

Referência Rápida Alpine Linux

Gerenciamento de pacotes, serviços, redes, imagem base Docker

Gerenciamento de Pacotes

Básico do apk

```
apk update # atualizar índice de pacotes
apk upgrade # atualizar todos os pacotes
apk add curl git vim # instalar pacotes
apk del curl # remover um pacote
apk search nginx # buscar pacotes
```

Informações de Pacotes

```
apk info # listar pacotes instalados
apk info -a nginx # informações detalhadas do pacote
apk info -L nginx # listar arquivos do pacote
apk policy nginx # exibir versões disponíveis
```

Pacotes Virtuais

```
# Install build deps as a group, remove later
apk add --virtual .build-deps gcc musl-dev
make && make install
apk del .build-deps
```

Repositórios

```
# /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
```

Serviços

Gerenciamento de Serviços OpenRC

```
rc-service nginx start # iniciar serviço
rc-service nginx stop # parar serviço
rc-service nginx restart # reiniciar serviço
rc-service nginx status # verificar status
```

Gerenciamento de Runlevels

```
rc-update add nginx default # habilitar na inicialização
rc-update del nginx default # desabilitar na inicialização
rc-update show # listar todos os serviços
rc-status # exibir serviços em execução
```

Runlevels

sysinit	Inicialização do sistema (sistemas de arquivos, relógio)
boot	Serviços básicos do sistema (rede, syslog)
default	Serviços normais (servidores web, daemons)
shutdown	Tarefas de desligamento

Configuração

Arquivos de Configuração Principais

/etc/apk/repositories	URLs dos repositórios de pacotes
/etc/hostname	Nome do host do sistema
/etc/network/interfaces	Configuração de interfaces de rede
/etc/conf.d/	Configuração específica de serviços
/etc/motd	Mensagem do dia

Configuração do Sistema

```
setup-alpine # configuração completa interativa
setup-timezone # definir fuso horário
setup-keymap # configurar layout do teclado
setup-hostname myhost # definir nome do host
```

Fuso Horário

```
apk add tzdata
cp /usr/share/zoneinfo/US/Eastern /etc/localtime
echo "US/Eastern" > /etc/timezone
apk del tzdata # optional: remove to save space
```

Redes

Configuração de Interface

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

Comandos de Rede

```
ip addr show # exibir endereços IP
ip route show # exibir tabela de rotas
ip link set eth0 up # ativar interface
setup-interfaces # configuração interativa de rede
```

DNS e Firewall

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

Usuários

Gerenciamento de Usuários

```
adduser alice # criar usuário (interativo)
adduser -D -s /bin/sh bob # não-interativo, definir shell
deluser alice # excluir usuário
passwd alice # definir/alterar senha
```

Grupos e Sudo

```
addgroup devs # criar grupo
addgroup alice devs # adicionar usuário ao grupo
apk add doas # lightweight sudo alternative
# /etc/doas.conf
permit persist alice as root
```

Usuários do Sistema

```
adduser -S -D -H -s /sbin/nologin myapp
# -S system user -D no password
# -H no home dir -s no shell
```

Disco e Armazenamento

Comandos de Sistema de Arquivos

```
df -h # resumo de uso do disco
du -sh /var/log # tamanho do diretório
lsblk # listar dispositivos de bloco
mount /dev/sda1 /mnt # montar dispositivo
umount /mnt # desmontar
```

LBU (Backup Local Alpine)

```
# For diskless/data modes - persist changes across reboots
lbu status # exibir alterações não confirmadas
lbu commit # salvar alterações na mídia de boot
lbu list # listar arquivos no backup
lbu include /etc/myconf # adicionar caminho ao backup
```

Configuração de Disco

```
setup-disk # instalação interativa em disco
setup-disk /dev/sda # instalar em disco específico
# Modes: sys (traditional), data, diskless
```

Imagem Base Docker

Por que Alpine para Docker

~5 MB base image	vs ~80 MB para Debian slim
musl libc	Menor que glibc (alguns problemas de compatibilidade)
apk package manager	Rápido, sem cache por padrão
Minimal attack surface	Menos pacotes = 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-Estágio

```
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"]
```

Armadilhas Comuns

--no-cache	Sempre use para manter a imagem pequena
musl vs glibc	Alguns binários precisam do pacote gcompat
No bash by default	Use /bin/sh ou apk add bash
Timezone missing	Instale tzdata se necessário

Padrões Comuns

Instalar Ferramentas de Build

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

Tarefas Cron

```
# Add a cron job
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
```

Atualizar Versão do Alpine

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