

dockerignoreの話

Container Build Meetup #2 #container_build
@orisano

docker buildの話をしてします

Docker CLI

dockerd

\$ docker build .

Docker CLI

dockerd

\$ docker build [build context]

Docker CLI

dockerd

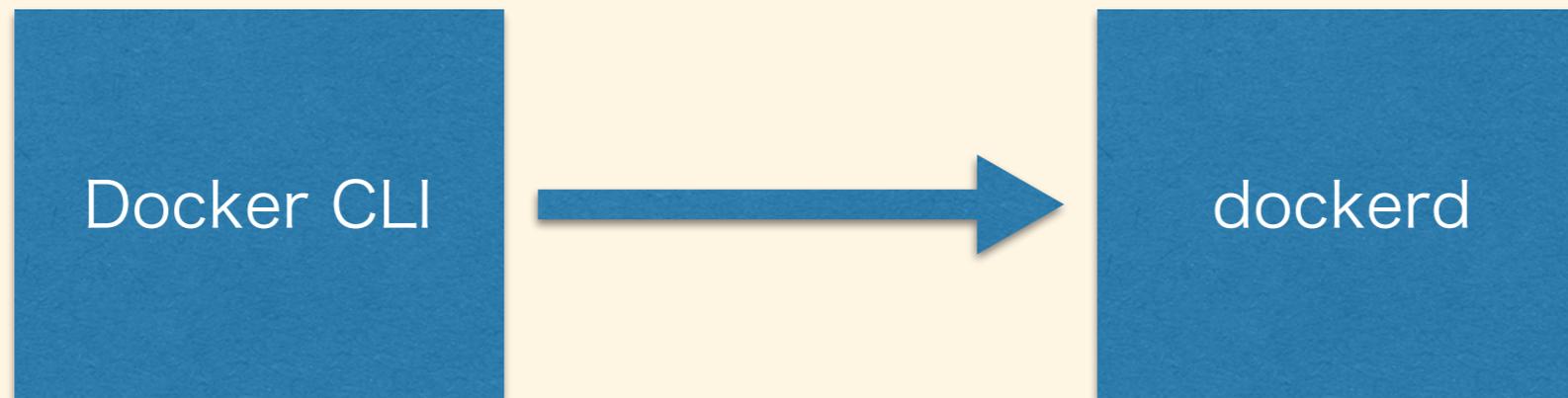
\$ docker build .

Docker CLI

dockerd

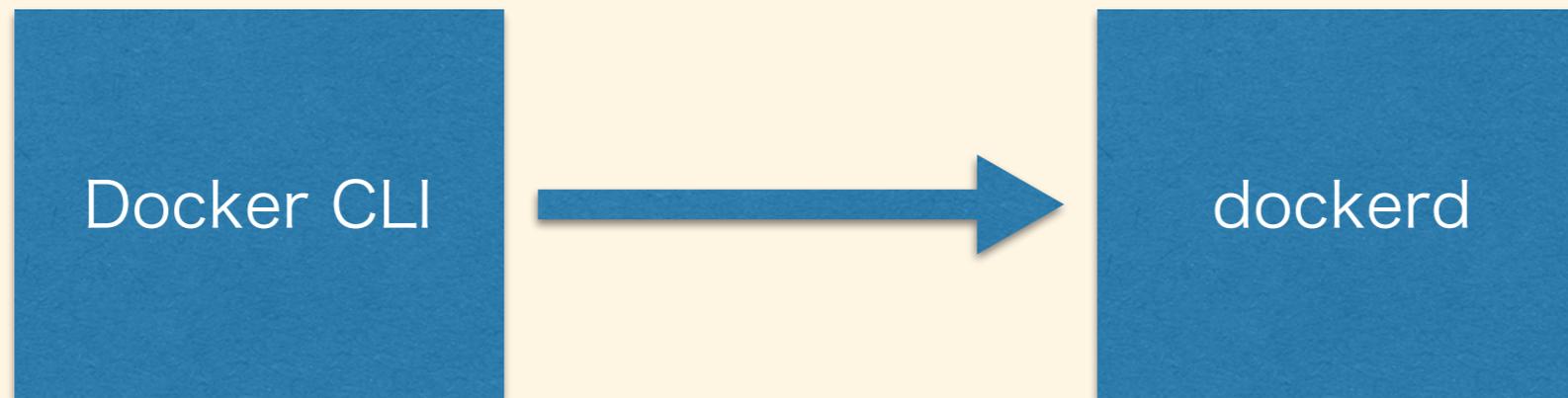
1. build contextをtarでarchiveする

\$ docker build .



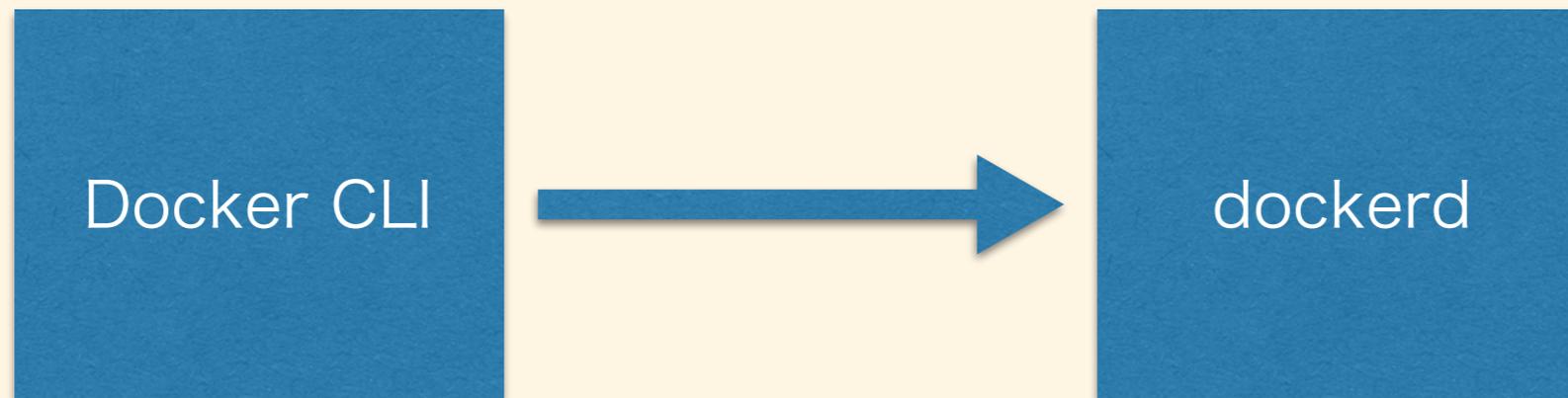
2. dockerdに対してPOSTする

\$ docker build .



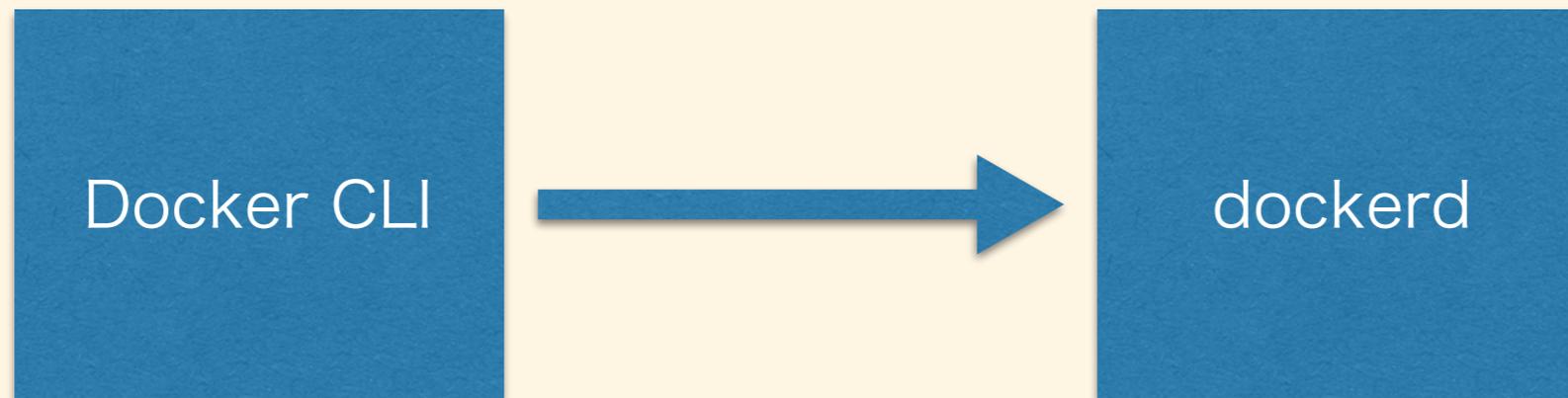
3. 受け取ったtarを展開

\$ docker build .



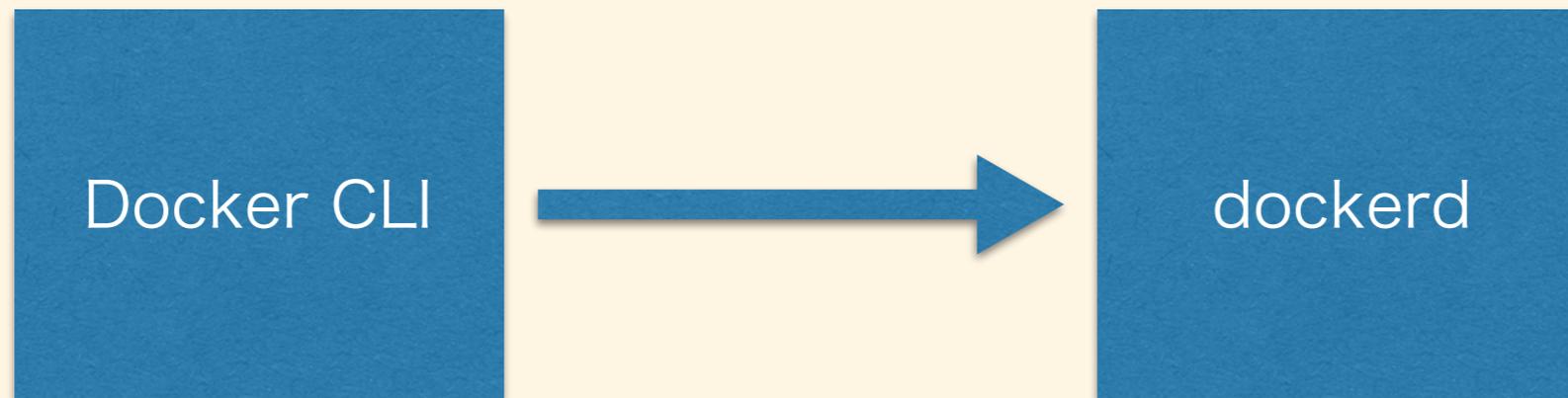
4. Dockerfileの解釈をする

\$ docker build .



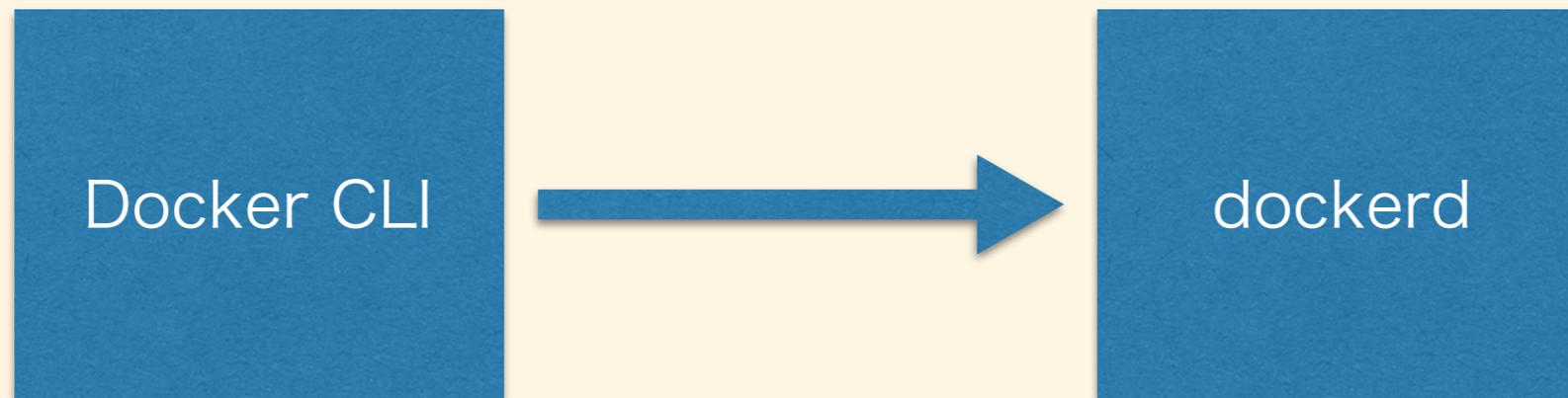
5. ステージごとに実行

\$ docker build .



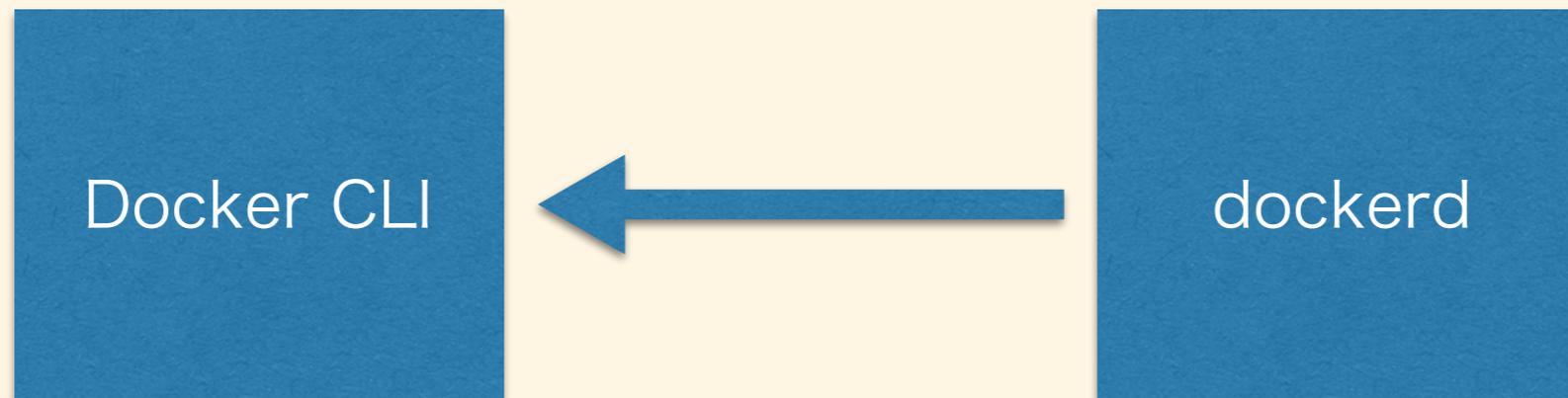
6. Instructionを実行

\$ docker build .



7. Instruction毎にimageができる

\$ docker build .



8.終わったら通知

不要なファイルは
build contextに含めたくない

なぜ？

- 1. Archiveのコスト**
- 2. 転送のコスト**
- 3. cacheのため**

1. Archiveのコスト

2. 転送のコスト

3. cacheのため

大量のファイルを
アーカイブするのは重い

(`node_modules`など)

クライアント側で行うが
現状2つの実装がある

docker/cli (Go)
docker/docker-py (Python)

docker/cli (Go)
dockerコマンド

docker/docker-py (Python)

**Docker Engine APIを
Pythonから使うためのもの**

**docker/compose
が使っている**

なんか遅い

AntiVirusなどが
I/Oをhookしている場合
I/Oコストがやばい

1. Archiveのコスト

2. 転送のコスト

3. cacheのため

大きなファイルを送るのは
大変

1. Archiveのコスト

2. 転送のコスト

3. cacheのため

COPYで

**意図してないファイルが
対象になっていると
cacheが壊れて最悪**

**build contextに
含めてはいけないファイル
がある**

1. 認証情報

2. アーキテクチャ依存

1. 認証情報

2. アーキテクチャ依存

言わずもがな

.envなど

.gitが危険な場合もある

**personal tokenでの
git cloneをしているなど**

1. 認証情報

2. アーキテクチャ依存

package managerによるが
その環境向けの**build**済み
バイナリが置いてあることが
(**npm**)

以上の問題を解決するために
dockerignore
を使う

dockerignore

後ろに書いてある
patternが優先される

先頭に!をつけると否定できる

使える表記

foo/*

foo/**

foo/?

内部で
正規表現に変換している

正規表現ならわかる
という人向けに

github.com/orisano/dire

対応する正規表現を出力

どうやって実現しているか

\$ docker build .

Docker CLI

dockerd

1. build contextをtarでarchiveする

\$ docker build .

Docker CLI

dockerd

**0. build contextの以下のファイルがread可能か確認
(ここもdockerignoreを考慮される)**

\$ docker build .

Docker CLI

dockerd

**0.5. directoryをtraverseしながらdockerignoreに
マッチしないファイルをtarに追加していく**

\$ docker build .

Docker CLI

dockerd

1. build contextをtarでarchiveする

余談

\$ docker build .

Docker CLI

dockerd

**0. build contextの以下のファイルがread可能か確認
(ここもdockerignoreを考慮される)**

ファイル数分dockerignoreが
コンパイルされるように
なっていた😱

docker/cli/pull/1577
で修正され

Docker 19.03
から入ります

ちょっと特殊な例

**monorepoで
subsystem間の依存がある
場合**

大**本**を**build context**に
しないといけなくなる

build contextがでかくなる

dockerignoreで解決!
することはできない

**dockerignoreはbuild毎に
ファイルを指定できない**

**subsystem毎に
dockerignoreを作り
buildのたびに
dockerignoreを自動生成する
アプローチが有効**

github.com/orisano/dignore

よくある話

gitignore と dockerignore
が
ほぼ一緒になる

そういう場合には
dockerignore
いらなくなる方法がある

git archiveって
コマンド使ったことありますか

repositoryをtarにできる
コマンド

tarにすればbuildできるので

git archive HEAD | docker build -