

MicroOpsCI with LC4RI ～Notebook手順書にCIを適用する～

富士通(株) 阿部 秀一 富士通(株) 井浦 陽一郎 富士通(株) 宇野 耕平
 (株)クレスコ 原野 昌幸 (株)日立製作所 山崎 航史

ITインフラの運用における問題点

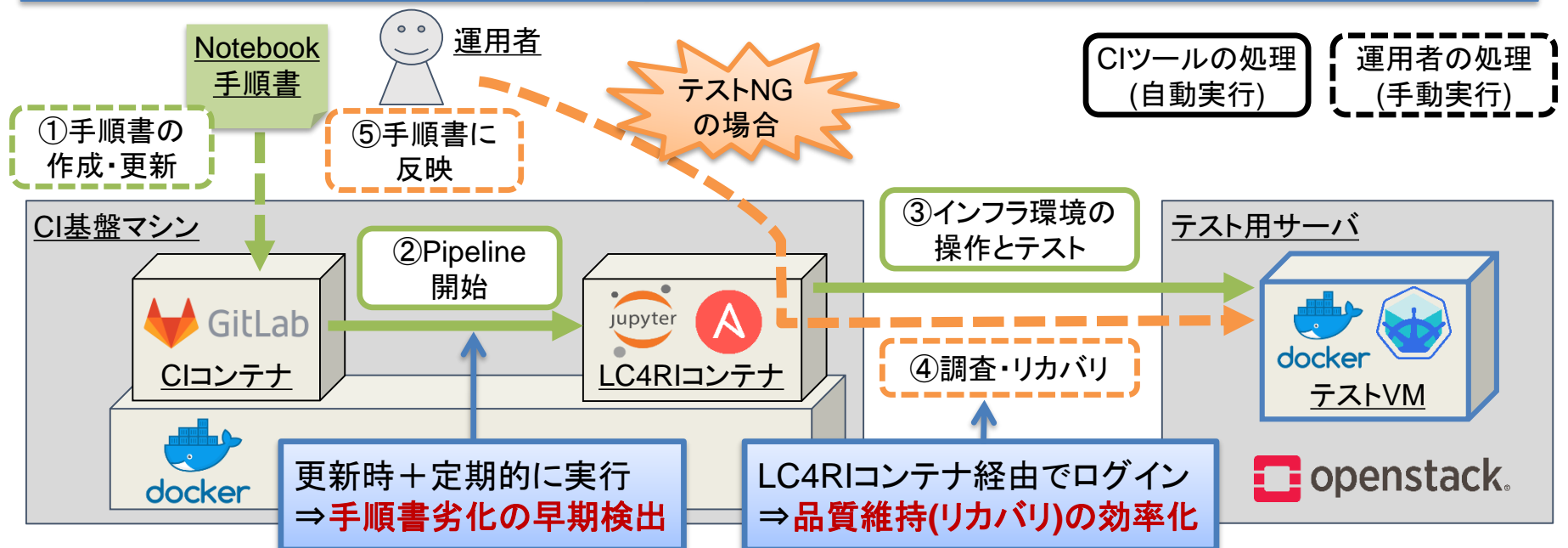
ITインフラの複雑化により, Infrastructure as Code などの自動化が取り組まれている. しかし, 単純な自動化は, 保守性が低く, 属人化が課題である. また, インフラの周辺環境は頻りにアップデートされるため, 自動化コードの品質を維持することは困難であり, エラー発生時の調査とリカバリには高いスキルと多くの工数が必要とされる.

手法・ツールの適用による解決

仕様書とコードを一元管理するLC4RI*に対して, **小さな(Micro)単位の操作(Ops)をテスト対象とする継続的インテグレーション(CI)手法を適用した.** 運用者は, **保守性に優れた実行可能な手順書**を作成し, その**品質を効率的に検証・維持することが可能**である.

[*] <https://literate-computing.github.io>

構築したMicroOpsCI with LC4RIのテスト環境



Notebook手順書

仕様記述 (Markdown)

1.2.1 Dokcerのインストール

2018/11/25現在, CentOS版の最新版**18.09**のDockerに Kubeadmが対応していないため, Minikubeの起動に失敗する. 今回は動作確認済みである**18.06.1-ce**をインストールする. 参考: <https://github.com/kubernetes/minikube/issues/3323>

コード実行

```
!ansible-playbook -l { target } playbook/docker.yml
```

実行結果

```
PLAY [minikube*] *****
TASK [Gathering Facts] *****
ok: [157.1.141.96]
```

テスト実行

1.2.2 Dockerの確認

Dockerのバージョンが**18.06.1-ce**であることを確認

テスト結果

```
!ansible { target } --become -a 'docker version'
157.1.141.96 | SUCCESS | rc=0 >>
Client:
Version: 18.06.1-ce
```

仕様・コード・実行結果を一元管理⇒**保守性の向上**

提案手法の位置付け

自動化

目標到達には
深い溝が...

目標(理想)

人の介入は必要

目標(現実)

Infrastructure
as Code中心の
自動化

**MicroOpsCI
with LC4RI**

LC4RI

属人的な
構築・運用

標準化