Operating Systems – Linux 1-2. Laboratory – Measurement Guide and Report Template

## Legend

This document displays commands to be entered or choices to be selected in blue bold text. When these commands or choices are only referenced **black bold text** is used.

## Introduction

You are a member of a group of young engineers who just created a prototype product that is promising enough to build a startup company on. Each members of the group have their own tasks and responsibilities: there are members working on the development, there is a member for fund raising, another for graphics and design and so on. Your task is to provide the IT background for the project. Unfortunately you are on a very low budget: as there are no investments yet, all costs are paid by the team members. As for the reputation of the young startup it is very important to only use legal software and not to conflict any licensing conditions.

By now the prototype is ready to be introduced to the investors and to the general public. For that aim the team decides to set up a trendy one page static HTML website. The designer member of the team will be responsible for the graphics and design, and your task is to set up the server computer for the homepage. When working on this task you have to focus on two key requirements: On the one hand the costs must be kept low, therefore setting up a standalone server is not an option. On the other hand the server might experience high load from a large number of visitors after the project has been announced, thus the solution should be easy to scale up when needed.

Based on these requirements you decide to rent a virtual private server (VPS), which starts pricing from 5 USD/month in 2024. The smallest VPS servers have 1 GB RAM, a single processor core and 20-40 GB storage. Most VPS service providers offer time based billing and dynamic server resizing, thus for high load times the underlying hardware can temporarily be upgraded, to fulfill the second requirement.

## 1. Find a suitable VPS service

Search the Internet for VPS services in the EU that fit your requirements for the VPS: less than 10 USD/month; at least 1 GB of RAM; at least 20 GB of storage!

**Enter the name of the chosen VPS provider and their offered package!**

**Insert a screenshot of the details of the VPS package you selected!**

## 2. Set up a virtual machine

After creating the VPS you decided to set up a virtual machine on your own computer to experiment with new software and configuration locally first, without affecting the live site.

We will use the VirtualBox application to create a local version of the VPS for testing purposes. On the Lab PC you have the empty virtual machine on your desktop.

If you don't have the files, you can download the prepared empty virtual machine from: <https://home.mit.bme.hu/~eredics/os/Operating-Systems-EN-Linux.zip> - Extract the files to your computer!

Open the "Operating Systems EN Linux lab.vbox" file to start the VM!

**Take a screenshot of your empty virtual machine!**

## 3. Install the operating system

Because of the limited resources of your VPS and the low budget you cannot install a commercial OS on your virtual machine like Windows. There is a wide variety of Unix based operating systems you can choose from: most of them are open source and free to use, and have very low hardware requirements. Your choice is the Ubuntu 24.04.

To start the Ubuntu installer:

* On the lab computers the installer image is already present on your desktop. If don't have it there you can download it from the following URL: <https://home.mit.bme.hu/~eredics/os/ubuntu-24.04.iso>
* Start your virtual machine! It will fail to boot as there is no bootable device yet.
* Open the **Devices** menu and the **Optical drives** submenu, and choose **Choose a disk file**!
* Find the installer ISO file (/tmp/guest/ubuntu-24.04.iso on the lab computers) and select **Open** to attach the installer ISO file to the virtual machine!

Reboot the virtual machine using the Machine menu's Reset option!

**Take a screenshot of the main menu of the installer disc!**

After booting from the install disc ISO, select **T**ry or Install Ubuntu Server!

**Select a language**

Choose English!

**Installer update available**

Choose Continue without updating!

Keyboard configuration

Select English (US) layout and English (US) variant and choose Done!

**Choose the type of installation**

Select Ubuntu Server and choose Done!

**Network** configuration

Defaults are good for now: Done!

**Proxy address**

Keep empty and Done!

**Ubuntu** archive mirror configuration

Wait for the speed test and Done!

**Guided storage configuration**

Select Use an entire disk and then Done!

**Storage configuration**

After reviewing the summary select Done!

**Confirm destructive action**

Select Continue!

Profile configuration

Your name: student

Your servers name: enter your Neptun code

Pick a username: student

Choose a password: ospass

Confirm your password: ospass

Select Done once finished.

**Upgrade to Ubuntu Pro**

Select Skip for now and Continue!

**SSH configuration**

Do not select to install the SSH server (we will install manually), just select Done!

**Featured server snaps**

Do not select any snap, just select Done!

Wait for the installer to finish the installation process. It might take a few minutes. Once the installation is done select Reboot Now!

If you see a message to remove the CD, please press Enter.

After rebooting the virtual machine log in with the student name and ospass password.

**Insert a screenshot of the console after your first login!**

Your first task is to check for available software updates. To do so execute the sudo apt update command to synchronize the local package database with the central servers. After that run sudo apt upgrade to install every available updates. Press y if asked about if you are sure to install the updates!

**Create a screenshot of the output of apt upgrade!**

## 4. Installing an SSH server

Accessing the physical console of a server computer is uncomfortable and in many cases it might be impossible as many servers do not have keyboard or display attached. To allow remote access you have to install an SSH (Secure Shell) server!

To install the necessary packages execute sudo apt install openssh-server!

After the installation you have to be aware of the IP address of the virtual machine to initiate an SSH connection. To determine the IP address enter the  
ip a command and look for the **eth0 or enp0s3** interface and the **inet** parameter!

**Make a note of the IP address of the virtual machine here! Make sure to only note the IP address (e.g. 152.66.89.1) and not the netmask (e.g. /123) at the end!**

Start an SSH client application:

* if you are using a Linux host OS, please open a terminal and enter the command ssh student@[ip-address-of-the-vm] replace **[ip-address-of-the-vm]** with the IP address you just noted above.
* if you are using a Windows host OS open the SSH client application (e.g. **PuTTY**)

When prompted for login use the student name and ospass password just as if you were using the console directly.

**Insert a screenshot of the SSH connection after logging in!**

From now on we suggest you to work through this SSH connection, as you can easily copy command from this guide into the SSH terminal window: to paste a command previously copied to the clipboard simply click right with your mouse in the **terminal** window! If necessary, you can open multiple SSH connection to the same server.

## 5. Installing a web server

The server was created to host a website, thus you need to install a web server application.

To install the most widely used web server application for Linux named **Apache2** execute the sudo apt install apache2 command! After finishing the installer open a web browser on your host machine and enter the IP address of the virtual machine into the address field. The default installation page of **Apche2** should be displayed by your browser!

**Take a screenshot of the default page in your web browser!**

As the default page says the content you see is served from the **/var/www/html** folder. To list the contents of this directory use the ls -la /var/www/html command!

From the output of **ls** you can see the name of the file being displayed in the web browser, and it is also visible that all files in the **html** directory are owned by the root user and the root group. From now on the content of this directory will be maintained by the student user, so you have to make this user the owner of the directory to grant the necessary permissions. To do so execute the sudo chown -R student:student /var/www/html command!

The **chown** command as most Unix commands does not display any output when the operation was successful, only error messages are shown when needed. To ensure that the permissions have been successfully updated execute the ls -la /var/www/htmlcommand again!

**Create a screenshot of the contents of the html directory displayed by the ls command!**

## 6. Upload the website

As the designer has not yet finished the final website only a basic template is provided to you to replace the default Apache website.

Use your SSH terminal and navigate to the html folder by issuing the cd /var/www/html command!

Download the files for the website: wget [**https://home.mit.bme.hu/~eredics/opre/static-html-website.zip**](https://home.mit.bme.hu/~eredics/opre/static-html-website.zip)

To extract the archive enter the unzip -o static-html-website.zip command!

The unzip operation fails. Why? Examine the error message, and solve the problem!

**What was the problem and how did you solve it?**

Extract the archive after solving the problem with the original unzip command. You can recall previous commands in the shell by using the up/down arrow keys on your keyboard.

After unzipping reload the page in your browser to see the changes!

**Take a screenshot of the static website in the browser served by your server!**