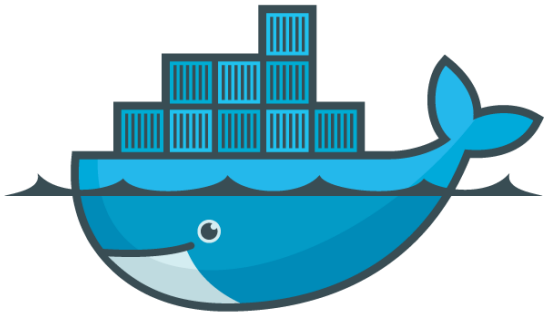
AITC
先端IT活用推進コンソーシアム

使ってわかる
今どきのdocker超入門

2016年2月18日

先端IT活用推進コンソーシアム
クラウド・テクノロジー活用部会
勉強会資料

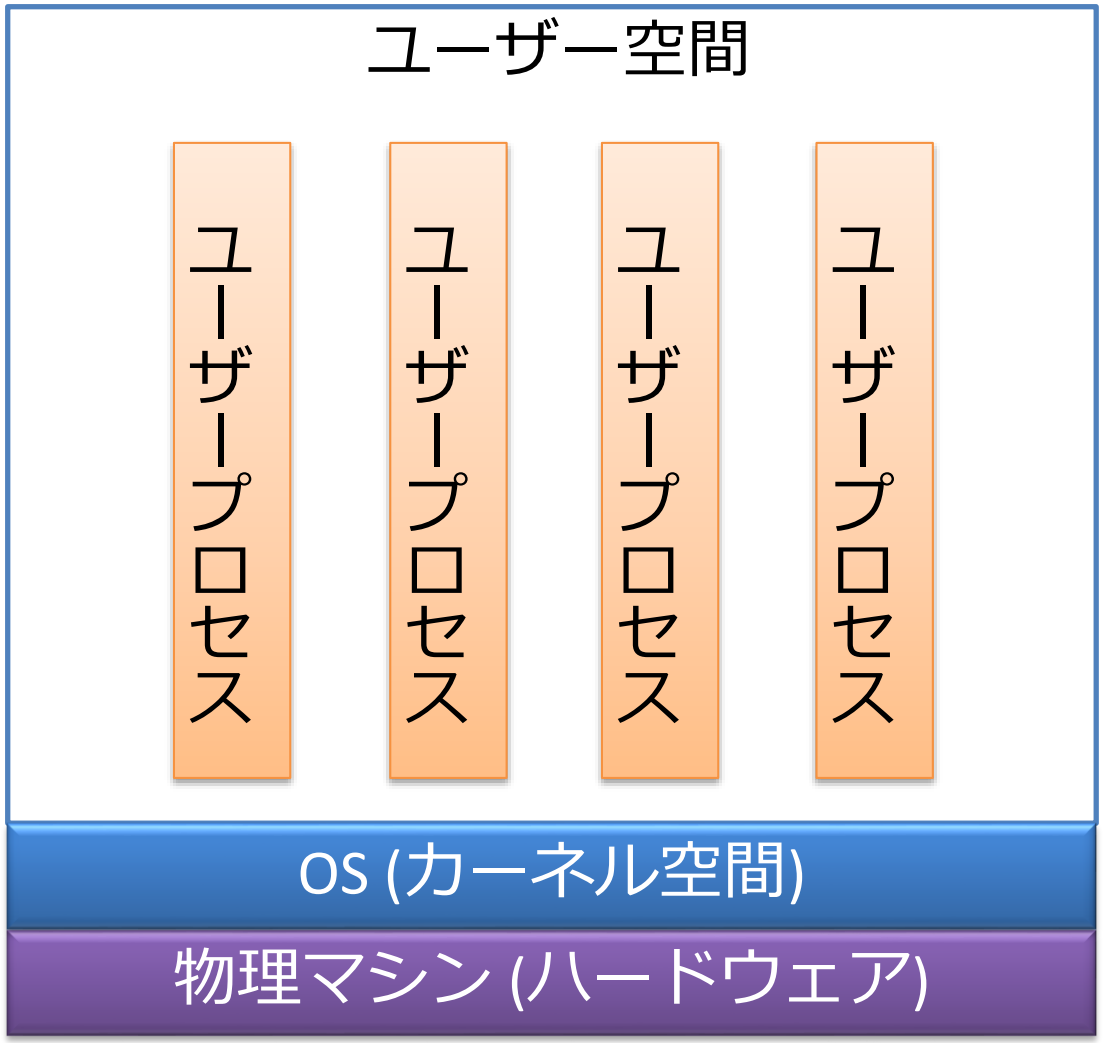
岡村　和英（株式会社テクリエ）



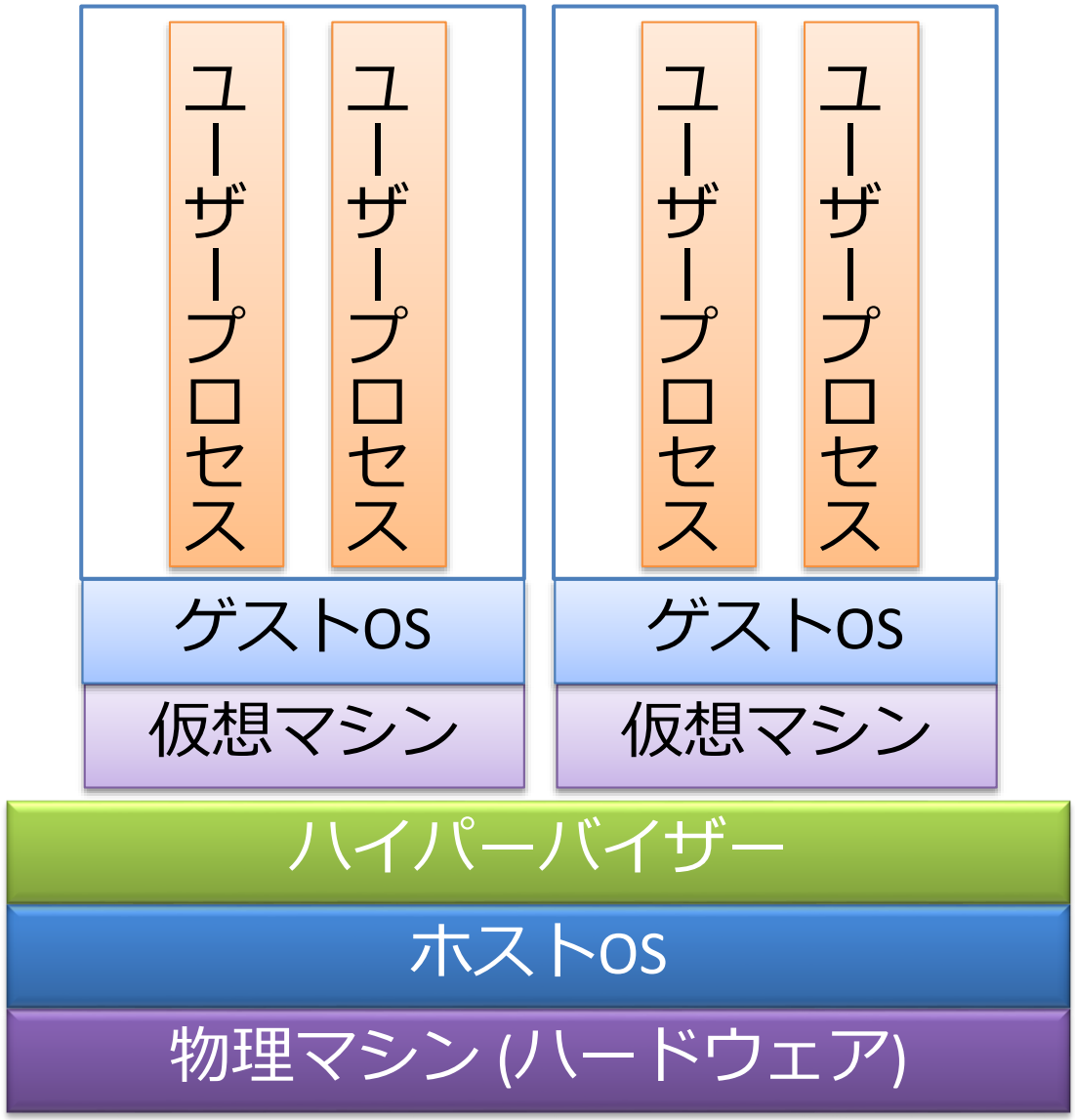
docker ?

Linuxコンテナによる
リソース分割を利用した
仮想化環境

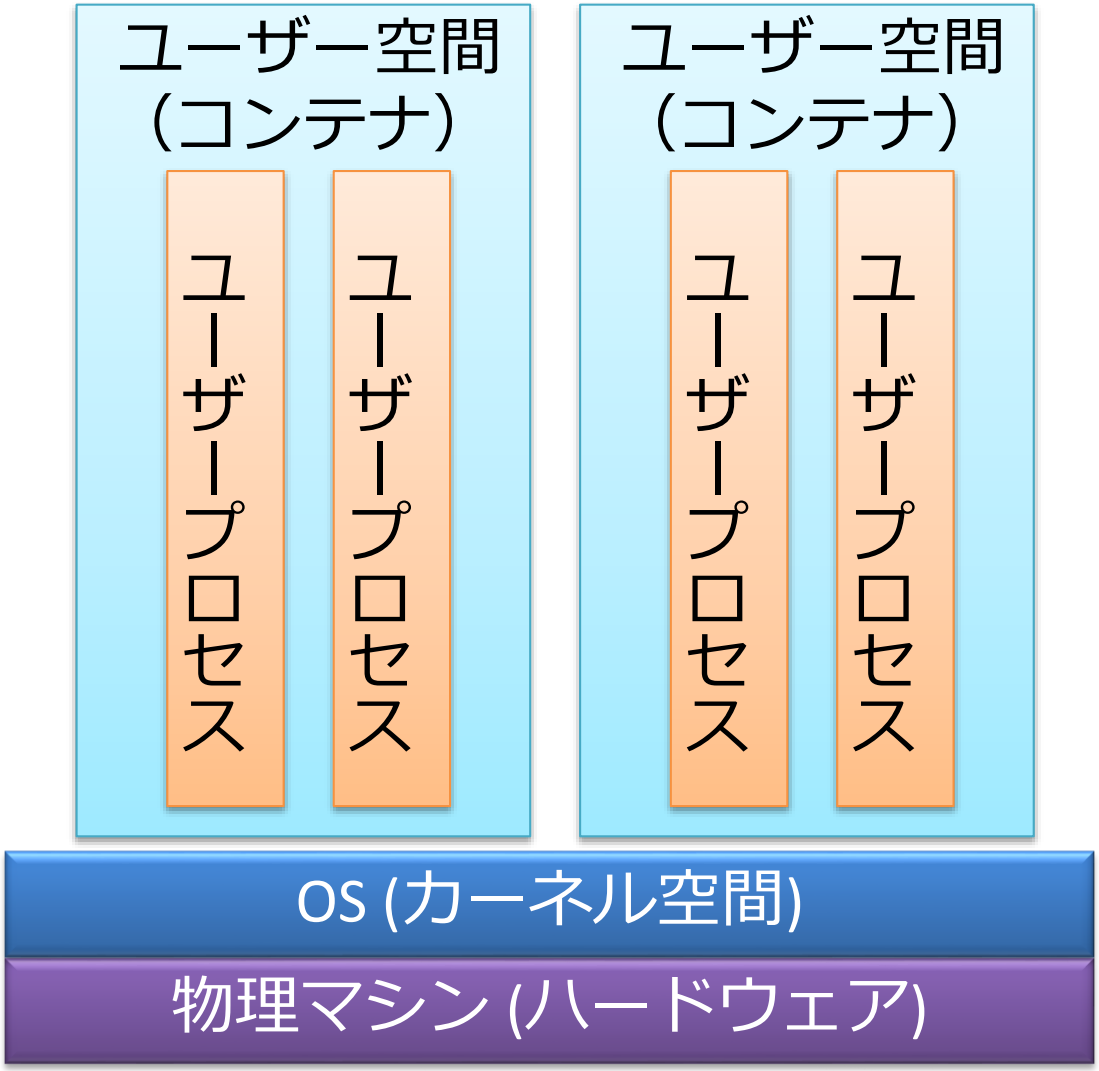
通常の非仮想化環境



ハイパーバイザー型仮想化環境



コンテナ型仮想化環境



コンテナ型仮想化
の
メリット

起動が早い

仮想マシンの起動を要するハイパーバイザー型仮想化とは異なり、ホストOS側の処理としては通常の非仮想化環境下におけるプロセスの起動とほとんど変わらない。

リソース消費が少ない

仮想マシンを介さないため、メモリ・CPU・HDDなどのリソースを有効に活用できる。
また、これにより一つの物理マシン上でより多くのプロセスを実行することが可能となる。

ポータビリティが高い

HTTPサーバのみのコンテナなど機能に応じた小さなコンテナを組み合わせて運用することで、物理マシン間におけるコンテナの再配置や、スケールアウトなどへの対応が容易となる。

面倒な説明はともかく
体験してみよう

Hands On

Today’s menu

コンテナを起動してみる
コンテナの中をのぞいてみる
コンテナをカスタマイズしてみる
コンテナ間で連携してみる

始める前に

今日のハンズオン勉強会ではAWS上に事前に用意した実習環境を用います。
後日おさらいをする場合には各自でLinux環境を用意して下さい。
DockerはVM上のLinux環境でも動作します。

https://docs.docker.com/engine/installation/

コンテナを起動する

【書式】
docker run [オプション...] イメージ名 : タグ [コマンド] [引数...]

コンテナー覧を表示する

【書式】
docker ps [オプション...]

コンテナの起動

Let’s Try!

console:1

```
$ sudo docker run --name=fpm-okamura php:fpm

[17-Feb-2016 12:23:20] NOTICE: fpm is running, pid 1
[17-Feb-2016 12:23:20] NOTICE: ready to handle connections
```

--name： コンテナに名前をつける
（今回の実習では他の人が実行したコンテナと区別するために明示的に名前をつけます）

コンテナがフォアグラウンドプロセスとして実行される

コンテナの表示

Let’s Try!

docker runを行ったのとは別の端末画面から実行すること！

console:2

```
$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS
PORTS               NAMES
94fc6fe8f0be        php:fpm             "php-fpm"           3 seconds ago       Up 2
seconds             9000/tcp            fpm-okamura
```

実行中のコンテナ情報が表示される

コンテナ内でコマンドを実行する

【書式】
docker exec [オプション...] コンテナID（又はコンテナ名）コマンド [引数...]

コマンドの実行

Let's Try!

console:2

```
$ sudo docker exec -ti fpm-okamura /bin/bash
```

-t： tty（端末デバイス）を割り当てる
-i： コンテナの標準入力を開く

```
root@94fc6fe8f0be:/var/www/html# ps ax
```

コンテナ内で実行中のプロセスを表示する

```
PID TTY      STAT   TIME COMMAND
 1 ?        Ss     0:00 php-fpm: master process (/usr/local/etc/php-fpm.conf)
 8 ?        S      0:00 php-fpm: pool www
 9 ?        S      0:00 php-fpm: pool www
10 ?        Ss     0:00 /bin/bash
15 ?        R+     0:00 ps ax
root@94fc6fe8f0be:/var/www/html# ls -a
. ..
root@94fc6fe8f0be:/var/www/html# exit
exit
$
```

コンテンツは存在しない

コマンドを終了する

コンテナを終了する

【書式】
docker stop [オプション...] コンテナID（又はコンテナ名）...

コンテナの終了

Let’s Try!

console:2

```
$ sudo docker ps fpm-okamura
$ sudo docker ps
CONTAINER ID    IMAGE     COMMAND    CREATED         STATUS          ....

$ sudo docker ps –a

CONTAINER ID        IMAGE           COMMAND          CREATED          STATUS
PORTS               NAMES
94fc6fe8f0be        php:fpm         "php-fpm"        42 minutes ago   Exited (0)
About a minute ago                  fpm-okamura
```

終了したコンテナは表示されない

-a： 全てのコンテナを表示する

終了したコンテナも表示される

イメージを作成する

【書式】
docker build [オプション...] パス名（又はURL）...

Dockerfileの記述内容に応じて、コンテナを起動する基となるイメージを作成する

イメージ一覧を表示する

【書式】
docker images [オプション...] [イメージ名[:タグ名]]

Dockerfileの記述内容に応じて、コンテナを起動する基となるイメージを作成する

Let’s Try!

console:1

```
$ mkdir ~/okamura
$ cd ~/okamura
$ git https://github.com/kzokm/aitc-cloud-20160217 .
$ ls -R .
.:
Dockerfile  www

./www:
index.php  static.html
```

各自の作業用ディレクトリを作成する

事前に用意された実習用ファイルを取得する

イメージの作成

Let's Try!

console:1

```
$ cat ./Dockerfile
FROM php:fpm
COPY www/* /var/www/html/
$ sudo docker build -t okamura/php:fpm .
Sending build context to Docker daemon 10.75 kB
Step 1 : FROM php:fpm
 ---> a630b021ad2a
Step 2 : COPY www/* /var/www/html/
 ---> e73d59c9b3f7
Removing intermediate container 809dd9b91a13
Successfully built e73d59c9b3f7
$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             VIRTUAL SIZE
okamura/php         fpm                 e73d59c9b3f7        19 seconds ago      495.8 MB
php                 fpm                 a630b021ad2a        11 days ago         495.8 MB
```

wwwディレクトリ以下のコンテンツをコンテナ内にコピーする

-t： イメージ名をつける

Dockerfileの内容が実行される

作成したイメージが表示される

イメージの確認

Let’s Try!

console:1

```
$ sudo docker run --name=fpm-okamura
okamura/php:fpm

Error response from daemon: Conflict. The name "fpm-okamura" is already in use by container 94fc6fe8f0be. You have to remove (or rename) that container to be able to reuse that name.
```

作成したイメージからコンテナを起動する

先に終了済みのコンテナと名前が同一のためコンテナの起動に失敗した！

コンテナを削除する

【書式】
docker rm [オプション...] コンテナID（又はコンテナ名）...

コンテナの削除

Let’s Try!

console:1

```
$ sudo docker rm fpm-okamura
fpm-okamura
$ sudo docker ps –a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS
PORTS               NAMES
```

終了済みのコンテナを削除する

削除されたことを確認する

イメージの確認（再）

Let’s Retry!

console:1

```
$ sudo docker run --name=fpm-okamura
okamura/php:fpm
[17-Feb-2016 14:43:45] NOTICE: fpm is running, pid 1
[17-Feb-2016 14:43:45] NOTICE: ready to handle connections
```

今度は正しく起動した

console:2

```
$ sudo docker exec -ti fpm-okamura /bin/bash
root@94fc6fe8f0be:/var/www/html# ls -a
. .. index.php  static.html
root@94fc6fe8f0be:/var/www/html# exit
exit
$
```

コンテンツがコピーされていることを確認する

コンテナを連携する

Let's Try!

console:2

```
$ sudo docker run -d -p 8081:80 ¥
 -v `pwd`/www:/var/www/html:ro ¥
 -v `pwd`/nginx:/etc/nginx/conf.d:ro ¥
 -v `pwd`/log/nginx:/var/log/nginx
 --link fpm-okamura:php-app
 nginx
a2311d93476feb085ee27ca782fa7879ebf79
$
```

-d： コンテナをバックグラウンドで実行する
-p : コンテナのポートをホスト側に公開する
（今回の実習では他の人が実行したコンテナと区別するために別々のホスト側ポートを指定します）
-v ホスト側のディレクトリをコンテナ内にマウントする
--link コンテナをリンクする

Let's Try!

console:2

```
$ sudo docker exec -ti a2311d93476f /bin/bash
root@a2311d93476f:/# ls /etc/nginx/conf.d
default.conf
root@a2311d93476f:/# ls /var/www/html
index.php static.html
root@a2311d93476f:/# printenv
PHP_APP_....
root@a2311d93476f:/# exit
$ curl http://localhost:8081/
<h1>Hello World!!!</h1>
<h3>PHP Version 7.0.3</h3>
<a href="/static.html">Static HTML Page</a>
$ ls log/nginx
access.log error.log
```

ホスト側ディレクトリがマウントされていることを確認する

リンクされたコンテナの情報が設定されていることを確認する

ホスト側ポートからnginxコンテナを経由してfpmコンテナにアクセスできることを確認する

ホスト側ディレクトリにnginxのログが出力されていることを確認する

あれ？？？

fpmコンテナ内に
コンテンツファイルを
コピーする必要
なかったんじゃない？

Let’s Try!

AITC
先端IT活用推進コンソーシアム

http://aitc.jp

https://www.facebook.com/aitc.jp

ハルミン
AITC非公式イメージキャラクター