

# Référence rapide Alpine Linux

Gestion des paquets, services, réseau, image de base Docker

## Gestion des paquets

### Bases d'apk

```
apk update # rafraîchir l'index des paquets
apk upgrade # mettre à jour tous les paquets
apk add curl git vim # installer des paquets
apk del curl # supprimer un paquet
apk search nginx # rechercher des paquets
```

### Infos sur les paquets

```
apk info # lister les paquets installés
apk info -a nginx # infos détaillées sur un paquet
apk info -L nginx # lister les fichiers du paquet
apk policy nginx # afficher les versions disponibles
```

### Paquets virtuels

```
# Installer les dépendances de build en groupe, à supprimer ensuite
apk add --virtual .build-deps gcc musl-dev
make && make install
apk del .build-deps
```

### Dépôts

```
# /etc/apk/repositories
https://dl-cdn.alpinelinux.org/alpine/v3.20/main
https://dl-cdn.alpinelinux.org/alpine/v3.20/community
@edge https://dl-cdn.alpinelinux.org/alpine/edge/testing
```

## Services

### Gestion des services OpenRC

```
rc-service nginx start # démarrer le service
rc-service nginx stop # arrêter le service
rc-service nginx restart # redémarrer le service
rc-service nginx status # vérifier le statut
```

### Gestion des niveaux d'exécution

```
rc-update add nginx default # activer au démarrage
rc-update del nginx default # désactiver au démarrage
rc-update show # lister tous les services
rc-status # afficher les services actifs
```

### Niveaux d'exécution

```
sysinit Initialisation du système (systèmes de fichiers, horloge)
boot Services système de base (réseau, syslog)
default Services normaux (serveurs web, démons)
shutdown Tâches d'arrêt
```

## Configuration

### Fichiers de config importants

```
/etc/apk/repositories URLs des dépôts de paquets
/etc/hostname Nom d'hôte du système
/etc/network/interfaces Config des interfaces réseau
/etc/conf.d/ Configuration spécifique aux services
/etc/motd Message du jour
```

### Configuration du système

```
setup-alpine # configuration interactive complète
setup-timezone # définir le fuseau horaire
setup-keymap # configurer la disposition clavier
setup-hostname myhost # définir le nom d'hôte
```

## Fuseau horaire

```
apk add tzdata
cp /usr/share/zoneinfo/US/Eastern /etc/localtime
echo "US/Eastern" > /etc/timezone
apk del tzdata # optionnel : supprimer pour économiser de l'espace
```

## Réseau

### Config des interfaces

```
# /etc/network/interfaces
auto eth0
iface eth0 inet dhcp
# --- statique ---
iface eth0 inet static
address 192.168.1.10/24
gateway 192.168.1.1
```

### Commandes réseau

```
ip addr show # afficher les adresses IP
ip route show # afficher la table de routage
ip link set eth0 up # activer l'interface
setup-interfaces # config réseau interactive
```

### DNS et pare-feu

```
# DNS: /etc/resolv.conf
nameserver 1.1.1.1
nameserver 8.8.8.8
# Pare-feu
apk add iptables
iptables -L -n # lister les règles
```

## Utilisateurs

### Gestion des utilisateurs

```
adduser alice # créer un utilisateur (interactif)
adduser -D -s /bin/sh bob # non-interactif, définir le shell
deluser alice # supprimer un utilisateur
passwd alice # définir/changer le mot de passe
```

### Groupes et sudo

```
addgroup devs # créer un groupe
addgroup alice devs # ajouter un utilisateur au groupe
apk add doas # alternative légère à sudo
# /etc/doas.conf
permit persist alice as root
```

### Utilisateurs système

```
adduser -S -D -H -s /sbin/nologin myapp
# -S utilisateur système -D sans mot de passe
# -H sans répertoire home -s sans shell
```

## Disques et stockage

### Commandes système de fichiers

```
df -h # résumé de l'utilisation disque
du -sh /var/log # taille d'un répertoire
lsblk # lister les périphériques blocs
mount /dev/sda1 /mnt # monter un périphérique
umount /mnt # démonter
```

### LBU (Sauvegarde locale Alpine)

```
# Pour modes diskless/data – persister les changements après redémarrage
lbu status # afficher les changements non sauvegardés
lbu commit # sauvegarder sur le support de démarrage
lbu list # lister les fichiers sauvegardés
lbu include /etc/myconf # ajouter un chemin à la sauvegarde
```

## Configuration disque

```
setup-disk # installation disque interactive
setup-disk /dev/sda # installer sur un disque spécifique
# Modes : sys (traditionnel), data, diskless
```

## Image de base Docker

### Pourquoi Alpine pour Docker

```
~5 Mo d'image de base vs ~80 Mo pour Debian slim
musl libc Plus léger que glibc (quelques problèmes de compatibilité)
gestionnaire apk Rapide, sans cache par défaut
Surface d'attaque minimale Moins de paquets = moins de CVE
```

### Dockerfile minimal

```
FROM alpine:3.20
RUN apk add --no-cache python3 py3-pip
COPY app.py /app/
CMD ["python3", "/app/app.py"]
```

### Build multi-étapes

```
FROM golang:1.22-alpine AS builder
WORKDIR /src
COPY . .
RUN go build -o /app
FROM alpine:3.20
COPY --from=builder /app /app
CMD ["/app"]
```

### Pièges courants

```
--no-cache Toujours utiliser pour garder l'image légère
musl vs glibc Certains binaires nécessitent le paquet gcompat
Pas de bash par défaut Utiliser /bin/sh ou apk add bash
Fuseau horaire manquant Installer tzdata si nécessaire
```

## Patterns courants

### Installer les outils de build

```
apk add --no-cache build-base # gcc, make, etc.
apk add --no-cache python3-dev # en-têtes Python
apk add --no-cache linux-headers # en-têtes noyau
```

### Tâches cron

```
# Ajouter une tâche cron
echo "*/5 * * * * /usr/local/bin/task.sh" \
| crontab -
rc-service cron start
rc-update add cron default
```

### Activer SSH

```
apk add openssh
rc-service sshd start
rc-update add sshd default
# Config : /etc/ssh/sshd_config
```

### Mettre à jour la version Alpine

```
# Éditer /etc/apk/repositories : changer v3.19 → v3.20
apk update
apk upgrade --available
sync && reboot
```