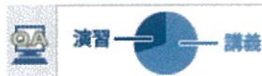


## 研修概要



### 【概要】

Dockerを実習環境として使用し、コンテナ型仮想化技術の基本的な使用方法を体験します。開発/テスト環境の構築方法や、コンテナ型仮想化によりアプリケーション開発やデプロイ、バージョンアップ作業などがどのように変化するかを学習します。初めてコンテナ型仮想化を使用するアプリケーション開発者向けの入門コースです。

### 【目標】

本コース修了後、次の事項ができることを目標としています。

1. コンテナ型仮想化技術の概要を説明できる。
2. コンテナ型仮想化技術を利用したPaaSの概要を説明できる。
3. アプリケーション開発における、コンテナ型仮想化技術のメリットを説明できる。
4. コンテナ型仮想化技術が導入済みの環境で、コンテナの起動やコンテナイメージのインポート、新たなコンテナイメージの作成などの作業ができる。

【期間】：1日間

### 【対象】

Dockerに関心があり、導入を検討しているアプリケーション開発者。コンテナ型仮想化による、アプリケーション開発手法の変化を把握したいアプリケーション開発者。

### 【前提知識】

Linuxの基本的な操作（ファイル操作やサービスの起動や停止など）ができることが望ましい。アプリケーション開発工程に関する基本的な知識があることが望ましい。

## 午前

### 第1章 コンテナ型仮想化の概要

- 1.1 コンテナの概要
- 1.2 Dockerの概要と特徴
- 1.3 Dockerの利用シーン

### 第2章 Dockerのアーキテクチャとコンポーネント

- 2.1 Dockerのコンポーネントの全体像
- 2.2 Dockerのアーキテクチャ
- 2.3 Dockerのイメージ管理と共有
- 2.4 コンテナの分離

## 午後

### 第3章 コンテナのライフサイクルとコマンド操作

- 3.1 Dockerコンテナのライフサイクル
- 3.2 コンテナ操作のためのコマンドリファレンス

### 第4章 Dockerイメージの管理と自動作成

- 4.1 Dockerイメージの作成の流れ
- 4.2 イメージ作成・管理のためのコマンドリファレンス

### 第5章 Docker ToolboxによるWindowsでのDocker利用

- 5.1 Docker Toolboxとは
- 5.2 Docker Toolboxに含まれるソフトウェア
- 5.3 Docker Machineとは
- 5.4 ローカルのVirtualBoxを利用する際のDocker Machineコマンド

### 第6章 サービス提供環境でのコンテナ利用の留意点

- 6.1 マイクロサービスとは
- 6.2 Docker Composeとは
- 6.3 Docker Swarmとは
- 6.4 OpenShiftとは