Theory

Virtual Machines

A virtual machine is an implementation of a computer that behaves like a physical machine. It emulates hardware components such as CPU, memory, storage and network interfaces.

Advantages:

- Allows you to install multiple operating systems simultaneously on a single physical machine;
- Virtual machines can be backed up and saved. This is useful in case of hardware failure.
- Working in a virtual machine prevents you from damaging the whole computer if something goes wrong when developing.

Debian vs Rocky

Debian

- Serves as upstream source of other linux distributions.
- Has a large community and therefore extensive documentation.
- It is versatile, and can be used in various architectures.

Rocky

- Created to fill the market gap left by CentOS. Is a downstream of RHEL (Red Hat Enterprise Linux).
- Is made for people familiarised with CentOS or RHEL.
- For enterprise use.

Debian

Is a robust and versatile operating system based in Linux.
Has 3 branches:

- Stable (the most tested and secure one);
- Testing (not intensively tested yet - for people that want the most recent software);
- Unstable (not tested yet - used by developers);

Package management

This is the process of installing, configuring and removing software packages on a computer. Debian's package manager can be either Apt or Aptitude.

Apt vs Aptitude

- Apt is a command-line tool. It provides a simple and straightforward interface for package management.
- Aptitude is a more advanced tool. It has a Text user interface (TUI) and advanced resolution algorithms.

Note: Resolution algorithms determine all the necessary dependencies to be installed when installing any package. They also deal with conflicts in package installation.

Security
APParmor is a Linux Security Module used by Debian. Administrators can define profiles and different permissions with text based configuration files (easy to manage by noobs).

LVM - Logical Volume Manager
It allows a flexible and dynamic management of storage without the need of interruption of services or data migration. It divides the memory into ‘logical volumes’. Partitions are sections or divisions of a physical device that are treated as separate units by the operating system, each with its own file systems and properties. A partition can be one or more logical volumes.

UFW - Uncomplicated firewall
Its a user-friendly interface for managing firewall rules. It’s simpler than “iptables”. UFW enhances system security by allowing users to define and control traffic.

TTY mode
A TTY represents a console or a terminal and requiring one ensures that users are executing commands directly from an interactive session on the system itself. This provides an extra security layer since it prevents users from using sudo commands remotely.

SSH
An SSH connection refers to a secure network protocol used to log into a remote machine and execute commands. SSH stands for "Secure Shell" and provides a secure channel over an unsecured network, such as the internet.
Key Features of SSH
- Encryption: SSH encrypts the data being transmitted, ensuring that any data sent over the connection cannot be easily intercepted and read by third parties.
- Authentication: SSH uses public key cryptography for authentication. Users can authenticate themselves using passwords, public keys, or other methods like certificates.
- Integrity: SSH ensures that the data has not been altered during transit by using message authentication codes (MACs).

Socket statistics
- Socket statistics provide information about the network sockets in use on a system. These statistics can help diagnose network issues, monitor network usage, and understand the state of network connections. The `ss` command in Unix-like operating systems is commonly used to display these statistics.
- A socket is an endpoint for sending or receiving data across a computer network. It is a combination of an IP address and a port number. Sockets can be used for various types of communication, including TCP (Transmission Control Protocol) and UDP (User Datagram Protocol).

Cron

Cron is a time-based job scheduling daemon in Unix-like operating systems. It allows users to schedule scripts or commands to run automatically at specified times and intervals. Cron is commonly used for system maintenance tasks, backups, and other repetitive tasks.

IP of the computer

An IP (Internet Protocol) address is a numerical label assigned to each device connected to a computer network that uses the Internet Protocol for communication. These can be static or dynamic.

Implementation

Summary of the used commands. The underlined commands will be needed for the evaluation.

See partitions: lsblk

Install sudo

1. su - (go to root user)
2. apt-get update -y
3. apt-get upgrade -y
4. apt install sudo
5. Verify if sudo is installed: which sudo

Manage groups and users

1. Create a new user: sudo adduser (username)
2. Delete an user: sudo deluser (username)
3. List users: cat /etc/passwd
4. Create new group: sudo groupadd (group)
5. Delete a group: sudo groupdel (group)
6. Show groups: cat /etc/group
7. Show groups in which the current user is in: groups
8. Add user to a group: usermod -aG (group) (username)
9. Verify which users are in a group: getent group (group)

Give permissions to your user

1. *sudo visudo* - to enter in the sudoers file
2. Look for #user privilege specification - add above “(username) ALL=(ALL) ALL”

Notes: you gave to be in the root user. To exit root mode type “exit”.

SSH

1. Install: sudo apt install openssh-server
2. Check server status: sudo systemctl status ssh
3. Change the port to 4242: sudo vim /etc/ssh/sshd_config

4. Check ports: sudo grep Port /etc/ssh/sshd_config
5. Restart the server: sudo service ssh restart

UFW

1. Install: apt-get install ufw
2. Enable: sudo ufw enable
3. Check status: sudo ufw status numbered
4. Configure rules: sudo ufw allow ssh
5. Configure port rules: sudo ufw allow (n. port)
6. Remove port: sudo ufw delete (nr.)
7. Restart ssh server: sudo systemctl restart ssh
8. Check ssh status: sudo service sshd status

Get IP: hostname -I

Change password policy

1. Install password quality checking library: sudo apt-get install libpam-pwquality
2. Access to the file to change rules: sudo vim /etc/pam.d/common-password
    - retry=3
    - minlen=10
    - ucredit = -1; dcredit = -1; lcredit = -1 (uppercase, digit, lowercase -> at least one)
    - enforce_for_root
    - difolk=7 (at least 7 characters should be different from the previous password)

Change password validity

1. Go to the document: sudo vim /etc/login.defs
2. Change: PASS_MAX_DAYS 30; PASS_MIN_DAYS 2; PASS_WARN_AGE 7
3. Reboot

Access my virtual machine through another terminal by SSH:

ssh (username)@(IP) -p (n. port)
exit

Save sudo actions into a file and change sudo defaults

1. Create a file sudo.log in the /var/log/sudo path
2. Add this path to the sudoers file: vim /etc/sudoers (or sudo visudo on root)

Note: requiretty default enables the TTY mode.

Change the hostname

sudo hostnamectl set-hostname new-hostname

Script - 10/10 min

#!/bin/bash

- To specify that the script should be executed using bash;

arc=$(uname -a)

- uname - prints system info
- -a - stands for all

pcpu=$(grep "physical id" /proc/cpuinfo | sort | uniq | wc -l)

- grep looks for the expression in the directory
- sort - sorts
- uniq - removes duplicates
- wc -l - counts lines

vcpu=$(grep "^processor" /proc/cpuinfo | wc -l)

- grep== ; ^ means starts with; wc -l ==;

fram=$(free -m | awk '$1 == "Mem:" {print $2}')
uram=$(free -m | awk '$1 == "Mem:" {print $3}')
pram=$(free | awk '$1 == "Mem:" {printf("%.2f"), $3/$2*100}')

- free: display information about the system's memory usage
- -m option specifies that the output should be in megabytes;
- awk - checks if the first field of each line is equal to "Mem:"; then prints row $nr.

fdisk=$(df -BG | grep '^/dev/' | grep -v '/boot$' | awk '{ft += $2} END {print ft}')
udisk=$(df -BM | grep '^/dev/' | grep -v '/boot$' | awk '{ut += $3} END {print ut}')
pdisk=$(df -BM | grep '^/dev/' | grep -v '/boot$' | awk '{ut += $3} {ft+= $2} END {printf("%d"), ut/ft*100}')

- df : displays disk space usage;
- -BG (gigabytes); -BM (megabytes);
- The -v flag in grep is to exclude. The $ means ending with;
- END - only in the end after the sum prints;

cpul=$(top -bn1 | grep '^%Cpu' | cut -c 9- | xargs | awk '{printf("%.1f%%"), $1 + $3}')

- top is a dynamic real time view of the running system
- -b bash mode
- -n1 to only be executed once
- xargs removes newlines and spaces
- cut -c 9- cuts the first 8 characters of the line
- printf: The command used for formatted output. "%.1f%%":
    - %: Introduces a format specifier

    - .1: Specifies one digit after the decimal point for floating-point numbers.
    - f: Specifies the format type as a floating-point number.
    - %%: Escapes the percent sign to include it in the output.

lb=$(who -b | awk '$1 == "system" {print $3 " " $4}')

- who - lists the users & details
- -b - last boot

lvmu=$(if [$(lsblk | grep "lvm" | wc -l) -eq 0]; then echo no; else echo yes; fi)

- If there are no lines of the partitions with llvm there - write “no”;
- Else - write yes;
- “fi” is marking the end of the if block;

ctcp=$(ss -neopt state established | wc -l)

- ss shows socket statistics

By using `ss`, administrators and users can gain insights into the state of network connections, diagnose issues, and manage network configurations more effectively.

- -t: Show TCP sockets.
- -p: Show the process using the socket.
- -n: Do not resolve service names (show numerical addresses)
- -o: Show timer information.

ulog=$(users | wc -w)

- Counts the no. of users logged in right now
- -w - words

ip=$(hostname -I)

mac=$(ip link show | grep "ether" | awk '{print $2}')

- ip - info about the system
- link - filters to only network interfaces
- Show - displays the info

cmds=$(journalctl _COMM=sudo | grep COMMAND | wc -l)

- journalctl - command to query system and service data
- _COMM=sudo - it’s a filter of journalctl that limits the output to entries where the command name is sudo

wall

- It is a command to broadcast a message to all logged-in users;

"
#Architecture: $arc
#CPU physical: $pcpu
#vCPU: $vcpu
#Memory Usage: $uram/${fram}MB ($pram%)
#Disk Usage: $udisk/${fdisk}Gb ($pdisk%)
#CPU load: $cpul
#Last boot: $lb
#LVM use: $lvmu
#Connections TCP: $ctcp ESTABLISHED
#User log: $ulog
#Network: IP $ip ($mac)
#Sudo: $cmds cmd"

Automate the script

1. Go to the crontab and call sleep.sh and monitoring.sh
2. @reboot - to run the script at every reboot
3. */10 - means every ten minutes
4. **** - 4 * for every hour, day, month, year
5. sleep.sh - calculates the delay to sleep in:
    - `uptime -s`: This command displays the system's uptime in a human-readable format.
    - `cut -d ":" -f 2`: This command extracts the second field from the input text, where fields (-f) are delimited (-d) by the colon (`:`) character.
      Delay:
    - `MIN % 10 * 60` converts the minutes value (0-9) to seconds by multiplying by 60, as there are 60 seconds in a minute.
    - `+ SEC` adds the seconds value (`SEC`) to the converted minutes value, resulting in the total delay in seconds.