

独自管理インターフェースを持つ未来大クラウドシステムのIaC対応

b1021204 西侑亮

指導教員：松原克弥

FUN Cloud System Which Have Unique Management Interface Implementations of IaC

Yusuke Nishi

概要：今日の多くの大学では、プログラミング実行環境を提供するクラウドシステム導入が加速している。クラウドシステムによる仮想マシンを用いることで、多様な端末上で統一された実行環境を用意できるようになった。しかし、学生人数分だけ、仮想マシンの作成や設定を用意することは、多くの手間や手順が必要となる課題がある。クラウドシステムをコードで管理する手法である Infrastructure as Code(IaC) に対応させることで効率的に管理することが期待できる。本研究では、公立はこだて未来大学（以降、未来大）にて使用されているクラウドシステムをIaCに対応させることにより、効率よく統一環境を用意し、講義の効率化を目指す。

キーワード：クラウドシステム, IaC, 公立はこだて未来大学

Abstract: Nowadays, the number of university is include which use cloud systems for programming develop environment. Virtual machines based on cloud systems can be prepared a unified development environment in deferent machines. However, the problem is that many processes are needed to create and configure virtual machines for the number of students. Using of Infrastructure as code (IaC) that the approach of managing Cloud systems by code can be expected to manage efficiently. The study's goal is to improve seminar efficiency by efficiently constructing a unified development environment for the cloud system used at Future University Hakodate (FUN) that is compatible with IaC.

Keywords: Cloud System, IaC, Future University of Hakodate

1 背景と目的

近年、情報科学分野を専攻とする大学や高等専門学校における授業スタイルが変化したことにより、クラウドシステムを利用して、統一した演習環境が求められている。授業スタイルは、学生人数分のPCを設置して統一された演習環境を用意する授業スタイルから、学生個人がPCを用意するBYOD(Bring Your Own Device)へと変化している [1]。BYODでは、学生が所有するPCのOSや内部設定が異なることから、統一の演習環境が必要な場合には、仮想マシンなどを用いて環境ごとの差異を吸収する必要がある。実際に、未来大や大阪大学では、クラウドシステムを利用して仮想マシンを用いて統一された演習環境を提供している [2]。

クラウドシステムを利用する事例が増加するとともに、仮想マシンの作成や設定などに、授業準備時間や教員の手間が必要となる課題が浮上している。例えば、未来大における情報科学の演

習講義の一つである並列分散処理では、複数台の仮想マシンを用意して一つのプログラムを並列分散実行する演習がある。2023年度に行われた同科目では未来大クラウドシステムでの実行方法やマシンの初期設定について合計10ページにわたる解説が行われている。講義を受講している学生分の、仮想マシンの準備にかかる手順を簡略化できれば、授業準備時間の短縮を見込める。

本研究は、統一した演習環境が必要な授業における、準備にかかる時間の短縮を目指す。実現のために、演習用クラウドシステムのIaC対応について実装・評価を行う。IaCとは、プログラミング言語で書かれたコードを実行することによりITインフラストラクチャを作成管理する手法のことを意味する [3]。Micheleら [4],IaCを用いることで再現性と効率性で利点をもたらすと述べている。再現性での利点とは、コードを配布することで、仮想環境構築の再現と再利用が容易にできることである。また、手作業による環境構築の手順を、簡略化することで効率性が向上する。

コードを配布し実行するだけで実行環境を再現できるため、従来のように多くのスライドを用いて説明する必要がなくなり、授業効率の向上が見込める。

2 未来大クラウドシステムの現状

未来大クラウドシステムでは、IaC 対応に必要な Application Programming Interface(以降,API) が公開されていない。学生の仮想マシンの管理を集約して、一つの Amazon Web Services(以降,AWS) アカウントが仮想マシンを管理しているためである。未来大クラウドシステムでは、学生がアカウント認証を行い、それぞれの仮想マシン管理画面 (WebAPI) にアクセスし、仮想マシンの作成や起動、削除などの管理を行っている。

一見、学生それぞれのアカウントが独立してクラウドシステムを管理しているように見えるが、未来大クラウドシステム内部では全てのアカウントの管理要求を集約し、一つの AWS アカウントが仮想マシンを管理している。そのため、アカウントひとつひとつに対して、個別に API を公開するためには、システム的大幅な改変が不可欠であり、多くの労力と費用を要する。そのため、本研究では、スクレイピングを用いて仮想マシンの管理を行う。

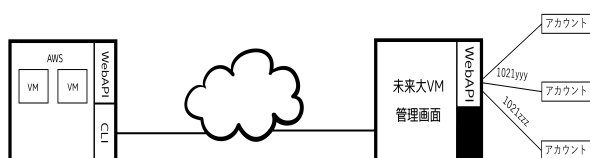


図1 未来大クラウドシステムの構造

3 関連技術

3.1 Terraform

本研究では、IaC フレームワークの一つである Terraform を用いる。Terraform は、デファクトスタンダードである。Terraform は Amazon Web Services(以降,AWS) や Microsoft Azure, Google Cloud Platform などのクラウドプロバイダに対応しているため、Terraform のコードを学習することで多くのプロバイダを同様のコマンドで管理することができる。

Terraform はプロビジョニング機能を備えている。プロビジョニング機能とは仮想マシンの作

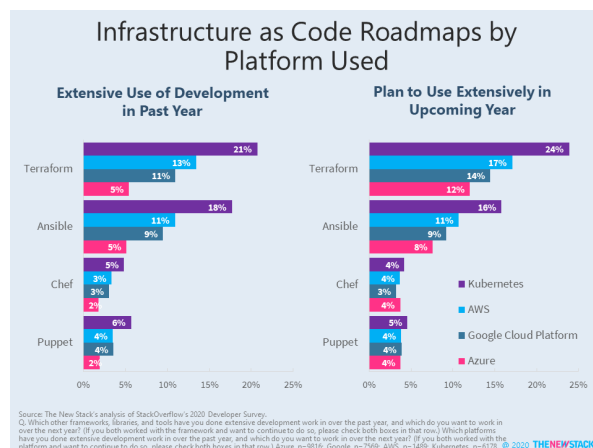


図2 2020年のIaCプラットフォームの利用率比較 [5]

成・起動時に、仮想マシンの環境設定やプログラムの自動実行を行う機能である。Terraformでは仮想マシン作成・起動後にマシンに対してSSH接続を行いプロビジョニング機能を実現している。

Terraform Custom Framework Providerという、新規にクラウドシステムをTerraformに対応させるプログラムが公開されている。Terraform Custom Framework Providerは対応させたいクラウドシステムに合わせて書き換えることでTerraform対応を実現する。作成したTerraform Custom Framework ProviderはTerraformを介して、世界中に公開することも可能である。

3.2 スクレイピング

スクレイピングとは、WEB ページ上の HTML や CSS に存在するタグやデータ構造を解析し、構造化されているデータをプレーンなデータへと抽出し変換する技術である。スクレイピングはプログラムを使用して行われ、目的の Web ページの HTML データを解析し、特定の情報をユーザからのリクエストに応じて自動的に取り出すことを可能にする。スクレイピングは、データ解析、自動化テスト、自動入力など、多様な分野で使用されており、データ収集手段として注目を集めている。ログイン認証が必要なページや、フォーム入力が必要な場合にも、スクレイピング技術は有効であり、ヘッドレスブラウザや専用のライブラリを使用し、自動的にフォームを入力、ログインプロセスを完了させることができる。

本研究は、未来大クラウドシステムをIaC対応化させることを提案する。仮想マシンのコードを用いた管理と、コードの配布によるマシン環境の再現が可能となることで、従来の演習用仮想マシンの準備と比べ、大幅な時間短縮を期待できる。

The diagram illustrates the process of creating a VM in AWS using Terraform. On the left, a box labeled 'AWS' contains 'VM' and 'CLI'. A cloud icon connects this to a box on the right labeled '将来大VM 管理画面' (Future Large VM Management Screen), which also contains 'VM' and 'CLI'. Below the cloud, a red 'X' marks a failed connection labeled '通信エラー' (Communication Error) between 'terraform' and the management screen. A blue line shows the successful path from 'terraform' to the management screen.

5 実装

```
resource "scraping_resource" "example" {
  environment = "Linux(Ubuntu22.04LTS) (2024前期) (10/31廃止)"
  username    = "b1021***"
  password    = "*****"
  machine_name = "EC2-geotail-155163"
  machine_stop = false
  instance_type = "t4g.large"
}
```

スクレイピングを用いて未来大クラウドシステムにアクセスすると、最初にユーザ認証画面へと遷移する。ユーザ情報とパスワードをフォームに自動入力し、ボタをクリックを行うことで環境選択画面に遷移する。環境選択では、年度や時期、受講している講義ごとに異なる環境を選択す

プロビジョニング機能を用いる際、仮想マシンのパスワードや IP アドレス、秘密鍵が必要となる。パスワードや IP アドレスはスクレイピングによって、HTML から自動抽出する機能を実装する。秘密鍵は WebAPI をスクレイピングして自動でダウンロードし、ダウンロード先の絶対パスを返す機能を実装する。

本研究では、スクレイピングを行う処理をクラウドコンピュータ上で行う。クラウドコンピュータ上で行うことにより、手間となる関連モジュールのインストール工程がなくなり、Terraformを実行する環境に左右されにくくなる。

本システムの有用性を確認するために、評価項目として定量的評価と定性的評価による測定を検討している。定量的評価では、システムの使用における実行時間、システムそのもののレスポンス時間や安定性などのパフォーマンスメトリクスを測定する。定性的評価では、未来大の2024年度後期の情報科学演習講義である並列分散処理の受講生を被験者として評価実験を行い、アンケートを通じて被験者の使用感や満足度や要望、システムに対する意見を収集する。

7 進捗と計画

進捗状況として、スクレイピングを用いて Terraform による未来大クラウドシステムの管理機能を実装した。システムを使用するにあたり、関連モジュールをインストールする手間を省くために、クラウドコンピューティング上で実行、実行環境に依存しないシステムの構築を目指している。今後の計画として、クラウドコンピューティング上での実装が終わった後、実際に未来大の学生に使用してもらい、フィードバックを元に機能改善を目指す。

8 結言

本研究では、独自の管理インターフェースを持つ未来大クラウドシステムの IaC 対応により、演習講義における統一環境準備の効率化を目指す。クラウドシステムを直接管理することが困難なインターフェースのため、スクレイピングを用いて管理を行う。今後の計画として、実行者の PC 環境に依存しないように実装し、評価実験を行いたい。

9 情報システムコースにおける本研究の位置づけ

本研究では、スクレイピングにより学内クラウドシステムを IaC に対応させることによって、演習講義に用いる統一環境準備の効率化を目標としている。これは、未来大での演習講義を情報システムによって支援するものであり、効率性と信頼性を考慮した情報システムの実現と言える。今後、実装したシステムを評価実験を行うことで、カリキュラムポリシーの結果の評価を通じて、新しい方法論や学問領域を切り拓く能力を育むことに繋がる。

参考文献

- [1] 大学 ICT 推進協議会: ICT 利活用調査部会, "BYOD を活用した教育改善に関する調査研究 結果報告書", AXIES 大学 ICT 推進協議会.[Online]. Available from : https://axies.jp/report/ict_survey/2016survey/. Accessed: 2024-10-25.
- [2] 大阪大学, "大阪大学キャンパスクラウドサービス", 大阪大学. [Online]. Avail-

able from: <https://ccc.osaka-u.ac.jp/hosting/>. Accessed: 2024-10-21.

- [3] K. Morris: "Infrastructure as Code", O'Reilly Media, 2016.
- [4] Michele Chiari, Bin Xiang, Sergio Canzoneri, Galia Novakova Nedeltcheva, Elisabetta Di Nitto, Lorenzo Blasi, Debora Benedetto, Laurentiu Niculut, Igor Škof : "DOML: A new modeling approach to Infrastructure-as-Code", Information Systems, Volume 125, November 2024.
- [5] B. Cameron Gain: "Terraform 1.0 Reflects What HashiCorp Has Learned About Infrastructure-as-Code", THE-NEWSTACK, [Online]. Available from: https://thenewstack.io/terraform1-0-reflects-what-hashicorp-has-learned-about-infrastructure-as-code/?utm_referrer=https%3A%2F%2Fwww.google.com%2F/. Accessed: 2024-10-22.