

2020年2月25日  
株式会社インプレスR&D  
<https://nextpublishing.jp/>

クラウドネイティブな継続的デリバリーを実現する！

## 『Spinnaker 入門』発行

技術の泉シリーズ、2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、『Spinnaker 入門』（著者：大澤 秀一）を発行いたしました。

最新の知見を発信する『技術の泉シリーズ』は、「技術書典」や「技術書同人誌博覧会」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。

### 『Spinnaker 入門』

<https://nextpublishing.jp/isbn/9784844378464>



著者：大澤 秀一

小売希望価格：電子書籍版 1600 円（税別）／印刷書籍版 2100 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 138 ページ

ISBN：978-4-8443-7846-4

発行：インプレス R&D

### <<発行主旨・内容紹介>>

Spinnaker（スピネーカー）は OSS の継続的デリバリーツールで、AWS、GCP などのマルチクラウド、Kubernetes、VM、PaaS といったマルチプロバイダに対応しています。

本書では AWS、GCP で Spinnaker 環境を構築して Kubernetes、Google App Engine をデプロイフローを構築する方法や、Spinnaker の機能の一つである、カナリーデプロイについて解説した入門書です。

（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。）

# Spinnaker の概要とインストールを丁寧に解説

## 第2章 Spinnakerのインストール

本章ではSpinnakerのインストール方法を紹介します。従来はHalyardのみで設定していくのが一般的ですが、今回はGCPが正式に提供している「Spinnaker for GCP」でインストールします。この方法だとCloud Shell上で指定されたコマンドを実行するだけで、プロダクション準備のSpinnakerをインストールできるためオススメです。  
 本章からGCPアカウントが必要になります。お持ちでなければ付録A「GCPアカウントの作成」を参考に作成してください。

### 2.1 GCPプロジェクトの作成

まず最初に、GCPプロジェクトを作成して環境を用意します<sup>1)</sup>。すでにGCPプロジェクトがある場合はそれを使っても構いませんが、使い終わった後にリソースをいちいち削除するよりGCPプロジェクトを閉じたほうが早いかつ間違いないため、用途に合わせてGCPプロジェクトを作成することをオススメします。  
 PCブラウザからマネジメントコンソールを開きます。

<https://console.cloud.google.com/>

プロジェクト選択欄をクリックし、「新しいプロジェクト」をクリックしてください(図21)。任意のプロジェクト名を入力して「作成」をクリックしてください。プロジェクトIDは入力したプロジェクト名から生成されますが、GCP全体で重複は許されないため、従来のような名前およびIDにしてください(図22)。

図21: プロジェクトの新規作成



図22: プロジェクト名の入力

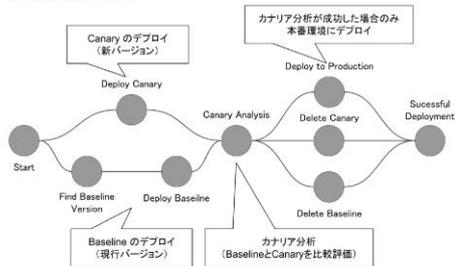


この本ではプロジェクト名およびIDを「techbookfest7-spinnaker」とします。数分待機すると新しいGCPプロジェクトが利用できるようになります。プロジェクト選択欄から先ほど作成したプロジェクトを選択します。

<sup>1)</sup> 正しい環境を回すことは継続的インテグレーションでなくてはなりません。

# GKE, GAE のデプロイと Canary Release の解説を掲載

図5.7: Canary Release パイプライン



- ・トリガー: 第3章と同様に Docker レジストリによるトリガー
- ・Find Baseline Version: ベースライン (現行バージョン) のイメージを特定する
- ・Deploy Baseline: ベースラインのデプロイ
- ・Deploy Canary: カナリア (新バージョン) のデプロイ
- ・Canary Analysis: カナリア分析
- ・Deploy to Production: カナリア分析が成功した場合のみ本番環境にデプロイ
- ・Delete Baseline: ベースラインを破棄
- ・Delete Canary: カナリアを破棄
- ・Successful deployment: Deploy to Production, Delete Canary, Delete Baselineの各ステージが正常に実行されたかどうかを確認

### 5.3.1 Find Baseline Version

ベースライン (現行バージョン) の Docker イメージを特定するステージです。  
 ・Type: Find Artifacts From Resource (Manifest)  
 ・Stage Name: Find Baseline Version  
 ・Depends on: なし  
 「Find Artifacts From Resource (Manifest) Configuration」では表5.3のように入力します。

表5.3: Find Artifacts From Resource (Manifest) Configurationの設定項目

項目名	入力値	説明
Account	spinnaker-1-acct	Kubernetes アカウント
Namespace	guestbook	Kubernetes の名前空間
Kind	deployment	リソース種別
Selecter	Choose a static target	ターゲットの指定方法
Name	frontend	リソース名

「Produces Artifacts」では「Add Artifact」をクリックして表5.4のように入力します。Baselineの Docker イメージのアーティファクト名を「frontend-baseline」とし、後続のステージで利用します。

表5.4: Produces Artifactsの設定項目

項目名	入力値	説明
Kind	Docker	アーティファクトの種類
Docker image	index.docker.io/obrowoff15/gb-frontend	Docker イメージ
Display name	frontend-baseline	後続ステージで表示する表記名

### 5.3.2 Deploy Baseline

先ほどのステージで特定した Docker イメージをもとにベースラインをデプロイするステージです。  
 ・Type: Deploy (Manifest)  
 ・Stage Name: Deploy Baseline  
 ・Depends on: Find Baseline Version

### Deploy (Manifest) Configuration

「Basic Settings」は次のように入力します。  
 ・Account: spinnaker-1-acct  
 ・Application: guestbook

### Manifest Configuration

Manifest Source を「Text」に選択して、techbookfest7-spinnaker-sample/guestbook/frontend-baselineの内容を貼り付けます。metadata.name を frontend-baseline にすることを除けば第3章の frontend-deployment.yaml と同じ内容です。  
 「Req. Artifacts To Bind」では frontend-baseline を選択します。これは「Find Baseline Version」ステージで特定した Docker イメージとなります。

### 5.3.3 Deploy Canary

カナリア (新バージョン) をデプロイするステージです。  
 ・Type: Deploy (Manifest)

## 第8章 GCPにおけるGitOps継続的デリバリー

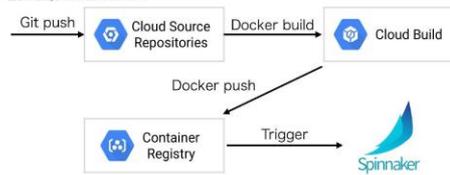
本章ではGitOpsを前提として継続的デリバリーパイプラインをGCPで作成する方法について説明します。

GitOps<sup>1)</sup>とは分散バージョン管理システムであるGit<sup>2)</sup>を起点としたCI/CDワークフローを行う方法です。ソースコードやKubernetesのマニフェストに変更が加わりGitリポジトリで変更が検知されると、テスト実行やビルドなどのCIやSpinnakerなどを用いたCDが自動的に実行される流れになります。Gitによってソースコードやマニフェストを管理することで、開発者は手動でマニフェスト適用やデプロイをする必要がありません。また、Gitの変更からCI/CDまでの流れがすべて自動化されているため、Gitリポジトリと本番サーバーにあるアプリケーションやKubernetesは常に同じ状態になっています。そのため、Gitリポジトリにあるソースコードやマニフェストを見れば稼働中のアプリケーションの設定が把握できるという利点があります。

本章で構築するフローの流れは図8.1の通りです。

1. Cloud Source RepositoriesのGitリポジトリへのソースコード変更を検知
2. Cloud BuildによってDockerイメージをビルドし、Google Container Registry (以下、GCR) にDockerイメージがプッシュされる
3. SpinnakerがGCRのDockerイメージプッシュを検知しGKEへのデプロイが実行される

図8.1: GitOpsによるCDパイプライン



<sup>1</sup> <https://www.aws.amazon.com/technology/gitops/>  
<sup>2</sup> <https://git-scm.com/>

### 8.1 Cloud Source Repositoriesの設定

Cloud Source Repositories<sup>3)</sup>はGCPによるフルマネージドのプライベートGitリポジトリです。Cloud Build、App Engine、Stackdriver、Cloud Pub/Subなどの他のGCPツールに接続して、Gitワークフローを拡張することができます。

Cloud ShellでCloud Source Repositoriesで新しいリポジトリを作成します。

```
$ gcloud source repos create guestbook
```

GitHubのkubernetes/examplesリポジトリをCloneしexamples/guestbookのみをCloud Source Repositoriesで管理できるようにします。

```
# 既存設定の読み込み
$ source ~/spinnaker-for-gcp/scripts/install/properties

# GCPプロジェクトの指定(すでに作成したGCPプロジェクトになっていればこのコマンドは不要)
$ gcloud config set project $PROJECT_ID

$ git clone https://github.com/kubernetes/examples.git
$ mv examples/guestbook ./
$ cd guestbook
$ git init && git add .
$ git commit -m "Initial commit"
$ git remote add origin https://source.developers.google.com/p/$PROJECT_ID/r/guestbook
$ git push -u origin master
```

### 8.2 Cloud Buildの設定

Cloud Build<sup>4)</sup>はGCPによるフルマネージド型のビルドサービスです。このサービスでソースコードのコンパイル・ビルド、テスト(CI)実行などを簡単に設定することができます。ビルド実行にかかった時間分しか料金が発生しないため、ビルド専用のGCEインスタンスを保持しておくよりコストパフォーマンスが高いのが特徴です。

Cloud Buildを初めて使う場合はAPIを有効化します。

<sup>3</sup> <https://cloud.google.com/source-repositories/>  
<sup>4</sup> <https://cloud.google.com/build/>

## <<目次>>

- 第1章 Spinnakerとは
- 第2章 Spinnakerのインストール
- 第3章 Google Kubernetes Engineのデプロイ
- 第4章 Google App Engineのデプロイ
- 第5章 Canary Release
- 第6章 Amazon EKSにSpinnakerをインストール
- 第7章 Jenkinsとの連携
- 第8章 GCPにおけるGitOps継続的デリバリー
- 第9章 AWSにおけるGitOps継続的デリバリー
- 付録A GCPアカウントの作成
- 付録B AWSアカウントの作成、初期設定

## <<著者紹介>>

大澤 秀一

Webエンジニアとして携帯電話向けコンテンツサイトの開発・運用、オンプレミスからAWSへの移行、AWSを活用したWebサービスのインフラ構築・運用業務を経て、2015年より表参道にあるIT企業のインフラエンジニアとして、インフラ運用や開発基盤の改善、AWSとGCPとのマルチクラウド化、クラウドコスト最適化の提案・実施を行っている。

Twitter、GitHub アカウント: @ohsawa0515

技術ブログ: <https://blog.jcoman.info/>

## <<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)