Getting apt-get on an alpine container

Asked 6 years, 3 months ago Modified 6 years, 3 months ago Viewed 223k times

64

I have to install a few dependencies on my docker container, I want to use python:3.6-alpine version to have it as light as possible, but apk package manager which comes with alpine is giving me trouble so I would like to get the apt-get package manager. I tried:

```
apk add apt-get
```

and it didnt work.

how can I get it on the container?

docker apt-get apt alpine-linux

Share Improve this question Follow

asked Nov 20, 2018 at 9:19
NotSoShabby
3,728 ● 9 ● 37 ● 65

7 `apk` is the default package manager on alpine. You don't need `apt` – lependu Nov 20, 2018 at 9:22

1 What do you want to install that is giving you trouble with apk? – Julio Daniel Reyes Nov 20, 2018 at 17:41

@JulioDanielReyes apk doesn't have all packages like `php-gearman` – SP Singh Jan 6, 2020 at 13:57

1 It has the php7-gearman package. You just need to add the additional repositories specified here unix.stackexchange.com/a/491257. Then a `apk update && apk add php7-gearman` should work... – Julio Daniel Reyes Jan 6, 2020 at 14:08

>"What do you want to install that is giving you trouble with apk?" - It stopped installing python. – Ark-kun Dec 16, 2020 at 1:49

1 Answer

Sorted by: Highest score (default)

55

Using multiple package systems is usually a very bad idea, for many reasons. Packages are likely to collide and break and you'll end up with much greater mess than you've started with. See this excellent answer for more detail: Is there a pitfall of using multiple package managers?

A more feasible approach would be troubleshooting and resolving the issues you are having with `apk`. `apk` is designed for simplicity and speed, and should take very little getting used to. It is really an excellent package manager, IMO.

For a good tutorial, I warmly recommend the `apk` introduction page at the Alpine Wiki site: https://wiki.alpinelinux.org/wiki/Alpine_Linux_package_management

If you're determined not to use `apk`, and for the sake of experiment want try bringing up `apt` instead, as a first step, you'll have first to build `apt` from source: https://github.com/Debian/apt. Then, if it is produces a functional build (not likely since it's probably not compatible with musl libc), you'll have to wire it to some repositories, but Alpine repositories are only fit for `apk`, not `apt`. As you can see, this is not really feasible, and not the route you want to go to.

Share Improve this answer Follow

answered Nov 20, 2018 at 19:29
valiano
18.7k ● 7 ● 71 ● 83

2 That's fair, but that got me thinking - why is that any different than using pip as well as other package manage (pip is the python package manager) which everybody do? – NotSoShabby Nov 21, 2018 at 8:10

4 Python has its own package system, which is managed by `pip`, as opposed to `apk` and `apt` which manage the Linux system binaries and programs. `pip` effect is limited to scope of Python programs. Think of it this way - if things break in `pip`, Python may break, but if things go wrong in `apk` or `apt`, your entire Linux system may break. – valiano Nov 21, 2018 at 8:36

4 @NotSoShabby or put in other words, `pip` is orthogonal to `apk` / `apt` - they are independent of each other. – valiano Nov 21, 2018 at 10:28

Start asking to get answers

Find the answer to your question by asking.

Ask question

Explore related questions

docker apt-get apt alpine-linux

See similar questions with these tags.