

コンテナ仮想化環境を活用したインフラ構築演習環境の提案

橋浦研究室 1155323 津村 隆弥

1. はじめに

現在のインフラエンジニア(以下, IE)を育成するための環境は十分ではない。例えば, OS やミドルウェアを誤った手順でインストール(以下, インフラ構築)した場合, 想定通りに動作しないことが考えられる。この想定通りに動作しない問題を修正するために必要なこととして,

- 1) どのように想定通りにインストールされていないのか確認すること
- 2) 環境をインストールする前の状態に戻すこと

上記の2点が考えられる。しかしながら, これらの作業を一般的な環境はサポートしないため, IE を育成する演習環境は十分ではないと考えた。

1.1. 確認することはなぜ難しいのか

インフラ構築が正しく行われたのか確認するために必要な項目は複数存在し, それぞれ方法は異なる。加えて, 確認に使うコマンドの出力は, 形式が統一されていないため, 学習者には表示された結果が本当に正しいものなのか分からないという問題がある。

1.2. 環境に戻すことはなぜ難しいのか

インフラ構築作業は, 一般的に“tar”や“yum”など専用のコマンドによって行われ, 大量のファイルの書き換えと展開を自動的に実行する。そのため, 学習者は何をすれば前の状態に戻せるのかが分からない。このため, 学習者が演習に専念することができなくなる問題がある。

2. 研究目的

本研究の目的は, 前述したこれら2点の問題に対し, インフラ構築演習環境を提供することで, IE の育成を支援することである。

3. 提案手法

前述の学習者にとって難しいとされる問題を解決するために,

- 1) 手順や設定のどこが誤っているのかについて, 個別・具体的に指摘する機能
- 2) 元の状態に戻すことが難しい状態になってしまった演習環境を, 学習者に代わって復元する機能。

上記の2点を用意することで, 演習環境を提供する。

機能1) に対しては, 教授者が予めインフラ構築演習で達成すべき内容を明確化しておき, 具体的なテストケースとして定義しておく。IE が演習環境を利用する際は自身で設定内容が正しく行われたか確認すると共に, 教授者が用意したテストケースも利用できるようにすることで, 適切なフィードバックを得ることが出来るようにする。

機能2) に対しては, 演習環境をコンテナ型仮想化環境上に用意し, 演習を行う前や教授者が設定したチェックポイントを達成した際に, 自動的に演習を行った環境のバックアップを取るようにする。これによって, 環境を想定通りに構築された以前の状態へ, 容易に復元することが可能となり, IE はインフラ構築の演習に専念することが可能となる。

4. 実現方法

本研究では, IE が操作する対象となる OS を CentOS とし, コンテナ型仮想化環境には Docker[1], 教授者が用意したテストケースを IE が演習を行った環境に対して実行するためのテストツールとして Container Structure Tests[2]を用いた。また, これらを統合するための UI として Java を用いてアプリケーションを開発した。UI の画面の表示例を次ページ図1に示す。

学習者は, 図1中の“チェックする”ボタンを押すことで, 機能1)を実行することができる。このボタンが押されると, ツールはテストを実行する。そして図1中の下部にある結果のテキストボックスへ, テストの可否を表示する。

学習者は、図 1 中の“やり直す”ボタンを押すことで、機能 2) を実行することができる。ツールは現状の環境のコンテナを破棄し、保存されたイメージを元に、前回のチェックポイント達成時の環境を復元する。



図 1 ツールの UI

5. 実験

作成したツールの有効性を検証するため、日本工業大学情報工学科所属の学生 8 人を被験者とする実験を行った。

実験内容は OS 上に Web サーバをインストールし、正しい内容の Web ページが見られる状態にするまでの一連のインフラ構築を行うものである。また、今回用意したテストケースは、パッケージマネージャーの設定ファイルが正しく設定されているか、Web サーバはインストールされているか、設定ファイルは正しい場所に存在し、内容が正しく設定されているか、Web サーバの状態は起動しており、OS の起動時に自動的に起動するようになっているか、などの合計 12 個であった。

評価は前章で述べた機能 1) と機能 2) の有無によって、演習終了時にテストケースの成功、或いは失敗した数がどのように変化するかについて比較する。

6. 実験結果と考察

今回の実験結果の概要を以下の表 1 に示す。ツールによる支援がない場合には、演習終了時に確認漏れと思われる項目が存在したが、ツールによる支援がある場合では、そのような漏れが無く、全て正しく動作する事が確認できた。

表 1 支援なしの時とありの時の比較

	支援なし	支援あり
テスト 成功数	90	96
テスト 失敗数	6	0
合計	96	96

ツールの支援による、テストの成功と失敗の差についてイエーツの補正を用いた χ^2 検定を行った。優位水準 α は 0.05 とし、帰無仮説 H_0 は「ツールによる支援の有無に関わらずテスト成功数は等しい」とした。表 2 に今回の実験での検定表を示す。

表 2 今回の実験での検定表

観測数	8
自由度	1
χ^2 値	4.301075269
p 値	0.038088284

表 2 より、今回 χ^2 検定の結果 $p < 0.05$ であったため、帰無仮説 H_0 は棄却され、ツールによる支援の効果が確認できた。

7. まとめと今後の課題

本研究では IE を育成する環境に必要な正しく出来ているか確認を行い、正しく出来なかった場合、失敗する前の状態を再現するインフラ構築演習環境の開発を行った。そして、提供した演習環境の有効性を検証するために実験を行い、ツールによる支援の効果を示した。

今後の課題としては、環境は正しく動作するかの機能性に加え、セキュリティなどの非機能性についてのテスト機能やアプリケーションのコンテナ化など、演習環境の範囲を拡大することである。

参考文献

- [1] Docker Inc., "Docker," <<https://www.docker.com/>> (Accessed 2018/12/13).
- [2] Google, "Container Structure Tests: Unit Tests for Docker Images," <<https://opensource.googleblog.com/2018/01/container-structure-tests-unit-tests.html>>, 2018/1/9, (Accessed 2018/12/13).