テーマ１：コンテナの体験
座学編

レッドハット株式会社
テクニカルセールス本部

Red Hat

コンテナ開発体験のアジェンダ

コンテナの作成からCI/CDまでの一連の流れがOpenShift上で体験できるカリキュラムとなっております。
各時間枠は、ハンズオンの進捗により多少前後します。

13:00-13:10 (10min)	オープニング、ハンズオン概要のご紹介、端末環境などのご説明 出席者自己紹介（業務のご担当内容など）
13:10-14:00 (50min)	テーマ１：コンテナの開発体験 座学：コンテナをデプロイするためのk8s/OpenShiftの機能・知っておくべきコンポーネントのご説明 ハンズオン：OpenShift上にて、アプリケーションのデプロイとコンテナのスケールを体験
14:10-14:20 (10min)	休憩
14:20-15:40 (80min)	テーマ２：CIの実践 座学：コンテナのビルド、CIの必要性とCIパイプライン作成の流れなどのご説明 ハンズオン：CIパイプライン(tekton)の作成とパイプラインへの静的診断やイメージ脆弱性診断の組込みを体験
15:40-15:50 (10min)	休憩
15:50-17:10 (80min)	テーマ３：CDの実践 座学：CIとCDの違い、Gitで継続的デリバリーを実現するGitOpsなどのご説明 ハンズオン：ArgCDを使った開発環境、本番環境へのCD(自動デプロイ)の体験
17:10-17:40 (30min)	PTPワークショップのご紹介 まとめ・クロージング ご出席者のご意見（今日学んだこと・業務に取り入れるにあたっての課題など）

コンテナ/K8s基礎が基本

ここで、コンテナのデプロイを体験

ここで、コンテナビルドとCIをご説明と体験

ここで、CDをご説明と体験

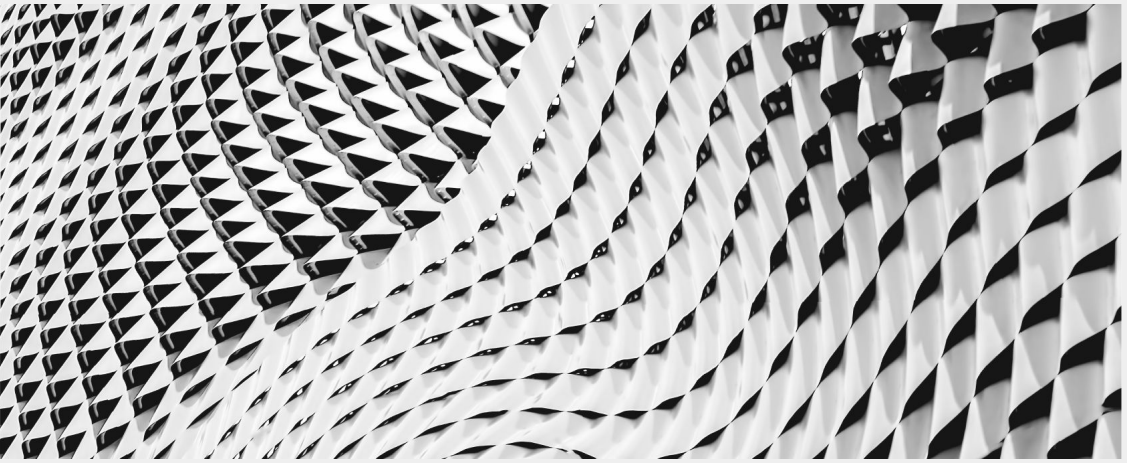
テーマ１：コンテナの体験のアジェンダ

1 **座学:コンテナ、OpenShift/Kubernetesについて**

2 **ハンズオン:コンテナのデプロイ体験**

Red Hat

1. 座学
コンテナ、OpenShift/Kubernetes について



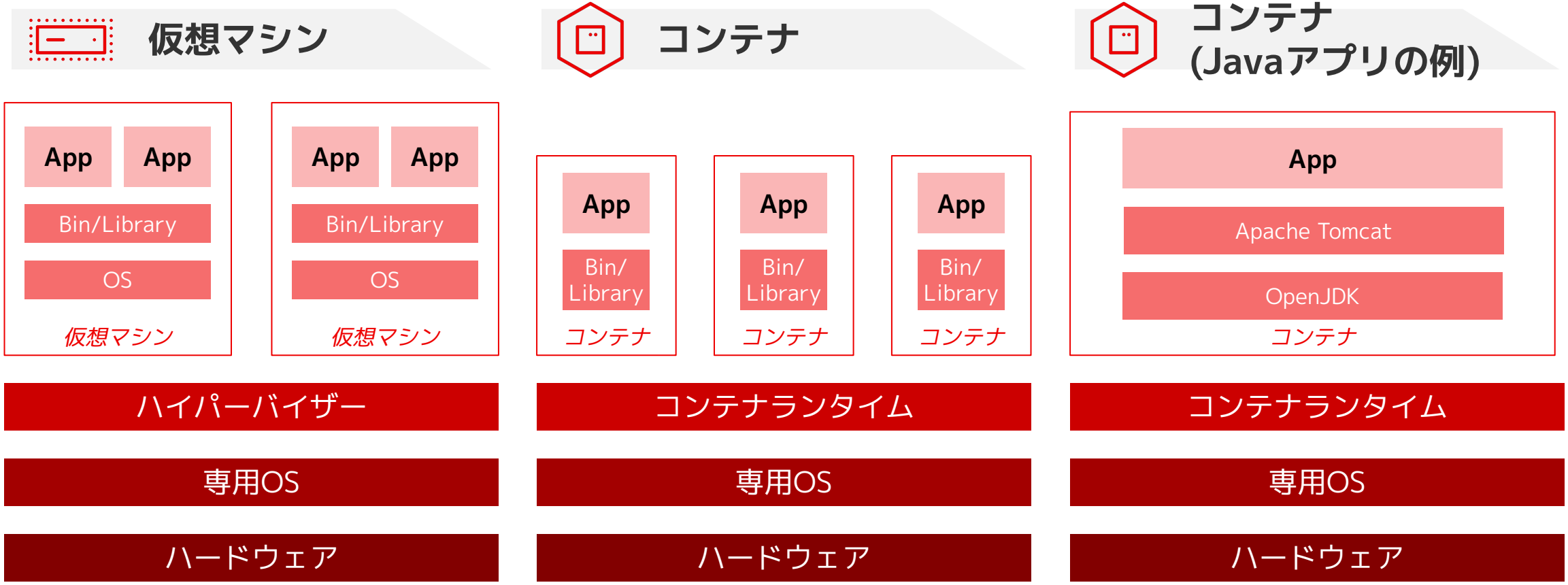
Red Hat

1.1 コンテナについて

Red Hat

1.1 コンテナと仮想マシンの違い

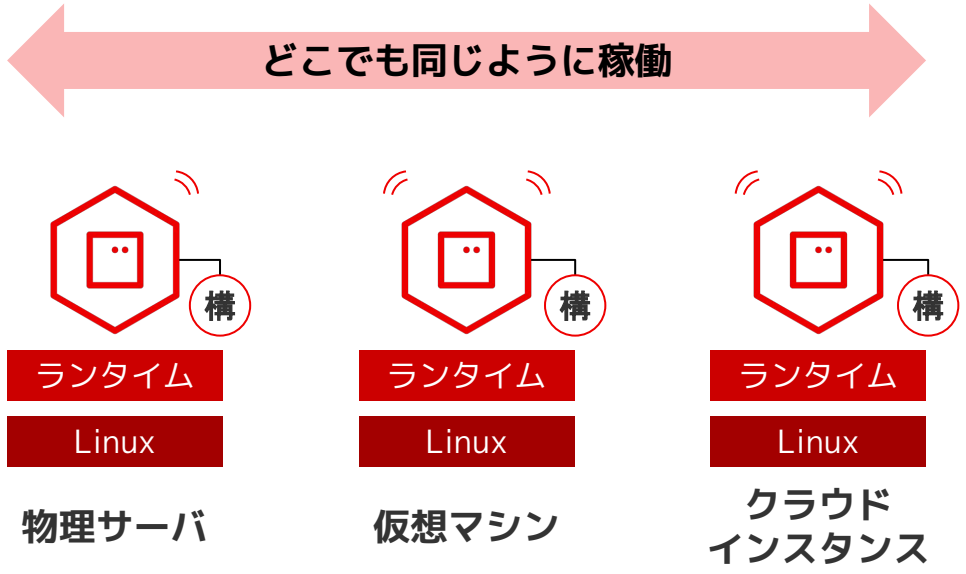
コンテナは、OSを持ちません。(すごく小さいLinuxカーネルを持ちます)
コンテナは、アプリケーションの動作に必要な実行環境がすべてコンテナの中に格納されます。
→OS全体に格納する共通ライブラリなどがないため、アプリ担当とインフラ担当の作業界面が減ります。
　また、OSバージョンアップに伴うアプリケーション影響度調査からの開放などのメリットがあります。



Red Hat

1.1 コンテナの特徴(1/2)

- **可搬性 (Portable)**
  - どの環境でも同じように稼働する。
    - 環境に依存する構成情報はコンテナとは別で持つ。

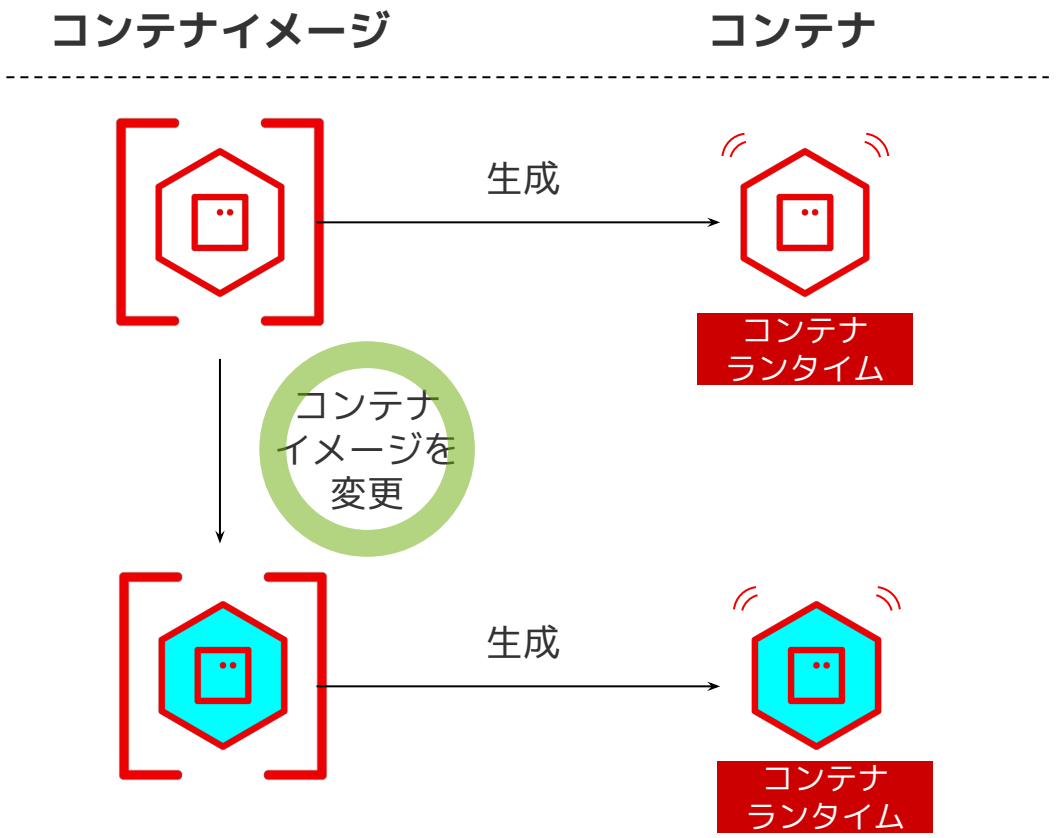


- **軽量**
  - OSが無く、必要最小限の要素のみ持つ。
- **起動が高速**
  - OS起動時間を省略できる。

	仮想マシン	コンテナ
容量	1桁 ～ 2桁 GB	2桁 MB ～ 1桁 GB
起動時間	数分	数秒

1.1 コンテナの特徴(2/2)

- **コンテナイメージから生成**
  - コンテナイメージから複製して作られる。
- **不変性 (Immutable)**
  - 同じコンテナイメージから起動したコンテナは、毎回必ず同じものとなる。
- **揮発性 (Ephemeral)**
  - コンテナに加えた変更は、コンテナが停止すると失われる。
    - コンテナ自身に永続性は無い。
    - コンテナに出力したログなども消えるため、永続化先に出力が必要。
  - コンテナに変更を加えたい場合は、コンテナイメージを変更して新しくコンテナを起動する。
    - 古いコンテナは破棄する。



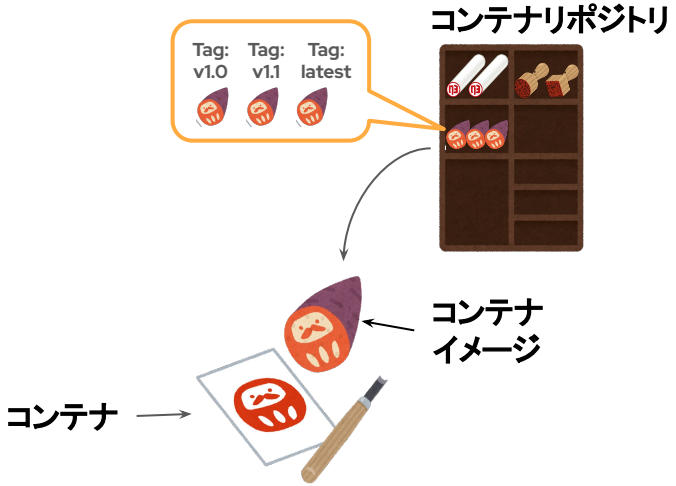
Red Hat

1.2 コンテナを使うために必要なもの

1.2 コンテナを使うために必要なもの

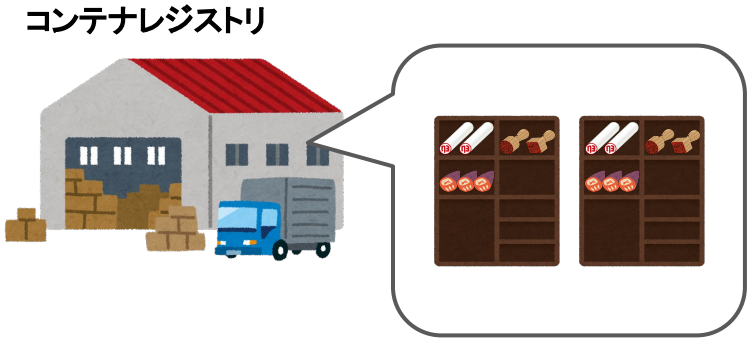
コンテナイメージ

- コンテナの素
- Public Image or Private Image
- 既存い追加変更し作成 (Build)
- セキュリティには要注意



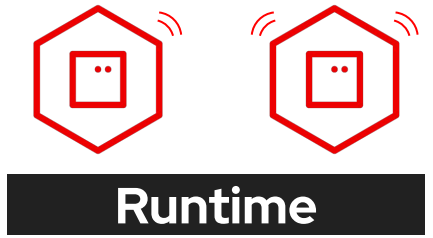
コンテナレジストリー

- コンテナイメージの保管庫
- イメージの引き出しをPull
- イメージのアップロードをPush
- Public or Private



コンテナランタイム

- コンテナを動かすためのソフト
  - ライフサイクル管理
  - ハードウェアリソースの分離
  - モニタリング・ロギング
  - イメージのPull・管理
  - イメージのBuild・Push



Red Hat

1.2.1 コンテナイメージについて

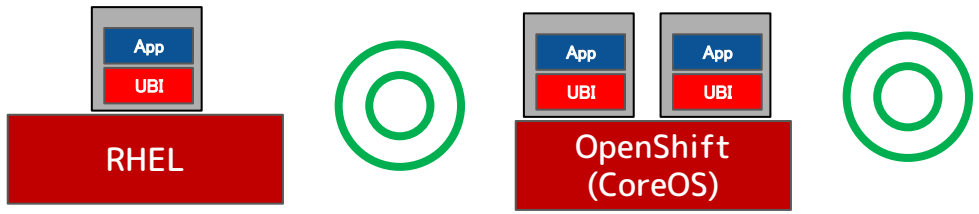
- コンテナイメージは、コンテナの素となるイメージ
  Red Hatは、RHELをベースとしたUBIというコンテナイメージを提供

再配布可能なコンテナイメージ

RHEL7および8のサブセットで、無償で入手・改変・再配布・実行が可能なコンテナイメージ

Red Hat基盤上の実行でサポートされる

RHEL / OpenShift / Red Hatが提供するコンテナエンジン上で利用する場合はUBIが正式にサポートされる。



ベース・イメージのバリエーション

- UBI　　標準ベースイメージ
- UBI-init　　複数サービス起動用
- UBI-minimal　サイズを切り詰めたイメージ
- UBI micro　　さらにサイズを切り詰めたイメージ

言語ランタイムやMW等の組込済のものも提供：

Node.js, Python, Php, Ruby,Tomcat,JBoss ...

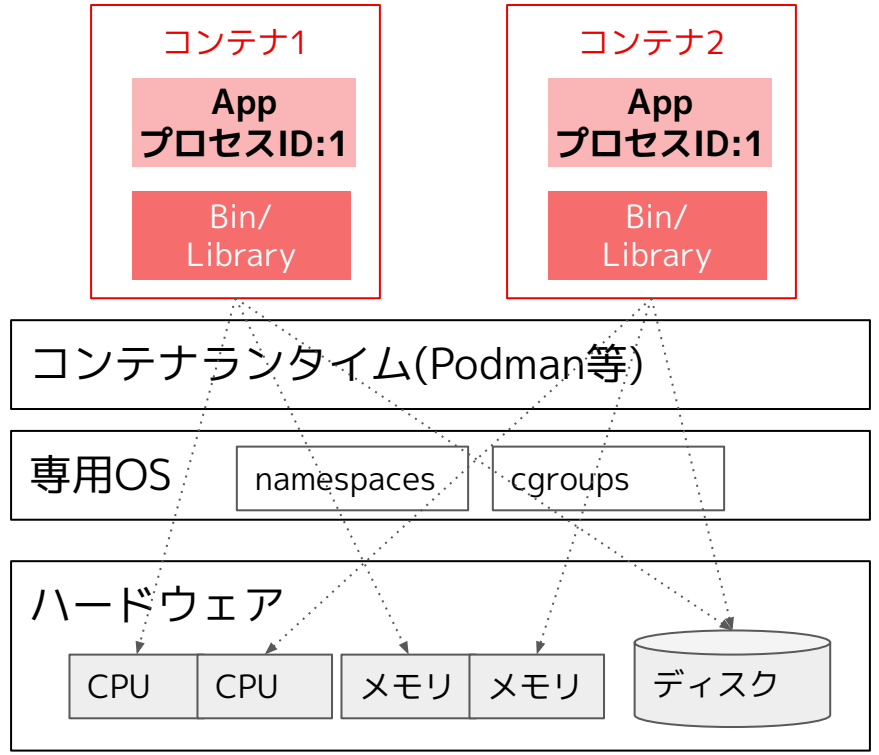
Red Hat

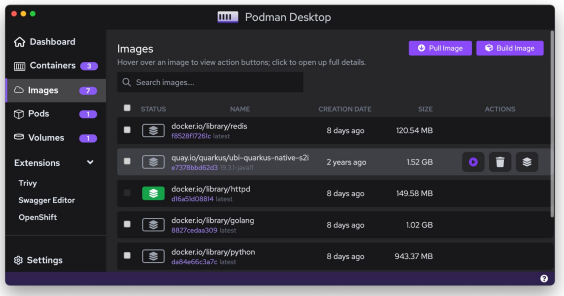
1.2.2 コンテナレジストリについて

- コンテナレジストリは、コンテナイメージの保管庫です
  コンテナイメージの操作(取得や格納)は、RESTでDocker Registry HTTP API V2がデファクトのプロトコルになっています。
- コンテナイメージは、パブリック、プライベートなものがあります
  - パブリック
    - 公に公開されていたり、インターネット経由で利用できるもの
      - Docker Hubが有名
      - Red Hat提供のコンテナイメージは、Red Hat Ecosystem Catalogで公開
        https://catalog.redhat.com/platform/red-hat-openshift/software/search
      - 開発成果物のコンテナなどの格納用で、SaaS型のサービスも多数あり
        Red Hatは、Red Hat Quay(SaaS版)を提供
  - プライベート
    - PJ用や自社用で利用するコンテナレジストリ
      - OpenShiftでは、内部コンテナレジストリを標準で提供
        - 複数の環境(正:東京、副:大阪)では、Red Hat Quay(インストール版)や、プライベートレジストリ用のソフトウェアを利用

1.2.3 コンテナランタイムについて

- コンテナが稼働するために必要なソフトウェア
  - コンテナの作成・実行・停止・削除などのライフサイクルの管理をする。
  - コンテナランタイムがなければコンテナは動かない。
- コンテナランタイムの仕組み
  - コンテナランタイムはLinuxカーネルの技術を使い、コンテナの実行環境の隔離を実現している。
    - namespaces(プロセスなどの分離)
    - cgroups(CPU/メモリ/IOの制御)　等
      →OSから見るとコンテナは制限付きプロセスです
- 有名なコンテナランタイム
  - ローカル環境で使うとき
    - Docker, Podman(RHEL同梱)
  - コンテナ基盤(Kubernetesなど)で使うとき
    - containerd, cri-o(OpenShiftで利用)





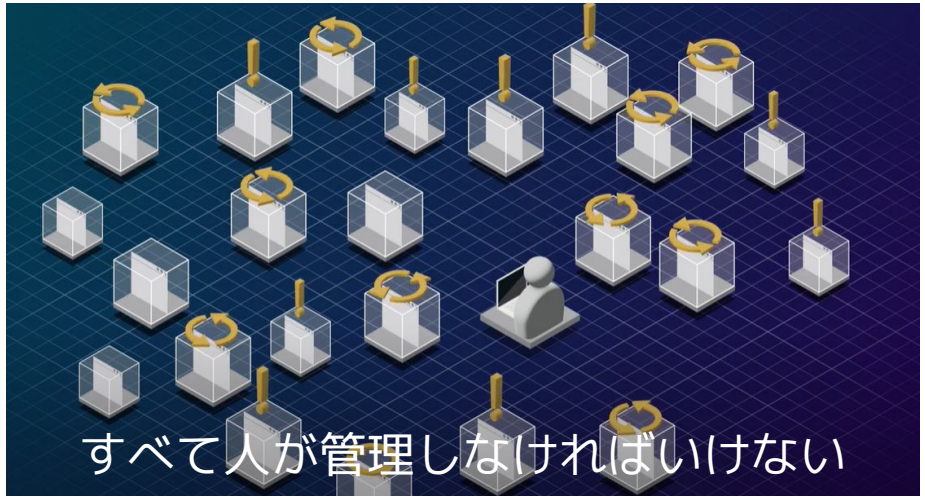
Podman Desktop(Windows,Mac,Linux対応)
(2023年に正式リリース予定 現時点でも使えます)

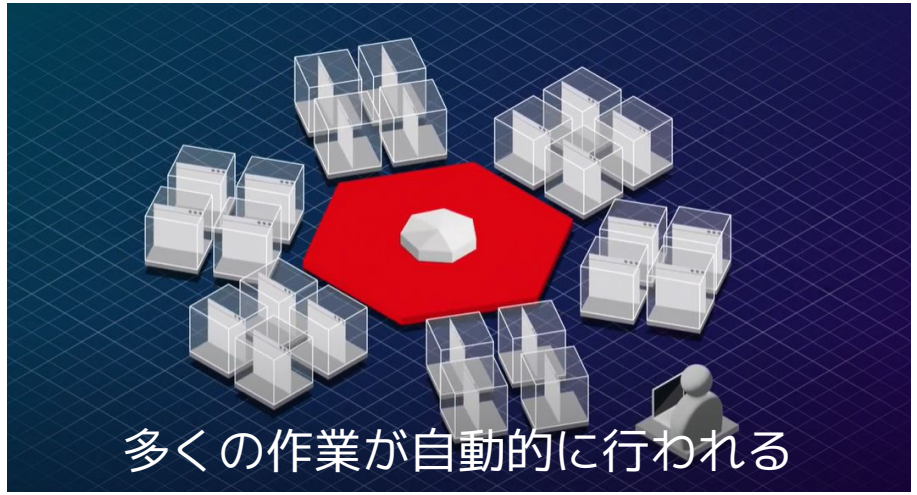
Red Hat

1.3 kubernetesについて

1.3 コンテナオーケストレーションについて

マイクロサービス化などによってコンテナの数が増えれば増えるほど、管理が複雑化します。
コンテナオーケストレーションは、コンテナの管理・運用を自動化するためのものです。





コンテナランタイムしかない場合の運用スタイル
- 属人的な障害復旧オペレーション
- 手動によるのコンテナ変更作業
- アプリケーションごとの設定管理
- 定期的な監視作業

コンテナオーケストレーションを使った場合の運用スタイル
- コンテナ運用操作の自動化
  - アクセス負荷分散
  - コンテナの死活監視
  - リソースの制御　など

コンテナオーケストレーションは、Kubernetesがスタンダードの地位を確立しています

Red Hat

1.3 Kubernetes(k8s)とは？

- コンテナオーケストレーションを実現する機能群を提供
  - コンテナデプロイ
  - コンテナ連携
  - サービスディスカバリー
  - 負荷分散
  - オートスケーリング
  - ログ表示
  - ストレージ管理
  - 宣言的な(=あるべき状態を記す)リソース定義
    - Manifestファイルで定義します
    - k8sは何でもリソースで管理します

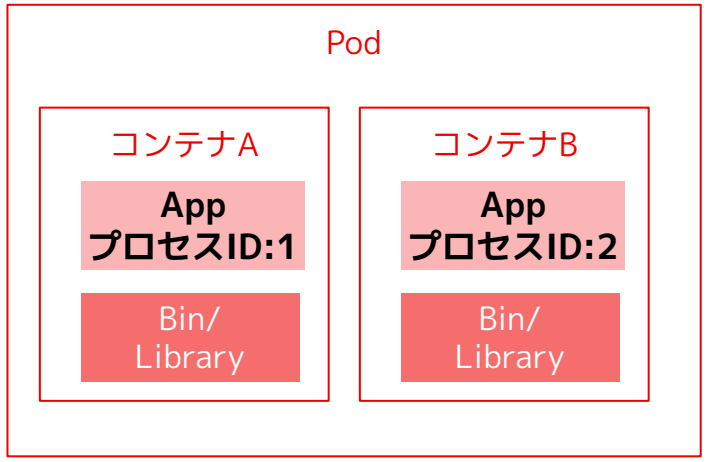
      → **k8sを利用するとIaCになる**

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: test-nginx
spec:
  selector:
   matchLabels:
      app: nginx
  replicas: 3
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:xxxxxxxxxx
        ports:
        - containerPort: 80
```

Manifestファイルの例(yaml形式)

1.3 Kubernetes(k8s)のコンテナ管理単位(Pod)

- k8sでは、コンテナを複数まとめたPodというもので管理します。
  - Pod内のコンテナは全て同じIPアドレスで管理され、
    CPU、メモリ、NWなどのリソースを共有
  - コンテナと同様に揮発性 (Ephemeral)であり、コンテナに出力したログなども消えるため、
    永続化が必要な場合は永続ボリューム等に出力が必要。
  - PodもK8sのリソースであり、Manifestファイルで定義
    (通常、イチからファイルを作成せず、定義の流用やコマンドで生成などが一般的です)



Podのイメージ図

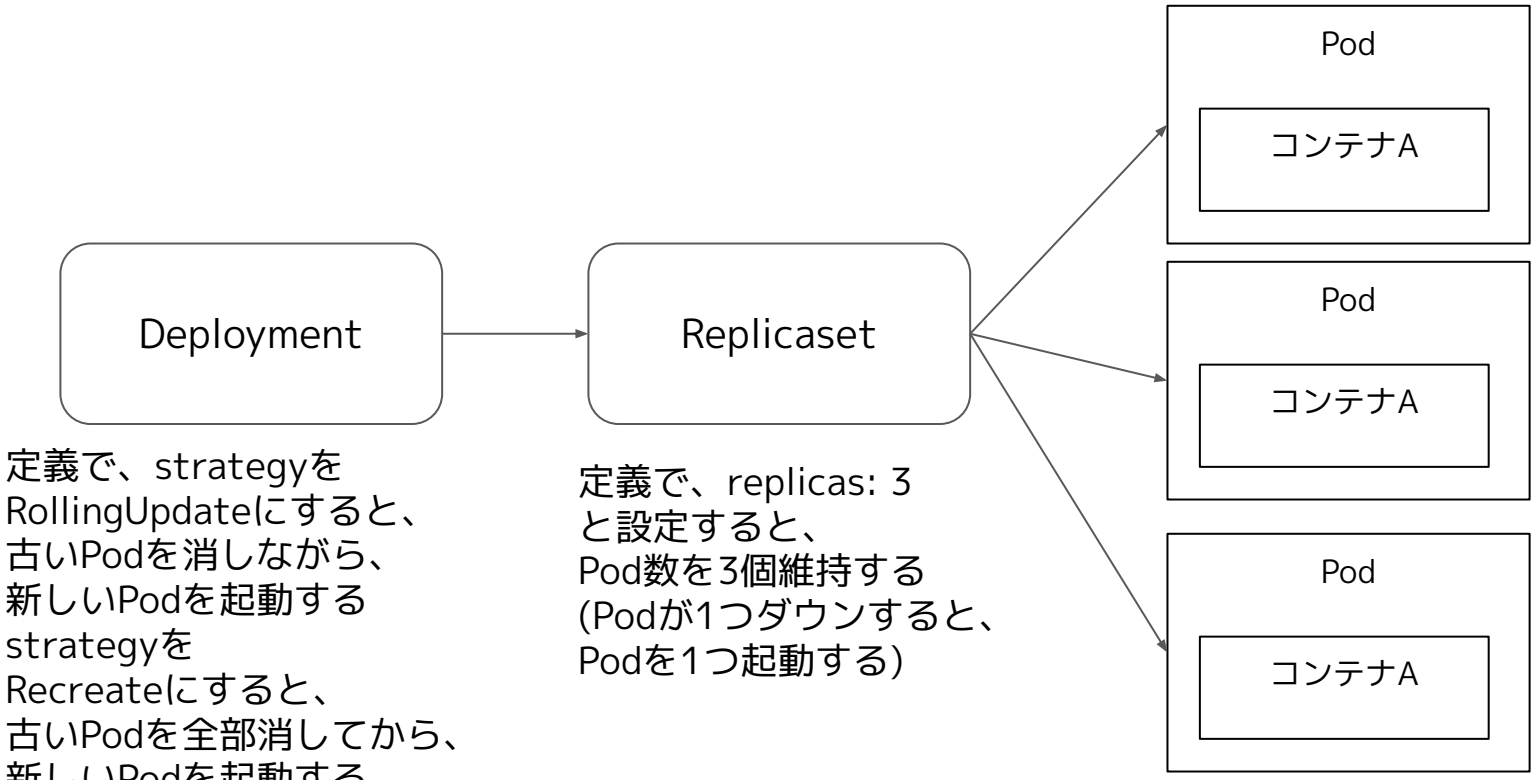
```
apiVersion: v1
kind: Pod ←リソースの種類(kind)がPod
metadata:
  name: httpd ←Podの名前
spec:
  containers:
  - name: httpd
    image: httpd:latest ←使うコンテナイメージ
    ports:
    - containerPort: 8080  ←コンテナで公開するポート番号
```

PodのManifestファイルの例

Red Hat

1.3 Kubernetes(k8s)のコンテナ管理-スケールアウト

- Podの数の管理は、Replicasetというリソースで実現します。
- Podのアップグレードの挙動(ローリング、全消し)や起動タイムアウトなどは、Deploymentで実現します。
- 通常、Deployment、ReplicasetとPodは一緒に使われます(Deploymentの定義ファイルに纏めて記載)



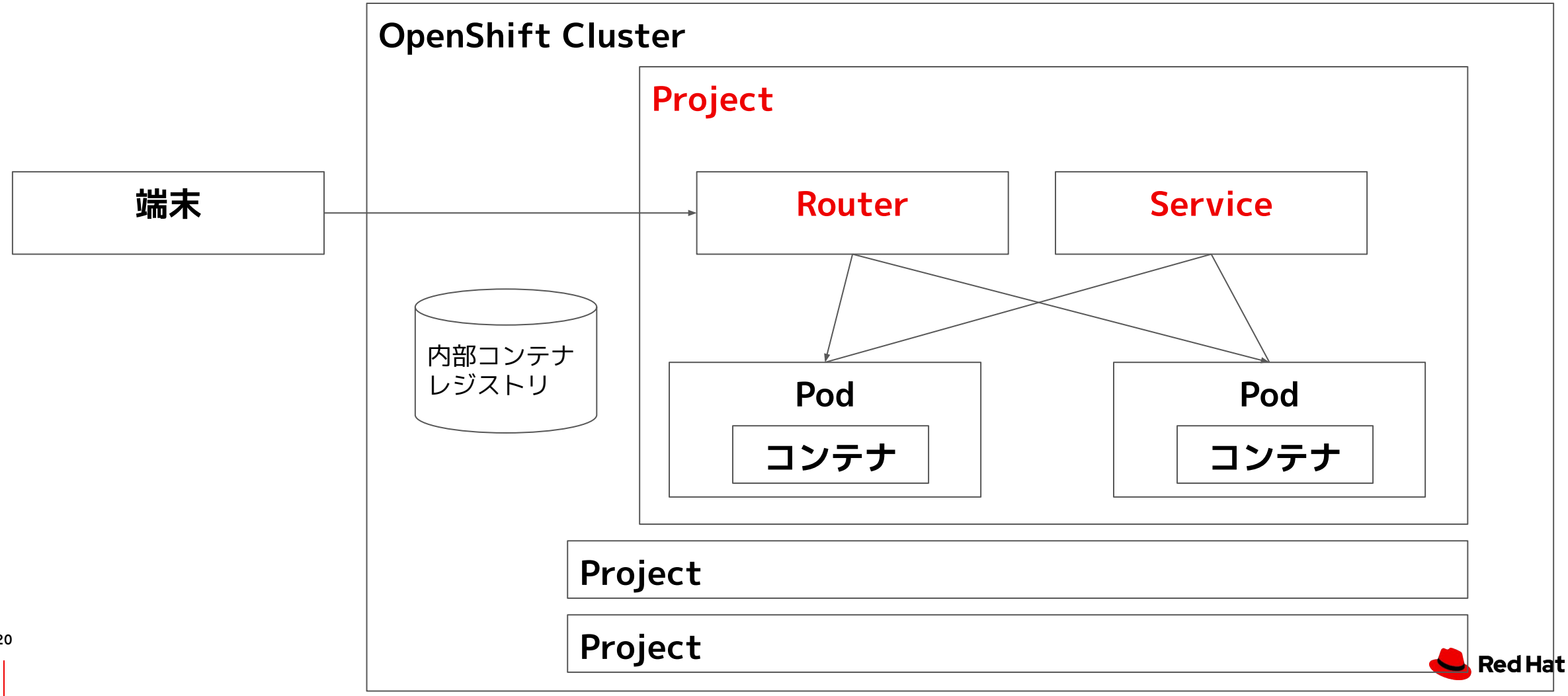
Red Hat

1.4 OpenShiftについて

Red Hat

1.4 OpenShiftの論理的な構成

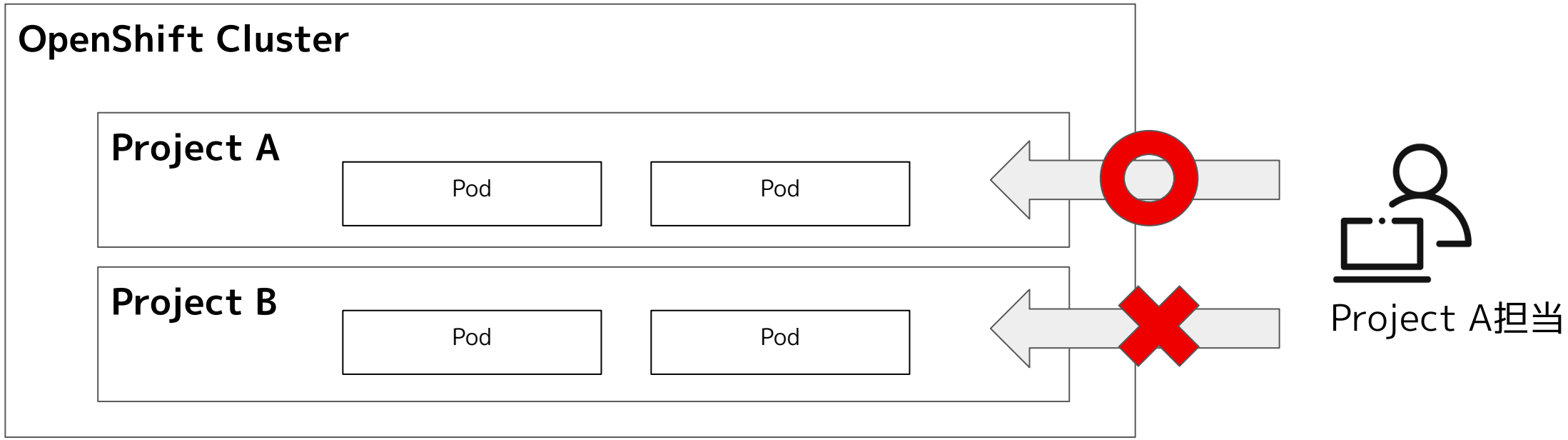
コンテナをデプロイする場合のOpenShiftの構成を次に示します。
(コンテナをビルドする場合の構成は、テーマ２で取り扱います)



Red Hat

1.4 Projectについて

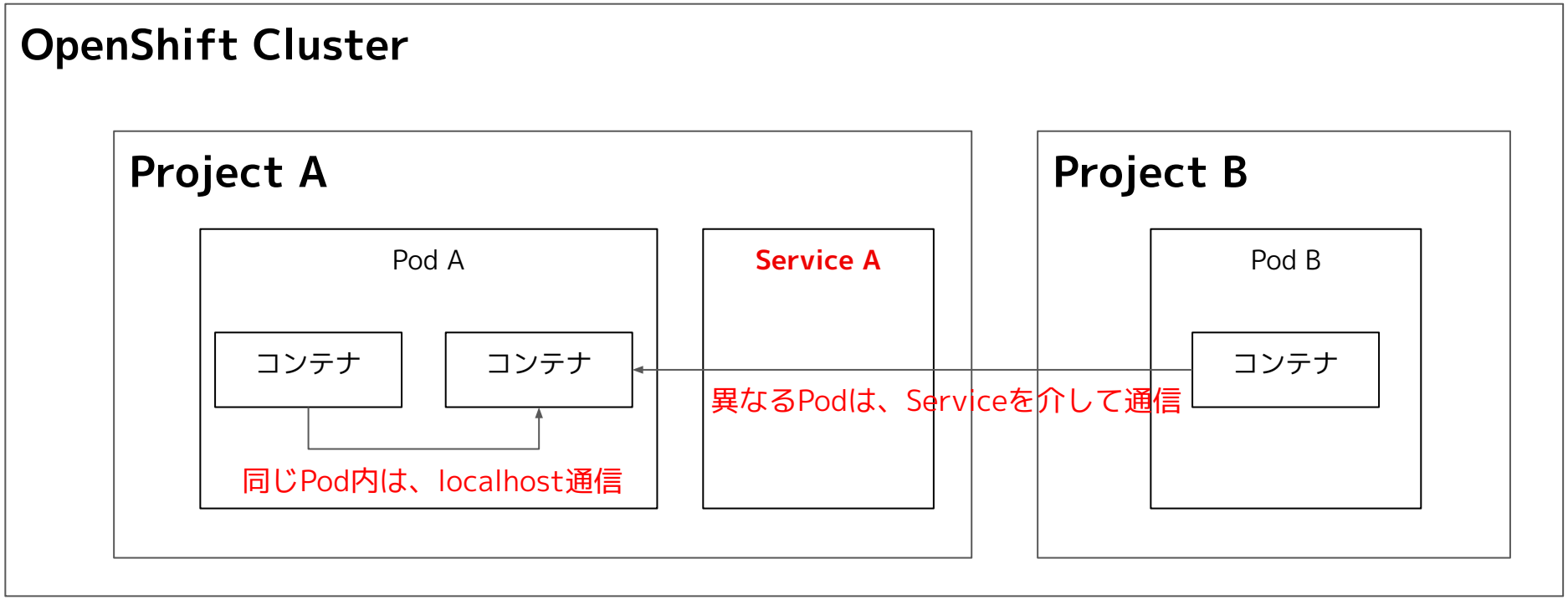
- Projectは、リソースのスコープやリソース制限を設定するための管理単位
  - Project単位で、アクセス制御や権限制御が可能(→マルチテナントの実現)
  - Project単位で、リソース制限(作成Pod数、CPU、メモリの上限)(→マルチテナントでの安定稼働を実現)
  - Project内に、Pod等の各種リソースを展開します。



Red Hat

1.4 Serviceについて

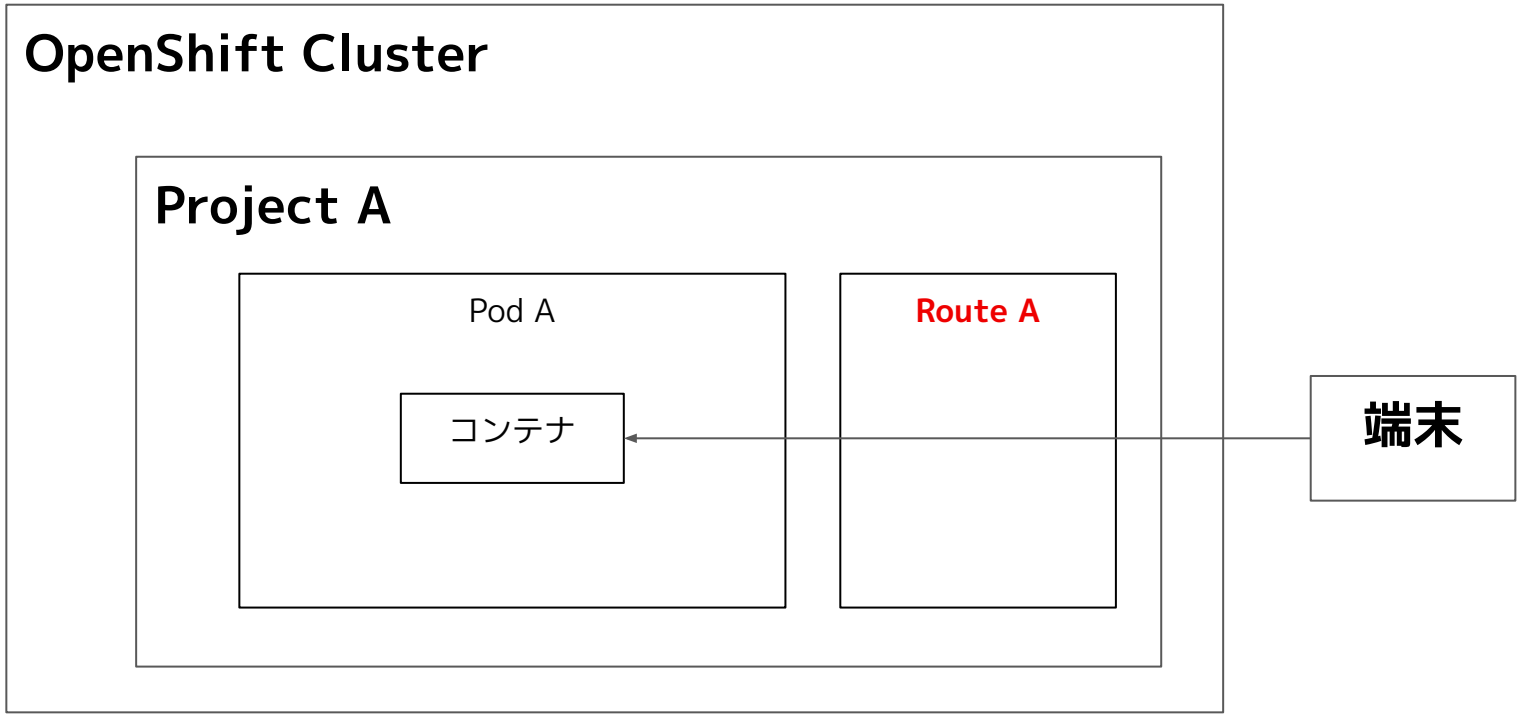
- Serviceは、OpenShift**内**のPod/コンテナ間通信を実施します。
  - Podは、配置されるNode(サーバ)が可変で、Podもスケーリングするため、各PodのIPアドレスなどではなく、Serviceを使って通信します。
  - 同じPod内の通信は、Pod 内のコンテナーはIP アドレスとポートマッピングを共有しているためlocalhostで通信します。



Red Hat

1.4 Routeについて

- Routeは、OpenShift**外**の通信を実施します。
  - アプリごとの固有のDNS名を指定して、アクセスを行います。
  - 負荷分散機能やSSL終端機能などもあります。



Red Hat

パートナー様
無償教育のご案内

Red Hat

効率的に次世代技術者を育てる

Red Hatでは、**有償のトレーニング**によって、効率的に知識を得るためのコースをご用意しています。
Red Hat のパートナー企業様向けには、一部の有償のトレーニングが無償で受講可能です。

開発編

OpenShiftを使用して、コンテナアプリやマイクロサービスの設計、開発について学びます

(DO101) Free for Partner	(DO180) Free for Partner	(DO288) Free for Partner	(DO292)
Introduction to OpenShift Applications	Introduction to Containers, Kubernetes, and Red Hat OpenShift	Red Hat OpenShift Development I: Containerizing Applications	Red Hat OpenShift Development II: Creating Microservices with Red Hat OpenShift Application Runtimes
コンテナアプリの概要	Kubernetes/OpenShift基礎	コンテナアプリ開発実装	マイクロサービス開発実践

運用編

OpenShiftを使用して、アプリやプラットフォームを作成、構成、管理するスキルについて学びます

(DO500) Free for Partner	(DO180) Free for Partner	(DO280 / DO380) Free for Partner	(DO425)
DevOps Culture and Practice Enablement	Introduction to Containers, Kubernetes, and Red Hat OpenShift	Red Hat OpenShift Administration I / II	Red Hat Security: Securing Containers and OpenShift
DevOpsのプラクティス概要	Kubernetes/OpenShift基礎	OpenShift運用管理実装	コンテナセキュリティの実践

(DO180,D0280)Kubernetes/OpenShift基礎が、本テーマで取り扱った内容の深堀りした教材になります。

無償で利用可能なレッドハットラーニングコース（日本語有）

Application Development Courses	
Cloud-Native Integration with Red Hat Fuse (AD221)	32h
Developing Application Business Rules with Red Hat Decision Manager (AD364)	24h
Red Hat AMQ Administration (AD440)	16h
Developing Event-Driven Applications with Apache Kafka and Red Hat AMQ Streams (AD482)	24h

Cloud Courses	
Red Hat OpenStack Administration 1: Core Operations for Cloud Operators (CL110)	40h
Red Hat OpenStack Administration 2: Day 2 Operations for Cloud Operators (CL210)	32h
Cloud Storage with Red Hat Ceph Storage (CL260)	40h

Platform Courses	
Red Hat System Administration I (RH124)	40h
Red Hat System Administration II (RH134)	40h
Red Hat Enterprise Linux Automation with Ansible (RH294)	32h
Red Hat Virtualization (RH318)	40h
Red Hat Enterprise Linux 8 New Features for Experienced Administrators (RH354)	32h

DevOps Courses			
Introduction to OpenShift Applications (DO101)	8h	Red Hat OpenShift Installation Lab (DO322)	16h
Red Hat OpenShift I: Containers & Kubernetes (DO180)	24h	Red Hat OpenShift Migration Lab (DO326)	24h
Cloud-Native API Administration with Red Hat 3scale API Management (DO240)	32h	Red Hat OpenShift Service Mesh (DO328)	24h
Red Hat OpenShift II: Operating a Production Kubernetes Clusters (DO280)	24h	Enterprise Kubernetes Storage with Red Hat OpenShift Data Foundation (DO370)	32h
Red Hat OpenShift Development II: Containerizing Applications (DO288)	32h	AAP 2.0 Developing Advanced Automation with Red Hat Ansible Automation Platform (DO374)	32h
Red Hat OpenShift Administration III: Scaling Kubernetes Deployments in the Enterprise (DO380)	32h	Red Hat Cloud-Native Microservices Development with Quarkus (DO378)	32h

Red Hat OpenShift I: Containers & Kubernetes (DO180)が、本テーマで取り扱った内容の深堀りした教材になります。

Red Hat

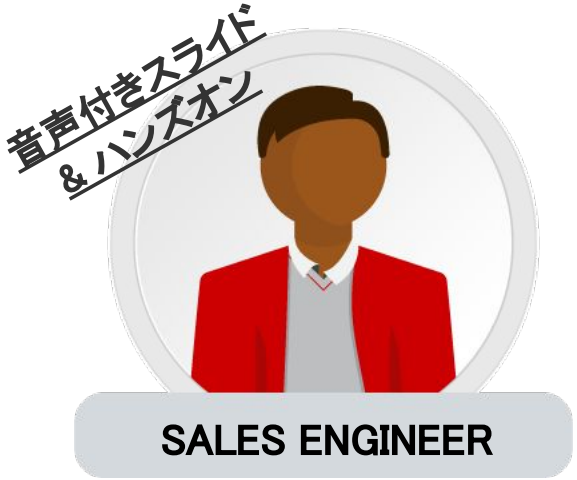
Partner Training Portal (PTP)

Role(役割)ごとのコンテンツロードマップを提供

- **Red Hat Partner Training Portal** は、Red Hat のパートナーの方向けの無償トレーニングの仕組みで、e-learning とオンラインのラボで構成されています。
- ほぼ全ての Red Hat 製品に対し、営業向け、セールスエンジニア向け、デリバリースペシャリスト向けのコースが提供されています。一部コースには日本語版も提供されています。
- パートナーの皆様は、**アカウントを作成するだけで無償利用が可能です。**

SALES

Red Hatの価値を理解
案件の精査
対競合優位性の理解
反論への対処
価格



SALES ENGINEER

技術的営業提案
案件の精査
対競合優位性の理解
反論への対処
デモの習得
製品の使い方に関するハンズオン

DELIVERY

製品導入
アプリケーション開発
PoCの実施
ソリューションアーキテクチャー
インプリメンテーション含めたハンズオン

Red Hat

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.


linkedin.com/company/red-hat


youtube.com/user/RedHatVideos


facebook.com/redhatinc


twitter.com/RedHat

Red Hat

コンテナ開発体験のアジェンダ

コンテナの作成からCI/CDまでの一連の流れがOpenShift上で体験できるカリキュラムとなっております。
各時間枠は、ハンズオンの進捗により多少前後します。

13:00-13:10 (10min)	オープニング、ハンズオン概要のご紹介、端末環境などのご説明 出席者自己紹介（業務のご担当内容など）
13:10-14:20 (50min)	テーマ１：コンテナの開発体験 座学：コンテナをデプロイするためのk8s/OpenShiftの機能・知っておくべきコンポーネントのご説明 ハンズオン：OpenShift上にて、アプリケーションのデプロイとコンテナのスケールを体験
14:10-**14:30** (10min)	休憩
14:30-**15:50** (80min)	テーマ２：CIの実践 座学：コンテナのビルド、CIの必要性とCIパイプライン作成の流れなどのご説明 ハンズオン：CIパイプライン(tekton)の作成とパイプラインへの静的診断やイメージ脆弱性診断の組込みを体験
15:40-15:50 (10min)	休憩
15:50-17:10 (80min)	テーマ３：CDの実践 座学：CIとCDの違い、Gitで継続的デリバリーを実現するGitOpsなどのご説明 ハンズオン：ArgCDを使った開発環境、本番環境へのCD(自動デプロイ)の体験
17:10-17:40 (30min)	PTPワークショップのご紹介 まとめ・クロージング ご出席者のご意見（今日学んだこと・業務に取り入れるにあたっての課題など）

コンテナ/K8s基礎が基本

ここで、コンテナのデプロイを体験

ここで、コンテナビルドとCIをご説明と体験

ここで、CDをご説明と体験