apk Alpine package manager

START HERE

`apk update`
Refresh the package index

`apk upgrade`
Upgrade installed packages

`apk add <pkg>`
Install a package

`apk search <name>`
Find a package

`apk add --no-cache <pkg>`
Install without keeping the index

`apk info <pkg>`
Show package details

`apk del <pkg>`
Remove a package

`apk info -L <pkg>`
List a package's files

Repositories /etc/apk/repositories → `apk update` → **Local index** package list → `apk add` → **World file** wanted packages → `resolve` → **Installed** world + deps

CONCEPTS

> apk keeps a world file (/etc/apk/world) of what you asked for; everything else is a dependency and goes away with it.

> Alpine uses musl and busybox, so some glibc binaries do not run; check for a -dev or -libs sibling package.

INSTALL AND REMOVE

`apk add <pkg>`
Install a package

`apk add <pkg1> <pkg2>`
Install several packages

`apk add --no-cache <pkg>`
Install, skip the local index cache
The standard form in Dockerfiles: no apk update step and no cache left in the image.

`apk del <pkg>`
Remove a package and unused deps

`apk del -r <pkg>`
Also remove packages depending on it

`apk add -t <name> <pkg>...`
Group packages as a virtual package

`apk del <name>`
Remove a virtual package and its deps

`apk add <pkg>=<version>`
Install a specific version

`apk add <pkg>~=<version>`
Install a fuzzy version match

`apk add <pkg>@<tag>`
Install from a tagged repository

`apk add -u <pkg>`
Upgrade a package if installed

`apk add cmd:<command>`
Install whichever package provides a command

`apk add so:<library>`
Install by shared library name

`apk add pc:<name>`
Install by pkg-config module

`apk add --allow-untrusted <file>.apk`
Install a local, unsigned package

INSTALL AND REMOVE (cont.)

`apk add -s <pkg>`
Simulate an install

`apk add`
Reconcile installed packages with the world file

UPDATE AND UPGRADE

`apk update`
Refresh the package index

`apk upgrade`
Upgrade all installed packages

`apk update && apk upgrade`
Refresh, then upgrade

`apk upgrade -s`
Preview an upgrade

`apk upgrade --available`
Sync to repos after a release switch
Use after editing the release in /etc/apk/repositories; also allows replacing packages with differing versions.

`apk upgrade <pkg>`
Upgrade only this package

`apk list -u`
List upgradable packages

`apk version -l '<'`
List packages with a newer version

`apk fix`
Repair or reinstall broken packages

> apk refreshes its index automatically when the cache is older than a few hours, but only with a cache enabled.

SEARCH AND INSPECT

`apk search <name>`
Search package names

`apk search -e <name>`
Search for an exact name

`apk search -d <text>`
Search descriptions

`apk search -v <name>`
Show versions and descriptions

`apk search cmd:<command>`
Find the package providing a command

SEARCH AND INSPECT (cont.)

`apk search so:<library>`
Find the package providing a library

`apk info <pkg>`
Show package details

`apk info`
List installed packages

`apk list -I`
List installed packages with versions

`apk info -L <pkg>`
List files in a package

`apk info -W <path>`
Find the package owning a file

`apk info -e <pkg>`
Check whether a package is installed

`apk info -R <pkg>`
List a package's dependencies

`apk info -r <pkg>`
List packages depending on it

`apk policy <pkg>`
Show available versions and repos

`apk list -O`
List orphaned packages

REPOSITORIES

`cat /etc/apk/repositories`
Show configured repositories

`apk add -X <url> <pkg>`
Add a repository for one command

`apk add --repository <url> <pkg>`
Same, long form

`apk update -v`
Show repository index status

> Repositories are main, community and testing; testing is edge-only. Use a @tag prefix to opt in per package.

repositories line

Part	Example
`Plain`	https://dl-cdn.alpinelinux.org/alpine/v3.20/main
`Community`	https://dl-cdn.alpinelinux.org/alpine/v3.20/community
`Edge testing`	@testing https://dl-cdn.alpinelinux.org/alpine/edge/testing
`Local path`	/var/apk/my-packages

CACHE AND WORLD

`apk cache clean`
Delete outdated cached packages

`apk cache -v sync`
Sync cache with the world file

`apk cache download`
Download packages into the cache

`apk fetch <pkg>`
Download a package without installing

`apk audit`
List changed files under /etc

`cat /etc/apk/world`
Show explicitly installed packages

`ls /etc/apk/cache`
Inspect the package cache

`update-conf`
Merge .apk-new config files
Package upgrades leave a .apk-new file when you modified the config; update-conf walks you through them.

`apk stats`
Show package database statistics

RECIPES

Dockerfile install

```
FROM alpine:3.20
RUN apk add --no-cache curl ca-certificates
```

Temporary build dependencies

```
apk add --no-cache -t .build-deps build-base
# compile something here
apk del .build-deps
```

Pin a testing package

```
echo '@testing https://dl-cdn.alpinelinux.org/alpine/edge/testing' \
  >> /etc/apk/repositories
apk add <pkg>@testing
```

Who provides this file?

```
apk info -W /usr/bin/<cmd>
apk search cmd:<cmd>
```