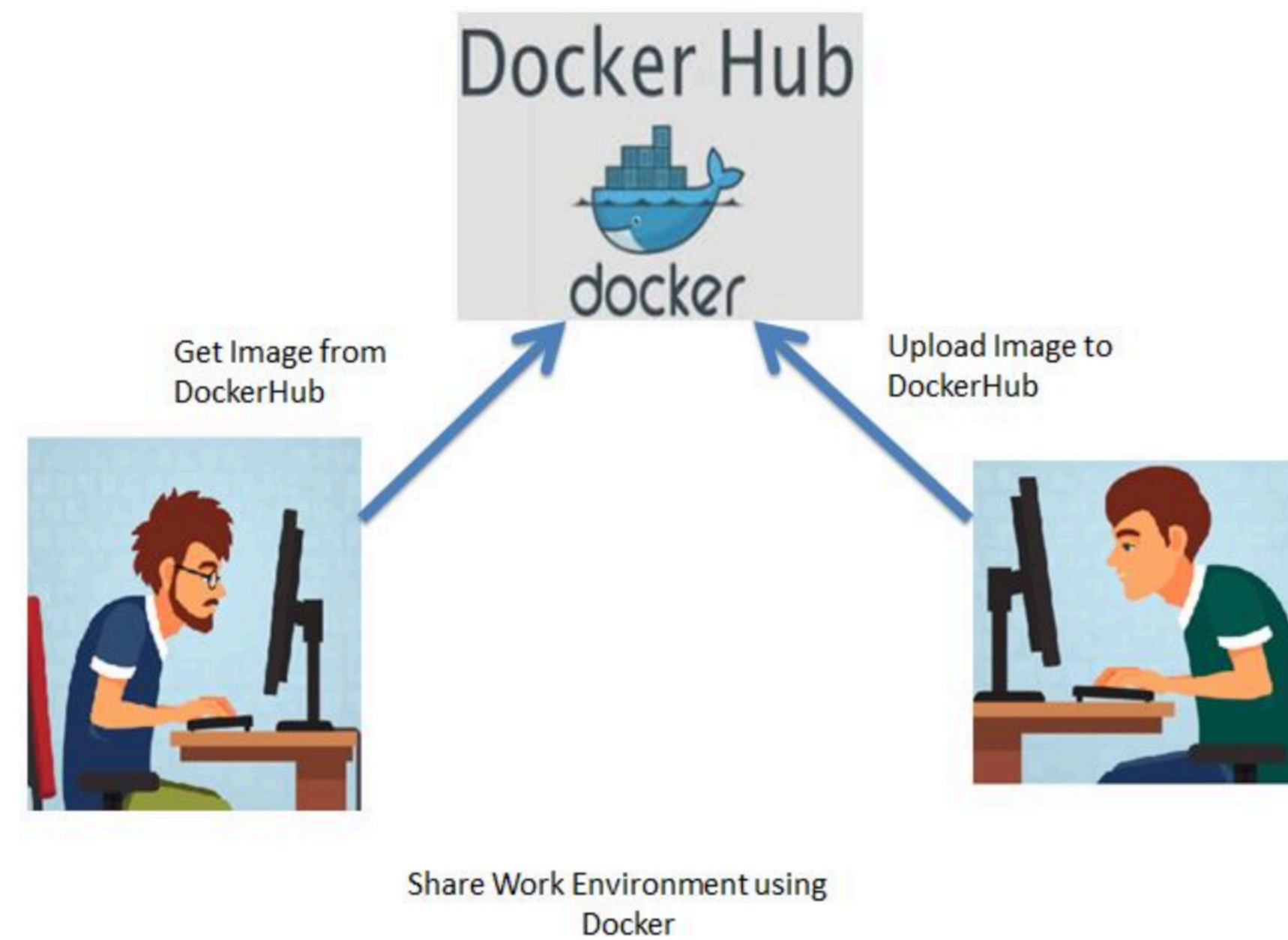


DOCKER

今回のゴール





dockerって何？

dockerとは、docker社が開発しているコンテナ型仮想化のプラットフォームです。



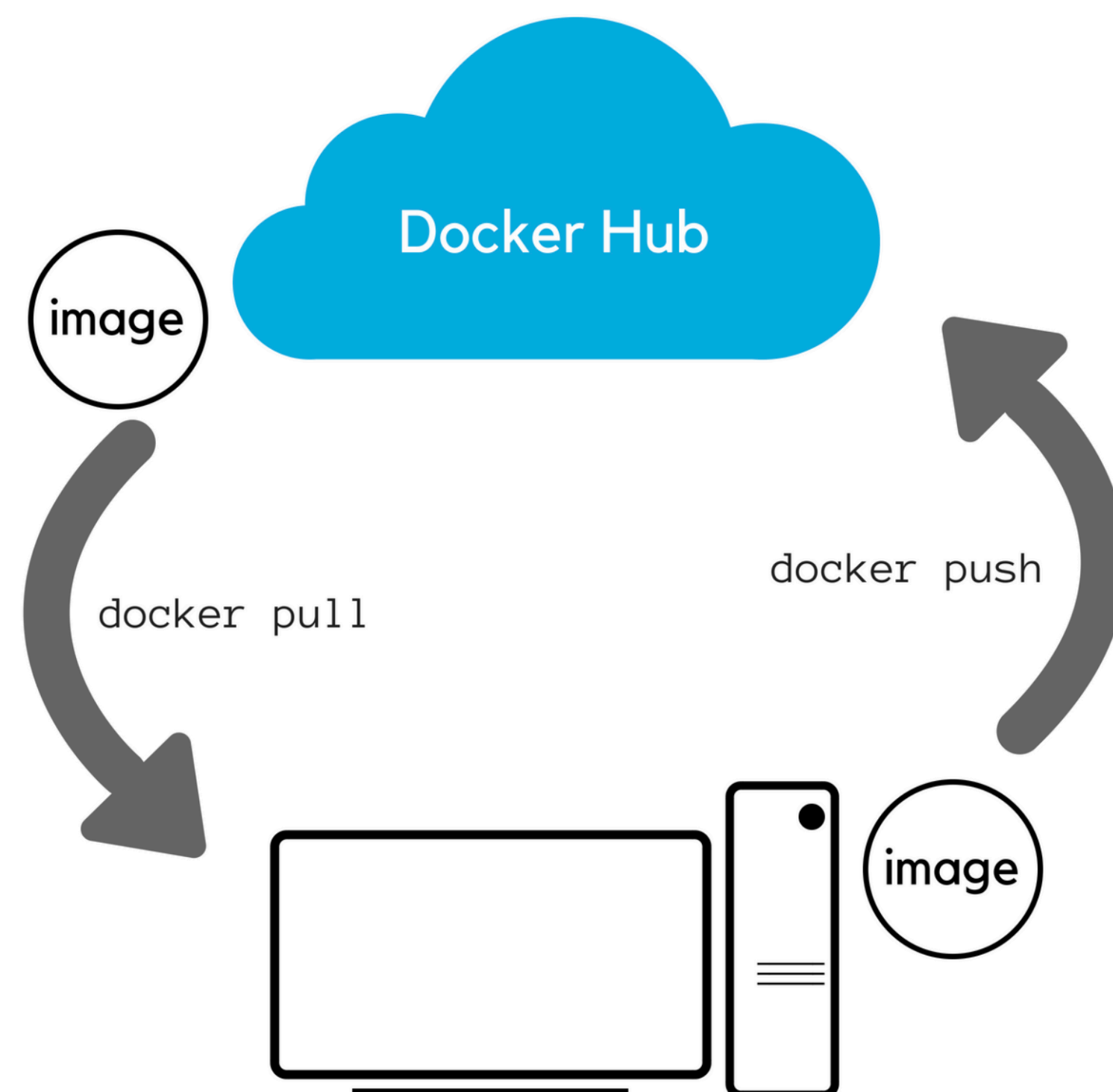
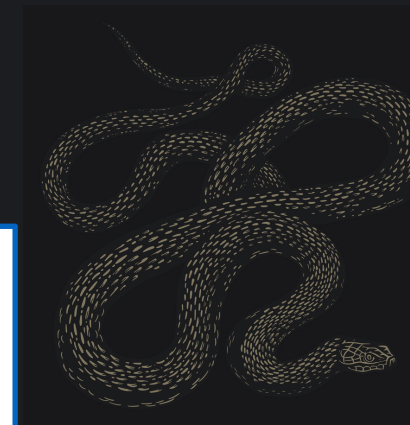
コンテナって何？

ホストOSの上で動作しているDocker Engineが作成する**隔離されたプロセスの状態**のこと。コンテナごとにファイルシステムやネットワークが独立して動き、その中でアプリケーションを動作させている。従来のホスト型仮想化とは異なり、ゲストOSを起動させない。

<https://www.slideshare.net/zembutsu/what-isdockerdoing>

p28-29

今回のゴール

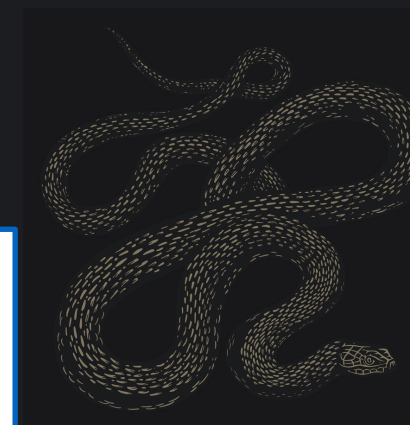




dockerイメージって何？

dockerイメージとは、コンテナ実行に必要なファイルやメタ情報をまとめた一式のことです。

イメージ上のファイルやメタ情報は、レイヤと呼ばれる階層構造で構成され読み取り専用である。



ここからはdockerの
操作コマンドを紹介していきます

docker run イメージ名

以下3つのコマンドを順に実行するのと等しい

イメージ取得

```
docker pull
```

コンテナ作成

```
docker create
```

コンテナ起動

```
docker start
```

つまり、`docker run` コマンドは
イメージ取得+コンテナ作成+コンテナ起動
を実行している

例：

```
docker run hello-world
```

※ `hello-world`イメージをrunする

Docker イメージ

ダウンロード済みのDockerイメージ一覧表示

```
docker images
```

イメージ削除

```
docker rmi イメージ名:タグ
```

例：

```
docker rmi hello-world:latest
```

Docker コンテナ 2

停止中のものも含めてコンテナ一覧表示

```
docker ps -a
```

起動中のコンテナ一覧表示

```
docker ps
```

コンテナ起動

```
docker start コンテナID
```

コンテナ停止

```
docker stop コンテナID
```

コンテナ削除

```
docker rm コンテナID
```

Docker コンテナ 2

コンテナログイン (attach)

```
docker attach コンテナID
```

attachからのコンテナログアウト

Ctrl P + Q (コンテナは停止しない)

exit (コンテナ停止)

コンテナログイン (exec)

```
docker exec -it コンテナID /bin/sh
```

execからのコンテナログアウト

Ctrl P + Q (コンテナは停止しない)

exit (コンテナ停止しない)

コンテナとローカル環境をつなぐ

Python3.6のイメージでコンテナを立て、ローカル環境とつなぐ

```
docker run --name ysy -it -v /Users/yoshiko/Desktop/  
docker:/home/docker python:3.7 /bin/bash
```

オプション

--rm

コンテナ終了時に自動的に削除

-v

バインドマウント

-it

i : アタッチされていない状態でも入力を保持できる

t : 擬似ターミナルを割り当てる

DockerHub へイメージを pushする

Docker Hubへログイン

```
docker login
```

```
Authenticating with existing credentials...  
Login Succeeded
```

停止したコンテナからイメージを作る

```
docker commit コンテナID 任意の名前:任意のtag
```

イメージにタグ付け

```
docker tag イメージID dockerUserName/リポジトリ名
```

Docker Hubにpushする

```
docker push dockerUserName/リポジトリ名
```

Docker Hubからイメージをpullする

```
docker pull dockerUserName/リポジトリ名
```