных и зависимости таблиц между собой, что бы понять какие данные мне нужно включить в отчетность и что бы ничего лишнего не попало туда.

Кратко описываю что мне нужно извлечь из базы

Есть таблицы **job**, **file**, **media**

в таблице **job** есть столбец **name**, из него мы считываем только содержимое с названием **BackupOnTape**,

так же есть столбец **jobid** и он идентичен в таблице **file**

Теперь в таблице **file** сопоставив **jobid** мы должны вытащить из столбца **name** который в таблице **file** название

В таблице **media** есть столбец **poolid** он понадобится для понимания какой используется пул

Но мне из таблицы **media** более важно знать что в столбце **volumename** и выводить эту информацию. Еще в таблице **media** есть вот такой столбец **mediatype**, мне важно что бы в финальном отчете он тоже фигурировал, но при условии что в **mediatype** будет исключительно содержимое с названием **LTO**

И из таблицы **job** нужны два столбца **starttime** и **endtime**, которые отвечают за начало и конец записи на ленту.

Таким образом резюмируем

Мне нужно сформировать отчет в котором будет:

- **volumename, poolid, mediatype** из таблицы **media**
- **name** из таблицы **file**
- **name, jobid, starttime, endtime** из таблицы **job**

У меня получился вот такой SQL запрос

download

```
SELECT
  m.volumename AS volumename,
  f.name AS file_name,
  j.name AS job_name,
  j.jobid AS jobid,
  m.mediatype AS mediatype,
  j.starttime AS starttime,
  j.endtime AS endtime,
  m.poolid AS poolid
FROM
  public.job j
```

```
JOIN
  public.file f ON j.jobid = f.jobid
JOIN
  public.media m ON m.poolid IS NOT NULL
WHERE
  j.name = 'BackupOnTape' AND
  m.mediatype = 'LTO' AND
  m.poolid = 2;
```

Для проверки можно сохранить в файл вывод запроса

download

```
COPY (
  SELECT
    m.volumename AS volumename,
    f.name AS file_name,
    j.name AS job_name,
    j.jobid AS jobid,
    m.mediatype AS mediatype,
    j.starttime AS starttime,
    j.endtime AS endtime,
    m.poolid AS poolid
  FROM
    public.job j
  JOIN
    public.file f ON j.jobid = f.jobid
  JOIN
    public.media m ON m.poolid IS NOT NULL
  WHERE
    j.name = 'BackupOnTape' AND
    m.mediatype = 'LTO' AND
    m.poolid = 2
) TO '/ВАШ ПУТЬ/report.txt' WITH (FORMAT csv, HEADER);
```

Но это все сырые данные и так мы только убедились что все работает, теперь нужно создать веб интерфейс и прикрутить кнопку в основной веб интерфейс bareos

Создание веб интерфейса

Создаем такую структуру

download

```
/var/www/html/
├── reports_bareos
│   ├── reports_bareos.py
│   └── static
```

```
├── style.css
└── templates
    ├── index.html
    └── report.html
```

Содержимое файла index.html

download

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Форма для отчета</title>
  <link rel="stylesheet" href="{ url_for('static', filename='style.css')
  } }">
</head>
<body>
  <h1>Форма для отчета</h1>
  <form action="/report" method="get">
    <label for="volumename">Номер ленты (Пример 000005):</label>
    <input type="text" id="volumename" name="volumename">

    <label for="start_date">Дата начала:</label>
    <input type="date" id="start_date" name="start_date">

    <label for="end_date">Дата окончания:</label>
    <input type="date" id="end_date" name="end_date">

    <input type="submit" value="Сформировать отчет">
  </form>
</body>
</html>
```

Содержимое файла report.html

download

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Отчет</title>
```

```
<link rel="stylesheet" href="{ url_for('static', filename='style.css')
}}">
</head>
<body>
  <h1>Отчет</h1>
  <table border="1">
    <tr>
      <th>Volume Name</th>
      <th>File Name</th>
      <th>Job Name</th>
      <th>Job ID</th>
      <th>Media Type</th>
      <th>Start Time</th>
      <th>End Time</th>
      <th>Pool ID</th>
    </tr>
    {% for row in reports %}
    <tr>
      <td>{{ row[0] }}</td>
      <td>{{ row[1] }}</td>
      <td>{{ row[2] }}</td>
      <td>{{ row[3] }}</td>
      <td>{{ row[4] }}</td>
      <td>{{ row[5] }}</td>
      <td>{{ row[6] }}</td>
      <td>{{ row[7] }}</td>
    </tr>
    {% endfor %}
  </table>
  <br>
  <a href="/">Вернуться назад</a>
</body>
</html>
```

Содержимое файла style.css

download

```
body {
  font-family: Arial, sans-serif;
  background-color: #f4f4f4;
  margin: 0;
  padding: 20px;
}

h1 {
  text-align: center;
  color: #333;
```

```
}

form {
  max-width: 600px;
  margin: auto;
  background: #fff;
  padding: 20px;
  border-radius: 8px;
  box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
}

label {
  display: block;
  margin-bottom: 8px;
  color: #555;
}

input[type="text"],
input[type="date"] {
  width: 100%;
  padding: 10px;
  margin-bottom: 15px;
  border: 1px solid #ccc;
  border-radius: 4px;
}

input[type="submit"] {
  background-color: #5cb85c;
  color: white;
  border: none;
  padding: 10px 15px;
  border-radius: 4px;
  cursor: pointer;
  font-size: 16px;
  transition: background-color 0.3s;
}

input[type="submit"]:hover {
  background-color: #4cae4c;
}

/* Стили для таблицы */
table {
  width: 100%;
  border-collapse: collapse;
  margin-top: 20px;
}

table, th, td {
  border: 1px solid #ccc;
}
```

```
th {
    background-color: #5cb85c;
    color: white;
    padding: 10px;
}

td {
    padding: 10px;
    text-align: center;
}

/* Стили для ссылки */
a {
    display: block;
    text-align: center;
    margin-top: 20px;
    color: #5cb85c;
    text-decoration: none;
}

a:hover {
    text-decoration: underline;
}
```

Создаем логику работы (backend)

Логику будем описывать с помощью **Python** и фреймворка **Flask**

Содержимое файла reports_bareos.py

download

```
from flask import Flask, request, render_template
import psycopg2

app = Flask(__name__)

def get_db_connection():
    conn = psycopg2.connect(dbname='ВАША БАЗА', user='ВАШ ПОЛЬЗОВАТЕЛЬ',
password='ВАШ ПАРОЛЬ', host='localhost')
    return conn

@app.route('/')
def index():
    return render_template('index.html')
```

```
@app.route('/report', methods=['GET'])
def report():
    volumename = request.args.get('volumename')
    start_date = request.args.get('start_date')
    end_date = request.args.get('end_date')

    conn = get_db_connection()
    cur = conn.cursor()

    query = '''
SELECT
    m.volumename AS volumename,
    f.name AS file_name,
    j.name AS job_name,
    j.jobid AS jobid,
    m.mediatype AS mediatype,
    j.starttime AS starttime,
    j.endtime AS endtime,
    m.poolid AS poolid
FROM
    public.job j
JOIN
    public.file f ON j.jobid = f.jobid
JOIN
    public.media m ON m.poolid IS NOT NULL
WHERE
    j.name = 'BackupOnTape' AND
    m.mediatype = 'LTO' AND
    m.poolid = 2
    ...

    if volumename:
        query += ' AND m.volumename = %s'
        cur.execute(query, (volumename,))
    else:
        cur.execute(query)

    reports = cur.fetchall()

    cur.close()
    conn.close()

    # Фильтруем строки, у которых file_name пустая
    reports = [row for row in reports if row[1]] # row[1] - это file_name

    return render_template('report.html', reports=reports)

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000, debug=True)
```

Тестируем

Откройте порт 5000 в вашем firewall если это необходимо
Далее нужно вручную запустить приложение командой

```
python /var/www/html/reports_bareos.py
```

Теперь идем в браузер

```
http://ВАШ_IP:5000
```

И вы увидите вот эту форму

Форма для отчета

Номер ленты (Пример 000005):

Дата начала:

Дата окончания:

Значит все идет как нужно. Но запускать таким образом приложение не удобно, нужно сделать его отдельным демоном в системе

Создаем демона для reports_bareos.py

Открываем

```
nano /etc/systemd/system/reports_bareos.service
```

Содержимое будет примерно такое

download

```
[Unit]  
Description=Flask Application for Reports Bareos
```

```
After=network.target

[Service]
User=bareos
Group=bareos
WorkingDirectory=/var/www/html/reports_bareos
ExecStart=/var/lib/bareos/.local/bin/gunicorn -w 4 -b 0.0.0.0:5000
reports_bareos:app
Restart=always
RestartSec=10
Environment="FLASK_ENV=development"
Environment="PATH=/var/lib/bareos/.local/bin:$PATH"
StandardOutput=append:/var/log/reports_bareos.log
StandardError=append:/var/log/reports_bareos.log

[Install]
WantedBy=multi-user.target
```

Запускаем демона и проверяем его

```
systemctl enable reports_bareos.service
systemctl start reports_bareos.service
```

```
systemctl status reports_bareos.service
```

```
root@server:~# systemctl status reports_bareos.service
● reports_bareos.service - Flask Application for Reports Bareos
   Loaded: loaded (/etc/systemd/system/reports_bareos.service; enabled; preset: disabled)
   Active: active (running) since Tue 2024-09-24 17:03:49 MSK; 16h ago
     Main PID: 254438 (gunicorn)
       Tasks: 5 (limit: 407792)
      Memory: 73.8M
         CPU: 8.494s
    CGroup: /system.slice/reports_bareos.service
            └─254438 /usr/bin/python3 /var/lib/bareos/.local/bin/gunicorn -w 4 -b 0.0.0.0:5000 reports_bareos:app
              └─254444 /usr/bin/python3 /var/lib/bareos/.local/bin/gunicorn -w 4 -b 0.0.0.0:5000 reports_bareos:app
                └─254445 /usr/bin/python3 /var/lib/bareos/.local/bin/gunicorn -w 4 -b 0.0.0.0:5000 reports_bareos:app
                  └─254446 /usr/bin/python3 /var/lib/bareos/.local/bin/gunicorn -w 4 -b 0.0.0.0:5000 reports_bareos:app
                    └─254962 /usr/bin/python3 /var/lib/bareos/.local/bin/gunicorn -w 4 -b 0.0.0.0:5000 reports_bareos:app

Sep 24 17:03:49 bacularestore systemd[1]: Started Flask Application for Reports Bareos.
```

Теперь идем в браузер

```
http://ВАШ IP:5000
```

И вы увидите вот эту форму

Форма для отчета

Номер ленты (Пример 000005):

Дата начала:

Дата окончания:

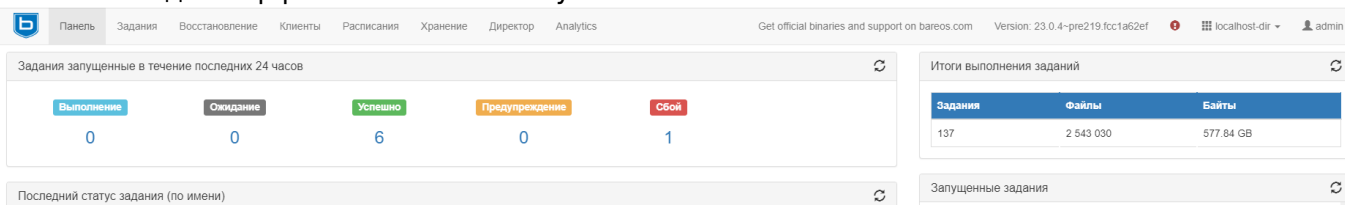
Сформировать отчет

Значит все получилось.

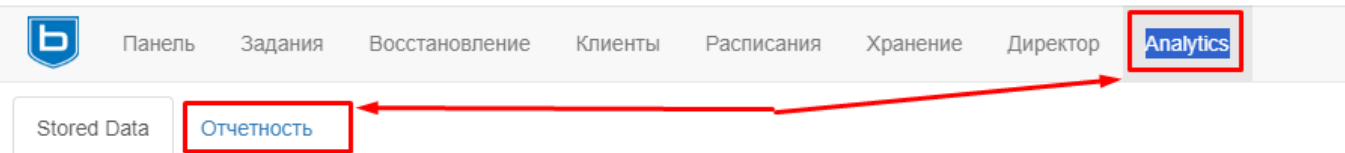
Прикручиваем кнопку отчетов в bareos-webui

Тут уже все индивидуально и куда вам будет удобнее, туда и прикручивайте, я покажу свой пример как сделал я

Внешний вид интерфейса вот такой по умолчанию



Я решил в пункте **Analytics** сделать кнопку отчетности



Открываем файл

```
nano /usr/share/bareos-webui/module/Analytics/view/analytics/analytics/index.phtml
```

Ищем в нем где упоминается **Stored Data** как видно на скрине выше это такой раздел И туда отдельным пунктом добавляем кнопку перехода в наш созданный интерфейс

```
<li><a href="http://10.0.5.210:5000" class="nav-link">Отчетность</a></li>
```

Визуально выглядит вот так

```
<?php if($this->acl_alert) : echo $this->ACLAlert($this->invalid_commands); elseif(!$this->acl_alert) : ?>
<ul class="nav nav-tabs">
<li class="active"><a href="<?php echo $this->url('analytics', array('action'=>'index')); ?>"><?php echo $this->translate('Stored Data'); ?></a></li>
<?php
if ($_SESSION['bareos']['configuration_resource_graph']) {
$url = $this->url('analytics', array('action'=>'configuration'));
$text = $this->translate('Configuration');
echo '<li><a href="' . $url . '">' . $text . '</a></li>';
}
?>
<li><a href="http://10.0.5.210:5000" class="nav-link">0тчетность</a></li>
</ul>
```

На этом все, мы проделали отличную работу по созданию интерфейса для отчетности. Его можно расширять если вам необходимы еще какие-то данные из базы визуализировать

From:

<https://wiki.fellk.ru/dokuwiki/> - **Игорь Fellk**

Permanent link:

https://wiki.fellk.ru/dokuwiki/doku.php/poleznosti/bareos_reports?rev=1727247953

Last update: **2024/09/25 07:05**

