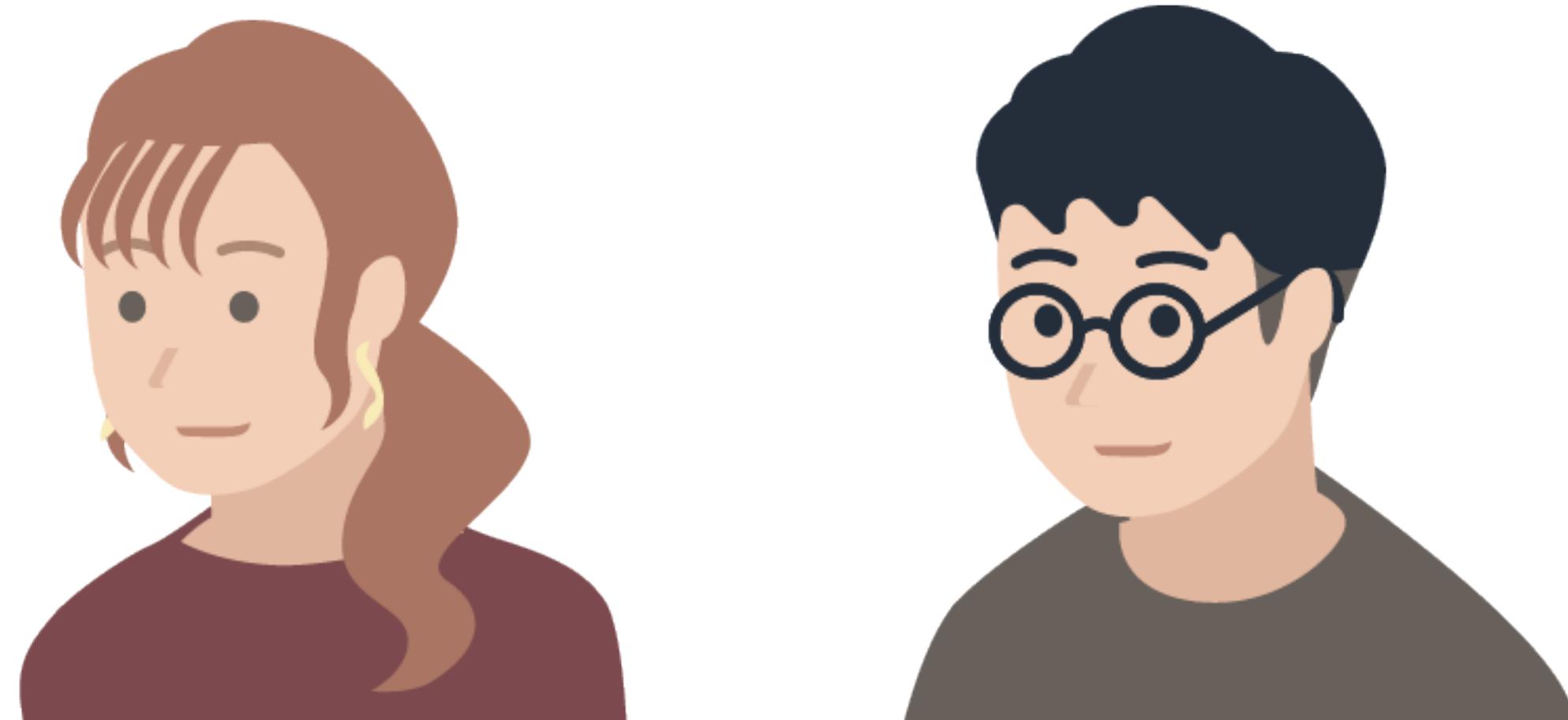




の理解を深めよう

docker? 何それ?



新入社員

目次

- dockerって何？
- コンテナとは？
- イメージとは？
- dockerをわかりやすく解説
- dockerコンテナを作成してみよう！
- docker compose
- おまけ(コンテナvs仮想化)
- まとめ

目次

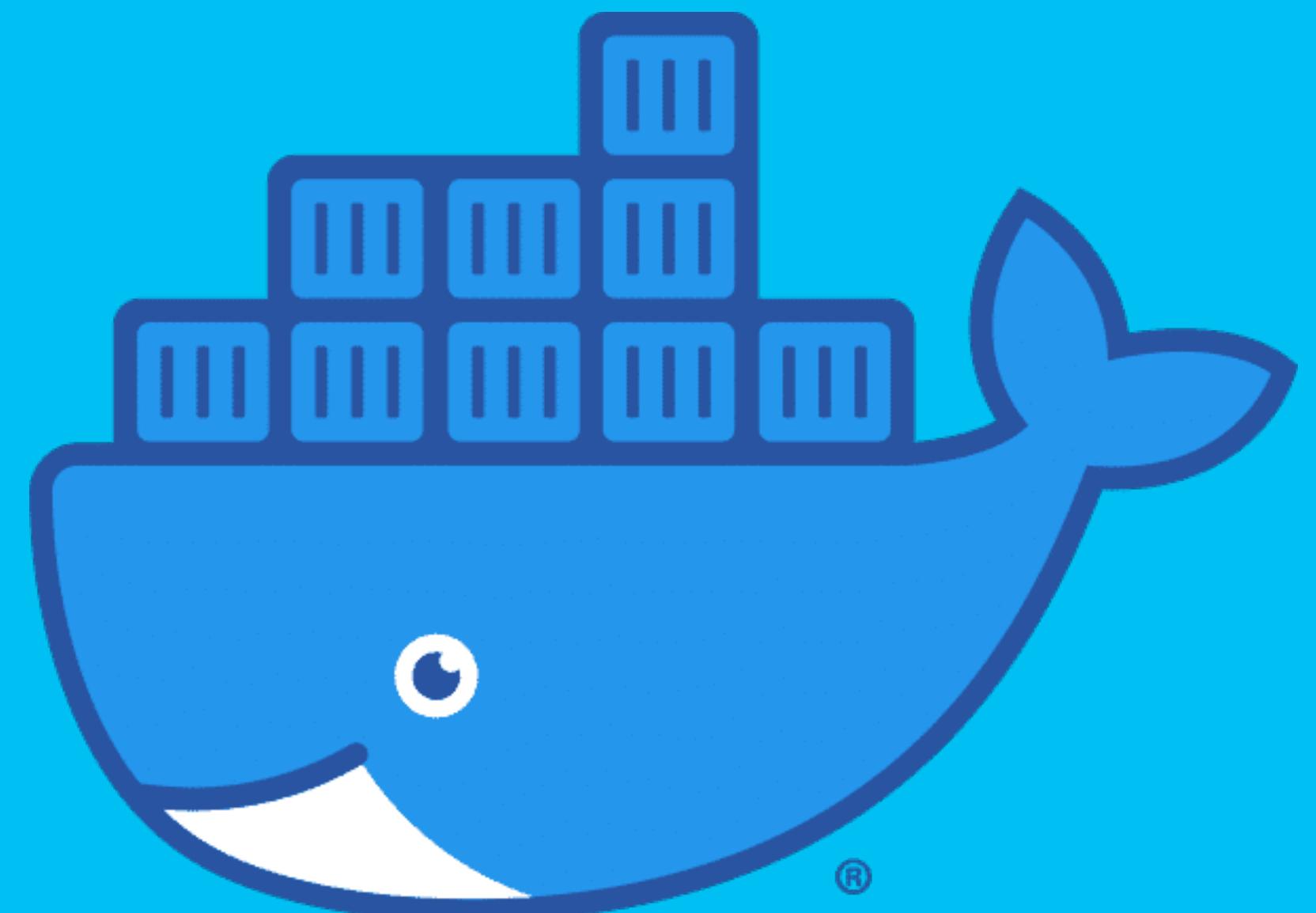
- dockerって何？
- コンテナとは？
- イメージとは？
- dockerをわかりやすく解説
- dockerコンテナを作成してみよう！
- docker compose
- おまけ(コンテナvs仮想化)
- まとめ

dockerって何？？

dockerは、コンテナ仮想化を用いてアプリケーションを開発・配置・実行するためのオープンプラットフォームである。

dockerはコンテナ仮想化を用いたOSレベルの仮想化によりアプリケーションを開発・実行環境から隔離し、アプリケーションの素早い提供を可能にする。

かつその環境 자체をアプリケーションと同じようにコード（イメージ）として管理可能にする。

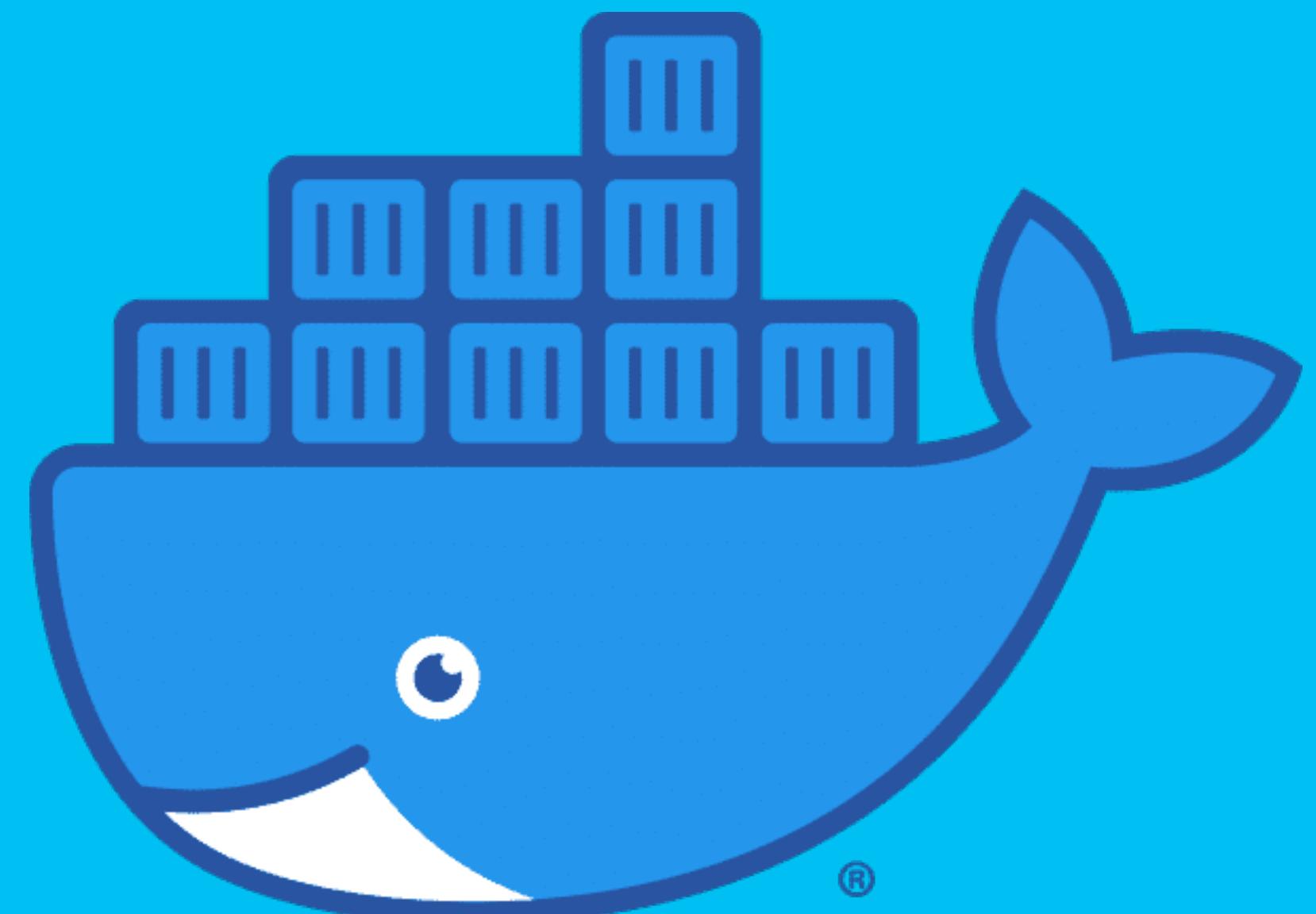
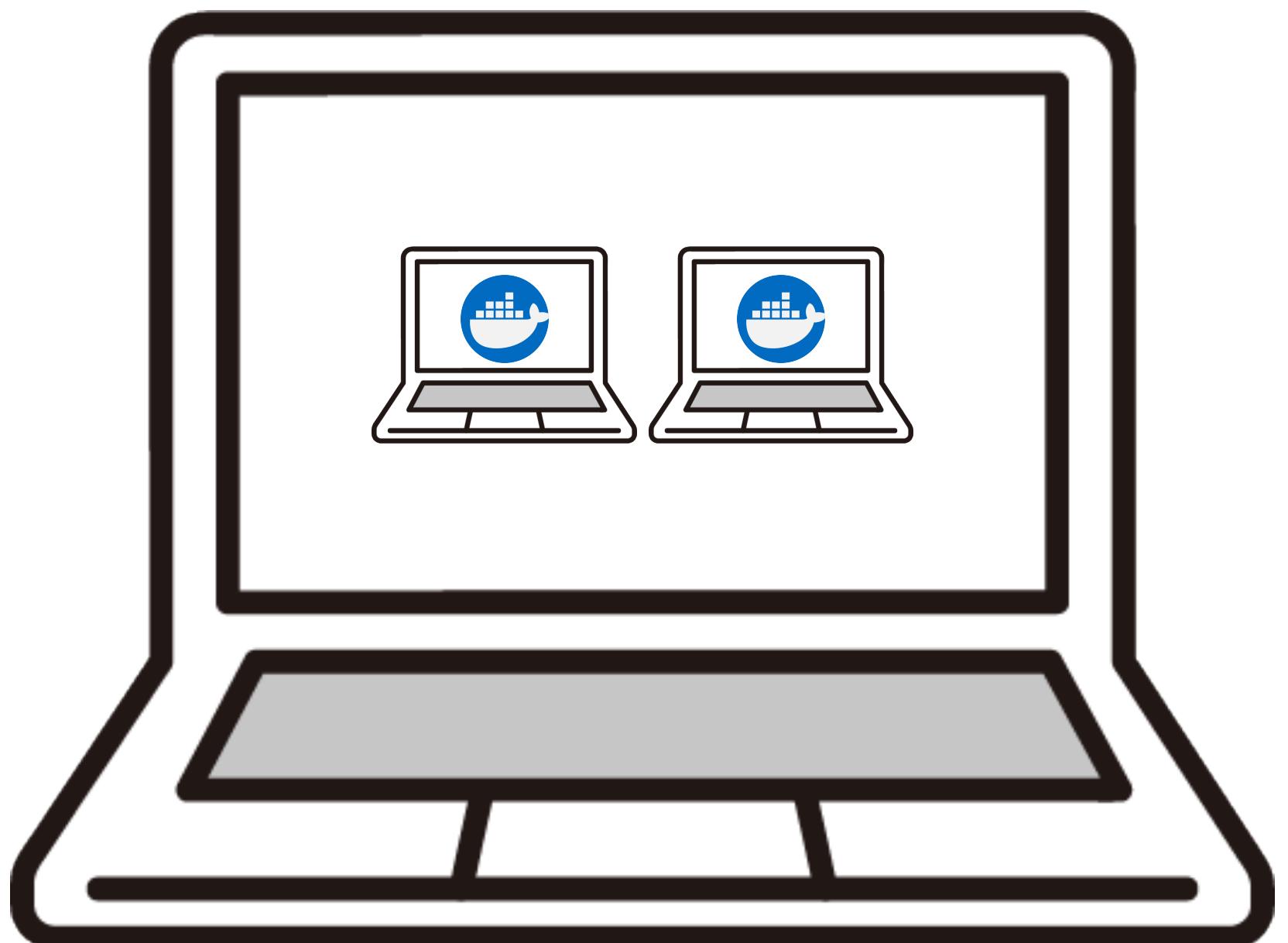


docker®

dockerって何？？

できること

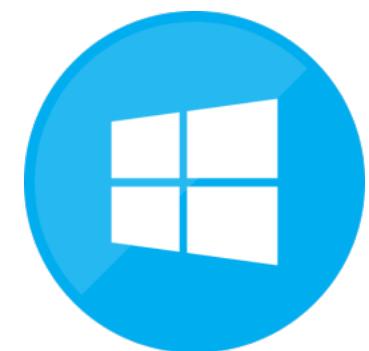
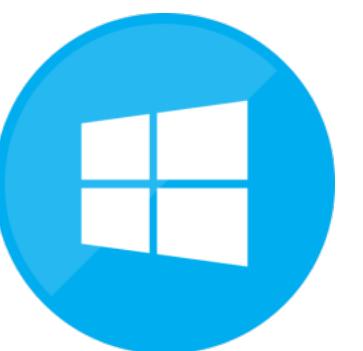
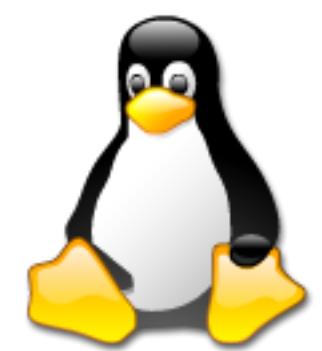
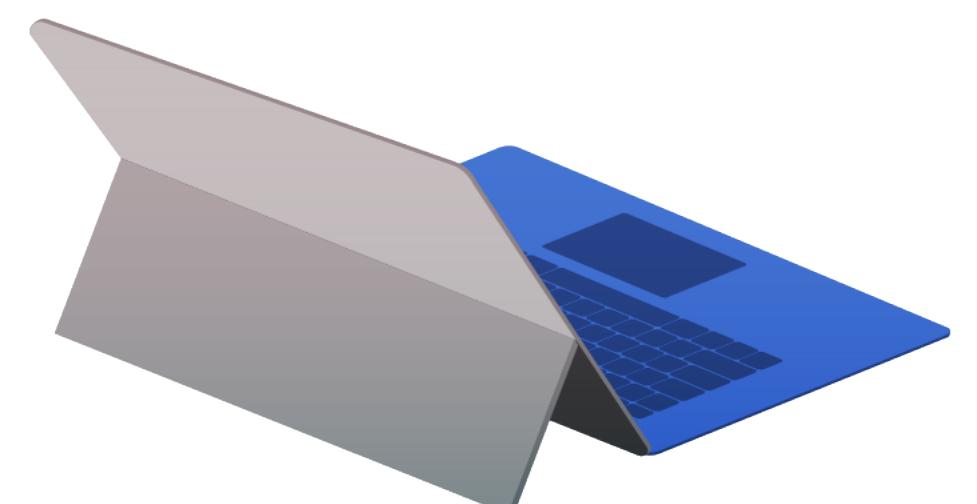
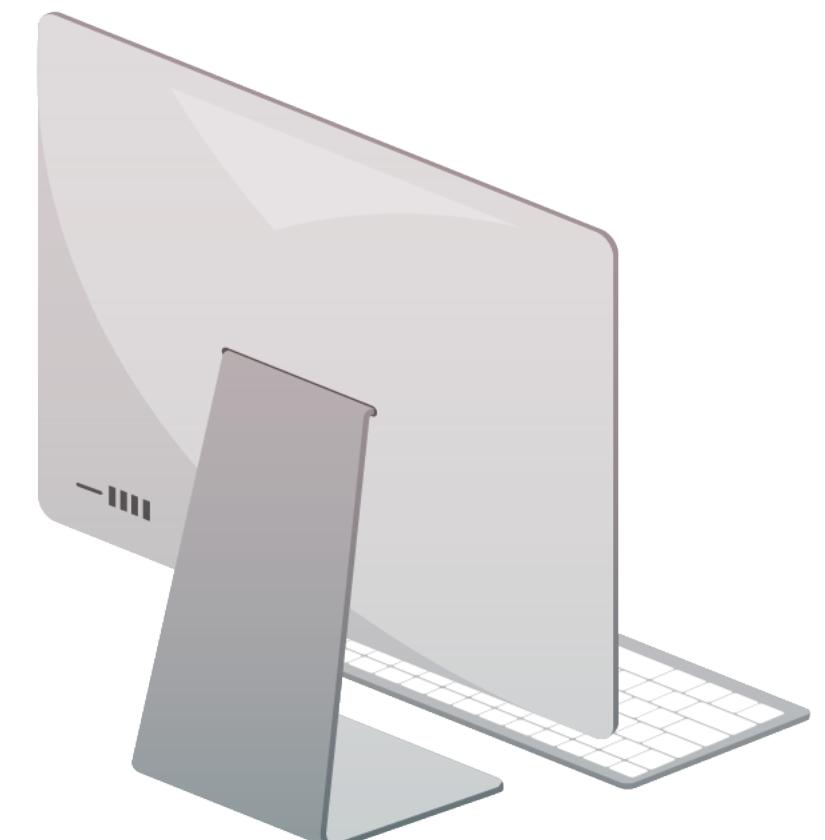
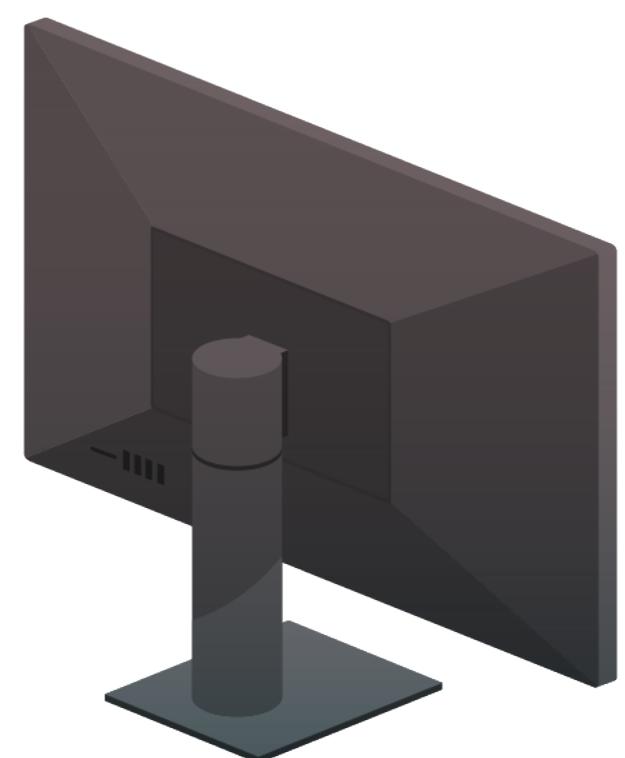
1台のPCの中に複数のコンテナ（仮想環境）を構築することができ、複数の開発環境・検証環境を揃え、対応することが可能です。



docker®

開発しますよ～！





8.0



8.0



7.4



5.7



5.6



5.6



-



5.7



7.2



-



環境が違うと動かない・・・

開発できない



みんな同じ環境や～！

開発できる



目次

- dockerって何？
- **コンテナとは？**
- イメージとは？
- dockerをわかりやすく解説
- dockerコンテナを作成してみよう！
- docker compose
- おまけ(コンテナvs仮想化)
- まとめ

コンテナって何？？



船や列車に荷物を乗せて
行き来するための鉄の箱

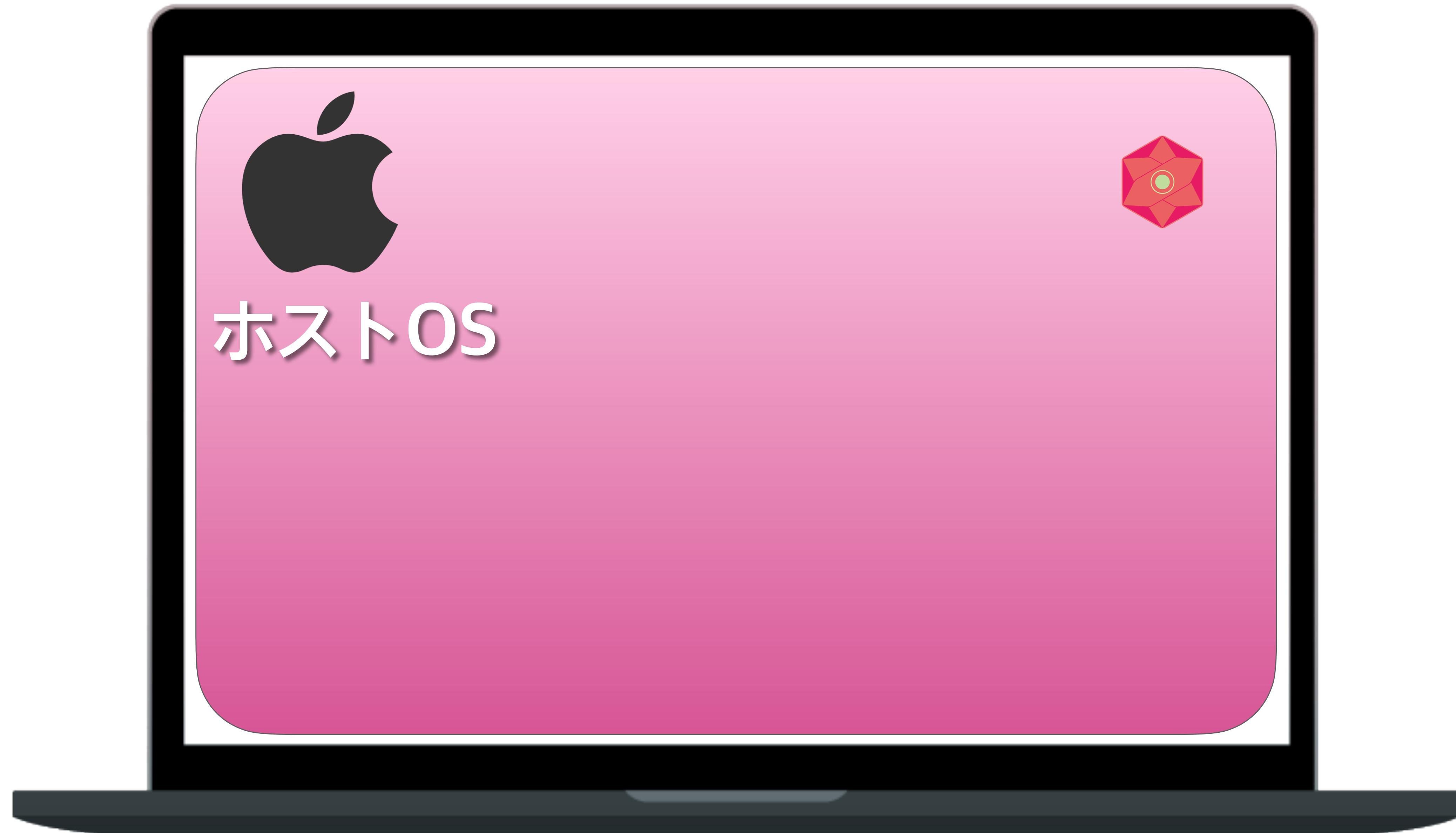


dockerコンテナ

プロセスだけを分離して動かす技術

※プロセス=アプリを動かす処理

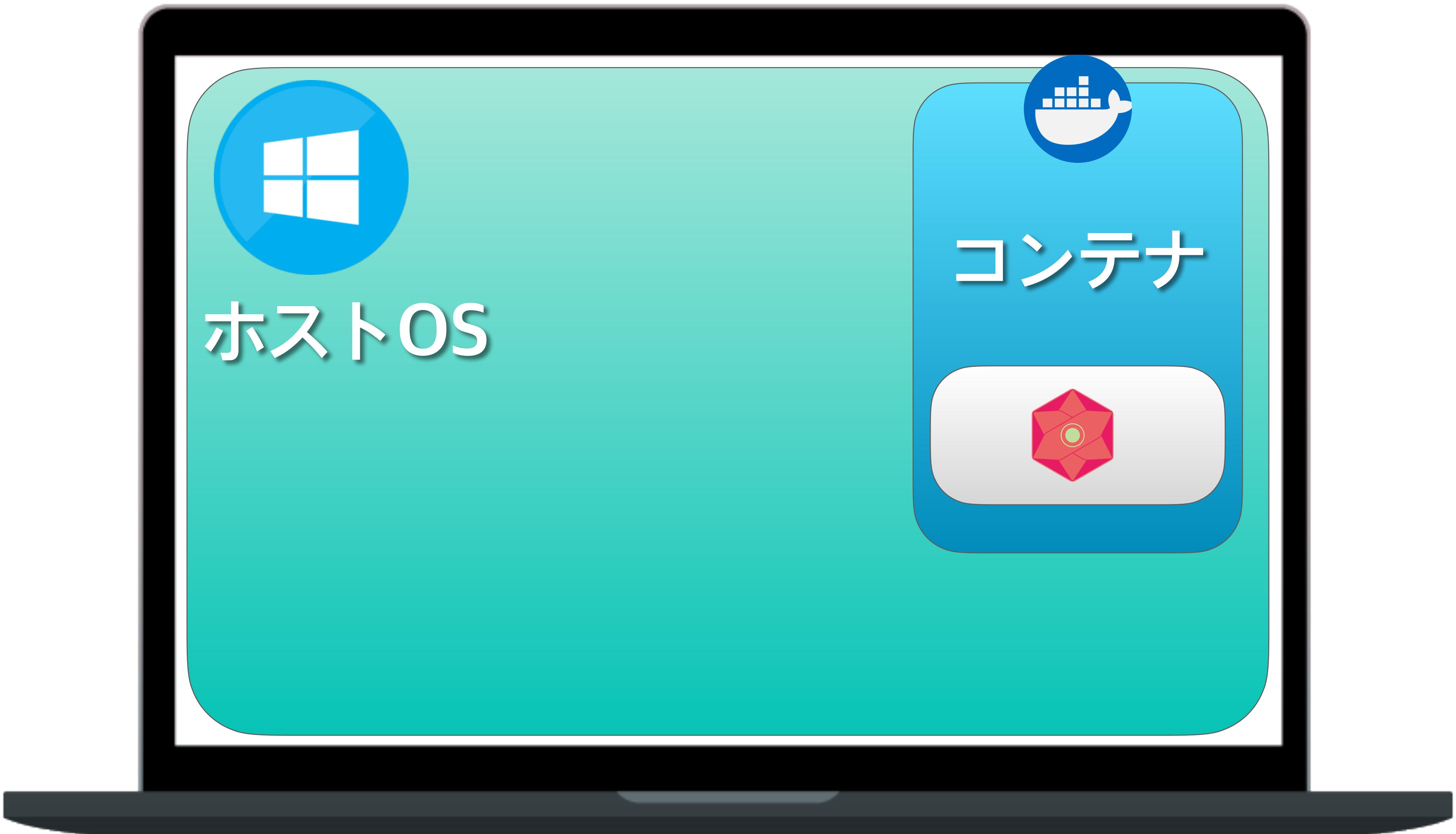




コンテナ



コンテナ



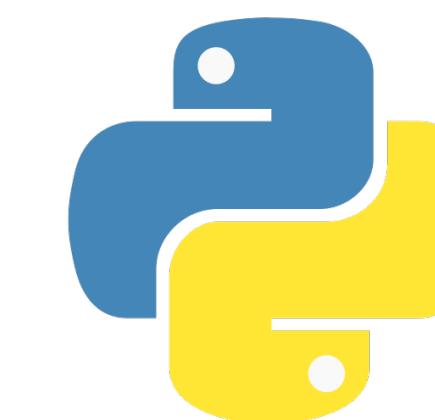
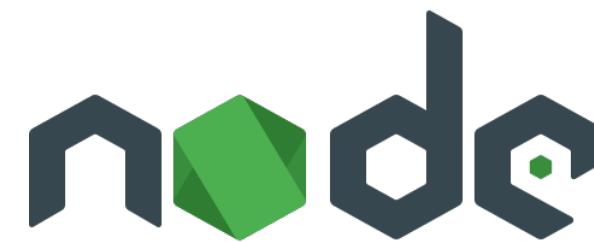
目次

- dockerって何？
- コンテナとは？
- イメージとは？**
- dockerをわかりやすく解説
- dockerコンテナを作成してみよう！
- docker compose
- おまけ(コンテナvs仮想化)
- まとめ

イメージって何？？

dockerイメージ

コンテナを起動させるためのベース
dockerファイルから作成される

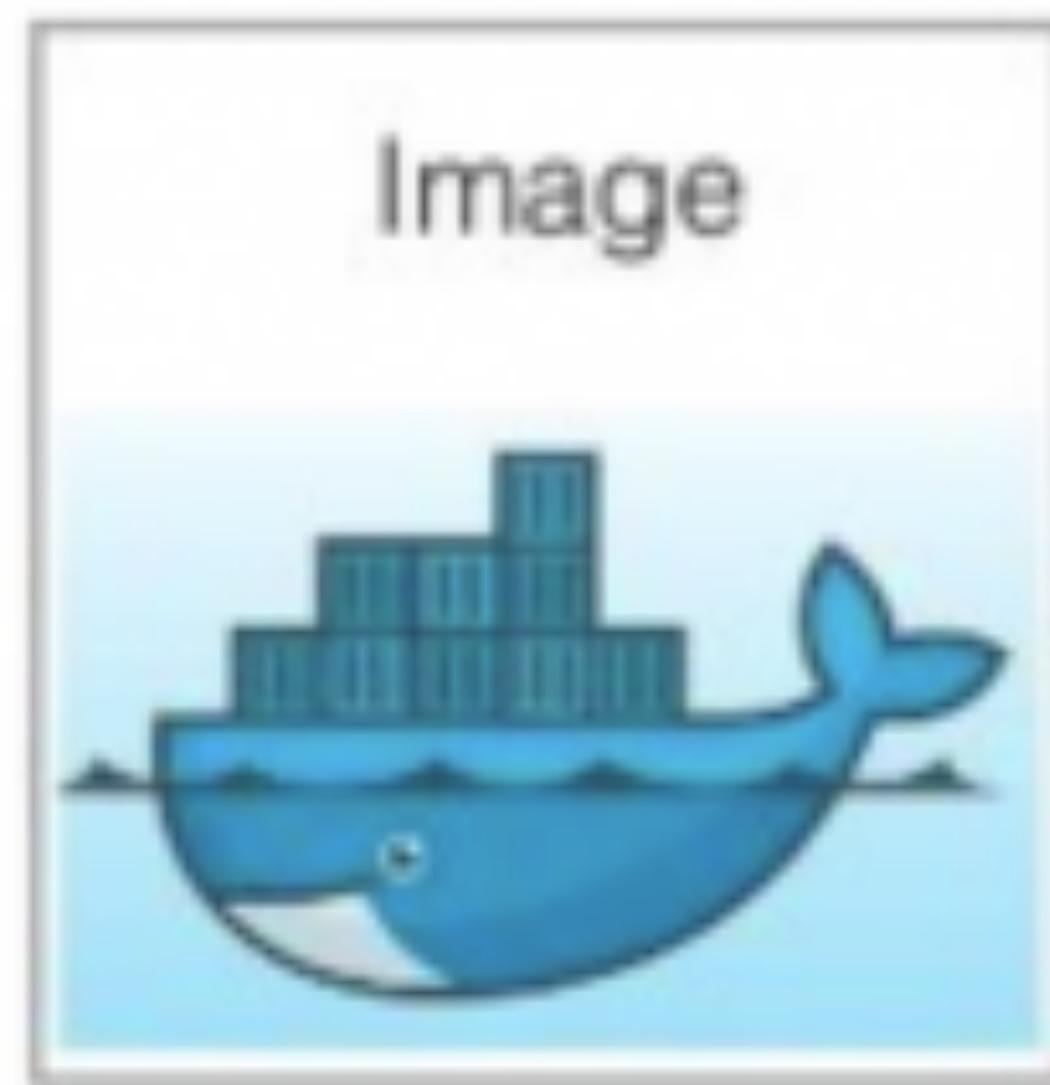


```
FROM alpine:3.10
MAINTAINER "John Doe <john.doe@example.com>"

RUN apk add curl
CMD curl -s https://api.github.com/repos/docker/dockerfile-parser/pulls
```

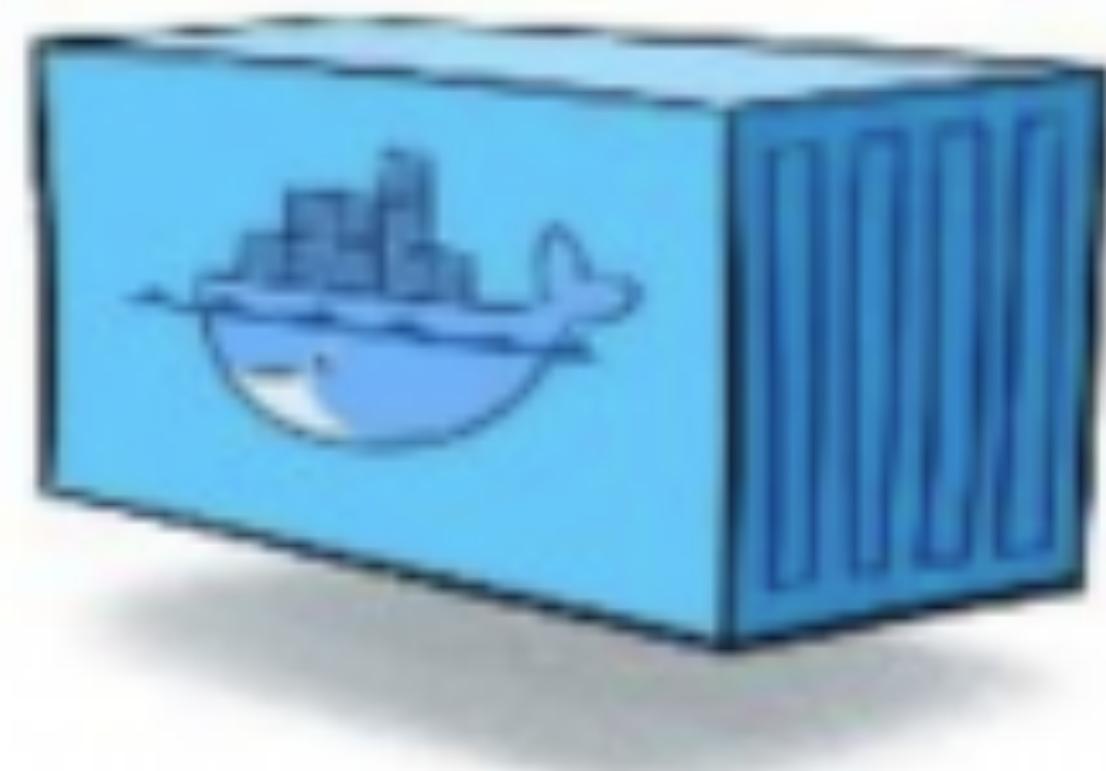
Dockerfile

build



Docker Image

run



Docker Container

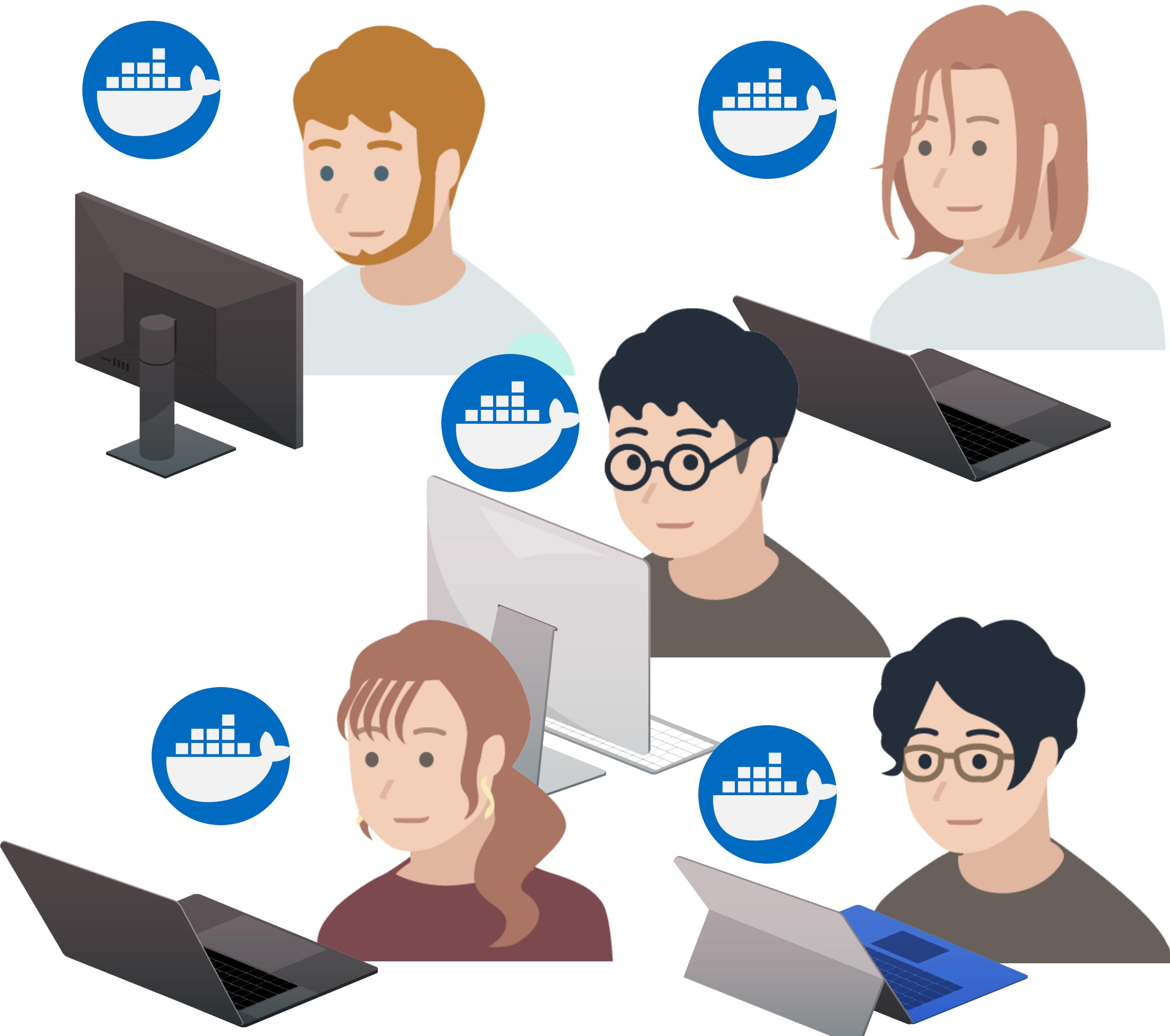
dockerファイルだよ

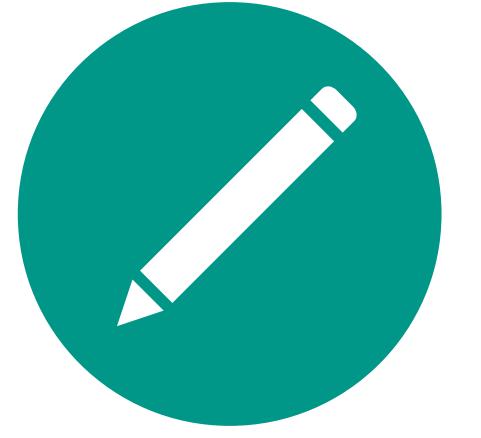


dockerイメージを作成



dockerコンテナを作成



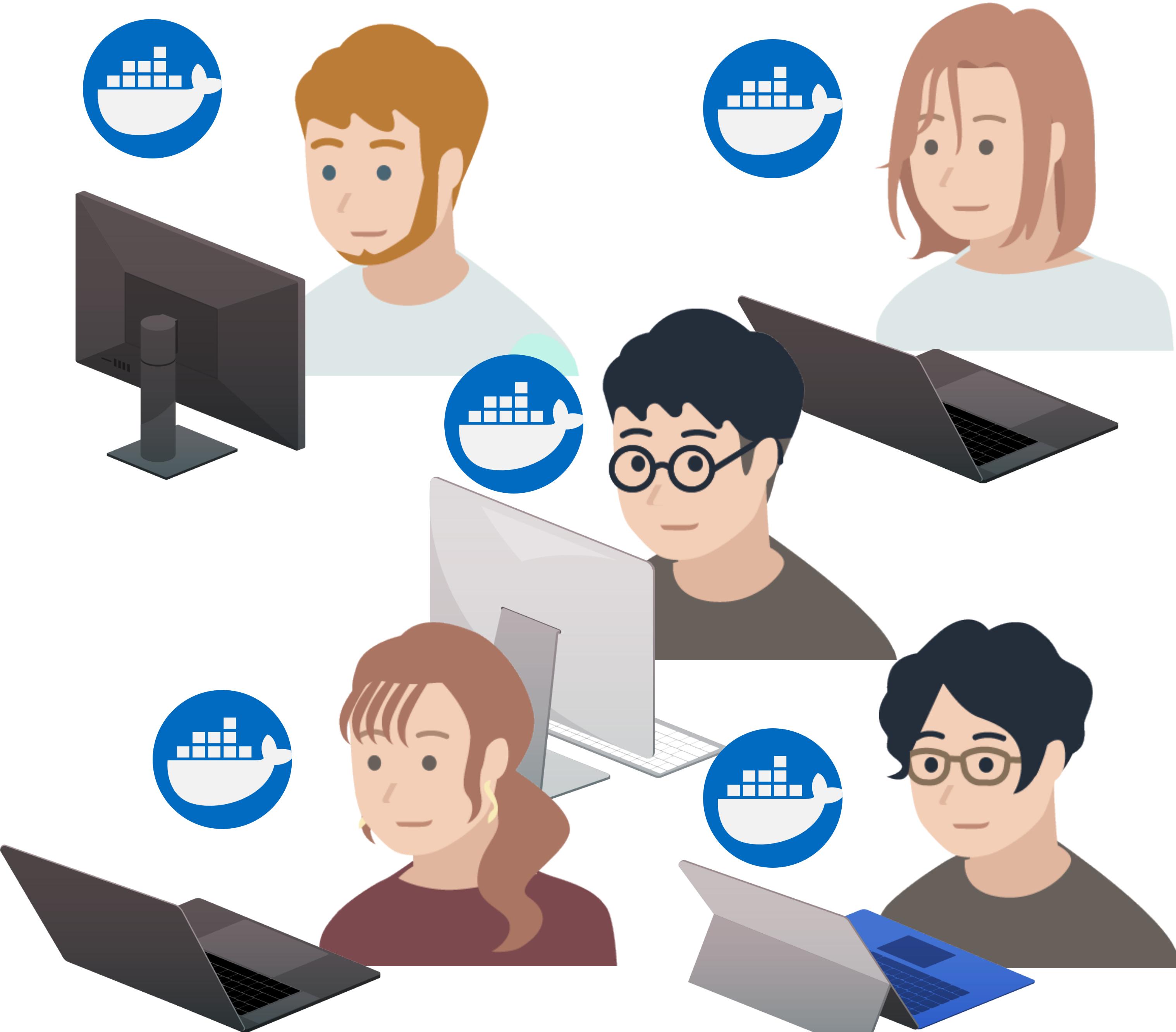


細かい変更にも対応しやすい

変更した
dockerファイルだよ



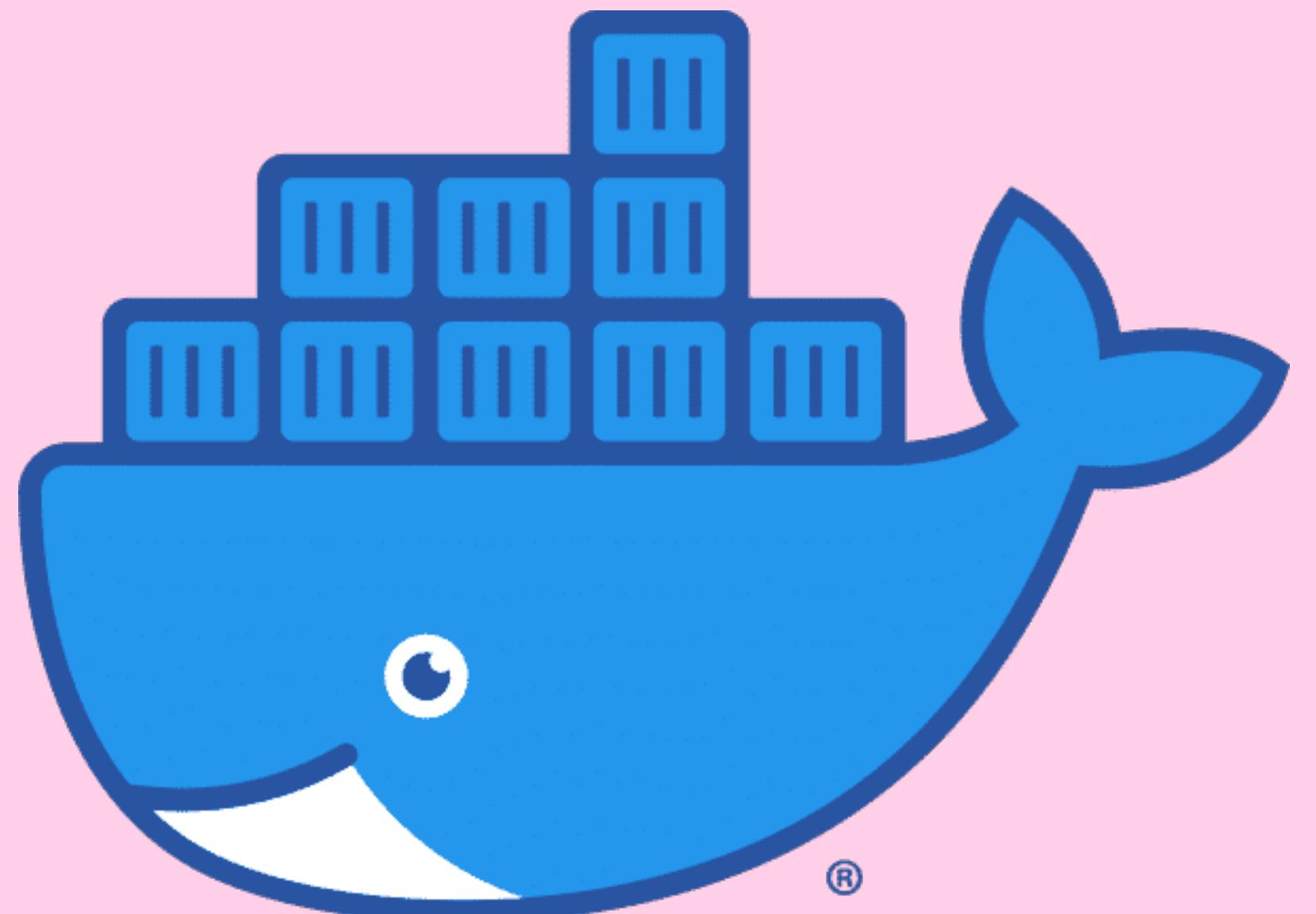
dockerコンテナを
作成しなおしてね



目次

- dockerって何？
- コンテナとは？
- イメージとは？
- **dockerをわかりやすく解説**
- dockerコンテナを作成してみよう！
- docker compose
- おまけ(コンテナvs仮想化)
- まとめ

dockerをわかりやすく解説！



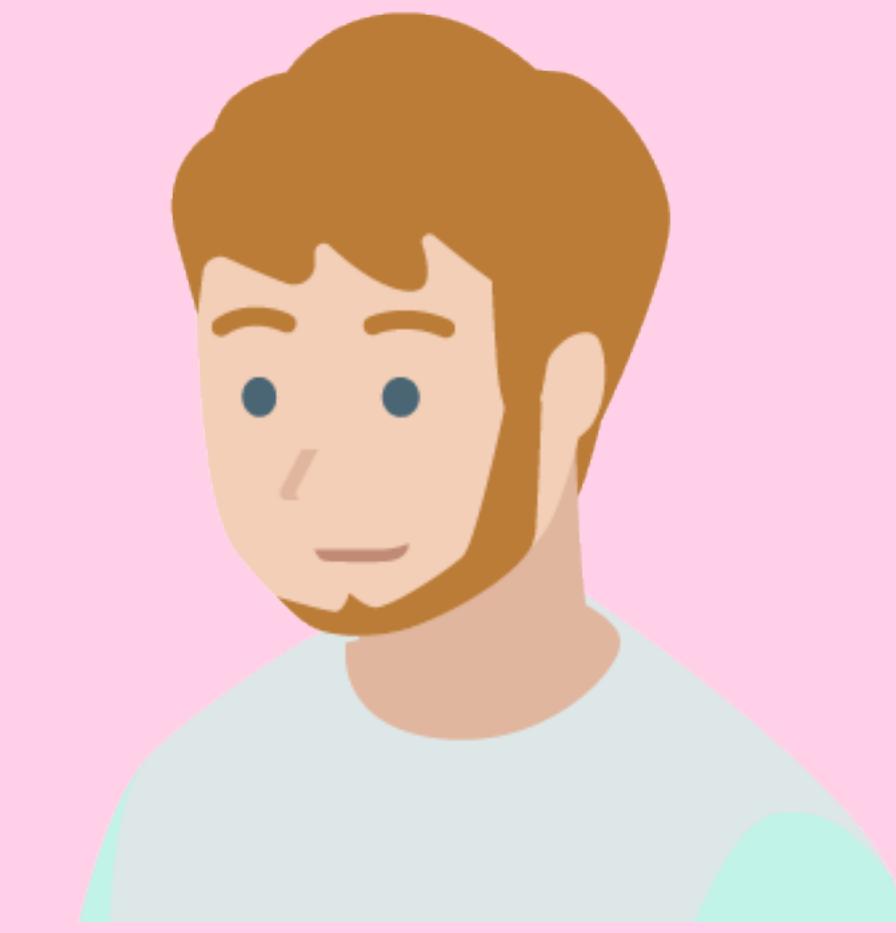
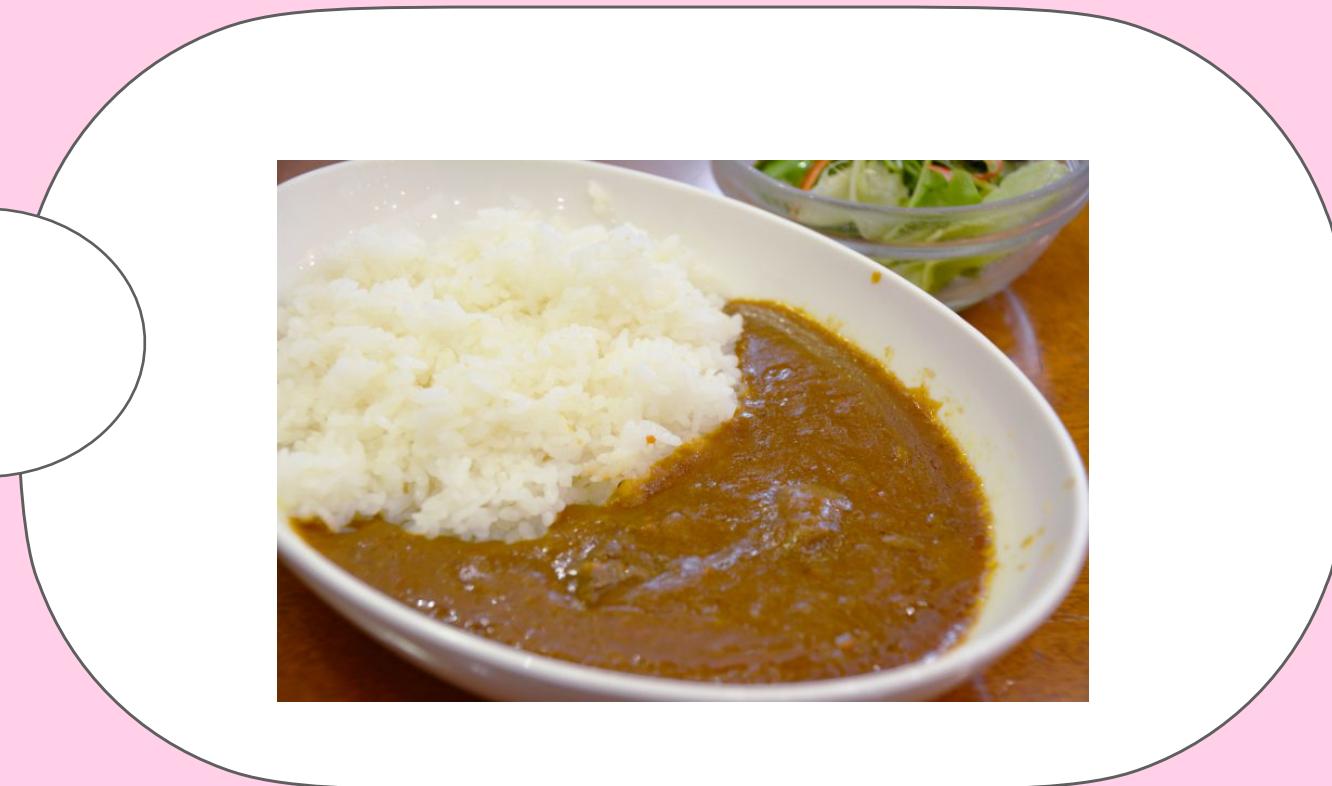
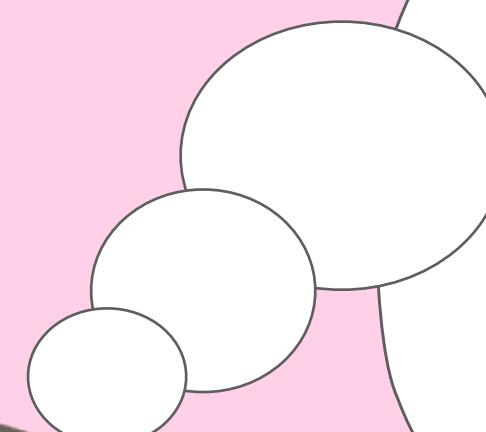
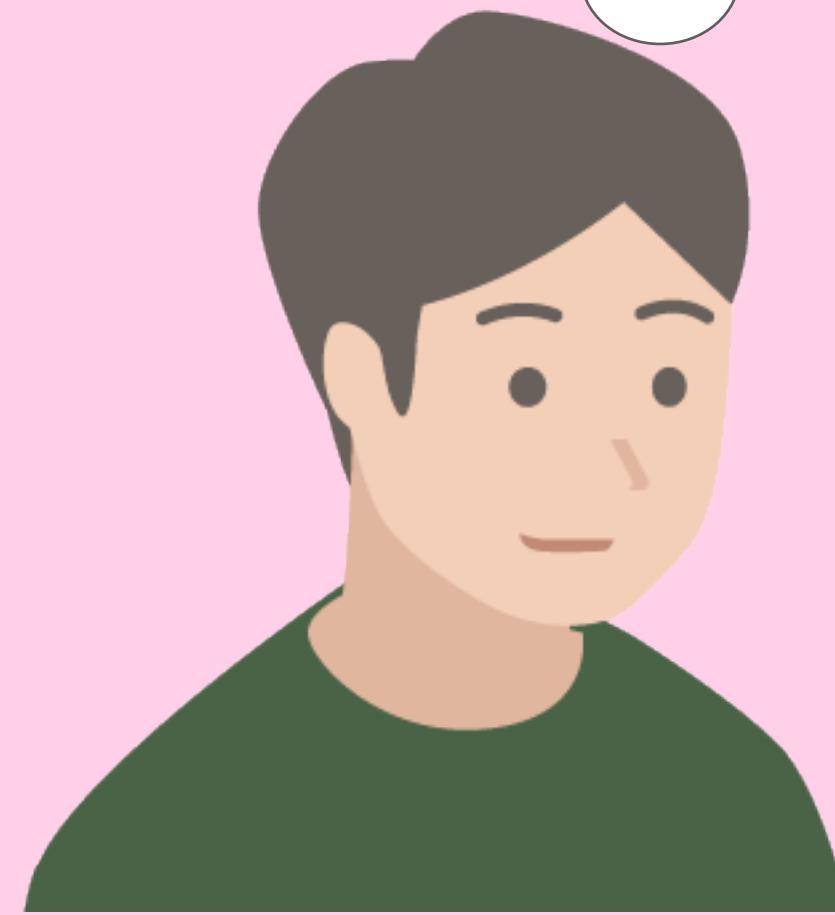
よくわからん



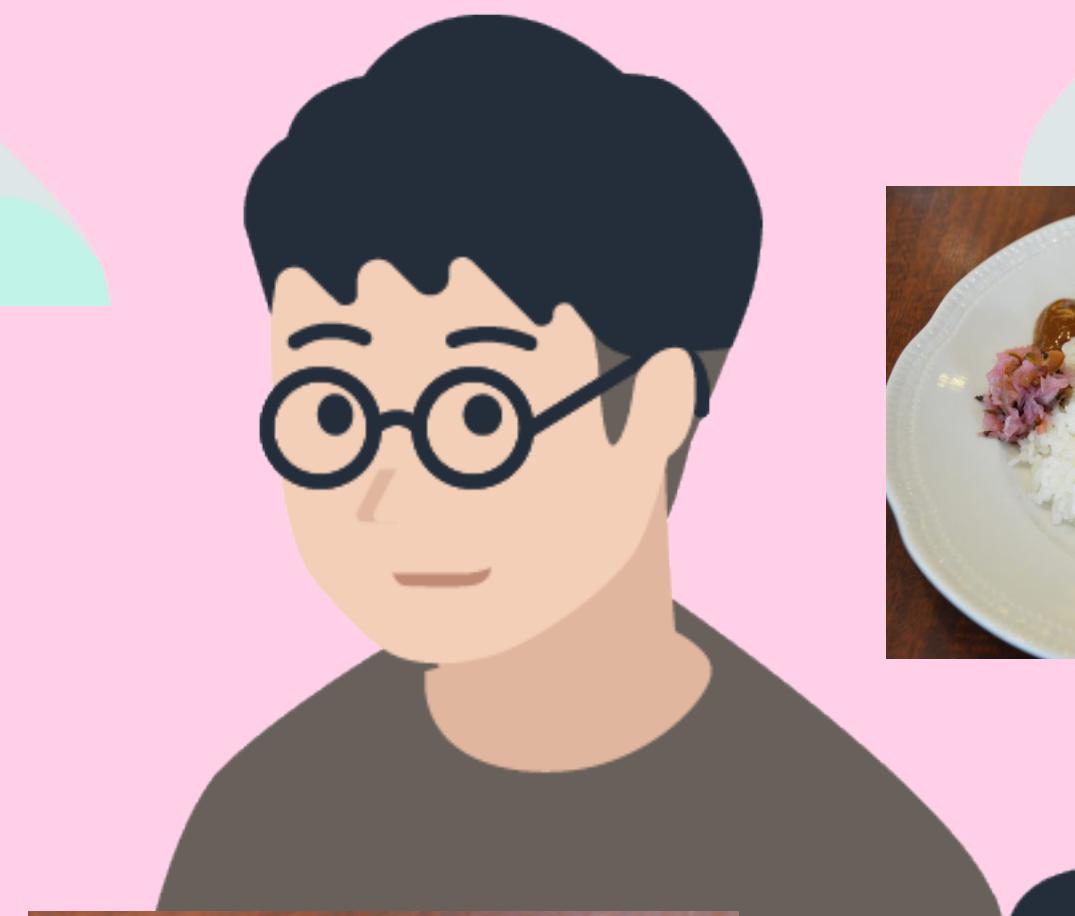
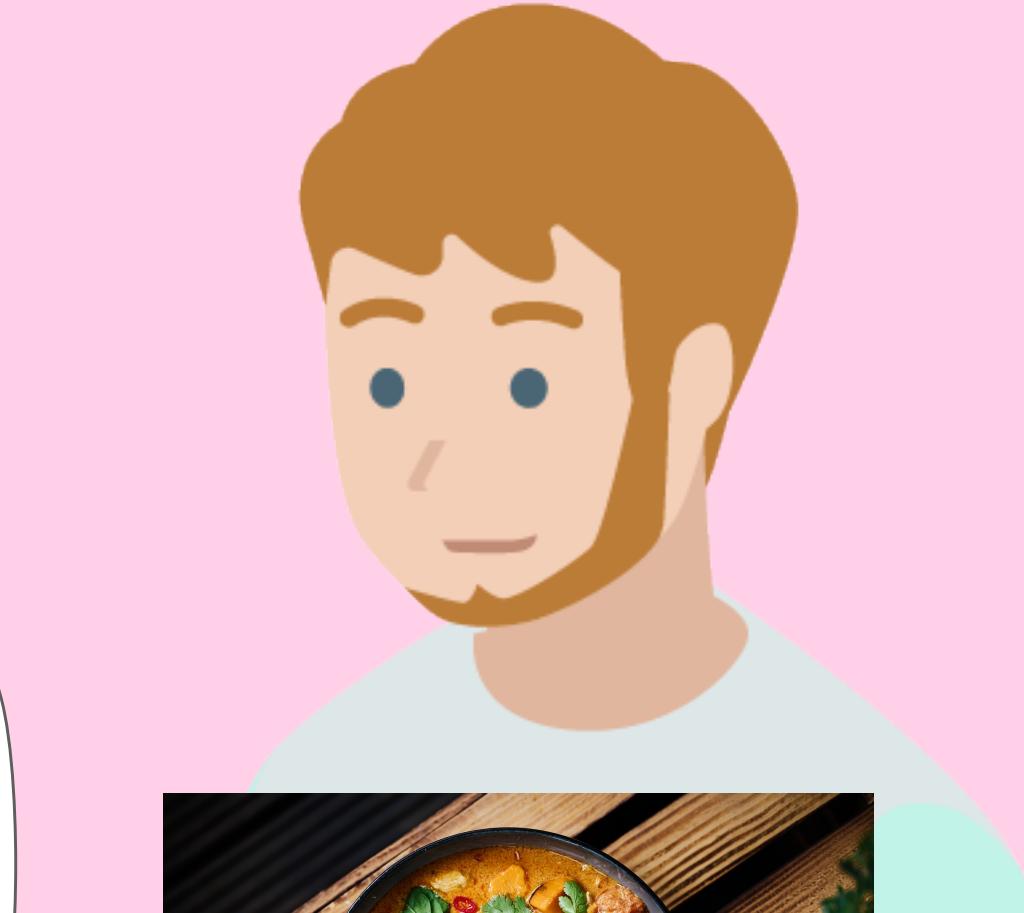
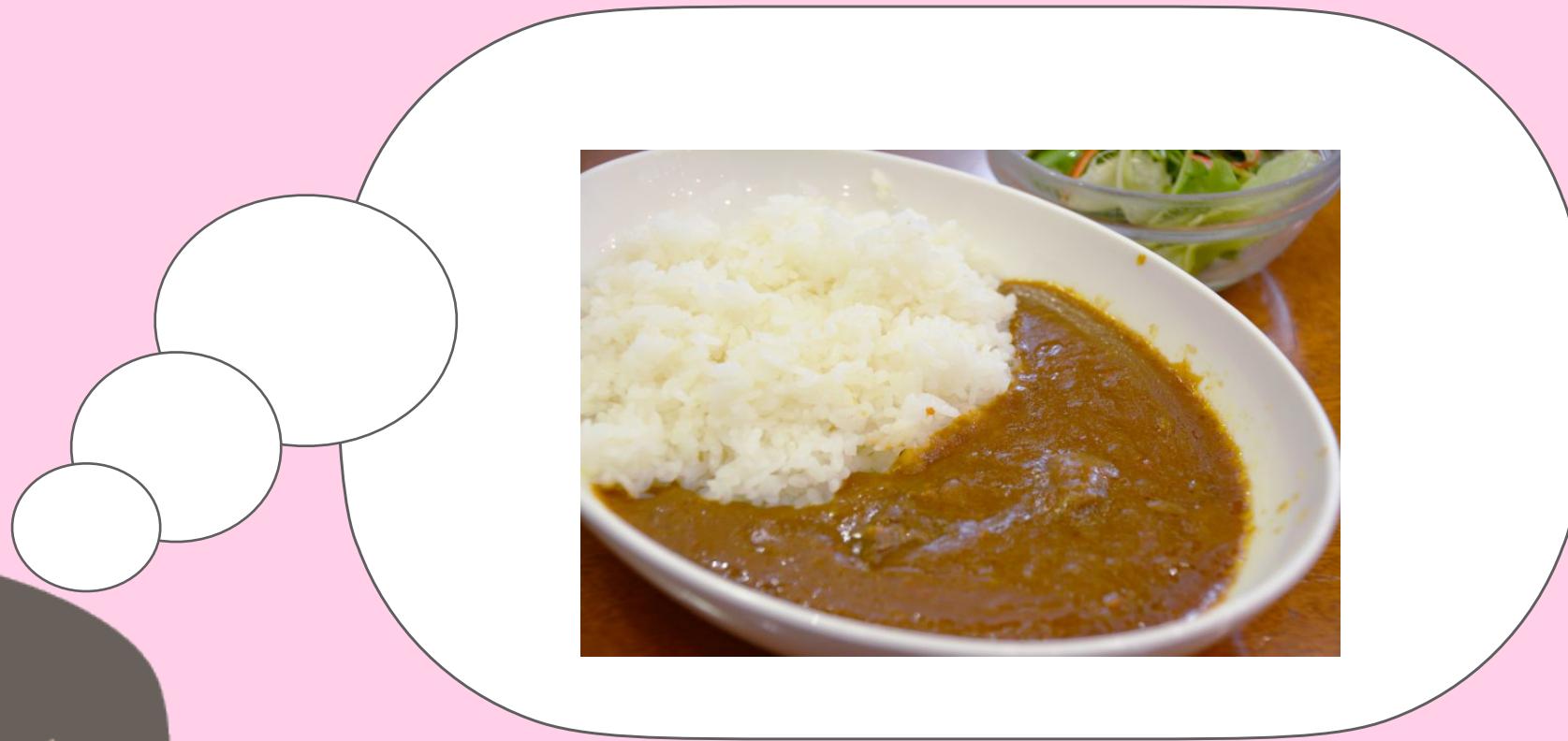
カレーに
例えてみる



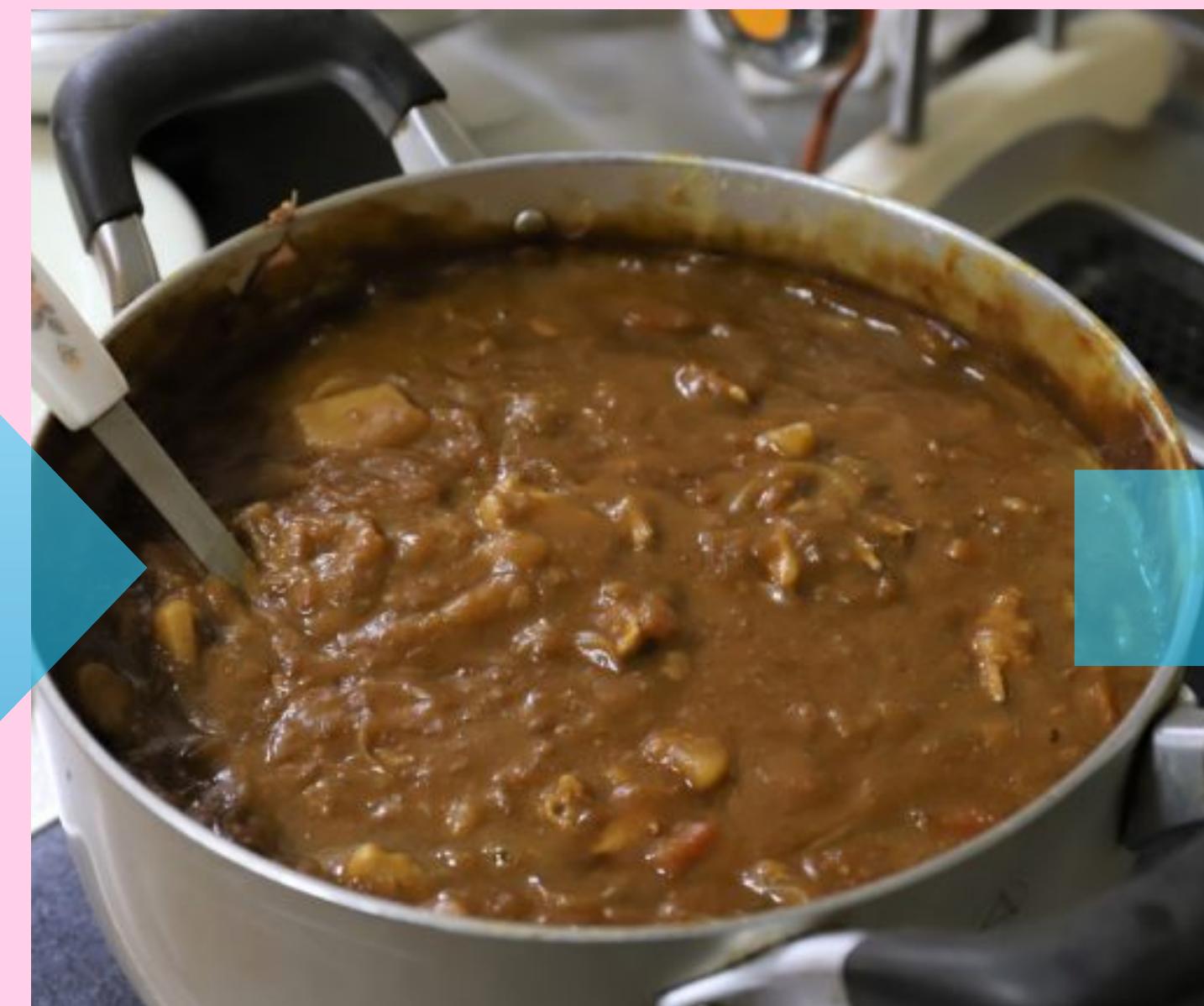
みんなで
おいしいカレーを食べよう！





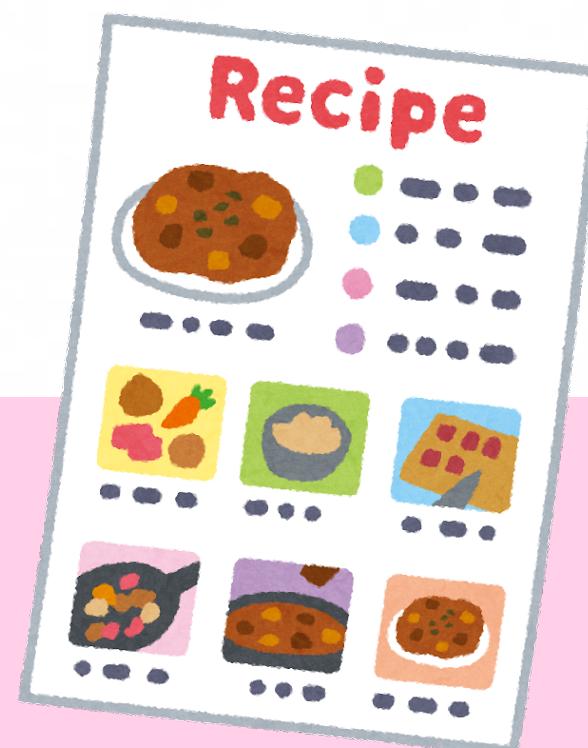


レシピが必要

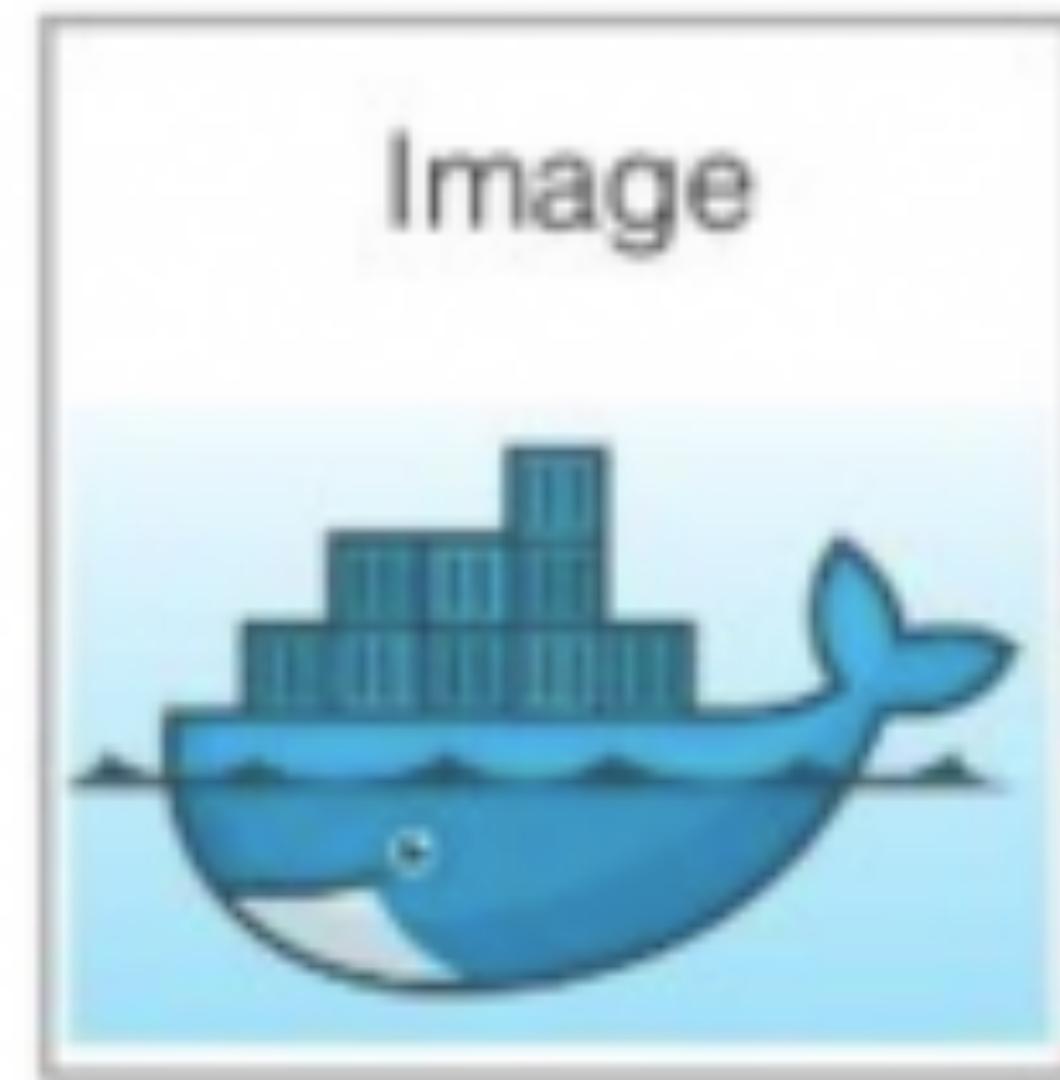




Dockerfile



build

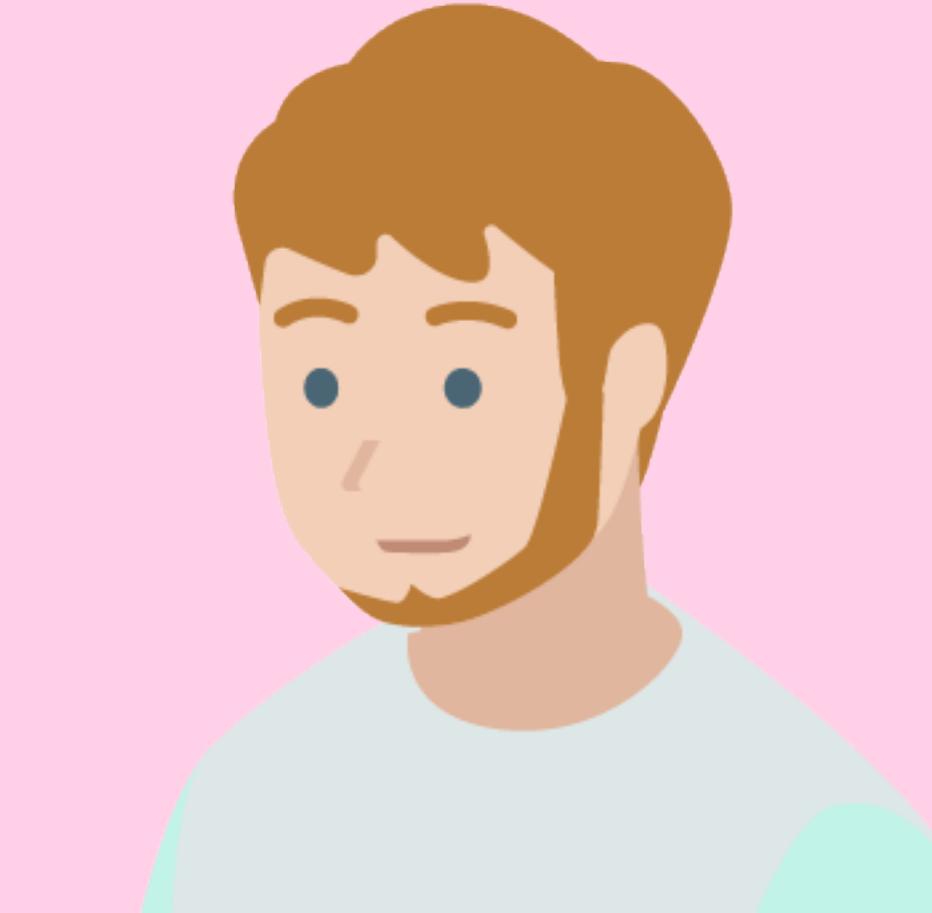
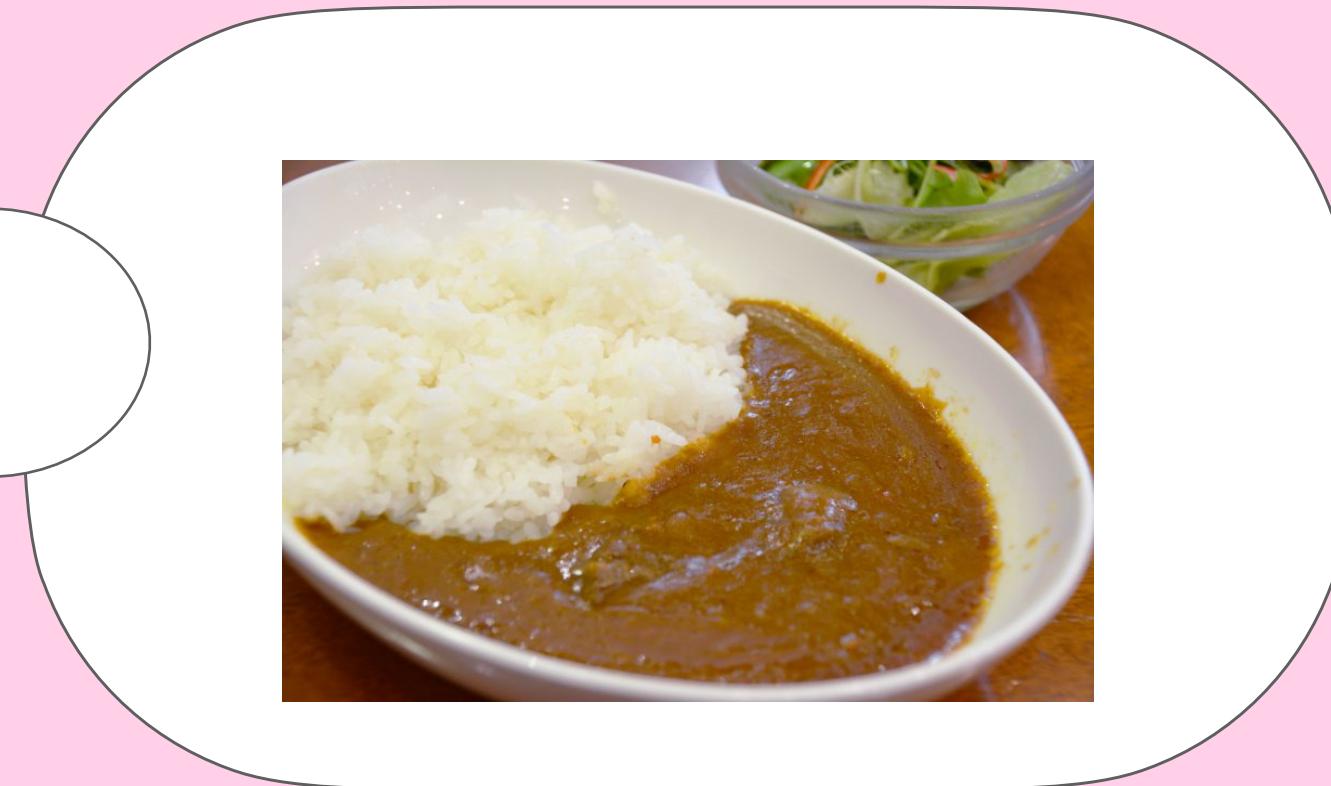
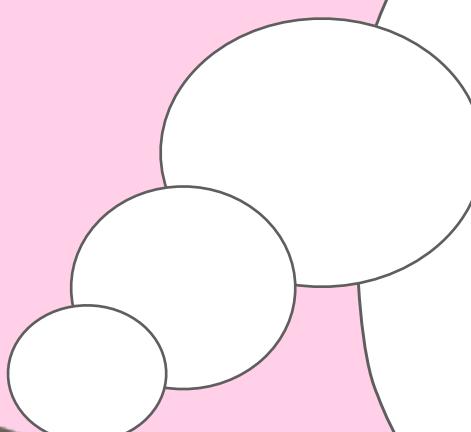


run



Docker Image

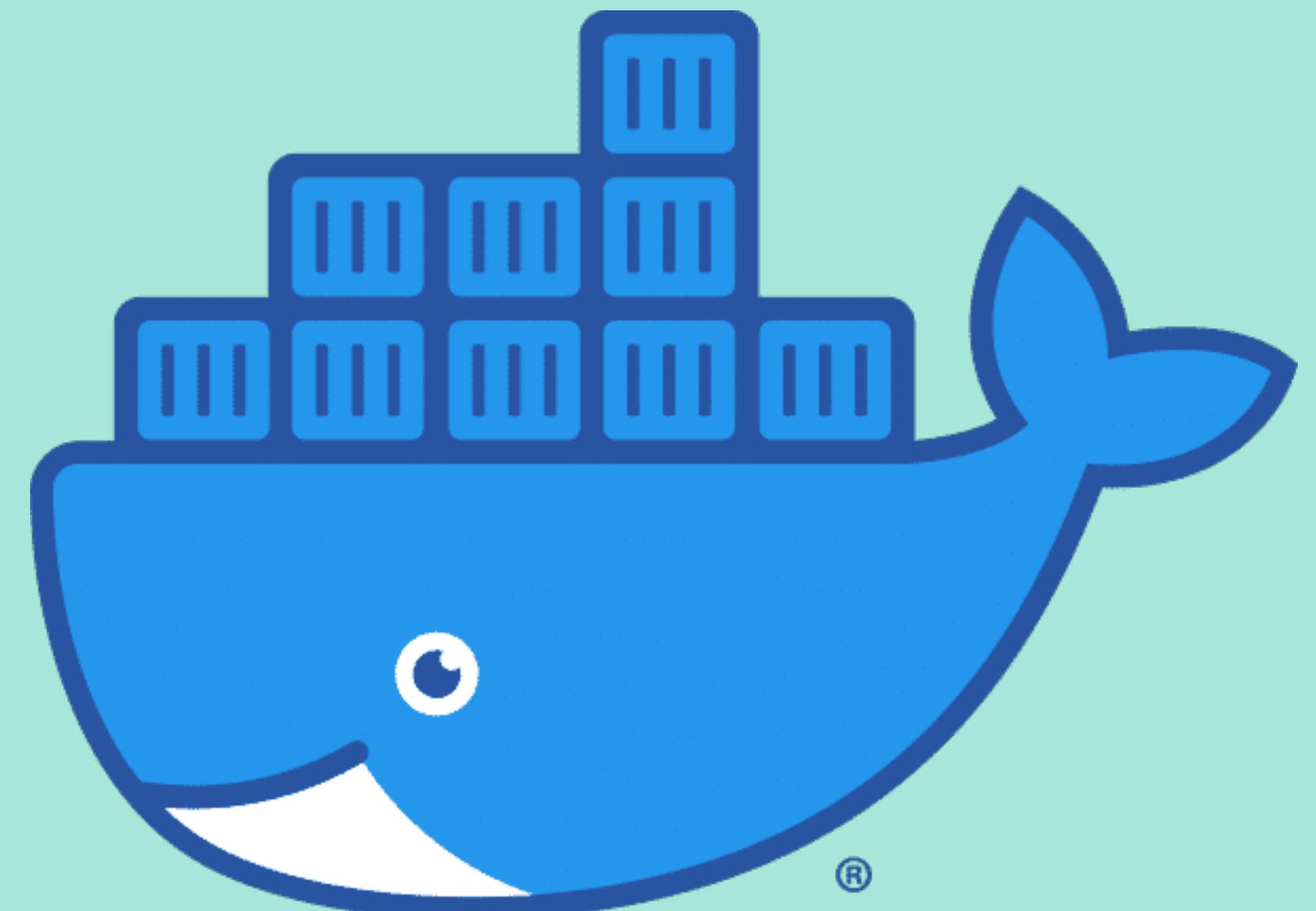




目次

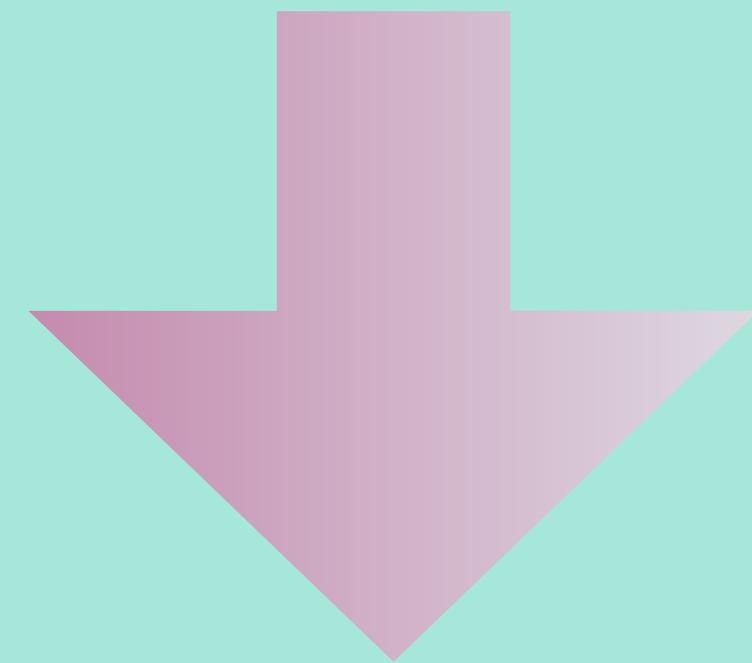
- dockerって何？
- コンテナとは？
- イメージとは？
- dockerをわかりやすく解説
- **dockerコンテナを作成してみよう！**
- docker compose
- おまけ(コンテナvs仮想化)
- まとめ

dockerファイルから
コンテナを作成してみよう！



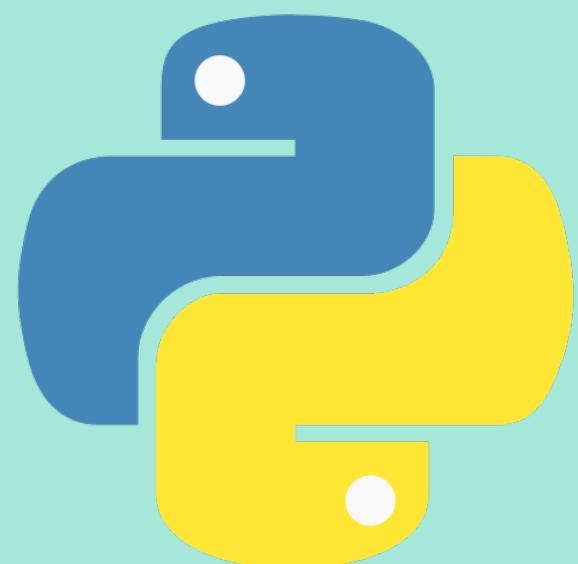


Dockerfileを作成



命令文を記載

```
↳ Dockerfile ×  
↳ Dockerfile > ...  
1  FROM python:3.9.10-buster  
2  USER root  
3  ·  
4  RUN apt-get update  
5  RUN apt-get -y install locales && \  
6  ··· localedef -f UTF-8 -i ja_JP ja_JP.UTF-8  
7  ENV LANG ja_JP.UTF-8  
8  ENV LANGUAGE ja_JP:ja  
9  ENV LC_ALL ja_JP.UTF-8  
10 ENV TZ JST-9  
11 ENV TERM xterm  
12 ·  
13 RUN apt-get install -y vim less  
14 RUN pip install --upgrade pip  
15 RUN pip install --upgrade setuptools  
16 ·  
17 RUN pip install jupyterlab==3.2.8  
18  
19
```

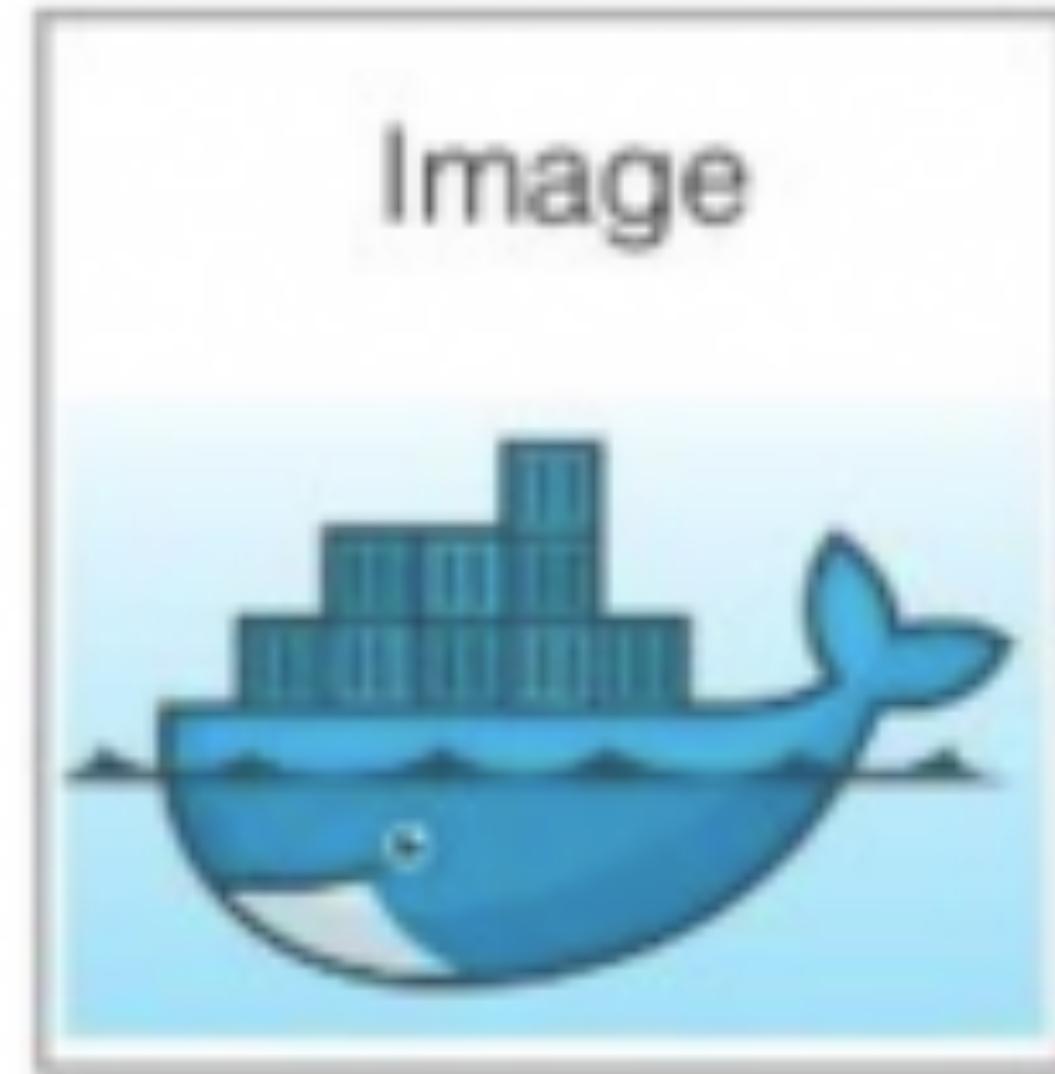




Dockerfile



build



Docker Image



run



Docker Container



```
$ docker build -t docker_test:1.0 .
```

```
$ docker images
```

dockerイメージ一覧を表示

```
(base) ~/docker-test (0.21s)
```

```
docker images
```

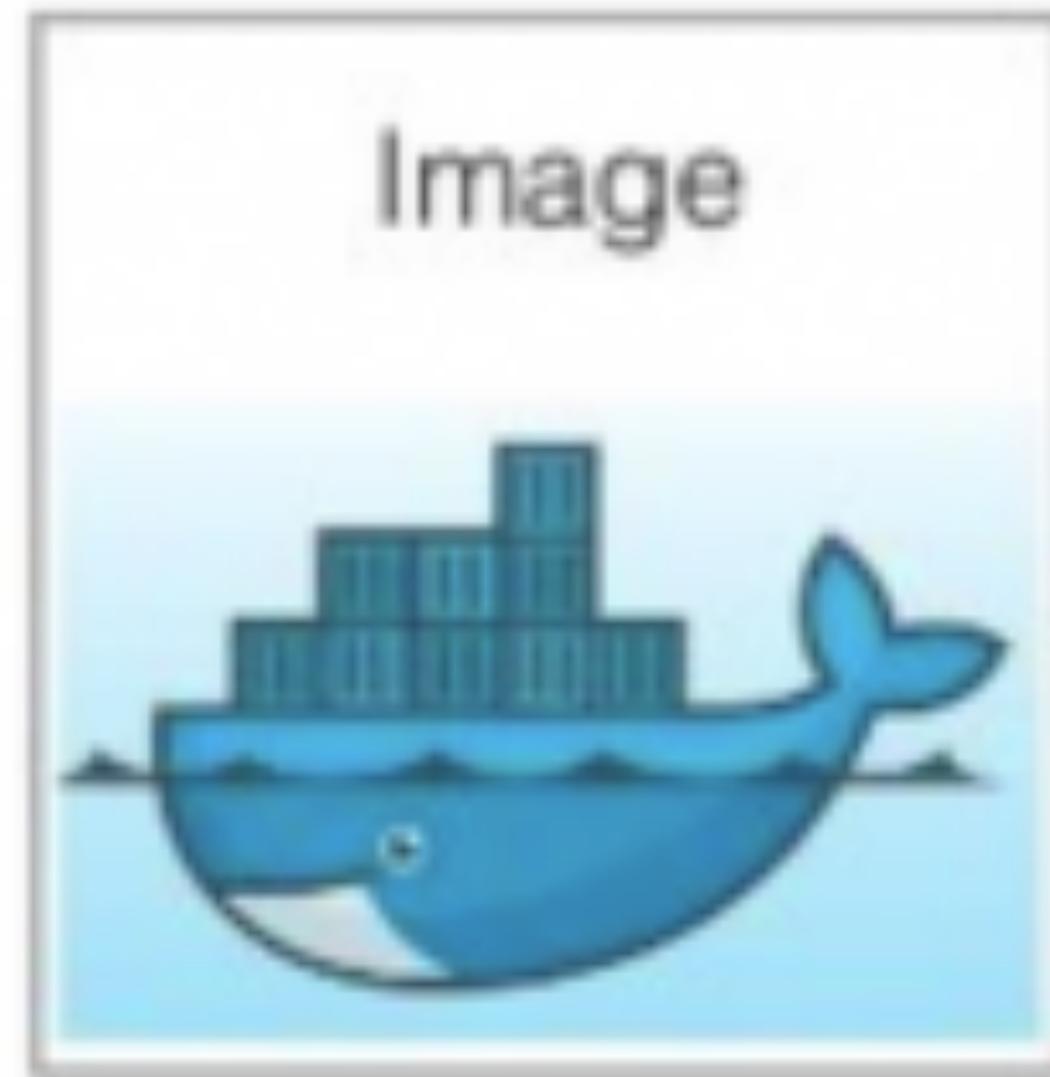
REPOSITORY	TAG
docker_test	1.0
ubuntu	20.04
develop_kp-app	latest
develop_kp-php-fpm56	latest
redis	latest
busybox	latest



Dockerfile



build



run



Docker Image



dockerイメージを起動

```
$ docker run --name test-docker -it -d docker_test:1.0
```

コンテナ名を指定

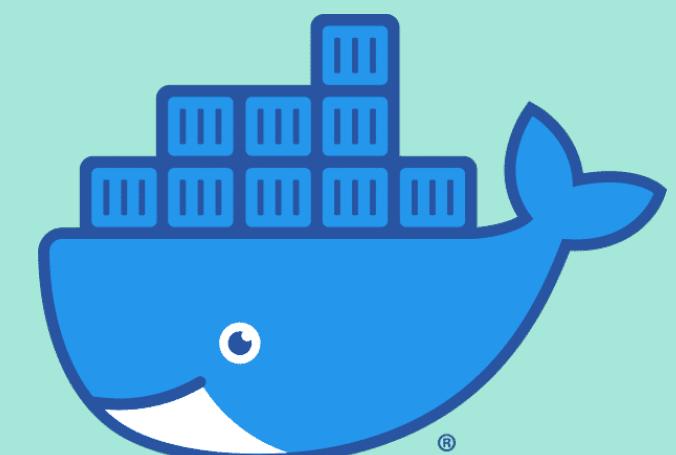
コンソールに結果を表示

デタッチドモード

イメージ名

```
(base) ~/docker-test (0.442s)
docker run --name test-docker -it -d docker_test:1.0
3d8ae0ac7136a0d80552051bff2bbbc5bb21ce94a2a71cd3cfce43ff5b488acb
```

これで起動できました



```
$ docker ps
```

起動しているdockerコンテナー一覧を表示

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
3d8ae0ac7136	docker_test:1.0	"bash"	6 seconds ago	Up 5 seconds		test-docker
199b0e0f7a1f	develop_kp-proxy	"nginx -g 'daemon of..."	2 months ago	Up 7 hours	0.0.0.0:8082->80/tcp	kp-proxy
d9ce29f64acb	develop_kp-app	"php-fpm5.6 --nodaem..."	2 months ago	Up 7 hours		kp-app
00b2e6772ae0	development_cart-proxy	"/docker-entrypoint..."	6 months ago	Up 7 hours	0.0.0.0:80->80/tcp	dev-cart-proxy
c7a9833ed775	03088287122.dkr.ecr.ap-northeast-1.amazonaws.com/dev-cart/app:latest	"docker-php-entrypoi..."	6 months ago	Up 7 hours	9000/tcp	dev-cart-app

CONTAINER ID IMAGE
3d8ae0ac7136 docker_test:1.0
199b0e0f7a1f develop_kp-proxy

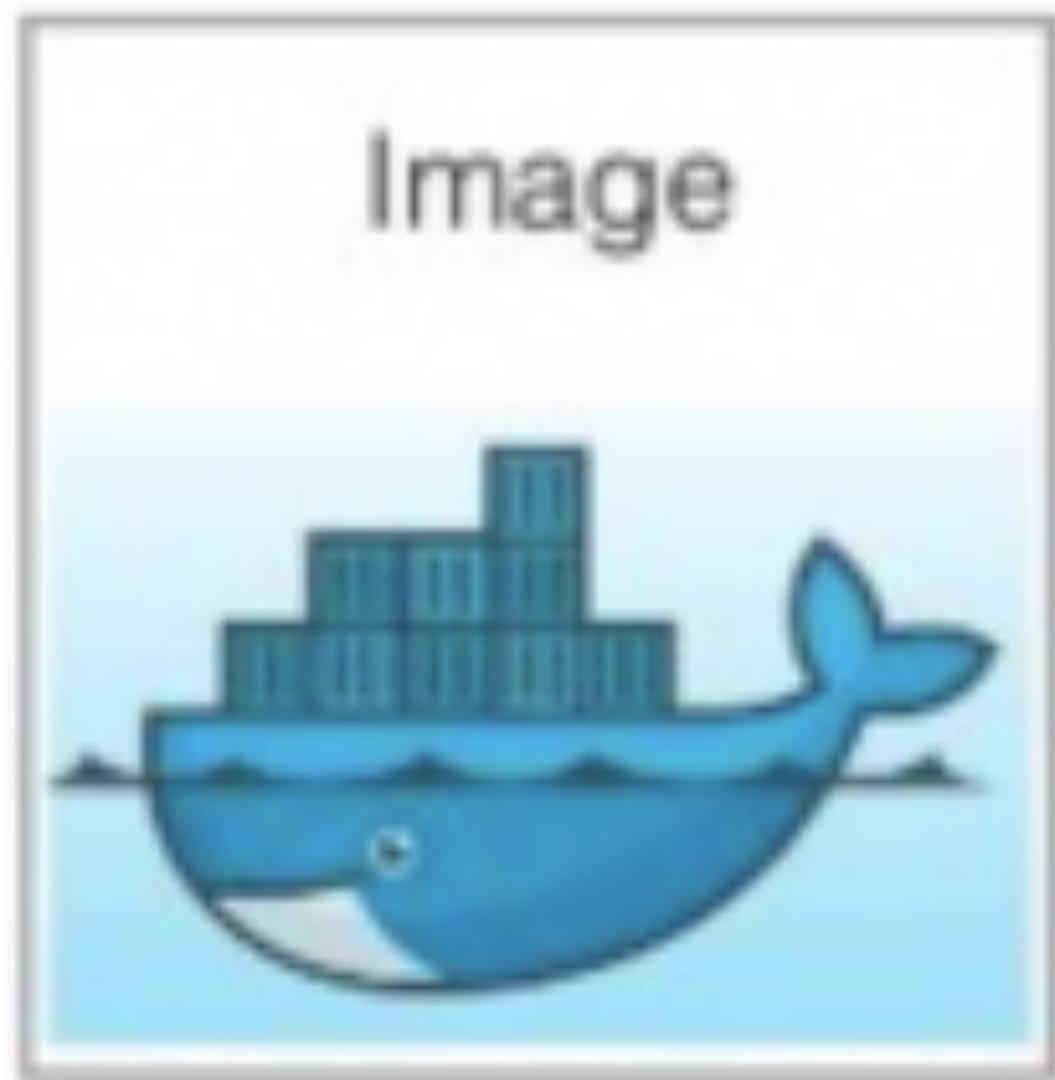
NAMES
test-docker
kp-proxy



Dockerfile



build



Docker Image



run



Docker Container



```
$ docker exec -it test-docker bash
```



コンソールに結果を表示

```
(base) ~/docker-test
docker exec -it test-docker bash
root@e54bd6c93b19:/# python --version
Python 3.9.10
root@e54bd6c93b19:/# █
```

docker hub

dockerイメージの倉庫

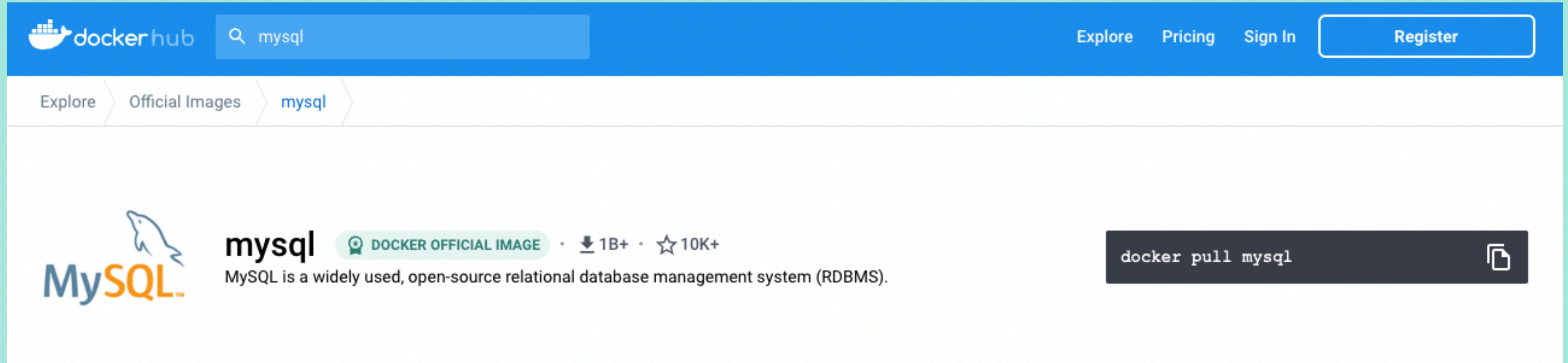
The screenshot shows the Docker Hub search interface with the query 'mysql' entered in the search bar. The results page displays four official Docker images:

- mysql** (Docker Official Image) - Updated 6 hours ago. MySQL is a widely used, open-source relational database management system (RDBMS). Available for Linux, x86-64, and ARM 64 architectures.
- mariadb** (Docker Official Image) - Updated 5 days ago. MariaDB Server is a high performing open source relational database, forked from MySQL. Available for Linux, PowerPC 64 LE, IBM Z, 386, x86-64, and ARM 64 architectures.
- percona** (Docker Official Image) - Updated 11 days ago. Percona Server is a fork of the MySQL relational database management system created by Percona. Available for Linux, x86-64, and 386 architectures.
- phpmyadmin** (Docker Official Image) - Updated 10 hours ago. phpMyAdmin - A web interface for MySQL and MariaDB. Available for x86 architecture.

On the left sidebar, there are filters for Products (Images, Extensions, Plugins), Trusted Content (Docker Official Image, Verified Publisher, Sponsored OSS), Operating Systems (Linux, Windows), Architectures (ARM, ARM 64, IBM POWER, IBM Z, PowerPC 64 LE, x86), and a search bar with the placeholder 'mysql'.

docker hub

dockerイメージの倉庫



```
$ docker pull mysql
```

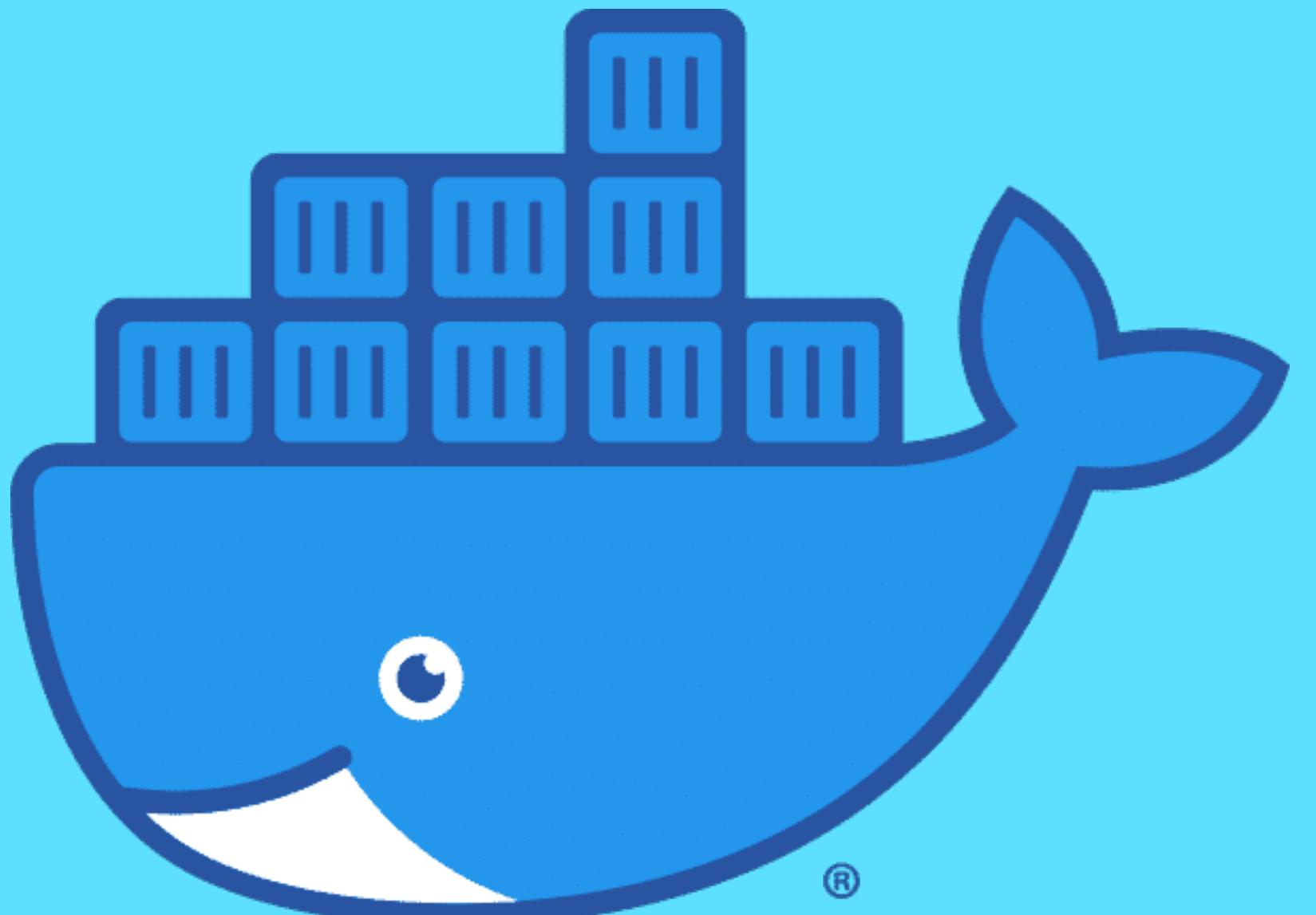
docker コマンド

docker	build	イメージを作成
docker	run	コンテナを起動
docker	stop	コンテナを停止
docker	exec	コンテナに入る
docker	rm	コンテナを削除
docker	rmi	イメージを削除
docker	Images	イメージ一覧
docker	ps	コンテナー一覧

目次

- dockerって何？
- コンテナとは？
- イメージとは？
- dockerをわかりやすく解説
- dockerコンテナを作成してみよう！
- **docker compose**
- おまけ(コンテナvs仮想化)
- まとめ

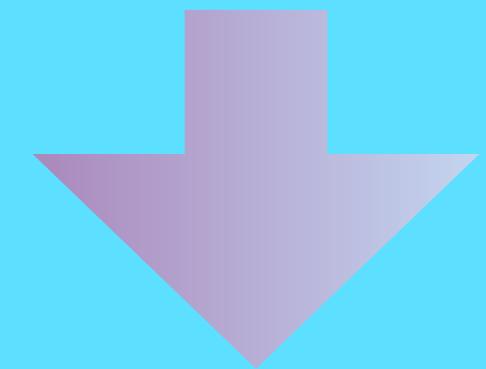
docker composeとは？





docker
Compose

docker composeとは



複数のdockerコンテナを実行するツール

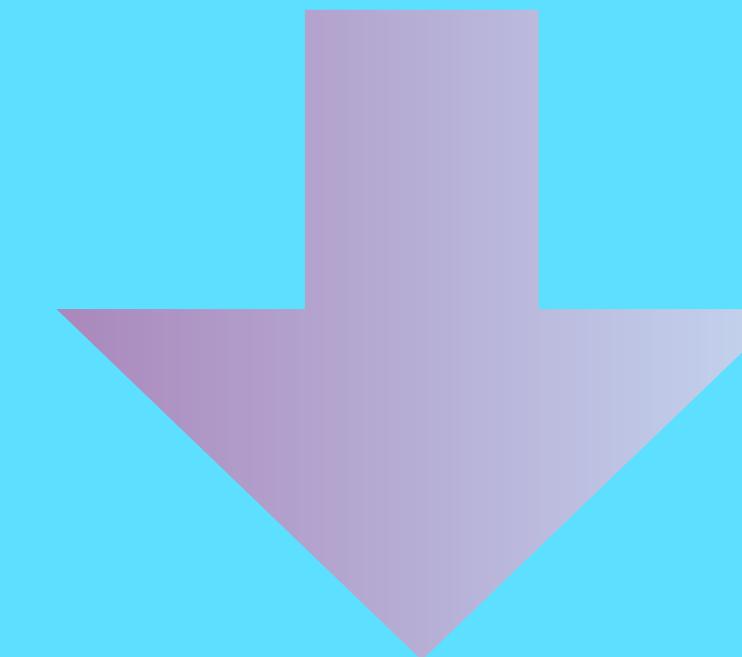
例えば



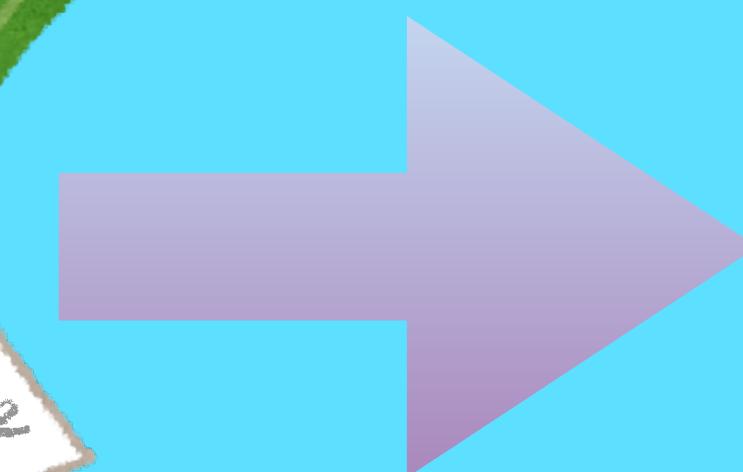
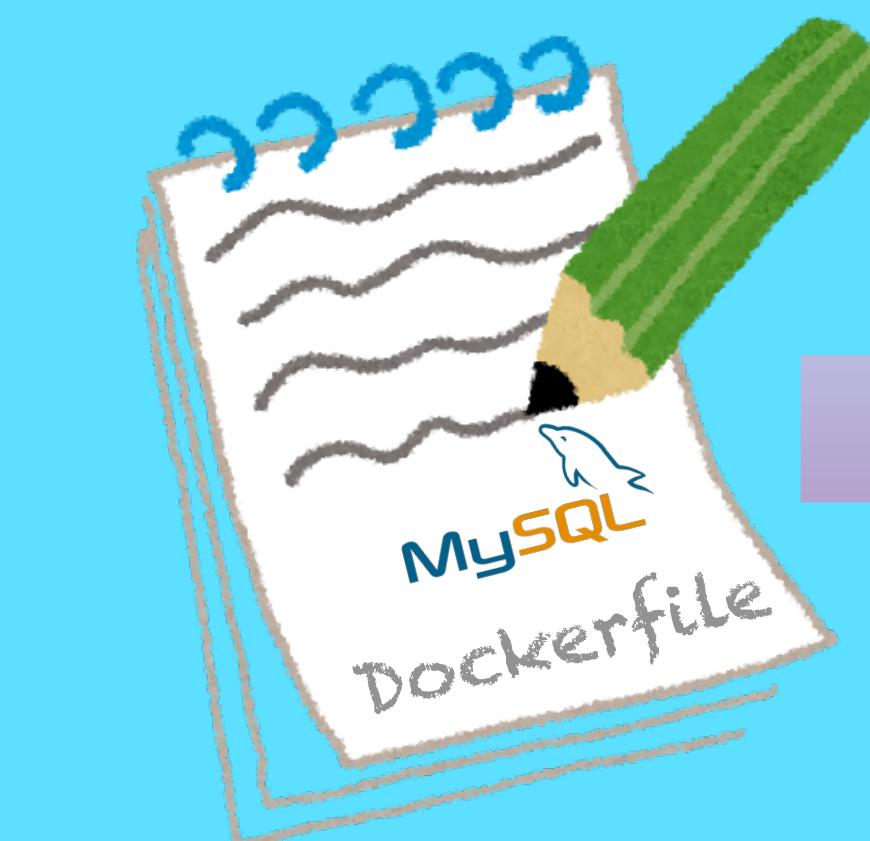
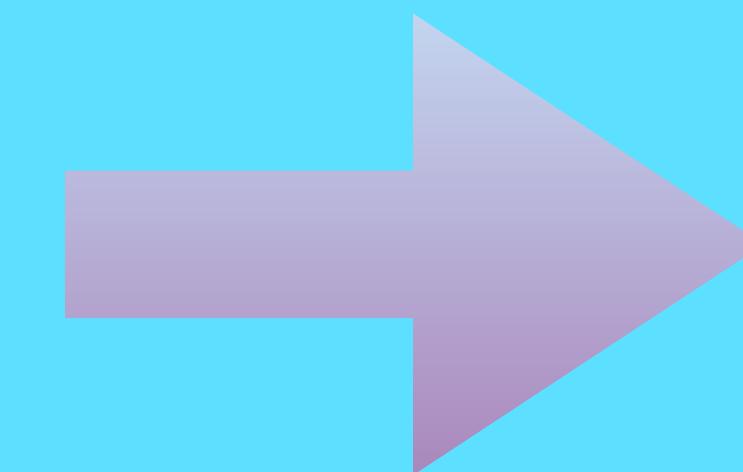
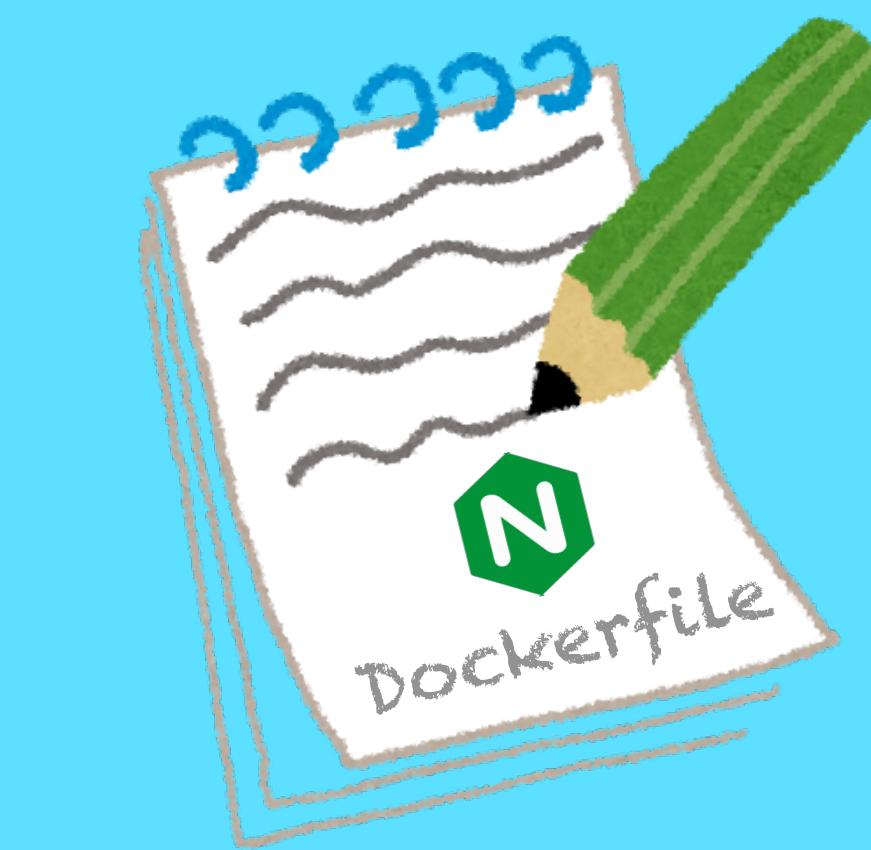
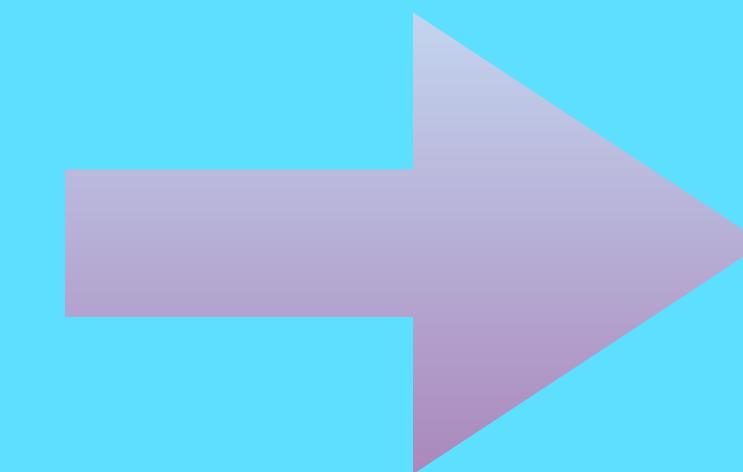
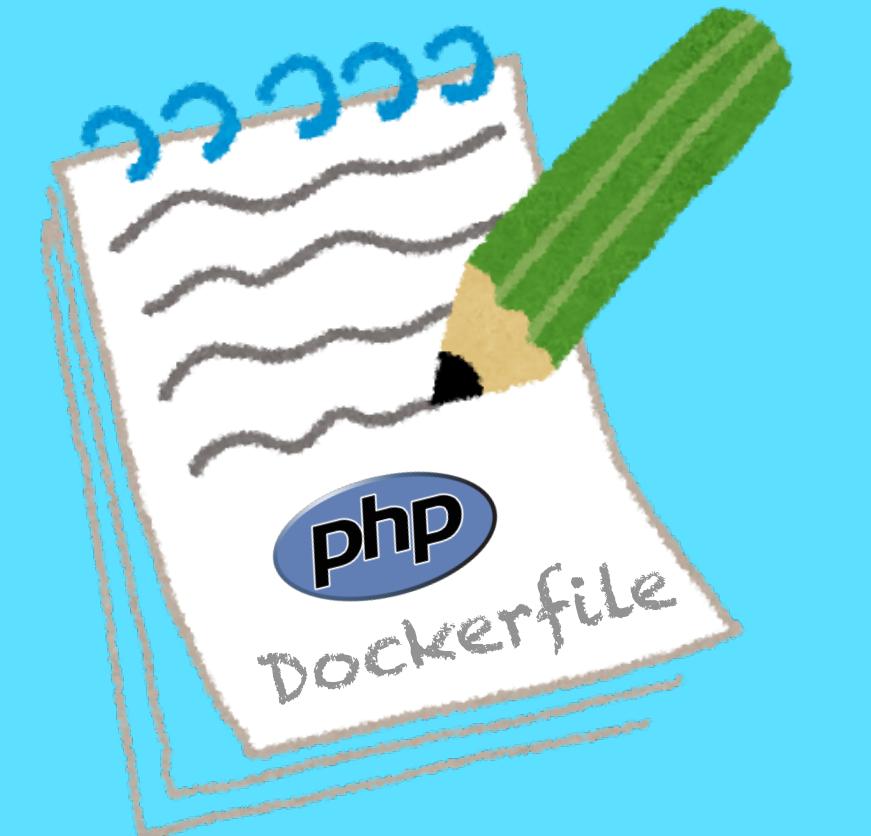
起動の順番を間違うと上手くいかない



docker-compose.ymlを作成



命令文を記載

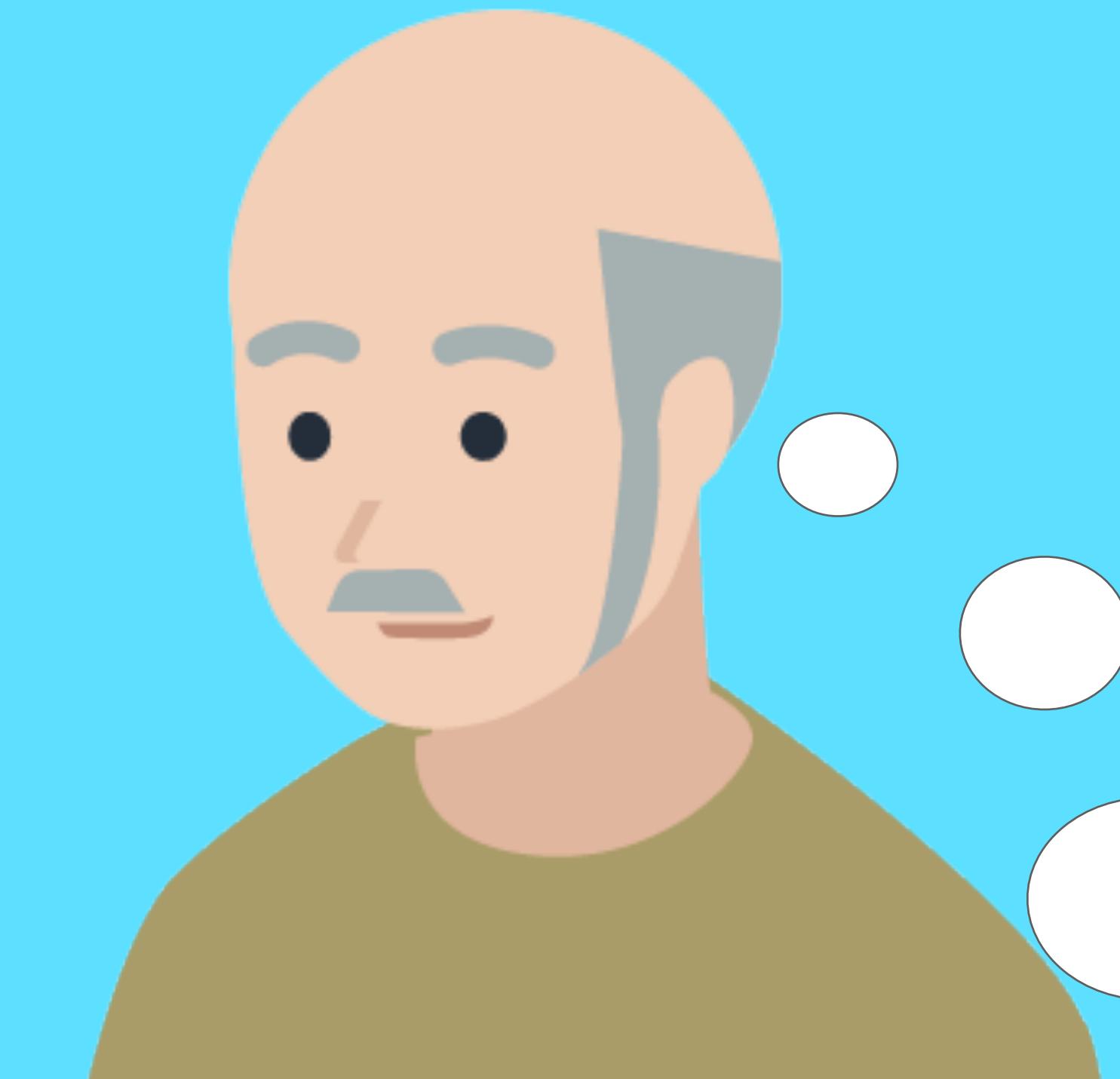


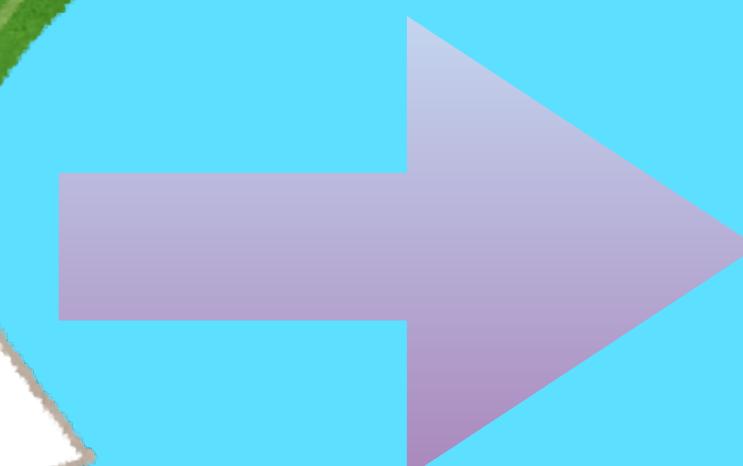
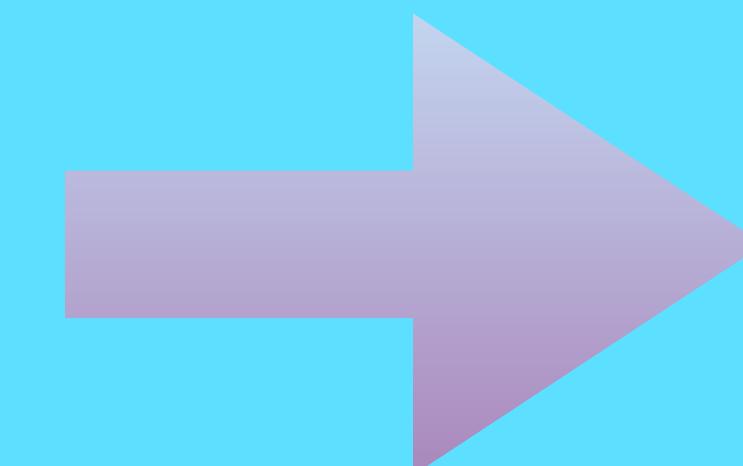
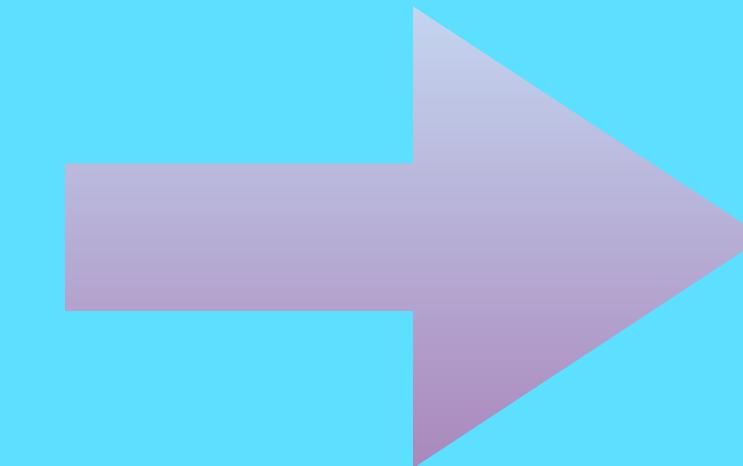
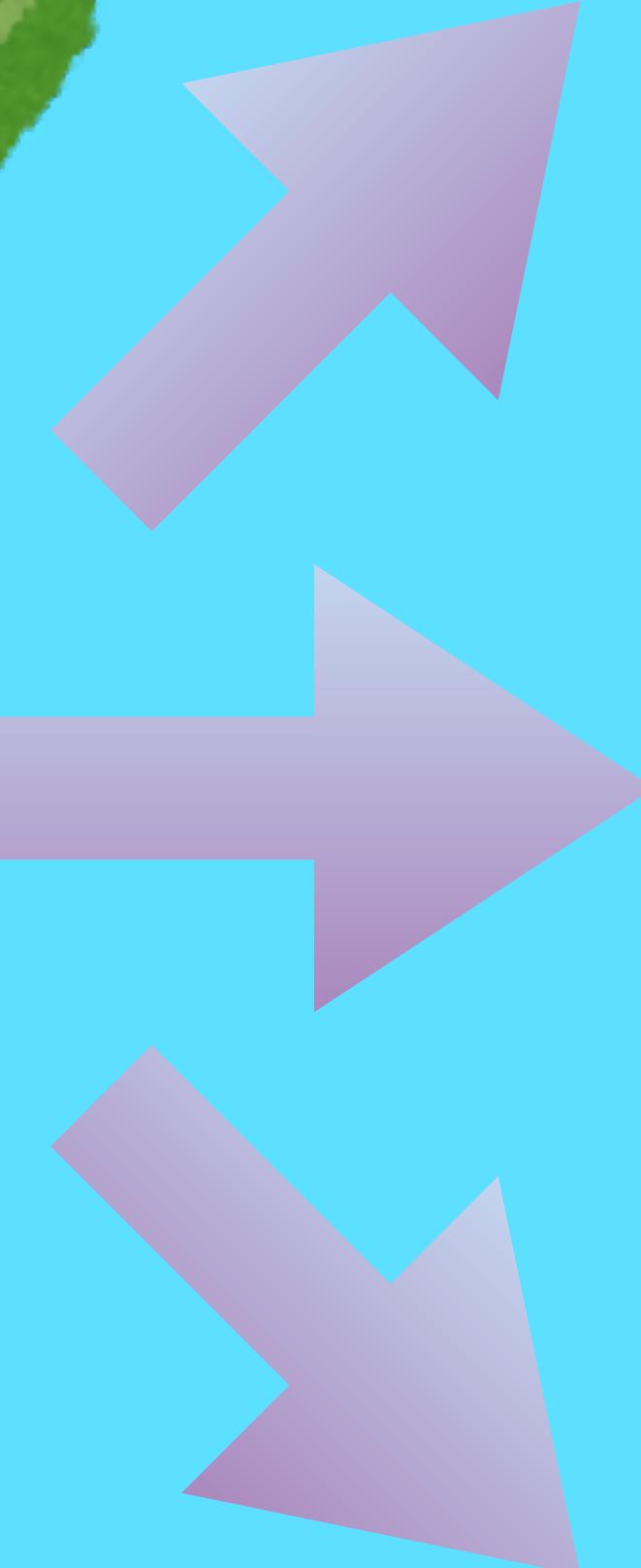
dockerコンテナ

カレー美味しいな～



でもサラダとラッシーも欲しい

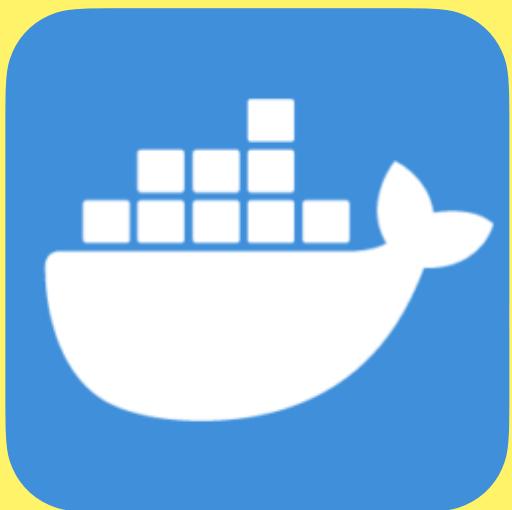




docker desktop



docker desktop



```
$ docker ps -a
```

A screenshot of the Docker Desktop application interface. The left sidebar has buttons for Containers, Images, Volumes, Dev Environments (BETA), Extensions (BETA), and Add Extensions. The main area is titled "Containers" with a "Give feedback" link. It shows a list of two running containers: "_development" and "develop", both running (2/2) and created 2 days ago. A search bar and a filter for "Only show running containers" are also present.

	NAME	IMAGE	STATUS	PORT(S)	STARTED	ACTIONS
<input type="checkbox"/>	_development	-	Running (2/2)		2 days ago	... ⋮ trash
<input type="checkbox"/>	develop	-	Running (2/2)		2 days ago	... ⋮ trash

docker desktop



```
$ docker images
```

The screenshot shows the Docker Desktop application window. The top bar includes the title "Docker Desktop", an "Upgrade plan" button, and user account settings. The left sidebar has navigation links: "Containers", "Images" (which is selected), "Volumes", "Dev Environments" (with a "BETA" badge), "Extensions" (with a "BETA" badge), and an "Add Extensions" button. The main content area is titled "Images on disk" and displays the "Images" section. It shows a list of local images with details like name, tag, status, creation time, size, and actions. The "python" image is tagged "3". The "ubuntu" image is tagged "20.04". The "busybox" image is tagged "latest". Other images listed are "develop_kp-app", "develop_kp-php-fpm56", and "a5ab4ab35b15". A search bar and filter icons are at the top of the list table.

<input type="checkbox"/>	NAME	TAG	STATUS	CREATED	SIZE	ACTIONS
<input type="checkbox"/>	develop_kp-app ebb3866a62ee	latest	In use	about 3 hours ago	415.9 MB	
<input type="checkbox"/>	python bf1bc951bd24	3	Unused	5 days ago	880.57 MB	
<input type="checkbox"/>	ubuntu 89867091bfb2	20.04	Unused	27 days ago	65.66 MB	
<input type="checkbox"/>	develop_kp-php-fpm56 e18752ae7bfa	latest	Unused	3 months ago	416.98 MB	
<input type="checkbox"/>	busybox a5ab4ab35b15	latest	Unused	4 months ago	1.41 MB	

docker desktop



\$ docker exec

The screenshot shows the Docker Desktop application window. The left sidebar has buttons for Containers, Images, Volumes, and Dev Environments (BETA). The main area shows a container named 'kp-app' (status: RUNNING). Below it are tabs for Logs, Inspect, Terminal (which is selected), and Stats. A message box says 'New integrated terminal' with a note to change preferences. The terminal window displays a command-line session:

```
# ls
app      bootstrap    composer-setup.php  Doxyfile   package-lock.json  readme.md  server.php
appspec.yml  composer.json  DB          node_modules  phpunit.xml    resources  sonar-project.properties
artisan    composer.lock  docs        package.json  public       scripts    vendor
#
#
```

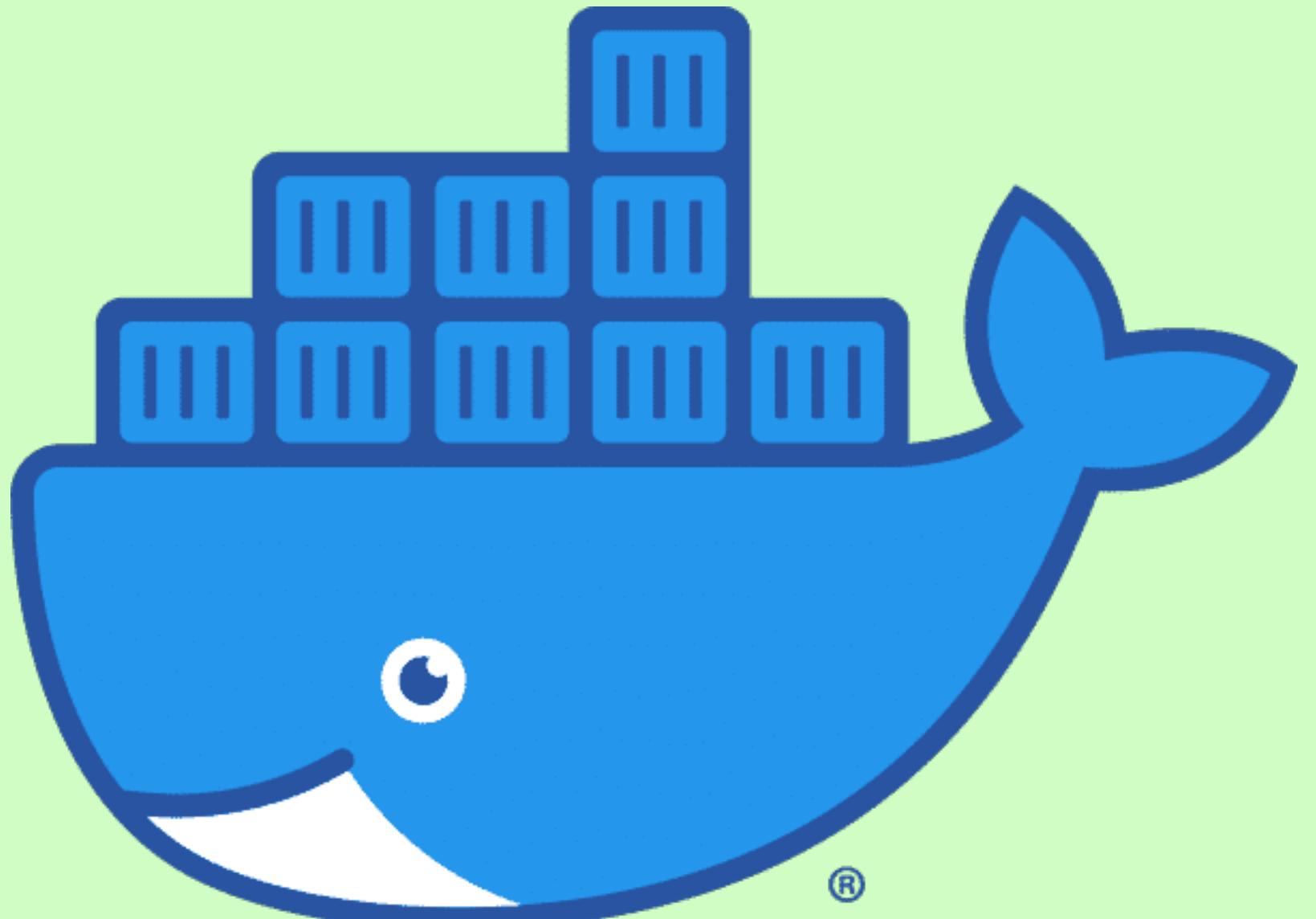
On the right side of the terminal window are icons for search, copy, and delete.

目次

- dockerって何？
- コンテナとは？
- イメージとは？
- dockerをわかりやすく解説
- dockerコンテナを作成してみよう！
- docker compose
- おまけ(コンテナvs仮想化)**
- まとめ

おまけ

コンテナ vs 仮想化



コンテナ vs 仮想化

どう違うの？

KO

docker

virtual box



コンテナ



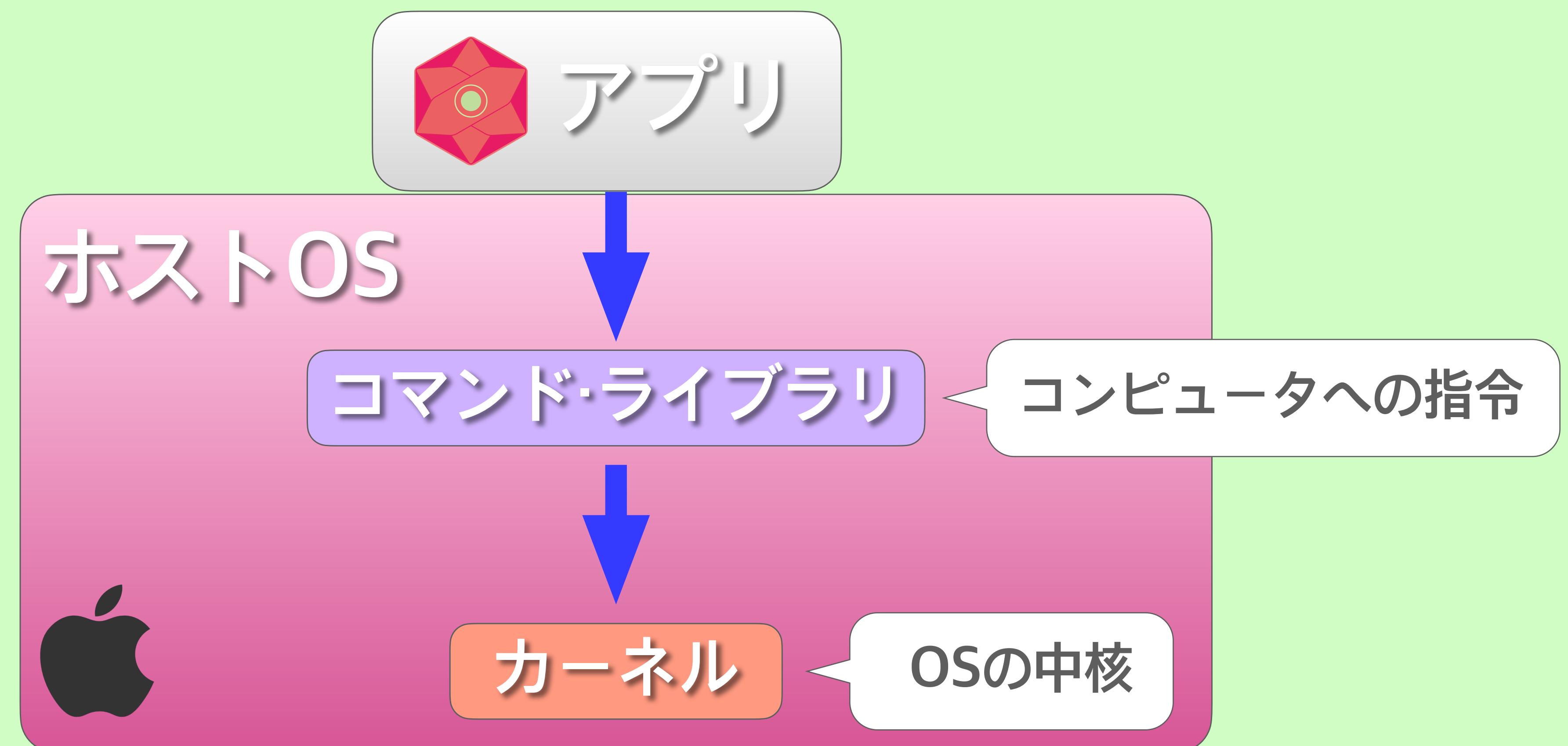
仮想化



コンテナ VS 仮想化

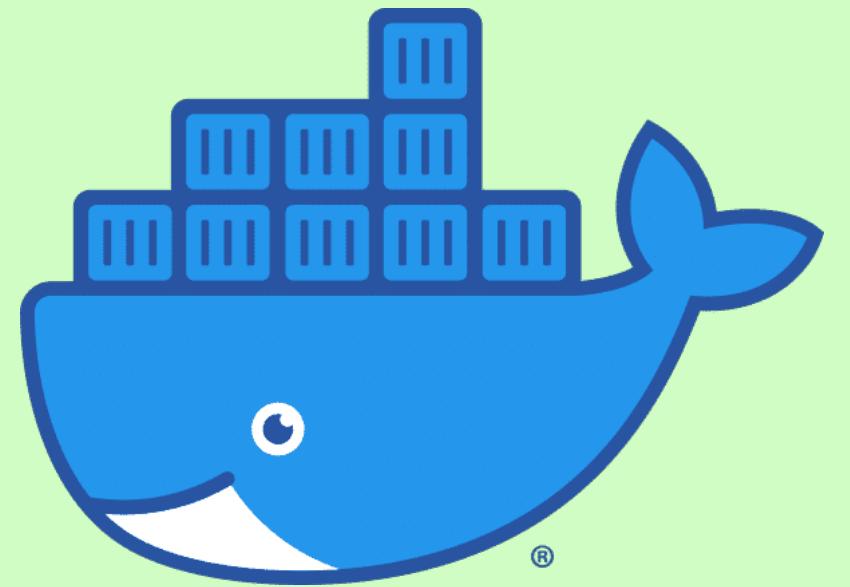


コンテナ VS 仮想化

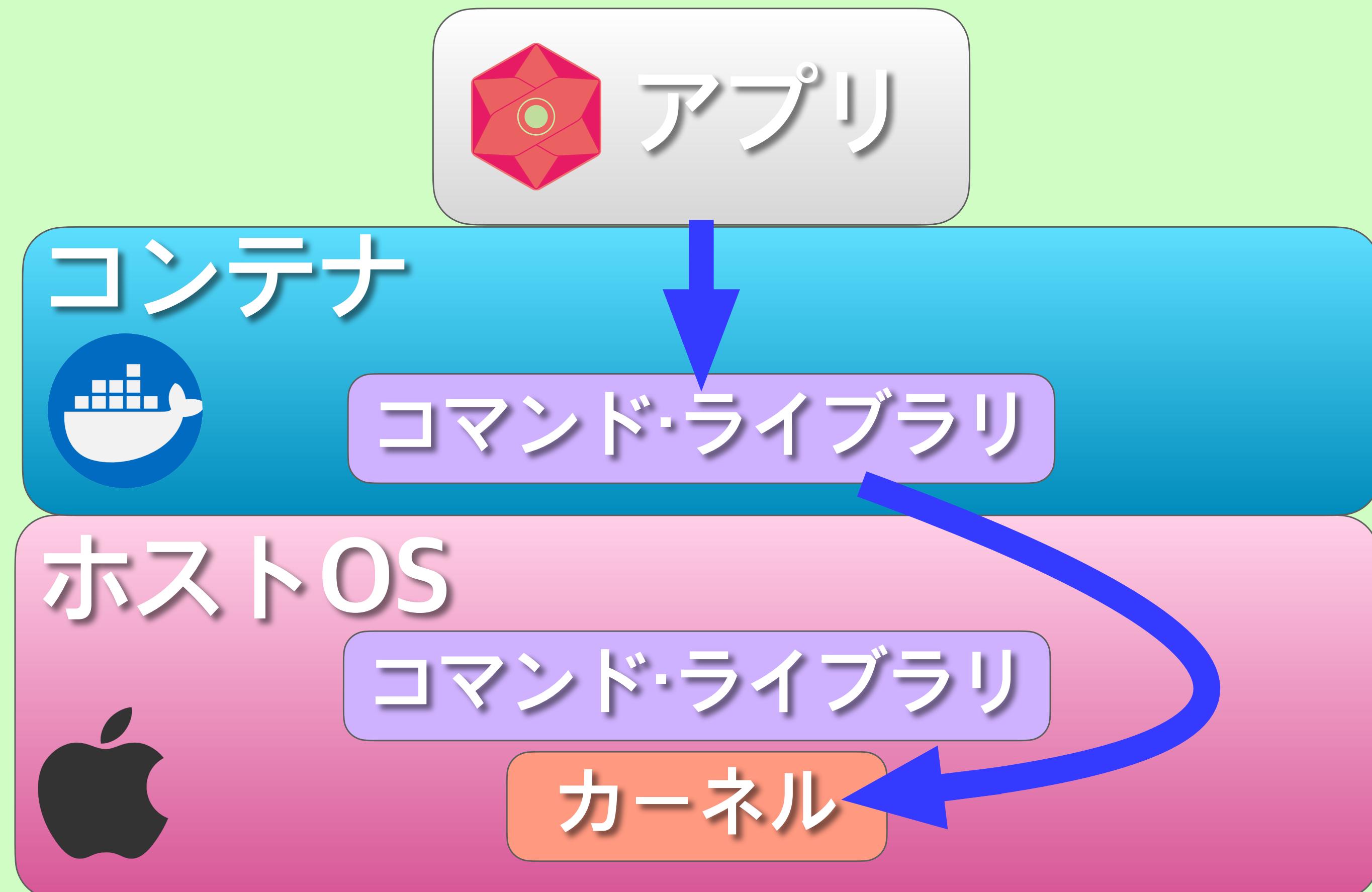


仮想化





コンテナ



コンテナ VS 仮想化

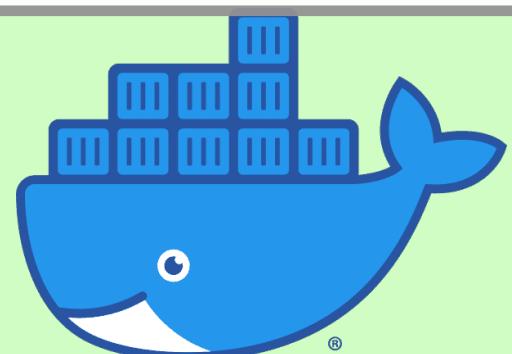
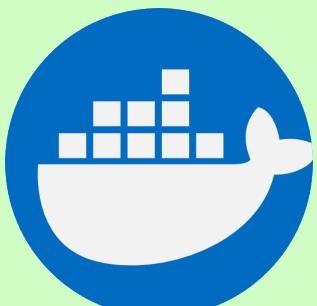
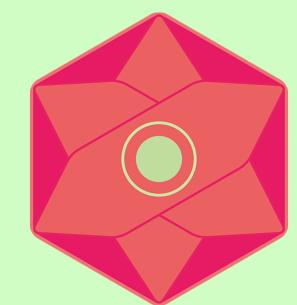
コンテナのメリット

仮想化ソフト + ゲストOSが不要

コンテナ

仮想化ソフト

軽量のため、一瞬で立ち上がる



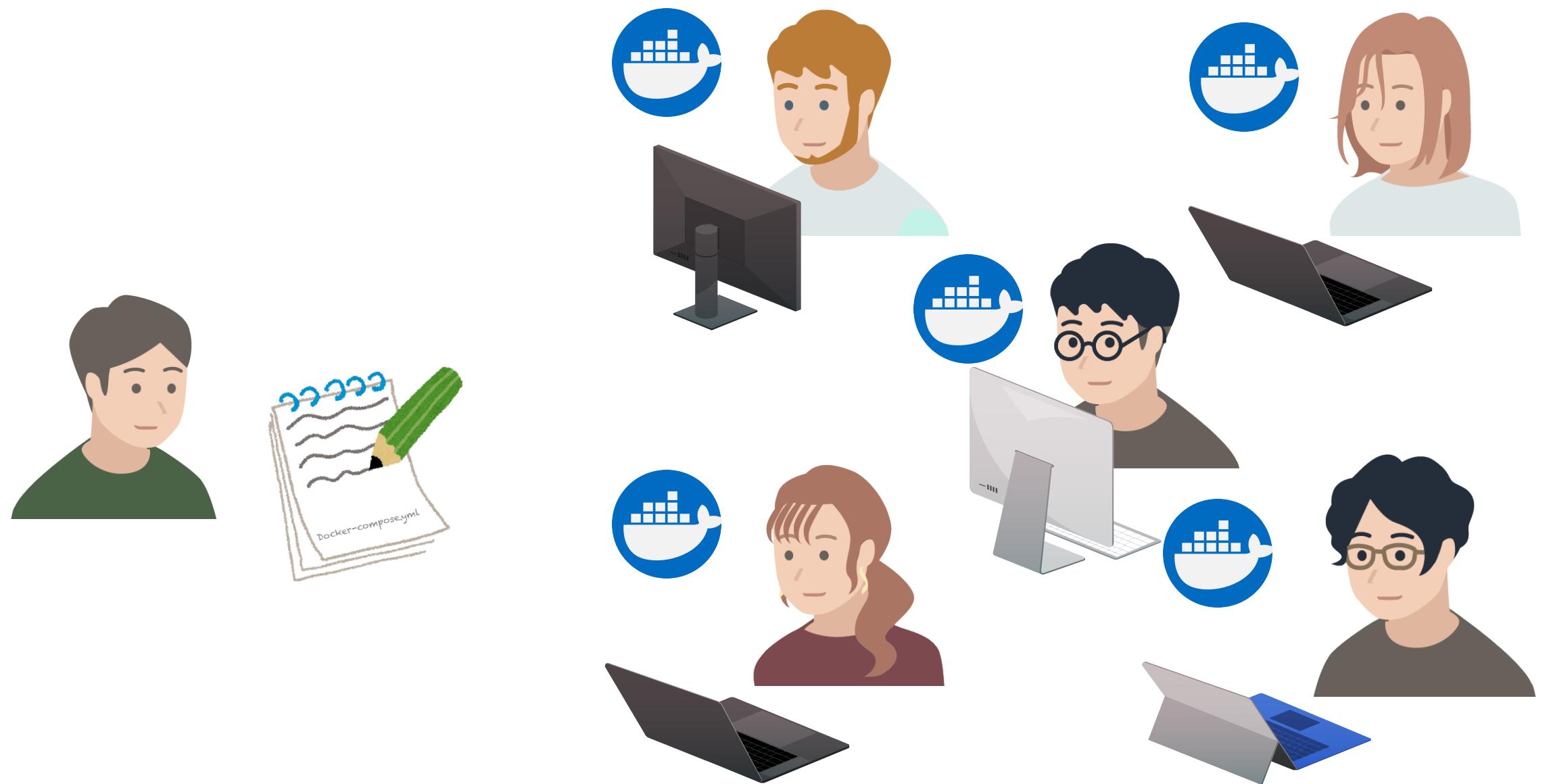
目次

- dockerって何？
- コンテナとは？
- イメージとは？
- dockerをわかりやすく解説
- dockerコンテナを作成してみよう！
- docker compose
- おまけ(コンテナvs仮想化)
- **まとめ**

dockerは

dockerは、コンテナ仮想化を実現できるツール。

dockerイメージからコンテナを作成し、開発者の環境を揃えることができる。



docker®

