
Introductory Lab

Introduction

This lab introduces you to the tools you will use throughout the course. It consists of three parts:

1. **Virtual Machine.** This part will introduce you to the remote virtual machine, how to connect to your virtual machine, how to use remote desktop clients, and how to run JupyterLab on the virtual machine. You will also learn to use SSH/SCP and clone the course repository.
2. **Introduction to Linux.** This part will introduce you to the basic concepts of Linux and the command line interface. You will learn to navigate the file system and use the essential commands.
3. **Optional.** This part is optional, and you can skip it. This part will help you set up a local virtual machine on your computer and introduce you to another option to have a remote desktop client.

Milestone 1 – Virtual Machine

In this course, you will use a virtual machine that runs a distribution of Linux called Ubuntu. You will use a virtual machine hosted on the NTNU server infrastructure. These virtual machines are available 24/7 and can be accessed via NTNU's internal network. If you are not on campus, you can connect to the virtual machine using the [NTNU VPN](#).

1.1 Connecting to the Virtual Machine

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

```
ssh ttm4200@your_VM_IP_address .
```

You can find your VM IP address and password in “Group Description on the Blackboard”. If you have Windows, you need to install an SSH client, for example [PuTTY](#) or OpenSSH Client.

SSH Key

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 ed25519 -f ~/.ssh/<key_name> -C "<your_email>"
```

Replace `<key_name>` with a name for the key, and `<your_email>` with your email address. In all the following commands, you need to replace what is inside `<...>` with your own values.

Introductory Lab

You will be asked to enter a passphrase to protect the key. Leave it empty for now. This will generate two files: `~/.ssh/<key_name>` and `~/.ssh/<key_name>.pub`. The first file is the private key, and the second file is the public key.

- Copy the public key to the remote VM, and add it to the authorized keys file:

```
# In your computer
scp ~/.ssh/<key_name>.pub ttm4200@<your_VM_IP_address>:~/ssh/
# In the remote VM
cat ~/.ssh/<key_name>.pub >> ~/.ssh/authorized_keys
```

Alternatively, you can use the following single command in your computer:

```
cat ~/.ssh/<key_name>.pub | ssh ttm4200@<your_VM_IP_address> 'cat >> ~/.ssh/authorized_keys'
```

- Connect to the remote VM using the private key:

```
ssh -i ~/.ssh/<key_name> ttm4200@<your_VM_IP_address>
```

SSH Config

Instead of typing the username and the IP address every time you connect to the remote VM, you can add the following lines to your `~/.ssh/config` file:

```
Host ntnu_vm
  HostName <your_VM_IP_address>
  User ttm4200
  IdentityFile ~/.ssh/<key_name>
```

If you have not used SSH before, you may not have a `~/.ssh/config` file. You can create it by running `touch ~/.ssh/config`.

Then, you can connect to the remote VM by running `ssh ntnu_vm`.

Copying Files to/from the Virtual Machine

You can 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> ntnu_vm:~/<destination_directory>`.
- To copy file from the remote VM: `scp ntnu_vm:~/<source_file> <local_directory>`.
- To copy directories add the `-r` flag: `scp -r <local_directory> ntnu_vm:~/<destination_directory>`.

Task: show your TA that you can connect to the remote VM using `ssh ntnu_vm`

Task: show your TA that you can copy a file to/from the remote VM using `scp`

Task: show your TA that you can copy a directory to/from the remote VM using `scp`

1.2 Running JupyterLab on a remote VM

JupyterLab is a web-based interactive development environment for Jupyter notebooks, code, and data. You can run JupyterLab on the remote VM and access it from your computer. To do so, you need to establish an [SSH tunnel](#) from your computer to the remote VM.

- On your computer, run the following command:

```
ssh ntnu_vm -L 8888:localhost:8888
```

This will establish an SSH tunnel from port 8888 in your computer to port 8888 in the remote VM. The first port number is the port in your computer, and the second port is the port in the remote VM. You can use any port number above 1024 and below 65535. You can also use different port numbers for the two ports, but using the same port number is easier.

- In the **remote VM**, start JupyterLab:

```
jupyter lab --no-browser --port=8888
```

The port number should be the same as the second port number in the SSH tunnel.

- In your computer browser, type `localhost:8888` to show JupyterLab. You can now use JupyterLab as you would do on your computer, but it is actually running on the remote VM.

The port number should be the same as the first port number in the SSH tunnel. If JupyterLab asks you for a token, you can find it in the terminal where you started JupyterLab in the remote VM.

1.3 Remote Desktop Client

Some of the tools you will use in this course require a graphical user interface, such as Wireshark. You can use the remote VM with a graphical user interface by connecting to it using a remote desktop client. Several options are available, but we recommend using [Xpra](#).

Xpra allows you to remotely connect to a graphical desktop session without installing any software on your computer. It is the easiest option because it uses HTML5 and does not require any configuration. It is also faster than X2Go or VNC.

- Open a terminal in your computer and establish an ssh tunnel to VM:

Introductory Lab

```
ssh ntnu_vm -L 7777:localhost:7777
```

- On the remote VM, run the following command to start Xpra:

```
xpra start --bind-tcp=0.0.0.0:7777 --html=on --start=gnome-terminal
```

- Open a browser on your computer and type `localhost:7777` to show the graphical desktop session. You can now use the graphical desktop session as you would on your computer, but it is actually running on the remote VM. You can open any application that you want from the terminal in the browser, for example:

```
wireshark &
```

1.4 Course Repository

We will use a [Git](#) repository to distribute the course material throughout the semester.

- Clone the course repository to your **VM** using the following command:

```
git clone https://github.com/ntnuttm4200/labs_2023.git /home/ttm4200/labs
```

This will create a folder named “labs” in your home directory. You can find the course material in this folder.

- To get the latest version of the course material, run the following command:

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

You need to run this command at the beginning of each lab.

Milestone 2 – Introduction to Linux

- Run JupyterLab in the remote VM and open the notebook `~/labs/00-lab0/tasks.ipynb`. Follow the instructions in the notebook to complete the tasks. We will use JupyterLab for the rest of the course.

Milestone 5 – Optional

5.1 Local Virtual Machine (Optional)

You can also use a local virtual machine on your own computer. This is optional, and we will **not** provide support for it.

- 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.sikt.no/?s=download&token=62a568a1-c841-40be-9ff1-a390cdc4e49c) (<https://filesender.sikt.no/?s=download&token=62a568a1-c841-40be-9ff1-a390cdc4e49c>). This VM image contains the software configuration, including an operating system and other packages that we will use in this course. This is the same image that we use in the remote VM.
- 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**.

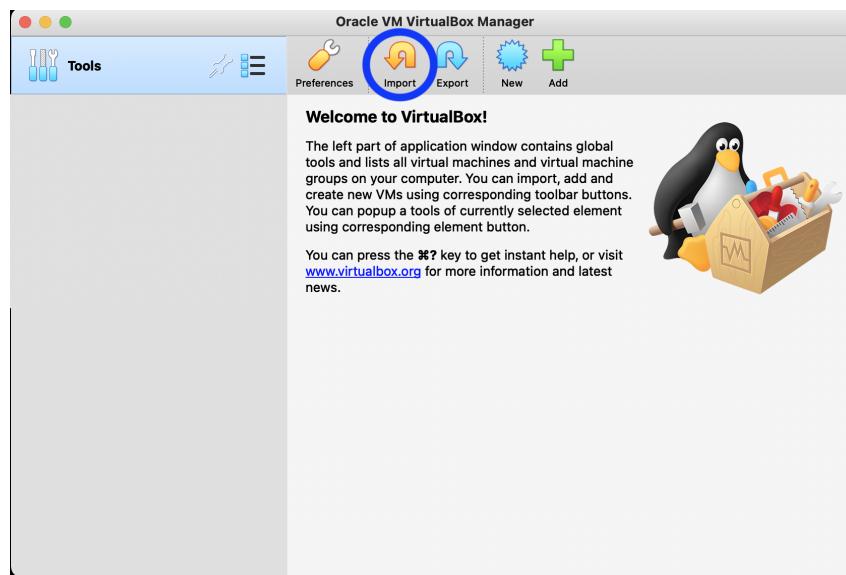


Figure 1: Importing VM image on VirtualBox.

Introductory Lab

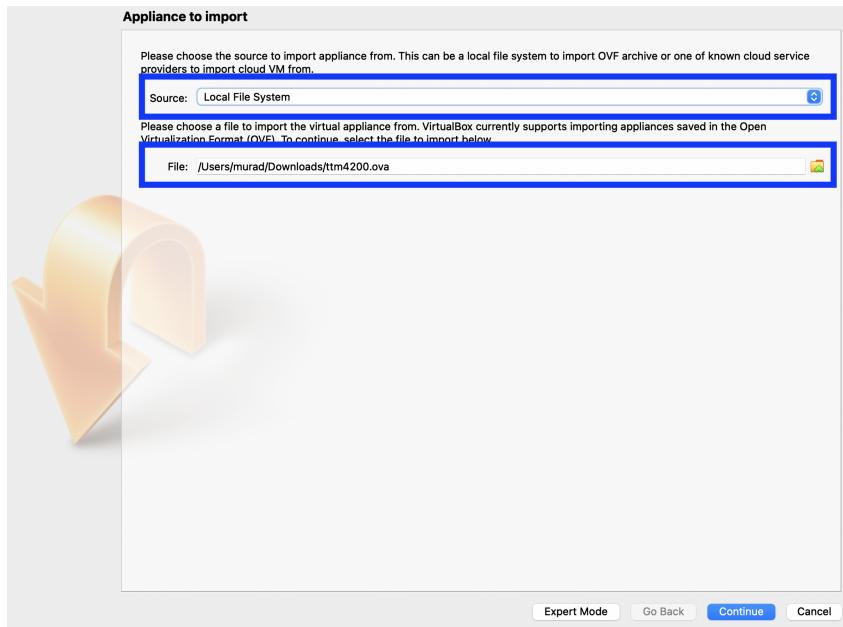


Figure 2: Selecting source OVA file.

- 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.

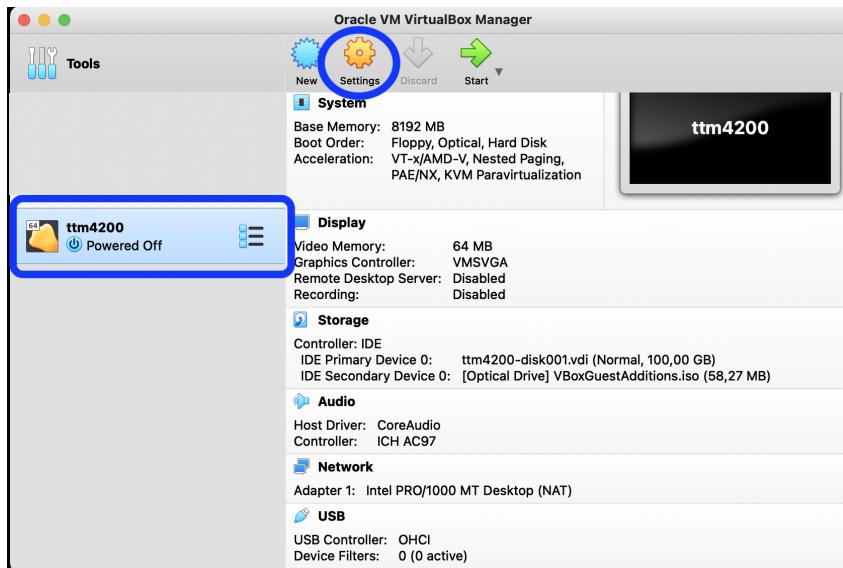


Figure 3: VM settings.

Introductory Lab

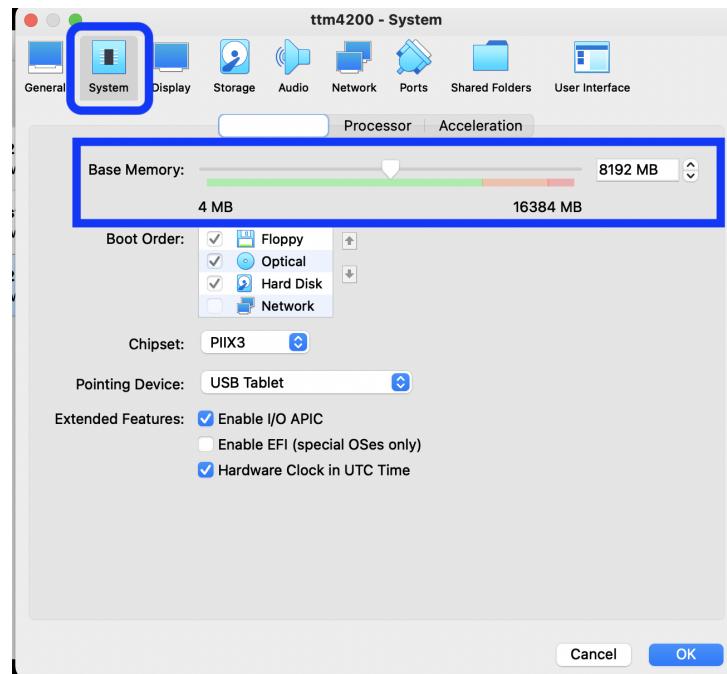


Figure 4: Allocating RAM.

- 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**.

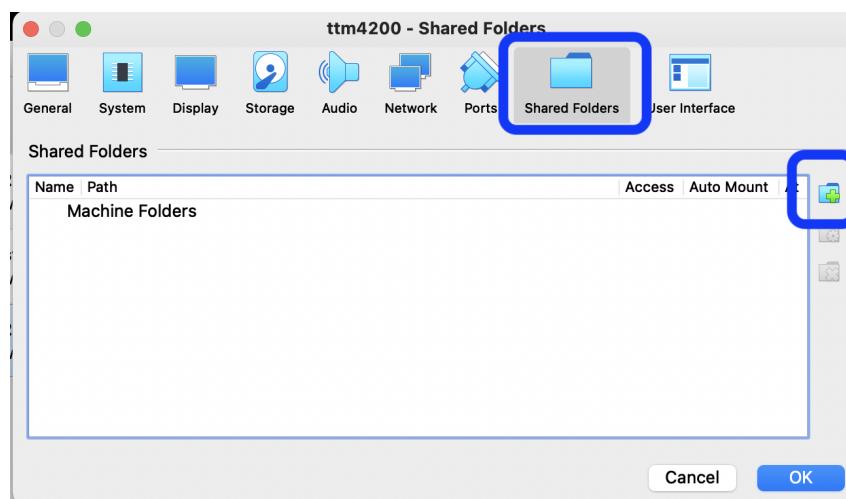


Figure 5: Setting a shared folder.

Introductory Lab

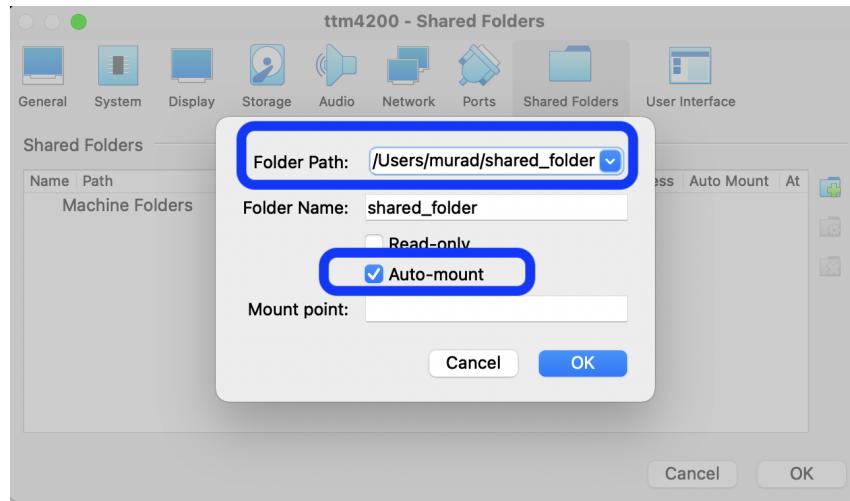


Figure 6: Auto-mounting the shared folder.

- Start the VM and log in using the password “ttm4200”.
- 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
```

5.2 X2Go (Optional)

Another option to access the remote VM is to use 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 7: Creating a new session in the X2Go client.

Introductory Lab

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

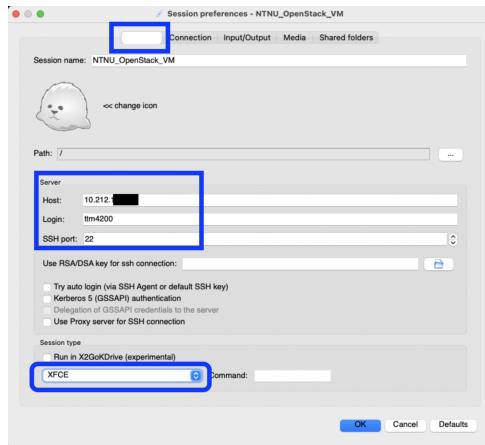


Figure 8: Details of an X2Go session.

- Disable sound and printing support:

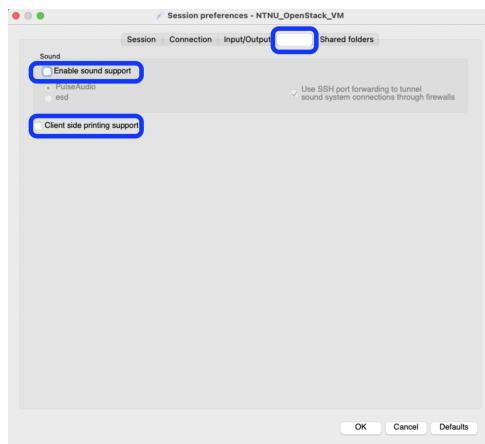


Figure 9: Disabling sound and printing support.

- 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

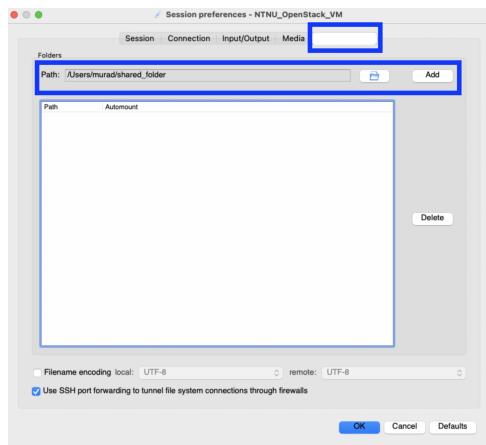


Figure 10: Sharing folders between the local machine and the remote VM.

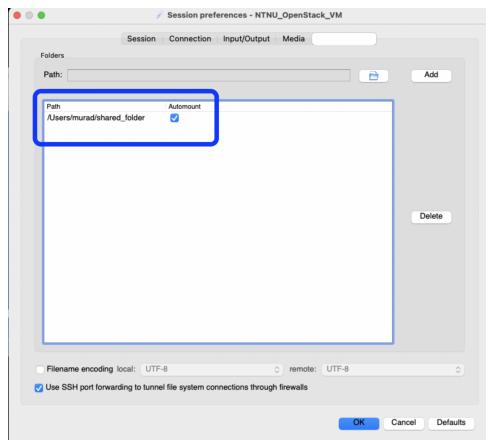


Figure 11: Automount shared folder.

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

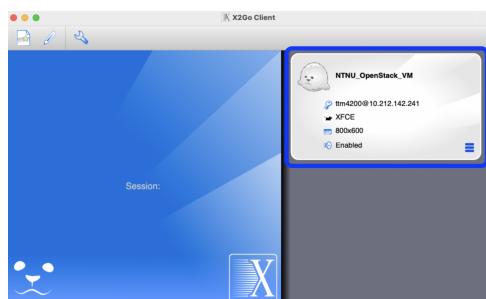


Figure 12: Starting the new session in the X2Go client.

Introductory Lab

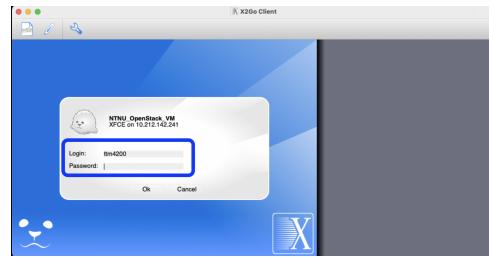


Figure 13: Starting the new session in the X2Go client.

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

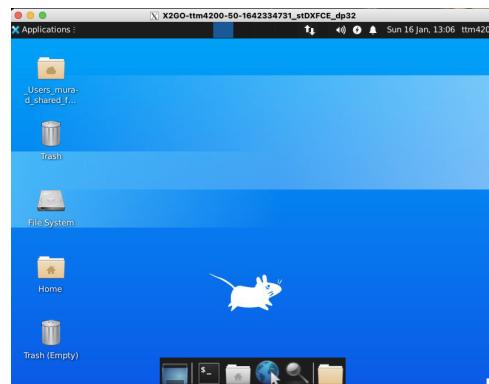


Figure 14: Remote graphical desktop using X2Go.