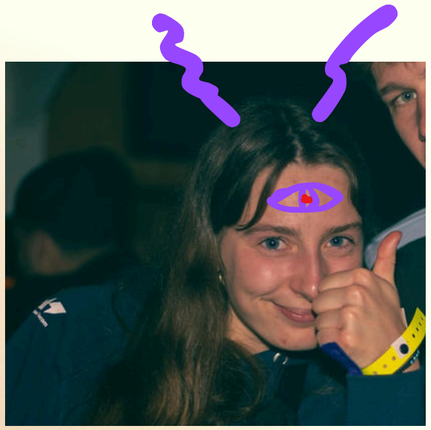
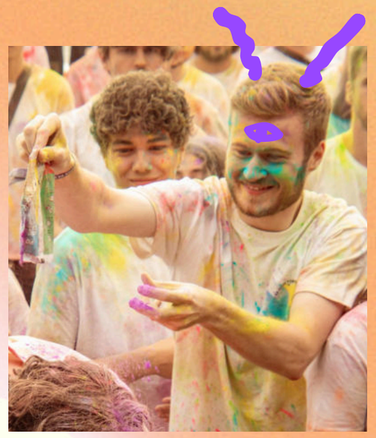
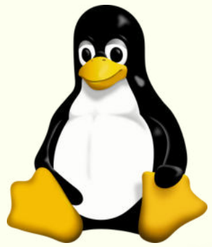




Victime

Formation Linux

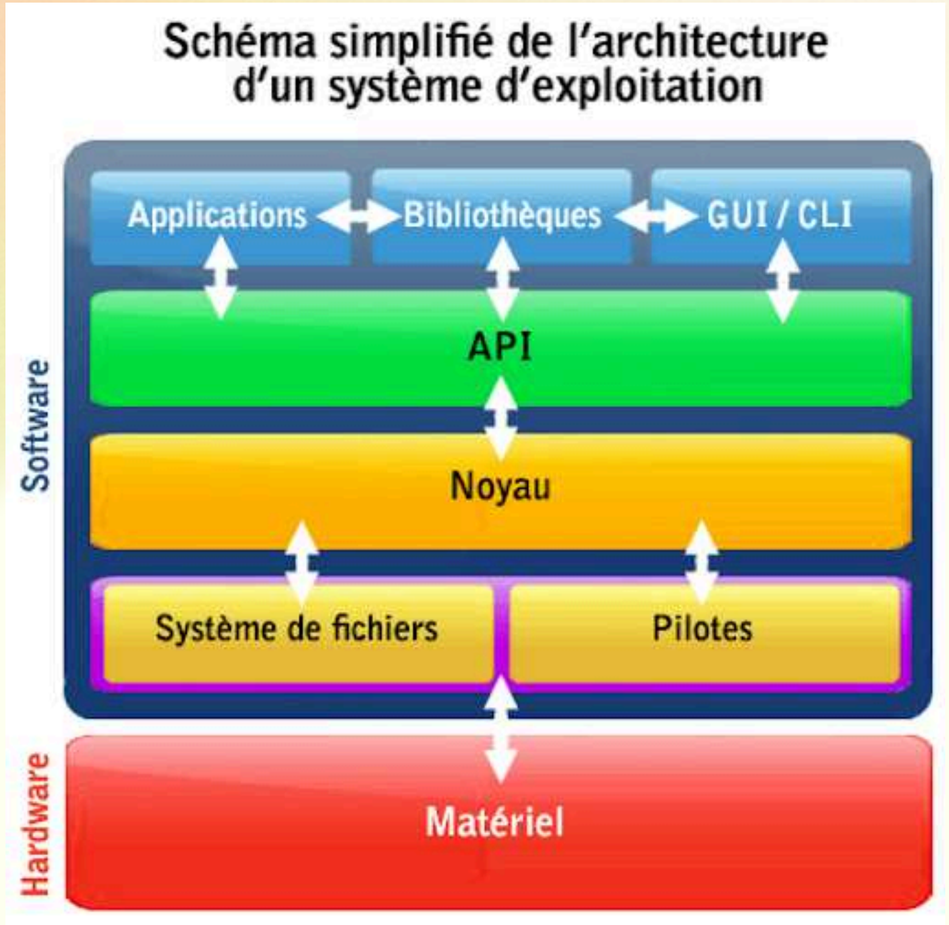


Avancée

MAINTENANT ON SAIT FAIRE PLUS QUE SURVIVRE

par rezyo et amélique

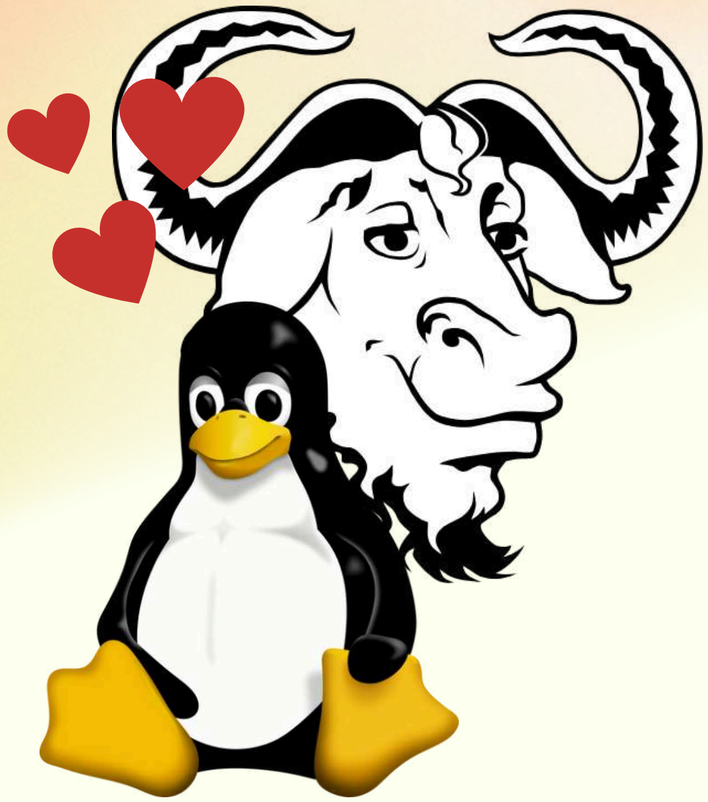
Récap : le kernel



Le kernel assure la **communication entre le matériel et les logiciels**.

kernel = chef d'orchestre de la machine

Récap : GNU / Linux



GNU/Linux

Ubuntu

fedora

debian

- **omagad : utilisé partout :**

minet

Google Cloud Platform

aws

android

- **omagad trop bien trop opensource :**

open source

GRATUIT

Récap : les commandes bash

on dit merci toto

Commandes	Explications
man <commande>	Manuel d'une commande
ls	Affiche les élèments
cd	Se déplacer
touch <fichier>	Créer un fichier
mkdir <répertoire>	Créer un répertoire
cat <fichier>	Afficher le contenu de <fichier>
nano <fichier>	Editer un <fichier>

Commandes	Explications
rm (-r si répertoire)	Suppression
cp <objet> <répertoire_destination>	Copier un objet
mv <objet> <répertoire_destination>	Déplacer un objet OU renommage
sudo	Utiliser une commande en administrateur
apt	Gestion des paquets

Pimp my linux

— quelques commandes en plus pour maîtriser la bête

apt update
apt upgrade

--mettre à jour votre PC

- fonctionne avec un **sudo** (droits avancés)
- up**date** pour charger les mises à jour
- up**grade** pour mettre à jour

7/ L'hygiène informatique c'est :

- se laver les mains avant de toucher son ordinateur
- faire les mises à jour de son système d'exploitation et de ses applications
- déconnecter son système d'information d'Internet
- faire une cartographie de ses équipements

```
amelique@ubuntuamelique: ~
amelique@ubuntuamelique: ~
amelique@ubuntuamelique: $ sudo apt update
[sudo] Mot de passe de amelique :
Réception de :1 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Atteint :2 http://fr.archive.ubuntu.com/ubuntu jammy InRelease
Réception de :3 http://fr.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Atteint :4 http://fr.archive.ubuntu.com/ubuntu jammy-backports InRelease
Réception de :5 https://dl.cloudsmith.io/public/symfony/stable/deb/ubuntu jammy InRelease [4 416 B]
Atteint :6 https://deb.nodesource.com/node_20.x nodistro InRelease
Atteint :7 https://ppa.launchpadcontent.net/mmk2410/intellij-idea/ubuntu jammy InRelease
Atteint :8 https://paulcarroty.gitlab.io/vscodium-deb-rpm-repo/debs vscodium InRelease
Réception de :9 http://fr.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [2 066 kB]
Réception de :10 http://fr.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [704 kB]
Réception de :11 http://fr.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1 128 kB]
Réception de :12 http://fr.archive.ubuntu.com/ubuntu jammy-updates/universe i386 Packages [735 kB]
4 895 ko réceptionnés en 2s (3 153 ko/s)
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
129 paquets peuvent être mis à jour. Exécutez « apt list --upgradable » pour les voir.
N: Le fichier configuré « main/binary-i386/Packages » ne sera pas pris en compte car le dépôt « https://paulcarroty.gitlab.
tecture « i386 »
amelique@ubuntuamelique: $ sudo apt upgrade
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Calcul de la mise à jour... Fait
Les paquets suivants ont été installés automatiquement et ne sont plus nécessaires :
  gyp libjs-events libjs-inherits libjs-is-typedarray libjs-psl libjs-source-map libjs-sprintf-js libjs-typedarray-to-buffe
  node-ansi-regex node-ansi-styles node-ansistyles node-aproba node-archy node-are-we-there-yet node-argparse node-arrify n
  node-chalk node-chownr node-clean-yaml-object node-cli-table node-clone node-color-convert node-color-name node-colors no
  node-core-util-is node-cssom node-cssstyle node-debug node-decompress-response node-defaults node-delayed-stream node-del
  node-escape-string-regexp node-events node-fancy-log node-foreground-child node-fs-write-stream-atomic node-fs.realpath n
  node-graceful-fs node-growl node-has-flag node-has-unicode node-hosted-git-info node-https-proxy-agent node-iconv-lite no
  node-ini node-ip node-ip-regex node-is-buffer node-is-plain-obj node-is-typedarray node-isarray node-isexe node-json-buff
  node-lodash-packages node-log-driver node-lowercase-keys node-lru-cache node-mimic-response node-minimatch node-minimist
  node-normalize-package-data node-npm-bundled node-npm-package-arg node-npmlog node-object-assign node-once node-osenv nod
  node-promise-inflight node-promise-retry node-promzard node-psl node-pump node-punycode node-quick-lru node-read node-rea
```

cron jobs

--automatisation de tâches

- **fichier** de cron à éditer
- syntaxe particulière

La méthode :

- écrire un script bash (ou autre) que vous voulez automatiser
- dans le fichier cron situé à **/etc/crontabs** : configurer le lancement du fichie à une récurrence donnée.

```
---------------- minute
| .------------- hour
| | .---------- day of month
| | | .------- month
| | | | .---- day of week
| | | | |
* * * * * /chemin/vers/ton/script
```

Exemple : tous les lundis à minuit

```
0 0 * * 1 /chemin/vers/ton/script.sh
```

le bashrc

--le fichier .bashrc

- fichier **caché**
- s'ouvre à chaque lancement du shell

voir un fichier caché : ls -a

```
amelique@ubuntuamelique: $ ls -a
.                .bashrc   Documents          extracted_data.bin   IdeaProjects
..               .bureau   .eclipse          .gitconfig           Images
.bash_history    .cache    eclipse            .gnupg               .java
.bash_logout     .config   eclipse-workspace  hash.txt             .jdks
amelique@ubuntuamelique: $ _
```

coucou

sudo nano ~/.bashrc :

Les services

--systemctl

- sudo systemctl
- sudo systemctl status {service}
- sudo systemctl enable/disable {service}
- sudo systemctl restart {service}

```
root@formation:/# systemctl
  UNIT                                                                LOAD   ACTIVE SUB       DESCRIPTION
  -.mount                                                             loaded active mounted   Root Mount
  dev-.lxc-proc.mount                                                 loaded active mounted   /dev/.lxc/proc
  dev-.lxc-sys.mount                                                  loaded active mounted   /dev/.lxc/sys
  dev-full.mount                                                      loaded active mounted   /dev/full
  dev-mqueue.mount                                                    loaded active mounted   POSIX Message Queue File System
  dev-null.mount                                                      loaded active mounted   /dev/null
  dev-ptmx.mount                                                      loaded active mounted   /dev/ptmx
  dev-random.mount                                                    loaded active mounted   /dev/random
  dev-tty.mount                                                       loaded active mounted   /dev/tty
  dev-tty1.mount                                                      loaded active mounted   /dev/tty1
  dev-tty2.mount                                                      loaded active mounted   /dev/tty2
  dev-urandom.mount                                                   loaded active mounted   /dev/urandom
  dev-zero.mount                                                      loaded active mounted   /dev/zero
  proc-cpuinfo.mount                                                  loaded active mounted   /proc/cpuinfo
  proc-diskstats.mount                                                loaded active mounted   /proc/diskstats
  proc-loadavg.mount                                                  loaded active mounted   /proc/loadavg
  proc-meminfo.mount                                                  loaded active mounted   /proc/meminfo
  proc-slabinfo.mount                                                 loaded active mounted   /proc/slabinfo
  proc-stat.mount                                                     loaded active mounted   /proc/stat
  proc-swaps.mount                                                    loaded active mounted   /proc/swaps
  proc-sys-net.mount                                                  loaded active mounted   /proc/sys/net
  proc-sysrq\x2dtrigger.mount                                         loaded active mounted   /proc/sysrq-trigger
  proc-uptime.mount                                                   loaded active mounted   /proc/uptime
  run-credentials-systemd\x2dsysctl.service.mount                     loaded active mounted   /run/credentials/systemd-sysctl.service
  run-credentials-systemd\x2dsysusers.service.mount                   loaded active mounted   /run/credentials/systemd-sysusers.service
  run-credentials-systemd\x2dtmpfiles\x2dsetup.service.mount          loaded active mounted   /run/credentials/systemd-tmpfiles-setup.service
  run-credentials-systemd\x2dtmpfiles\x2dsetup\x2ddev.service.mount   loaded active mounted   /run/credentials/systemd-tmpfiles-setup-dev.service
  sys-devices-system-cpu.mount                                        loaded active mounted   /sys/devices/system/cpu
  sys-devices-virtual-net.mount                                       loaded active mounted   /sys/devices/virtual/net
  sys-fs-fuse-connections.mount                                       loaded active mounted   FUSE Control File System
● sys-kernel-config.mount                                             loaded failed failed    Kernel Configuration File System
```

```
root@formation:/# systemctl --failed
  UNIT                                LOAD   ACTIVE SUB    DESCRIPTION
● sys-kernel-config.mount             loaded failed failed Kernel Configuration File System
● systemd-networkd-wait-online.service loaded failed failed Wait for Network to be Configured
```

```
root@formation:/# systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; disabled; preset: enabled)
     Active: active (running) since Sun 2024-03-03 21:40:25 UTC; 6 months 26 days ago
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 72578 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 72579 (sshd)
      Tasks: 1 (limit: 154144)
     Memory: 5.3M
        CPU: 225ms
     CGroup: /system.slice/ssh.service
             └─72579 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Jul 26 21:52:50 formation sshd[2980754]: Failed password for invalid user kaguya from 192.168.102.24 port 46324 ssh2
```

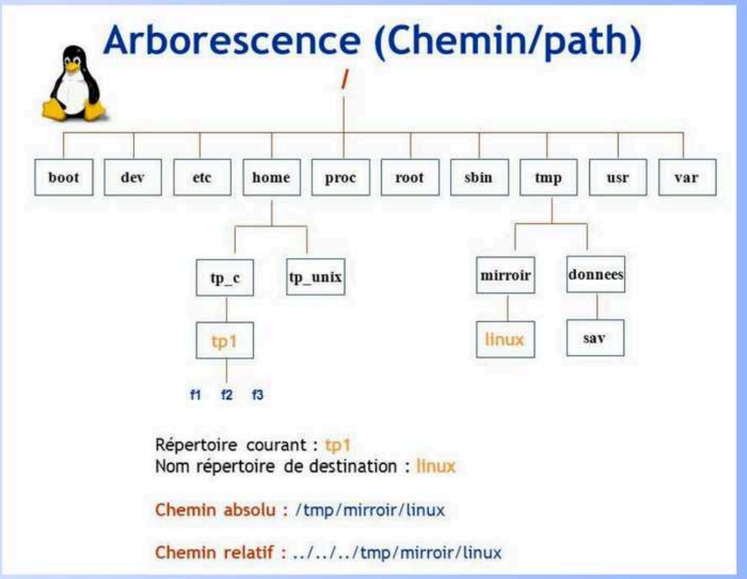
La sécurité

— quelques commandes en plus pour maîtriser la bête

Petit rappel

CHEMIN/PATH

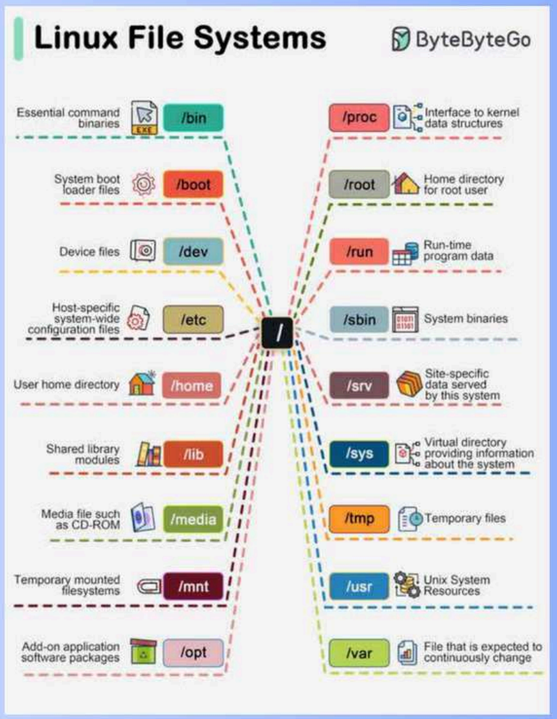
. = répertoire courant
.. = répertoire parent



LES RÉPERTOIRES

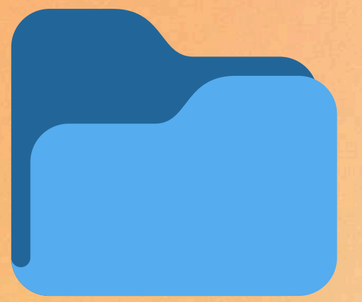
~ = répertoire de l'utilisateur

ex : pour un nom d"utilisateur "thomas",
~ = /home/thomas



on dit merci toto

Les dossiers utiles:



/var/log/

/etc/

$HOME/

Les logs

--le dossier /var/log/

- Journal de bord de chaque service
- Track les erreurs et autres info
- Bonus: tail -{nb de lignes a afficher} /...

tail -20 /var/log/... :

```
root@ZenbookOled-Ambroise:~# tail -20 /var/log/boot.log.1
[  OK  ] Reached target Network.
         Starting Network Manager Wait Online...
         Starting containerd container runtime...
         Starting CUPS Scheduler...
         Starting MySQL Community Server...
         Starting A high performanc… and a reverse proxy server...
         Starting OpenVPN service...
         Starting Advanced key-value store...
         Starting Permit User Sessions...
[  OK  ] Started Unattended Upgrades Shutdown.
[  OK  ] Finished OpenVPN service.
[  OK  ] Finished Permit User Sessions.
[  OK  ] Started CUPS Scheduler.
[  OK  ] Started Thunderbolt system service.
[  OK  ] Started Modem Manager.
[  OK  ] Started Disk Manager.
         Starting Terminate Plymouth Boot Screen...
         Starting Set console scheme...
[  OK  ] Started Dispatcher daemon for systemd-networkd.
[  OK  ] Finished Set console scheme.
```

“root” ?

Les utilisateurs

- **useradd** -> Ajouter un utilisateur
- **passwd** -> Changer le mot de passe
- **usermod** -> Pleins de trucs

- **root** -> Super-utilisateur, tout les droits

Average Linux enjoyer

Side-note

Le fichier passwd

- Tout est un fichier au bout du compte
- Mot de passes pas en clair =(

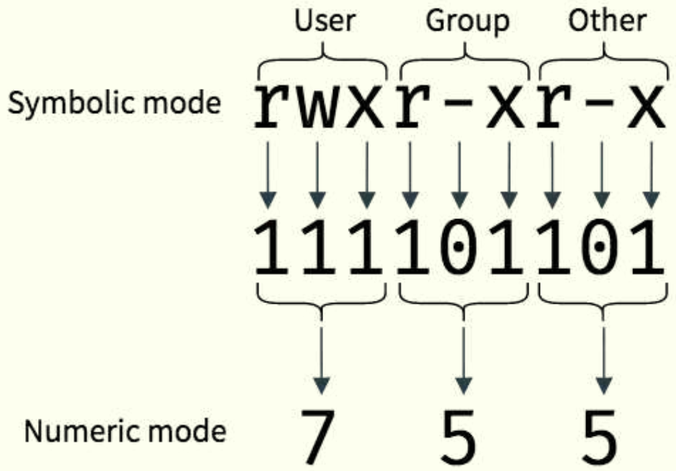
cat /etc/passwd

```
root@ZenbookOled-Ambroise:~# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
```

Les permisions

- **Les permissions:**
  - **r: read**
  - **w: write**
  - **x: execute**
- **ex: chmod 777 {fichier}**

Les codes de permissions :



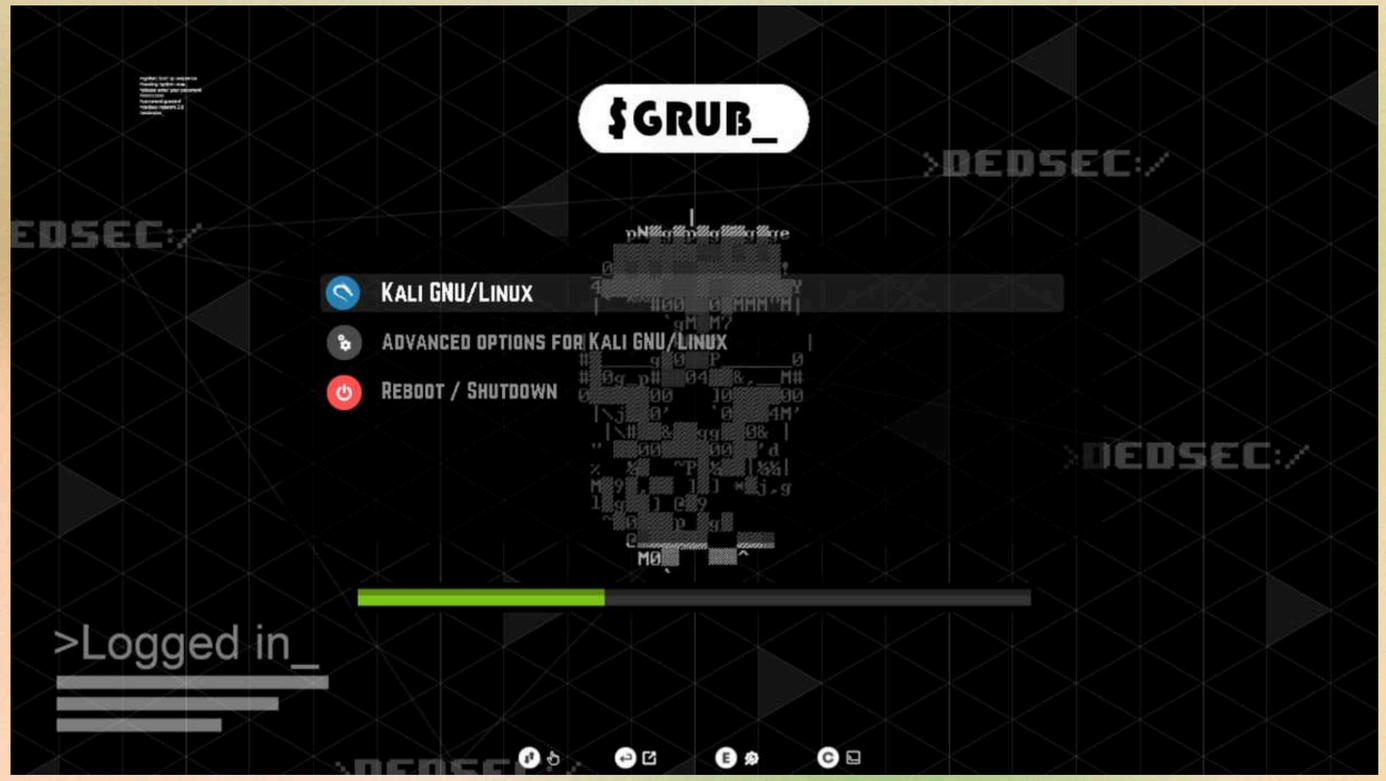
Afficher les permissions :

```
root@ZenbookOled-Ambroise:~# ls -l
total 4
drwx------ 7 root root 4096 mai   26 16:39 snap
-rw-r--r-- 1 root root    0 mai   15 17:01 sudo
```

- “d” / “-” -> dossier ou fichier
- rw-r--r-- -> paramètres de permissions

BONUS : GRUB

A faire post-TP



- Aller sur : gnome-look.org/browse?cat=109&page=1&ord=latest
- Choisir le thème et télécharger la version voulue
- “cd /tmp” puis “tar xvf ~/Téléchargements/{nom du fichier}”
- sudo ./install.sh

Si ça ne marche pas -> malekal.com/theme-grub-personnaliser-grub/

BONUS 2 : ALIAS

A faire post-TP

```
alias minetvpn="sudo openvpn ~/Clés_VPN_important/config.conf"
alias maj="sudo apt update && sudo apt upgrade"
```

Pour mettre à jour plus facilement :

- nano ~/.bashrc
- Ajouter *alias maj="sudo apt update && sudo apt upgrade"*
- *F*aire Ctrl+X, taper “o”, et enter, on a bien sauvegardé le fichier

Et plus qu’à écrire “maj” pour mettre à jour tout son système rapidement !

Votre PC et le reste du monde

— quelques commandes en plus pour maîtriser les autres bête

comment fonctionne minet ?

Pourquoi on utilise un serveur ?

Mais dis moi Jamy :

Yakwa sur un serveur ?

ça ressemble à quoi ?

comment on s’en sert ?

pourquoi Thomas est si blond ?

Un exemple :

minet.net

De quoi on a besoin pour créer minet.net ?

Un exemple : minet.net

De quoi on a besoin pour créer minet.net ?

Du stockage = mémoire morte

stocker les images, le système d’exploitation (apache), la structure visuelle = le code du site

Un exemple : minet.net

De quoi on a besoin pour créer minet.net ?

Du stockage = mémoire morte

stocker les images, le système d'exploitation (apache), la structure visuelle = le code du site

De la RAM = mémoire vive

"l'intelligence" du site

Un exemple : minet.net

De quoi on a besoin pour créer minet.net ?

Du stockage = mémoire morte

stocker les images, le système d'exploitation (apache), la structure visuelle = le code du site

De la RAM = mémoire vive

"l'intelligence" du site

De la connectivité

pour mettre minet.net en ligne

Un exemple : minet.net

De quoi on a besoin pour créer minet.net ?

Du stockage = mémoire morte

stocker les images, le système d'exploitation (apache), la structure visuelle = le code du site

De la RAM = mémoire vive

"l'intelligence" du site

De la connectivité

pour mettre minet.net en ligne

De l'alimentation

tout ça, ça fonctionne pas gratuitement

Un exemple : minet.net

De quoi on a besoin pour créer minet.net ?

Du stockage = mémoire morte

stocker les images, le système d'exploitation (apache), la structure visuelle = le code du site

De la RAM = mémoire vive

"l'intelligence" du site

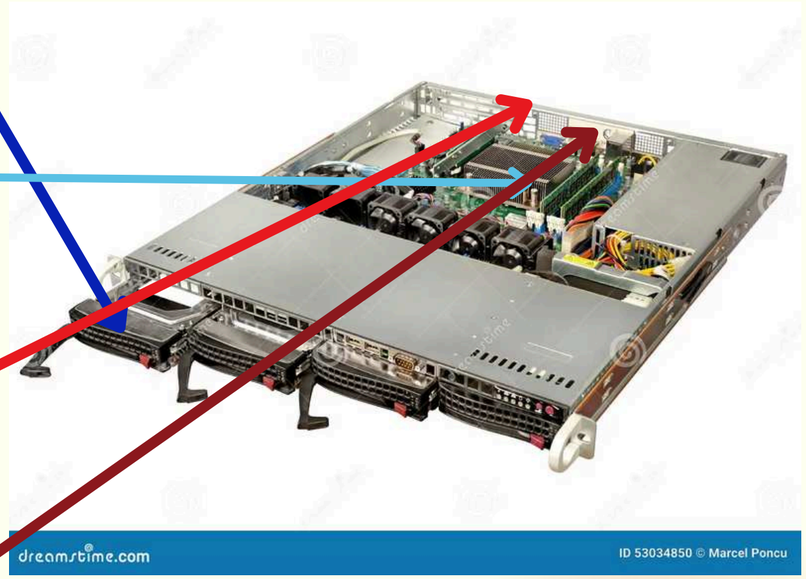
De la connectivité

pour mettre minet.net en ligne

De l'alimentation

tout ça, ça fonctionne pas gratuitement

boum le serveur



minet = 133 services comme minet.net

minet = 133 serveurs ???

Et là

C’est le drame !

un serveur = 4000 euros

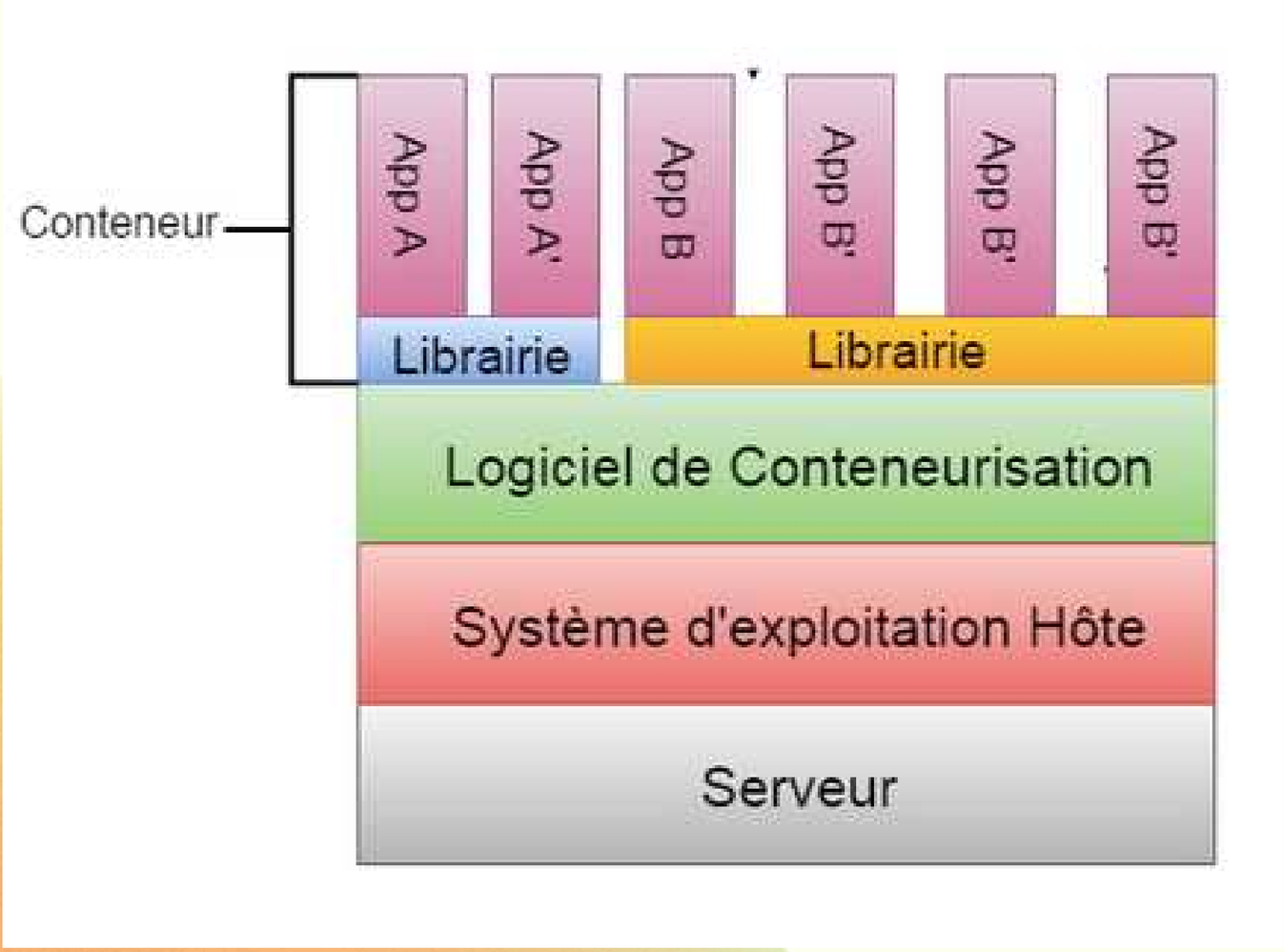
minet = 532 000 euros ??

Evan en sueur :/

mais POURQUOI Thomas est si blond ?

La solution :
la conteneurisation

Simuler des serveurs à l'interieur du serveur



faut descendre voir le serveur pour travailler sur les VMs et CTs ?

en salle serveur il fait froid, baisse la clim

Et là

T’as la flemme

flemme d’aller au U7 voir Altaïr

les minetiens en sueur :/

Thomas fait qq chose stp

commande ssh

commande de la flemme

-->accès **à distance** au shell d'une machine
-->pour les serveurs et les VMs, les switch...

ssh utilisateur@ip-de-la-machine

exemple avec notre serveur de prod sirius **:**

```
amelique@ubuntuamelique: $ ssh amelique@192.168.102.86
Linux sirius 6.5.13-1-pve #1 SMP PREEMPT_DYNAMIC PMX 6.5.13-1 (2024-02-05T13:50Z) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Sep 29 13:15:43 2024 from 192.168.102.24
amelique@sirius: $ cd /
amelique@sirius: $ ls
bin  boot  dev  etc  home  lib  lib64  lost+found  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
amelique@sirius: $ _
```

Annexes

La solution :
la conteneurisation

Simuler des serveurs à l'interieur du serveur

Deux manières de le faire :

Après avoir installé un logiciel de conteneurisation = hyperviseur sur le serveur :

Créer des **Machines Virtuelles** (VM)

OU

Créer des **Containers** (CT)

(ou un mélange des deux)

La solution :
la conteneurisation

Simuler des serveurs à l'interieur du serveur

PROXMOX

docker

Deux manières de le faire :

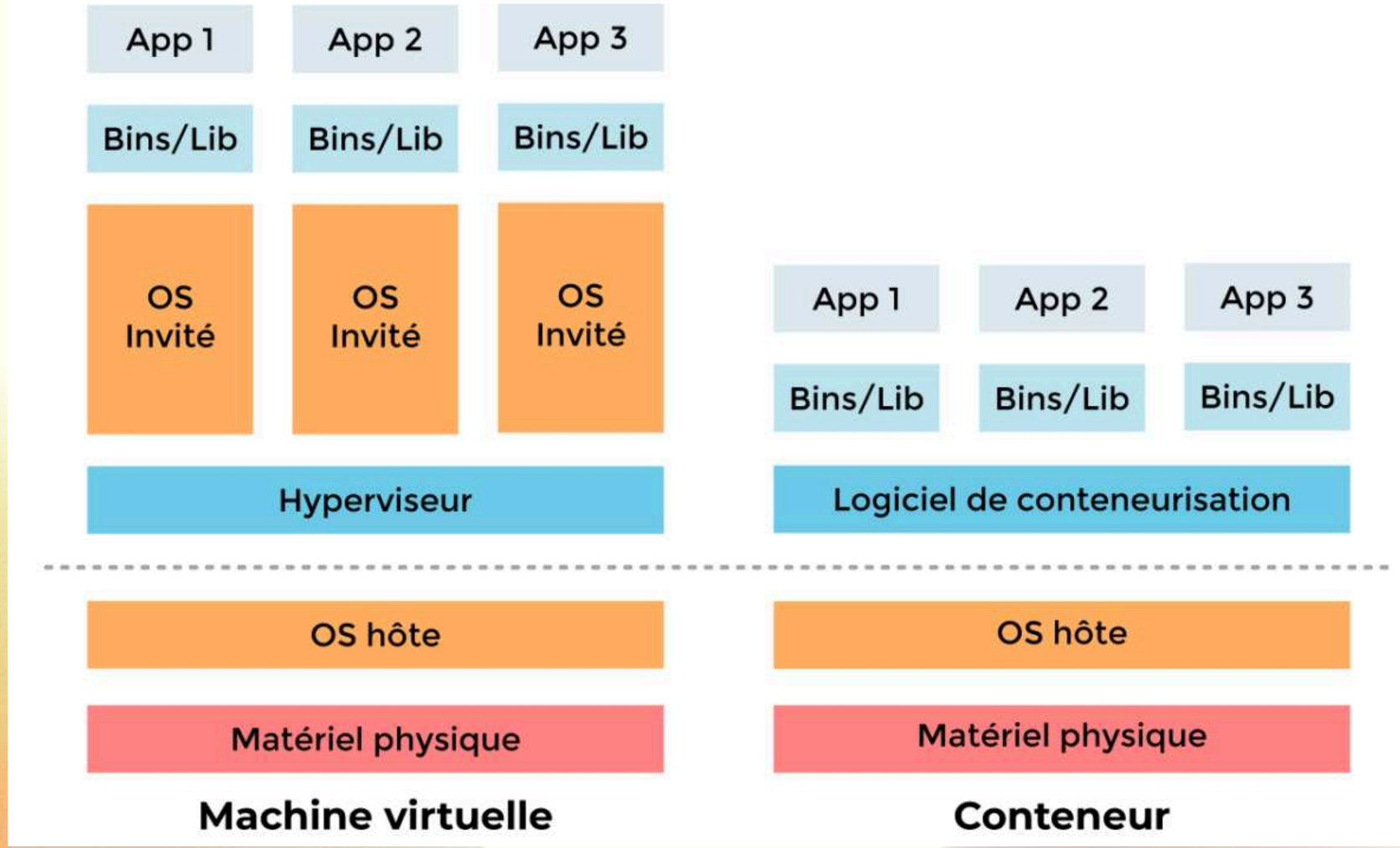
Après avoir installé un logiciel de conteneurisation = hyperviseur sur le serveur :

Créer des **Machines Virtuelles** (VM)

OU

Créer des **Containers** (CT)

(ou un mélange des deux)



La solution :
la conteneurisation

Simuler des serveurs à l'interieur du serveur

PROXMOX

docker

Deux manières de le faire :

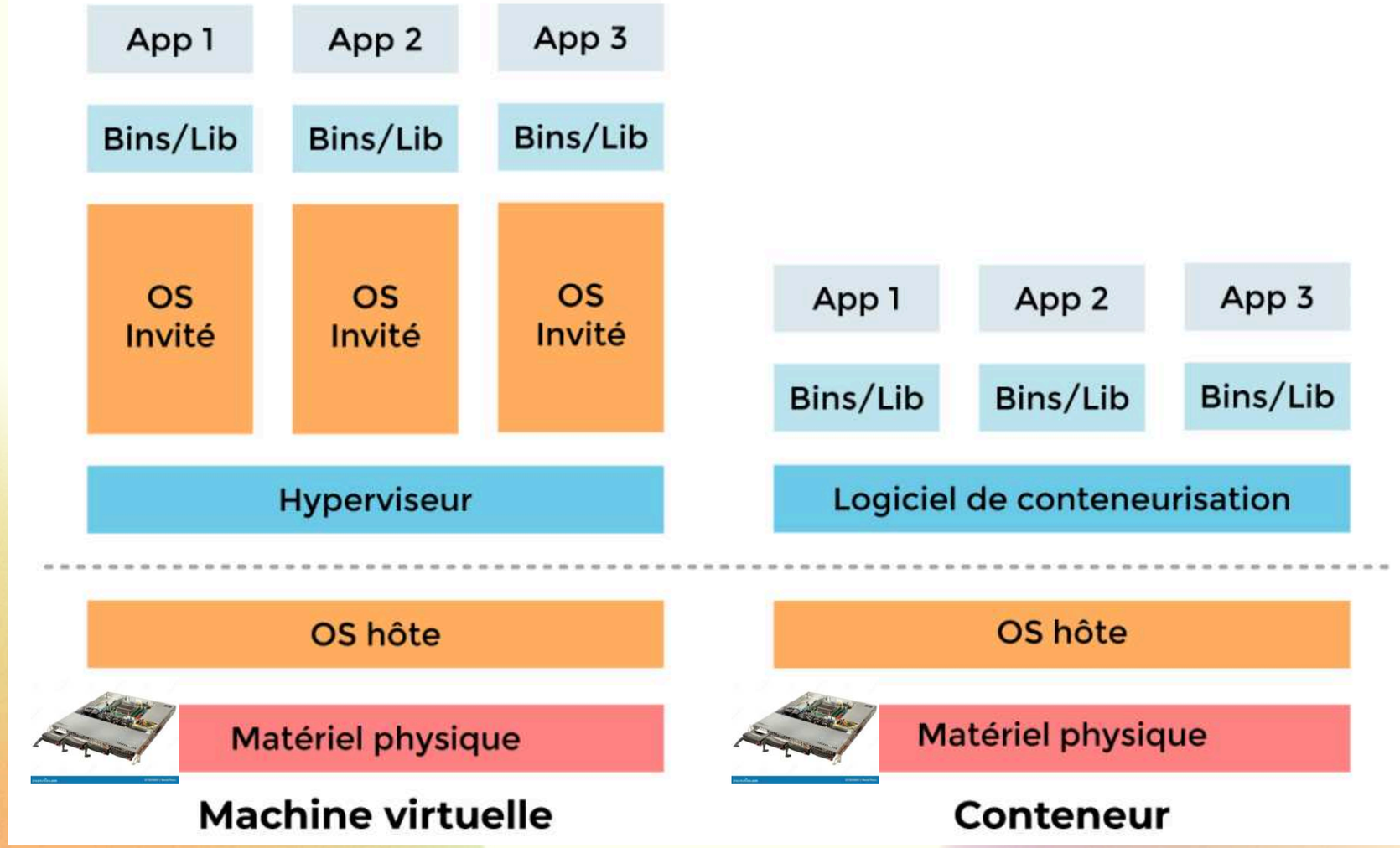
Après avoir installé un logiciel de conteneurisation = hyperviseur sur le serveur :

Créer des **Machines Virtuelles** (VM)

OU

Créer des **Containers** (CT)

(ou un mélange des deux)



La solution :
la conteneurisation

Simuler des serveurs à l'interieur du serveur

Deux manières de le faire :

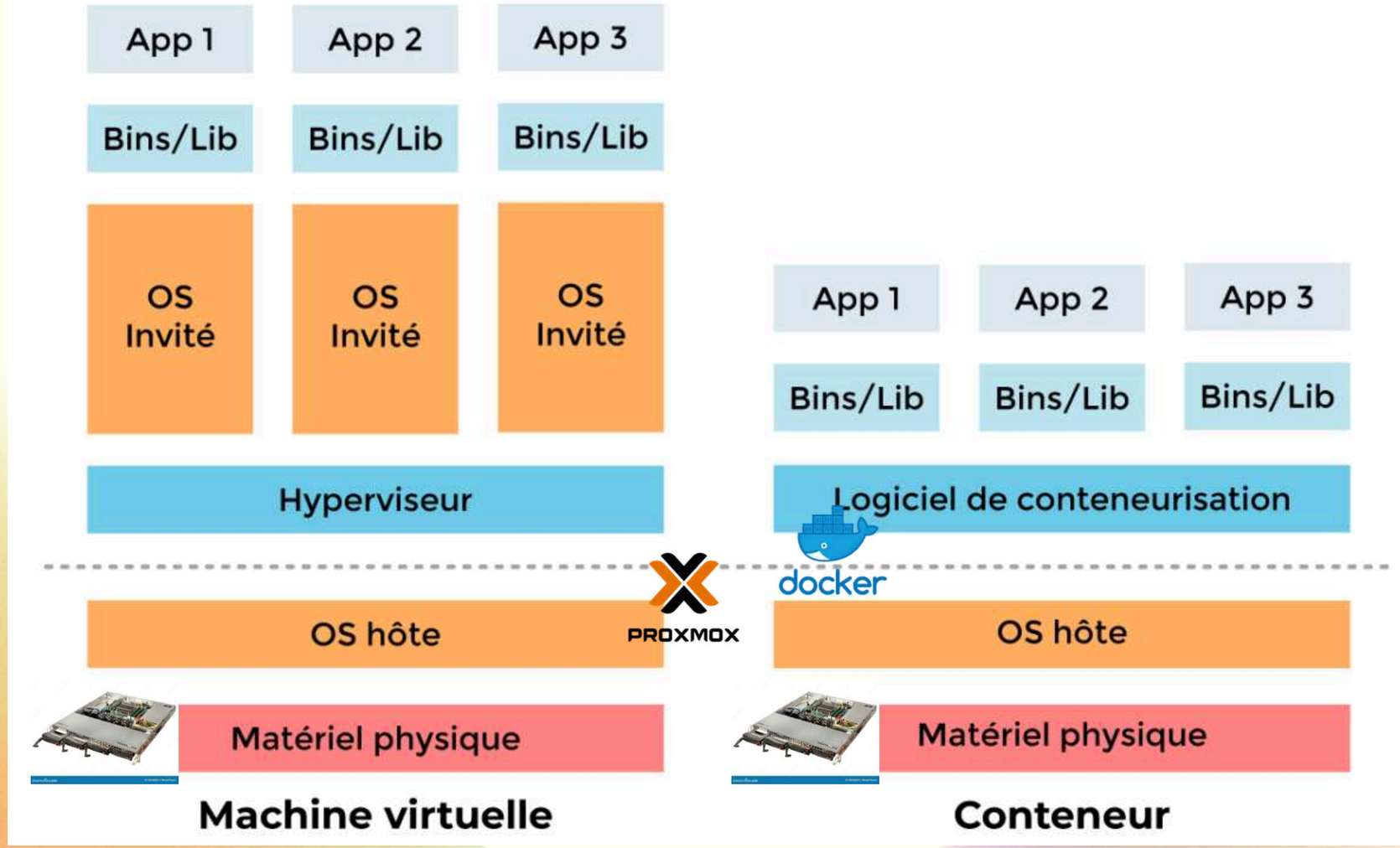
Après avoir installé un logiciel de conteneurisation = hyperviseur sur le serveur :

Créer des **Machines Virtuelles** (VM)

OU

Créer des **Containers** (CT)

(ou un mélange des deux)



La solution :
la conteneurisation

Simuler des serveurs à l'interieur du serveur

PROXMOX

docker

Deux manières de le faire :

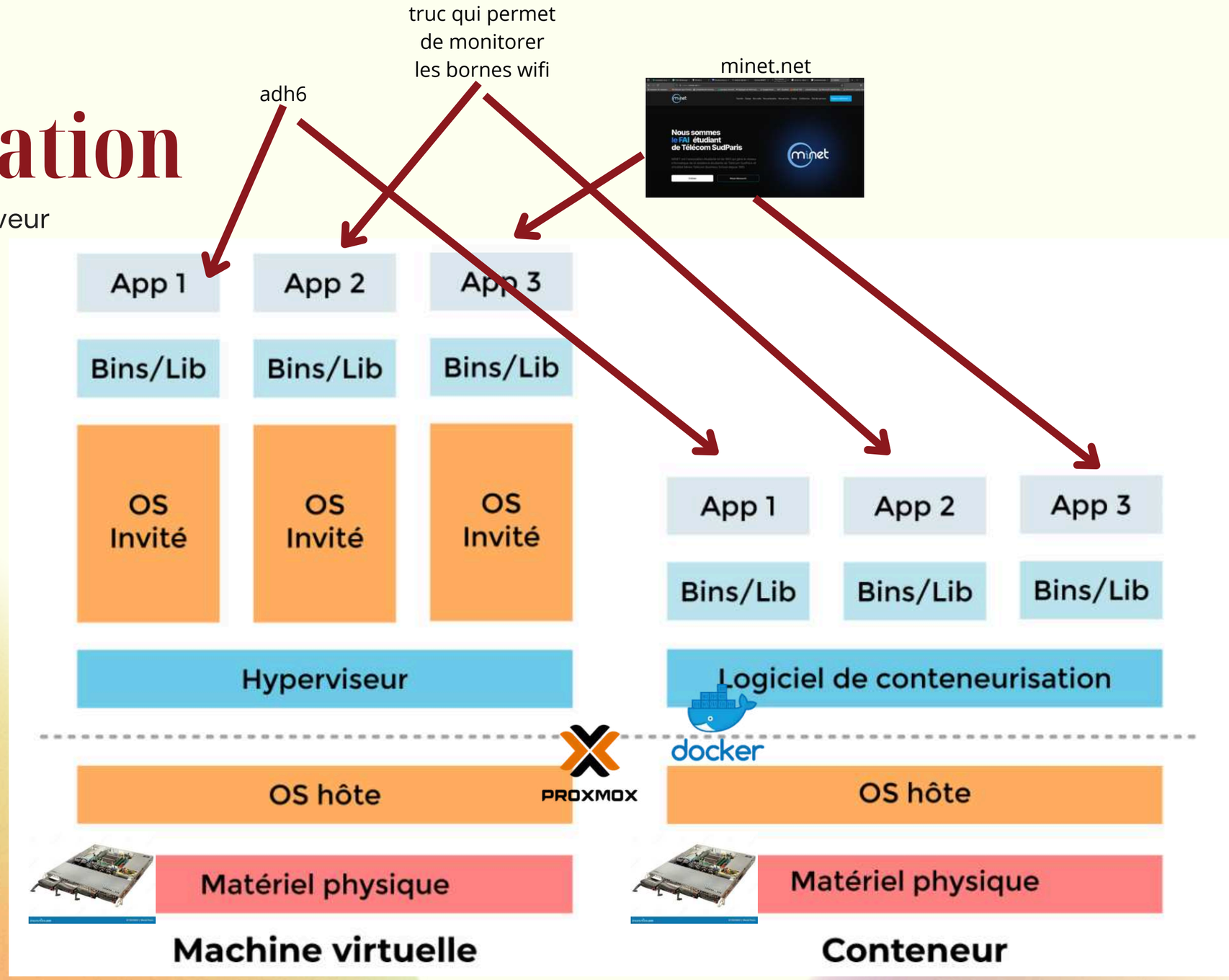
Après avoir installé un logiciel de conteneurisation = hyperviseur sur le serveur :

Créer des **Machines Virtuelles** (VM)

OU

Créer des **Containers** (CT)

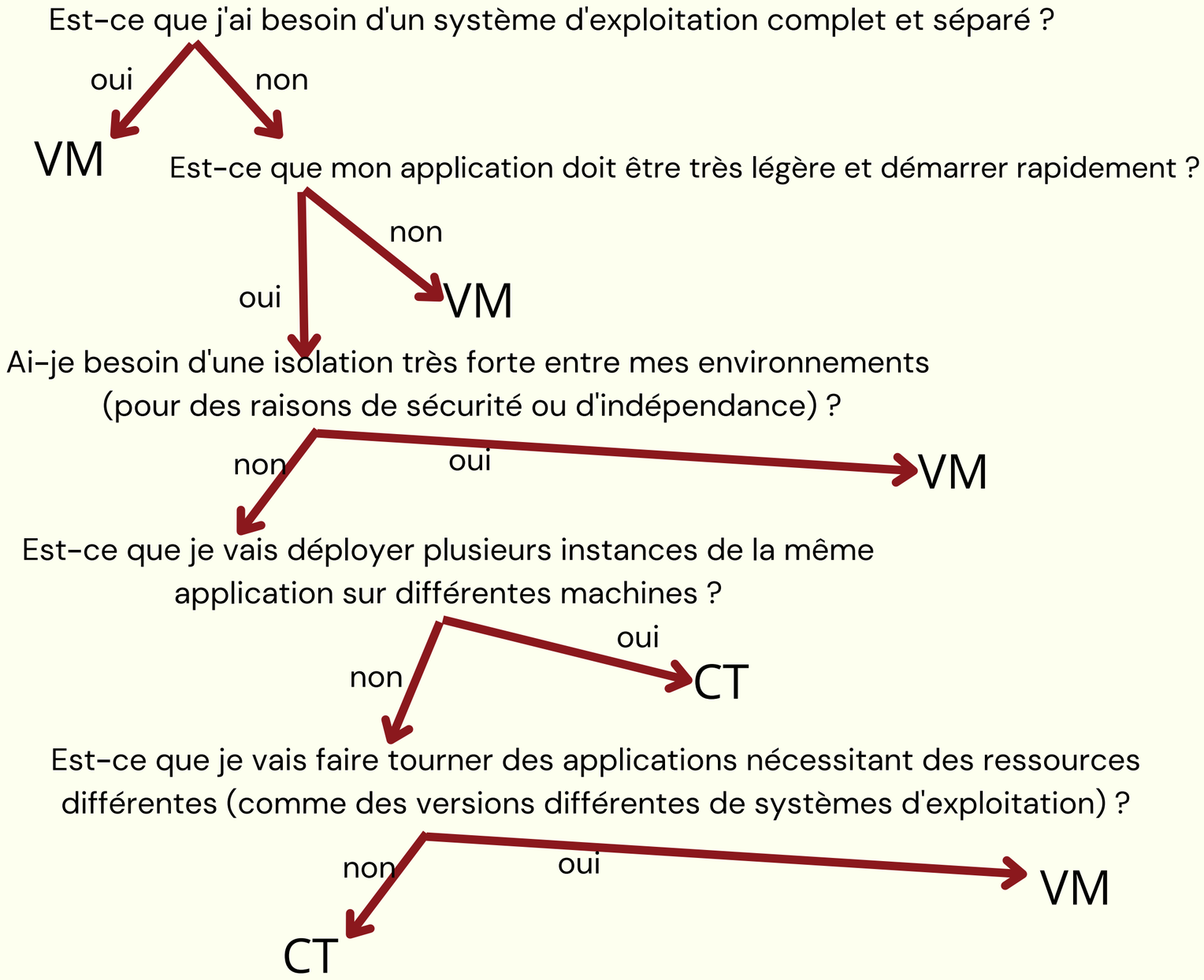
(ou un mélange des deux)



plutôt VMs ou CTs ?

faire son choix

-->généralement, dans les serveurs, c’est un mélange de VM et CT



Conclusion :

VM = environnement complet, sécurisé et isolé, ou si tu fais tourner plusieurs systèmes d'exploitation différents.

CT = quelque chose de plus léger, rapide, et que tu n'as pas besoin d'un système d'exploitation entier pour chaque application.

et pour les CTs ?

CTs != ssh

-->ssh au serveur qii stocke la VM
-->sudo enter la VM

exemple avec minet.net sur sirius**:**

```
amelique@ubuntuamelique: $ ssh amelique@192.168.102.86
Linux sirius 6.5.13-1-pve #1 SMP PREEMPT_DYNAMIC PMX 6.5.13-1 (2024-02-05T13:50Z) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Sep 29 13:15:43 2024 from 192.168.102.24
amelique@sirius: $ sudo enter minet.net
[sudo] password for amelique:
```

```
PVE enter
Choose one of the following LXCs :

local:105 portail.minet.net
local:109 formation.minet.net
local:116 bbq.minet.net
local:122 imagine.minet.net
local:136 zabbix3bdd.minet.net
local:146 webmail.minet.net
local:147 kea.minet.net
local:155 etherpad.minet.net
local:167 webmail.minet.net
local:168 vote.minet.net
```

selection de la CT à utiliser

Formation
Linux avancé

1
Formation

1.1 Gestion des utilisateurs

Commandes à mener sur son PC

— Créez trois utilisateurs : ‘user1‘, ‘user2‘, et ‘adminuser‘.

— Ajoutez ‘adminuser‘ au groupe ‘admin‘.

— Configurez les permissions suivantes :
  — ‘adminuser‘ a tous les droits sur ‘/data/secret‘.
  — ‘user1‘ a accès en lecture seule au répertoire ‘/data/secret‘.
  — ‘user2‘ a des droits en lecture et écriture sur ‘/data/secret‘.

1.2 Script

Faire un script "coucou.sh" qui affiche "coucou" dans le terminal.
(N'oubliez pas de le rendre exectuable)

1.3 Gestion processeur

— Installer "hotp" avec "sudo apt install htop"
— Faites la commande "yes MiNET"
— Lancer "htop" et essayez de kill le processus

1.4 Gestion réseau

Vous pouvez afficher les ports réseaux utilisé avec :
ss -tuna | grep ESTAB | grep -v '127.0.0.1

Ecrivez un script pour afficher les ports puis automatisez son fonctionnement avec un cron.

1.5 Défi final

Petit chall donnez la couleur de caleçon de Thomas à l'aide de l'archive Thomas.tar qui publiée sur le discord.