CSE321 Lab Ubuntu Setup for xv6

Using a Virtual Machine (Recommended)

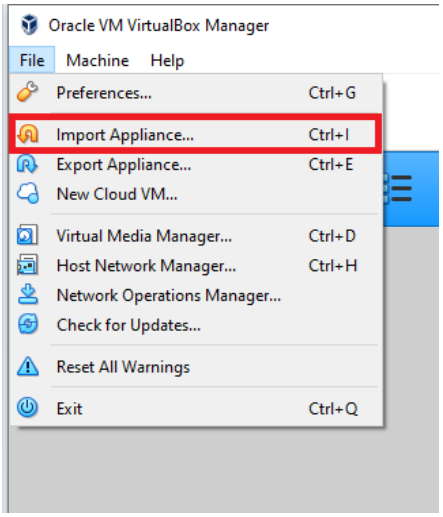
Machine Credentials:

OS: Ubuntu 24.04.3 LTS

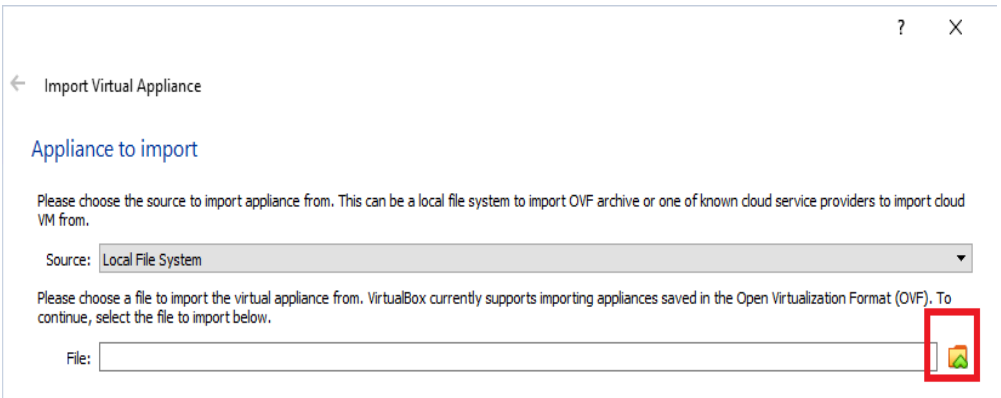
Username: cse321

Password: cse321

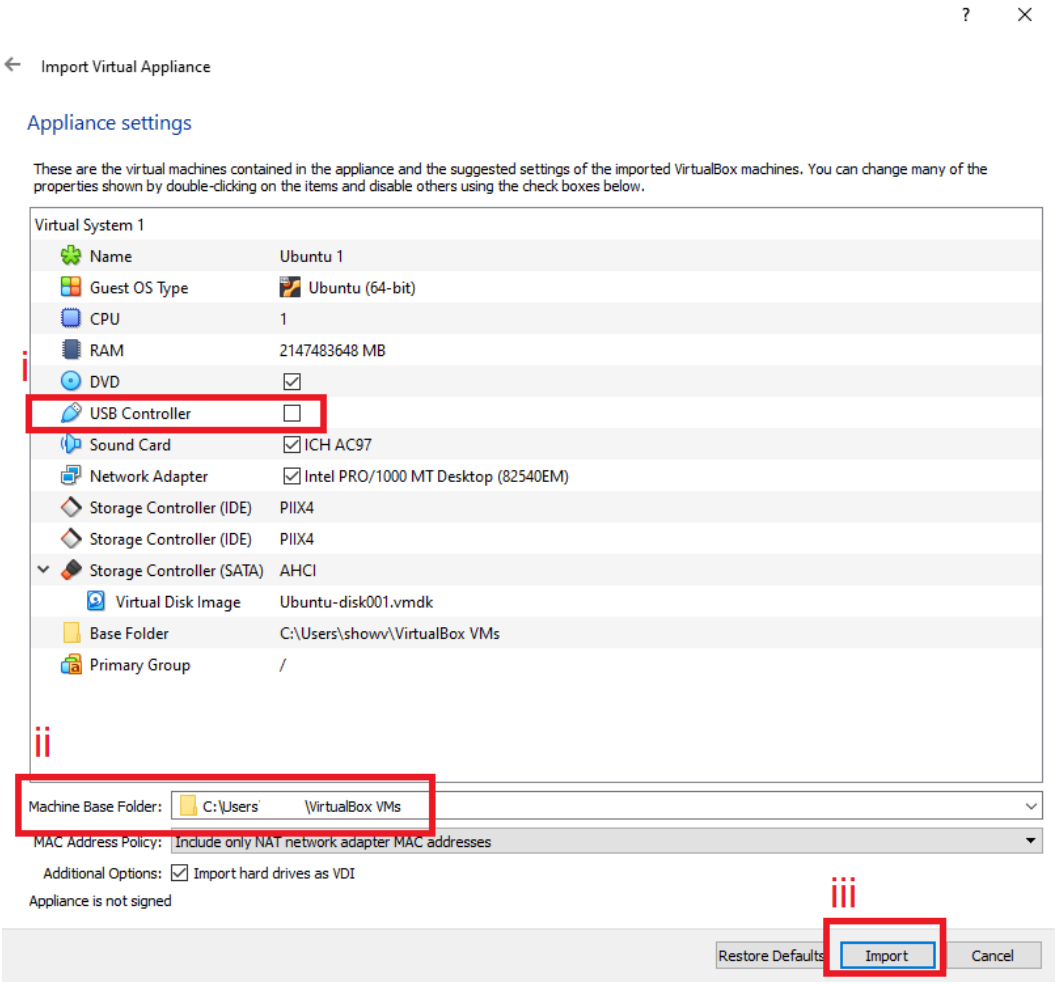
1. Download and install VirtualBox.
2. Download this Ubuntu Virtual Machine Image. It has all the tools needed for the lab pre-installed. Username/password of the machine are given above.
3. Once you install VirtualBox, perform the following:
4. Go to File -> Import Appliance:



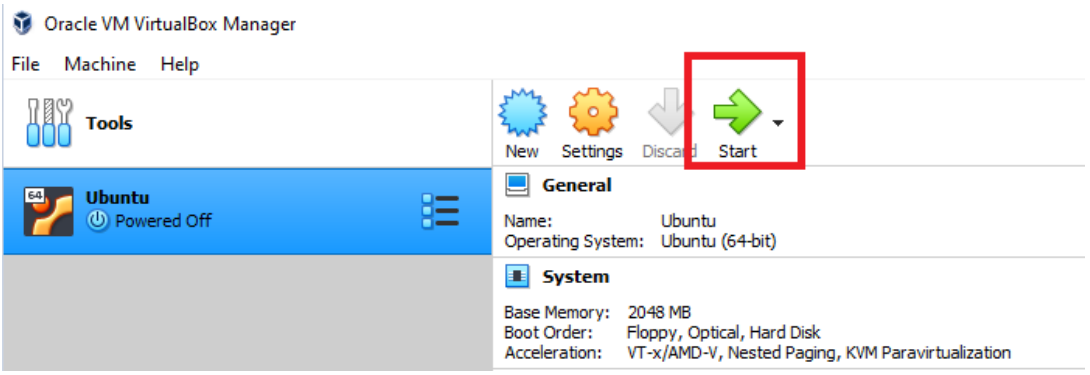
5. Click on the File icon as shown and then select the *Ubuntu.ova* file downloaded in Step 2, and click on **Next**.



6. Ensure to **uncheck** “USB Controller”, shown in (i). Optionally, you may change the Machine Base Folder, shown in (ii). Keep in mind that you will require about 30 GB of space so select a location accordingly. Then, click on **Import**, as shown in (iii).



7. Once Importing is complete, select it and click **Start** to power the machine up.



8. Log in to the machine using the provided credentials.
9. **Post-Login**: Once Ubuntu boots up properly, ensure the VM can access the Internet connection of your host device by opening up the terminal and running: *sudo apt update*. You will be prompted for the password again.
10. Run xv6: Move to ~/cse321/xv6-riscv, and run *make clean && make qemu*

Installing and Building Everything Yourself

This process will be a bit more complicated. It is assumed that you have downloaded, and installed **Ubuntu 24.04.3 LTS** natively on your OS, in a single OS or a dual-boot/multiple-boot setup, and are running it on an x86 processor. Once you have your OS running, perform the following:

1. Run *sudo apt update* to update all your repository indices.
2. Build the RISC-V GNU toolchain:
    a. Clone the toolchain repository using *git clone* *https://github.com/riscv/riscv-gnu-toolchain*
    b. Install a few prerequisites using the following:
    sudo apt install autoconf automake autotools-dev curl python3 python3-pip python3-tomli libmpc-dev libmpfr-dev libgmp-dev gawk build-essential bison flex texinfo gperf libtool patchutils bc zlib1g-dev libexpat-dev ninja-build git cmake libglib2.0-dev libslirp-dev make gcc
    c. Move into the repository folder of 2(a), and select the build path for the tools using *./configure --prefix=/opt/riscv*
    d. Run *sudo make* to build the toolchain. This process will take a lot of time depending on your CPU, so you should be patient.
3. Install the latest version of qemu using *sudo apt install qemu qemu-system-x86*.
4. Download xv6: From your parent folder, run *git clone* *https://github.com/mit-pdos/xv6-riscv.git*
5. Move into the xv6 folder, and check whether your prerequisites are installed
    a. *qemu-system-riscv64 --version*
    b. *riscv64-linux-gnu-gcc --version*
6. Run xv6: Inside the xv6 folder, run *make qemu*.

For installation issues, kindly reach out to your faculty with a detailed description of the error you're facing.