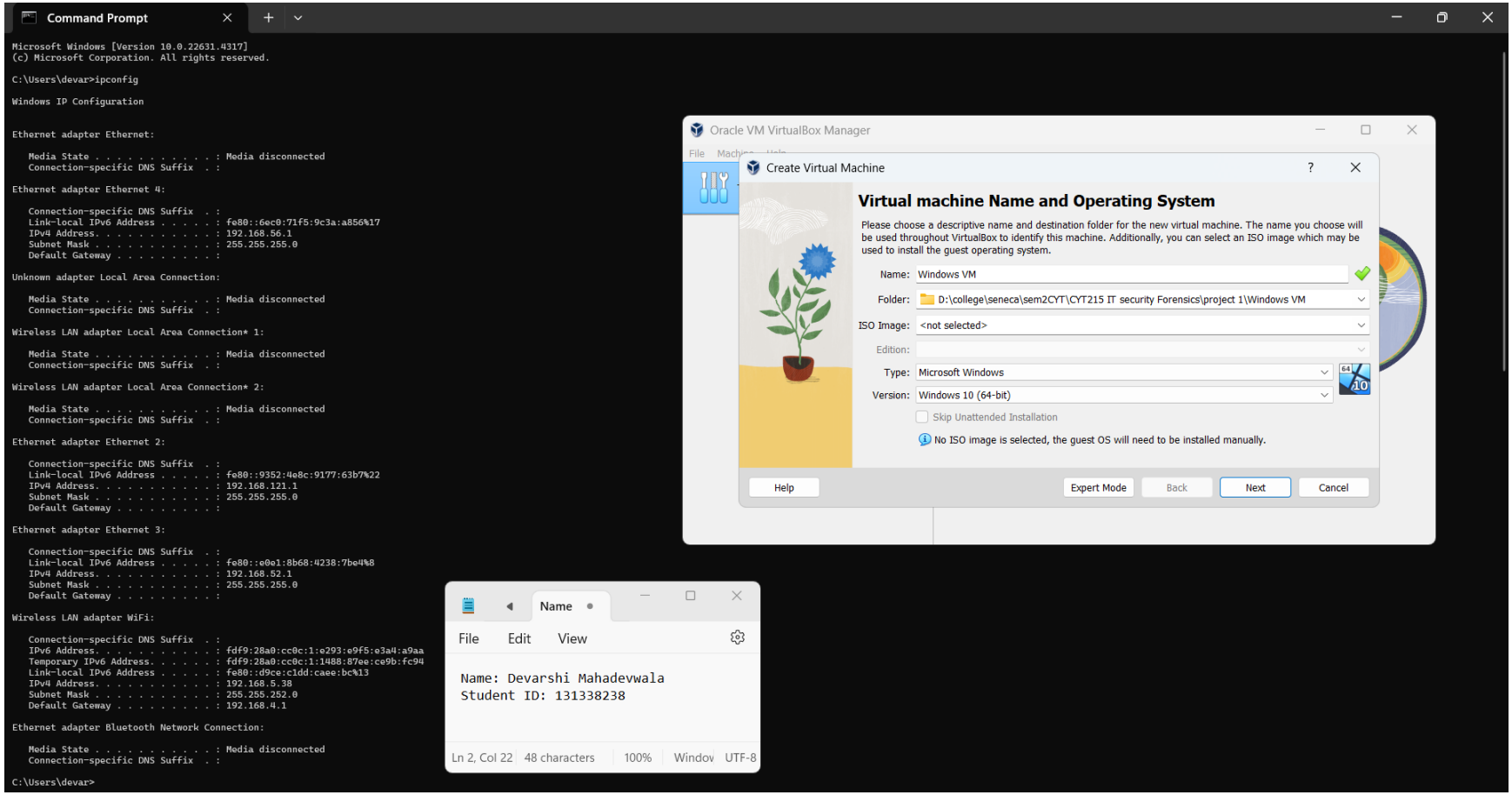
Name ↓			
Devarshi Mahadevwala			

Name	Project1: Build Your own Forensic Workstation
Main Goal	Set up your functional forensic workstation to conduct forensics investigations using variety of popular tools

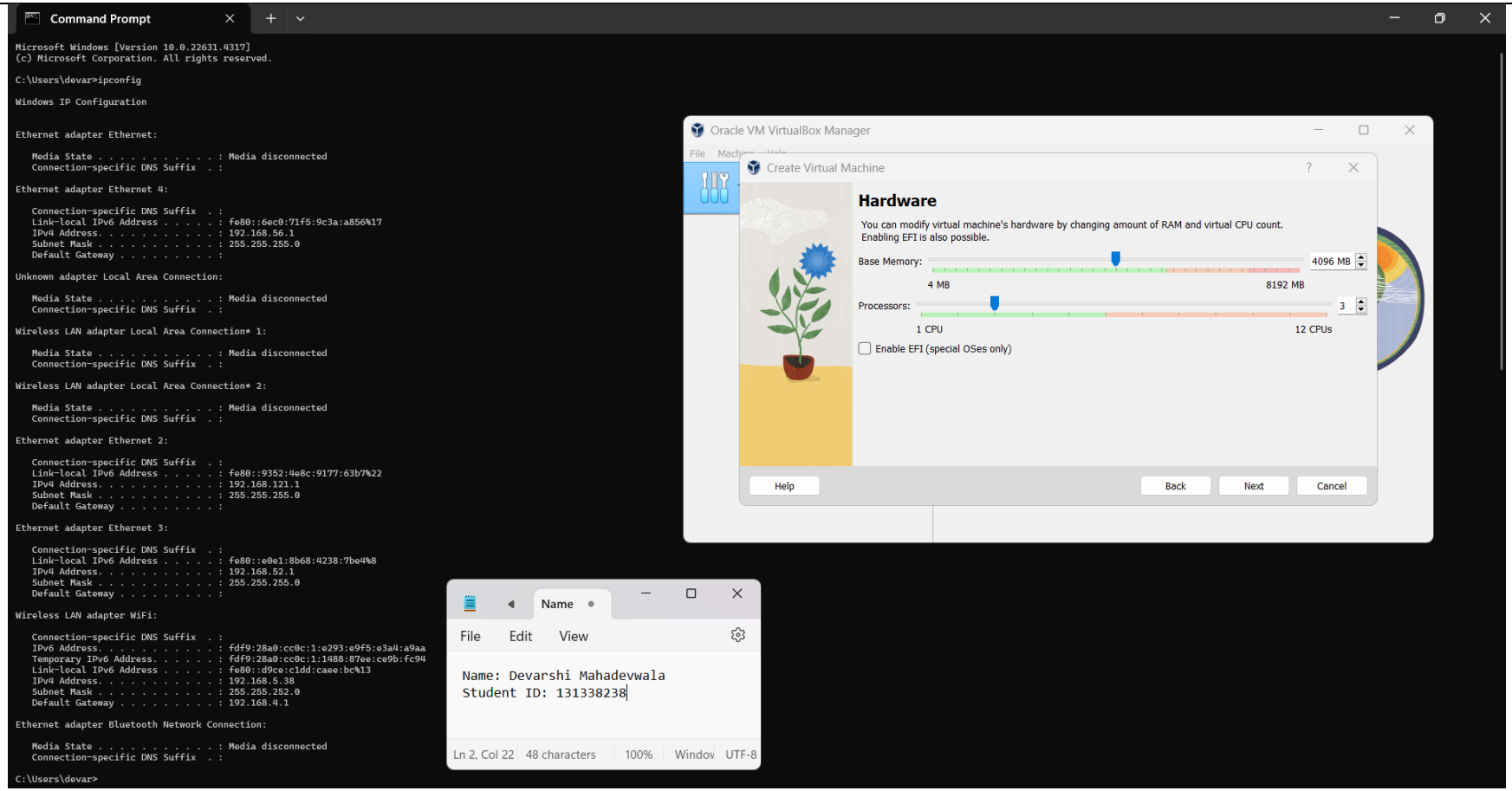
Instructions	• It is an Individual assignment. Put your name + Student ID in the empty spaces above.
	• Submit via the BB relevant link ONLY. NO submission via email please. Be sure to submit the final version file ONLY.
	• Show me genuine signs of your work is done on your machine. This includes:
	  ○ Screenshots that show your desktop background with Date/Time
	  ○ Show me a pop-up bx that shows "your name + IP."
	  ○ Show your logged in account, if applicable
	  ○ Optional: Show your photo.
	• Use this same template to include your work in the specified fields below. Submit in PDF.
	• Submit your report name with the name: CYT215-Project1-Student Name & ID

Students Work required for this activity	1. You will follow instructions to setup your own forensics workstation on your machine. 2. You will check that your forensics workstation is functioning. 3. You will use your workstation for memory & malware analysis. 4. You will Prepare Your Target System (your own machine). You will Build Your basic Lab.
How to start	• Read thoroughly & follow the instructions mentioned in this link. The instructions will guide you to a step-by-step of how you complete your work successfully https://bluecapesecurity.com/build-your-forensic-workstation/ • Take an image of your machine memory. Use this link for guidance: https://dfirmadness.com/case-001-memory-analysis/ • You can use any installed tool or focus on common tools for memory analysis e.g.: Volatility; Cyber Triage; Rekall; Redline;
Important	• Your target system should be your own machine
Students Reports	1. Take screenshots of all your works steps 2. Show you have practiced 3 relevant tools to investigate memory. 3. Write a detailed report of personal learning experience (free writing).
Grading Rubrics	• 5 Marks for completing the setup of your forensics workstation successfully. • 5 Marks for using your workstation for memory analysis. Ate least, use 3 tool/memory analysis. • 2 Marks for your detailed personal learning experience (free writing)

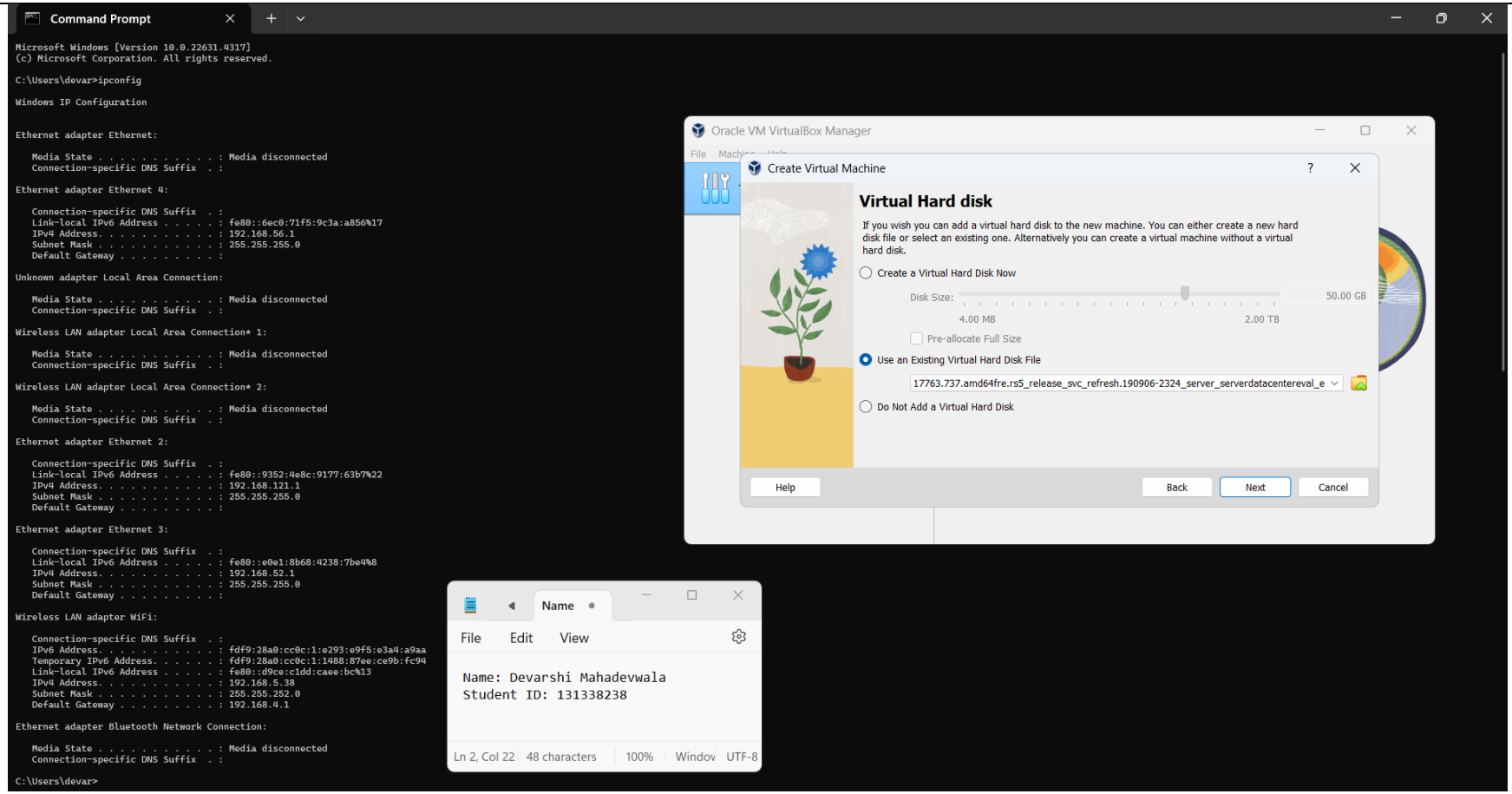
Student work	First, we need to setup the forensics lab along with the tools and necessary applications to conduct memory analysis. For this, I have setup a windows virtual machine environment. I have downloaded FTK imager to capture memory image from my windows machine and run memory analysis on that image from several applications like volatility, cyber triage, and redline. I have used windows VM in VMware, but below are the steps to setup windows VM in virtualbox.



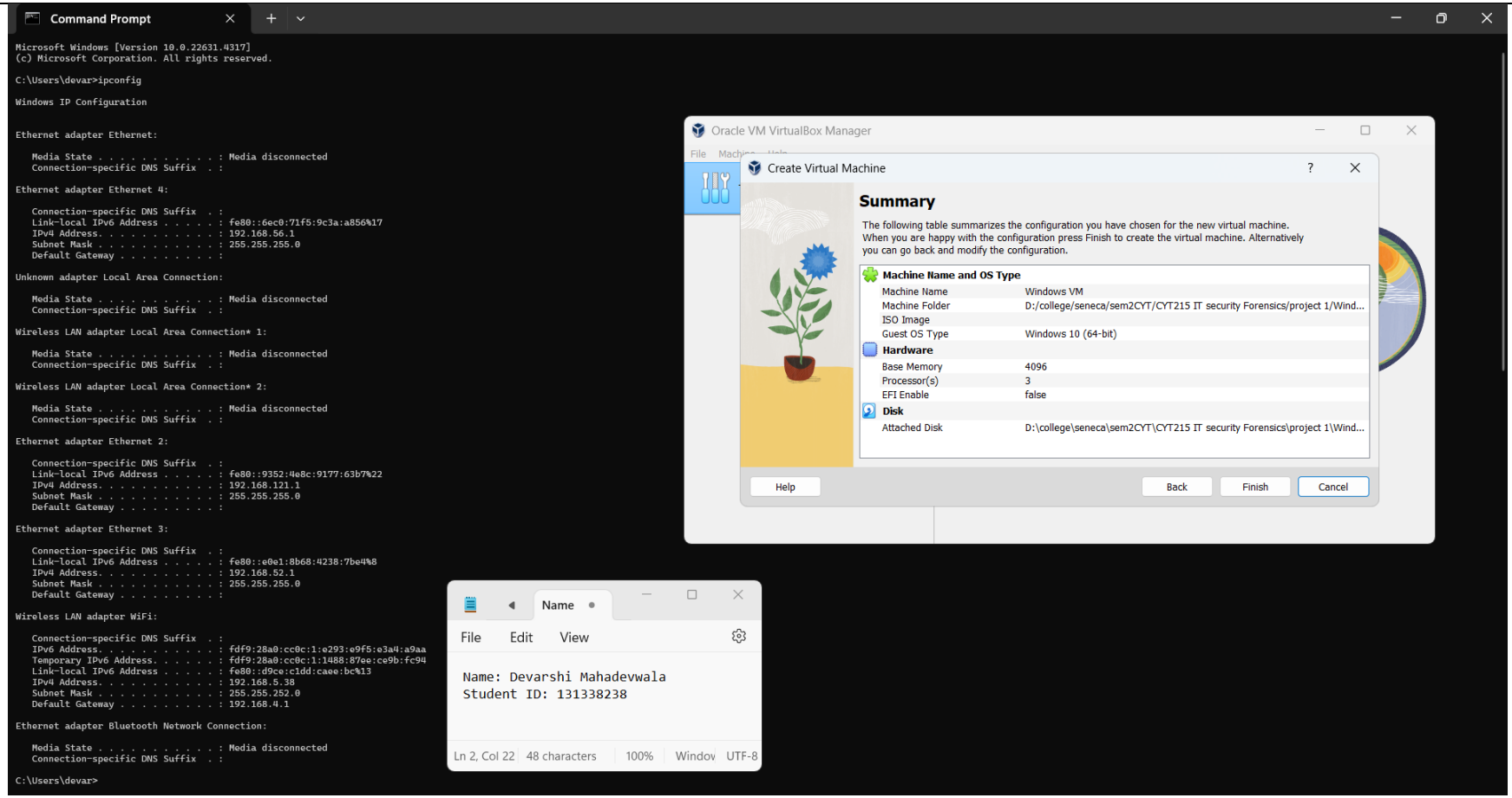
The above photo shows the start of setting up of windows VM in virtual box. We define the destination of the VM and select the ISO image where the VM will be built upon.



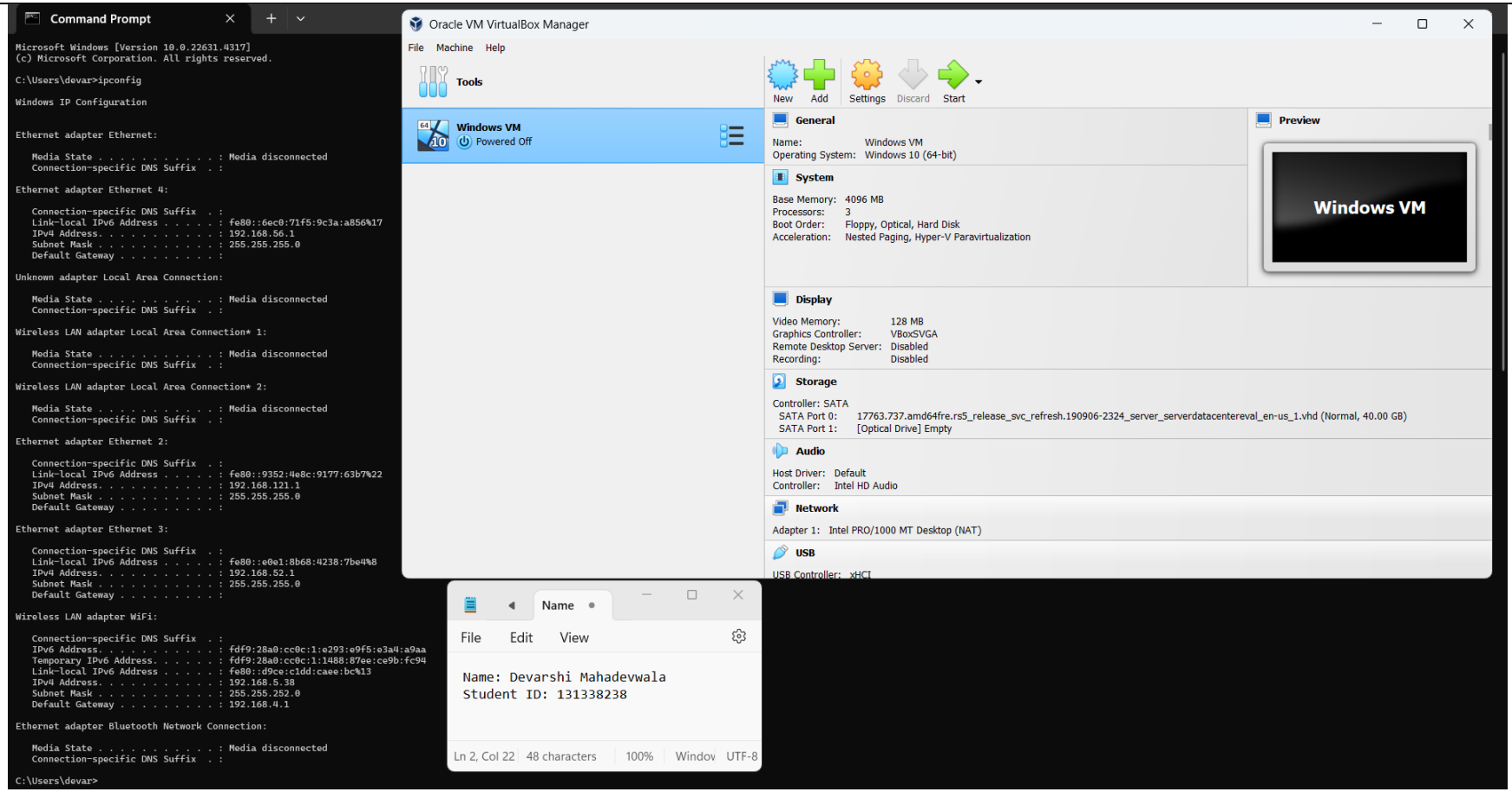
Specify the details of the virtual machine hardware. I have selected 3 processors and 4GB RAM for my windows virtual machine.



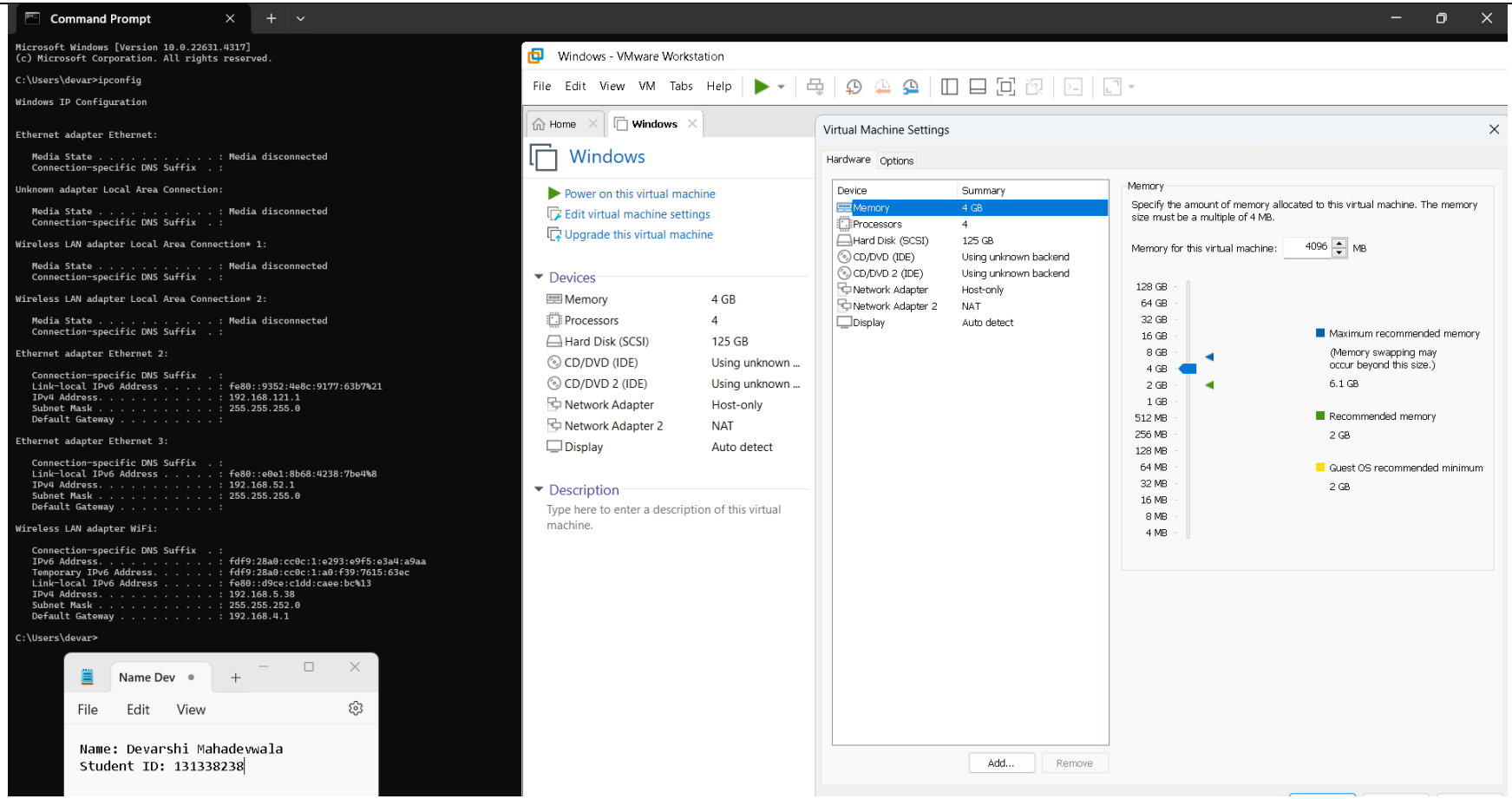
A virtual hard disk needs to be specified so that the virtual machine has memory storage.



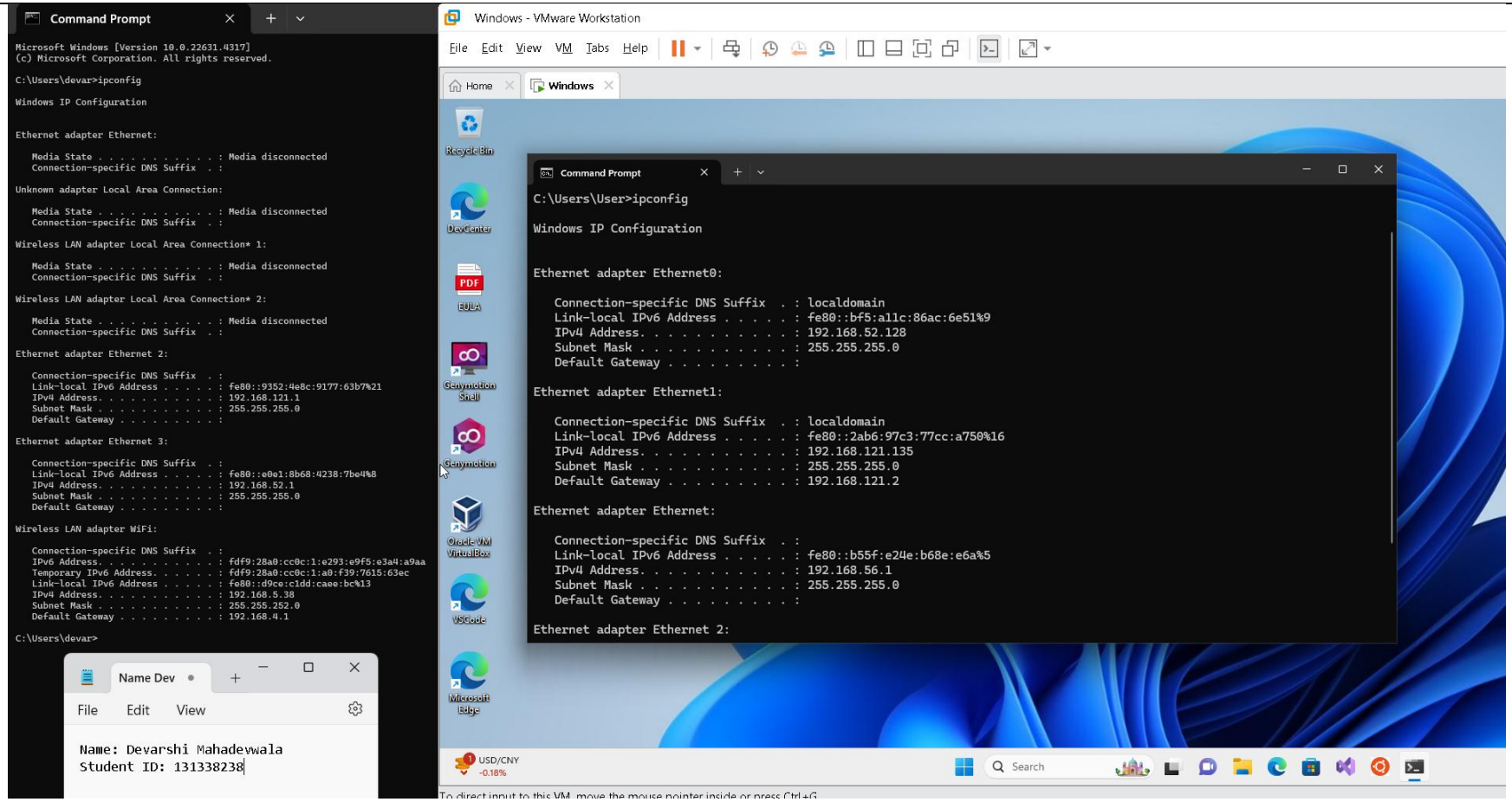
Finally, check the specifics of the virtual machine in the summary and if it is to your liking, finish the setup.



Virtual machine setup is now complete. You just need to start the machine to run windows in the virtual environment.

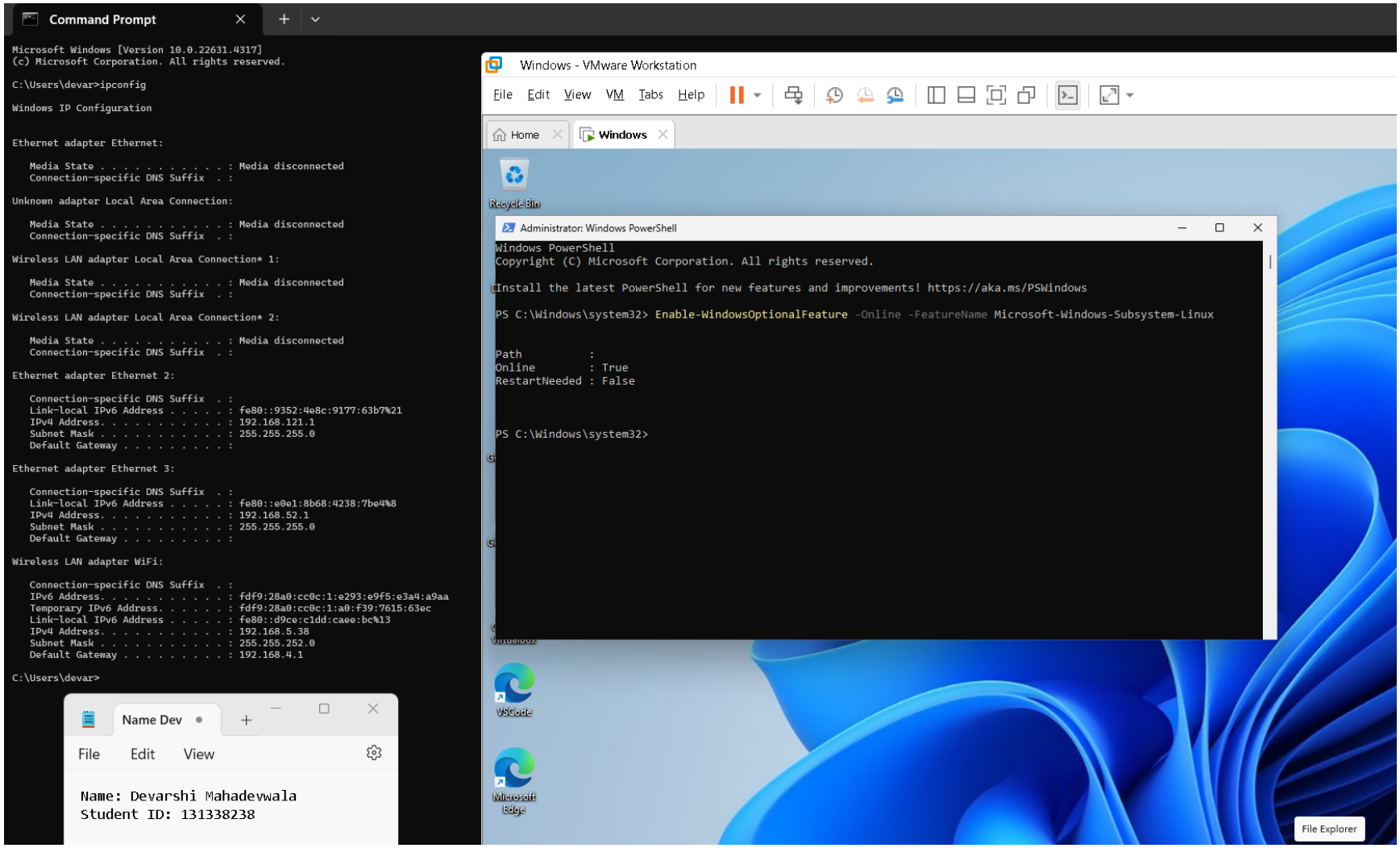


Before starting the virtual machine, check out the settings of the virtual machine and make changes you feel fit for your liking.

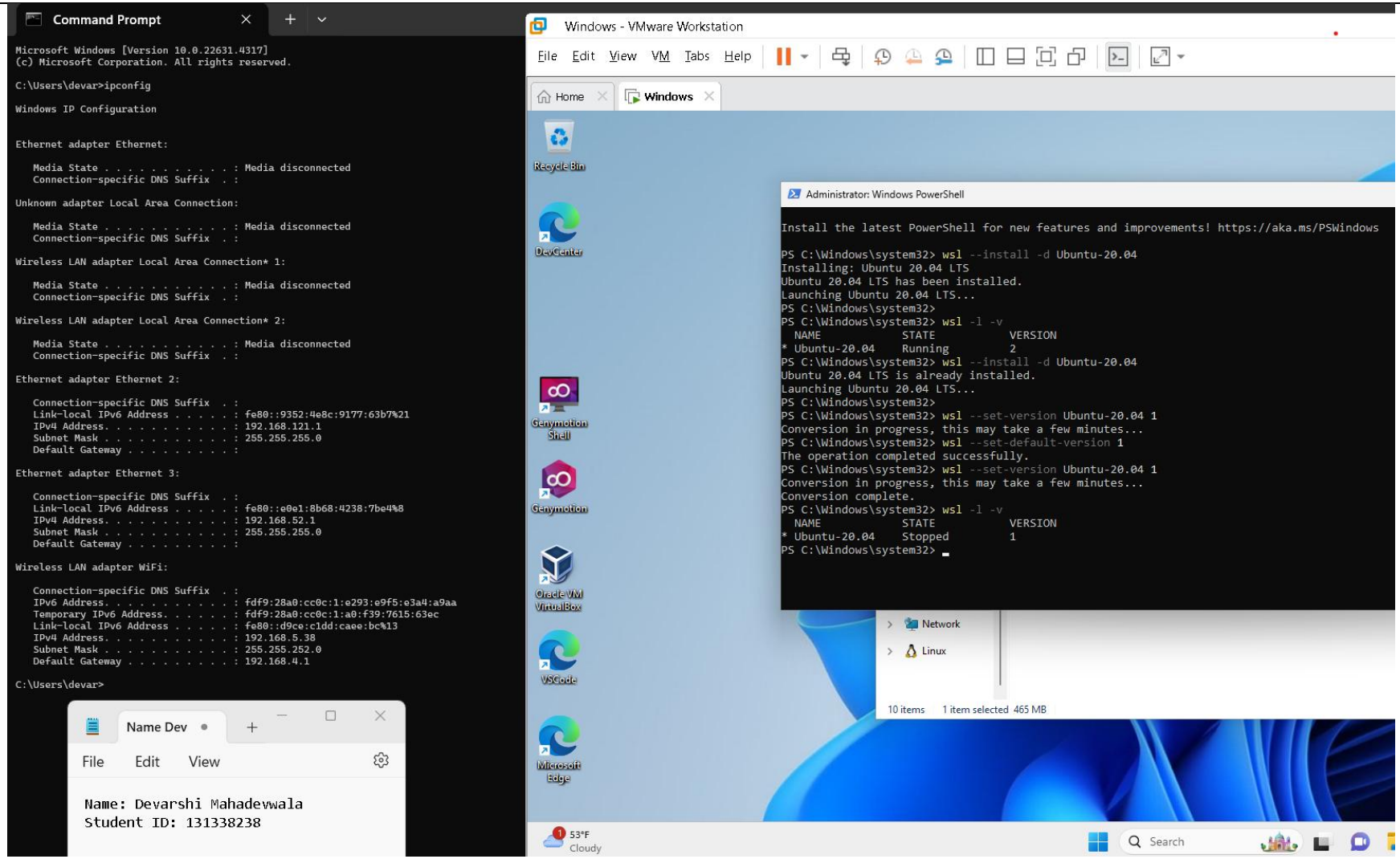


The windows machine has been powered up and above states my virtual machine's IP address.
Note: The virtual machine is working on VMware as it is the application I prefer to run my virtual environments.

We need to setup a WSL ubuntu environment on windows machine.

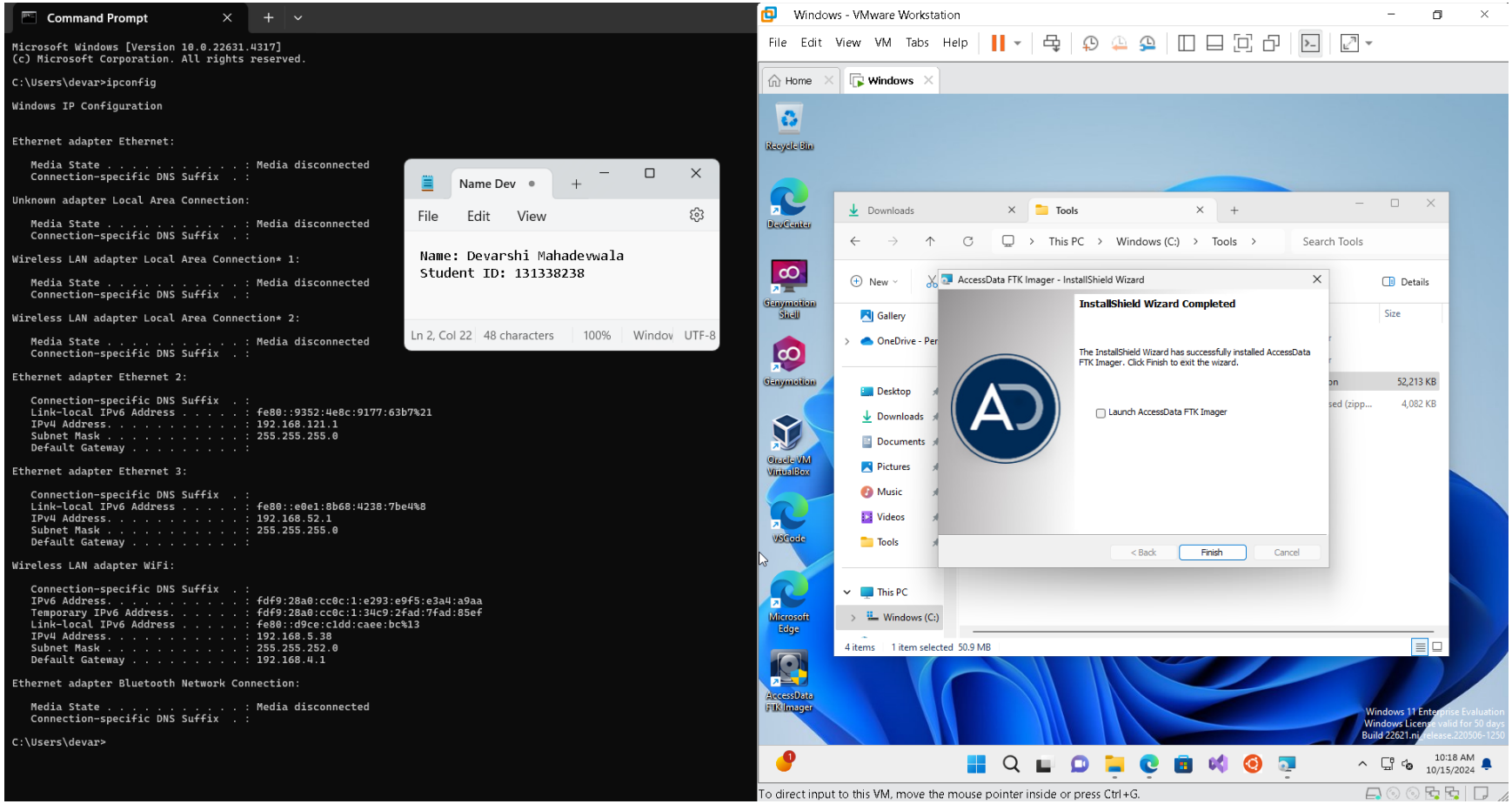


We enable the feature of using ubuntu terminal environment on windows virtual machine.

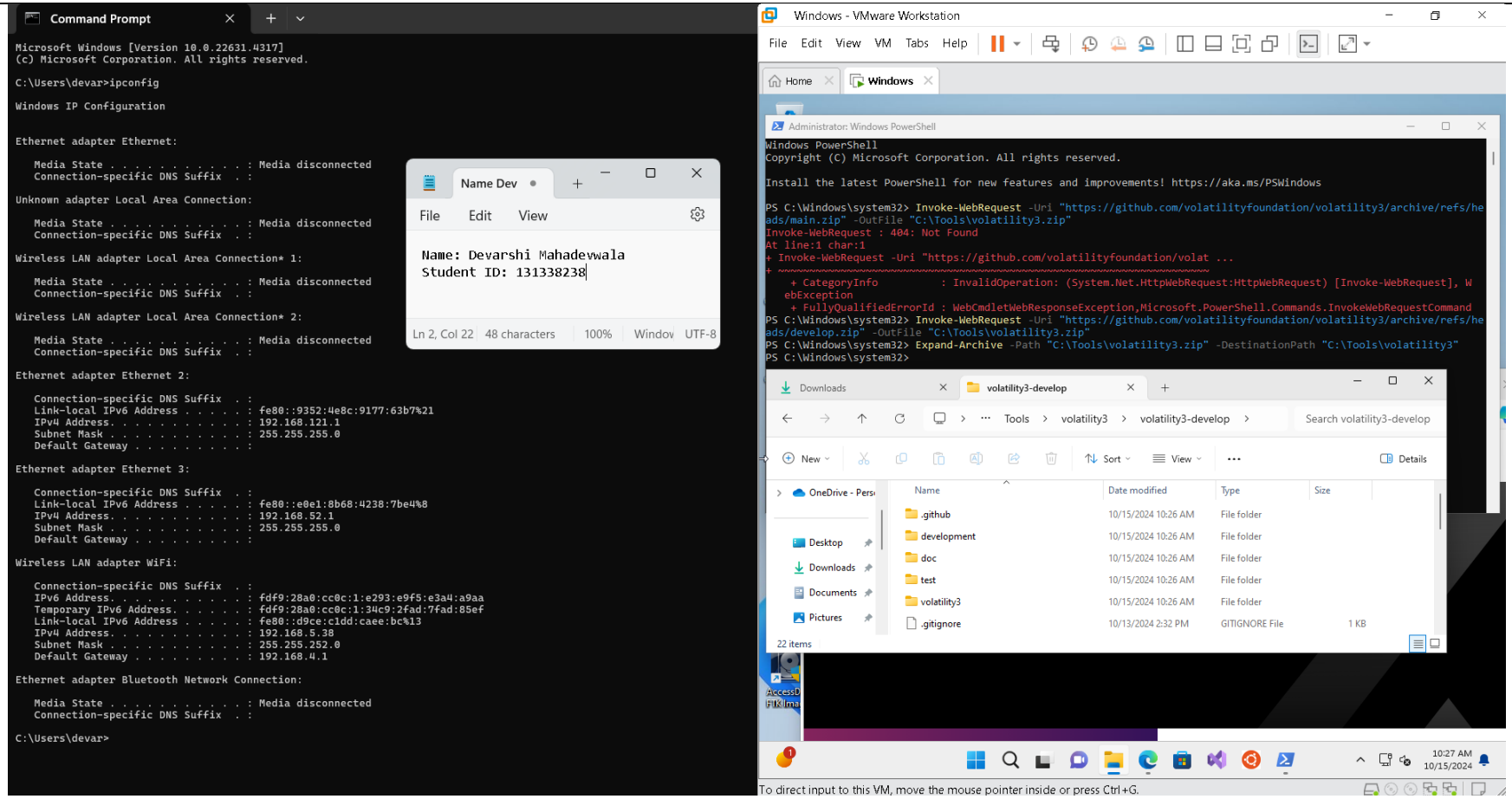


WSL environment download and setup is now complete and the version we are using for this lab is version 1.

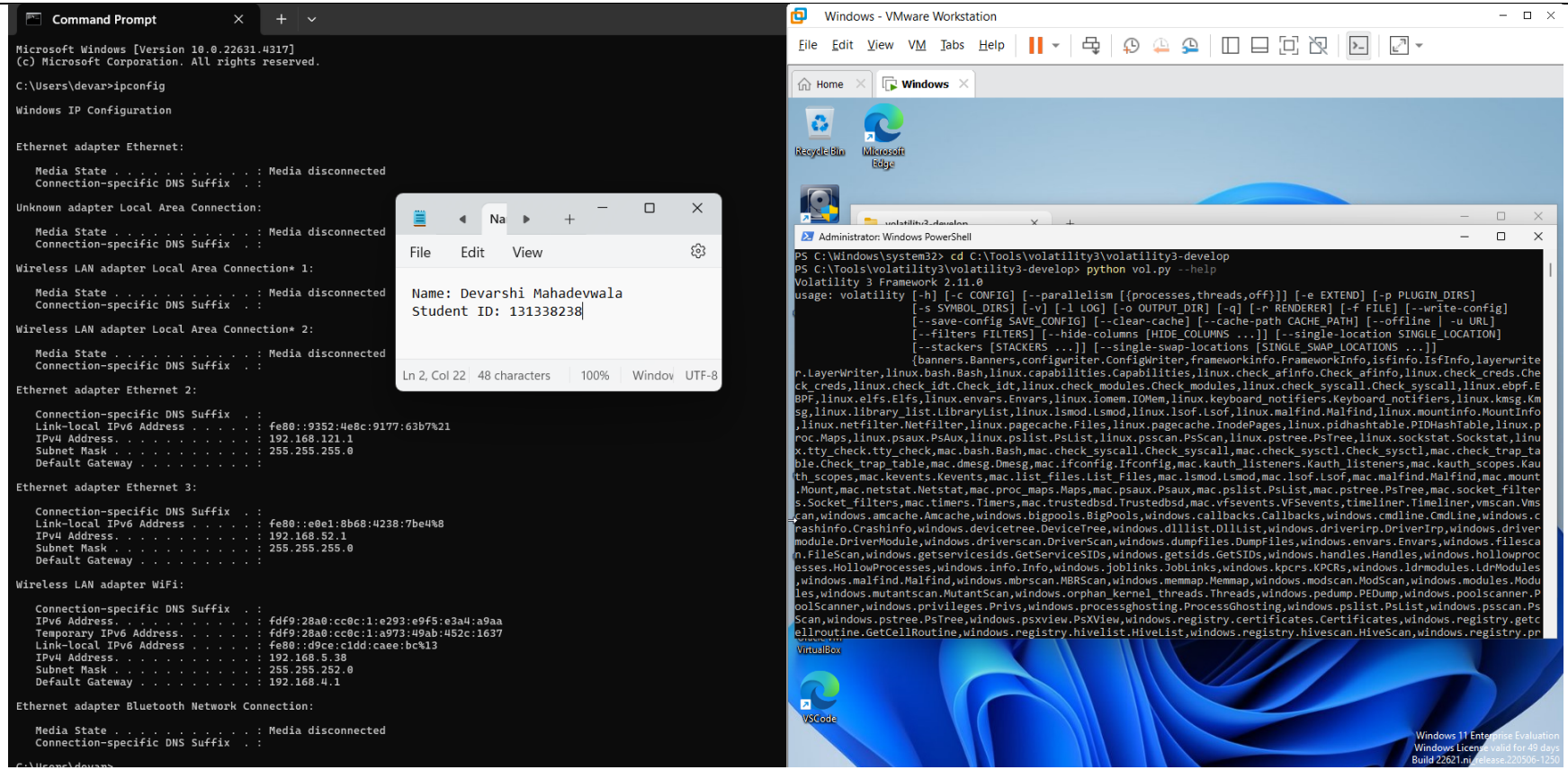
Now, we need to setup some tools on the virtual machine to create our virtual forensics lab. Let us start with FTK imager which will help us capture a memory image file on which we will run our memory analysis operations.



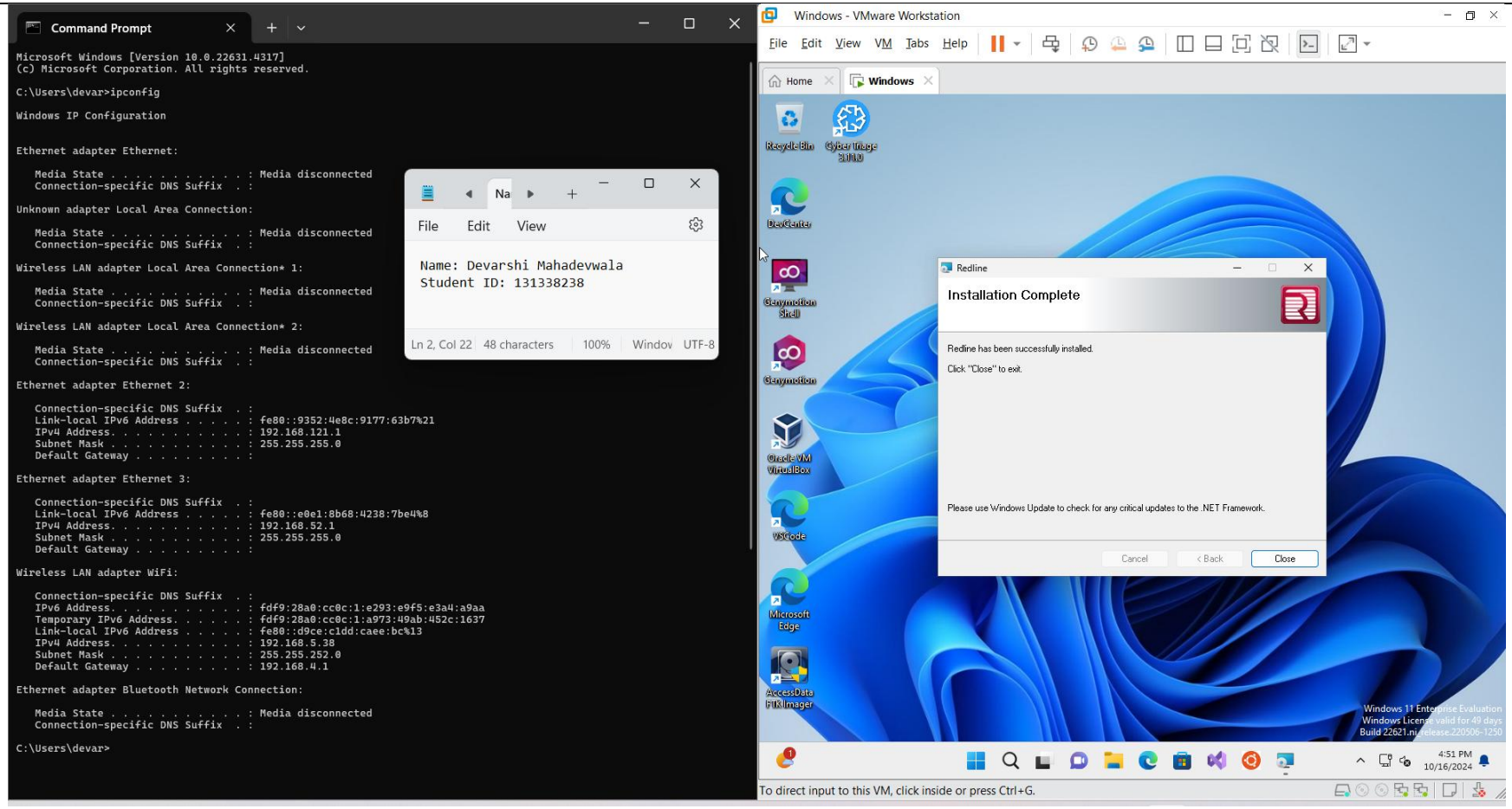
FTK imager has a GUI application. Above is the photo which shows that FTK imager has been successfully downloaded on our virtual windows machine.



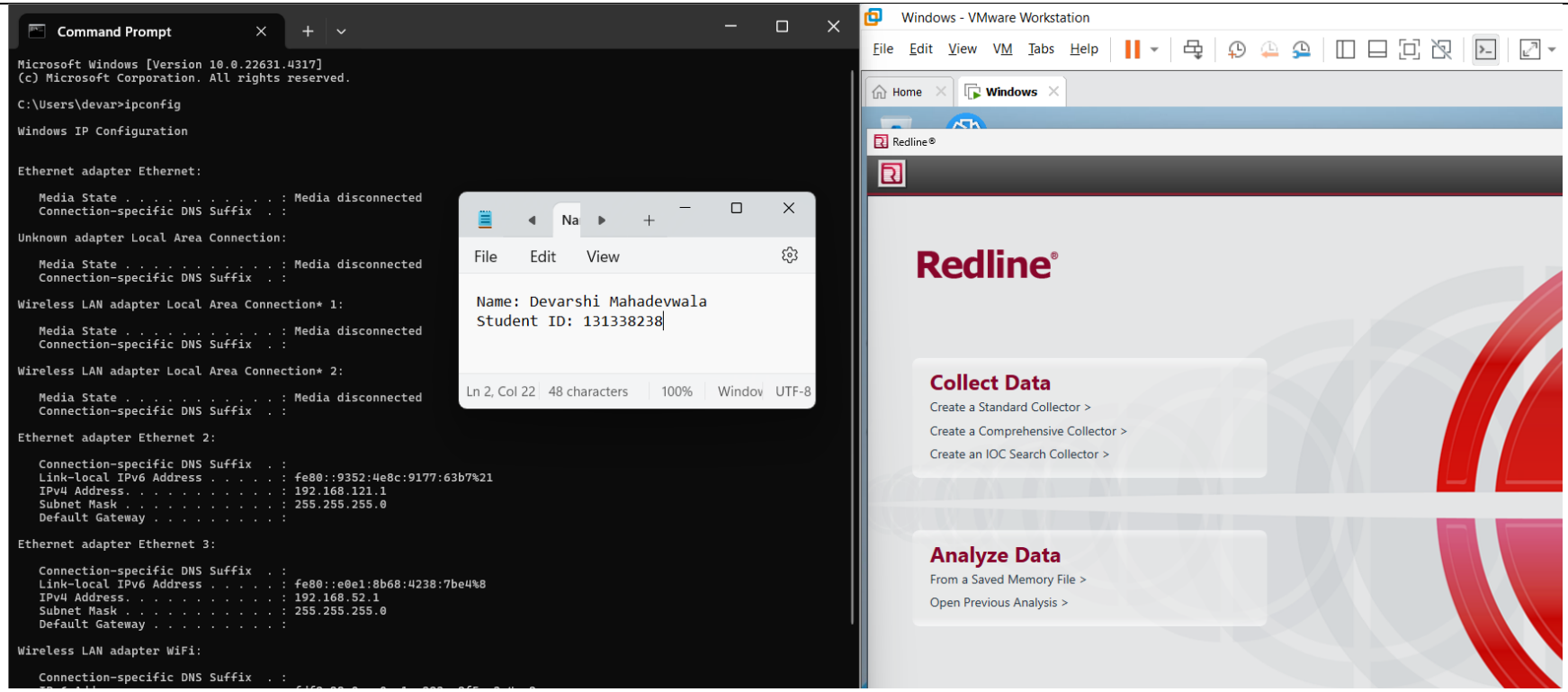
For volatility, I cloned the github repository of volatility 3 and expanded the archive of volatility in C:\Tools in my C drive.



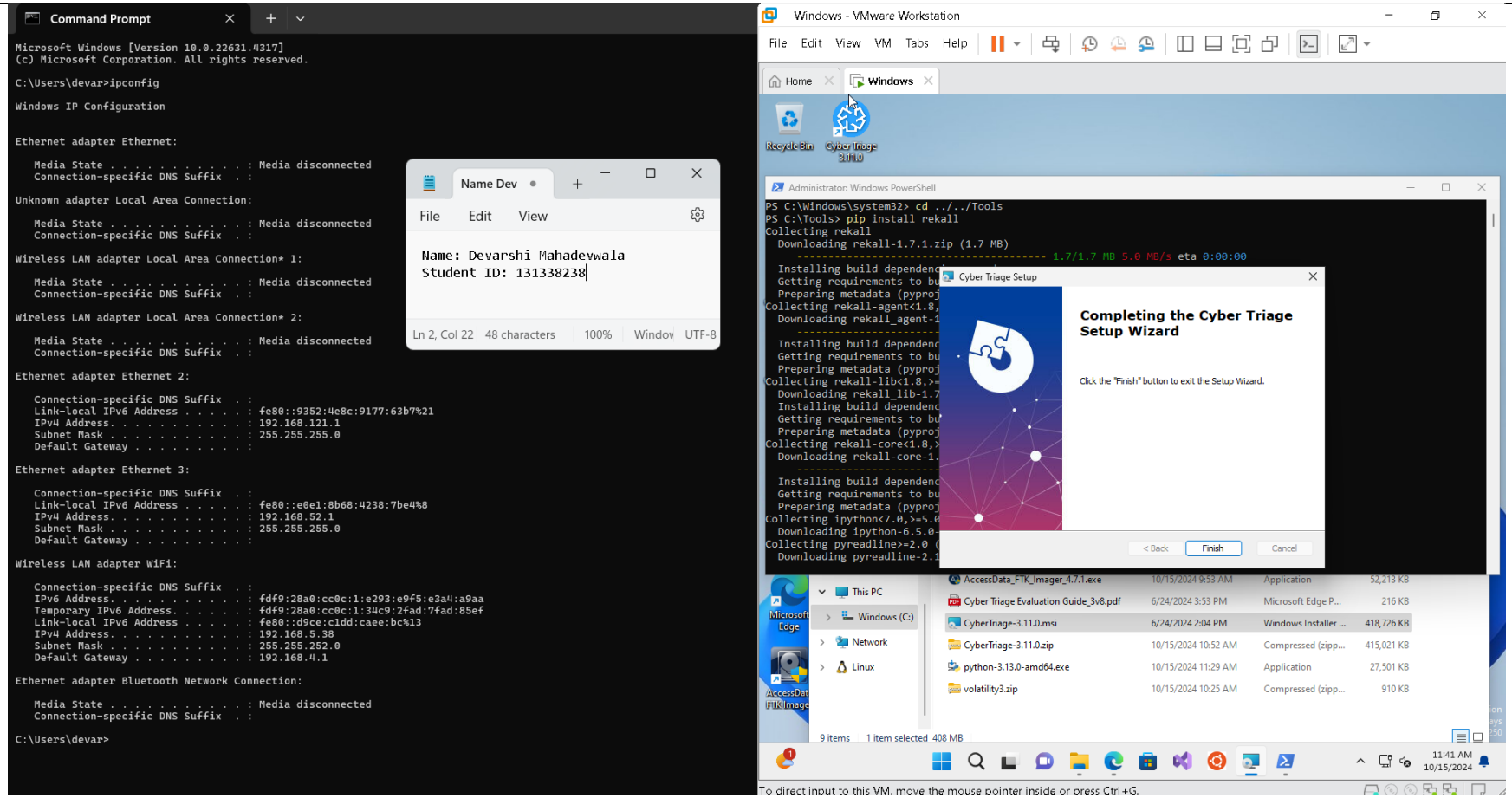
In PowerShell, from the folder where volatility was expanded, we run a help function to check if volatility has been successfully downloaded and is working on our system. Now we can use volatility to conduct scans on our image file.



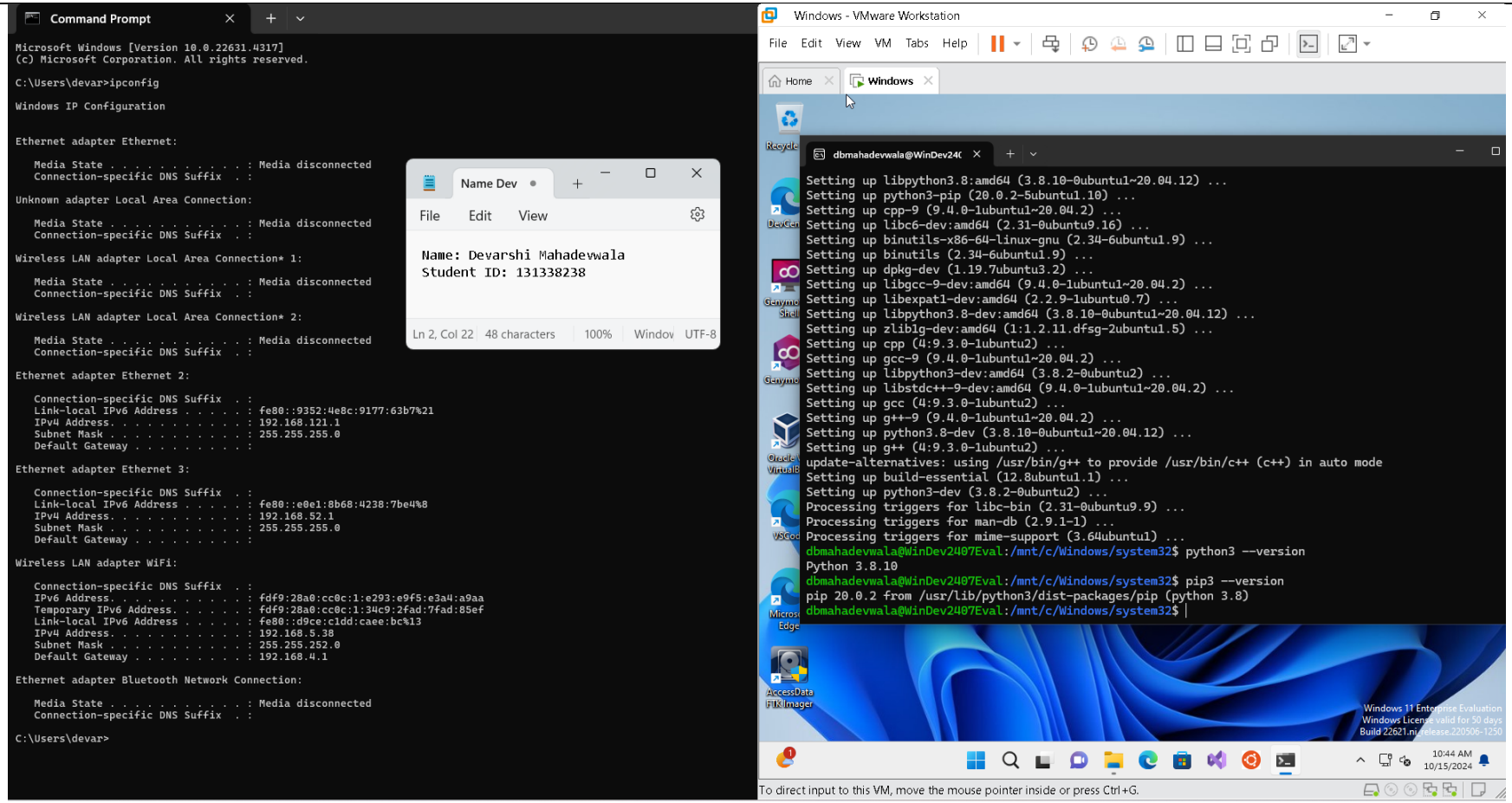
Redline is a GUI application where we can run analysis on our memory image. As it a GUI application, above is the screenshot showing that redline has been successfully downloaded on our virtual environment.



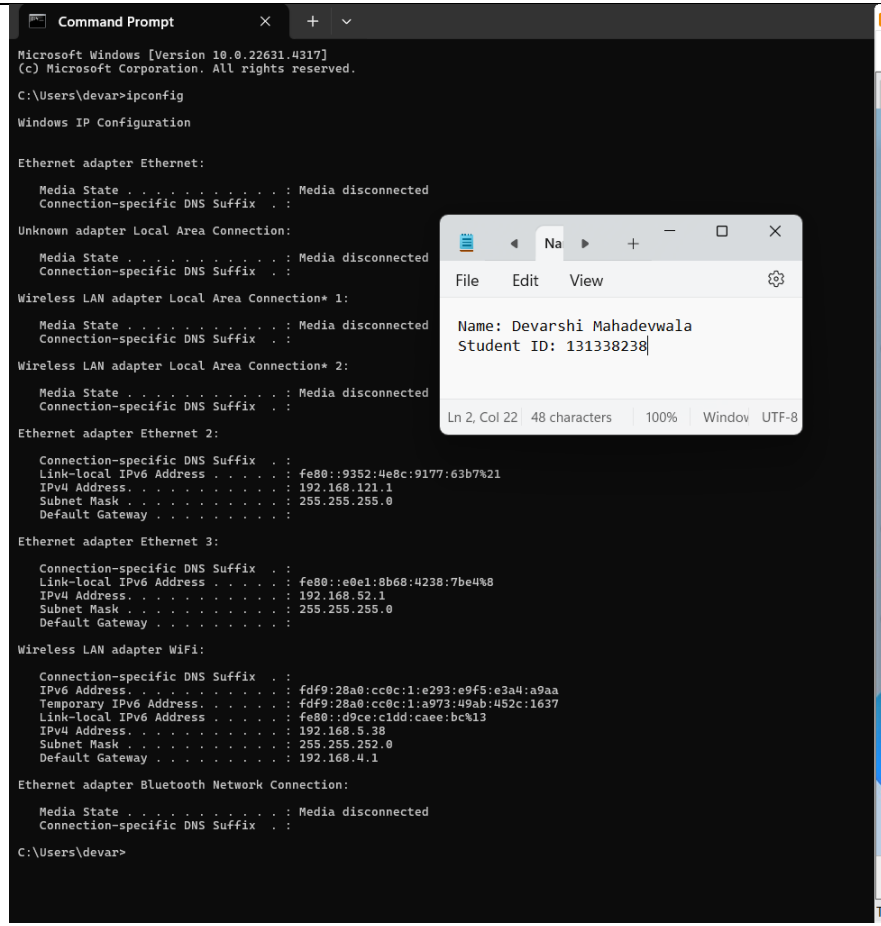
Above is the successful running environment of redline where we can collect data and analyze it.



Cyber triage wizard setup complete. It is again a GUI application that you can easily download from the internet.



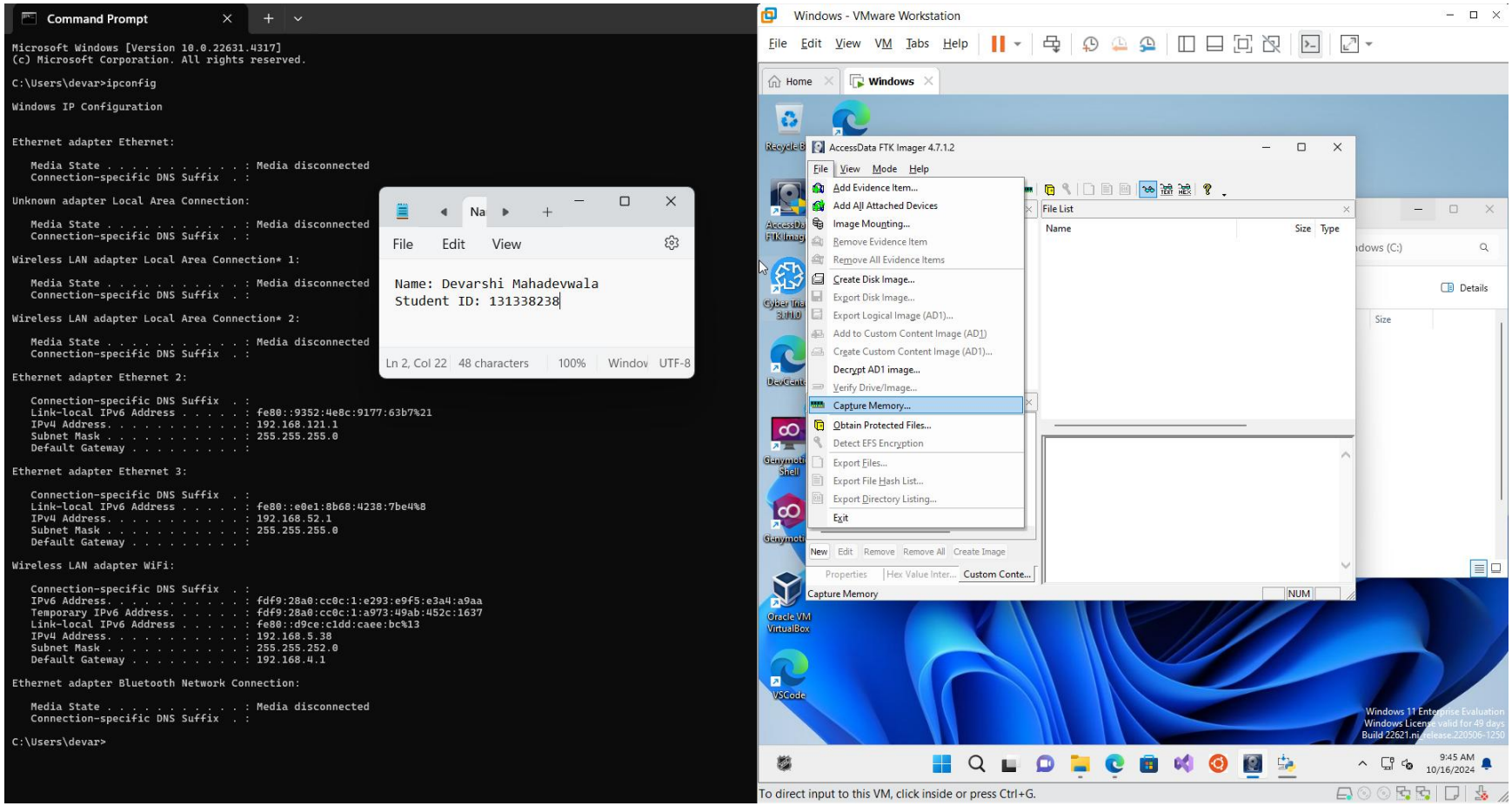
Finally, we install python and corresponding version of pip on our environment.



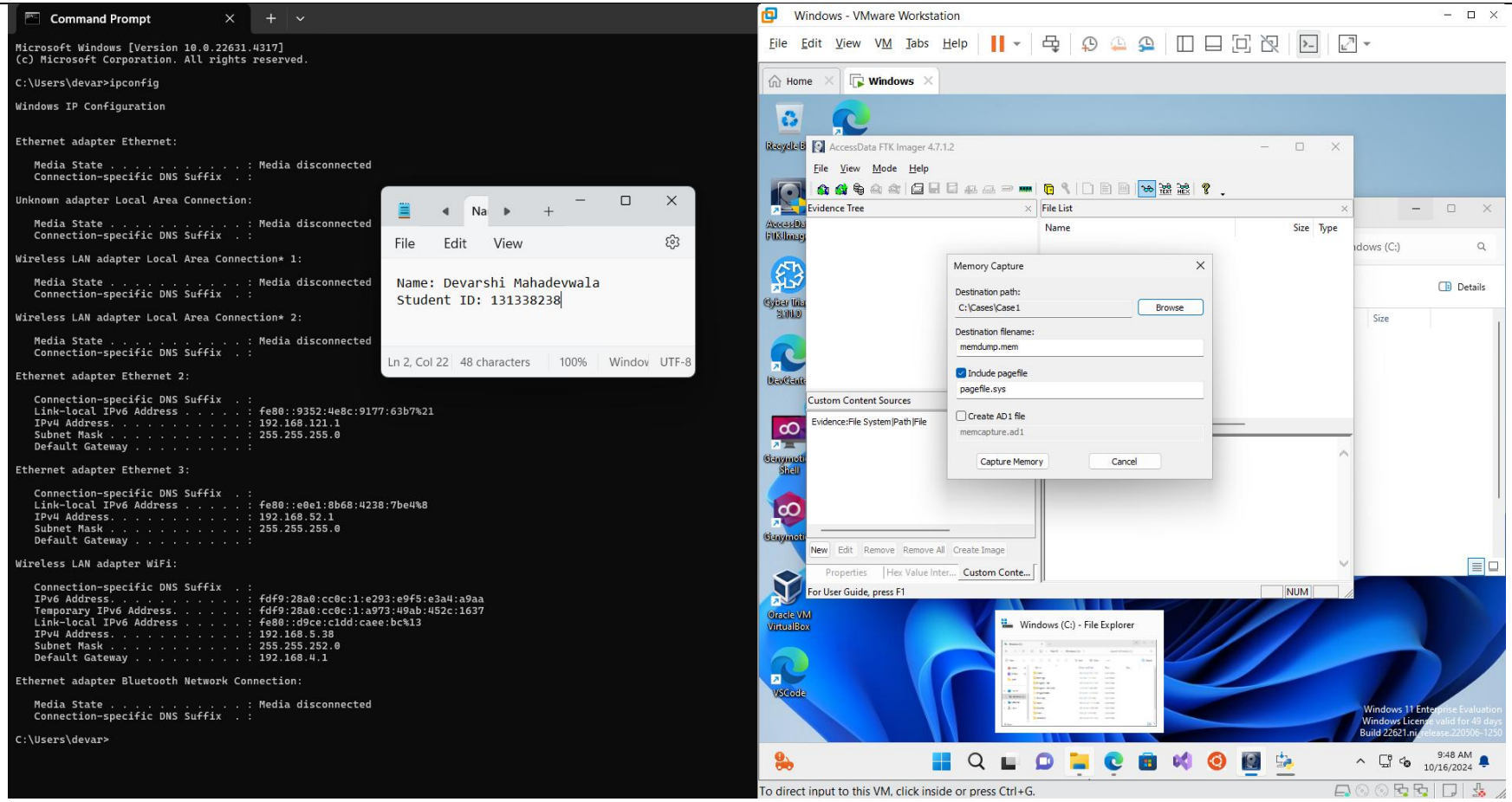
I also tried downloading rekall on the virtual environment. But there is a package named pypiwin32 which need to have the version 220 for recall to download, install, and run successfully. That package has been deprecated and there are currently no vendors that distributes that package, so I needed to scrape using rekall here.

The forensics lab has been setup with FTK imager, cyber triage, redline and volatility along with WSL ubuntu environment and python installed and up and running on our virtual environment. You can also download other tools like Belka soft to run memory image analysis.

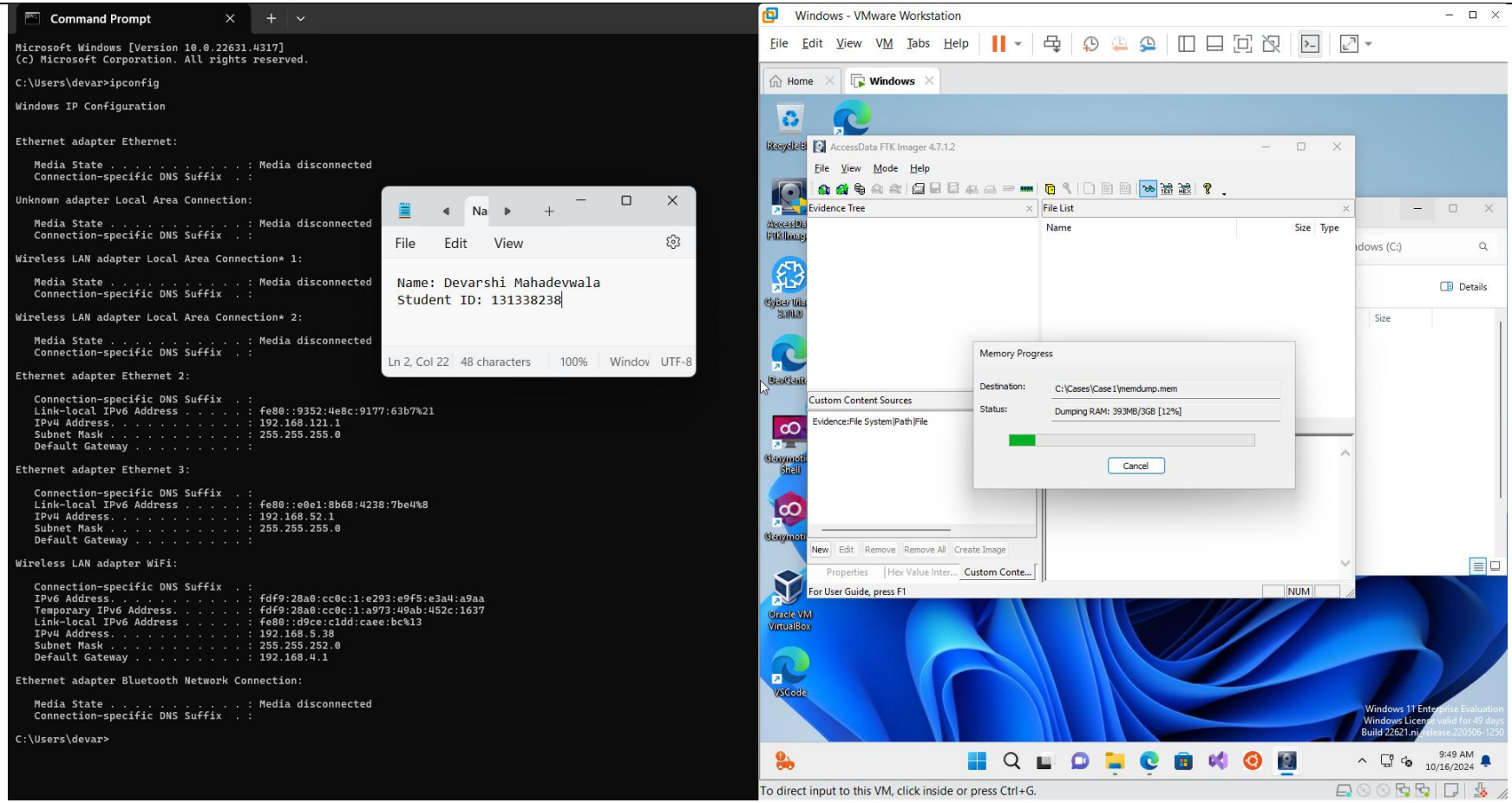
Now, we can start conducting the memory image analysis. But before that, let us capture a memory image from FTK imager.



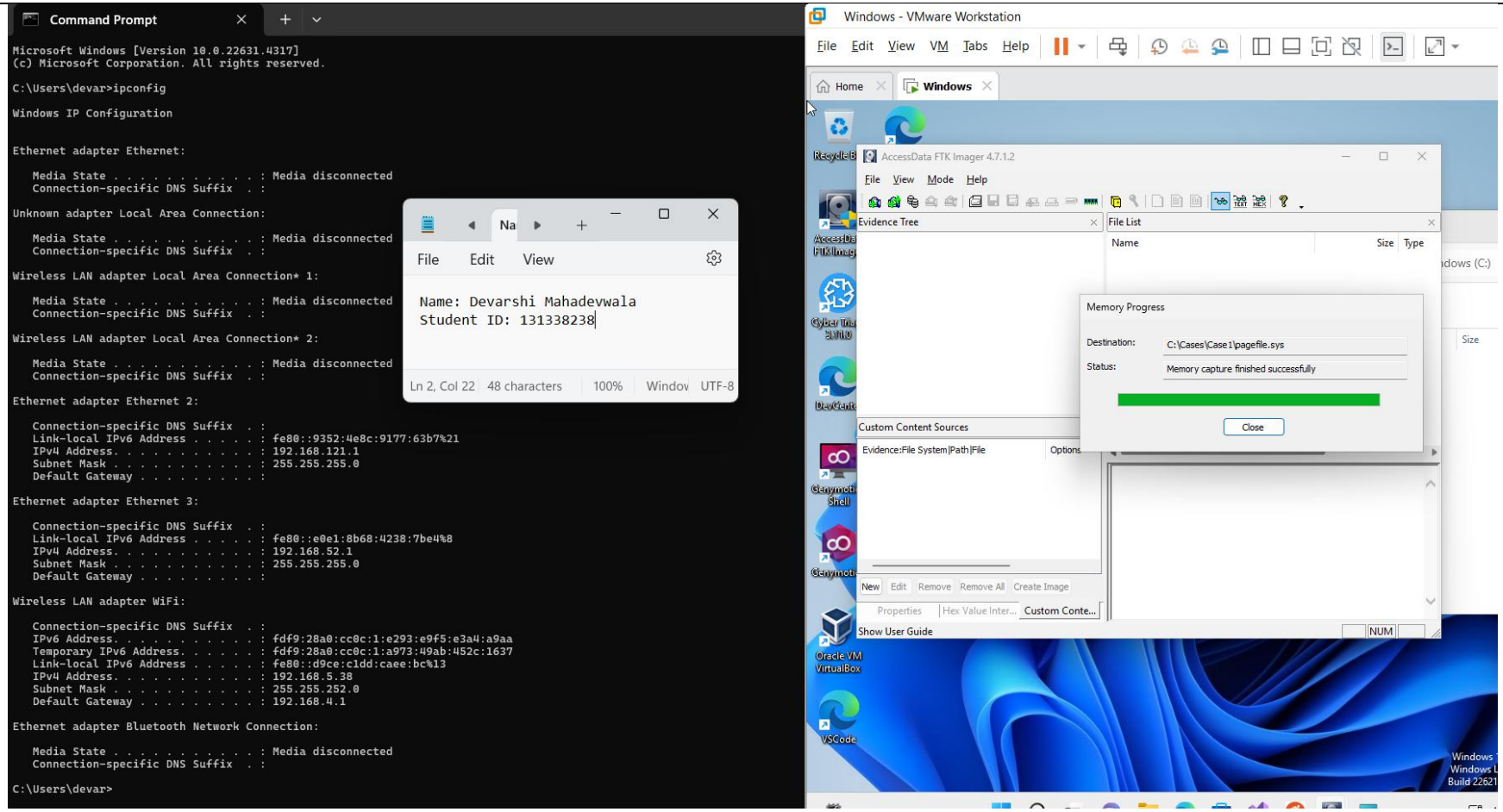
Start the FTK imager application and click on file on the taskbar. We need to select capture memory from it.



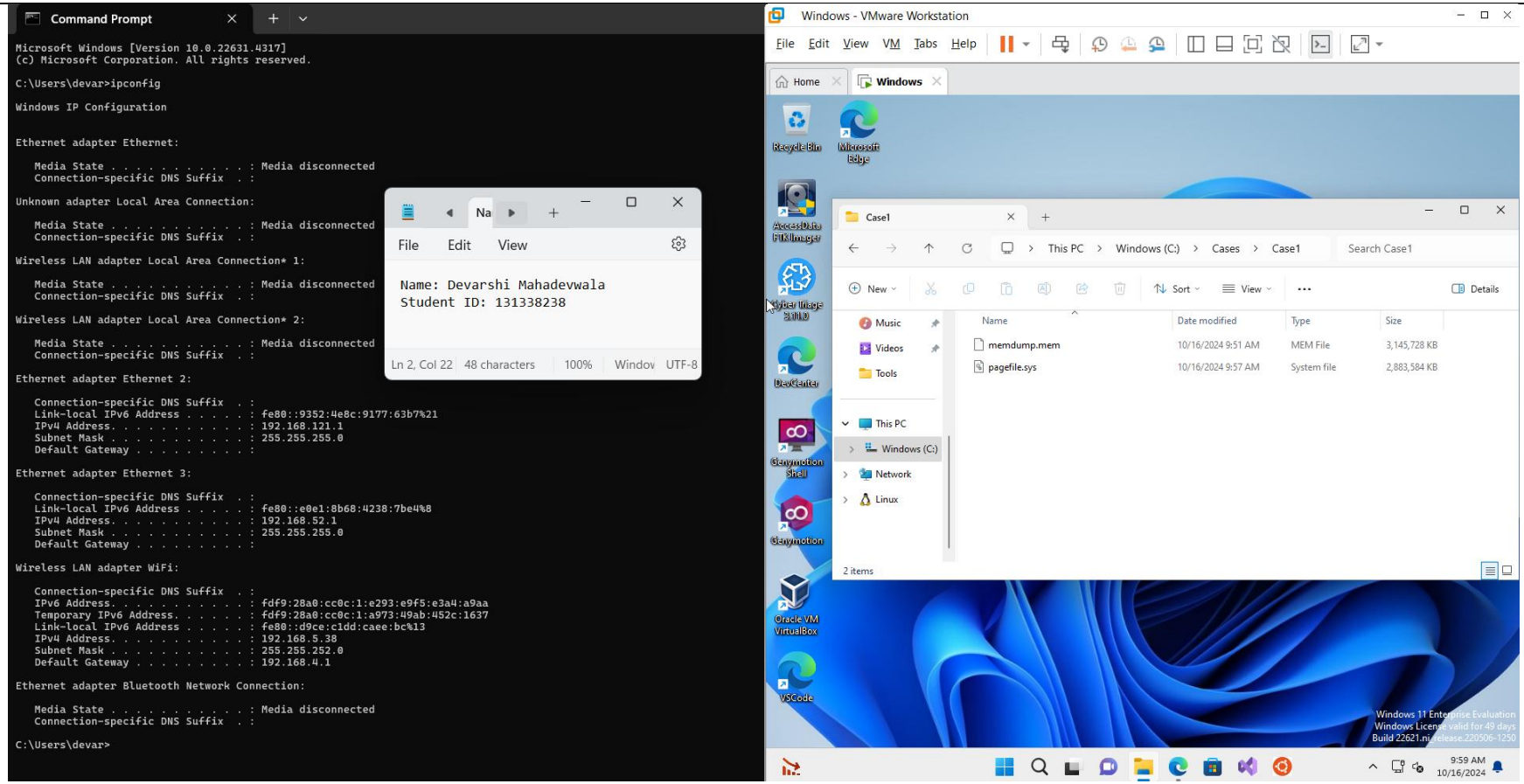
The memory capture wizard will ask us the name we want for our memory file and the destination of our memory image. I am storing my memory image in C:\Cases\Case1 as it is the dedicated directory where I will create cases for memory analysis.



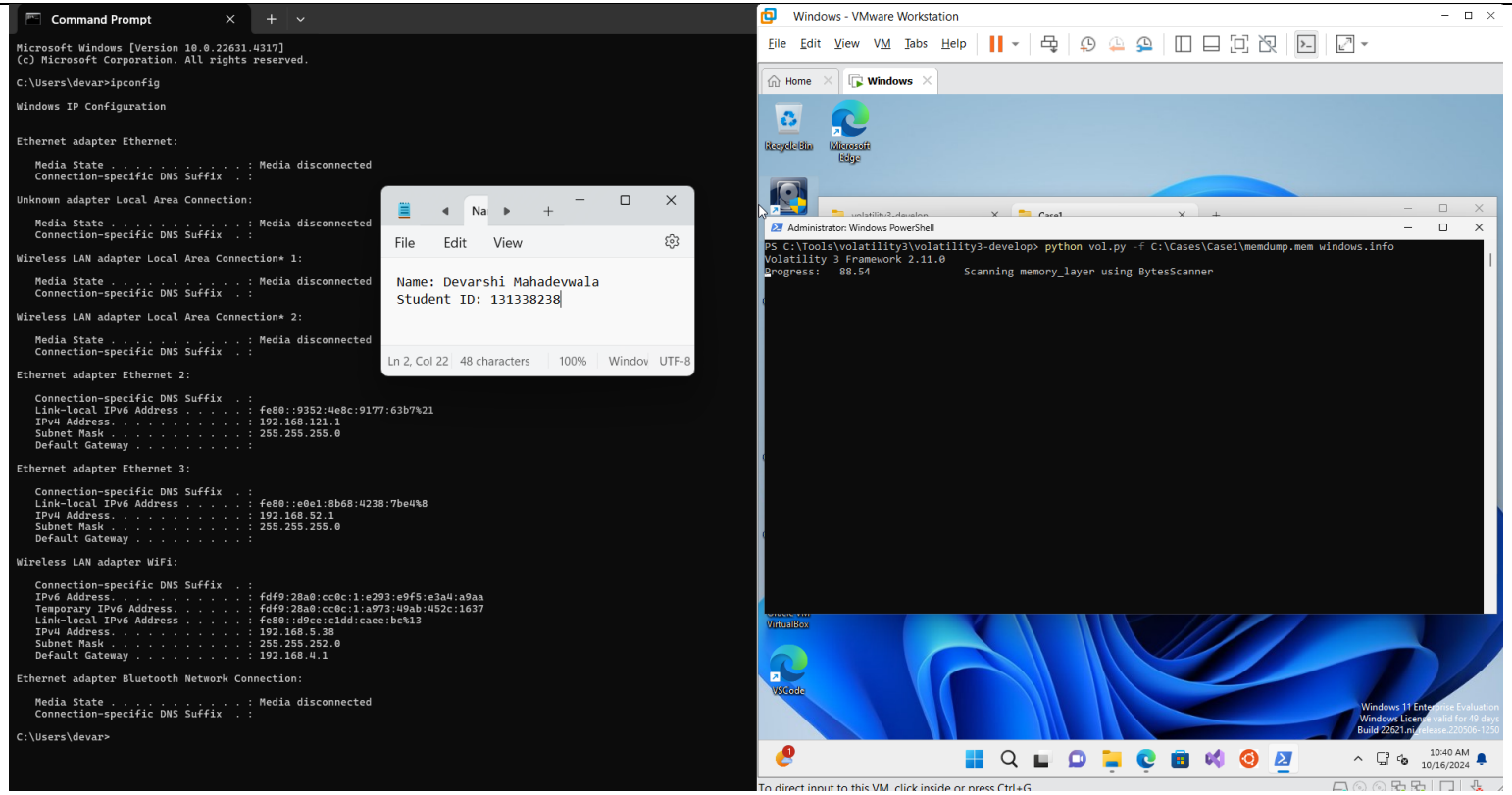
After clicking on capture memory, FTK imager will start capturing the image.



Memory has been captured, click on close and check up if the memory image has been captured at the specified location.



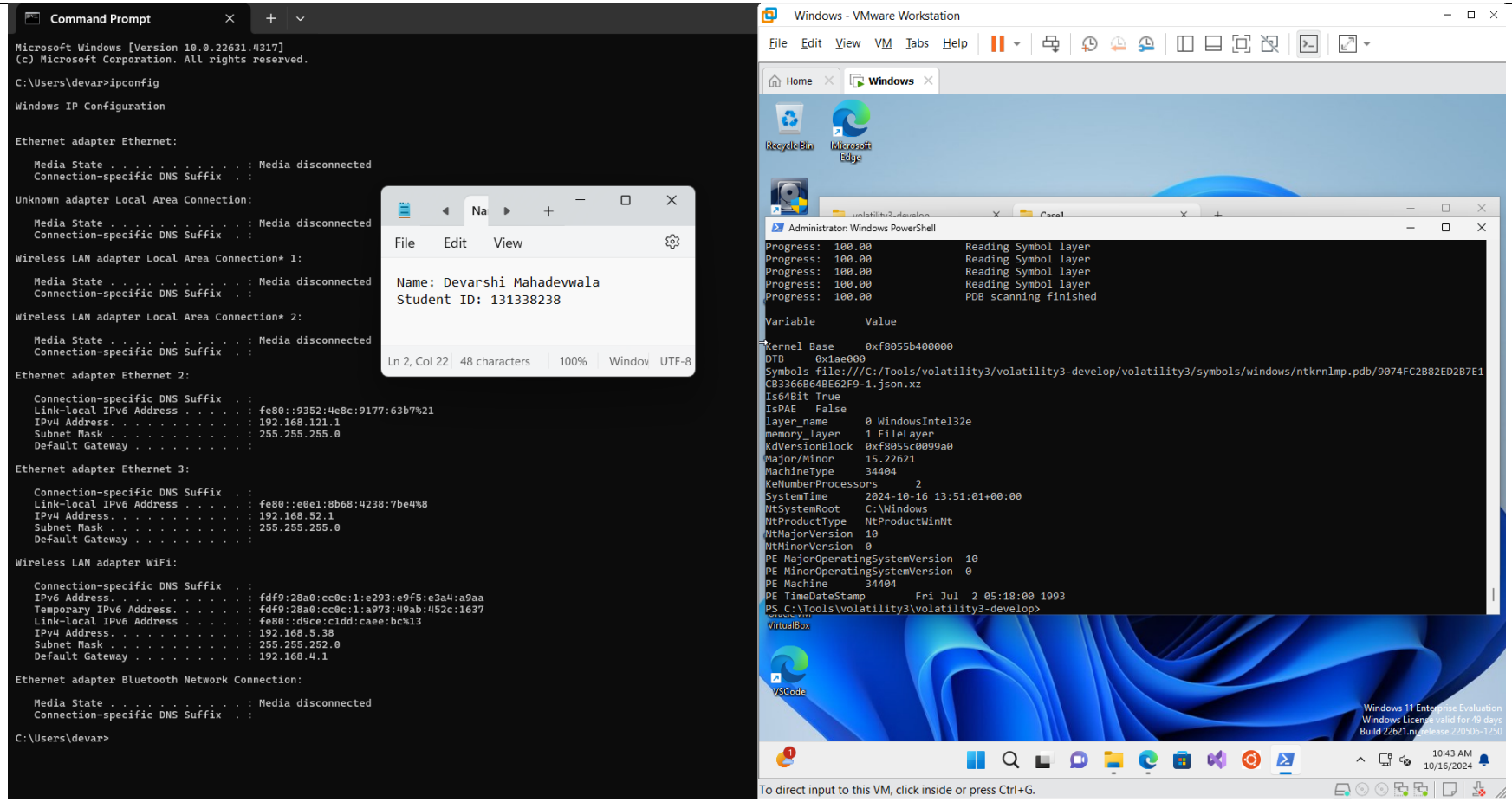
In C:\Cases\Case1, we can see the memdump.mem file indicating that memory capture was successful. Now let us run our tools for memory image analysis.



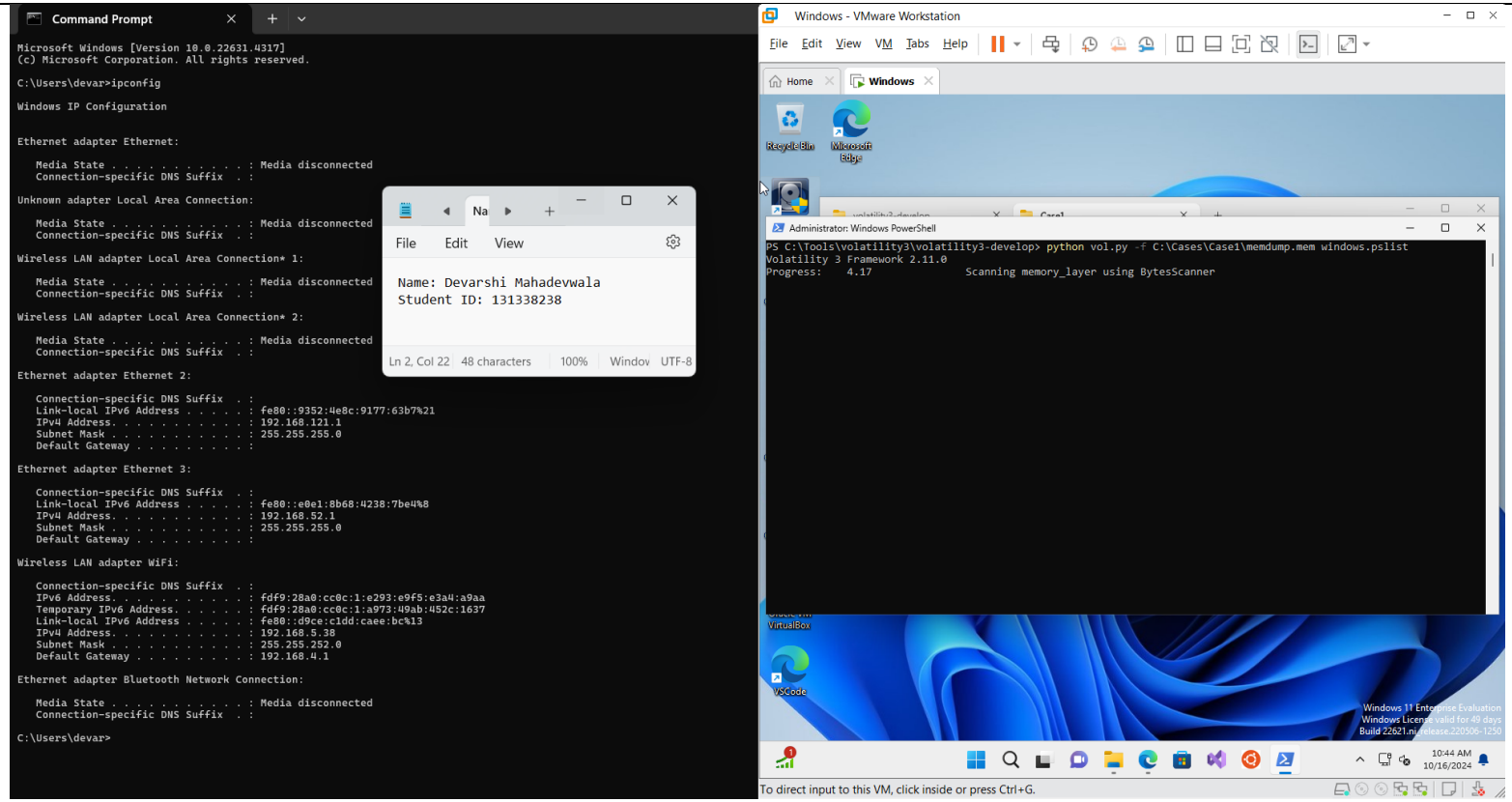
Let us run our first scan from volatility on the image file that we captured. Starting off with a basic scan, I ran a simple windows information step. When you run the 'windows.info' plugin in Volatility on a memory image from FTK Imager, it extracts and displays various details about the Windows operating system. This includes:

- The version and build number of Windows
- The service pack level installed
- The system root directory
- The username of the logged-in user
- The current process ID and name
- The system uptime
- Information about any system crashes or blue screens

This plugin provides a quick and easy way to gather essential information about the Windows system from the memory image, which can be useful in digital forensics and incident response investigations.



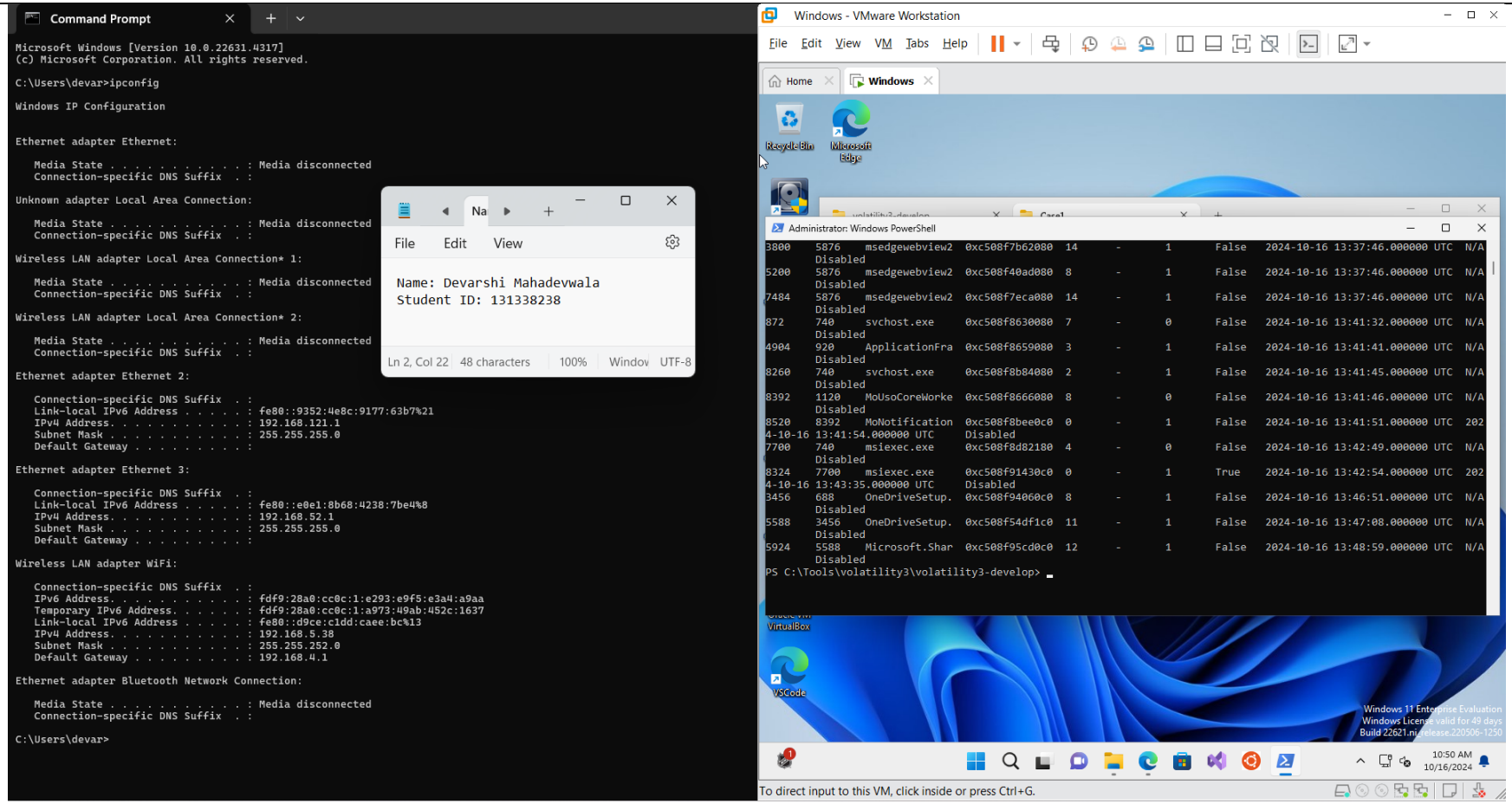
First scan has been completed and the details of those scans can be clearly visible from the above screenshot.



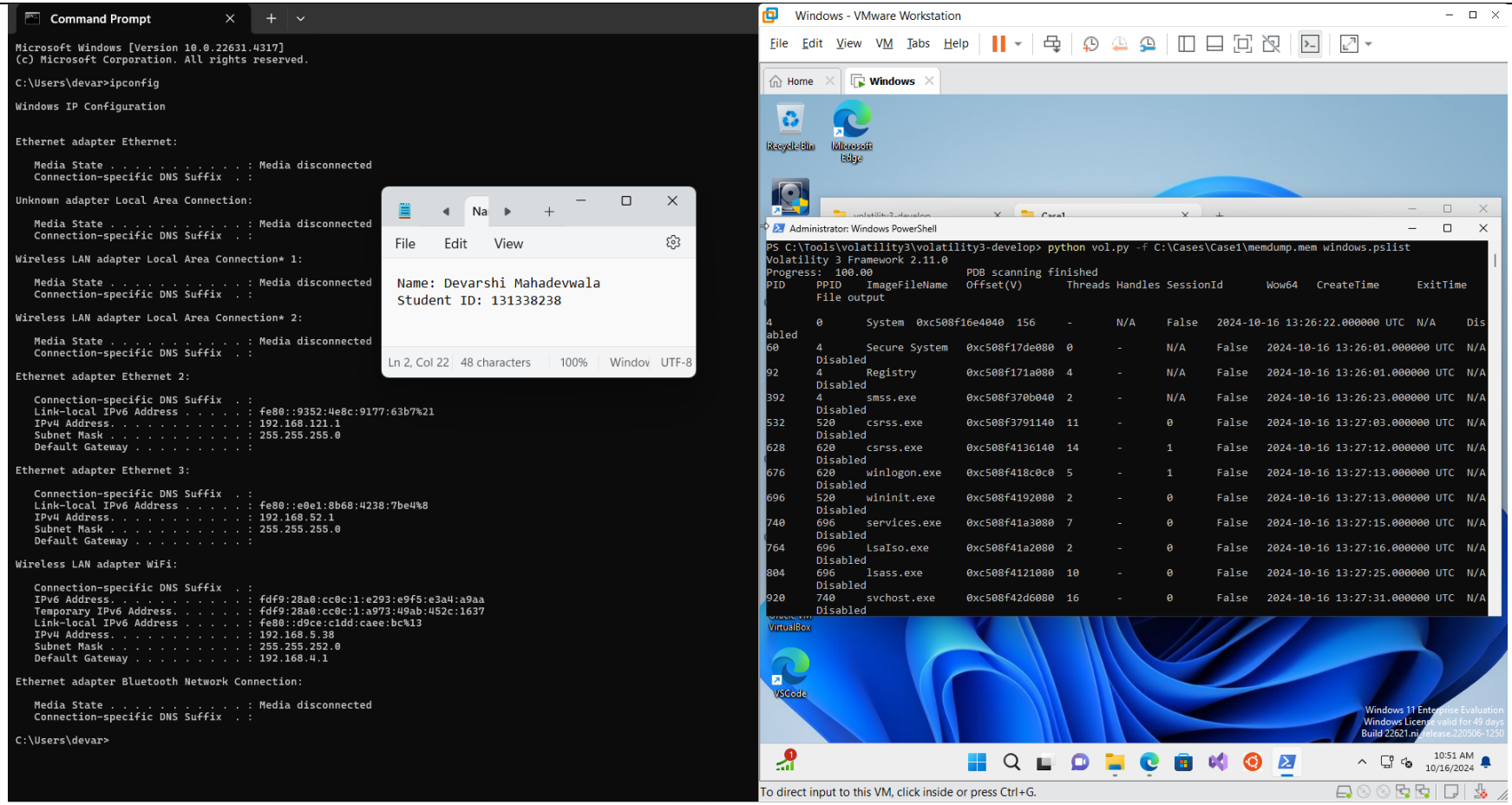
Now, I am running a windows.pslist scan on the memory image. The windows.pslist plugin in Volatility generates a list of processes running on the Windows system at the time the memory dump was captured. This includes:

- Process ID (PID)
- Process name
- Parent process ID (PPID)
- Process start time
- Process exit time (if applicable)
- Thread count
- Handle count
- Memory usage (private bytes, working set, and peak working set)

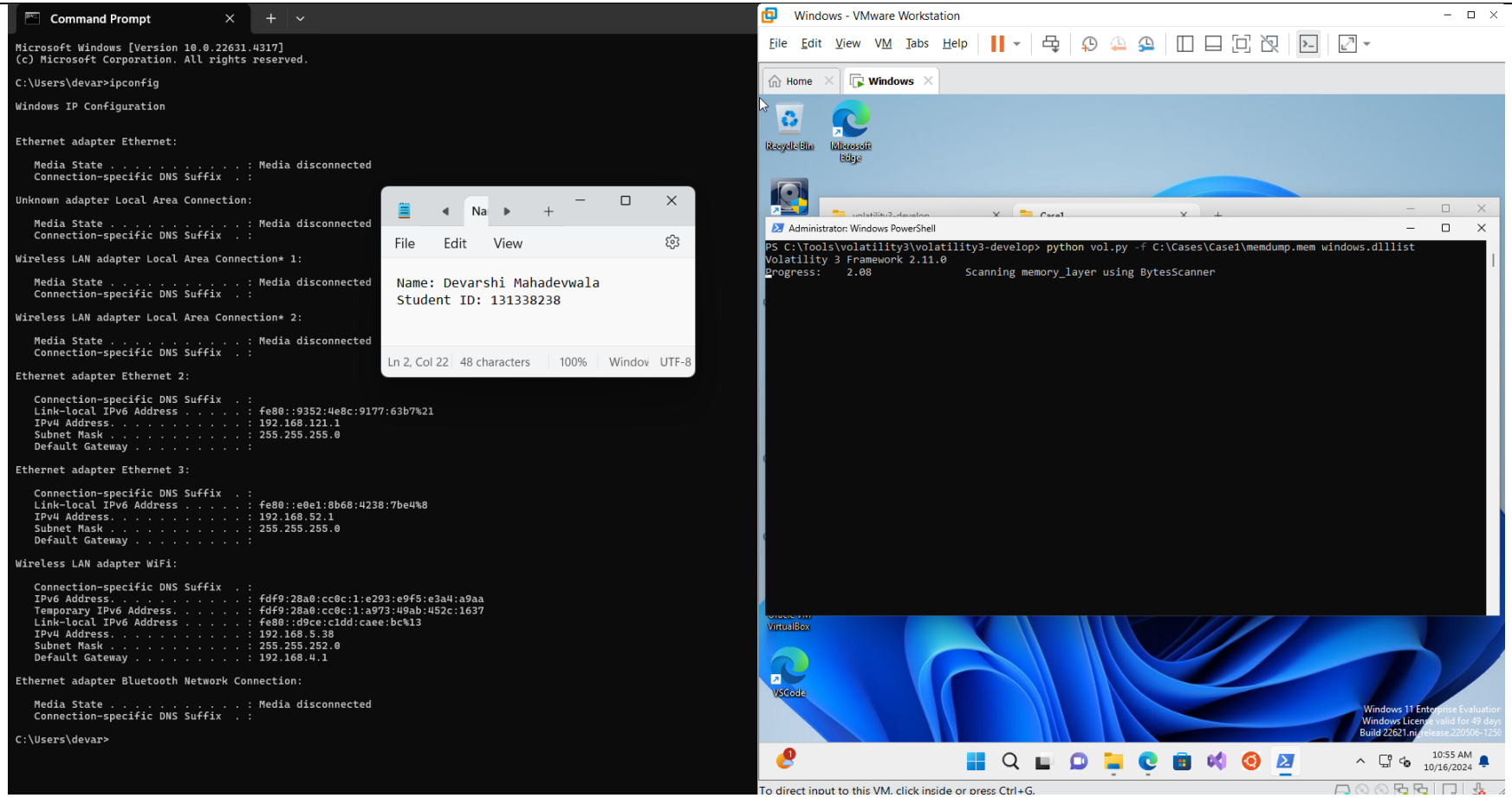
This plugin provides a comprehensive view of the system's process landscape, allowing investigators to identify and analyze running processes, including potential malware or suspicious activity.



We can see all the specified list on the result of the scan when completed in the screenshot above. It gives us a very detailed list of processes and their specifics which we can go through to run an analysis later.



Above we can see the resulting column wise specification of windows.pslist analysis.

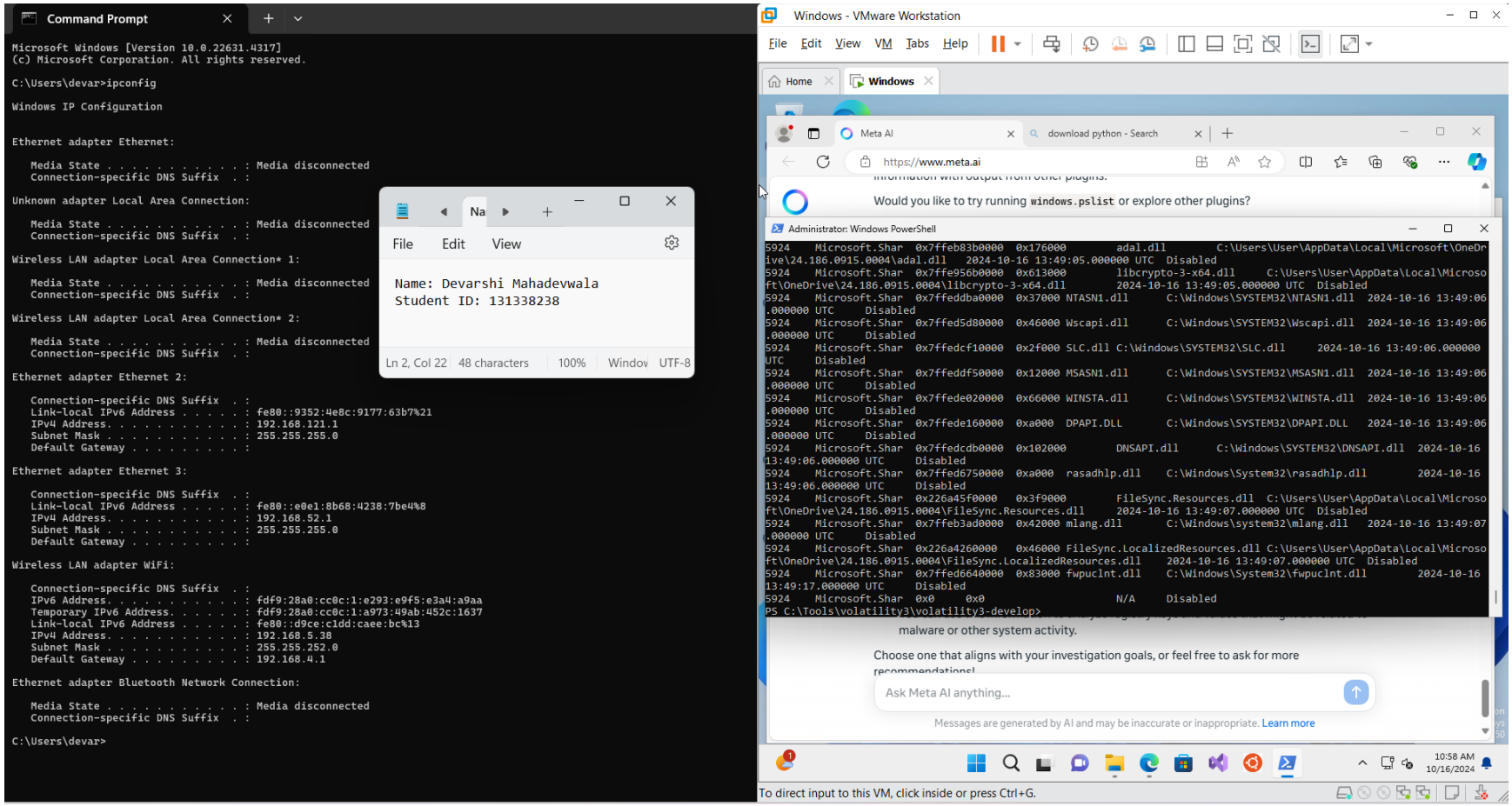


Next scan that we will conduct is a dll list scan. The windows.dlllist plugin in Volatility enumerates the DLLs (Dynamic Link Libraries) loaded by each process running on the Windows system at the time the memory dump was captured. This includes:

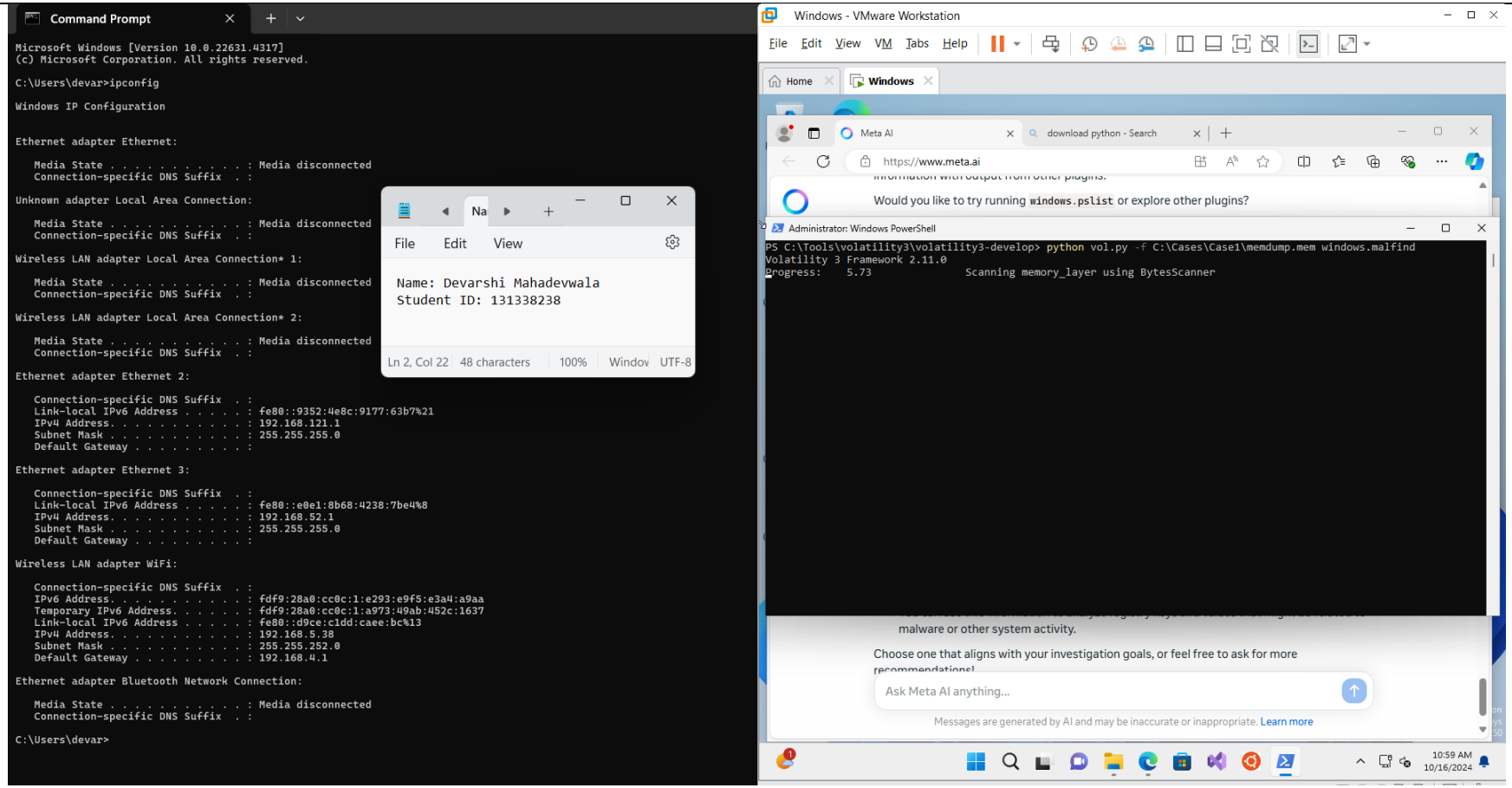
- Process ID (PID)
- Process name
- DLL name
- DLL base address
- DLL size
- DLL load count
- DLL load time

This plugin provides a detailed view of the DLLs used by each process, allowing investigators to:

- Identify potentially malicious or unknown DLLs
- Analyze DLL load patterns and dependencies
- Detect potential DLL hijacking or injection attacks
- Investigate DLL-related system crashes or instability



We can see all of the specific details of each **dll** process running on windows in the above screenshot.



Final scan we conduct is the windows.malfind. The windows.malfind plugin in Volatility is designed to detect and identify potential malware artifacts in the memory dump of a Windows system. This includes:

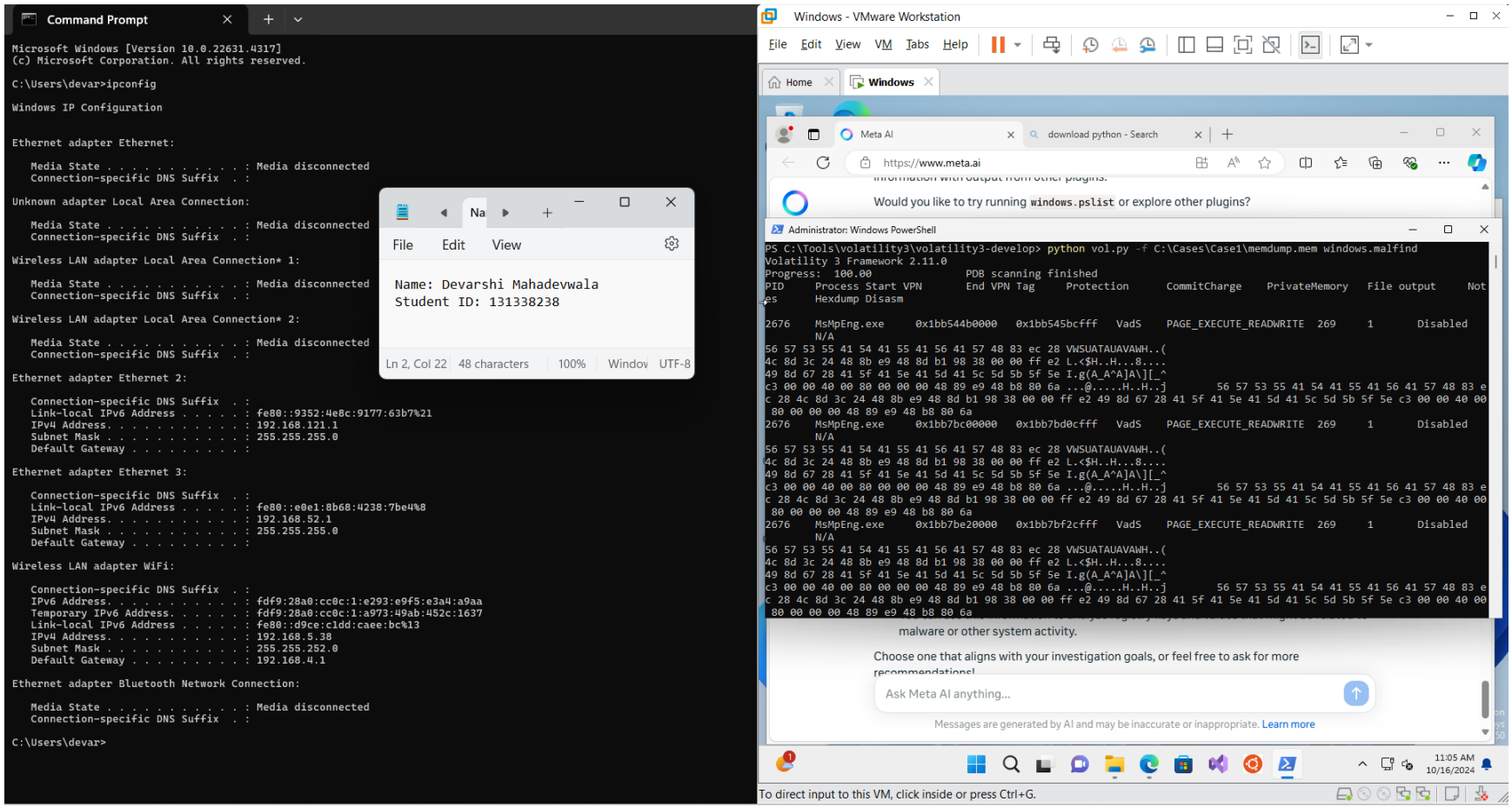
- Suspicious memory allocations
- Unusual API hooks
- Hidden or masked processes
- Mysterious system calls
- Anomalies in system memory

This plugin uses various techniques to identify potential malware, including:

- Heuristics-based detection

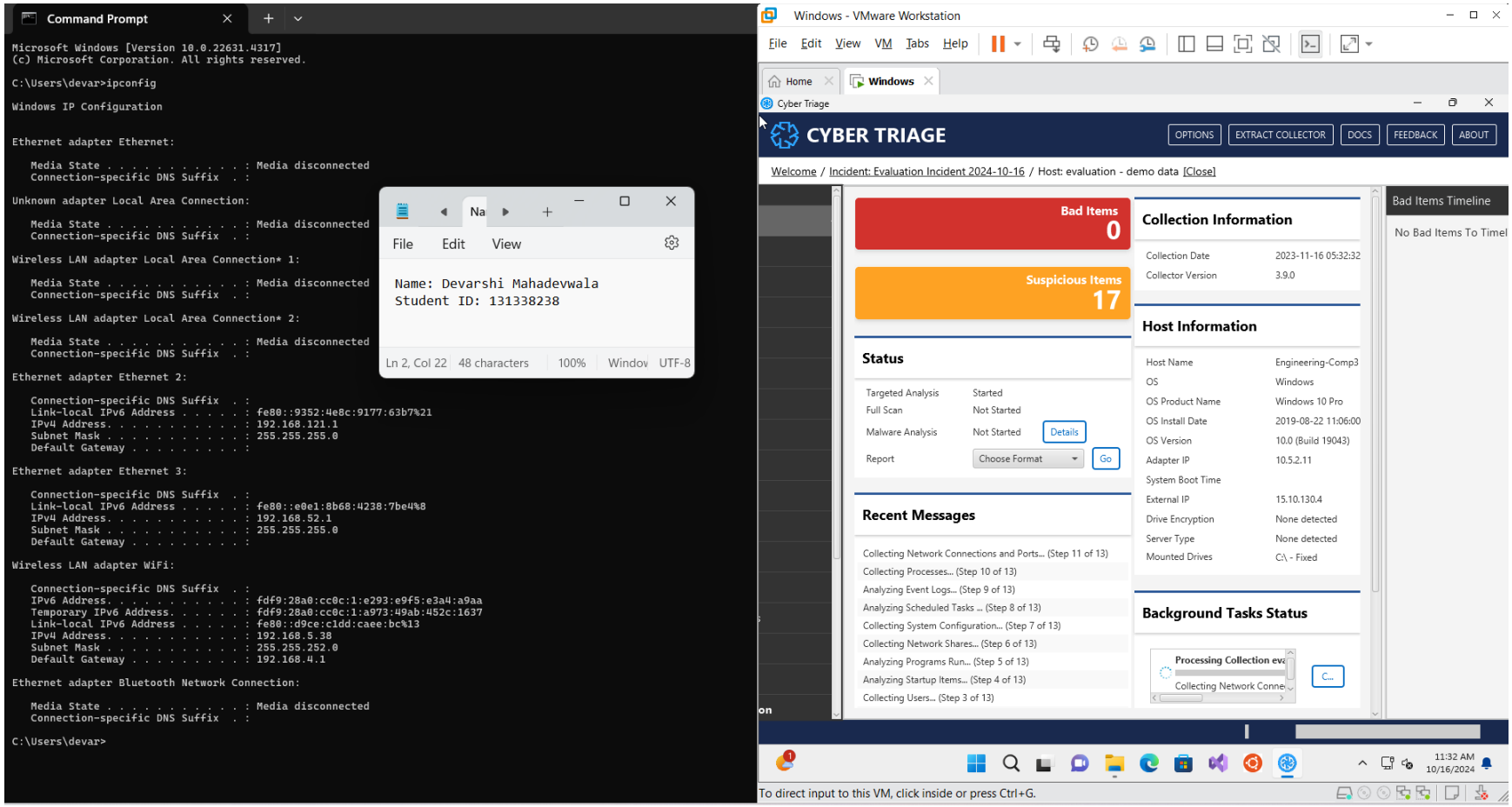
- Signature-based detection
- Anomaly-based detection

By running windows.malfind, investigators can quickly identify potential malware activity and take further action to contain and eradicate the threat.

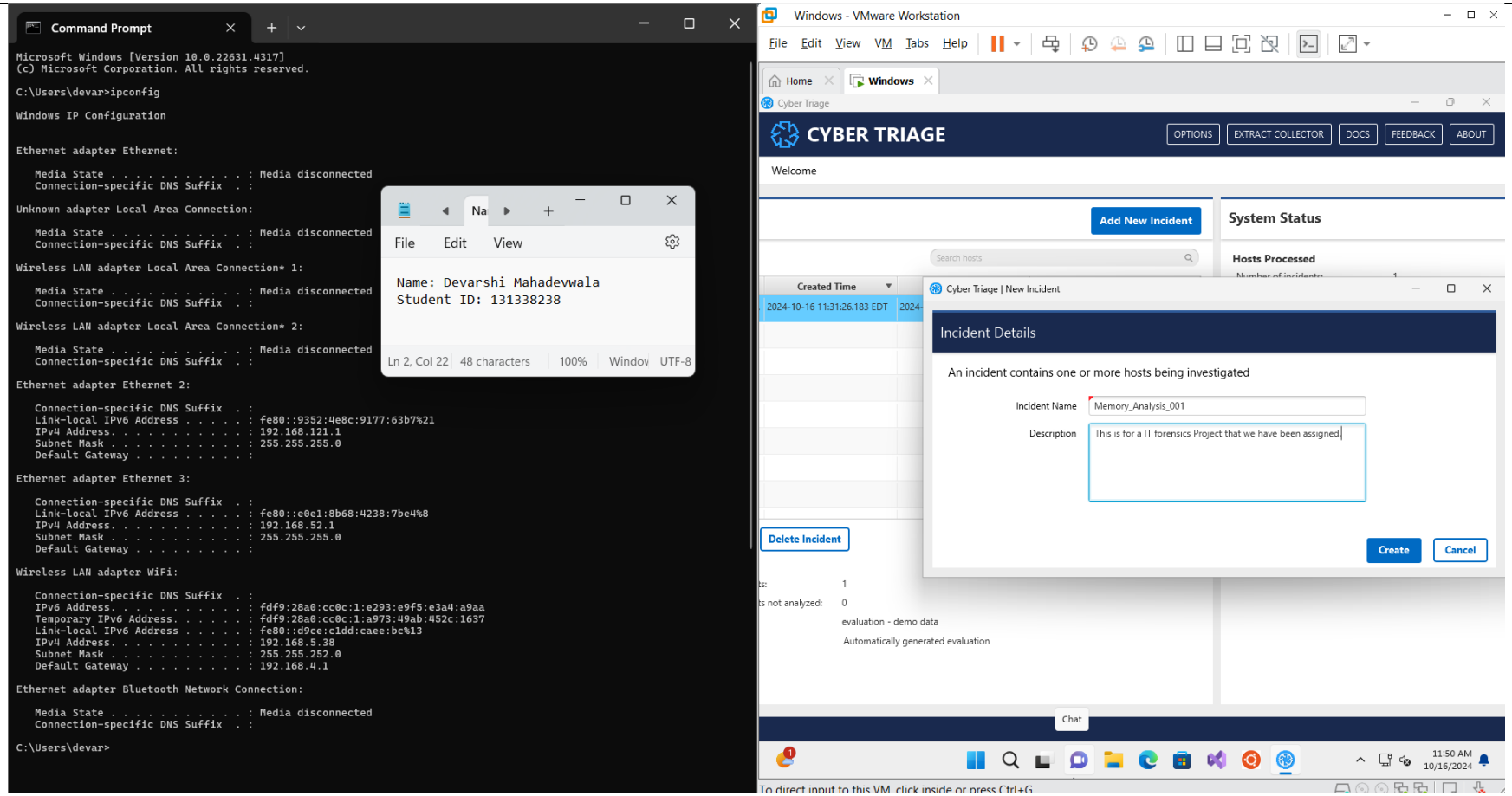


The resulting screenshot shows us the potential malware artifacts of the memory dump with PID process and other details.

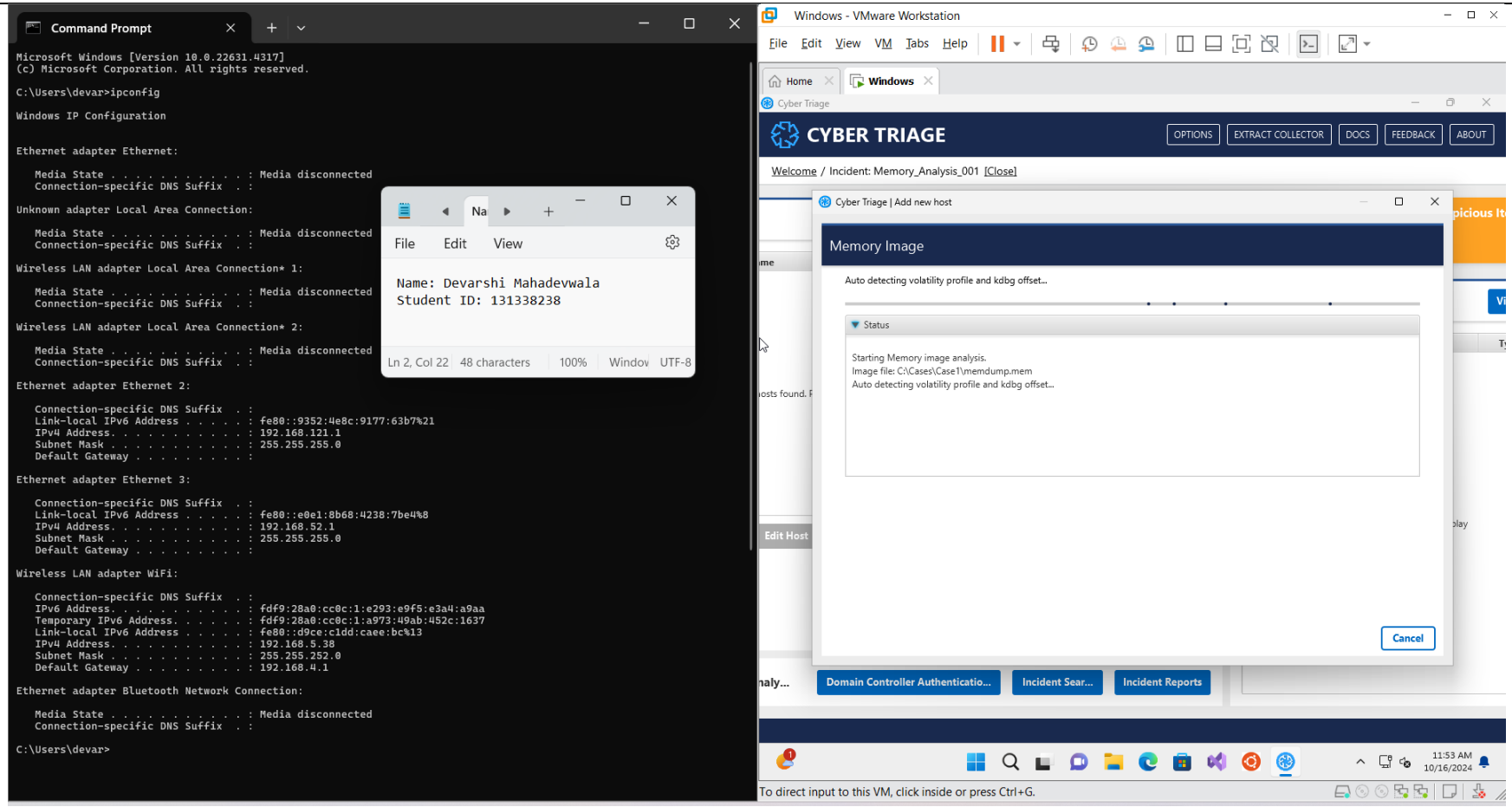
Let us now look at cyber triage.



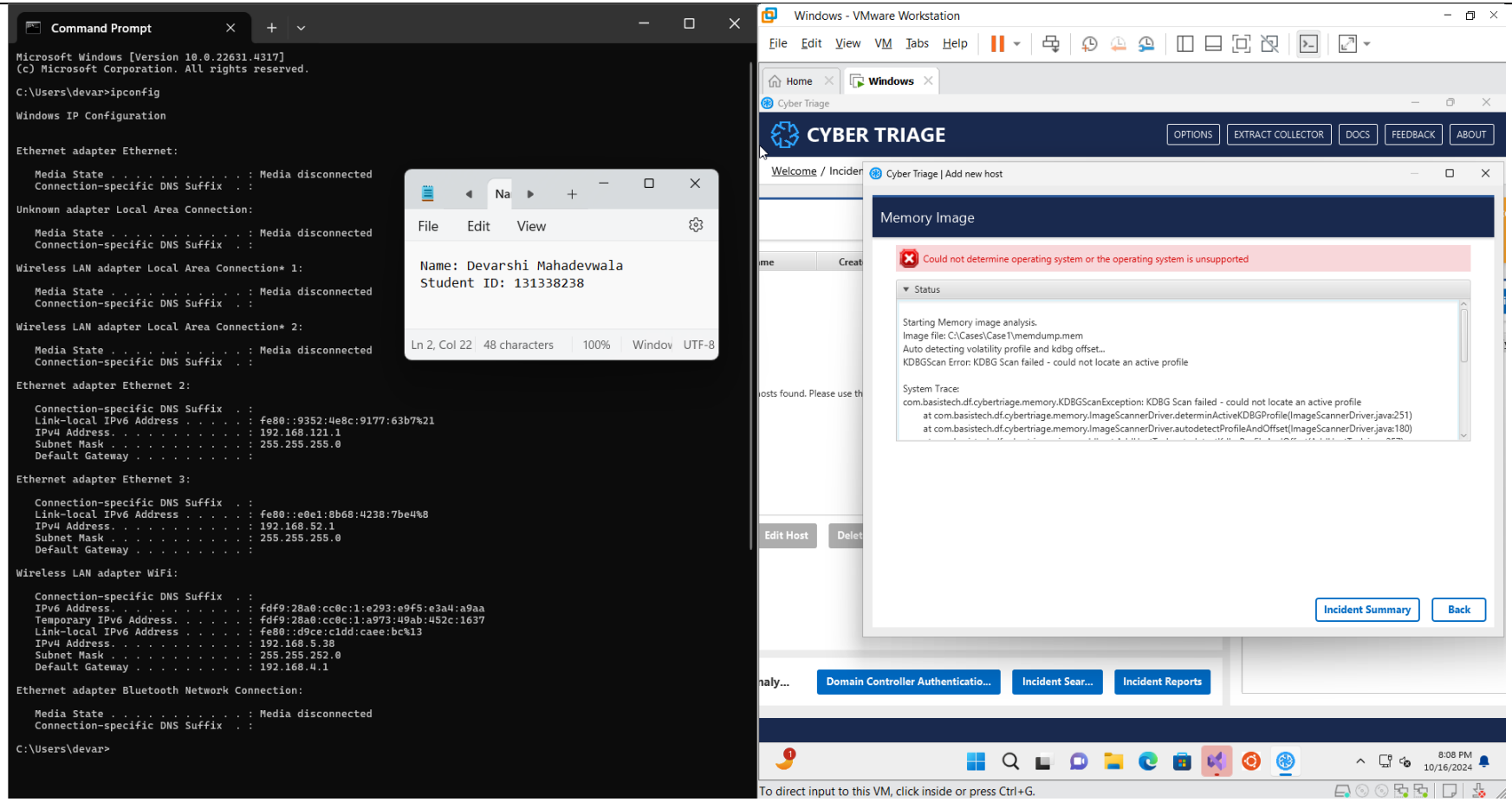
Here, cyber triage is running a scan on the host to find any bad or suspicious items.



We need to create a new incident to conduct analysis as this will create a proper pathway for us to navigate to the scans we have conducted.

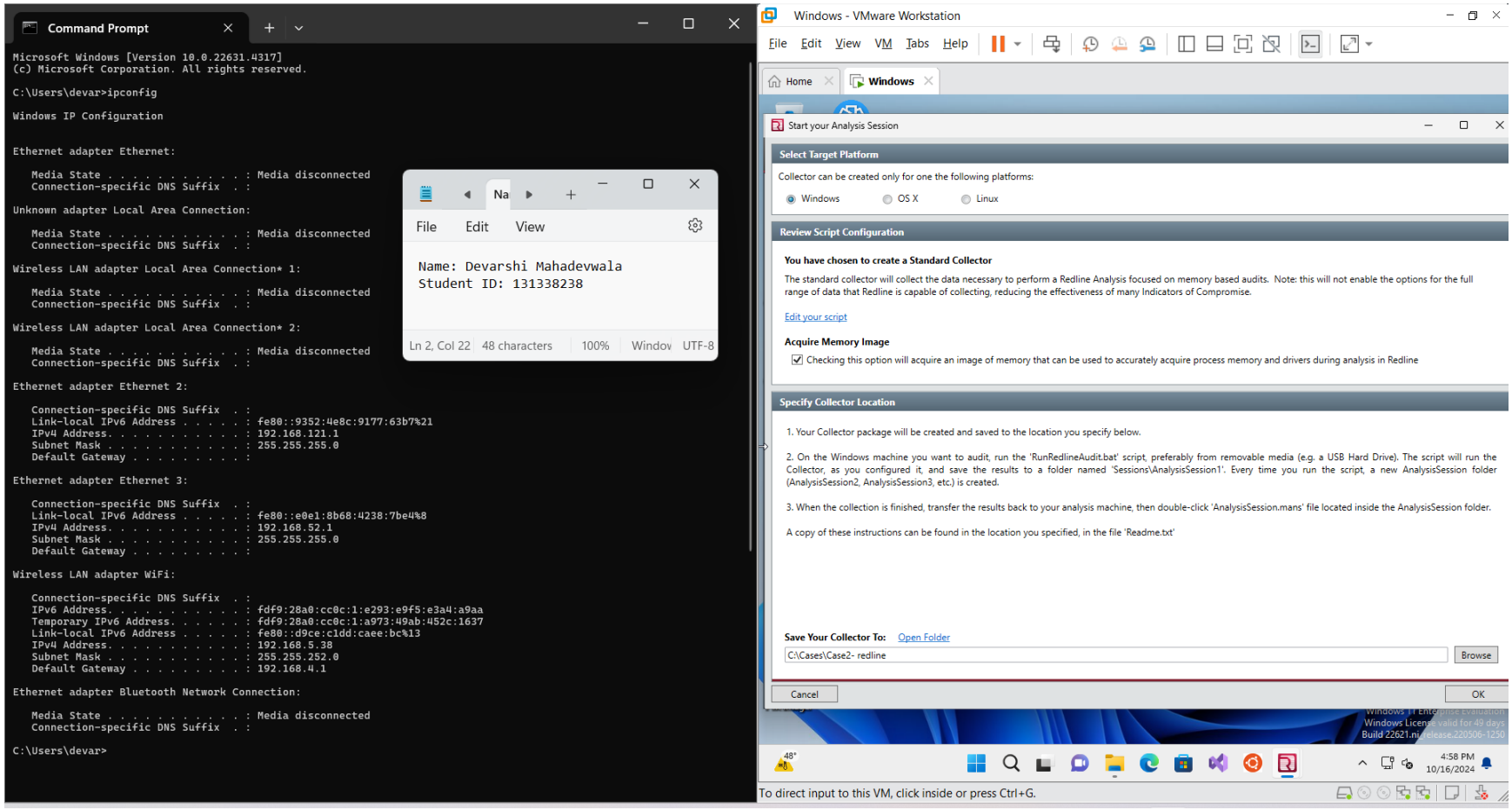


Now, we start our scan on the FTK image file. The application is currently running the scan on our image file.

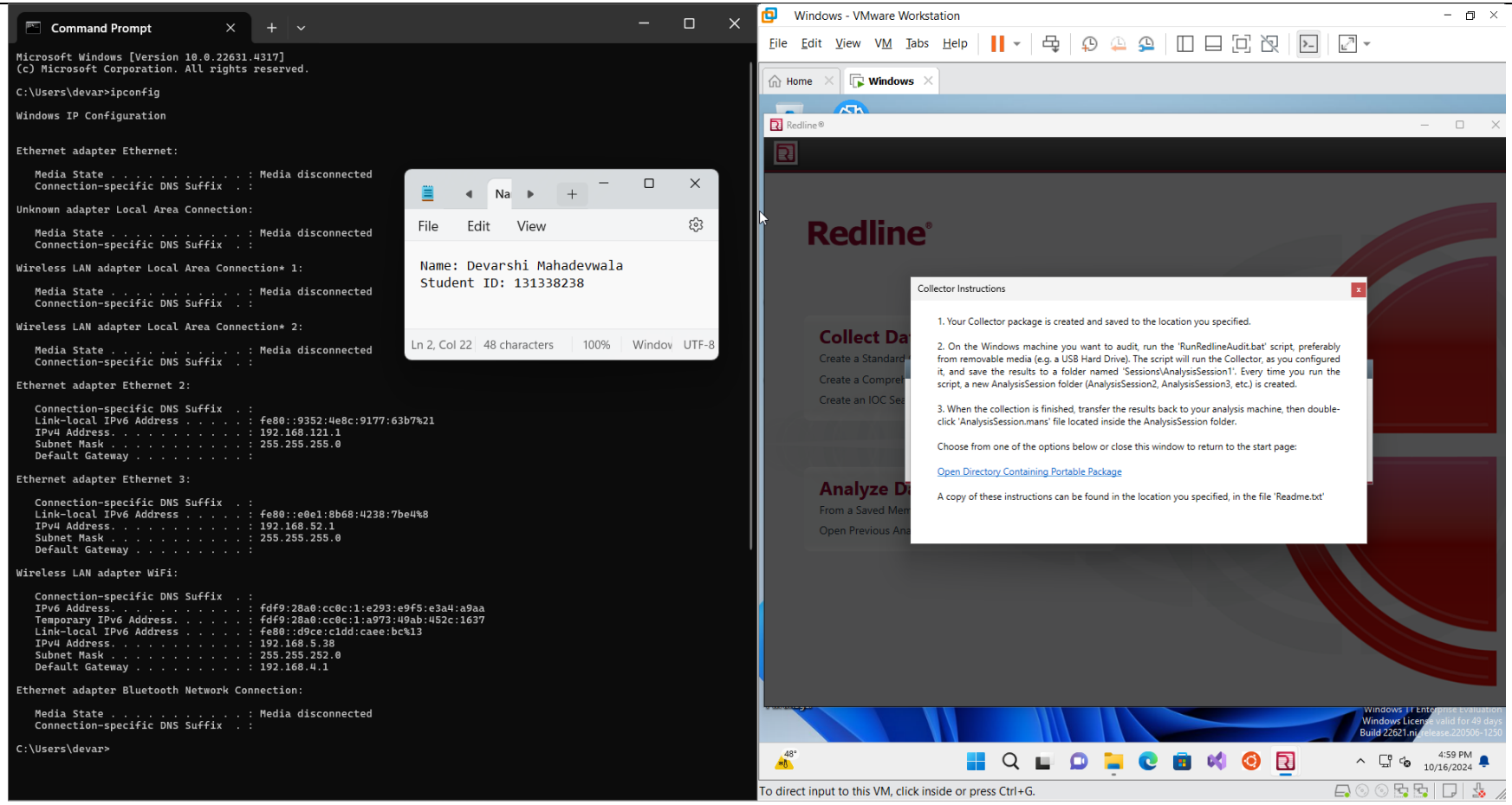


Usually, at this point the memory image analysis should be complete and give you a detailed report about your image file. But I was getting some error about which operating system is supported. I tried this for 5-6 times with every bug fix that I can conduct but it gave me the same result every time.

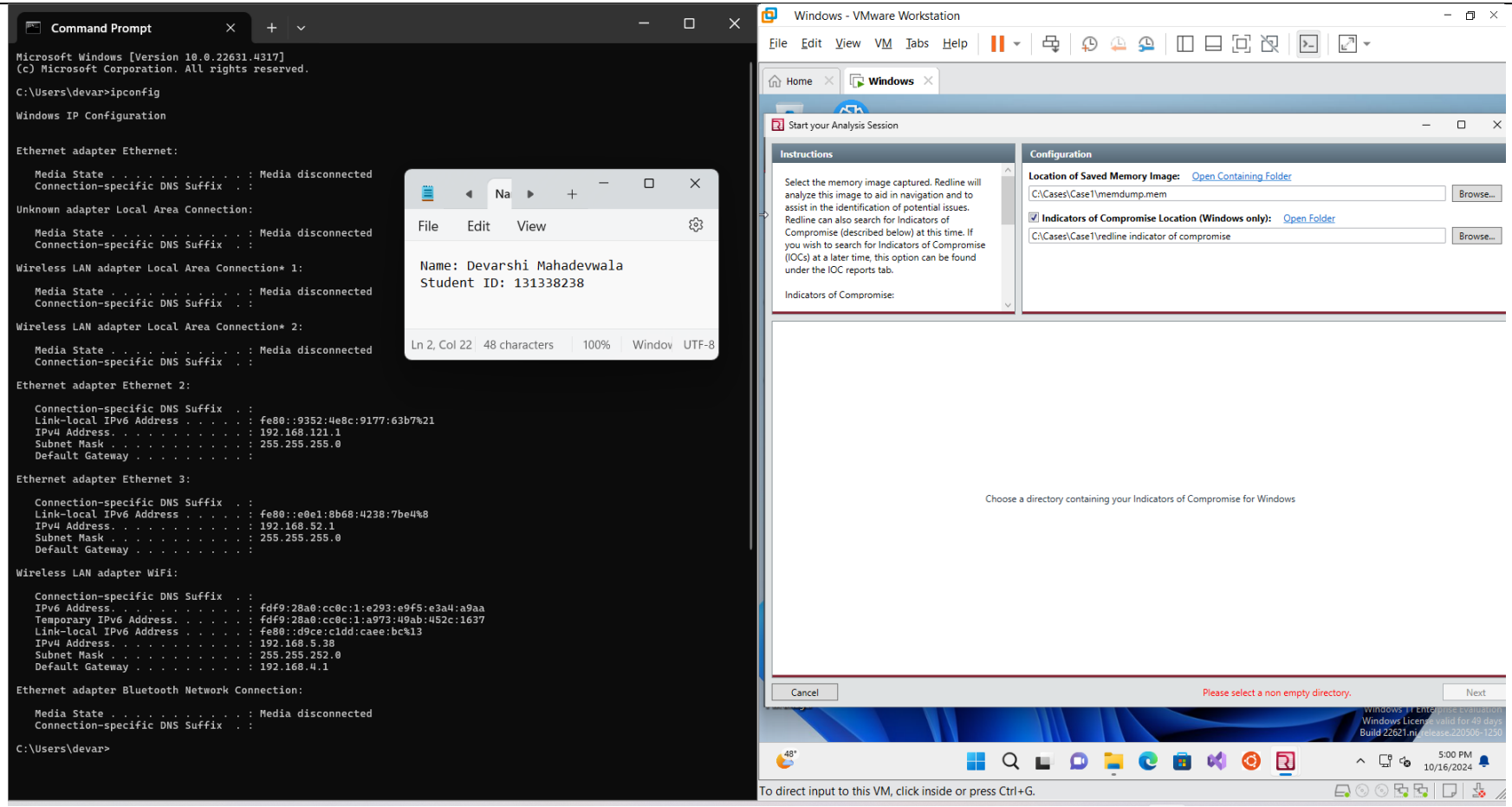
Now, let us move on to redline.



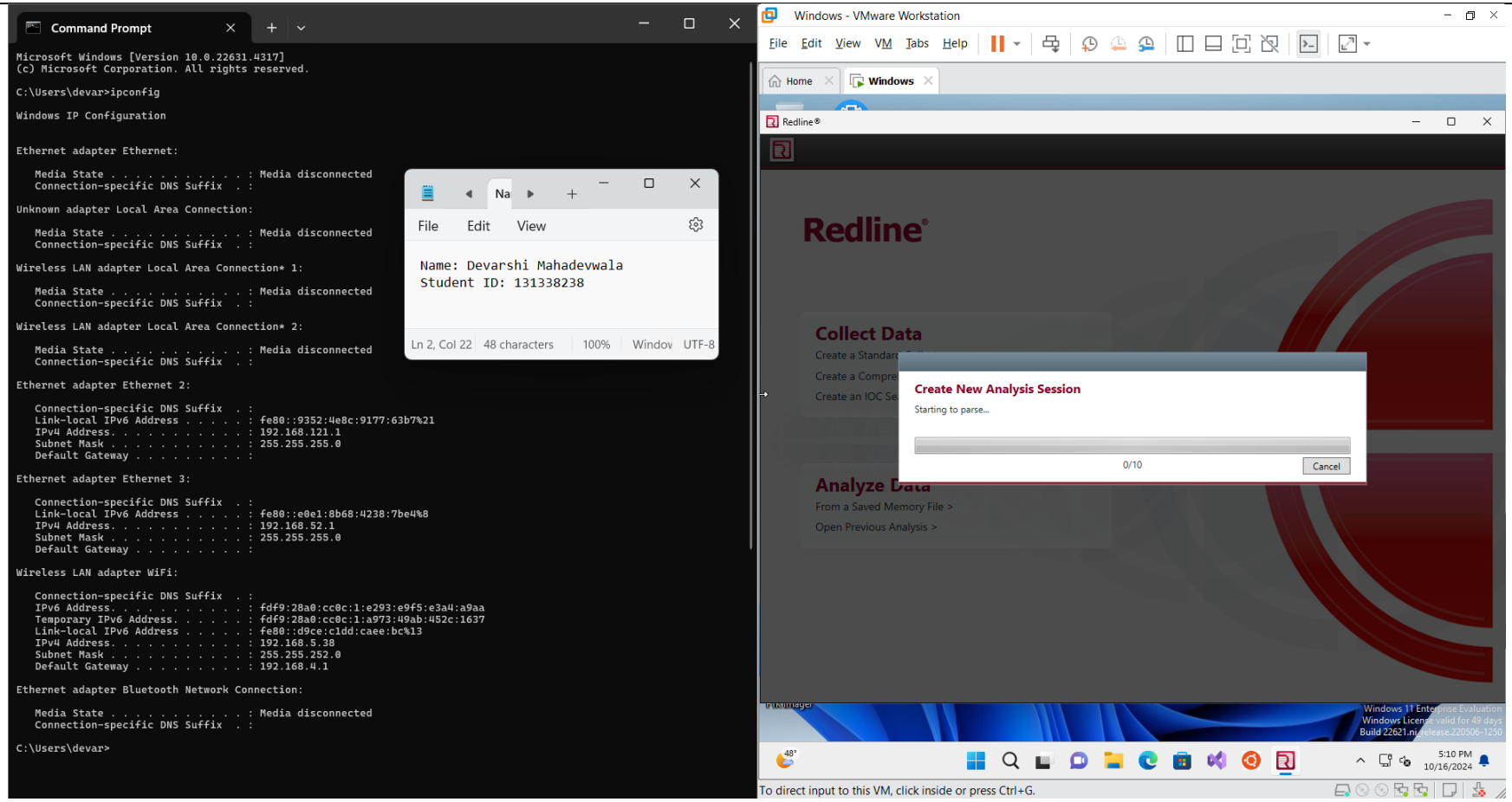
We can collect data from redline so we will create a new case in our C:\Cases to store our data.



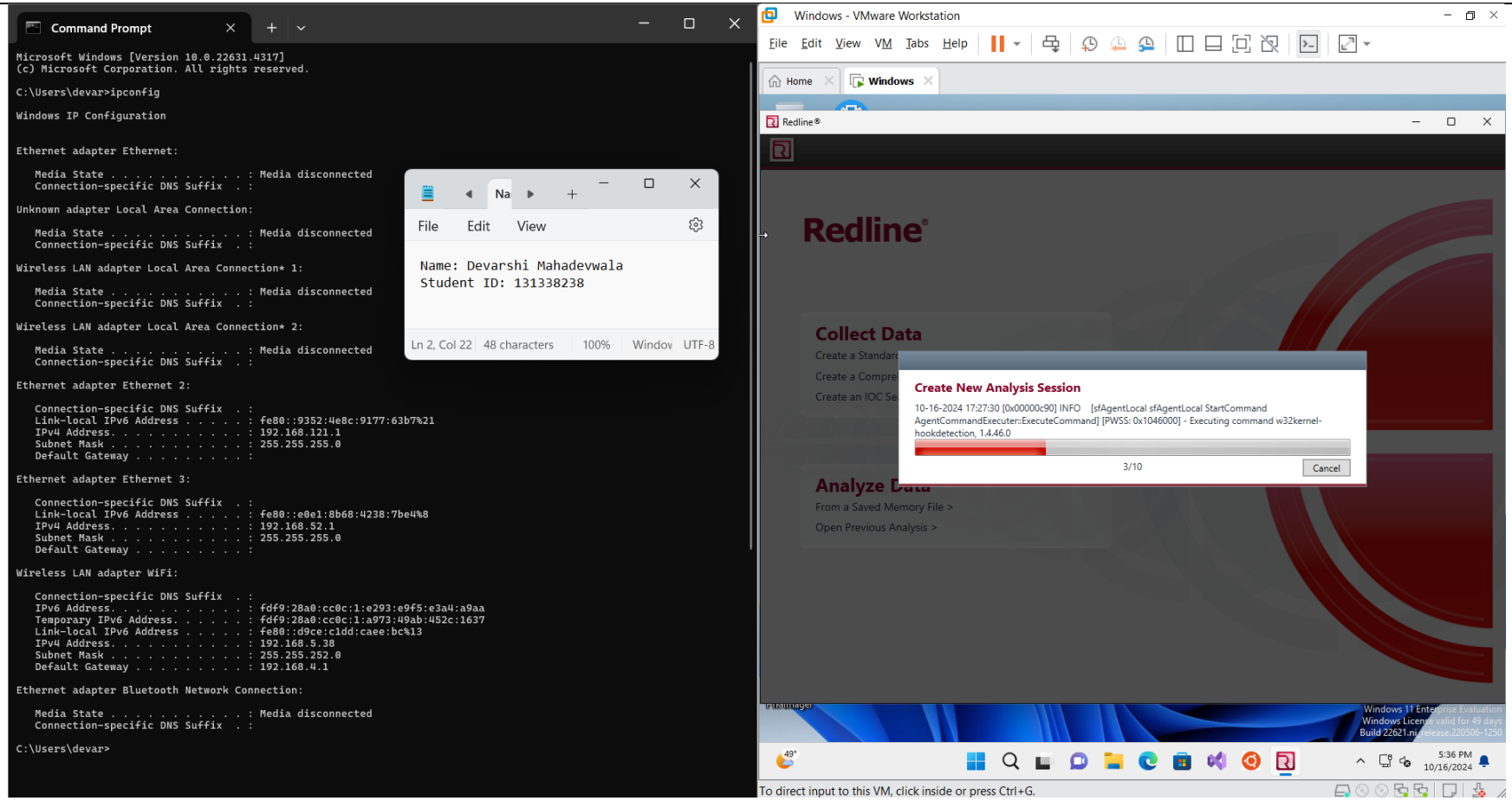
The package has been collected and stored at the specified destination.



Now to conduct a memory analysis on our memory image, we need to start our analysis and specify the file on which we need to conduct our analysis.



Our analysis is running, and this will create a new session for us to analyse our packages.



The scan is in process, it will take some time.

I faced a problem. When the analysis completed, I needed to include some indicators of compromise (IOC) which I tried solving but I found no way around this as the IOC needed was specifically for the memory image. Thus, I couldn’t get redline to work too.

Personal Learning Experience Report on Memory Analysis Workstation Setup

1. Introduction to Memory Analysis

Setting up a forensics workstation was an enlightening experience, providing insight into the tools and techniques used for memory analysis. Memory forensics is critical for uncovering malicious activity and gaining a deeper understanding of an incident. The process allowed me to experiment with tools like Redline, Cyber Triage, and Volatility, offering hands-on experience in real-world digital forensics.

2. Building the Workstation

The first step involved setting up a forensic workstation following a guide from Blue Cape Security. The process included configuring the necessary software, setting up the hardware, and ensuring a stable environment. This setup was essential for preparing a dedicated machine for forensic analysis, providing a controlled environment for working on memory analysis tasks without compromising the system's integrity.

Challenges included navigating software configurations and troubleshooting dependencies. For example, installing Python for Volatility required specific library versions. Despite these hurdles, configuring the workstation taught me the importance of meticulous preparation and checking for compatibility with the software.

3. Memory Image Acquisition

I followed guidance from DFIR Madness to capture a memory image of my machine. The process required using tools like FTK Imager to generate a .mem file. This experience emphasized the importance of maintaining the integrity of the data, ensuring no alterations were made during acquisition.

Capturing memory from a live system proved to be an engaging task, as it required careful handling to avoid tampering with potentially volatile data. The .mem file provided a complete snapshot of system memory, allowing for in-depth analysis using various forensic tools.

4. Memory Analysis Using Redline, Cyber Triage, and Volatility

Exploring multiple tools helped to understand different capabilities and approaches to memory analysis.

Redline: I used this tool for comprehensive memory analysis, which involved scanning the captured memory image for potential indicators of compromise. Its user-friendly interface and detailed reports made it easier to identify suspicious processes and anomalies.
Cyber Triage: This tool allowed for automated analysis, which quickly highlighted potentially malicious activities. The incident-based approach of Cyber Triage streamlined the investigation process, providing quick insights into various artifacts.
Volatility: As an open-source tool, Volatility provided more control over memory analysis through command-line usage. Running specific commands to detect malicious processes, network connections, and registry keys provided a deeper understanding of how memory artifacts correlate with potential threats.

	Each tool had its strengths; for instance, Redline was efficient for initial triage, while Volatility offered more granular analysis capabilities. **5. Personal Insights and Learning Outcomes** This experience highlighted the complexity and necessity of memory forensics in cybersecurity. Understanding how to navigate different tools and handle the nuances of memory data was challenging but rewarding. Key takeaways included: Tool Capabilities and Limitations: Each tool provides different insights based on its approach to memory forensics. It's essential to understand their capabilities to utilize them effectively. Challenges in Data Interpretation: Memory analysis often yields large volumes of data, requiring the use of filters and advanced techniques to identify significant indicators of compromise. Importance of Documentation and Reporting: Keeping detailed records of each step during the analysis process helps in tracking progress, troubleshooting issues, and ensuring accurate reporting. **6. Conclusion** Overall, setting up the forensic workstation and conducting memory analysis provided a comprehensive learning experience in digital forensics. Using multiple tools demonstrated the strengths and weaknesses of various approaches to memory analysis. The practical application of these tools to detect potential threats greatly improved my understanding of cybersecurity practices and prepared me for future forensic tasks.
Grading Alerts	• If you do NOT use this template or delete any part of it or use any other template, you will be degraded. • If you do NOT follow the fie naming convention, you will be degraded. • If you do NOT submit your file in PDF; you will be degraded. • If you do NOT show your account real name (when applicable); you will be degraded. • If you do NOT show your machine desktop background (with date & time) and IP, you will be degraded. • If you do NOT write (in your own words) your learning experience for the activity practices, you will be degraded.