

ALPINE LINUX QUICK REFERENCE

Package management, services, networking, Docker base image

Package Management

apk Basics

```
apk update           # refresh package index
apk upgrade          # upgrade all packages
apk add curl git vim # install packages
apk del curl         # remove a package
apk search nginx     # search for packages
```

Package Info

```
apk info            # list installed packages
apk info -a nginx  # detailed package info
apk info -L nginx  # list files in package
apk policy nginx    # show available versions
```

Virtual Packages

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

Repositories

```
# /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

OpenRC Service Management

```
rc-service nginx start # start service
rc-service nginx stop  # stop service
rc-service nginx restart # restart service
rc-service nginx status # check status
```

Runlevel Management

```
rc-update add nginx default # enable at boot
rc-update del nginx default # disable at boot
rc-update show               # list all services
rc-status                    # show running services
```

Runlevels

```
sysinit System initialization (filesystems, clock)
boot     Basic system services (networking, syslog)
default Normal services (web servers, daemons)
shutdown Shutdown tasks
```

Configuration

Key Config Files

```
/etc/apk/repositories Package repository URLs
/etc/hostname         System hostname
/etc/network/interfaces Network interface config
/etc/conf.d/          Service-specific configuration
/etc/motd             Message of the day
```

System Setup

```
setup-alpine # interactive full setup
setup-timezone # set timezone
setup-keymap # configure keyboard layout
setup-hostname myhost # set hostname
```

Timezone

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

Networking

Interface Config

```
# /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
```

Network Commands

```
ip addr show # show IP addresses
ip route show # show routing table
ip link set eth0 up # bring interface up
setup-interfaces # interactive net config
```

DNS & Firewall

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

Users

User Management

```
adduser alice # create user (interactive)
adduser -D -s /bin/sh bob # non-interactive, set shell
deluser alice # delete user
passwd alice # set/change password
```

Groups & Sudo

```
addgroup devs # create group
addgroup alice devs # add user to group
apk add doas # lightweight sudo alternative
# /etc/doas.conf
permit persist alice as root
```

System Users

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

Disk & Storage

Filesystem Commands

```
df -h # disk usage summary
du -sh /var/log # directory size
lsblk # list block devices
mount /dev/sdal /mnt # mount device
umount /mnt # unmount
```

LBU (Alpine Local Backup)

```
# For diskless/data modes - persist changes across reboots
lbu status # show uncommitted changes
lbu commit # save changes to boot media
lbu list # list backed-up files
lbu include /etc/myconf # add path to backup
```

Disk Setup

```
setup-disk # interactive disk install
setup-disk /dev/sda # install to specific disk
# Modes: sys (traditional), data, diskless
```

Docker Base Image

Why Alpine for Docker

```
~5 MB base image vs ~80 MB for Debian slim
musl libc Smaller than glibc (some compat issues)
```

```
apk package manager Fast, no cache by default
Minimal attack surface Fewer packages = fewer CVEs
```

Minimal Dockerfile

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

Multi-Stage Build

```
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"]
```

Common Gotchas

```
--no-cache Always use to keep image small
musl vs glibc Some binaries need gcompat package
No bash by default Use /bin/sh or apk add bash
Timezone missing Install tzdata if needed
```

Common Patterns

Install Build Tools

```
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
```

Cron Jobs

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

Enable SSH

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

Upgrade Alpine Version

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