Homework_Lesson8_OS_7

Задание:

1. RSYNC либо RCLONE. Сделать синхронизацию с облачных хранилищем (S3, GCP Storage) с указанием NAME_SURNAME в имени папки.
2. FDISK + LVM
3.* синхронизировать между собой две папки на двух разных вм-ках
4. *синхронизировать папки на двух вмках и ещё на GCP
Ссылки на GCP bucket:
https://console.cloud.google.com/storage/browser/tms_123121419djscj_test
gs://tms_123121419djscj_test

Выполнение 1-го задания:

RSYNC либо RCLONE.
Сделать синхронизацию с облачных хранилищем (S3, GCP Storage) с указанием NAME_SURNAME в имени папки.

В нашей ситуации используем утилиту RSYNC.

1.1 Скачиваем архив gcloud Linux и устанавливаем:

$ curl -O https://dl.google.com/dl/cloudsdk/channels/rapid/downloads/google-cloud-cli-linux-x86_64.tar.gz

$ tar -xf google-cloud-cli-linux-x86_64.tar.gz

$ ./google-cloud-sdk/install.sh

```
makarov@DESKTOP-UG6J7T7:~$ sudo curl -O https://dl.google.com/dl/cloudsdk/channels/rapid/downloads/google-cloud-cli-linu
x-x86_64.tar.gz
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  125M  100  125M    0     0  7992k      0  0:00:16  0:00:16 --:--:-- 8152k
makarov@DESKTOP-UG6J7T7:~$ ls -l
total 128412
drwxr-xr-x 10 makarov makarov      4096 Nov  6 14:05 Aliaksandr_Makarau_DOS24
-rw-r--r--  1 root    root    131489459 Nov 11 12:29 google-cloud-cli-linux-x86_64.tar.gz
makarov@DESKTOP-UG6J7T7:~$ tar -xf google-cloud-cli-linux-x86_64.tar.gz
makarov@DESKTOP-UG6J7T7:~$ ls -l
total 128416
drwxr-xr-x 10 makarov makarov      4096 Nov  6 14:05 Aliaksandr_Makarau_DOS24
-rw-r--r--  1 root    root    131489459 Nov 11 12:29 google-cloud-cli-linux-x86_64.tar.gz
drwxr-xr-x  9 makarov makarov      4096 Nov 11 12:30 google-cloud-sdk
makarov@DESKTOP-UG6J7T7:~$ ./google-cloud-sdk/install.sh
Welcome to the Google Cloud CLI!

To help improve the quality of this product, we collect anonymized usage data
and anonymized stacktraces when crashes are encountered; additional information
is available at <https://cloud.google.com/sdk/usage-statistics>. This data is
handled in accordance with our privacy policy
<https://cloud.google.com/terms/cloud-privacy-notice>. You may choose to opt in this
collection now (by choosing 'Y' at the below prompt), or at any time in the
future by running the following command:

    gcloud config set disable_usage_reporting false

Do you want to help improve the Google Cloud CLI (y/N)?
```

```
To update your SDK installation to the latest version [500.0.0], run:
  $ gcloud components update


Modify profile to update your $PATH and enable shell command completion?

Do you want to continue (Y/n)?  y

The Google Cloud SDK installer will now prompt you to update an rc file to bring the Google Cloud CLIs into your
environment.

Enter a path to an rc file to update, or leave blank to use [/home/makarov/.bashrc]:
Backing up [/home/makarov/.bashrc] to [/home/makarov/.bashrc.backup].
[/home/makarov/.bashrc] has been updated.

==> Start a new shell for the changes to take effect.


For more information on how to get started, please visit:
  https://cloud.google.com/sdk/docs/quickstarts
```

1.2 Делаем авторизацию в GCP:

$ gcloud auth login

```
makarov@DESKTOP-UG6J7T7:~$ gcloud auth login
Go to the following link in your browser, and complete the sign-in prompts:

    https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=32555940559.apps.googleusercontent.com&redire
ct_uri=https%3A%2F%2Fsdk.cloud.google.com%2Fauthcode.html&scope=openid+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinf
o.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.adm
in+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fsqlservice.login+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A
%2F%2Fwww.googleapis.com%2Fauth%2Faccounts.reauth&state=AVk9OntRChWkJjm4OjbN0tWHaxVQwf&prompt=consent&token_usage=remote
&access_type=offline&code_challenge=5ayUGobYVJO4v9YRwlckrY04Unx12dj6uBYSr3n0PbQ&code_challenge_method=S256

Once finished, enter the verification code provided in your browser: 4/0AeanS0bHQG9TdzWAUqg0MCokzvEDXKOrKPpN_58Rf2xYu274
XLTow37tLm4MJgtzK2JJFA

You are now logged in as [sashamkr1995@gmail.com].
Your current project is [None].  You can change this setting by running:
  $ gcloud config set project PROJECT_ID
makarov@DESKTOP-UG6J7T7:~$
```

1.3 Делаем установку rsync:

$ apt-get install rsync

```
root@DESKTOP-UG6J7T7:~# apt-get install rsync
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  python3 python3-braceexpand
The following NEW packages will be installed:
  rsync
0 upgraded, 1 newly installed, 0 to remove and 31 not upgraded.
Need to get 417 kB of archives.
After this operation, 795 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian bookworm/main amd64 rsync amd64 3.2.7-1 [417 kB]
Fetched 417 kB in 0s (2,570 kB/s)
Selecting previously unselected package rsync.
(Reading database ... 19923 files and directories currently installed.)
Preparing to unpack .../rsync_3.2.7-1_amd64.deb ...
Unpacking rsync (3.2.7-1) ...
Setting up rsync (3.2.7-1) ...
invoke-rc.d: could not determine current runlevel
root@DESKTOP-UG6J7T7:~#
```

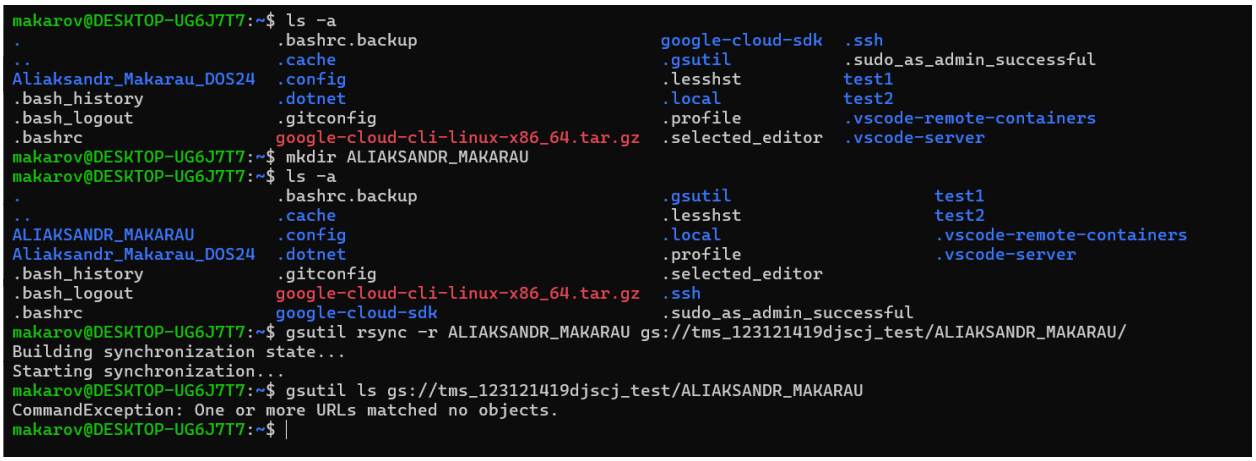
1.4 Начинаем синхронизацию с облачным хранилищем GCP Storage. Создаем локальную папку для синхронизации с названием NAME_SURNAME. В папке создаем файл типа test_alex.sh

$ mkdir ALIAKSANDR_MAKARAU

$ touch test_alex.sh

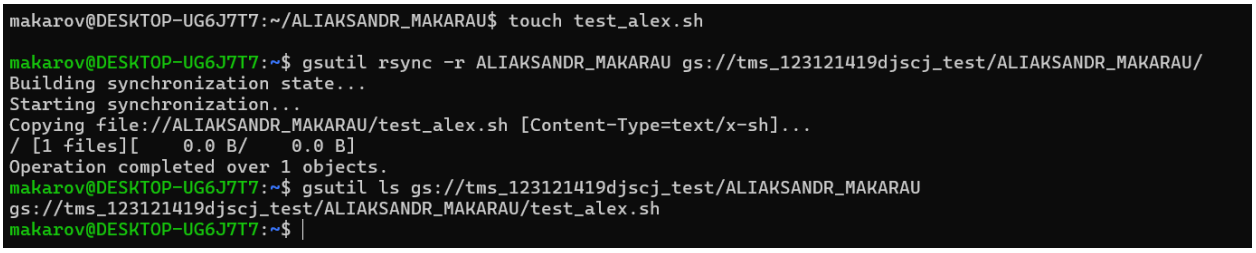
Синхронизируем файлы с бакетом:

$ gsutil rsync -r ALIAKSANDR_MAKARAU gs://tms_123121419djscj_test / ALIAKSANDR_MAKARAU/



Проверяем, что файлы успешно синхронизировались:

$ gsutil rsync ls ALIAKSANDR_MAKARAU gs://tms_123121419djscj_test / ALIAKSANDR_MAKARAU/



Синхронизация с облачным хранилищем GCP Storage завершена.

Выполнение 2-го задания:

FDISK + LVM

2.1 В нашей ВМ уже есть не занятый диск /dev/sdb, к которому относится раздел /u01 - пустой. Допустим, нам нужно добавить (новый созданный) ему vdi_диск на 5GB (/dev/sdc) и обьединить в общую группу тома vg01 и логический том lv01.

```
Disk identifier: A6CAC559-B34A-42D1-AA21-BD236056557D

Device        Start      End  Sectors  Size Type
/dev/sda1      2048  3999743  3997696  1,9G Linux swap
/dev/sda2   3999744 43999231 39999488 19,1G Linux filesystem
/dev/sda3  43999232 67106815 23107584   11G Linux filesystem


Disk /dev/sdb: 10 GiB, 10737418240 bytes, 20971520 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 0D426D42-442B-41DB-9577-A8D1ED2819E6

Device     Start      End  Sectors Size Type
/dev/sdb1   2048 20969471 20967424  10G Linux filesystem


Disk /dev/sdc: 5 GiB, 5368709120 bytes, 10485760 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/loop8: 12,93 MiB, 13553664 bytes, 26472 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

Т.к. диск /dev/sdb уже был разделен на /u01 (ext4), нам нужно с данного диска через fdisk удалить данный раздел. Вводим fdisk /dev/sdb и в интерактивном режиме (fdisk) даем команду d (удаление раздела).

После пробуем инициализировать диск /dev/sdb для его использования в LVM, получаем ошибку “Cannot use /dev/sdb: device is partitioned”. Выяснилось, что почему-то фаловые системы с диска не удалились. Оказалось, что на каждом разделе или диске, есть подпись и метаданные строки, которые мешают добавить данный диск в LVM. Есть отдельная утилита: wipefs -a /dev/sdb, которая также полноценно не отрабатывалась (Device or resource busy). Прочитавши ключи для данной утилиты, есть ключ -f, которая рекурсивно все стирает с диска:

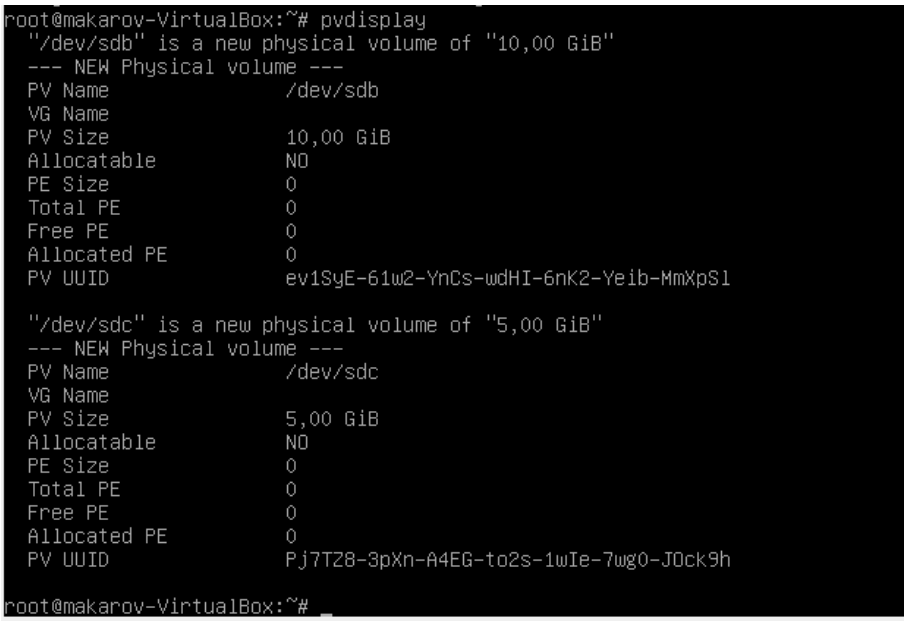
wipefs -af /dev/sdb

```
root@makarov-VirtualBox:~# fdisk -l /dev/sdb
Disk /dev/sdb: 10 GiB, 10737418240 bytes, 20971520 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 0D426D42-442B-41DB-9577-A8D1ED2819E6
root@makarov-VirtualBox:~# wipefs -a /dev/sdb
wipefs: error: /dev/sdb: probing initialization failed: Device or resource busy
root@makarov-VirtualBox:~# wipefs -af /dev/sdb
/dev/sdb: 8 bytes were erased at offset 0x00000200 (gpt): 45 46 49 20 50 41 52 54
/dev/sdb: 8 bytes were erased at offset 0x27fffe00 (gpt): 45 46 49 20 50 41 52 54
/dev/sdb: 2 bytes were erased at offset 0x000001fe (PMBR): 55 aa
root@makarov-VirtualBox:~#
```

2.2 Добавляем наши диски /dev/sdb и /dev/sdc в LVM.

$ pvcreate /dev/sdb
$ pvcreate /dev/sdc

Команда pvdisplay позволяет нам посмотреть, что наши диски могут использоваться в LVM:

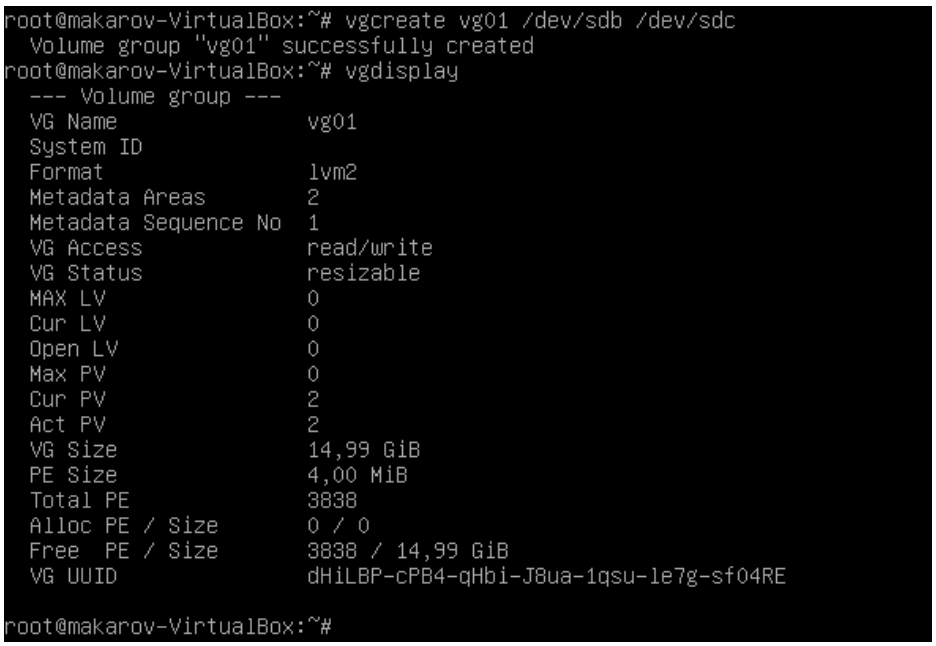


2.3 Создаем группу томов

$ vgcreate vg01 /dev/sdb /dev/sdc

vg01 - имя создаваемой группы

$ vgdisplay - дает информацию о созданных группах



2.4 Создаем логические тома.

В нашем случае используем все свободное пространство группы vg01 при создании логического тома.

lvcreate -l 100%FREE -n lv01 vg01

lv01 - название логического тома.

```
root@makarov-VirtualBox:~# lvcreate -l 100%FREE -n lv01 vg01
WARNING: ext4 signature detected on /dev/vg01/lv01 at offset 1080. Wipe it? [y/n]: y
  Wiping ext4 signature on /dev/vg01/lv01.
  Logical volume "lv01" created.
root@makarov-VirtualBox:~# lvdisplay
  --- Logical volume ---
  LV Path                /dev/vg01/lv01
  LV Name                lv01
  VG Name                vg01
  LV UUID                QMhkjP-RsJ7-FLvU-CpBT-FTQJ-Owyg-CE6QyL
  LV Write Access        read/write
  LV Creation host, time makarov-VirtualBox, 2024-11-12 05:06:02 +0300
  LV Status              available
  # open                 0
  LV Size                14,99 GiB
  Current LE             3838
  Segments               2
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           252:0

root@makarov-VirtualBox:~# _
```

Вводим команду lvdisplay, которая дает информацию о созданном томе:

```
root@makarov-VirtualBox:~# mkfs.ext4 /dev/vg01/lv01
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 3930112 4k blocks and 983040 inodes
Filesystem UUID: bfc8cef4-0f42-4e94-9702-30f64648686b
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

root@makarov-VirtualBox:~# mount /dev/vg01/lv01 /mnt
root@makarov-VirtualBox:~# _
```

2.5 Создаем файловую систему и монтируем том.
Для создания файловой системы ext4 вводим:

$ mkfs.ext4 /dev/vg01/lv01

Монтируем данный созданный том:

$ mount /dev/vg01/lv01 /mnt

Наш созданный том монтируется в директорию /mnt.

Командой df -hT мы просматриваем, примонтирован ли наш диск.

```
root@makarov-VirtualBox:/mnt# df -hT
Filesystem           Type   Size  Used Avail Use% Mounted on
tmpfs                tmpfs  197M  964K  196M   1% /run
/dev/sda2            ext4    19G  9,7G  8,0G  55% /
tmpfs                tmpfs  985M     0  985M   0% /dev/shm
tmpfs                tmpfs  5,0M  4,0K  5,0M   1% /run/lock
/dev/sda3            ext4    11G   15M   11G   1% /home
/dev/mapper/vg01-lv01 ext4    15G   24K   14G   1% /mnt
root@makarov-VirtualBox:/mnt# _
```

Для того, чтобы постоянно вручную не монтировать диски, прописываем данный диск в /etc/fstab.

```
root@makarov-VirtualBox:/mnt# cat /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
# / was on /dev/sda2 during installation
UUID=a93ab16c-653a-4af7-addc-a2e0acbf707f /               ext4    errors=remount-ro 0       1
# /home was on /dev/sda3 during installation
UUID=52812ac6-d487-4e7e-addb-ffe964c2082c /home           ext4    defaults        0       2
# /u01 was on /dev/sdb1 during installation
UUID=3a5d52e1-fb9e-4c4c-b3b5-77f1964b22d2 /u01            ext4    defaults        0       2
# swap was on /dev/sda1 during installation
UUID=653802ae-47b3-4d27-98ec-abb6a8d10bcc none            swap    sw              0       0

/dev/vg01/lv01                                            /mnt    ext4  defaults        1 2
root@makarov-VirtualBox:/mnt# _
```

Выполнение 3-го задания.

3.* синхронизировать между собой две папки на двух разных вм-ках

С предыдущих ДЗ у нас есть две ВМ vm1server (192.168.56.102) и vm2server (192.168.56.101).

Для начала нужно пробросить между ВМ SSH-ключи.

```
makarov1@vm1server:~$ ssh-copy-id makarov2@192.168.56.101
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/makarov1/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
makarov2@192.168.56.101's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'makarov2@192.168.56.101'"
and check to make sure that only the key(s) you wanted were added.

makarov1@vm1server:~$ ssh makarov2@192.168.56.101
Welcome to Ubuntu 24.10 (GNU/Linux 6.11.0-9-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Tue Nov 12 03:06:27 PM UTC 2024

  System load:    0.0             Processes:             104
  Usage of /home: 0.0% of 5.82GB  Users logged in:       0
  Memory usage:   7%              IPv4 address for enp0s3: 10.0.2.15
  Swap usage:     0%


1 update can be applied immediately.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Wed Oct 23 09:12:31 2024 from 192.168.56.1
makarov2@vm2server:~$ exit
logout
Connection to 192.168.56.101 closed.
```

vm1server - Пробрасываем ssh-ключи на vm2server

vm2server - Пробрасываем ключи на vm1server

Проверить установлен ли у нас rsync на двух ВМ:

$ apt-get install rsync

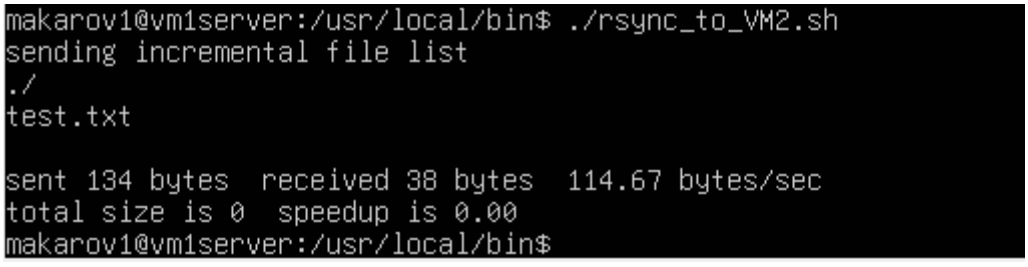
Можно попробовать автоматизировать синхронизацию rsync между двумя ВМ с помощью systemd. Для этого необходимо создать bash-скрипт и положить его в /usr/local/bin/*.sh. Назавем его rsync_to_VM2.sh и внесем скрипт:

#!/bin/bash

VM1_DIR=”/home/makarov1/rsync/”
VM2_DIR=”makarov2@192.168.56.101:/home/makarov2/rsync/”

rsync -av --delete “$VM1_DIR” “$VM2_DIR”

Пробуем запустить из под пользователя makarov1 созданный bash-скрипт. Для начала создал на двух ВМ директории: ”/home/makarov1/rsync/” и “/home/makarov2/rsync/”.



На ВМ vm2server появился данный файл.

Создаем файл сервиса. Все файлы сервиса находятся в /etc/systemd/system/. Назовем его rsync_to_VM2.service.

В нем прописываем:

[Unit]
Description=Myunit rsync

[Service]
User=makarov1
Type=oneshot
ExecStart=/usr/local/bin/rsync_to_VM2.sh

Во время проверки были проблемы с запуском сервиса. В журнале journalctl ошибки были типа Host key verification failed. Проблемы были SSH судя из ошибки. Только с временем понял, что данный сервис запускался из под root. Поэтому в файле rsync_to_VM2.service в секции [Service] указал строку User=makarov1, чтоб данный сервис запускался из под этого пользователя.

Также создаем таймер в той же директории. Даем название типа rsync_to_VM2.timer.

[Unit]
Description=Run rsync service

[Timer]
Unit=rsync_to_VM2.service
OnCalendar=*:0/5

[Install]
WantedBy=timers.target

Пробуем запустить. Данный таймер должен каждые 5 минут запускать rsync_to_VM2.service.

```
makarov1@vm1server:~$ sudo systemctl start rsync_to_VM2.timer
makarov1@vm1server:~$ sudo systemctl status rsync_to_VM2.timer
● rsync_to_VM2.timer - Run rsync service
     Loaded: loaded (/etc/systemd/system/rsync_to_VM2.timer; disabled; preset: enabled)
     Active: active (waiting) since Tue 2024-11-12 16:52:45 UTC; 7s ago
 Invocation: 2e466705a69647a69a3b74225a87cc68
    Trigger: Tue 2024-11-12 16:55:00 UTC; 2min 7s left
   Triggers: ● rsync_to_VM2.service

Nov 12 16:52:45 vm1server systemd[1]: Started rsync_to_VM2.timer - Run rsync service.
makarov1@vm1server:~$ _
```

Также можно проверять статус таймера:

```
makarov1@vm1server:~$ systemctl list-timers
NEXT                        LEFT LAST                        PASSED UNIT                         ACTIVATES
Tue 2024-11-12 17:00:00 UTC 4min 14s Tue 2024-11-12 16:55:01 UTC   44s ago rsync_to_VM2.timer          rsync_to_VM2.service
Tue 2024-11-12 17:00:00 UTC 4min 14s Tue 2024-11-12 16:50:16 UTC  5min ago sysstat-collect.timer       sysstat-collect.service
```

Аналогичные действия нужно проделать, чтоб все изменения и проходили со стороны vm2server.