https://github.com/pgomez-a/born2beroot

Basic functioning of the machine :

VM → software capable of installing an OS within itself making it think the OS is hosted on a real computer. VM uses the host hardware and is isolated from it (hypervisor uses part of the host machine and distributes them).
There can be multiple VMs on a machine and they behave as if they were hosted on a physical device → same experience when using os on virtual or physical machine

Why Debian?
Easier to setup and use than Centos.

Diff between Debian and Centos :
Architecture is generally the same except for centos7
Package management difference between the two YUM/DNF vs dpkg/APT
Upgrading easier on Debian
Debian more community supported

Why virtual machines?
Different guest machines hosted on our computer can run different operating systems → diff OS working on the same machine.
Provide env to safely test unstable programs.
Failsafe → if crash the VM can be migrated to another physical machine
Less hardware hungry = less money → 1 hardware = several virtual servers
More perf → better use of shared ressouces
Easy to implement → cloning

APT and Aptitude?
Aptitude = high level package manager —> does apt-get apt-mark -apt-cache —> if computer has UI is what the computer uses —> more interactions
apt = lower level, no UI only command line use dkpg but also install the dependencies for the program to work correctly.
dkpg to install but sometimes need dependencies so apt lets you install the program + dependencies

APPArmor?
Provides MAC (mandatory access control) security.
Linux kernel security module —> confines programs according to a set of rules that specify what files a given program can access. Helps against known and unknown vulnerabilities.
For example, if an installed application can take photos by accessing the camera application, but the administrator denies this privilege, the application will not be able to access the camera application. If a vulnerability occurs (some of the restricted tasks are performed), AppArmor blocks the application so that the damage does not spread to the rest of the system.
Profile system, enforce mode (prohibits), complain mode (allow but create of log of the complaint)

---

in sudo
service ufw status
service ssh status
uname -a

Sudo adduser <name>
Getent group <name>
Sudo adduser <name> <group> / usermod -aG <group> <username>
passwd -e to change passwd even for root

Groupadd <name>

how did i setup the rules for passwd
sudo vi /etc/login.defs
sudo apt install libpam-pwquality → sudo vi /etc/pam.d/common-password
difok = diff between new and old password
ucredit = uppercase
lcredit = lowercase
dcredit = digits

Password takes average 7month to brute force with 1u 1l 1d
More difficult to crack but at the same time come with drawback for user → password hygiene is bad, either forget or use previously easy password with a few more characters. also they are biases for example 3millions 8 characters pwd ‘e’ has been used 1.5 million times where as ‘f’ only 250k times in a study

---

Hostnamectl / hostname / cat /proc/sys/kernel/hostname
Hostnamectl set-hostname / vi /etc/hostname
Lsblk
What is LVM?
Abstraction layer between storage device and a file system.
LVM → easier to manage partitions → we can expand it using avalable storage located on diff physical disks. we can also move the lvm between physical devices, they will still behave the same.

Dpkg -l | grep sudo
Sudo adduser <name> sudo
Sudo
Sudo allows a user to run commands with elevated privileges (usually root level) as superuser
sudo cat /var/log/sudo/sudo.log / sudo apt-get

---

dpkg -l | grep ufw
Check UFW status —> sudo ufw status
Check SSH status —> sudo service ssh status / systemctl status ssh

UFW

UFW (Uncomplicated Firewall) is a software application responsible for ensuring that the system administrator can manage iptables in a simple way. Since it is very difficult to work with iptables, UFW provides us with an interface to modify the firewall of our device (netfilter) without compromising security. Once we have UFW installed, we can choose which ports we want to allow connections, and which ports we want to close. This will also be very useful with SSH, greatly improving all security related to communications between devices.
UFW rule to 8080 —> sudo UFW allow 8080
UFW status numbered —> sudo status numbered, sudo UFW delete number of the rules

---

Dpkg -l | grep ssh
Sudo service ssh status

SSH

SSH or Secure Shell is a remote administration protocol that allows users to control and modify their servers over the Internet thanks to an authentication mechanism. Provides a mechanism to authenticate a user remotely, transfer data from the client to the host, and return a response to the request made by the client. SSH was created as an alternative to Telnet, which does not encrypt the information that is sent. SSH uses encryption techniques to ensure that all client-to-host and host-to-client communications are done in encrypted form. One of the advantages of SSH is that a user using Linux or MacOS can use SSH on their server to communicate with it remotely through their computer's terminal. Once authenticated, that user will be able to use the terminal to work on the server.
Ssh username@ip -p 4242
grep -i port /etc/ssh/sshd_config

Cron

Linux task manager that allows us to execute commands at a certain time. We can automate some tasks just by telling cron what command we want to run at a specific time. For example, if we want to restart our server every day at 4:00 am, instead of having to wake up at that time, cron will do it for us.

Wall

command used by the root user to send a message to all users currently connected to the server. If the system administrator wants to alert about a major server change that could cause users to log out, the root user could alert them with wall.

Cron every 30s —> (sleep 30; ….)
systemctl disable service