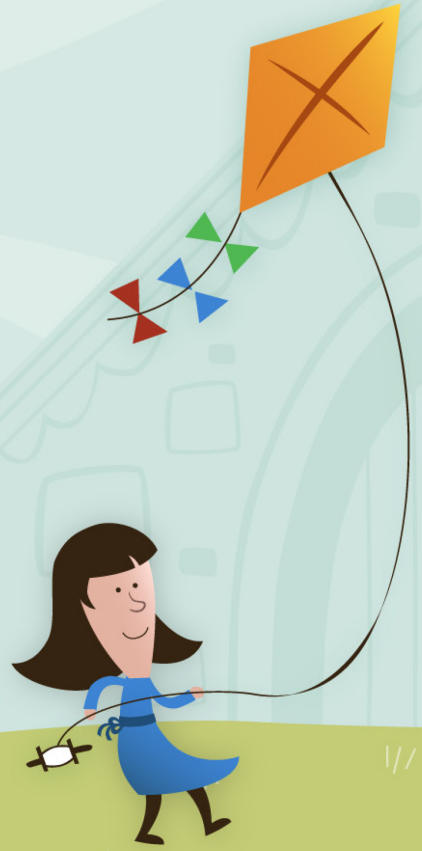
Internet Week 2018

Kubernetesの基礎



2018年11月28日
日本アイ・ビー・エム株式会社
クラウド事業本部　高良 真穂

発音／略称／Logo

綴り Kubernetes

発音 koo-ber-net-ees

略称 K8s

Kubernetes

12345678

クーベネティスのロゴ

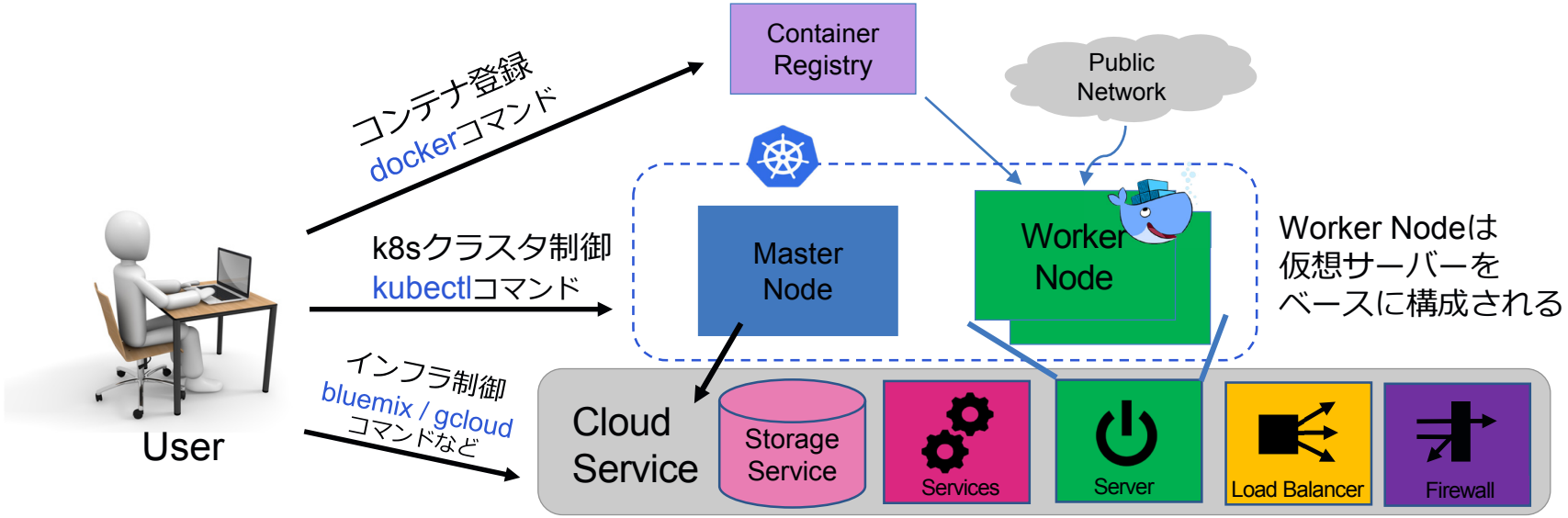
K8sは一言で何ができる？

K8sは、コンテナのアプリ運用のためのOSS

1. コンテナの組み合わせ利用
2. スケールアウト
3. ロールアウト＆ロールバック
4. 永続ストレージ利用
5. 自己修復（可用性）
6. クラスタの分割利用
7. 監視＆ログ分析

k8s共通とベンダー個別の階層構造

コンテナ層（共通）　docker コンテナのビルド、レジストリへの登録＆取出し
オーケストレータ層（共通）　kubectl ロールアウト、負荷分散、リソース制限、コンテナ間連携
インフラ層（ベンダー個別）　クラスタ構成、レジストリ操作、FW/LB設定、ストレージ操作
IKS bx cs, ICP bx pr, GKE gcloud container,



コマンド・カットの階層構造View

- K8sは、ベンダー個別のインフラ機能を共通化 (抽象化) して、共通のオペレーション環境を提供する。
- K8sは、HA構成、負荷分散、監視、オートスケールなどの機能を提供

利用者

アプリのビルド
開発言語固有のコマンド

コンテナのビルド
docker build …

デプロイ&負荷分散
kubectl create –f xxx.yaml

クラスタ環境の構成
gcloud container create…

アプリケーション層

コンテナ層
(Docker)

オーケストレータ層
(Kubernetes)

インフラ層
（クラウドVSI, オンプレ物理サーバーなど）

Java, Node.js, Python, Ruby, PHP
C, Go などのコンパイル、リンクなどビルド

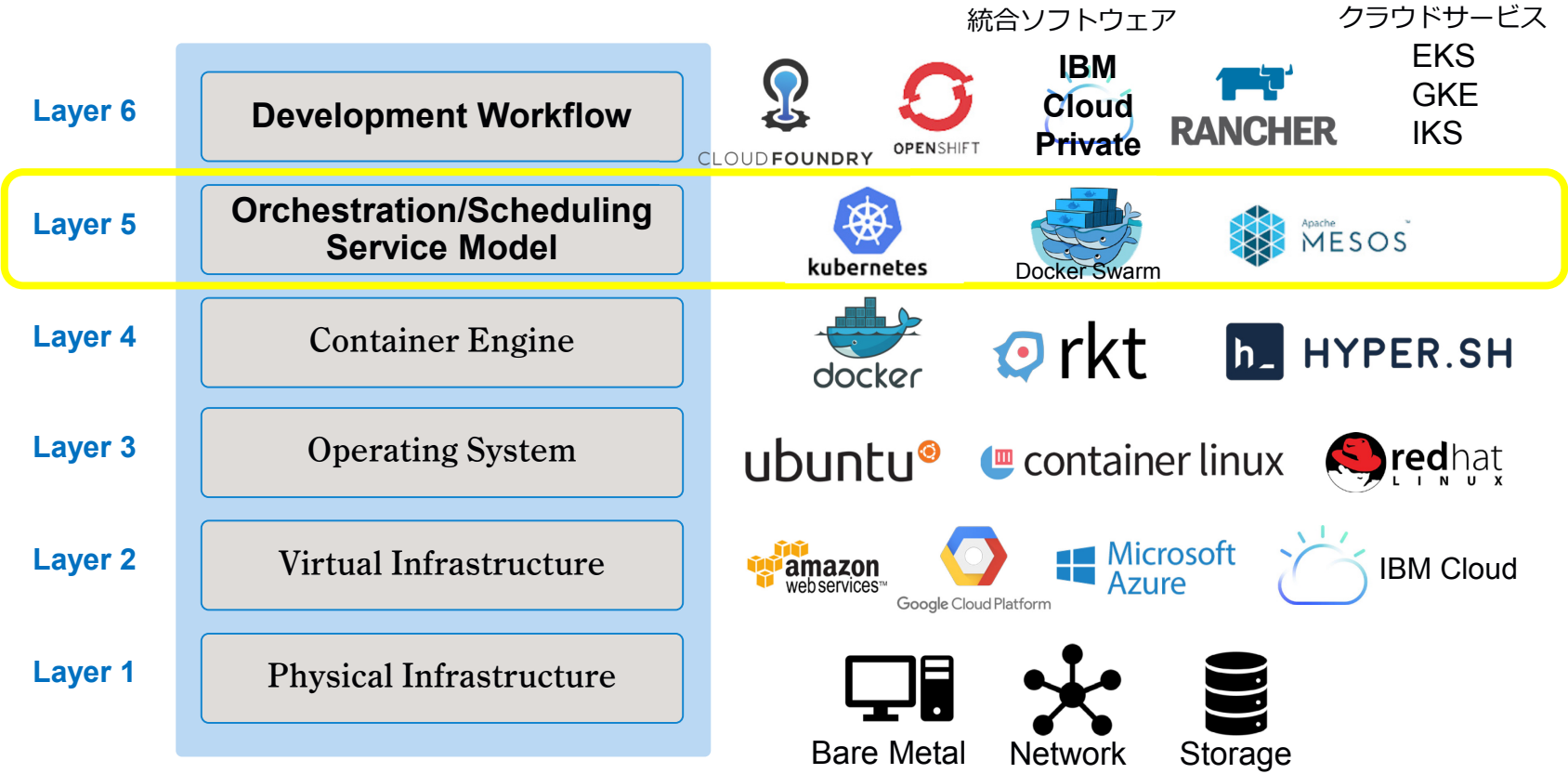
ビルドしたアプリのコンテナ化、リポジトリ登録
WAS,DB2,MQなどのコンテナ化ミドルウェア利用

各社クラウド、ソフトウェア製品
ICCS, ICP, GKE, AKE, RedShiftなどの
k8s kubectl コマンドを利用

各社クラウドのIaaS機能
オンプレのOpenStack や Vmware 利用
物理サーバー利用

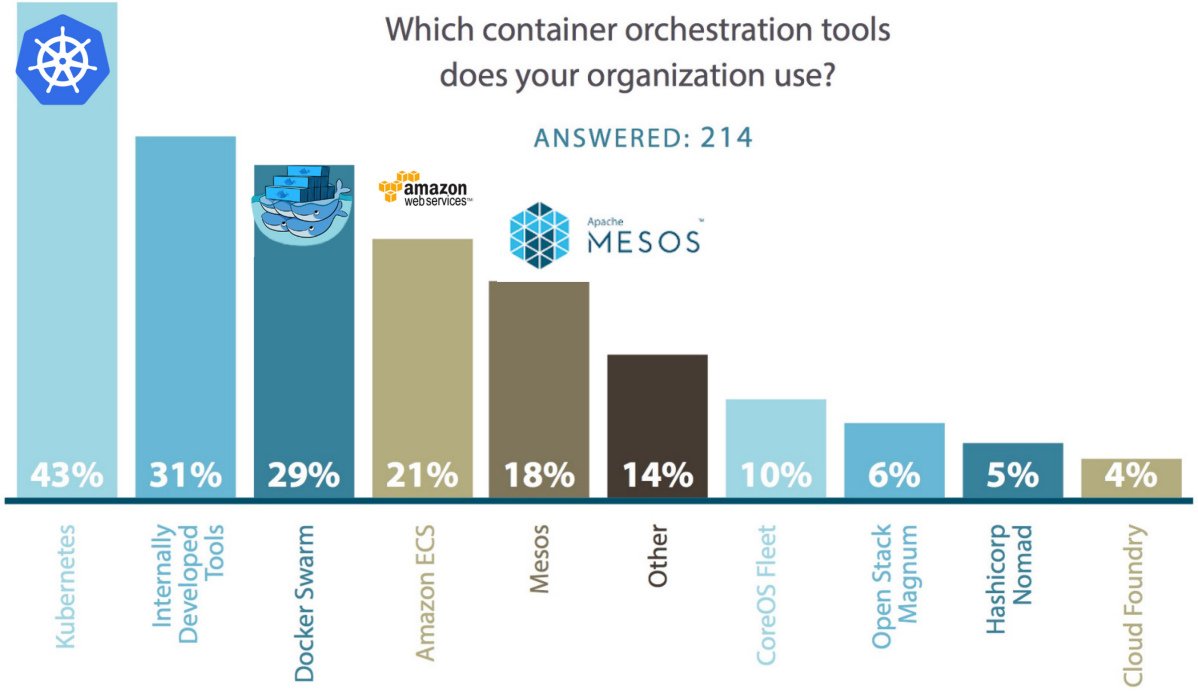
階層の中でのKubernetesの概念的な位置づけ

- DockerはSwarmとKubernetesの統合を発表　**DockerCon EU 2017**



コンテナのオーケストレーション・ツールの人気

- この調査結果ではKubernetesが首位



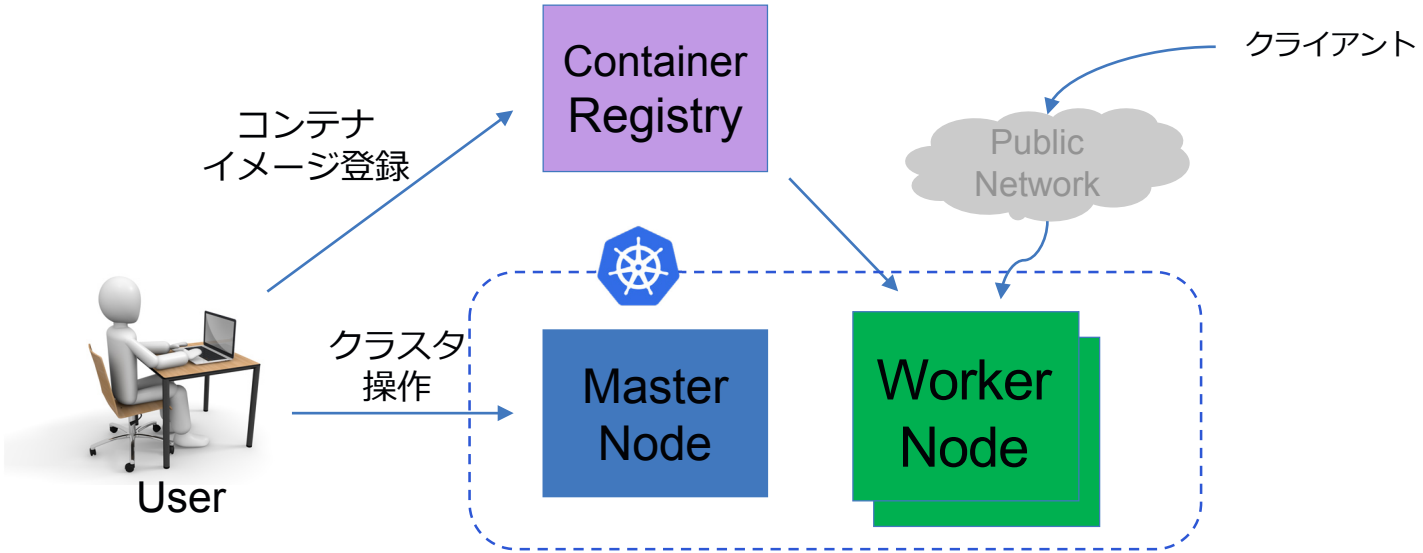
Source: devops.com

ココを抑えれば
もう怖くない
Kubernetesの要点

k8sの三大構成要素

K8sの３大構成要素は３つ

- **マスターノード**　　k8sクラスタの制御　パブリック・クラウドの場合はマネージド
- **ワーカーノード**　　アプリのコンテナ稼働環境、ノード数可変
- **コンテナ・レジストリ**　イメージの保管場所

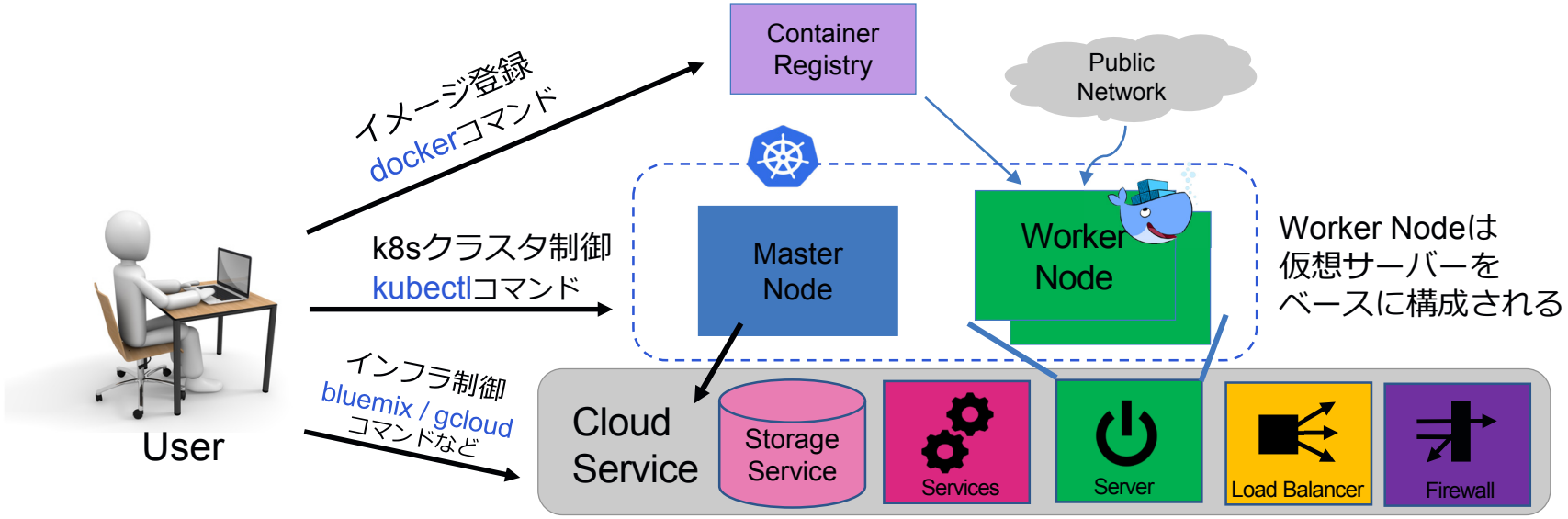


k8s共通とベンダー個別の階層構造

コンテナ層（共通） docker コンテナのイメージをビルド、レジストリへの登録＆取出し

オーケストレータ層 (共通) kubectl ロールアウト、負荷分散、リソース制限、コンテナ間連携

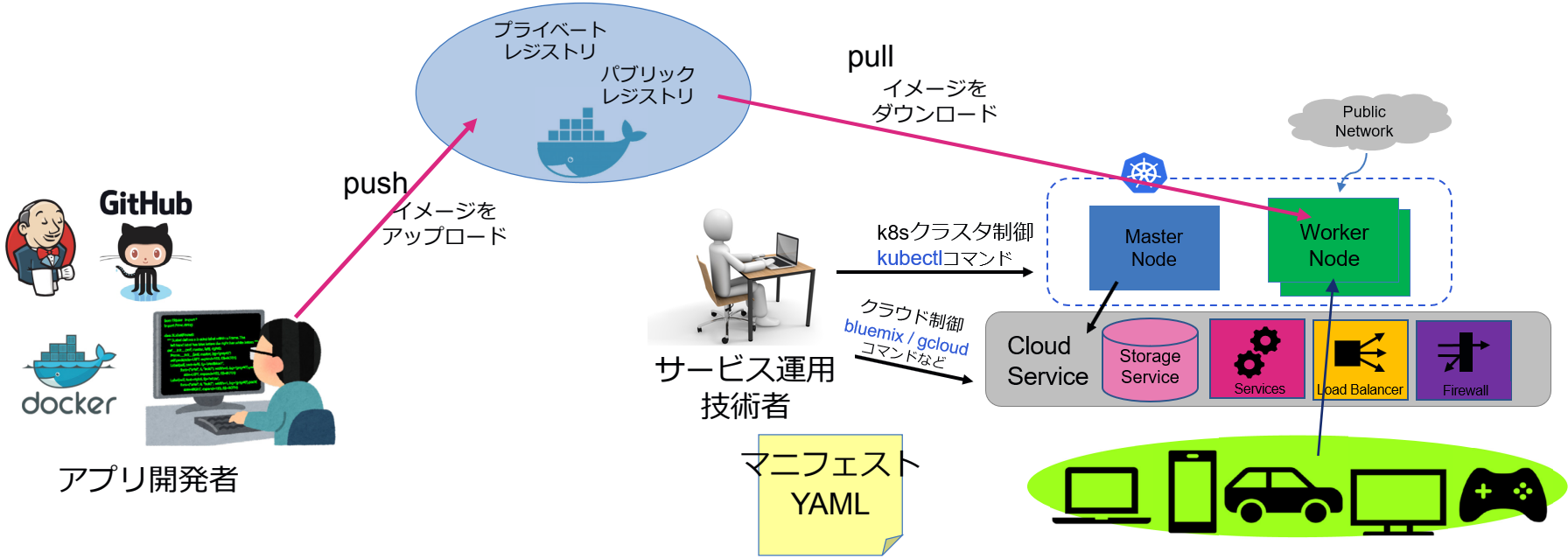
インフラ層（ベンダー個別） クラスタ構成、レジストリ操作、FW/LB設定、ストレージ操作



アプリのデプロイ

サービスを開始するまでの作業は３ステップ

1. **アプリのコードを開発してDockerコンテナ化**
2. **Dockerイメージをレジストリへ登録**
3. **マニフェスト(YAML)を利用してデプロイ**



YAMLの定義は？ 基本これだけ

Nginxのマニフェスト

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: ms-apl-ws
spec:
  replicas: 3
  selector:
    matchLabels:
      app: ms-apl-x
  template:
    metadata:
      labels:
        app: ms-apl-x
    spec:
      containers:
      - name: application-x
        image: appl-x:1.18
        ports:
        - containerPort: 80
```

コンテナの可用性を制御

起動数

コンテナのイメージ名

サービス ロードバランサー

```
apiVersion: v1
kind: Service
metadata:
  name: apl-x
spec:
  selector:
    app: ms-apl-x
  ports:
  - protocol: TCP
    port: 80
```

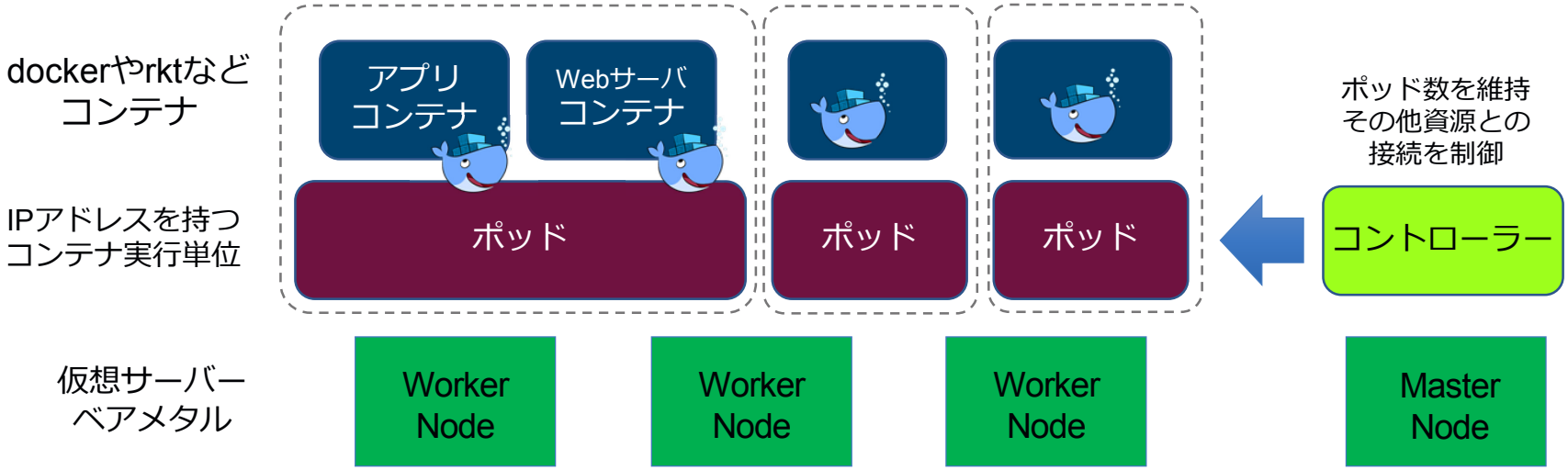
APL-Xでアクセス可能

実行はこれだけ

```
$ kubectl apply –f manifest.yml
```

ポッドはコンテナを内包して実行

- コンテナは、ポッドによって起動される
- ポッドは、内部IPアドレスを持つ、一つの仮想マシンの様な存在、ポッド単位で起動と破棄され、永続データは保持できない一時的な存在
- コントローラーは、ポッドの起動停止、回復などの制御を受け持つ



k8sうんちく

- Podは、なんの短縮形？

  Pod is not “Point Of Delivery”

podとは

クイック再生　プレーヤー再生　ピン留め　単語を追加

主な意味　(エンドウなどの)さや、さや状のもの、(蚕の)まゆ、(イナゴの)卵袋、ポッド、(宇宙船の)離脱部、(アザラシ・クジラなどの)小群

音節　p.o.d.　発音記号・読み方　/ pɑ́d (米国英語), pˈɔd (英国英語) /

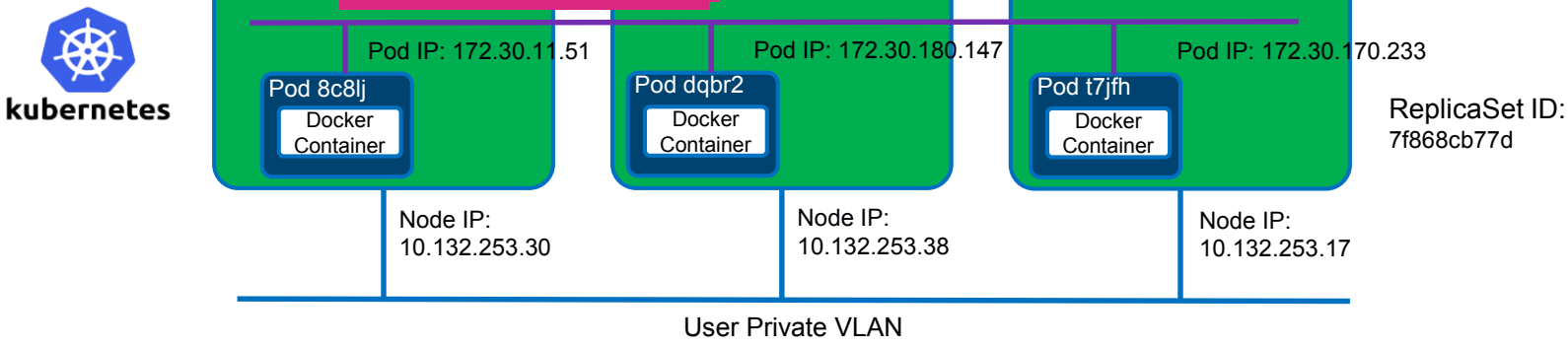
podの変形一覧	動詞： podding(現在分詞)　podded(過去形)　podded(過去分詞)　pods(三人称単数現在) 名詞： pods(複数形)
podのイディオムやフレーズ	in pod
	レベル：9　英検：1級以上の単語　学校レベル：大学院以上の水準　TOEICスコア：950点以上の単語

クラスタ内部ネットワーク

- ポッドはクラスタ・ネットワークに、ポッドが接続され、Worker Nodeの境界を越えて通信できるノード横断ネット
- しかし、ポッドのIPアドレスでは外側と通信はできない内部専用、ポッド作成ごとにIPアドレスを自動付与

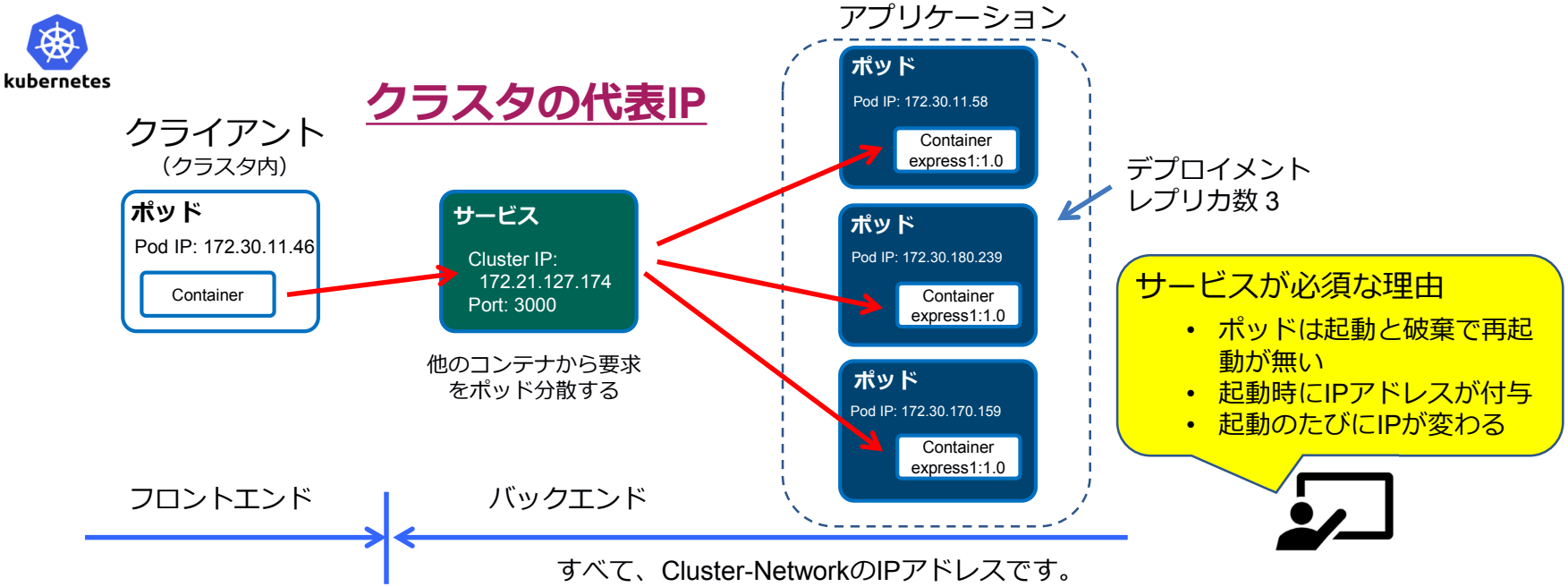
```
vagrant@vagrant-ubuntu-trusty-64:~$ kubectl get deployment
NAME          DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
express-app   3         3         3            3           2h
vagrant@vagrant-ubuntu-trusty-64:~$ kubectl get pod -o wide
NAME                           READY     STATUS    RESTARTS   AGE       IP              NODE
express-app-7f868cb77d-8c8lj   1/1       Running   0          2h        172.30.11.51    10.132.253.30
express-app-7f868cb77d-dqbr2   1/1       Running   0          2h        172.30.180.147  10.132.253.38
express-app-7f868cb77d-t7jfh   1/1       Running   0          2h        172.30.170.233  10.132.253.17
```

ポッドの
IPアドレス



内部ロードバランサー

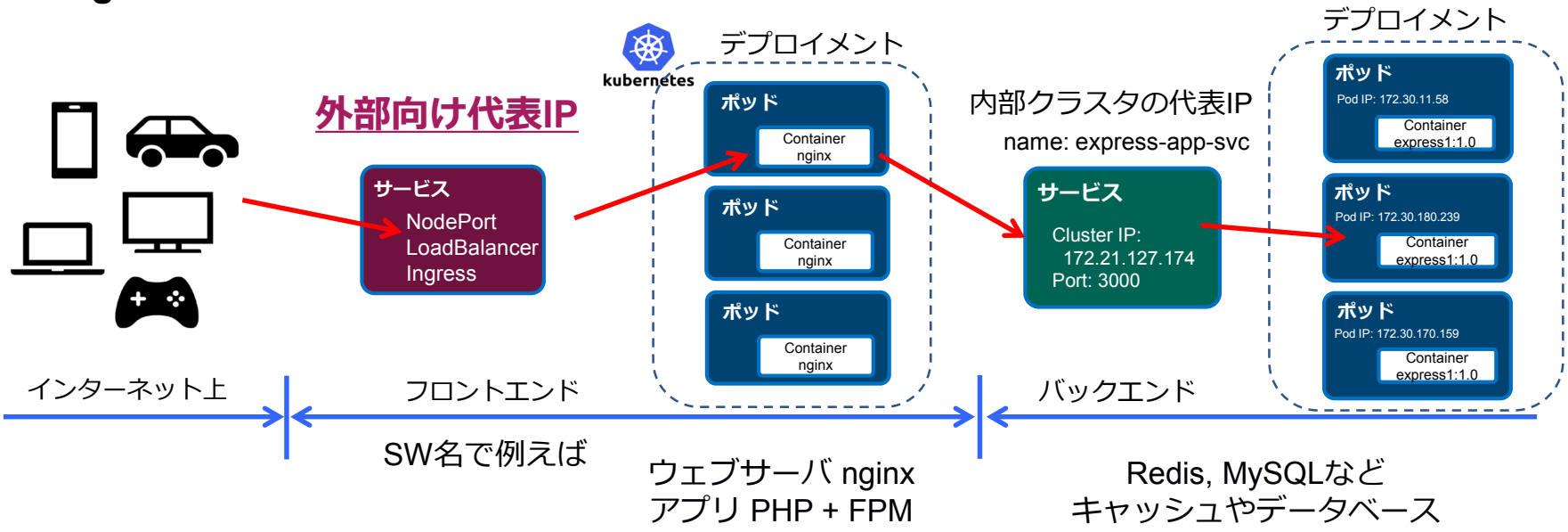
- k8sは、ポッドのクラスタの代表IPを作り要求を分配するサービスを提供
  - kubectl コマンドからYAMLを投入して”サービス”を定義する
  - サービス作成時に、内部DNSにサービス名を登録、クライアントからDNS名で参照可能
  - フロントエンドの接続先バックエンドは、YAMLの定義 “セレクタ”で指定



外部公開用ロードバランサー

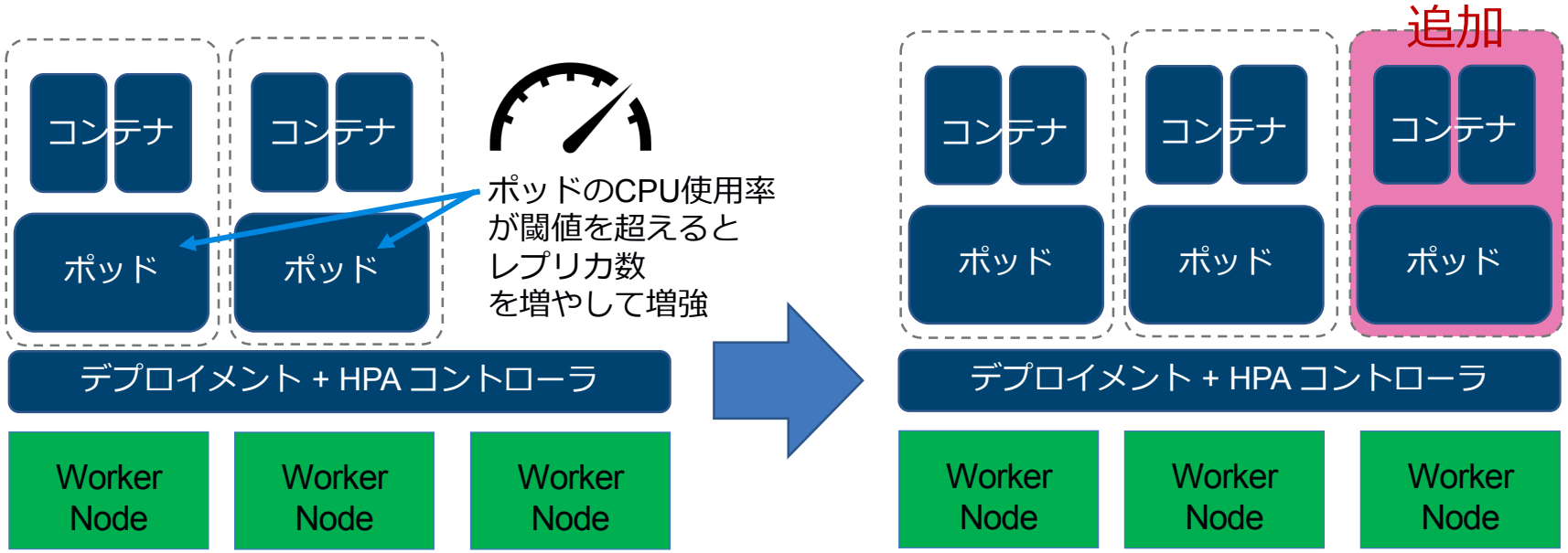
- **k8sクラスタから外へ公開するロードバランサのサービスを提供**

  - **NodePort** ノードのポート番号で公開、KubeProxyと連携して複数のポッドへアクセスを分配します
  - **LoadBalancer** クラウドプロバイダのロードバランサを使用して外部にサービスを公開
  - **Ingress** 各社クラウドの実装でロードバランサーとHTTPSなどの機能で公開



オートスケール

- CPUの使用率の閾値越えで、ポッド数を増加して処理能力を増強
- 反対に閑散な状態となれば、ポッド数を縮小
- ノードの増強は手動のケースありに注意、ノードは余裕をもって運用は必須

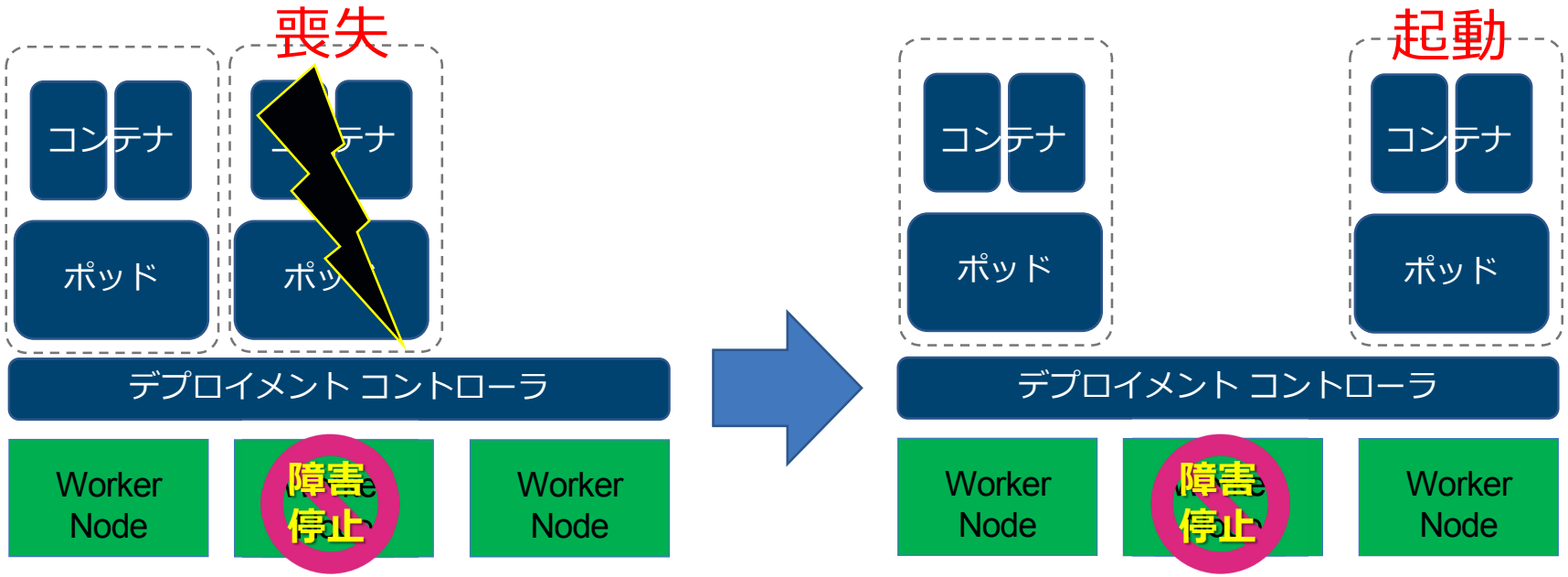


HPA : Horizontal Pod Autoscaler

自己回復

ノード障害に対して、サービスは無停止で継続可能

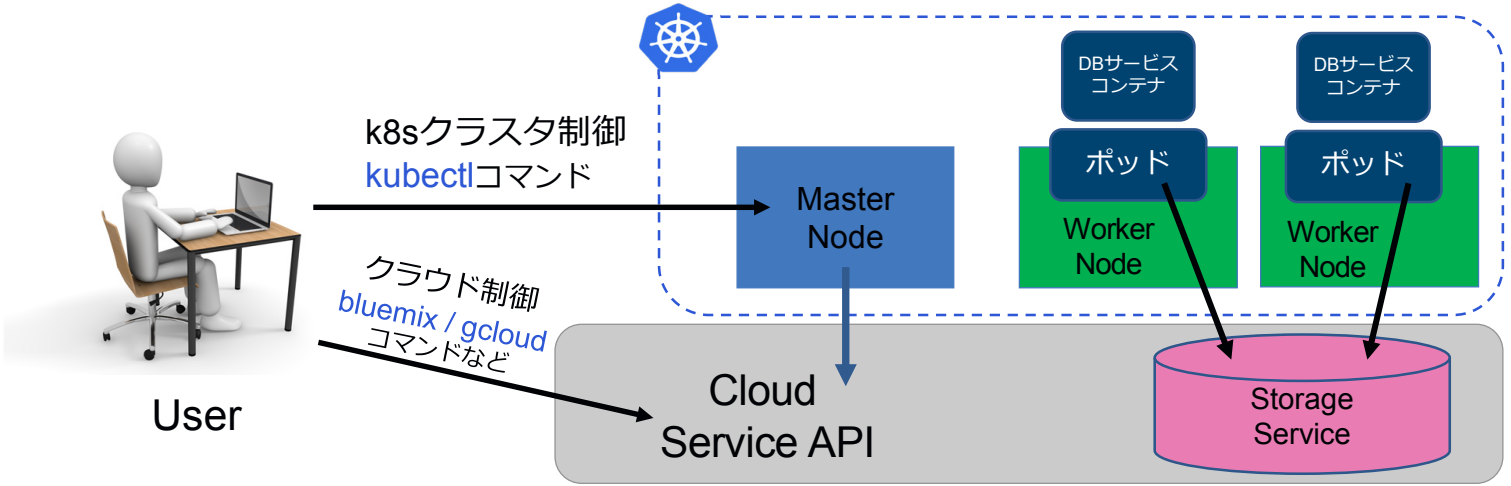
- **ワーカーノードの障害に対して、必要数のポッドを起動して処理能力を補う**
- **N+1構成の様に能力的に余裕もった設計が必要であるが、ポッドの再配置は自動**



永続ストレージの利用

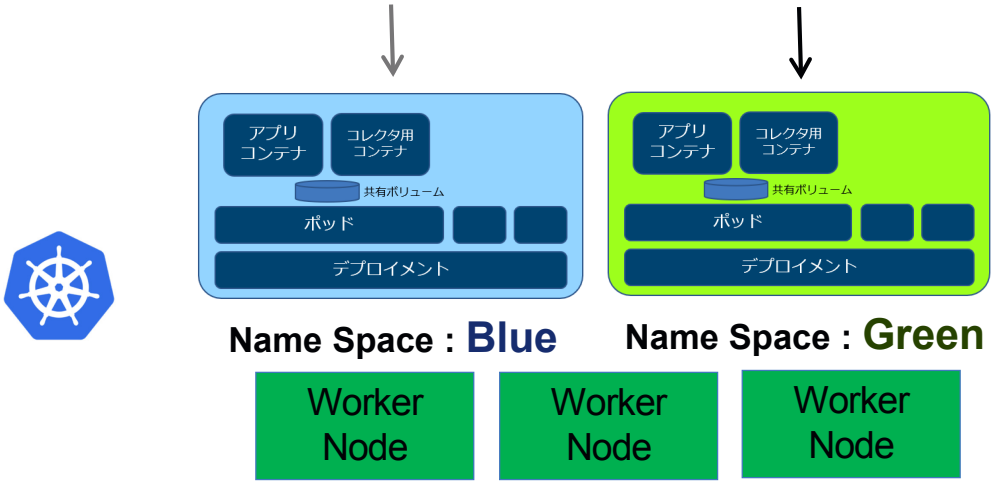
- K8sのコンテナ環境ではデータをストレージに永続的に保管できます。
  - クラウド・プロバイダのストレージ・サービスを利用できます。
  - YAMLファイルに、”kind: PersistentVolume” や “kind: PersistentVolumeClaim” とすることで、既存ボリュームをマッピングしたり、新規に作成するなどの永続ストレージの利用ができます。
  - クラウド・プロバイダや既存ストレージ系プロトコルのための複数のプラグインが提供されています。

参考　https://kubernetes.io/docs/concepts/storage/persistent-volumes/#types-of-persistent-volumes



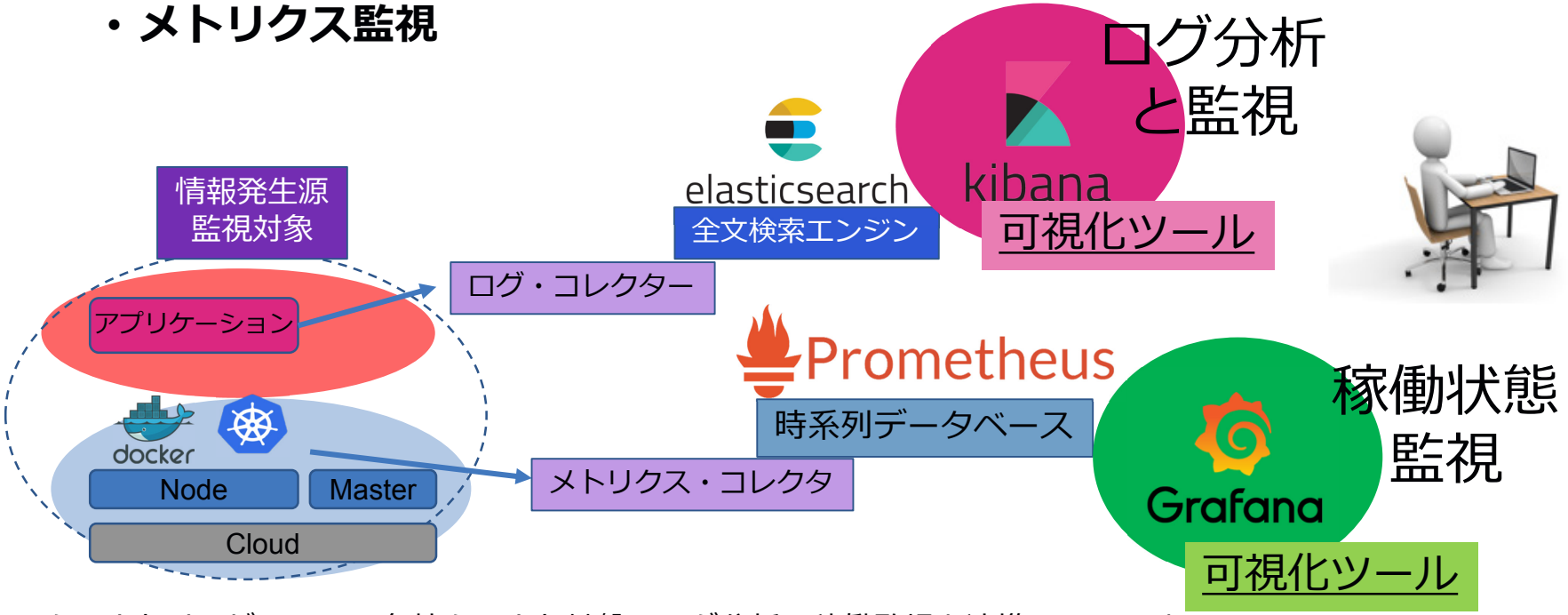
名前空間によるクラスタの仮想化

- 名前区間毎にCPUとメモリの利用制限を設定
- ネットワークポリシーを設定してアクセス制限
- RBACとサービスアカウントによるアクセス権管理



分散環境のモニタリングと洞察

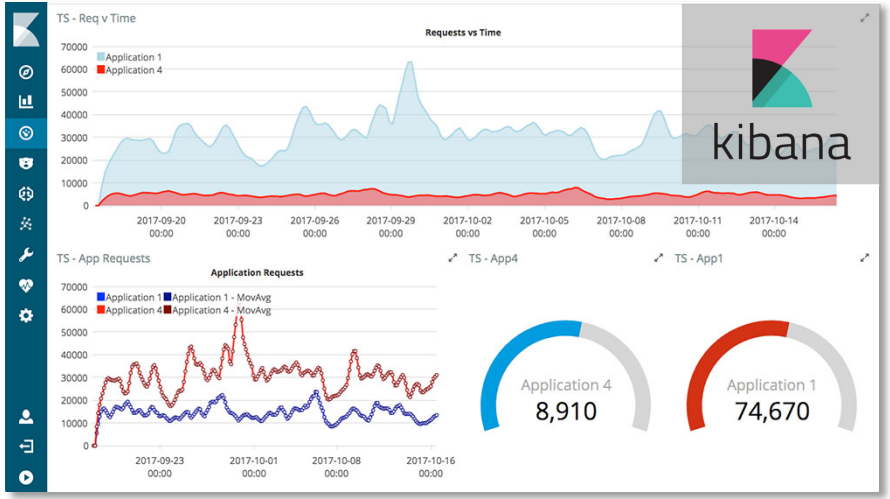
- **Kubernetesは２つの機能を組み込み済み**
  - **ログ分析**
  - **メトリクス監視**



クラウドサービスでは、各社クラウド基盤のログ分析＆稼働監視と連携しています。

ブラウザで閲覧する視覚化ツール

Kibana（ログ分析）



特徴

- Elasticsearchの視覚化ツール
- ログ分析など、対話的に操作しながらの発見に向く
- 豊かな表現形式に対応

Grafana（稼働分析）

特徴

- 時系列DB (time series database)の視覚化ツール
  influxDB, Prometheus, Graphiteなどの時系列データを視覚化
- 設定を保存して繰り返し利用するダッシュボードに向く
- シンプルな操作

ここで、ご紹介したのは
Kubernetesの一部の機能であり
説明者により選定されたものです。

メガクラウド各社 Kubernetes 対応状況

2017年はクラウド各社がk8s対応した年となった

- GCP
  - Google Kubernetes Engine (GKE)
    - 2014年6月 **オープンソース化**とGCPのサポートを発表
    - KubernetesをCNCFへ移管
- IBM
  - クラウド IBM Cloud Kubernetes Service (IKS)
    - 2017年3月23日 IBM Cloud （当時 Bluemix）で提供開始を発表
  - ソフトウェア IBM Cloud Private (ICP) v2.1 提供
    - 2017年10月24日 発表　オンプレのサーバーに導入できるソフトウェア製品
    - 無料で利用できるIBM Cloud Private Community Editionのダウンロード提供
- Azure
  - Azure Container Service (AKS)
    - 2017年10月24日 発表
- AWS
  - Amazon Elastic Container Service for Kubernetes (Amazon EKS)
    - 2017年11月29日発表

CLOUD NATIVE COMPUTING FOUNDATION

KCSP認定制度
による互換性確保

K8sでクラウド・レースの展開に変化があるかもしれない

まとめ

- Kubernetesはコンテナの運用基盤
- オンプレ＆クラウドで共通のオペレーションで運用できる
- 必要なインフラ機能が提供され、高効率な運用を実現
- 主要クラウドベンダー、ソフトウェア企業が賛同

Kubernetes
ハンズオンへ