

AWS 講習会 -LAMP環境構築編-

目次:

1. AWS 上に仮想サーバを立ち上げる
 2. SSH でログイン
 3. LAMP をインストールし、Apache の起動確認
 4. PHP を使って初期画面変更、確認
 5. MySQL, phpMyAdmin の設定ファイルを変更、確認
 6. FileZila を使って変更を反映できるようにする
 7. 変更した内容が反映できているかを確認
-

はじめに...

今回は初級編ということで、AWS_EC2 内に仮想サーバを立ち上げ、その中に動作環境の構築をして行きます。

構成は「Amazon Linux AMI」, 「t2.micro」で LAMP 環境を構築し、「FileZila」で確認できるようにします。

本当に基本の構成のみとなりますので、セキュリティの強化や、S3 や RDS を使ったWEBアプリの構築などは、

別途実施しますので、そちらにご参加をお願いします。

1. AWS 上に仮想サーバを立ち上げる

- AWS マネジメントコンソールにログインする (<https://aws.amazon.com/jp/console/>)
 - 画面右上のリージョンに「東京」と表示されていない場合は「アジアパシフィック(東京)」を選択する
 - 画面左上のサービスからEC2をクリック(検索してでも大丈夫です)
 - 左パネルから EC2ダッシュボードを選択し、「インスタンス作成」をクリック
 - 「Amazon Linux AMI」を選択
 - 「t2.micro」を選択
 - 変更を加えず、作成をクリック
 - キーペアの作成画面になるので、新しいキーペアで「test」と入力
 - 「キーペアのダウンロード」をクリックし「インスタンスを作成」をする
-

2. SSH でログイン

- EC2 上の画面左から「インスタンス」を選択し、インスタンスが作成されているか確認
- 作成したインスタンス上で右クリックから「接続」を選択
- インスタンスへの接続コマンドを確認し、メモしておく
- EC2 上の画面左から「セキュリティグループ」を対象

- インスタンスに対応するセキュリティグループを右クリックし、「インバウンドルールの編集」を選択
- [HTTP/TCP/80/マイIP] と [SSH/TCP/22/マイIP] の2つを作り、保存する
- 作成した鍵を Desktop 配下に移動する

(ターミナル)

```
* cd Desktop
* ls                      #test.pem があることを確認
* chmod 400 test.pem
* ssh -i ...              #先ほどメモしたコードを貼り付け、実行する
* (yes/no)? yes          #初回に聞かれるので yes と入力する
* [ec2-user...]$         となったら SSH での接続完了
```

3. LAMP をインストールし、Apache の起動確認

(ターミナル)

```
* sudo yum install -y httpd24 php56 mysql55-server php56-mysqld #LAMP のインストール
* sudo service httpd start #Apache 起動
* sudo chkconfig httpd on  #Apache 自動起動の設定
* sudo chkconfig --list httpd #自動移動の設定確認
(# 結果: httpd 0:off 1:off 2:on 3:on 4:on 5:on 6:off)
```

- AWS_EC2 のインスタンスから、IPv4パブリックIP の項目をコピーする
- 「http://IPアドレス/」をブラウザで開いてApacheの動作確認

4. PHP を使って初期画面変更、確認

(ターミナル)

```
* sudo groupadd www          #www グループをインスタンスに追加
* sudo usermod -a -G www ec2-user #ユーザー(今回は ec2-user)を www グループに追加
* exit                      #一度ログアウト
* ssh -i ...                #再度 SSH で接続
* groups                     #www グループのメンバーシップを検証
(# 結果: ec2-userwheel www)
* sudo chown -R root:www /var/www #/var/ とそのコンテンツのグループ所有権を www グループに変更
* sudo chmod 2775 /var/www      #/var/www およびそのサブディレクトリのディレクトリ許可を変更
* find /var/www -type d -exec sudo chmod 2775 {} \; #将来のサブディレクトリにグループ ID を設定
* sudo find /var/www -type f -exec sudo chmod 0664 {} \; #/var/www およびサブディレクトリのファイル許可を繰り返し変更
* echo "<?php echo 'Hello World'; ?>" > /var/www/html/index.php #index.php 作成
* echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php #phpinfo.php 作成
```

- 「http://IPアドレス/index.php」をブラウザで開いて HTML の動作を確認
 - 「http://IPアドレス/phpinfo.php」をブラウザで開いて php の動作を確認
-

5. MySQL, phpMyAdmin の設定ファイルを変更、確認

(ターミナル)

```
* sudo service mysqld stop          #MySQL を起動
* sudo chkconfig mysqld on          #MySQL を自動設定に設定
* sudo yum-config-manager --enable epel #Fedora プロジェクトの Extra Packages for Enterprise Linux (EPEL) リポジトリを有効化
* sudo yum install -y phpMyAdmin    #phpMyAdmin をインストール
* sudo sed -i -e 's/127.0.0.1/your_ip_address/g' /etc/httpd/conf.d/phpMyAdmin.conf
#phpMyAdmin の設定ファイルを編集
(your_ip_address は各自のIPアドレス)
* sudo chown -R apache /etc/phpMyAdmin #phpMyAdmin のディレクトリの所有権を apache に設定
* sudo chown -R ec2-user /etc/phpMyAdmin #phpMyAdmin のディレクトリの所有権を ec2-user に設定
* sudo vi /etc/phpMyAdmin/config.inc.php #phpMyAdmin の設定を変更(詳細は下記にて)
* sudo service httpd restart          #Apache の再起動
* sudo service mysqld restart        #MySQL の再起動
```

(sudo vi /etc/phpMyAdmin/config.inc.php の内容変更部分)

```
$cfg['Servers'][$i]['controluser'] = 'root';
$cfg['Servers'][$i]['user'] = 'root';
$cfg['Servers'][$i]['AllowRoot'] = true;
$cfg['Servers'][$i]['AllowNoPassword'] = true;
```

- 「http://IPアドレス/phpmyadmin」をブラウザで開き、ユーザーを「root」にして実行
-

6. FileZila を使って変更を反映できるようにする

- FileZila をダウンロードする

Mac版URL: <https://filezilla-project.org/download.php?platform=osx>

Windows版URL: <https://filezilla-project.org/download.php?platform=win64>

- FileZilaを起動し、「鍵ファイルを追加」から鍵ファイルを指定 [設定→SFTP、鍵ファイル「test.pem」]
- サイトマネージャの詳細から、限定のリモートディレクトリを「/var/www/html」にする
- 一般から、ホスト名にIPアドレスまたは PulibcDNSプロトコル を SFTP に変更
- ユーザー名に「ec2-user」に指定
- パスワードを入力せずに「接続」をクリック

- ターミナルで作成したファイルが置いてあることを確認
-

おわりに...

AWS のサービスは無料枠を利用できるものもあれば、有料のものもあります。

クラウドのメリットは、気軽に構築でき、また壊すことができることです。

講義の終わりとして後片付けをし、興味があればまたこの PDF を参考に構築してみてください。

EC2:

インスタンス: 右クリックから「インスタンスの状態」 -> 「削除」

キーペア: 右クリックから「削除」

セキュリティグループ: 右クリックから「セキュリティグループの削除」(#インスタンスが「**terminated**」(削除済み) になってから)

(* キーペア、セキュリティグループについては費用が発生しないので、消さなくても問題はありません)

主催:

ENGINEER STYLE TOKYO

<https://tec.connpass.com/>