Dockerと分散処理

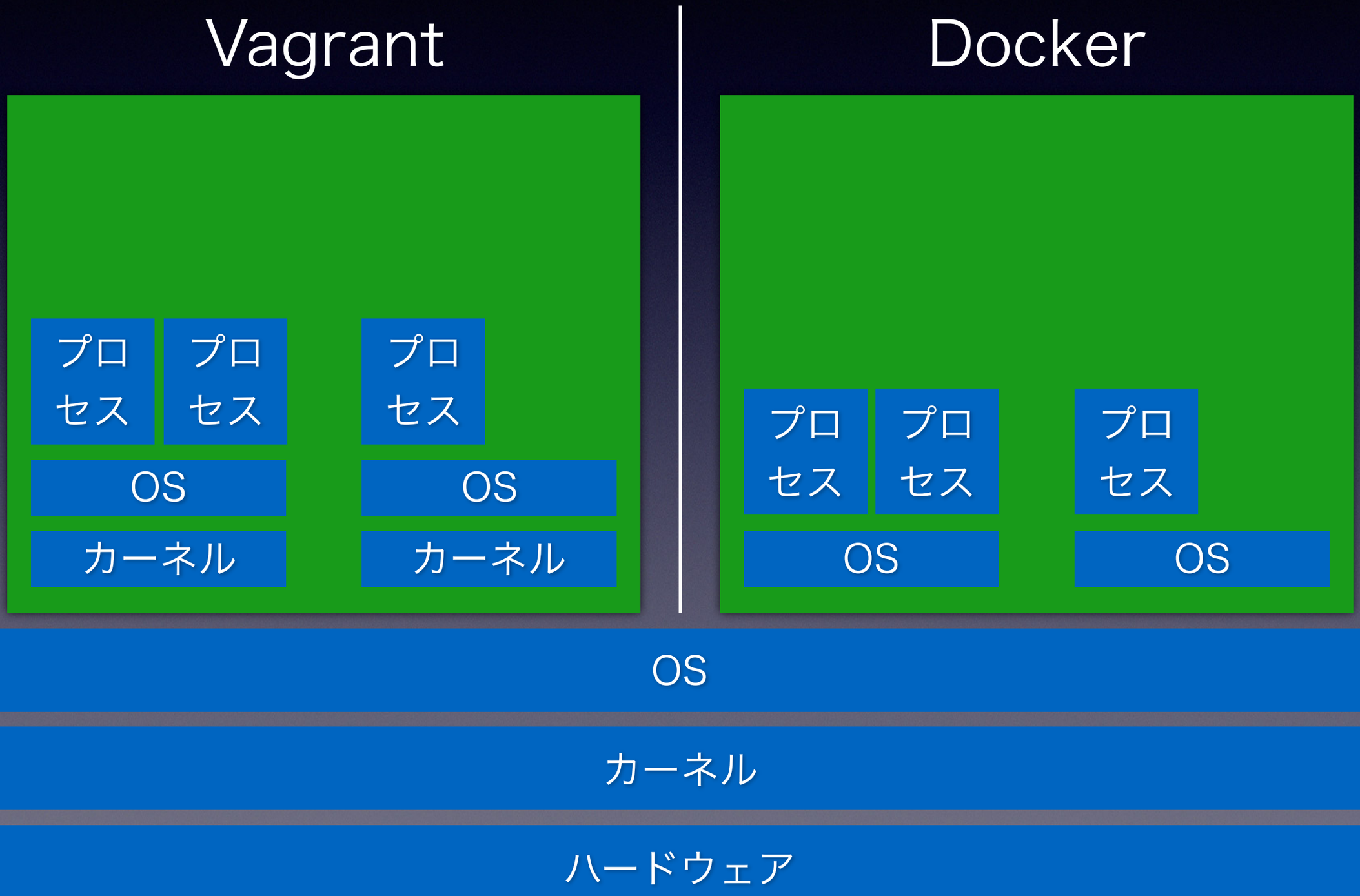
宮島健太

Dockerとは何か

・コンテナ型の仮想環境

通常の仮想環境(VM・Vagrant)とは何が違うのか

VagrantとDocker



何が違うのか

Docker

・カーネルを共有する
→共通部分は使っちゃおう

・基本的に１プロセス１コンテナ
・コンテナ同士がやり取り

Vagrant

・カーネルは共有しない
→マシンは全て仮想化！

・プロセスは仮想マシンの上
・仮想マシン内部で完結

※Vagrant上でDockerは動かせる。

Dockerに注目した理由

・軽い

→プロセスだけを仮想化するので、動くプログラムも少ない。

・素早く簡単に仮想環境が作れる。

・環境の破棄も容易

・最近注目されているので、波に乗っておきたかった。

Dockerを使って思うこと

1. アプリごとにコンテナを作成して、それぞれ連携させたい
2. 別々のサーバで別々にコンテナを動かして、連携させたい
3. リソースが空いてるサーバに優先的に立ち上げたい
4. コンテナの起動を確認し、停止してたら再起動させたい

1. アプリごとにコンテナを作成して

それぞれ連携させたい

docker-compose.yamlを使用する。

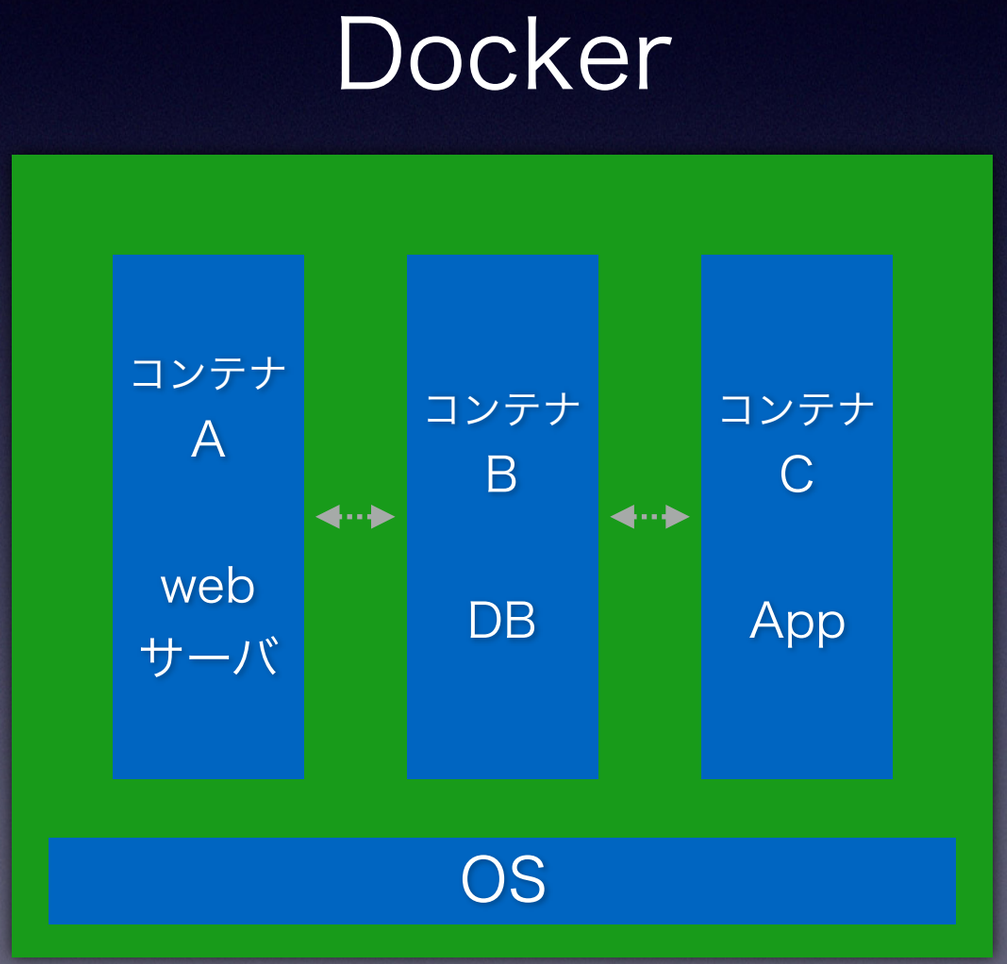
・コンテナの組み合わせ

・起動準

・コンテナ間の接続情報

コマンド１つで複数コンテナの連携ができる。

比較的簡単。



ここが重要！

2. 別々のサーバで別々にコンテナを動かして、連携させたい
3. リソースが空いてるサーバに優先的に立ち上げたい
4. コンテナの起動を確認し、停止してたら再起動させたい

これらを実現するために
・Docker Swarm
・Kubernetes
でクラスタの管理をする。

Docker Swarm

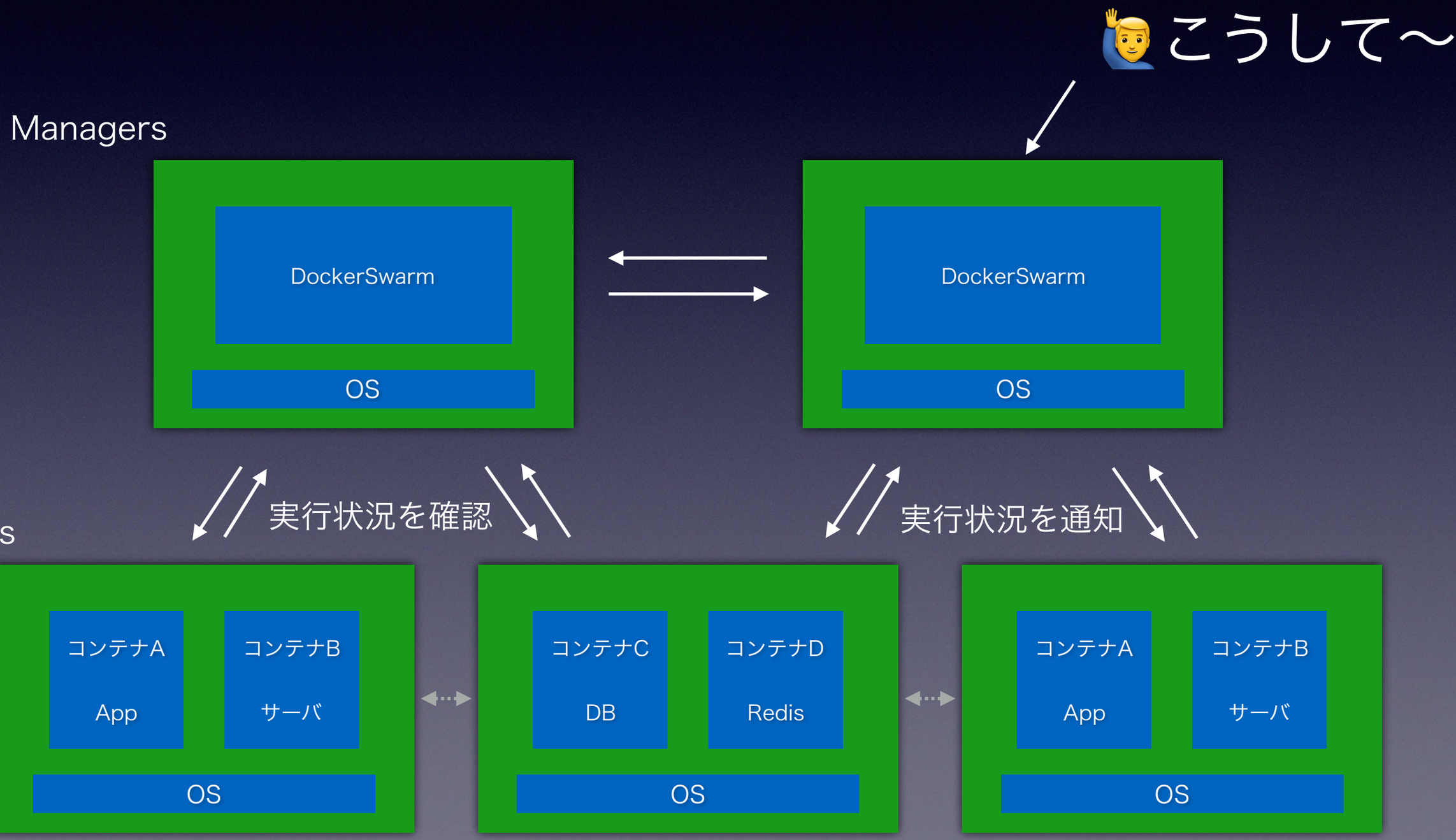
２つの役割に分かれている。

・Managers…コンテナの利用状況をチェックする

・Agents　…コンテナを実行する

Docker Swarm

Docker社 公式



Kubernetes

２つの役割に分かれている。

・Master Node …コンテナの管理

・Node　　…コンテナを実行

Docker Swarmとの相違点

・Node中に、複数のコンテナをまとめて管理

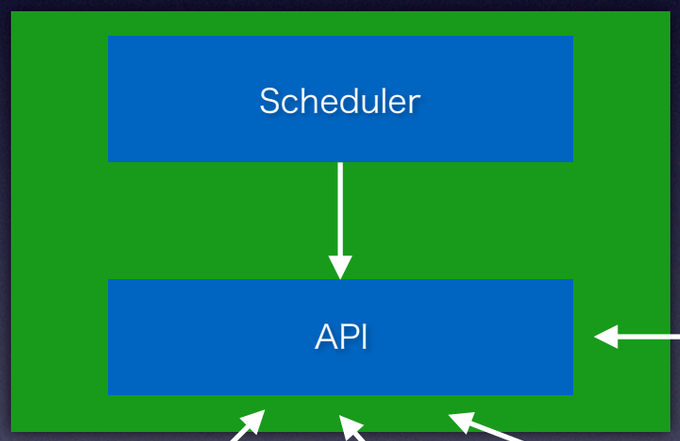
→Podと言う単位

・コンテナはKubeletと言うプログラムで起動

Kubernetes

Google社

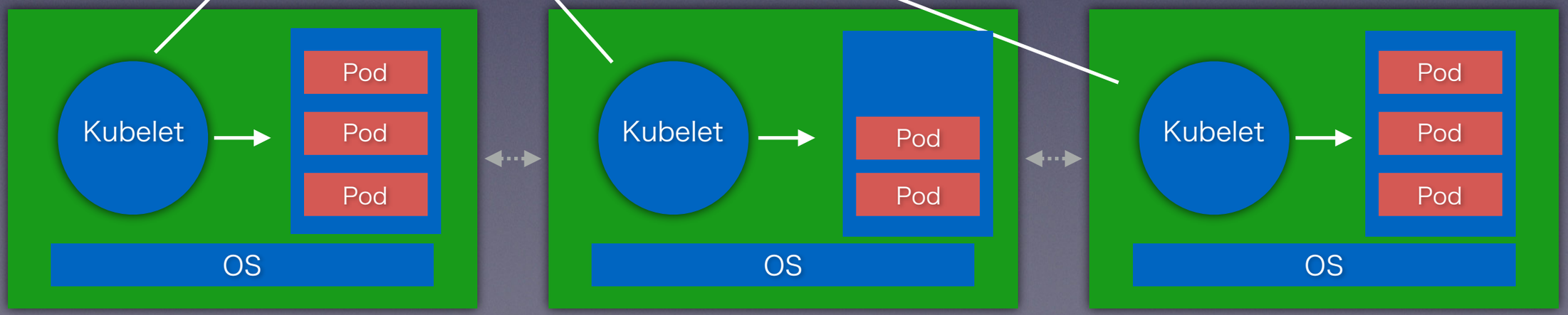
Master Node



Spec

こうして〜

Node



Kubernetes

1. オペレータは動かしたいコンテナの情報をSpecとしてKubernetesに渡す。

2. Schedulerは空いているリソースを見ながら、どのようにコンテナを配置するかを決定

3. 各ノードに常駐するKubeletと言うプログラムがその決定に従ってコンテナを起動

まとめ

Dockerを分散処理させるには

・コンピュータを複数台用意し、クラスタ化する

・コンテナの管理部と実行部で分割

・オペレータはSpecと言う情報を渡す

・コンピュータの空きリソースを効率的に使う

補足

Kubernetesをベースとしたクラウド上の
コンテナ管理サービス

- Google Kubernetes Engine
- Azure Kubernetes Service
- Amazon Elastic Kubernetes Service