

# Introduction aux systèmes GNU/Linux

S23E03 inetdoc.net



Philippe Latu / Université Toulouse 3

Document sous licence GNU FDL v1.3  
<http://www.gnu.org/licenses/fdl.html>

# Objectifs de la séance

- Configurer un Système GNU/Linux
  - **Utiliser** les ressources du shell Bash
  - **Identifier** & **gérer** les processus
  - **Gérer** les permissions sur les fichiers et répertoires
  - **Différencier** les processus entre système hôte & conteneur
- Manipuler sur machine virtuelle & conteneur
  - Installer & configurer des services Internet
  - Repérer les propriétaires & les propriétés des processus correspondant

# Bash → Shell et langage de commandes

- 1979 → SysV version 7 → Bourne Shell
- 1989 → GNU BASH → Bourne-again Shell → scripts
  - Interpréteur de commandes
    - IEEE POSIX P1003.2/ISO 9945.2 Shell and Tools standard
  - Environnement de développement
    - Algorithmique de base : conditions, boucles, etc.
    - Fonctions et alias
    - Arithmétique et tableaux
    - Manipulations de chaînes de caractères
  - Documentation shell Bash
    - Advanced Bash Scripting
      - <http://www.tldp.org/LDP/abs/html/>

```
$ man sh
```

Documentation  
syntaxe

```
$ help
```

Documentation  
commandes internes

# Bash → séquençement entre les commandes

- Éditions et corrections en ligne de commande
  - Auto-correction → commande `shopt -s dirspell`
  - Auto-completion → touche tabulation
  - Historique et rappel des commandes antérieures
    - Commande `history`
    - Séquences de touches
      - Ctrl+R, Ctrl+A, Ctrl+E
      - Shift+PageUp, Shift+PageDown, flèches haut et bas

# Bash → séquençement entre les commandes

- Séquençement entre les opérations
  - Tubes ou *pipes* → |
  - Redirections → > → >>
  - Enchaînements logiques → ; → && → ||
  - Gestionnaire de fenêtres ou de tâches console
    - Tmux, byobu, VSCode
  - Commandes internes → jobs, suspend

# Bash → exemple de script

- Script `checkup.sh` : gestion des machines virtuelles

```
#!/bin/bash
```

Shebang

```
set -e
```

Affectation directe d'une variable

```
GreenOnBlack='\E[32m'
```

```
echo "~> Check-up gestion des machines virtuelles"
```

```
# Arborescence
```

Test de la présence d'un lien symbolique

```
if [ ! -L ~/masters ]; then
```

```
    echo "Création du lien vers le catalogue des images de machines virtuelles."
```

```
    ln -s /var/cache/kvm/masters ~
```

```
fi
```

```
if [ ! -d ~/vm ]; then
```

Test de la présence d'un dossier

```
    echo "Création du dossier des images de machines virtuelles."
```

```
    mkdir ~/vm
```

```
    ln -s ~/masters/scripts ~/vm/
```

```
fi
```

```
echo -e "${GreenOnBlack}~> Arborescence prête."
```

```
tput sgr0
```

# Shell Bash

- Script `checkup.sh` : gestion des machines virtuelles

```
# Analyse des fichiers image
```

Nom du fichier de rapport d'analyse

```
count=0
```

```
listFilename="$HOME/image-list-info.json"
```

Boucle de traitement des fichiers image

```
echo -n "[" > $listFilename
```

```
imageList=$(find ~ -type f \( -iname \*.qcow2 -o -iname \*.raw \) -printf '%p ')
```

```
for file in $imageList
```

Recherche du nom de fichier dans les processus actifs

```
do
```

```
    echo "$((++count)) : $file"
```

```
    if [ -z "$(pgrep -u ${USER} | grep -o "${file##*/}\ ")" ]; then
```

```
        qemu-img info --output=json --backing-chain $file | tr -d '[' >> $listFilename
```

```
        sed -i '/^$/d' $listFilename && sed -i '$s/$/\,/ ' $listFilename
```

```
    fi
```

Suppression ligne vide

Ajout virgule en fin de fichier

```
done
```

```
sed -i '$ s/,/\n]/g' $listFilename
```

```
echo -e "${GreenOnBlack}~> $count fichier(s) image(s) présent(s)."
```

```
tput sgr0
```

# Shell Bash

- Script `checkup.sh` : gestion des machines virtuelles

```
# Liste des machines virtuelles actives
IFS=$'\n'
count=0
vmArray=( $(ps aux | grep ${USER} | grep 'qemu-system-x86_64\ ' || true) )
if [ ${#vmArray[@]} -gt 0 ]; then
  for vm in "${vmArray[@]}"
  do
    echo -n "$((++count)) : "
    echo $vm | grep -Po '(?<=-name\ ).*(?=-m)' | tr -d '\n'
    echo -n "utilise le cordon "
    echo $vm | grep -Po 'tap\d{1,3}'
  done
fi
echo -e "${GreenOnBlack}~> $count machine(s) virtuelle(s) active(s)."
tput sgr0

exit 0
```

Redéfinition du délimiteur de champs

Création d'un tableau pour chaque processus

Boucle de parcours des éléments du tableau des processus

Nom de machine virtuelle

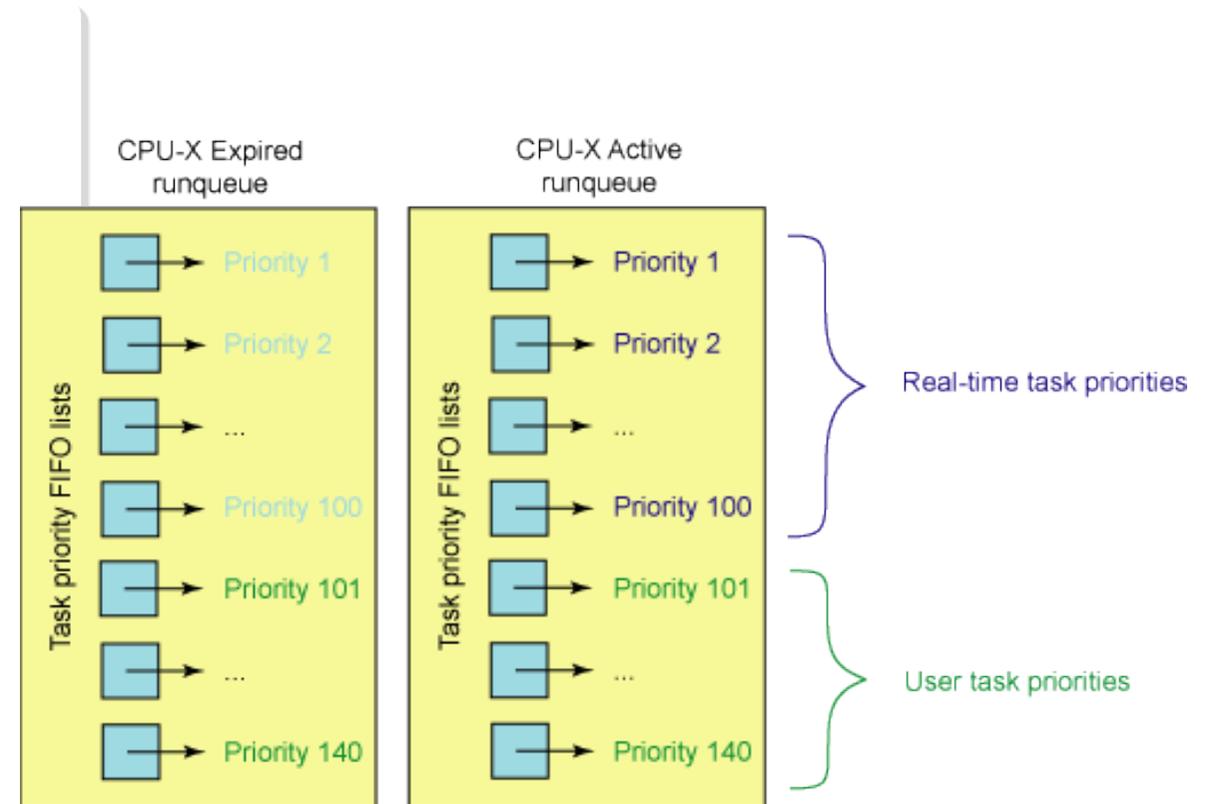
Numéro du cordon de brassage

# Bash → Exercices

- Créer et tester le script `checkup.sh`
- Comment lister l'historique des commandes ?
- Quel est l'effet de la séquence de touche `Ctrl+E` ?
- Comment rappeler une commande dans l'historique ?
- Quel est l'effet de la séquence de touche `Ctrl+D` ?
- Comment accéder à la documentation de la commande intégrée `shopt` ?
- Comment activer l'auto-correction à l'ouverture d'un shell Bash ?
  - [https://www.gnu.org/software/bash/manual/html\\_node/The-Shopt-Builtin.html](https://www.gnu.org/software/bash/manual/html_node/The-Shopt-Builtin.html)
- Comment activer la personnalisation Oh My Bash ?
  - <https://github.com/ohmybash/oh-my-bash>

# Processus → programme en cours d'exécution

- Fonction Unix de base  
Partage des ressources du système entre différents programmes
- Ordonnanceur (*Scheduler*)  
Attribution des tranches de temps processeur en fonction du type de tâche
- Fonctions multi-tâches préemptives du noyau Linux
  - Planification de l'exécution des processus
  - Contrôle au début et à la fin de chaque tranche de temps processeur



# Processus → quelles propriétés ?

```
0[ |
1[ |
2[ |
3[ |||
Mem[ |||||
Swp[ |

0.7%] Tasks: 32, 84 thr, 134 kthr; 1 running
0.0%] Load average: 0.83 0.29 0.10
0.0%] Uptime: 00:00:48
2.0%]
536M/1.89G]
0K/4.00G]

Main I/O
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
563 jenkins 20 0 4099M 356M 28080 S 2.7 18.4 0:17.96 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.w
1150 etu 20 0 8920 5120 3584 R 1.3 0.3 0:00.17 htop
707 jenkins 20 0 4099M 356M 28080 S 0.7 18.4 0:00.04 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.w
1 root 20 0 163M 12304 8976 S 0.0 0.6 0:01.15 /sbin/init
313 root 19 -1 41160 19456 18560 S 0.0 1.0 0:00.14 /lib/systemd/systemd-journald
350 root RT 0 282M 27264 8832 S 0.0 1.4 0:00.02 /sbin/multipathd -d -s
353 root 20 0 26976 6784 4736 S 0.0 0.3 0:00.07 /lib/systemd/systemd-udev
356 root 20 0 282M 27264 8832 S 0.0 1.4 0:00.00 /sbin/multipathd -d -s
357 root RT 0 282M 27264 8832 S 0.0 1.4 0:00.00 /sbin/multipathd -d -s
```

## ▪ Gestion des processus

- Qui est le propriétaire d'un processus ?
- Quelles sont les ressources utilisées par un processus ?
- Comment changer le niveau de priorité d'un processus ?
- Comment tuer un processus défectueux ?

# Processus → quelles commandes ?

- Commandes de gestion des processus
  - Qui, quoi, combien : `ps`, `w`, `top`, `htop`, `iostat`
  - Priorités : `nice`, `renice`, `ionice`
  - Signalisation et arrêt : `kill`, `killall`
  - Utilisation mémoire : `free -m`, `vmstat -w -a -S M`, `vmstat -s`

# Processus → exercices

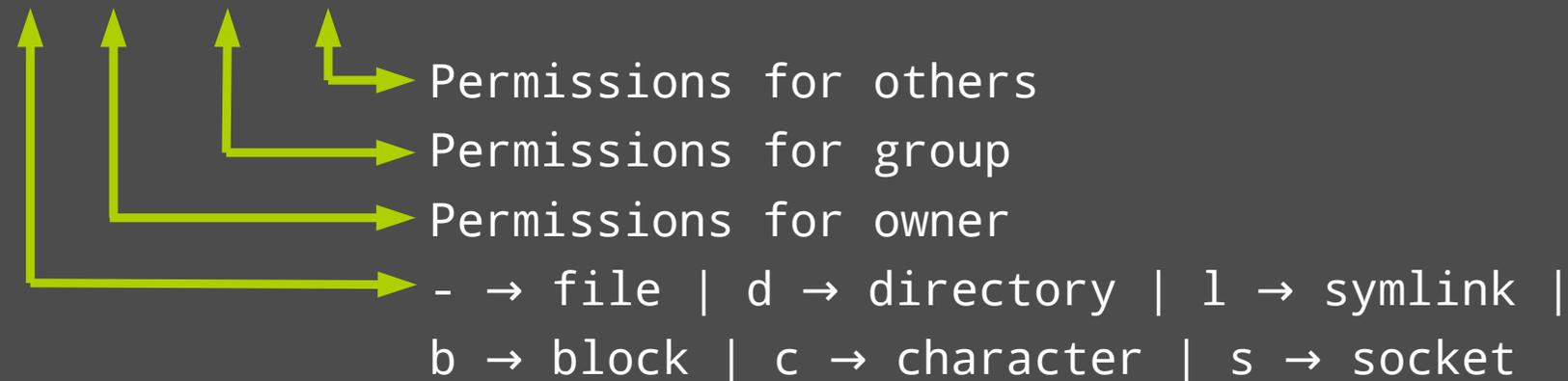
- Commande `ps`
  - Comment visualiser les processus, les propriétaires et les terminaux ?
  - Quelle est la signification des 4 options dans la commande 'ps faux' ?
- Commandes `kill` et `killall`
  - Quelle est la signification du terme signal ?
  - Comment relancer un processus ?
  - Comment tuer un processus «en force» ?
- Processus ou services inutiles
  - Comment supprimer un service inutile de façon permanente ?
  - Comment caractériser le gain en occupation mémoire correspondant ?



# Permissions sur le système de fichiers

- Masque des permissions de base → 10 indicateurs
  - Partant de la gauche
    - Premier indicateur → nature de l'objet
      - fichier, répertoire, périphérique ou socket Unix
    - Autres indicateurs → droits
      - lecture, écriture, exécution
      - Propriétaire, groupe et autre

d rwx rwx rwx



# Permissions sur le système de fichiers

- Visualisation/Édition du masque des permissions
- Commandes usuelles
  - `ls` → visualisation
  - `chown`, `chgrp` → changement de propriétaire ou de groupe
  - `chmod` → changement de masque
  - `umask` → masque utilisateur utilisé par défaut lors de la création d'objets
- Codage des permissions
  - Notation littérale
    - `r` → read → droit de lecture
    - `w` → write → droit d'écriture
    - `x` → execute → droit d'exécution
  - Notation en octal
    - `r` →  $2^2$  → 4
    - `w` →  $2^1$  → 2
    - `x` →  $2^0$  → 1

```
$ touch emptyfile
$ ls -l emptyfile
-rw- r-- r-- 1 etu etu 0 mai 16 11:20 emptyfile
420 400 400
6 4 4 → 644
$ chmod +x emptyfile
-rwx r-x r-x 1 etu etu 0 mai 16 11:20 emptyfile
421 401 401
7 5 5 → 755 → chmod +x = chmod 755
```

# Permissions sur le système de fichiers

- Applications
  - À l'aide de la commande `'ls'`, donner un exemple de :
    - programme exécutable
    - lien symbolique
    - périphérique en mode caractère
    - périphérique en mode bloc
    - socket Unix
  - Permissions sur les fichiers et répertoires
    - Donner les valeurs numériques des masques d'un fichier de données et d'un programme
    - Quel est l'effet de l'instruction `'umask 0027'` ?
  - Shell script Bash «Hello, World!»
    - Créer le fichier script `hello.sh` et le rendre exécutable

```
#!/bin/bash  
echo "Hello, World!"
```

# Permissions sur le système de fichiers

- Masque étendu : 3 bits supplémentaires → extension des permissions
  - **SUID** : Set User ID bit
  - **SGID** : Set Group ID bit
  - directory **Sticky bit**
- Ces 3 bits prennent la place du bit d'exécution **x**
  - Pour le propriétaire du fichier
    - **s** indique qu'il a aussi le droit d'exécution
    - **S** indique qu'il n'a pas le droit d'exécution
  - Pour le groupe du fichier
    - **s** indique qu'il a aussi le droit d'exécution
    - **S** indique qu'il n'a pas le droit d'exécution
  - Directory Sticky bit
    - Utile pour les répertoires partagés
    - Un utilisateur ne peut effacer que les fichiers qu'il a créé

# Permissions sur le système de fichiers

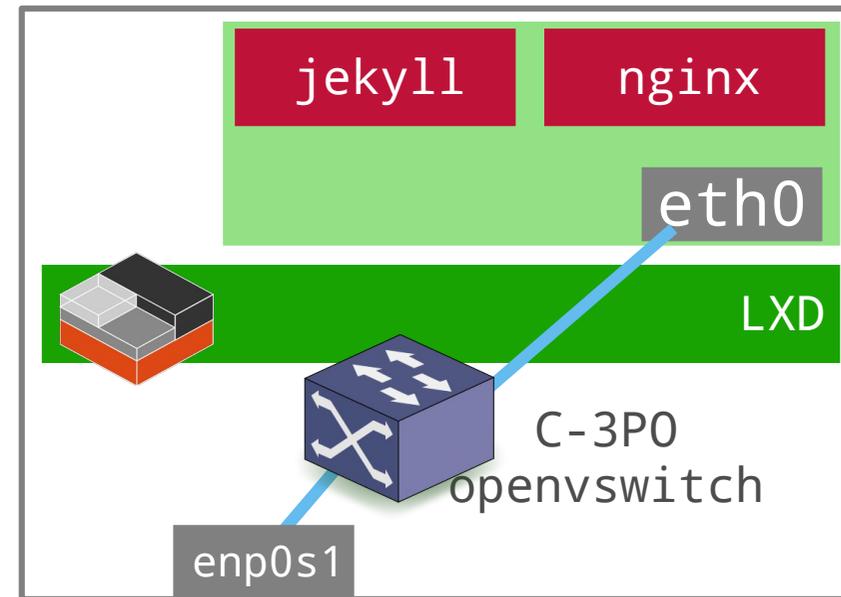
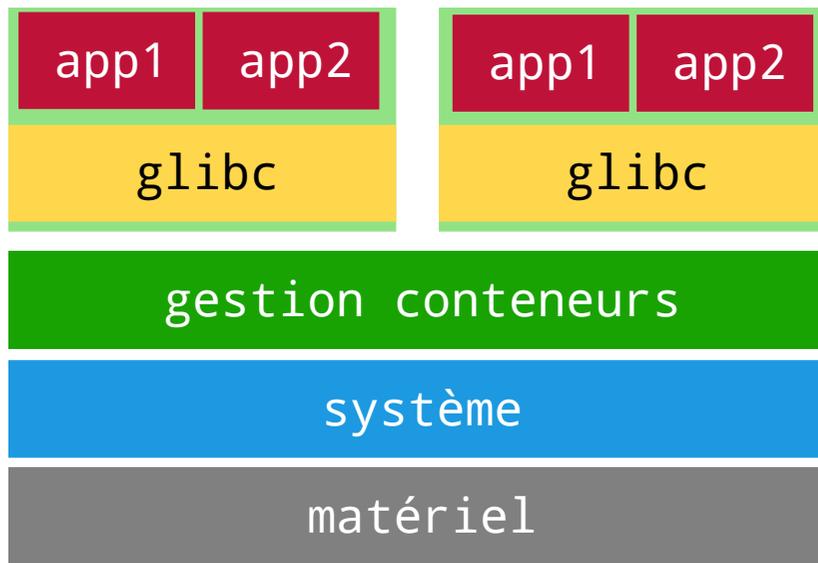
## ▪ Applications

- Quel est le rôle du masque étendu pour les objets suivants ?  
/usr/bin/passwd et /tmp
- Comment configurer un environnement de développement Web statique ?
  - Installer le paquet `task-web-server`
  - Fixer la valeur du masque utilisateur à `0027`
  - Créer le répertoire `/var/www/newhtml` avec le masque `rwxr-s---`  
L'utilisateur propriétaire doit être `etu` et le groupe propriétaire `www-data`
  - Créer un fichier `index.html` dans ce nouveau répertoire
  - Ajouter une entrée `newsite` dans `/etc/hosts` avec l'adresse `127.0.0.2`
  - Copier le fichier `/etc/apache2/sites-available/000-default.conf` en `newsite.conf`
  - Activer le nouveau site avec la commande `a2ensite`  
Recharger la configuration du service
  - Tester l'ouverture de la page Web et retrouver les traces dans les journaux du service Web

# Application → Cas jekyll

## ▪ Objectifs

- Transformer le système hôte en commutateur/routeur réseau
- Installer de gestionnaire de conteneurs LXD
- Créer un conteneur 'jekyll'



# Application → Cas jekyll

- Installation des paquets, du service snapd et de LXD
  - Installer les paquets openvswitch-switch et snapd
  - Installer de gestionnaire de conteneurs LXD
  - L'utilisateur 'etu' doit appartenir au groupe système 'lxd'

```
etu@vm0:~$ sudo apt install openvswitch-switch snapd
etu@vm0:~$ sudo snap install lxd
etu@vm0:~$ sudo adduser etu lxd
Déconnexion / Reconnexion

etu@vm0:~$ id | grep -o lxd
lxd
etu@vm0:~$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games:/snap/bin
```

# Application → Cas jekyll

## ▪ Interconnexion réseau

1. Éditer le fichier

```
/etc/network/interfaces
```

2. Redémarrer la machine virtuelle

3. Vérifier l'état des interfaces

```
ip addr ls
```

```
etu@vm0:~$ cat /etc/network/interfaces
# This file describes the network
# interfaces available on your system
# and how to activate them.
# For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto C-3P0
iface C-3P0 inet dhcp
        ovs_type OVSBridge
        ovs_ports enp0s1

allow-C-3P0 enp0s1
iface enp0s1 inet manual
        ovs_bridge C-3P0
        ovs_type OVSPort
        up ip link set dev $IFACE up
        down ip link set dev $IFACE down
```

# Application → cas jekyll

- Gestion des conteneurs
  - Configuration initiale
  - Raccordement au commutateur C-3P0

```
etu@vm0:~$ lxd init
Would you like to use LXD clustering? (yes/no) [default=no]:
Do you want to configure a new storage pool? (yes/no) [default=yes]:
Name of the new storage pool [default=default]:
Name of the storage backend to use (btrfs, dir, lvm, ceph) [default=btrfs]:
Create a new BTRFS pool? (yes/no) [default=yes]:
Would you like to use an existing empty block device (e.g. a disk or partition)? (yes/no) [default=no]:
Size in GB of the new loop device (1GB minimum) [default=13GB]:
Would you like to connect to a MAAS server? (yes/no) [default=no]:
Would you like to create a new local network bridge? (yes/no) [default=yes]: no
Would you like to configure LXD to use an existing bridge or host interface? (yes/no) [default=no]: yes
Name of the existing bridge or host interface: C-3P0
Would you like LXD to be available over the network? (yes/no) [default=no]:
Would you like stale cached images to be updated automatically? (yes/no) [default=yes]:
Would you like a YAML "lxd init" preseed to be printed? (yes/no) [default=no]:

etu@vm0:~$ lxc profile device set default eth0 nictype bridged
```

# Application → Cas jekyll

- Créer le conteneur et installer les paquets

```
lxc launch images:debian/12 jekyll
lxc exec jekyll -- apt update
lxc exec jekyll -- apt -y install nginx htop ruby-full build-essential
```

- Configurer le *reverse proxy* nginx

```
config=$(cat << EOF
server {
    listen 80;
    listen [::]:80;
    location / {
        proxy_pass http://127.0.0.1:4000;
    }
}
EOF
)
lxc exec jekyll -- bash -c "echo \"${config}\" | tee /etc/nginx/sites-enabled/reverse.conf"
```

```
lxc exec jekyll -- rm /etc/nginx/sites-enabled/default
lxc exec jekyll -- systemctl restart nginx
lxc exec jekyll -- ss -tan
```

# Application → Cas jekyll

- Installer jekyll

```
lxc exec jekyll -- gem install jekyll bundler
```

- Créer un premier site web

```
lxc exec jekyll -- jekyll new my0wnB10g
```

- Lancer le service

```
lxc exec jekyll -- bash -c "cd my0wnB10g && bundle exec jekyll serve"
```

!! Consulter la page web avant d'interrompre le service !!

# Application → cas jekyll

- Accès au service Web via un tunnel SSH
- Relever les adresses du conteneur jekyll

```
etu@vm0:~$ lxc ls
+-----+-----+-----+-----+-----+-----+
| NAME | STATE | IPV4 | IPV6 | TYPE | SNAPSHOTS |
+-----+-----+-----+-----+-----+-----+
| jekyll | RUNNING | 198.18.28.212 (eth0) | 2001:678:3fc:1c:216:3eff:feaa:2b68 (eth0) | CONTAINER | 0 |
+-----+-----+-----+-----+-----+-----+
```

- Créer un tunnel via la connexion SSH à l'hyperviseur depuis son poste
  - Utiliser l'adresse IPv4 ou IPv6
  - Le nom « oscar » est défini dans la configuration du client SSH

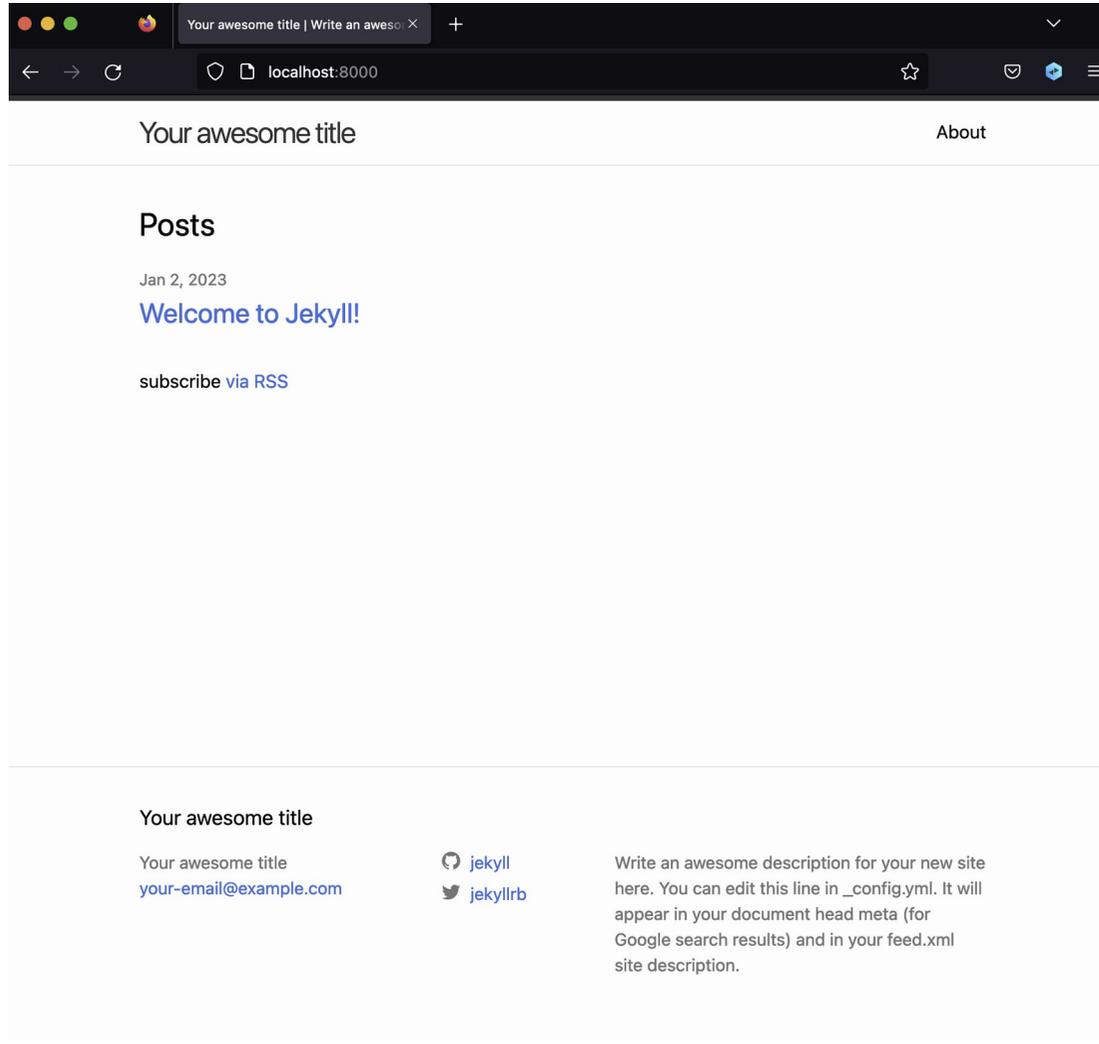
```
ssh -L 8000:198.18.28.212:80 oscar
```

```
ssh -L "8000:[2001:678:3fc:1c:216:3eff:feaa:2b68]:80" oscar
```

- Identifier les processus `nginx` et `jekyll` avec `htop` dans le conteneur et sur le système hôte

# Application → cas jekyll

- Accès au service Web via un tunnel SSH



- L'URL à consulter depuis le poste de travail est :

`http://localhost:8000`

# Bilan séance

- Shell Bash
  - Interpréteur de commandes aux fonctions étendues
  - Interface de base de l'administration système → contexte infrastructure
- Processus
  - Tout programme en cours d'exécution a l'«identité» de son propriétaire
- Droits sur le système de fichiers
  - Principes de gestion des droits Unix
  - Compromis efficacité/simplicité
- **Compétences essentielles en administration système**
  - Identifier les processus et leurs propriétaires
  - Différencier l'appartenance des processus entre système hôte et conteneur ou machine virtuelle

# Défi pour la prochaine séance !

- Quelle est la relation entre uid/gid et subuid/subgid ?
- Consulter la page à l'adresse ci-dessous
  - <https://ubuntu.com/blog/custom-user-mappings-in-lxd-containers>
  - Quel est le nom de l'utilisateur normal qui obtient la valeur subuid 100000 ?
  - À quel uid cette valeur correspond dans le conteneur ?

