Worcester Polytechnic Institute
Department of Computer Science

Lab 2: Windows Acquisition Tools
Module 1

Objectives

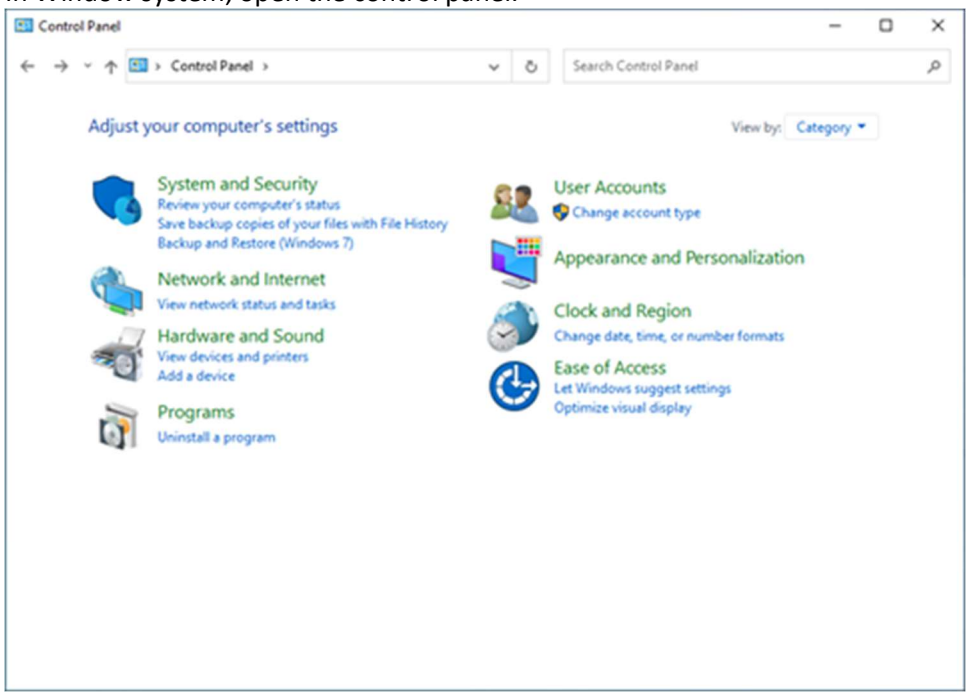
- Create a virtual drive on Windows, and then do disk formatting, and then do acquisitions using a windows tool such as FTK imager.
- Create a virtual drive on Linux, and then do zero out, and then do acquisitions using a Linux tool such as dd.
- Using Autopsy to compare the results and differences.

Software Preparation

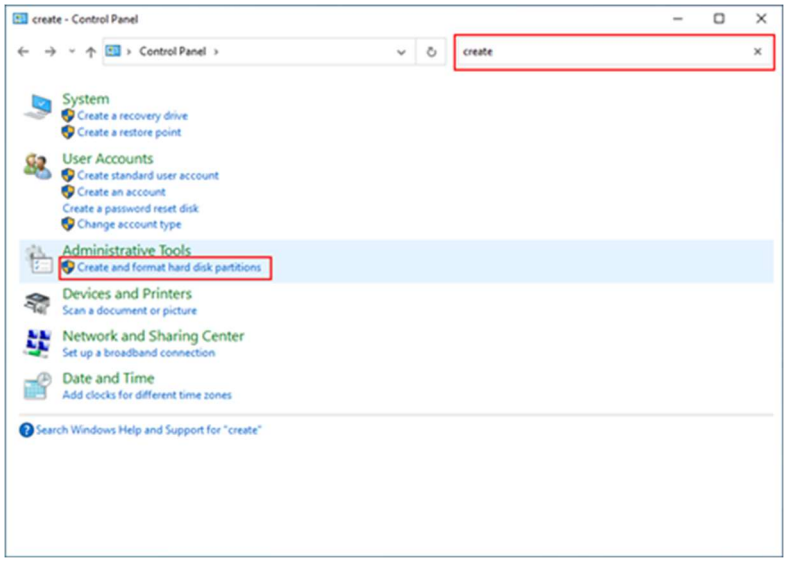
Install the CAINE virtual machine. If you are not using a Windows machine, you also need to install the Windows virtual machine. The download page for CAINE Linux is: https://www.caine-live.net/page5/page5.html
Install the FTK Imager and Autopsy on your machine.
FTK Image download link: https://go.exterro.com/l/43312/2023-05-03/fc4b78
Autopsy download link: https://www.autopsy.com/download/

Part1. Perform data acquisition in Windows using FTK Imager

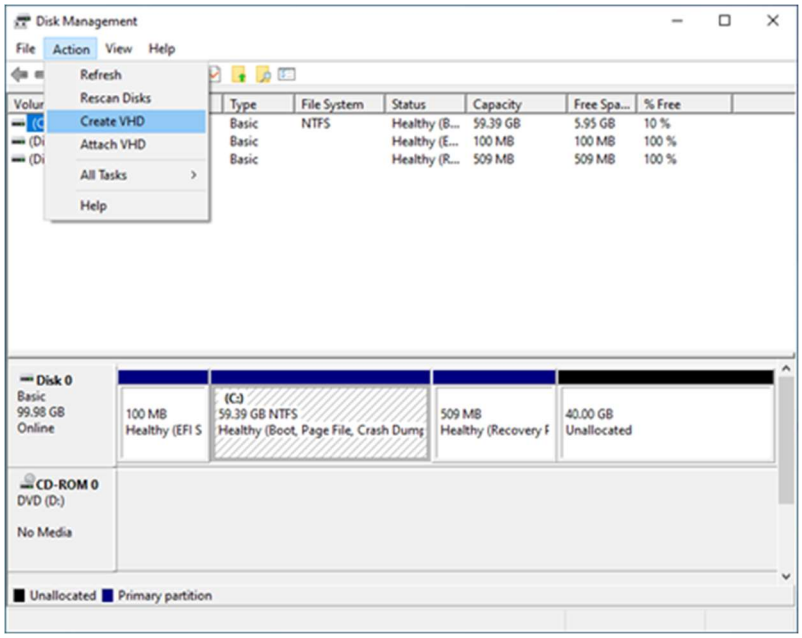
In Window system, open the control panel.

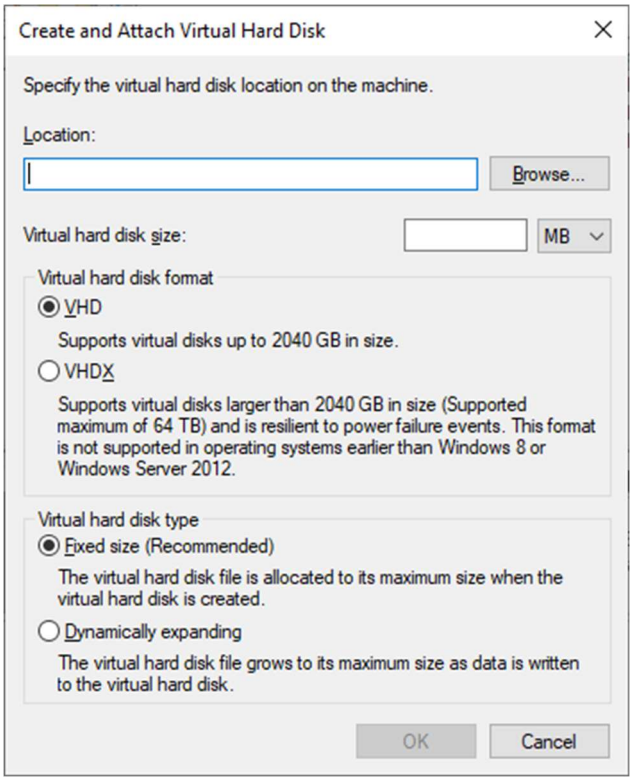


Then, on the top right, search for “create and format hard disk partitions” and open the disk management.

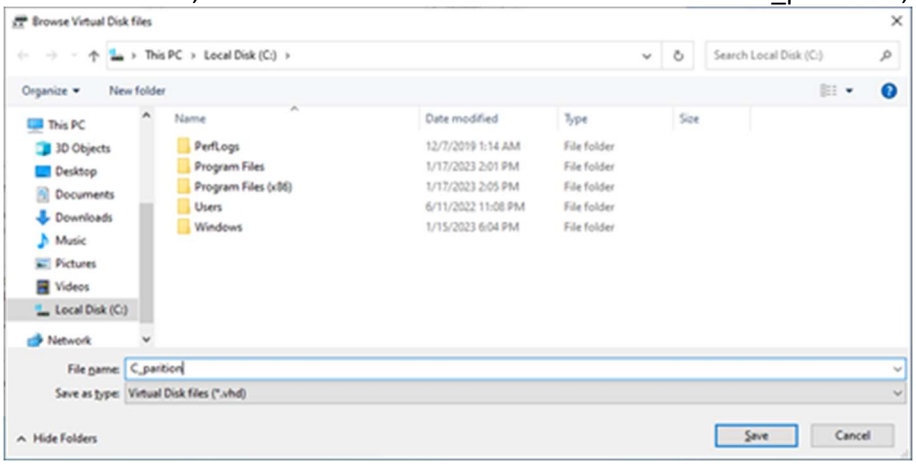


Select “Action”-> “Create VHD”

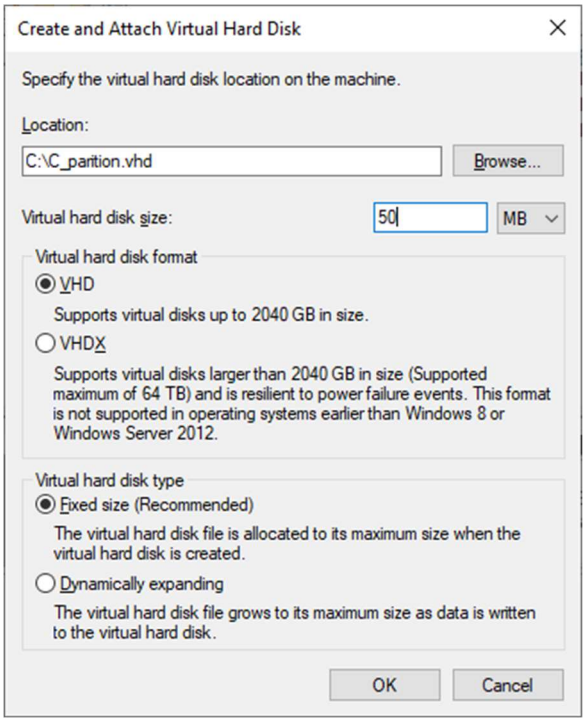




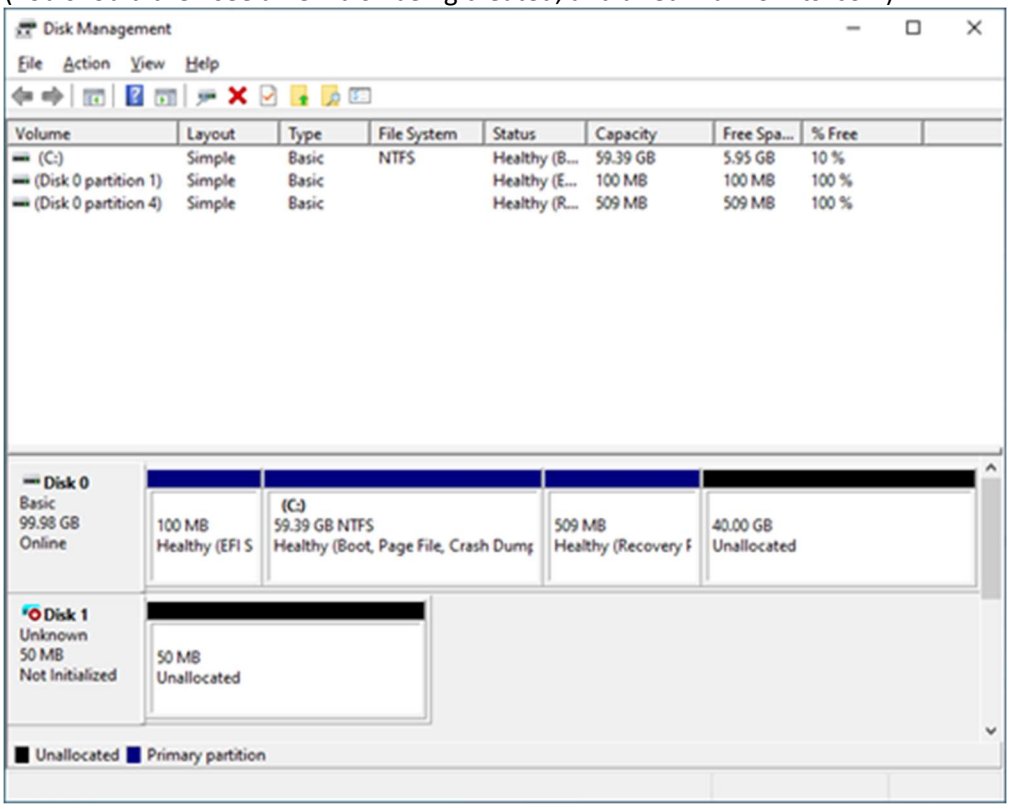
For the location, click on “Browse” and select C: and named it “C_partition”, click save.



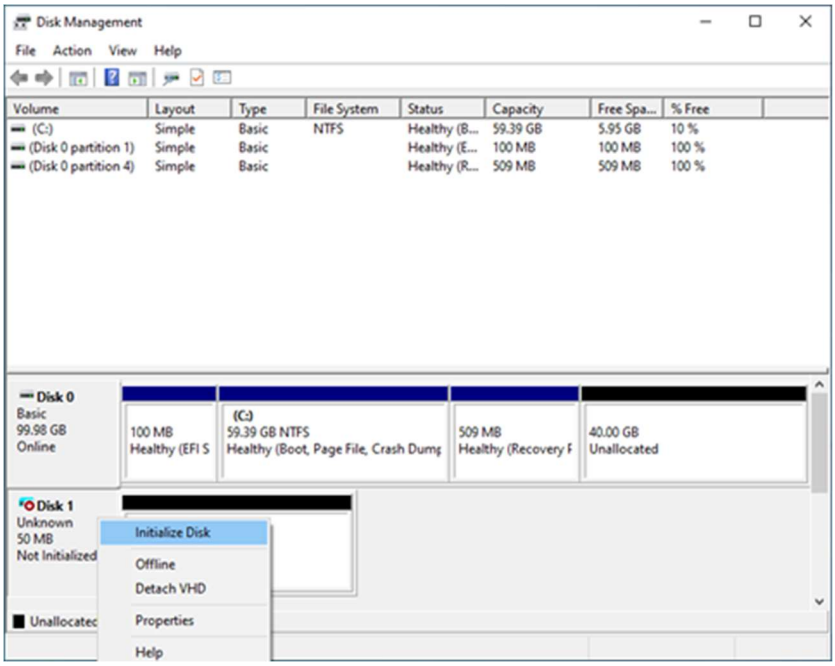
Then, enter 50 MB for the virtual hard disk size and leave the other settings as default, click OK.



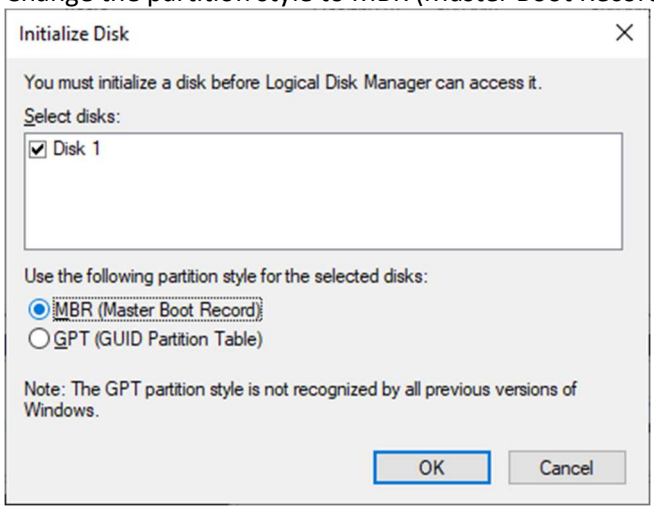
(You should then see a new disk being created, and a red mark on its icon.)



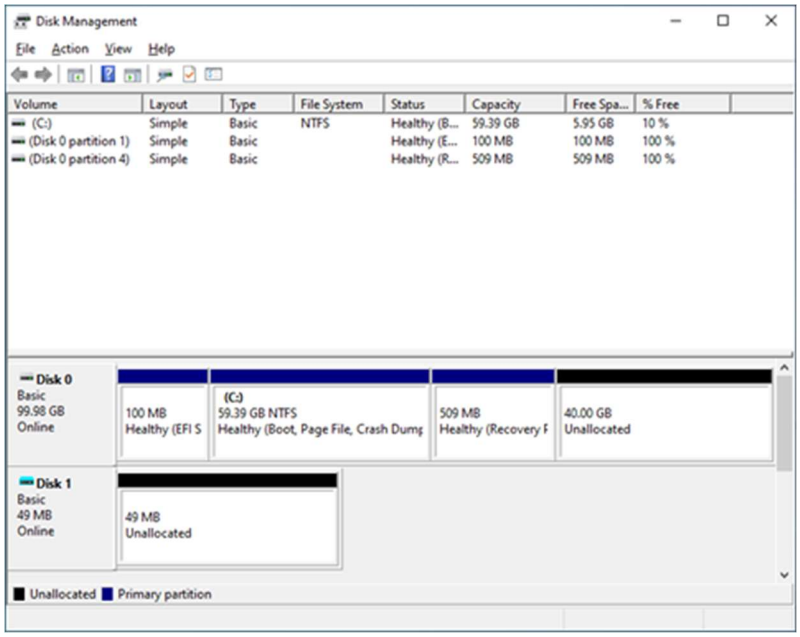
Right click on of disk and select “Initialize Disk”



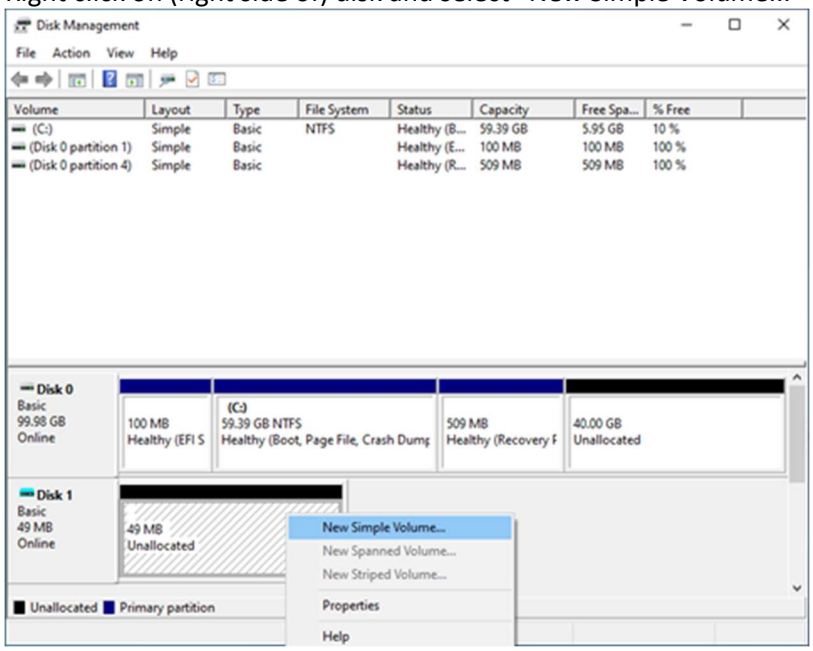
Change the partition style to MBR (Master Boot Record)



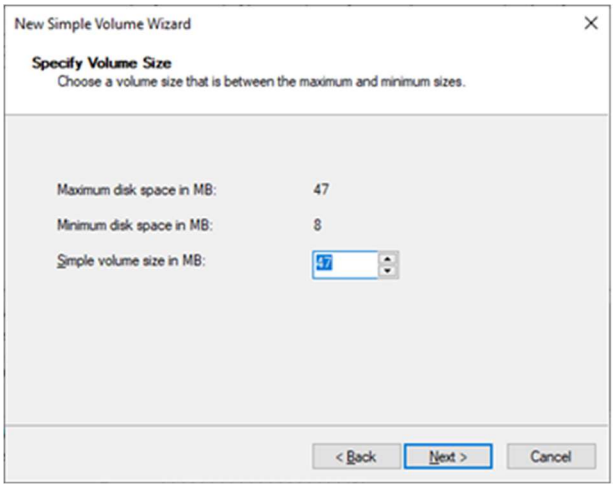
(You should now see the red mark icon is gone.)

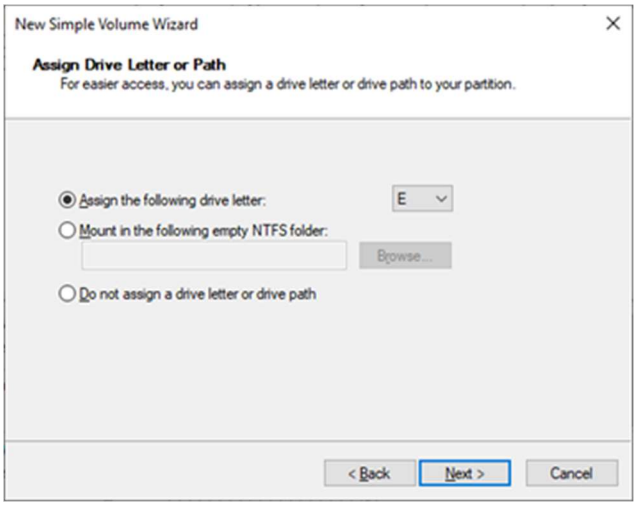


Right click on (right side of) disk and select “New Simple Volume...”

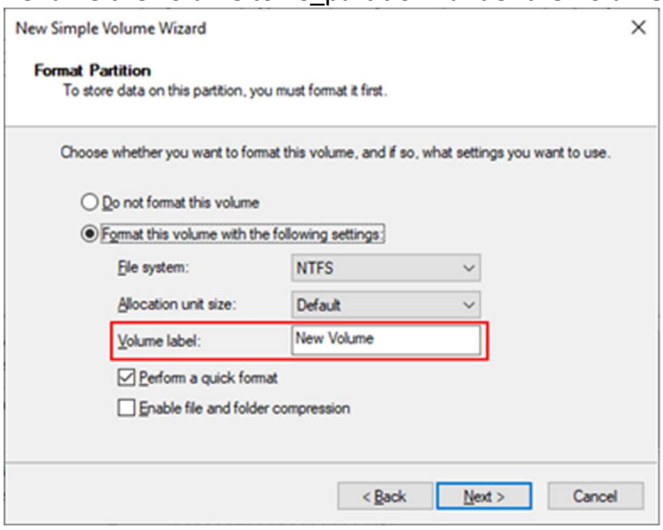


Follow the setup instruction and click “Next”, keep the default setting and click “Next”

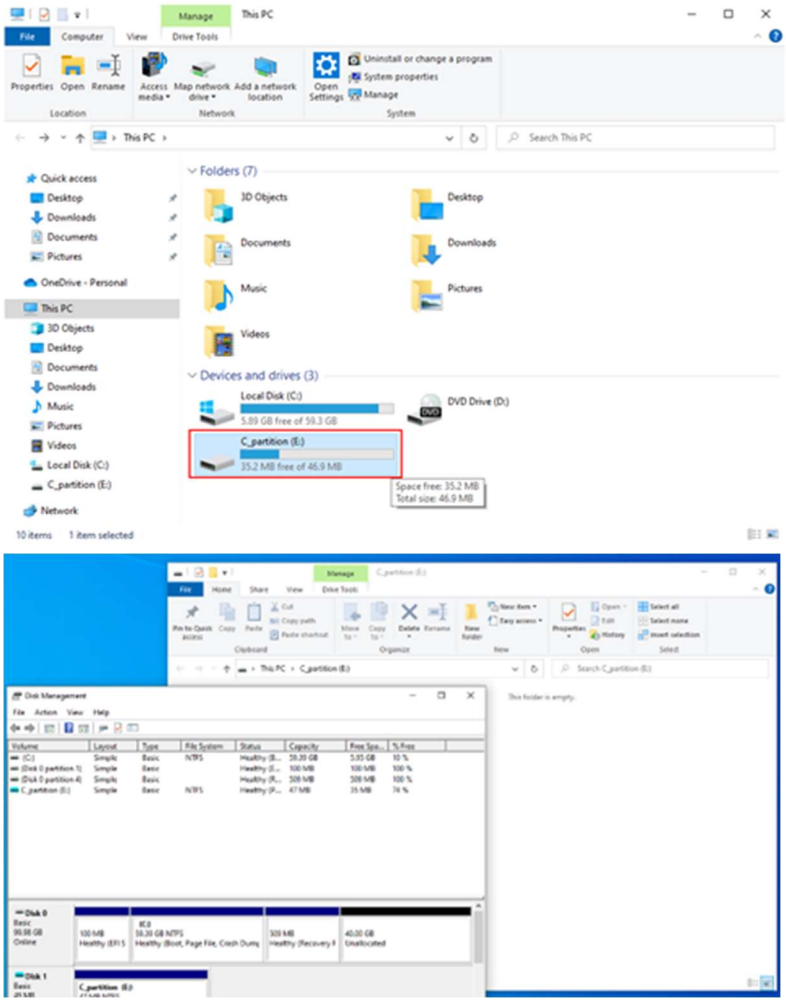




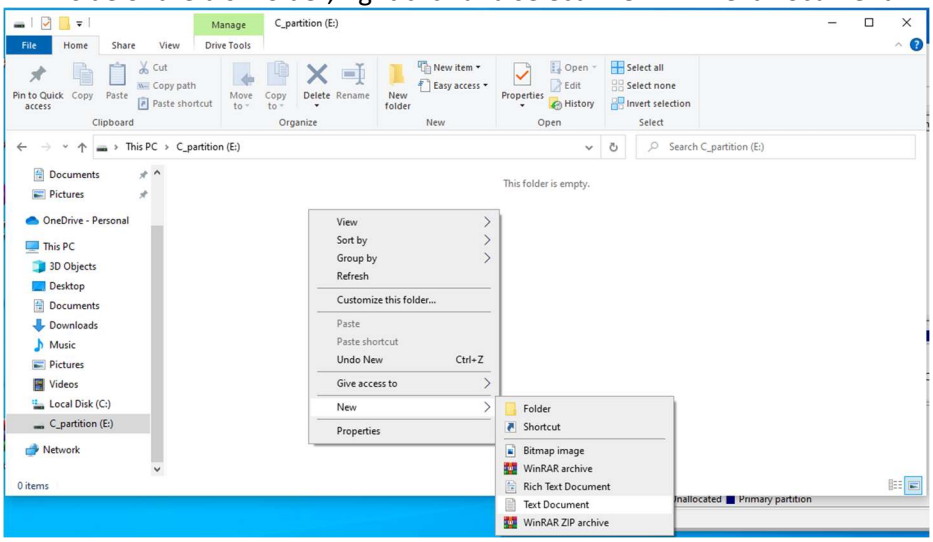
Rename the volume to “C_partition” under the Volume label. Then click “Next” and “Finish”.



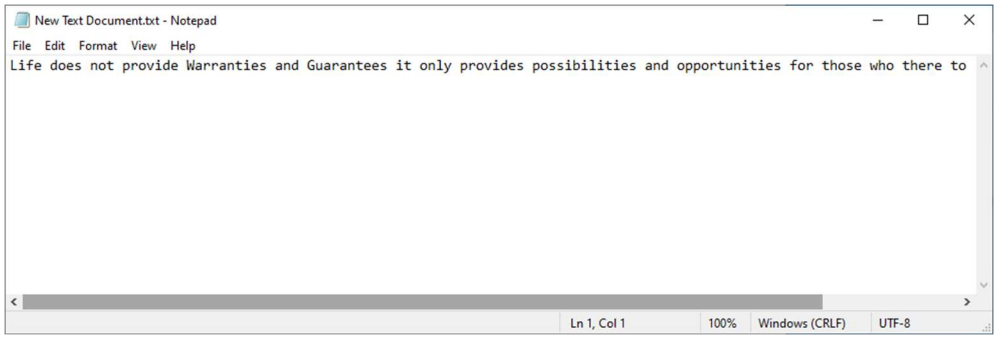
Windows should then automatically open the C_partition folder in the E: disk.
(If not, open the start menu and enter “this pc”)



12. Inside of the disk folder, right click and select “New” > “Text Document”



13. Enter the message: "Life does not provide Warranties and Guarantees it only provides possibilities and opportunities for those who there to make best use of it!"

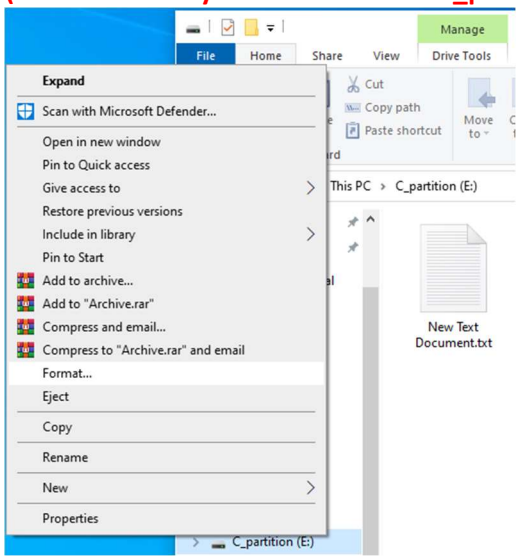


14. Then save and close the window.

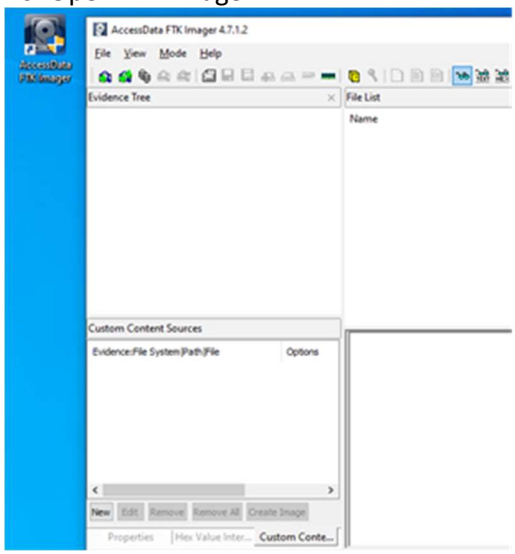
Now, we are going to delete and format the disk.

15. Right click the C_partition (E:) disk and select “Format”.

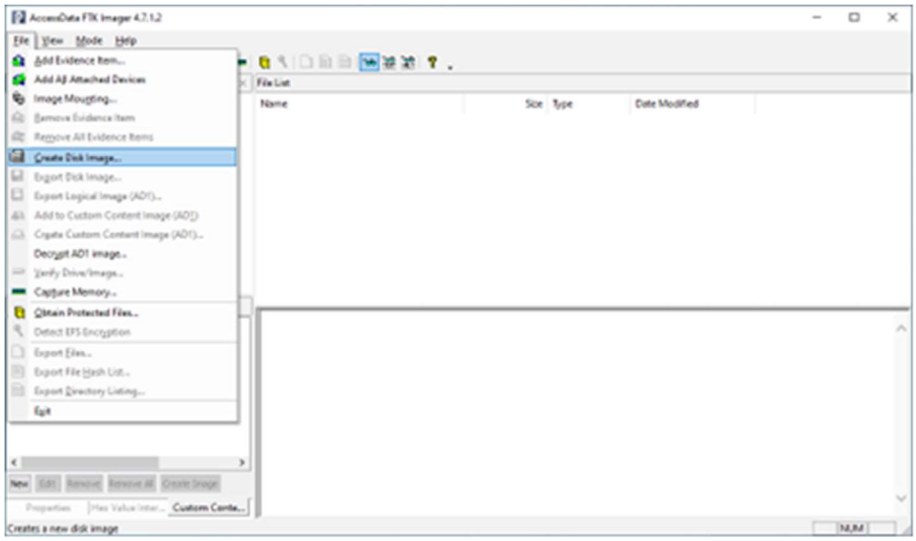
(* Make sure you have selected “C_partition” (E:) and not other disks ***)**



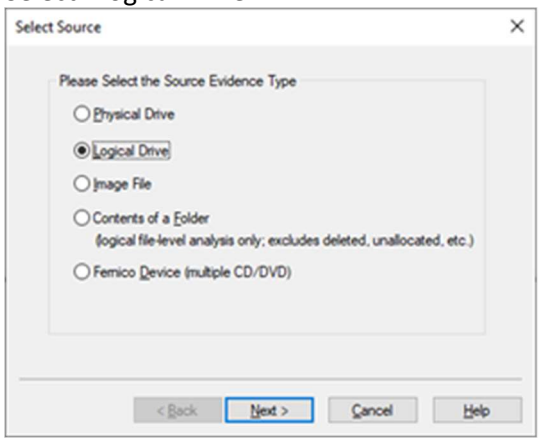
16. Open FTK Imager.



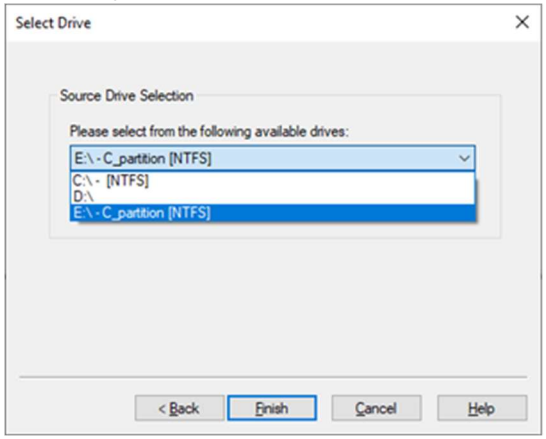
Select “File” -> “Create Disk Image…”



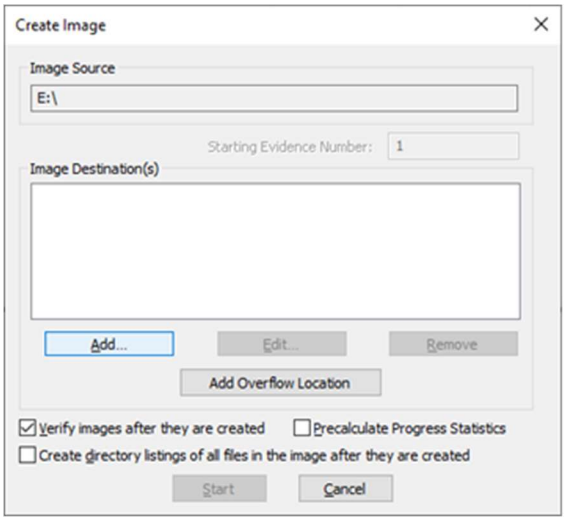
Select “Logical Drive”



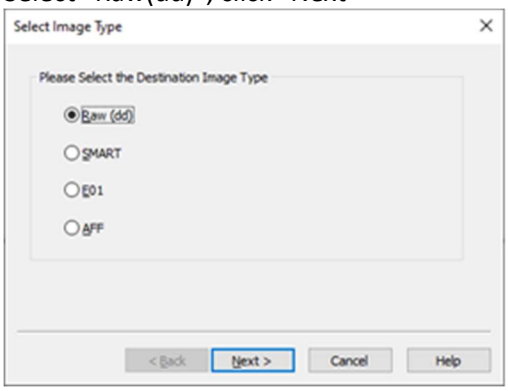
Select E:\ as the source drive. Click “Finish”



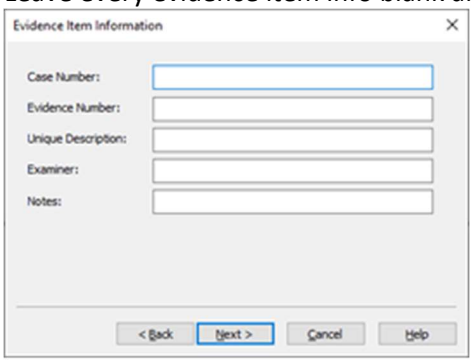
In the Create Image window, click on “Add” image destination.



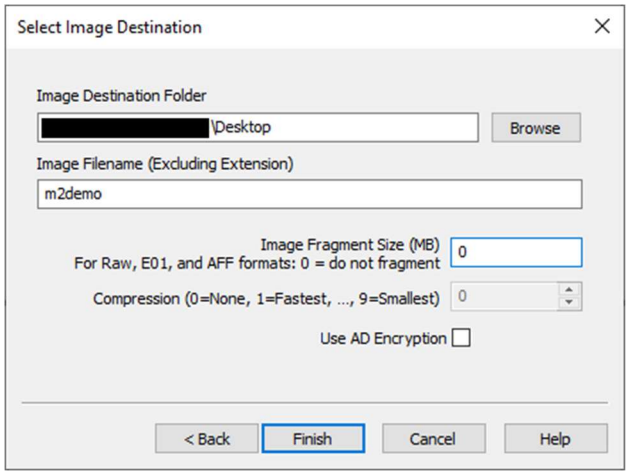
Select “Raw(dd)”, click “Next”



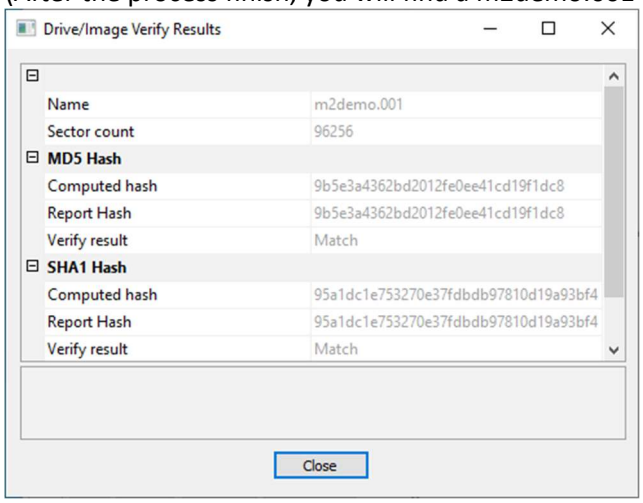
Leave every evidence item info blank and click “Next”



17. For the image destination folder, select under your computer desktop.
For the image filename, enter “m2demo”.
For the Image Fragment Size, enter 0. Then click “Finish”



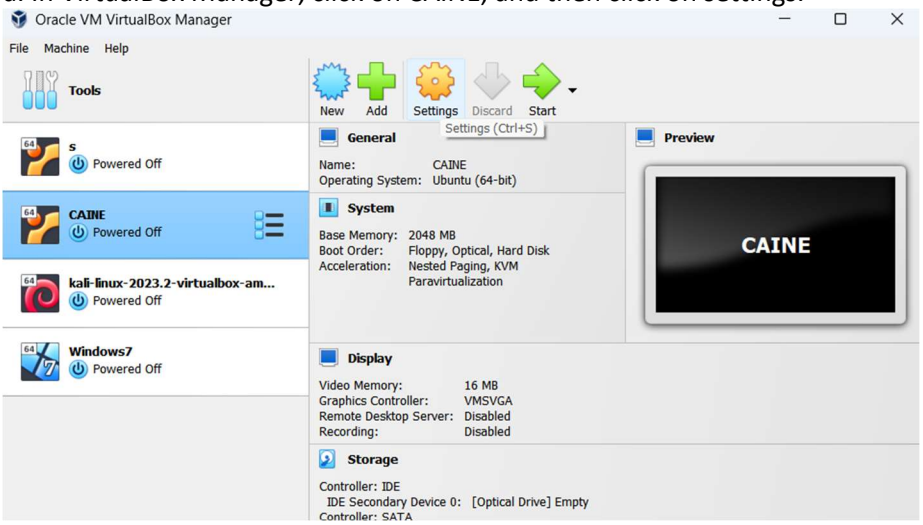
18. Then, back to the Create Image window, click “Start”
(After the process finish, you will find a m2demo.001 on the desktop)



Part2. Perform data acquisition in Linux using dd

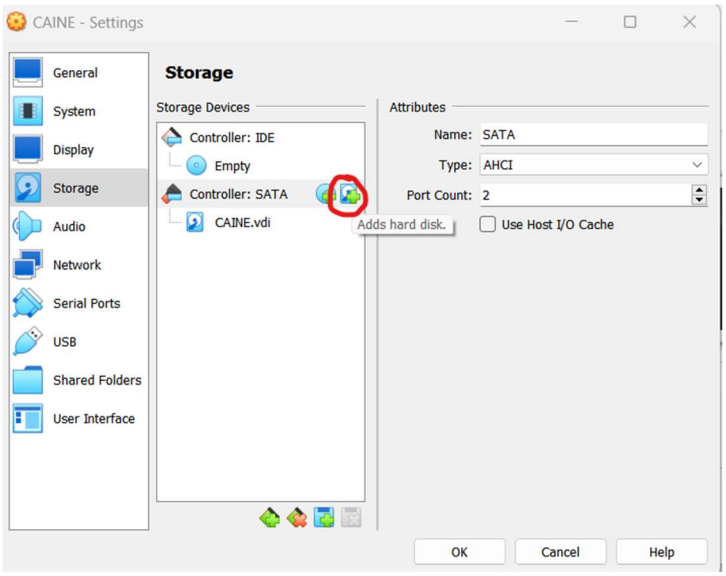
1. Create two virtual drives in CAINE.

a. In VirtualBox Manager, click on CAINE, and then click on settings.

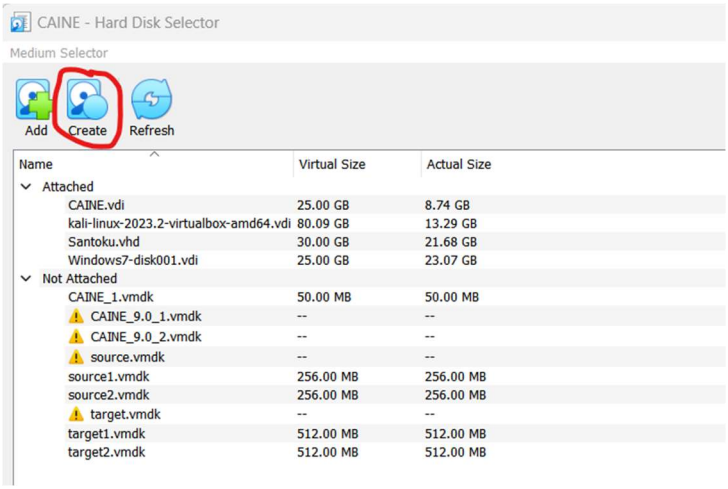


b. In the settings, click on Storage, select Controller: SATA. Then, click on “Adds hard disk” as

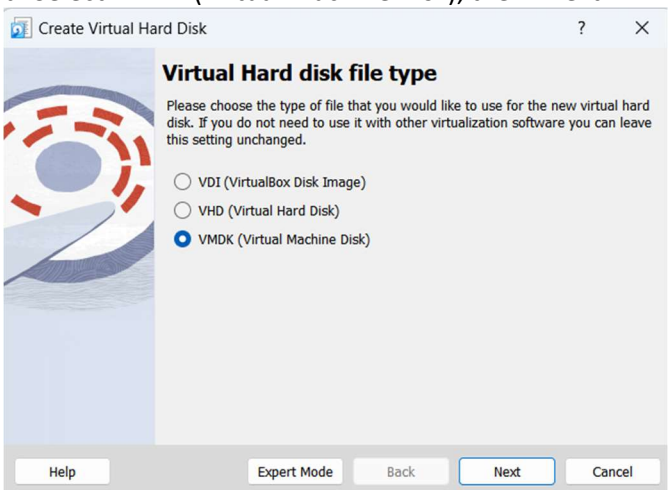
shown below.



c. Click “Create”

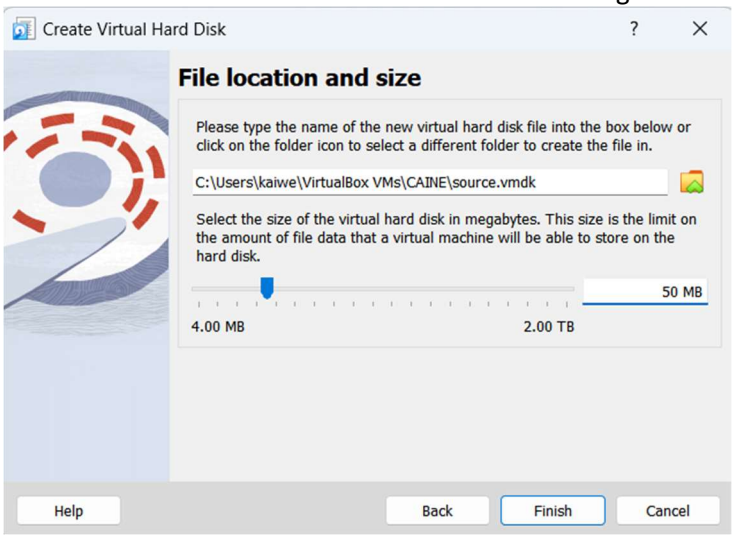


d. Select VMDK (Virtual Machine Disk), then “Next”’

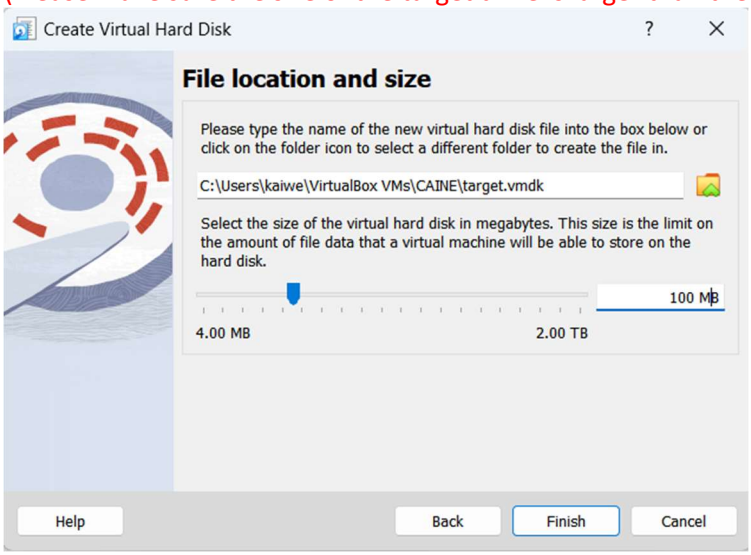


e. Select “Pre-allocate Full Size”, then “Next”

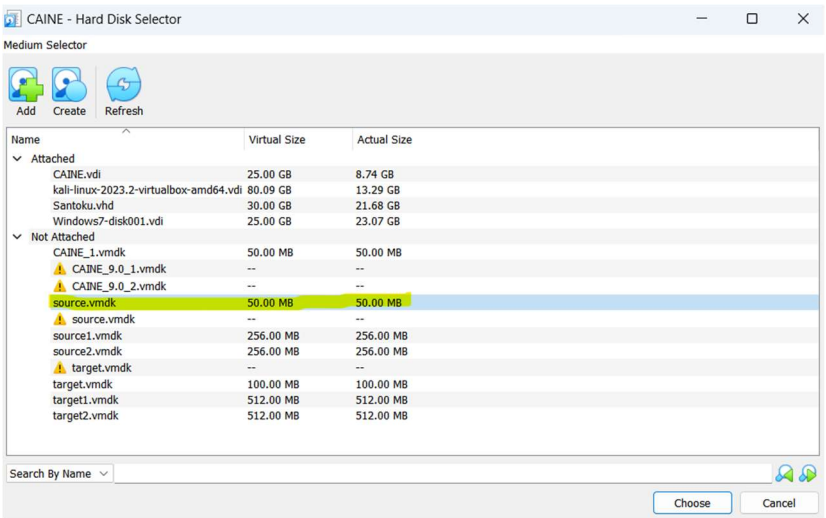
f. Rename the disk name to “source.vmdk” and assign 50 MB to the disk size. Then click “Finish”



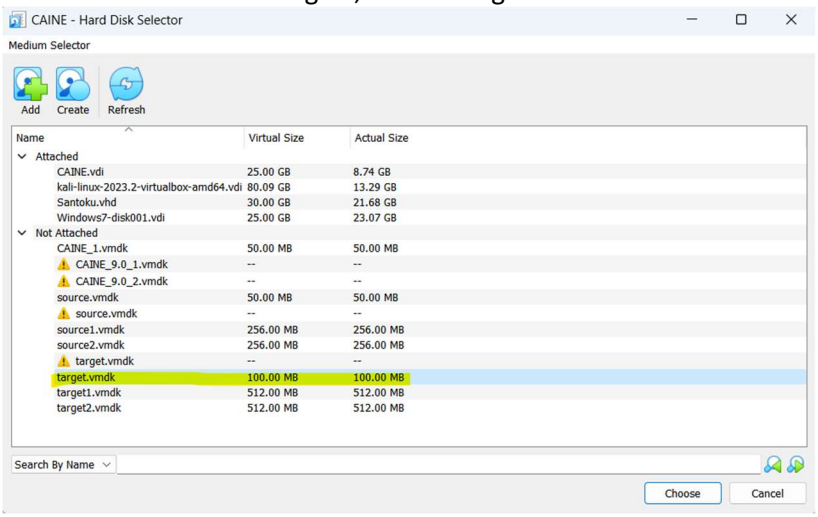
g. Rename the disk name to “target.vmdk” and assign 100 MB to the disk size. Then “Finish”
(Please make sure the size of the target drive is larger than the size of the source drive)



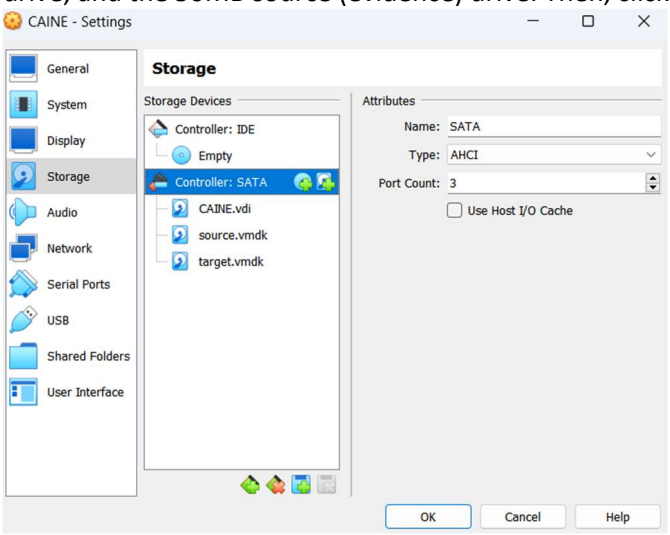
h. Back to the hard disk selector, select “source.vmdk” and click “Choose”.



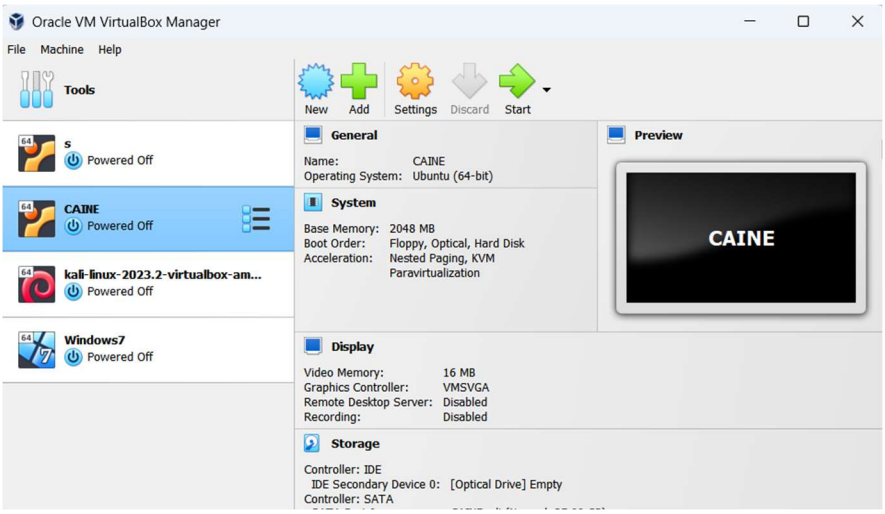
i. Click “Adds hard disk” again, select “target.vmdk” and click “Choose”.



j. Now you can see the VM has three hard drives: the 20GB original hard drive, the 100MB target drive, and the 50MB source (evidence) drive. Then, click “OK”



k. Next you can start the virtual machine.



2. In CAINE, find terminal in Main->system tools->MATE Terminal. In Kali Linux, find the terminal on the panel to the left of the screen.

3. Type **fdisk –l** to show the current disks. You can see there are three disks shown in the screenshot.

a. /dev/sda is the original hard drive assigned to the CAINE vm;

b. /dev/sdb (50MB) is the source (evidence) drive we added;

c. /dev/sdc (100MB) is the target drive we'll use to store the image.

d. !!!! Please always make sure you are using the smaller driver as the source/evidence drive, and the bigger drive as the target drive.

```
root@cainecf:/home/cainecf# fdisk -l
Disk /dev/sda: 25 GiB, 26843545600 bytes, 52428800 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
Disklabel type: dos
Disk identifier: 0x6e89d69f

Device     Boot Start      End  Sectors Size Id Type
/dev/sda1        2048 52428799 52426752  25G 83 Linux


Disk /dev/sdb: 50 MiB, 52428800 bytes, 102400 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


Disk /dev/sdc: 100 MiB, 104857600 bytes, 204800 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
root@cainecf:/home/cainecf#
```

4. Create a new partition on the evidence drive /dev/sdb

a. Type **fdisk /dev/sdb**

b. Type **p**

c. Type **n**

d. Type **p**

e. Type **1**

f. Hit enter

g. Hit enter

h. Type **p**

```
root@cainecf:/home/cainecf# fdisk /dev/sdb

Welcome to fdisk (util-linux 2.27.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognised partition table.
Created a new DOS disklabel with disk identifier 0x9e67000a.

Command (m for help):
Command (m for help): p
Disk /dev/sdb: 50 MiB, 52428800 bytes, 102400 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
Disklabel type: dos
Disk identifier: 0x9e67000a

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-102399, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-102399, default 102399):

Created a new partition 1 of type 'Linux' and of size 49 MiB.

Command (m for help):
```

5. Create the file system

a. Type **t**

b. Type **c**

c. Type **p**

d. Type **w**

e. Type **fdisk -l**

f. Type **mkfs.msdos -vF32 /dev/sdb1**

```
Command (m for help): p
Disk /dev/sdb: 50 MiB, 52428800 bytes, 102400 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
Disklabel type: dos
Disk identifier: 0x9e67000a

Device     Boot Start    End Sectors Size Id Type
/dev/sdb1        2048 102399  100352  49M 83 Linux

Command (m for help): t
Selected partition 1
Partition type (type L to list all types): c
Changed type of partition 'Linux' to 'W95 FAT32 (LBA)'.

Command (m for help):
```

6. Mount the drive

a. Type **ls /mnt**

b. Type **mkdir /mnt/sdb1**

c. Type **ls /mnt**

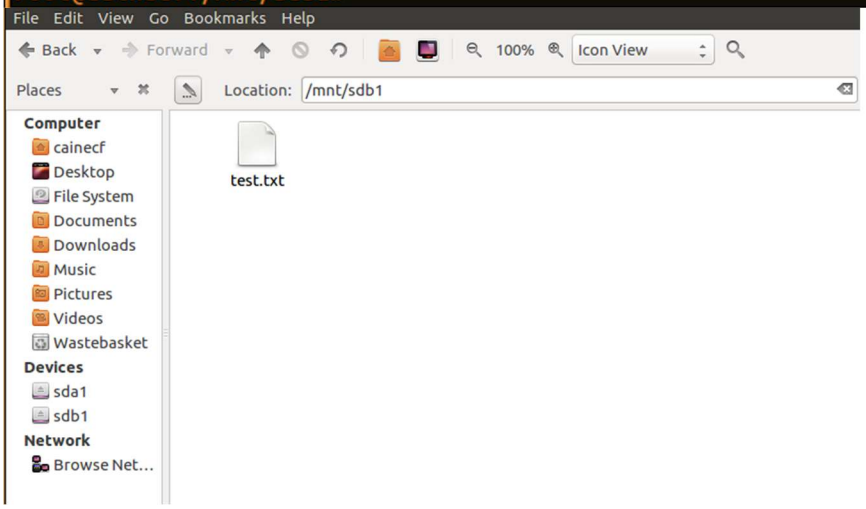
d. Type **mount -t vfat /dev/sdb1 /mnt/sdb1** to mount the /dev/sdb1 to /mnt/sdb1

e. Type **ls /mnt/sdb1**
f. Type **cd /mnt/sdb1**
g. Type **ls**
h. Type **touch test.txt**

```
root@cainecf:/# mkdir /mnt/sdb1
root@cainecf:/# ls /mnt
sdb1  sdc1  sde2
root@cainecf:/# mount -t vfat /dev/sdb1 /mnt/sdb1
root@cainecf:/# ls /mnt/sdb1
root@cainecf:/# cd /mnt/sdb1
root@cainecf:/mnt/sdb1# ls
root@cainecf:/mnt/sdb1# touch test.txt
```

i. Add content to the file by entering the command:
echo "Life does not provide Warranties and Guarantees it only provides possibilities and opportunities for those who there to make best use of it!" >> test.txt
*You can enter different content, but please enter the same content that you used in part1.

```
root@cainecf:/mnt/sdb1# echo "Life does not provide Warranties and Guarante
es it only provides possibilities and opportunites for those who there to m
ake best use of it!" >> test.txt
root@cainecf:/mnt/sdb1#
```



*The file is created successfully.

7. Type command **cd /** and go back to the root folder, type command **fdisk -l**

```
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x6e89d69f

Device     Boot Start      End  Sectors Size Id Type
/dev/sda1        2048 52428799 52426752  25G 83 Linux


Disk /dev/sdb: 50 MiB, 52428800 bytes, 102400 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
Disklabel type: dos
Disk identifier: 0x9e67000a

Device     Boot Start    End Sectors Size Id Type
/dev/sdb1        2048 102399  100352  49M  c W95 FAT32 (LBA)


Disk /dev/sdc: 100 MiB, 104857600 bytes, 204800 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
root@cainecf:/#
```

Zero out the evidence drive using the following command:

dd if=/dev/zero of=/dev/sdb

This step is to zero out the evidence drive.

```
root@cainecf:/# dd if=/dev/zero of=/dev/sdb
dd: writing to '/dev/sdb': No space left on device
102401+0 records in
102400+0 records out
52428800 bytes (52 MB, 50 MiB) copied, 1.51406 s, 34.6 MB/s
root@cainecf:/#
```

8. Zero out the target drive using the following command:

dd if=/dev/zero of=/dev/sdc

This will take some time depending on the virtual machine configuration. In this activity, it is very fast because we have a very small hard drive: 100MB. If you are zero-out a big drive, please be patient as it is writing bit by bit. The time can vary from seconds to days.

Note:

1) Make sure you are zeroing out the target drive, NOT the original drive or other drives!

2) Make sure you are zeroing out the entire drive, not just a partition under it. For example, /dev/sdb is the entire drive, but /dev/sdb1 is just a partition.

```
root@cainecf:/# dd if=/dev/zero of=/dev/sdc
dd: writing to '/dev/sdc': Operation not permitted
1+0 records in
0+0 records out
0 bytes copied, 0.000127913 s, 0.0 kB/s
root@cainecf:/#
```

Currently the target drive is completely empty with all 0s. Next, you need to create a partition on the target drive and then create a file system on the partition, so that (image) files can be stored on the drive.

9. Type **fdisk /dev/sdc** to begin creating a partition on the target drive.

a. Then type **m** to show the menu.

```
root@cainecf:/# fdisk /dev/sdc

Welcome to fdisk (util-linux 2.27.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognised partition table.
Created a new DOS disklabel with disk identifier 0x5840b510.

Command (m for help): m

Help:

  DOS (MBR)
   a   toggle a bootable flag
   b   edit nested BSD disklabel
   c   toggle the dos compatibility flag

  Generic
   d   delete a partition
   F   list free unpartitioned space
   l   list known partition types
   n   add a new partition
   p   print the partition table
   t   change a partition type
   v   verify the partition table
   i   print information about a partition

  Misc
   m   print this menu
   u   change display/entry units
   x   extra functionality (experts only)

  Script
   I   load disk layout from sfdisk script file
   O   dump disk layout to sfdisk script file

  Save & Exit
   w   write table to disk and exit
```

b. Type **p** to print the partition table and see if there are any partitions on /dev/sdb**.** If there are no partitions on target drive, the output will sometimes be similar to:
Disk /dev/sdb doesn't contain a valid partition table
Or the output simply doesn't show any partition information.
Then you'll need to create a new partition following the steps below.
c. Type **n** and hit enter, to create a new partition. It lists two partition types: primary and extended.
d. Type **p** and hit enter, to choose a primary partition table.
e. Type **1** and hit enter, to select the first partition;
then **hit enter** for first section and **hit enter** again for the last sector to use the default value.
f. After the new partition is created, type **p** again to show current partitions on /dev/sdb and you'll see the newly created partition. In this case /dev/sdc1 is the new partition created.

```
Command (m for help): p
Disk /dev/sdc: 100 MiB, 104857600 bytes, 204800 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
Disklabel type: dos
Disk identifier: 0xf79005f1

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-204799, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-204799, default 204799):

Created a new partition 1 of type 'Linux' and of size 99 MiB.
```

```
Command (m for help): p
Disk /dev/sdc: 100 MiB, 104857600 bytes, 204800 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
Disklabel type: dos
Disk identifier: 0xf79005f1

Device     Boot Start    End Sectors Size Id Type
/dev/sdc1        2048 204799  202752  99M 83 Linux
```

10. In the last step we created a new linux partition /dev/sdc1. In this step, we'll change the newly created partition to windows 95 FAT32 file system.

a. Type **m** to show the menu.

```
Command (m for help): m

Help:

  DOS (MBR)
   a   toggle a bootable flag
   b   edit nested BSD disklabel
   c   toggle the dos compatibility flag

  Generic
   d   delete a partition
   F   list free unpartitioned space
   l   list known partition types
   n   add a new partition
   p   print the partition table
   t   change a partition type
   v   verify the partition table
   i   print information about a partition

  Misc
   m   print this menu
   u   change display/entry units
   x   extra functionality (experts only)

  Script
   I   load disk layout from sfdisk script file
   O   dump disk layout to sfdisk script file

  Save & Exit
   w   write table to disk and exit
   q   quit without saving changes

  Create a new label
   g   create a new empty GPT partition table
   G   create a new empty SGI (IRIX) partition table
   o   create a new empty DOS partition table
   s   create a new empty Sun partition table
```

b. Type **t** to change the partition type.

c. Type **l** (lowercase L) to show available file systems and their code values.

```
Command (m for help): t
Selected partition 1
Partition type (type L to list all types): l

 0  Empty           24  NEC DOS         81  Minix / old Lin bf  Solaris
 1  FAT12           27  Hidden NTFS Win 82  Linux swap / So c1  DRDOS/sec (F
AT-
 2  XENIX root      39  Plan 9          83  Linux           c4  DRDOS/sec (F
AT-
 3  XENIX usr       3c  PartitionMagic  84  OS/2 hidden or  c6  DRDOS/sec (F
AT-
 4  FAT16 <32M      40  Venix 80286     85  Linux extended  c7  Syrinx
 5  Extended        41  PPC PReP Boot   86  NTFS volume set da  Non-FS data
 6  FAT16           42  SFS             87  NTFS volume set db  CP/M / CTOS
/ .
 7  HPFS/NTFS/exFAT 4d  QNX4.x          88  Linux plaintext de  Dell Utility
 8  AIX             4e  QNX4.x 2nd part 8e  Linux LVM       df  BootIt
 9  AIX bootable    4f  QNX4.x 3rd part 93  Amoeba          e1  DOS access
 a  OS/2 Boot Manag 50  OnTrack DM      94  Amoeba BBT      e3  DOS R/O
 b  W95 FAT32       51  OnTrack DM6 Aux 9f  BSD/OS          e4  SpeedStor
 c  W95 FAT32 (LBA) 52  CP/M            a0  IBM Thinkpad hi ea  Rufus alignm
ent
 e  W95 FAT16 (LBA) 53  OnTrack DM6 Aux a5  FreeBSD         eb  BeOS fs
 f  W95 Ext'd (LBA) 54  OnTrackDM6      a6  OpenBSD         ee  GPT
10  OPUS            55  EZ-Drive        a7  NeXTSTEP        ef  EFI (FAT-12/
16/
11  Hidden FAT12    56  Golden Bow      a8  Darwin UFS      f0  Linux/PA-RIS
C b
12  Compaq diagnost 5c  Priam Edisk     a9  NetBSD          f1  SpeedStor
```

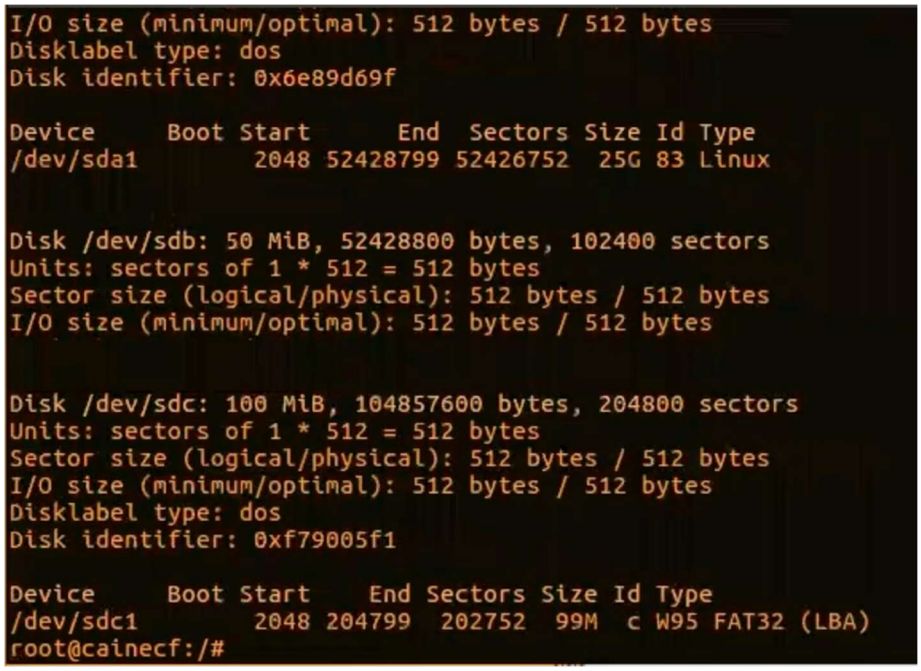
d. Type **c** to change the partition to Windows 95 FAT32(LBA).

e. Type **p** to display the newly changed drive.

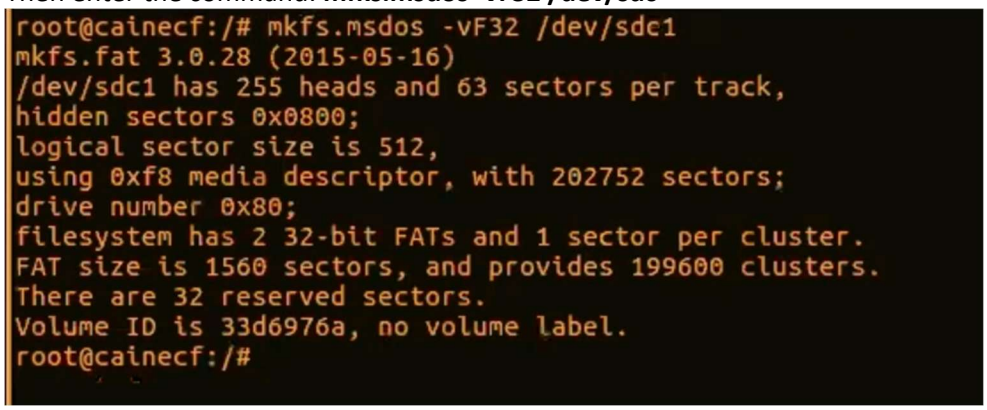
f. Type **w** to save/write the newly created partition to the /dev/sdc drive.

```
Partition type (type L to list all types): c
Changed type of partition 'Linux' to 'W95 FAT32 (LBA)'.

Command (m for help): p
Disk /dev/sdc: 100 MiB, 104857600 bytes, 204800 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
Disklabel type: dos
Disk identifier: 0xf79005f1

Device     Boot Start    End Sectors Size Id Type
/dev/sdc1        2048 204799  202752  99M  c W95 FAT32 (LBA)

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Synching disks.

root@cainecf:/#
```

11. Type **fdisk –l** to show all disks.



Then enter the command: **mfks.msdos -vF32 /dev/sdc**



In order to store files to the target drive, we need to first mount the target drive to the operating system. The following steps will mount the target drive to /mnt folder.

12. Type **ls /mnt** to show files under /mnt.
13. Type **mkdir /mnt/sdc1** to create a folder under /mnt.
14. Type **ls /mnt** again to show files under /mnt.
15. Type **mount –t vfat /dev/sdc1 /mnt/sdc1** to mount the target drive partition. (–t vfat means the file system type is vfat).
16. Type **cd /mnt/sdc1** to change to default directory to target drive
17. Type **ls –al** to show the contents of the target drive's root level.
18. Type **mkdir case1** to create a target directory.
19. Type **ls** to confirm that the new directory has been created.

```
root@cainecf:/# ls /mnt
sdb1
root@cainecf:/# mkdir /mnt/sdc1
root@cainecf:/# ls /mnt
sdb1  sdc1
root@cainecf:/# mount -t vfat /dev/sdc1 /mnt/sdc1
root@cainecf:/# cd /mnt/sdc1
root@cainecf:/mnt/sdc1# ls -al
total 5
drwxr-xr-x 2 root root  512 Jan  1  1970 .
drwxr-xr-x 4 root root 4096 Jul 21 01:29 ..
root@cainecf:/mnt/sdc1#
```

Next, we'll need to calculate the hash of the evidence drive /dev/sdb, so that later we can compare this hash with the hash of image after acquisition to see if they are the same. If they are the same, that means the acquisition is successful.

20. Type **md5sum /dev/sdb |tee /mnt/sdc1/case1/pre-image.md5.txt**
a. Md5sum calculates the hash of the evidence drive using MD5 algorithm
b. |tee means the output is added to the .txt file and displayed in terminal.

```
root@cainecf:/mnt/sdc1# md5sum /dev/sdb |tee /mnt/sdc1/case1/pre-image.md5.txt
25e317773f308e446cc84c503a6d1f85  /dev/sdb
```

Now we can perform data acquisition.

21. To acquire data from the evidence drive /dev/sdb, type
dcfldd if=/dev/sdb of=/mnt/sdc1/case1/image1.dd conv=noerror,sync hash=md5 hashwindow=0 hashlog=/mnt/sdc1/case1/post-image.md5.txt
Special note here: to acquire the entire drive, if=/dev/sdb is correct. If you do if=/dev/sdb1, that means only acquire the sdb1 partition. This is not recommended in real investigation.

```
root@cainecf:/mnt/sdc1# dcfldd if=/dev/sdb of=/mnt/sdc1/case1/image1.dd conv=noe
rror,sync hash=md5 hashwindow=0 hashlog=/mnt/sdc1/case1/post-image.md5.txt
1536 blocks (48Mb) written.
1600+0 records in
1600+0 records out
root@cainecf:/mnt/sdc1# cd case1
```

a. If you want to create segmented volumes of 2MB each, you may use (**Please note: This step is optional.** If you have already done acquisition using dcfldd, you cannot do this step again, otherwise you'll do acquisition twice.)
dcfldd if=/dev/sdb split=2M of=/mnt/sdc1/case1/image1.dd conv=noerror,sync hash=md5 hashwindow=0 hashlog=/mnt/sdc1/case1/post-image.md5.txt
b. If use dd command for data acquisition, the command should be (**Please note: This step is optional.** If you have already done acquisition using dcfldd, you can NOT do this step again, otherwise you'll do acquisition twice.)
dd if=/dev/sdb |split -b 2m – image_sdc
if you want to create multiple segments.
Otherwise you can simply use
dd if=/dev/sdb of=/mnt/sdc1/case1/image1-dd.dd

22. To validate the acquired data, there are two ways if using dcfldd for data acquisition (Please note: only choose one way below. You don't need to do both. First way is recommended):

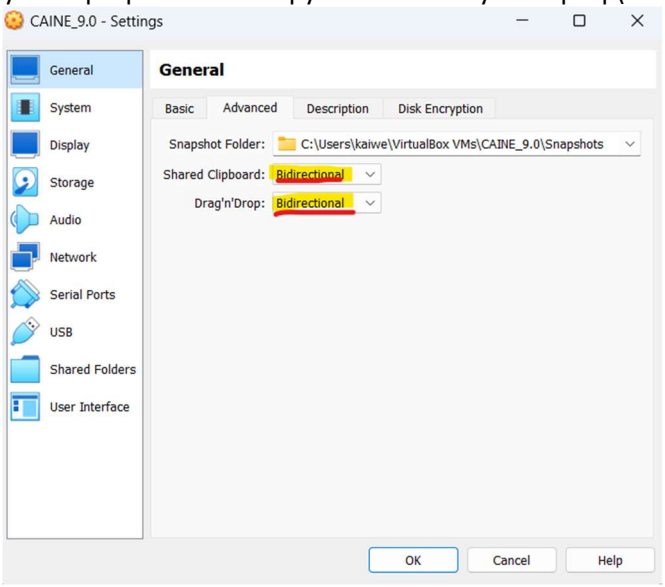
a. Go to the /case1 folder by
cd case1
Type
cat pre-image.md5.txt
to show the hash value of original evidence drive;
The hash of the acquired image is already computed and put into post-image.md5.txt.
Type
cat post-image.md5.txt
to show the hash value and then compare the value in pre-image.md5.txt
If pre-image.md5.txt and post-image.md5.txt have the same value, that means the acquisition is successful.

```
root@cainecf:/mnt/sdc1# cd case1
root@cainecf:/mnt/sdc1/case1# cat pre-image.md5.txt
25e317773f308e446cc84c503a6d1f85  /dev/sdb
root@cainecf:/mnt/sdc1/case1# cat post-image.md5.txt
Total (md5): 25e317773f308e446cc84c503a6d1f85
```

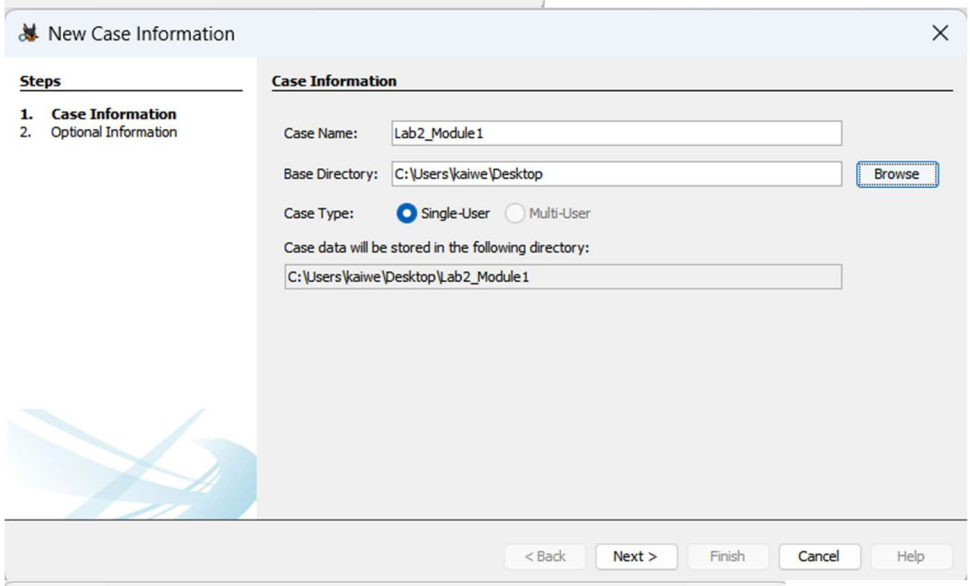
b. Type **dcfldd if=/dev/sdb vf=/mnt/sdc1/case1/image1.dd** (**Note: this only applies to the nonsegmented image file**).
23. To validate the acquired data if using dd for data acquisition (Please note: only use this if you have used dd to do data acquisition):
a. Type **md5sum /dev/sdb |tee /mnt/sdc1/case1/pre-image.md5.txt**
b. Type **cat image_sdb.*|md5sum > post-image.md5.txt**
And then compare the values in both files to see if they are the same. This applies to the segmented image file too.

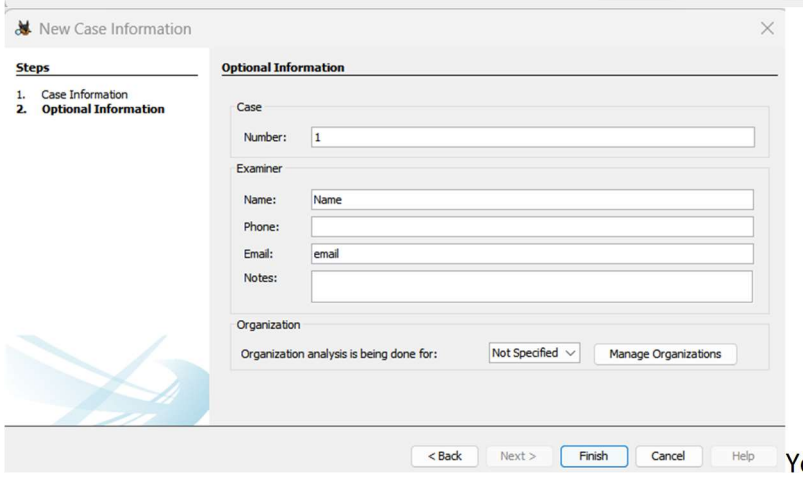
Part3. Compare the results using Autopsy.

1. Copy the image files to the windows system. For CAINE, you can open the setting and then set the "shared Clipboard" and "Drag'n'Drop" as Bidirectional. Copy the image.dd file from CAINE machine to your laptop and then copy the file from your laptop(host) to windows machine.



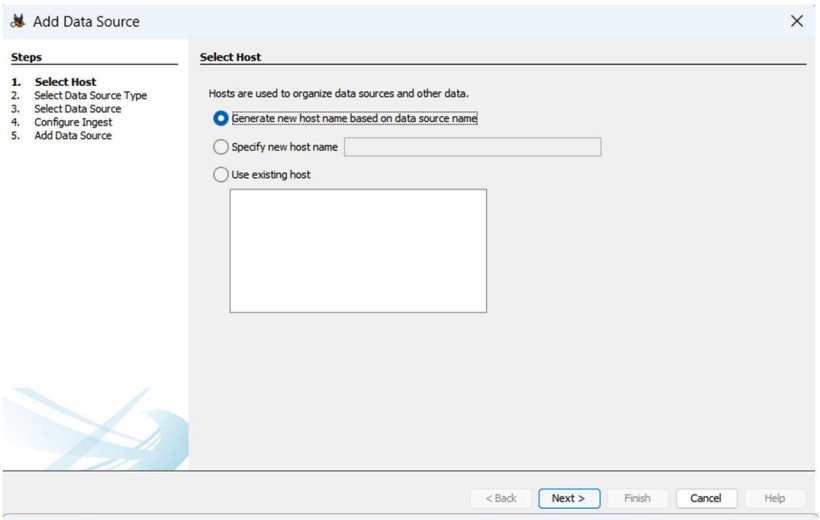
2. Open Autopsy and then create a new case. Name the new case as "Lab2_Module1".

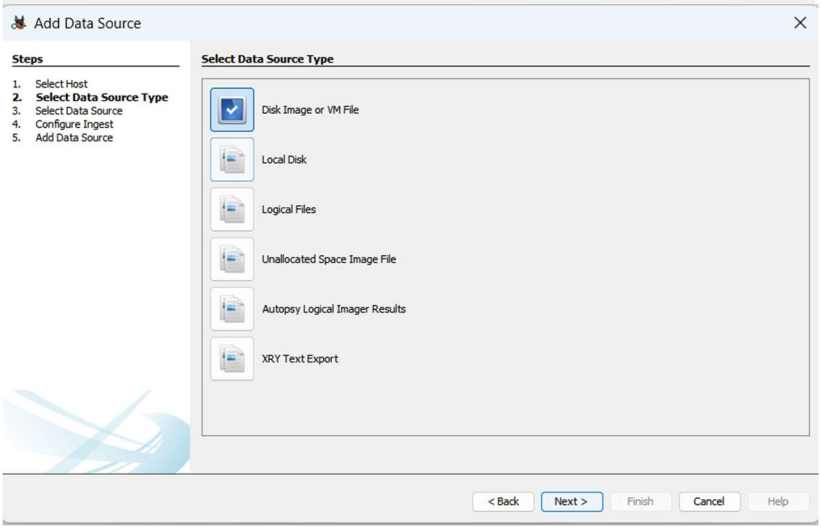




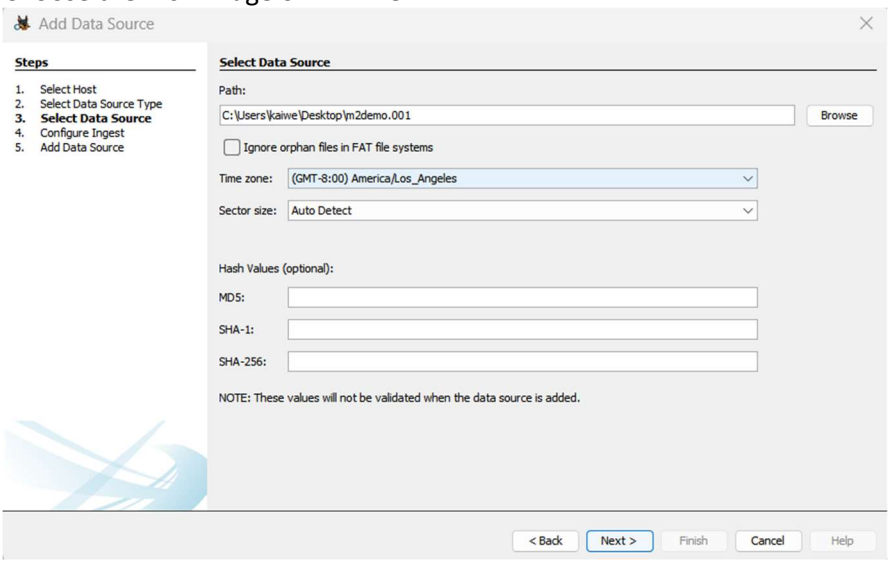
You can enter your name and email.

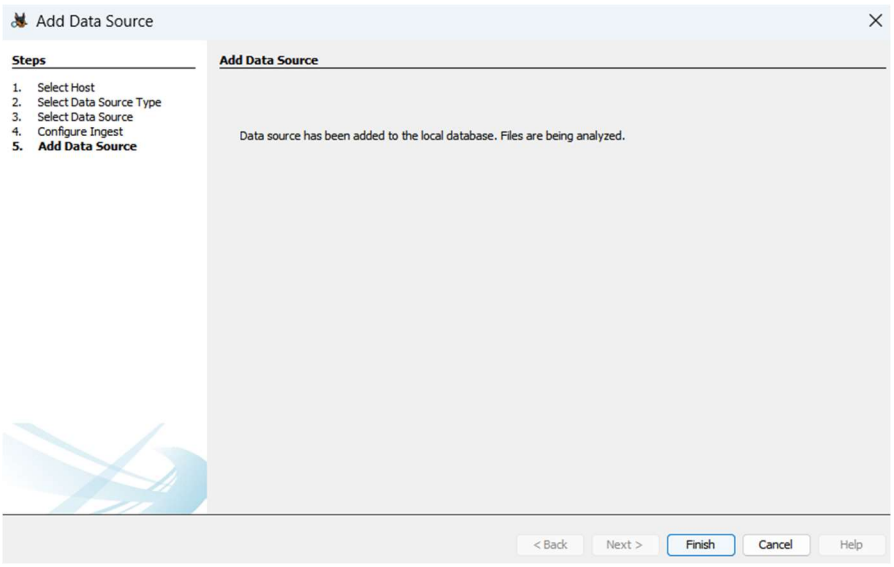
3. Add two data sources, m2demo.001, and image1.dd.
Add m2demo.001 first.



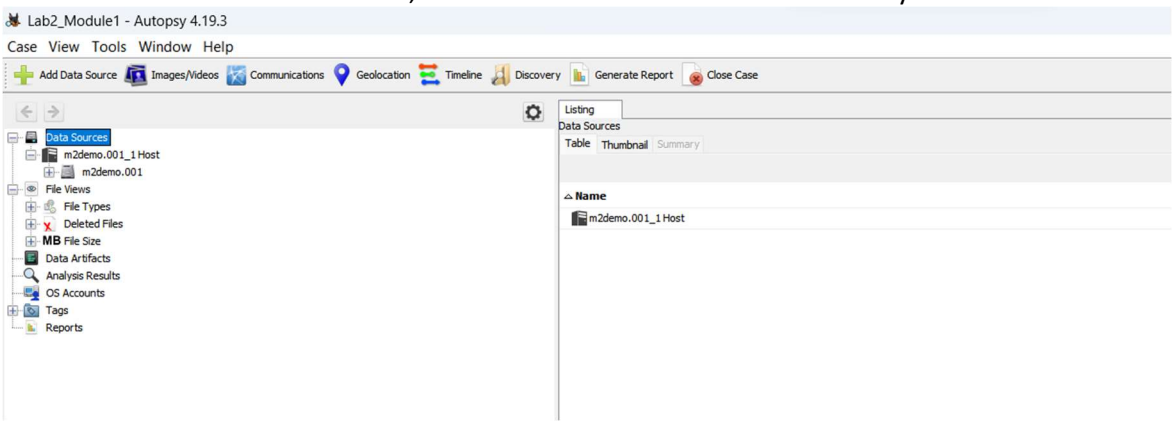


Choose the Disk Image or VM File.

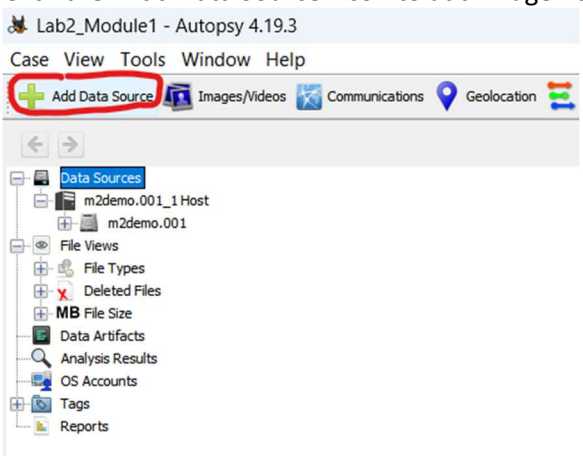


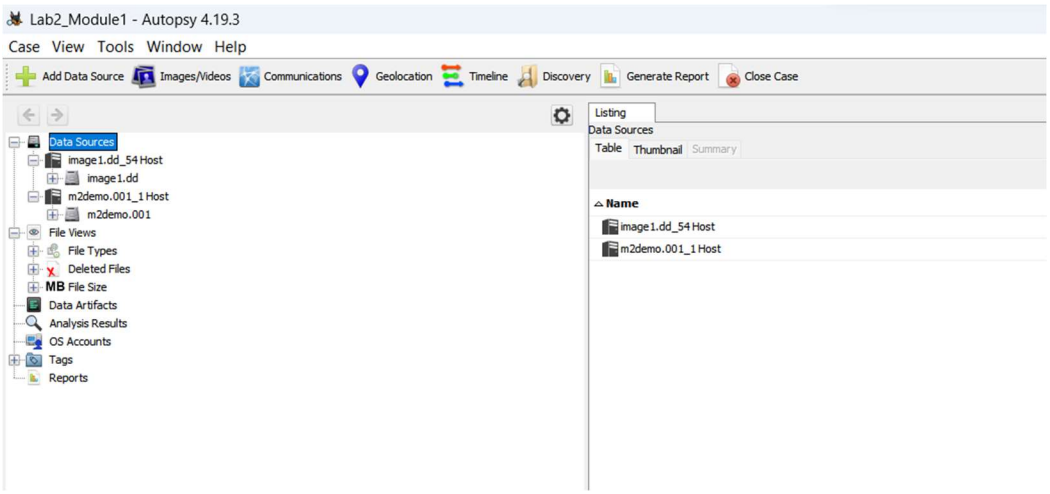


As shown in the screenshot below, the m2demo.001 was added successfully.



Click the “Add Data Source” icon to add image1.dd





After adding these two files successfully, you can start analyzing the data. As shown in the screenshot below, there is no data for the CAINE zero-out one. But some data still exists even though the virtual disk is formatted in the Windows system.

M2demo.dd

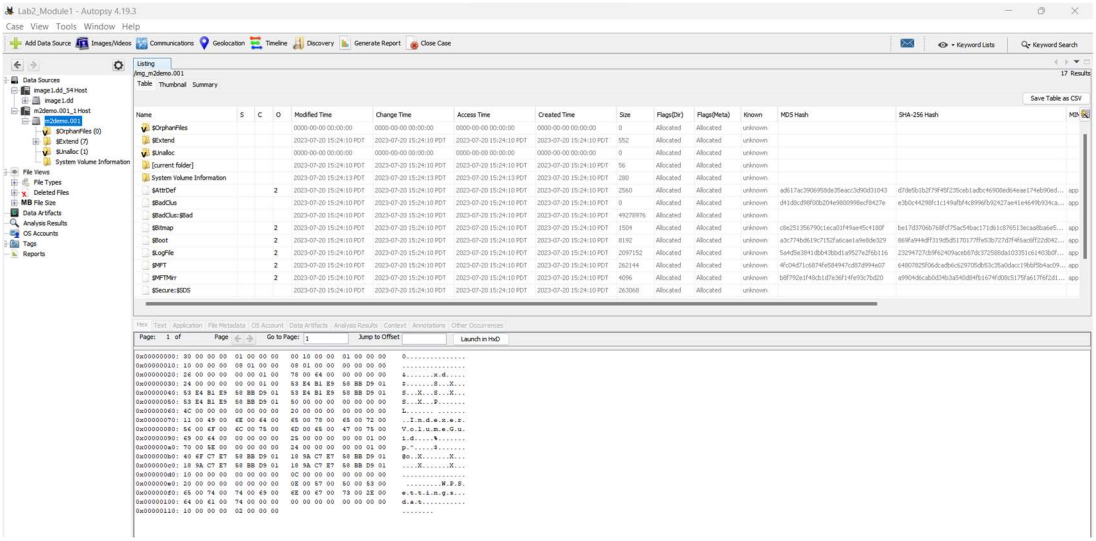
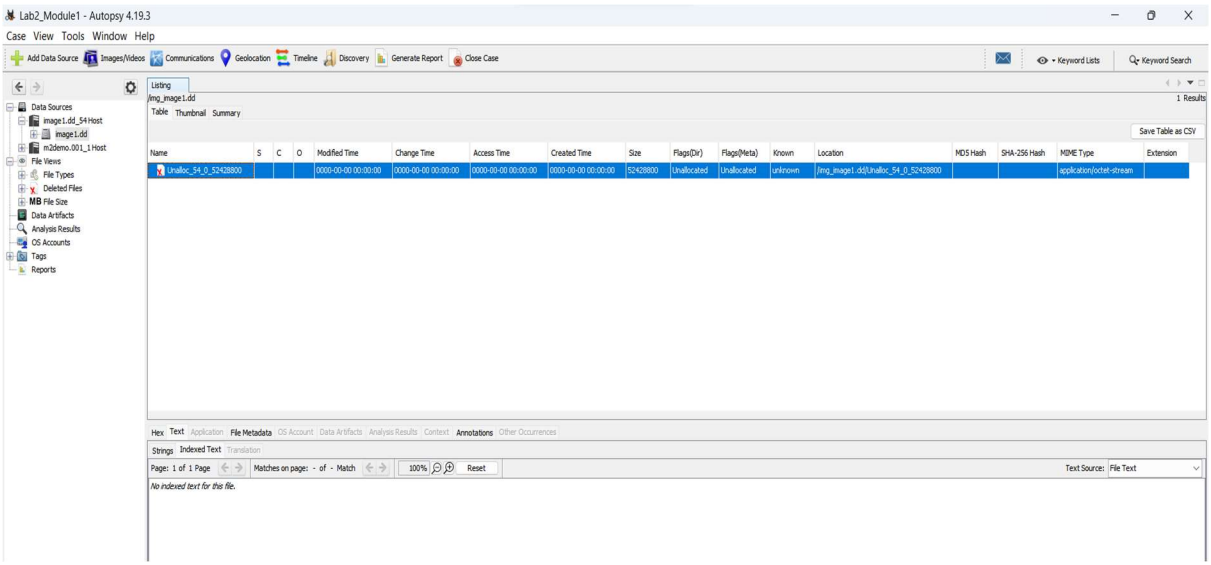


Image1.dd



Or you can also use FTK Image to examine the result, for the image1.dd one, there is no data, everything is zeroed out.