

Отказоустойчивость: Резервное копирование. Vasula.



Александр
Зубарев



Александр Зубарев

Председатель цикловой комиссии “Информационной
безопасности инфокоммуникационных систем”

АКТ (ф) СПбГУТ



[Александр Зубарев](#)



Вспомним материал прошлой лекции

Вопрос: Для чего используется Peacemaker?



Вспомним материал прошлой лекции

Вопрос: Для чего используется Peacemaker?

Ответ: Для создания кластера высокой доступности.



Вспомним материал прошлой лекции

Вопрос: Для чего используется DRDB?



Вспомним материал прошлой лекции

Вопрос: Для чего используется DRDB?

Ответ: Для репликации ресурсов и создания надежности и доступности ресурсов.


Предисловие

Сегодня мы:

- поговорим о технологиях и методах построения резервного копирования различных компонентов системы, программ;
- рассмотрим программы, позволяющие выполнять данные процессы;
- настроим резервное копирование отдельных компонентов системы и баз данных.

План занятия

1. [Виды резервного копирования](#)
2. [Представители рынка резервного копирования](#)
3. [Bacula](#)
4. [rsync](#)
5. [Резервирование важных ресурсов](#)
6. [Итоги](#)
7. [Домашнее задание](#)



Виды резервного копирования

Резервирование

А вообще, зачем резервировать (делать backup)?



Но как скажет любой системный администратор:





Резервирование

Резервирование — процесс повышения надежности путем добавления избыточности данных или устройств для быстрого восстановления работоспособности системы.

Надёжность изделия — количественный и качественный показатель, отвечающий за то, сколько времени может работать то или иное устройство. Например, циклы записи ячейки памяти на SSD.

Резервирование

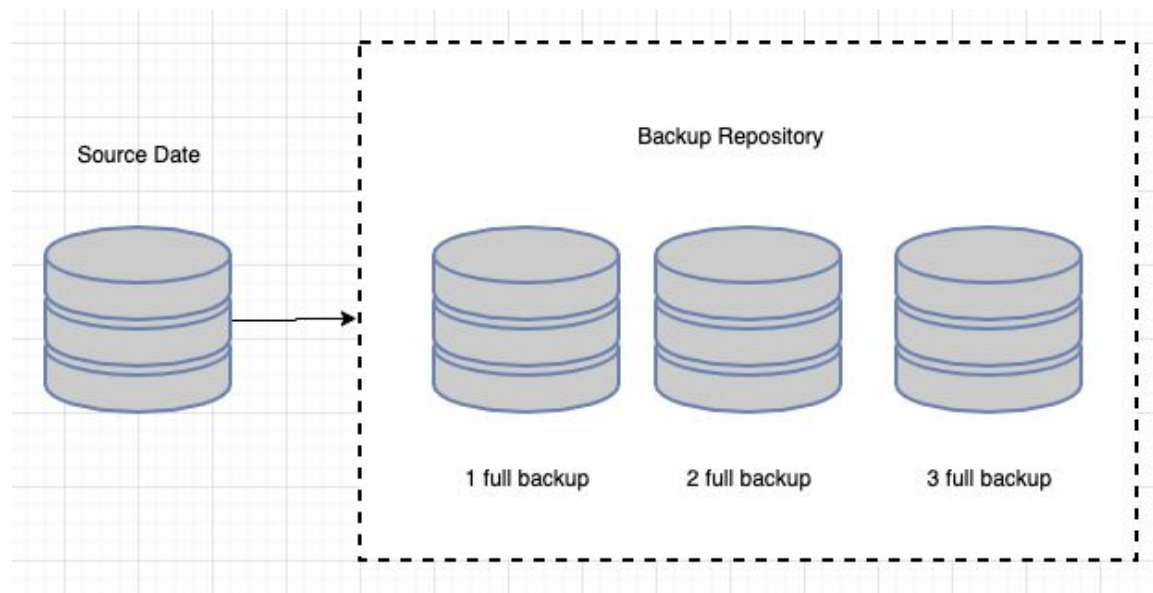
Виды резервного копирования:

- полное резервное копирование;
- дифференциальное резервное копирование;
- инкрементное резервное копирование;
- клонирование;
- резервное копирование в виде образа;
- резервное копирование в режиме реального времени;
- холодное резервирование;
- горячее резервирование.

Полное резервное копирование

Full BackUp делает полное копирование всего.

- Самый долгий с точки зрения процесса;
- Дает нагрузку на диски и на сеть, если она сетевая;
- Самый надежный и быстрый с точки зрения восстановления данных.



Дифференциальное резервное копирование

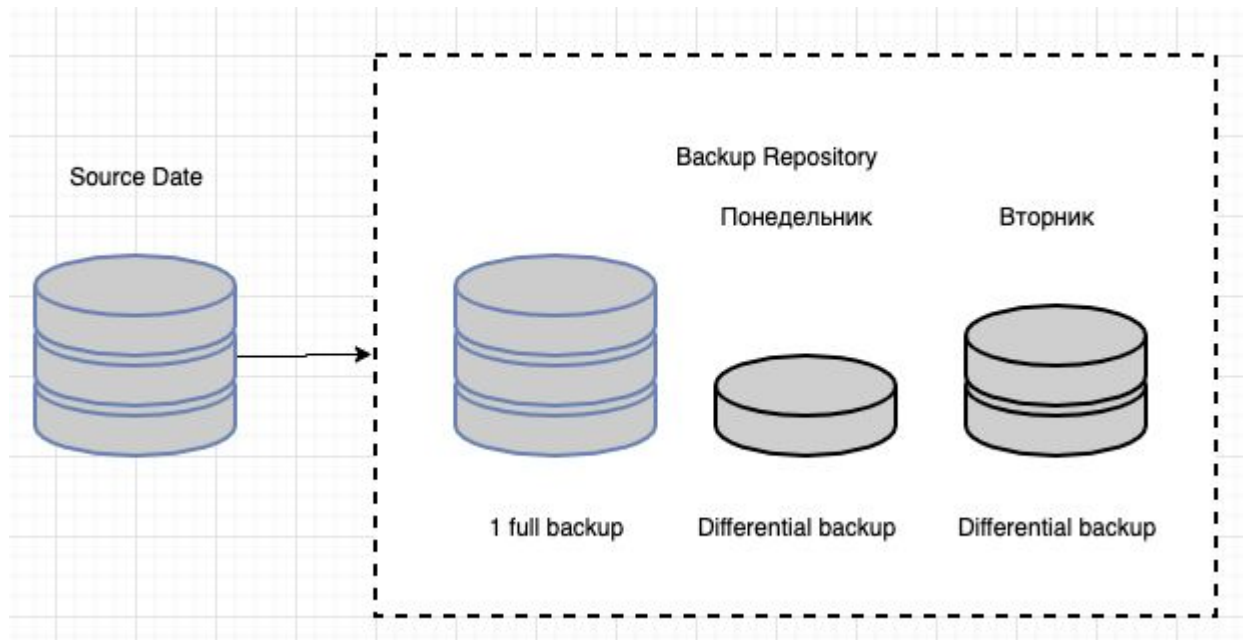
Принцип работы:

- сначала делается полное резервное копирование,
- затем при каждом запуске процесса резервируются только измененные данные, но точкой отсчета является состояние времени полного бэкапа.

Резервное копирование быстрее, чем полное, но медленнее, чем инкрементное. Восстановление наоборот.

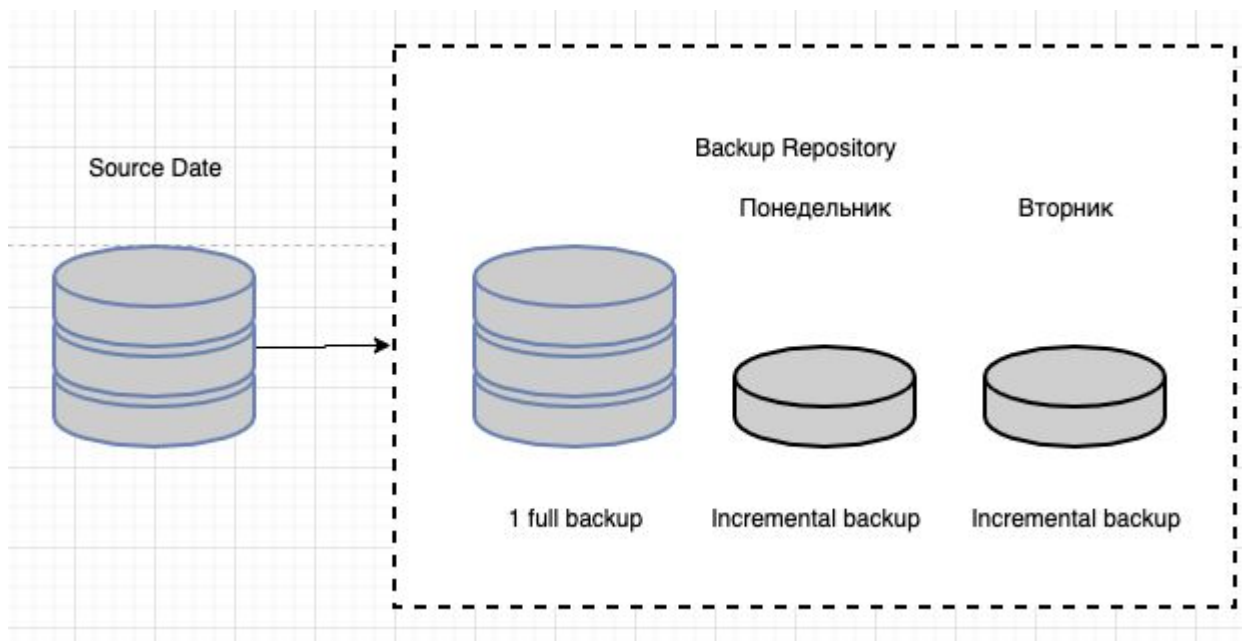
Памяти на определенный период меньше, чем у полного, но больше, чем у инкрементного .

Дифференциальное резервное копирование



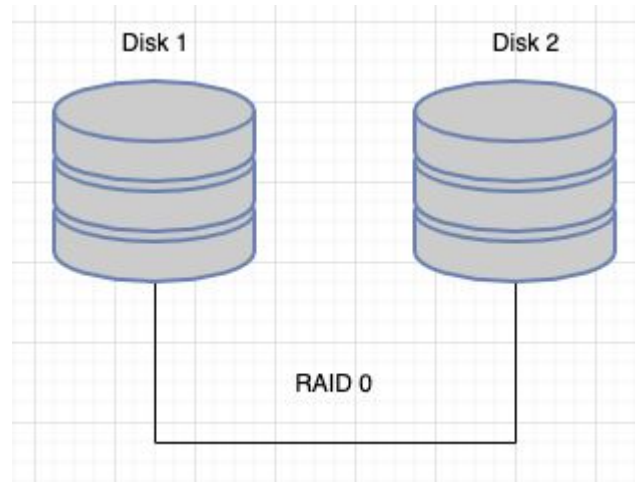
Инкрементное резервное копирование

Инкрементное копирование работает как дифференцированное копирование, но в отличии от него бэкапятся данные, которые были изменены из последнего слепка, то есть отправная точка каждого нового бэкапа это бэкап $n-1$.



Клонирование

При **клонировании** делается резервное копирование не системы или компонента, а всего диска. Например, целого раздела или всего диска. Если произойдет сбой с устройством, можно восстановить весь раздел с данными. С другой стороны, это рейд массив зеркальный, который клонирует все данные на двух (или более) носителях.





Резервное копирование в виде образа

Резервное копирование в виде образа, как и клонирование, производится в какой-то слепок системы. Отличие в том, что образ системы можно, например, поместить в контейнер или виртуализировать. Тем самым мы получаем образ с возможностью развертки без привязки к чему-либо (ну кроме всего аппаратного =)).

Резервное копирование в режиме реального времени


- происходит либо на логическом уровне, либо на аппаратном — RAID на аппаратно/программном;
- механизмы типа теневого копирования;
- процесс идет при изменении любого файла, он сохраняет копию состояния до и после;
- процесс происходит в реальном времени.



Холодное и горячее резервирование

Холодное и горячее резервирование — это разновидности резервирования. При работающей системе выполняется горячее резервирование, при перезагрузке компьютера — холодное.

Исходя из этого, холодное резервное копирование проходит быстрее, но в это время невозможно пользоваться системой.



Представители рынка резервного копирования

Представители рынка резервного копирования

Виды программных продуктов:

- [bacula](#)
- [rsync](#)
- [BackupPC](#)
- [luckyBackup](#)
- [rdiff-backup](#)



Данный список исключительно свободное программное обеспечение, но не будем забывать про Acronis, iDrive, Microsoft и много другое.



Bacula



Bacula

Bacula — кроссплатформенное клиент-серверное программное обеспечение, позволяющее управлять резервным копированием, восстановлением и проверкой данных по сети для компьютеров и операционных систем различных типов.

Bacula

Bacula состоит из трех компонентов:

- **Bacula Director (сервис bacula-dir)** — основной сервис, который управляет всеми другими процессами по резервному копированию и восстановлению;
- **Bacula Storage (сервис bacula-sd)** — хранилище, предназначенное для сохранения резервных копий на диске;
- **Bacula File Daemon (сервис bacula-fd)** — клиентская часть сервиса, которая нужна для доступа к файлам на сервере, с которого будет выполняться резервное копирование.

Bacula

Установка есть в стандартных репозиториях. Для работы потребуется база данных.

```
sudo apt install bacula postgresql
```

В ходе установки потребуется выбрать, где будет база данных, выберем localhost.

Далее установка пароля для входа, можно установить любой (*главное запомнить*).

Следующим шагом требуется настроить bacula:

```
nano /etc/bacula/bacula-sd.conf
```

Bacula

В настройках есть 4 основные секции:

- **Storage** — секция с основными настройками хранилища, здесь настраивается имя хранилища, а также IP адрес на котором оно будет доступно, для локальной сети пока достаточно 127.0.0.1.
- **Director** — настраивается авторизация для управляющего сервиса, тут надо прописать имя сервиса, который может подключиться и пароль, который он должен использовать.
- **Device** — здесь настраивается способ хранения файлов на физическом диске и путь к папке, где они будут храниться.
- **Messages** — отправка сообщений, можно оставить как есть.

Bacula

```
Storage { # definition of myself
    Name = nodeOne-sd
    SDPort = 9103                # Director's port
    WorkingDirectory = "/var/lib/bacula"
    Pid Directory = "/run/bacula"
    Plugin Directory = "/usr/lib/bacula"
    Maximum Concurrent Jobs = 20
    SDAddress = 127.0.0.1
}

Director {
    Name = nodeOne-dir
    Password = "tMLJYlOKqe2U7EAFXJ8803EWRpfIyb05t"
}
```

Bacula

```
Device {
  Name = FileChgr1-Dev2
  Media Type = File1
  Archive Device = /nonexistant/path/to/file/archive/dir
  LabelMedia = yes;                # lets Bacula label unlabeled
media
  Random Access = Yes;
  AutomaticMount = yes;            # when device opened, read it
  RemovableMedia = no;
  AlwaysOpen = no;
  Maximum Concurrent Jobs = 5
}
```

Bacula

После редактирования настроек выполняем проверку получившийся конфигурации:

```
sudo /usr/sbin/bacula-sd -t -c /etc/bacula/bacula-sd.conf
```

Если ошибок не найдено, то перезагружаем сервис:

```
sudo systemctl restart bacula
```

Bacula

Настройка локального клиента производится:

```
nano /etc/bacula/bacula-fd.conf
```

В принципе, все тоже самое, но посмотрим на работающей конфигурации.

После редактирования проверяем все командой:

```
bacula-dir -t -c ./bacula-dir.conf
```

Настройка консоли

Конфигурационный файл консоли находится здесь:

etc/bacula/bconsole.conf.

В нём есть только одна секция и здесь нужно указать:

- на каком IP адресе ожидает соединений Bacula Director,
- а также пароль к нему, настроенный в секции Director файла ***/etc/bacula/bacula-dir.conf.***

```
Director {  
  Name = nodeOne-dir  
  DIRport = 9101  
  address = localhost  
  Password = "m4b745enxHN-o_Mnxdv9JSAU5iP3WHVKH"  
}
```

Bacula

Следующим шагом перезапускаем службы:

```
systemctl restart bacula-fd  
systemctl restart bacula-sd  
systemctl restart bacula-dir
```

Bacula

Запускаем консоль:

`bconsole`

Выполняем:

`run`

и выбираем нужный Backup.

Для просмотра сообщения

требуется ввести:

`messages`

```
root@nodeOne:/home/user# bconsole
Connecting to Director localhost:9101
1000 OK: 103 nodeOne-dir Version: 9.4.2 (04 February 2019)
Enter a period to cancel a command.
*run
Automatically selected Catalog: MyCatalog
Using Catalog "MyCatalog"
A job name must be specified.
The defined Job resources are:
    1: BackupClient1
    2: NodeBackup
Select Job resource (1-2): 2
Run Backup job
JobName:  NodeBackup
Level:    Full
Client:   nodeOne-fd
FileSet:  Node-configuration
Pool:     NodePool (From Job resource)
Storage:  File1 (From Job resource)
When:     2021-07-18 13:56:43
Priority:  10
OK to run? (yes/mod/no): yes
Job queued. JobId=2
You have messages.
*
```



rsync



Rsync

rsync — программа для UNIX-подобных систем, которая эффективно выполняет синхронизацию файлов и каталогов в двух местах с минимизированием трафика, используя кодирование данных при необходимости.

Установка:

```
apt install rsync
```

Rsync

Настройка конфигурации:

```
nano /etc/default/rsync
```

Меняем значение:

```
RSYNC_ENABLE = true
```

После чего создаем файл:

```
touch /etc/rsyncd.conf
```

Rsync

Вносим в конфигурационный файл:

```
nano /etc/rsyncd.conf
```

```
pid file = /var/run/rsyncd.pid  
log file = /var/log/rsyncd.log  
transfer logging = true  
munge symlinks = yes
```

Rsync

```
# папка источник для бэкапа
[data]
path = /data
uid = root
read only = yes
list = yes
comment = Data backup Dir
auth users = backup
secrets file = /etc/rsyncd.scr
```

Rsync

Запускаем rsync:

```
systemctl start rsync
```

Проверяем rsync, что работает по сети:

```
netstat -tulnp | grep rsync
```

Создаем файл с учетными данными:

```
nano /etc/rsyncd.scr
```

```
backup:12345
```

```
chmod 0600 /etc/rsyncd.scr
```


Rsync

Теперь можно настроить синхронизацию со вторым сервером.

Для этого устанавливаем rsync на втором сервере.

Настраиваем скрипт для выполнения синхронизации:

```
mkdir /root/scripts/  
touch /root/scripts/backup-node1.sh  
nano /root/scripts/backup-node1.sh
```

Rsync

```
#!/bin/bash
date
# Папка, куда будем складывать архивы – ее либо сразу создать либо не
создавать а положить в уже существующие
syst_dir=/backup/
# Имя сервера, который архивируем
srv_name=node0ne #из тестовой конфигурации
# Адрес сервера, который архивируем
srv_ip=192.168.0.1
```

Rsync

```
# Пользователь rsync на сервере, который архивируем
srv_user=backup
# Ресурс на сервере для бэкапа
srv_dir=data
echo "Start backup ${srv_name}"
# Создаем папку для инкрементных бэкапов
mkdir -p ${syst_dir}${srv_name}/increment/
```

Rsync

```
/usr/bin/rsync -avz --progress --delete
--password-file=/etc/rsyncd.scrpt ${srv_user}@${srv_ip}::${srv_dir}
${syst_dir}${srv_name}/current/ --backup
--backup-dir=${syst_dir}${srv_name}/increment/`date +%Y-%m-%d`/
/usr/bin/find ${syst_dir}${srv_name}/increment/ -maxdepth 1 -type d
-mtime +30 -exec rm -rf {} \;
date
echo "Finish backup ${srv_name}"
```

Rsync

Устанавливаем исполняемость скрипта:

```
chmod 0744 /root/scripts/backup-node1.sh
```

Создаем файл с паролем для авторизации на сервере
источнике:

```
nano /etc/rsyncd.scrpt  
12345  
chmod 0600 /etc/rsyncd.scrpt
```


Rsync

Тестируем */root/scripts/backup-node1.sh*

Если ошибок нет, то заносим скрипт в cron:

```
nano /etc/crontab
```

```
...  
30 23 * * * root /root/script/backup-node1.sh
```



Резервирование важных ресурсов



Резервирование важных ресурсов

В действительности резервировать требуется важные для системы компоненты например: [/etc/](#) [/usr/](#).

Также бэкапятся базы данных, веб-сервисы и все, что является критической информации.

Рассмотрим пример на работающей конфигурации.



Итоги

Итоги

Сегодня мы:

- рассмотрели механизмы резервирования данных;
- настроили два решения bacula и rsync;
- рассмотрели на примере, как настроить синхронизацию;
- вспомнили главную мудрость: делай бэкап, чтобы не было, как в фильме:





Домашнее задание



Домашнее задание

Давайте посмотрим ваше [домашнее задание](#).

- Вопросы по домашней работе задавайте **в чате** мессенджера Slack.
- Задачи можно сдавать **по частям**.
- Зачёт по домашней работе проставляется после того, как **приняты все задачи**.

**Задавайте вопросы и
пишите отзыв о лекции!**

Александр Зубарев