Virtual Machine: virtual ver of a computer, w/ storage borrowed from physical computer

- advantages :
    - saves physical space
    - customization : can control system versions, storage, etc

Rocky vs. Debian:

- rocky for business, debian for personal
- debian supports many architectures

Apt: interactive command line tool for managing deb packages

- deb packages : deb file format, for Debian Linux stuff

Aptitude : offers visual interface, for example can display debian change log

AppArmor : Linux security kernel security model, allows the system administer to restrict programs' capabilities such as network access, read/write access, etc

```
ufw status
service ssh status
cat /etc/os-release or uname -a
```

```
getent group
vi /etc/login.defs <- password expiry for security reasons (hacker may try to use the old logins)
vi /etc/pam.d/common_password <- password policy to make it harder to guess
sudo adduser eval
sudo addgroup evaluating
sudo adduser eval evaluating
groups eval
```

```
uname -n
sudo adduser eval sudo
sudo login eval
sudo vi /etc/hostname <- change to eval
sudo reboot
(maybe repeat, to restore original hostname)
lsblk <- partitions
```

LVM: logical volume management

- form of storage visualization that offers system admistrators a more flexible approach to managing disk storage space
- combine different physical storage spaces even for base layers of laptop

```
dpkg -l | grep sudo
sudo adduser eval sudo
vi /etc/sudoers.d/sudoconfig (sudo visudo?)
        - 3 attempts
        - custom message
(ex: chmod?)
sudo cat /var/log/input
sudo cat /var/log/output
```

```
sudo ufw status
```

ufw: uncomplicated firewall : tool for easily managing a net filter firewall

- command line interface
- firewall : network security system that monitors and controls incoming and outgoing network traffic (based on predetermined security rules) , typically between trusted network and an untrusted network

```
sudo ufw allow 8080
sudo ufw status numbered
sudo ufw delete $NUMBER
```

sudo service ssh status
ssh : secure shell protocol : allows two computers to communicate

- allows connection to linux servers remotely

sudo vi /etc/ssh/sshd_config
login w ssh from host machine: ssh anakasuji42@127.0.0.1 -p 4242
login root <- check that you can”t login w root user

sudo vi /usr/local/bin/monitoring.sh <- script

- bash script for providing system’s key metrics and information to all logged in users
- arc=$(uname -a): This line stores the output of the uname -a command in the variable arc, which provides information about the system's architecture.
- pcpu=$(grep "physical id" /proc/cpuinfo | sort | uniq | wc -l): This line counts the number of physical CPUs by extracting lines with "physical id" from the /proc/cpuinfo file, sorting and removing duplicates, and then counting the lines.
- vcpu=$(grep "^processor" /proc/cpuinfo | wc -l): This line counts the number of virtual CPUs by counting the lines that start with "processor" in the /proc/cpuinfo file.
- fram=$(free -m | awk '$1 == "Mem:" {print $2}'): This line uses the free command to get the total system memory (in megabytes) and stores it in the variable fram.
- uram=$(free -m | awk '$1 == "Mem:" {print $3}'): This line uses the free command to get the used system memory (in megabytes) and stores it in the variable uram.
- pram=$(free | awk '$1 == "Mem:" {printf("%.2f"), $3/$2*100}'): This line calculates the percentage of used memory and stores it in the variable pram.
- fdisk=$(df -BG | grep '^/dev/' | grep -v '/boot$' | awk '{ft += $2} END {print ft}'): This line calculates the total disk space (in gigabytes) by summing the sizes of all mounted partitions except /boot.
- udisk=$(df -BM | grep '^/dev/' | grep -v '/boot$' | awk '{ut += $3} END {print ut}'): This line calculates the used disk space (in megabytes) by summing the used space of all mounted partitions except /boot.
- pdisk=$(df -BM | grep '^/dev/' | grep -v '/boot$' | awk '{ut += $3} {ft+= $2} END {printf("%d"), ut/ft*100}'): This line calculates the percentage of used disk space.
- cpul=$(top -bn1 | grep '^%Cpu' | cut -c 9- | xargs | awk '{printf("%.1f%%"), $1 + $3}'): This line calculates the CPU load as a percentage using the top command.
- lb=$(who -b | awk '$1 == "system" {print $3 " " $4}'): This line retrieves the last system boot time.
- lvmu=$(if [$(lsblk | grep "lvm" | wc -l) -eq 0]; then echo no; else echo yes; fi): This line checks if LVM is in use on the system.
- ctcp=$(ss -neopt state established | wc -l): This line counts the number of TCP connections in the "ESTABLISHED" state.
- ulog=$(users | wc -w): This line counts the number of logged-in users.
- ip=$(hostname -I): This line retrieves the system's IP address.
- mac=$(ip link show | grep "ether" | awk '{print $2}'): This line retrieves the MAC (Ethernet) address of the system.
- cmds=$(journalctl _COMM=sudo | grep COMMAND | wc -l): This line counts the number of sudo commands in the system's journal.
- wall " ... ": This line sends a message to all users logged into the system using the wall command. It includes various system information collected earlier. The message is formatted with comments (lines starting with #) to make it more readable.

cron : job scheduler for unix like operating systems

- can use cron to schedule jobs, or run periodically at fixed times, dates, or intervals

sudo crontab -u root -e
*/1 * * * * sleep 30s && script path <- to run it every 30 secs
to make script stop running after reboot: delete @reboot /home/monitoring.sh
*/1 * * * * /home/monitoring.sh

sudo reboot
sudo vi /usr/local/bin/monitoring.sh