Installing Neurodesk on Windows environment

Guy Weizman | Department of Biomedical Engineering | Technion - Israel Institute of Technology

13 בספטמבר 2023

Contents

1 Introduction

What is Neurodesk? Neurodesk [1] is a Linux (type of operating system) based environment, with different engineering tools and programs built inside it, that facilitates the analysis of neuroimaging data (MRI). Many of those tools and programs require specific operating system, configuration and architecture that not all computers necessarily meet. Neurodesk provides a convenient platform that satisfies all the minimum requirements, and allows end users to simply use them without further actions.

Neurodesk operates as a container. In software engineering [2], containerization is operating system-level virtualization or application-level virtualization over multiple network resources, so that software applications can run in isolated user spaces called **containers**, that can be compared to virtual machines, in that they allow users to create a virtual, isolated computing environment with an operating system separate to that of the host machine.

The containers of Neurodesk are automatically built in an application called Docker, that requires seperate installation, like Neurodesk. The installation of both of them on Windows based environments poses several challenges, that this document comes to help overcoming.

2 Installation procedure on physical stations

2.1 Docker installation

2.1.1 Prerequisites

The installation of Docker requires several components [4]:

1. **Operating System.**
   (a) Windows 11 64-bit: Home or Pro version 21H2 or higher, or Enterprise or Education version 21H2 or higher.
   (b) Windows 10 64-bit: Home or Pro 21H2 (build 19044) or higher, or Enterprise or Education 21H2 (build 19044) or higher.
2. **WSL.** WSL version 1.1.3.0 or later. WSL stands for **W**indows **S**ubsystem for **L**inux, that lets [3] developers run a GNU/Linux environment (including most command-line tools, utilities, and applications) directly on Windows, unmodified, without the overhead of a traditional virtual machine or dualboot setup. The WSL 2 feature on Windows should be turned on. In order to successfully run the WSL feature on Windows, several prerequisites should be satisfied:
   (a) 64-bit processor with Second Level Address Translation (SLAT).
   (b) 4GB system RAM.
   (c) BIOS-level hardware virtualization support must be turned on in the BIOS settings.

How can we check whether these requirements are satisfied or not? Let's check the components one by one:

Regarding the operating system's edition and version, you should press the winkey button on the keyboard, write "system" in the start menu search bar, and choose the first result. Under the "Windows specifications" field, the edition and version can be found (figure 2.1-a). If the version is older than required, you can simply update it: in the start menu write "check for updates" and click the first result. Then click "check for updates" (figure 2.1-b). If new version of windows is available, it will be installed. If not, it's possible to download it **online**.

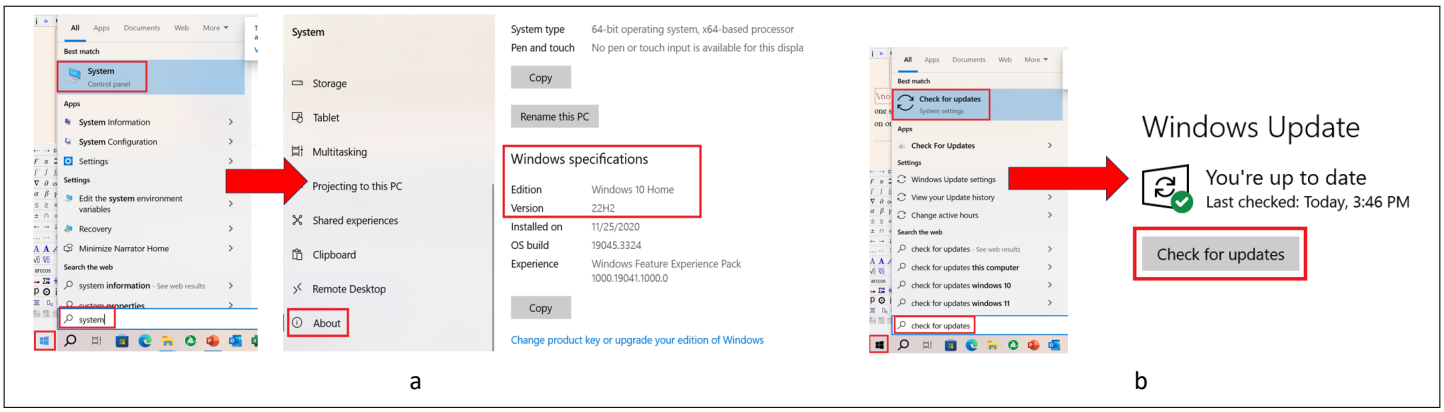

Figure 2.1: (a) Checking Windows' version and edition; (b) Windows update service

Now, let's check if the computer satisfies the hardware requirements in order to successfully run the WSL feature:

1. **Processor.** Go to the "System" window (figure 2.1-a), and look at the Device specifications field, in which the processor type is described (figure 2.2-a).
2. **RAM.** Go to the "System" window (figure 2.1-a), and look at the Device specifications, in order to see the amount of installed RAM (figure 2.2-a).

3. **BIOS & SLAT.** At the start menu search bar, write “cmd”, and run it as an administrator. Then write “systeminfo” and press enter. Find the title of “Hyper-V requirements”, and look at the lines “Virtualization Enabled In Firmware” and “Second Level Address Translation” (SLAT). If the word “yes” appears next to them, then the virtualization is enabled in the BIOS, and the processor supports this feature (figure 2.2-b).

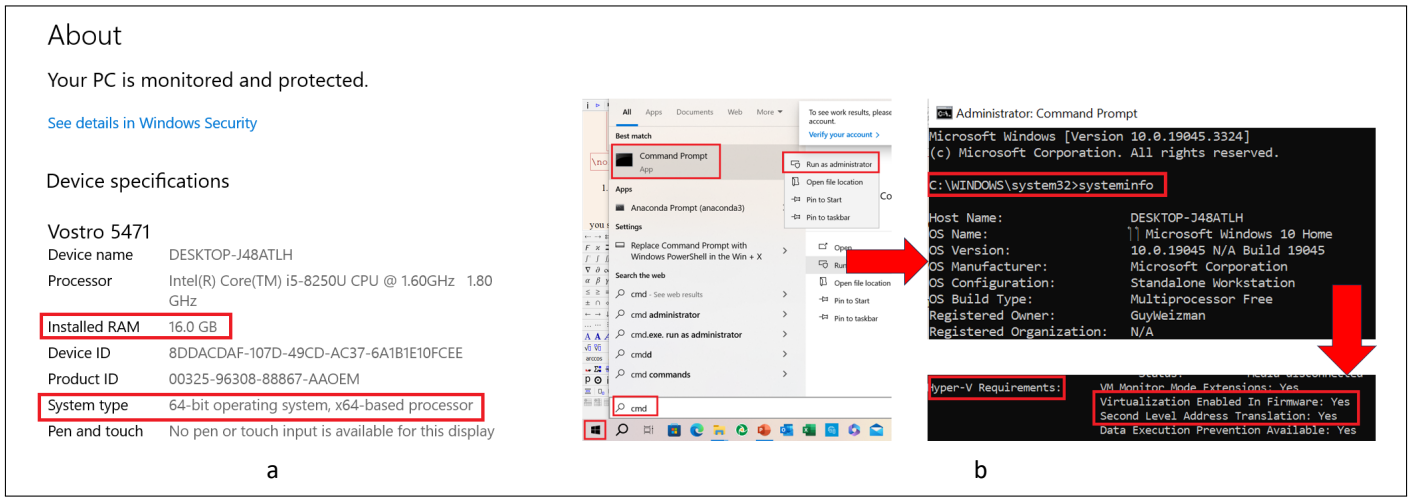

Figure 2.2: (a) Hardware information; (b) Hyper-v Requirements (BIOS and SLAT)

After you make sure your computer can run the WSL feature, it’s time to turn it on. In order to do so, press the winkey button, write appwiz.cpl, and then choose the first result. After the “Programs and Features” window opens, click the “Turn Windows features on or off” option (admin privileges are required), and in the following list (figure 2.3), check the “Windows Subsystem for Linux” option, and then reboot the computer.

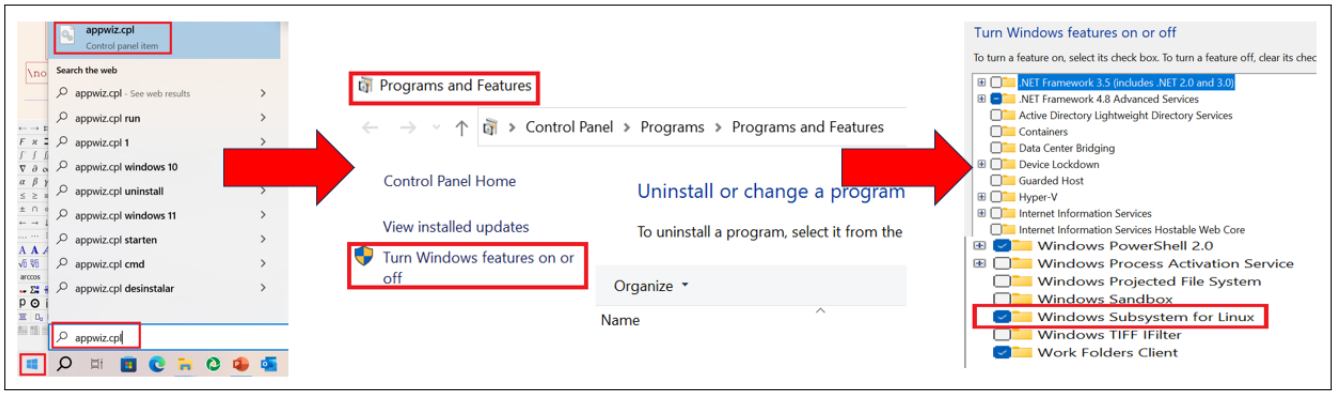

Figure 2.3: Turning “Windows Subsystem for Linux” on

2.1.2 Installation

Now you can install Docker on your computer. Go to this **website**, and click the “Docker Desktop for Windows” button. When the download is complete, go to the “downloads” folder, and double click the “Docker Desktop Installer” file (make sure you run it as administrator). When the installation begins, make sure you check both options: “Use WSL 2 instead of Hyper-V” and “Add shortcut to desktop (figure 2.4). Then press OK, and wait until the installation is finished. When it’s done you will have to reboot your computer.

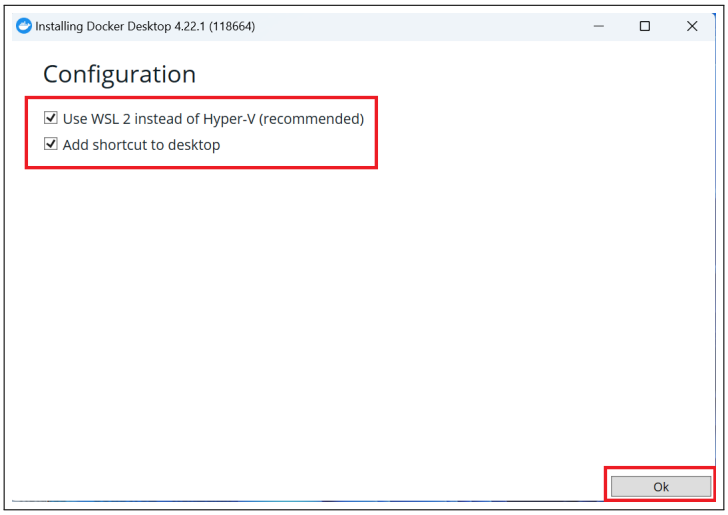

Figure 2.4: Installing Docker Desktop

After the reboot, log in and try running Docker. You might encounter the following error: "Docker Desktop - WSL kernel version too low. Docker Desktop requires a newer WSL kernel version" (figure 2.5).

Figure 2.5: "WSL kernel version too low" error

In this case, there are two solutions:

1. Run cmd as administrator (as shown in figure 2.2-b), an write the command: **wsl –update** (notice the double hyphen before the word "update"). If everything works fine, a window of "User account control" will open, and ask you allow the app to make changes (figure 2.6). Choose yes, and wait until the installation is finished.

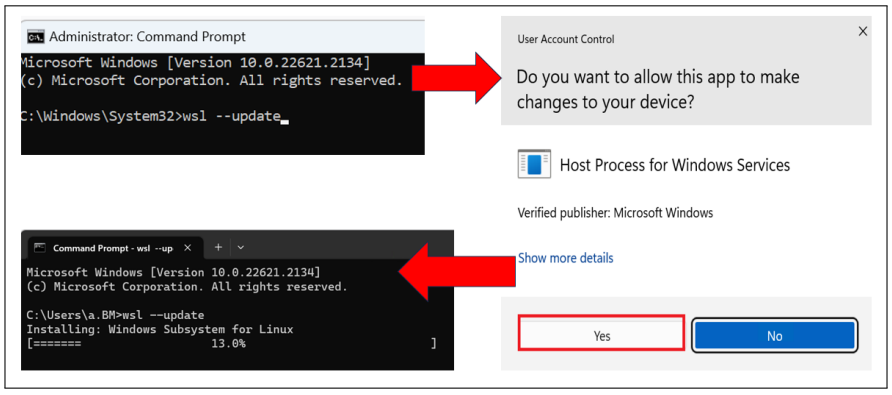

Figure 2.6: Updating WSL kernel version

If you encounter an error after running the command (figure 2.7), move to the second solution.

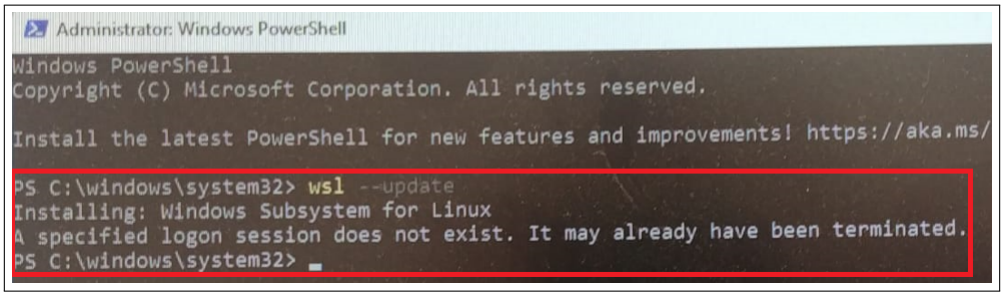

Figure 2.7: "A specified logon session does not exist" error

2. Go to this **website**, and download the update "WSL2 Linux kernel update package for x64 machines", and run it with administrator privileges. It should solve the previous errors (figure 2.5 and 2.7).

Now Docker is properly installed on your station. The next step is to add all the users that will work with Docker to the right security group. In the start menu search bar, write "compmgmt.msc" and run it as administrator: right click on the result and choose "run as administrator" (figure 2.8).

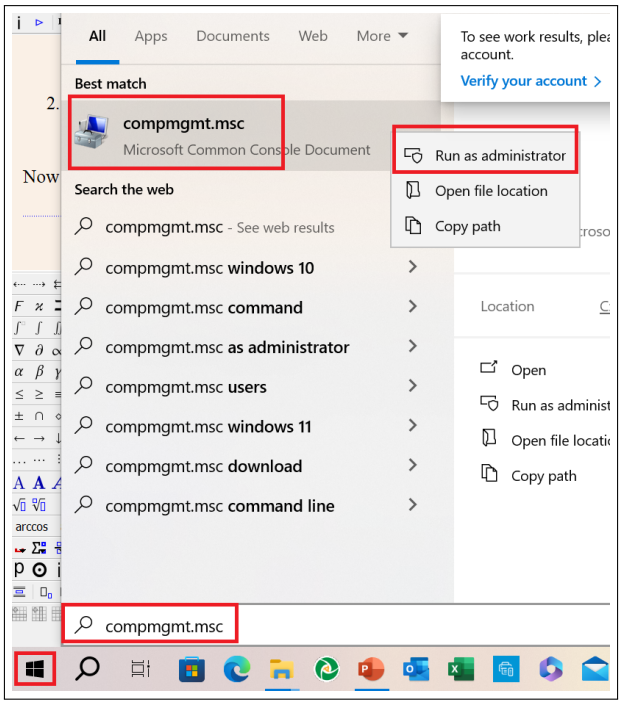

Figure 2.8: Opening computer management

In the computer management window, choose "system tools", then "local users and groups" then "groups". Look for the "docker-users" group, and double click it. In the "properties" window, press the "add" button, and insert the names of the users that will be using Docker. Pay attention: if the users are local (i.e. exist only on your local machine), make sure you write the name of your computer before the user name. For exmaple, if the station you are working on is "T-BM-135-135", and the user is "new", you should add: **t-bm-135-135\new**. You can add several users, as many as you like, but make sure to use the validation button "check names", on the right. After you have finished adding all the relevant users, press "OK", and then "apply" to apply the changes you have made (figure 2.9).

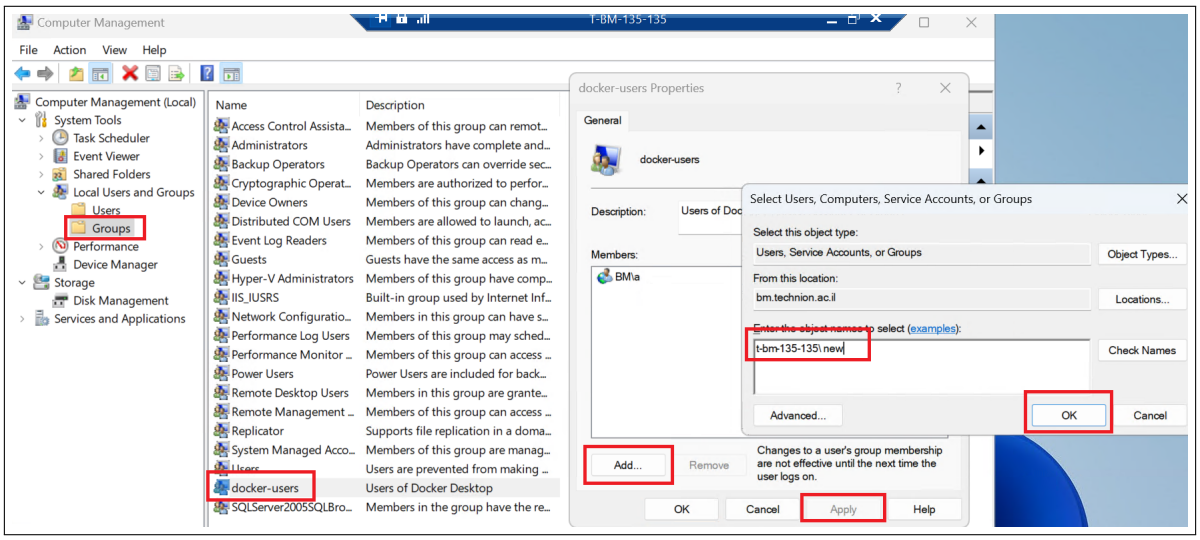

Figure 2.9: Adding users to the “docker-users” security group

If one of the users you have added is “regular” (i.e. without admin privileges), you might encounter the same error whown in figure 2.5, when you try to run Docker. In this case, there are two solutions:

1. After logging in using the “regular” user’s credentials, run Docker with administrator privileges. How? search Docker in the start menu, right click its logo, and then choose “run as administrator”. You may be required to insert the administrator’s credentials (user name and password). If this still doesn’t work, or you don’t want to run Docker with administrator privileges, move to the second solution.

2. Log in using an administrator account. Open the file explorer (recycle bin is also fine), copy this path “**C:\Program Files\Docker\Docker**” and paste it in the url bar at the top of the window. Find the “Docker Desktop” exe file, right click it, and choose “properties” (figure 2.10).

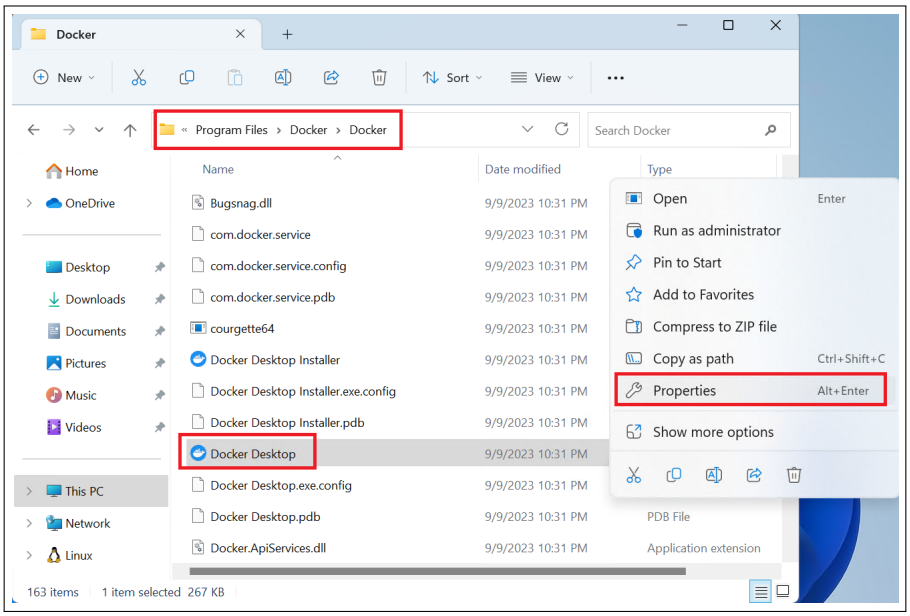

Figure 2.10: Finding the “Docker Dekstop” exe file

In the “Docker Desktop properties” go to the “security” window, and press edit. In the “persmissions” window, press add. In order to make sure that anyone who uses your computer can run Docker without problems, add “everyone” and press the validation button (figure 2.11).

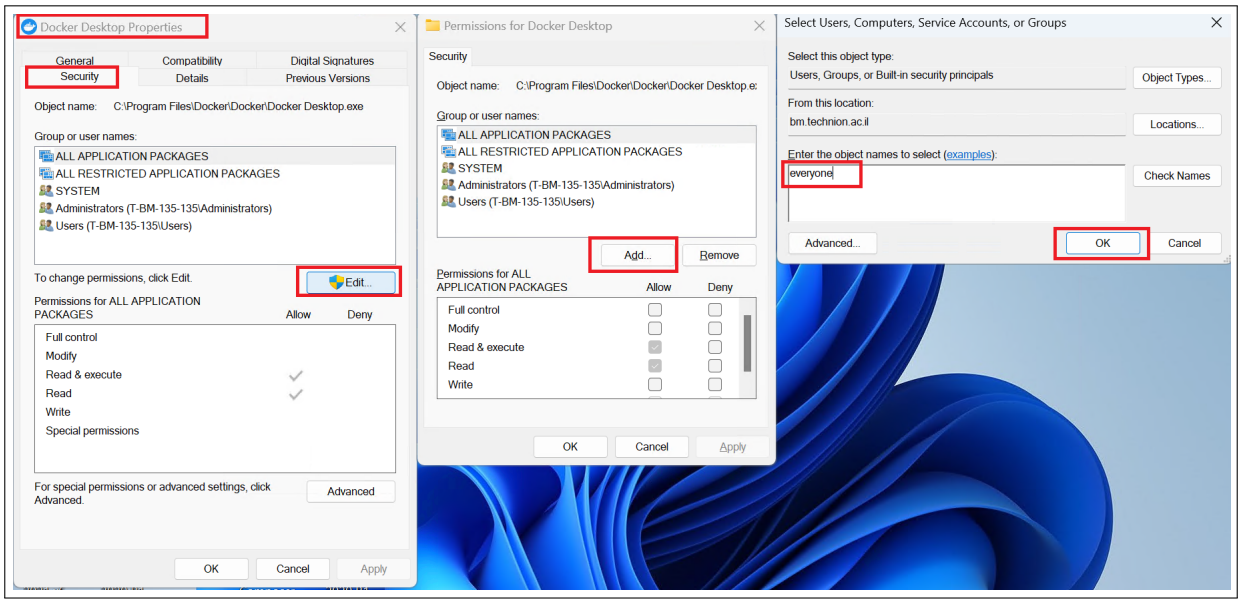

Figure 2.11: Changing Docker Dekstop permissions

After you have added everyone to the list, change the permissions to “Full Control”: you want to make sure everyone can run the file without admin privileges (figure 2.12). Then click “apply” to apply the changes you have made.

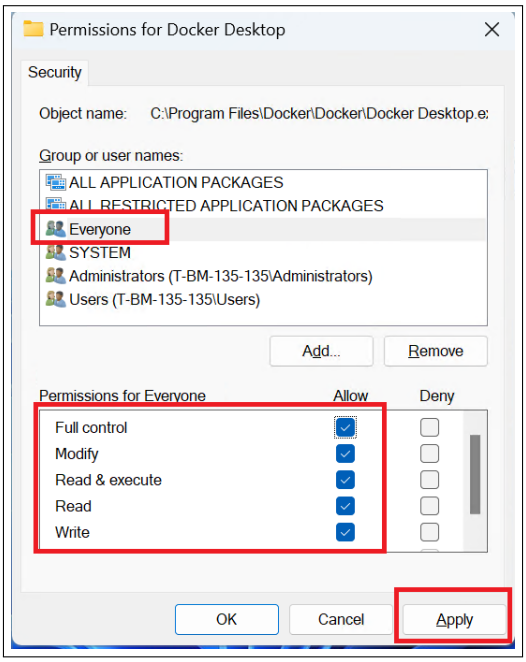

Figure 2.12: Changing Docker Dekstop permissions

2.2 Neurodesk installation

Once you have installed Docker and managed to overcome all the problems, you can move on and install Neurodesk. Go to this **website**, and download "Windows Installer". Run it with administrator privileges, and wait until the installation process is complete. If Windows is "reluctant" to install this program, just click "more info" and then click "run anyway" (figure 2.13).

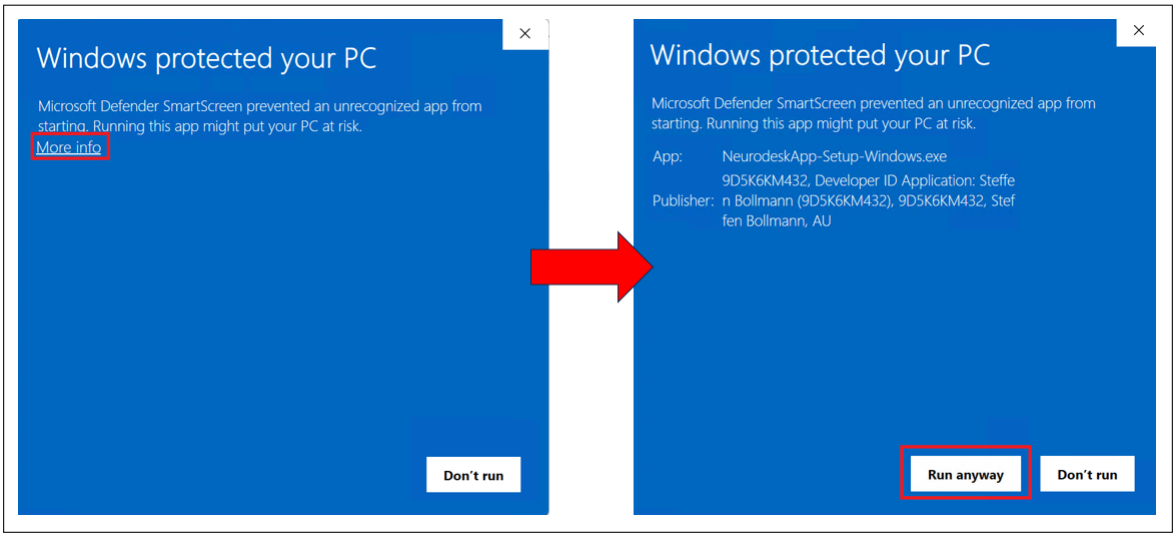

Figure 2.13: Installing Neurodesk

After the installation is finished, you may want to run Neurodesk to check that everything works fine. Make sure to first run Docker, and only then run Neurodesk. When the Neurodesk window is opened, choose "Open local Neurodesk". If the update message pops up, try downloading and installing the newest version of Neurodesk. If the newest version is already installed, just ignore the message and click the X button (figure 2.14).

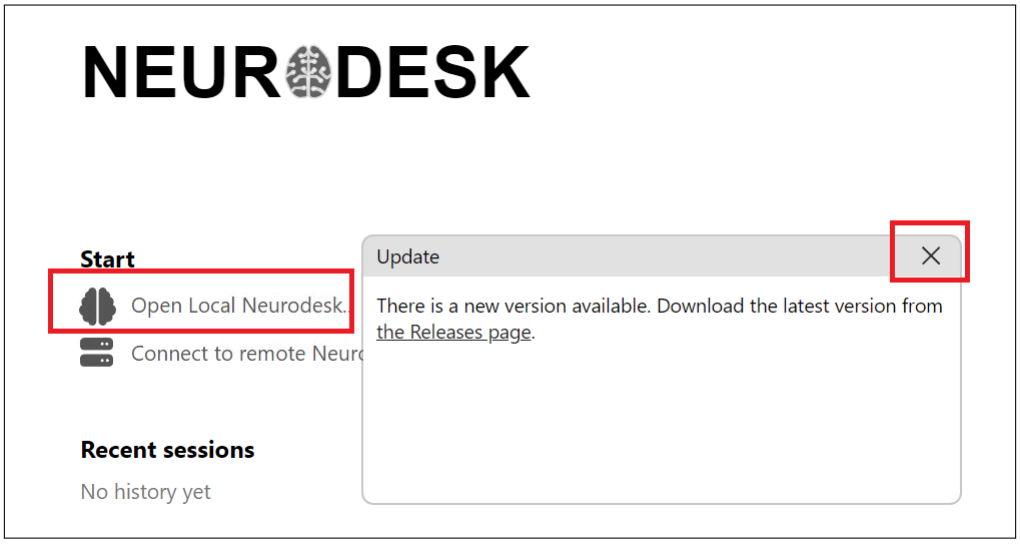

Figure 2.14: Running Neurodesk

If you run Neurodesk for the first time, it may take some time before the program opens. If the "Windows Security Alert" message pops up - allow the access (figure 2.15).

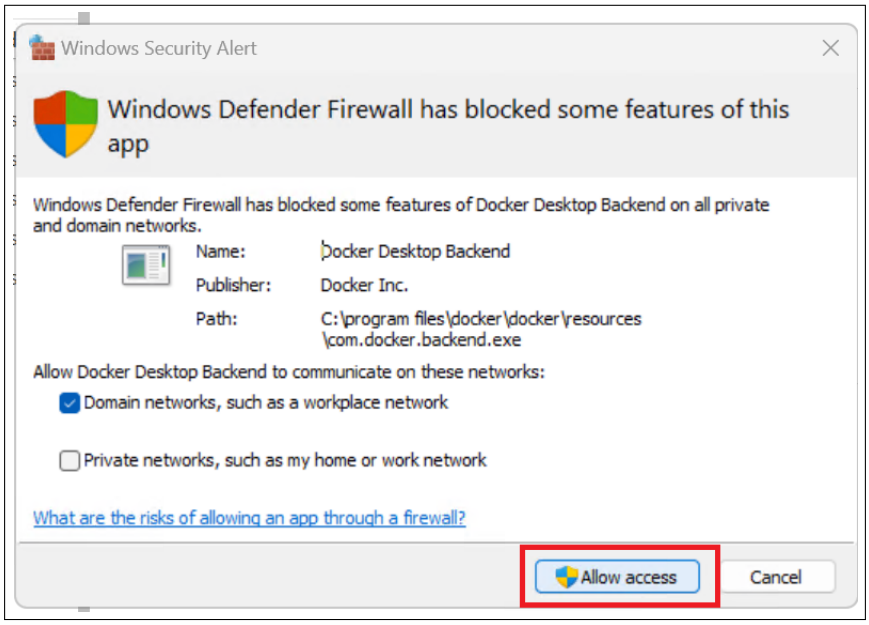

Figure 2.15: Windows Security Alert message

Once Docker and Neurodesk are working, this is what they are supposed to show:

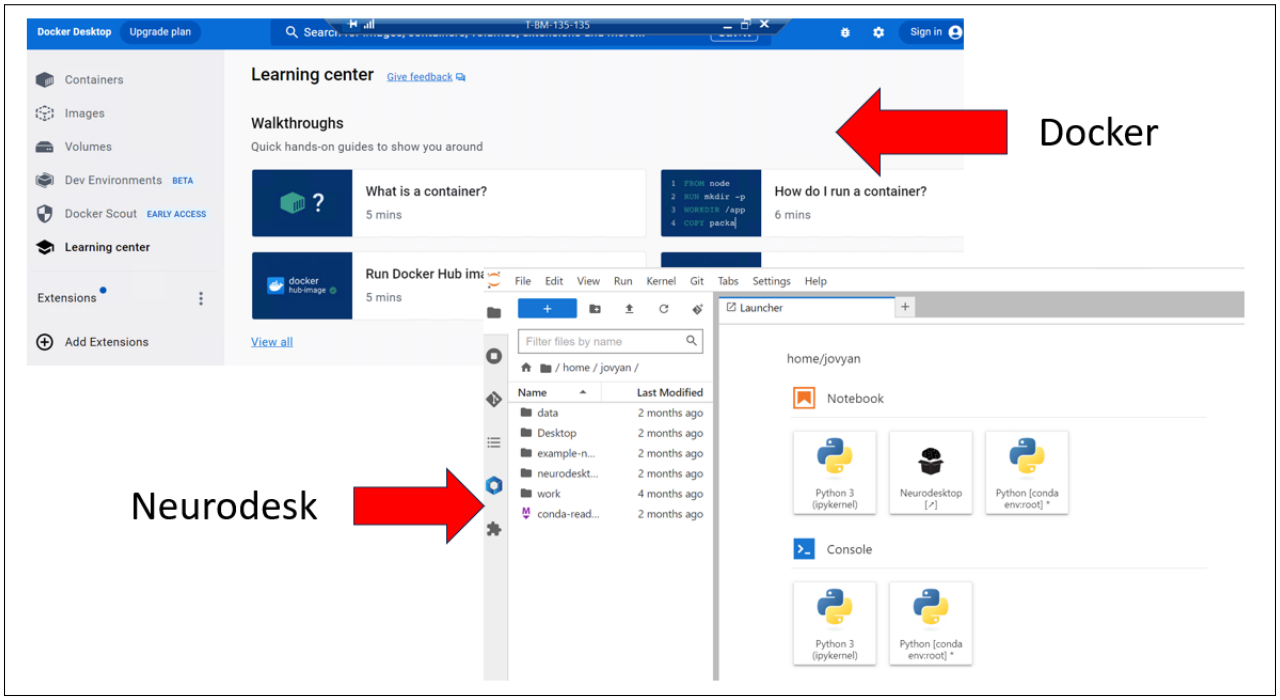

Figure 2.16: Docker and Neurodesk

3 Installation procedure on virtual machines

3.1 Motivation

One might want to install Neurodesk on virtual machines, rather than physical machines for two main reasons:

- Creating a convenient environment to run Neurodesk by many end users.
- Creating an environment to run Neurodesk in case the physical station does not satisfy the minimum requirements described previously.

Either way, virtual machine is a convenient solution that can fulfill both needs mentioned above. Virtual machine simulates a physical computer that can run Windows based operating systems (or Linux, for that matter), only it's created on a server, rather than a stand alone computer. Past experience teaches that not all virtual machines are suitable to install Docker and Neurodesk, but only the ones that support the nested virtualization feature. What does that mean? Since Neurodesk itself runs on a Linux based virtual machine, and we install it on a Windows based virtual machine, we need to make sure the "external" virtual machine has the right characteristics to support this structure. In fact, we install a virtual machine inside a virtual machine, to form some kind of "nest".

After many trials and failures, we have come to the conclusion that the best platform to create the virtual machines is Azure Cloud. We have also created several virtual machines, and successfully managed to install Docker and Neurodesk on all of them. The main challenge to be addressed is creating the virtual machines. Installing Docker and Neurodesk on them is as described in the previous chapter.

3.2 Creating the virtual machine

Before creating anything, you should first need to subscribe to Azure Cloud and settle the billing with the customer service. Make sure you do it as soon as possible, so you could access the console and create the right machine for you. Go to Portal Azure: **Link**, and make sure you are logged in with the right credentials (figure 3.1).

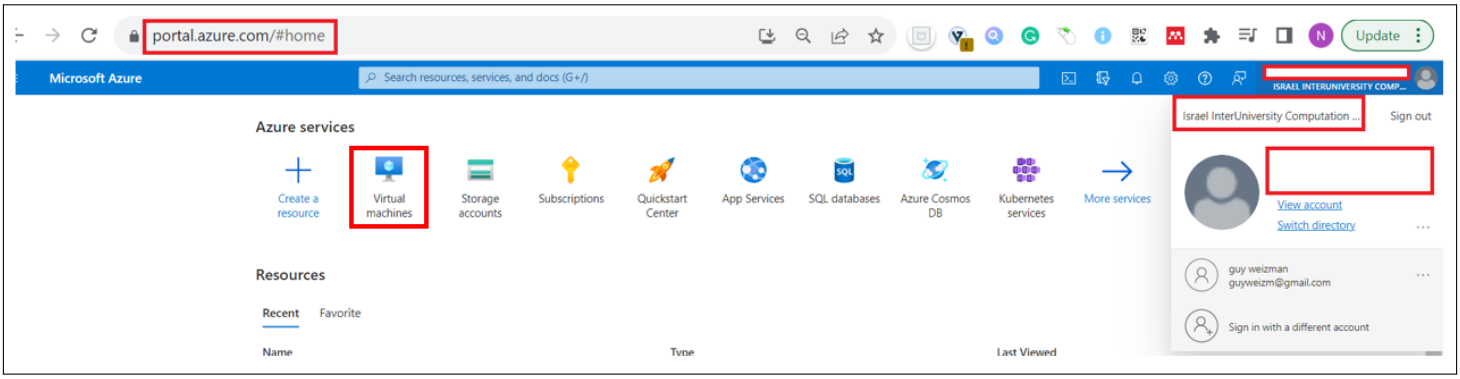

Figure 3.1: Accessing Azure portal with the right credentials

Figure 3.1 shows the main screen of Azure portal. In order to create a virtual machine, click "Virtual Machines" as shown in the picture. Now, in the next screen, click the "create" button, and choose the first option: Azure virtual machine (figure 3.2). Pay attention: before creating many machines (as the number of people that will be working with Neurodesk), create ONE sample and make sure everything works fine on it. Only then you should move on and create more machines with the same configuration.

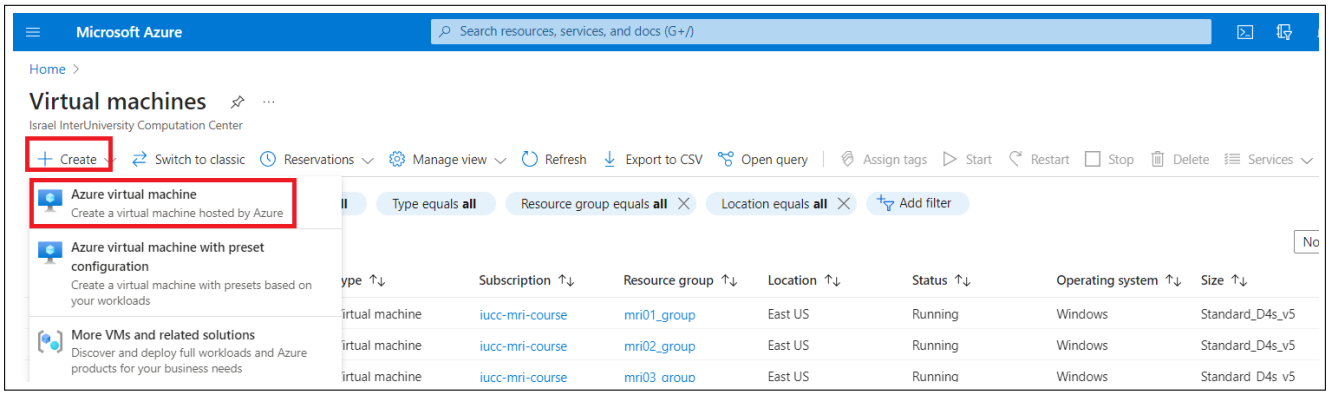

Figure 3.2: Creating a virtual machine

You are now on the “create a virtual machine” screen, inside the “Basics” tab. There is no need the change the subscription and the resource group definitions, as well as the availability options and availability zone. Regarding the region - consult the customer service and ask them what would be the best option. Please note that different regions can change the cost of the virtual machines’ operation (figure 3.3).

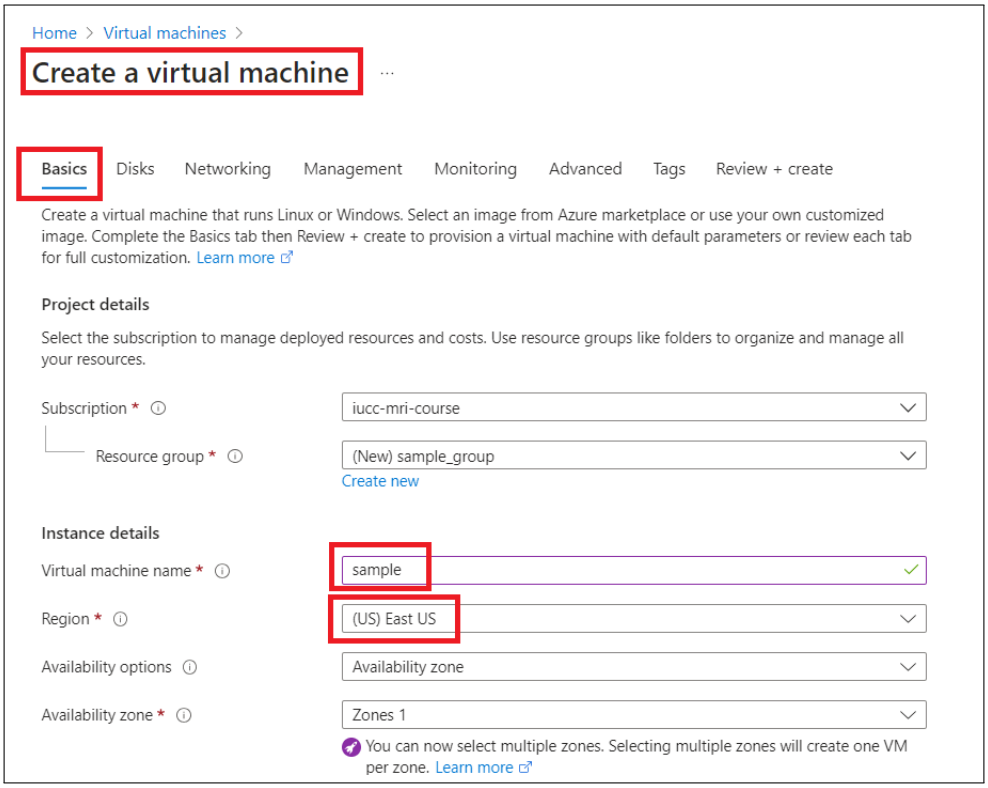

Figure 3.3: The “Basics” tab

You do need to choose a name for the virtual machine: try using an informative name, related to your project. Under security type choose “Standard”, and under image choose “Windows 11 Pro, version 22H2 - x64 Gen2”. It’s possible to choose also “Windows 10 Pro, version 22H2 - x64 Gen2” (figure 3.4). However past experience shows that working with Windows Server based operating systems may cause many problems, and is **not** suitable for working with Neurodesk.

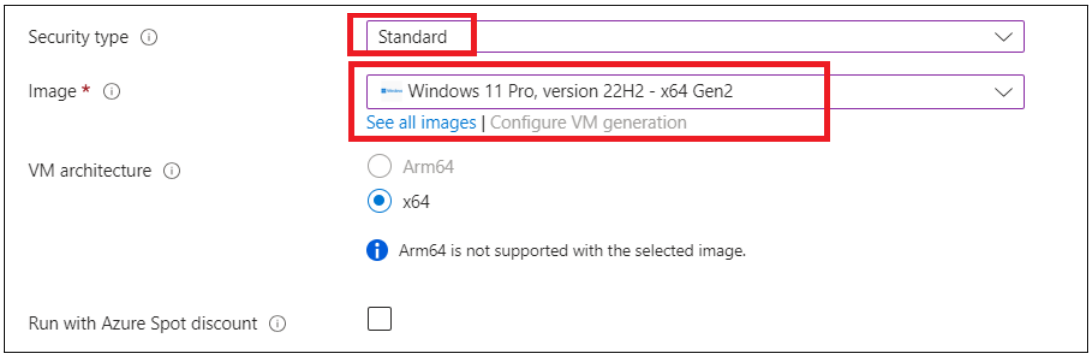

Figure 3.4: Choosing the security type and the image

Under size, choose “Standard_D4s_v5 - 4cpus, 16 GiB memory” (figure 3.5).

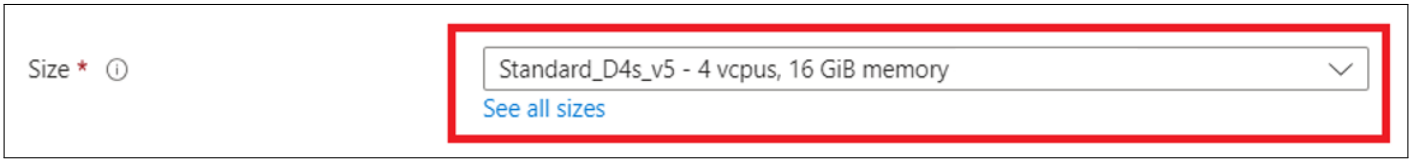

Figure 3.5: dize definition

Click the “see all sizes” button located under the drop down menu, and in the search bar write d4s, then press enter. You will get several options, choose the D-series v5 and click the “select” button at the bottom of the screen (figure 3.6).

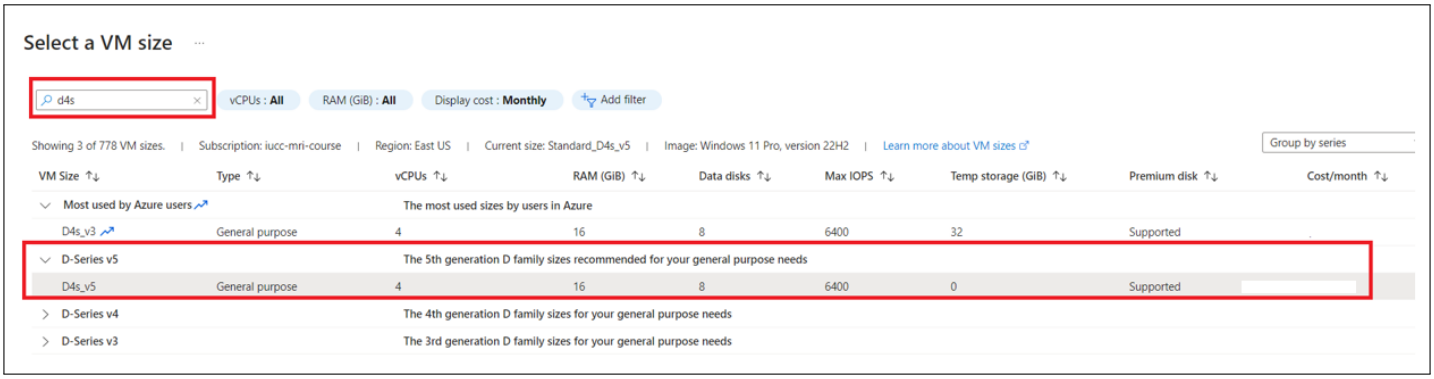

Figure 3.6: Selecting a VM size

If you think you need a different type of machine, talk to Azure’s customer service, and make sure the machine they recommend can support the installation of Docker and Neurodesk. Next you need to create an administrator account: write a name, and a password as required. Under “inbound ports” check “HTTP”, “HTTPS”, and “RDP”. Under licensing, check v on the statement, and then click “Review + create” (figure 3.7). There is no need to change anything in other tabs (Disks, Networking, Management, etc.), unless the customer service told you specifically what to do.

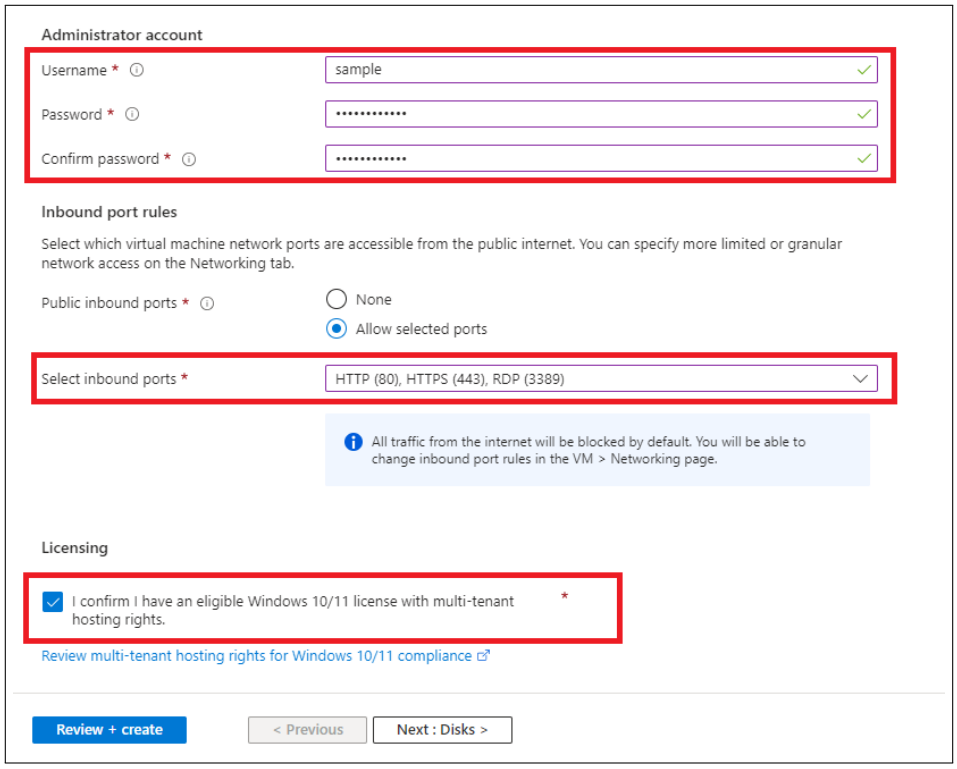

Figure 3.7: Creating the administrator account and selecting the inbound ports

You need to wait a few minutes until the validation is completed, and then click “create” (figure 3.8). Then wait a few more minutes until the deployment is complete.

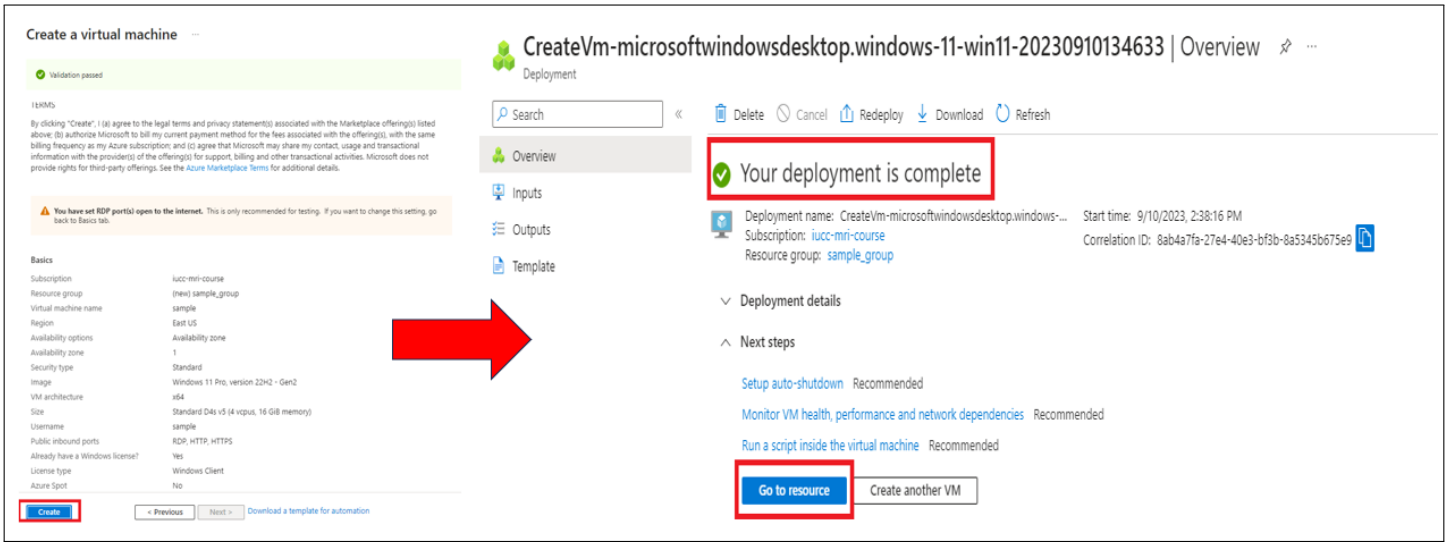

Figure 3.8: Deployment is complete

Once the deployment is complete, click “go to resource”. In the follwing screen you can control the machine you have created and see detailed information about it.

3.3 Remote Desktop Protocol

After you have created the virtual machine, you need to connect to it in order to install Docker and Neurodesk. Navigate to the virtual machine **interface**, and find the relevant machine. If you check the square on the left, you notice that several options at the top menu suddenly "unlock": (1) **Start.** You can turn your machine on; (2) **Restart.** You can reboot your machine; (3) **Stop.** You can turn your machine off; (4) **Delete.** You can delete your machine (figure 3.9).

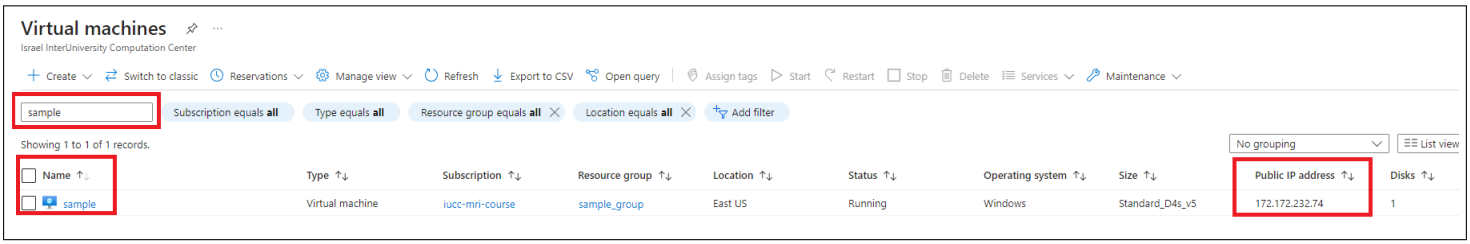

Figure 3.9: Operation of the virtual machine

If you click the name of the virtual machine (sample, in this example) a new screen will open, allowing you to connect to the machine remotely. Make sure you work on the "Overview" tab, and click "connect" (figure 3.10).

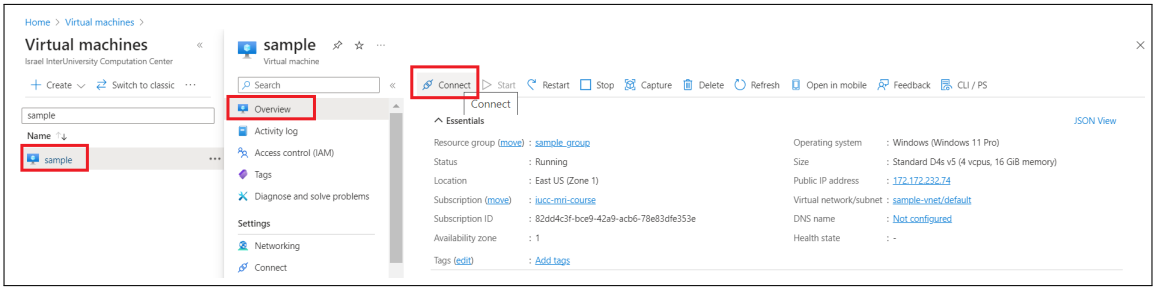

Figure 3.10: Connecting to the virtual machine

In the following screen, make sure you connect to the virtual machine using a public IP address (not a private one). Then go to the bottom of the screen, choose the "Native RDP" option, and click "select".

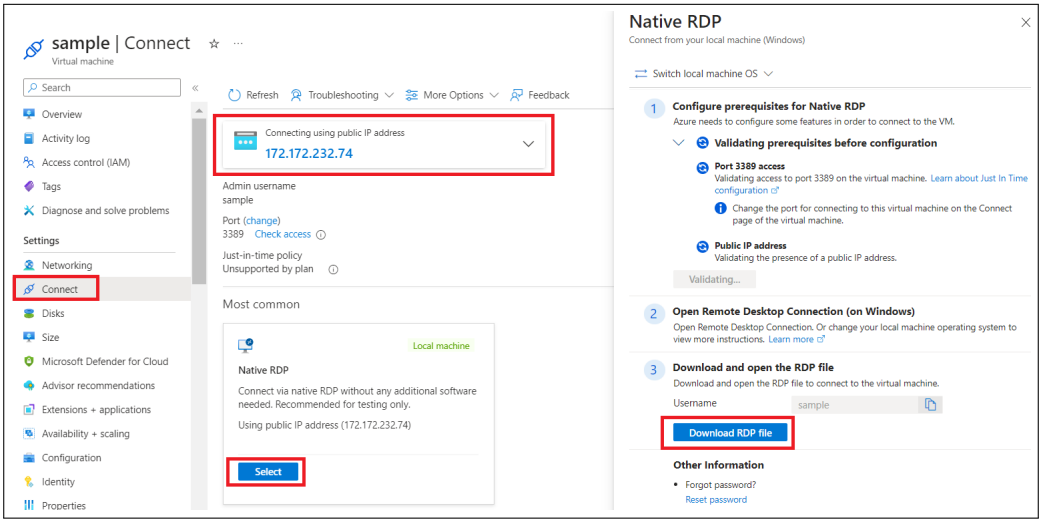

Figure 3.11: RDP connection to the virtual machine

A new window will open on the right - go straight to step 3, and click “download RDP file” (figure 3.11). After the download is complete, open the “downloads” folder, and double click the RDP file. A “Remote Dekstop Connection” window will open, asking you to click the “connect” button. After a few seconds, you will be required to insert a user name and a password - use those you chose eariler, when you created the administrator account (figure 3.7), and then click “OK” (figure 3.12). Make sure you try connecting using a private wifi, a hotspot, or a **not** secured wifi network. Secured wifi networks (like those in universities) might block the connection to the virtual machine.

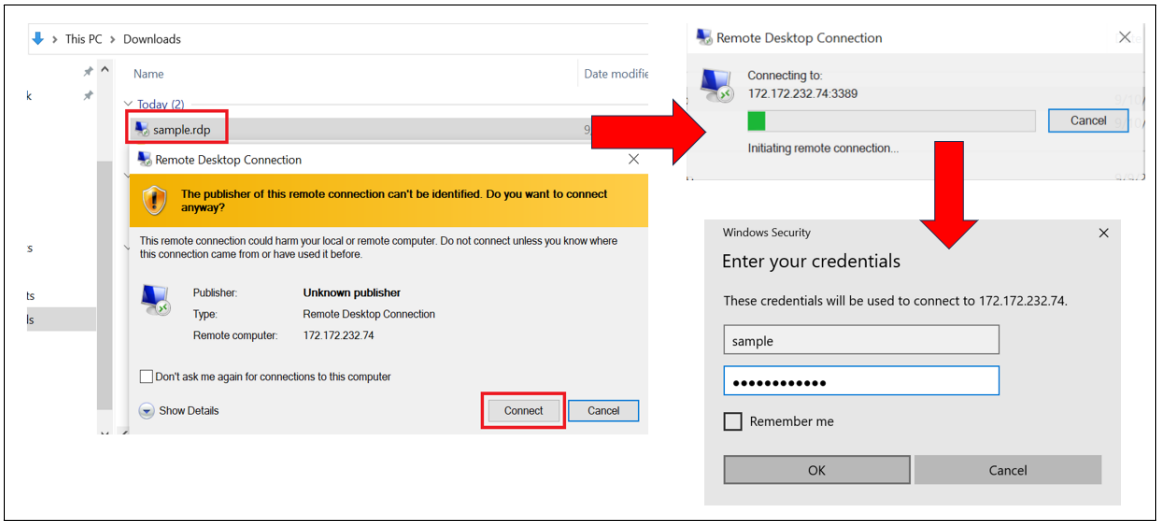

Figure 3.12: Remote connection process

A new message will appear, asking your attention regarding the remote control session. You can check the “Don’t ask me again” option at the bottom of it, and then click “yes” (figure 3.13).

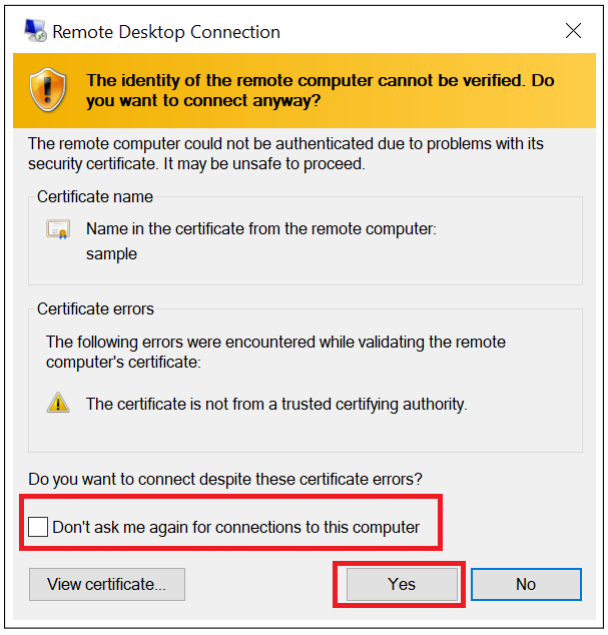

Figure 3.13: Remote Desktop Connection message

At the bottom of your screen, you will see a small logo that symbolizes the remote desktop connection. If you click it, it will occupy the entire screen, displaying the desktop of your virtual machine (figure 3.14). The blue bar at the top of your screen, allows you to control the

remote session: clicking the hyphen (-) sign will minimize the remote control window, and clicking the X sign will close it completely (but will NOT shut down the virtual machine, that will continue to run). You can use the regular power button to shut down and reboot your machine. Pay attention that once you shut it down - you can turn it on **only** through the control panel (start button), as shown in figure 3.9.

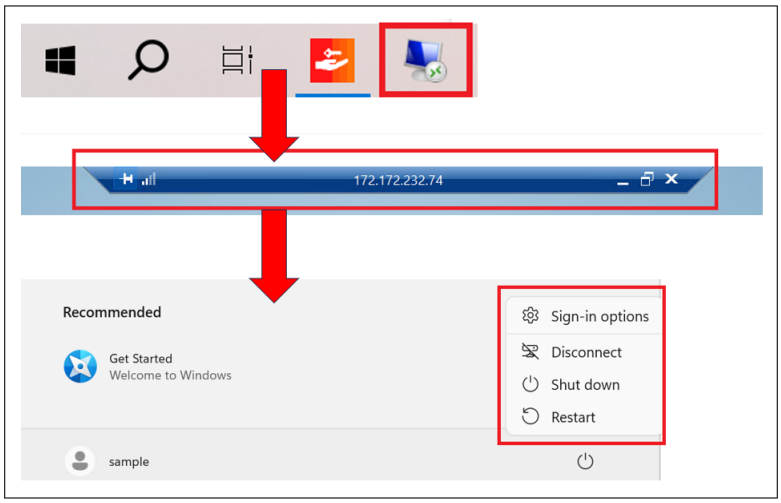

Figure 3.14: Control panel of the remote session

One of the first things you may want to do, is creating a backup user, just in case you forget the password of the original administrator user. Notice that if you do forget your password, and insert a wrong one too many times, your account might be locked out. But don't worry because it will unlock itself within 10 minutes. But, if you don't want to wait, you can always use the other administrator account to manually unlock it. In order to create a new local account (i.e, that exists only on your virtual machine), open the computer management window, as shown in figures 2.8 and 2.9, but this time, under "local users and groups" choose "users". Then right click the "users" option, and choose "New user" (figure 3.15).

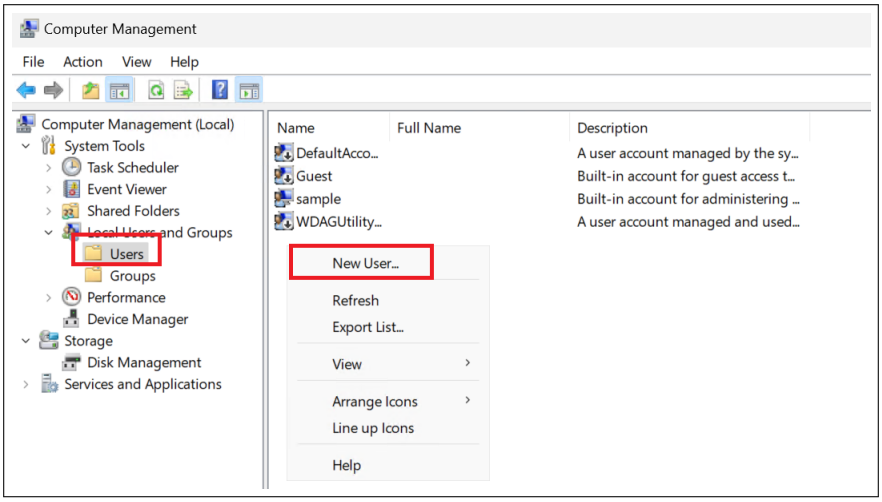

Figure 3.15: Computer management window

In the window that opens, write the name and the password of the account you want to create. Don't forget to repeat the password, to prevent confusion. Then remove the V from the "user must change password at next logon" option, and check the "password never expires" option instead. Choose the password wisely, and write something catchy and simple you would easily remember. After the new account is created, right click it and choose "properties". You can of course choose "set password" to reset the new account's password.

In the properties window, go to the “member of” tab, and click “add”. In the “select group” window write “administrators”. This will make sure that you will be able to use the new account to control other accounts, and particularly unlock them. Don’t forget to click the “Check Names” validation button”, and then “OK” (figure 3.16).

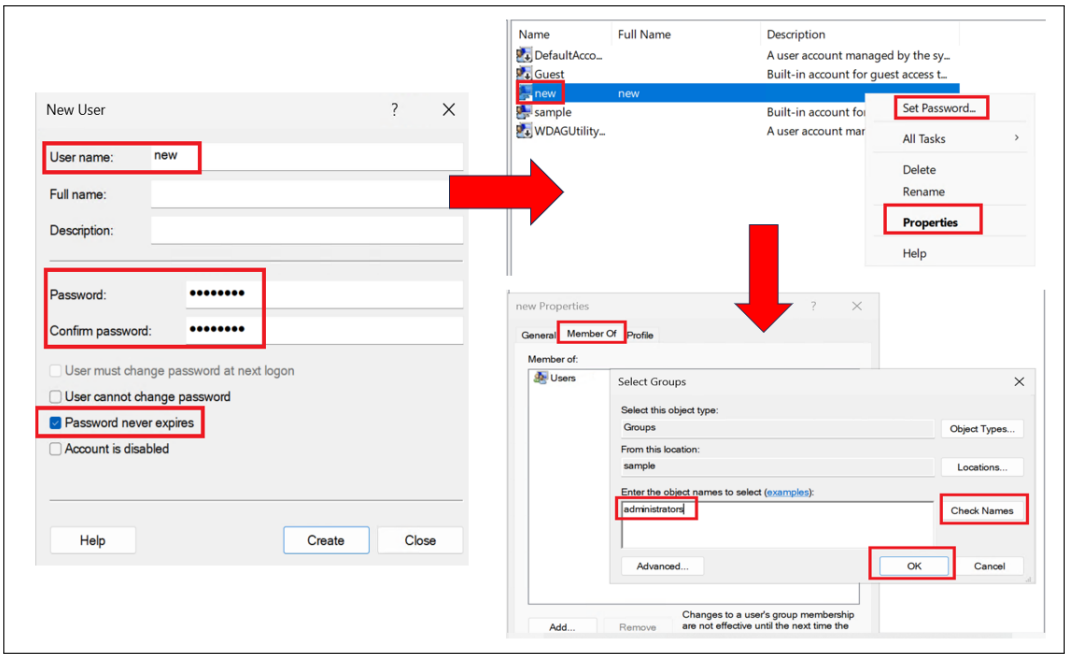

Figure 3.16: Crearing a new administrator account

You can modify the rdp file you downloaded eariler, and add the user name you want to log in with for the next times. Go back to the “downloads” folder, find the rdp file, right click it and choose “edit”. In the “Remote Desktop Connection” window, make sure you work on the “General” tab. Then go the “user name” field, and fill in the name of the account you will want to log in with in the future. Check the “Allow me to save credentials” option, and click “save” (figure 3.17).

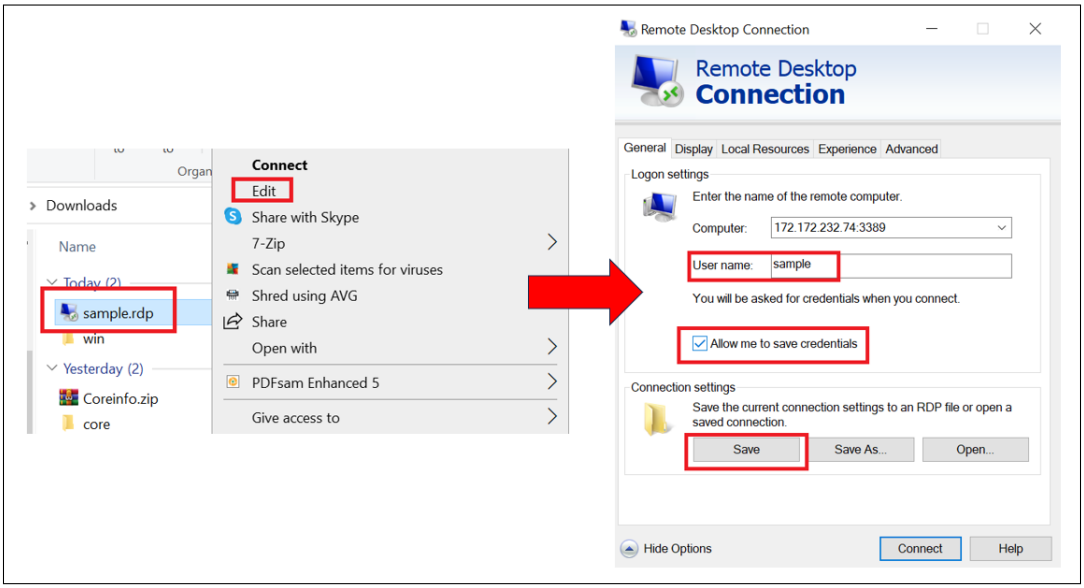

Figure 3.17: Editing the rdp file

3.4 Adding more space to the virtual machine's disk

Imagine the following scenario: you are working on your virtual machine, analyzing MRI scans, but suddenly you notice your C: drive is almost full, and you need more space. How can you add more space to your virtual machine's C: drive? First, you need to shut it down, according to the instructions described before (stop button), or you won't be able to change the size of your C: drive. Then navigate to the virtual machine screen (in this **link**), find your virtual machine, click its name, and choose the "Disks" option from the menu on the left. Then click the disk name (figure 3.18).

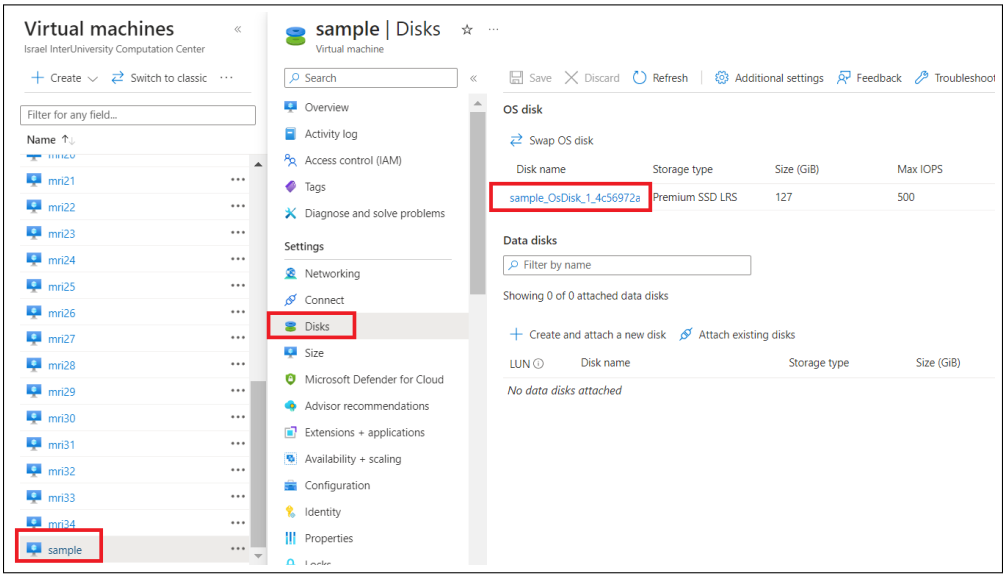

Figure 3.18: Navigating to the "Disks" screen

In the next screen, choose the "size + performance" option from the menu on the left. You can then change the storage type, and the size. You can choose one of the options from the list, or write a specific size. Let's change the size to 160 GB for example, and then click "save" (figure 3.19). Wait a few minutes and turn on your machine.

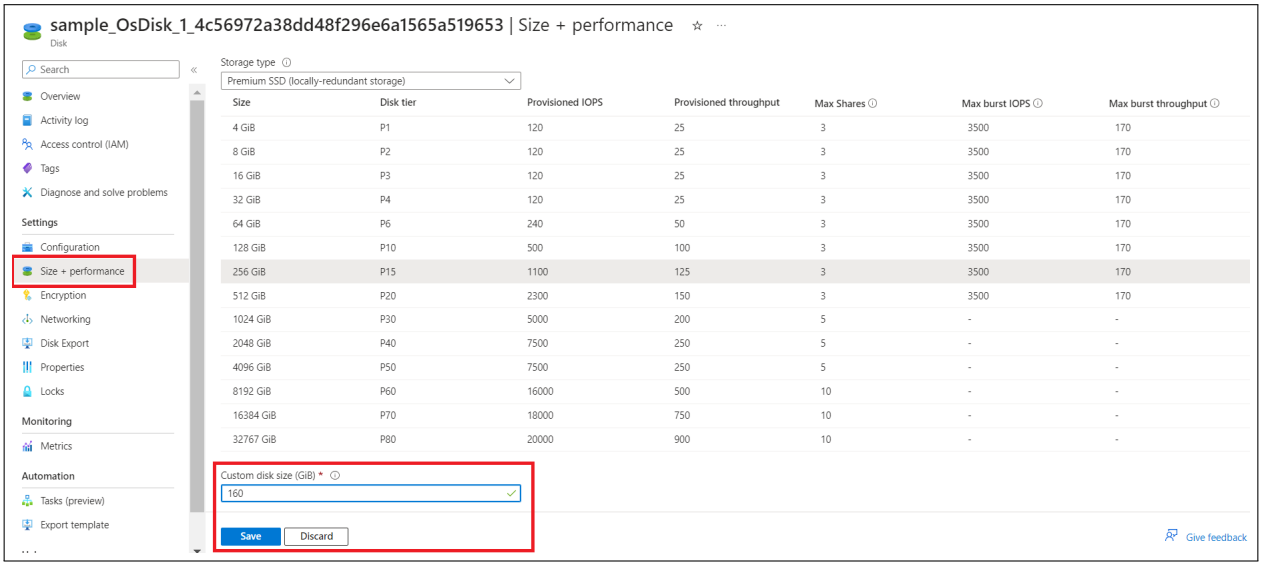

Figure 3.19: Chaning the volume of C drive

If you connect again to your machine, you would notice that C: drive remains the same, because you still need to "activate" it. Go to computer management again (you know the drill), and choose the "Disk Management" option under the "storage" field. You will see an unallocated space under Disk 0 (this is the disk of your virtual machine). Right click the "Windows C" partition, and choose "Extend Volume" (figure 3.20).

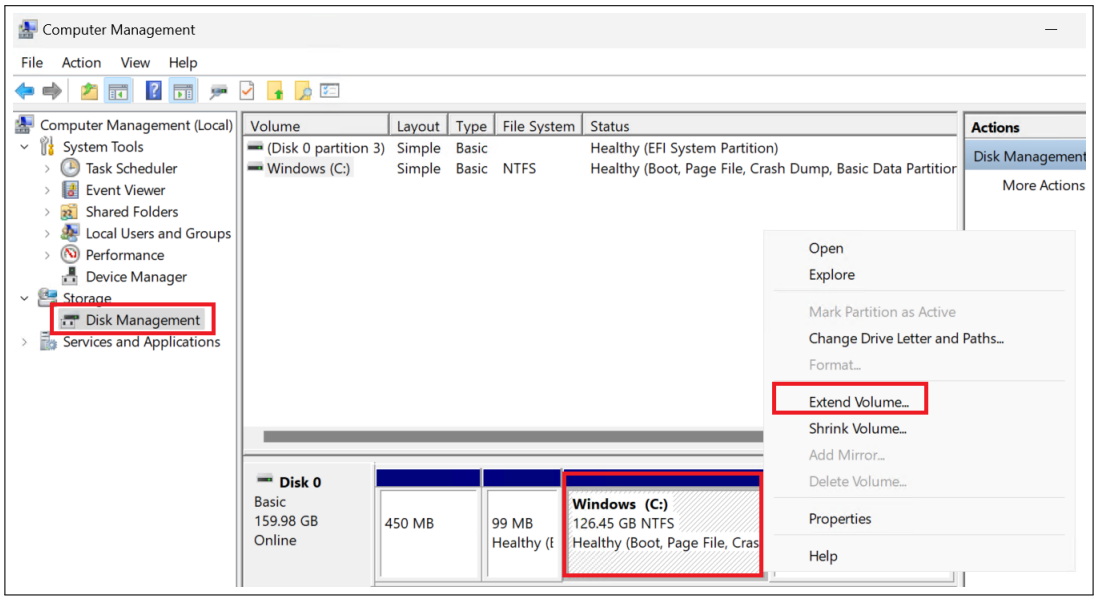

Figure 3.20: Extending C: drive's volume - Part 1

In the "Extend Volume Wizard" click next, next and finish - no need to change anything (figure 3.21). If you now open file explorer, and look at C: drive, you would notice that its volume is increased to the size you have stated eariler.

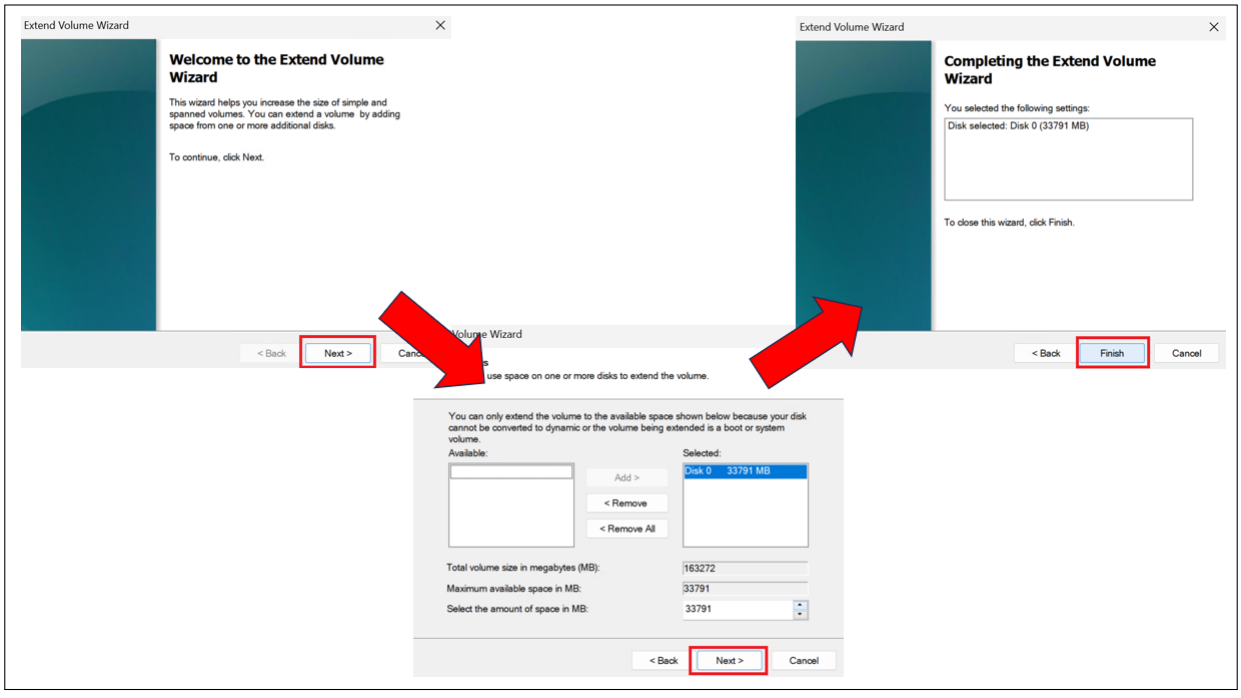

Figure 3.21: Extending C: drive's volume - Part 2

3.5 Creating a shared folder

Another useful tool that exists in a virtual environment is a shared folder, accessible by many users simultaneously. Navigate back to the main screen (in this **link**), choose “Storage accounts”, and then click the “create” button (figure 3.22).

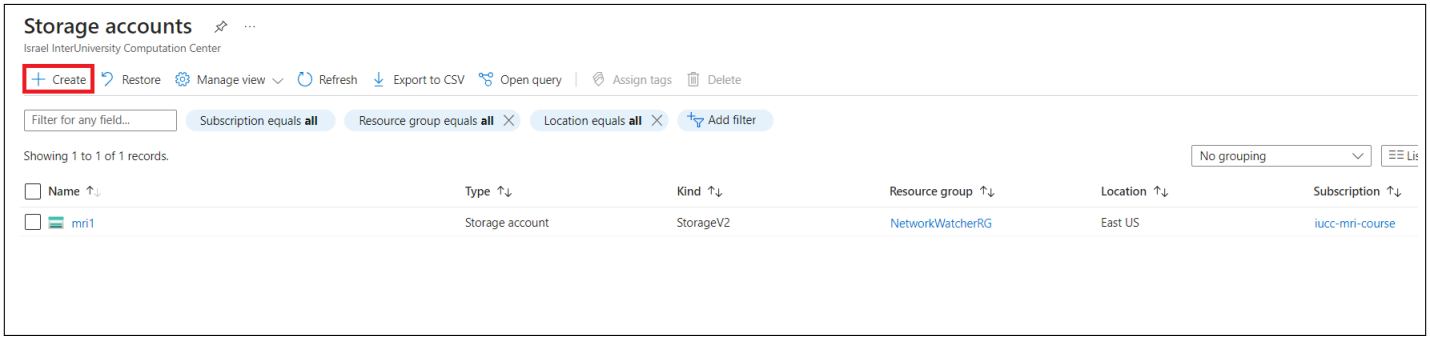

Figure 3.22: Storage accounts screen

In the following screen (figure 3.23), make sure you work in the “Basics” tab, choose “NetworkWatcherRG” under the Resource group field, and choose a name for the storage account. Under “Redundancy” choose the option “Locally-redundant storage” (LRS). No need to change other definitions (under the other tabs), unless you have been told otherwise by the customer service. Then you can click the “Review” button, wait (a few minutes) until the validation process is complete, and then click “create”.

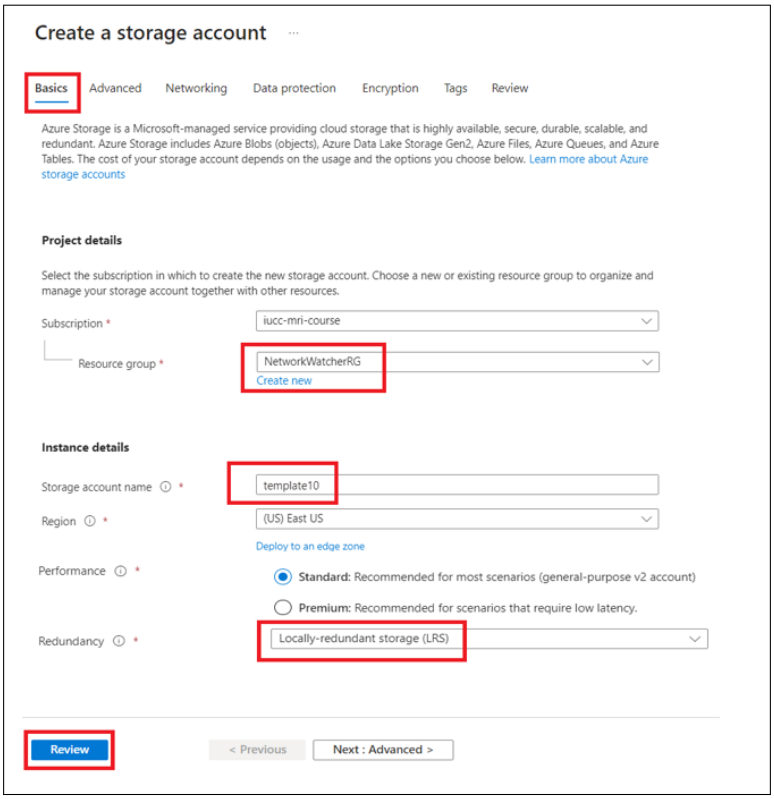

Figure 3.23: Creating a storage account - part 1

After the deployment is complete, click the “go to resource” button (figure 3.23).

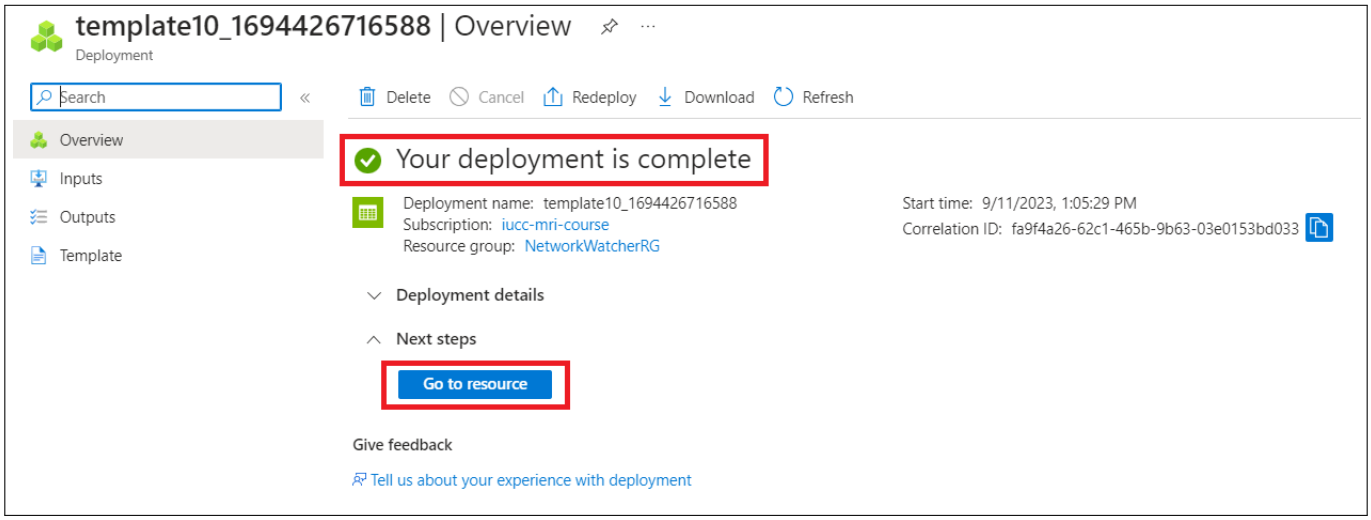

Figure 3.24: Creating a storage account - part 2

In the following screen, choose the “File shares” tab in the menu on the left, and then click the “File share” button (figure 3.25).

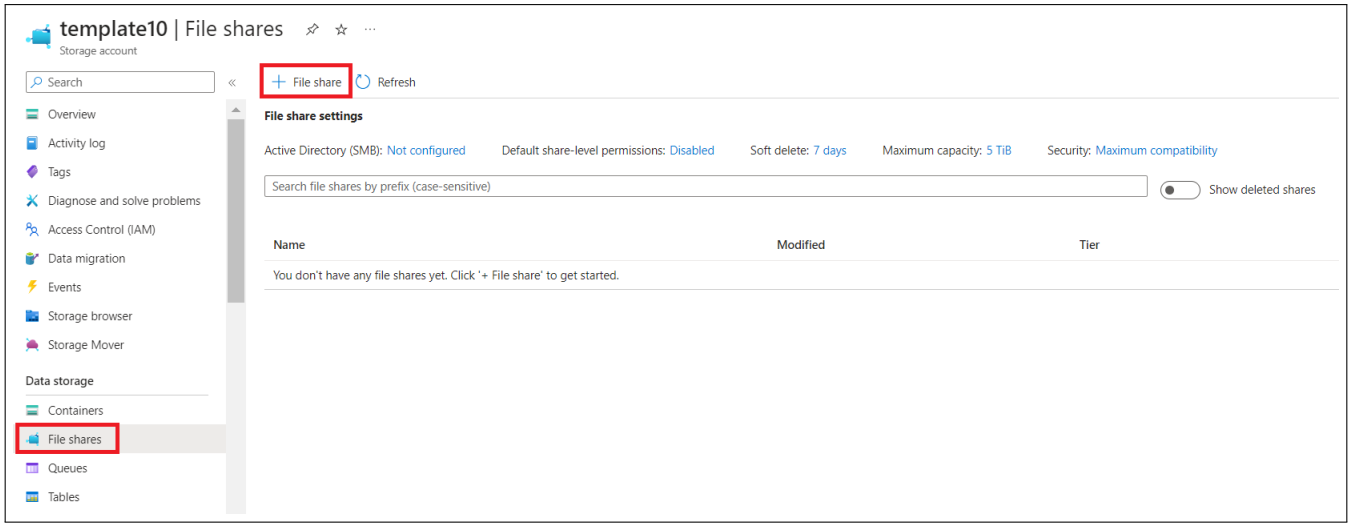

Figure 3.25: Creating a new file share - part 1

Then you need to choose a name for the file share (think of something as informative and relevant as you can). No need to change anything else, unless you have been told otherwise. You can then click the “Review + create” button, wait (a few minutes) until the validation is complete, and then click “create”.

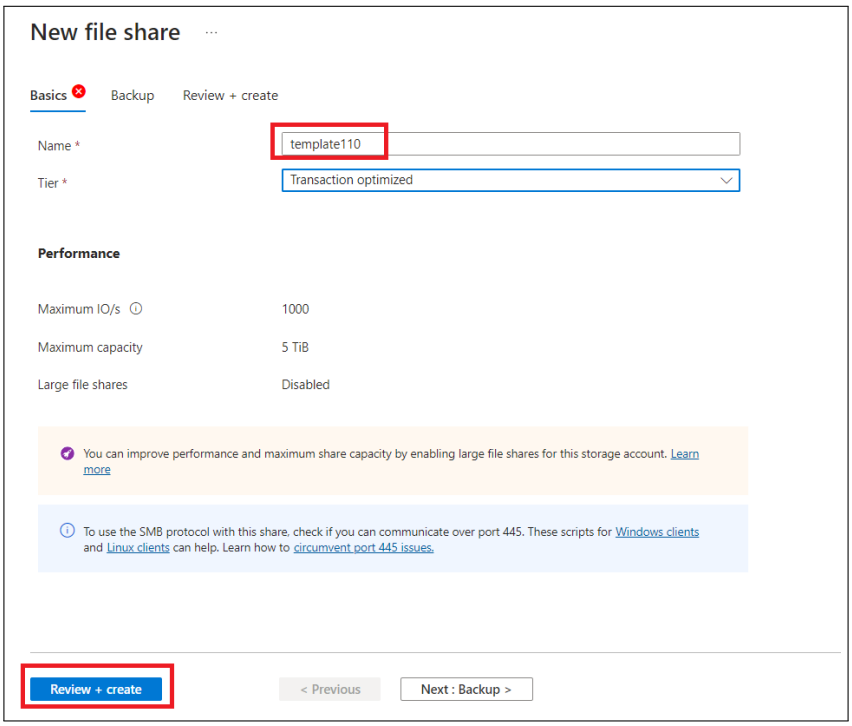

Figure 3.26: Creating a new file share - part 2

If the validation fails, make sure all the mandatory fields are not empty, and then click (again) the “Review + create” button (figure 3.26). Notice that the maximum capacity is 5TB, but you pay **only** for the space you use in practice. Navigate back to the “storage accounts” screen and choose the relevant account. Choose the “file shares” option and then click the name of file share you have created eariler (figure 3.27).

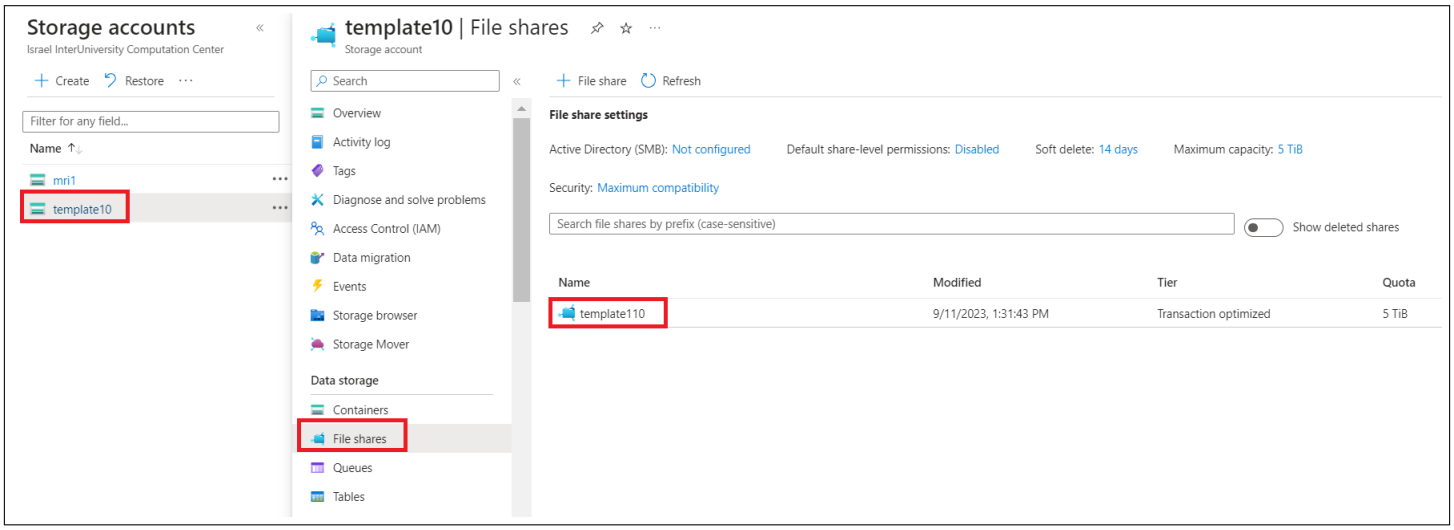

Figure 3.27: Navigating to the “file share” tab

In the following screen click “connect”. A new windows will open on the right: choose the letter of the drive (any letter is good), click the “show script” button, and copy the script to your clipboard (figure 3.28).

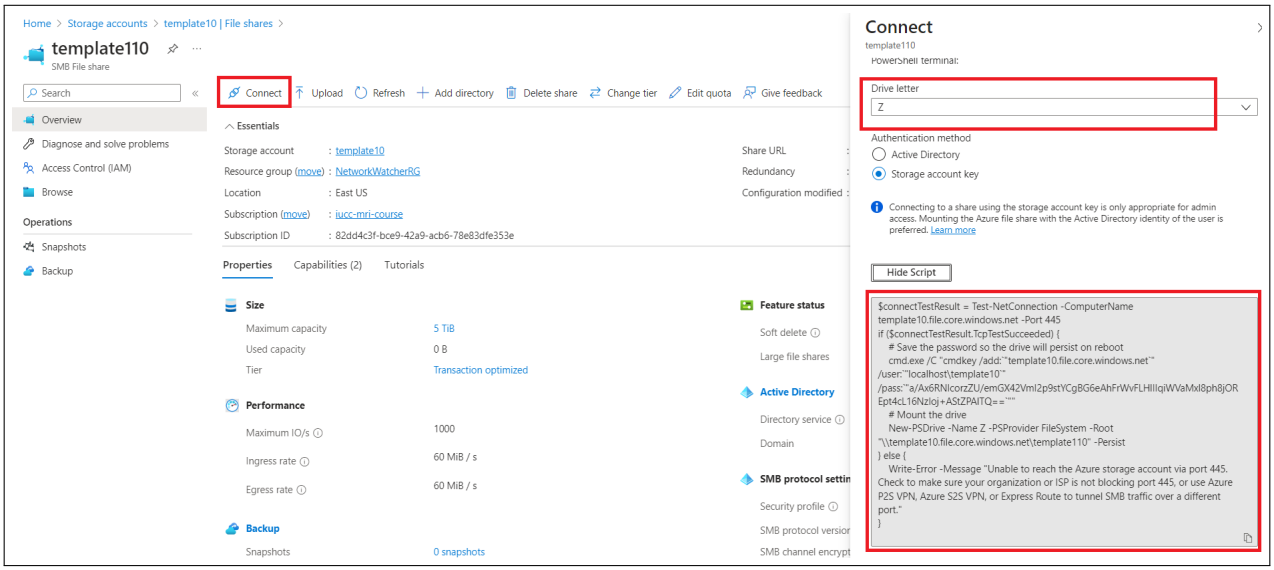

Figure 3.28: Copying the script of the file share

Now, connect to the virtual machine remotely, and open Windows Powershell as administrator. In the Powershell window, paste the script you copied eariler, and press enter. Wait until you see the line “CMDKEY: Credential added successfully”, open the file explorer and click “This PC”. You would notice that a new drive (with the letter Z) appears, with a capacity of 5 TB. This is the shared folder you created (figure 3.29). Anyone with the script can add the link to the shared drive, as described here. Anyone who can access the shared folder can modify the files and folders inside it (everyone has the same permissions, so it’s important to be careful and avoid deleting files that other people may use). The link to the shared drive will continue to exit after the virtual machine reboots.

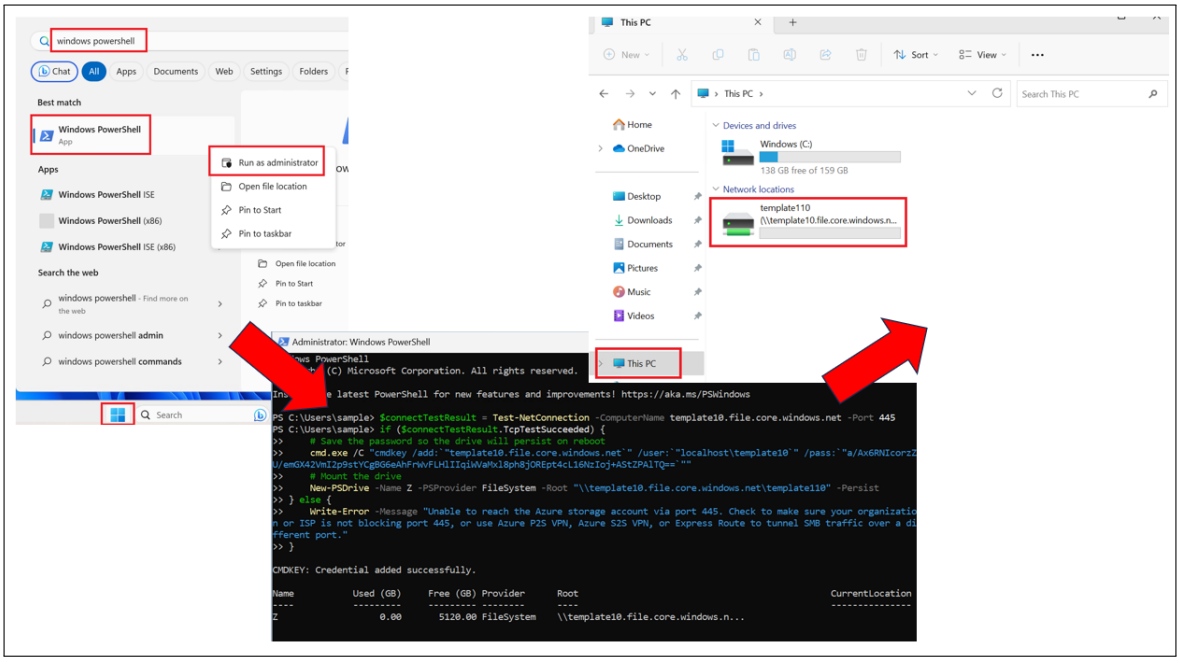

Figure 3.29: Adding a link to the shared drive

3.6 Installing Docker and Neurodesk

Now that the virtual machine is ready, you can continue and install Docker and Neurodesk. The installation process is almost identical to the one on a physical station, except there is no need to check the prerequisites and turn on the WSL 2 feature. You do need to install the WSL kernel update (in case you encounter the error message shown in figure 2.5). If you create "regular" users on the virtual machines (without administrator privileges), make sure to give everyone a "full control" permission to the Docker application, as explained previously.

3.7 Tips and insights

Let's try summarizing few tips and insights regarding the installation of Docker and Neurodesk on virtual machines:

- Don't create many virtual machine before you make sure you can successfully install Docker and Neurodesk on ONE machine.
- Don't install Windows Server (2019, 2022 or other versions) on your virtual machine - the internet is full of people who tried installing Docker on this operating system, but failed.
- Consult Azure customer service about your machine. Don't hesitate to tell them what you need, and if something is not right - let them know, so you could fix it together.
- If you work with Neurodesk - work only with the local C: drive of your machine. The shared drive does not "interact" well with Linux. It's best to use local paths while working on Neurodesk.
- If you want Neurodesk to work fast - ask the costomer service for a strong machine with powerful processor, and big amount of RAM.
- Monitor the C: drives of the machines, and extend their volume if necessary.
- If end users need to work with the machine - write specific instructions how to do it. They might not know how to use the RDP file properly.
- You can ask Azure for a virtual machine that already contains a Docker installed. If you do that - make sure it works fine without additional actions or installations.
- Working with Azure console requires some patience - when you create, delete, stop, start or do other actions, wait a few minutes until their completion. You can click the bell button at the top of your screen and read notifications regarding actions you have made.
- You are charged for every hour each machine works. If not in use - shut the machines down.
- Try preparing the machines few weeks before you actually need to use them. You will need this time to fix bugs and unexpected errors.
- Don't hesitate to use the internet in order to find useful information - many people have tried installing Neurodesk in the past. Their experience is priceless, and they will probably be happy to help if you ask politely.
- Stack overflow is a great **platform** to exchange ideas and thoughts. You can also consult Neurodesk's developers, in this **link**.

References

[1] **Neurodesk Website**

[2] **Container** on Wikipedia.

[3] **WSL** on Microsoft.

[4] **Docker** installation.