

4.7Докер часть 2 Практика докер 2

Что нужно сделать:

1. Придумать и описать сценарий использования кэша для сборки, который даст неожиданный результат (для наивного пользователя, который о кэше не задумывается и просто жмет `docker build`)
2. Придумать и описать сценарий появления мусора в истории слоев образа, который будет незаметен при запуске контейнера из финального образа. Желательно, чтоб этот мусор неадекватно увеличивал размер образа. (без использования `docker commit`, с ним все очевидно)

1.

Можно привести пример, когда в `Dockerfile` есть родительский образ или же мультистейдж, когда пользователь поменял на удаленном репозитории родительский образ, но не спуллил его себе. Таким образом, сборка пройдет быстро, ведь если использовать `tag latest`, а в локальной репе такой образ не менялся, то по идее в сборку пойдет родительский образ без изменений.

2.

Допустим что родительский образ весил 500 MB. Девопс взял себе этот образ для создания образа для своего приложения, какое нибудь `python` приложение, что пишет логи себе под ноги в контейнер, без `logrotate` и т.д. Этот контейнер крутился на виртуальной машине, пользователи по многу раз в день заходили на его сайт, информация по запросам логировалась, допустим, что с избытком.

В какой то момент такой же сайт понадобился другому девопсу, но репозиторий с исходным кодом и Докерфайлом был удален. Тогда хитрый Девопс решил скоммитить новый образ, запустить его в репозиторий и стянуть себе на свою VM. Именно так он и сделал. После этого он запустил контейнер с энтрипойнтом баша, чтобы поменять код пришлось докачать множество тулзов,удалил старые логи, немного поменял код, закинул некоторые статические файлы и снова коммитнул для создания нового образа. Таким образом, его конечный образ стал весить гораздо больше и будет увеличиваться, поскольку образ хранит предыдущие логи, тулзы в слоях и т.д.