

雰囲気ですら理解しない Docker 入門

Docker とはなんぞや

仮想的な実行環境

コンテナ型

めっちゃかるい

なにがうれしいの

OSや、ローカル環境に依存しない

ローカルが汚れない

共通化しやすい

なうい

仮想化環境っていっぱいあるねん

DockerとかVagrantとか。。。

ややこしいねん

ホスト型とかコンテナ型とか

別に気にせんでええ

とりあえず

ホスト型はOSごと

コンテナ型はOS上

とでもおもったとき

気になったら調べたらええ

Dockerと触れ合ってみる

Dockerをつかってみよう

```
$ docker run centos
```

```
$ docker ps
```

```
$ docker ps -a
```

```
$ docker rm [CONTAINER ID or NAMES]
```

```
$ docker run centos echo "Hello World"
```

```
$ docker ps -a
```

```
$ docker rm [CONTAINER ID or NAMES]
```

いっしゅんでこんてなしぬやん

```
$ docker run -it centos
```

```
[root@123456789 /]# exit
```

```
$ docker ps
```

```
$ docker stop [CONTAINER ID or NAMES]
```

```
$ docker rm [CONTAINER ID or NAMES]
```


いますぐこんテナはいりたいわけとちゃうねん

```
$ docker -itd centos
```

```
$ docker exec -it [CONTAINER ID or NAMES] bash
```

```
[root@123456789 /]# exit
```

```
$ docker ps
```

```
$ docker stop [CONTAINER ID or NAMES]
```

```
$ docker rm [CONTAINER ID or NAMES]
```

こんてなになまえつけたくない？つけたくないの？

```
$ docker run -itd --name mycent centos
```

```
$ docker exec -it mycent bash
```

```
# exit
```

```
$ docker ps
```

```
$ docker stop mycent
```

```
$ docker rm mycent
```

え、どうやってぶろぐらむかくんすか

```
$ docker run -itd --name py -v $(PWD):/app python
```

```
$ docker exec -it py bash
```

```
# cd /app
```

```
# ls
```

```
$ vi test.txt
```

```
# ls
```

じっこうしてもみれんくない？

```
$ docker run -d --name test -p 8888:80 nginx
```

localhost:8888 にアクセス

おぶしょんず

- -it ターミナルをつなげるで
- -d バックグラウンドで起動するで
- --name コンテナに名前つけんで
- --rm stopしたら自動でrmするで
- -v ファイル共有すんで
- -p ポート繋ぐで
- -w ここではじめるで

```
$ docker run -itd --name nginx --rm -v $(PWD):/usr/share/nginx/html -p 8888:80  
-w=/usr/share/nginx/html nginx
```

こまんど

- `docker run [OPTION] [CONTAINER ID or NAMES]`
- `docker exec [OPTION] [CONTAINER ID or NAMES]`
- `docker ps [OPTION]`
- `docker stop [CONTAINER ID or NAMES]`
- `docker rm [CONTAINER ID or NAMES]`

こんでなとこんでなつなげたいねん

docker-compose ってなんだ

なんかいっぺんにふくすーのこんてなたてれるやつ

こまんど

- `docker-compose run [CONTAINER NAMES]`
 - (起動前の)コンテナ内でコマンド実行
- `docker-compose exec [CONTAINER NAMES]`
 - (起動後の)コンテナ内でコマンド実行
- `docker-compose up`
 - コンテナを起動
- `docker-compose down`
 - コンテナを終了
- `docker-compose ps`
 - `docker-compose`内のコンテナの状態を確認
- `docker-compose logs [CONTAINER NAMES]`
 - コンテナのログを確認