TABLE DES MATIERES

INTRODUCTION

HYPER-V TOOLBOX - PROJECT PRESENTATION.

This project was carried out in order to answer a very specific problem and needs that I will take the time to state in this document.

Of course, I do not claim to have reinvented the wheel with this project, although to my knowledge, no such comprehensive tool existed before this one.

The ultimate interest of this project and of this tool as a whole is to save time and to improve the comfort of use considerably. Whether you are a technician, an administrator, a student, a teacher (or even just a computer enthusiast) wishing to use **Hyper-V** for your various projects, this tool will be an undeniable ally.

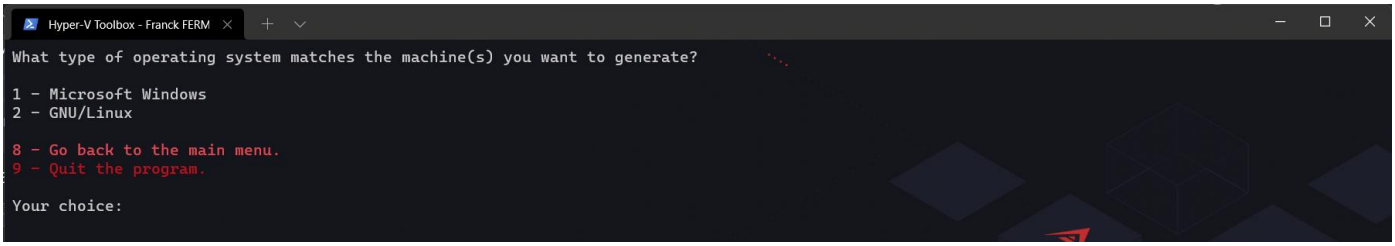
The principle of **Hyper-V Toolbox** is simple. It is an interactive script (with multiple choices adapting to the user's answers) strongly user-friendly and allowing (among other things) the management and advanced manipulation of **Hyper-V** from your terminal.

OVERVIEW OF FEATURES

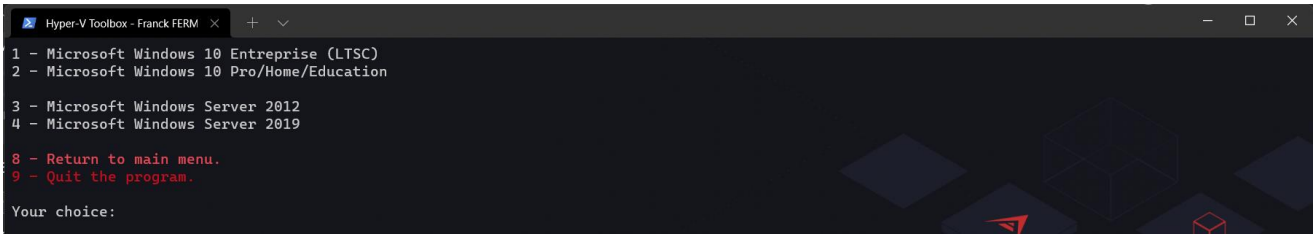
HERE IS AN OVERVIEW OF THE AVAILABLE FEATURES.

- **Creation of (basic) virtual machines**.

It is very simple, fast and effective. You just have to answer the questions asked by the script.

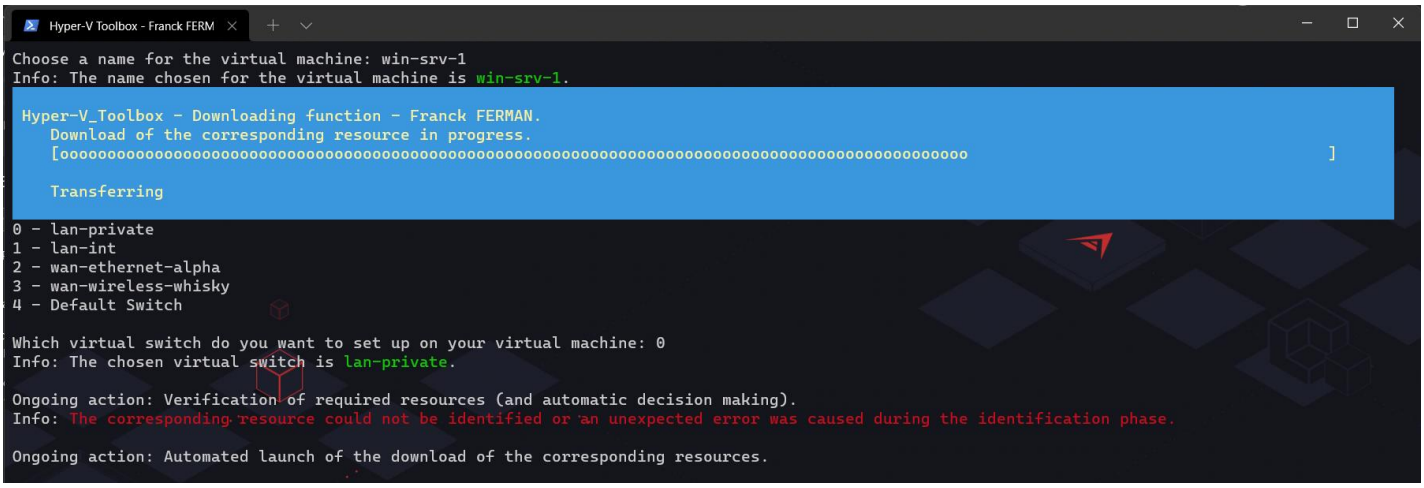


Many choices of systems are offered to you for the creation of your virtual machine. In order not to lose the user and to keep a coherent, orderly and harmonious structure, I have grouped the systems by separating them into two distinct categories: **Microsoft Windows** on one side and **GNU/Linux** on the other.



After selecting the desired system, two basic questions are asked by the script, the choice of a name for your machine and the possibility or not to choose a network card.

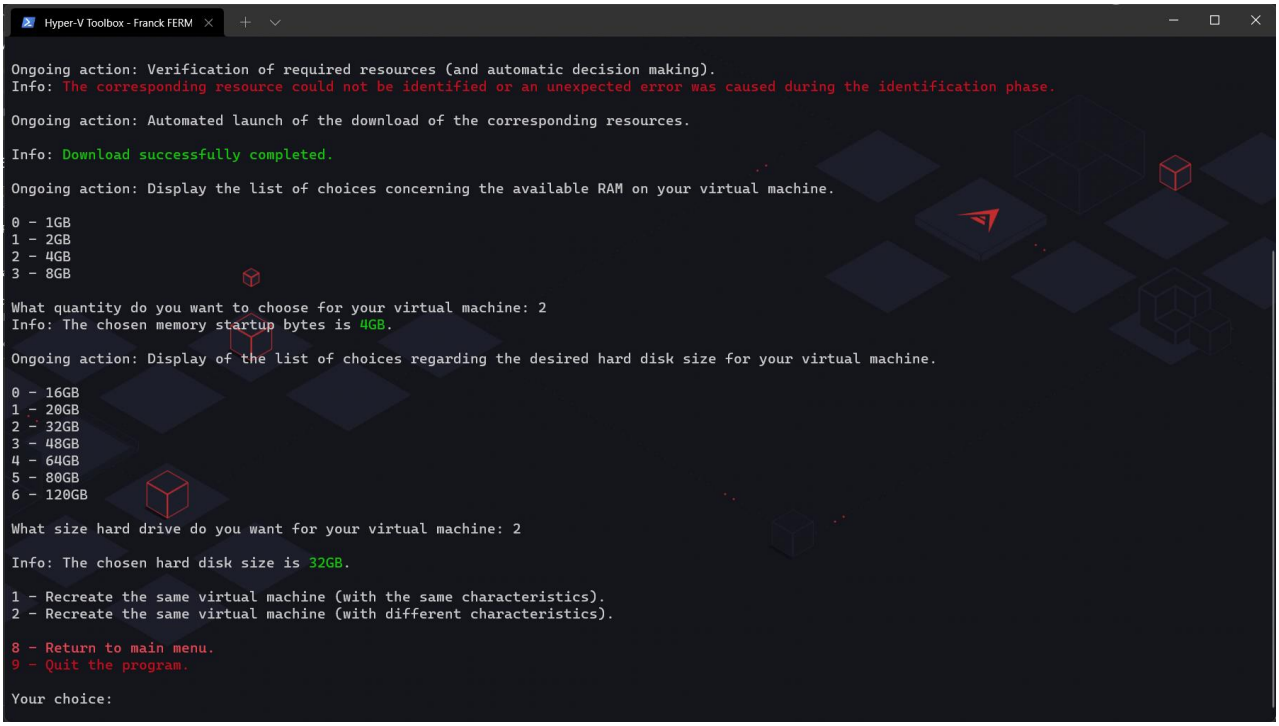
Then, in a first step, the script checks if the tree structure initially created by the script is present or not, if not, takes care of it accordingly. In a second step, the script checks if the desired resources are present or not. By resources, I mean everything that the machine will need to be correctly created and used. In this case, when creating simple virtual machines, it will most often just be an ISO image. If the resource(s) are not identified in the tree, the download will run by itself to complete the creation of the machine.



Some other basic questions (for the good continuity of the creation of the machine) are asked such as the choice of the quantity of RAM or the size of the hard disk desired for the machine.

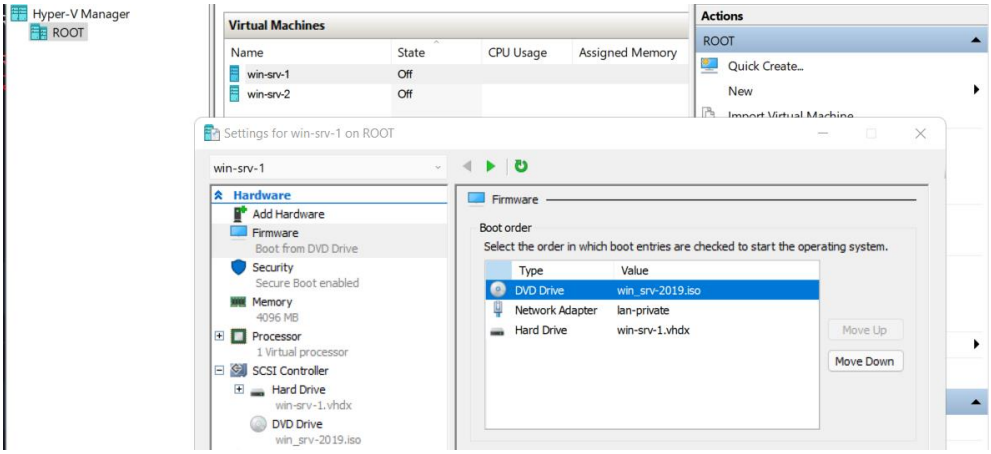
Once the machine has been created, which is generally extremely fast (a few seconds at best), several choices are available to you, two of which are particularly interesting. You can quickly create the same type of machine with different characteristics or recreate the same type of machine with the same characteristics.

To take a concrete example, in the context of practical work in class, or even for professional demonstrations or numerous tests for system administration or computer security, we sometimes need to create two machines of the same type such as two **Windows Servers**, two **Windows 10** (*clients*). This function allows the preparation of these elements in just a few seconds.



By default, the script optimizes the virtual machine parameters to allow the user to have a machine already ready to use. At least, to allow him to limit as much as possible the actions to perform. For example, the automatic restore points are disabled and the boot order is also modified so that the user does not have to modify anything.

On Linux, secure boot is disabled and dynamic RAM management is disabled because it is often poorly managed by **Hyper-V**.



The tool already handles the creation of many systems such as Microsoft Windows systems including **Windows 10 pro**, **enterprise**, **server 2012**, and **2019** but also many **GNU/Linux** systems such as **pfSense**, **Debian**, **Parrot Security**, **Rocky Linux** (with two editions available) and **Kali Linux** (also with two editions available).

Most of the ISOs have been retrieved from official repositories. This is not the case for **PfSense**, since the download from the official repository retrieves a compressed image. **PowerShell** handles well the Expand-Archive command which allows extraction of compressed files but does not handle decompression of compressed .gz files.

For this reason, I simply hosted on some of my rented servers the **PfSense** iso.

The **Microsoft Windows** systems ISOs (**10 pro**, **enterprise**, **server 2012** and **2019**) were also hosted on my servers.

- **Creation of preconfigured virtual machines**.

Just like the (basic) virtual machine creation function, the user will be asked some basic questions. With a few notable differences, at first, some choices will allow you to further speed up the creation process as only the name and network card of the desired machine will be asked.

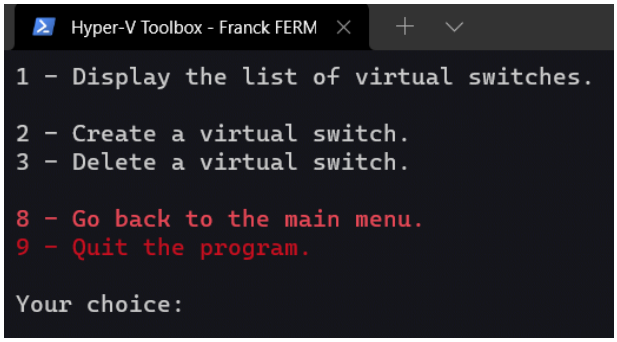
The RAM will be automatically adjusted according to the desired machine, and the hard disk? This is where it gets interesting.

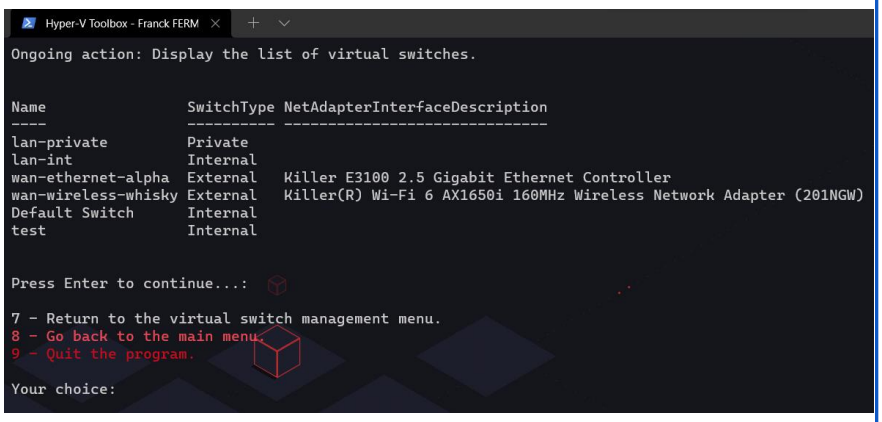
A differentiation disk will be created from the parent disk corresponding to the desired machine. What's the point? What does this famous parent hard disk contain? It all depends on the desired machine. But we will take the example of a **Windows 10** client.

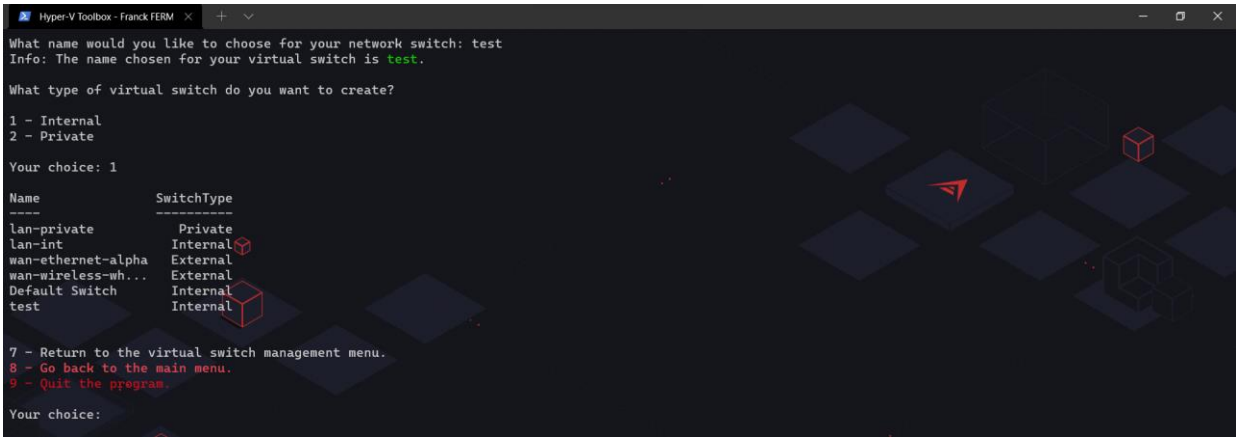
After having performed various manipulations on the machine (software installations that could be useful, system settings, etc.), I made a sysprep on the machine, sysprep to which we can link a response file (autounattend.xml). What I did, and what will allow, at the startup of the machine to pass all the "long" and "boring" steps of the **Windows** OOBE a.k.a. Out-of-box experience (acceptance of the terms of use, setting of the Windows privacy options, creation of a local account etc.) and at the end, the goal for the user will be to arrive on a Windows desktop, connected to a local account, already ready to use.

- **Virtual switch management**.

This feature is just a plus for management support, it simply allows (non-advanced) management of virtual switches. More precisely, it allows to display, create and delete virtual switches.

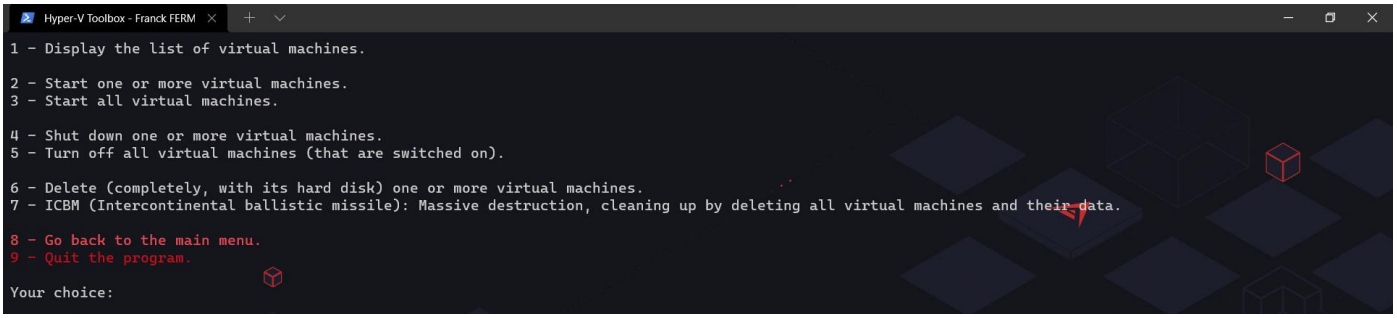






- **Virtual machines management**.

Just like the virtual switch management, these are basic functions (although this time much more useful and offering some additional features) allowing the manipulation of **Hyper-V** machines. By manipulation I mean all sorts of classic but useful tasks, such as: displaying the list of virtual machines and their status. Starting one, several, or all the machines. Shutting down one, several, or all machines. The deletion of one, several, or all the machines. In short, a relatively classical management of virtual machines.



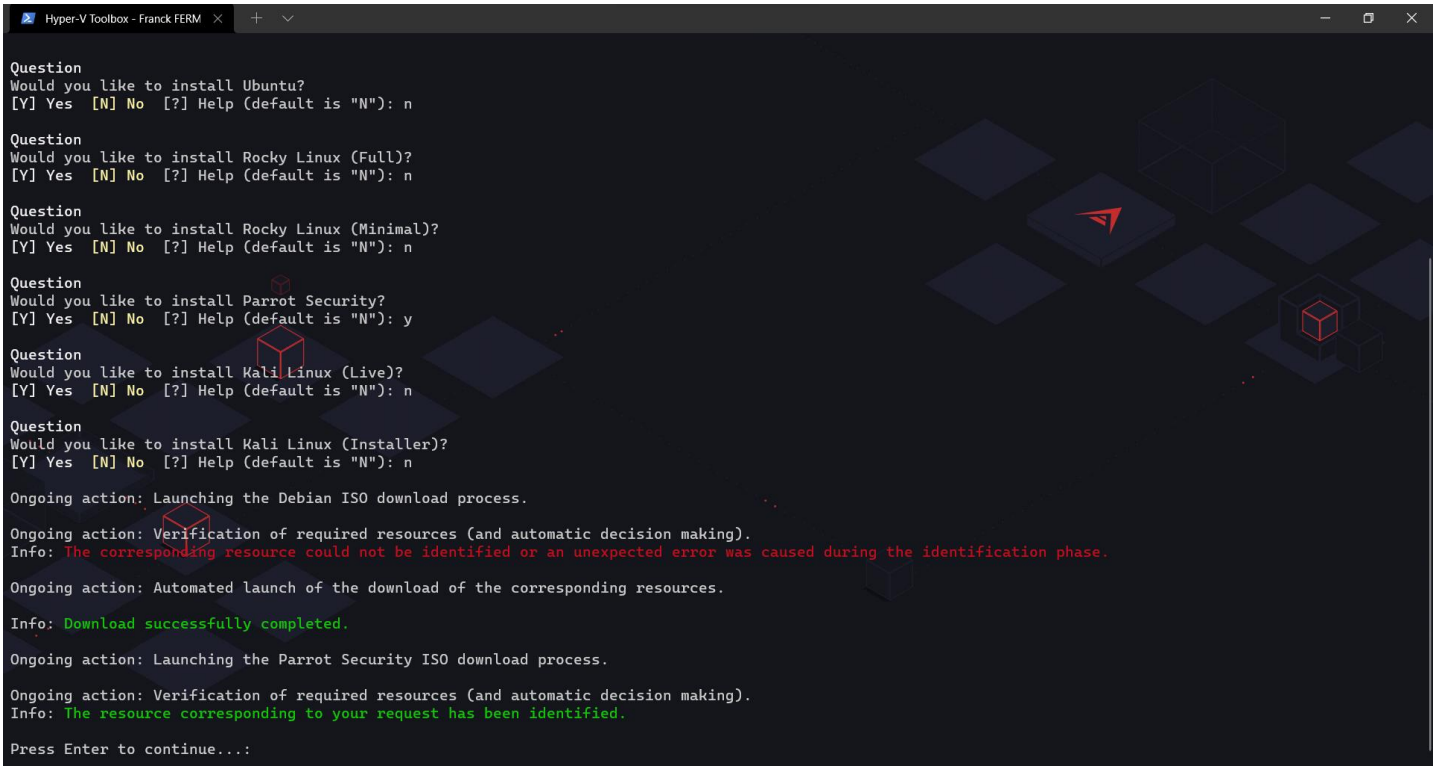
- **An advanced resource management system**.

As seen before, before each virtual machine creation, the script checks if the tree structure has been initially created by the script and acts accordingly then checks if the desired resources are present or not and takes care of downloading them if needed.

As a result, this first download might take some time although once it has been downloaded once, it will be present and available for the next times.

So I thought it would be interesting to set up a resource management system that downloads the desired needed resources in advance.

There are multiple possibilities: downloading resources for the creation of **Microsoft Windows** virtual machines only, downloading resources for the creation of **GNU/Linux** virtual machines only, the possibility to download absolutely all resources, and one last solution that I strongly recommend, I have made a customization function, by customization I mean a certain form of interactivity, which will allow the user to choose precisely and only the resources that he will really need.



AVAILABILITY OF THE TOOL

WHY IS IT ON GITHUB.

I fully support the philosophy of Free Software in general.

Free software is about freedom: everyone should be free to use software in any way that is socially useful. Software differs from material objects -- such as chairs, sandwiches, and gasoline -- in that it can be copied and modified much more easily. These possibilities make software as useful as it can be; we believe that software users should be able to use them.

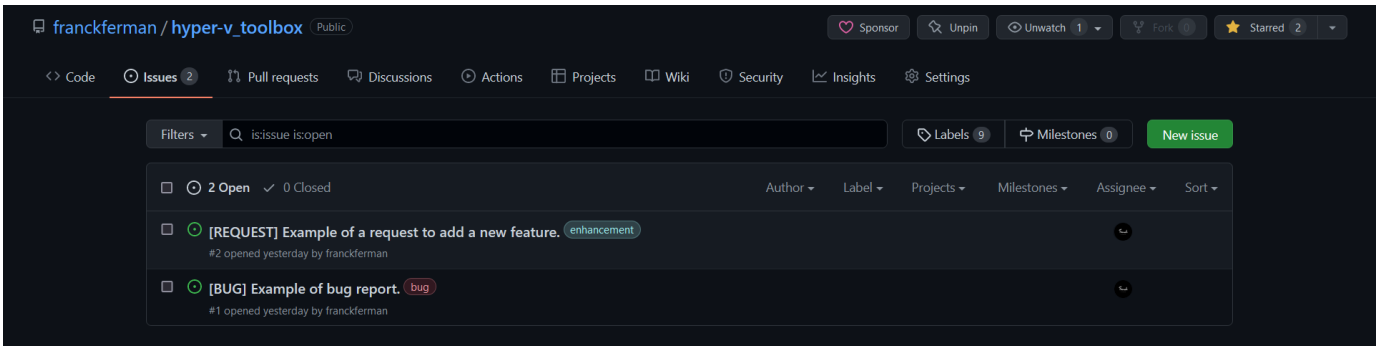
But not only. Let's put aside the political and philosophical side for the moment. The fact that my tool is free and that it is on **Github** is a real advantage, for many reasons.

I will argue, I will start by quoting the following adage: "*alone we go faster together we go further*". Anyone who has a **Github** account will be able to participate in the project and improve it by optimizing the code, fixing bugs, adding features, etc.

Teachers, students or even simple enthusiasts can join the project and get involved.

Github offers a great community project management system with change history, code review, collaborator management, etc.

Another very interesting feature is that it offers a system for managing bugs and proposals made by users. Anyone who encounters a problem can submit a new ticket describing the problem. The feature request system works the same way.



Finally, it allows to provide this tool in a rather simple way (*it is also a facility to update the code*).

The user just has to go to the project repository and click on download. Even simpler in a single PowerShell command : *Start-BitsTransfer https://raw.githubusercontent.com/franckferman/hyper-v_toolbox/main/hyper-v_toolbox.ps1*.

As for the ease of updating my code, I just have to make a commit and in a few seconds, the code is automatically updated.

- The link to my **Github** project: *https://github.com/franckferman/hyper-v_toolbox.git*.
- Links to the source code of my tool: *https://github.com/franckferman/hyper-v_toolbox/blob/main/hyper-v_toolbox.ps1* and *https://raw.githubusercontent.com/franckferman/hyper-v_toolbox/main/hyper-v_toolbox.ps1*.

THE WHY OF THE HOW

SKETCH OF A DESCRIPTION, WHAT WERE THE ISSUES AND NEEDS.

I had previously worked on a similar project but considerably less complete than this one. Although this one is not an inconceivable puzzle to realize either.

So I already had a little idea of what I could and couldn't implement, what I could draw inspiration from, etc.

So how did this project come about? Well, it was after many discussions throughout my life and my career with many students, professors, professionals, noting the considerable loss of time that students, professors or even professionals took, even if it was only during the phases of preparation of the infrastructure to be set up to start working on virtual machines.

Another problem that many companies, schools or institutions have already encountered is bandwidth problems.

My tool allows to solve this problem easily by editing the source code of my tool to change the download sources of the different ISO and to host them locally.