BitCurator

Quick Start Guide
Last updated: August 9, 2023
Release(s): 4.3.0 and later

BitCurator
CONSORTIUM

Table of Contents

Getting Started

Visit https://github.com/BitCurator/bitcurator-distro/wiki/Releases to find the latest release. Virtual Appliance files may also be available for some recent releases.

Note: BitCurator must be deployed on an x86/amd64 version of Ubuntu. Currently, it is not possible to use the Virtual Appliance or deploy it on systems with ARM processors (including Apple M1).

Using the BitCurator Virtual Appliance:

If you have downloaded one of our pre-configured Virtual Appliance files, follow the instructions in the section **Importing the BitCurator Virtual Appliance**.

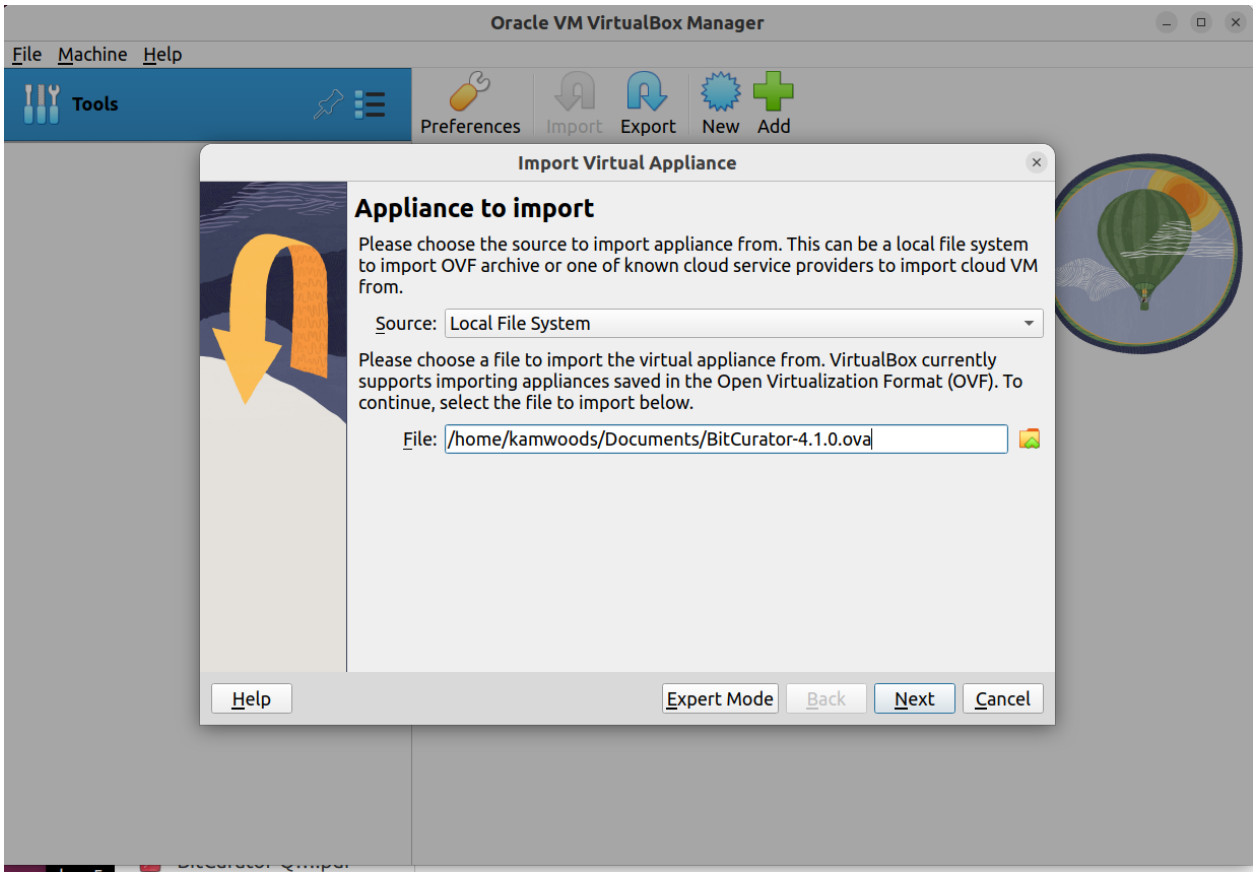
Building BitCurator on your own host or in a new VirtualBox VM:

If you wish to build the environment from scratch on your own physical host or VM, follow the instructions in the section **New BitCurator Installations (Step 1): Install Ubuntu 22.04LTS.**

An internet connection is required during all stages of the installation.

Importing the BitCurator Virtual Appliance

The fastest way to get started with BitCurator is to download our latest Virtual Appliance (.ova) file and import it into VirtualBox. Once you have downloaded the latest **.ova** file from https://github.com/BitCurator/bitcurator-distro/wiki/Releases, start VirtualBox on your host and select **Import Appliance** in the **File** menu.



Click on the folder icon, navigate to the location of your downloaded file, select it, and then click **Next**.

Importing the appliance may take several minutes. Once the import has been completed, you may wish to check the **Settings** for the machine. By default, it is configured with **4GB RAM** and **2 Processor Cores**. We recommend increasing this to at least 8GB and 4 Processor cores, if your hardware is capable.

Start the machine and continue with the section **Working in the BitCurator Environment.**

New BitCurator Installations (Step 1): Install Ubuntu 22.04LTS

Download the latest 64-bit Ubuntu 22.04 desktop image from https://releases.ubuntu.com/22.04/ and install on your local machine or in a VM[1].

Configure your Ubuntu installation as desired. To (optionally) reproduce the default configuration of BitCurator, when prompted use **BitCurator** for **Your name**, **bitcurator** for **Your computer's name**, and **bcadmin** for **Pick a username**. Enter a strong password of your choice. In the examples used in this document, our password will simply be **bcadmin**.

When installation is completed, reboot and log in.

If you are installing Ubuntu 22.04LTS as the host operating system on a dedicated machine, continue by following the instructions in the section **Deploy BitCurator in Ubuntu 22.04LTS.**

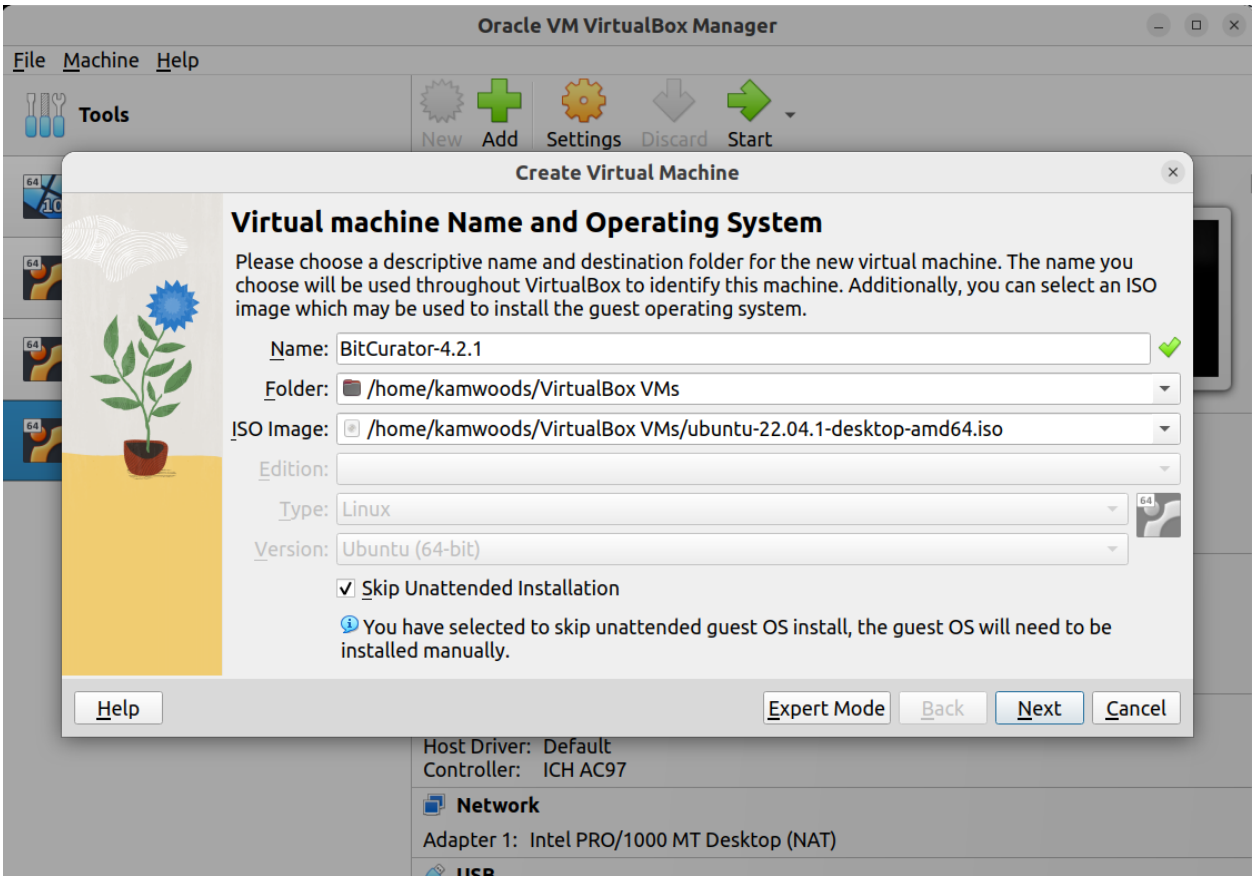
If you are installing Ubuntu 22.04LTS in a VM, see the section **Important Notes for VirtualBox VM Installations**

1. BitCurator may also be deployed in Ubuntu 20.04, using the same instructions in the following sections. We recommend that you only deploy in 20.04LTS if you have a specific compatibility requirement with an existing installation, or specific tools that are unavailable for or incompatible with 22.04LTS.

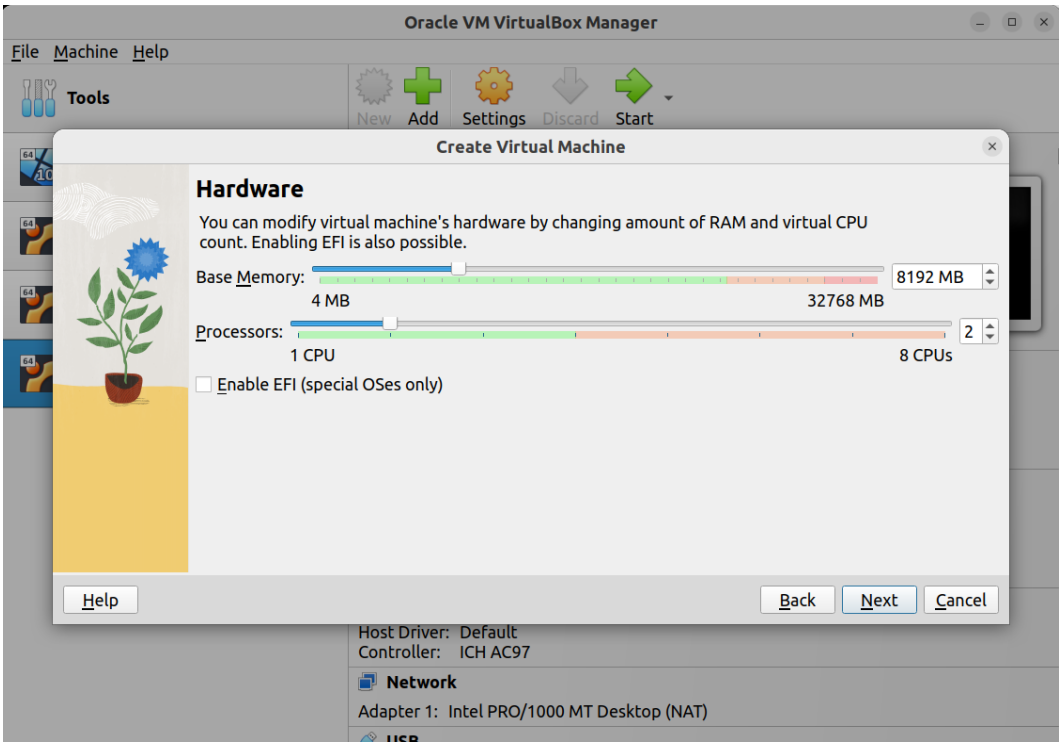
Important Notes For VirtualBox VM Installations

Visit the VirtualBox website to download and install the latest version of VirtualBox for your machine. You **must** also install the **VirtualBox Extension Pack** on your host machine in order for USB 3.0 support to work.

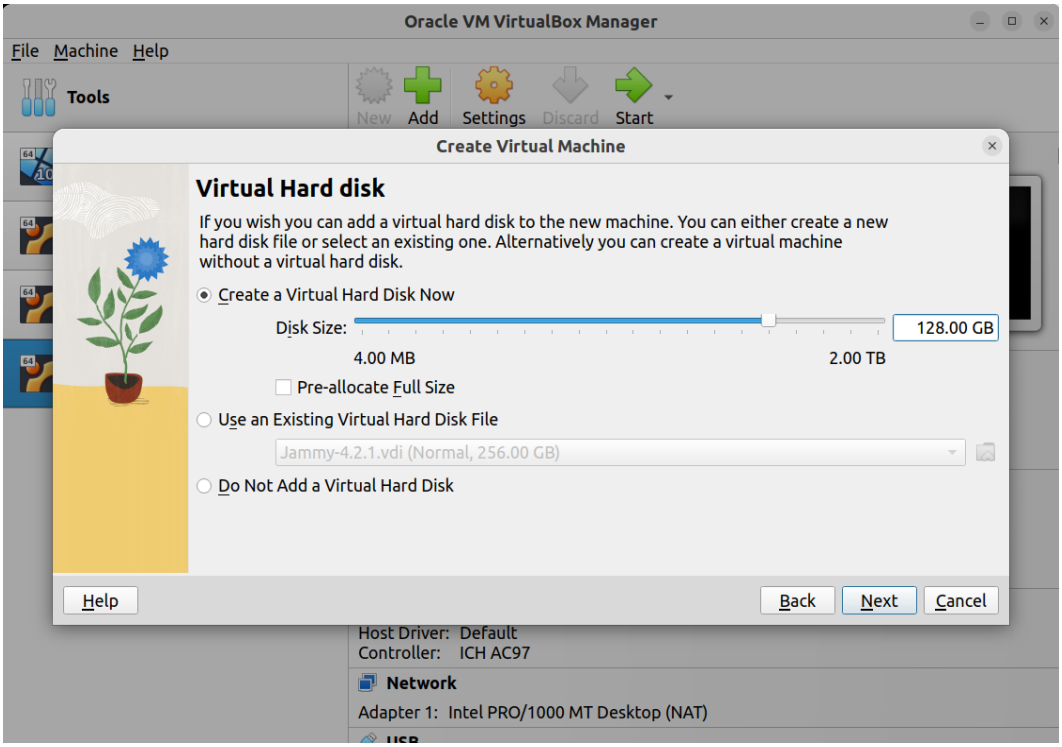
Note that installation options have changed in VirtualBox 7.x and later. In the VirtualBox VM Manager, after clicking **New** to create a new VM, you must specify the name for your VM, the location of the Ubuntu ISO you downloaded earlier, and check the box for **Skip Unattended Installation**.



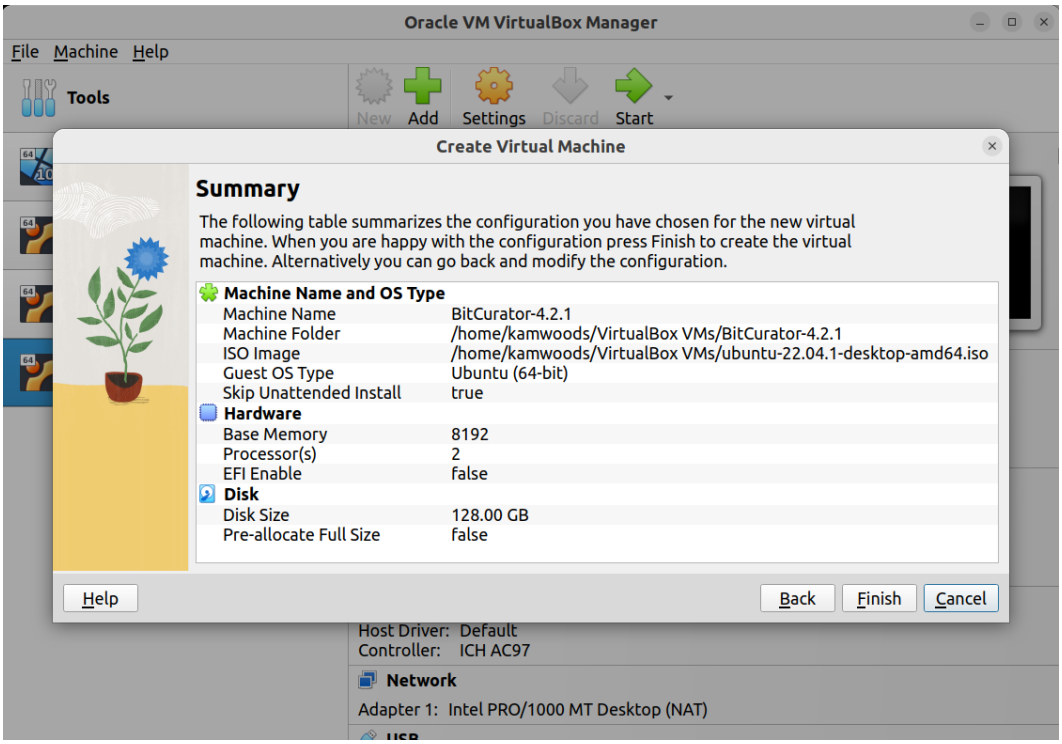
Click **Next**, and on the following screen select at least 8GB RAM and 2 processor cores:



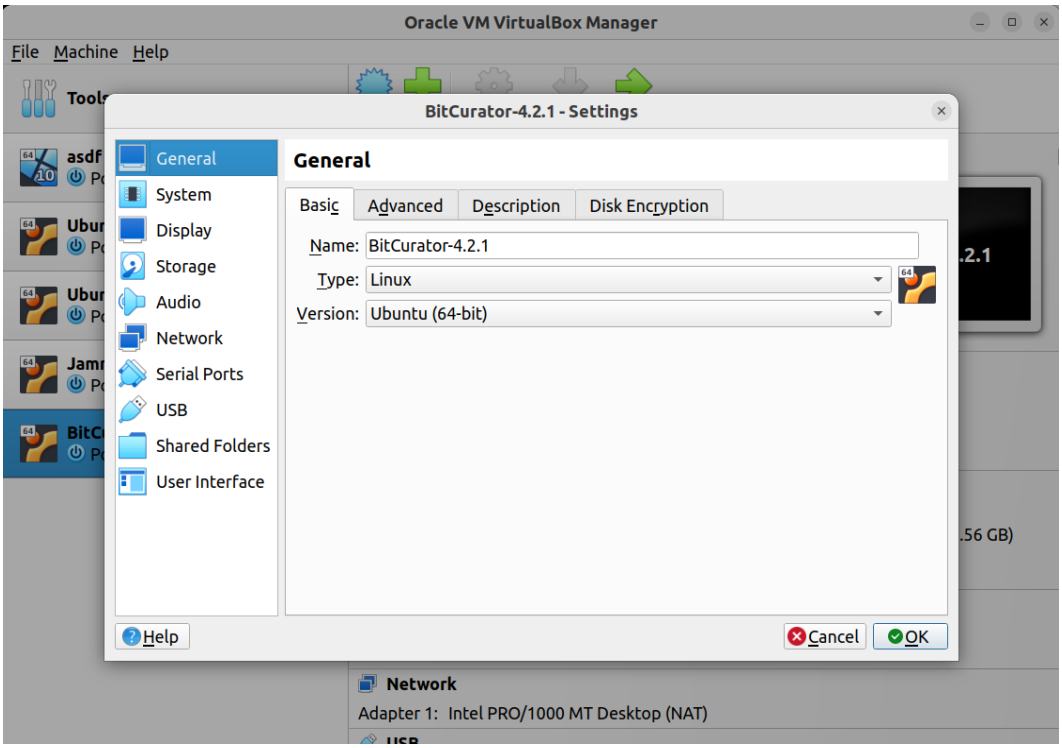
Click **Next** and set the size of the virtual disk. We recommend at least **128GB:**



Click **Next**, and review the details:



Now, click **Finish**, and you will be returned to the main management interface. With your new image selected, click **Settings:**



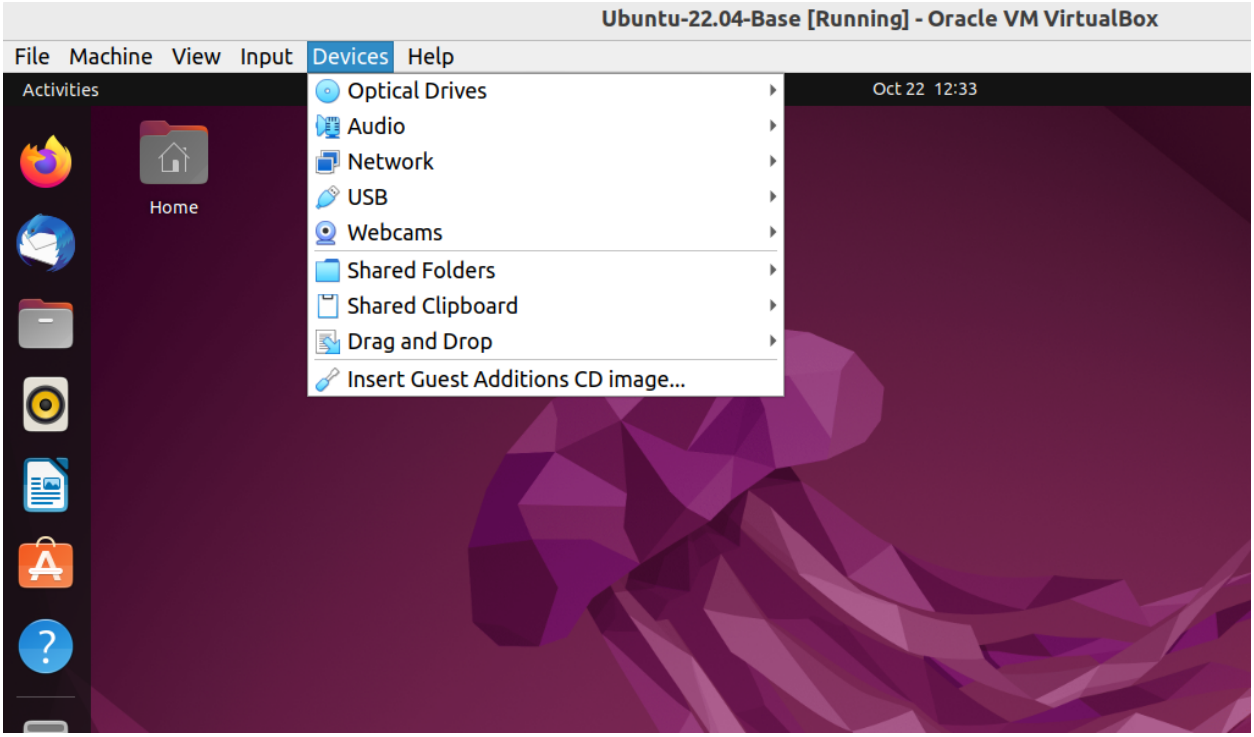
In **Settings**, you will need to make several modifications:

- In the **General -> Advanced** tab, change **Shared Clipboard** to **Bidirectional** and **Drag 'n Drop** to **Host to Guest.**
- In the **Display -> Screen** tab, increase the **Video Memory** to **128MB**
- In **USB**, ensure that **Enable USB Controller** is checked, select the **USB 3.0 (xHCI) Controller,** and click the **small blue USB plug** icon on the right hand side of the dialog to create a new USB filter. Click **OK**.

With the machine selected, click **Start**, and proceed with the Ubuntu installation as normal.

Once you have completed installing Ubuntu in a VirtualBox virtual machine, it's a good time to **install the VirtualBox guest additions**, since both media automounts and software autorun will be disabled after deploying the BitCurator install script in the next section.

With the machine powered up and logged in, select **Insert Guest Additions CD Image** from the **Devices** menu:



Now, click the **Files** icon in the dock, and you should see the Guest Additions CD listed on the left of the new window. Click it, and in the new dialog select **Run Software.**

Important: If you **have trouble** getting the VirtualBox guest additions automatic installer to run, you can also execute the following command from a terminal with the guest additions ISO mounted

(replacing X.X.X with the current release number of the guest additions you are using; this number will match the release number of VirtualBox itself):

```
sudo /media/bcadmin/VBox_GAs_X.X.X/VBoxLinuxAdditions.run
```

(**Note:** as you are typing, you can hit the **Tab** key on your keyboard to autocomplete each part of the command after typing a few letters)

If you plan to use Shared Folders, you should also add the **bcadmin** user to the **vboxsf** group in Ubuntu (after the guest additions have been installed). To do this:

1. Install the **gnome-system-tools** package in order to access the Users and Groups GUI. In a terminal, run the following command:

   ```
   sudo apt-get install gnome-system-tools
   ```

2. Click on the **Show Applications** icon (grid icon) at the bottom left of the screen. then type “Users and Groups”. Click on the icon that appears in the search results.

3. Click **Manage Groups** for the **bcadmin** user

4. Scroll down the groups list until you see **vboxsf**. Select it and click **Properties**,

5. Enable the checkbox.

6. Click **Ok**. You will be prompted for the password for the bcadmin user. Once you’ve completed this step, you can close the Users and Groups window.

Continue by following the instructions in the section **New BitCurator Installations (Step 2): Deploy BitCurator in Ubuntu 22.04LTS.**

New BitCurator Installations (Step 2): Deploy BitCurator in Ubuntu 22.04LTS

1. Prepare your environment

To ensure you have all of the tools, and updates necessary for the BitCurator environment to succeed, you should update the local `apt` repository and install the necessary tools. Open a terminal and use the following commands:

```
sudo apt-get update && sudo apt-get upgrade -y
sudo apt-get install gnupg curl git -y
```

`gnupg` is required for the BitCurator installer to validate the signature of the BitCurrator configuration files during install. `curl` can be used when developing or testing state files. `git` is used to clone local GitHub repos, and can be used when testing state files from the BitCurator SaltStack Repo.

2. Download the BitCurator CLI installer

BitCurator uses a standalone command-line tool for installation and upgrade. First, **open a terminal** and download the latest release of the tool with the following command:

```
wget https://github.com/BitCurator/bitcurator-cli/releases/download/v1.0.0/bitcurator-cli-linux
```

Verify that the SHA-256 has of the downloaded file matches the value below:

```
5acab7abcafa24864d49e4872f8e2b562c16bf4842256ad3f994aae8d0df77c1
```

You can generate the hash of your downloaded file with:

```
sha256sum bitcurator-cli-linux
```

Next, adjust some permissions and move the BitCurator installer to the correct location:

```
sudo mv bitcurator-cli-linux /usr/local/bin/bitcurator
sudo chmod +x /usr/local/bin/bitcurator
```

(Instructions continue on next page)

3. Run the BitCurator CLI Installer

Simple install: Run the BitCurator installer. This may up to an hour to complete, depending on your system and network speeds:

```
sudo bitcurator install
```

The command will install BitCurator for the user that is currently logged in. For other installation options, read the section below.

Optional install modes:

The BitCurator installer provides several additional installation modes. To see the options provided by the command, run the following:

```
bitcurator --help
```

You will be presented with the usage information:

```
Usage:
  bitcurator [options] list-upgrades [--pre-release]
  bitcurator [options] install [--pre-release] [--version=<version>] [--mode=<mode>]
[--user=<user>]
  bitcurator [options] update
  bitcurator [options] upgrade [--pre-release] [--mode=<mode>] [--user=<user>]
  bitcurator [options] version
  bitcurator [options] debug
  bitcurator -h | --help | -v

Options:
  --dev                 Developer Mode (do not use, dangerous, bypasses checks)
  --version=<version>   Specific version install [default: latest]
  --mode=<mode>         bitcurator installation mode (dedicated or addon, default: dedicated)
  --user=<user>         User used for bitcurator configuration [default: bcadmin]
  --no-cache            Ignore the cache, always download the release files
  --verbose             Display verbose logging
```

If you wish to install BitCurator for a different user, you may use the following command:

```
sudo bitcurator install --user=<user>
```

Where <user> is a username other than the one you are currently logged in as.

If you wish to install BitCurator in **addon mode** (installing the tools, but leaving the default Ubuntu theme intact), use the following command:

```
sudo bitcurator install --addon
```

3.1 What to do if something goes wrong

If you encounter an error, you may be able to identify the issue by reviewing the **saltstack.log** file under **/var/cache/bitcurator/cli** in the subdirectory that matches the BitCurator state-files version you're installing. Search for the log file for **“result: false”** messages and look at the surrounding 5 lines or the 8 lines above each message to see the state file that caused the issue. You can do this with:

```
grep -i -C 5 'result: false' or grep -i -B 8 'result: false'
```

4. Reboot

When the installation is complete, reboot your system from the terminal:

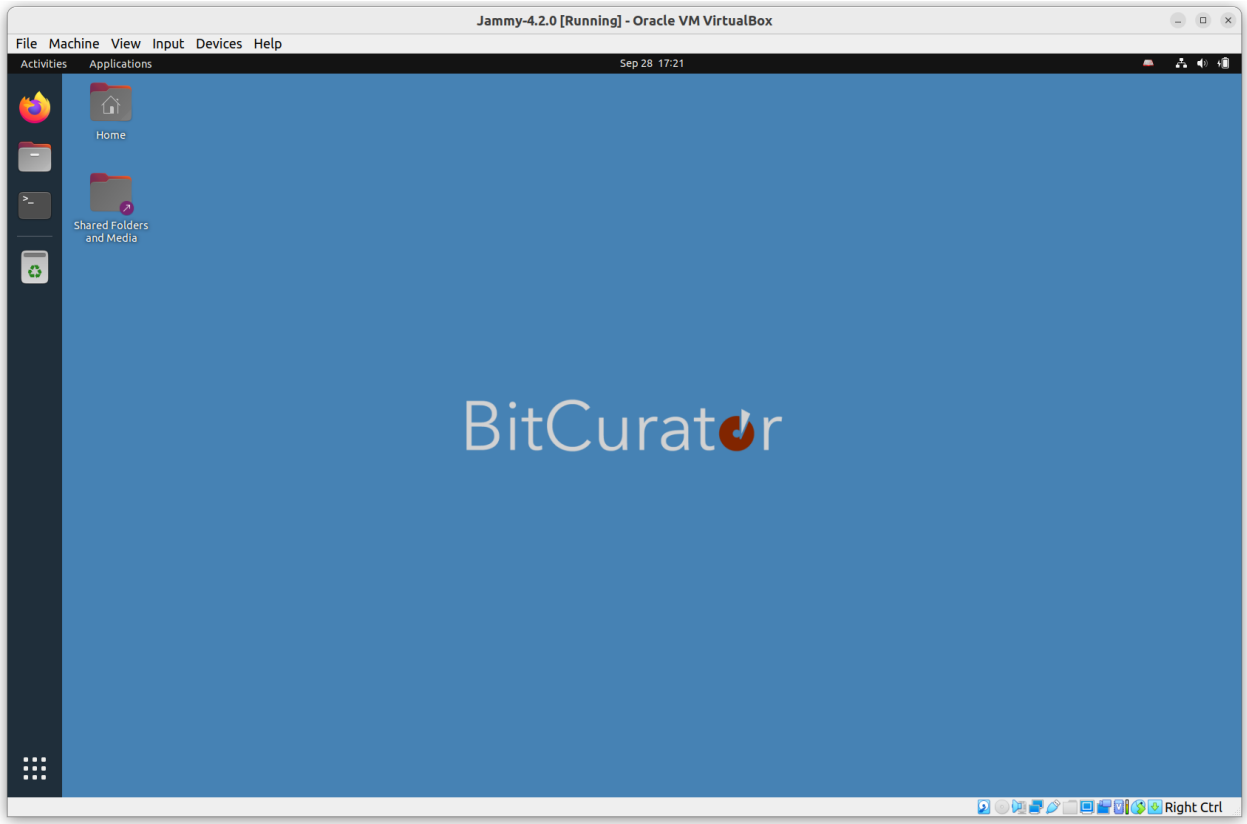
```
sudo reboot
```

After the reboot, you will be automatically logged in if you selected “Automatic Login” during the Ubuntu install. Otherwise, you can log in to BitCurator with the username and password used during the install. To change this behavior:

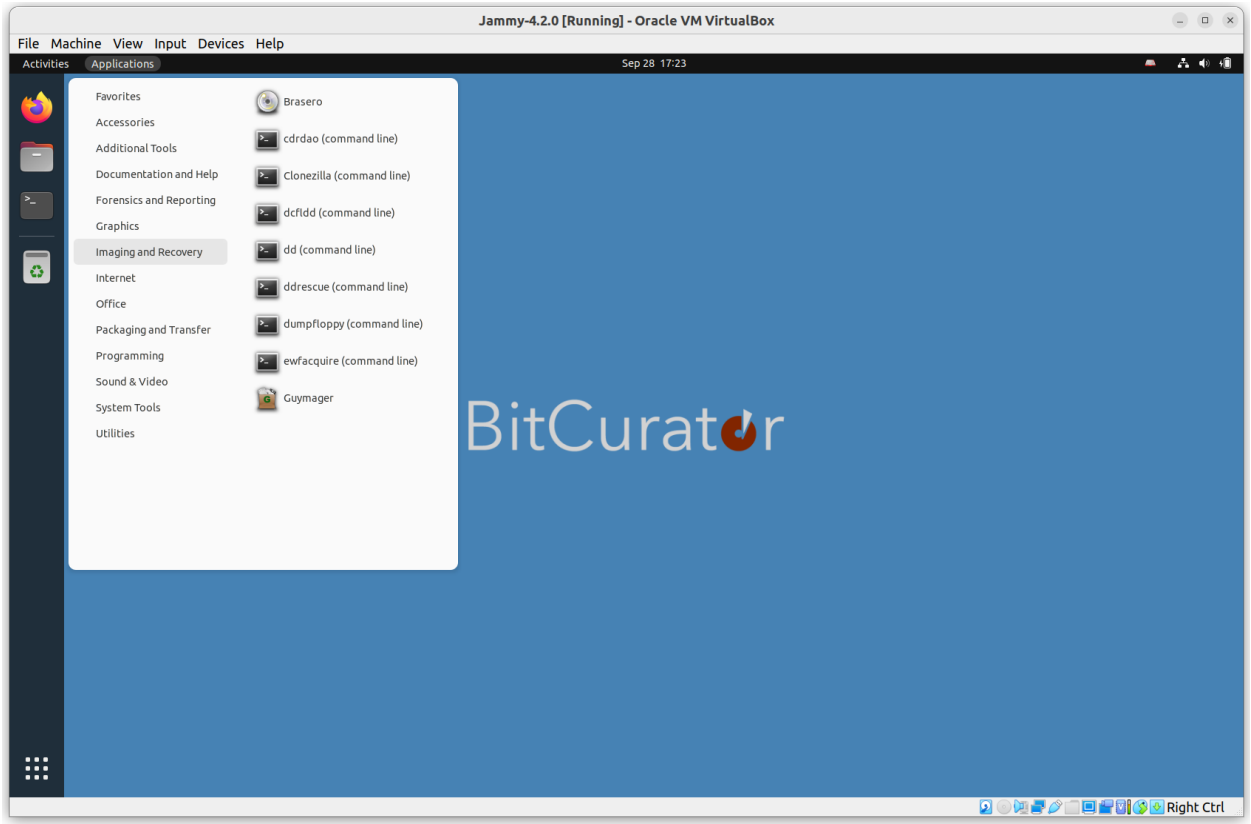
1. Open the Activities overview and start typing Users.
2. Click Users to open the panel.
3. Select the user account that you want to set the log in behavior for at startup.
4. Press Unlock in the top right corner and type in your password when prompted.
5. Switch the Automatic Login switch to on or off, depending on the desired behavior.

Working in the BitCurator Environment

The BitCurator Environment includes various customizations and tools to assist in mounting and analyzing digital media and their contents. When you log in, you should see a desktop similar to the one shown below:

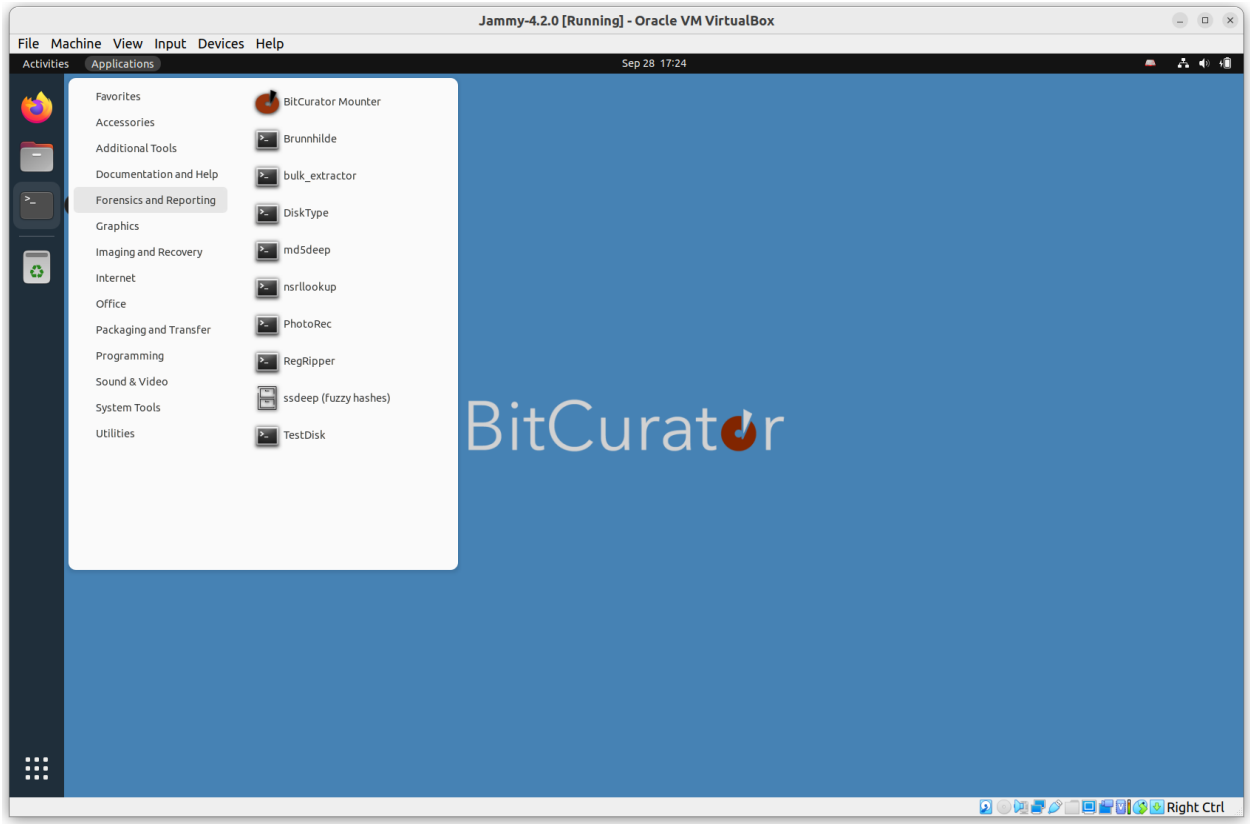


In previous releases, tools to support a variety of workflows were organized into folders on the desktop. In BitCurator 4.x, these tools are organized in submenus under the **Applications** menu in the top left (to the right of the **Activities** menu).



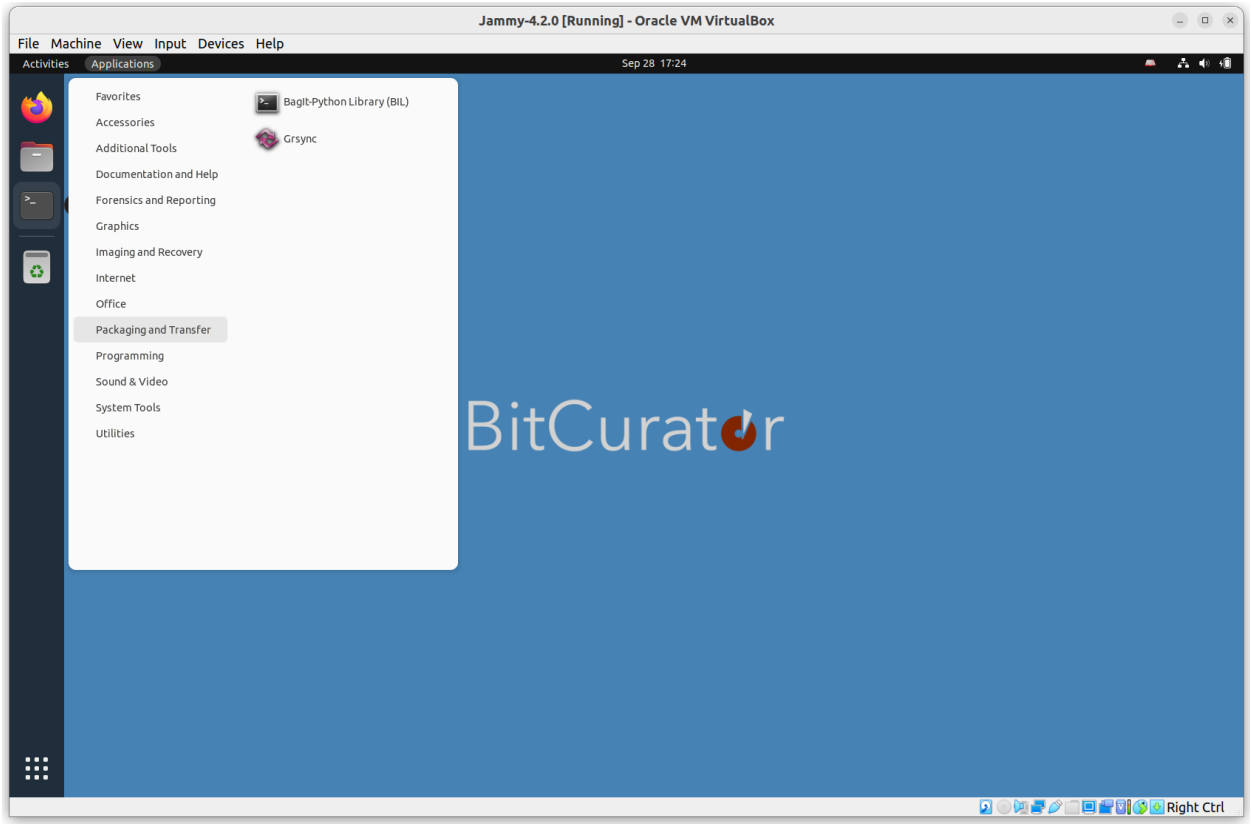
The **Imaging and Recovery** submenu lists tools to assist in device imaging and recovery. These include:

Brasero: A GNOME application to burn CDs and DVDs, and create 1:1 copies of CDs and DVDs.
cdrdao: An application to record audio or data CD-Rs in disk-at-once (DAO) mode based on a textual description of the CD contents (*toc-file*).
Clonezilla: A partition and disk imaging/cloning program.
dcfldd: An enhanced version of GNU dd with features useful for forensics and security, including hashing on-the-fly and split outputs.
dd: dd copies input to output with a changeable I/O block size, while optionally performing conversions on the data. Can be used to create raw images of devices.
ddrescue: A data recovery tool with advanced features to assist in rescue of good data from devices with read errors.
dumpfloppy: A tool to read floppy disks in arbitrary formats supported by the PC floppy controller, and work with the resulting image files.
ewfacquire: A utility to acquire media data from a source and store it in EWF format (Expert Witness Compression Format).
Guymager: A GUI tool to create raw and EWF-packaged images from devices.



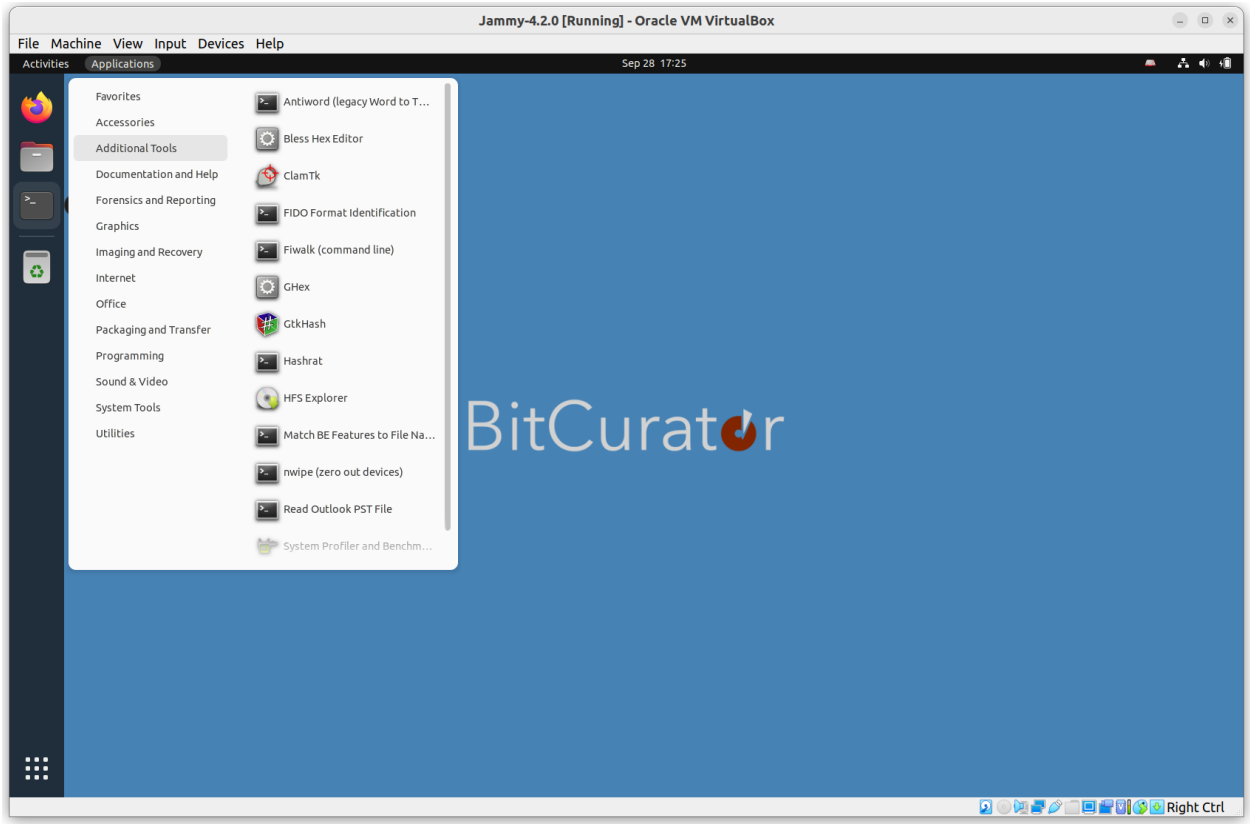
The **Forensics and Reporting** submenu lists forensics and forensics-adjacent tools. These include:

BitCurator Mounter: A lightweight GUI tool to assist with mounting and unmounting devices.
Brunnhilde: A tool to generate aggregate reports of files in a directory or disk image based on input from Richard Lehane's Siegfried. Can optionally analyze content using bulk_extractor.
Bulk Reviewer: A tool to scan disk images and assist in the review of bulk_extractor reports.
bulk_extractor: A tool to scan disk images and directories for PII and other features.
DiskType: A tool to detect the content format of a disk or disk image. It knows about common file systems, partition tables, and boot codes.
md5deep: a set of programs to compute MD5, SHA-1, SHA-256 and other digests
nsrllookup: Query NSRL's MD5 hashes of known pieces of software.
PhotoRec: File data recovery software designed to recover lost files including video, documents and archives from media.
RegRipper: Extract the contents of Windows registry backups.
TestDisk: Data recovery software, companion to PhotoRec.



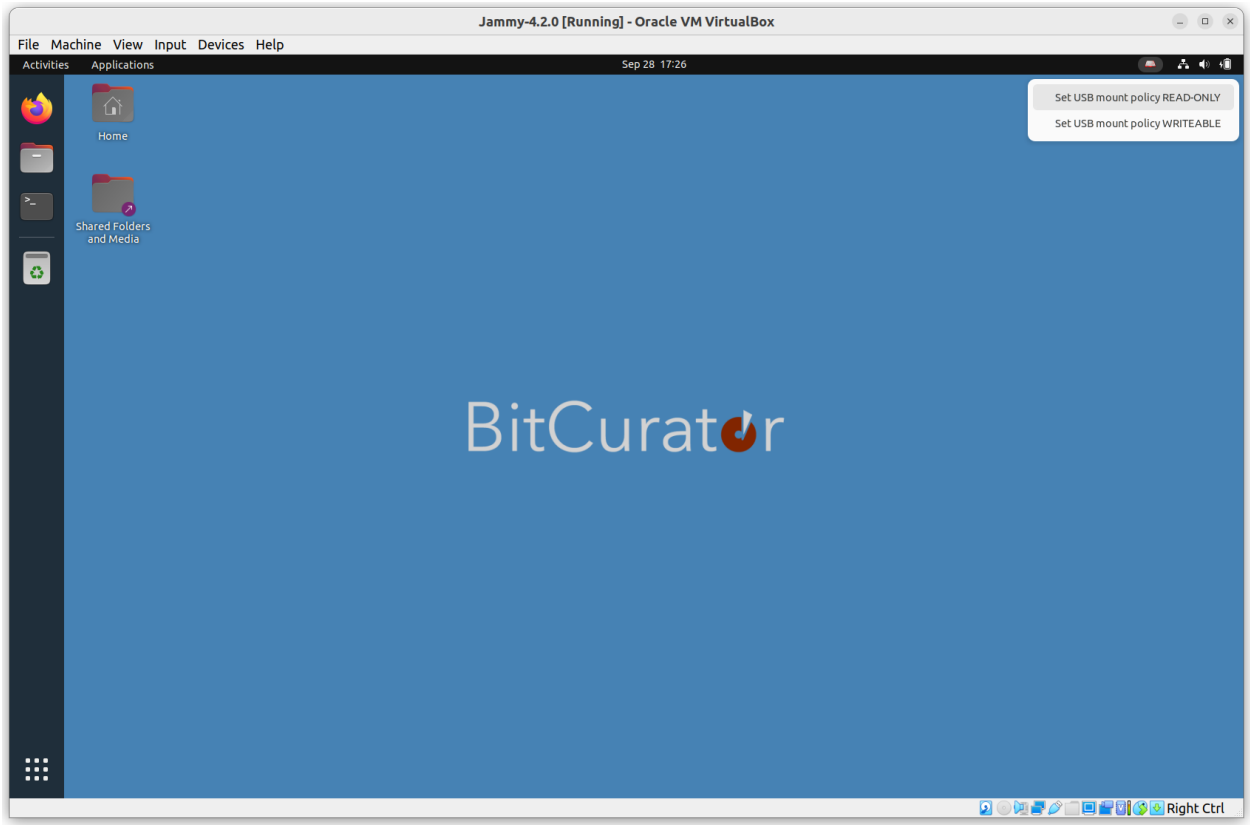
The **Packaging and Transfer** submenu lists some tools to assist with safe packaging and network transfer of materials. These include:

Bagit-Python: A Python library and command line utility for working with BagIt style packages.
Grsync: A GUI front-end for rsync.



The **Additional Tools** submenu includes a selection of extra tools that may be useful in various workflows. Full documentation for these can be found elsewhere online.

Antiword: Extract the contents of legacy Word (binary) documents.
Bless Hex Editor and **GHex**: Hex editors to inspect and edit hexadecimal representations of files and devices.
Fiwalk: Command line tool to process disk images using Sleuthkit. Outputs machine-readable metadata in Digital Forensics XML.
HFSExplorer: A tool to read and export files from Apple HFS file systems.
ClamTk: A GUI front end for the scriptable Clam Antivirus software.
FIDO Format Identification: The FIDO (Format Identification of Digital Objects) command line tool. Uses the PRONOM file format and container descriptions.
nwipe: Securely zero out (wipe) devices.



BitCurator includes a convenience application indicator (small disk icon in the top right of the menu bar) that may be used to quickly set the current mount policy for USB Devices.

When the disk icon is red, the mount policy is set “writeable”. USB storage devices plugged into the system when this state is set can be read from, and written to.

When the disk icon is green, the mount policy is set “read-only”. USB storage devices plugged into the system when this state is set cannot be written to.

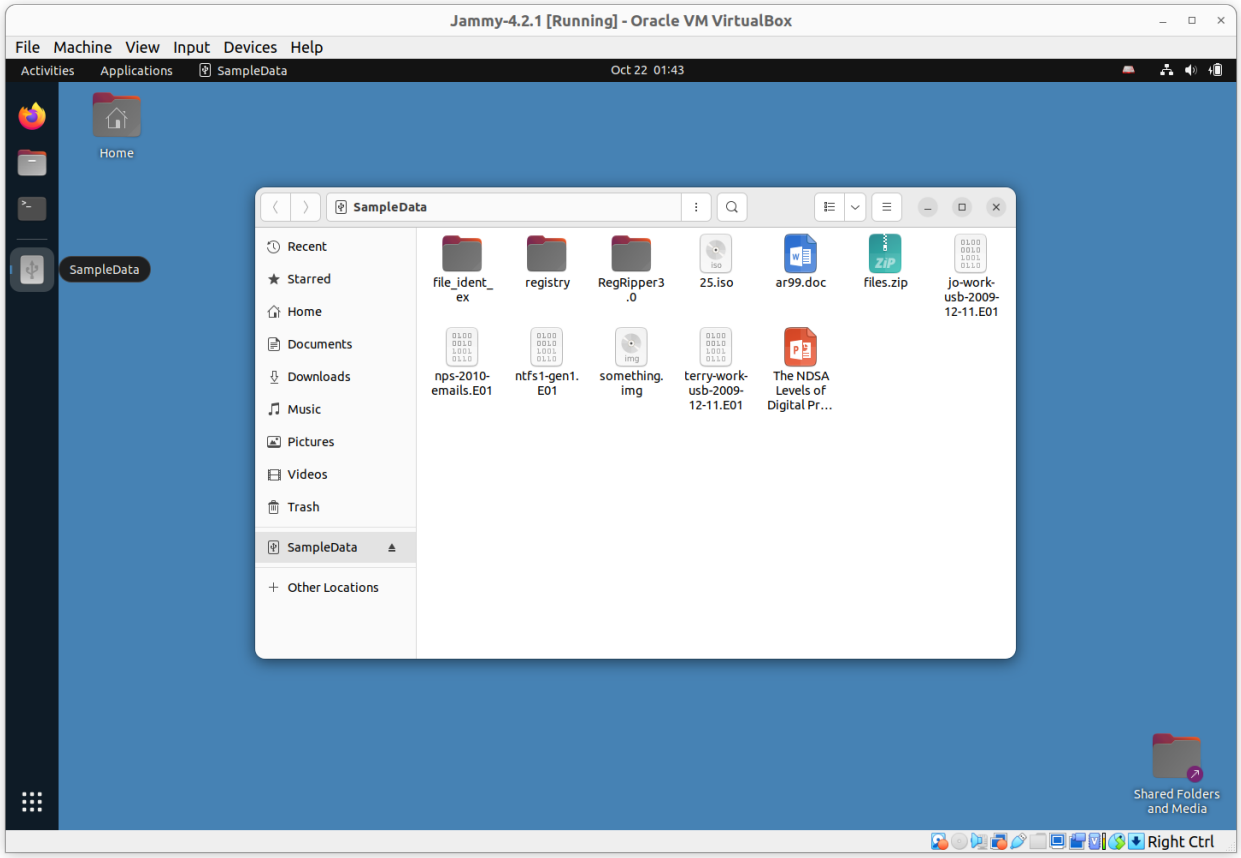
Important notes: Changing the mount policy state **does not** change the mount policy for devices currently plugged into your machine.

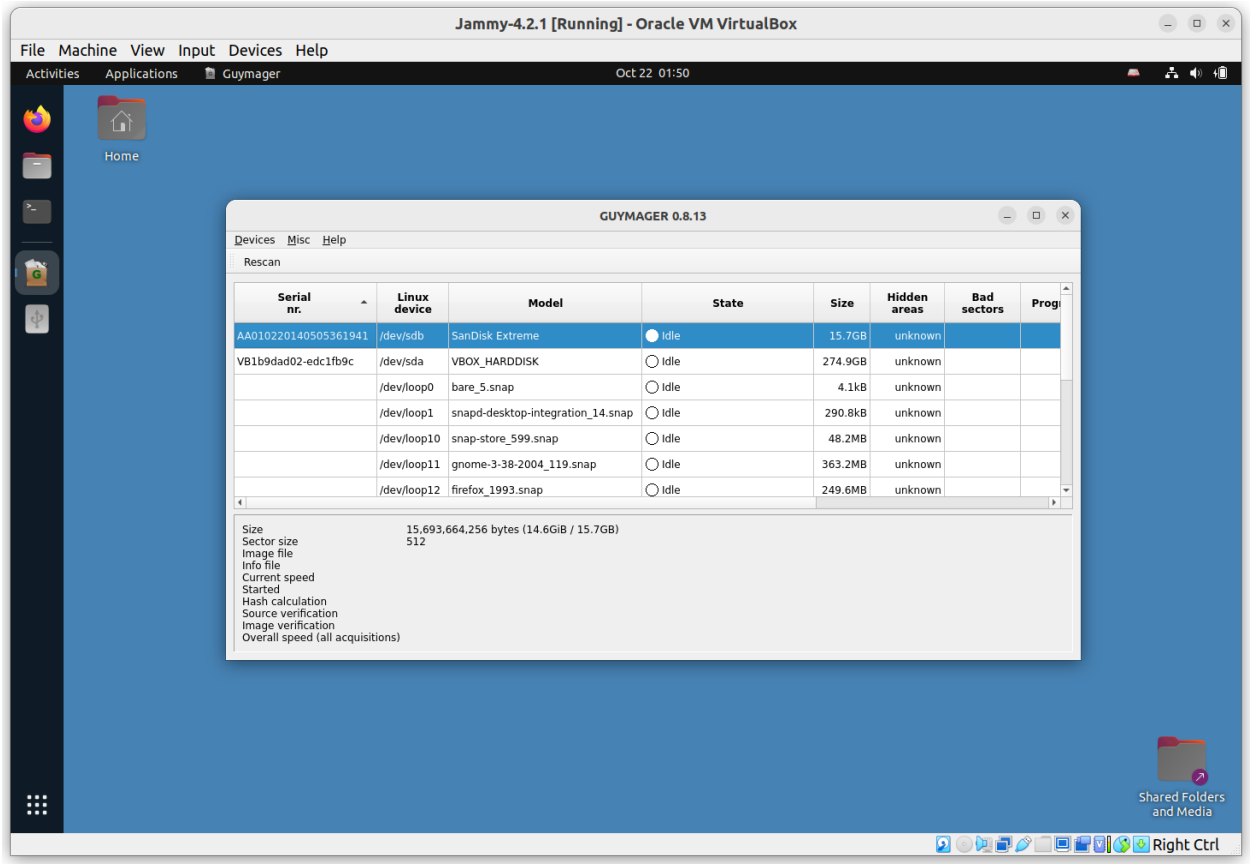
Imaging Physical Media

In this example, we will acquire a USB flash drive connected to a laptop running the BitCurator environment in a VirtualBox VM. The process is similar for other devices.

When properly configured (with VirtualBox Guest Additions installed and a USB Device Filter enabled for all USB devices in the VirtualBox settings for that machine), the USB device should be automatically captured by the VM. However, **it will not be automatically mounted.**

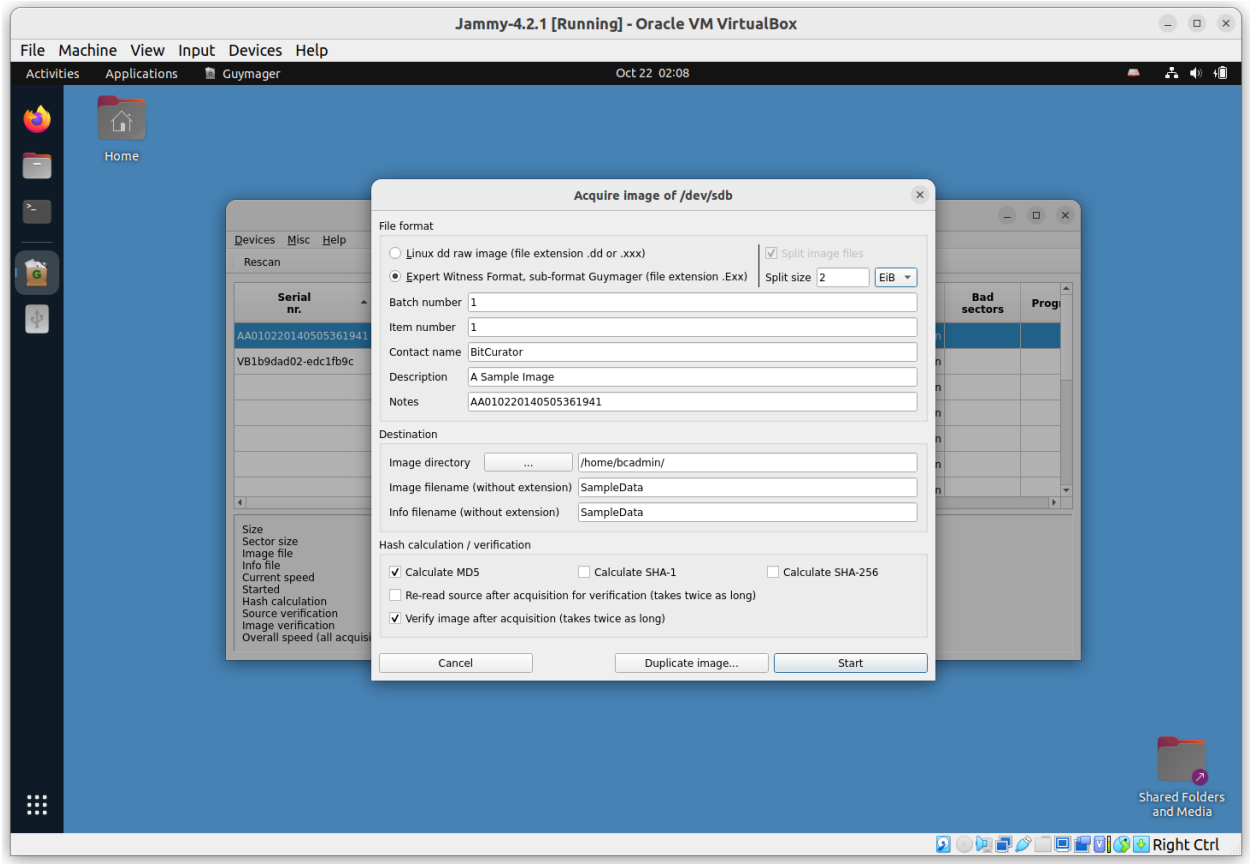
Mounting the device is not required to create an image of it. If you wish to mount the device, click on the **Files** icon in the dock, and select the name of the indicated volume on the device to mount. **If you are not using a hardware write blocker, or if the USB device read-only policy is not enabled, your device is now mounted and writable!**





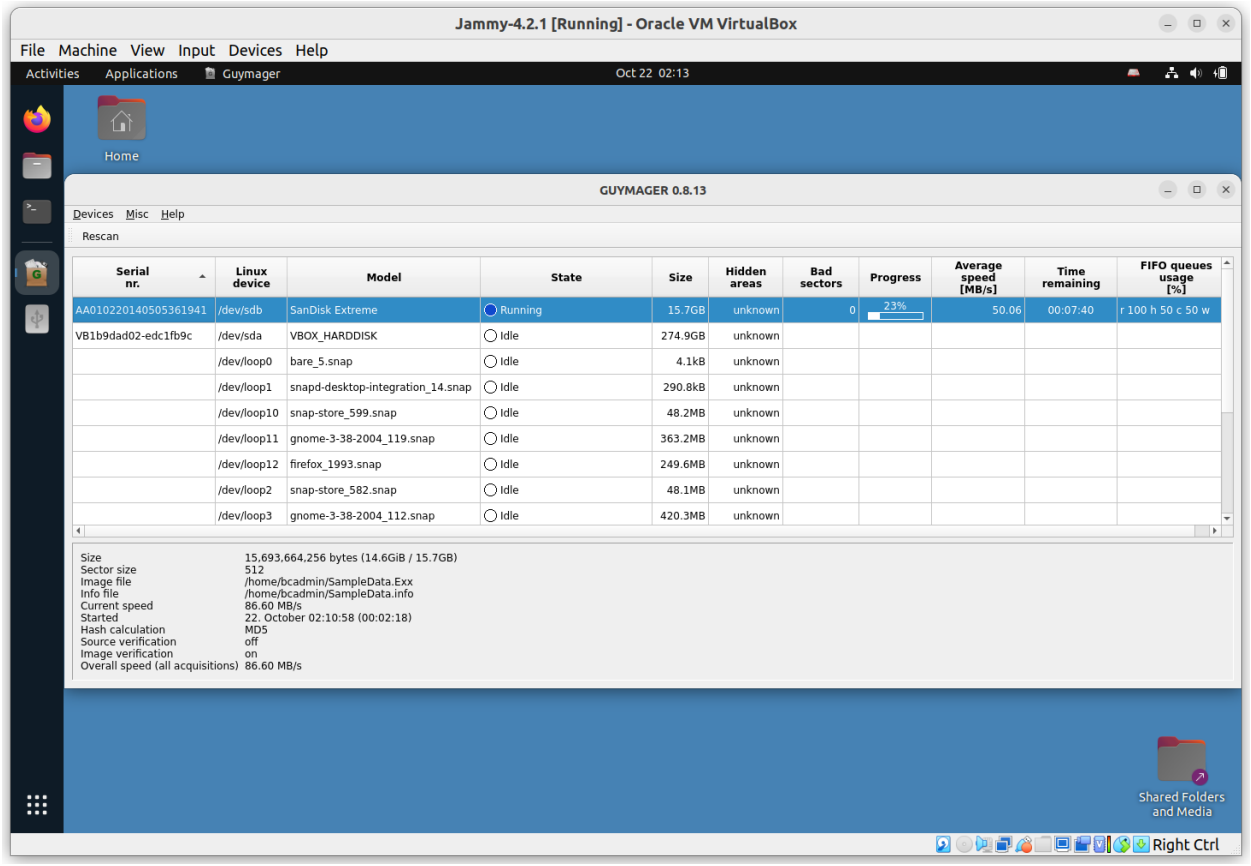
Click on the **Applications** menu in the top left of the screen, then navigate to the Imaging and Recovery submenu. Then click on **Guymager**. Guymager requires elevated privileges for access to physical devices; you will be prompted for your password to enable this. Once Guymager has loaded, the main interface appears as in the picture above. In this example, the USB flash drive is selected.

Next, right-click on the selected device (in this example, a 16GB SanDisk Extreme flash drive) and select **Acquire Image** from the context menu. A new dialog will appear:



This disk image will be acquired using the Expert Witness Format (the second option at the top). Guymager will split EWF images into 2048MiB segments by default. If you do not wish to split the image, set the **Split size** to something very large (2 EiB, for example).

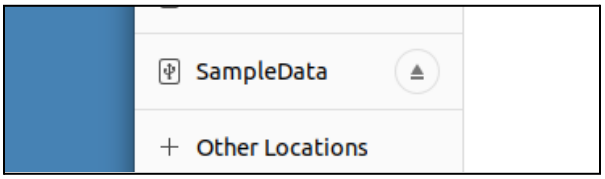
The five metadata fields starting with **Batch number** are optional. Don't forget to select the directory you made on the desktop in the Image directory field. In this case, we have simply chosen to write the image to our home directory. Finally, provide a name for the image. Then click **Start**.



You will see the main dialog state change to **Running**. When the acquisition finishes, you will see a **Finished - Verified & Ok** message in the State column.

Tip: If you're running the BitCurator environment in a VM, the default screen resolution may be small enough that you can't see the whole dialog. Increase the size of your VM window by dragging a side or corner. The desktop should adjust automatically.

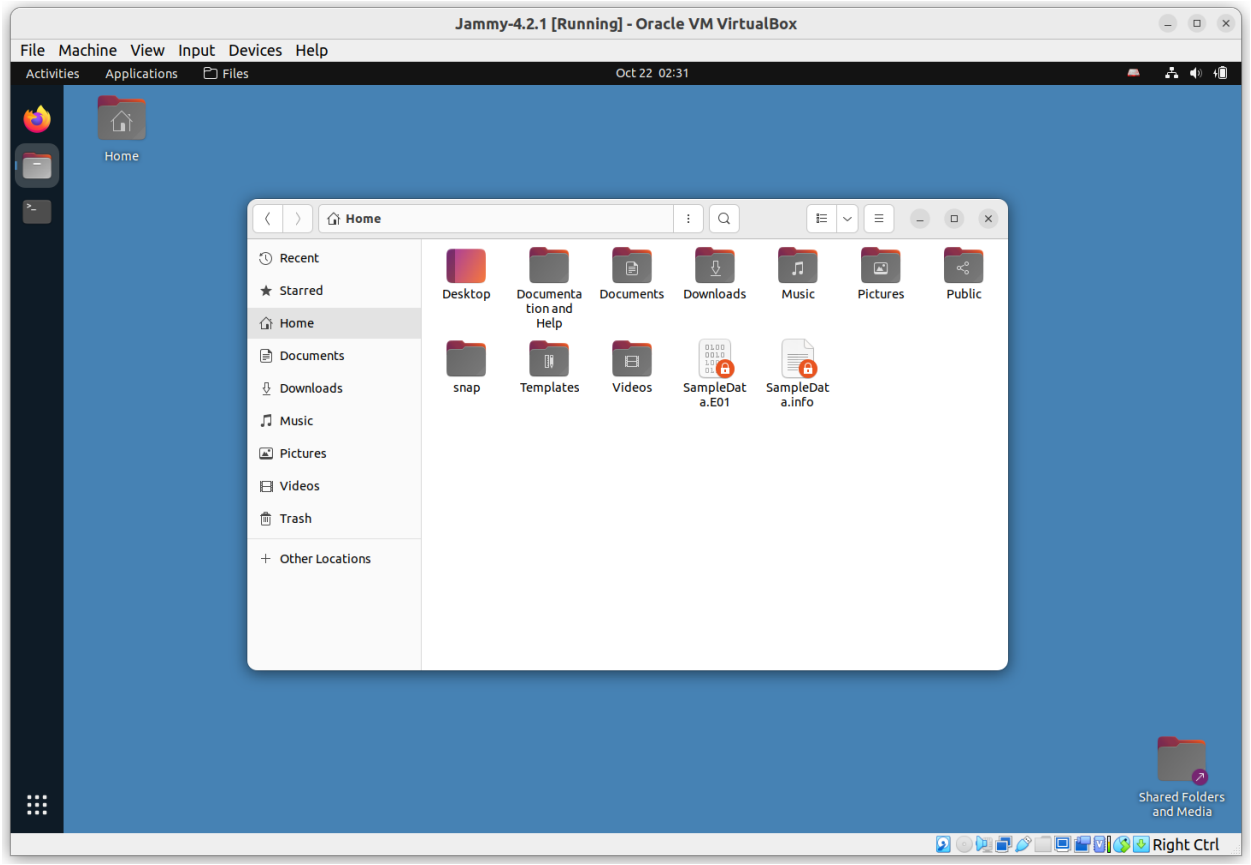
Close **Guymager**, and double-click on the **Home** icon on the Desktop. If your physical device is mounted, it will appear in the window with an eject symbol next to it.



Click the eject symbol, and you should see a notification appear at the top of the screen indicating that the filesystem is being unmounted. Once the name of the filesystem disappears from the list, you can power off and or unplug the physical device.

Changing Permissions for our Disk Image Files

Since Guymager runs with elevated permissions, the images it creates are **read-only** for users other than **root**. Double-click on the **Home** icon on the Desktop. You will see that the image file (and associated metadata file) we have created have small **“locked padlock”** indicators on them, indicating restricted permissions for our user.



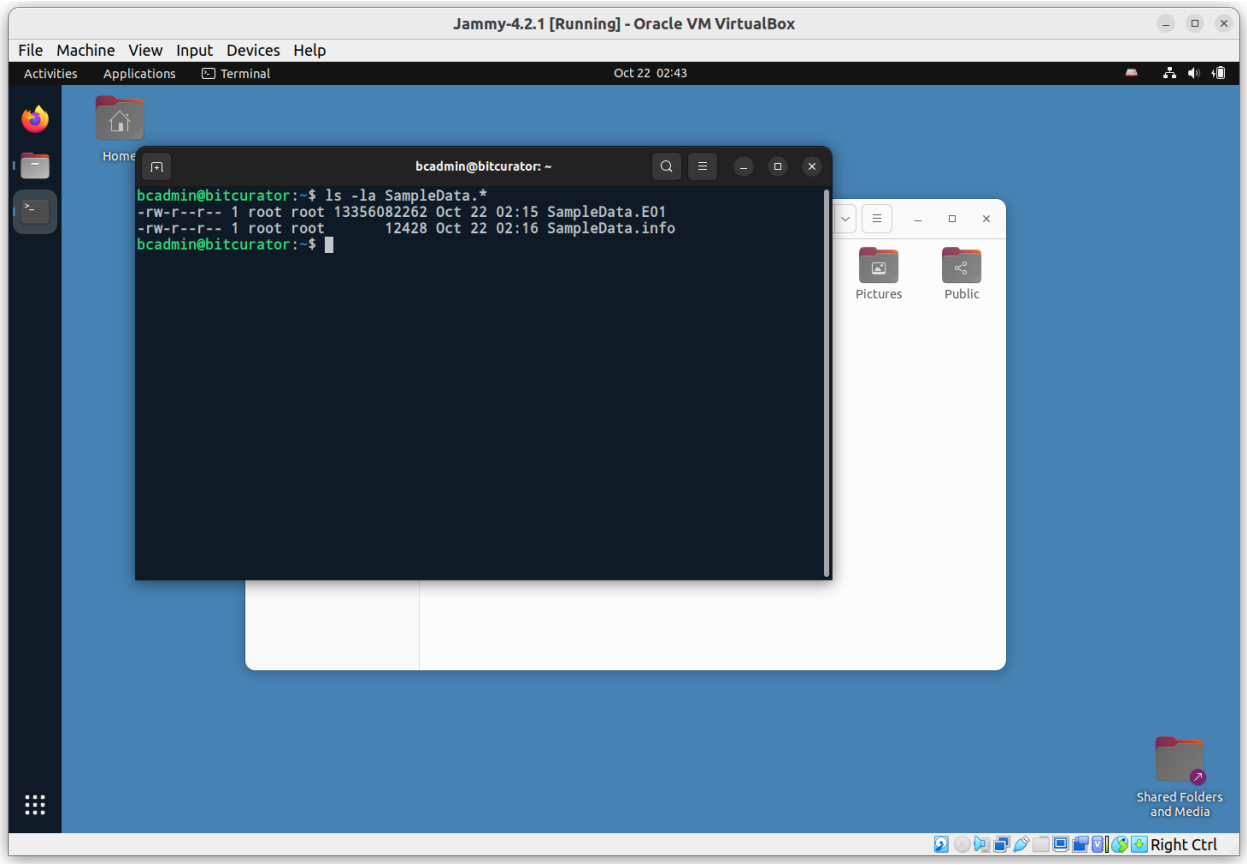
If you’re unfamiliar with how Linux permissions work, now is a good time to read an introductory guide such as the one at https://ubuntu.com/tutorials/command-line-for-beginners.

There are multiple ways to update the permissions on these images so we have full read-write access, but the fastest is with some terminal commands. Right-click in the white space of the window we just opened, and select **Open in Terminal** from the context menu.

In the **Terminal** window that appears, type the following:

```
ls -l SampleData.*
```

This tells the system to list, using the long listing format (-l) files with the name SampleData and any extension.



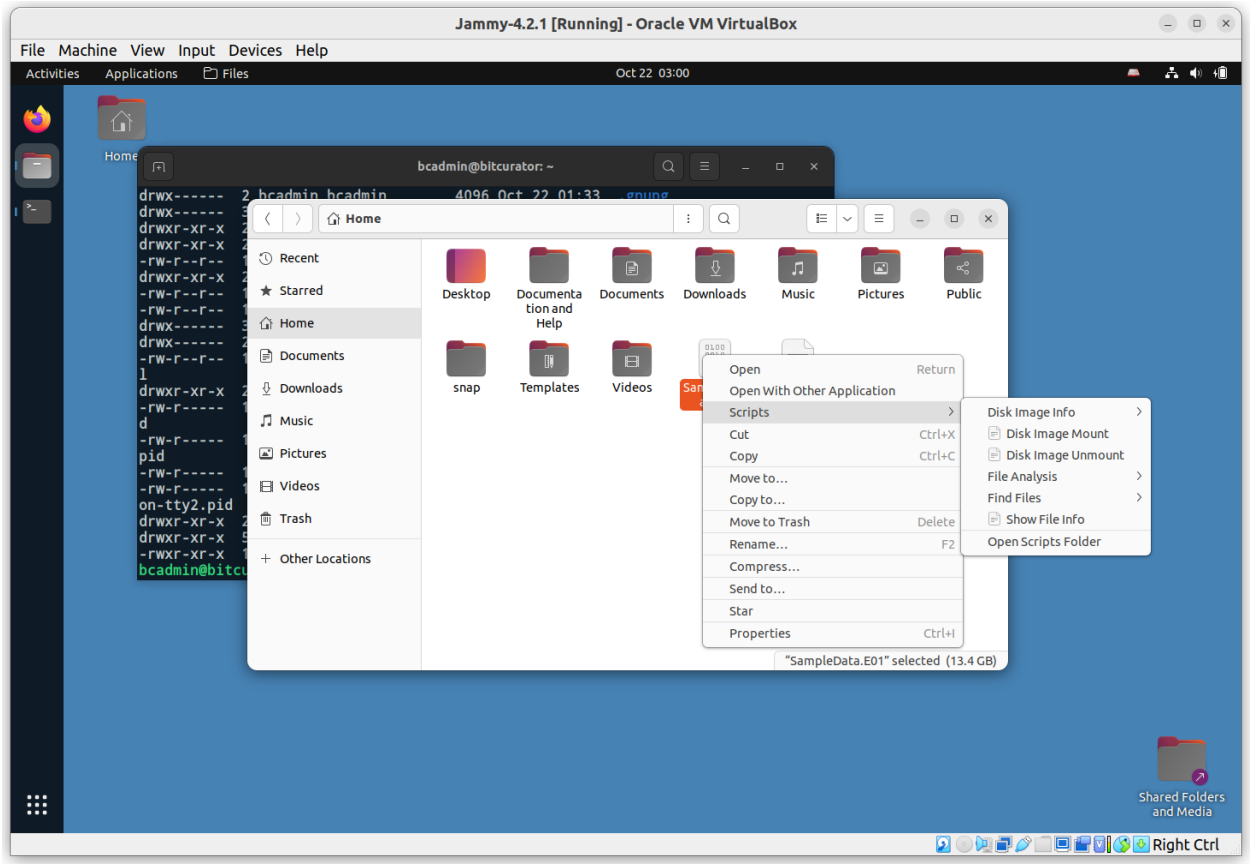
We can see that these two files have read-write permissions for the file owner (root), read-only permissions for the group owner (anyone part of the root group) and read-only permissions for all other users. We could change these permissions so that **everyone** has read-write access, but instead for this example we will be changing the file owner and group owner to our user, bcadmin. To do this, enter the following commands:

```
sudo chown bcadmin SampleData.*
sudo chgrp bcadmin SampleData.*
```

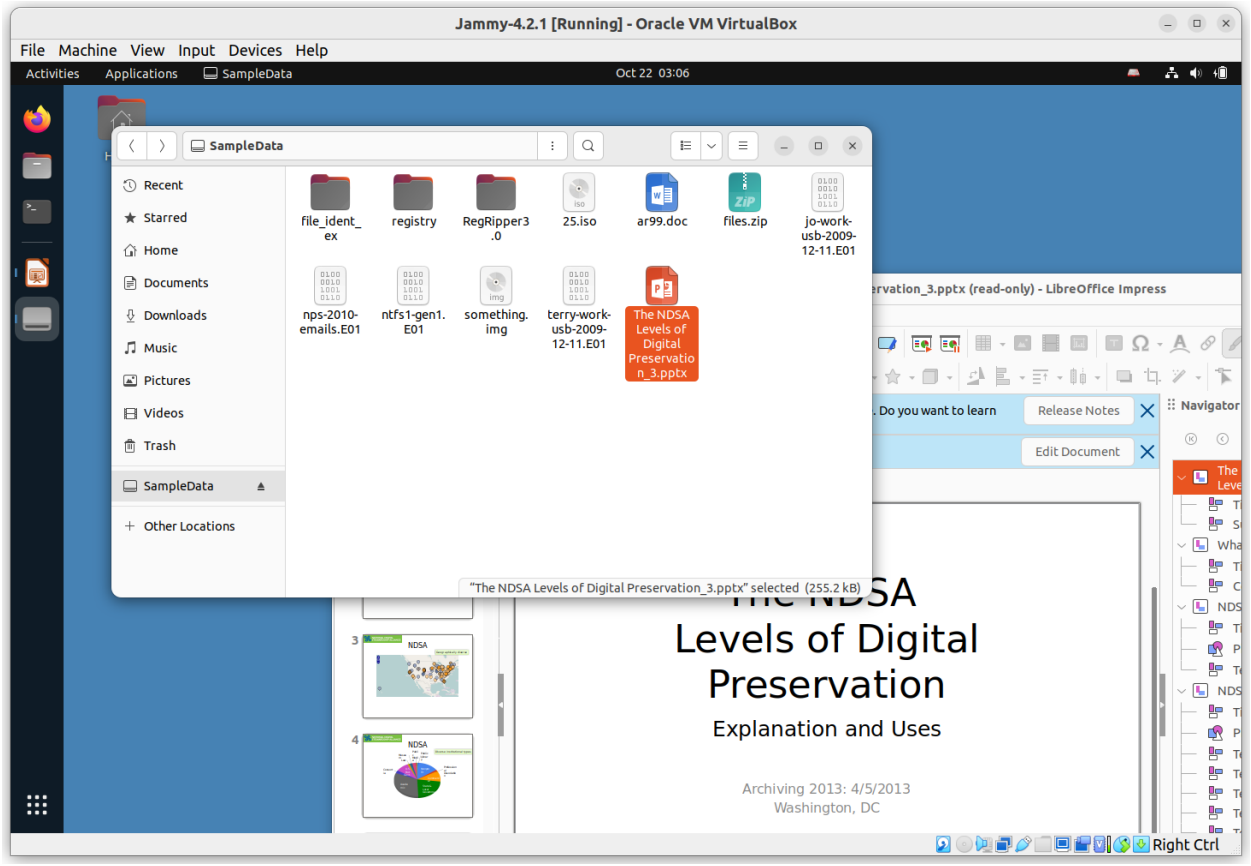
If you run the previous **ls** command again, you'll see that the file owner and group owner for both of the SampleData files is now **bcadmin**. The padlock icons are also now gone in the file manager window. Close the terminal window.

Mounting and Examining a Disk Image

The BitCurator environment includes many different tools to interact with disk images. For **raw** (.dd or .raw) and **Expert Witness Format** (.E01) images that contain filesystems readable by the underlying Ubuntu OS, BitCurator includes some convenience scripts to automatically mount those filesystems.



In the file manager dialog, right click on the **SampleData.E01** image we created, select **Scripts**, and then select **Disk Image Mount**. This script serves as a wrapper for **libewf** and some mounting tools to attempt to automatically mount any identified file systems. If such a filesystem is found, you will see it appear as a mountable device in the list on the left.



In this example, we have mounted the FAT32 filesystem contained within the image of the 16GB drive we imaged, and can now browse, open, and copy any files we require. In this example, we have opened a PPTX file in LibreOffice.

Note: This mount is read-only. You cannot alter the content of a filesystem mounted from an E01 file (modifying, adding new files, or deleting) from this desktop interface.

Once you have finished examining the content, click the **eject** indicator next to the filesystem name in the file dialog. You will get a prompt for your user password in order to complete this step.

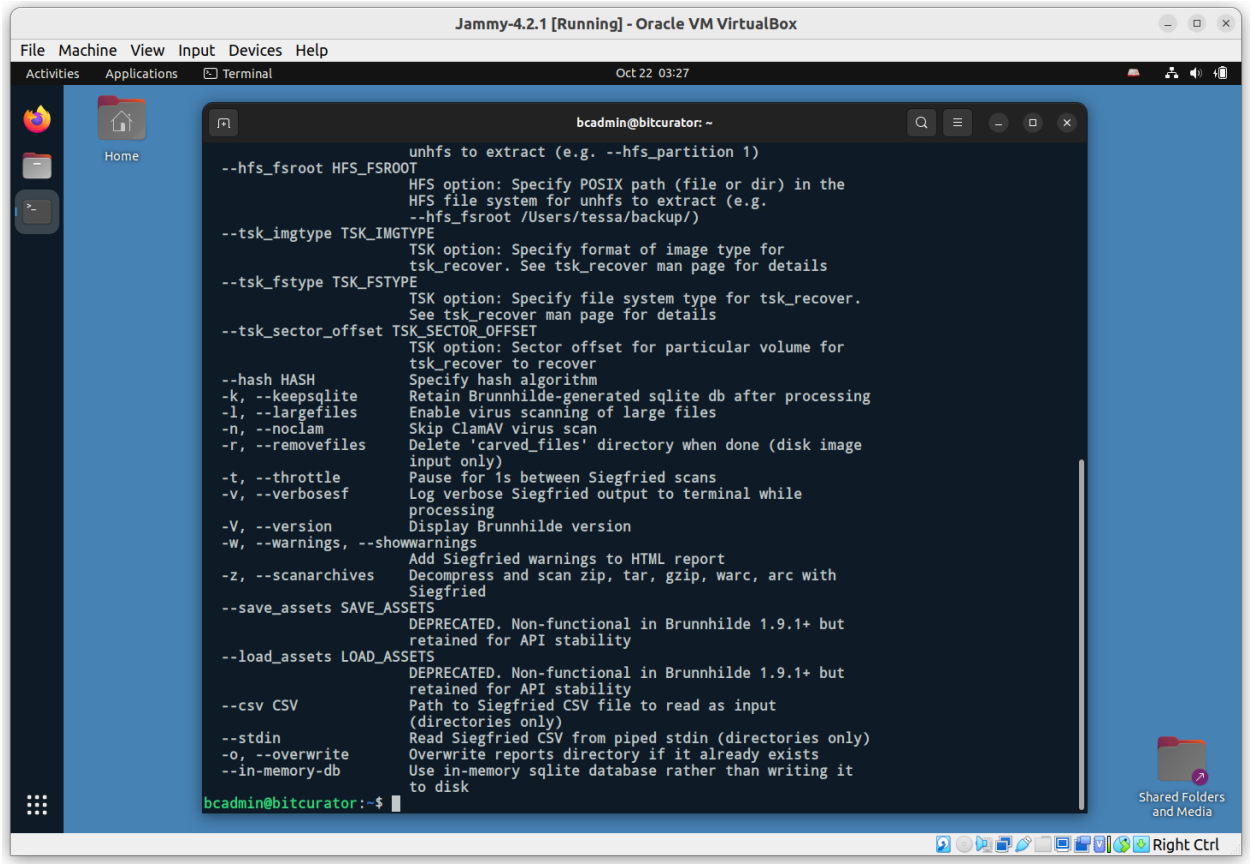
Analyzing a Disk Image with Brunnhilde

BitCurator includes a variety of tools to analyze and report on disk images and the filesystems they contain. For this Quickstart guide, we will focus on the command line tool **Brunnhilde**, a Siegfried-based characterization tool for directories and disk images.

In the **Applications** menu in the top left**,** navigate to the **Forensics and Reporting** submenu, and select **Brunnhilde**. A new terminal window will appear, in which the following command has been run to show Brunnhilde's help documentation:

```
brunnhilde.py -h
```

Your terminal window may be too small to see all of the flags and options in the help output. Make the window larger, or scroll up if you'd like to see the full list of options.



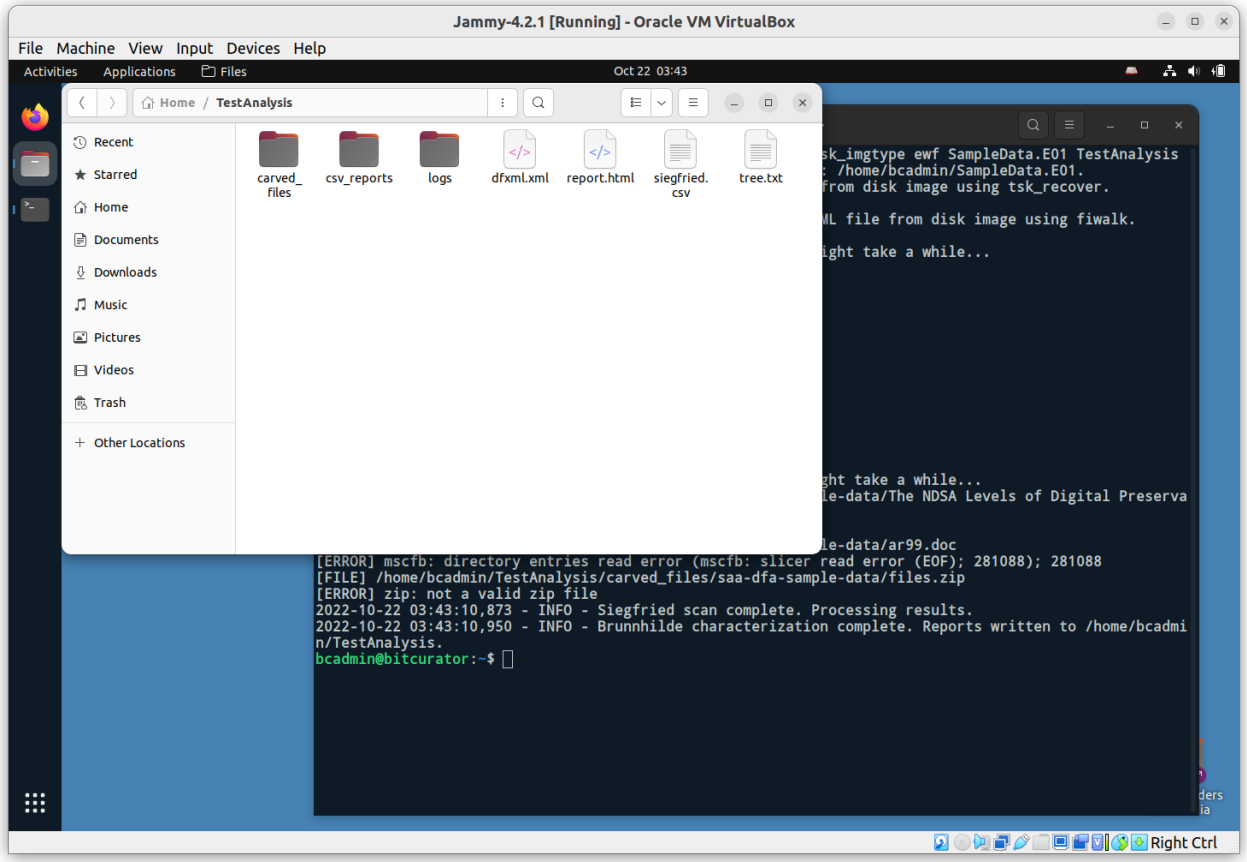
For this example, we'll be instructing Brunnhilde to analyze an **EWF** disk image containing a **FAT** filesystem, carve out all of the files it can find, create a set of common reports, and run a Clam virus scan. The command we will issue is as follows:

```
brunnhilde.py -d --tsk_fstype fat --tsk_imgtype ewf SampleData.E01 TestAnalysis
```

Using this command, Brunnhilde will create a new **TestAnalysis** directory under **/home/bcadmin**, where we will find the output.

Brunnhilde will carve files out of the image using **tsk_recover**, generate a **DFXML** report for the filesystem using **fiwalk**, execute the virus scan, and run an analysis with **Siegfried**.

You can examine the output by navigating to the new **TestAnalysis** folder:



Brunnhilde provides many other options, including scanning files with **bulk_extractor**. To enable **bulk_extractor**, simply add the -b flag to the previous command:

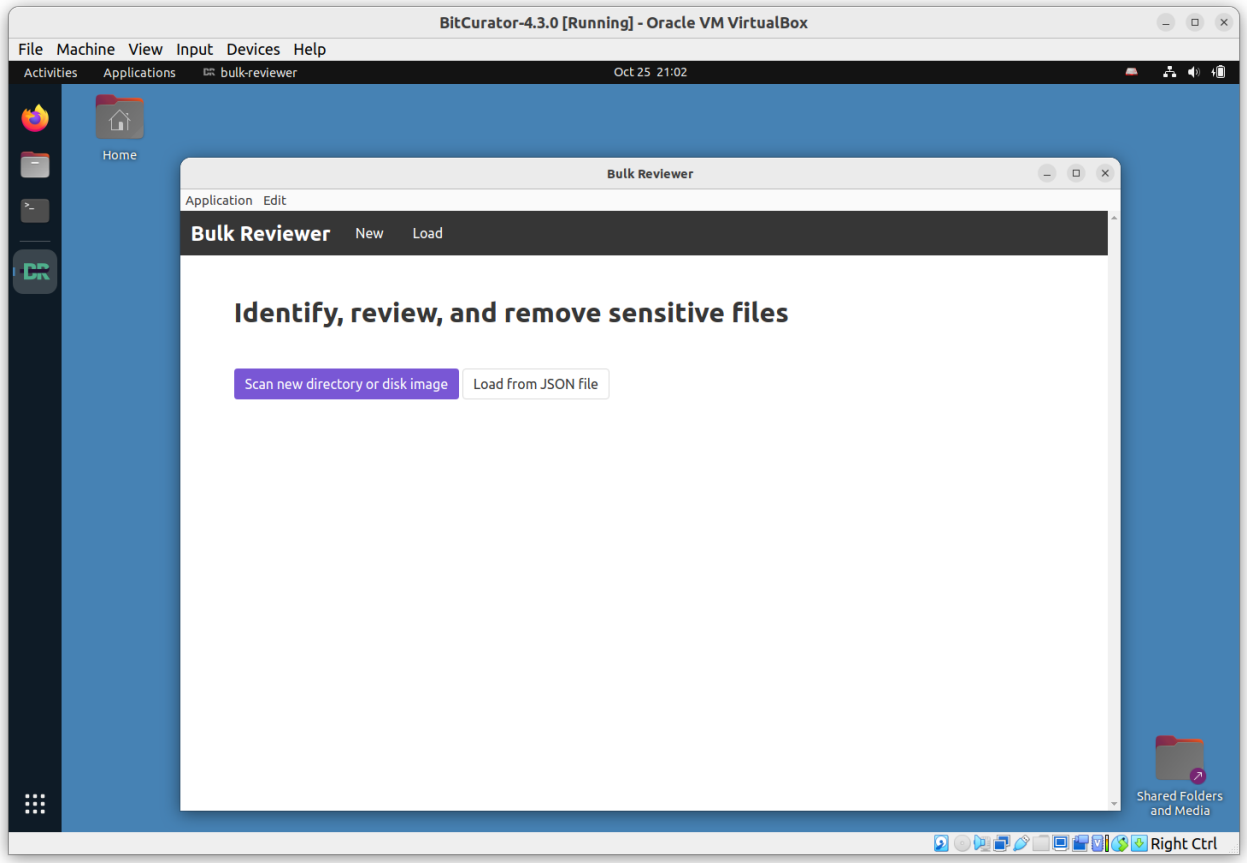
```
brunnhilde.py -d -b --tsk_fstype fat --tsk_imgtype ewf SampleData.E01 TestWithBE
```

You should see a new directory, **TestWithBE** under **/home/bcadmin**, that now also contains a bulk_extractor directory with the scanner output files in it.

Consult the [Brunnhilde documentation] for additional detail.

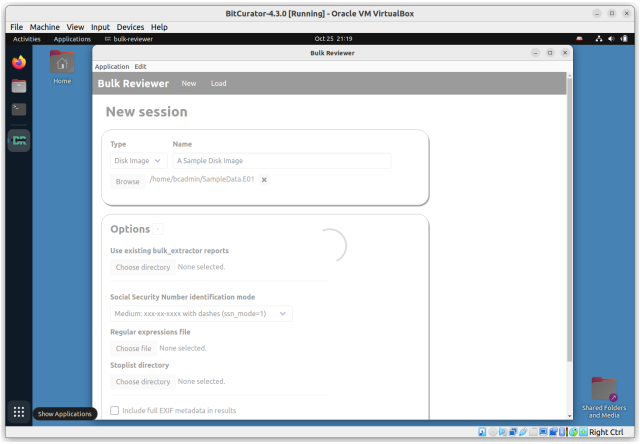
Scanning Disk Images and Directories with Bulk Reviewer

BitCurator 4.x.x releases include **Bulk Reviewer** (version 0.3.1). Bulk Reviewer "is an Electron desktop application that aids in identification, review, and removal of sensitive files in directories and disk images. Bulk Reviewer scans directories and disk images for personally identifiable information (PII) and other sensitive information using bulk_extractor, a best-in-class digital forensics tool."

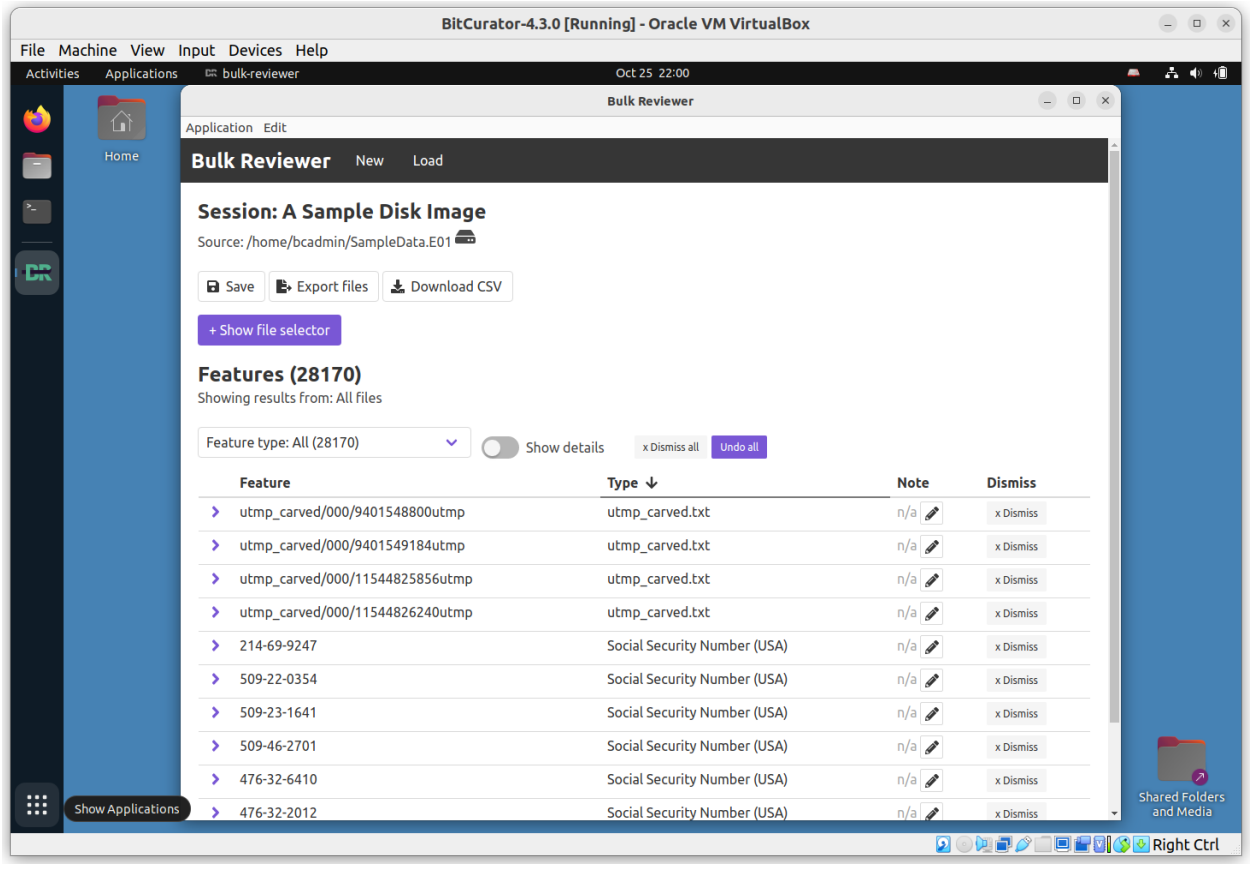


Click **Scan new directory or disk image**. On the following page, select **Disk Image** under **Type** and click **Browse** to add our **SampleData.E01** image. Enter a name; we'll use **A Sample Disk Image** in this example. Click Options to view available options (consult the Bulk Reviewer documentation for additional details). We'll use the default settings in this example. Scroll down and click **Start Scan**.

You will see a spinning progress indicator appear. If your disk image is large, the program may remain in this state for some time. (**Tip**: You can reduce the runtime of this program by generating the **bulk_extractor** reports ahead of time. **Bulk_extractor** provides dynamically updated estimates of completion time. See the next section for how to run **bulk_extractor** independently.)



This process will write a new **bulk-reviewer** directory to the base directory where our image is located. In our example, this directory contains another new directory, **/home/bcadmin/A\ Sample\ Disk\ Image_reports**, and logfile.



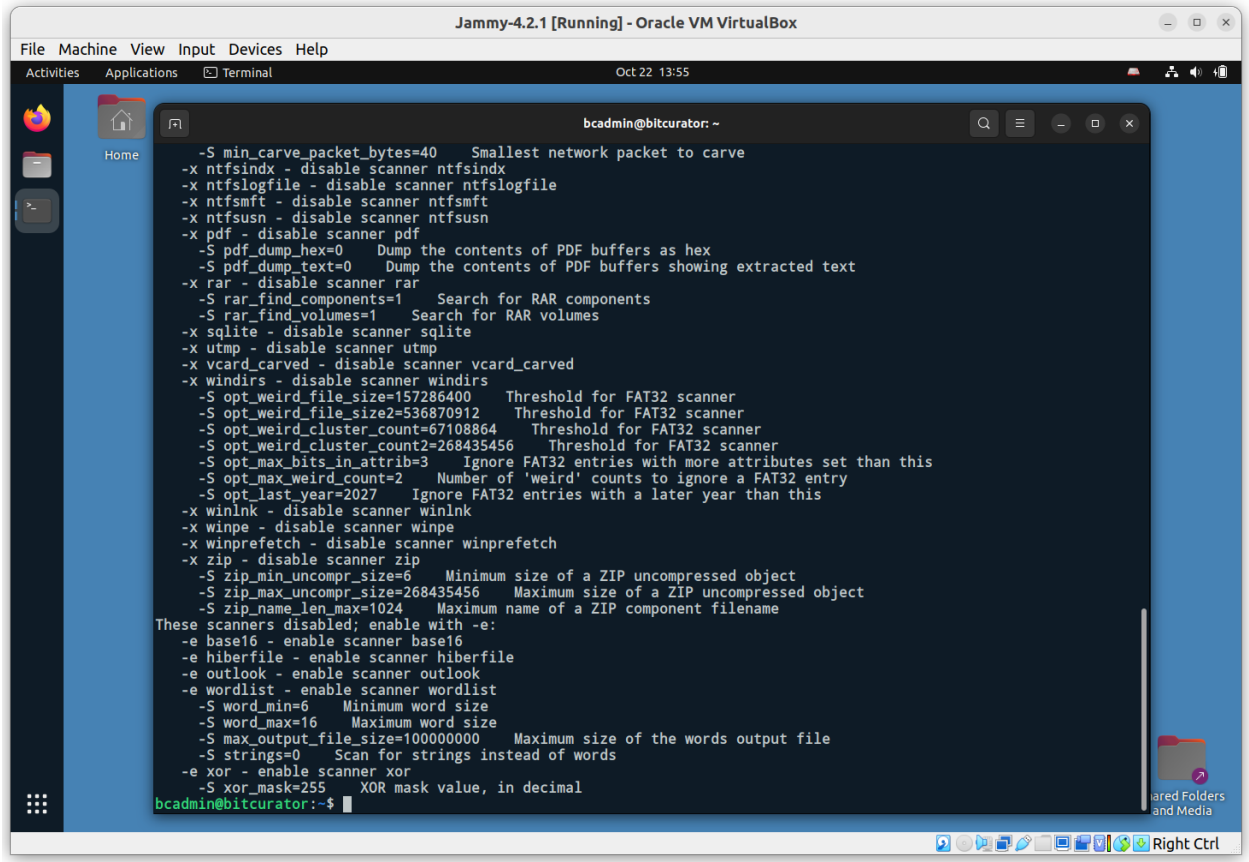
When the scan has completed, a new page will appear showing a navigable list of all discovered features. Consult the [Bulk Reviewer documentation] for additional information.

Scanning Disk Images, Files, and Directories with bulk_extractor

BitCurator 4.x.x releases include **bulk_extractor** (version 2). The bulk_extractor tool “scans any kind of input (disk images, files, directories of files, etc) and extracts structured information such as email addresses, credit card numbers, JPEGs and JSON snippets without parsing the file system or file system structures. The results are stored in text files that are easily inspected, searched, or used as inputs for other forensic processing” (https://github.com/simsong/bulk_extractor/blob/main/README.md).

Note: The 2.x releases of **bulk_extractor** do not include the **BEViewer** GUI front end.

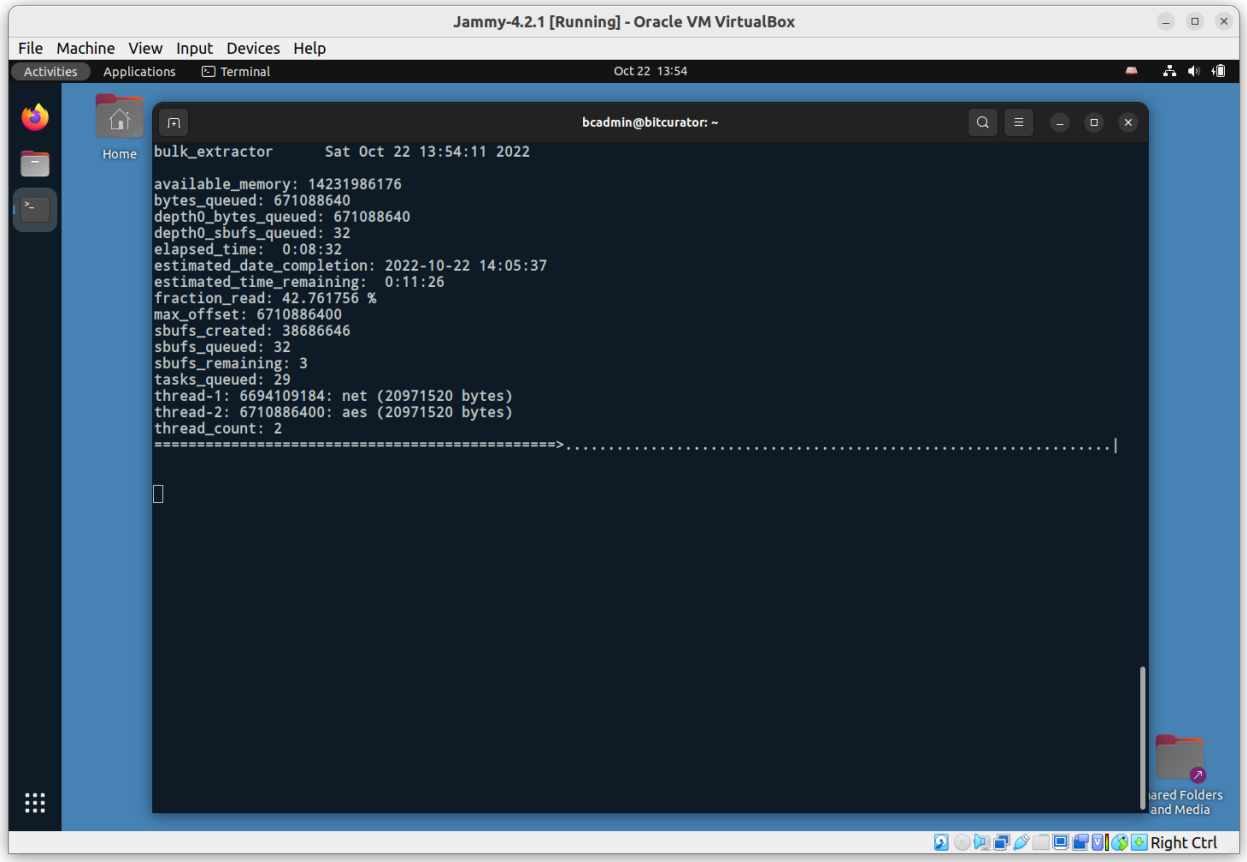
In this example, we will use the **bulk_extractor** CLI to analyze the EWF disk image (SampleData.E01) we created in an earlier step. Click on the **Applications** menu in the top left, navigate to the **Forensics and Reporting** submenu, and click **bulk_extractor**.



This runs the command **`bulk_extractor -help`**, displaying all the options available for the command. A full description of these options can be found in the **bulk_extractor** manual. For this simple test, we will leave all of the default scanners on, and specify only the required output directory, and the source material we wish to scan (our disk image. Enter the following command:

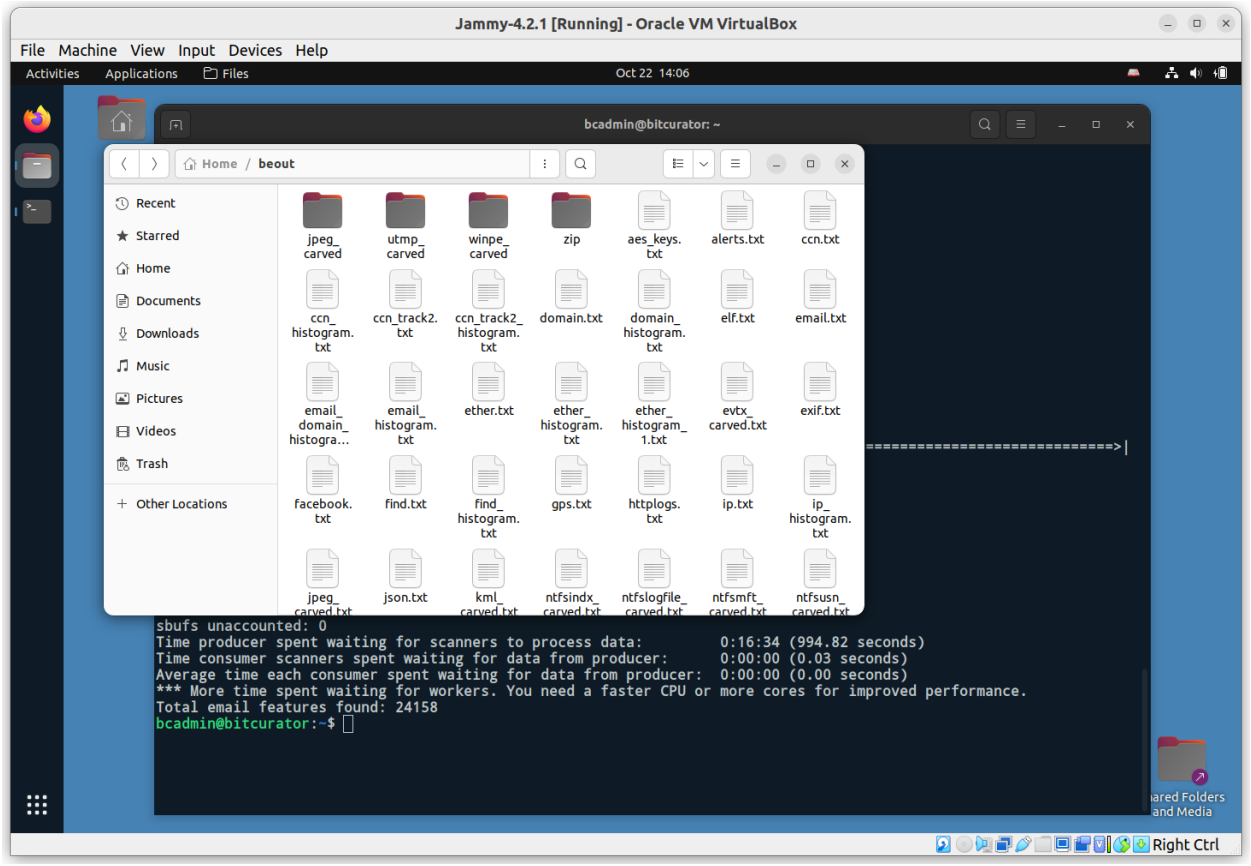
```
bulk_extractor -o beout /home/bcadmin/SampleData.E01
```

This tells bulk_extractor to write its output to a new directory **beout** in our current working directory (/home/bcadmin), and to scan the disk image found at **/home/bcadmin/SampleData.E01.** Once **bulk_extractor** begins running, the terminal will redraw to display a dynamically updated progress page:



Once the scan is complete, you will see a confirmation printed in the terminal with some statistics on the run. Click on the **Files** icon in the dock, and navigate to the **beout** directory in **/home/bcadmin**.

This directory contains a range of items, including folders for JPEG images that have been carved out of the disk image, files that were extracted from zipped materials, and text file reports for each scanner that was run:



Note: The bulk_extractor tool does not parse the filesystem(s) contained within the disk image, so all features identified are referenced only by their byte offset into the source. If you wish to link **specific features** to **the files they originated from within the filesystem(s)**, you will need to run the **fiwalk** tool to create a report of the filesystems, and the **identify_filenames** tool to map features in the **bulk_extractor** output to those filenames.

To run **fiwalk**, in the same terminal type**:**

```
fiwalk -f -X /home/bcadmin/SampleData.xml /home/bcadmin/SampleData.E01
```

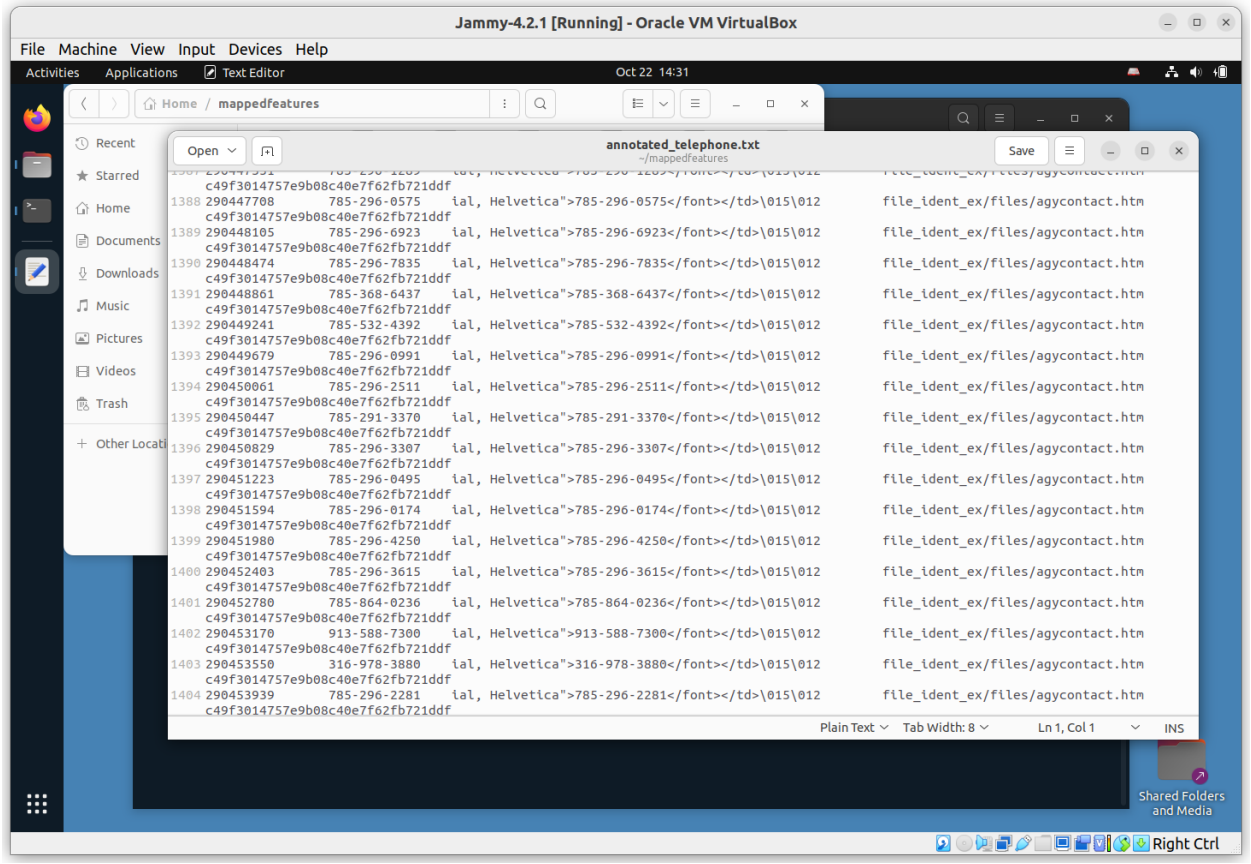
This will create a new file, SampleData.xml, that contains a **DFXML** map of the filesystem(s) within the disk image.

Next, run the following command:

```
identify_filenames.py --all --image_filename /home/bcadmin/SampleData.E01 --xmlfile
/home/bcadmin/SampleData.xml /home/bcadmin/beout /home/bcadmin/mappedfeatures
```

In order, the parts of this command tell the tool to **process all feature files**, use the **image filename SampleData.E01**, use the **output from fiwalk**, the location of the bulk_extractor output, and where to put our new output (a new directory named **mappedfeatures**).

Click on the **Files** icon in the dock, navigate to the **mappedfeatures** directory, and examine one of the files (in this example, we're looking at **annotated_telephone.txt**).

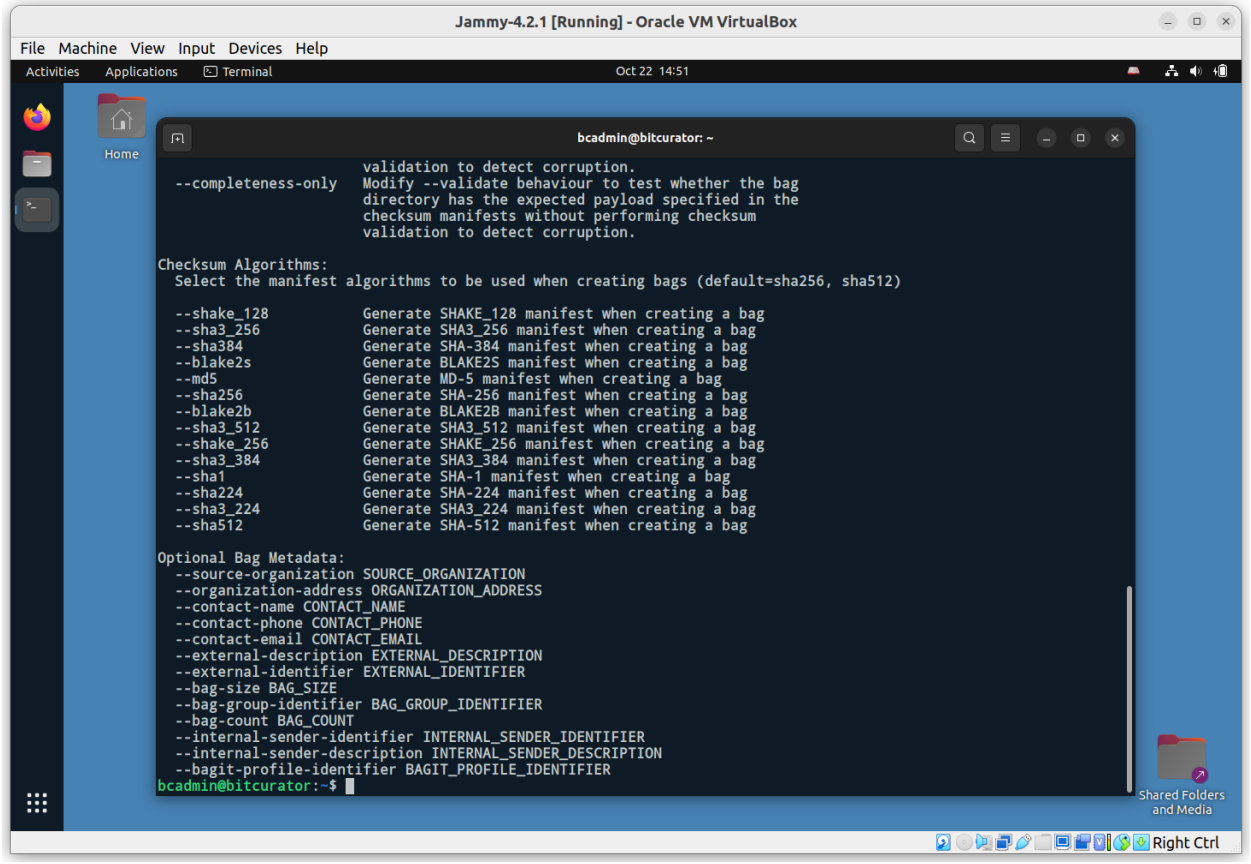


Compare this to the equivalent report in **beout**, and you will see there are now two new columns: the full file path of the file from which the feature originated, and the MD5 of that file.

Creating BagIt Packages for Secure Transfer with BagIt-Python

BitCurator 4.x.x includes the latest release of **bagit-python**, a simple command line tool to create BagIt-style packages to facilitate data transfer.

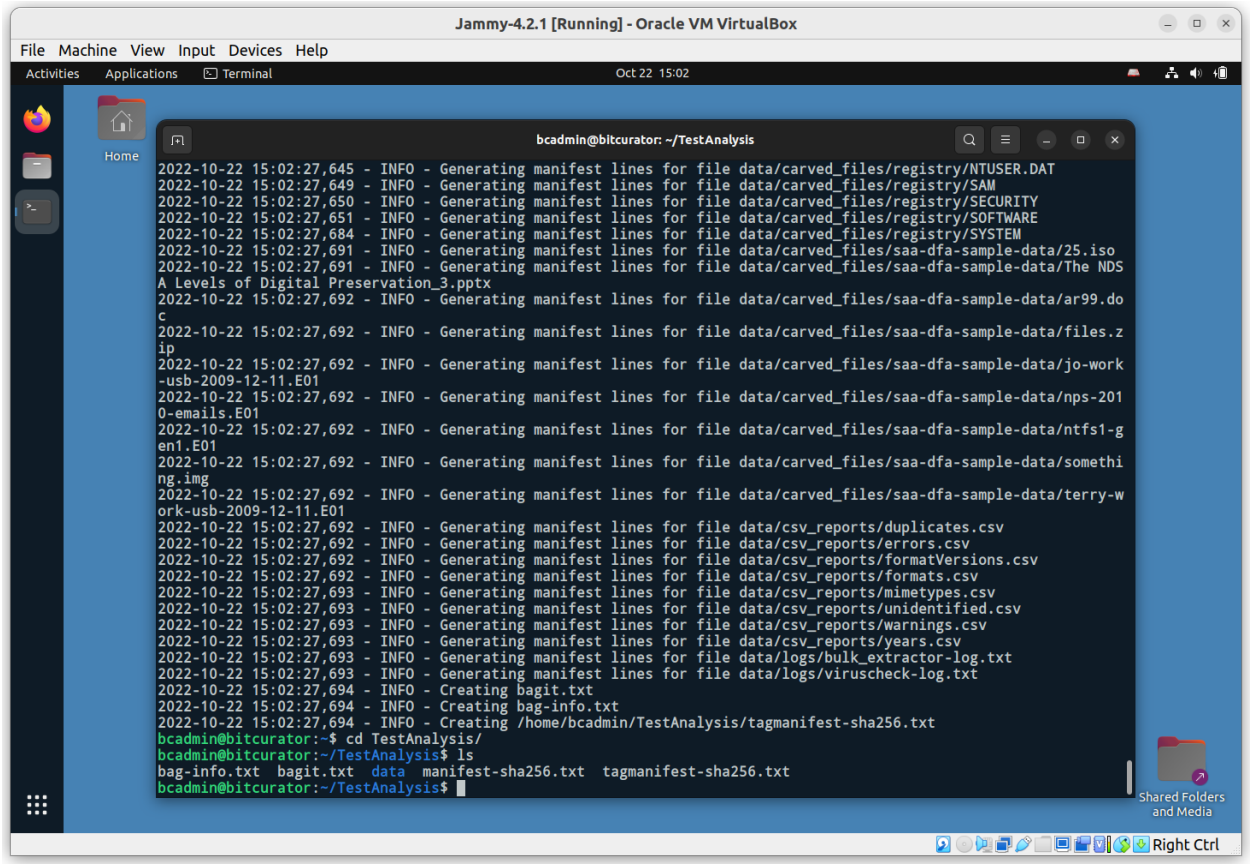
To see the help options for **bagit-python**, click the **Applications** menu in the top left, navigate to **Packaging and Transfer**, and click **Bagit-Python Library (BIL)**. A new terminal window will open:



The **bagit.py** command has many available options. Consult the README for a detailed explanation. For this simple example, we will create a bag that includes a metadata identifier for the source organization, specifies that we will use the SHA256 hash algorithm, and that we are creating a bag of the **TestAnalysis** directory we created in the **Brunnhilde** exercise:

```
bagit.py --source-organization “BitCurator” --sha256 /home/bcadmin/TestAnalysis
```

Note: This will perform a “bag in place” operation, which rebuilds the original **TestAnalysis** directory, moving its content into a **data** subdirectory and writing out the relevant bagit metadata into a selection of new files:



Navigating into the **TestAnalysis** directory on the command line, and listing the content, we can see the result of the bag creation operation.

Running fiwalk with the ClamAV Plugin to Scan Disk Images for Viruses and Malware

The “Domex Gateway Interface” (DGI) is a plugin specification for fiwalk that enables metadata generated by other programs to be embedded in fiwalk’s output (for example, a DFXML file) as a key/value pair.

BitCurator includes a plugin, **ficlam.sh**, that enables ClamAV to scan files identified within disk images when fiwalk is run. To do this, open a terminal and navigate to the .fiwalk directory with the following command:

```
cd ~/.fiwalk
```

This directory contains the **ficlam.sh** script, but before we run it we need to provide it with some configuration parameters in a **ficonfig-formatted** configuration file. This has the following layout:

```
# globpattern    channel    args

*.*              dgi        ./ficlam
```

This specific configuration pattern tells the program to scan all filenames with all extensions (*.*). If desired, we could replace this pattern with something more specific - for example, scan only files with the .exe file extension (*.exe).

For now, we’ll create a new configuration file, **clamconfig.txt**, from the command line with this single line:

```
echo ‘*.* dgi ./ficlam.sh’ > clamconfig.txt
```

Now, run **fiwalk** with the appropriate flag to tell it to use the plugin:

```
fiwalk -c clamconfig.txt -X /home/bcadmin/SampleDataWithClam.xml /home/bcadmin/SampleData.E01
```

A new file, **SampleDataWithClam.xml** will be created in **/home/bcadmin**/. Each identified file result will now include the result of this plugin: