

CS 2200 - VM Installation Guide For M1 Macs Only

Written By: Patrick Liu

Introduction

If you are like me and have a shiny new M1 Mac, you most likely have run into issues with lots of software. This is because the new M1 chips are using an ARM architecture, and in this specific case, programs such as VirtualBox/Vagrant do not work with these chips.

(If you are unsure of if your Mac is an M1 chip as opposed to an Intel one, simply find the **About this Mac** in the spotlight search, and next to **Chip** there it should say what chip your Mac is using)

The solution we will be using is called **UTM**. Don't worry about how it works, effectively it is an alternative to VirtualBox for the new M1 chips.

Installations

Go to <https://mac.getutm.app/> to install the dmg, and open the dmg and follow the instructions to add the application to your Mac.

Once this is complete, you need to download the **Ubuntu 20.04 64 bit ARM server ISO**. PLEASE MAKE SURE THAT YOU ARE DOWNLOADING THE ARM VERSION. The link to download the ISO is here: <https://cdimage.ubuntu.com/releases/20.04.2/release/>

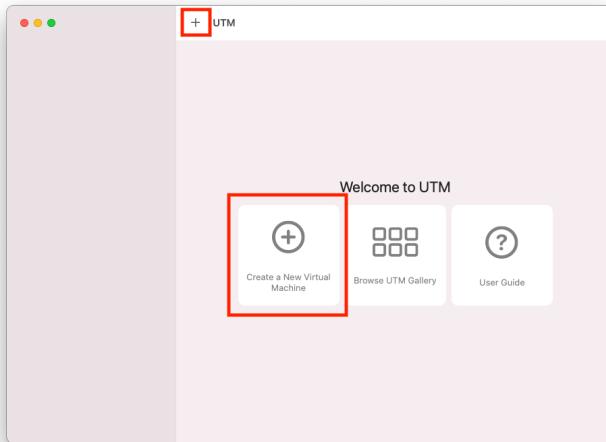
This is a 1GB download, so as long as you have a reliable connection, this installation should be quick.

Setting Up Your 2200 VM

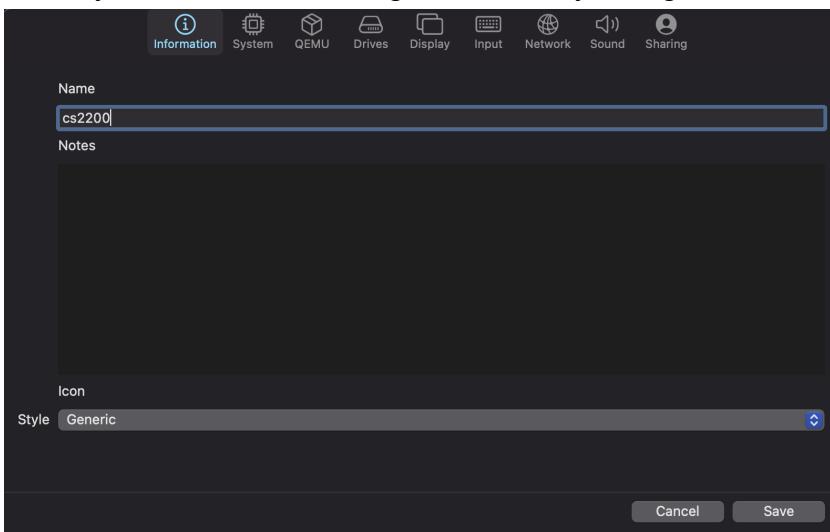
Most of what I say here is covered in this guide: <https://mac.getutm.app/gallery/ubuntu-20-04>, but I do think the instructions can be explained more clearly.

If you want a video walkthrough, here is a tutorial on Youtube (although it might be a bit more difficult to understand what he is saying): <https://www.youtube.com/watch?v=A0LNG4rlTw0>

1. Open UTM and click **Create new Virtual Machine**.

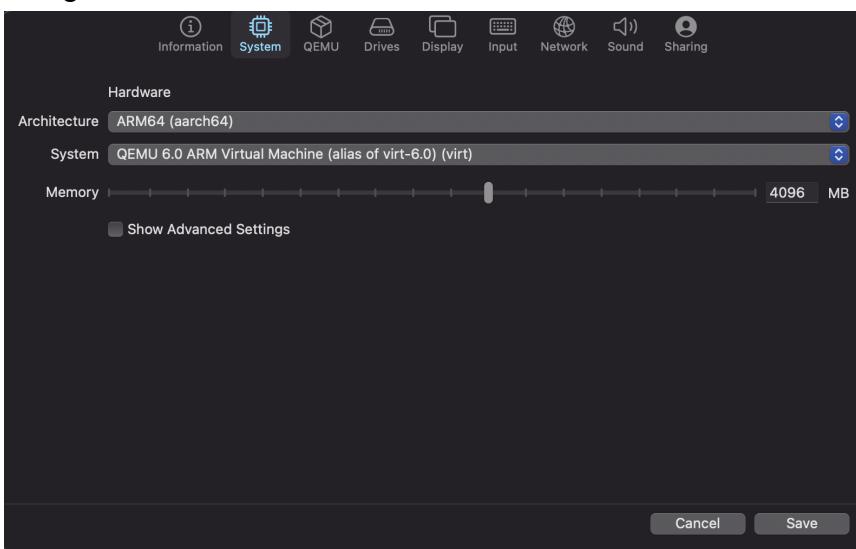


2. Name your VM to something that is easily recognizable, in this case I called it **cs2200**.

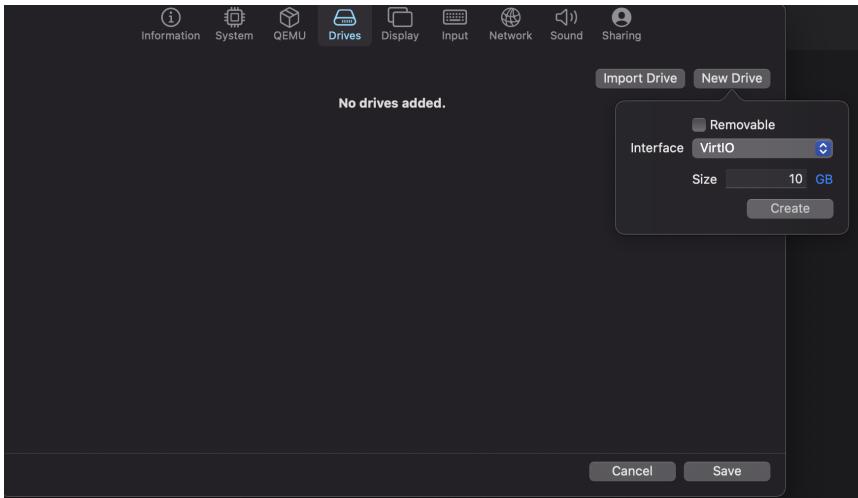


3.

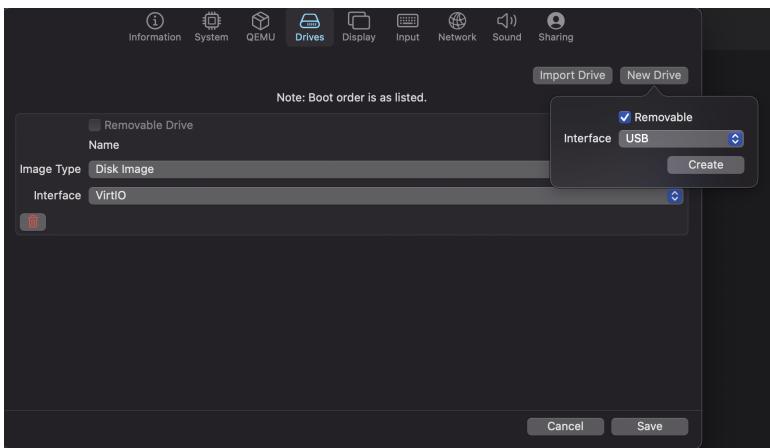
4. In the **System** tab, under the **Architecture** dropdown, make sure you select **ARM64 (aarch64)**. For the **System** dropdown, you do not need to change anything. For the memory, this is how much memory the VM can use. I would highly suggest you allocate 4096 MB(4GB) or higher.



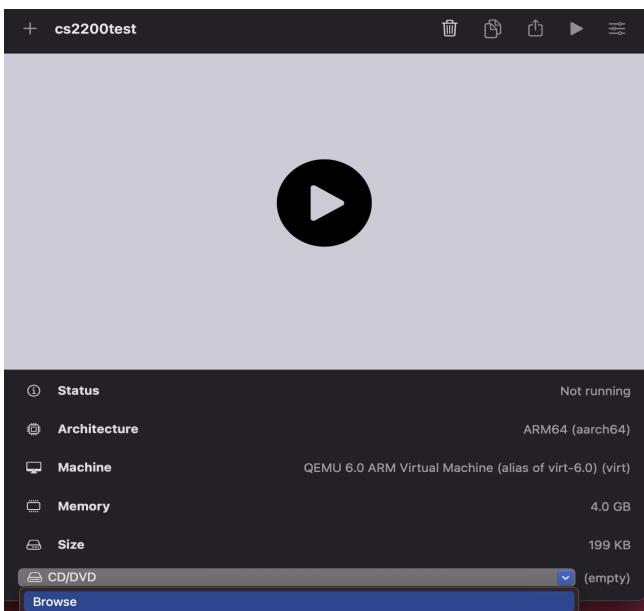
5. Under the Drives tab, click New Drive. The default settings should work just fine, click **Create**.



6. Create another New Drive using the same process as step 4, but in the step, check the **Removable** box. Now click **Save** in the bottom right to save the VM configuration.



7. Now, you should be taken back to the UTM home screen, and your screen should look something like this. Under the CD/DVD tab, click **Browse** and choose the 20.04 Ubuntu 64 bit ARM ISO that you downloaded earlier.



8. Now, click the **Play** icon in the middle or top right to launch the VM. It might take a while to boot, so be patient. If you are taken to an EFI shell (looks like a terminal and you see the

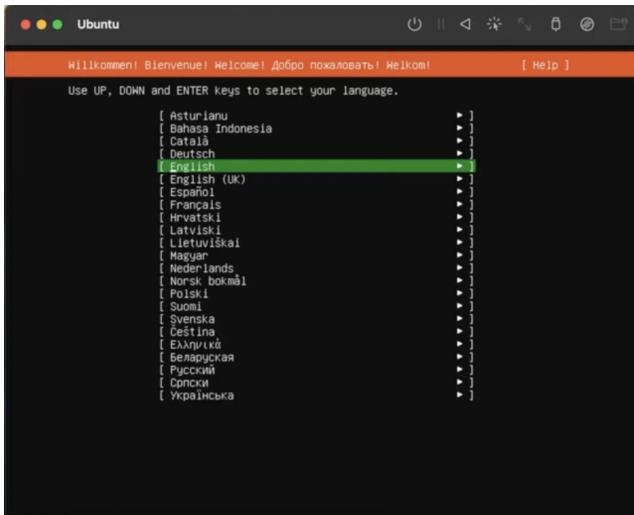
keywords EFI), then follow these troubleshooting steps here (this can also be caused by having the wrong ISO, so make sure your ISO is the 64 bit ARM Ubuntu server ISO):

Cannot boot into installer

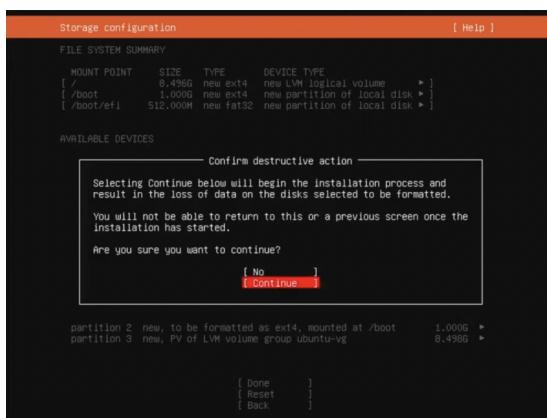
If you start the VM and are stuck at the EFI screen (BdsDxe: failed to load Boot0001 or UEFI Interactive Shell), try the following in order.

1. Make sure you have the installer ISO selected. Click the disk icon on the toolbar and check that there is a menu option for CD/DVD (ISO) Image (usb): ubuntu-xxx.iso. If it says CD/DVD (ISO) Image (usb): none, then highlight that menu and choose Change and then select the ISO. If you don't have any selectable menu option, follow the guide again and make sure you have added a removable drive. Then restart the VM.
2. Next, try to get into the EFI Shell. If you see UEFI Interactive Shell then you are already in the shell. Otherwise, restart the VM and quickly press the Esc key to enter the shell.
3. In the EFI shell make sure you see FS0: Alias(s):CD0h0a0a:BLK1: near the top or something similar. If not, then double check your configuration and make sure you have a removable drive configured and the installer ISO mounted. Also check that your ISO is valid.
4. Type in: fs0:\efi\boot\bootaa64.efi and you should see GRUB. Then select Ubuntu Server to continue with the install.

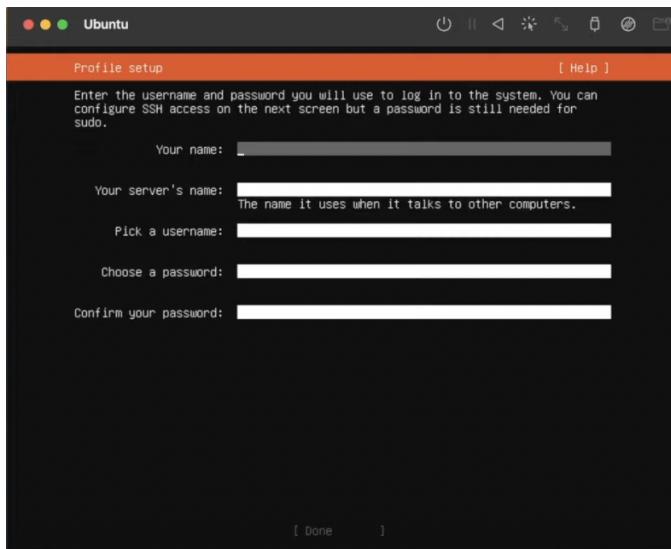
9. If you have successfully booted, you should come to a screen like this, and follow these steps:



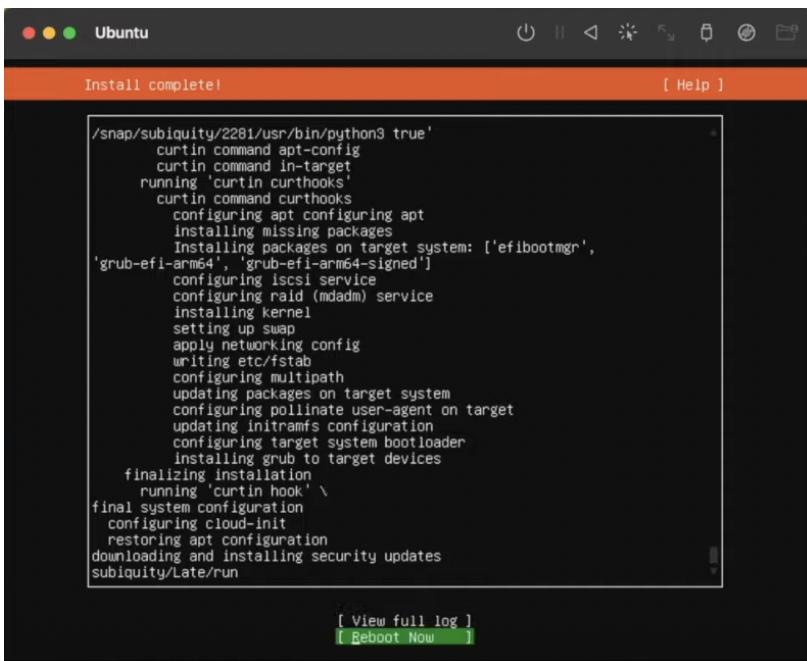
- a. Select your language by using the arrow keys and pressing ENTER
- b. Keep the default keyboard configuration
- c. Keep the default network connections (simply go to Done)
- d. For configure proxy, keep it blank (go to Done)
- e. For configure archive mirror, keep default
- f. For guided storage configuration, keep all defaults, arrow down to Done
- g. For storage configuration, go to Done
- h. You will then see a popup saying **Confirm destructive action**. This sounds scary but its not; select **continue**.



- i. You will then be prompted to select a name, server name, username, and password. I simply chose **cs2200** for all of these fields, but you can choose whatever you want.

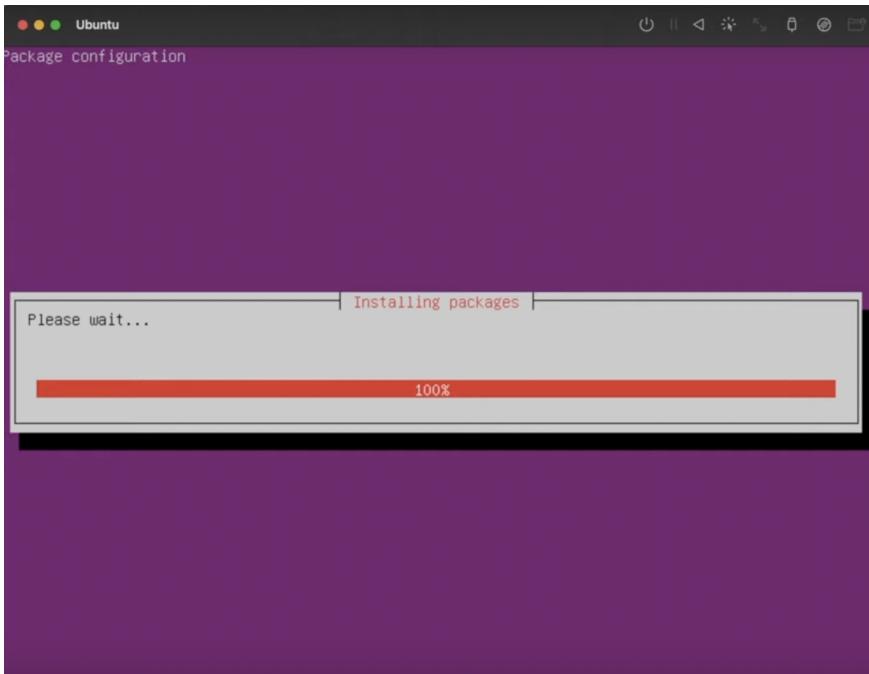


10. Now you will be taken to an installation log screen, wait until this is finished to continue, it might take around 5 minutes or so.
11. Once you see the **Install complete!** Header, select the **reboot now** option. Rebooting might take a while, and if you are stuck on a blank screen for more than 5 minutes, try manually closing out the VM and relaunching it.

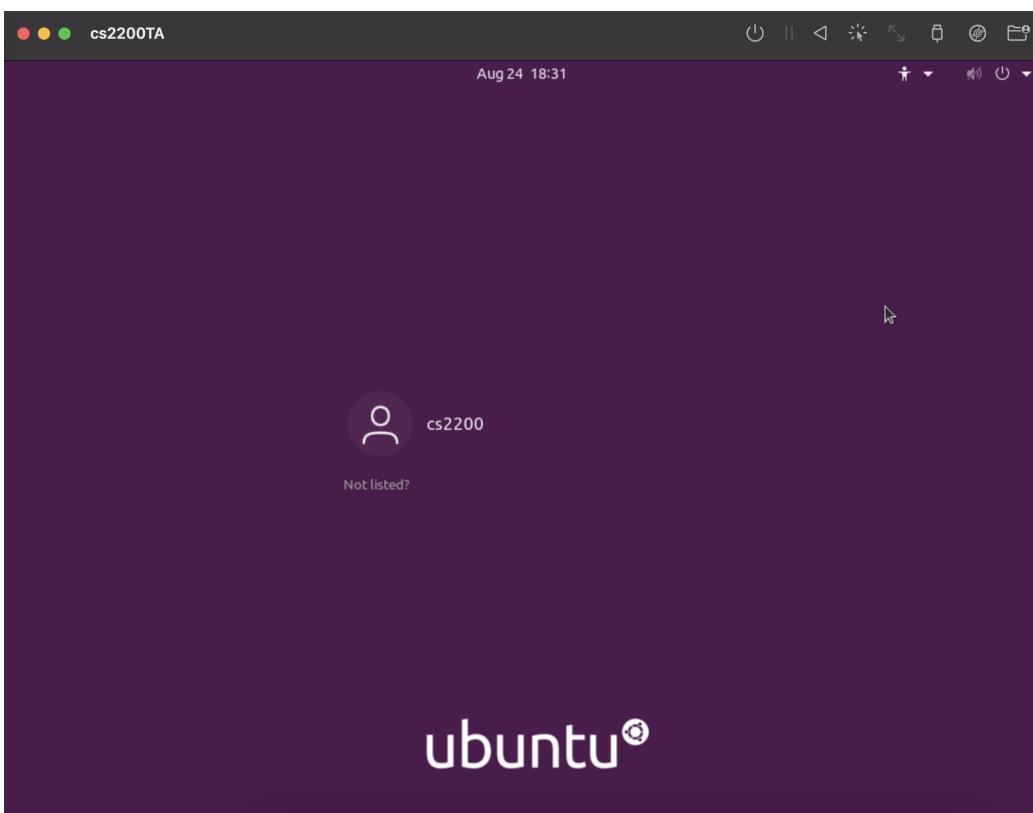


12. If the reboot succeeded, you will be taken to a terminal and asked to put in your username and password that you set earlier.
13. After you enter your username and password, you will now install a GUI. Type **sudo apt install tasksel** in your terminal.

14. Once this installation is complete, type **sudo tasksel install ubuntu-desktop**. Once you see this screen, the installation is complete, so now reboot your VM by typing **sudo reboot**.



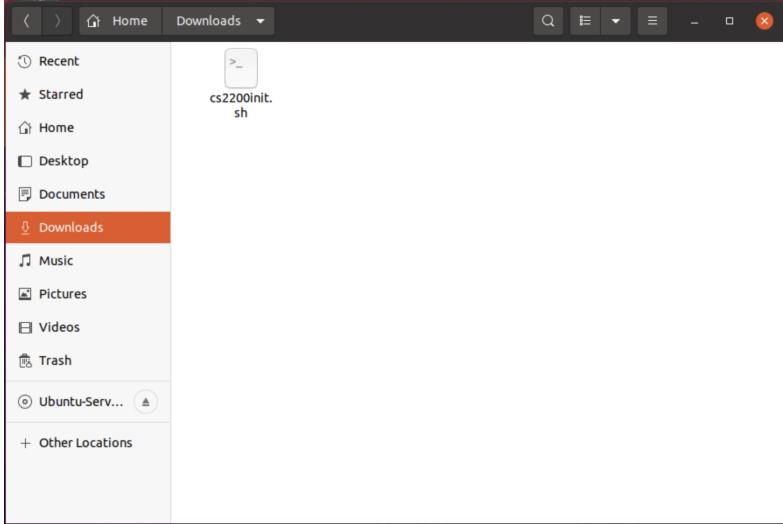
15. Once you reboot your VM, you should be taken to a GUI, with a login screen that looks like this:



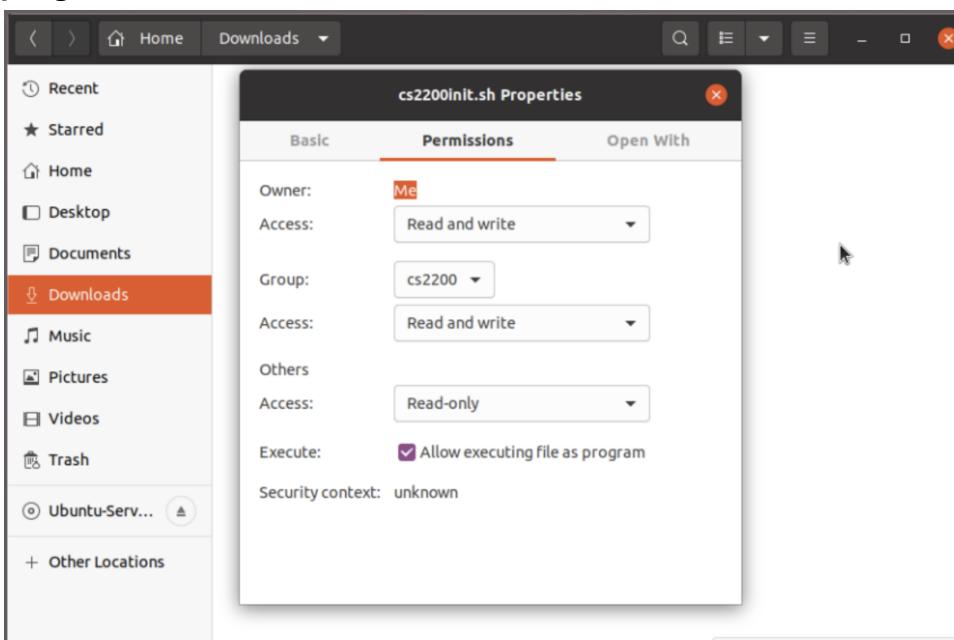
16. Login using the credentials that you specified earlier, and you should be taken to a setup screen. You can deny all of the location/data tracking.

Package installation

- Once you have completed the setup, you can access the Firefox browser within the VM and download the script **cs2200init.sh** through Canvas (Modules > VM Setup (P3,P4,P5)). Once this is done, the script should be in your **Downloads** folder.
Update: You will likely need to download the **run_circuitsim.sh** as well. Place it in the same directory as **cs2200init.sh**. (Not shown in image)

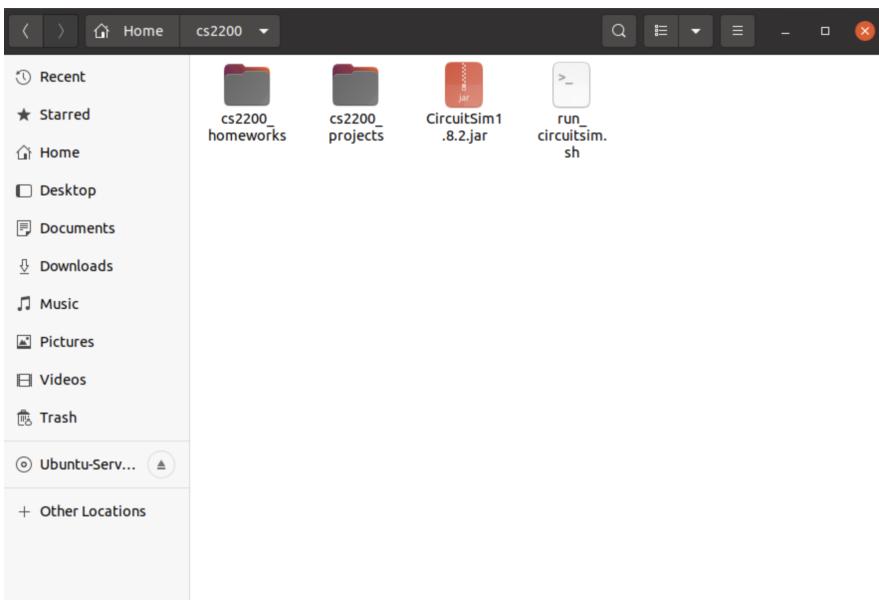


- Right click on this init file and make sure under **Permissions** that the **Allow executing file as program** box is checked.

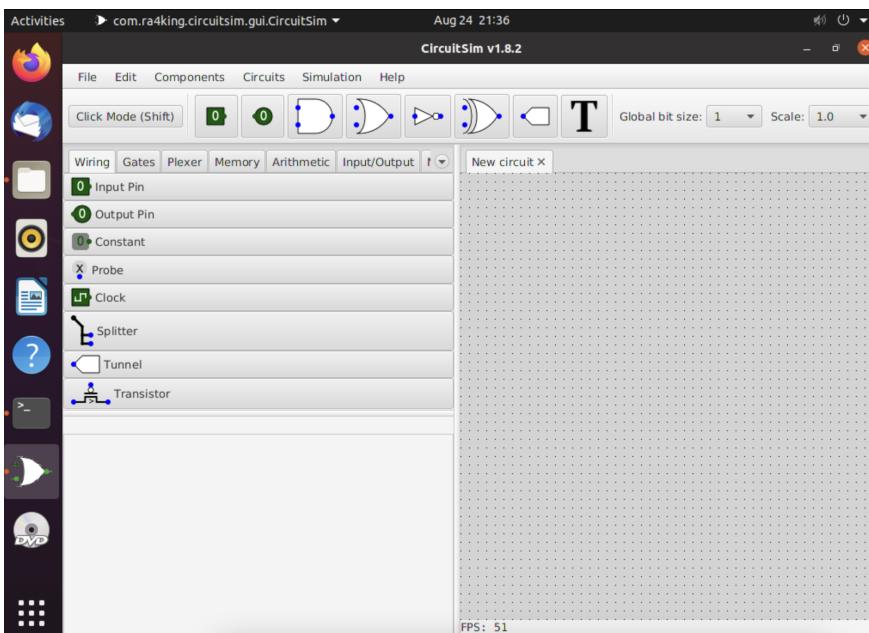


- Now, open the native terminal using the shortcut **control+option+t**, and change directories into the **Downloads** folder. Run the command **chmod +rx cs2200init.sh**
- Now run the script using **./cs2200init.sh** in the same terminal. This will take a few minutes to install everything, and it might ask you for the administrator password, which is the root user password you set up earlier. Once the download is complete, there should be a message that says so.
- This command should create a **cs2200** folder within your **Home** directory. Ensure that within your **Home/cs2200** folder, has the following files/folders:

- a. cs2200_homeworks
- b. cs2200_projects
- c. circuitSim1.8.2.jar
- d. run_circuitsim.sh



6. Now, to run circuitsim within your VM, simply open the terminal again (control+option+t), change directories into the **cs2200** folder, and run **./run_circuitsim.sh**. If you did the prior steps correctly, it should look like this!



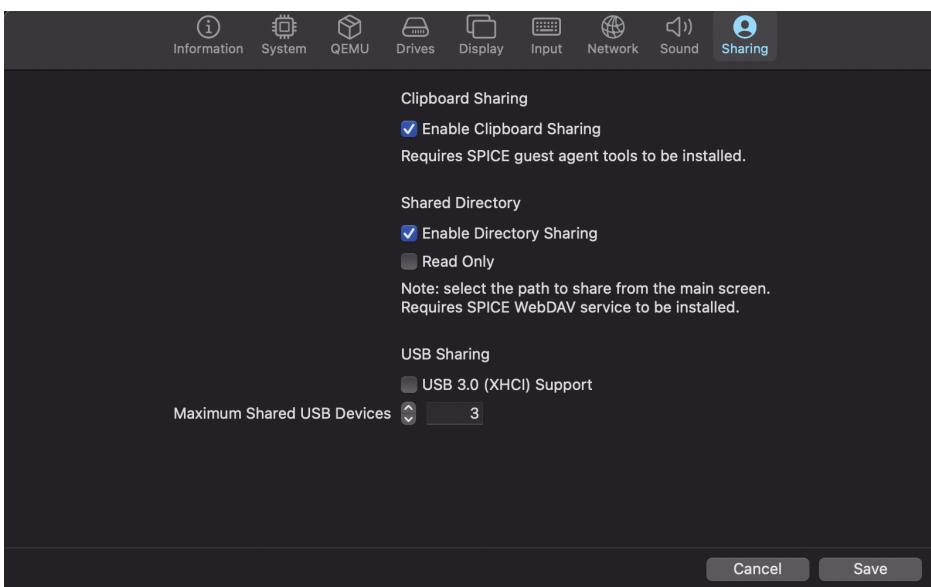
Setting Up a Shared Directory

If you would like to setup a shared directory between the VM and your physical computer (not mandatory, since you can login to Canvas on your VM and download the files that way), follow these steps:

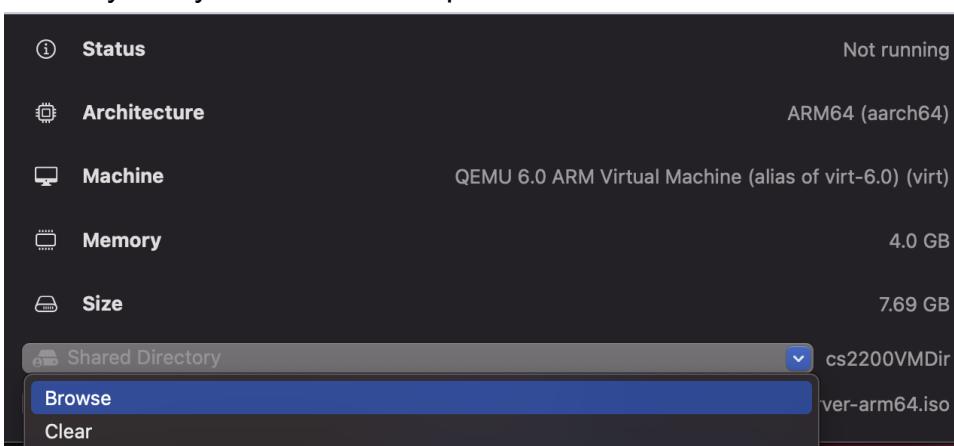
1. Before logging into your VM, create a folder on your local machine that you would like to designate as the shared folder. In this case, I named mine **cs2200VMDir** and added a test file **test.txt**.
2. Open up UTM and find your VM that you created earlier. Now click on the edit selected VM icon in the top right corner:



3. Click on the **Sharing** tab, and make sure the box next to **Enable Directory Sharing** is clicked, and click save.



4. Now, under your VM, there should be an option to select a shared directory, set it to the directory that you created in step 1.



5. Now, launch/login to your VM and open the terminal, and type the command **sudo apt install spice-vdagent spice-webdav**
6. Restart your VM to apply the changes.
7. Now, if you did all the prior steps properly, when you access the link <https://127.0.0.1:9843> in your VM browser, you should be taken to screen that looks like this:



- a. Notice that the test.txt file that I put on my local machine is visible here! And if I want to have the file on my VM, simply right click on the link to the file and download it.