

IBM Cloudのツールチェーンで DevOps

クラウド・アプリケーションの開発・運用効率を劇的に向上させる

ある調査会社のIT市場予測によると、2017年から2020年までの年平均の市場規模成長率は、従来型ITがおおよそマイナス3%であるのに対し、クラウドはおおよそプラス12%となっています[1]。この予測から、エンタープライズ領域におけるクラウドの適用が急激に進行していることが推察できます。こうしたクラウド活用の広がり背景には、コスト削減効果への期待に加えて、デジタルネイティブ・カンパニーとの競争に不可欠な“アジリティーの確保”が影響していると考えられます。ITライフサイクルにおけるアジリティーの実現に欠かせない取り組みがまさにDevOpsであり、「IBM Cloud」ではそのためのツール群を統合したツールチェーンを提供しています。クラウド活用の現場では、DevOpsは今や当たり前の取り組みとなっています。

本コラムでは、IBM CloudでDevOpsが簡単にできることを、実際の利用手順とともにお伝えします。

1. 方法論とツールでクラウドネイティブな開発を支援する

IBMはこれまでRUP(Rational Unified Process)やDAD(Disciplined Agile Delivery)といったエンタープライズ・システムの開発におけるアジリティー向上や各種リスク低減のための開発方法論を体系化し、ツールとともに提供してきました。そしてクラウドネイティブなアプリケーション開発のための方法論として「IBM Garage Method」(以下、Garage Method) [2]を体系化し公開しています(技術解説「クラウド時代の新しいアプリケーション開発」36ページ参照)。これは、単なるアプリケーションの開発方法論ではなく、スタートアップ企業のようなスピードとイノベーションを実現すべく革新的なアイデアを創発し、それをすばやくITで具現化し、その可否をユーザーに問うための方法論です。DevOpsの取り組みはGarage Methodの一部であり、ツールチェーンはGarage Methodの各プラクティスの実践に適したツール(開発者から高い支持を得ているOSSなど)を簡単に統合し、IBM Cloud上におけるGarage Methodの実践を推進するものです。Garage Methodとツールチェーンのイメージを図1に示します。

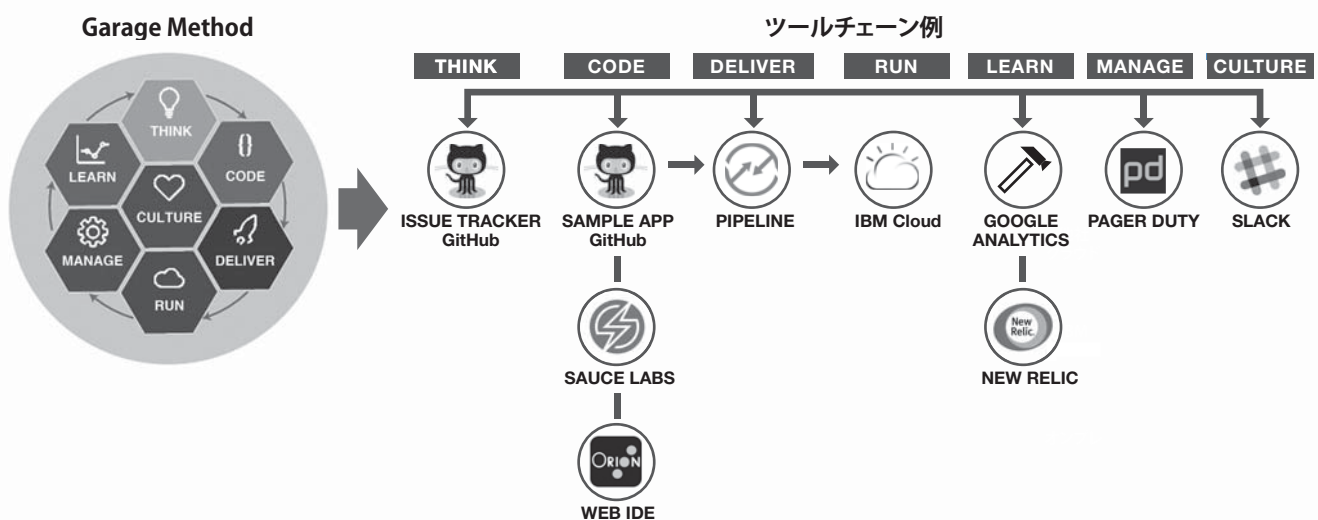


図1. IBM Garage Methodとツールチェーン

2. 開発からデプロイまでに必要なツールを楽々構成する

ここからは、実際にIBM Cloudでツールチェーンを使う手順を紹介します。

まず、IBM Cloudが提供するアプリケーションの稼働環境とそれぞれの概要について図2、表1に示します。ツールチェーンはこれら稼働環境の中でも特に、さまざまなサービスを組み合わせて迅速な開発を行うのについてつけであるCloud Foundryでのアプリケーション開発で威力を発揮します。

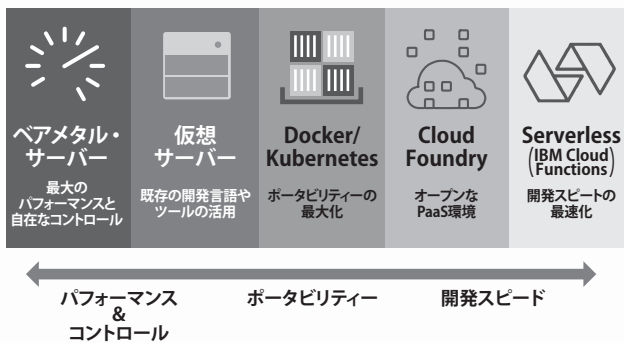


図2. IBM Cloudのアプリケーション稼働環境

表1. アプリケーション稼働環境の概要

稼働環境	Cloudサービスタイプ	概要
ベアメタル・サーバー	IaaS	物理サーバー。CPU、メモリー、OS、内蔵Disk、GPU、ネットワークポートなど任意に構成することが可能。VMwareなどのハイパーバイザーの利用も可能。
仮想サーバー	IaaS	5種類のFlavorからモデルを選択するいわゆるクラウドのサーバー。
Docker/Kubernetes	PaaS	コンテナ・テクノロジーのデファクト・スタンダードとその運用管理基盤。デファクトゆえにポータビリティが高い。
Cloud Foundry	PaaS	OSSのPaaS基盤。Docker/Kubernetesとは異なるコンテナ・テクノロジーを持つ。魅力的なサービスを簡単に使える仕組みが整っている。
Serverless	FaaS (Function as a Service)	イベント駆動型で短時間で完結する処理の実行機能を提供。実行環境に関して一切考慮する必要がないことが名前の由来。

それでは、ツールチェーンを使ってみましょう。ツールチェーンを使うまでの手順は、①IBM Cloudにログインする、②Continuous Deliveryサービスを作成(有効化)する、③必要なだけ(つまりプロジェクトごとに)ツールチェーンを作成する、となります。

①IBM Cloudにログインする

IBM Cloudへは“https://console.bluemix.net/”からログインします。

IBM Cloudのアカウントを持っていない場合は、“https://console.bluemix.net/registration/free”からIBM Cloudを無料で試せるライト・アカウントの登録をしてすぐに試すことができます。

②Continuous Deliveryサービスを作成(有効化)する

2-1. Continuous Deliveryサービスを見つける

画面上部の「カタログ」をクリックし、検索ボックスに‘conti’と入力すると、Developer ToolsカテゴリにあるContinuous Deliveryサービスを見つけることができます。「Continuous Delivery」をクリックします(図3)。



図3. カタログ画面

2-2. Continuous Deliveryサービスを作成する

次に、Continuous Deliveryサービスの作成画面が表示されるので、表示内容を確認し、「作成」ボタンをクリックします。サービス名、ロケーション、リソースグループが指定・選択できますが、お試し目的の場合にはデフォルトで表示・選択されているもので問題ありません(図4)。

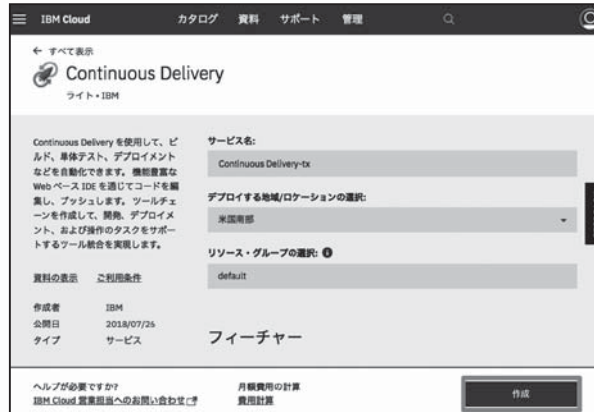


図4. Continuous Deliveryサービスの作成画面

これでツールチェーンを作成する準備ができました。ツールチェーンはContinuous Deliveryサービスを作成(有効化)することで、初めて使えるようになります。一度Continuous Deliveryサービスを作成すると、そのサービス下で複数のツールチェーンを作成できるようになります。

③ ツールチェーンを作成する

3-1. ツールチェーンを作成する

画面左上のハンバーガーメニューをクリックして「DevOps」を選択するとツールチェーン画面が表示されます。図5左の画面は筆者の環境のためすでに複数のツールチェーンが表示されていますが、初めて作成する場合には「ツールチェーンの作成」ボタンのみ表示されています。ボタンをクリックすると、ツールチェーンのテンプレート画面が表示されます。



図5. ツールチェーン画面

ここで「Cloud Foundryアプリの開発」をクリックすると、作成されるツールチェーンの概要と含まれるツールの統合に関する設定画面が表示されます。デフォルトで統合されるツールは、「Gitリポジトリ」とブラウザで使える開発ツールの「Eclipse Orion Web IDE」、そして「Delivery Pipeline」の3つです。

図6真ん中の画面でGitの設定を見てみると、ソース・リポジトリのURLに「https://github.com/open-toolchain/node-hello-world」が指定されており、リポジトリ・タイプは「クローン」となっています。これはGithubというインターネット上の公開リポジトリに格納されているこのテンプレート用のサンプルコードを、自分用にIBM Cloud上のリモート・リポジトリとして複製(クローン)することを意味します。

Eclipse Orion Web IDEについては、改めてここで設定する項目はありません。

次に図6右の画面のDelivery Pipelineの設定を見てみると、API Keyの入力フィールドから下のフィールドが赤線表示されています。API KeyはDelivery PipelineがアプリケーションをCloud Foundry環境にデプロイする際に必要ですので、「作成+」ボタンをクリックします。するとAPI Keyが自動設定されるとともにその下のフィールドも自動的に読み込まれ、赤線表示が解消されます。これでツールチェーンの設定ができましたので、画面最下部にある「作成」ボタンをクリックします。



図6. ツール統合の設定画面

3-2. ツールチェーンが作成されると同時に、最初のビルド&デプロイが自動実行される

しばらくすると作成されたツールチェーンの画面が表示されます。「Delivery Pipeline」をクリックすると、自動的にサンプルのアプリケーションのビルドとデプロイのプロセスが実行されていることが確認できます。

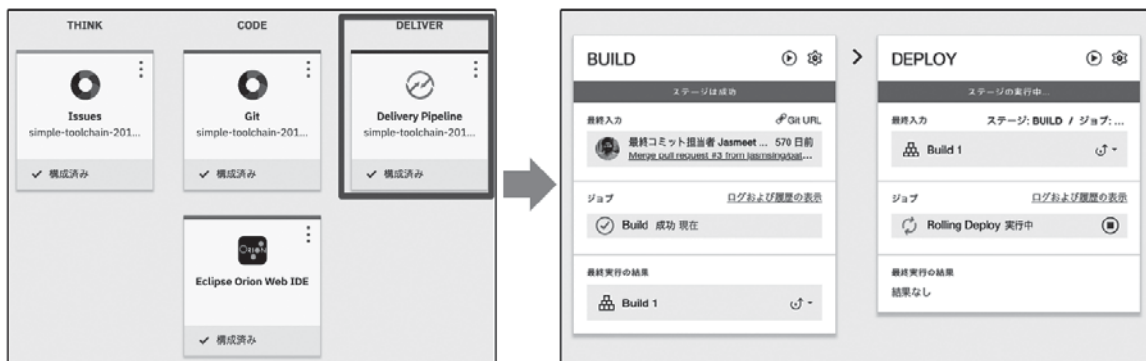


図7. ビルドとデプロイの画面

再びツールチェーンの画面に戻り、「Eclipse Orion Web IDE」をクリックすると、開発ツール画面が表示されます。図8上部中央にアプリケーションの実行状態が表示されます。緑色の○が表示され実行中になったら、「デプロイされたアプリケーションを開く」ボタンをクリックします。すると「Hello world!」というメッセージのWebページが表示されます。



図8. アプリケーションの実行画面

ここまでで最初のアプリケーションがビルドされCloud Foundry上にデプロイされ、きちんと動作していることが確認できました。時間はわずか10分程度です。

3-3. おまけ:アプリケーションに変更を加えた場合のツールチェーン動作

アプリケーションに変更を加えた場合のツールチェーンの動作を確認することで、開発者の作業によってDevOpsのサイクルが回ることを実感できます。

“Hello world!”が“こんにちは、世界!”と表示されるように、index.htmlの一行を修正して変更内容をコミットし、リモート・リポジトリにプッシュします。その後、3-2に示したDelivery Pipelineを確認すると、先程と同様にビルドとデプロイが自動実行されていることが確認できます。



図9. アプリケーションの変更

ここまで見てきたツールチェーン作成の一連の動作を整理したのが図10です。

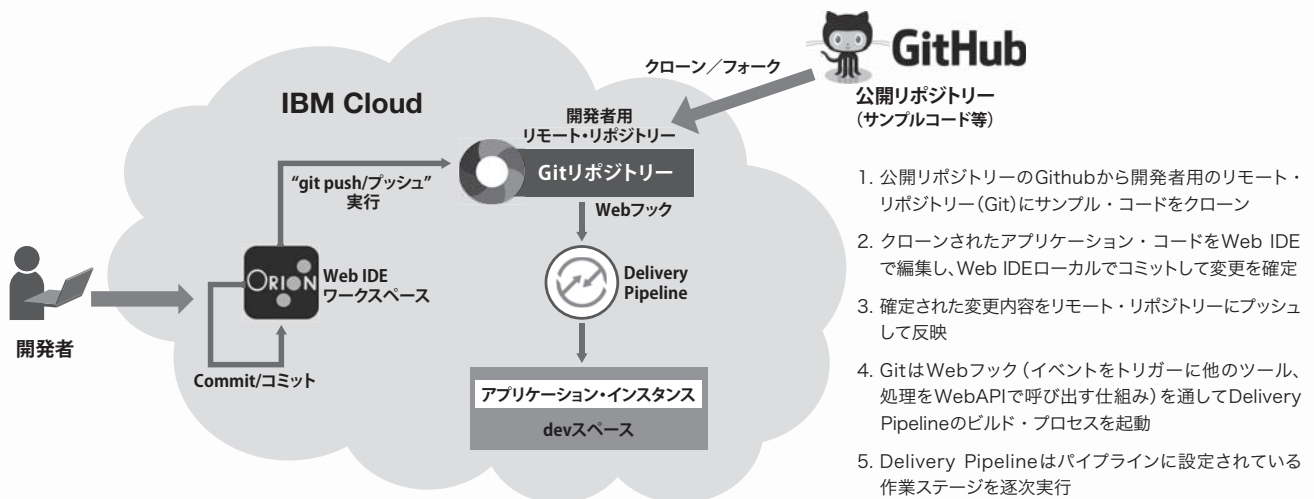


図10. ツールチェーンによる開発からデプロイまでの流れ

ここから分かるように、アプリケーションが実際にデプロイされて動くようになるまでに人が行ったことは、開発者がコードを編集してコミット、プッシュしたのみです。それ以降の処理はすべて自動実行されアプリケーションが公開されます。最もシンプルなDevOpsサイクルがツールチェーンによって自動実行される様子が理解できたかと思います。

3. プロジェクト・ライフサイクルとともにツールチェーンも成長させる

ここまで見てきたものは非常にシンプルなツールチェーンであり、Delivery Pipelineも然りです。実際のエンタープライズ・アプリケーションの開発では、テストやコミュニケーションのためのツールを活用したり、アプリケーションのデプロイ環境は開発、テスト、本番といった複数環境を持つのが一般的です。ツールチェーンはそうしたニーズにも簡単に対応できます。

● ツールチェーンへのツールの統合

ツールチェーンの画面にある「ツールの追加」ボタンをクリックすると、現在開発者から大きく支持されている20を超えるツールをツールチェーンに組み込むことができます。またここに表示されていないツールであっても、Webフックに対応しているツールであればツールチェーンへの統合が可能です(図11)。



図11. ツールの統合

● 自動実行ステージとジョブの追加

さらにDelivery Pipelineにおける処理として、テスト実行や複数環境へのデプロイの自動実行も簡単に設定できます。例えば、開発環境へのデプロイが終わったら機能テストを自動実行し、テストが正常に終了したら統合テスト環境にデプロイしてUIテストを自動実行、それがOKであれば本番環境にデプロイするといったことが簡単に設定できます(図12)。



図12. ジョブの追加

ツールチェーンを使いだした段階ではシンプルなところから始めて、プロジェクトの進行とともに、ツールチェーンを基盤とした開発スタイルの習熟度が上がってきたから新たなツールを統合する。自動実行させるテストは、開発初期段階ではスモークテストのみを行い、アプリケーションの形ができあがってくるに従ってUIテスト、統合テストと増やしていく。このように、プロジェクトのライフサイクルに合わせて適切なツールチェーンを設定・利用することが、無理なくDevOpsを実践するための重要なポイントです。

免許取り立ての初心者がスーパーカーを乗りこなせないように、最初から完成形のツールチェーンを目指す并使用こなす前に挫折することにもなりかねません。肩の力を抜いて、シンプルなツールチェーンから始めてみることをお勧めします。

[参考文献]

- [1] IDC : Share of Spending on IT Infrastructure Continues to Shift to Cloud IT Environments at an Accelerated Rate in 2017, According to IDC, https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=6&cad=rja&uact=8&ved=2ahUKEwiYm8rf47ndAhVGUbwKHVhMALMQFjAFegQICRAC&url=https%3A%2F%2Fwww.idc.com%2Furl.do%3Furl%3D%2Fincludes%2Fpdf_download.jsp%3FcontainerId%3DprUS43508918%26position%3D34&usq=AOvVaw3rPIXNd-igXO4DVcyMQozH
- [2] IBM Garage Method, <https://www.ibm.com/cloud/garage/>



日本アイ・ビー・エム株式会社
IBMクラウド事業本部
Watson & Cloud Platform テクニカルセールス
コンサルティング・テクニカル・スペシャリスト

今村 智
Satoshi Imamura

金融情報サービスベンダーにて開発者として6年間、その後SlerにてPM、フレームワーク・アーキテクト、開発ツールベンダーにてアジャイルを中心とした開発プロセス・コンサルティングに従事した後、ITベンチャーCTOなどを経て日本IBMに入社。2015年よりIBMのクラウド事業の展開に従事。