

2019年7月1日

株式会社インプレスR&D

<https://nextpublishing.jp/>

コマンドひとつで Kubernetes 上にアプリケーションが構築できる！「Helm」最新解説書  
『実践 Helm—自作アプリを Kubernetes クラスタに簡単デプロイ！』発行  
技術の泉シリーズ、7月の新刊

インプレスグループで電子出版事業を手がける株式会社インプレス R&D は、『実践 Helm—自作アプリを Kubernetes クラスタに簡単デプロイ！』（著者:磯 賢大）を発行いたします。

最新の知見を発信する『技術の泉シリーズ』は、「技術書典」をはじめとした各種即売会や、勉強会・LT 会などで頒布された技術同人誌を底本とした商業書籍を刊行し、技術同人誌の普及と発展に貢献することを目指します。

『実践Helm—自作アプリをKubernetesクラスタに簡単デプロイ！』

<https://nextpublishing.jp/isbn/9784844378051>



著者:磯 賢大

小売希望価格:電子書籍版 2000 円(税別)／印刷書籍版 2200 円(税別)

電子書籍版フォーマット:EPUB3／Kindle Format8

印刷書籍版仕様:B5 判／カラー／本文 170 ページ

ISBN:978-4-8443-7805-1

発行:インプレス R&D

### <<発行主旨・内容紹介>>

本書は、オープンソースのコンテナオーケストレーションシステム「Kubernetes」のパッケージマネージャー「Helm」の解説書です。

Helm の特徴は、アプリケーションのデプロイの容易さです。既存の Chart を活用したり Chart を作成したりすることで、「helm install」のコマンド一つでアプリケーションを Kubernetes 上に構築できます。本文中では Docker・Kubernetes 初心者が、Helm を使って Helm のテンプレートである Chart を自作できるまでを目指します。

Docker・Kubernetes は知っているけど「Helm ってなに？」という方にもおすすめの一冊です。

(本書は、次世代出版メソッド「NextPublishing」を使用し、出版されています。)

## Helm の概要の前に、パッケージマネージャーの変遷を解説

```
# JDKのインストール
$ yum -y install java-1.8.0-openjdk
# リポジトリの設定
$ curl -o /etc/yum.repos.d/jenkins.repo \
https://pkg.jenkins.io/redhat-stable/jenkins.repo
$ rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key
# Jenkinsのインストール
$ yum -y install jenkins
# サービスの有効化
$ systemctl enable jenkins
# サービスの開始
$ systemctl start jenkins
```

図1.1: CentOS7上にJenkinsを構築する例

サーバーorVM  
Jenkins  
Java  
CentOS

このように、「コマンドひとつで簡単インストール」というわけにはいきません。JenkinsはMaster-Slave構成を取ることができます。当然Slave構成を取る場合はサーバーを用意し、JavaやGitなどの必要なミドルウェアを自分で設定する必要があります。

### 1.1.2 コンテナの利用

YumやAPTを利用するのはオンプレミスのサーバーやVMにソフトウェアを入れる場合のユースケースですが、ここでコンテナを利用すると導入が容易になります。コンテナの特徴のひとつに「Build Once. Run Anywhere.」という言葉があります。文字通り、一度構築してしまえばDockerが動く環境ならばどこでも実行が可能になるのです。Jenkinsをコンテナで構築・実行する場合は、次のDockerコマンドを実行します。

```
$ docker run -p 8888:8888 -p 50000:50000 jenkins
```

図1.2: DockerでJenkinsを構築する例

サーバーorVM  
Jenkins Container  
Jenkins Layer  
Java Layer  
Docker  
LinuxOS

YumやAPTを利用する場合より、コマンドひとつで起動できる分、非常に構築が簡単です。

### 1.1.3 複数のコンテナの連携

単純にJenkinsサーバーを立てるだけならこれで完了です。では、続いてMaster-Slave構成を立てるケースを考えてみます。

これを実現するには

1. Jenkins Master コンテナ (一個)
2. Jenkins Slave コンテナ (N個)

の2種類のコンテナを起動し、連携させる必要があります。複数のコンテナを起動させるために、次のdocker-compose.ymlを起動します。

```
リスト1.1: docker-compose.yml
version: "3"
services:
  master:
    container_name: jenkins-master
    image: jenkins
    ports:
      - 8888:8888
      - 50000:50000
```

10 | 第1章 Helmの概要・基本構成

第1章 Helmの概要・基本構成 | 11

## Helm アーキテクチャーの全体構成を図を用いて解説

- istio
- jaeger
- kafka
- vault
- zookeeper
- etc.

Helmの公式Chartは、Docker Hubでコンテナイメージ探すために、HelmHub<sup>6</sup>というサイトで検索できます。

図1.9: Helm Hub Datadog Chartの例

Helm Hub  
datadog  
6.0.0 - stable  
Datadog Agent

Installing a centralized infrastructure monitoring platform. This chart takes the Datadog Agent to all nodes in your cluster as a DaemonSet. It also optionally, depends on the kube-state-metrics chart. For more information about monitoring Kubernetes with Datadog, please refer to the Datadog documentation website.

Datadog offers two variants, `helm.io - v3x` and `tgz` if you need to run Jenkins integrations. The chart also supports templating for installation-dependent things.

See the Datadog Helm integration to learn more.

Prerequisites

- Kubernetes 1.4+ or OpenShift 3.4+, 1000 nodes
- The Datadog Agent supports Kubernetes 1.3+

Chart Versions

### 1.2.1 Helmのアーキテクチャーとコンポーネント

本章のテーマである、Helmの基本構成について説明します。Helmはクライアント・サーバー構成のソフトウェアです。個別に詳細を説明する前に、Helm全体のアーキテクチャーを図1.10の概要図で確認してみましょう。なお図1.10や本書で説明しているHelmは「バージョン2系」のものです。執筆時点(2019年04月現在)3系の開発が進んでいます。このHelm v3ではTillerサーバーが廃止され、構成がシングルサービスアーキテクチャーに変更になります。詳細は次のリンクを参照してください。

- <https://github.com/helm/community/blob/master/helm-v3/000-helm-v3cmd#summary>

©Helm Hub/istio / Hub Helm.io/

図1.10: Helmアーキテクチャー全体概要図

Helm Charts Repository (Private, Stable, Incubator, Other) - xxx.tgz, index.yaml

Helm Client - Helm コマンド

Kubernetes Cluster - kube-system, Tiller, api server, Namespace (Deployment, Ingress, Service, etc.), Kubernetesリソースの操作

Go言語で書かれたHelmClientが、Kubernetesクラスター上で稼働するTillerサーバーと応答することで、Kubernetes上にソフトウェアの構築や削除を行います。

### 1.2.1.1 HelmClientとTiller

Tillerは、Kubernetesクラスターの「kube-system」というシステム用の名前空間上で、コンテナとして稼働します。HelmClientは、gRPCプロトコルでTillerと応答します。そして、TillerがREST API通信でKubernetesのAPI Serverに、HelmClientからの命令を伝達します。

### 1.2.1.2 Helm Chartsリポジトリ

構築するソフトウェアは、パッケージのテンプレートであるChartに定義が詰め込まれています。Chartはtgz形式で固められており、通常はHelm Chartsリポジトリから取得されます。Chartリポジトリには、公式リポジトリや企業が公開しているリポジトリがあります。外部に公開したくないChartを配置するためにプライベートリポジトリを構築することも可能です。GitHubやGitLabなどのGitリポジトリにtgz形式のChartを配置することでHelmリポジトリとして利用できます。リポジトリにはCharts(複数のChart)の目録となるindex.yamlを配置します。

7.0 - 0.4.4 構築で作成したChartをリポジトリに格納して利用することも可能です

22 | 第1章 Helmの概要・基本構成

第1章 Helmの概要・基本構成 | 23

# 自作の Web アプリケーションを独自 Chart 化できるまでを解説

## 第4章 Helm Chartを自作しよう

本章では、Helm Chartを自作します。自作のWebアプリケーションを独自Chart化できるようになるのがゴールです。公式ドキュメントのチュートリアルを確認した後、自作WebアプリケーションをChart化します。

### 4.1 Chart作成チュートリアル

アプリケーションのChartに取り掛かる前に、Helm公式ドキュメントのチュートリアル<sup>1</sup>を参考に簡単なChartを作ってみましょう。本節はChartを作る際の考え方やテクニック（変数やFunctionの理解）の感覚を掴むことを目的とします。公式ドキュメントのチュートリアルに沿っていますが、全てを掲載しているわけではありません。全文を確認したい方は公式ドキュメント<sup>2</sup>を参照してください。

#### 4.1.1 helm create

第3章「Helmでアプリケーションをデプロイ」で登場したhelm createでサンプルのChartを作成します。すでにこれまでの章で見てきたとおり、templateには変数化されたYAMLマニフェストがあり、values.yamlにはその変数定義が記載されています。

```
# mychart を作成
$ helm create mychart

# ディレクトリの中身を確認
$ tree mychart/
mychart/
├── Chart.yaml
├── charts/
├── templates/
│   ├── NOTES.txt
│   ├── _helpers.tpl
│   ├── deployment.yaml
│   ├── ingress.yaml
│   ├── service.yaml
│   ├── tests/
│   └── test-connection.yaml
└── values.yaml
```

<sup>1</sup>The Chart Template Developer's Guide [https://helm.sh/docs/chart\\_template\\_guide/the-chart-template-developer-s-guide](https://helm.sh/docs/chart_template_guide/the-chart-template-developer-s-guide)  
<sup>2</sup>The Chart Template Developer's Guide [https://helm.sh/docs/chart\\_template\\_guide/the-chart-template-developer-s-guide](https://helm.sh/docs/chart_template_guide/the-chart-template-developer-s-guide)

```
3 directories, 8 files
```

#### 4.1.2 Templateファイルの作成

template配下の全てのファイルを削除します。これはフルスクラッチで作るためです。合わせてvalues.yamlの中身を初期化します。

```
# templates配下のファイルを削除
$ rm -rf mychart/templates/*

# values.yamlを初期化
$ echo "" > mychart/values.yaml
```

全てのtemplateファイルを削除できましたね。それでは最初からtemplateファイルを作成していきます。

まずはConfigMapを作ります。ConfigMapは環境変数や設定情報を定義するのに利用する基本的なリソースです。key「myvalue」にvalue「Hello World」を定義します。

```
リスト4.1: mychart/templates/configmap.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: mychart-configmap
data:
  myvalue: "Hello World"
```

templatesディレクトリに配置したファイルは、Helmコマンド実行時にテンプレートエンジンに送られます。今回は何も変数化していないため、Tillerがこのtemplateを読んでも、そのままの値がKubernetesに読み込まれます。

この単純なtemplateだけでも、すでにインストール可能なChartになっています。次のコマンドでChartをインストールできます。もちろんインストールしても、ConfigMapがインストールされるだけで他には何も起こりません。

```
$ helm install ./mychart
NAME: winsome-wombat
LAST DEPLOYED: Sun Mar 24 20:23:04 2019
NAMESPACE: default
STATUS: DEPLOYED

RESOURCES:
==> v1/ConfigMap
```

## <<目次>>

### 第1章 Helmの概要・基本構成

- 1.1 具体例で考えるパッケージマネージャーの変遷
- 1.2 Helmの特徴・構成

### 第2章 Kubernetes クラスタ構築・Helm インストール

- 2.1 Kubernetes クラスタの構築の選択肢
- 2.2 Google Kubernetes Engine の構築
- 2.3 Helm のインストール

### 第3章 Helm でアプリケーションをデプロイ

- 3.1 Helm コマンド
- 3.2 MySQLとPrometheus/GrafanaをHelmでインストールする

### 第4章 Helm Chartを自作しよう

- 4.1 Chart 作成チュートリアル
- 4.2 GoアプリケーションをChart化する

### 第5章 Helm Chartを発展させよう

- 5.1 SubChartsとは
- 5.2 Happy Helming ChartをSubChart化する

### 付録A コマンドチートシート

### 付録B Chart用変数チートシート

### 付録C Chart用Sprig Functions チートシート

## <<著者紹介>>

磯 賢大

システムインテグレーターで、アプリケーションエンジニアとして 4 年間業務システムの構築に従事する。2019 年 5 月より転職し、インフラエンジニアとして基盤構築に携わる。最近の趣味は CNCF のプロダクトに触れること。

## <<販売ストア>>

電子書籍:

Amazon Kindle ストア、楽天 kobo イーブックストア、Apple Books、紀伊國屋書店 Kinoppy、Google Play Store、honto 電子書籍ストア、Sony Reader Store、BookLive!、BOOK☆WALKER

印刷書籍:

Amazon.co.jp、三省堂書店オンデマンド、honto ネットストア、楽天ブックス

※ 各ストアでの販売は準備が整いしだい開始されます。

※ 全国の一般書店からもご注文いただけます。

## 【インプレス R&D】 <https://nextpublishing.jp/>

株式会社インプレスR&D(本社:東京都千代田区、代表取締役社長:井芹昌信)は、デジタルファーストの次世代型電子出版プラットフォーム「NextPublishing」を運営する企業です。また自らも、NextPublishing を使った「インターネット白書」の出版など IT 関連メディア事業を展開しています。

※NextPublishing は、インプレス R&D が開発した電子出版プラットフォーム(またはメソッド)の名称です。電子書籍と印刷書籍の同時制作、プリント・オンデマンド(POD)による品切れ解消などの伝統的出版の課題を解決しています。これにより、伝統的出版では経済的に困難な多品種少部数の出版を可能にし、優秀な個人や組織が持つ多様な知の流通を目指しています。

## 【インプレスグループ】 <https://www.impressholdings.com/>

株式会社インプレスホールディングス(本社:東京都千代田区、代表取締役:唐島夏生、証券コード:東証1部9479)を持株会社とするメディアグループ。「IT」「音楽」「デザイン」「山岳・自然」「旅・鉄道」「学術・理工学」を主要テーマに専門性の高いメディア&サービスおよびソリューション事業を展開しています。さらに、コンテンツビジネスのプラットフォーム開発・運営も手がけています。

## 【お問い合わせ先】

株式会社インプレス R&D NextPublishing センター

TEL 03-6837-4820

電子メール: [np-info@impress.co.jp](mailto:np-info@impress.co.jp)