



二畳半堂

八 月

14

火曜日

2018

## コンテナからコンテナを操作する

**Docker**コンテナからコンテナを操作するのは特殊なユースケースで縁遠い存在と思いきや、アプリケーションをコンテナ化しそれをコンテナベースのCIで扱おうとすると割と直ぐに直面することになったのでメモ。

コンテナ内でコンテナを操作する主立ったやり方には

- DinD (Docker in Docker)
- DooD (Docker outside of Docker)

の二つがある。

またコンテナのビルドに限定すればDockerに依存せず独立してコンテナイメージをビルドするDaemon-less Image Builderを使う方法もある。

\* \* \*

**DinD**

---

## 概要

Docker in Docker。

Dockerインストール済みのコンテナを使用しコンテナ内でホストとは別にDockerデーモンを動かす方法。

Dockerの `privileged` フラグのコントリビュータでありDinDの走りとなったの `docker:dnd` の作者でもある `jpetazzo` 氏のブログ

『Using Docker-in-Docker for your CI or testing environment? Think twice』

によると本来DinDはDocker本体の開発を効率化するために編み出された方法でありそれ以外の用途としてはいくつか難点があるとのこと。

タイトルにもある通りCI界限でDinDの乱用を危惧しており代替手段としてDooDについて言及されている。

DinDを行うためのイメージとしてalpineベースのDocker公式イメージ(dindタグ)や、ubuntuベースのteracy/ubuntuなどがある。

## 試してみる

Docker公式のDinD対応イメージを動かしてみる。

```
$ docker run --privileged --name dind -d docker:stable-dind
```

デフォルトではDockerコンテナはデバイスファイルにアクセスできないためコンテナ内でDockerデーモンを起動することはできない。

DinDなコンテナを起動する際は `privileged` フラグを付与し特権モードで動かす必要がある。

```
$ docker ps
CONTAINER ID        IMAGE               COMMAND              CREATED             STATUS
91ee3e6eeca8        docker:stable-dind "dockerd-entrypoint..." 11 seconds ago      Up 10 seconds
```

コンテナが起動したことを確認し中に入ってみる。  
コンテナ内でdockerコマンドが利用できることを確認する。

```
$ docker exec -it dind /bin/ash

/ # docker -v
Docker version 18.06.0-ce, build 0ffa825
```

プロセスも確認。  
dockerd と containerd が上がっていることを確認。

```
/ # ps
PID USER TIME COMMAND
1 root 0:13 dockerd --host=unix:///var/run/docker.sock --host=tcp://0.0.0.0:2375
18 root 0:28 docker-containerd --config /var/run/docker/containerd/containerd.toml
139 root 0:00 /bin/ash
195 root 0:00 ps
```

イメージとコンテナを確認。  
当然起動直後なので何もない。

```
/ # docker ps -a
CONTAINER ID        IMAGE               COMMAND              CREATED             STATUS             P
```

/ # docker images				
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
<div></div>				

このようにDinDではホストのDockerとは完全に別物のDocker環境が利用可能になる。

ちなみにDocker公式イメージの場合、コンテナ内の `/var/lib/docker` はData Volumeまたはホスト側の任意のデータディレクトリをマウントして使用する。

デフォルトではData Volumeが使用される。コンテナ作成時に対となるData Volumeも自動的に作成されるためコンテナの作成を繰り返していくと不要になったData Volumeが溜まっていくため注意が必要。

## DooD

---

### 概要

Docker outside of Docker。

Dockerインストール済みのコンテナを使用するのはDinDと同じだがdockerd、containerdは使用しない。

代わりコンテナ側からホストのdocker.sock ( `/var/run/docker.sock` ) をマウントすることでコンテナ上のDockerコマンドはホスト側のDocker環境で実行される。

## 試してみる

Docker公式コンテナを起動しシェルを実行する。

```
$ docker run -ti --rm -v /var/run/docker.sock:/var/run/docker.sock docker /bin/ash

/ #
```

`docker image` でイメージ一覧を確認するとホスト側のイメージが表示されている。

```
/ # docker image
REPOSITORY TAG          IMAGE ID      CREATED      SIZE
docker      dind             1e09f740386c 3 weeks ago  164MB
docker      stable-dind     1e09f740386c 3 weeks ago  164MB
docker      latest          b22ef854da32 3 weeks ago  153MB
alpine      3.5             6c6084ed97e5 7 months ago 3.99MB
```

当然 `docker ps` でもホスト上のコンテナ一覧が表示される。

```
/ # docker ps
CONTAINER ID IMAGE          COMMAND                  CREATED      STATUS      PORTS      NAMES
1efd57fdd79c docker        "docker-entrypoint.s..." 4 hours ago  Up 4 hours  laughing_engelbart
046efc45afa5 docker:stable-dind "dockerd-entrypoint...." 24 hours ago Up 24 hours  2375/tcp    di
```

このようにDooDではホスト側のDocker環境を共用することができる。

要はDockerのクライアントサーバシステムのクライアント部分だけをコンテナ側にもってきた形態。コンテナ側はホスト側のDockerバージョンとの整合を気をつける必要がある。

# Daemon-less Image Builder

---

Dockerデーモンに依存せずコンテナイメージをビルドするためのツールには、

- ポスト docker build 『[moby/buildkit](#)』
- MS [Jess Frazelle](#)氏発の『[genuinetools/img](#)』
- RedHat社発 『[projectatomic/buildah](#)』
- Google製 『[GoogleContainerTools/kaniko](#)』

等がある。

詳細は『[2018年のDocker・Moby](#)』 P27 3章 「Docker buildの後継・代替」 参照。

#docker

2021 2020 2019 2018 2017 2016  
apple applematch aws blender chrome circleci clojure clojurescript cloudfront codebuild core.async css cui dakoku design  
direnv docker drone ecr emacs fish fzf gadget ghq git gitlab golang google gpg gradle hub hugo intellij ios iphone  
java javascript jenkins lambda leiningen mac naruko network nodenv postgres raspberry-pi s3 ses slack sns sonarqube  
spring-boot sqs suica terraform theme tmux twitter vim webstrom