
Introductory Lab

Introduction

This lab aims to show you how to set up a virtual machine (VM) on your computer, clone the labs' repository, and set up JupyterLab.

1. Virtual Machine (VM)

A Virtual machine is a software that emulates a computer system, thus allowing you to run multiple operating systems on a single computer.

- Start with downloading and installing [VirtualBox](https://www.virtualbox.org/wiki/Downloads)(<https://www.virtualbox.org/wiki/Downloads>) if you don't have it already. Choose the “platform packages” according to your operating system (If you do not know how to install it, see this link: <https://www.wikihow.com/Install-VirtualBox>).
- Download the VM image “ttm4200.ova”[Link](https://filesender.uninett.no/?s=download&token=3a3c3529-f947-4a7b-bc88-a7e5b5308126) (<https://filesender.uninett.no/?s=download&token=3a3c3529-f947-4a7b-bc88-a7e5b5308126>). This VM image contains the software configuration, including an operating system and other packages that we will use in this course.
- Open VirtualBox and select **Tools** → **Import**. In the “Appliance to import” window, set the **Source** to be **Local File System** and the **File** to be the path to the downloaded **ttm4200.ova**. Then → **Next** → **import**.

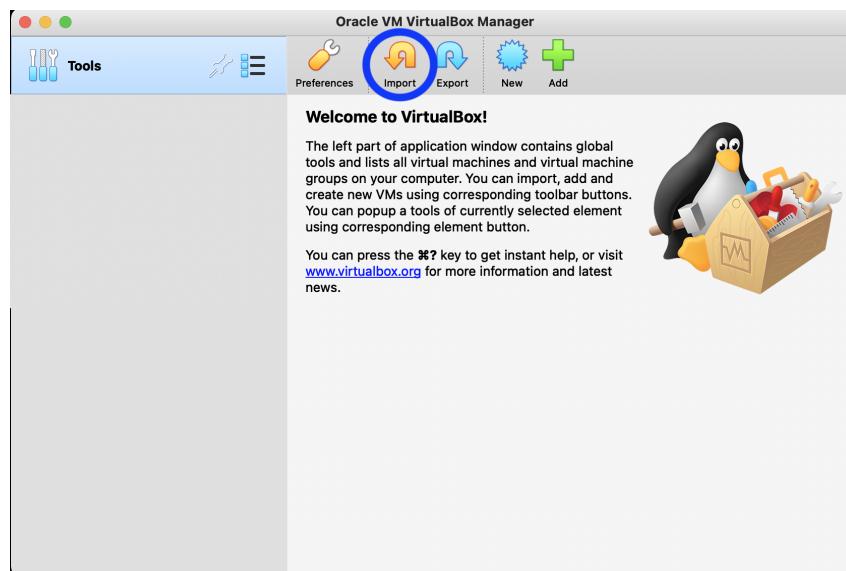


Figure 1: Importing VM image on VirtualBox.

Introductory Lab

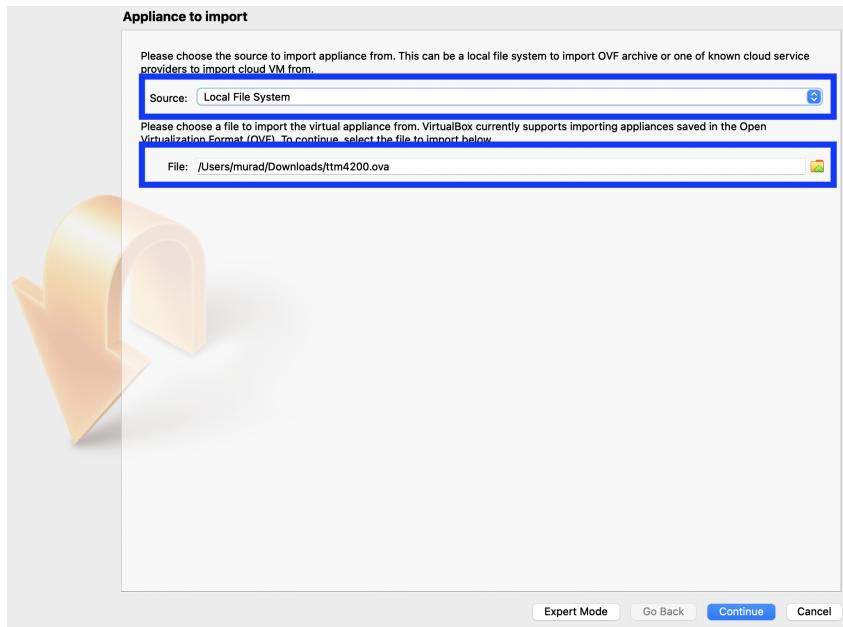


Figure 2: Selecting source OVA file.

- After the import is done, select **ttm4200** → **Settings** → **System**. Allocate enough “Base Memory” to the VM depending on your computer. There should be enough remaining for the host machine. For example, for a computer with 16 GB RAM, you can allocate 8 GB (8192 MB) to the VM.

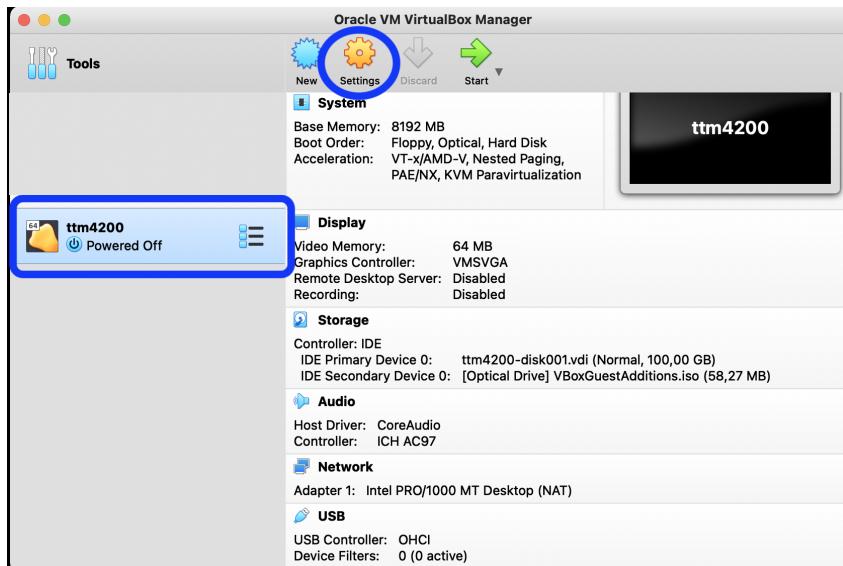


Figure 3: VM settings.

Introductory Lab

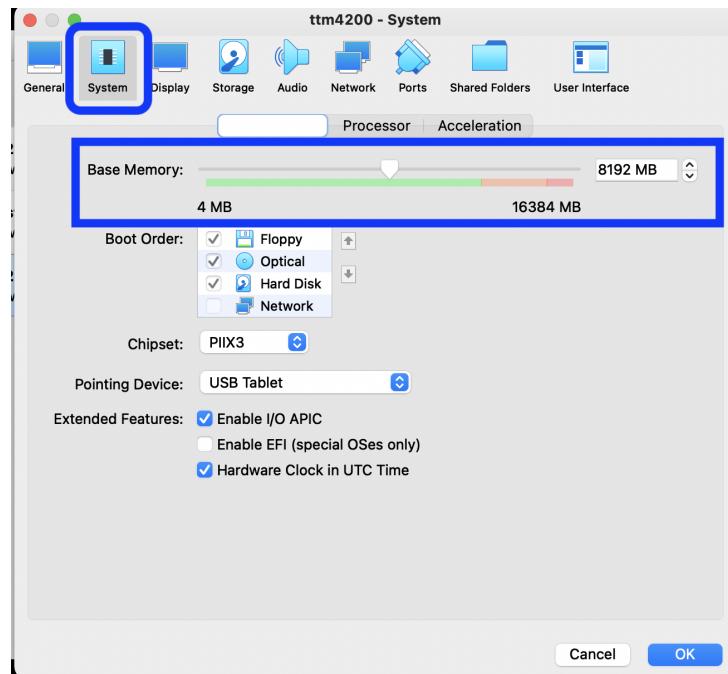


Figure 4: Allocating RAM.

- To access files on your computer from within the VM, you can share folders between them. Select **Shared Folders** → **Add new shared folder** (small icon on the right). Set **Folder Path** to a folder you want to share in your computer and set **Folder Name** to be “shared_folder,” then select **Auto-mount**.

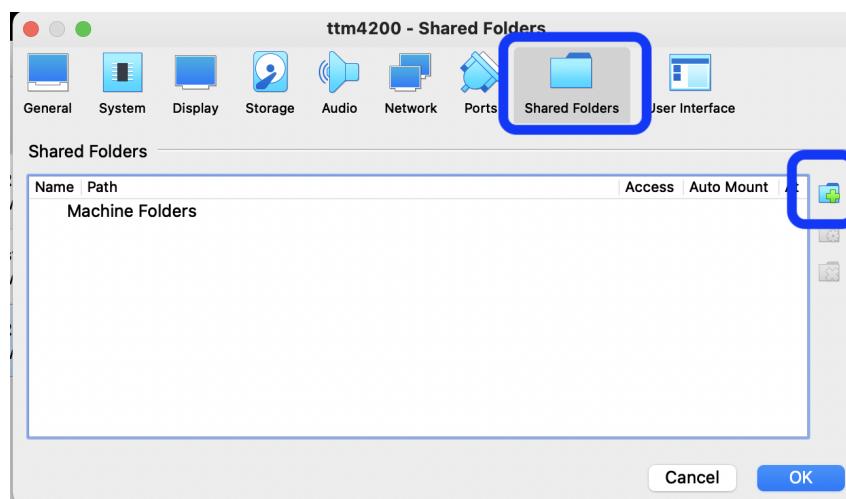


Figure 5: Setting a shared folder.

Introductory Lab

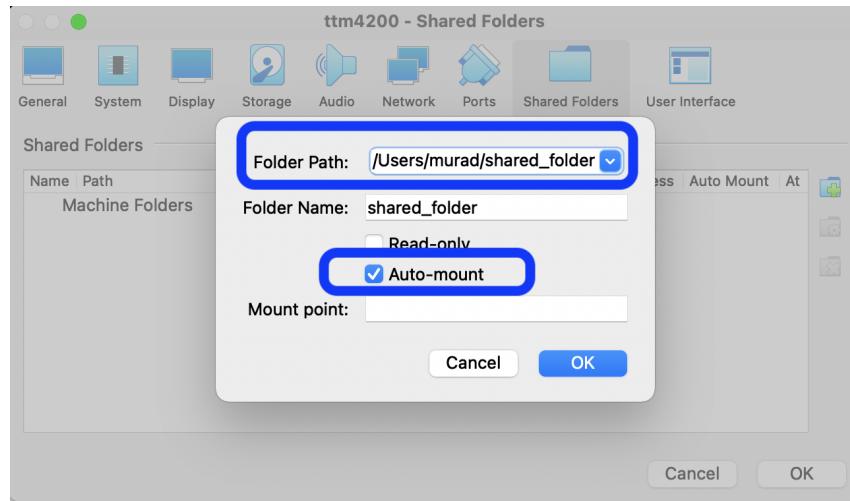


Figure 6: Auto-mounting the shared folder.

- Start the VM and log in using the password “ttm4200”. This VM uses [Lubuntu](#), a lightweight Linux operating system.
- Change the password using [passwd](#).
- By default, the shared folder will be mounted to “/media/sf_shared_folder/” (if Auto-mount is selected). You can create a symbolic link to the home directory for easier access.

```
ln -s /media/sf_shared_folder/ shared_folder
```

- We will heavily use Linux commands throughout this course, so you must get familiar with basic commands. For example, read this tutorial: <https://ubuntu.com/tutorials/command-line-for-beginners#1-overview> or any other resources online.

1.1 Course Repository

We will use a Github repository for all labs materials during this course.

- Clone the labs’ repository from Github:

```
git clone https://github.com/ntnuttm4200/labs.git /home/ttm4200/labs
```

A new directory called “labs” will be created in your home directory.

- We will update the course repository frequently, so you should fetch the new updates at the beginning of each lab.

```
cd ~/labs
git pull origin main
```

Introductory Lab

1.2 JupyterLab

We will use JupyterLab notebooks for lab tasks throughout this course. A Notebook document will contain instruction text and a python code to check your progress.

- To start JupyterLab inside the folder “lab0”:

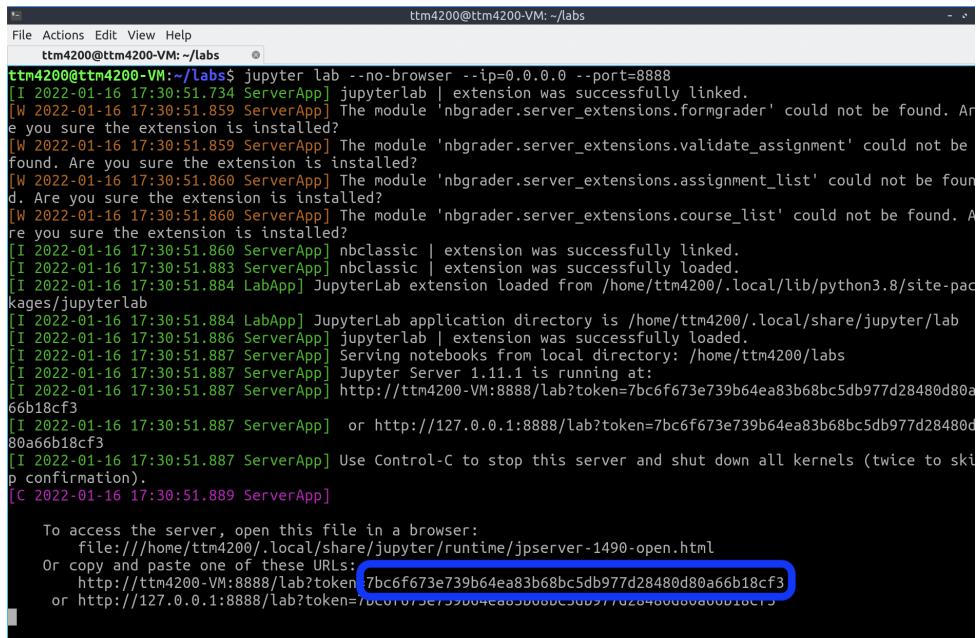
```
cd ~/labs/lab0
jupyter lab
```

This will automatically open JupyterLab in a browser in your VM.

- A more convenient way is to access the JupyterLab from your computer browser by running the following in your VM:

```
cd ~/labs/lab0
jupyter lab --no-browser --ip=0.0.0.0 --port=8888
```

Then in your computer browser, type: 127.0.0.1:8888. It will ask you for the token, which you can copy from the terminal output in your VM.

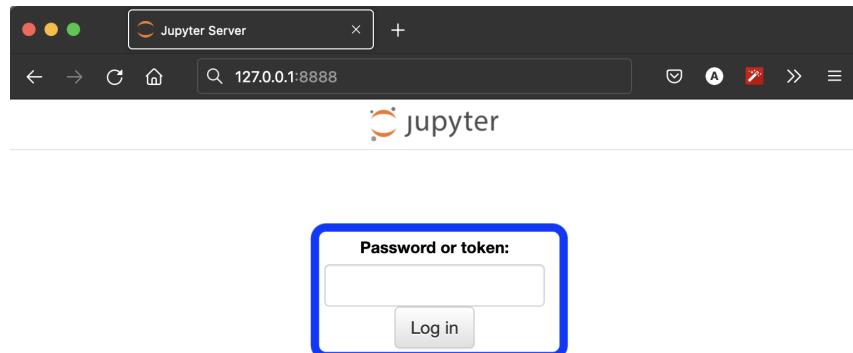


```
ttm4200@ttm4200-VM:~/labs$ jupyter lab --no-browser --ip=0.0.0.0 --port=8888
[I 2022-01-16 17:30:51.734 ServerApp] jupyterlab | extension was successfully linked.
[W 2022-01-16 17:30:51.859 ServerApp] The module 'nbgrader.server_extensions.formgrader' could not be found. Are you sure the extension is installed?
[W 2022-01-16 17:30:51.859 ServerApp] The module 'nbgrader.server_extensions.validate_assignment' could not be found. Are you sure the extension is installed?
[W 2022-01-16 17:30:51.860 ServerApp] The module 'nbgrader.server_extensions.assignment_list' could not be found. Are you sure the extension is installed?
[W 2022-01-16 17:30:51.860 ServerApp] The module 'nbgrader.server_extensions.course_list' could not be found. Are you sure the extension is installed?
[I 2022-01-16 17:30:51.860 ServerApp] nbclassic | extension was successfully linked.
[I 2022-01-16 17:30:51.883 ServerApp] nbclassic | extension was successfully loaded.
[I 2022-01-16 17:30:51.884 LabApp] JupyterLab extension loaded from /home/ttm4200/.local/lib/python3.8/site-packages/jupyterlab
[I 2022-01-16 17:30:51.884 LabApp] JupyterLab application directory is /home/ttm4200/.local/share/jupyter/lab
[I 2022-01-16 17:30:51.886 ServerApp] jupyterlab | extension was successfully loaded.
[I 2022-01-16 17:30:51.887 ServerApp] Serving notebooks from local directory: /home/ttm4200/labs
[I 2022-01-16 17:30:51.887 ServerApp] Jupyter Server 1.11.1 is running at:
[I 2022-01-16 17:30:51.887 ServerApp] http://ttm4200-VM:8888/lab?token=7bc6f673e739b64ea83b68bc5db977d28480d80a66b18cf3
[I 2022-01-16 17:30:51.887 ServerApp] or http://127.0.0.1:8888/lab?token=7bc6f673e739b64ea83b68bc5db977d28480d80a66b18cf3
[I 2022-01-16 17:30:51.887 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 2022-01-16 17:30:51.889 ServerApp]

To access the server, open this file in a browser:
  file:///home/ttm4200/.local/share/jupyter/runtime/jpserver-1490-open.html
Or copy and paste one of these URLs:
  http://ttm4200-VM:8888/lab?token=7bc6f673e739b64ea83b68bc5db977d28480d80a66b18cf3
  or http://127.0.0.1:8888/lab?token=7bc6f673e739b64ea83b68bc5db977d28480d80a66b18cf3
```

Figure 7: Authentication token in JupyterLab Server.

Introductory Lab



Token authentication is enabled

Figure 8: Adding authentication token into JupyterLab client.

If you are wondering how does this work? We set up a port forwarding on the VM image, which tells the VM to forward any traffic on port 8888 to the host machine (on port 8888). You can see this setting as shown in the following figures

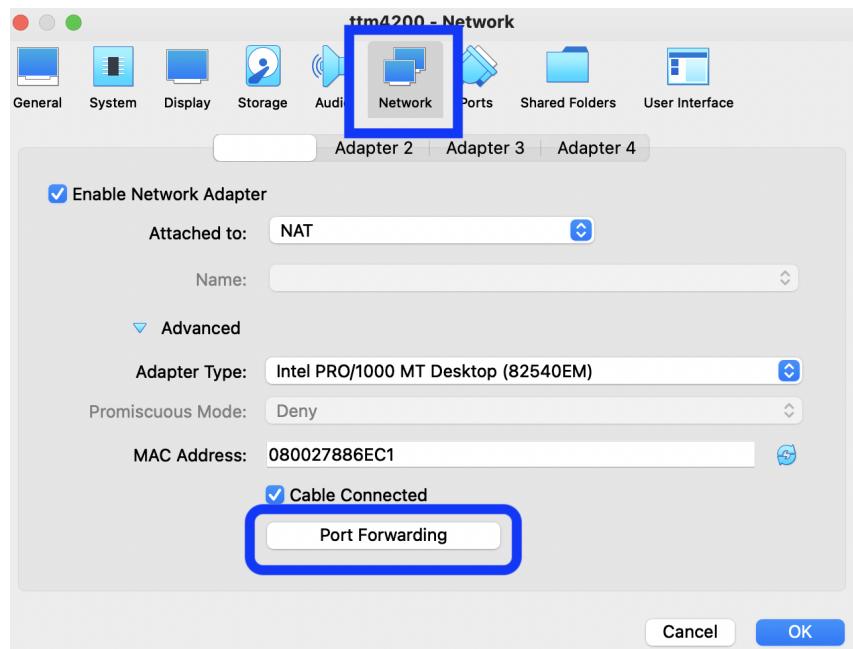


Figure 9: Port forwarding on VM.

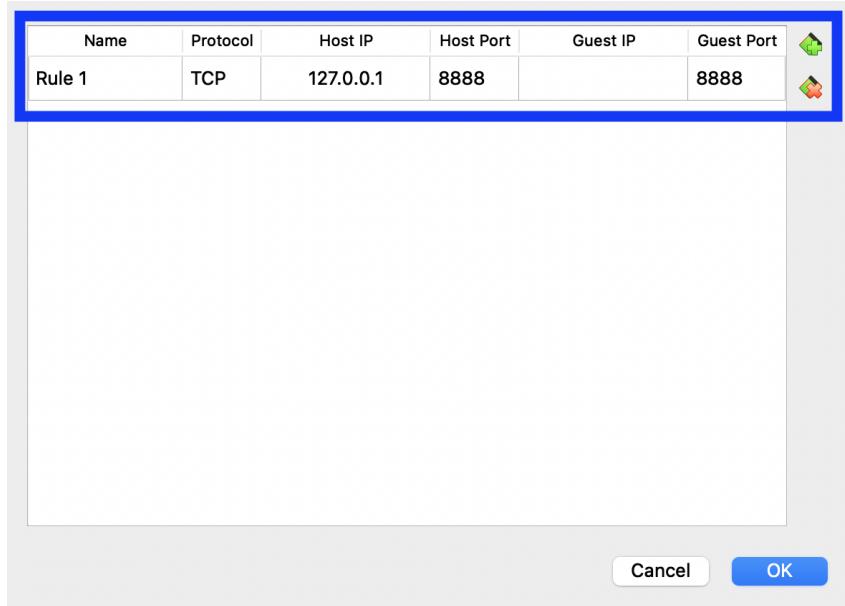


Figure 10: Port forwarding on VM.

- Try to get familiar with the JupyterLab interface (<https://jupyterlab.readthedocs.io/en/stable/user/interface.html>). Open “tasks.ipynb” inside “lab1”, it will contains instruction for lab1. From now on, you will get only JupyterLab notebooks for lab instructions (not a pdf).

2. Virtual Machine on NTNU Cloud (Optional)

If you are having trouble setting up the VM on your computer, you can use a VM hosted on the NTNU OpenStack cloud. We have created a single VM for each team. These VMs have private IP addresses, thus only accessible via NTNU internal network. If you are not using the campus network, you need to connect to the NTNU network using VPN (Instruction are available at: <https://i.ntnu.no/wiki-/wiki/English/Install+vpn/>).

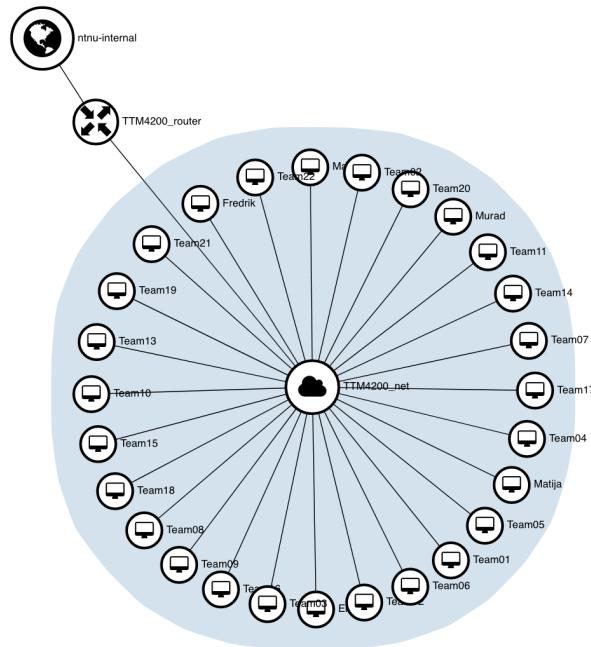


Figure 11: TTM4200 VMs on NTNU cloud.

We will post the IP and the password of your VM in your “Group Description” on Blackboard.

2.1 Accessing a remote VM through SSH

- You can access the remote VM through SSH by running the following command in a terminal:

```
ssh ttm4200@your_VM_IP_address .
```

In Windows, you need to install an SSH client, for example [PuTTY](#) or OpenSSH Client

- Use `scp` to copy files and directories to/from the remote VM. For example:

- To copy a file to the remote VM: `scp local_file ttm4200@your_VM_IP_address:~/ .`
- To copy file from the remote VM: `scp ttm4200@your_VM_IP_address:~/path_to_remote_file .`
- To copy directories add the `-r` flag: `scp -r local_directory ttm4200@your_VM_IP_address:~/ .`

2.2 Running JupyterLab on a remote VM

- On the remote VM, run JupyterLab on localhost and on a specific port (e.g., 9999):

```
jupyter lab --no-browser --ip=localhost --port=9999
```

Introductory Lab

- Establish an SSH tunnel from the port you run the JupyterLab (in the remote VM) to a port in your computer:

```
sss ttm4200@your_VM_IP_address -L 9999:localhost:9999
```

Then in your computer browser, type `127.0.0.1:9999` to open the JupyterLab.

In Windows, you can use PuTTY to establish the SSH Tunnel (you can see this [tutorial](#)).

You need to keep the SSH tunnel open; otherwise, you will lose connection if you close the terminal with the SSH tunnel.

2.3 Remote Access to a Graphical User Interface

Sometimes you need to use applications with GUI (e.g., Wireshark); thus, we added two options to have remote access to desktops in your VM:

2.3.1 Xpra

One advantage of using Xpra is that it supports HTML5 clients; thus, you can have a desktop screen of the remote VM on your browser. You only need to establish an SSH tunnel from your computer to the VM.

- On the remote VM, start the Xpra server and forward its traffic to a port on localhost (e.g., 7033):

```
xpra start --bind-tcp=localhost:7033 --start=xterm
```

- Establish an SSH tunnel from the port on the remote VM to a port in your computer:

```
ssh ttm4200@your_VM_IP_address -L 7033:localhost:7033
```

Then in your computer browser, type `127.0.0.1:7033` to show xterm. You can start any application from the command line, as shown in the following figure:

Introductory Lab

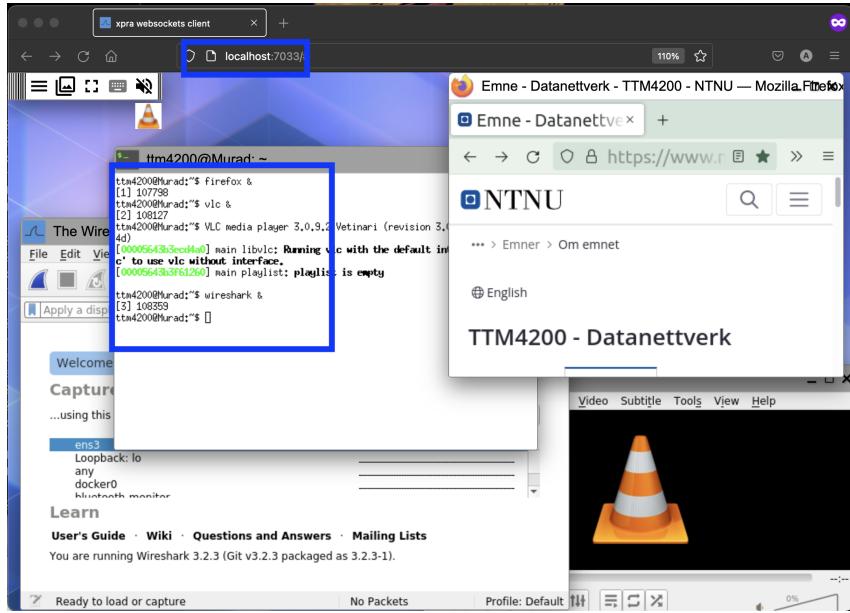


Figure 12: Xpra client using HTML5.

- Xpra is a persistent remote display server; thus, you will see the same windows if you close your browser and open it again. After you are done, stop the Xpra server: `xpra stop`.

2.3.2 X2Go

X2Go allows you to have a remote connection to a graphical desktop session, but it can be slow depending on your Internet bandwidth.

- Install the X2Go client on your computer. Installing instructions are available at: <https://wiki.x2go.org/doku.php/doc:installation:x2goclient/>.
- Launch the X2Go client and create a new session:



Figure 13: Creating a new session in the X2Go client.

- Fill in “Host” with your remote VM IP address. The “Login” is the username, which is “ttm4200”. Select “XFCE” in the session type.

Introductory Lab

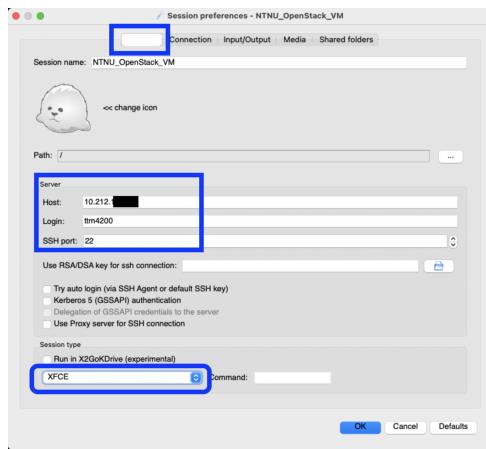


Figure 14: Details of an X2Go session.

- Disable sound and printing support:

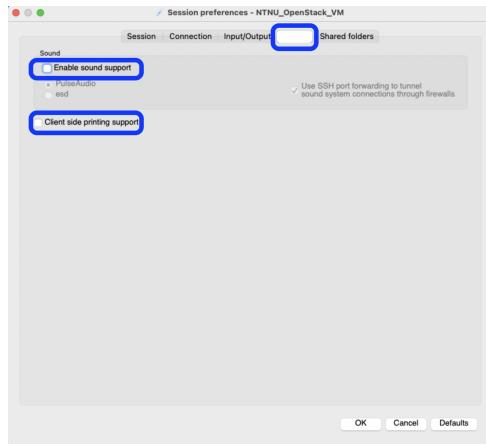


Figure 15: Disabling sound and printing support.

- For convenient file exchange between your computer and the remote VM, you can share a folder between them. The shared folder will be mounted at (`/tmp/.x2go-ttm4200/media/disk/...`).

Introductory Lab

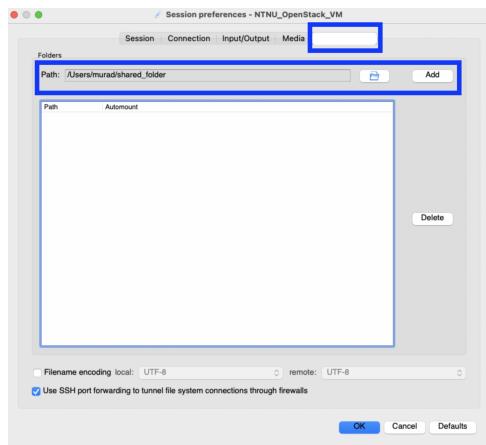


Figure 16: Sharing folders between the local machine and the remote VM.

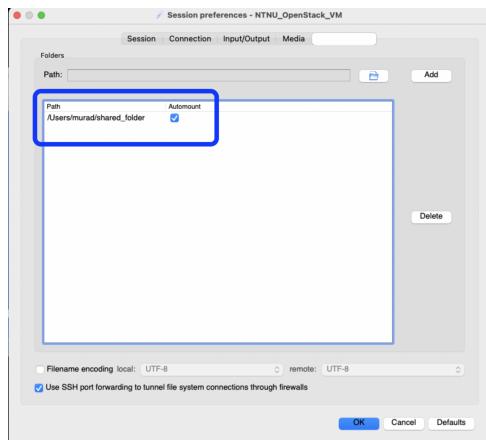


Figure 17: Automount shared folder.

- After completing the setup, click on the newly created session and input the password of the remote VM:

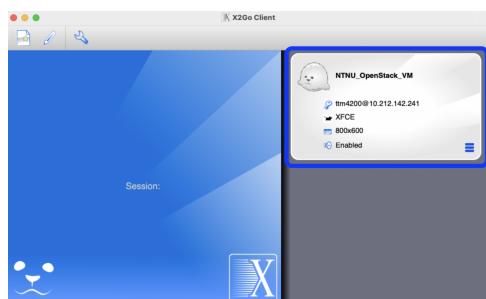


Figure 18: Starting the new session in the X2Go client.

Introductory Lab

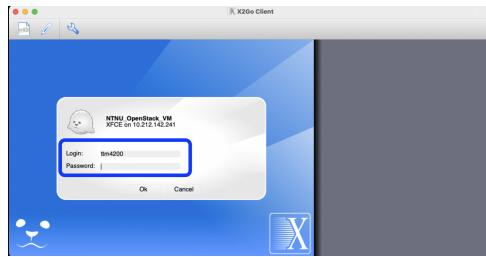


Figure 19: Starting the new session in the X2Go client.

- Now, you are able to interact with the remote VM in the same way as if you would set it up locally:

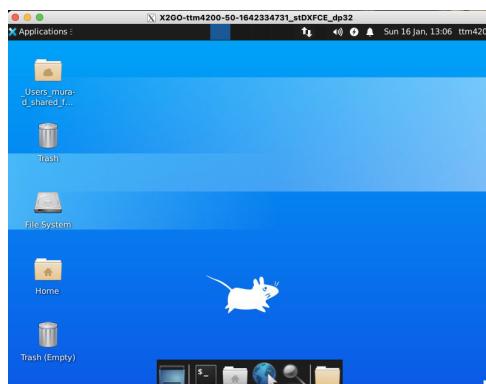


Figure 20: Remote graphical desktop using X2Go.

2.4 Adding an SSH key to the remote VM (Optional)

- Instead of using the password every time you connect to the remote VM, you can use an SSH key:
 - Generate an SSH key in your computer: `ssh-keygen -t rsa`
 - Copy the public key to the remote VM:

```
scp ~/.ssh/id_rsa.pub ttm4200@your_VM_IP_address:~/ssh/
```

- In the remote VM, add the public key to the authorized keys:

```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

- Restart the SSH server: `sudo systemctl restart ssh`

3. Lastly but Most Importantly: Backup your Data

Always backup your progress to your own computer. The local VM can fail due software upgrade. The remote VM can become unavailable or reset due to crashes, misconfiguration, or other issues.