

REFERENCIA RÁPIDA DE ALPINE LINUX

Gestión de paquetes, servicios, red, imagen base Docker

Gestión de paquetes

Básico de apk

```
apk update           # actualizar índice de paquetes
apk upgrade          # actualizar todos los paquetes
apk add curl git vim # instalar paquetes
apk del curl         # eliminar un paquete
apk search nginx     # buscar paquetes
```

Info de paquete

```
apk info            # listar paquetes instalados
apk info -a nginx  # información detallada del paquete
apk info -L nginx  # listar archivos del paquete
apk policy nginx    # mostrar versiones disponibles
```

Paquetes virtuales

```
# Instalar deps de compilación como grupo, eliminar después
apk add -virtual build-deps gcc musl-dev
make && make install
apk del .build-deps
```

Repositorios

```
# /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
```

Servicios

Gestión de servicios con OpenRC

```
rc-service nginx start # iniciar servicio
rc-service nginx stop  # detener servicio
rc-service nginx restart # reiniciar servicio
rc-service nginx status # verificar estado
```

Gestión de niveles de ejecución

```
rc-update add nginx default # habilitar al arrancar
rc-update del nginx default # deshabilitar al arrancar
rc-update show              # listar todos los servicios
rc-status                   # mostrar servicios activos
```

Niveles de ejecución

sysinit Inicialización del sistema (sistemas de archivos, reloj)
boot Servicios básicos del sistema (red, syslog)
default Servicios normales (servidores web, daemons)
shutdown Tareas de apagado

Configuración

Archivos de configuración principales

/etc/apk/repositories URLs de repositorios de paquetes
/etc/hostname Nombre de host del sistema
/etc/network/interfaces Configuración de interfaces de red
/etc/conf.d/ Configuración específica de servicios
/etc/motd Mensaje del día

Configuración del sistema

```
setup-alpine        # configuración completa interactiva
setup-timezone      # establecer zona horaria
setup-keymap        # configurar distribución del teclado
setup-hostname myhost # establecer nombre de host
```

Zona horaria

```
apk add tzdata
cp /usr/share/zoneinfo/US/Eastern /etc/localtime
echo "US/Eastern" > /etc/timezone
apk del tzdata # opcional: eliminar para ahorrar espacio
```

Red

Configuración de interfaz

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

Comandos de red

```
ip addr show # mostrar direcciones IP
ip route show # mostrar tabla de rutas
ip link set eth0 up # activar interfaz
setup-interfaces # configuración de red interactiva
```

DNS y Firewall

```
# DNS: /etc/resolv.conf
nameserver 1.1.1.1
nameserver 8.8.8.8
# Firewall
apk add iptables
iptables -L -n # listar reglas
```

Usuarios

Gestión de usuarios

```
adduser alice # crear usuario (interactivo)
adduser -D -s /bin/sh bob # no interactivo, establecer shell
deluser alice # eliminar usuario
passwd alice # establecer/cambiar contraseña
```

Grupos y Sudo

```
addgroup devs # crear grupo
addgroup alice devs # agregar usuario al grupo
apk add doas # alternativa ligera a sudo
# /etc/doas.conf
permit persist alice as root
```

Usuarios de sistema

```
adduser -S -D -H -s /sbin/nologin myapp
# -S usuario sistema -D sin contraseña
# -H sin directorio home -s sin shell
```

Disco y almacenamiento

Comandos de sistema de archivos

```
df -h # resumen de uso de disco
du -sh /var/log # tamaño de directorio
lsblk # listar dispositivos de bloque
mount /dev/sdal /mnt # montar dispositivo
umount /mnt # desmontar
```

LBU (Respaldo local de Alpine)

```
# Para modos diskless/data - persistir cambios entre reinicios
lbu status # mostrar cambios sin confirmar
lbu commit # guardar cambios en medio de arranque
lbu list # listar archivos respaldados
lbu include /etc/myconf # agregar ruta al respaldo
```

Configuración de disco

```
setup-disk # instalación interactiva de disco
setup-disk /dev/sda # instalar en disco específico
# Modos: sys (tradicional), data, diskless
```

Imagen base Docker

Por qué Alpine para Docker

~5 MB imagen base vs ~80 MB para Debian slim
musl libc Más pequeño que glibc (algunos problemas de compatibilidad)
gestor de paquetes apk Rápido, sin caché por defecto
Superficie de ataque mínima Menos paquetes = menos CVEs

Dockerfile mínimo

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

Build multi-etapa

```
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"]
```

Problemas comunes

--no-cache Usar siempre para mantener imagen pequeña
musl vs glibc Algunos binarios necesitan el paquete gcompat

Sin bash por defecto Usar /bin/sh o apk add bash
Zona horaria ausente Instalar tzdata si se necesita

Patrones comunes

Instalar herramientas de compilación

```
apk add --no-cache build-base # gcc, make, etc.
apk add --no-cache python3-dev # cabeceras Python
apk add --no-cache linux-headers # cabeceras del kernel
```

Tareas cron

```
# Agregar una tarea cron
echo "* * * * * /usr/local/bin/task.sh" \
| crontab -
rc-service crond start
rc-update add crond default
```

Habilitar SSH

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

Actualizar versión de Alpine

```
# Editar /etc/apk/repositories: cambiar v3.19 -> v3.20
apk update
apk upgrade --available
sync && reboot
```