#019 Docker entrypoint

Introduction

this is part 19 from the journey it's a long journey(360 day) so go please check previous parts , and if you need to walk in the journey with me please make sure to follow because I may post more than once in 1 Day but surely I will post daily at least one 😍.

And I will cover lot of tools as we move on.

---

entrypoint vs cmd

I found this article on dev.to explain it very well so go and read it and when you fully understand it cameback here.

---

entrypoint

what I need to focus on right now is entrypoint , why I can use it and how?

basically entrypoint is an shell script that run first when container run , so it's not stick to the image so he does not make a new layer.

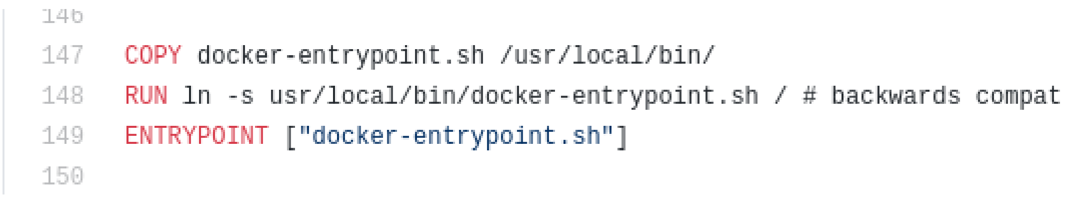
entrypoint can be a script in linux is sh (don't worry there is an entire section for linux bash :D in our Journey!)

I am going to get help from the postgres image , they have an big entry point.

go to this link , and you will see a big script called docker-entrypoint.sh

I will focus on some things that can be combined with docker.

first to be able to run this script we can see that the postgres alpine template they provide contain this.



```
COPY docker-entrypoint.sh /usr/local/bin/
RUN ln -s usr/local/bin/docker-entrypoint.sh / # backwards compat
ENTRYPOINT ["docker-entrypoint.sh"]
```

```
COPY docker-entrypoint.sh /usr/local/bin/
```

this path is special path used in linux to put your own bin files or scripts bin is simply an excitable in linux like exe or bat in windows.

```
RUN ln -s usr/local/bin/docker-entrypoint.sh /
```

ln -s is mean symbolic link.

> *Symbolic links offer a convenient way to organize and share files. They provide quick access to long and confusing directory paths. They are heavily used in linking libraries in Linux.*

```
ENTRYPOINT ["docker-entrypoint.sh"]
```

this say to Dockerfile here is the script I need to run.

Explain docker-entrypoint.sh

```
#!/usr/bin/env bash
```

in this line in the head of the script is meaning that this script is will work on bash (since we have alpine we use sh insted)

```
set -Eeo pipefail
```

this is for safety if things go wrong on executing the script immediately exit. pipefail take in count also if let's say cmd1|cmd2|cmd3 if one of them fail also immediately exit. Don't continue to others.

```
exec "$@"
```

this is where the magic it's simply say take all what you pass as argument from terminal and execute it as 1 line , so when it's done running all the script will start to run what we pass as arguments.

End

I know this is hard topic maybe because it's contain lot of shell things , but when we reach the bash section there lot to cover and you will be able to back here and easily understand what is going on.