

Lab BSY FILES II

Introduction & Prerequisites

This laboratory is to learn how to:

- Learn how to create, format and mount volumes.
- Learn how to test the performance of volumes and file systems.

The following resources and tools are required for this laboratory session:

- Any modern web browser,
- Any modern SSH client application
- OpenStack Horizon dashboard: <https://ned.cloudlab.zhaw.ch>
- OpenStack account details: please contact the lab assistant in case you already have not received your access credentials.

Important access credentials:

- Username to login with SSH into VMs in ned.cloudlab.zhaw.ch OpenStack cloud from your laptops
 - **ubuntu**

Time

The entire session will take 90 minutes.

Task 0 – Setup

In order to get started, create a VM instance from the standard image with a flavor size of your choice. Access the VM with SSH using the username `ubuntu`. Once you have SSH'ed into the VM, change your shell to a root shell (`su -i`). This will not require you run 'sudo' in front of every command that requires root/system level privilege¹. Most of the tools should be installed on the VM, missing ones can be installed using "apt".

¹ This is for convenience and not recommended to do on production systems

Task 1 – Uncompressed and compressed btrfs, defragmentation

In this task you will create two volumes in the ned.cloudlab.zhaw.ch OpenStack environment and attach them to an existing instance. Then you will partition the volumes and format the partitions with btrfs and mount them uncompressed and compressed. At the end you will defragment and simultaneously compress the uncompressed file system.

Create two volumes of 8 GB each in the ned.cloudlab.zhaw.ch OpenStack environment. To do this, click on the button “Create Volume” in the Volume section.

Attach the two volumes to an existing running instance. Select therefore the action “Manage Attachments”. Verify that volumes are attached.

Create three directories /data, /data1 and /data2 and a primary partition on the first volume (/dev/vdx) and on the second volume (/dev/vdy). The partitions should cover 25% of the block devices.

How can you be sure that a volume is not already partitioned?

Format the first partition of the first volume /dev/vdx1 with btrfs and mount it on /data1.

Format the second partition of the second volume /dev/vdy1 also with btrfs and mount it on /data2 forcing compression with the zlib algorithm.

Read about compression in <https://wiki.archlinux.org/title/btrfs#Compression>.

Verify the disk usage in /dev/vdx1 and /dev/vdy1 with btrfs and Linux commands. Look at

```
$ man btrfs filesystem
```

```
$ man df
```

```
$ man du
```

Is the disk usage equal ?

List the compression algorithms/methods you can choose from.

Create a few big files in /data1 and /data2.

Verify with the df command, if the file has been compressed in /data2.

Create a big file in /data and copy it to /data1 and /data2

You can verify again with the df and du commands, if the file has been compressed in /data2.

Look at \$ man btrfs filesystem. Defragment /dev/vdx applying the compression algorithm zlib and compare the disk usage of /data1 and /data2.

Task 2 – Btrfs file encryption and automatic volume mount

In this task you will create and automatically mount an encrypted storage device.

Does btrfs support native file encryption?

To encrypt files with the 3rd party tool dm-crypt install btrfs-progs and cryptsetup if needed.

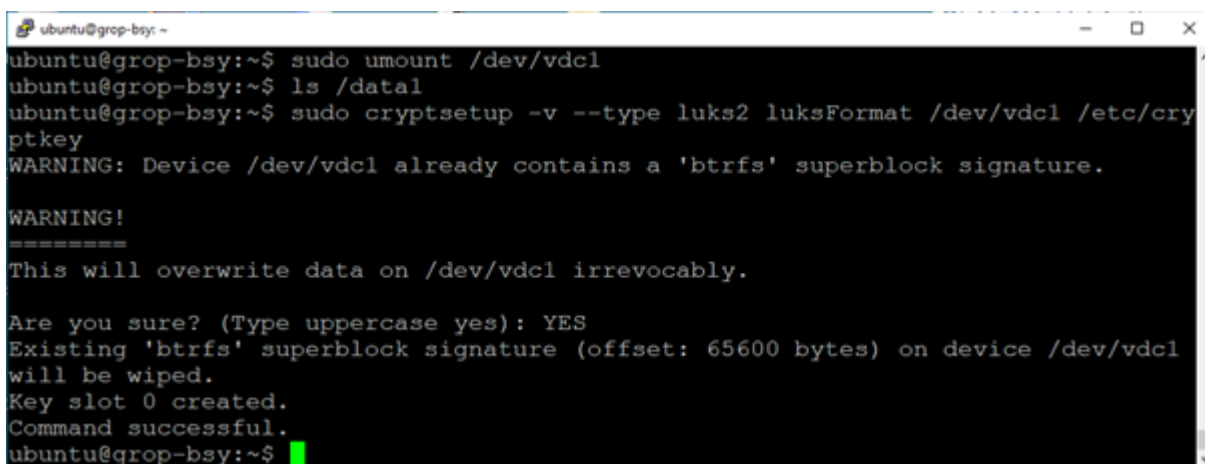
Generate a new 64 bytes long encryption key and store it in the file /etc/cryptkey

```
$ sudo dd if=/dev/urandom of=/etc/cryptkey bs=64 count=1
```

Change the mode of the cryptkey file to 600, so that only the root has r/w access to it

Look also at `$ man cryptsetup` and encrypt the storage device

```
$ sudo cryptsetup -v --type luks2 luksFormat /dev/vdx /etc/cryptkey
```



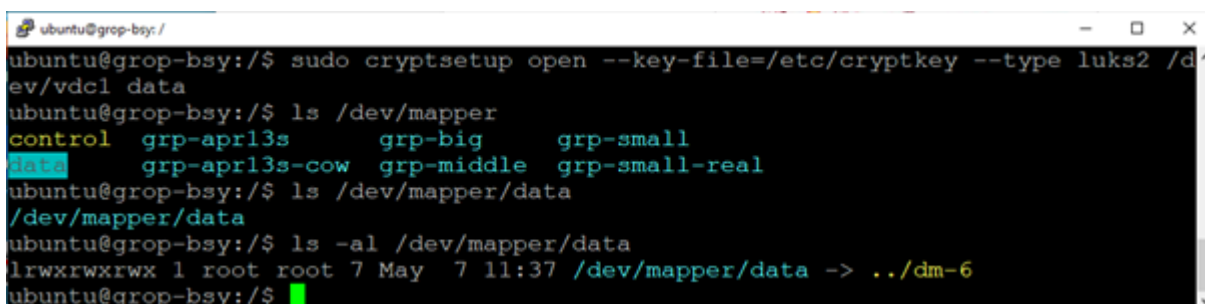
```
ubuntu@grop-bsy: ~  
ubuntu@grop-bsy:~$ sudo umount /dev/vdc1  
ubuntu@grop-bsy:~$ ls /data1  
ubuntu@grop-bsy:~$ sudo cryptsetup -v --type luks2 luksFormat /dev/vdc1 /etc/cryptkey  
WARNING: Device /dev/vdc1 already contains a 'btrfs' superblock signature.  
  
WARNING!  
=====
```

This will overwrite data on /dev/vdc1 irrevocably.

Are you sure? (Type uppercase yes): YES
Existing 'btrfs' superblock signature (offset: 65600 bytes) on device /dev/vdc1 will be wiped.
Key slot 0 created.
Command successful.
ubuntu@grop-bsy:~\$

Open the encrypted storage device vdx and map it to the computer as a data storage device.

```
$ sudo cryptsetup open --key-file=/etc/cryptkey --type luks2 /dev/vdx data
```

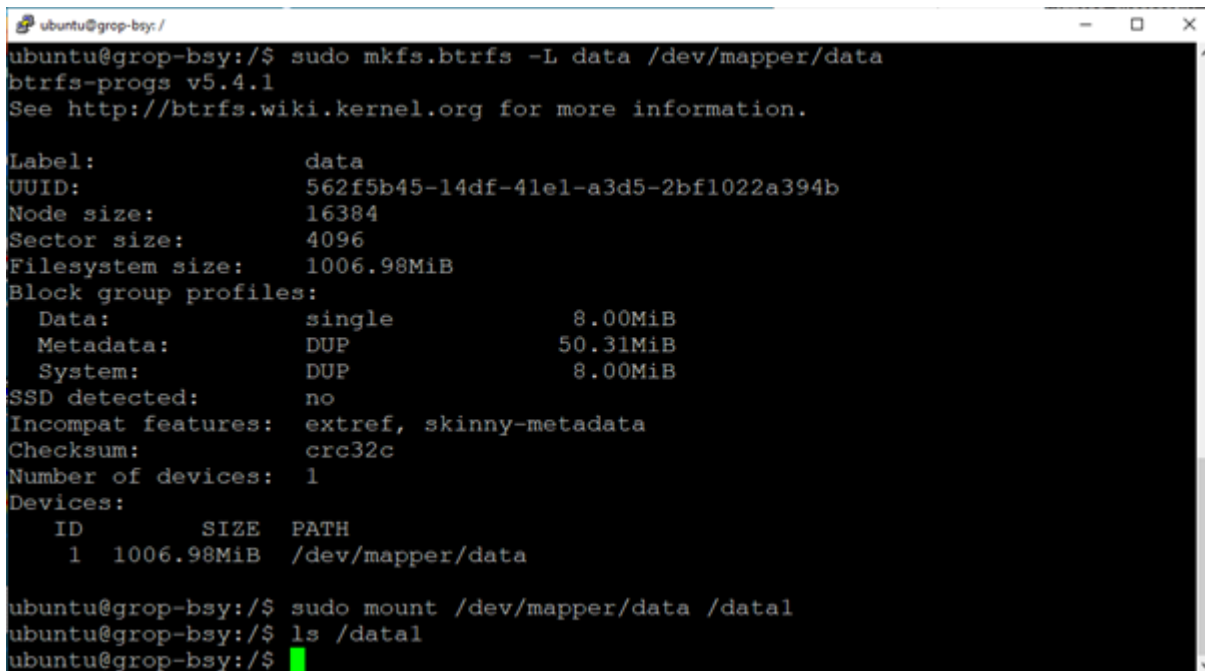


```
ubuntu@grop-bsy: /  
ubuntu@grop-bsy:/$ sudo cryptsetup open --key-file=/etc/cryptkey --type luks2 /dev/vdc1 data  
ubuntu@grop-bsy:/$ ls /dev/mapper  
control  grp-apr13s  grp-big  grp-small  
data     grp-apr13s-cow  grp-middle  grp-small-real  
ubuntu@grop-bsy:/$ ls /dev/mapper/data  
/dev/mapper/data  
ubuntu@grop-bsy:/$ ls -al /dev/mapper/data  
lrwxrwxrwx 1 root root 7 May  7 11:37 /dev/mapper/data -> ../dm-6  
ubuntu@grop-bsy:/$
```

Now, the decrypted storage device will be available in the path `/dev/mapper/data`. The desired btrfs files system has to be created in the `/dev/mapper/data` device and then mounted to the `/dev/mapper/data` device instead of `/dev/vdx` from now on.

```
$ sudo mkfs.btrfs -L data /dev/mapper/data
$ sudo mount /dev/mapper/data /data1
```

As you can see, the btrfs file system created on the encrypted storage device vdx is mounted in the `/data1` directory.

A terminal window titled 'ubuntu@grop-bsy: /' showing the execution of btrfs commands. The first command is 'sudo mkfs.btrfs -L data /dev/mapper/data', which outputs 'btrfs-progs v5.4.1' and a URL. It then displays detailed filesystem information including label, UUID, node and sector sizes, and block group profiles. The second command is 'sudo mount /dev/mapper/data /data1', followed by 'ls /data1' which shows a green cursor.

```
ubuntu@grop-bsy:/$ sudo mkfs.btrfs -L data /dev/mapper/data
btrfs-progs v5.4.1
See http://btrfs.wiki.kernel.org for more information.

Label:                data
UUID:                 562f5b45-14df-41e1-a3d5-2bf1022a394b
Node size:            16384
Sector size:          4096
Filesystem size:      1006.98MiB
Block group profiles:
  Data:               single             8.00MiB
  Metadata:           DUP                50.31MiB
  System:             DUP                8.00MiB
SSD detected:         no
Incompat features:    extref, skinny-metadata
Checksum:             crc32c
Number of devices:    1
Devices:
  ID     SIZE  PATH
  1  1006.98MiB /dev/mapper/data

ubuntu@grop-bsy:/$ sudo mount /dev/mapper/data /data1
ubuntu@grop-bsy:/$ ls /data1
ubuntu@grop-bsy:/$
```

Create the file `HelloWorld.txt` in `/data1` with some readable content.

What happens, if you try to mount `/dev/vdx` directly to another directory?

To automatically mount the encrypted btrfs file system at boot time proceed as follows:

Decrypt the storage device `/dev/vdx` at boot time using the encryption key in `/etc/cryptkey`

Find therefore the UUID of `/dev/vdx` using the `blkid` command

Edit the file `/etc/crypttab` so, that the `vdx` storage device is automatically decrypted at boot time.

```

ubuntu@grop-bsy: /etc
GNU nano 4.8 crypttab Modified
# <target name> <source device> <key file> <options>
data UUID=f138783e-bc2e-49ca-a3f2-790ceaf7922e /etc/cryptkey luks,noearly
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line

```

Mount the decrypted storage device `/dev/mapper/data` to the `/data1` directory.

Find therefore the UUID of the decrypted `/dev/mapper/data` storage device

Edit the `/etc/fstab` file so, that `/dev/mapper/data` is automatically mounted to `/data1`

```

ubuntu@grop-bsy: /etc
GNU nano 4.8 fstab
LABEL=cloudimg-rootfs / ext4 defaults 0 0
LABEL=UEFI /boot/efi vfat defaults 0 0
UUID=6ef3d16d-0ad9-475a-bb28-60920844387e /mnt/data ext4 defaults 0 0
UUID=562f5b45-14df-41e1-a3d5-2bf1022a394b /data1 btrfs defaults 0 0
[ Read 4 lines ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line

```

Reboot the system and verify that the encrypted storage device `vdx` is decrypted, mounted and the `HelloWorld.txt` file is still there.

```

ubuntu@grop-bsy: /data1
vdc      252:32  0    1G  0 disk
└─vdc1   252:33  0   1023M  0 part
└─data   253:6    0   1007M  0 crypt /data1
vdd      252:48  0    1G  0 disk
└─vdd1   252:49  0   1023M  0 part /data2
vde      252:64  0    1G  0 disk
vdf      252:80  0    1G  0 disk
ubuntu@grop-bsy:/data1$ ls -al
total 24
drwxr-xr-x  1 root root   28 May  7 11:59 .
drwxr-xr-x 23 root root 4096 May  7 12:59 ..
-rw-r--r--  1 root root   12 May  7 11:59 HelloWorld.txt
ubuntu@grop-bsy:/data1$

```

Cleanup!

IMPORTANT: At the end of the lab session:

- **Delete** all OpenStack VMs, volumes, security group rules that are no longer needed. You may want to keep some data for exam preparations.

Additional Documentation

OpenStack Horizon documentation can be found on the following pages:

- User Guide: <https://docs.openstack.org/horizon/latest/>