

exec form vs shell form

Command forms

`RUN`, `ENTRYPOINT`, and `CMD` are all directives to run a command in your Dockerfile. All three takes two forms of command, shell form and exec form.

1. Shell form

The commands are written without the `[]` brackets and are run by the container's shell as a child process, the normal way of running things in Linux and by default the shell is `/bin/sh -c`

```
FROM alpine:latest

# /bin/sh -c 'echo $HOME'
RUN echo $HOME

# /bin/sh -c 'echo $PATH'
CMD echo $PATH
```

2. Exec form

The commands are written in `[]` brackets and are run directly, not through a shell, mean which means is the main process with PID 1. You will not get shell features like variable substitution from environment variable. In addition, you will not be able to pipe output, chain commands, IO redirections. Those features are only possible with shell form.

```
FROM alpine:latest

RUN ["pwd"]

CMD ["sleep", "1s"]
```

Differences between shell and exec form

In shell form, commands will inherit the environment variables from the shell

```
FROM alpine:latest

# This will echo out /root because it is run in the shell
```

```
RUN echo $HOME
```

```
# This will just echo "$HOME" because it isn't using variable substitution, a shell feature.
```

```
RUN ["echo", "$HOME"]
```

Signals

In addition, because the shell form runs the command via shell, i.e. it will spawn child process to run the commands if you are going to stop the container it will require 2 signals to stop it. In exec form because your process is the main process i.e. with PID of 1, when you stop the container docker only needs to send one signal to stop the main process.

The container stops when the main process PID of 1 stops.

In essence, shell form since the executable is not the main process, it will not receive signals.

In exec form since the executable is the main process (PID 1) it will receive signals.

Why does it take some time for container to stop

Docker when you stop a container will sent the main process (PID 1) a SIGTERM to ask it nicely to shutdown. If the main process doesn't handle SIGTERM like a shell script without `trap` then after a grace period (10 seconds), SIGKILL is sent to forcefully stop shutdown the process. Which is why if you run custom script that runs forever without handling SIGTERM it will take 10 seconds for it to be forcefully shutdown.

Recommended forms

- RUN: Use shell form
- ENTRYPOINT: Use exec form
- CMD: Use exec form

Revision #7

Created 2023-02-20 15:53:42 UTC by Tamarine

Updated 2023-07-09 03:21:16 UTC by Tamarine