# Born2beroot

## First steps - knowing virtual machines and Debian

▼ What you should know before starting

What is a VM and how does it work?

Virtual machines are made possible by virtualisation technology. Virtualisation uses software to simulate virtual hardware that allows multiple VMs to run on a single machine. The physical machine is called the host while the VMs running on it are called the guests.

Type 1 hypervisors (also known as bare metal hypervisors) are installed natively on the underlying physical hardware. VMs interact directly with hosts to allocate hardware resources to intermediate software layers. Host machines running Type 1 hypervisors are only used for virtualisation. They are often found in server-based environments such as enterprise data centers.

Type 2 hypervisors (also known as hosted hypervisors) run on top of the host computer's operating system. Hosted hypervisors pass VM requests to the host OS, which then provides the appropriate physical resources to each guest. Type 2 hypervisors are slower than their Type 1 counterparts because every VM action must first go through the host operating system.

Debian vs CentOS

Debian is known as Debian GNU/Linux, a distribution composed of free open-source software, developed b the community- supported Debian Project.

Some advantages of Debian as developers:

**Multiple Hardware Architectures.**Debian supports a long list of CPU architectures, including amd64, i386, multiple versions of ARM and MIPS, POWER7, POWER8, IBM System z and RISC-V. Debian is also available for niche architectures.

**IoT and Embedded Devices**Debian runs on a wide range of devices, like the Raspberry Pi, variants of QNAP, mobile devices, home routers and a lot of Single Board Computers (SBC).

**Huge Number of Software Packages**Debian has a large number of packages (currently in stable: 59000 packages) which use the deb format.

**Different Releases**Besides our stable release, you can install newer software versions by using the testing or unstable releases.

**Public Bug Tracker**Our Debian bug tracking system (BTS) is publicly available for everybody via a web browser. We do not hide our software bugs, and you can easily submit new bug reports or join the discussion.

**Debian Policy and Developer Tools**Debian offers high-quality software. To learn more about our standards, read the policy which defines technical requirements for every package included in the distribution. Our Continuous Integration strategy involves Autopkgtest (runs tests on packages), Piuparts (tests installation, upgrade and removal), and Lintian (checks packages for inconsistencies and errors).

CentOS is a Linux partition that strives to provide a free enterprise-class computing program that has a 100% binary dealing with its upstream source, Red Hat Enterprise Linux. It does not support many different architectures. This should gain it very, very stable, and become used as a web server. Debian has much more packages in its default repositories than CentOS.

AppArmor (if you are planning of use Debian you can skip this this step of installation but it is important to know a bit about it).

**apt install apparmor apparmor-utils auditd**

**mkdir -p /etc/default/grub.d** → enable command, because it is a Linux kernel module

This will be storage, in a file with the address earlier put it.

**update-grub** → To save and exit.

**aa-status** → to see if AppArmor is active and running

AppArmor is an effective and easy-to-use Linux application security system. AppArmor proactively protects the operating system and applications from external or internal threats, even zero-day attacks, by enforcing good behaviour and preventing both known and unknown application flaws from being exploited.

▼ Now, the commands to follow!

It is important to mention that the following commands are in order to keep track of what are you doing, but certainly you will need to check the information behind to understand better and get ready for the evaluation.

▼ VM / installation with Debian

Enter to https://www.debian.org/ and download the latest version, this will do it automatically(or in any case the instructions are there to follow).

Opening the Virtual Box is necessary since it is mandatory in the PDF, the next step is going through the section “New” and configure it as the PDF. (Small tip, try to save it in the computer that you will use, /desktop, look for the number of the machine and Macintosh HD/goinfre/yourusername).

▼ Steps to consider when you open the Debian installer

**During the installation it is necessary to keep track of the passwords used.** (hostname, root, layer)

Once there you will need to follow the option without a graphic terminal (as it is signal in the PDF). This mean to take the option LMV with encrypted → /home

Choose delete and overwrite

Encryption password.

Fresh partitioning → Remove [*]SSH Server . [*]Standard System Ut...

Finally install Grub.

▼ Launch VM

Log with your login + password.

1. Let’s start to install sudo:

```
  su -                     #To enter to the root
apt-get install sudo
```

2. Add user to sudo group:

```
adduser username sudo  -> replace the user name
```

3. Install SSH:

```
sudo apt-get install openssh-server
sudo nano /etc/ssh/sshd_config
  #Modify the #port 22 -> port 4242 (without comment)
  #Modify the #PermitRootLogin ... -> PermitRootLogin no (without comment)
sudo nano /etc/ssh/ssh_config
  #Modify the #port 22 -> port 4242 (without comment)
sudo service ssh force reload
sudo service ssh status
sudo apt update
```

It is important to understand the difference in between the ssh server and the sshd.

4. Install UFW

```
sudo apt-get install ufw
sudo ufw enable
sudo ufw allow 4242
sudo ufw status

#to turn off the machine you can use:
sudo shutdown -h now
```

5. Extra step is to reboot, be use in the normal terminal for more efficient use.

```
Enter to VM  -> Settings -> Network -> Advanced -> Port -> AddNewRule
After Add New Rule
8:10
```

```
New Rule :
8:10
* Protocol       Host IP       Host Port       Guest IP       Guest Port *
* TCP            127.0.0.1     4242            10.0.2.15      4242
Leave the VM & open (relaunch)
```

6. Open the terminal

   ```
   To be able to login through ssh, it is necessary to open
   the Terminal, #replace the username:
   ssh username@localhost -p 4242
   ```

   ```
   sudo systemctl enable ssh
   sudo systemctl start ssh
   #(apt is an advanced package tool)
   ```

7. AppArmor

   Linux app security system.

   > Protects OS & apps from the external and internal threats.
   >
   > Normally, kernel module had it integrated. To be certain let’s check it:

   ```
   sudo aa-status
   ```

▼ Some information to keep in mind

- To create a new user:

  ```
  sudo adduser username        #check it in the terminal
  ```

- To list all users

  ```
  compgen -u
  ```

- to list all the groups

  ```
  sudo groups
  ```

- To an specific user group to specific group:

  ```
  sudo groups username         #replace the username
  ```

    - To check if I have sudo permissions:

      ```
      sudo -l
      ```

    - To create a new group:

      ```
      sudo addgroup groupname         #replace with the groupname
      ```

    - Add user to a group:

      ```
      sudo adduser username groupname #replace with the groupname/username
      ```

▼ Change the hostname

This step is important to know when you will get evaluated.

- Check the hostname:

  ```
  hostnamectl
  ```

- Change the hostname

  ```
  sudo hostnamectl set-hostname name    #replace with the desired name
  ```

- Now lets update it in the host-lists:

  ```
  sudo nano /etc/hosts                 #replace old hosts with the new ones
  ```

▼ Configure sudoers rules:

Add the new lines (TYPE THEM!!!)

```
(**BE AWARE THAT YOU HAVE A SUDOLOG FILE, IF NOT, CREATE ONE :)!)
sudo nano /etc/sudoers
    Defaults          secure_path = "from subject pdf" #copy the path in example of the PDF
    Defaults          passwd_tries = 3
    Defaults          badpass_message = "Wrong password"
    Defaults          logfile = "/var/log/sudo/sudolog"
    Defaults          log_input, log_output

***
FOR INFORMATION

root ALL = (ALL : ALL ) ALL
```

```
Rules apply to all the hosts
Root can be run commands as users
Root can run commands as all users
These rules apply to all commands

this is a command : TTY : teletypewriter
```

▼ Password Set-up 👽IF YOU ENTER TO ROOT YOU DON’T NEED TO USE “SUDO”

There is a quick reminder, Password age 🙂.

```
sudo nano /etc/login.defs
#uncomment and change the days:

  Maximum of days 30
  Minimun of days 2
  Warning advice  7
```

Password strength

```
sudo apt-get install libpam-pwquality #download package
#to check it:
    dpkg -l grep libpam-pwqality
```

Configure Password

TYPE THE RULES!!!

```
sudo nano /etc/pam.d/common-password
 #password requisite: (** REMEMBER ALL IN ONE LINE)
    pam_pwquality.so retry=3 minlen=10
    ucredit=-1 decredit=-1 maxrepeat=3
    reject_username difok=7 enforce_for_root
```

Change password

```
/To change password run as a root:
    passwd username           #replace username
/Verify password age:
    chage -l username       #replace username
/to add changes manually (in case it didn't apply to old passwords):
    chage -m 2 -M 30 username
```

YOU NEED TO CHANGE THE PASSWORD OF THE USER AND THE ROOT, BECAUSE IT WON’T LET YOU ENTER IN THE NEXT TIME.

but do not panic if you forgot, because you can recovered with the grub 🙂

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install cron
sudo systemctl status cron
/to configure (this must be done as ROOT):
  sudo nano /etc/crontab
  sudo crontab -e
/add in the end of file: (**CREATE A MONITORING.SH IF YOU DONT HAVE IT AND REMEMBER THE PATH OF THE FILE)
  (YOU CAN CREATE WHEREVER YOU WANT EX. /HOME/USERNAME/MONITORING.SH)
  */10 * * * * <path to monitoring.sh script>
  (m h dom mon dow)
/check:
  sudo crontab -l
```

Try to understand the path of the folders to change or create the necessary ones.

▼ BONUS PART

```


/lighttpd
  sudo apt install lighttpd
  sudo ufw allow 80        (part 80 -HTTP)
/PHP -> Hypertext Preprocessor
  sudo apt install php-cgi php-mysql
/verify:
  dpkg -l | grep php
/mariadb & sql
  sudo apt install mariadb-server
  sudo mysql_escure_installation
      no pass | y
          no  | y
      yes     | y

#login into mariadb:
sudo mariadb
```

```
# create a database:
CREATE DATABASE test42;   #replace test with any datebasename
#check the database
SHOW DATABASES;
```

```
#create a user:
CREATE USER ‘test’@localhost IDENTIFIED BY ‘password42’;
#replace test with any username and any password !!!Do not forget it, you will
need it to login into Wordpress page

#grant permission:
GRANT ALL ON test42.* TO 'test'@localhost IDENTIFIED BY 'password42'

#enable changes:
FLUSH PRIVILEGES;

#exit mariadb
ex
```

Download WORDPRESS

```
sudo apt install wget
sudo wget http://wordpress.org/latest.tar.gz -P /var/www/html
sudo tar -xzvf /var/www/html/latest.tar.gz
sudo rm /var/www/html/latest.tar.gz
sudo cp -r /var/www/html/wordpress/* /var/www/html
sudo rm -rf /var/www/html/wordpress
sudo cp /var/www/html/wp-config-sample.php /var/www/html/wp-config.php
sudo nano /var/www/html/wp-config.php
  DB_NAME -> test42 (database name)
  DB_USER -> test (user name)
  DB_PASS -> password42 (password)
sudo lighty-enable-mod fastcgi
sudo lighty-enable-mod fastcgi-php
sudo service lighttpd force-reload
```

Script monitoring.sh:

The way to fill your script is by having the guide open and check the requirements of the script itself, once read lets look at the script with variables.

```
#USERname  Script
#TO see OS and kernel version
arquitecture=$(uname -a)
#Number of physical processors
cpu_physical=$(cat /proc/cpuinfo | grep 'physical id' | wc -l)
#Number of virtual processor
vcpu=$(cat /proc/cpuinfo | grep 'processor' | wc -l)
#Memory usage
memory_usage=$(free -m | awk 'NR==2{printf"%s/%sMB(%.2f%%)",$3,$2,$3*100/$2}')
#Current memory available on your server and itrs utilization rate as percentage
disk_usage=$(df -h | awk '$NF=="/"{printf"%d%dGB(%s)",$3,$2,$5}')
#Current utilization rate of the processor as percentage
cpu_load=$(top -bn1 | grep "Cpu(s)" |sed "s/.*, *\([0-9.]*\)%* id.*/\1/" | awk '{print 100 - $1"%"}')
#The date and the time
last_boot=$(who -b | awk '{printf$3" "$4" "$5}')
#Whether LVM is active or not
lvm_use=$(lsblk | grep lvm | awk '{if ($1) {print "yes"; exit;}else{print"no"}}')
#Number of active connections
connextion_tcp=$(netstat -ant | awk '{print $37}' | sort | uniq -c | sort -n)
#Number of user using the server
num_usrs=$(who | sort -u |wc -l)
#Network
net=$(hostname -I)
mac=$(ifconfig -a | grep "ether" | cut -c 15-31)
#The number of commands executed with the sudo program
n_log=$(grep -a 'sudo' /var/log/auth.log |wc -l)
wall "Arquitecture:  $arquitecture
Number of physical processor: $cpu_physical
Number of virtual processor: $vcpu
```

```
Memory usage: $memory_usage
Disk usage: $disk_usage
CPU load: $cpu_load
Last boot: $last_boot
LVM USE: $lvm_use
Connexins TCP: $connextion_tcp
User log: $num_usrs
Network: IP $net($mac)
Sudo: $n_log cmd
"
```

It is important to remember that must have an enter at the end of the file (like Norminette). That way will compile without problems. Finally let’s review some of the concepts required to understand what we did.

- uname or unix name; prints the name , version and other details in the current OS
- cat or concatenate; allows yu to create a single or multiple files, view the content.
- cat grep, instead of printing out every word that contains the pattern, GREP allow us to to print the word by itself. It does the same thing for the entire lines with the -x flag, so if oyu are looking for a phrase or a single line in a configuration file, that can really help.

  <<Grep will print out any lines in the file containing the word that you told it to look for.>>
- free command provides info about the total amount fo the physical and swap memory, as well as the free and used memory
- -m, Megabytes
- awk, scripting language used to manipulating data in generating reports. Requires no compiling and allows the user to use variables, numeric functions, string functions, and logical operators. “In other words, allows the programmer to find the exact statements in a document and the actions and the actions is to be taken when a match is found in the line”.

  **AWK Operations:** (a) Scans a file line by line (b) Splits each input line into fields (c) Compares input line/fields to pattern (d) Performs action(s) on matched lines

  **Useful For:** (a) Transform data files (b) Produce formatted reports

  **Programming Constructs:** (a) Format output lines (b) Arithmetic and string operations (c) Conditionals and loops

  ```
  awk options 'selection _criteria {action }' input-file > output-file
  ```

  for more information: https://www.geeksforgeeks.org/awk-command-unixlinux-examples/
- top, very useful to see the state of the your unix system, by default present you the list of top users of yours system’s resources (CPU shares and memory).
- -h, means human readable format (ex. 42M= meaning 23 MB)

- df or disk free, a standard command for display the amount of available disk space for the file systems on which the invoking user has appropriate read access.
- bn1, displays an calculate average CPU usage, it goes along with “top”, s recommendable to skip the first one, because might produce error, better bn2.
- who, displays a list of users who are currently logged into de computer, along with -b, show the time when the system was last rebooted.

  For more info review: https://en.wikipedia.org/wiki/Who_(Unix)
- lsblk, is used to display details about the block devices (except RAM), basically files that representing devices connected to the pc
- sort, prints the lines of its input or concatenation of all the files listed in its argument list in sorted order
- uniq, reports or filters out the repeated lines in a file. along with -c (count) shows the amount of the repeated prefix with the line

  For more info: https://www.geeksforgeeks.org/uniq-command-in-linux-with-examples/
- -ant, when arguments are not specified, ant looks for a build .https://www.tutorialspoint.com/ant/ant_build_files.htm
- netstat, list out all the network(socket) connections on a system. Can also list listening sockets that are waiting for incoming connections.
- ifconfig, is use to configure and view the status of the network interfaces in Linux operating systems