

Alpine Linux 快速参考

包管理、服务、网络、Docker 基础镜像

包管理

apk 基础

```
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
```

包信息

```
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
```

虚拟包

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

软件源

```
# /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
```

服务管理

OpenRC 服务管理

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

运行级别管理

```
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
```

运行级别

```
sysinit 系统初始化 (文件系统、时钟)
boot    基础系统服务 (网络、syslog)
default 普通服务 (Web 服务器、守护进程)
shutdown 关机任务
```

配置

关键配置文件

```
/etc/apk/repositories 软件源 URL 列表
/etc/hostname         系统主机名
/etc/network/interfaces 网络接口配置
/etc/conf.d/          各服务专属配置
/etc/motd            登录提示信息
```

系统初始化

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

时区设置

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

网络

接口配置

```
# /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
```

网络命令

```
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 与 防火墙

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

用户管理

用户操作

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

用户组与 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
```

系统用户

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

磁盘与存储

文件系统命令

```
df -h # disk usage summary
du -sh /var/log # directory size
lsblk # list block devices
mount /dev/sda1 /mnt # mount device
umount /mnt # unmount
```

LBU (Alpine 本地备份)

```
# 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
```

磁盘安装

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

Docker 基础镜像

为何选 Alpine 做 Docker 基础镜像

```
~5 MB base image 对比 Debian slim 约 80 MB
musl libc 比 glibc 更小 (部分兼容性问题)
apk package manager 快速, 默认不留缓存
Minimal attack surface 包越少, CVE 越少
```

最简 Dockerfile

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

多阶段构建

```
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"]
```

常见坑

```
--no-cache 务必加, 保持镜像小
musl vs glibc 部分二进制需要 gcompat 包
No bash by default 用 /bin/sh 或 apk add bash
Timezone missing 需要时安装 tzdata
```

常用模式

安装构建工具

```
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
```

定时任务

```
# Add a cron job
echo "*/5 * * * * /usr/local/bin/task.sh" \
| crontab -
rc-service crond start
rc-update add crond default
```

启用 SSH

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

升级 Alpine 版本

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