

誤解しない！ Dockerの肝と価値 そして使い方 ——開発者編——

モウフカブール
大澤文孝



インフラ担当者向けと 開発者向け

インフラ担当者に向けた全体のお話は、OSC京都でお話しました
<https://event.ospn.jp/osc2021-online-kyoto/>

今回は開発者向け！

インフラ向けはOSC京都のアーカイブで！



Who Am I?

大澤文孝

テクニカルライター／システムエンジニア／プログラマ／インフラエンジニア。
著書はもうすぐ100冊。100冊記念パーティーやりたい(けど反対されている)。

主な著書

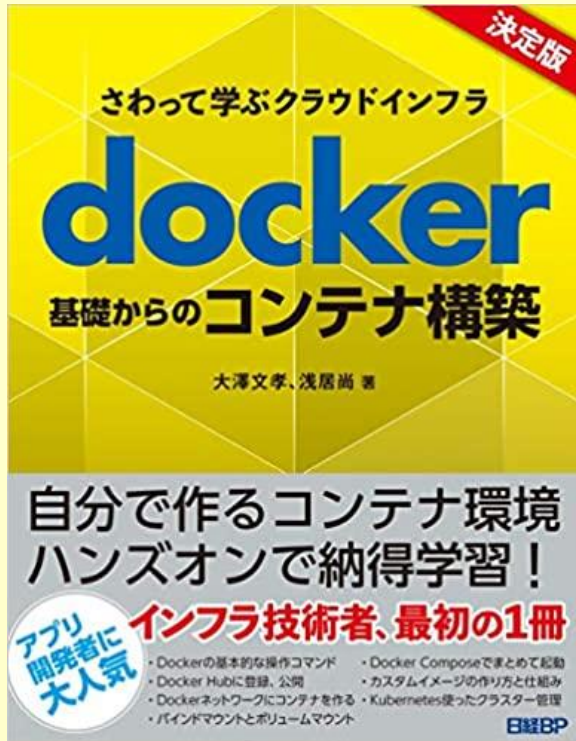


もうすぐ100冊
パーティー
やりたい！





今日のお話はDocker



日経BP社より好評発売中

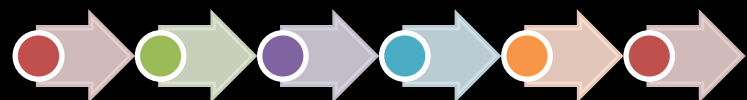
※ときどきAmazon Unlimitedになったり、セールしたりします。

AWSで
Dockerを学ぶ
Kubernetes
も解説！





アジェンダ



Presented by
Mofukabur.いんく



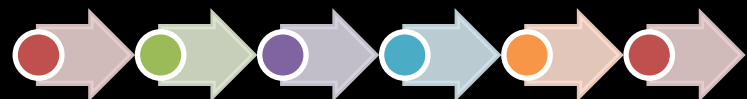


- Dockerとは**隔離したLinux環境**を動かす技術である
仮想マシンと何が違うの？
WindowsやmacOSで使うDocker(Docker Desktop)
- Dockerの基本的なこと
コンテナとイメージ
コンテナの実行
- Apacheを使ってみよう
起動、ネットワークにつなげる、中身をいじる、ホストのディスクを見せる
- 開発現場で使うDocker
どんな場面で使うか、開発者は何を知らなければならないか、チームリーダーがすべきこと





Dockerとは隔離である





• Dockerとは

(主に)Linuxにおいて、プログラムを隔離した場所で動かす「コンテナ」という仕組みを実現する実装のひとつ

【開発者目線】

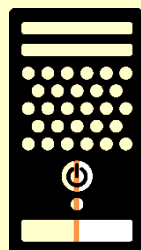
→DockerはWindowsやmacOSで動かせるよ(Docker Desktop)

→WindowsやmacOSでLinuxのソフトが隔離した状態で動くよ





・ 隔離するとは？



192.168.0.1



/bin
/sbin
/home
/usr
/var
...

メインの環境
(ホスト)

172.17.0.2



/bin
/sbin
/home
/usr
/var
...

コンテナ1

172.17.0.3



/bin
/sbin
/home
/usr
/var
...

コンテナ2

ポイント①
独立したネットワーク

ポイント②
独立したストレージ

※IPアドレスはすべて仮





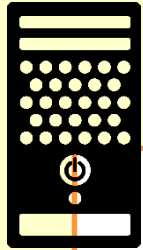
- 隔離すると何が嬉しいの？

それぞれ別々にソフトのインストールやアップデートができる





・ 隔離しない(Docker利用しない)場合



192.168.0.1



/bin
/sbin
/home
/usr
/var

app1

app2

...

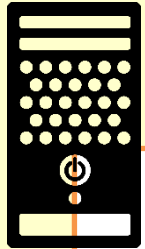
アプリ1

アプリ2





・ 隔離しない(Docker利用しない)場合



Linuxがインストールされた
PC(またはサーバ)

192.168.0.1



/bin
/sbin
/home
/usr
/var

app1

app2

...

アプリ1は、
PHP8にアップ
デートしたい！

アプリ1

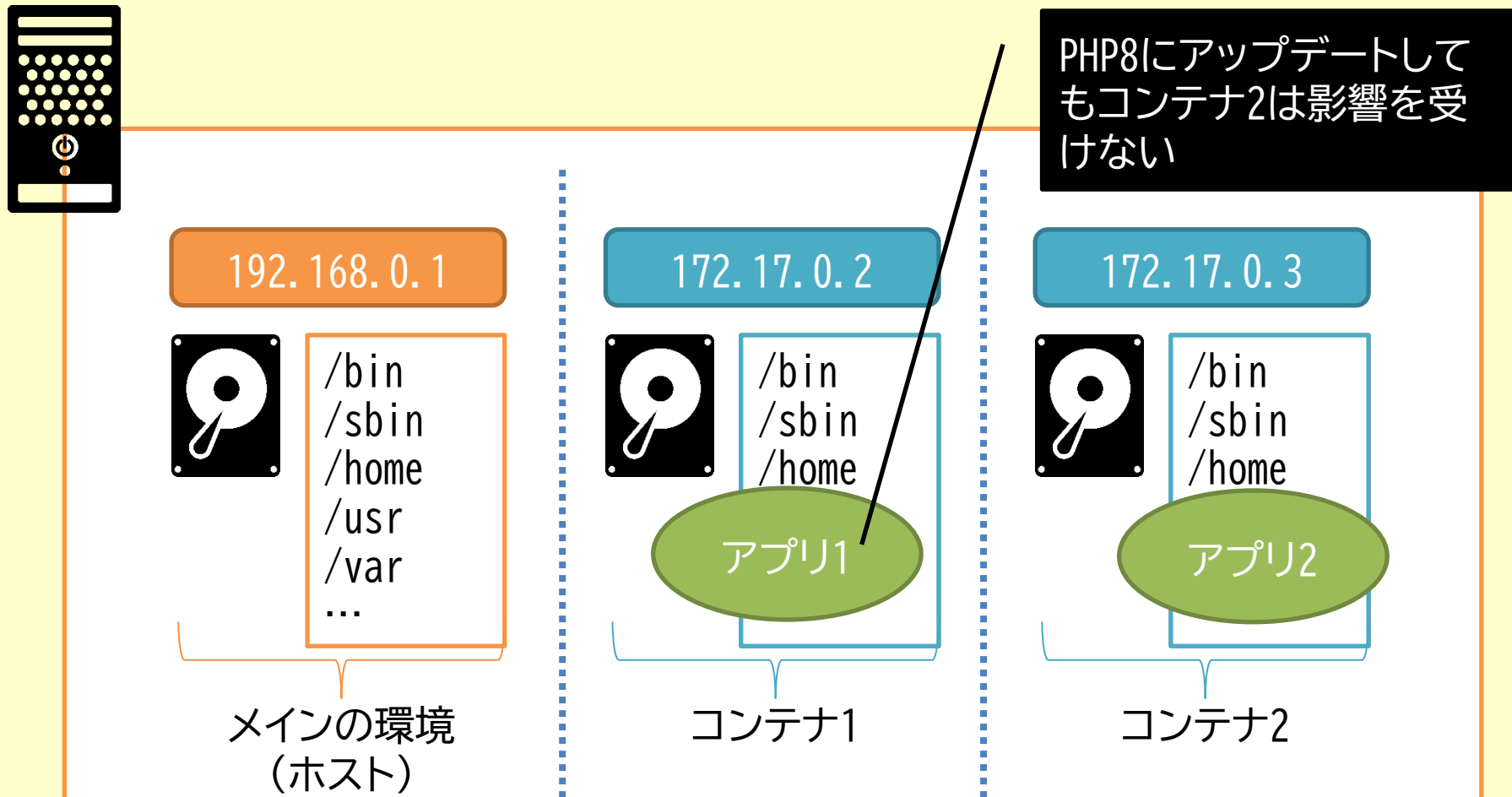
アプリ2

? アプリ2が動か
なくなるかも



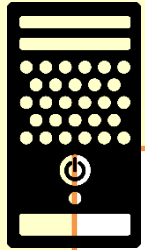


・コンテナなら同居ができる





・実験が容易



192.168.0.1



/bin
/sbin
/home
/usr
/var
...

メインの環境
(ホスト)

172.17.0.2



/bin
/sbin
/home
/usr
/var
...

コンテナ1

インストール、アンインストール、設定ファイルの書き換え、さまざまな実験をしても、メインの環境に影響がない！

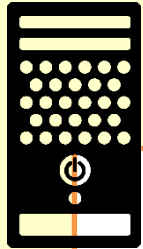
〔開発者目線〕
CI/CDのテストの
サーバーに便利

ごちゃごちゃしてきたら、
コンテナを作り直して、
まっさらな状態から始め
直すのも簡単





・ いろんな環境の同居ができる



192.168.0.1



/bin
/sbin
/home
/usr
/var
...

メインの環境
(ホスト)

172.17.0.2



PHP7環境

172.17.0.4



Ruby環境

172.17.0.3



PHP8環境

172.17.0.5



MySQL環境

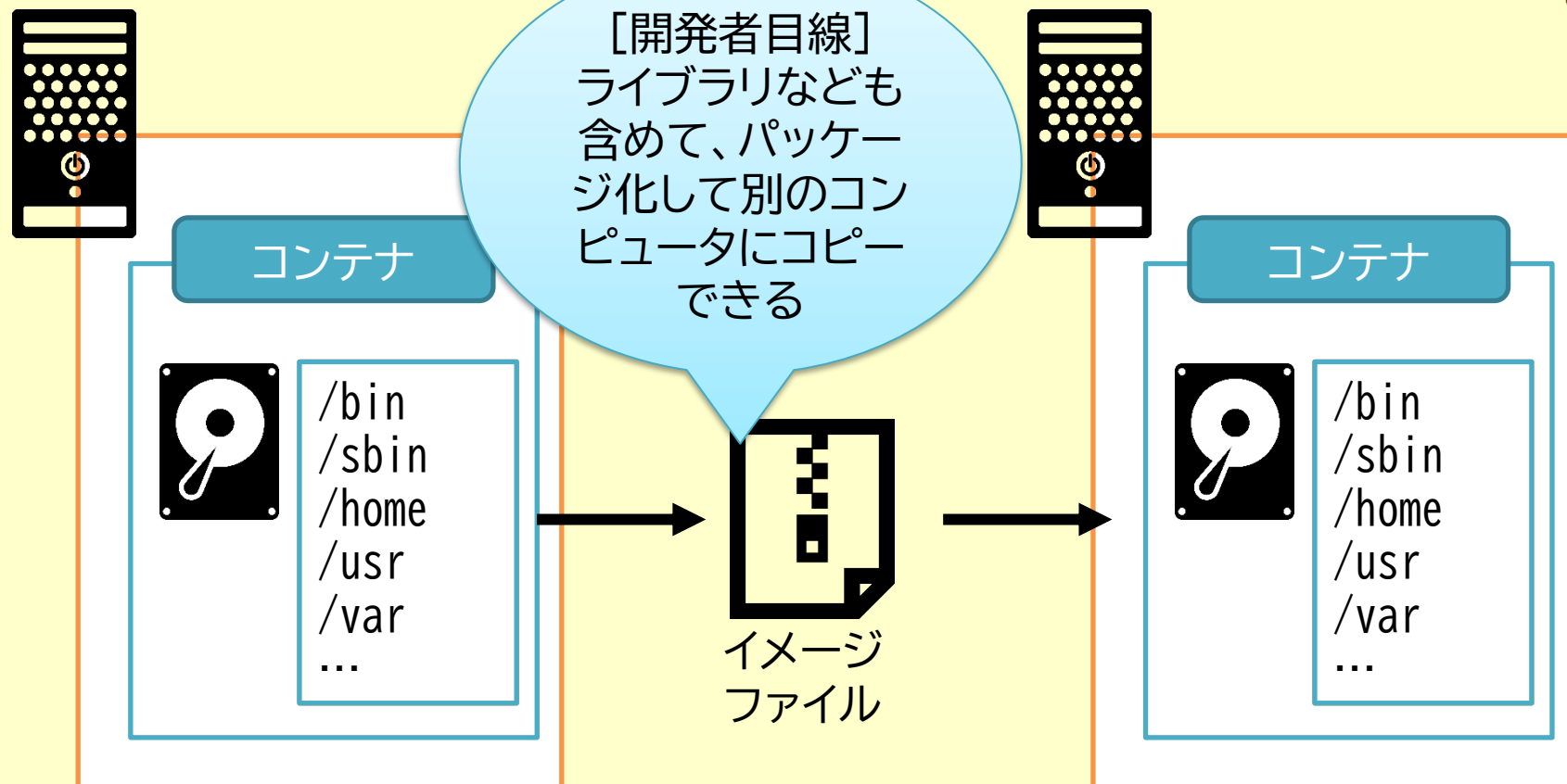
「開発者目線」
バージョン違いの
同居OK。検証が
はかどる





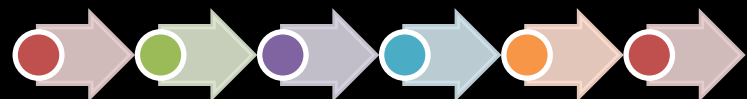
・移動・コピーが容易

納品物のアー
カイブ、差分の
確認から解放
される！





仮想マシンと何が違うの？

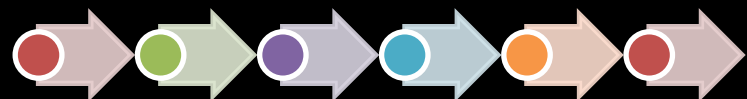
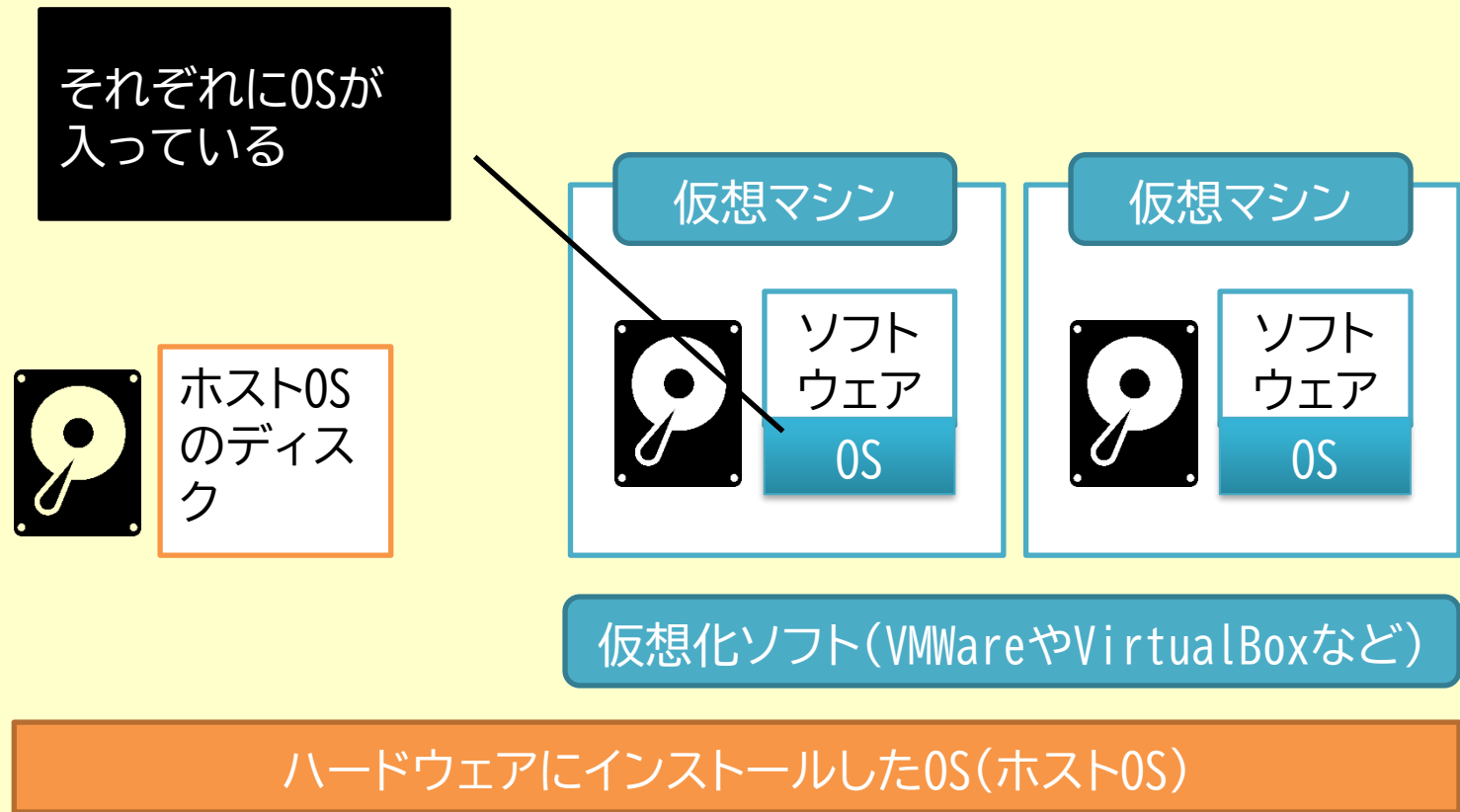


Presented by
Mofukabur.いんく



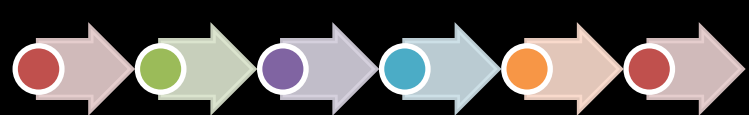
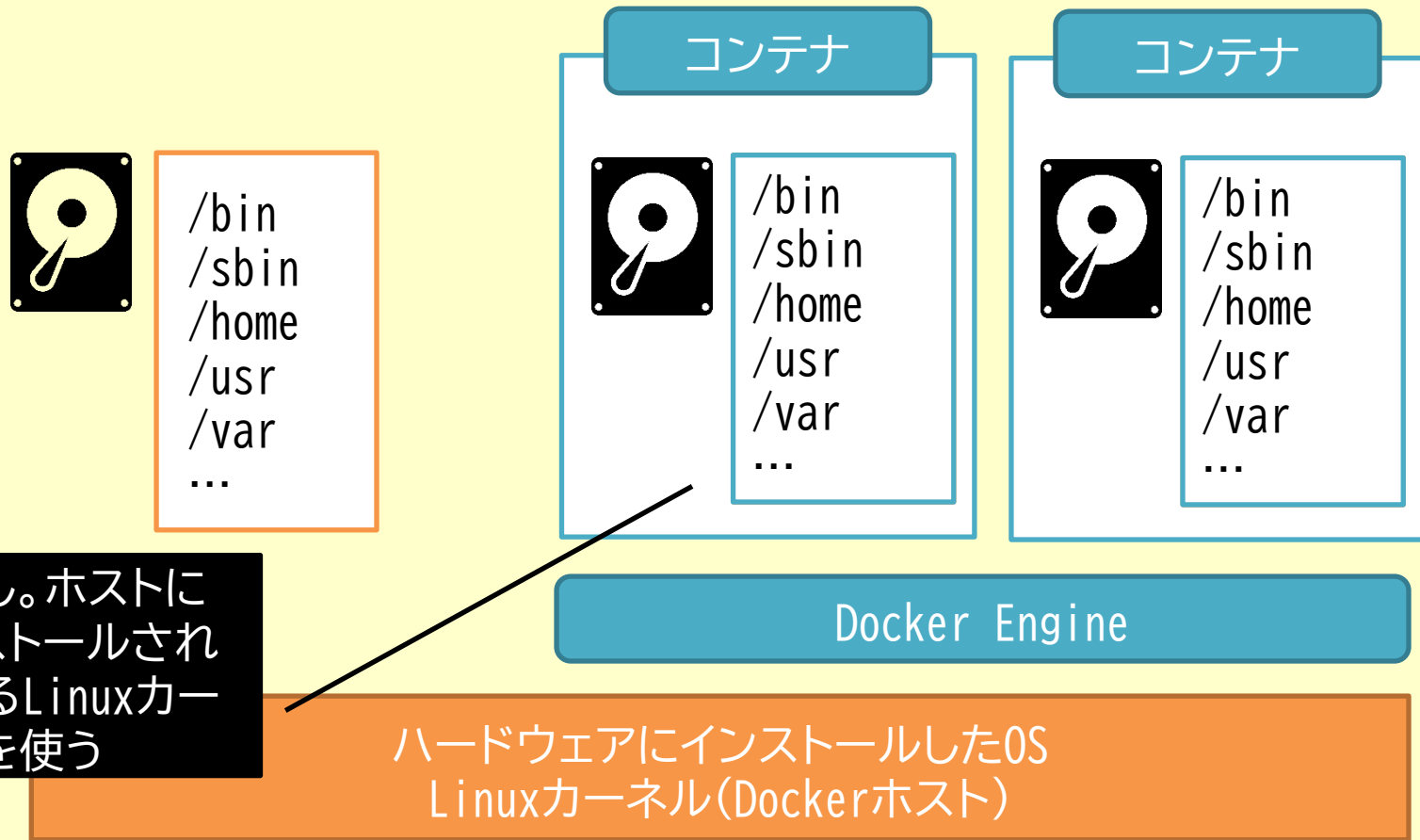


• 仮想マシンの場合





• Dockerの場合





• DockerはLinuxじゃないとダメ

コンテナのなかで
動かすものLinux



/bin
/sbin
/home
/usr
/var
...

コンテナ



/bin
/sbin
/home
/usr
/var
...

コンテナ

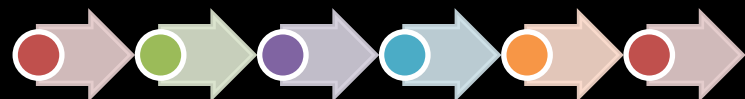


/bin
/sbin
/home
/usr
/var
...

ホストのOSも
Linux

Docker Engine

ハードウェアにインストールしたOS
Linuxカーネル(Dockerホスト)



Presented by
Mofukabur.いんく





・ ディストリビューション



/bin
/sbin
/home
/usr
/var
...

ホストのOSを
Ubuntuで構成した
場合



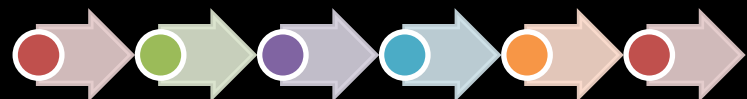
/bin
/sbin
/home
/usr
/var
...

RedHat
コンテナ

ホストである
Ubuntuに入っ
ているLinuxカー
ネルで動く

Docker Engine

Ubuntu

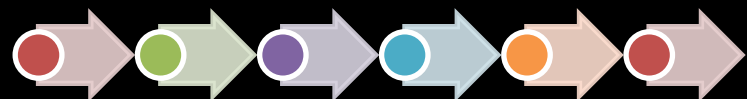


Presented by
Mofukabur.いんく





WindowsやmacOSで使うDocker

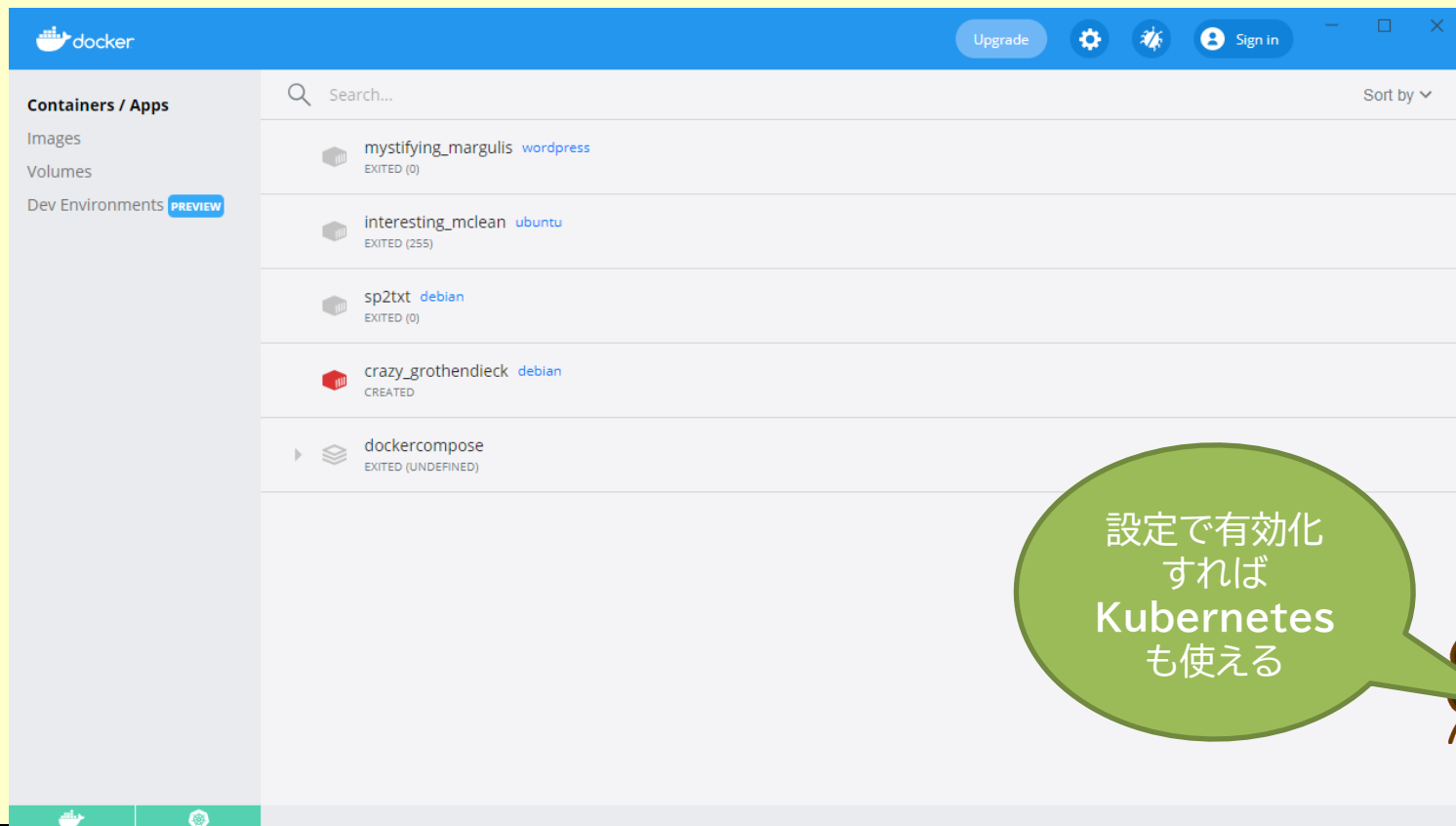


Presented by
Mofukabur.いんく

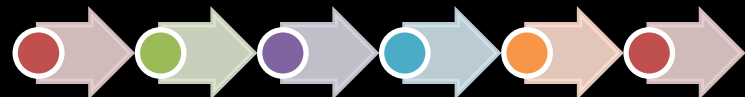




• Docker Desktop for Windows/for Mac



設定で有効化
すれば
Kubernetes
も使える



Presented by
Mofukabur.いんく





• Linuxを被せて動かす

動くのはLinuxのソフト



Windows
やmacOS
のディス
クシステ
ム

コンテナ



/bin
/sbin
/home
/usr
/var
...

コンテナ



/bin
/sbin
/home
/usr
/var
...

Linuxカーネルが
暗黙的に実行さ
れている

Docker Engine

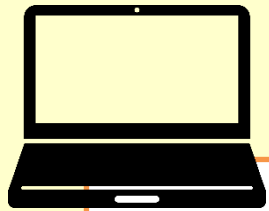
Linuxカーネル
(仮想マシン・ハイパーバイザー・WSL2など)

WindowsやmacOS





・自分のPCで好きなだけLinuxを使える



Docker DesktopをインストールしたWindowsやmacOS

Linux環境

192.168.0.1



C:¥
...

WindowsやMac

172.17.0.2



PHP環境

172.17.0.4



Ruby環境

172.17.0.3



MySQL環境

172.17.0.5

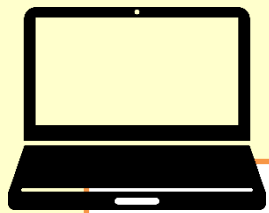


Go環境





・完成物のコピーも容易



Docker DesktopをインストールしたWindowsやmacOS

192.168.0.1



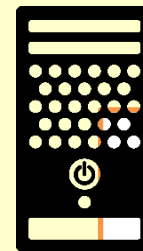
C:¥
...

WindowsやMac

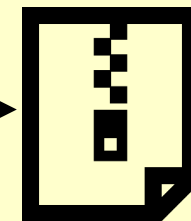
172.17.0.2



開発した
もの



コンテナ



イメージ
ファイル

本番への
コピー

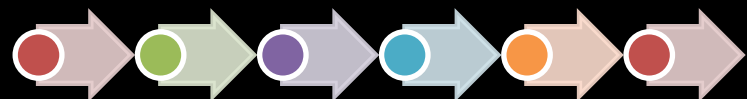
開発現場では
Dockerがフィー
バー！！(死語)

Presented
Mofukabur.いんく





コンテナとイメージ



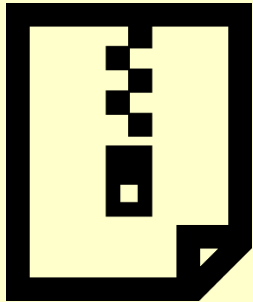
Presented by
Mofukabur.いんく





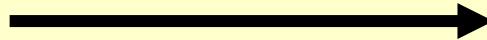
・イメージからコンテナを作る

Dockerレジストリ



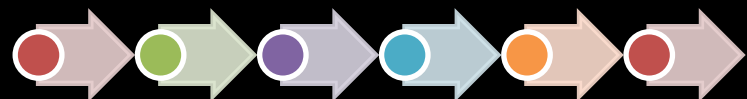
イメージ
ファイル

`docker run ...`



Debian、Ubuntu、CentOS、
Apache、Nginx、MySQL、...

※`docker run`は、`docker pull`(イメージダウンロード)、`create`(コンテナ作成)、`start`(コンテナ実行)をまとめて行うもの



Presented by
Mofukabur.いんく





• どんなイメージがあるの？

ディストリビューションだけのイメージ

Debian、Ubuntu、CentOS、Alpine、BusyBox

サーバー系のソフトが入ったイメージ

MySQL、Apache、Nginx、WordPress、...

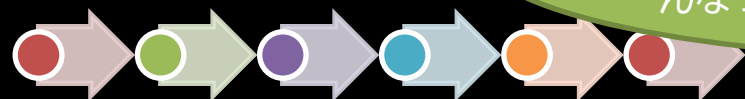
便利なツール

Cコンパイラ、TeX、(同人組版で使う)Re:VIEW、...

BusyBoxはコンテナにマウントした
ディスクをtarして取り出したいとか、そ
んなつまらないことに、よく使われる。



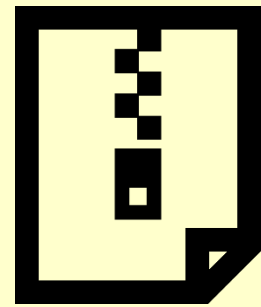
Presented by
Mofukabur.いんく





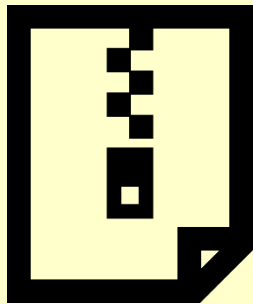
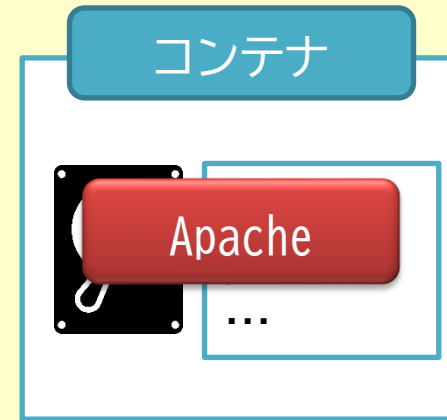
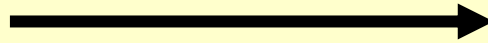
• 1コンテナ=1サービス

ApacheとMySQLを起動したいなら、それぞれ起動する。



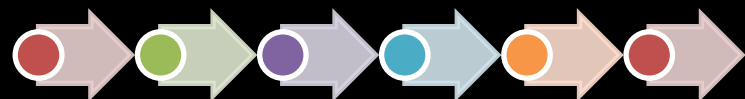
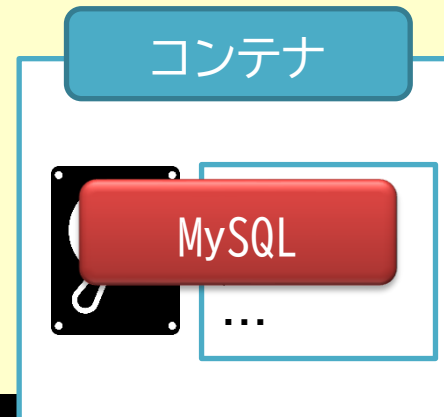
Apacheイメージ

docker run ...



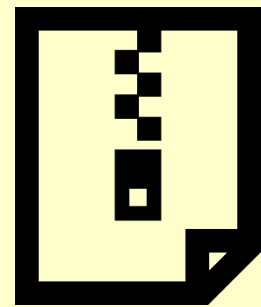
MySQLイメージ

docker run ...



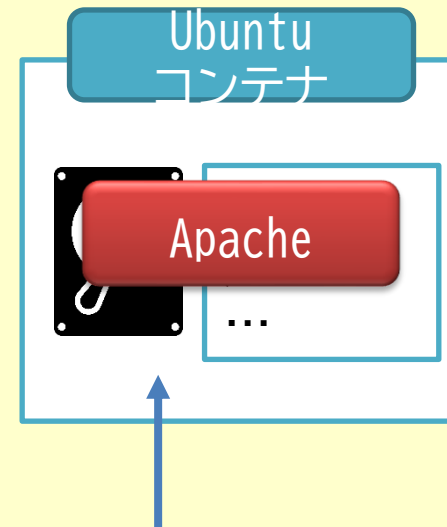
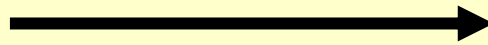


・ 自前で作ってもOK



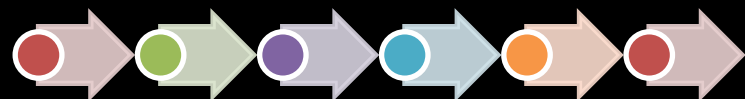
Ubuntuイメージ

`docker run ...`



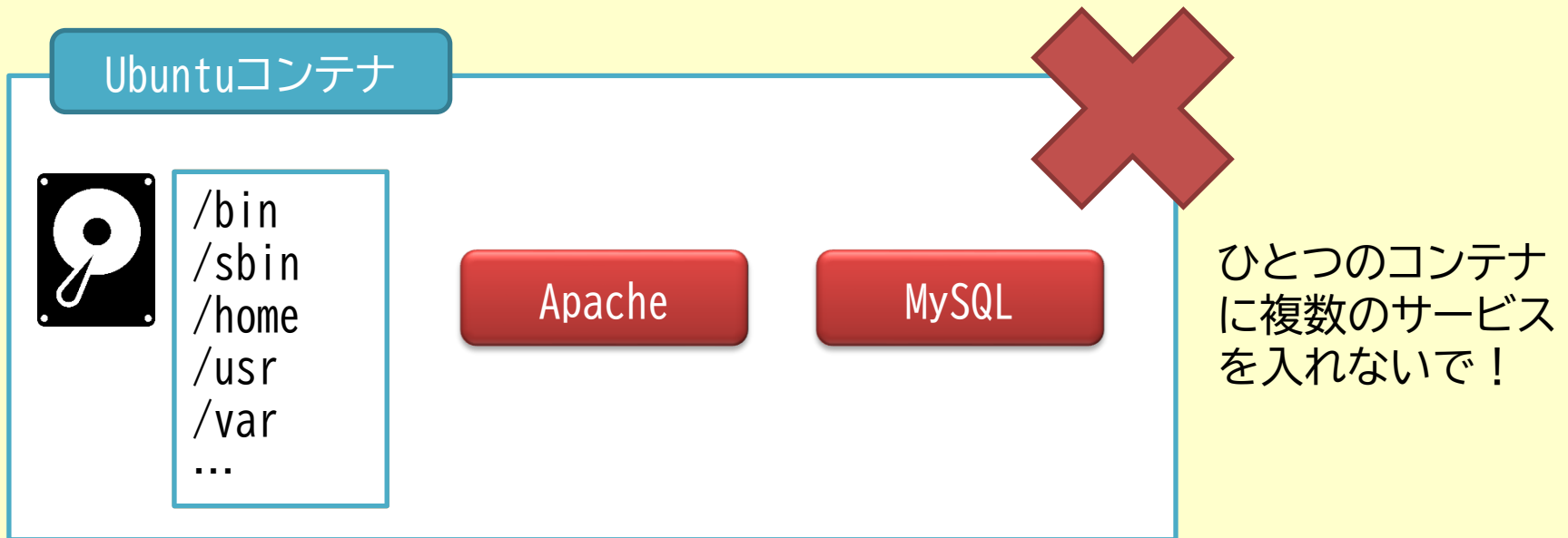
コンテナのなかに入ってaptなどでインストールする。

ただしデーモンの起動方法が違う。`systemctl`とか使えないので、出来合のイメージ使うほうが簡単(0SC京都のアーカイブ見てね！)

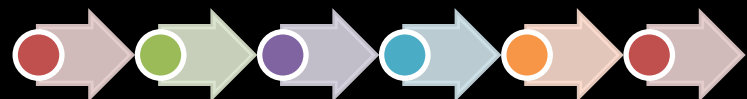




• 1コンテナに複数サービスは入れられない



Ubuntuなどの素のコンテナを作って、自前でApacheとかMySQLをインストールする
→相当工夫しないと、どちらかひとつしかデーモンとして動かない
→コンテナではデーモンとして動かせるのは1本だから。詳細はOSC京都のアーカイブで！





・イメージを探す

キーワード検索する

The screenshot shows the Docker Hub search results for the keyword 'httpd'. The search bar at the top contains 'httpd'. Below the search bar, the 'Containers' tab is selected. The search results show 10,338 results for 'httpd'. The first result is 'httpd', which is an 'Official Image' from 'The Apache HTTP Server Project'. It has over 10 million downloads and 3.6K stars. The image is updated 14 minutes ago. The architecture is listed as 'Container', 'Linux', 'ARM 64', 'x86-64', 'IBM Z', 'ARM', 'mips64le', '386', and 'PowerPC 64 LE'. The application infrastructure is 'Application Infrastructure'. Below this, another result 'centos/httpd-24-centos7' is partially visible.

Filters

1 - 25 of 10,338 results for **httpd**. [Clear search](#)

Most Popular

Images

☐ Verified Publisher ⁱ

☐ Official Images ⁱ
Official Images Published By
Docker

httpd Official Image

Updated 14 minutes ago

The Apache HTTP Server Project

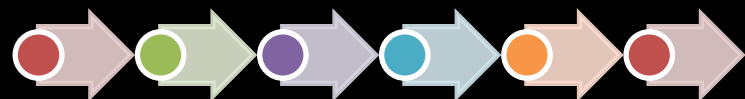
Container Linux ARM 64 x86-64 IBM Z ARM mips64le 386 PowerPC 64 LE

Application Infrastructure

centos/httpd-24-centos7

オフィシャルかを確認する(イメージ名が「ユーザー名/イメージ名」ではなくて、イメージ名のみ)

ダウンロード数を確認する



Presented by
Mofukabur.いんく





• タグ

latest

最新版

2

2系統の最新版

2.4.48

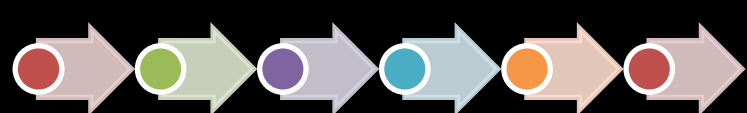
2.4.48版

2.4.48-buster

2.4.48のbuster版

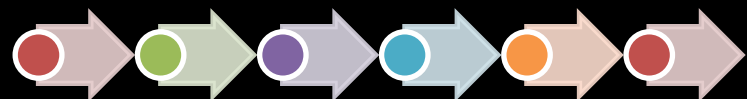
httpd:2.4.48

イメージの作成者が決めた識別子
(バージョン番号が慣例だが、そうとは限らない)





コンテナの実行



Presented by
Mofukabur.いんく





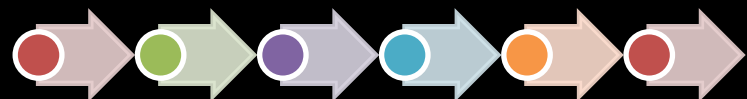
• コンテナの実行

①1回限り(フォアグラウンド)

何かコマンドを1回実行して終わらせる。コンパイル作業やTeXなどのコマンド実行、シェルを使って何かするなどなど。

②実行しっぱなしにするもの(バックグラウンド)

サーバー系。Apache、Nginx、MySQLなどさまざまなサービスを動かしたい場合。

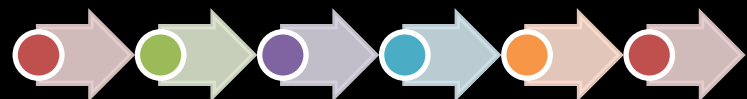




• フォアグラウンドの例

「hello-world」の例

```
docker run hello-world
```





コマンドプロンプト

```
C:\Users\osawa>docker run hello-world
```

```
Hello from Docker!
```

```
This message shows that your installation appears to be working correctly.
```

```
To generate this message, Docker took the following steps:
```

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
(amd64)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

```
To try something more ambitious, you can run an Ubuntu container with:
```

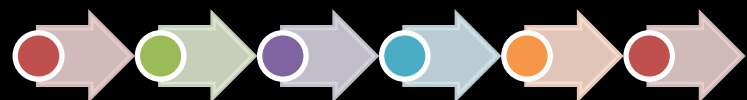
```
$ docker run -it ubuntu bash
```

```
Share images, automate workflows, and more with a free Docker ID:
```

```
https://hub.docker.com/
```

```
For more examples and ideas, visit:
```

```
https://docs.docker.com/get-started/
```



Presented by
Mofukabur.いん<





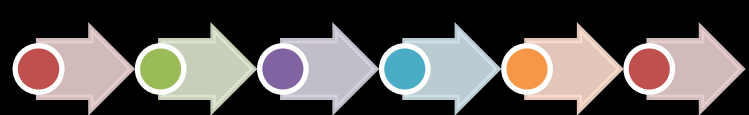
• コンテナの終了

コンテナは、**コマンドの実行が終わると終了する**

```
cmd コマンドプロンプト

C:\Users\Yosawa>docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
C:\Users\Yosawa>
```

docker psコマンドは、実行中のコンテナ一覧を見る。docker run hello-worldでは、内部の「hello-worldを実行するコマンド」を実行するだけなので、それが終わればコンテナ終了する





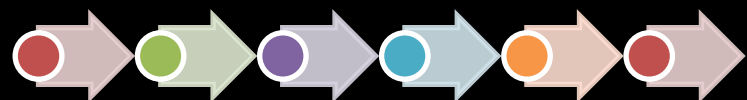
・残った残骸に注意

残骸はディスクを消費する！

```
C:\ コマンドプロンプト

C:\Users\Yosawa>docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
C:\Users\Yosawa>docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
4818ff4a2c20   hello-world  "/hello"                3 minutes ago  Exited (0)    3 minutes ago
167d1275689e   hello-world  "/hello"                16 minutes ago  Exited (0)    15 minutes ago
ann
792e4a9418f2   hello-world  "/hello"                16 minutes ago  Exited (0)    16 minutes ago
lion
```

docker ps コマンドに「-a」を付けると、停止したコンテナが残骸として残っていることがわかる。不要なら、docmer rm コマンドで削除する。
もしくは、起動時に、「docker run --rm hello-world」とすると、コンテナ終了にともないコンテナが自動的に削除される

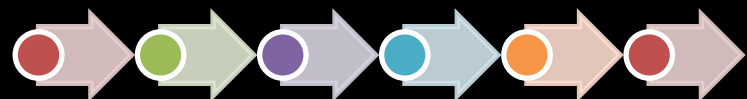




• 起動時には名前を付ける

起動時には、`--name`オプションで名前を付けておくことを推奨。
付けないと、適当な名前が付けられるため、コンテナの終了や
破棄などの際、名前で指定できずに辛い。

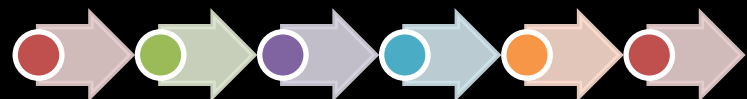
```
docker run --name mycontainer hello-world
```





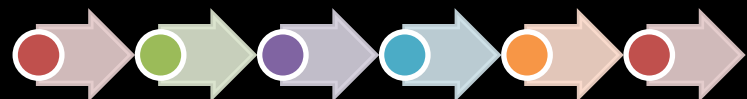
• コンテナの名前

STATUS	PORTS	NAMES
Exited (0) 7 seconds ago		mycontainer
Created		confident_wozniak
Exited (0) 2 hours ago		sad_raman
Exited (0) 2 hours ago		funny_lehmann
Exited (0) 2 hours ago		pedantic_elion
Exited (0) 4 days ago		mystifying_margulis





Apacheを使おう



Presented by
Mofukabur.いんく

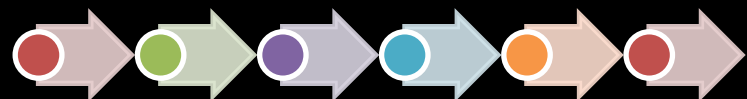




• Apacheの実行例

```
docker run -d httpd
```

httpdというのは、
Apacheが入った
イメージ名





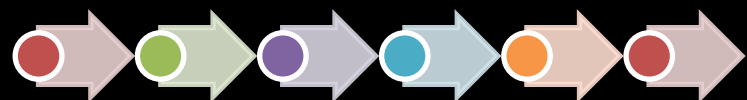
・ 起動すると…

コンテナID

```
C:\Users\Yosawa>docker run -d httpd  
177b33bf60d09f8ef2526836c856d176a7532018372030d52d590cbd3fcd6d80
```

```
C:\Users\Yosawa>docker ps  
CONTAINER ID    IMAGE    COMMAND                  CREATED          STATUS  
177b33bf60d0    httpd    "httpd-foreground"      24 seconds ago  Up 21 seconds
```

コンテナID

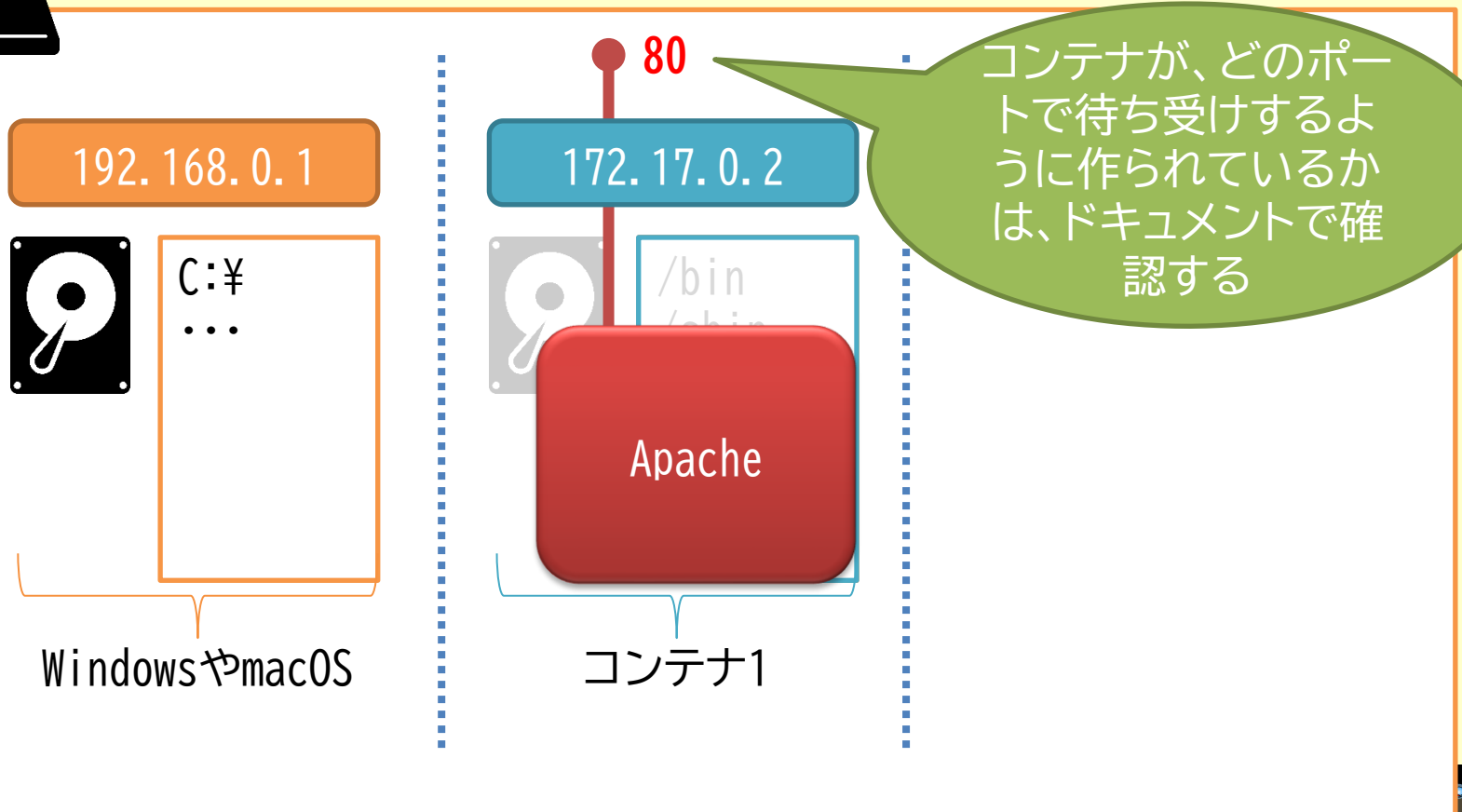
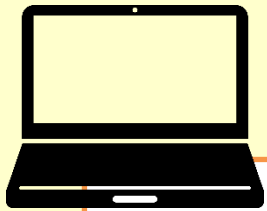


Presented by
Mofukabur.いんく



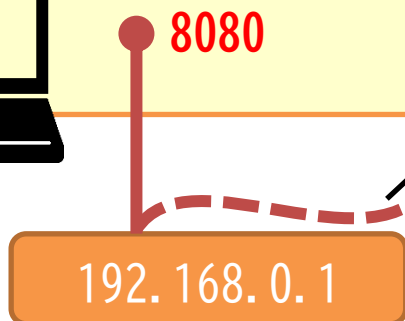
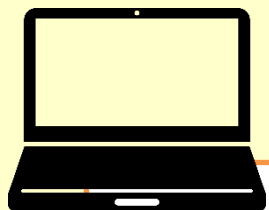


• これではつながらない

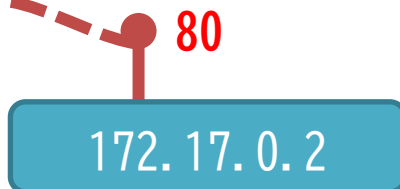




・外からつなぐ



WindowsやmacOS



コンテナ1

```
docker run -d -p 8080:80 httpd
```

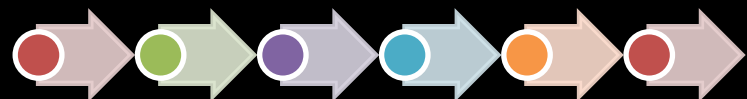
あとから-p
の指定変更
不可





- 停止するには

docker stop コンテナ名 or コンテナID





• コンテナの停止

```

C:\> コマンドプロンプト

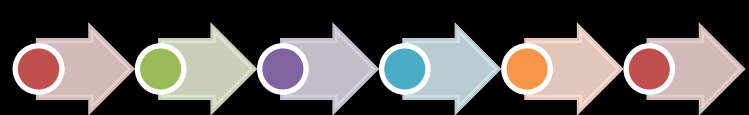
C:\Users\Yosawa> docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
177b33bf60d0        httpd              "httpd-foreground" 2 minutes ago

C:\Users\Yosawa> docker stop 177b
177b

C:\Users\Yosawa> docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED        STATUS        PORTS
177b33bf60d0        httpd              "httpd-foreground" 3 minutes ago  Up 3 minutes  80/tcp

C:\Users\Yosawa> docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED        STATUS        PORTS
177b33bf60d0        httpd              "httpd-foreground" 3 minutes ago  Up 3 minutes  80/tcp
88-178878f1-11      httpd              "httpd-foreground" 5 minutes ago  Up 5 minutes  80/tcp

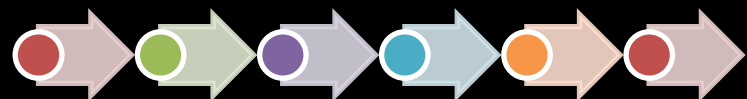
```





- -pオプションを付けて起動

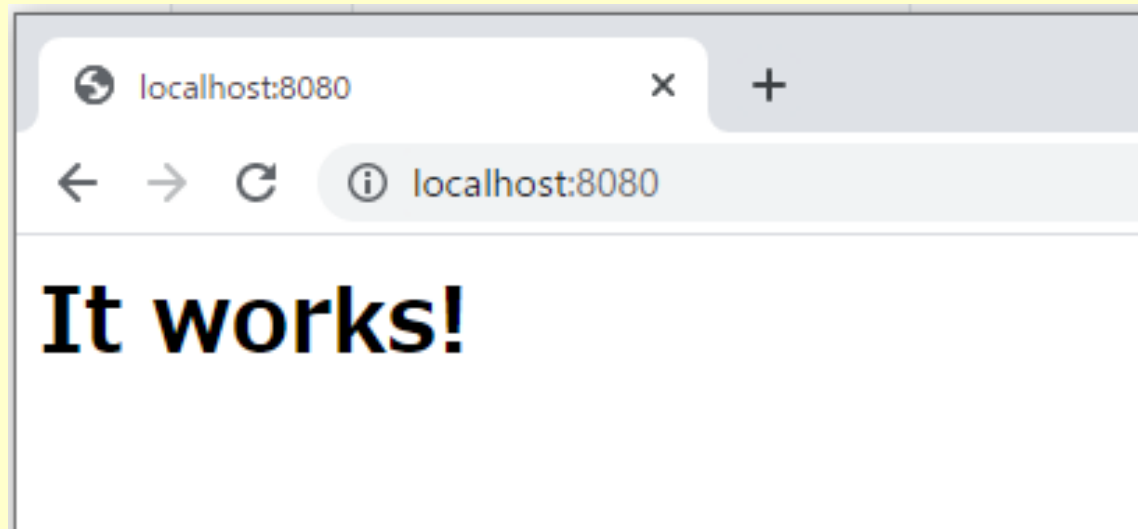
```
docker run -d -p 8080:80 httpd
```





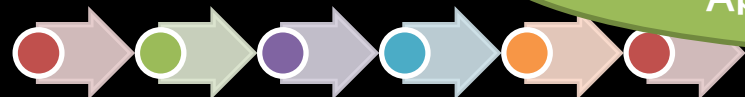
- ブラウザでつながる

`http://localhost:8080/`にアクセス



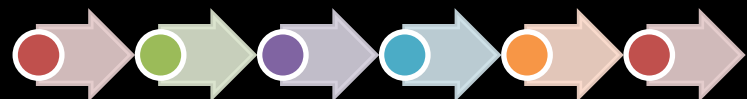
Docker Desktopなら、まさに、
WindowsやmacOSで「Linuxの
Apacheが動いた！」ってこと。

Presented by
Mofukabur.いんく



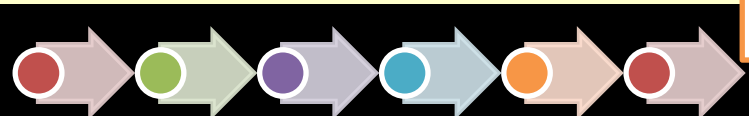
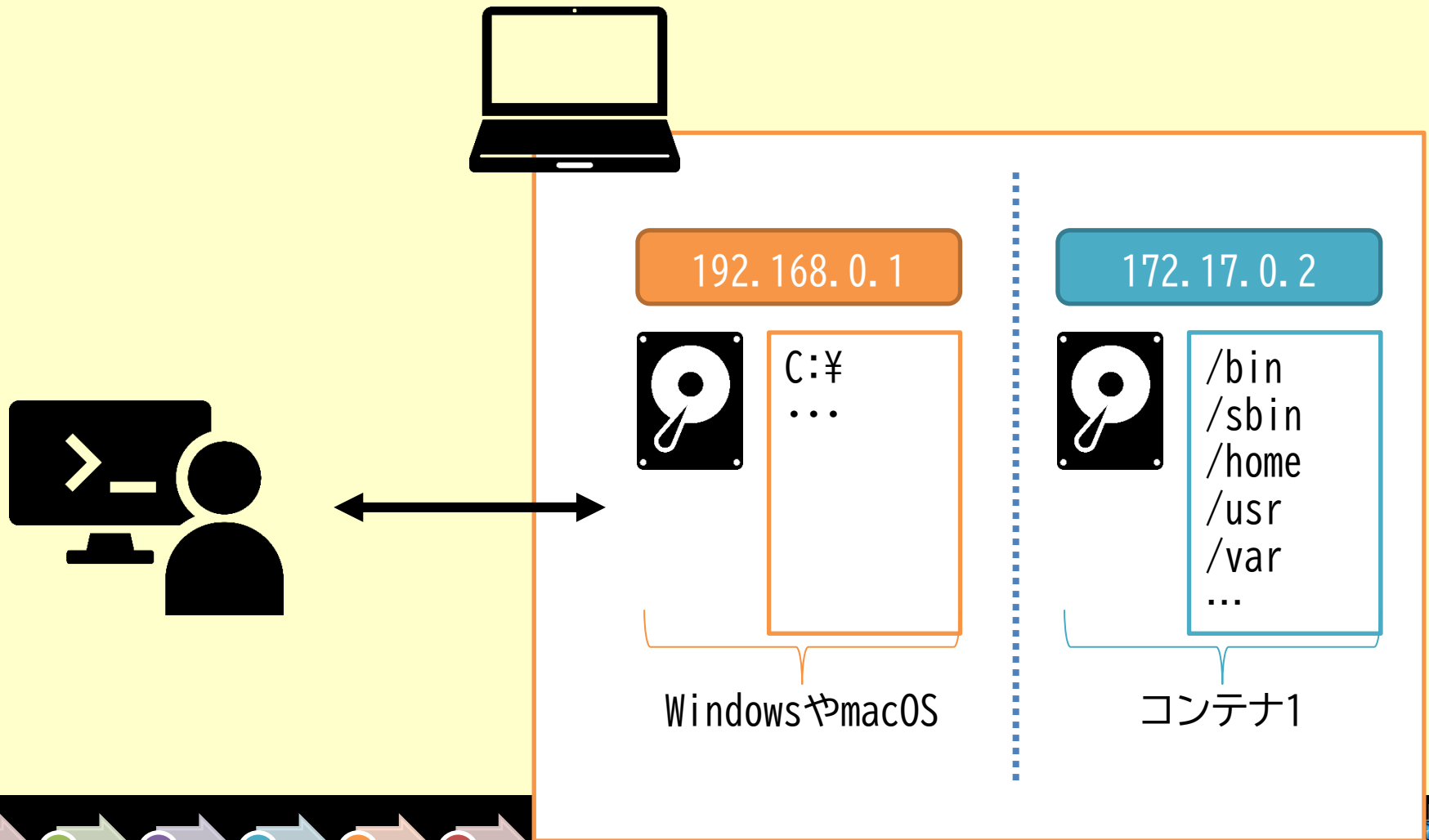


コンテナの中身をいじる



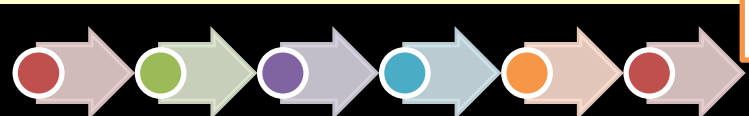
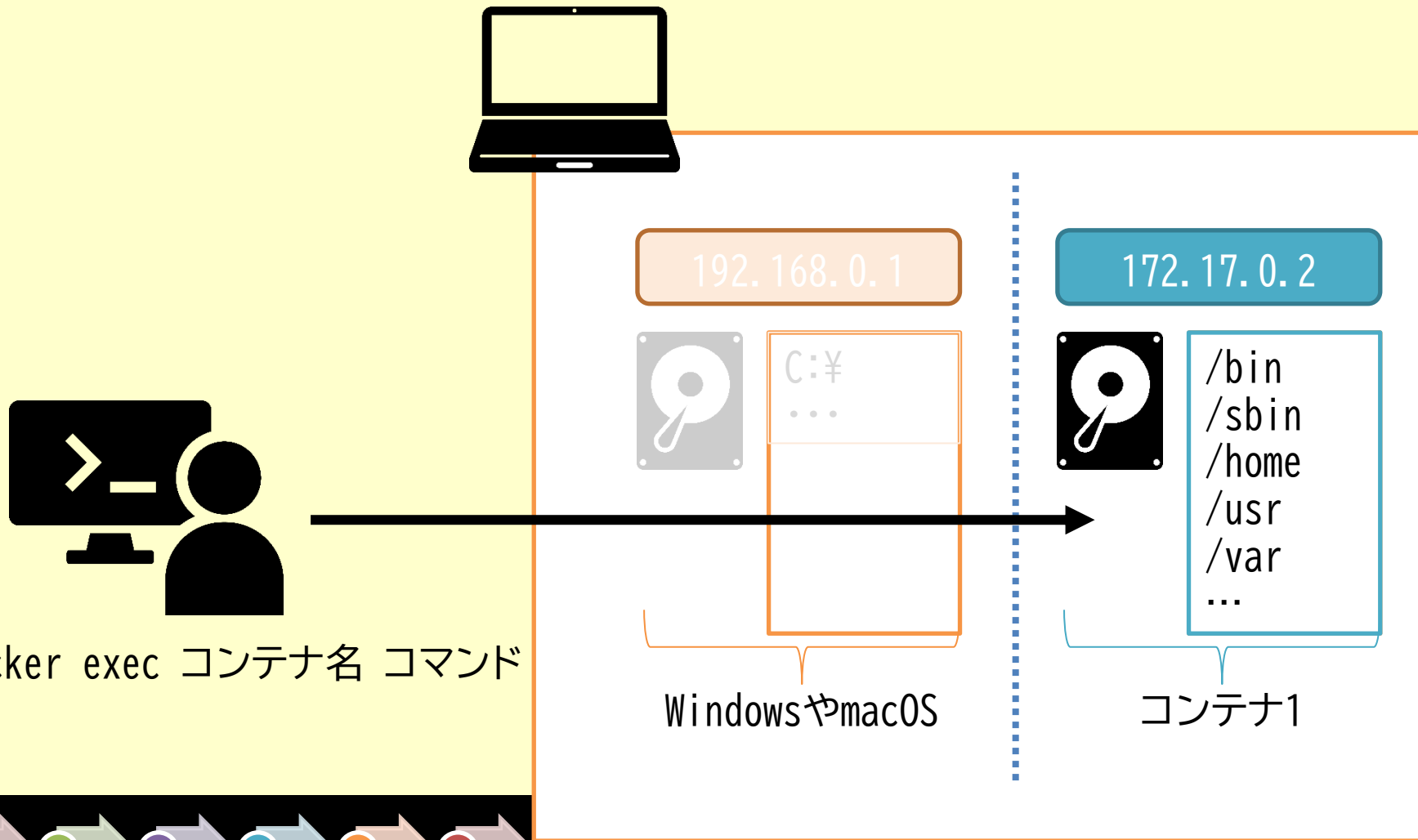


・ 隔離されている





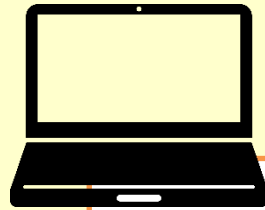
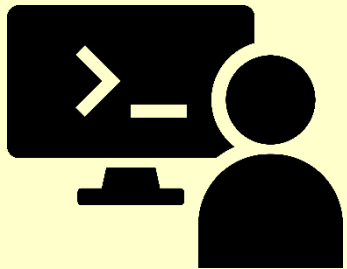
• 任意のコマンドを実行する





・シェルで操作する

-tを指定しないとコントロールキーやカーソルキーが効かない



192.168.0.1



C:¥
...

WindowsやmacOS

172.17.0.2

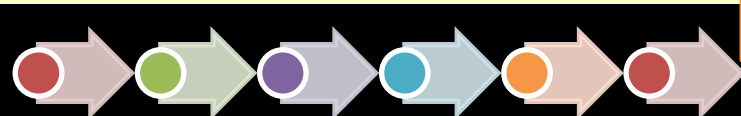


/bin
/sbin
/home
/usr
/var
...

コンテナ1

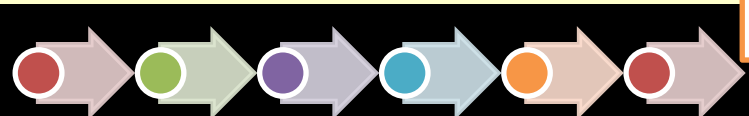
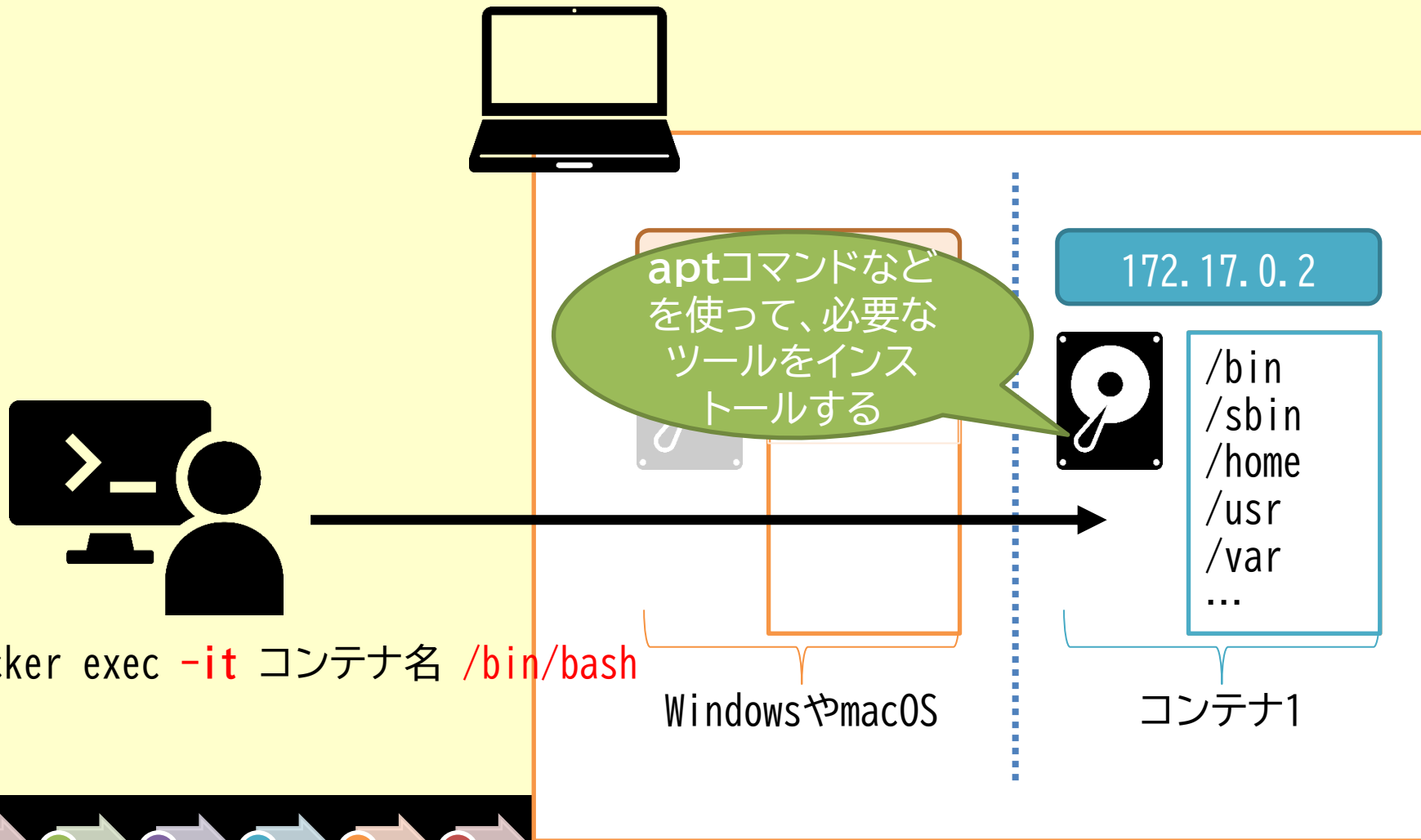


`docker exec -it コンテナ名 /bin/bash`



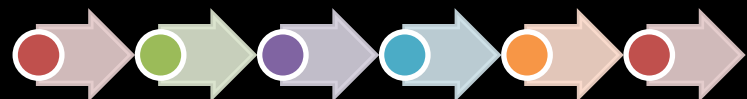


・ 必要なツールは自分でインストール





コンテナのファイル操作する

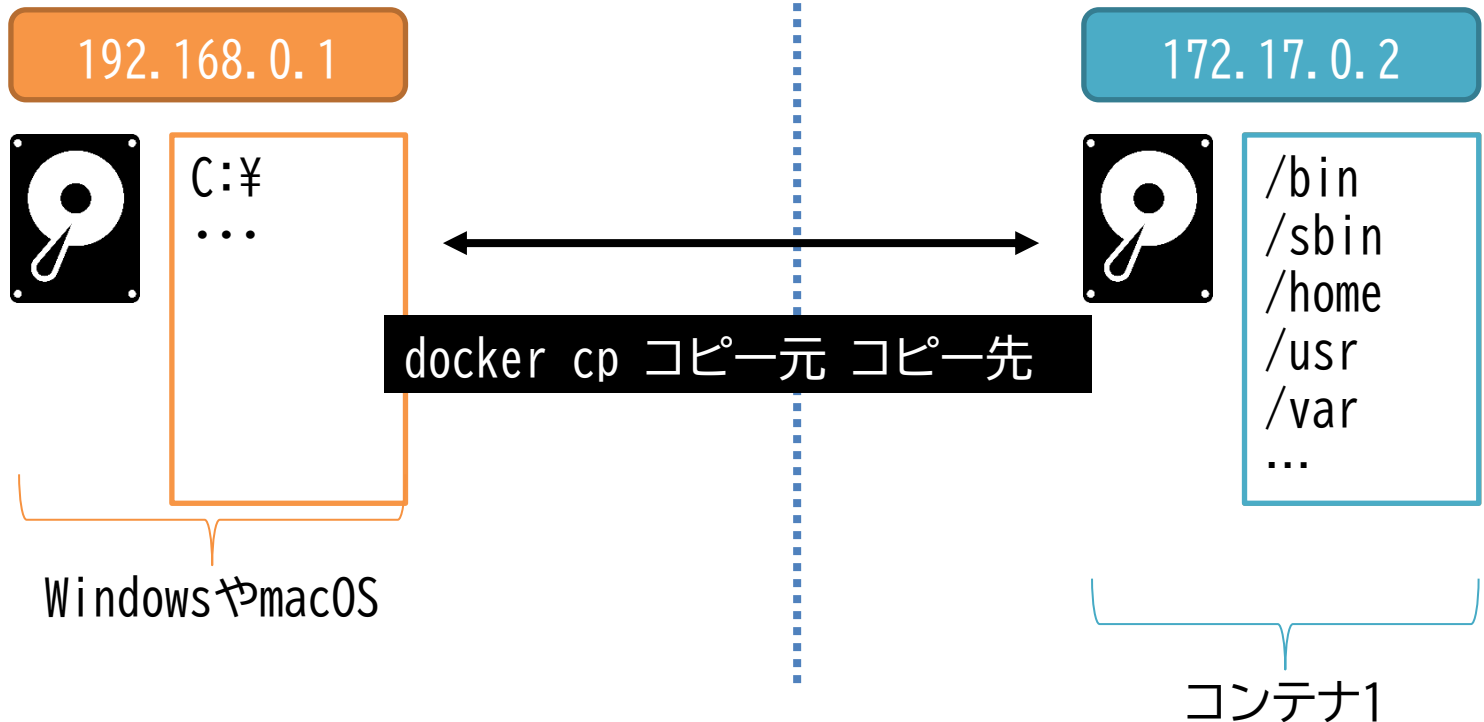
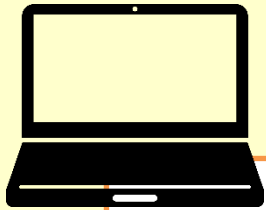


Presented by
Mofukabur.いんく



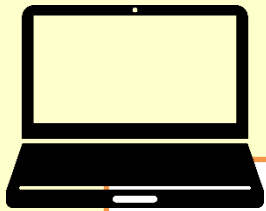


• ホストとコンテナとのファイルコピー





• バインドマウント



Windowsの世界

192.168.0.1



C:¥
...
Users¥
foo
Documents and Settings
hoge

ここをVSCodeとかWindowsのエディタで
編集すると、Apacheから見える

Linuxの世界

172.17.0.2



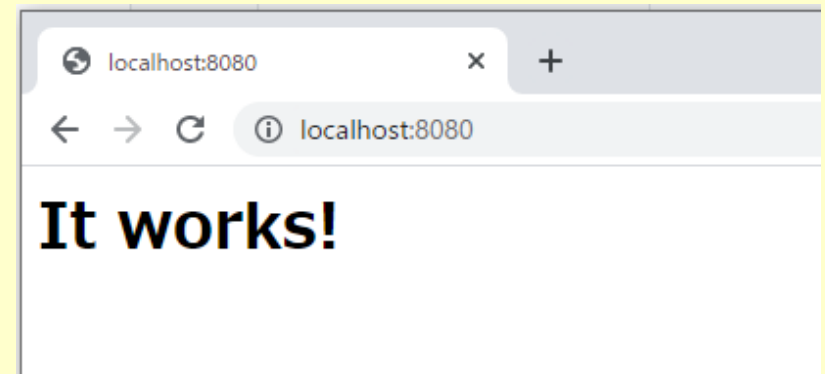
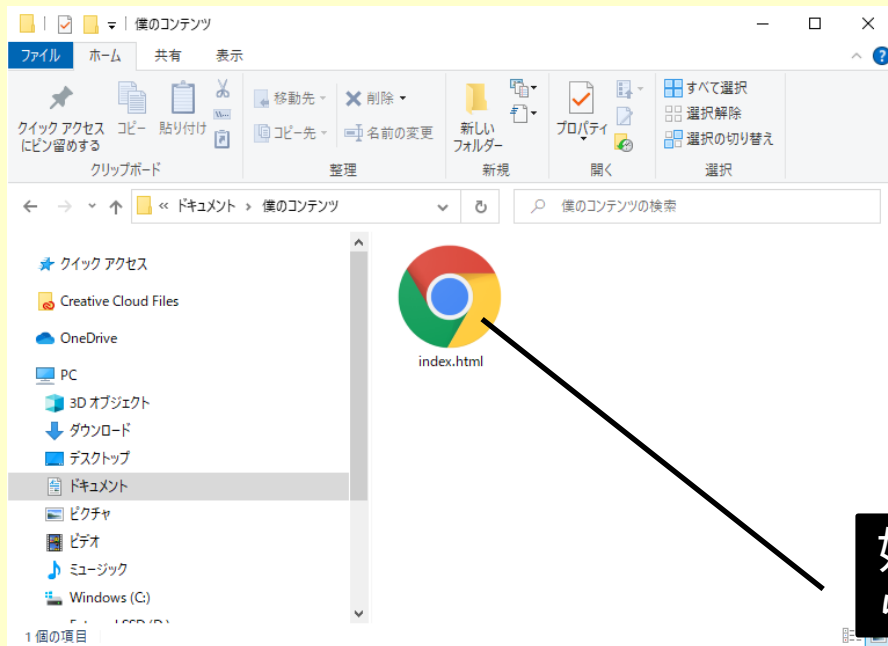
/bin
/sbin
/home
/usr
local
apache2
htdocs
/var
...

```
docker run ...  
-v C:¥Users¥...hoge:/usr/local/apache2/htdocs
```



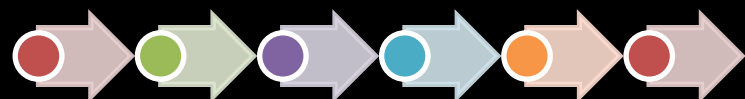


・ はかどる



好きなエディタで編集すればApacheから見えるコンテンツが変わる

/usr/local/apache2/htdocsをボリュームマウントしたフォルダ

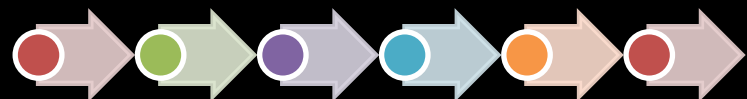


Presented by
Mofukabur.いんく





余談)ボリュームマウント

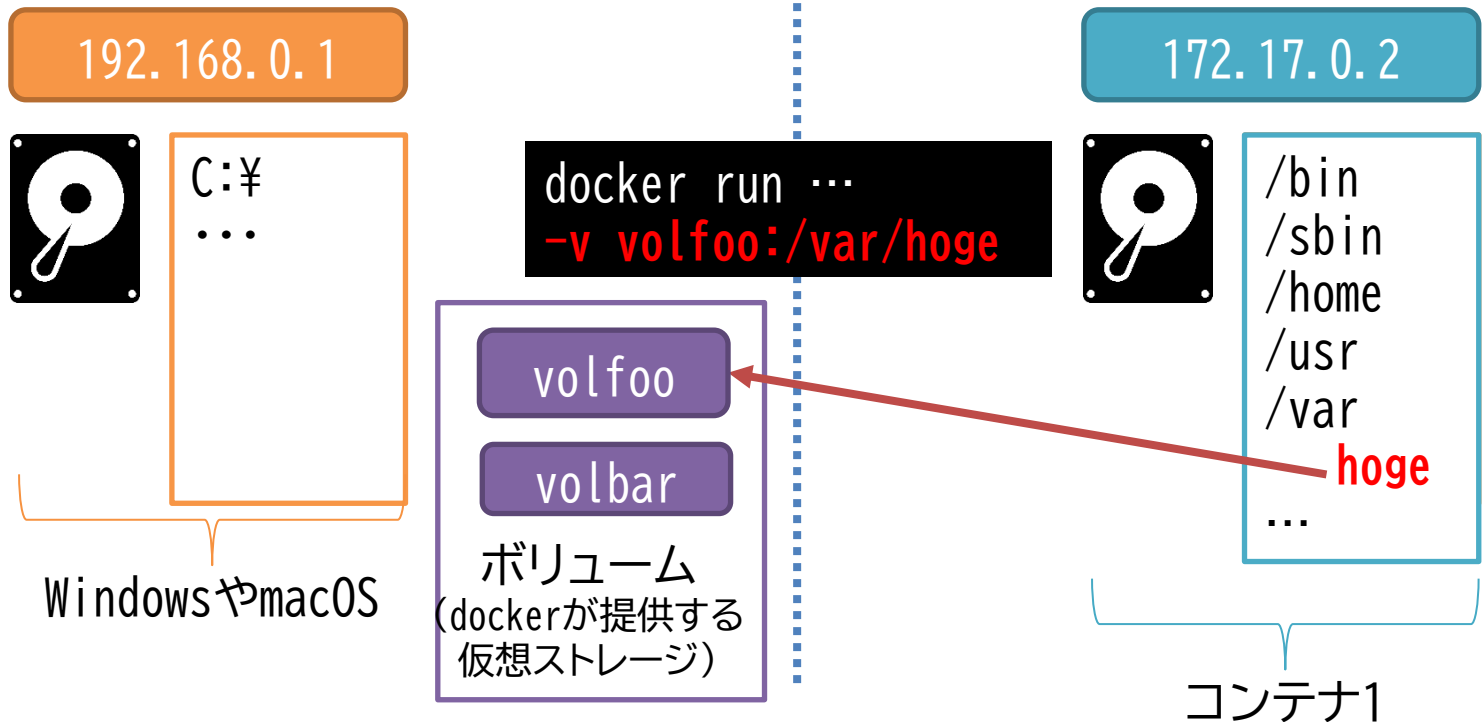
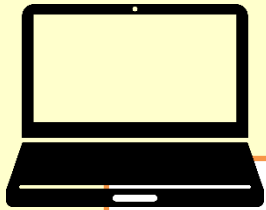


Presented by
Mofukabur.いんく



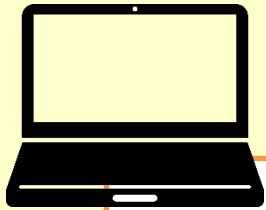


・ ボリュームマウント





• ボリュームマウント



Linuxがインストールされた
PC(またはサーバ)

192.168.0.1



/bin
/sbin
/home
/usr
/var
...

メインの環境
(ホスト)

volfoo

volbar

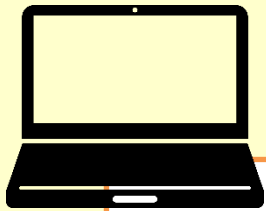
ボリューム
(dockerが提供する
仮想ストレージ)

ここに、/var/hogelに
保存した内容が格納
されて残っている





・ ボリュームマウント



Linuxがインストールされた
PC(またはサーバ)

前に保存しておいたデータを引き続き、
コンテナで使える

192.168.0.1



/bin
/sbin
/home
/usr
/var
...

メインの環境
(ホスト)

volfoo

volbar

ボリューム
(dockerが提供する
仮想ストレージ)

172.17.0.2



/bin
/sbin
/home
/usr
/var
...

コンテナ1

hoge

DBとか共有
ファイル置き
場によく使
われる



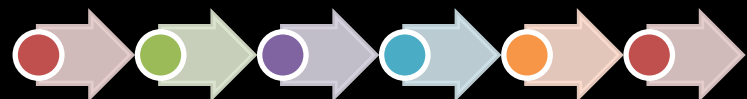


• データは外だしが基本

コンテナが削除されると、そのなかのデータが消える(当たり前)(あくまでも削除、rmした場合。stopした段階では消えない)。

よって、データはマウントして外だしする

コンテナを削除すると、コンテナのデータは消えるよ。
永続化したいデータは外に出してね

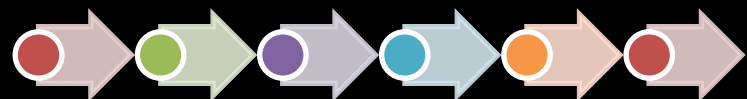


Presented by
Mofukabur.





開発現場で使うDocker



Presented by
Mofukabur.いんく





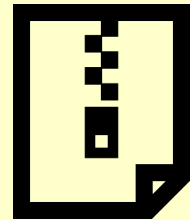
・ チームリーダーはイメージを作る

開発環境のイメージを作って、配布してあげる。

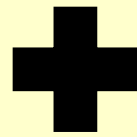
配布先はプログラマのほかデザイナーも考えられる。



チームリーダー

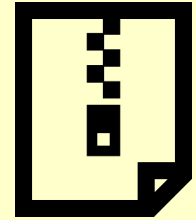


基のイメージ
(Debianとかの
ベースイメー
ジやApacheな
どのイメージ)

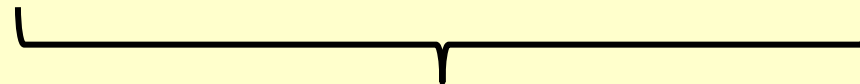


追加したファイル

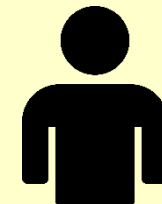
実行コマンド



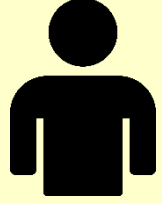
配布イメージ



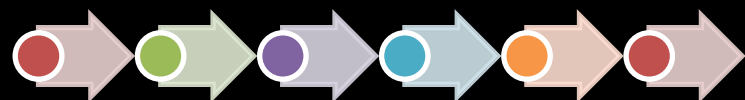
Dockerfileの書き方、ビルド方法な
どの知識が必要



プログラマ



デザイナー

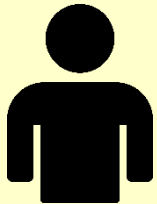




• 使うだけのユーザー

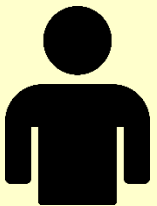
プログラマやデザイナーは、docker runやstop、rmなどの「使い方」だけ知ればOK。

(複数のコンテナを起動するdocker composeの使い方も理解していればよりよい)



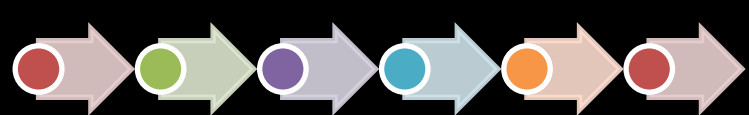
プログラマ

```
docker run -p 8080:80  
-v C:\Users\¥...hoge:/usr/local/apache2/htdocs https
```



デザイナー

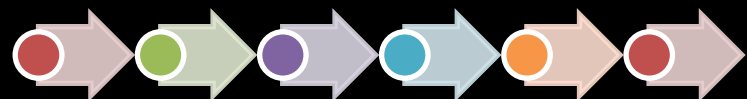
-pでポートの公開がわかること
-vで自分が操作するフォルダにマッピングできること





• CI/CDにおける活用

テストの際の実行サーバとしてDockerを採用する

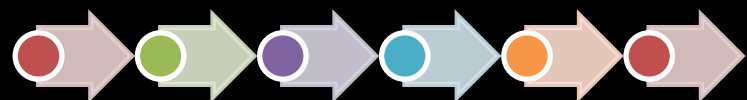


Presented by
Mofukabur.いんく





まとめ



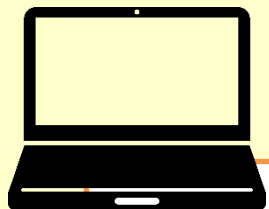
Presented by
Mofukabur.いんく





・ 隔離

dockerコマンド
で、穴開けして操
作していく



Linuxがインストールされた
PC(またはサーバ)

ポイント①
独立したネットワーク

192.168.0.1



/bin
/sbin
/home
/usr
/var
...

メインの環境
(ホスト)

172.17.0.2



/bin
/sbin
/home
/usr
/var
...

コンテナ1

172.17.0.3



/bin
/sbin
/home
/usr
/var
...

コンテナ2

※IPアドレスはすべて仮

ポイント②
独立したストレージ





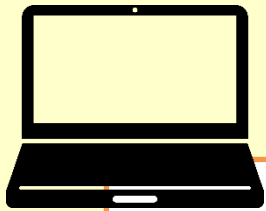
• docker runさえ理解できれば…

- pオプション ポートの設定
- vオプション マウントの設定

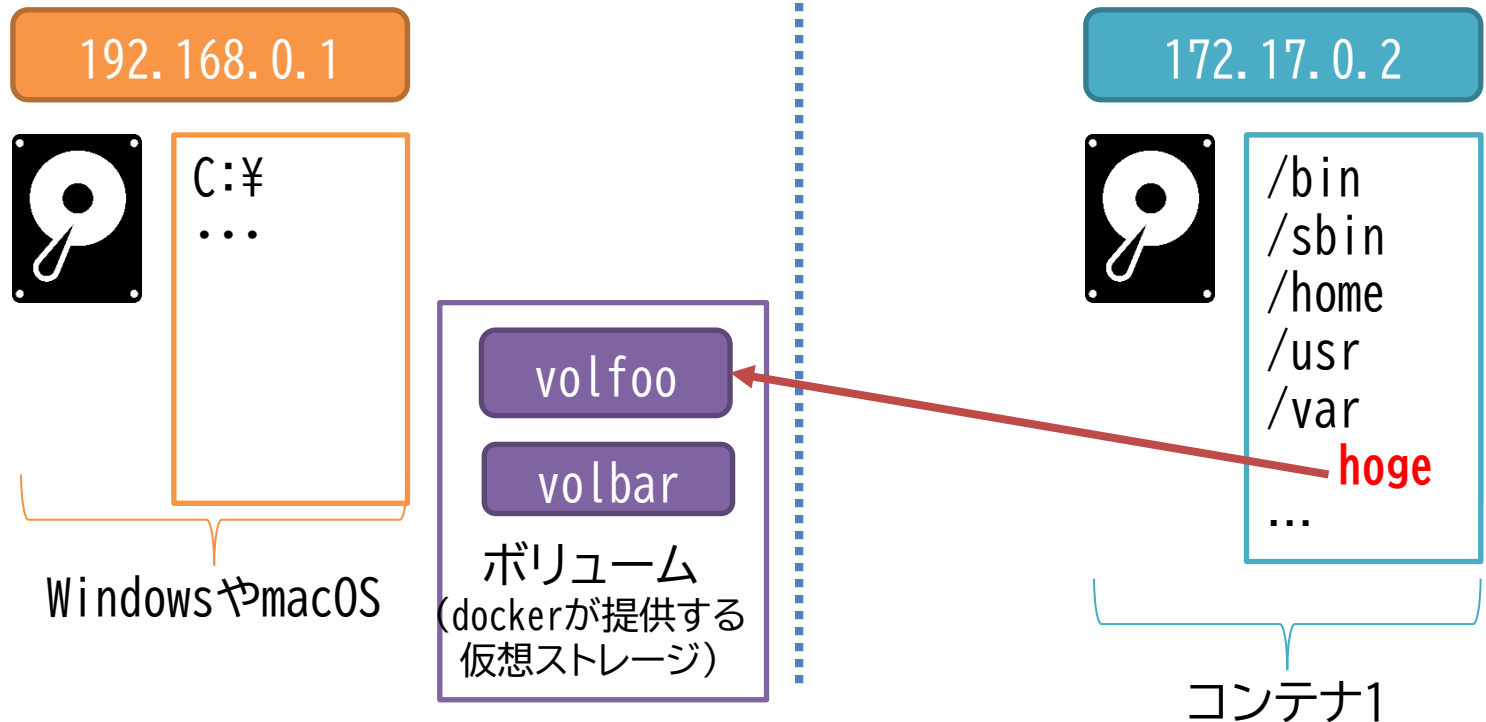




• データは外に出す

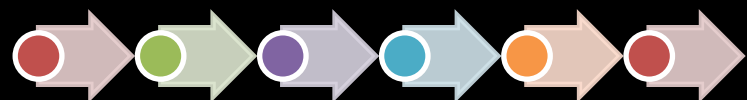


Linuxがインストールされた
PC(またはサーバ)





今後の活動



Presented by
Mofukabur.いんく



モウフカブール



小笠原種高



モウフカブール

<http://mofukabur.com/>



大澤文孝



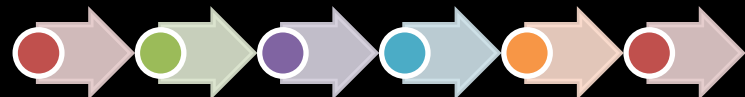
同人活動

次回参加イベント
未定

モウフカブール
(<http://mofukabur.com/>)



技術書典、技術書博覧会(技書博)、コミケなどで活動中。
技術同人のほかの音楽CD、写真集などを頒布しています。



Presented by
Mofukabur.いんく





通販でも買える

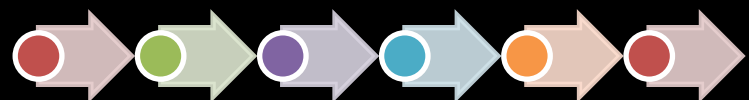
- とらのあな
- メロンブックス
- booth

など



【期間外は電子のみ】

• 技術書典オンラインマーケット



Presented by
Mofukabur.いんく



音楽CD



恋するデータベース

- ・海とポスグレ
- ・MyLove, MySQL
- ・ぼくの大好きなMaria



メロンブックスに委託中。
ボーナストラックにDBの曲
が入ってます。

フルバージョンは、YouTubeで
視聴できます。
モウフカブールで検索！

デバッグ音頭は、
YouTubeで検索！



Presented by
Mofukabur.いんく



アンケートのご協力 お願いいたします

https://bit.ly/ODC2021_Form



Follow me

 @sour23

(所要時間3分程度)



https://bit.ly/ODC2021_Form



AWSで学ぶ
docker入門書



Follow me

 @sour23

