

Alpine Linux 빠른 참조

패키지 관리, 서비스, 네트워킹, Docker 기반 이미지

패키지 관리

apk 기본

```
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
```

패키지 정보

```
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
```

가상 패키지

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

저장소

```
# /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
```

서비스

OpenRC 서비스 관리

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

런레벨 관리

```
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
```

런레벨

```
sysinit 시스템 초기화 (파일시스템, 시계)
boot 기본 시스템 서비스 (네트워킹, syslog)
default 일반 서비스 (웹 서버, 데몬)
shutdown 종료 작업
```

설정

주요 설정 파일

```
/etc/apk/repositories 패키지 저장소 URL
/etc/hostname 시스템 호스트명
/etc/network/interfaces 네트워크 인터페이스 설정
/etc/conf.d/ 서비스별 설정
/etc/motd 오늘의 메시지
```

시스템 설정

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

시간대

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

네트워킹

인터페이스 설정

```
# /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
```

네트워크 명령

```
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 & 방화벽

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

사용자

사용자 관리

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

그룹 & 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
```

시스템 사용자

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

디스크 & 스토리지

파일시스템 명령

```
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 로컬 백업)

```
# 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
```

디스크 설정

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

Docker 기반 이미지

Docker에서 Alpine을 쓰는 이유

```
~5 MB base image Debian slim 약 80 MB 대비
musl libc glibc보다 작음 (일부 호환성 문제 있음)
apk package manager 빠르고 기본적으로 캐시 없음
Minimal attack surface 패키지가 적을수록 CVE도 적음
```

최소 Dockerfile

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

멀티 스테이지 빌드

```
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"]
```

자주 하는 실수

```
--no-cache 이미지를 작게 유지하려면 항상 사용
musl vs glibc 일부 바이너리는 gcompat 패키지 필요
No bash by default /bin/sh 사용 또는 apk add bash
Timezone missing 필요한 경우 tzdata 설치
```

일반 패턴

빌드 도구 설치

```
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
```

크론 작업

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

SSH 활성화

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

Alpine 버전 업그레이드

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