



Методическое пособие по выполнению домашнего задания по курсу
«Администратор Linux. Professional»

Домашнее задание Резервное копирование

Цель домашнего задания

Научиться настраивать резервное копирование с помощью утилиты Borg

Описание домашнего задания

1. Настроить стенд Vagrant с двумя виртуальными машинами: backup_server и client. (Студент самостоятельно настраивает Vagrant)
2. Настроить удаленный бэкап каталога /etc с сервера client при помощи borgbackup. Резервные копии должны соответствовать следующим критериям:
 - a. директория для резервных копий /var/backup. Это должна быть отдельная точка монтирования. В данном случае для демонстрации размер не принципиален, достаточно будет и 2GB; (Студент самостоятельно настраивает)
 - b. репозиторий для резервных копий должен быть зашифрован ключом или паролем – на усмотрение студента;
 - c. имя бэкапа должно содержать информацию о времени снятия бэкапа;
 - d. глубина бэкапа должна быть год, хранить можно по последней копии на конец месяца, кроме последних трех. Последние три месяца должны содержать копии на каждый день. Т.е. должна быть правильно настроена политика удаления старых бэкапов;
 - e. резервная копия снимается каждые 5 минут. Такой частый запуск в целях демонстрации;
 - f. написан скрипт для снятия резервных копий. Скрипт запускается из соответствующей Cron джобы, либо systemd timer-a – на усмотрение студента;
 - g. настроено логирование процесса бэкапа. Для упрощения можно весь вывод перенаправлять в logger с соответствующим тегом. Если настроите не в syslog, то обязательна ротация логов.

Формат сдачи:

Vagrantfile + ansible

Критерии оценивания

Статус «Принято» ставится при выполнении следующих условий:

1. Ссылка на репозиторий GitHub.
2. Vagrantfile, который будет разворачивать виртуальные машины
3. Настройка виртуальных машин происходит с помощью Ansible.
4. Документация по каждому заданию:

Создайте файл README.md и снабдите его следующей информацией:

- название выполняемого задания;
- текст задания;
- схема сети;
- особенности проектирования и реализации решения,
- заметки, если считаете, что имеет смысл их зафиксировать в репозитории.

Функциональные и нефункциональные требования

- ПК на Unix с 8 ГБ ОЗУ или виртуальная машина с включенной Nested Virtualization.

Предварительно установленное и настроенное следующее ПО:

- Hashicorp Vagrant (<https://www.vagrantup.com/downloads>)
- Oracle VirtualBox (https://www.virtualbox.org/wiki/Linux_Downloads).
- Ansible (версия 2.8 и выше) - https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html
- Любой редактор кода, например Visual Studio Code, Atom и т.д.

Инструкция по выполнению домашнего задания

Все дальнейшие действия были проверены при использовании Vagrant 2.2.19, VirtualBox v6.1.26 r145957. В качестве ОС на хостах установлена Ubuntu 22.04. Серьезные отступления от этой конфигурации могут потребовать адаптации с вашей стороны.

Тестовый стенд:

```
backup 192.168.11.160 Ubuntu 22.04
client 192.168.11.150 Ubuntu 22.04
```

Устанавливаем на client и backup сервере borgbackup

```
apt install borgbackup
```

На сервере backup создаем пользователя и каталог /var/backup (в домашнем задании нужно будет создать диск ~2Gb и примонтировать его) и назначаем на него права пользователя borg

```
mkdir /var/backup
chown borg:borg /var/backup/
```

На сервер backup создаем каталог ~/.ssh/authorized_keys в каталоге /home/borg

```
su - borg
mkdir .ssh
touch .ssh/authorized_keys
chmod 700 .ssh
chmod 600 .ssh/authorized_keys
```

На сервере client:

```
ssh-keygen
```

Все дальнейшие действия будут проходить на client сервере.

Инициализируем репозиторий borg на backup сервере с client сервера:

```
borg init --encryption=repokey borg@192.168.11.160:/var/backup/
```

Запускаем для проверки создания бэкапа

```
borg create --stats --list
borg@192.168.11.160:/var/backup/::"etc-{now:%Y-%m-%d_%H:%M:%S}"
/etc
```

Смотрим, что у нас получилось

```
borg list borg@192.168.11.160:/var/backup/
Enter passphrase for key ssh://borg@192.168.11.160/var/backup:
etc-2021-10-15_23:00:15          Fri, 2021-10-15 23:00:21
[573f7b4071bd2e079957217f397394c336eaf172208755110b311ada735e16d3]
```

Смотрим список файлов

```
borg list
borg@192.168.11.160:/var/backup/::etc-2021-10-15_23:00:15
```

Достаем файл из бекапа

```
borg extract
borg@192.168.11.160:/var/backup/::etc-2021-10-15_23:00:15
etc/hostname
```

#Автоматизируем создание бэкапов с помощью systemd

Создаем сервис и таймер в каталоге /etc/systemd/system/

```
vim /etc/systemd/system/borg-backup.service
```

```
[Unit]
```

```
Description=Borg Backup
```

```
[Service]
```

```
Type=oneshot
```

Парольная фраза

```
Environment="BORG_PASSPHRASE=Otus1234"
```

Репозиторий

```
Environment=REPO=borg@192.168.11.160:/var/backup/
```

Что бэкапим

```
Environment=BACKUP_TARGET=/etc
```

```

# Создание бэкапа
ExecStart=/bin/borg create \
    --stats \
    ${REPO}::etc-{now:%%Y-%%m-%%d_%%H:%%M:%%S} ${BACKUP_TARGET}

# Проверка бэкапа
ExecStart=/bin/borg check ${REPO}

# Очистка старых бэкапов
ExecStart=/bin/borg prune \
    --keep-daily 90 \
    --keep-monthly 12 \
    --keep-yearly 1 \
    ${REPO}

# /etc/systemd/system/borg-backup.timer
[Unit]
Description=Borg Backup

[Timer]
OnUnitActiveSec=5min

[Install]
WantedBy=timers.target

# Включаем и запускаем службу таймера
systemctl enable borg-backup.timer
systemctl start borg-backup.timer

# Проверяем работу таймера
systemctl list-timers --all

```

NEXT PASSED	UNIT	LEFT	LAST ACTIVATES
C6 2021-10-16 11:37:51 UTC	borg-backup.timer	3min 25s left	C6 2021-10-16 11:32:51
UTC 1min 34s ago			borg-backup.service

Полезные источники:

- <https://habr.com/ru/company/flant/blog/420055/>
- https://www.opennet.ru/tips/3180_borg_backup.shtml
- <https://borgbackup.readthedocs.io/en/stable/usage/general.html>
- <https://github.com/alisiano/borg-systemd>
- https://dextervolkman.com/posts/borg_backups

