Vol.2020-CLE-30 No.2
2020/3/8

情報技術系演習に用いるサーバ群の管理のための kubernetes の機能拡張

小林 優吾[1]　新村 正明[1,a)]

概要：プログラミング等の情報技術系の演習のために，学習者に演習用サーバを提供する方法が広く用いられている．使用するソフトウェアによっては演習用サーバを学習者毎に用意する必要があり，先行研究ではこれを解決するために docker による仮想サーバを kubernetes を用いて構築することを提案した．しかし，kubernetes は大規模なアプリケーションを運用するために作成されており，演習のためのシステムをそのまま構築・運用することは困難である．そのため，これを解消するために kubernetes の機能の拡張をカスタムコントローラの作成という手法で行い，演習のためのシステムに最適な運用を可能にした．

Kobayashi Yugo[1]　Niimura Masaaki[1,a)]

1. 研究背景

現在，コンピュータを用いた演習は，多くの教育機関で利用されており，今後も情報系の科目に限らず利用の拡大が見込まれる技術である．

また，従来の演習では教育機関の用意したコンピュータで演習を行なう形式が一般的だったが，現在では BYOD(Bring Your Own Device) と呼ばれる形式で行なう場合も多い．BYOD とは，各ユーザが所有している私的な端末を職場や学校などで利用する取り組みであり，さまざまな機関で導入されている．しかし，BYOD を演習で利用する場合には各ユーザの端末の環境の異差が問題になる場合がある．このため，先行研究 [1] では演習環境を Web アプリケーションとして提供することで，Web ブラウザを介すことによりユーザの端末の差異への対応を行なった．

また，演習で使用するアプリケーションによってはユーザごとに 1 ホストを占有する物があり，これらのアプリケーションを演習で利用したい場合はユーザの数だけ物理サーバが必要となってしまう．このため仮想化技術を用いて仮想的なホストでアプリケーションを動作させユーザごとに提供する方法を提案した．

本研究は，教授者のような環境を提供する側がサーバ構築などの複雑な作業を行なわなくても，簡単に環境を構築および運用することができるようにすることで授業などでコンピュータを用いた演習を行う難易度を下げることにより，実践的な授業内容を行うことを容易にすることを目的とする．

先行研究ではコンテナ型仮想化技術である docker を用いて演習環境を構築し，kubernetes による運用を行うことを提案したが，kubernetes をそのまま本研究で目的とする演習システムの作成に使用することは，システム全体の複雑化を招き，運用の難易度が高くなりユーザに提供することが困難になる．そこで，新規の演習環境の作成をより容易にするような実装方法を再検討した．

2. 既存演習システムの課題

演習環境全体のことを以後，演習システムと呼ぶ．

実際に演習システムを構築するにあたって，いくつか問題がある．先行研究で行われている手法も含め，**図 1** に示すように，既存の仮想化技術基盤では仮想化した演習環境 1 つ 1 つを管理する必要がる．本研究で目的としているような演習システムの場合にはユーザと同数以上の環境を提供者が管理しなければならず，これは演習システムを運用する上で非常に大きな負荷となる．

そこで本研究では，このユーザごとに利用する仮想サーバ (以後，各ユーザの利用する仮想化され各々分離された，ユーザ専用の環境を仮想サーバと呼称する) の作成や管理を行うプラットフォームを作成し，提供者が個別の演習環

[1] 信州大学
Shinshu University, 4-17-1 Wakasato, Nagano, 380–8553, Japan
[a)] niimura@cs.shinshu-u.ac.jp

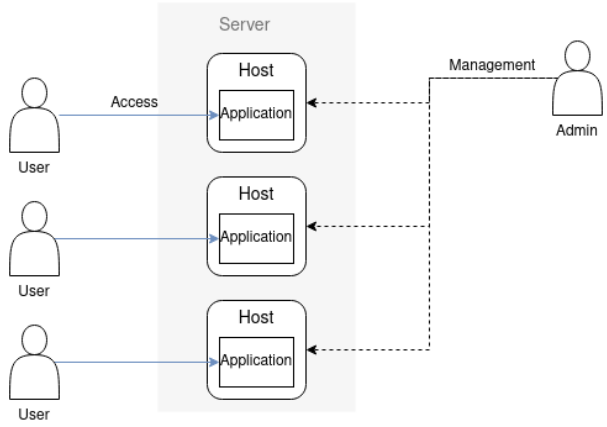

図 1　これまでの仮想化システム
Fig. 1　Virtualization system

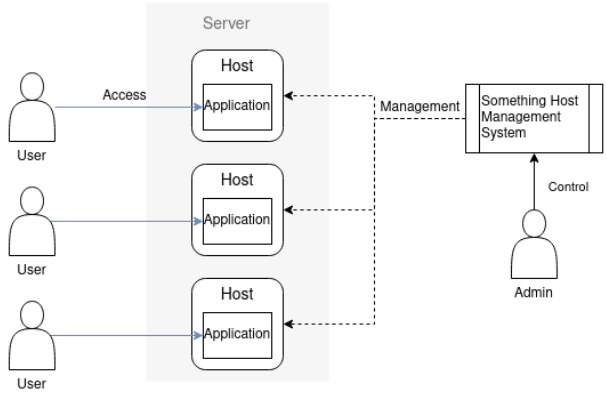

図 2　演習システムの提案
Fig. 2　Proposal system

境の管理を行う負荷を減らすことを目的としたプラットフォームの構築を行なう．この概要を図 2 に示す．また，各ユーザの仮想化した演習環境を制御するシステムを，以後プラットフォームと呼称する．

3. 提案

3.1 必要とされる要素

本研究では提供者が演習環境を提供する際の負荷の軽減を目的としており，そのために演習システムを作成する際に必要とされる要素はいくつか想定されるが，本研究では以下のものに絞った．

- 演習システムの作成を簡単にする
- 演習システムの運用の負荷を下げる

これらの要素を満たすプラットフォームを構築することで，提供者が演習環境の作成や運用を簡単に行う機能を提供することができる．

次に，演習を行う演習システムを構築する際に必要とされる機能の要件を定義する．

3.2 要件定義

ユーザや演習で使用するアプリケーションなどの情報から，仮想サーバの管理を行うプラットフォームを構築する．実際の仮想サーバ自体の管理は既存のソフトウェアを用い，本研究で構築するプラットフォームはそれらを操作することで要件を達成することを目指す．

演習環境を管理するにあたって，必要な機能は以下の物とする．

- 複数サーバへの展開
- ユーザ用環境の自動復旧
- ユーザ用環境の作成，削除
- 認証認可などの，各環境で共通の処理を行う部分

演習においてユーザの人数が増えた場合に，物理的なマシンを複数台にし展開できるようにすることが必要になると考えられる．さらに，仮想サーバ自体に障害が発生した場合などは，その仮想サーバに対して提供者が起動処理などの対応をしなければならない場合があると考えられるので，これを行う機能が必要である．

提供者が演習システムを提供する際に，ユーザごとに存在する仮想サーバを 1 つ 1 つ管理することは大変な手間になり，ユーザ数が増加した場合に対応できなくなることが考えられる．そのため，提供者に変わり，ユーザの利用する仮想サーバを作成，削除する機能を自動化する必要がある．

また，認証認可および利用者の識別やユーザが存在するかの管理など，仮想サーバを作ること以外の演習環境として必要とされる要素は，作成する演習システムを利用する演習の内容に依存すると考えられ，共通化が難しい．そのため，本プラットフォームは仮想サーバの作成管理のみの機能を実装し，認証認可などの機能は外部のシステムとの連携により実現することとする．

4. 実装

各ユーザの環境を構築する技術は，先行研究と同様にコンテナ型仮想化を使用する．またコンテナの管理には以下の理由から kubernetes を使用する．

kubernetes には，コンテナの高速なデプロイや障害発生時の自動復旧などの機能がある．これにより，演習を行うアプリケーションに障害があった場合に提供者が手作業で復旧作業などを行う必要が無くなり，運用上の負荷を軽減できる．

また，コンテナを複数台のサーバに分散する機能があるため，演習を大規模に提供することを容易にすることができる．さらに，kubernetes 上で実装することで，さまざまな kubernetes 環境に容易に展開することができる．例えば，マネージド kubernetes を提供しているクラウド事業者の環境に展開することができ，演習環境の提供のクラウド化も容易である．

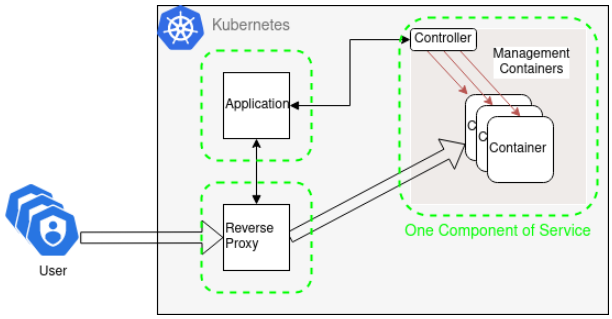

図 3　プラットフォームのコンポーネント化
Fig. 3　One Component in Platform

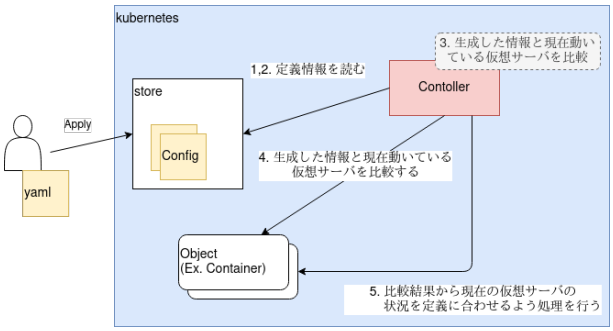

図 4　カスタムコントローラの動作
Fig. 4　Custom Controller

これらのことから要件にまとめた

- 複数サーバへの展開
- ユーザ用環境の自動復旧

の機能は仮想化基盤に kubernetes を採用することで解決可能だと考えられる．

4.1　プラットフォームの実装方針

本プラットフォームでは運用の簡略化という目的がある．運用を簡単にするため，複雑な機能を持つ単一のシステムではなく，シンプルな機能の組み合わせによるシステムの方が，構成を理解しやすく運用が簡単になると考えられる．そのため，本研究では，ユーザや演習環境に必要なソフトウェアの情報から，演習で使用するコンテナを，ユーザごとに作成する部分の機能のみを提供することとした．

また，演習において必要となる認証認可などの機能は，本プラットフォーム以外の演習システムでも共通であることから，外部のアプリケーションとの連携で実現することとした．そこで，仮想サーバ群を管理するプラットフォーム自体をコンポーネントとして扱い，演習環境全体にかかわるアプリケーションやリバースプロキシなどとの連携を取りやすくした．これを図 **3** に示す．

4.2　kubernetes の機能拡張

kubernetes を仮想化基盤に採用することだけでは，ユーザごとに必要となるコンテナの作成や削除を自動的に行うことができず，依然として提供者がユーザ一人一人のコンテナの起動や削除を行う必要がある．これは運用の手間となりユーザ数を増やす際に問題となることが考えられるため，自動化すべきである．

この自動化を kubernetes で実現するためには，個々のコンテナをユーザごとに割り当てるという機能を持ったシステムを作成する必要がある．しかし，システムを外部に作成してしまうと，kubernetes の持つ自動復旧や高い耐障害性などの恩恵を受けることができなくなってしまい，外部システムが単一障害点になり運用が難しくなるなどの課題がある．また，システムと kubernetes の連携のために，API エンドポイントやデータストアを別途用意する必要があり，運用の負荷の増加につながる．

そのため，可能な限り kubernetes の機能を有効に活用するため，kubernetes の機能の 1 つである機能拡張という手法で実装することとした．

4.3　実装

これを実現するため，CustomResourceDefinition を用いて API の拡張を行い，これを利用するカスタムコントローラの作成するという方法を選択した．実際に作成したプラットフォームを図 **4** に示す．

今回作成したカスタムコントローラでは仮想サーバの管理のために以下の処理を実装をした．

(1) 演習で利用するアプリケーションの定義情報を読む
(2) 演習システムを使用するユーザの定義情報を読む
(3) 上記 2 つの定義情報から仮想サーバの情報を生成する
(4) 生成した情報と現在動いている仮想サーバを比較する
(5) 比較結果から現在の仮想サーバの状況を定義に合わせるよう処理を行う

この実装により，運用負荷の小さいプラットフォームの構築を行うことができた．

4.4　まとめ

本研究により，web アプリケーションを使用した演習システムの作成において，仮想化技術を用いて演習システムの作成や運用を容易に行うことができるプラットフォームを構築することが可能になった．

今後はシステムにさらなる機能追加を行いより汎用性のあるプラットフォームの作成を目指す．

参考文献

[1]　新村正明，田中篤志，國宗永佳: LTI とリバースプロクシの連携による演習サーバ接続システム，情報処理学会研究報告，2018-CLE-25，Jun 2017.
入手先 ⟨https://research.google/pubs/pub43438/⟩