Kubernetes: アプリケーションの実行

～Pod、ReplicaSet、Deploymentでアプリを動かしてみよう！～

所要時間: 約30分
対象者: Kubernetesの基本を学びたい方、アプリケーションのデプロイ方法を知りたい方

このスライドのゴール 🎯

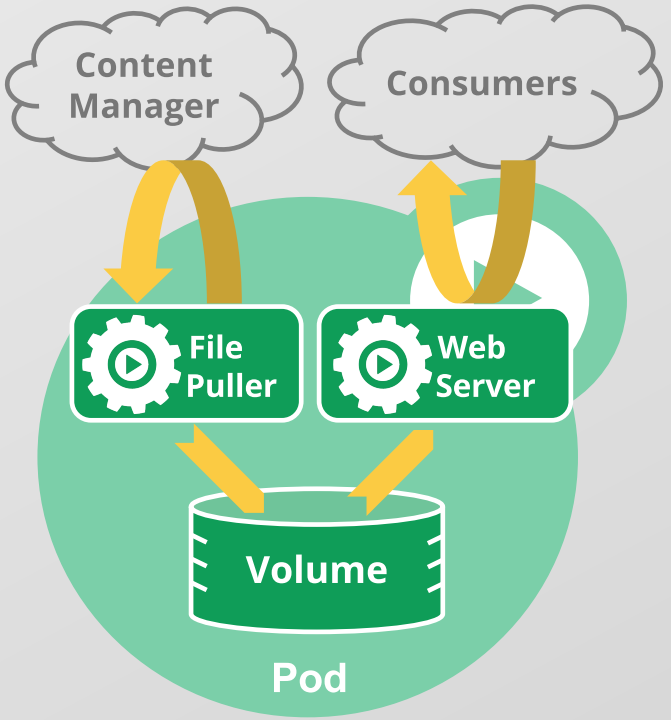
Kubernetesの主要なワークロードリソースを理解する

- Pod: コンテナの基本的な実行単位と特徴
- ReplicaSet: 複数のPodを維持する仕組み
- Deployment: アプリケーションの無停止更新と管理
- StatefulSet/Job: 特殊な要件を持つアプリの実行方法

Podとは？

Pod: コンテナの家 🏠

- Kubernetesの最小単位
- コンテナを入れる「箱」みたいなもの
- 同じPodの中のコンテナは仲良し（ネットワークやストレージを共有）



Podの特徴：コンテナの同居生活 👥

- **同じIPアドレスで暮らす**
  - 同じPodの中なら`localhost`で会話できる
  - 外からは1つのアプリとして見える
- **一生は短い**
  - 作られて、動いて、終わる
  - 終わったら新しいPodが作られる

実際のPod定義を見てみよう 👀

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  labels:
    app: nginx
spec:
  containers:
  - name: nginx
    image: nginx:1.14.2
    ports:
    - containerPort: 80
```

シンプルでしょ？でも、これだけだと...

問題：Podが1つだけだと... 😱

- Podが落ちたらアプリが使えなくなる
- トラフィックが増えたら対応できない
- 更新するときは一旦止める必要がある

→ どうすればいい？

ReplicaSetとは？

ReplicaSet: Podのコピー機 📋

- 指定した数のPodを常に維持
- Podが落ちたら自動で新しいのを作る
- スケーリングも簡単！

ReplicaSetの定義例

```yaml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: nginx-replicaset
spec:
  replicas: 3  # 3つコピーを作る
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
〜〜〜省略〜〜〜
```

Deploymentとは？

Deployment: アプリの運営マネージャー 👨‍💼

- 複数のPodを管理
- 無停止で更新できる
- 問題があったら元に戻せる
- ReplicaSetの上司的存在

Deploymentの特徴：賢い管理 🧠

1. **レプリカ管理**
    - 指定した数のPodを維持
    - スケーリングも簡単
    - Podが落ちたら自動で再起動
2. **更新戦略**
    - ローリングアップデート（デフォルト）
        - 1つずつ新しいバージョンに更新
        - サービスは止まらない！

Deploymentの定義例

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 1         # 最大で何個多く作れるか
      maxUnavailable: 1   # 最大で何個止められるか
〜〜〜省略〜〜〜
```

ReplicaSet vs Deployment: どっちがいい？🤔

機能	ReplicaSet	Deployment
基本的な役割	単純なPodの複製のみ	ReplicaSetを管理
更新機能	更新機能なし	無停止で更新可能
ロールバック	ロールバック機能なし	問題があれば元に戻せる
履歴管理	履歴管理なし	更新履歴を保持
運用面	基本的に直接使用しない	実際の運用で最も使用される

→ Deploymentの方が圧倒的に便利！ReplicaSetはDeploymentの内部で使われる存在

StatefulSetとは？

StatefulSet: 順序と一意性が重要なアプリのためのコントローラー 🔢

- **順序付きのデプロイ**
  - Podに一意の識別子（0, 1, 2...）
  - 順番に起動・停止
  - データベースなどに最適
- **永続的なストレージ**
  - 各Podに専用のストレージ
  - Podが再作成されてもデータは保持
  - 一意のネットワーク識別子

StatefulSetの定義例

```yaml
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: mysql
spec:
  serviceName: "mysql"
  replicas: 3
  template:
    spec:
      containers:
      - name: mysql
        image: mysql:5.7
  volumeClaimTemplates:  # 各Podに永続ストレージを割り当て
  - metadata:
      name: data
    spec:
      accessModes: [ "ReadWriteOnce" ]
```

Jobとは？

Job: 一度だけ実行するタスクのためのコントローラー⚡

- **バッチ処理に最適**
  - 一度だけ実行するタスク
  - 成功するまで再試行
  - 並列実行も可能
- **CronJobとの組み合わせ**
  - 定期的なバッチ処理
  - スケジュール実行
  - 履歴管理

Jobの定義例

```yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: batch-job
spec:
  template:
    spec:
      containers:
      - name: batch
        image: batch-image:latest
        command: ["/bin/sh", "-c", "echo 'Processing...' && sleep 30"]
      restartPolicy: OnFailure
  backoffLimit: 4  # 失敗時の再試行回数
```

CronJobの定義例

```yaml
apiVersion: batch/v1
kind: CronJob
metadata:
  name: scheduled-job
spec:
  schedule: "*/5 * * * *"  # 5分ごとに実行
  jobTemplate:
    spec:
      template:
        spec:
          containers:
          - name: scheduled
            image: scheduled-image:latest
            command: ["/bin/sh", "-c", "echo 'Running scheduled task...'"]
          restartPolicy: OnFailure
```

よく使うコマンド集 🛠️

```
# デプロイする
kubectl apply -f deployment.yaml

# スケールする（5個に増やす）
kubectl scale deployment <deployment-name> --replicas=5

# イメージを更新する
kubectl set image deployment/<deployment-name> <container-name>=<new-image>

# 問題があったら元に戻す
kubectl rollout undo deployment/<deployment-name>
```

実践的なTips 💡

1. **リソース制限は必須**
    - CPU/メモリの制限を設定
    - 他のアプリに影響を与えないように
2. **ヘルスチェックを設定**
    - アプリが動いているか確認し、問題があれば自動で再起動
3. **更新戦略を最適化**
    - アプリの特性に合わせて設定
    - 安全に更新できるように

まとめ：Kubernetesの基本構成 🎓

- Pod: コンテナの家
- ReplicaSet: Podのコピー機
- Deployment: アプリの運営マネージャー

これらを組み合わせて、安定したアプリ運用を実現！

参考リンク 📚

- Kubernetes公式ドキュメント
  - Pod
  - ReplicaSet
  - Deployment
- Kubernetes チュートリアル
- Kubernetes 用語集