

Kubernetes のマニフェスト作成ツール Kurouching の開発

19T312 三枝 泰士 (最所研究室)

1 はじめに

急速に変化する IT 社会に対応するためには短期間でソフトウェアにアップデートを重ねることが重要である。しかし、リリース作業を行うための学習コストや作業負担は高く、リリースの頻度が上がるにつれ、その負荷はより高くなる。コンテナの運用管理プラットフォーム Kubernetes は、設定ファイルであるマニフェストを記述すれば、システムの構成は自動で行うことができることから、リリース作業をより容易に行うことができる。しかし、Kubernetes を使いこなすためには、その複雑な仕組みやそれを扱うための複雑なルールなどの学習が必要となる。

本研究では、Kubernetes におけるコンテナの作成のためのマニフェストをより小さな学習コストで作成するシステム Kurouching を開発した。

2 課題

筆者は Kubernetes を用いて BEYOND というシステムをリリースした。BEYOND とは、当研究室で開発している、脆弱性対策のためのセキュリティ保護システムである。この経験から、以下の課題がわかった。

① Kubernetes の学習が困難

マニフェストには、リソースと呼ばれるコンテナを管理するための部品の定義を行う。リソースの種類や設定項目は非常に多く、それらの意味を理解しつつ適切な値を定義することが求められる。

② 細かいルールに則った作業によるミスが頻発

マニフェストをエラーなく記述するためには、リソースごとに、押さえなければならぬ記述ルールの理解が求められる。タイプミスなどの小さなミスは、マニフェストを Kubernetes に適用する際に発生するエラーが出るまでは気づきにくい。

③ 設定値の改善が困難

設定値の改善のためには、Kubernetes についての理解を基にした、ベストプラクティスの理解が必要である。例えば、コンテナを物理マシン間で分散して配置すると、冗長性を持たせることができる。しかし、その設定を記述するためには Kubernetes のノード (物理マシン) や Pod (コンテナの動作環境) などの概念や Pod の選択方法など多くの事項の理解が求められる。

3 設計・実装

システムの構成を図 1 に示す。

課題を解決するために、要件を以下に決定した。

① 入力項目数の軽減 (課題 ①)

開発者は、開発したアプリケーションの要求事項を Kurouching に少量入力するだけで、リリースができる状態のマニフェストを作成する。マニフェストに必要な、運用の知識を要する設定項目などは Kurouching が補完する。例えば、コンテナ名とコンテナイメージ名を入力するだけで、コンテナを順次更新するなどの運用の設定も含めたマニフェストを出力する。また、マニフェストの生成

時に、エラーがあるか Kurouching がチェックする。

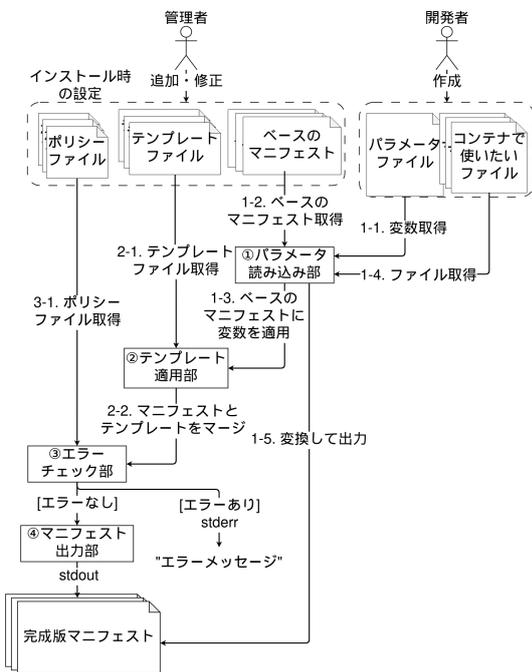


図 1: 構成

② 構成の部品化 (課題 ②)

マニフェストで繰り返し書かれる設定部分を分離しテンプレートファイルという形で部品化する。開発者は、テンプレートファイルの名前を入力するだけで、それに記載されたマニフェストを使用できるため、タイプミスなどの記述ミスを防げる。

③ 良い構成の提供 (課題 ③)

Kurouching で作成するマニフェストは、マニフェストの静的解析ツール kube-score[1] を参考にしたベストプラクティスに沿った設定を提供する。開発者は、Kubernetes や運用に関する知識レベルに関係なく、良い構成でアプリケーションを動作させることができる。

Kurouching は、①パラメータ読み込み部、②テンプレート適用部、③エラーチェック部、④マニフェスト出力部によって動作する。①で

は、開発者によってアプリケーションの要求事項が入力されるパラメータファイルからマニフェストを作成する。②では、部品化した機能であるテンプレートファイルを読み込み、パラメータ読み込み部で作成したマニフェストに適用する。③では、エラー検知ルールを定義したポリシーファイルを読み込み、テンプレート適用部で作成したマニフェストを検査する。④では、エラーチェック部の検査を通過したマニフェストをひとまとめにして出力する。

4 評価

Kurouching にて作成したアプリケーションが、期待した通りに動作するか確認するための機能評価では、Kubernetes 上で動作させられることが確認できた。また、Kurouching と手動でマニフェストを作成した場合の、操作量 (記述行数) と学習コスト (設定項目の種類数) と時間と充実さ (マニフェストの静的解析ツールによるベストプラクティスの違反検知エラー数) の 4 項目を比較した。手動でマニフェストを作成した場合と比べて、操作量は約 1/4 に、学習コストは約 1/2 に、時間は約 1/3 に、減少させられた。充実さは、マニフェストの静的解析ツールによるエラーの出力数の少なさを測定したが、同等もしくは約 1/2 のエラーの出力数となった。

5 まとめ

Kurouching は手動でマニフェストを作成する場合に比べて、より簡単に、短い時間でマニフェストを作成できることが確認できた。今回は筆者が被験者となり実験を行ったが、被験者によって、マニフェストの作成時間などの計測結果にどのような差が出るかは調査の余地がある。

参考文献

- [1] zegl. “kube-score - Kubernetes object analysis with recommendations for improved reliability and security”.
<https://kube-score.com/>. 2023/1/14.