# Homework M1: Local Storage and Additional Techniques

Main goal is to build further on what was demonstrated during the practice Prerequisites may vary between different tasks. You should adjust your infrastructure according to the task you chose to implement Tasks Chose and implement two or more of the following:

* Create a RAID10-based pool in ZFS out of six devices each 5 GB in size. Do it in two different configurations – 3x2 and 2x3
* Create a RAID6-based pool in ZFS out of five devices each 5 GB in size
* Research on how to use a key file to automount an encrypted volume on boot and demonstrate it for *one drive Please note that even if you choose to implement more than one task, they are quite independent and different. So, you may need to create a separate infrastructure (environment) for each or at least clean it every time*

# Proof

Prepare a document that shows what you accomplished and how you did it. It can include (not limited to):

* The commands you used to achieve the above tasks
* A few pictures showing intermediary steps or results

# Solutions

* alma/homework-alma.md
* debian/homework-debian.md
* suse/homework-suse.md

# Debian solution

## Create a RAID10-based pool in ZFS out of six devices each 5 GB in size. Do it in two different configurations – 3x2 and 2x3

### Preparations

1. Add **contrib** repository

$ sudo apt-get install software-properties-common  
$ sudo apt-add-repository contrib

1. Install the ZFS necessary packages

$ sudo apt-get update  
$ sudo apt-get install zfsutils-linux

1. Create folder for mount point

$ sudo mkdir -p /homework/zfs

#### Creating 3x2 Configuration (three mirrored pairs)

Each mirrored pair is created using two devices, and then the three pairs are striped.

$ lsblk  
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS  
sda 8:0 0 20G 0 disk  
├─sda1 8:1 0 19G 0 part /  
├─sda2 8:2 0 1K 0 part  
└─sda5 8:5 0 975M 0 part [SWAP]  
sdb 8:16 0 5G 0 disk  
sdc 8:32 0 5G 0 disk  
sdd 8:48 0 5G 0 disk  
sde 8:64 0 5G 0 disk  
sdf 8:80 0 5G 0 disk  
sdg 8:96 0 5G 0 disk  
sr0 11:0 1 1024M 0 rom

1. Create a mount point

$ sudo mkdir -p /homework/zfs

1. Crate a RAID10 3x2

$ sudo zpool create -m /homework/zfs raid10\_3x2 mirror /dev/sdb /dev/sdc mirror /dev/sdd /dev/sde mirror /dev/sdf /dev/sdg  
  
$ sudo zpool status raid10\_3x2  
 pool: raid10\_3x2  
 state: ONLINE  
config:  
  
 NAME STATE READ WRITE CKSUM  
 raid10\_3x2 ONLINE 0 0 0  
 mirror-0 ONLINE 0 0 0  
 sdb ONLINE 0 0 0  
 sdc ONLINE 0 0 0  
 mirror-1 ONLINE 0 0 0  
 sdd ONLINE 0 0 0  
 sde ONLINE 0 0 0  
 mirror-2 ONLINE 0 0 0  
 sdf ONLINE 0 0 0  
 sdg ONLINE 0 0 0  
  
errors: No known data errors  
  
$ lsblk  
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS  
sda 8:0 0 20G 0 disk  
├─sda1 8:1 0 19G 0 part /  
├─sda2 8:2 0 1K 0 part  
└─sda5 8:5 0 975M 0 part [SWAP]  
sdb 8:16 0 5G 0 disk  
├─sdb1 8:17 0 5G 0 part  
└─sdb9 8:25 0 8M 0 part  
sdc 8:32 0 5G 0 disk  
├─sdc1 8:33 0 5G 0 part  
└─sdc9 8:41 0 8M 0 part  
sdd 8:48 0 5G 0 disk  
├─sdd1 8:49 0 5G 0 part  
└─sdd9 8:57 0 8M 0 part  
sde 8:64 0 5G 0 disk  
├─sde1 8:65 0 5G 0 part  
└─sde9 8:73 0 8M 0 part  
sdf 8:80 0 5G 0 disk  
├─sdf1 8:81 0 5G 0 part  
└─sdf9 8:89 0 8M 0 part  
sdg 8:96 0 5G 0 disk  
├─sdg1 8:97 0 5G 0 part  
└─sdg9 8:105 0 8M 0 part  
sr0 11:0 1 1024M 0 rom

1. Clean up

$ sudo umount /homework/zfs  
$ sudo zpool destroy raid10\_3x2  
$ sudo wipefs --all /dev/sd[b-g]

#### Creating 2x3 Configuration (two groups of three striped devices, mirrored together)

Three devices are striped together in two groups, and these groups are mirrored.

1. Crate a RAID10 2x3

$ sudo zpool create -m /homework/zfs raid10\_2x3 mirror /dev/sdb /dev/sdc /dev/sdd mirror /dev/sde /dev/sdf /dev/sdg  
  
$ sudo zpool status raid10\_2x3  
 pool: raid10\_2x3  
 state: ONLINE  
config:  
  
 NAME STATE READ WRITE CKSUM  
 raid10\_2x3 ONLINE 0 0 0  
 mirror-0 ONLINE 0 0 0  
 sdb ONLINE 0 0 0  
 sdc ONLINE 0 0 0  
 sdd ONLINE 0 0 0  
 mirror-1 ONLINE 0 0 0  
 sde ONLINE 0 0 0  
 sdf ONLINE 0 0 0  
 sdg ONLINE 0 0 0  
  
errors: No known data errors

1. Clean up

$ sudo umount /homework/zfs  
$ sudo zpool destroy raid10\_3x2  
$ sudo wipefs --all /dev/sd[b-g]

## Create a RAID6-based pool in ZFS out of five devices each 5 GB in size

1. Create a mount point

$ sudo mkdir -p /homework/zfs-raid6

1. Create the RAID-Z2 Pool

$ sudo zpool create -m /homework/zfs-raid6 raid6\_pool raidz2 /dev/sdb /dev/sdc /dev/sdd /dev/sde /dev/sdf  
  
$ sudo zpool status raid6\_pool  
 pool: raid6\_pool  
 state: ONLINE  
config:  
  
 NAME STATE READ WRITE CKSUM  
 raid6\_pool ONLINE 0 0 0  
 raidz2-0 ONLINE 0 0 0  
 sdb ONLINE 0 0 0  
 sdc ONLINE 0 0 0  
 sdd ONLINE 0 0 0  
 sde ONLINE 0 0 0  
 sdf ONLINE 0 0 0  
  
errors: No known data errors

1. Perform test

$ sudo touch /homework/zfs-raid6/testfile  
$ ls -l /homework/zfs-raid6/  
total 1  
-rw-r--r-- 1 root root 0 Jan 18 15:11 testfile

1. Clean up

$ sudo umount /homework/zfs-raid6  
$ sudo zpool destroy raid6\_pool  
$ sudo wipefs --all /dev/sd[b-f]

## Research on how to use a key file to automount an encrypted volume on boot and demonstrate it for one drive

$ lsblk  
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS  
sda 8:0 0 20G 0 disk  
├─sda1 8:1 0 19G 0 part /  
├─sda2 8:2 0 1K 0 part  
└─sda5 8:5 0 975M 0 part [SWAP]  
sdb 8:16 0 5G 0 disk  
sdc 8:32 0 5G 0 disk  
sdd 8:48 0 5G 0 disk  
sde 8:64 0 5G 0 disk  
sdf 8:80 0 5G 0 disk  
sdg 8:96 0 5G 0 disk  
sr0 11:0 1 1024M 0 rom

#### Prepare encrypted volume

1. Check is crypt module is available

$ grep -i DM\_CRYPT /boot/config-$(uname -r)  
CONFIG\_DM\_CRYPT=m

1. Load the module

$ sudo modprobe dm\_crypt  
  
$ sudo lsmod | grep dm\_crypt  
dm\_crypt 61440 0  
dm\_mod 184320 1 dm\_crypt

1. Install **cryptsetup** package

$ sudo apt-get install cryptsetup

1. Install parted tool

$ sudo apt update  
$ sudo apt install parted

1. Create partition on sdb drive

$ sudo parted -s /dev/sdb -- mklabel msdos mkpart primary 2048s 5G

1. Initialize LUKS encryption

$ sudo cryptsetup luksFormat /dev/sdb1  
WARNING: Device /dev/sdb1 already contains a 'zfs\_member' superblock signature.  
  
WARNING!  
========  
This will overwrite data on /dev/sdb1 irrevocably.  
  
Are you sure? (Type 'yes' in capital letters): YES  
Enter passphrase for /dev/sdb1:  
Verify passphrase: # enter passcode twice

1. Open the encrypted volume

$ sudo cryptsetup open /dev/sdb1 encrypted\_volume  
Enter passphrase for /dev/sdb1: # enter passcode

1. Format the volume

$ sudo mkfs.ext4 /dev/mapper/encrypted\_volume  
mke2fs 1.47.0 (5-Feb-2023)  
Creating filesystem with 1306368 4k blocks and 327040 inodes  
Filesystem UUID: e2623e6c-934b-4251-90f4-eff4cd19ac41  
Superblock backups stored on blocks:  
 32768, 98304, 163840, 229376, 294912, 819200, 884736  
  
Allocating group tables: done  
Writing inode tables: done  
Creating journal (16384 blocks): done  
Writing superblocks and filesystem accounting information: done

1. Mount the volume

$ sudo mkdir -p /homework/encr-data  
  
$ sudo mount /dev/mapper/encrypted\_volume /homework/encr-data

#### Generate a key file and configure autofs

1. Create a random key file

$ sudo dd if=/dev/urandom of=/root/crypto\_keyfile.bin bs=1024 count=4  
4+0 records in  
4+0 records out  
4096 bytes (4.1 kB, 4.0 KiB) copied, 0.00221607 s, 1.8 MB/s

1. Secure the key file

$ sudo chmod 400 /root/crypto\_keyfile.bin

1. Add the key file to LUKS volume

$ sudo cryptsetup luksAddKey /dev/sdb1 /root/crypto\_keyfile.bin  
Enter any existing passphrase: # enter passcode

1. Install autofs

$ sudo apt install autofs

1. Add line to map file sudo nano /etc/auto.master. The timeout option - how long device stay inactive before unmount.

/homework /etc/auto.encrypted --timeout=60

1. Create the mapping file sudo nano /etc/auto.encrypted

encr-data -fstype=ext4 :/dev/mapper/encrypted\_volume

1. Configure crypttab to use the key file sudo nano /etc/crypttab

encrypted\_volume /dev/sdb1 /root/crypto\_keyfile.bin luks

1. Restart autofs.

$ sudo systemctl restart autofs

#### Test

1. Reboot VM

sudo reboot now

1. Try accessing the directory

Linux debian 6.1.0-30-amd64 #1 SMP PREEMPT\_DYNAMIC Debian 6.1.124-1 (2025-01-12) x86\_64  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/\*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Sat Jan 18 09:18:54 2025 from 10.0.2.2  
vagrant@debian:~$ ls /homework/encr-data  
lost+found

1. Check if it’s mounted

$ mount | grep encrypted\_volume  
/dev/mapper/encrypted\_volume on /homework/encr-data type ext4 (rw,relatime)