私が半年間で学んだ Kubernetesの3つのこと

日本オラクル株式会社
クラウド・プラットフォーム本部
Middlewareソリューション部

自己紹介

@cotoc88

- 茂 こと（しげる こと）
- 日本オラクルのソリューションエンジニア
  - Oracle Database 数年
  - Docker/Kubernetes 6ヶ月くらい
  - Vitess/MySQL 3週間くらい
- キーボードはREALFORCE（光らない）

撮影：JapanContainerDays実行委員会

Kubernetesを
さわったことがありますか?

Kubernetesとは

- Kubernetesはコンテナ・オーケストレーションツール
  - Google発のオープンソース
- 2014年6月にローンチされ、2015年7月にCloud Native Computing Foundation（CNCF）に開発が移管
  - CNCFは著名な開発者、エンドユーザ、大手クラウドプロバイダ、ベンダーが参加しており、現在はCNCFが主体となり中立的な立場で開発が進められています

私が半年間で学んだKubernetesの3つのこと

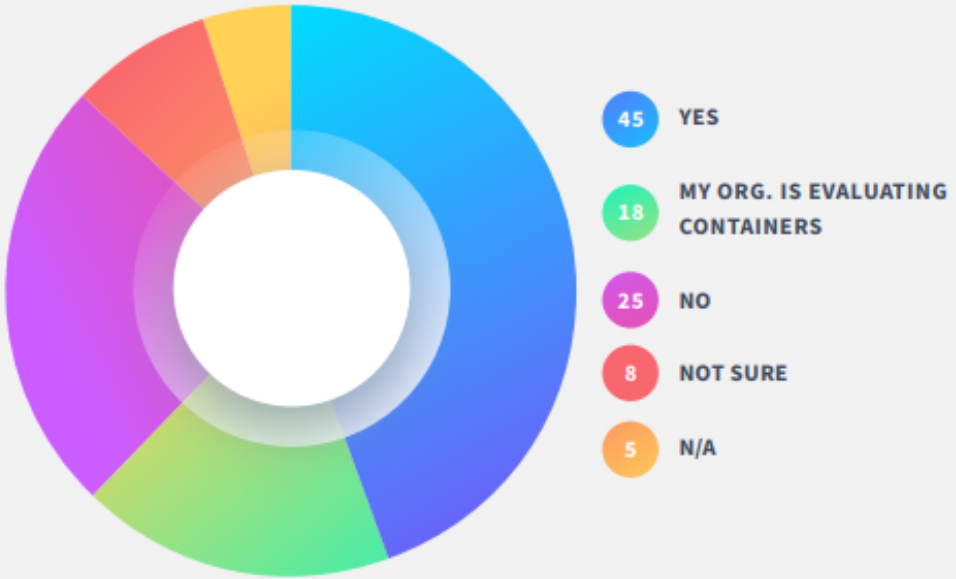
1. Kubernetes、なんだかすごそうだ！
   - Kubernetes登場の背景・メリット
2. Kubernetes、なんだか大変そうだ！
   - 最初にぶつかる壁
3. Kubernetesで広がる世界！

Container/Kubernetesってなぜ流行ってるの?

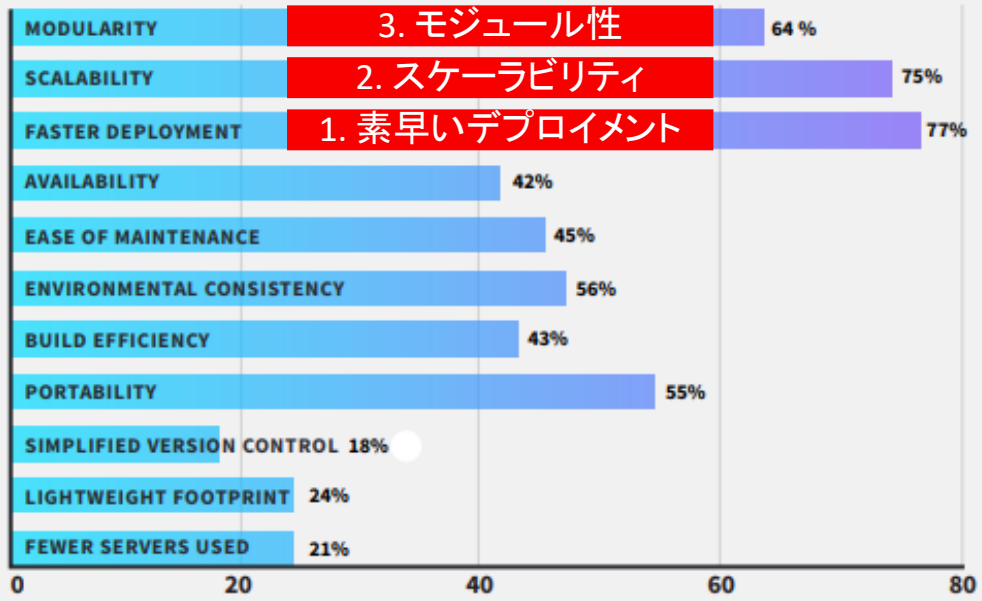
コンテナは流行っているらしい

世界中の企業で採用が進んでいる

GRAPH 01. Does your organization currently use container technologies?



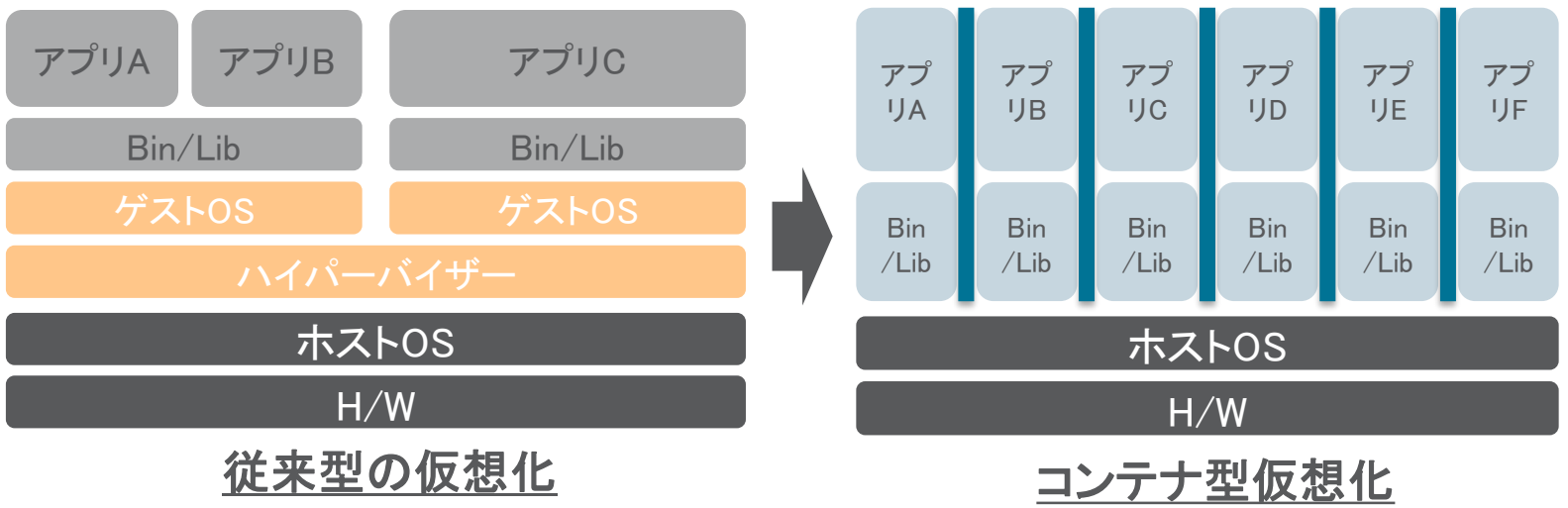
GRAPH 02. What benefits do container technologies offer your organization?



コンテナってなに…?

コンテナ ＝ 軽量・高速な仮想マシン

- コンテナ型仮想化という技術で実現されている、仮想マシンの一種
- 軽量・高速・高密度にアプリケーションを実行させることが可能

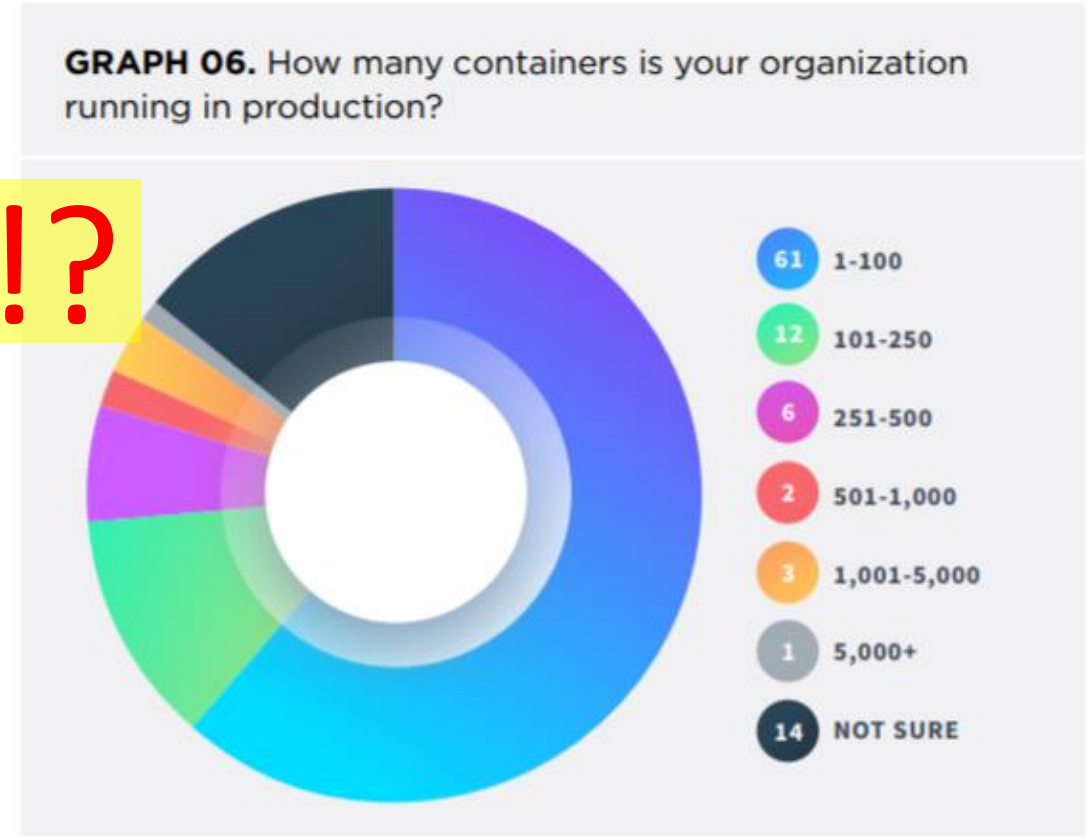


従来型の仮想化

コンテナ型仮想化

- ✓ 軽容量
  - カーネル部分をホストOSと共有するため軽量
  - 高い可搬性
- ✓ 高速
  - 起動が早い
  - オーバヘッドが少なくパフォーマンスが高い
- ✓ モジュール化
  - 他の構成要素に及ぼす影響が最小限

5000+!?



どうやって面倒を見れば…

軽くて集約できるもんだからついつい数が増えちゃう

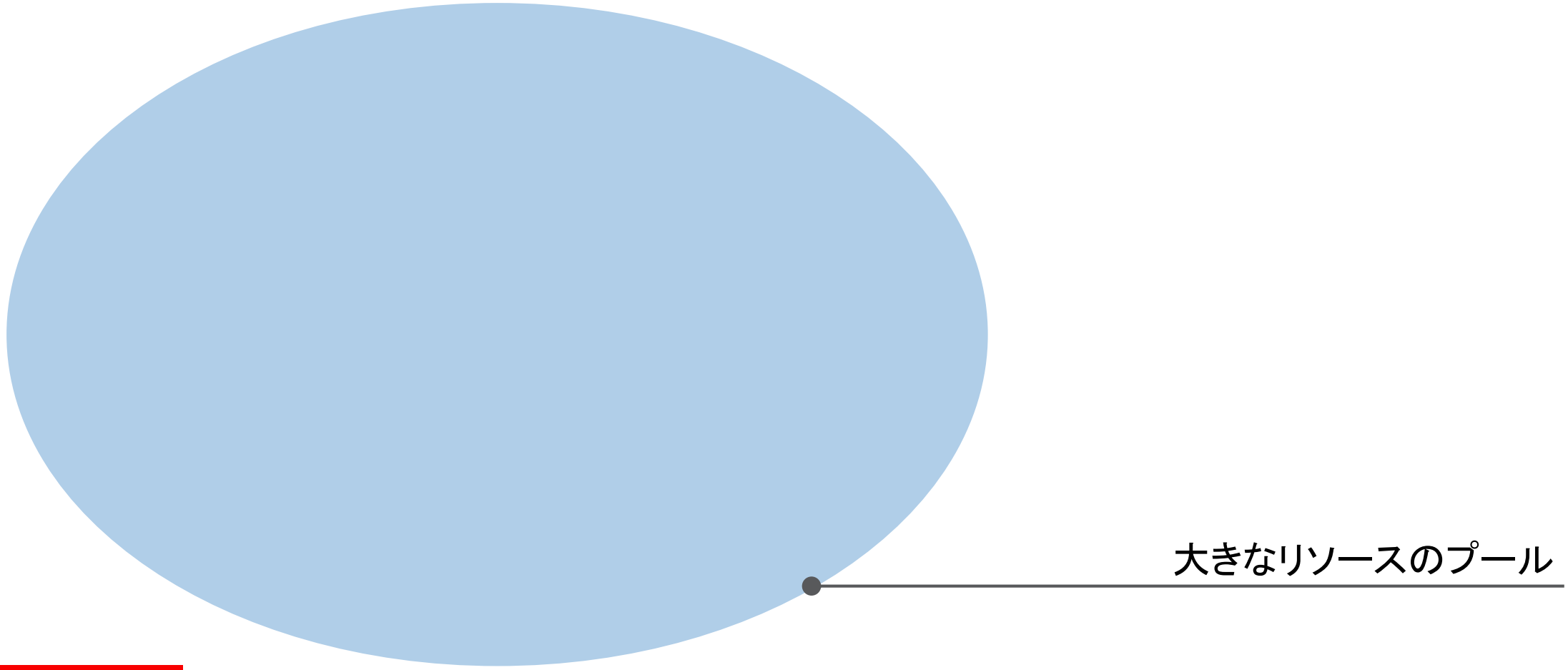
コンテナ・オーケストレーター

大量のコンテナを管理・運用するプラットフォーム

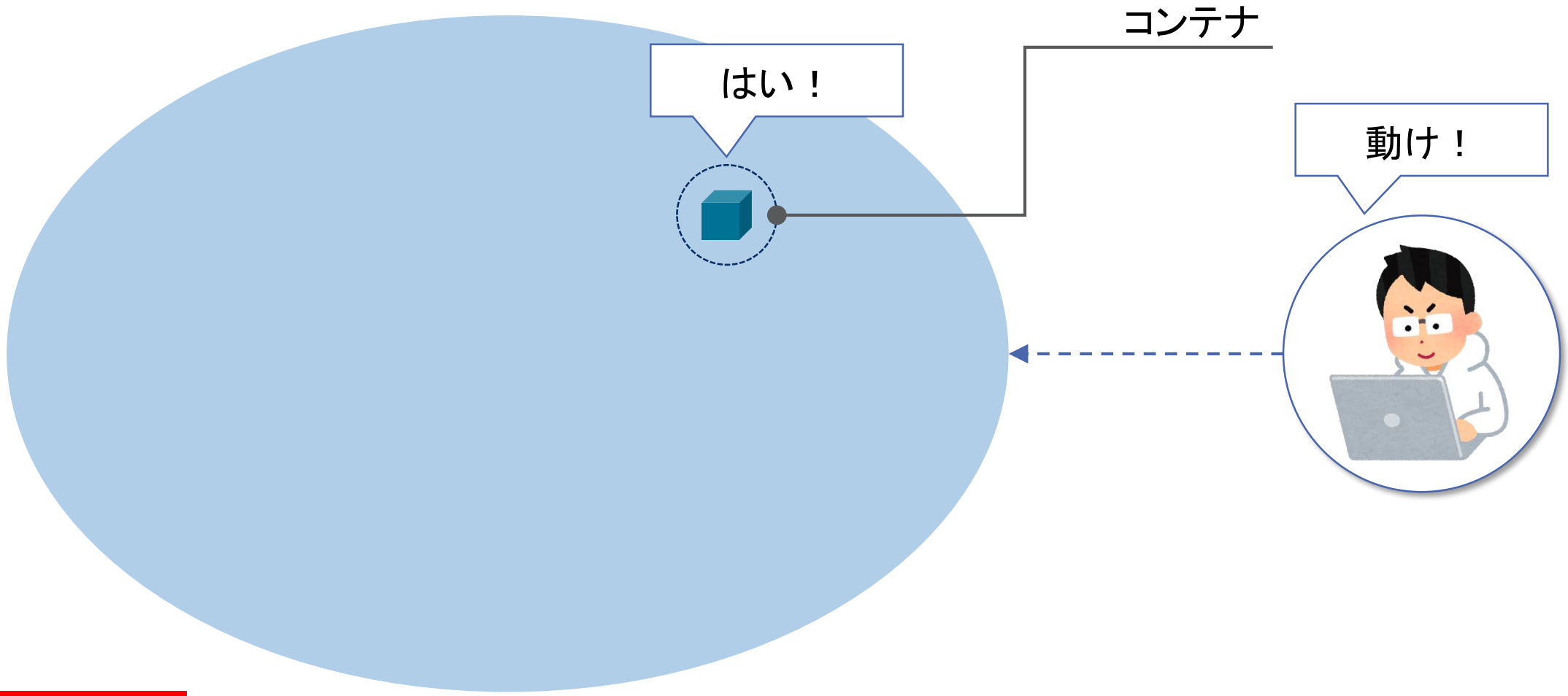
- 複数コンテナのデプロイ、スケーリング等を自動管理するプラットフォーム
  - 複数ホストにコンテナをデプロイ（HWを意識しない）
  - 手動／自動でスケーリング
  - 複数コンテナをまとめて制御
  - コンテナの死活監視 障害時のコンテナ再立ち上げ
  - クラスター内／外のネットワークアクセスの管理
- Kubernetesがデファクト・スタンダードになりつつある

kubernetes

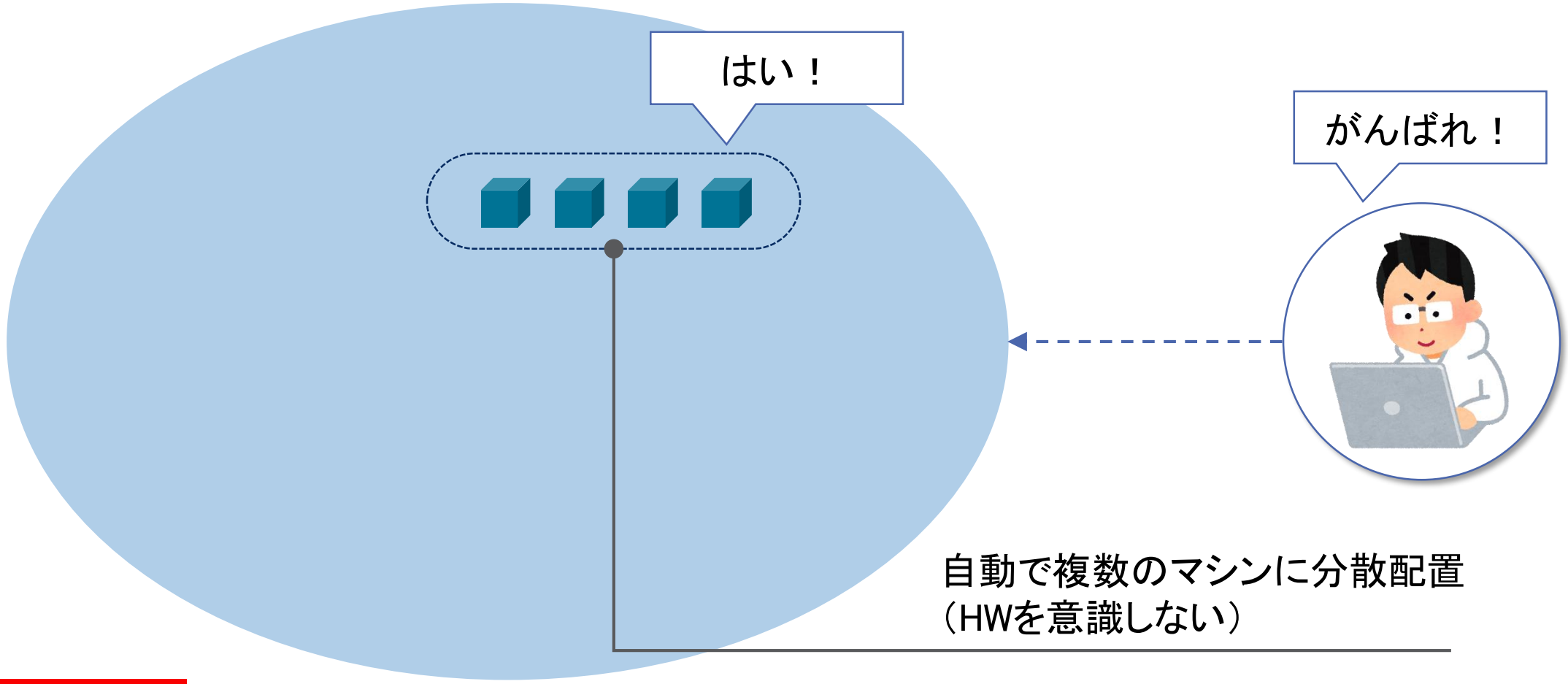
コンテナオーケストレーションが実現すること



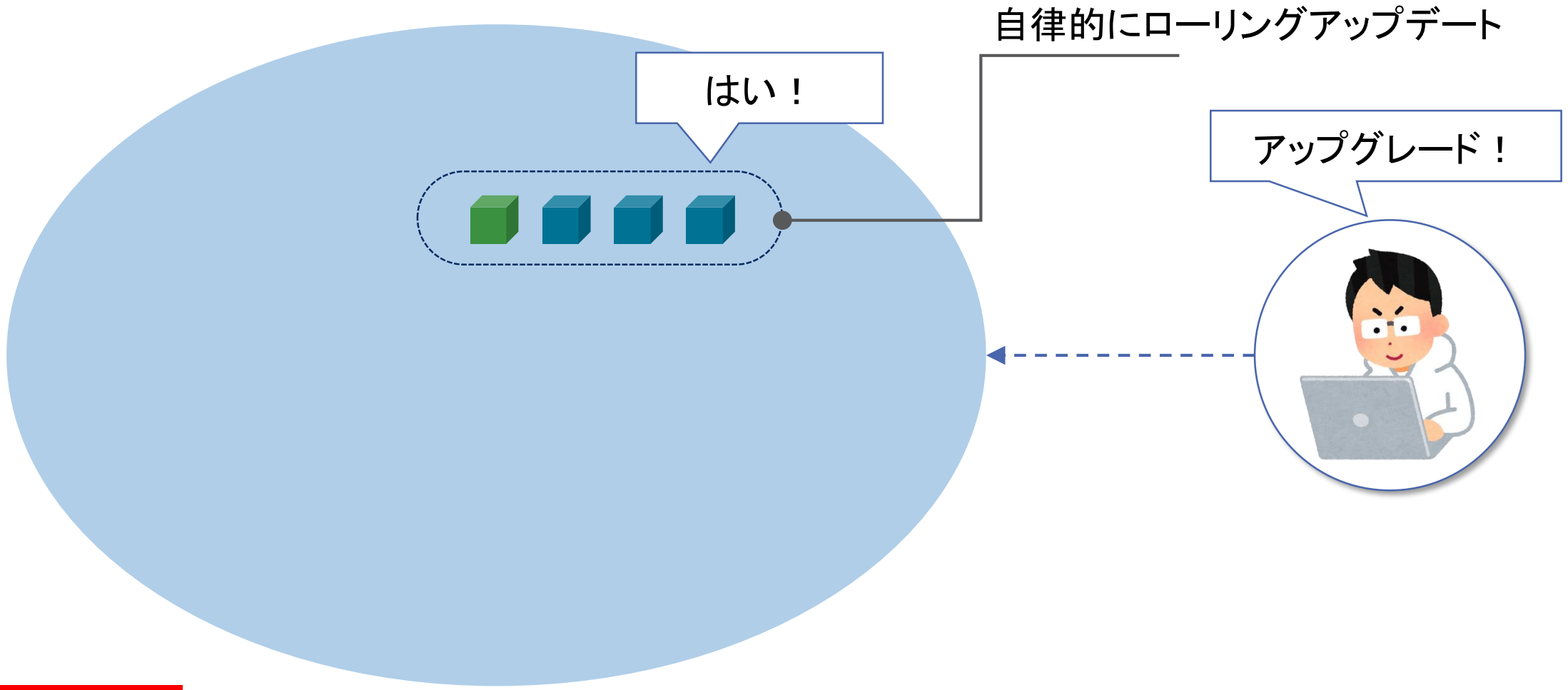
コンテナオーケストレーションが実現すること



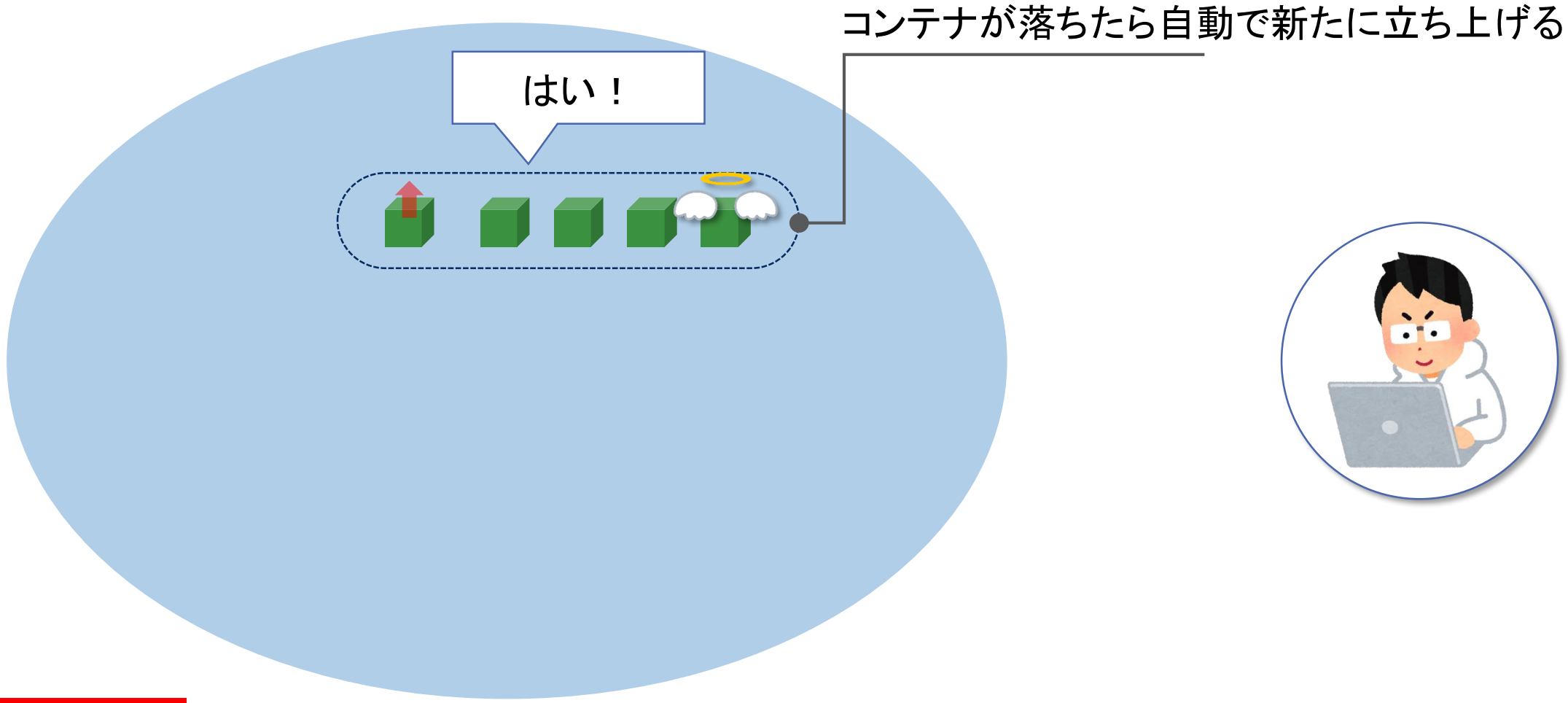
コンテナオーケストレーションが実現すること



コンテナオーケストレーションが実現すること



コンテナオーケストレーションが実現すること

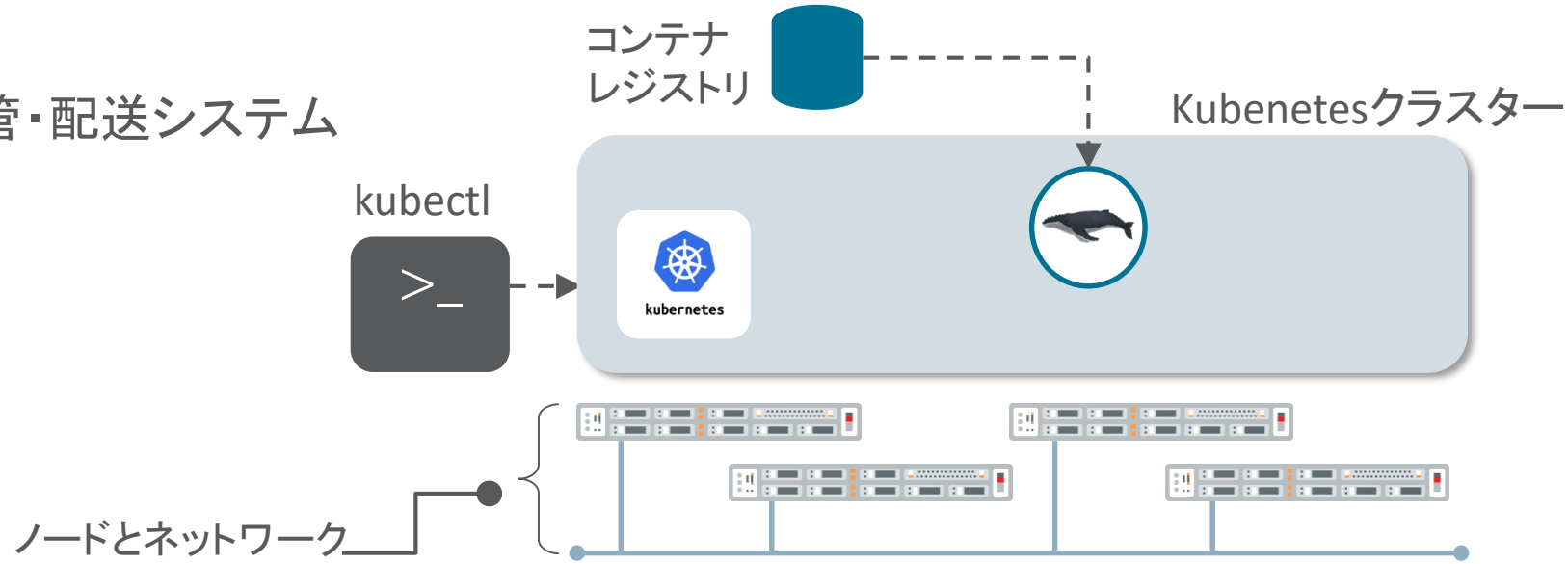


なんだかすごそうだ！

よし、
さっそくやってみよう！

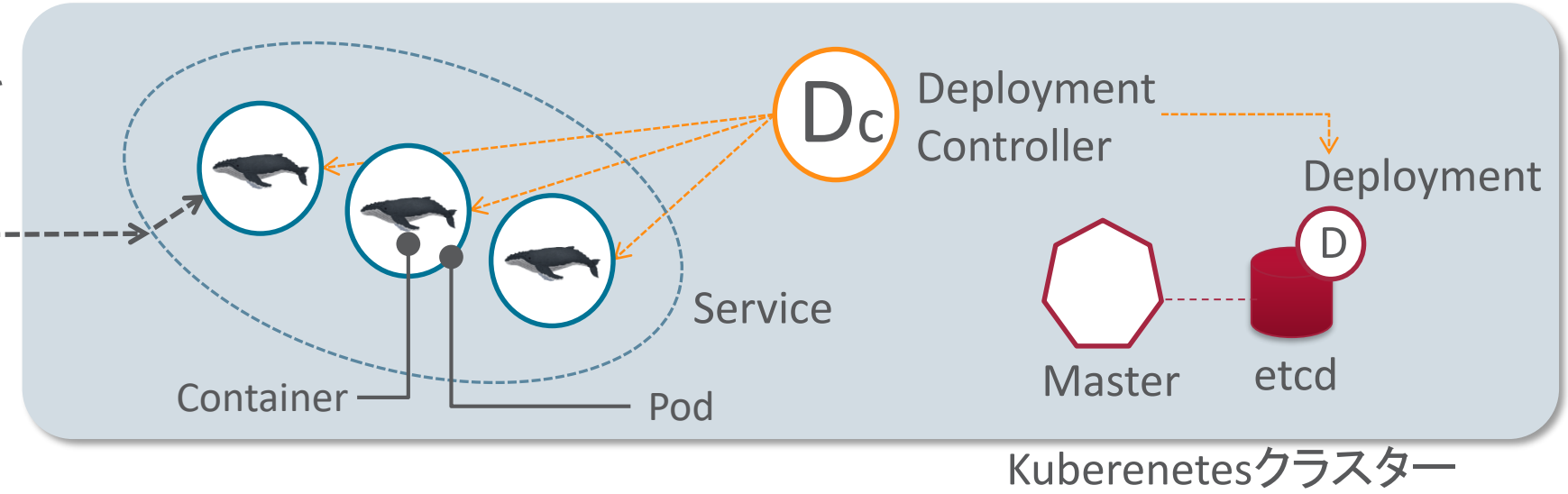
Kubernetes利用環境の主要な構成要素

- 大きく分けて、Kubernetesクラスター、kubectl、コンテナレジストリで構成
  - kubectl
    - Kubernetesクラスターの管理操作を行うためのコマンドラインツール
  - Kubernetesクラスター
    - クラスター本体、実際のワークロードが可動する場所
  - コンテナレジストリ
    - コンテナイメージの保管・配送システム



Kubernetes上でコンテナが動作する様子

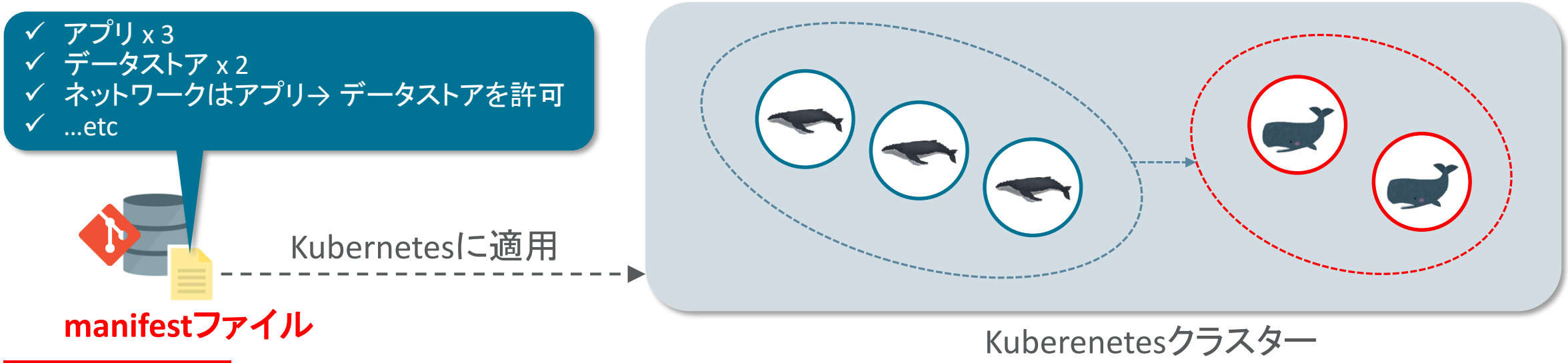
- Kubernetesでアプリケーションを動かすときの最も基本的な構成
  - Pod / Container
    - 複数のコンテナを束ねる単位
  - Deployment / Deployment Controller
    - Podの状態（レプリカ数など）を定義するデータ/制御プロセス
  - Service
    - クラスタ内外の通信をルーティング



Kubernetesは宣言的オペレーション

最終的にどうあるべきかということを宣言的に記述

- クラスターに配備する要素を全てマニフェスト・ファイル（＝コード）に記述
  - 環境構成の変更内容をバージョン管理し、追跡可能にする
  - 運用オペレーションはコードの変更によって実施し、作業を簡素化する



マニフェスト・ファイルの例

- yaml形式で記述するのが主流

- 右例では以下のような構成を定義
  - 3つのPodで冗長化
  - コンテナイメージは、
    cotoc/cowweb:v1.0
  - ポート8080を公開してアプリに流す
    …

```
1   apiVersion: apps/v1
2   kind: Deployment
3   metadata:
4     name: cowweb
5   spec:
6     replicas: 3
7     selector:
8       matchLabels:
9         app: cowweb
10    strategy:
11      type: Recreate
12    template:
13      metadata:
14        labels:
15          app: cowweb
16          version: v1.0
17      spec:
18        containers:
19        - name: cowweb
20          image: cotoc/cowweb:v1.0
21          ports:
22          - name: api
23            containerPort: 8080
24  …（以下略）…
```

マニフェスト・ファイルの運用

本番環境で運用するためには様々な考慮が必要

Agenda

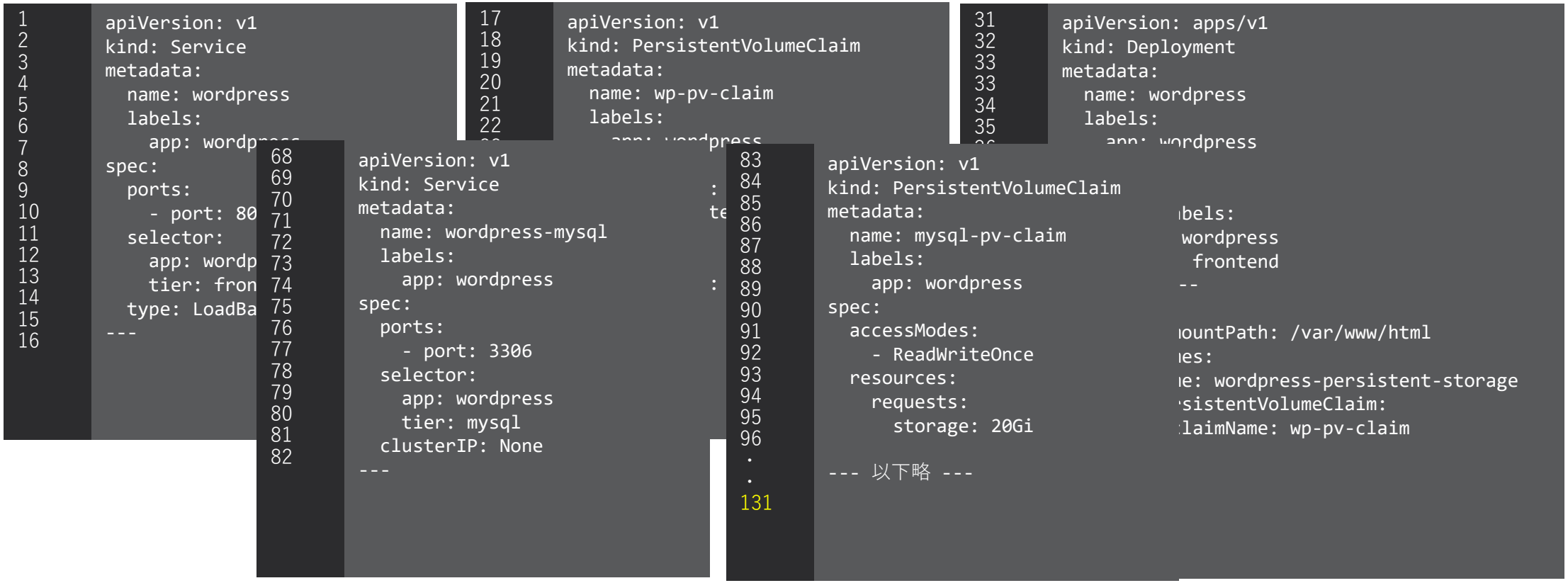
- コンテナのライフサイクル（起動時）
- コンテナのライフサイクル（停止時）
- ヘルスチェック
- メンテナンスとアップデート
- スケジューリング
- リソースの割り当てと基準
- カーネルパラメータのチューニング
- インターネットからのアクセス制御

https://speakerdeck.com/masayaaoyama/jkd1812-prd-manifests

Kubernetes完全ガイド impress top gearシリーズ

@amsy810

単純なアプリケーションでも3桁・・・



なんだか大変そうだ…

通称：YAMLの壁

でも、その先に待つのは！

クラスター内/外のネットワーク
アクセスの管理

複数ホストにコンテナをデプロイ
（HWを意識しない）

手動／自動でスケーリング

障害時のコンテナ再立ち上げ

複数コンテナをまとめて制御

kubernetes

コンテナの死活監視

Cloud Nativeプロダクトと組み合わせて最高のDevOpsを！

DevOpsを実現するためのCloud Nativeプロダクト群

- コンテナのCI/CD
  - 自動でビルド,テスト,デプロイを実施
- Wercker
  - 2012年オランダ発、現在はOracleが提供しているCIツール
- Spinnaker
  - Netflixが開発したCDツール
    - カナリーリリースの自動解析など

- 運用・監視
  - k8sクラスターやPodの監視
- Prometheus
  - システムのモニタリング・アラートのためのソフトウェア
  - K8sとの連携機能あり
- Grafana
  - メトリックの可視化ツール
    - 豊富なグラフテンプレート

スケーラブルなアーキテクチャに不可欠なインフラ

- サービスメッシュ
  - マイクロサービス間に張りめぐされたメッシュ状のトラフィックや経路の制御
- Istio
  - Pod間の通信にプロキシ（Envoy）を挟むことでトラフィックのモニタリングやコントロールを行う
  - アプリの変更なし

- 分散データベース
  - k8s上でスケーラブルな分散データベース
- Vitess
  - シャーディング技術を用いることでMySQLをスケーラブルに扱うことができるクラスタリングシステム

本日お話した3つのこと

1. Kubernetes、なんだかすごそうだ！
   - Kubernetes登場の背景・メリット
2. Kubernetes、なんだか大変そうだ！
   - 最初にぶつかる壁
3. Kubernetesで広がる世界！

YAMLの壁を乗り越えて、KubernetesとCloud Nativeプロダクトを活用して最高の開発/プロダクション環境を実現しよう！

ORACLE

さいごにお知らせ

Containers on Oracle Cloud

OracleもKubernetesのサービスを提供しています

- コンテナベースのマイクロサービスおよびサーバーレス・アプリケーションを構築、配置、管理するためのフルマネージド・サービスを提供します

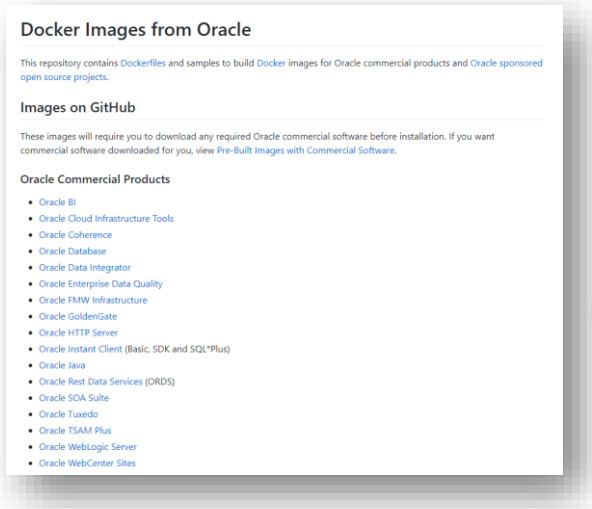
- **Oracle Container Engine for Kubernetes (OKE)**
  - マネージドKubernetesサービス

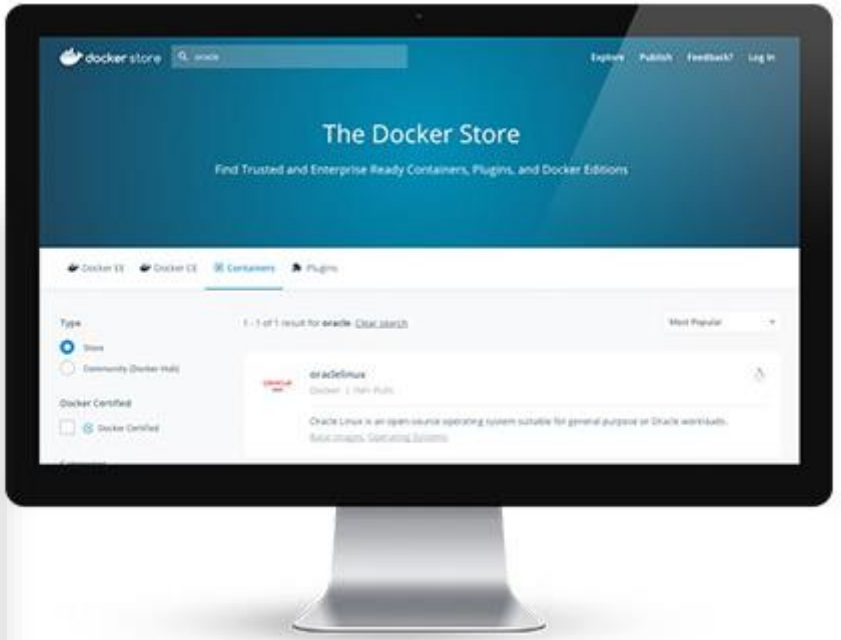
- **Oracle Cloud Infrastructure Registry (OCIR)**
  - マネージドなコンテナレジストリ

Oracle on Docker Store

Oracleの製品がDockerで簡単に動かせるようになりました

- Docker StoreでOracle Database、Oracle WebLogic Server、Oracle Coherence、Oracle Javaなどの主要製品を公式なDockerイメージとして利用できるようになりました
  - OTN開発者ライセンスで無償利用可能

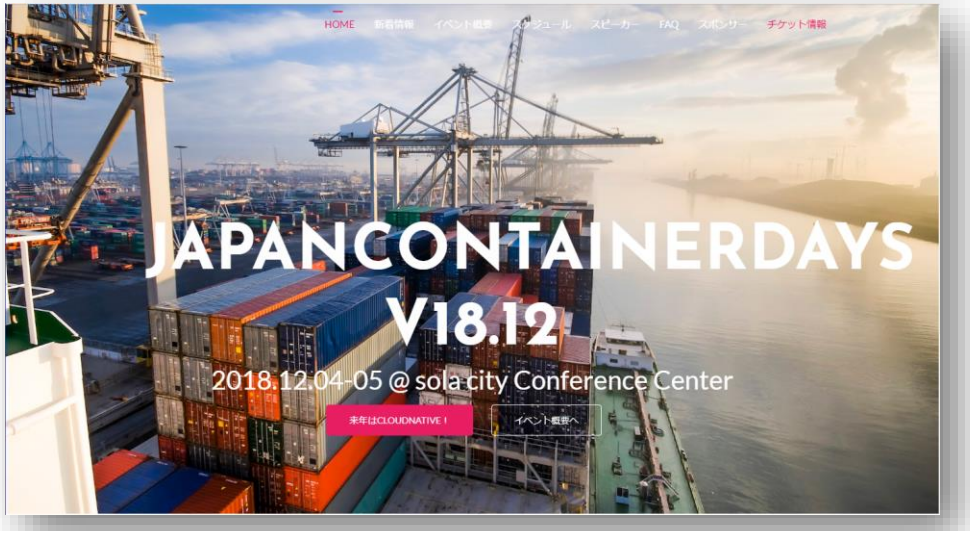




https://store.docker.com/publishers/oracle

Container/Cloud Nativeの世界はコミュニティが活発

もっと詳しくCloud Nativeな技術を共有し学び会える場



※ https://containerdays.jp

Community

Docker Tokyo	Kubernetes Meetup Tokyo	Rancher JP	Japan OpenStack User Group
Container SIG	Mesos User Group Tokyo	PaaS JP	Bluemix Users Group
Serverless Community (JP)	GitLab Tokyo	Cloud Native Meetup Tokyo	Cloud Native Developers JP
Cloud Native Deep Dive	NoOps Japan		

※来年はCloudNative Daysに名称が変更

cndjp - Cloud Native Developers JP

- **Cloud NativeなOSSスタック**を取り上げる勉強会シリーズ、およびコミュニティ
- OSS中心（最近はOSSとも限らない傾向）
- 楽しく学ぶ、深く学ぶ

Oracle Cloud Hangout Cafe（OchaCafe）

カフェで一息つきながらしっかり学べる、Cloud Nativeな勉強会シリーズ

	テーマ	キーワード
12/20	コンテナを使ったCI/CD	Wercker,Kubernetes,Spinnaker
1月開催予定	Microservicesの運用・管理	Grafana,Prometheus,istio,
2月開催予定	Microservicesな Javaアプリケーション	Helidon,MicroProfile,GraalVM
3月開催予定	明日から使える Enterprise Blockchain	Hyperledger Fabric
4月開催予定	避けては通れない 認証/認可	OAuth 2.0,OpenID Connect
5月開催予定	OpenAPIのエコシステム	OpenAPI

※発表順序・テーマは変更する可能性があります。

#ochacafe

https://ochacafe.connpass.com/event/108009/

Safe Harbor Statement

The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.


ORACLE®