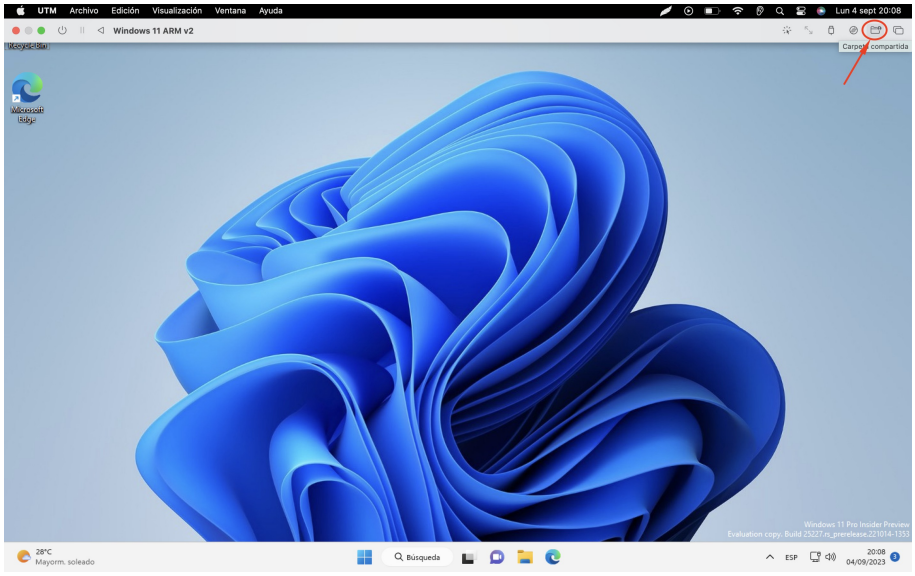
Sharing Files in the UTM Virtual Machine.

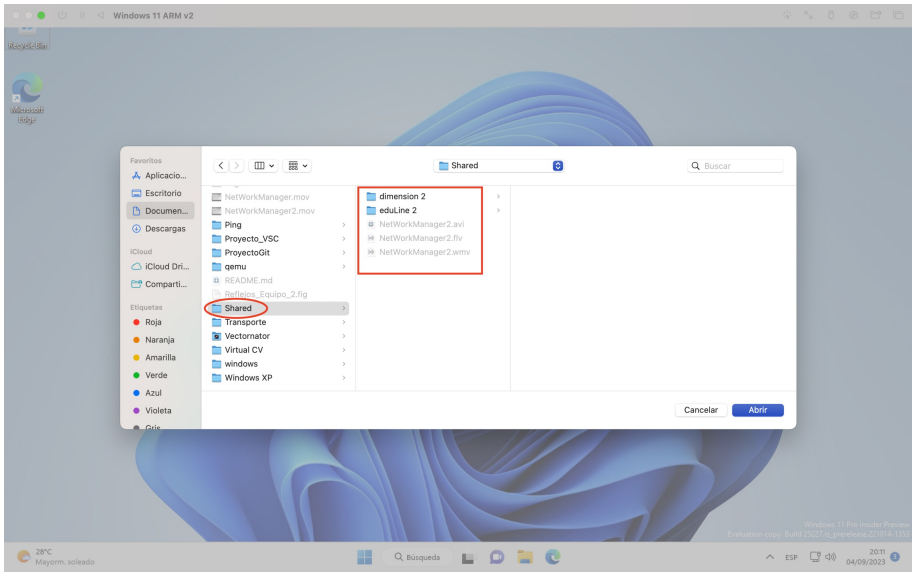
Shared files through a directory.

Windows 11 ARM:

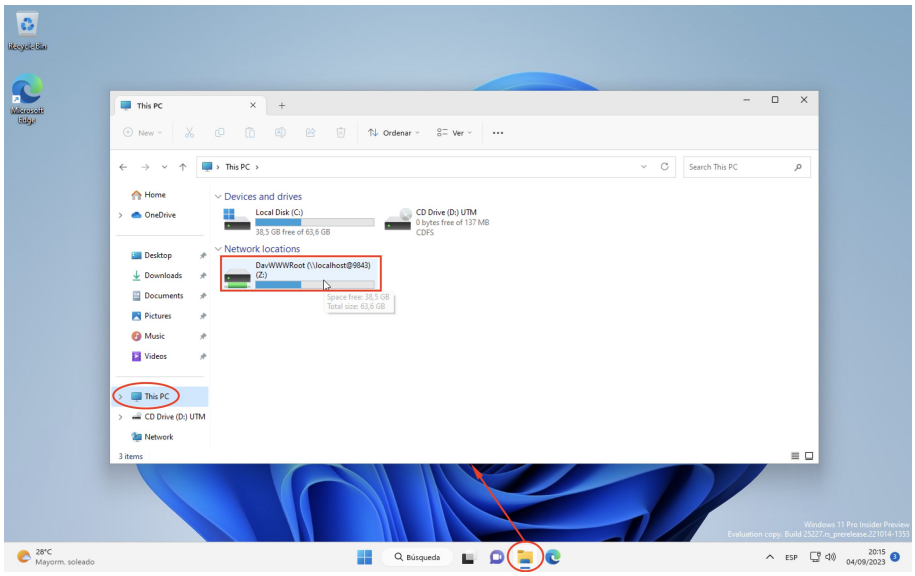
When we want to exchange files in Windows 11, we can go to the top right menu in the virtual machine window, where we'll see a **folder icon**.



*Click on it to open, and choose the path you want to share. You can select any folder; in this case, I chose a folder named **Shared**, which contains some items inside. Click **Open**.*



In Windows 11, open **File Explorer** (folder icon), and in the new window that opens, go to the bottom left and select **This PC**. You'll see a network drive labeled **Z:** — select it.



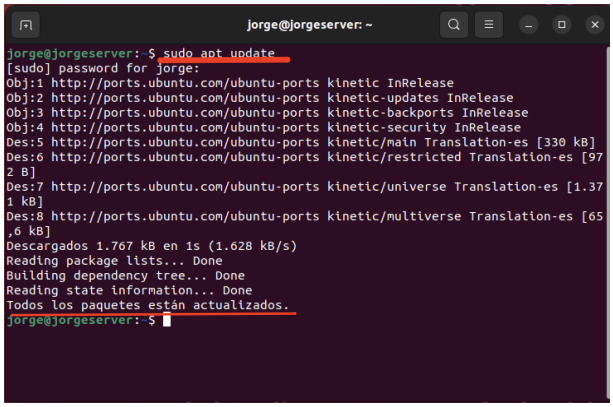
And we can see that it contains the same items as the previous **Shared** folder. This directory will be used to share all files between the host and guest machines—in other words, between our **Mac** and the **virtual machine**.

Ubuntu ARM Server:

On any GNU/Linux system, we need to install tools from the Linux package manager to improve and enable file sharing between the two machines.

To do this, first run the following command in the terminal:

```
sudo apt update
```



It's also a good idea to update the operating system, so run the following command:

```
sudo apt upgrade
```

Now, let's install the required packages (for more information: https://docs.getutm.app/guest-support/linux/). To do this, first run:

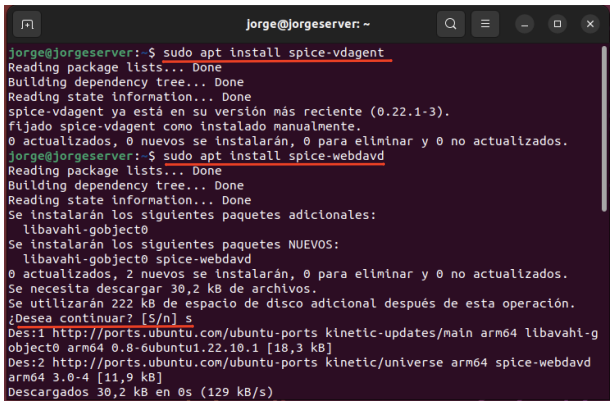
```
sudo apt install spice-vdagent
```

Then:

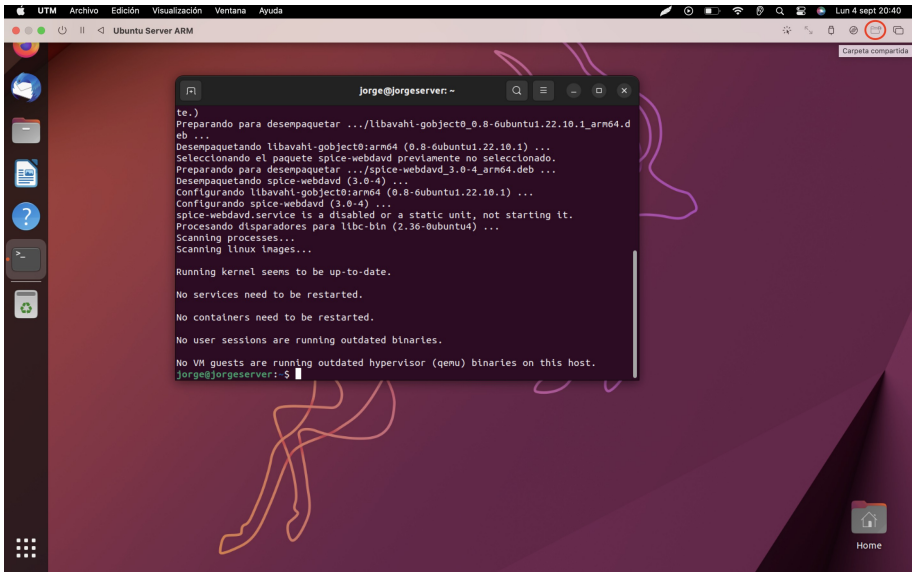
`sudo apt install spice-webdavd`

And finally:

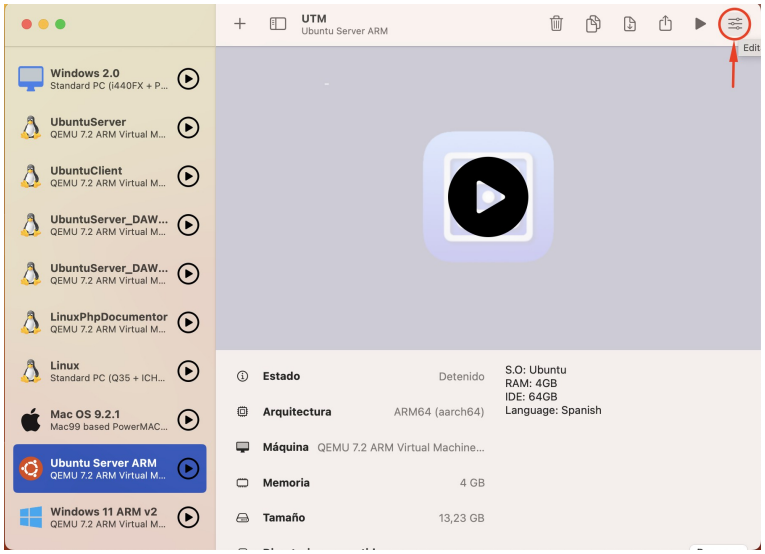
`sudo apt install qemu-guest-agent`



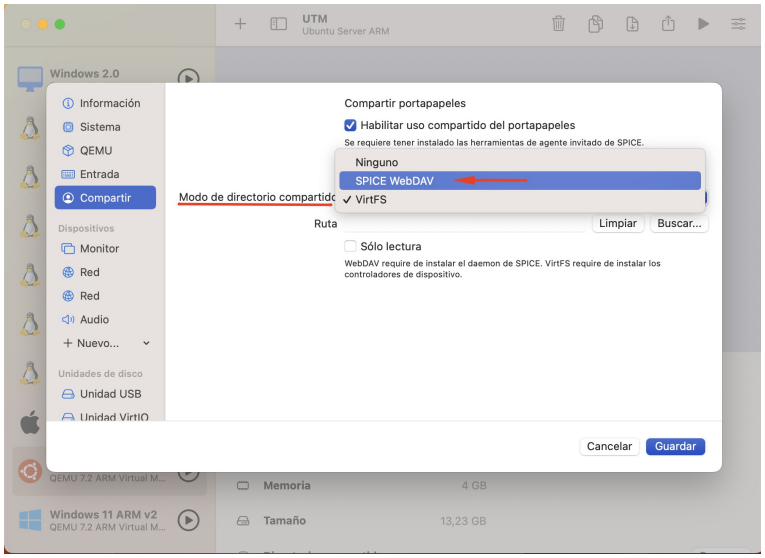
We can see that the folder icon is disabled:



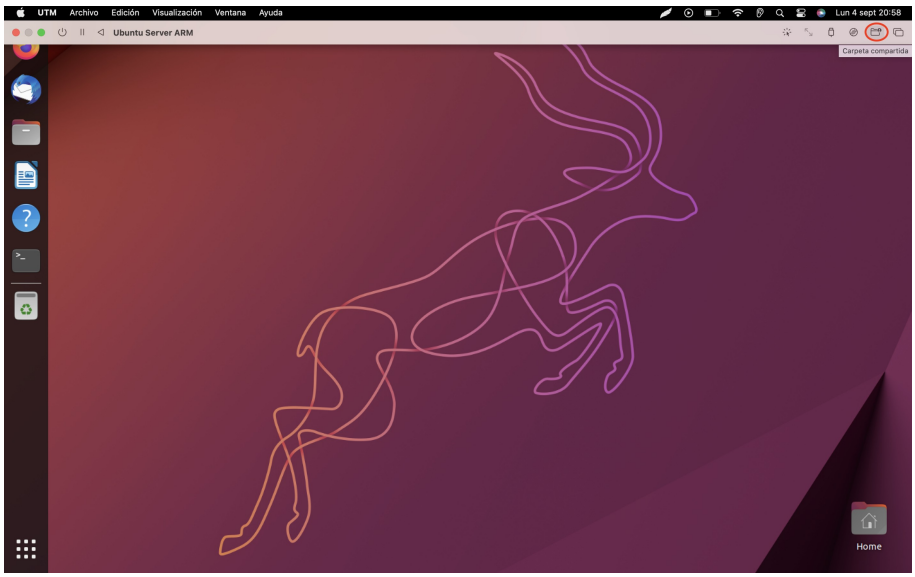
Shut down the virtual machine and go to the machine’s settings.



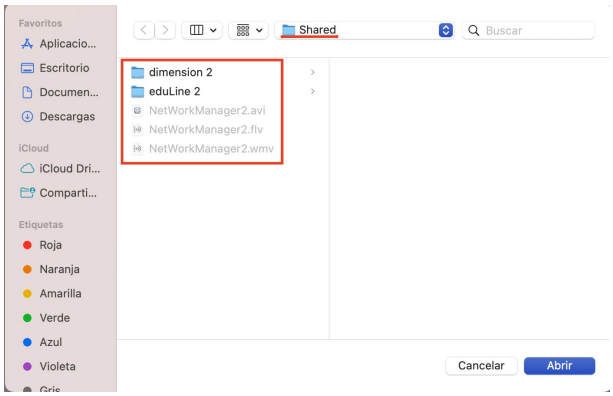
*Go to the **Sharing** section in the settings menu and select **SpiceWebDAV** for the **Shared Directory Mode**, then click **Save**.*



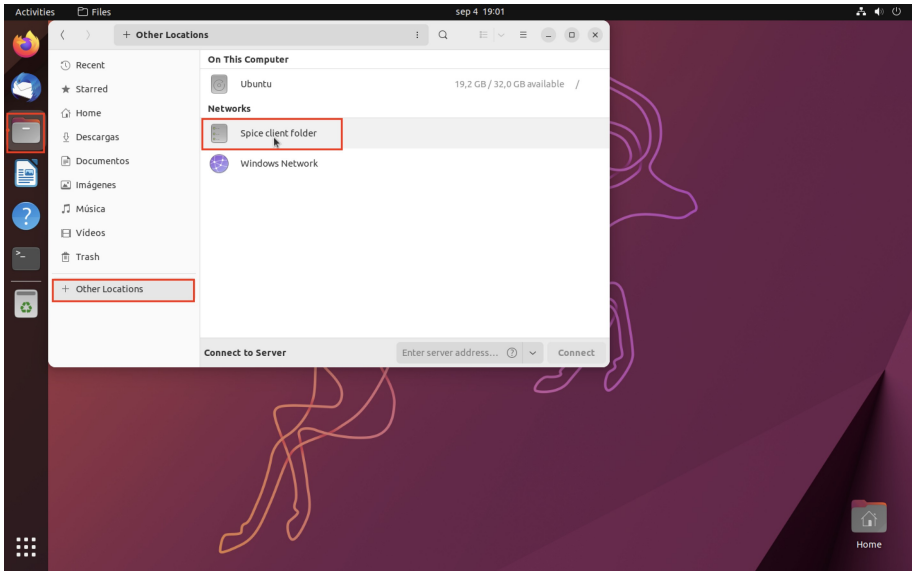
Start the virtual machine, and now you should see the **shared folder icon** in the top-right corner.



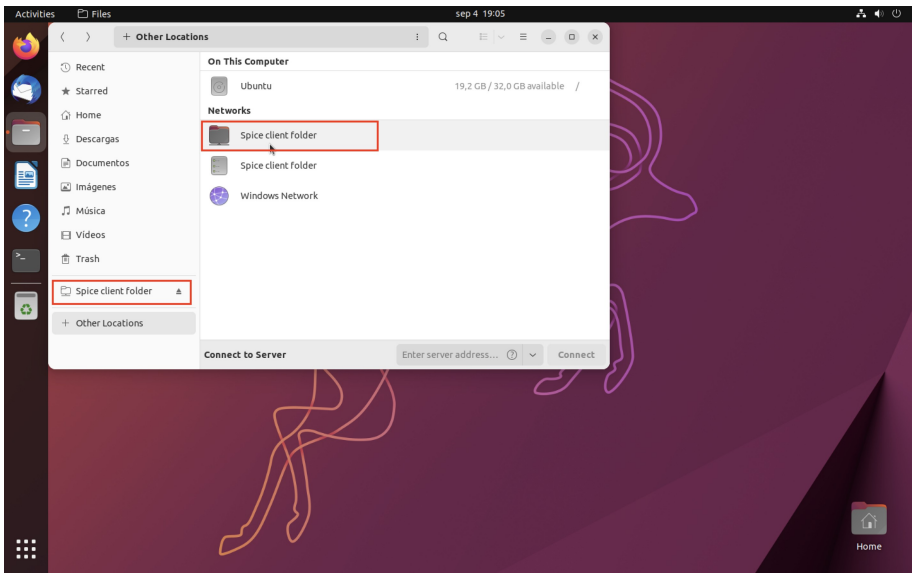
Click on the icon and select the path you want to choose. In this case, select ***Shared*** *and click* ***Open****.*



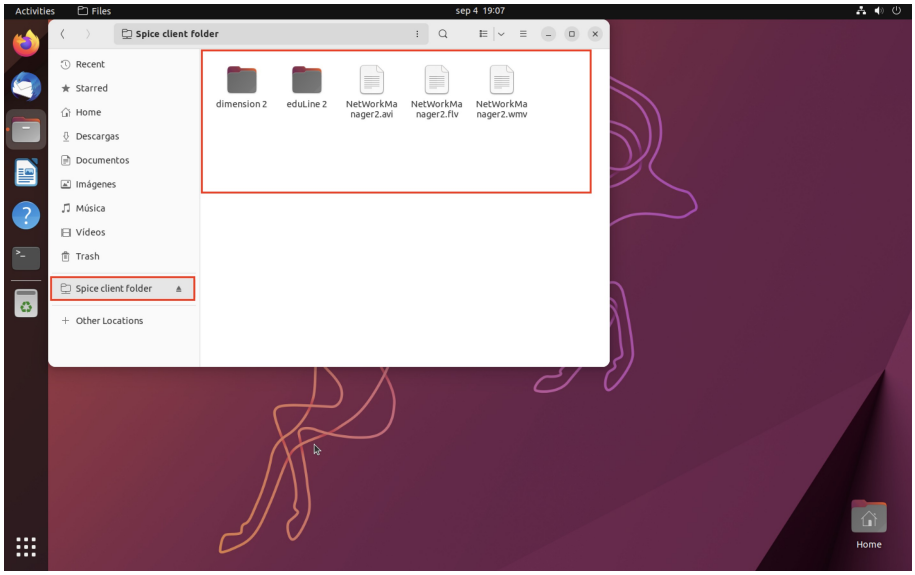
Now, in the Ubuntu virtual machine, click on ***File Explorer*** *(the Folder icon)* → ***+ Other Locations*** → ***Spice Client Folder****.*



Now, we can see that a network drive opens for us.



*Inside this directory, you will find the contents of the **Shared** folder.*

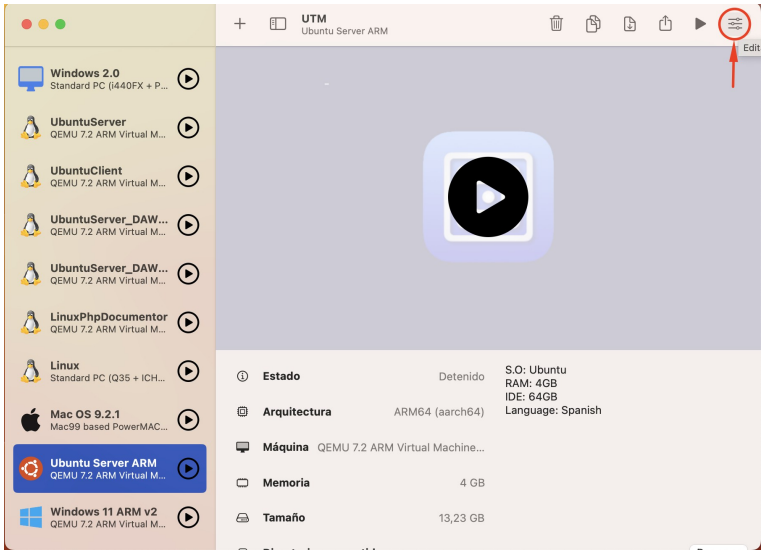


Note: This folder doesn't always work well for me; sometimes it gets stuck or takes a long time to load and appear. But don't worry, there's another way to do it, using **VirtFS**.

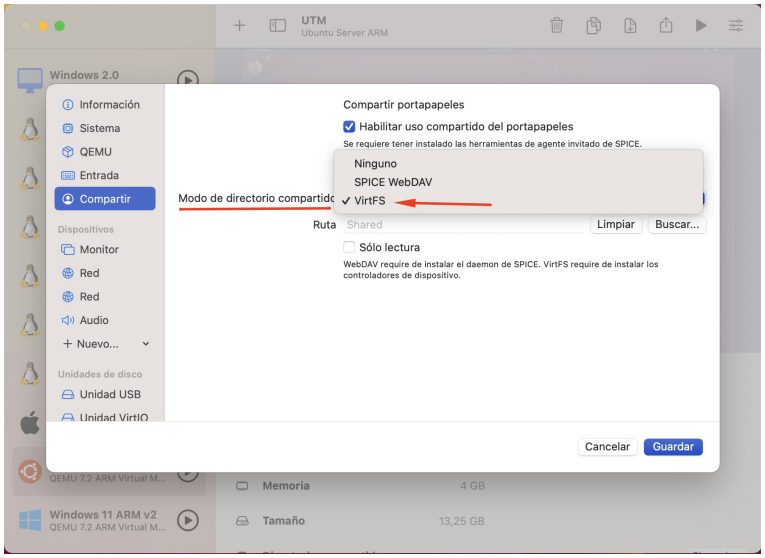
With VirtFS:

Learn more: https://docs.getutm.app/guest-support/linux/#virtfs

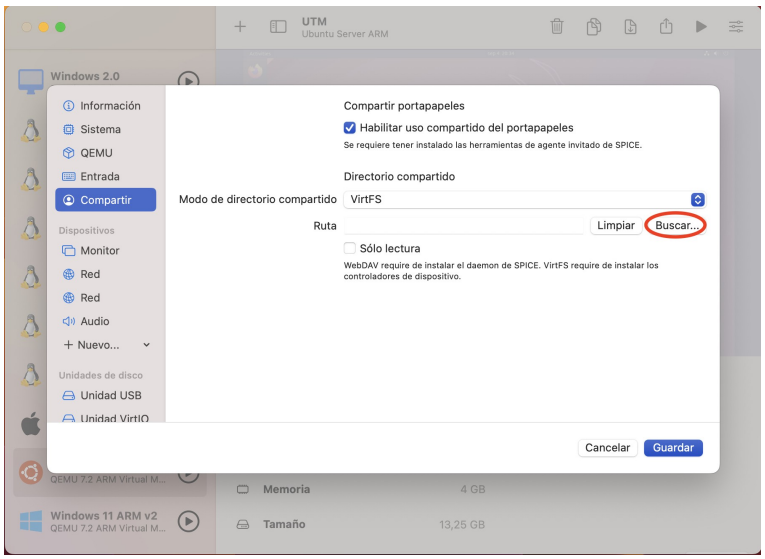
Shut down the virtual machine and go to the machine's settings.

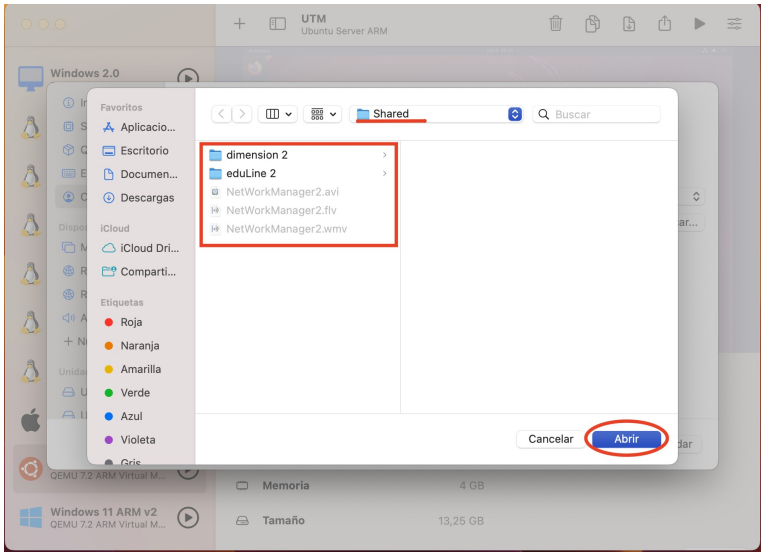


Go to the **Sharing** section in the settings menu and select **VirtFS** for the **Shared Directory Mode**.



*In the **Path** field, choose a directory path, any path you prefer. In this case, select **Shared**.*





Start the Ubuntu machine and open the terminal.

First, check which directory you're in by running:

```
pwd
```

(This will show something like /home/jorge, but it may vary for you. Remember this directory, as we will need it later to configure a file so that the shared folder always appears when Ubuntu starts.)

Create a directory where the shared drive will appear:

```
sudo mkdir Host-home
```

(You can name it whatever you want; I used **Host-home** in this case. It's better not to use spaces in the name.)

Now, let's mount the shared drive in the directory using VirtFS:

```
sudo mount -t 9p -o trans=virtio share Host-home -oversion=9p2000.L
```

(We are mounting **Host-home** from a relative path, as we are already in /home/jorge. If you prefer, you can use the absolute path: /home/jorge/Host-home.)

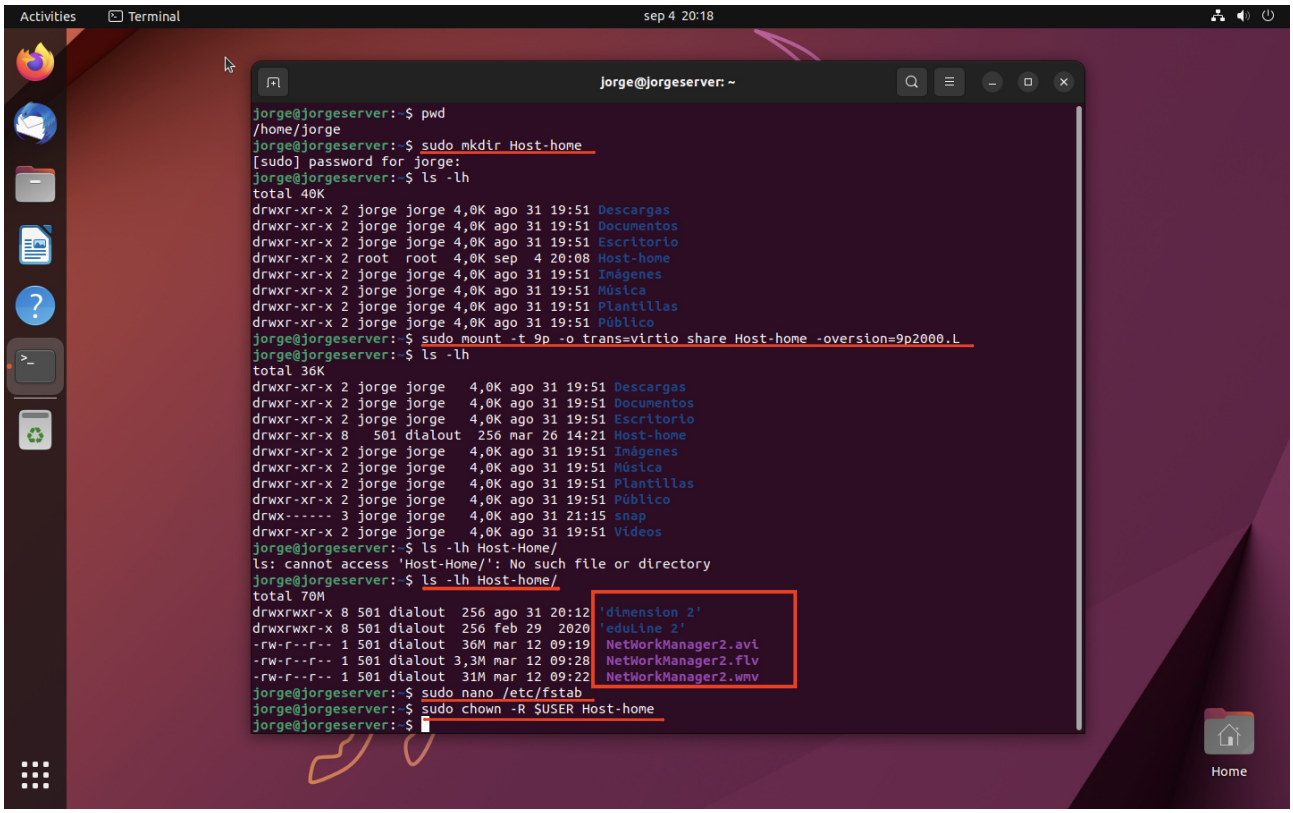
Next, let's give superuser permissions to this directory, otherwise, you won't be able to create directories, unmount the drive, etc.:

```
sudo chown -R $USER Host-home
```

Finally, let's view the contents of the directory:

```
ls -lh Host-home
```

(You should now see the same files that were in the **Shared** folder on your Mac.)



Now, let's make sure the shared folder appears every time you start Ubuntu (this is necessary because, without this, you would have to manually mount it each time you boot up). To do this, we need to modify the /etc/fstab file:

1. Open the file for editing:

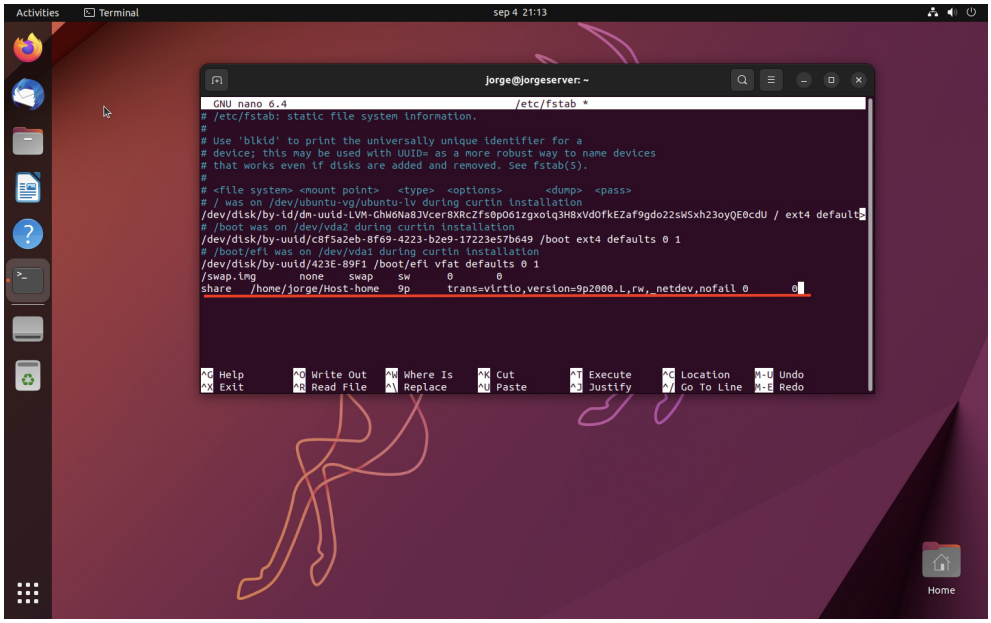
```
sudo nano /etc/fstab
```

2. Add the following line at the end of the file:

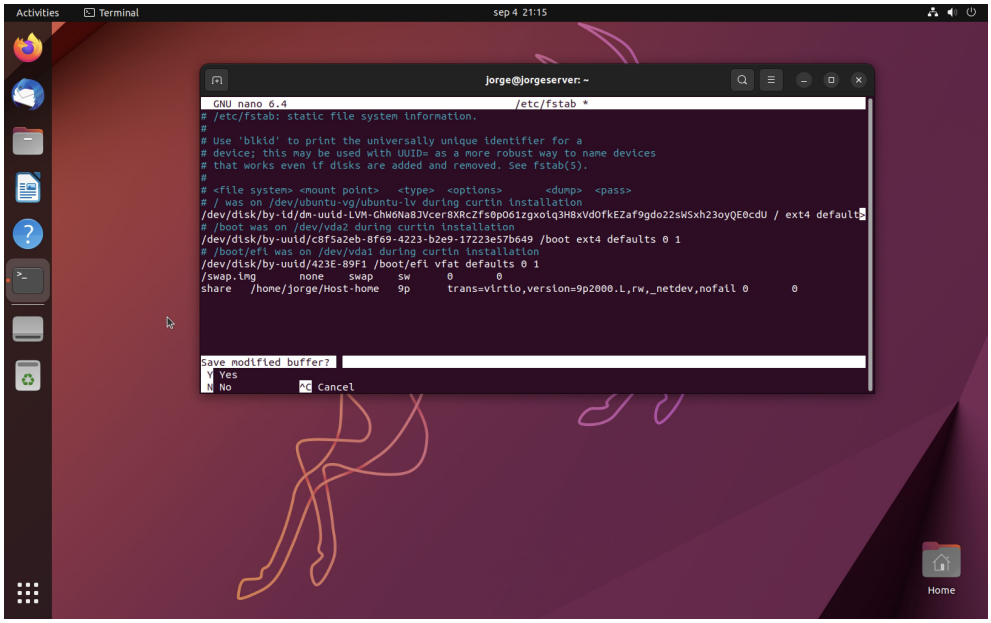
```
share /home/jorge/Host-home 9p trans=virtio,version=9p2000.L,rw,_netdev,nofail 0 0
```

3. Save the changes by pressing Ctrl + O, then press Enter.
4. Exit the editor by pressing Ctrl + X.

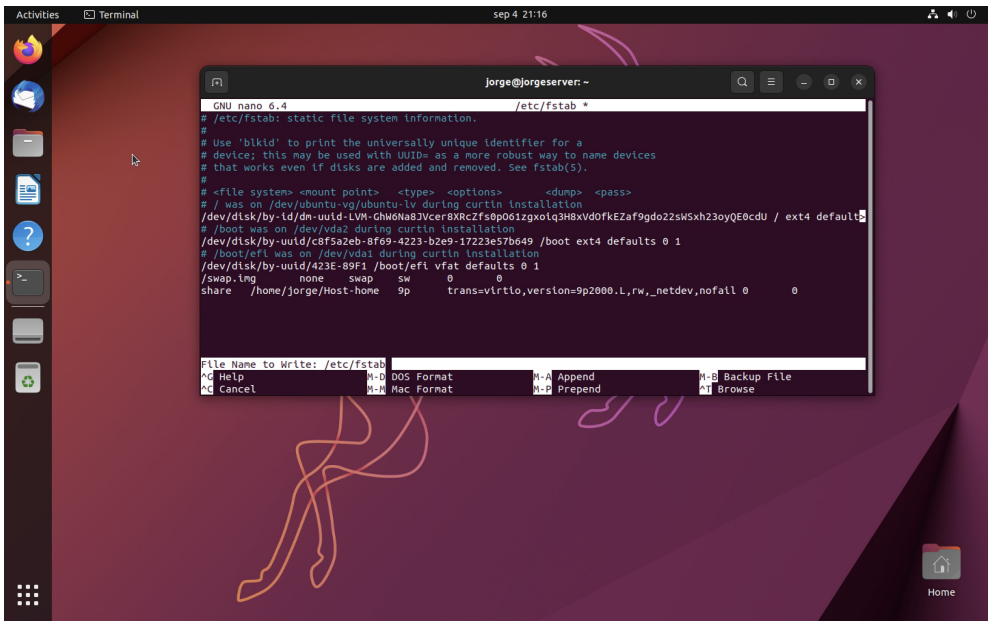
This will automatically mount the shared folder at boot.



*Press **Ctrl** + **X** to exit the editor.*



Save modified buffer? Press **Y** to confirm saving the changes.

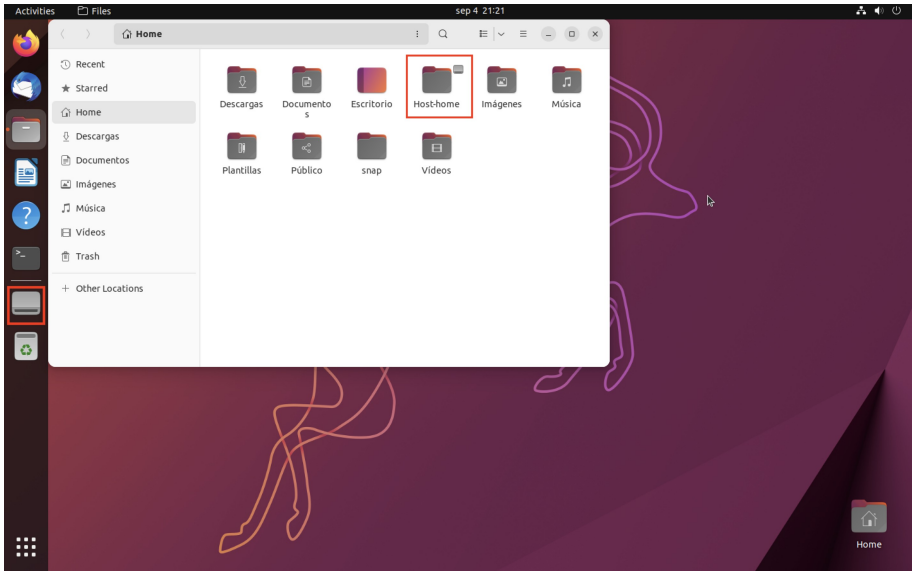


When prompted with **File Name to Write: /etc/fstab**, press **Enter** to confirm saving the file.

Now that the changes are saved, to check if everything is working, restart the system by running:

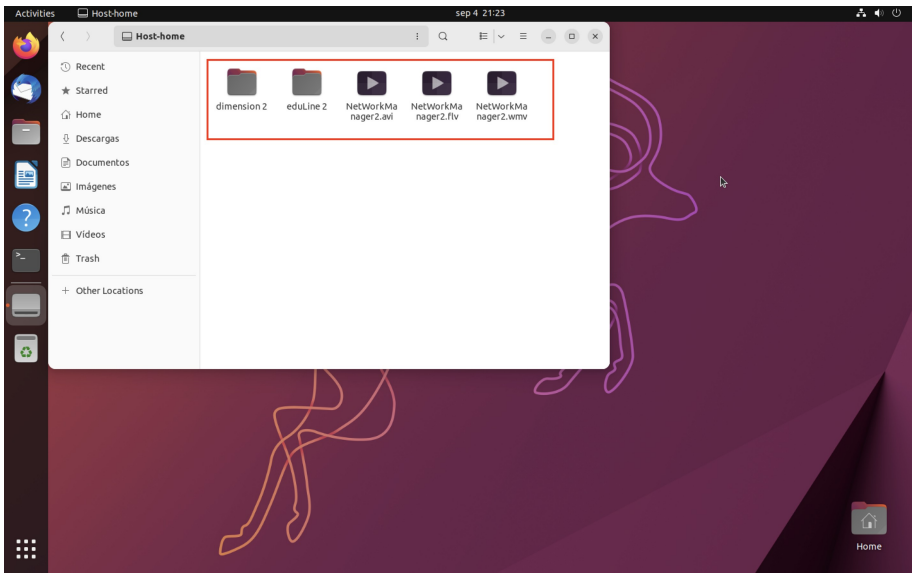
```
sudo reboot now
```

This will reboot your machine and apply the new configuration.



*Now, we can see a direct icon for the shared drive, or if we open **File Explorer**, we can see a **Host-home** shared folder.*

*Inside this folder, we can confirm that it contains the same items as the **Shared** folder on our Mac.*



This is the best way to share files through a directory in Ubuntu/Linux.

Note: If at any point you change the path in the **Sharing** section in the virtual machine settings, in the new path, when you restart Ubuntu, you won't have administrator rights. To fix this, you'll need to run:

```
sudo chown -R $USER Host-home
```

This will grant administrator rights to the new path.

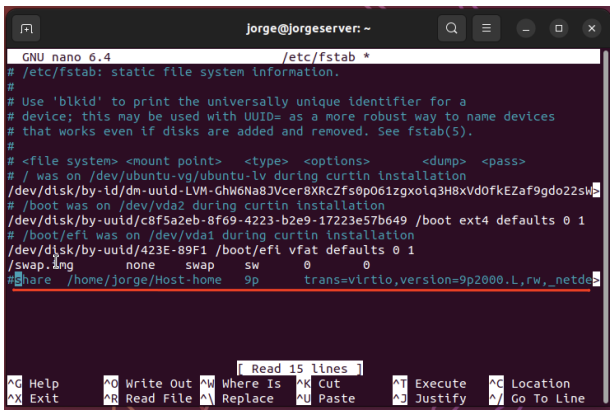
If you revert to the previous path, you won't lose your administrator rights on the old directory; you'll still have those permissions.

In other words, if you change the shared folder path in **VirtFS** and grant administrator rights to each new path, the guest machine will retain administrator rights over those paths on the host machine.

Unmount and remove the shared directory:

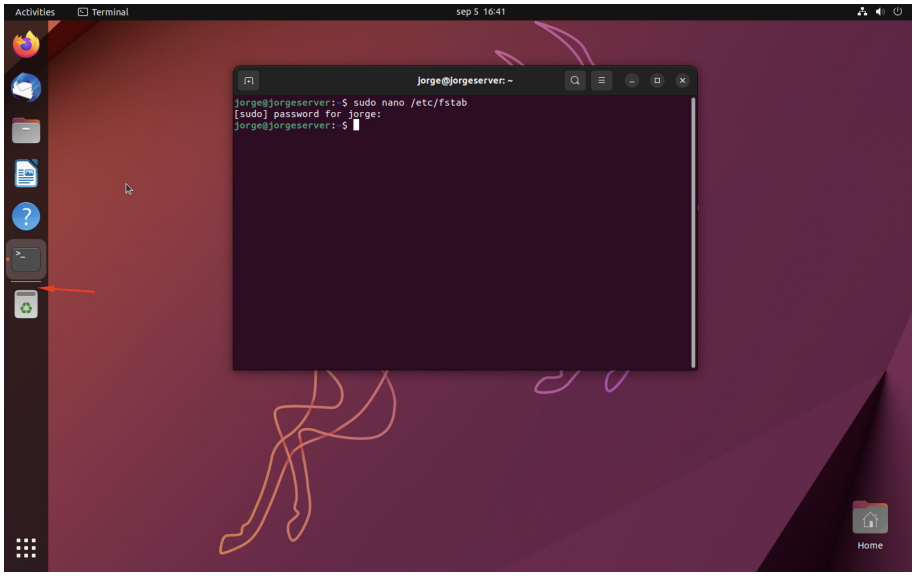
Open the terminal and modify the file /etc/fstab → sudo nano /etc/fstab.

The last line we added can either be deleted or commented out so it's ignored (I recommend the latter, as it will keep it handy in case you need it later. If you use it again, make sure to update it with the new absolute path if it has changed). To do this, add # at the beginning of the line you previously added.



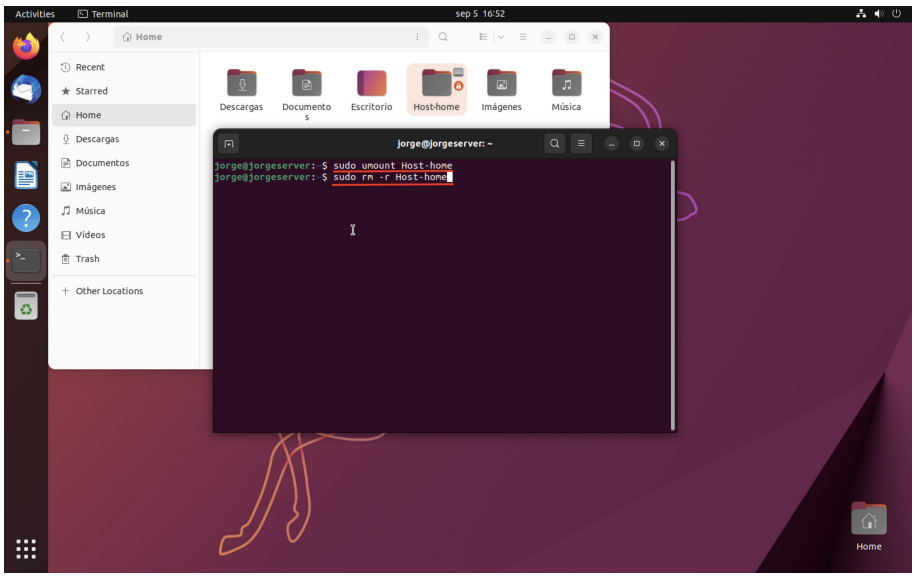
*To save → **Ctrl + X → Y → Enter**.*

Now it will have been removed from the sidebar menu.



*And to remove it from **File Explorer**, from the terminal, type → sudo umount Host-home.*

To delete it completely → sudo rm -r Host-home.

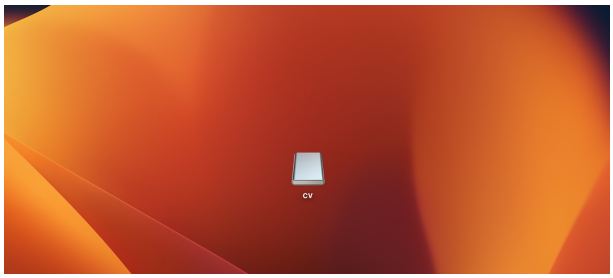


*Yes, with that, we have removed the **Host-home** folder.*

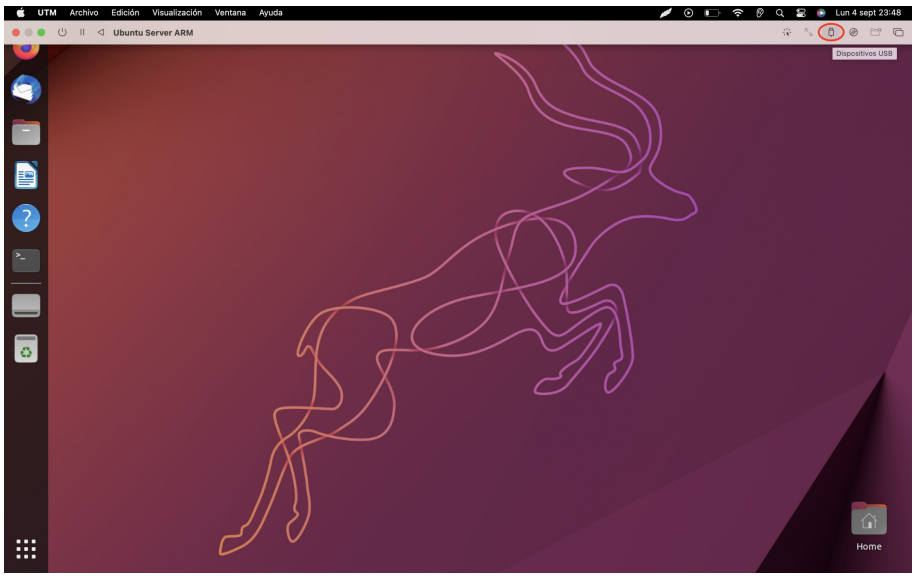
Shared Files via USB:

Finally, we just need to see how files can be shared via USB.

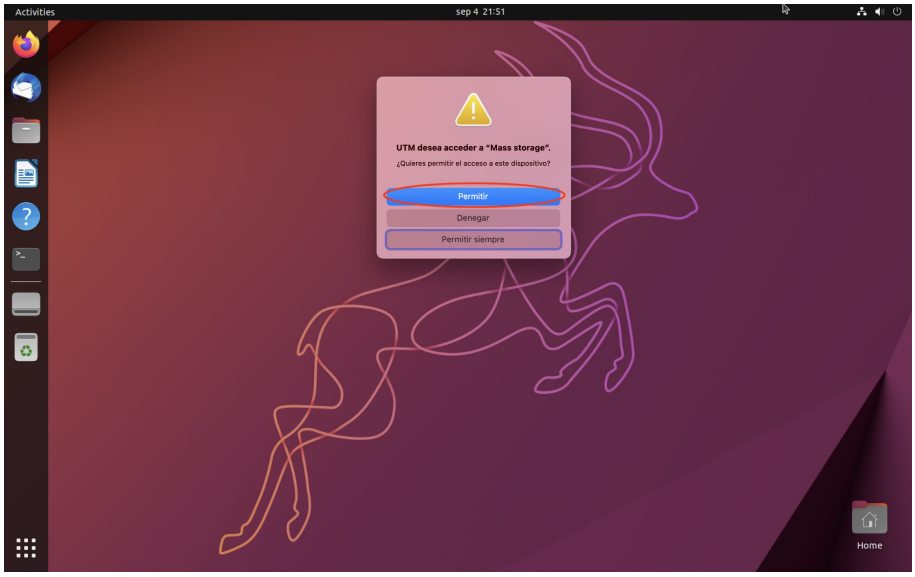
We connect our USB to the Mac, and the drive icon will appear on the desktop.



Now, go to the virtual machine, and in the icon menu at the top-right of the window, click on the USB icon.

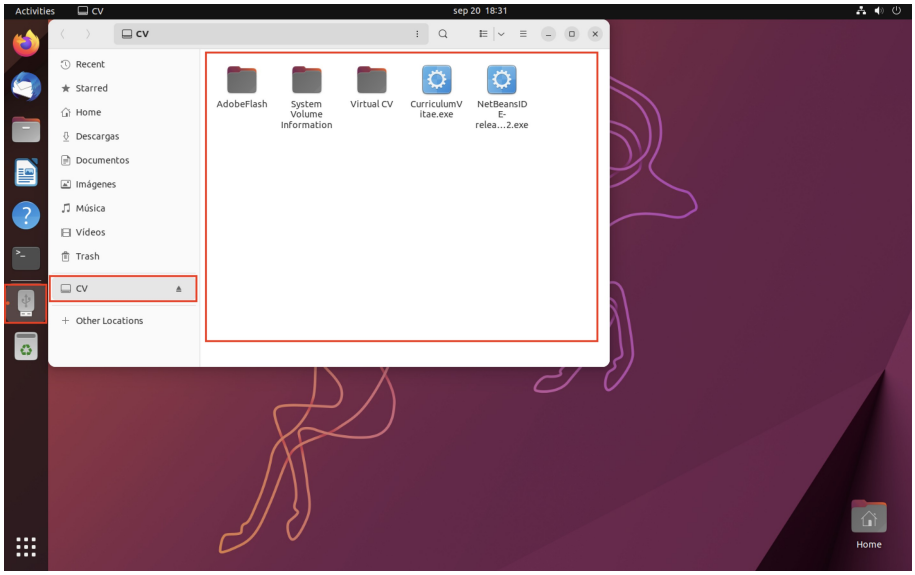


Two options will appear, select **mass storage (0:2)** and click **Allow**.

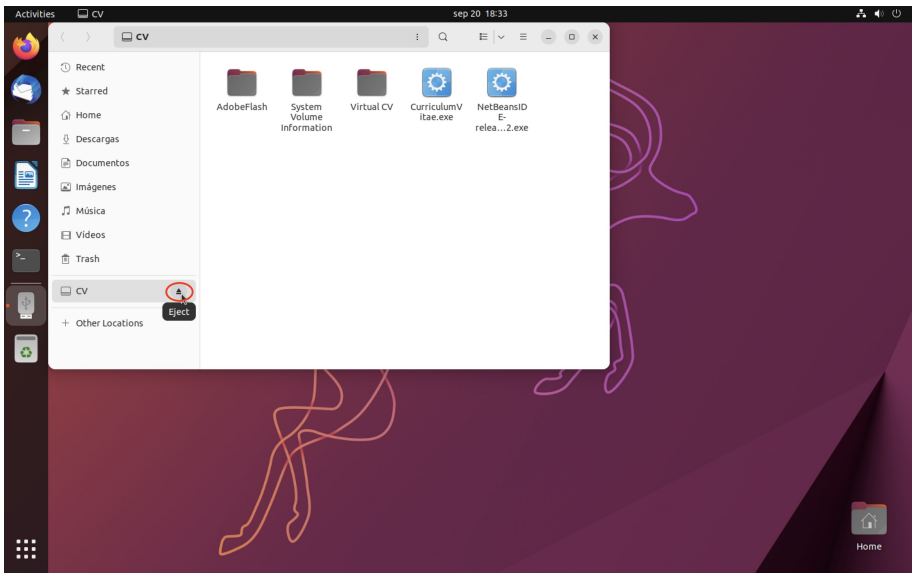


Now, the USB will no longer appear on the desktop of our Mac, and a pop-up warning will appear stating that the disk was ejected improperly. Don’t worry, just close the message.

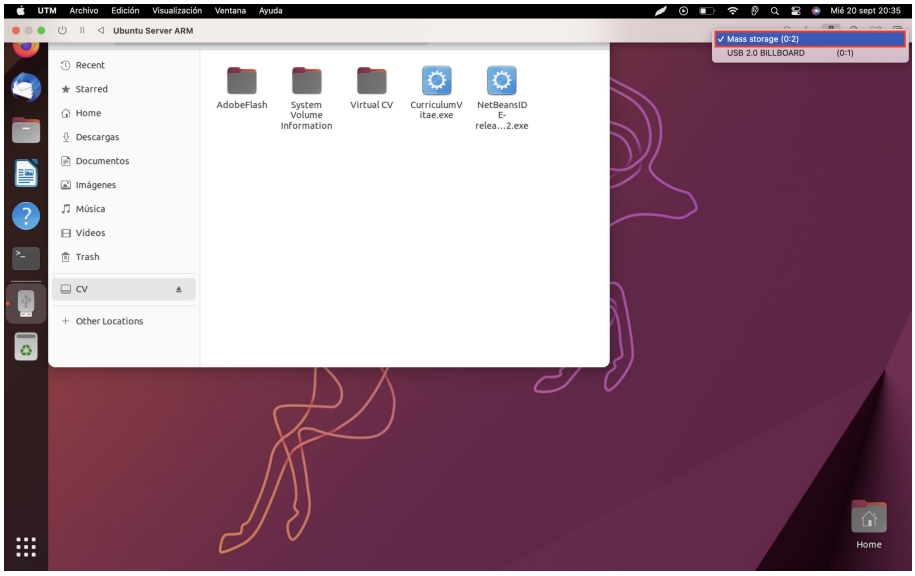
Now, go to your virtual machine, and you will see that the USB device appears.



We are going to eject the USB, and there are two ways to do it. One way is by clicking the ***Eject*** *icon.*



It will only eject the USB with the first method, but the second method is to go to the USB icon, as we did before, and deselect ***Mass storage (0:2)****.*



With this method, the USB is ejected and reappears on the Mac desktop. When we want it to be shown again in Ubuntu, we just have to select **Mass storage (0:2)** — it's a way to use the USB as a container between the two machines.

The same applies to Windows 11 ARM, following the same process.