Администратор Linux

Резервное копирование

otus.ru

Проверить, идет ли запись

Меня хорошо видно && слышно?

Ставим “+”, если все хорошо
“-”, если есть проблемы

Тема вебинара

Резервное копирование

Федоров Иван Романович

Технический директор ГК “Инотех”

Опыт:

Более 10 лет в IT-сфере

Аспирант университета ИТМО по направлению “Информационная безопасность”

Многократный победитель различных конкурсов и хакатонов (команда IBI Solutions)

Эл. почта: ifedorov.devops@gmail.com

Правила вебинара

Активно
участвуем

Off-topic обсуждаем
в группе Telegram

Задаем вопрос
в чат или голосом

Вопросы вижу в чате,
могу ответить не сразу

Маршрут вебинара

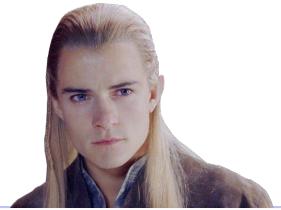
- Знакомство
- Введение и базовые понятия
- Создание резервных копий
- Bacula
- Borg
- Рефлексия

Цели вебинара

К концу занятия вы сможете

1. Подобрать оптимальный для вашей задачи способ резервного копирования

2. Создавать резервные копии с помощью rsync



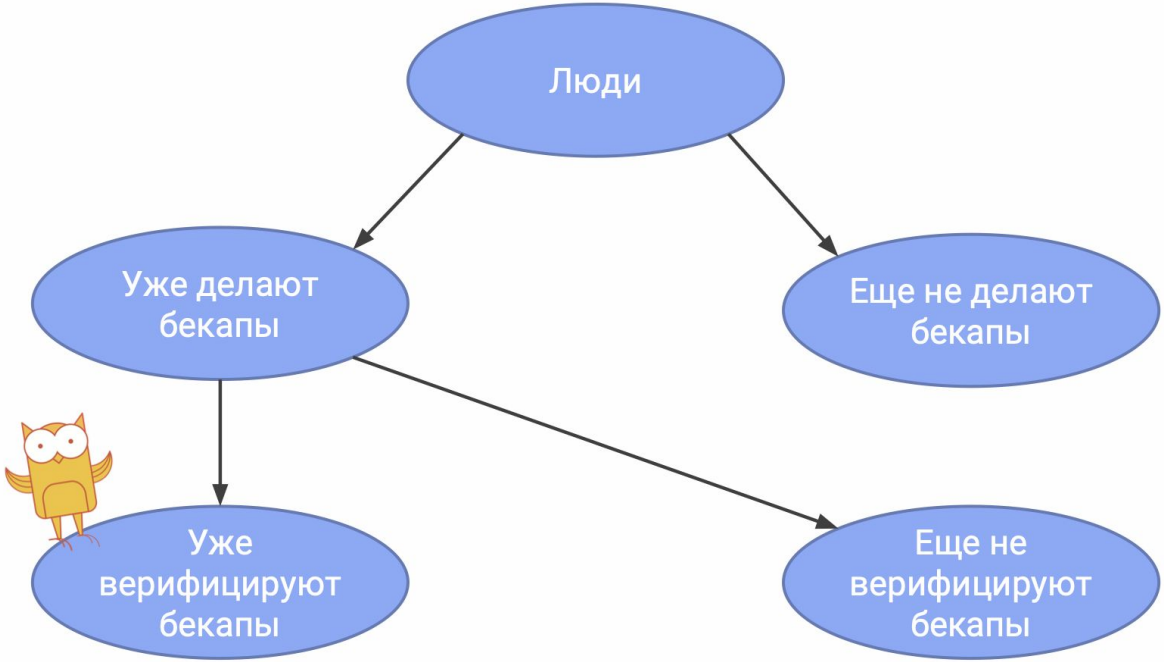
3. Создавать резервные копии с помощью borg

Смысл

Зачем вам это уметь

Введение
и базовые понятия

Сова находится где нужно :)

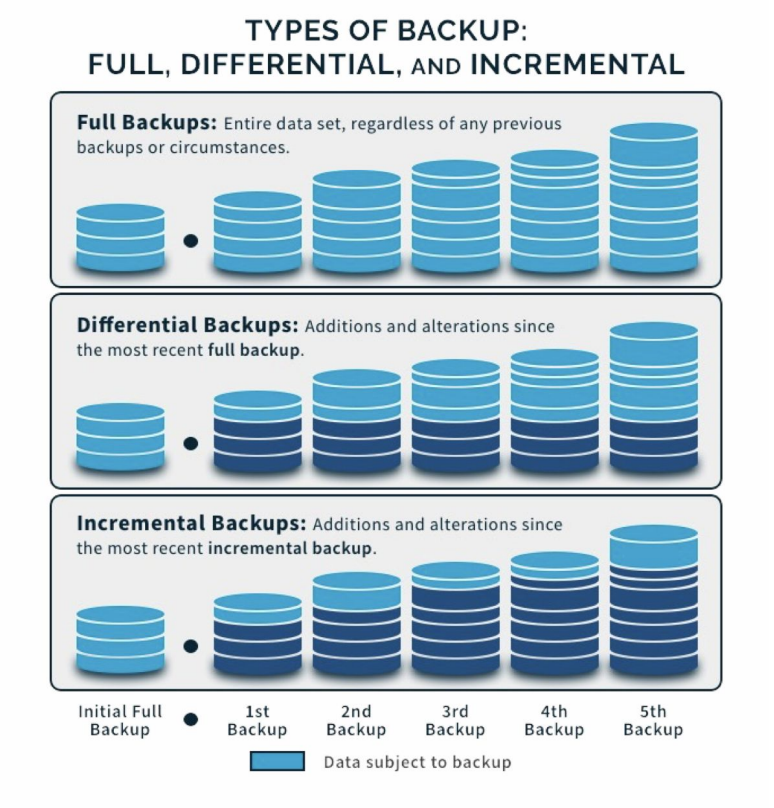


Термины резервного копирования

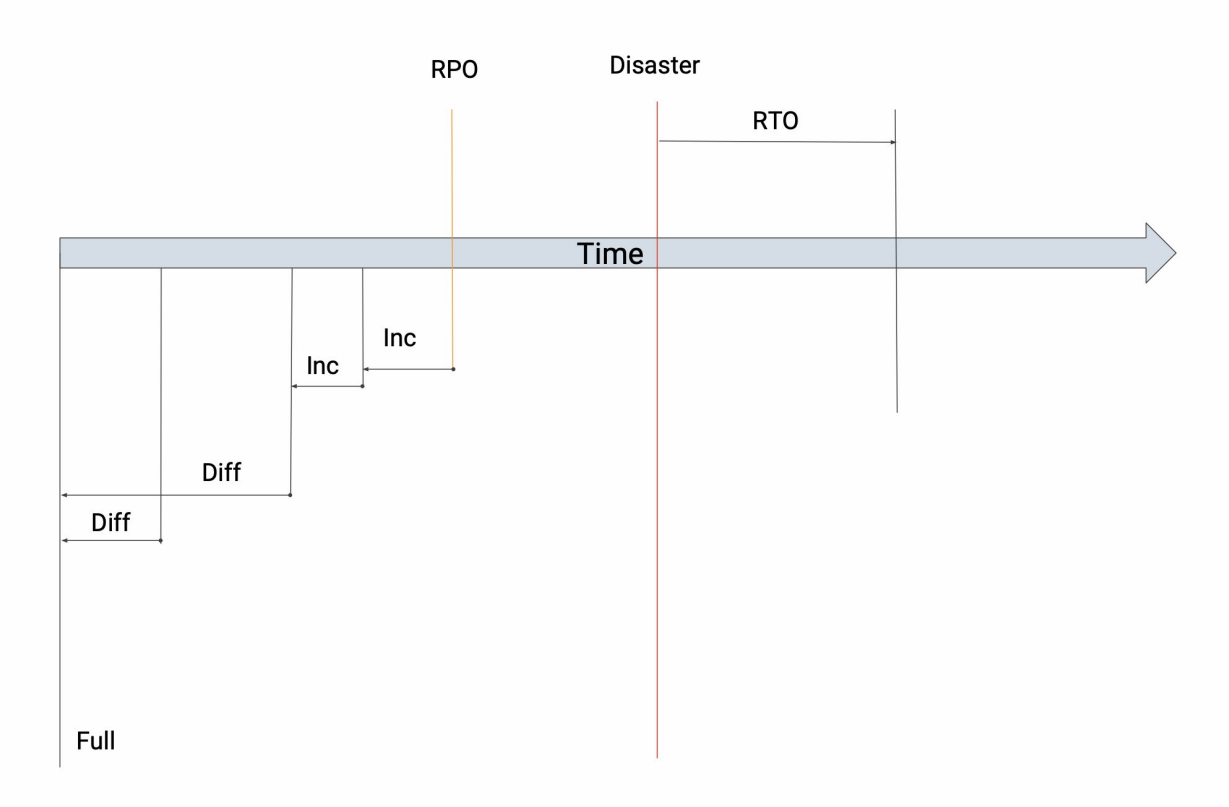
- **RTO (Recovery Time Objective)** — определяет время, требуемое на восстановление из резервной копии. Это не то, что диктуется процессом РК, это то требование, которому процесс должен соответствовать. Например, “восстановление из РК должно занимать не более 1 часа”.
- **RPO (Recovery Point Objective)** — точка во времени (Point in Time), на которую должны быть восстановлены данные. Например, “Данные должны быть восстановлены по состоянии не “дальше”, чем 24 часа с момента сбоя”.
- **Уровень резервного копирования (Backup Level)** — 0-1-2, Full, Differential, Incremental. Различные стратегии выбора данных для копирования.
- **Глубина резервного копирования** — определяет как долго хранятся резервные копии.
- **Стратегия хранения** — определяет “детализация”, с которой можно восстановиться.

Уровни резервного копирования

- **Full** — полное резервное копирование. Для восстановления требуется только эта резервная копия.
- **Differential** — разностное резервное копирование. Копируется только то, что изменилось с последнего резервного копирования. Для восстановления требуется последняя полная и последняя дифференциальная копии.
- **Incremental** — инкрементальное резервное копирование. Копируется только то, что изменилось с последнего прохода резервного копирования. Для восстановления требуется: последняя полная; последняя дифференциальная (если есть); ВСЕ инкрементальные копии с момента последней полной/дифференциальной копии.



Уровни резервного копирования



Требования к системе РК

- Хранение на отдельном носителе или в другом месте
- Надежность места хранения
- Доступность места хранения
- Простота использования

Возможные проблемы

- Размеры резервной копии — негде хранить.
- Время на получение резервной копии
- Время развертывания резервной копии
- Нагрузка на систему для получения резервной копии — резервное копирование либо создает дополнительную нагрузку, либо взводит проблемы перечисленные выше.

Возможные сопутствующие техники

- Репликация/Дублирование
- Снэпшоты (снимки)
- Журналирование
- Зеркалирование

Все эти техники сами по себе НЕ ЯВЛЯЮТСЯ резервными копиями и НЕ ГАРАНТИРУЮТ восстановление данных в случае штатного удаления и/или сложного аппаратного сбоя

Вопросы для планирования РК

- Репликация/Дублирование
- От каких сбоев мы хотим защититься? (Зачем?)
- Что надо копировать?
- Как быстро надо восстанавливать? (RTO)
- На сколько “близко” точка восстановления? (RPO)
- Где все это хранить?
- Сколько это стоит?

Чуть глубже

В зависимости от типа данных/приложения может потребоваться свой собственный подход к резервному копированию.

Например:

- БД — большой объем данных, сложность получения консистентной копии “файловым” копированием. Примеры: mysql, redis
- Объектные хранилища/большие хранилища файлов. Большой объем данных, большое количество объектов, сложность файлового доступа.

Сопутствующие действия

- Проверка целостности копий/проверка хранилища
- Мониторинг:
  - Хранилища
  - Агентов
  - ПО
  - Процесса
- Проверка восстанавливаемости.

Все хорошо?
Вопросы?

Создание резервных копий

Ручное копирование

Ручное создание копий.

Основные проблемы:

- Создание упорядоченного архива (решается `date+format`)
- Удаленное хранение (решается с помощью SSH, NFS)

Инструменты:

- tar
- rsync
- dump
- dd
- rsync + inotify = lsyncd

Ручное копирование (пример 1)

CentOS 7

```
# С помощью утилиты tar (архивирование)
$ tar -czvf nginx_conf.tar.gz /etc/nginx # xvf

# С помощью утилиты dd (побайтовое копирование)
$ dd if=/dev/sdb of=/mnt/backup/sdb.img

# С помощью утилиты rsync
$ rsync -avz --delete /etc/ /mnt/backup/hostname/etc
$ rsync -az -e ssh --delete 192.168.2.1:/etc/nginx/ /etc/nginx
```

Ручное копирование (пример 2)

Скрипт для бэкапа статических сайтов

```
$ vim backup.sh
#!/bin/bash

# Pages directory
PAGESDIR=/var/www/

# Pages backup directory
BACKUPDIR=/mnt/Backup/pages

# Get current weekday number
WDN=$(date +%u)

# Get month number and name
MONNUM=$(date +%m)
MONNAME=$(date +%b)

# Do not run on weekends
[ ${WDN} -eq 6 ] && exit 0
[ ${WDN} -eq 7 ] && exit 0

# Daily dump
for pages in $(ls -l ${PAGESDIR} | grep -e '^d' | awk {'print $9'})
do
  # Sync pages
  rsync -a --delete ${PAGESDIR}/${pages} ${BACKUPDIR}/w.${WDN}/
done

# Monthly archive on first Monday
[ $(date +%-d) -lt 8 ] && [ ${WDN} -eq 1 ] && rsync -a --delete ${BACKUPDIR}/w.${WDN}/ ${BACKUPDIR}/${MONNUM}.${MONNAME}/
```

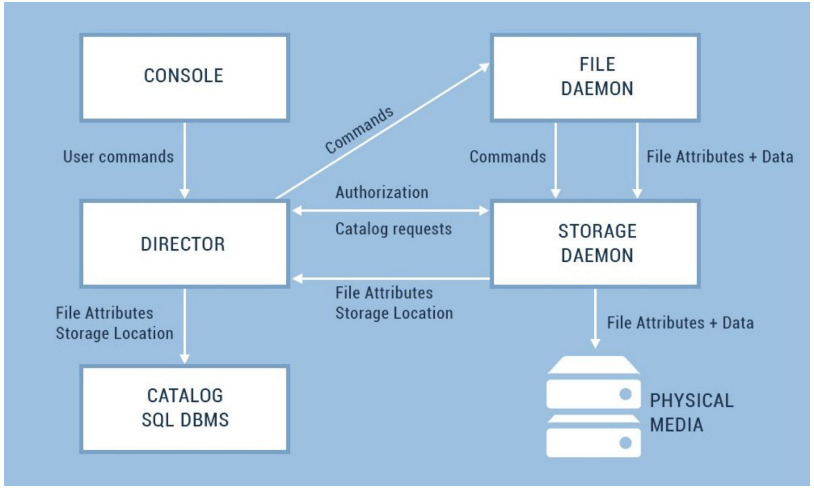
Как настроение?
Есть ли вопросы?

Bacula

Bacula

- Bacula Director
- Bacula File Daemon
- Bacula Storage Daemon
- Bacula console

Обоение между клиентами требует минимальной аутентификации и используется пароли.
Для разных сущностей могут быть разные пароли.



Bacula Director

Центральный и самый важный компонент системы резервного копирования. Работает с каталогом (Catalog) и осуществляет все планирование и постановку задач остальным компонентам системы РК. Поскольку остальные компоненты не наделены “разумом”, то вся конфигурация сосредоточена в director’е.

Сущности:

- FileSet — набор данных для резервного копирования (Что копировать)
- Schedule — расписание резервного (Когда копировать)
- Client — сервер который надо “подвергнуть” резервному копированию (Откуда копировать)
- Pool — описание того как и где хранить резервные копии (Куда копировать)
- Job, JobDefs — описание задачи на резервное копирование, связующее звено Fileset, Schedule, Client, Pool. Описывает что, когда, откуда и куда копировать
- Storage — описание хранилища для бэкапа (Где хранить)
- Catalog - хранилище данных о бэкапах - “сердце” системы

Bacula: минусы

- Schedule — расписание резервного (Когда копировать)
- Нужно генерировать конфиги и клиенту, и директору
- Сложность этих самых конфигов
- На каталогах с большим количеством данных может начаться медленная утечка памяти
- На бэкапах с большим количеством файлов Bacula и Bareos очень сильно зависит от производительности используемой СУБД.
- И вообще зависимость от СУБД

Borg

Borg

- Дедупликация: исключение дублирующих копий повторяющихся данных. Файлы в рамках одного Borg repository (т.е. специальном каталоге в специфичном для Borg формате) делятся на блоки по N мегабайт, а повторяющиеся блоки Borg дедуплицирует.
- Сжатие: после дедупликации данные сжимаются.
- Работает через SSH: отсюда простота и безопасность. На обеих сторонах только и нужно, что поставить Borg
- Прост в установке: PPA, Epel-Release, бинарники
- Гибкая очистка от старых бэкапов

Borg (установка на сервере)

Centos 7

```
# Из репозитория
$ yum install borgbackup

# Можно также просто скачать бинарник
$ wget https://github.com/borgbackup/borg/releases/download/1.1.6/borg-linux64 -O
/usr/local/bin/borg
$ chmod +x /usr/local/bin/borg

# Создание пользователя borg (на клиенте предварительно нужно сгенерировать SSH-ключи)
$ useradd -m borg
$ su - borg
$ cd ~
$ mkdir .ssh
$ vim .ssh/authorized_keys
command="/usr/bin/borg serve" ssh-rsa ...
$ chmod 700 ~/.ssh
$ chmod 600 ~/.ssh/authorized_keys
```

Borg (установка на клиенте)

Centos 7

```
# Из репозитория
$ yum install borgbackup

# Можно также просто скачать бинарник
$ wget https://github.com/borgbackup/borg/releases/download/1.1.6/borg-linux64 -O
/usr/local/bin/borg
$ chmod +x /usr/local/bin/borg

# Генерация ключей
$ ssh-keygen
```

Borg (создание бэкапа)

Centos 7

```
# На клиенте:
# Инициализация репозитория
$ borg init -e none borg@172.17.0.3:MyBorgRepo

# Создание РК
$ borg create --stats --list borg@172.17.0.3:MyBorgRepo::"MyFirstBackup-{now:%Y-%m-%d_%H:%M:%S}"
/etc /root


# На сервере
# Просмотр репозитория
$ borg list MyBorgRepo/
# Просмотр бэкапа
$ borg list MyBorgRepo::MyFirstBackup-2018-08-04_16:55:53
# Извлечение из бэкапа
$ borg extract MyBorgRepo::MyFirstBackup-2018-08-04_16:55:53 etc/hostname
```

Вопросы?

Ставим “+”,
если вопросы есть

Ставим “–”,
если вопросов нет

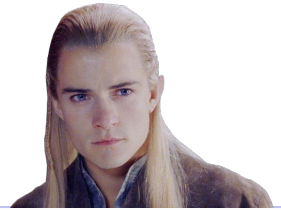
Рефлексия

Цели вебинара

К концу занятия вы сможете

1. Подобрать оптимальный для вашей задачи способ резервного копирования

2. Создавать резервные копии с помощью rsync



3. Создавать резервные копии с помощью borg

Рефлексия

С какими впечатлениями уходите с вебинара?

Как будете применять на практике то,
что узнали на вебинаре?

Заполните, пожалуйста, опрос о занятии по ссылке в чате