LINUX

quelques commande utiles sous linux.

- alpine linux
- install docker ubuntu

alpine linux



> “
> Configuration sous **LXC**

Config timezone

```
setup-timezone -z Europe/Zurich && date
```

Ajout de dépot

```
vim /ect/apk/repositories
https://dl-cdn.alpinelinux.org/alpine/edge/main
https://dl-cdn.alpinelinux.org/alpine/edge/community
https://dl-cdn.alpinelinux.org/alpine/edge/testing
```

Mise à jours & installation des paquet par exemple docker.

```
apk update &&  apk upgrade
apk add curl wget vim htop openssh nmon neofetch docker
```

Démarrage auto du service

```
rc-update add docker default
/etc/init.d/docker start
```

Connection root ssh

```
rc-update add sshd default
vim /etc/ssh/sshd_config
service sshd start
reboot
```

Config docker

```
service docker status \
&& mkdir -p /usr/local/lib/docker/cli-plugins \
&& curl -SL "https://github.com/docker/compose/releases/download/v2.5.0/docker-compose-linux-x86_64" -o /usr/local/lib/docker/cli-plugins/docker-compose \
&& chmod +x /usr/local/lib/docker/cli-plugins/docker-compose \
&& rc-service docker restart \
&& rc-update add docker boot \
service docker start
```

install docker ubuntu



> “
> Installation de docker sous ubuntu 22 | **LXC**

Install des packages

```
sudo apt update \
&& sudo apt install -y fish-common vim wget nmon curl htop
```

Ajout utilisateur

```
adduser mon_user
vim /etc/passwd
vim /etc/sudoers
```

Install de fish

```
usermod -s /usr/bin/fish dezed
```

Correction de langue

```
sudo vim /etc/environment
sudo vim /etc/default/locale

LANG="fr_CH.utf8"
LC_ALL="fr_CH.utf8"
```

Mise à jour

```
sudo apt update && sudo apt upgrade -y && sudo apt dist-upgrade -y \
&& sudo apt autoremove -y && sudo apt purge && sudo apt autoclean \
&& reboot
```

Installation de docker

Ajout de package au fonctionnement de docker

```
sudo timedatectl set-timezone Europe/Zurich \
&& sudo apt-get update && sudo apt-get install \
    ca-certificates \
    curl \
    gnupg \
    lsb-release -Y
```

Ajout dépot docker

```
sudo mkdir -p /etc/apt/keyring` \
&& curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/etc/apt/keyrings/docker.gpg \
&& echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

installation du dépot docker

```
sudo apt-get update \
&& sudo apt-get install -y docker-ce docker-ce-cli containerd.io docker-compose-plugin \
&& sudo systemctl enable docker.service && sudo systemctl enable containerd.service
```

Utilisateur avec permission docker

```
sudo usermod -aG docker mon_user \
&& sudo usermod -aG docker $USER
```

Activation de l'accès distant docker

```
sudo mkdir -p /etc/systemd/system/docker.service.d/ \
&& sudo touch /etc/systemd/system/docker.service.d/options.conf
&& echo "[Service]" >> /etc/systemd/system/docker.service.d/options.conf \
&& echo "ExecStart=" >> /etc/systemd/system/docker.service.d/options.conf \
&& echo "ExecStart=/usr/bin/dockerd -H unix:// -H tcp://0.0.0.0:2375" >>
/etc/systemd/system/docker.service.d/options.conf
```

Fichier final: /etc/systemd/system/docker.service.d/options.conf

```
[Service]
ExecStart=
ExecStart=/usr/bin/dockerd -H unix:// -H tcp://0.0.0.0:2375
```

Docker resolution dns

```
sudo touch /etc/docker/daemon.json \
&& echo "{"dns": ["1.1.1.1","1.0.0.1","8.8.8.8","8.8.4.4"]}" >> /etc/docker/daemon.json
```

Si on utilise un conteneur dns

> **ajouter au fichier /etc/systemd/resolved.conf ces lignes:**
> DNS=1.1.1.1
> DNSStubListener=no
>
> ** ACTIVER IPV6 ** ==> lien

Solution complète pour Proxmox LXC

Étape 1 : Modifier la configuration LXC (sur l'hôte Proxmox)

Connectez-vous à votre serveur Proxmox et éditez le fichier de configuration du conteneur:

```
nano /etc/pve/lxc/VOTRE_CT_ID.conf
```

Ajoutez cette ligne:

```
lxc.apparmor.profile = unconfined
```

Sauvegardez et redémarrez le conteneur depuis Proxmox.

Étape 2 : Appliquer le hack AppArmor (dans le conteneur LXC)

Une fois le conteneur redémarré, connectez-vous à l'intérieur du conteneur et exécutez:

```
sudo mount --bind /dev/null /sys/module/apparmor/parameters/enabled
sudo systemctl restart docker
```

Étape 3 : Rendre le hack permanent (optionnel)

Pour que le hack survive aux redémarrages du conteneur, créez un service systemd:

```
sudo nano /etc/systemd/system/docker-apparmor-fix.service
```

Contenu du fichier :

```
[Unit]
Description=Fix AppArmor for Docker in LXC
```

```
Before=docker.service
DefaultDependencies=no

[Service]
Type=oneshot
ExecStart=/bin/mount --bind /dev/null /sys/module/apparmor/parameters/enabled
RemainAfterExit=yes

[Install]
WantedBy=multi-user.target
```

Activez le service:

```
sudo systemctl enable docker-apparmor-fix.service
sudo systemctl start docker-apparmor-fix.service
```