

# 情報工学科演習用のコンテナ技術を用いた新規サービスの設計・実装

宮平 賢<sup>2,a)</sup> 河野 真治<sup>2,b)</sup>

概要：IT 技術を学ぶ時の学習環境の 1 つとして、OS 上の隔離された環境を構築する技術であるコンテナがある。これらはローカルに設置された計算機、あるいはクラウド上に作られる。作成されるコンテナは学生、あるいは教員側から適切に管理するシステムが必要となる。管理システムはマルチユーザで動作するのは当然として、利用者や管理者に適した UI, sudo 権限で動作するコンテナへの対処などが含まれる。学生の演習には、Web サービスの実装や人工知能の学習などがある。そのため、気軽に開発環境やテスト環境などを用意できる利用のしやすさが重要である。本稿ではコンテナ管理ソフトウェアである Docker, Singularity を用いた新規 Web サービスの設計・実装を行う。

## 1. はじめに

情報通信技術の普及に伴い学生が学ぶ学習環境が必要となる。その学習環境として VM や コンテナにより、手軽に開発し試せる技術が普及している。だが、手元の PC 上で VM や コンテナを立ち上げ、開発を行うことはできるが、VM や コンテナの使用には高性能 PC や 有料のクラウドサービスが必要になる場合がある。この大きな負担を学生に負わせない仕組みが必要である。

琉球大学工学部工学科知能情報コースでは、希望の学生に学科のブレードサーバから仮想環境を貸出すサービスを行なっている。貸出をする VM のデフォルトのスペックでは不足場合要望に応じてスペックの変更を行なっている。だが、貸出サービスでは GPU を利用した処理環境を提供することができない。GPU が搭載されている PC は研究室によっては用意されているが、研究室に所属していない学生は利用することができない。そのため、新たな仕組みが必要である。

学科のブレードサーバに搭載される GPU は VM の貸出サービスでは利用することができないため、コンテナ技術を利用する。コンテナ管理ソフトウェアである Docker では NVIDIA Container Toolkit である nvidia-docker を利用することで、複数のコンテナで GPU を使用することができる。Docker では基本的に root 権限で動作する。また一般ユーザが docker コマンドを使用するには docker グ

ループに追加する必要がある。そのため Docker をマルチユーザで使用すると、他ユーザのコンテナへアクセスできるなどセキュリティの問題がある。

そこで、本論文では、Docker と マルチユーザ環境で利用しやすい Linux コンテナである Singularity を利用したコンテナ貸出サービスを提案する。このコンテナ貸出サービスでは、Web コンソールからコンテナの操作を行うことで他ユーザのコンテナへの操作をさせない。また、先行研究の課題でもあったデータの永続化を Singularity で、外部リポジトリの利用を Docker の操作を API で提供することで利用によるで解消する。

## 2. Docker

Docker とは Docker, Inc. が開発した Linux コンテナ管理ソフトウェアであり、2013 年にオープンソースのプロジェクトとして公開された。Docker ではコンテナを実行するだけでなく、コンテナイメージの作成

## 3. Singularity

Singularity とは、Berkeley Lab が開発した Linux コンテナであり、オープンソースソフトウェアで提供されている。Singularity には他のコンテナランタイムとは異なるセキュリティ機能がある。Singularity はコンテナを作成時にコンテナ内に UID 及び GID 情報を動的に書き込み、ユーザの権限を引き継ぐ。そのため、ユーザが非特権ユーザの場合コンテナ内でも非特権ユーザとして動作する。また、Singularity は他のユーザのコンテナを操作することができないためマルチユーザ環境で安全に使用できる。

<sup>1</sup> 琉球大学大学院理工学研究科情報工学専攻

<sup>2</sup> 琉球大学工学部工学科知能情報コース

<sup>a)</sup> mk@cr.ie.u-ryukyu.ac.jp

<sup>b)</sup> kono@ie.u-ryukyu.ac.jp

Singularity のコンテナではデフォルトで \$HOME, /tmp, /proc, /sys, /dev がマウントされる。そのため、ユーザがコンテナ内にファイルをコピーする手間を省くことができる。

#### 4. GitLab

GitLab とは、GitLab.Inc. が開発した Git リポジトリマネージャーであり、オープンソースソフトウェアで提供されている。GitLab には無償版の GitLab Community Edition (以下 GitLab CE) と有償版の GitLab Enterprise Edition (以下 GitLab EE) がある。本コースでは GitLab CE を運用している。本研究では GitLab CE に統合されている CI/CD 機能を利用する。

#### 5. 関連研究

本サービスに関連した研究として、Docker をラップし複数のユーザで利用することを目的とした ie-docker, Kubernetes を利用した教育用コンテナ貸出を目的とした, digdog がある。この 2 つの研究について軽く概要を説明する。

##### 5.1 ie-docker

ie-docker とは Docker をラップし複数のユーザで利用することのできるコンテナ管理ツールである。利用する学生は ssh でブレードサーバへ接続し、ie-docker を使用してコンテナを操作することができる。ie-docker は UID 及び GID 情報を取得し他のユーザのコンテナを操作させない。またユーザが使える docker の機能を制限する。表 1 が ie-docker の機能である。

表 1: ie-docker のコマンド

ps	起動中のコンテナの一覧を表示する
run	コンテナを作成する
start	コンテナを起動する
stop	コンテナを停止する
attach	起動しているコンテナに attach する
cp	コンテナにファイルを送信する
rm	コンテナを削除する

##### 5.2 digdog

digdog とは Kubernetes を利用したコンテナ貸出サービスである。学生は Dockerfile を GitLab CI/CD を利用して GitLab Registry に Docker イメージを登録する。コンテナを利用するには学科アカウントを使用して Web サービスへログインし、登録したイメージでコンテナを作成することができる。また学生の Namespace が作成される。RBAC を用いたリソース操作のアクセス制御を行っており、学生は Kubernetes コマンドである kubectl コマン

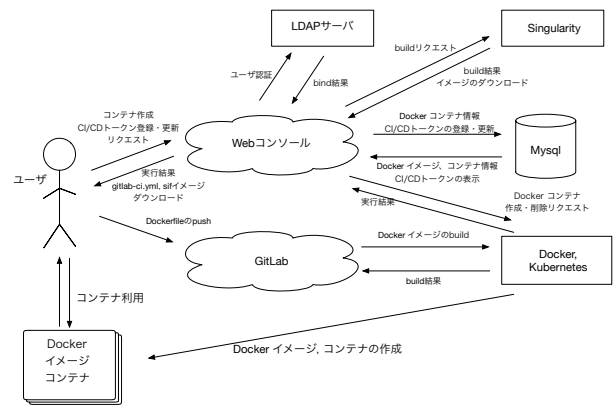


図 1: システム構成

ドで Pod の操作を行うことができる。RBAC で許可されているリソース操作は表 2 である。

表 2: kubectl のコマンド

get	Pod の一覧を表示する
log	Pod の Log を表示する
exec	Pod にアクセスする

#### 6. WhaleMountain

本サービスの構成を図 1 に示す。本サービスは digdog を参考にして作られた コンテナ貸出を行う Web サービスである。学生は学科アカウントを使用して Web サービスへログインし、コンテナの作成や操作を行うことができる。また、GitLab CI/CD を使用し学生が作成した Dockerfile を GitLab へ Push することで、学生が自由に Docker イメージを作成することもできる。作成した Docker イメージは他の学生に共有できる。また学生は他の学生が作成したコンテナやイメージを操作することはできない。WhaleMountain が提供するサービスは以下である。

- GitLab CI/CD Token の更新
- Docker Image の削除
- Singularity Image のダウンロード
- Docker コンテナの作成, 操作
- Kubernetes Pods の作成

#### 7. WhaleMountain と他ツールとの比較

関連研究である ie-docker と digdog との比較を行う。

##### 7.1 ie-docker と比較

ie-docker では学生が自由に Docker イメージを作成することができない。またコンテナで使用できる Docker イメージも制限される。WhaleMountain の Docker コンテナでは学生がコンテナの log の確認することができない。また ie-docker の attach と比べ WhaleMountain では、コンテナに接続するのではなく、コンテナでコマンドを実

行し出力を表示するため、ファイルの編集などを行うことができない。そのため、WhaleMountain では、Docker コンテナだけでなく Kubernetes の Pods を作成することができる。Kubernetes で作成した Pods に対して 学生は kubectl コマンドで操作ができる。

## 7.2 digdog と比較

WhaleMountain は digdog を参考にして作られたため、基本構成は似ている。だが、digdog はコンテナ作成に学生が作成した Docker イメージのみに制限されている。また、GitLab CI/CD の CI で Dockerfile を build する際に利用する GitLab Runner が共有される。そのため複数の学生が同時に Dockerfile を build することができない。WhaleMountain では コンテナ作成で Docker イメージに制限をかけないため Docker Hub に登録されている Docker イメージも利用できる。また、学生用の GitLab Runner を Token の設定時に用意するため、複数の学生が同時に Dockerfile を build することができる。

## 8. 今後の課題

## 9. まとめ

参考文献