

Amazon Web Servicesによるサーバの構築

システム科学技術学部 情報工学科

2年 松田 来央

2年 金原 隆斗

2年 織田 拓希

1年 中林 舜葵

指導教員 システム科学技術学部 情報工学科

准教授 廣田 千明

1 はじめに

昨今のインターネット上のサーバの多くは、Amazon Web Services (以下AWS) で運用されている。数年前までは自社でサーバの管理をする、オンプレミス環境が主であった。大学の講義ではAWSをはじめとしたクラウド環境でのサーバ構築について学ぶことはなく、この機会にAWSがここまで用いられているのかを自分たちで確かめることにした。また、オンプレミス環境と比較することでそれぞれのメリットを調べてみようと考えた。

2 AWSによるサーバの構築

AWS は Amazon 社が提供するクラウドサービスの総称である。AWS ではレンタルサーバをはじめ、データベースなど、大きく分けて 90 以上のサービスが提供されている。サーバの構築、運用は Elastic Compute Cloud (EC2 と呼ばれる) を用いることで実現できる。また、サービスを構築する際にライブラリやアプリケーションを仮想化する Docker を利用した。大まかな手順は図1の通りである。以下では手順を詳しく解説する。

- (1) AWS のアカウントの作成
- (2) EC2 のインスタンスの作成
- (3) Docker のインストール
- (4) WordPress の構成

図1 AWSによるサーバ構築の手順

2.1 AWS アカウントの作成

AWS を利用するには AWS のアカウントを作成する必要がある。登録ページ (<https://portal.aws.amazon.com/billing/signup#/start>) から作成することができる。この際、デビットカードもしくはクレジットカードが必要となるので準備しておく。

2.2 EC2 のインスタンスの作成

AWS アカウントの認証情報を使ってアクセスすることは、共同でサーバを扱う場合にはセキュリティ上好ましくない。そこで AWS Identity and Access Management (IAM) ユーザを作成し、権限の管理を行う。続けて、EC2 に ssh を使ってログインするためのキーペアを作成する。最後に、セキュリティグループを作成し、ssh を使用して IP アドレスからインスタンスに接続できるようにするためのルールをセキュリティグループに追加する。

2.3 Docker のインストール

ここではサーバの OS に Ubuntu 18.04 LTS を使用したため、Docker 公式ドキュメント[1]に沿ってインストールすることで簡単にインストールが可能である。

2.4 WordPress の導入と公開

まず、docker-compose.yml を作成する必要がある。このファイルはコンテナ管理をより簡単に行うために作成するファイルで、使用するコンテナやそれらの設定を記述する。続けて、コンテナを起動する。コンテナを起動するには、図 1 のように命令する必要がある。最後にポートを開放する。ここまでの操作で EC2 上に WordPress が立ち上がっているが、この段階ではポートが開いておらず外部からアクセスできない。そこで AWS コンソール上でポートを開放する必要がある。ポートの開放には、まず AWS コンソールにログインし、EC2 の管理画面を開く。次に、WordPress を構成したインスタンスを選択し、セキュリティグループのリンクを開く。その後、表示された画面の編集ボタンから設定を行うことができる。図 2 の設定画面が表示されるので、ルールの追加から、HTTP と HTTPS、MySQL のポートを開放する。以上により、WordPress で作成したページをインターネットに公開することができる。

```
docker-compose up -d
```

図 1 WordPress に必要なコンテナの起動コマンド

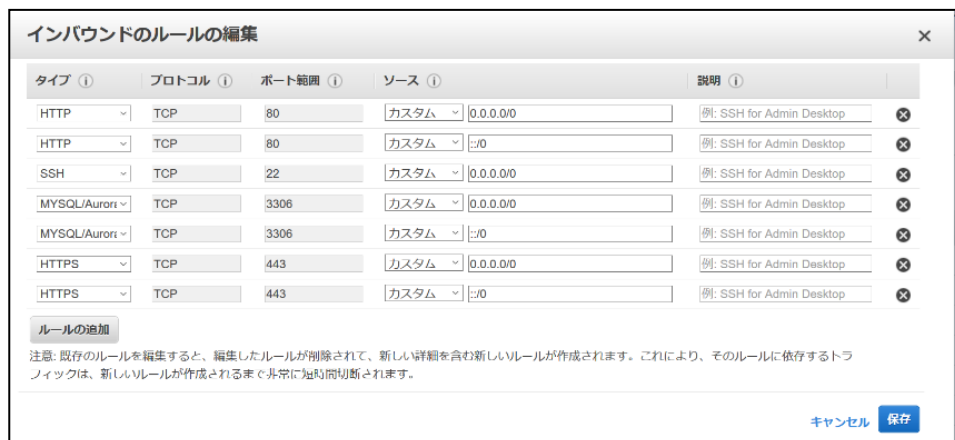


図 2 ルールの編集画面

3 オンプレミス環境でのサーバ構築

オンプレミス環境でのサーバ構築にはハードウェアと OS（ここでは Linux を用いる）のイメージが必要である。ここで構築するサーバは WordPress を動かすためのウェブサーバで LAMP 環境を構築する。LAMP 環境とはウェブアプリケーション開発で推奨されている環境のことで、Linux、Apache、MariaDB、PHP の4つの頭文字をとったものである。Apache は外部からのリクエストを処理するために、MariaDB は WordPress のデータを保存管理するために、PHP は WordPress を動かすために使われる。

まずは OS のインストールをする。ここでは Ubuntu server 18.04 LTS という Linux ディストリビューションを使用する。Linux のインストールは Linux のイメージを公式サイトからダウンロードし DVD や USB メモリに書き込み、サーバ本体からインストーラを起動することでインストールできる。図3にインストール時の様子を示す。

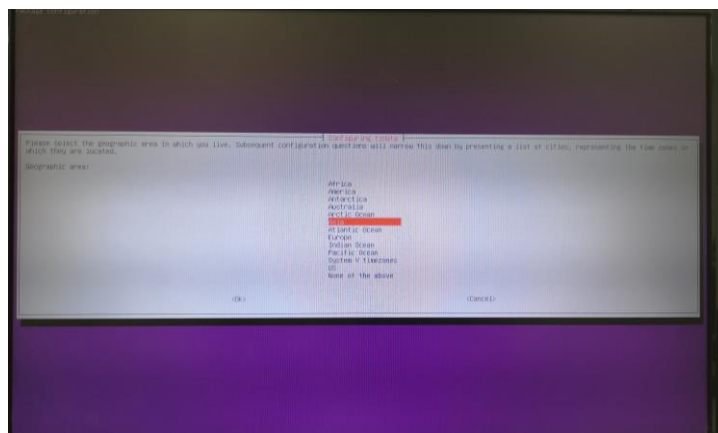


図3 Ubuntu Linux のインストールの様子

Linux のインストール後は LAMP 環境構築のために必要なソフトウェアのインストールをする。Linux の多くはソフトウェアをインストールする際には、ソフトウェアのインストールと同時に依存するライブラリやソフトウェアのインストールも自動でインストールされるパッケージ管理システムを使う。Ubuntu server はパッケージ管理システム apt を使いコマンドラインからソフトウェアをインストールすることができる。例えば、apache2 というパッケージをインストールする場合には

```
$ sudo apt install apache2
```

と命令することで簡単にインストールすることができる。

LAMP 環境に必要なソフトウェアをインストール後 systemd により apache や MariaDB のサービスを起動し、WordPress で使用するデータベースの作成などの設定を行う。WordPress のインストールは公式サイトから WordPress をダウンロードし、Apache が動いているディレクトリに展開し、データベースなどの設定を自分の環境に合わせて設定することでサーバの IP アドレスにアクセスし画面の指示に従い WordPress のインストールができる。WordPress のインストール後に簡単なセキュリティ対策として Apache での http 通信に使われる 80 番ポートのみを開放しそれ以外の全てのポートを閉じる必要がある。そのためには、

```
$ sudo ufw allow 80
```

と命令する必要がある。最後に、試しに作成した情報工学科のウェブページを図4に載せる。researchmap[2]から業績のデータを取得し、ページ内に挿入することを検討したが、そこまで完成させることができなかつた。これが達成できると、教員の業績を自動的に表示することで、ウェブページの更新の手間が大幅に削減できる。この点は今後の課題としたい。



図4 WordPress を用いて作成した情報工学科のウェブページ

4 クラウドとオンプレミスの比較

実際に自分たちでクラウドとオンプレミスでそれぞれサーバを構築してみて、オンプレミスではサーバの構築の際に予期せぬエラーが起きる、そもそも事前にある程度の準備が必要と大変なことが多かった。従来の方法で実際起きたエラーの例としては、WordPress を更新しようとした際に、apache 実行ユーザに書き込み権限がなく更新できなかったことが挙げられる。一方 AWS を用いた方では、時間もそこまでかからず、エラーもなく簡単に立ち上げることができた。運用するまでの段階では圧倒的にクラウドの方が簡潔であることが分かった。また、AWS のサービスは EC2のみを使用した、静的な Web ページを提供でき、システムダウンがない S3を使い WordPress の一部のページを移行したりすることで、オンプレミスに比べより安定したサービスを提供できると考えられる。

5 終わりに

本研究を通してクラウド、オンプレミスそれぞれのサーバ構築について学ぶことができた。また、AWS に対する理解が深まり、その便利さも知ることができた。この経験を将来生かしていきたい。

参考文献

- [1] Get Docker Engine - Community for Ubuntu | Docker Documentation (<https://docs.docker.com/install/linux/docker-ce/ubuntu/>)
- [2] Researchmap (<https://researchmap.jp/>)