

ApacheとPHP,MySQLでアプリを作成し、EC2で公開する

目次

| | |
|--|----------|
| ApacheとPHP,MySQLでアプリを作成し、EC2で公開する | 1 |
| 目次 | 1 |
| まえがき | 4 |
| この本について | 4 |
| 対象読者 | 4 |
| ゴール | 4 |
| アプリケーションの機能 | 4 |
| EC2インスタンスを作成する | 5 |
| EC2とは | 5 |
| aws アカデミーについて | 5 |
| SSH鍵を取得 | 5 |
| インスタンスを作成する | 6 |
| クイズ | 9 |
| インスタンスに接続する | 10 |
| クイズ | 11 |
| Linuxコマンド | 11 |
| ls | 11 |
| pwd | 11 |
| cd | 11 |
| touch | 12 |
| cp | 12 |
| mkdir | 12 |
| cat | 12 |
| > | 12 |
| < | 12 |
| | 12 |
| --version | 13 |
| rm | 13 |
| apt | 13 |
| クイズ | 13 |
| Apache をインストール・起動 | 14 |
| Apacheとは | 14 |
| インストールする | 14 |
| 表示されない場合 | 15 |
| HTMLを公開する | 15 |
| HTMLを書き換える | 16 |

| | |
|-------------------------------|----|
| vi の使い方 | 16 |
| PHPをインストールする | 16 |
| PHPとは | 17 |
| インストールする | 17 |
| PHPを公開する | 17 |
| クイズ | 19 |
| MySQLをインストールする | 19 |
| MySQLとは | 20 |
| インストールする | 20 |
| SQLを実行する | 20 |
| MySQLを操作(初期設定) | 21 |
| root ユーザーのパスワードの変更する | 21 |
| 注意 | 21 |
| 初期設定を変更しセキュリティを向上させる | 22 |
| 設定の詳細 | 23 |
| MySQL に入る | 23 |
| 注意 | 24 |
| MySQLを操作する(データベース・テーブル) | 24 |
| MySQLのユーザーを作成 | 24 |
| ユーザーが作成できているか確認する | 24 |
| クイズ | 25 |
| データベースを作成する | 25 |
| テーブルを作成する | 26 |
| SQLについて | 26 |
| クイズ | 27 |
| テーブルにデータを追加する | 27 |
| クイズ | 28 |
| ファイルに書かれたSQLを実行する | 28 |
| クイズ | 29 |
| MySQLのユーザーに権限を与える | 30 |
| ユーザーに権限を付与する | 30 |
| 確認する | 30 |
| クイズ | 31 |
| アプリケーションからデータベースのデータを取得する | 31 |
| アプリケーションからデータベースにアクセスできるようにする | 32 |
| php-pdoをインストールする | 32 |
| php-mysqlをインストールする | 32 |
| アプリケーションを作成する | 32 |
| ソースコードの解説 | 35 |
| データベースに接続する | 35 |
| データを取得する | 35 |
| データを追加する | 36 |
| データを削除する | 37 |
| ソースコードをGitHubにpushする | 38 |

| | |
|--------------------------|----|
| レポジトリを作成する | 38 |
| EC2 から GitHub に SSH 接続する | 38 |
| 公開鍵をGitHubに設定する | 39 |
| SSHの設定ファイルを書く | 39 |
| 接続をテストする | 40 |
| GitHubにソースコードをアップロードする | 40 |
| Gitコマンド | 41 |
| git init | 41 |
| git add | 41 |
| git commit | 41 |
| git remote add | 42 |
| git push | 42 |
| おわりに | 42 |

まえがき

この本について

この本は、リモートサーバに接続する(SSH)から、実際にサーバ内でコマンドライン操作を行いながら、アプリケーションを公開するまでの流れを問題形式で解説しています。サンプルのアプリの解説は少なく、公開するまでのコマンドライン操作に重点をおいて解説しています。

資料通りにアプリケーションを構築し、各クイズにも回答してください。回答は以下のフォームに回答してください。

<https://forms.gle/nP6cZczhhZQLz2zr9>

対象読者

- アプリケーションの開発やプログラミングの勉強は行ったことがあるが、データベースやWEBサーバを扱ったことがない方。
- コマンド操作に慣れたい方。
- windowsの方。macの解説はないので、最初の部分は自力でお願いします。

ゴール

この本では、IPアドレスをブラウザに入力し、アプリケーションにアクセスできる状態にする。なおかつ、そのアプリケーションが正常に動作している状態にする。

▼完成図

| | | | | |
|-----------|----------------------|-----------------------------------|----------------------|-----------------------------------|
| 商品名 | <input type="text"/> | 価格(円) | <input type="text"/> | <input type="button" value="送信"/> |
| 商品名 価格(円) | | | | |
| ノート 100円 | | <input type="button" value="削除"/> | | |

アプリケーションの機能

- 商品名と価格を入力し、送信すると商品を追加することができる。
- 削除ボタンを押すと、該当商品を削除することができる。

EC2インスタンスを作成する

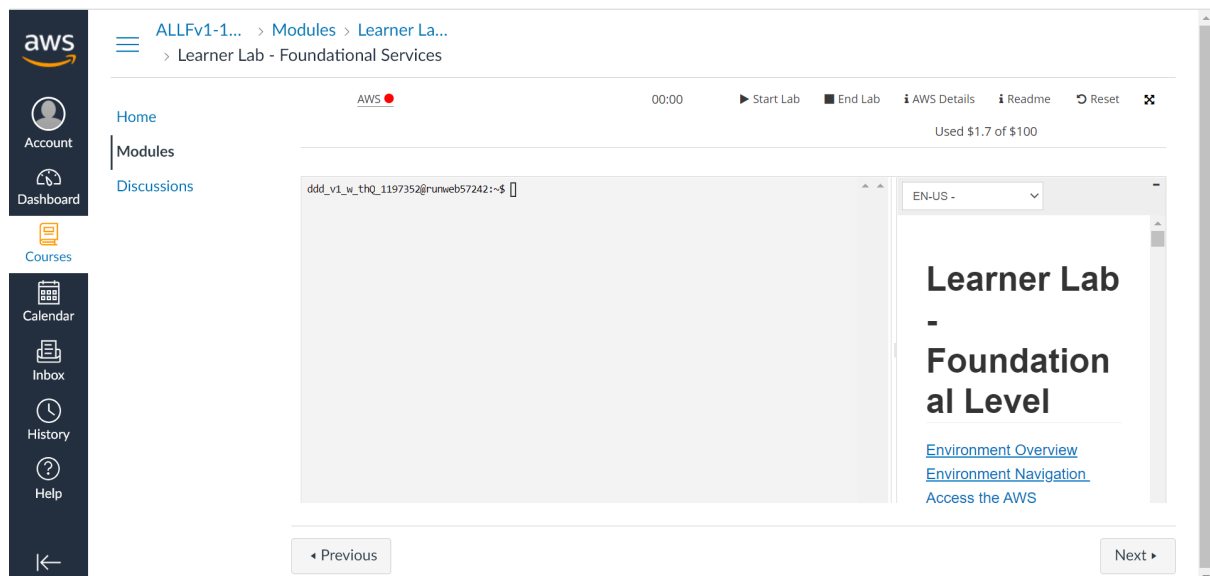
EC2とは¹

AWSというクラウドプラットフォームの中のサービスで、以下の機能を持っています。

- インスタンスと呼ばれる仮想コンピューティング環境
- Elastic IP アドレスと呼ばれる、動的クラウドコンピューティング用の静的な IPv4 アドレス²

このクラウドコンピュータにアプリケーションを構築していきます。

aws アカデミーについて



awsアカデミーにログインすると上記のような画面になります。

赤い点の時は EC2 が停止しています。▶start labをクリックしてください。

SSH鍵を取得

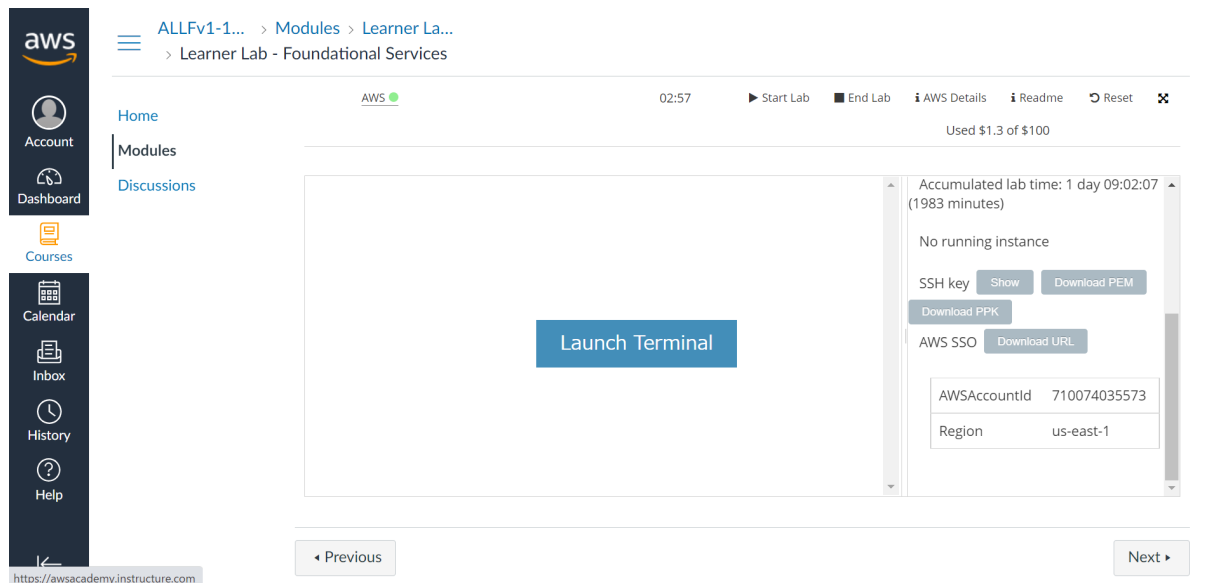
awsアカデミーの画面からEC2を操作するために必要な「鍵」を取得します。

¹ 「Amazon EC2 の機能」(一部抜粋)

https://docs.aws.amazon.com/ja_jp/AWSEC2/latest/UserGuide/concepts.html

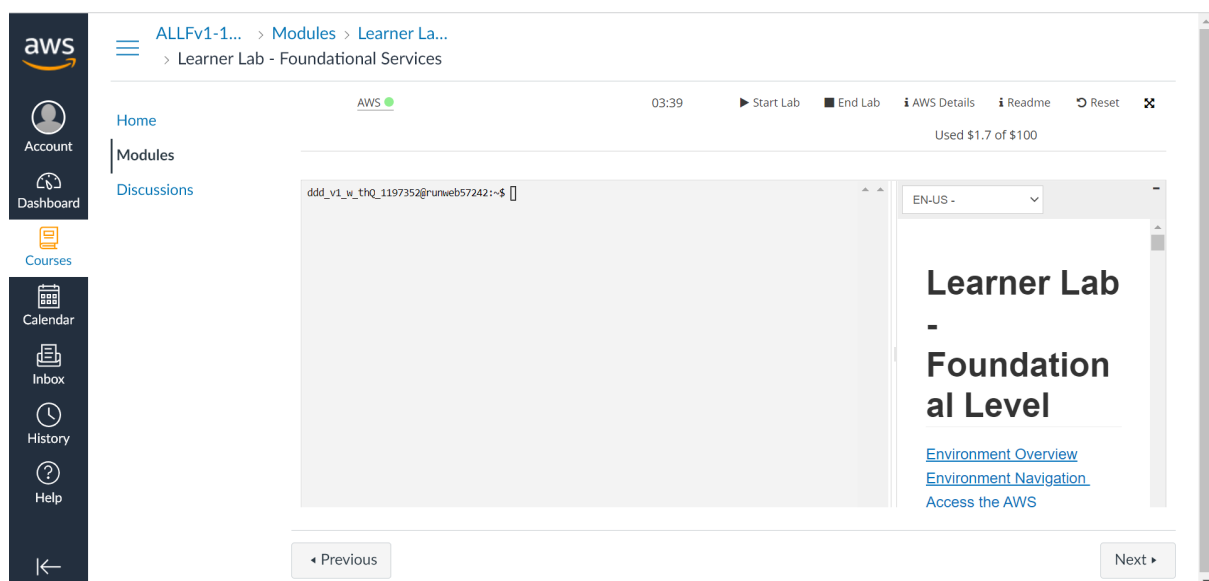
² インターネットに存在するコンピュータの場所を示すものである。EC2 には、世界中に1つのグローバル IP アドレスが割り当てられています。そのため、IP アドレスを指定することで、EC2 にアクセスすることが可能であり、コンテンツを配信することが可能です。

EC2ではIPアドレスを固定する設定を行わない限り、グローバルIPアドレスは変化します。



1. aws details をクリック
2. Download pem をクリック

labsuser.pemというファイルがダウンロードされます。大切に管理してください。



EC2 の起動が完了すると緑に変化します。

緑の点をクリックすると**AWS** コンソールの画面に飛びます。

インスタンスを作成する

awsコンソールに移動後に画面上部の検索ボックスに「EC2」と入力し、EC2の管理画面に移動してください。

aws

サービス ▼

🔍 サービス、機能、マーケットプレイスの製品、ドキュメントを検索し [Alt+S]

New EC2 Experience
Tell us what you think

EC2 ダッシュボード
イベント
タグ
制限

▼ インスタンス
インスタンス New
インスタンスタイプ
テンプレートの起動
スポットリクエスト
Savings Plans
リザーブドインスタンス New
専用ホスト
スケジュールされたイン
スタンス
キャパシティの予約

▼ イメージ
AMI

▼ Elastic Block Store
ボリューム
スナップショット
ライフサイクルマネー
ジャー New

▼ ネットワーク & セキュリ
ティ

リソース

🔄 ⚙️

米国東部 (バージニア北部) リージョンで次の Amazon EC2 リソースを使用しています。

| | | | | | |
|--------------|---|------------|---|------------|---|
| インスタンス (実行中) | 1 | Elastic IP | 0 | インスタンス | 1 |
| キーペア | 0 | スナップショット | 0 | セキュリティグループ | 2 |
| プレースメントグループ | 0 | ボリューム | 1 | ロードバランサー | 0 |
| 専用ホスト | 0 | | | | |

📌 AWS Launch Wizard for SQL Server を使用すると、Microsoft SQL Server Always On 可用性グループのサイズ調整、設定、デプロイを簡単に行うことができ
ます。詳細はこちら

インスタンスを起動

🔄

開始するには、クラウド内の仮想サーバーである Amazon EC2 インスタンスを起
動します。

インスタンスを起動 ▼

注意: インスタンスは、米国東部 (バージニア北部) リージョンで起動されます

予定されているイベント

🔄

米国東部 (バージニア北部)

予定されているイベントはありません

サービス状態

🔄 サービス状態ダッシュボード 📄

リージョン
米国東部 (バージニア北部)

ステータス
🟢 このサービスは正常に動作していま
す

ゾーン

| ゾーンのネーム | ゾーン ID |
|------------|----------|
| us-east-1a | use1-az1 |
| us-east-1b | use1-az2 |
| us-east-1c | use1-az4 |

「インスタンスを起動」をクリックしてください。

名前とタグ

名前

web

さらにタグを追加

▼ アプリケーションおよび OS イメージ (Amazon マシンイメージ)

AMI は、インスタンスの起動に必要なソフトウェアとオペレーティングシステム、アプリケーションサーバー、アプリケーションを定義したテンプレートです。お望みのAMIが以下に表示されない場合は、AMI を検索または登録してください。

Q 何千ものアプリケーションイメージと OS イメージを含むカテゴリ全体を検索します。

最新

クイックスタート



その他の AMI を
探す

AWS Marketplace
コミュニティからの
AMI を探す

Amazon マシンイメージ (AMI)

Ubuntu Server: 22.04 LTS (Focal Fossa) | SSD Volume Type
ami-09d56966ab235b1 (64 ビット-64GB) / ami-034a975821f25213 (64 ビット-64GB)
仮想化: hvm | ENA 有効: true | ルートデバイスタイプ: ebs

無料利用枠の対象

説明

Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2022-04-20

アーキテクチャ

AMI ID

64 ビット (x86)

ami-09d56966ab235b1

▼ インスタンスタイプ

インスタンスタイプ

t2.micro
ファミリー: t2 | vCPU: 1 | GB メモリ: 1
オンデマンド Linux 利用: 0.016 USD / 1 時間あたり
オンデマンド Windows 利用: 0.062 USD / 1 時間あたり

無料利用枠の対象

インスタンスタイプを比較

▼ キーペア (ログイン)

キーペアを使用してインスタンスに安全に接続できます。インスタンスを起動する前に、適切なキーペアにアクセスできることを確認してください。

キーペア名 - 必須

webkey

新しいキーペアの作成

▼ ネットワーク設定

編集

ネットワーク

vpc-0a07213ac685810e9

サブネット

標準サブネット (アベイラビリティゾーンのデフォルトサブネット)

パブリック IP の自動割り当て

有効化

ファイアウォール (セキュリティグループ)

A security group is a set of firewall rules that control the traffic for your instances. Add rules to allow specific traffic to reach your instance.

☒ セキュリティグループを作成する

☐ 既存のセキュリティグループを選択する

次のルールを使用して、「launch-wizard-1」という新しいセキュリティグループを作成します。

☒ からの SSH トラフィックを許可する
インスタンスへの接続

許可の範囲
0.0.0.0/0

☐ インターネットからの HTTP トラフィックを許可する
エンドポイントを設定アップするには (ウェブサーバーの作成時など)

☐ インターネットからの HTTP トラフィックを許可する
エンドポイントを設定アップするには (ウェブサーバーの作成時など)

⚠ 送信元が 0.0.0.0/0 のルールを設定すると、すべての IP アドレスからインスタンスにアクセスすることが許可されます。セキュリティグループのルールを設定して、適切な IP アドレスからのみアクセスできるようにすることをお勧めします。

▼ ストレージを設定

アドバンスド

サイズ

8

GB

gp2

ルートボリューム

ⓘ 無料利用枠の対象のお客値は、最大 30 GB の EBS 標準 (gp2) ストレージまたはマグネティックストレージに制限されます。

▼ 概要

インスタンス名

1

ソフトウェアイメージ (AMI)

Canonical, Ubuntu, 22.04 LTS, ...
ami-09d56966ab235b1

インスタンスタイプ (インスタンスタイプ)

t2.micro

ファイアウォール (セキュリティグループ)

新しいセキュリティグループ

ストレージ (ボリューム)

1 ボリューム - 8 GB

ⓘ 無料利用枠: In your first year includes 750 hours of t2.micro (or t1.micro in the regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 GiB of bandwidth to the internet.

キャンセル

インスタンスを開始

名前とタグに「web」、アプリケーションおよびOSイメージを「Ubuntu」、キーペアを「vockey」と入力してください。最後に「インスタンスを起動」をクリックしてください。

EC2インスタンスが作成されました。

クイズ

1. なぜ、秘密鍵は公開してはいけないのでしょうか。
 - 1.1. 同じ内容の鍵を持っている人がいる可能性があるため。
 - 1.2. この鍵を持っている人はインスタンスにアクセスできてしまうため。
 - 1.3. メールアドレスなどの個人情報が漏れてしまうため。
 - 1.4. 公開すると鍵が無効化され、インスタンスにアクセスできなくなるため。
2. なぜ、22 番 TCP を許可するのでしょうか。
 - 2.1. ssh接続を行うため。
 - 2.2. telnet接続を行うため。
 - 2.3. httpで接続するため。
 - 2.4. httpsで接続するため。
3. なぜ、80 番 TCP を許可するのでしょうか。
 - 3.1. ssh接続を行うため。
 - 3.2. telnet接続を行うため。
 - 3.3. httpで接続するため。
 - 3.4. httpsで接続するため。
4. なぜ、443 番 TCP を許可するのでしょうか。
 - 4.1. ssh接続を行うため。
 - 4.2. telnet接続を行うため。
 - 4.3. httpで接続するため。
 - 4.4. httpsで接続するため。
5. なぜ、ソースで0.0.0.0/0を指定するのでしょうか。
 - 5.1. 自分以外がアクセスできないようにするため。
 - 5.2. 自分以外もアクセスできるようにするため。
 - 5.3. この値以外は無効なため、設定できないため。
 - 5.4. 海外からのアクセスを制限するため。

インスタンスに接続する

インスタンスを作成し、起動しているインスタンスを選択すると以下のような画面になります。



この画面に表示されている「パブリックIPv4アドレス」をSSHの際に使います。また、今後の動作確認の際にも使います。

自身のパソコンのコマンドプロンプトを起動して、以下のコマンドを入力してください。

```
$ ssh -i C:\Users\1234\Documents\labsuser.pem ubuntu@12.34.567.890
```

SSH³コマンドを用いてインスタンスに接続する。

1. -iはSSHの秘密鍵のファイルを指定する。
2. ubuntuはSSHする際のユーザー名を指定する。
3. 12.34.567.890はIPアドレスである。自身のEC2のIPアドレスに書き換えて実行してください。

これで、SSHでインスタンスに接続することができました。

EC2の操作をやめたい場合は、以下のコマンドでEC2を抜けることができます。

```
ubuntu@ip-172-31-85-199:~$ exit
```

クイズ

1. IPアドレスが23.45.678.901で、ユーザーがec2-user、SSHの秘密鍵がC:\Users\hoge\Documents\labsuser.pemにある場合はどのようにコマンドを入力しますか。
 - 1.1. sshconnect -i C:\Users\hoge\Documents\labsuser.pem ec2-user@23.45.678.901

³ SSHとは、Secure Shell(セキュアシェル)の略で、リモートコンピュータと通信するためのプロトコルである。認証部分を含めネットワーク上の通信がすべて暗号化されるため、安全に通信することが可能です。従来は、TelnetやFTPなどの手法でリモート通信が行われていましたが、これらはパスワードを暗号化のない平文で送信してしまうため、盗聴のリスクがあります。

- 1.2. `ssh -i C:\Users\hoge\Documents\labsuser.pem 23.45.678.901@ec2-user`
- 1.3. `ssh -i C:\Users\hoge\Documents\labsuser ec2-user@23.45.678.901`
- 1.4. `ssh ec2-user@23.45.678.901 -i C:\Users\hoge\Documents\labsuser.pem`

回答

1. `ssh ec2-user@23.45.678.901 -i C:\Users\hoge\Documents\labsuser.pem`(正解:4)

Linuxコマンド

簡単にLinuxコマンドについて解説します。これらのコマンドを使っていきます。

ls

```
ubuntu@ip-172-31-85-199:/$ ls
bin  dev  home  lib32  libx32  media  opt  root  sbin  srv  tmp  var
boot  etc  lib   lib64  lost+found  mnt   proc  run   snap  sys  usr
```

現在のディレクトリに存在するファイル、ディレクトリを出力します。(lsを入力し、現在のディレクトリにファイルまたはディレクトリがない場合は何も出力されません。)

pwd

```
ubuntu@ip-172-31-85-199:~$ pwd
/home/ubuntu
```

現在のディレクトリの位置を表示します。

cd

```
ubuntu@ip-172-31-85-199:/$ cd /home/ubuntu/
```

ディレクトリを移動します。今回の場合は、/(ルートディレクトリ)から/home/ubuntuディレクトリに移動しています。⁴

touch

```
ubuntu@ip-172-31-85-199:~$ touch main.py
ubuntu@ip-172-31-85-199:~$ ls
main.py
```

ファイルを作成することができます。ここでは、main.pyを作成しています。

⁴ `ubuntu@ip-172-31-85-199:~$ pwd`
`/home/ubuntu`

現在のディレクトリの場所がわからなくなったら、pwdを入力し、現在のディレクトリを確認しましょう。

`ubuntu@ip-172-31-85-199:~$ cd ..`
`cd ..`とすることで1つ上のディレクトリに移動できます。

cp

```
ubuntu@ip-172-31-85-199:~$ cp main.py sample.py
```

main.pyをsample.pyにコピーしています。

mkdir

```
ubuntu@ip-172-31-85-199:~$ mkdir hoge
ubuntu@ip-172-31-85-199:~$ ls
main.py  hoge
```

ディレクトリを作成することができます。ここでは、os-lectureディレクトリを作成しています。

cat

```
ubuntu@ip-172-31-85-199:~$ cat main.py
```

ファイルの中身をターミナルに出力することができます。
main.pyの中身を出力することができます。

>

```
ubuntu@ip-172-31-85-199:~$ echo 'print("Hello World !")' > main.py
```

リダイレクトコマンド。echoで出力した文字列をmain.pyに書き込みます。⁵

<

```
ubuntu@ip-172-31-85-199:~$ cat < main.py
```

上記の>コマンドの逆です。
main.pyの内容をcatコマンドに渡しています。

|

```
ubuntu@ip-172-31-85-199:~$ echo 'ls' | sh
```

パイプコマンドです。|の前のコマンドの内容を|の後のコマンドに渡しています。
echo 'ls'でlsという文字を出力し、shコマンドにlsを渡しています。⁶

⁵ python3 main.pyを実行してみましょう。

UbuntuにはPython3がインストールされています。

⁶ shコマンドは、shellとして実行することを示しています。

--version

```
ubuntu@ip-172-31-85-199:~$ python3 --version
Python 3.10.4
```

ソフトウェアなどのバージョンを確認するコマンドです。⁷

rm

```
ubuntu@ip-172-31-85-199:~$ ls
main.py test
ubuntu@ip-172-31-85-199:~$ rm -r test/
ubuntu@ip-172-31-85-199:~$ ls
main.py
ubuntu@ip-172-31-85-199:~$ rm -f main.py
ubuntu@ip-172-31-85-199:~$ ls
```

ファイル・ディレクトリを削除するコマンドです。
-rのオプションでディレクトリを再帰的に削除できます。
-fのオプションで強制削除することができます。

apt

```
ubuntu@ip-172-31-85-199:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
...
```

APT (Advanced Packaging Tool) とは、ソフトウェアのインストールとアンインストールを自動的に行い、ソフトウェアの管理を簡単に行えるようにするための仕組みの 1 つです。⁸
sudoはroot権限で実行する際にコマンドの前につけます。

クイズ

1. SSHで接続したあとにはどのユーザーで操作を行っていますか。
 - 1.1. ubuntu
 - 1.2. root
 - 1.3. ec2-user
 - 1.4. b2XXXXXX(あなたの学籍番号)

接続したインスタンス内で、これらのコマンドを試してみましょう。コマンド操作に慣れてきたら、アプリケーションの構築を進めていきましょう。

⁷ --versionや-Vなどソフトウェアによって様々です。
うまく動作しないときや、質問する際はバージョンを明記すると、回答者が問題を解決しやすくなる場合があります。

⁸ 今回使用しているUbuntuではaptコマンドによってパッケージをインストールします。
これから紹介するソフトウェア、ミドルウェアをインストールする際にaptコマンドを用います。

Apache をインストール・起動

Apacheとは

Web サーバーの役割を担うソフトウェアです。Web サーバーとは、パソコンやスマートフォンなどの端末から HTTP/HTTPS で送られたリクエストに対して HTML、CSS、JavaScript などの情報を返す役割を持ったソフトウェアです。

ユーザーとアプリケーションの橋渡しの役割を担っています。ユーザーからのアクセスが増加した場合に、待機している複数のサーバーに分散させて処理させることが可能です。

インストールする

aptコマンドを用いてインストールします。

```
ubuntu@ip-172-31-85-199:/var/www/html$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [109 kB]
...
```

sudo apt updateにより、apt installでインストールされるソフトウェアが最新のものになるようにします。

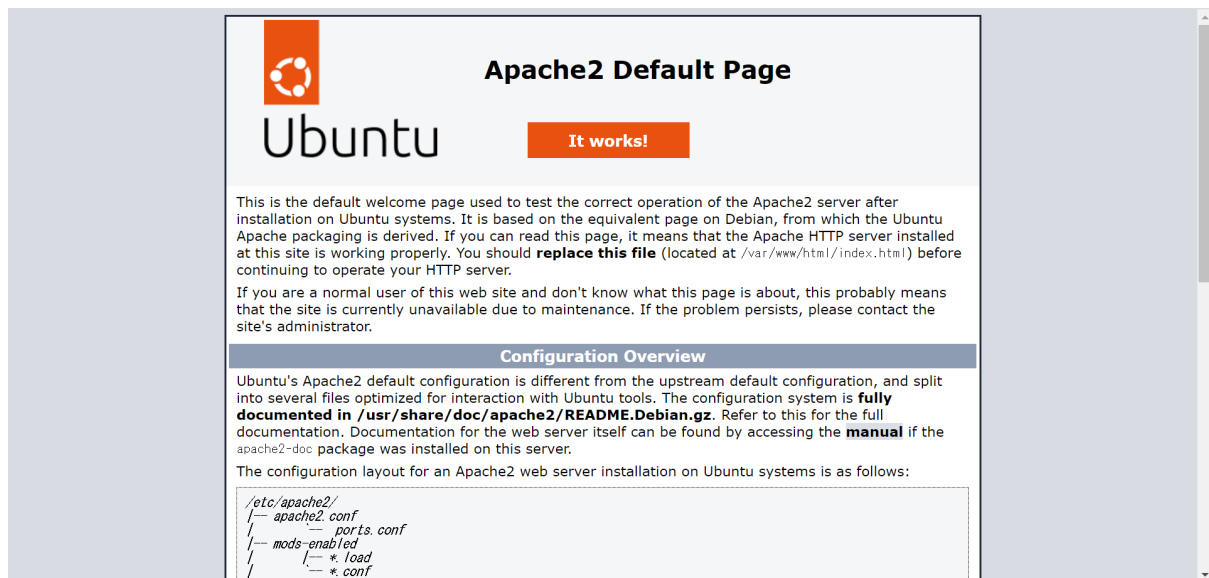
Apacheをインストールします。

```
ubuntu@ip-172-31-85-199:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
...
```

途中で、Do you want to continue? [Y/n]と聞かれるので、Yと入力しましょう。⁹

インストールが完了したら、SSH接続した際に使ったIP アドレスをブラウザに入力します。Apacheが正常にインストールされ、起動している場合は以下のような画面が出ます。

⁹ nと入力すると、インストールせずに終了します。



表示されない場合

Apacheが起動していない可能性があります。Apacheを起動しましょう。

```
ubuntu@ip-172-31-85-199:~$ sudo systemctl start apache2
```

systemctlコマンドはソフトウェアの操作を行うコマンドです。起動や停止¹⁰を行うことができます。

HTMLを公開する

先ほど、ブラウザに表示された画面のソースコードを見てみましょう。この画面は HTML¹¹と CSS¹² というもので表示されています。この HTMLとCSS は、/var/www/htmlディレクトリの index.htmlです。

```
ubuntu@ip-172-31-85-199:~$ cd /var/www/html/  
ubuntu@ip-172-31-85-199:/var/www/html$ cat index.html
```

cdでディレクトリを移動しています。catでファイルの中身を出力させています。

cat index.htmlを実行すると、文字列が出力されます。このコードが画面を構成しています。

HTMLを書き換える

```
ubuntu@ip-172-31-85-199:/var/www/html$ sudo vi index.html
```

¹⁰ 停止させたいときは `sudo systemctl stop apache2`

¹¹ HTML とは、HyperText Markup Language(ハイパーテキスト・マークアップ・ランゲージ)の略であり、拡張子は.htmlです。ウェブページを作成するために開発された言語である。HTML により、ブラウザ上に文字を表示させることができます。

¹² CSS とは、Cascading Style Sheets(カスケーディング・スタイル・シート)の略であり、拡張子は.cssです。HTML で記述された WEB サイトにスタイルを適用する際に用いられます。

viを使ってファイルを書き換えます。
viはターミナルで使うことができるテキストエディタです。

試しに、index.htmlに<h1>Hello World !</h1>と入力します。

vi の使い方

黒い画面が出たら、

1. iと入力する(入力モードになる)
2. ↑→キーでカーソルを移動
3. 自由に編集
4. Escを押下
5. :wqと入力(変更を保存する)
(保存しないで終了する場合は:q!と入力)

変更後、SSH接続した際に使ったIP アドレスをブラウザに入力します。正しく変更できている場合は、以下のような画面になります。

Hello World !

自由にHTMLやCSSを書き換えてみましょう。ファイルを変更したらブラウザをリロードしましょう。

PHPをインストールする

今回のアプリケーションではデータベースに接続するのでPHPの環境を構築します。

PHPとは

PHP は、"PHP: Hypertext Preprocessor" を意味し、広く使用されているオープンソースの汎用スクリプト言語です。HTML に埋め込むことができ、Web アプリケーションの開発に特に適しています。PHP の構文の多くは C、Java、Perl 言語から転用したもので、簡単に習得することができます。¹³

¹³ 参考

インストールする

Apacheをインストールした時と同様にaptコマンドを用いてインストールします。

```
ubuntu@ip-172-31-85-199:/var/www/html$ sudo apt install php
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libapache2-mod-php8.1 php-common php8.1 php8.1-cli php8.1-common
  php8.1-opcache php8.1-readline
Suggested packages:
```

インストールが完了したら、インストールできているか確認します。

```
ubuntu@ip-172-31-85-199:/var/www/html$ php --version
PHP 8.1.2 (cli) (built: Apr 7 2022 17:46:26) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.1.2, Copyright (c) Zend Technologies
with Zend OPcache v8.1.2, Copyright (c), by Zend Technologies
```

PHPのバージョンを確認しています。正常にインストールされていれば上記のような結果になります。今回はバージョン8.1.2がインストールされました。

PHPを公開する

/var/www/htmlディレクトリ¹⁴に移動したら、以下のコマンドを入力しPHPファイルを作成し、編集しましょう。HTMLファイルを編集した時と同様に編集しましょう。

```
ubuntu@ip-172-31-85-199:/var/www/html$ sudo vi index.php
```

index.phpには以下のように入力してください。¹⁵

```
<?PHP
phpinfo();
?>
```

編集が完了したら、SSH接続した際に使ったIP アドレスとIPアドレスの後に/index.phpとブラウザに入力します。正しく編集できている場合は以下のような画面が出ます。

例


12.34.567.890/index.php

<https://www.php.net/manual/ja/index.php>

¹⁴ apacheはこのディレクトリ内のファイルを公開することができます。

¹⁵ phpinfo関数

<https://www.php.net/manual/ja/function.phpinfo.php>

| PHP Version 8.1.2 | |
|---|---|
|  | |
| System | Linux ip-172-31-85-199 5.15.0-1011-aws #14-Ubuntu SMP Wed Jun 1 20:54:22 UTC 2022 x86_64 |
| Build Date | Apr 7 2022 17:46:26 |
| Build System | Linux |
| Server API | Apache 2.0 Handler |
| Virtual Directory Support | disabled |
| Configuration File (php.ini) Path | /etc/php/8.1/apache2 |
| Loaded Configuration File | /etc/php/8.1/apache2/php.ini |
| Scan this dir for additional .ini files | /etc/php/8.1/apache2/conf.d |
| Additional .ini files parsed | /etc/php/8.1/apache2/conf.d/10-opcache.ini, /etc/php/8.1/apache2/conf.d/10-pdo.ini, /etc/php/8.1/apache2/conf.d/20-calendar.ini, /etc/php/8.1/apache2/conf.d/20-ctype.ini, /etc/php/8.1/apache2/conf.d/20-exif.ini, /etc/php/8.1/apache2/conf.d/20-ffi.ini, /etc/php/8.1/apache2/conf.d/20-fileinfo.ini, /etc/php/8.1/apache2/conf.d/20-ftp.ini, /etc/php/8.1/apache2/conf.d/20-gettext.ini, /etc/php/8.1/apache2/conf.d/20-iconv.ini, /etc/php/8.1/apache2/conf.d/20-phar.ini, /etc/php/8.1/apache2/conf.d/20-posix.ini, /etc/php/8.1/apache2/conf.d/20-readline.ini, /etc/php/8.1/apache2/conf.d/20-shmop.ini, /etc/php/8.1/apache2/conf.d/20-sockets.ini, /etc/php/8.1/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.1/apache2/conf.d/20-sysvsem.ini, /etc/php/8.1/apache2/conf.d/20-sysvshm.ini, /etc/php/8.1/apache2/conf.d/20-tokenizer.ini |
| PHP API | 20210902 |
| PHP Extension | 20210902 |
| Zend Extension | 420210902 |
| Zend Extension Build | API420210902 NTS |

サーバーの情報などが表示されます。

自由にindex.phpを書き換えて、表示を変えてみましょう。ファイルを変更したらブラウザをリロードしましょう。

以下のようにindex.phpを変更すると、Hello World ! と出力されます。

```
<?PHP
print("Hello World !");
?>
```

Hello World !

クイズ

1. 新しく/var/www/htmlディレクトリ内にhogeディレクトリを作成し、hogeディレクトリ内にindex.htmlを作成しました。このファイルを作るところからブラウザでアクセスするまでを説明してください。IPアドレスは12.34.567.890です。

- 1.1. `cd /var/www/html`でディレクトリを移動し、`mkdir hoge`でhogeディレクトリを作成する。また、hogeディレクトリ内で、`sudo vi index.html`でindex.htmlを作成して書き込む。最後に、ブラウザに12.34.567.890/index.htmlと入力しアクセスする。
- 1.2. `cd /var/www/html`でディレクトリを移動し、`mkdir hoge`でhogeディレクトリを作成する。また、hogeディレクトリ内で、`vi index.html`でindex.htmlを作成して書き込む。最後に、ブラウザに12.34.567.890/hoge/index.htmlと入力しアクセスする。
- 1.3. `mkdir hoge`でhogeディレクトリを作成する。また、hogeディレクトリ内で、`sudo vi index.html`でindex.htmlを作成して書き込む。最後に、ブラウザに12.34.567.890/hoge/index.htmlと入力しアクセスする。
- 1.4. `cd /var/www/html`でディレクトリを移動し、`mkdir hoge`でhogeディレクトリを作成する。また、hogeディレクトリ内で、`sudo vi index.html`でindex.htmlを作成して書き込む。最後に、ブラウザに12.34.567.890/hoge/index.htmlと入力しアクセスする。

MySQLをインストールする

データベースを準備します。今回のアプリケーションでは、商品を管理するために用います。

MySQLとは

MySQLはデータベース¹⁶の役割を担うソフトウェアです。データベースはデジタルでデータを保持し、膨大なアクセスに対しても整合性を保ったままデータを管理することが可能な機能を持ったソフトウェアです。

インストールする

MySQLも他のソフトウェアと同様にaptコマンドを用いてインストールします。

```
ubuntu@ip-172-31-85-199:~$ sudo apt install mysql-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
...
```

途中で、Do you want to continue? [Y/n]と聞かれるので、Yと入力しましょう。

インストールが完了したら、インストールできているか確認します。

```
ubuntu@ip-172-31-85-199:~$ mysql --version
mysql Ver 8.0.29-0ubuntu0.22.04.2 for Linux on x86_64 ((Ubuntu))
```

MySQLのバージョンを確認しています。正常にインストールされていれば上記のような結果になります。今回はバージョン8.0.29がインストールされました。

¹⁶「データベースの定義」

<https://www.oracle.com/jp/database/what-is-database/>

SQLを実行する

MySQLをインストールできたので、SQLで操作します。データベースをコマンドラインから操作してみましょう。¹⁷

```
ubuntu@ip-172-31-85-199:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.29-0ubuntu0.22.04.2 (Ubuntu)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

sudo mysqlと入力し、MySQLのコンソールに入りましょう。すると、ターミナルの入力がmysql>となっています。ここからデータベースを操作できます。

試しに、userテーブルを確認してみましょう。

```
mysql> SELECT Host, User FROM mysql.user;
+-----+-----+
| Host   | User           |
+-----+-----+
| localhost | debian-sys-maint |
| localhost | mysql.infoschema |
| localhost | mysql.session   |
| localhost | mysql.sys       |
| localhost | root            |
+-----+-----+
5 rows in set (0.00 sec)
```

SQLが実行され、テーブルの内容が出力されていることが分かります。

mysqlコンソールを抜けましょう。

```
mysql> exit
Bye
ubuntu@ip-172-31-85-199:~$
```

¹⁷ MySQLをrootでアプリケーションから操作することは危険です。ユーザーを作成して、操作しましょう。SQLインジェクションなどで被害にあう可能性が高まります。

「安全なウェブサイトの作り方 - 1.1 SQLインジェクション」

https://www.ipa.go.jp/security/vuln/websecurity-HTML-1_1.html

これで、MySQLをコマンドラインから操作できるようになりました。

MySQLを操作(初期設定)

MySQL の初期設定には安全ではない設定があるので、変更します。

root ユーザーのパスワードの変更する

初期設定では、root ユーザーのパスワードがデフォルト値に設定されています。これでは、攻撃者から容易に推測されてしまうため、root ユーザーにパスワードを設定します。

mysqlのコンソールに入り、以下のコマンドを入力してください。

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password by 'rootのパスワード';
```

rootのパスワードは好きなパスワードに設定してください。

注意

rootのような簡単なパスワードは設定できません。

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password by 'root';  
ERROR 1819 (HY000): Your password does not satisfy the current policy requirements  
mysql>
```

特殊文字(#\$%&など)、大文字(ABCなど)、小文字(abcなど)、数字(123など)を用い、8文字以上のパスワードにしてください。¹⁸

初期設定を変更しセキュリティを向上させる

mysql_secure_installationコマンドにより、MySQL に安全な設定をコマンドから対話的にを行います。

```
ubuntu@ip-172-31-85-199:~$ sudo mysql_secure_installation  
  
Securing the MySQL server deployment.  
  
Enter password for user root:  
The 'validate_password' component is installed on the server.  
The subsequent steps will run with the existing configuration  
of the component.  
Using existing password for root.  
  
Estimated strength of the password: 100
```

¹⁸ mysqlのrootのパスワード

<https://dev.mysql.com/doc/refman/8.0/ja/validate-password.html>

Change the password for root ? ((Press y|Y for Yes, any other key for No) : n

... skipping.

By default, a MySQL installation has an anonymous user, allowing anyone to log into MySQL without having to have a user account created for them. This is intended only for testing, and to make the installation go a bit smoother. You should remove them before moving into a production environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : Y
Success.

Normally, root should only be allowed to connect from 'localhost'. This ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : Y
Success.

By default, MySQL comes with a database named 'test' that anyone can access. This is also intended only for testing, and should be removed before moving into a production environment.

Remove test database and access to it? (Press y|Y for Yes, any other key for No) : Y
- Dropping test database...
Success.

- Removing privileges on test database...
Success.

Reloading the privilege tables will ensure that all changes made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : Y
Success.

All done!

設定の内容を解説します。

設定の詳細

1. Change the password for root ? ((Press y|Y for Yes, any other key for No) : n
root ユーザーのパスワードを変更するかどうか。しない。
2. Remove anonymous users? (Press y|Y for Yes, any other key for No) : Y
誰でも MySQL にログインできる状態をできないように変更する。
3. Disallow root login remotely? (Press y|Y for Yes, any other key for No) : Y
root ユーザーのサーバー外からのアクセスを拒否する。

4. Remove test database and access to it? (Press y|Y for Yes, any other key for No) : Y
test データベースを作成しない。
5. Reload privilege tables now? (Press y|Y for Yes, any other key for No) : Y
設定を即時反映させる。

MySQL に入る

先ほど、MySQL の root ユーザーに設定したパスワード(ルートのパスワード)を引数で渡し、MySQLのコンソールに入ります。

```
ubuntu@ip-172-31-85-199:~$ mysql -uroot -pルートのパスワード
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.29-0ubuntu0.22.04.2 (Ubuntu)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

-uはユーザー名を示しており、-pはユーザーのパスワードを示しています。
これで、rootユーザーのパスワードが変更できていることが確認できました。

注意

-u、-pの後のユーザー名やパスワードは間にスペースを入れないで入力してください。

MySQLを操作する(データベース・テーブル)

ここから、今回作成するアプリケーションのデータを管理するためのテーブルを作成し、商品データを入れていきます。

MySQLのユーザーを作成

アプリケーションからrootユーザーとしてデータベースにアクセスすることもできますが、セキュリティ上で問題があります。そのため、データベースを操作するための権限を絞ったユーザーを作成します。このユーザーでアプリケーションからデータベースを操作します。

```
mysql> create user 'webapp'@'localhost' identified by 'webappのパスワード';
Query OK, 0 rows affected (0.01 sec)

mysql>
```

- ユーザー名をwebapp
- ホスト名をlocalhost
- パスワードをwebappのパスワード

で作成します。

webappユーザーもrootユーザーと同様に複雑なパスワードにします。パスワードは自身で入力してください。

ユーザーが作成できているか確認する

```
mysql> SELECT Host, User FROM mysql.user;
+-----+-----+
| Host   | User           |
+-----+-----+
| localhost | debian-sys-maint |
| localhost | mysql.infoschema |
| localhost | mysql.session   |
| localhost | mysql.sys       |
| localhost | root            |
| localhost | webapp          |
+-----+-----+
6 rows in set (0.00 sec)
```

正しくwebappユーザーが作成されていることが確認できます。

クイズ

1. hogeユーザーをホスト名localhostで作成したい場合、どのようなコマンドを入力しますか。
 - 1.1. mysql> create user 'hoge'@'localhost' identified by 'hogeのパスワード';
 - 1.2. mysql> create user 'localhost'@'hoge' identified by 'hogeのパスワード';
 - 1.3. mysql> create user 'hoge'@'localhost' password is 'hogeのパスワード';
 - 1.4. mysql> createUser 'hoge'@'localhost' identified by 'hogeのパスワード';
2. mysqlのコンソールにhogeユーザーで入るにはどのようなコマンドを入力しますか。
 - 2.1. ubuntu@ip-172-31-85-199:/etc\$ mysql -u hoge -p hogeのパスワード
 - 2.2. ubuntu@ip-172-31-85-199:/etc\$ mysql uhoge phogeのパスワード
 - 2.3. ubuntu@ip-172-31-85-199:/etc\$ mysql user hoge password hogeのパスワード
 - 2.4. ubuntu@ip-172-31-85-199:/etc\$ mysql -uhoge -phogeのパスワード

データベースを作成する

今回作成するアプリケーションで使うデータを管理するデータベースを作成します。¹⁹
作成するデータベース内にテーブルを作成し、商品データを管理します。

MySQLのrootユーザーでデータベースやテーブルを作成してください。

¹⁹ 13.1.12 CREATE DATABASE ステートメント
<https://dev.mysql.com/doc/refman/8.0/ja/create-database.html>


```
mysql> create database webapp;  
Query OK, 1 row affected (0.01 sec)
```

```
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| sys |  
| webapp |  
+-----+  
5 rows in set (0.00 sec)
```

create databaseコマンドでデータベースを作成します。webappデータベースを作成します。

show databasesコマンドで存在するデータベースを表示しています。webappデータベースが作成されていることが確認できます。

webappデータベース内に商品を管理するためのテーブルを作成します。

テーブルを作成する

今回作成するアプリケーションで使うデータを管理するテーブルを作成します。²⁰

```
mysql> create table webapp.product (name varchar(255), price int);  
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> show tables from webapp;  
+-----+  
| Tables_in_webapp |  
+-----+  
| product |  
+-----+  
1 row in set (0.00 sec)
```

create tableコマンドでwebappデータベースにproductテーブルを作成しています。

show tables from webappでwebappデータベース内のテーブルを表示させています。webappデータベース内にproductテーブルが作成できていることが確認できました。

SQLについて

今回用いたSQLについて紹介します。

```
create table webapp.product (name varchar(255), price int);
```

²⁰ 13.1.20 CREATE TABLE ステートメント
<https://dev.mysql.com/doc/refman/8.0/ja/create-table.html>

このSQLはテーブルを作成するSQLです。

1. create tableでテーブルを作成することを示しています。
2. webapp.productでwebappデータベース内にproductテーブルを作成することを示しています。
3. (name varchar(255), price int)でproductテーブルの形を示しています。

productテーブルの形は以下のようになっています。

| | |
|------|-------|
| name | price |
|------|-------|

nameカラムがvarchar(255)²¹、priceカラムがint型です。nameに商品名、priceに商品の価格のデータが保存されます。

varchar(255)には255文字のデータを保存することができます。

クイズ

1. webappデータベース内にuserテーブルを作成します。userテーブルはnameカラム varchar(255)、ageカラムはint型を持っています。userテーブルを作成する場合はどのようなSQLを入力しますか。
 - 1.1. create table webapp.user (varchar(255) name, int age);
 - 1.2. create table webapp.user (name varchar(255), age int);
 - 1.3. create table user (name varchar(255), age int);
 - 1.4. create table webapp.user (name varchar(255), age number);

テーブルにデータを追加する

作成したwebappデータベース内のproductテーブルに商品データを追加していきます。

MySQLのrootユーザーでデータを追加してください。

```
mysql> select * from webapp.product;
Empty set (0.01 sec)

mysql> INSERT INTO webapp.product(name, price) VALUES ('apple', 100);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO webapp.product(name, price) VALUES ('pen', 200);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO webapp.product(name, price) VALUES ('pineapple', 500);
Query OK, 1 row affected (0.01 sec)

mysql> select * from webapp.product;
+-----+-----+
```

²¹ 11.4.1 CHAR および VARCHAR 型
<https://dev.mysql.com/doc/refman/8.0/ja/char.html>

```
| name    | price |
+-----+-----+
| apple   | 100   |
| pen     | 200   |
| pineapple | 500   |
+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

1. `select * from webapp.product;`
webappデータベース内のproductテーブルの中身を確認してます。データがないので、Emptyと出ています。
2. `INSERT INTO webapp.product(name, price) VALUES ('apple', 100);`
テーブルにデータを追加するSQLです。webappデータベース内のproductテーブルにnameがappleでpriceが100のデータを1つ追加しています。
3. 同様にnameがpen、priceが 200のデータ、nameがpineapple、priceが 500のデータを追加しています。
4. `select * from webapp.product;`
webappデータベース内のproductテーブルの中身を確認してます。データが追加されていることが確認できます。

これで、サンプルの商品データを追加できました。
自由にデータを追加してみましょう。また、データの削除なども試してみましょう。

このデータをPHPのアプリケーションで表示させます。また、アプリケーションから商品データの追加や削除もできるようにします。

クイズ

1. webappデータベース内のproductテーブルにnameがpeach、priceが 300のデータを追加する場合はどのようなSQLを入力しますか。
 - 1.1. `INSERT INTO webapp.product(name, price) VALUES (300,'peach');`
 - 1.2. `INSERT INTO webapp.product(name 'peach', price 300);`
 - 1.3. `INSERT INTO webapp.product(name, price) VALUES ('peach',300);`
 - 1.4. `INSERT INTO product(name, price) VALUES ('peach',300);`
2. webappデータベース内のproductテーブルにnameがpeach、priceが 300のデータを削除する場合はどのようなSQLを入力しますか。
 - 2.1. `DELETE FROM webapp.product WHERE name = 'peach' AND price=300;`
 - 2.2. `DELETE FROM product WHERE name = 'peach' AND price=300;`
 - 2.3. `DELETE FROM webapp.product WHERE name = 'peach' OR price=300;`
 - 2.4. `DELETE FROM webapp.product WHERE 'peach' AND 300;`

ファイルに書かれたSQLを実行する

mysqlコンソールからSQLを実行してきましたが、ファイルに書かれたSQLを実行してMySQLに指示を与えることができます。linuxコマンドで紹介したコマンドを用います。

cd コマンドでホームディレクトリに移動してください。

```
ubuntu@ip-172-31-85-199:~$ cd /home/ubuntu/  
ubuntu@ip-172-31-85-199:~$ pwd  
/home/ubuntu
```

ホームディレクトリでSQLファイルを作成し、以下のようなSQLを入力し、保存してください。

▼create_table.sql

```
DROP TABLE IF EXISTS webapp.product;  
CREATE TABLE webapp.product (name varchar(255), price int);
```

DROP TABLE IF EXISTS webapp.product;

このSQLは、もしwebappデータベース内にproductテーブルが存在していたらproductテーブルを削除するというSQLです。

▼insert_data.sql

```
INSERT INTO webapp.product(name, price) VALUES ('apple', 100);  
INSERT INTO webapp.product(name, price) VALUES ('pen', 200);  
INSERT INTO webapp.product(name, price) VALUES ('pineapple', 500);
```

作成後、以下のコマンドを実行してください。

```
ubuntu@ip-172-31-85-199:~$ mysql -uroot -pルートのパスワード < create_table.sql  
mysql: [Warning] Using a password on the command line interface can be insecure.  
ubuntu@ip-172-31-85-199:~$ mysql -uroot -pルートのパスワード < insert_data.sql  
mysql: [Warning] Using a password on the command line interface can be insecure.22
```

このようにすることで、mysqlに入ることなく、SQL を実行することができます。

コマンド < ファイル名 により、ファイルの中の指示を実行するコマンドに与えています。
ここでは、mysqlのルートユーザーとして、.sqlのファイルの中のSQLを実行しています。

クイズ

1. <を使わずにコマンドを書くとするとどのようなコマンドになりますか。|を使って書いてみてください。
 - 1.1. mysql -uroot -pルートのパスワード | cat insert_data.sql
 - 1.2. cat insert_data.sql >> mysql -uroot -pルートのパスワード
 - 1.3. cat 'mysql -uroot -pルートのパスワード' > insert_data.sql
 - 1.4. cat insert_data.sql | mysql -uroot -pルートのパスワード

回答

²² コマンドラインにmysqlのルートパスワードが入力されているので警告が出ています。

1. `cat insert_data.sql | mysql -uroot -p`ルートのパスワード
(補足) パイプ(`|`)により、`cat insert_data.sql`の結果をパイプの後ろのコマンドに渡しています。(正解:4)

MySQLのユーザーに権限を与える

MySQLのrootユーザーで実行してください。

rootユーザーで作成したwebappデータベースやproductテーブルを操作する権限がwebappユーザーにはありません。実際に権限がないことを確認しましょう。²³

```
mysql> SHOW GRANTS FOR 'webapp'@'localhost';
+-----+
| Grants for webapp@localhost          |
+-----+
| GRANT USAGE ON *.* TO `webapp`@`localhost` |
+-----+
1 row in set (0.00 sec)
```

1. `SHOW GRANTS FOR 'webapp'@'localhost';`
ユーザーの権限を確認するコマンドです。ユーザー名がwebapp、ホスト名がlocalhostのユーザーの権限を確認しています。

webapp ユーザーには、権限が付与されていないことが確認できます。
USAGEは権限が何も無いことを示します。

ユーザーに権限を付与する

アプリケーションからはwebappユーザーとして操作します。そのため、webappユーザーはwebappデータベース内のproductテーブルを操作する権限が必要になります。そのため、webappユーザーにproductテーブルを操作する権限を付与します。

```
mysql> GRANT DELETE, INSERT, SELECT, UPDATE ON webapp.product TO
'webapp'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql>
```

1. `GRANT DELETE, INSERT, SELECT, UPDATE ON webapp.product TO 'webapp'@'localhost';`
ユーザー名がwebapp、ホスト名がlocalhostのユーザーにwebappデータベース内のproductテーブルのDELETE(カラムの削除), INSERT(データの追加), SELECT(データの取得), UPDATE(データの更新)の権限を与える設定をしています。

²³ 6.2.2 MySQL で提供される権限

<https://dev.mysql.com/doc/refman/8.0/ja/privileges-provided.html>

13.7.1.6 GRANT ステートメント

<https://dev.mysql.com/doc/refman/8.0/ja/grant.html>

確認する

ユーザー名がwebapp、ホスト名がlocalhostのユーザーに権限を付与できたか確認します。

```
mysql> SHOW GRANTS FOR 'webapp'@'localhost';
+-----+
| Grants for webapp@localhost |
+-----+
| GRANT USAGE ON *.* TO `webapp`@`localhost` |
| GRANT SELECT, INSERT, UPDATE, DELETE ON `webapp`.`product` TO |
| `webapp`@`localhost` |
+-----+
2 rows in set (0.01 sec)
```

1. SHOW GRANTS FOR 'webapp'@'localhost';
ユーザーの権限を確認するコマンドです。ユーザー名がwebapp、ホスト名がlocalhostのユーザーの権限を確認しています。

webapp ユーザーにwebappデータベースのproductテーブルのDELETE(カラムの削除), INSERT(データの追加), SELECT(データの取得), UPDATE(データの更新)の権限を付与できてることが確認しました。

クイズ

1. webappデータベース内にuserテーブルを作成してください。userテーブルはnameカラム varchar(255)、ageカラムはint型を持っています。
 - 1.1. create table user (name varchar(255), age int);
 - 1.2. createTable webapp.user (name varchar(255), age int);
 - 1.3. create table webapp.user (name, age);
 - 1.4. create table webapp.user (name varchar(255), age int);
2. ユーザー名がwebapp、ホスト名がlocalhostのユーザーにwebappデータベース内のuserテーブルに対して、DELETE(カラムの削除), INSERT(データの追加), SELECT(データの取得), UPDATE(データの更新)の権限を与える設定をしてください。
 - 2.1. GRANT ALL ON webapp.user TO 'webapp'@'localhost';
 - 2.2. GRANT DELETE, INSERT, SELECT, UPDATE ON webapp.user TO 'localhost'@'webapp';
 - 2.3. GRANT DELETE, INSERT, SELECT, UPDATE ON user TO 'webapp'@'localhost';
 - 2.4. GRANT DELETE, INSERT, SELECT, UPDATE ON webapp.user TO 'webapp'@'localhost';

アプリケーションからデータベースのデータを取得する

データを管理するためのデータベースの準備はできました。アプリケーションからデータベース内の商品データを取得し、ブラウザの画面に表示させましょう。

mysqlコンソールで操作している方はexitでmysqlコンソールを抜けましょう。

```
mysql> exit
Bye
ubuntu@ip-172-31-85-199:~$
```

アプリケーションからデータベースにアクセスできるようにする

アプリケーションからデータベースを操作するために必要なPHPモジュールをインストールします。

今回はPHPのアプリケーションからMySQLを操作するので、以下をインストールします。

- php-pdo²⁴
- php-mysql²⁵

php-pdoをインストールする

aptコマンドを用いてインストールします。

```
ubuntu@ip-172-31-85-199:/var/www/html$ sudo apt install php-pdo
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
...
```

php-mysqlをインストールする

aptコマンドを用いてインストールします。

```
ubuntu@ip-172-31-85-199:/var/www/html$ sudo apt install php-mysql
Do you want to continue? [Y/n] Y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64
php8.1-mysql amd64 8.1.2-1ubuntu2.1 [130 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 php-mysql all
2:8.1+92ubuntu1 [1834 B]
Fetched 132 kB in 0s (2329 kB/s)
...
```

これでPHPのアプリケーションからMySQLに接続できるようになりました。

アプリケーションを作成する

Apacheで公開されているディレクトリである/var/www/htmlディレクトリに移動してください。

```
ubuntu@ip-172-31-85-199:~$ cd /var/www/html/
```

²⁴ PDO クラス

<https://www.php.net/manual/ja/class.pdo.php>

²⁵ PHPがMySQLと通信できるようにするために必要です。

cdコマンドでディレクトリを移動します。

/var/www/htmlディレクトリに移動したら、以下のコマンドを入力しPHPファイルを編集しましょう。
PHPファイルを編集した時と同様に編集しましょう。

viエディタの使い方は前に紹介したので、戻って確認してください。

```
ubuntu@ip-172-31-85-199:/var/www/html$ sudo vi index.php
```

index.phpには以下のように入力してください。

```
<?php

$dsn = 'mysql:dbname=webapp;host=localhost';
$user = 'webapp';
$password = 'webappのパスワード';

try {
    $dbh = new PDO($dsn, $user, $password);
} catch (PDOException $e) {
    print('Error: ' . $e->getMessage());
    die();
}

// POST
if ($_POST['method'] == 'post') {
    $stmt = $dbh->prepare("INSERT INTO product (name, price) VALUES (:name, :price)");
    $stmt->bindParam(':name', $_POST['name'], PDO::PARAM_STR);
    $stmt->bindParam(':price', $_POST['price'], PDO::PARAM_INT);
    $res = $stmt->execute();

    header('Location: index.php');
}

// DELETE
if ($_POST['method'] == 'delete') {
    $stmt = $dbh->prepare("DELETE FROM product WHERE name = :name AND price=:price");
    $stmt->bindParam(':name', $_POST['name'], PDO::PARAM_STR);
    $stmt->bindParam(':price', $_POST['price'], PDO::PARAM_INT);
    $res = $stmt->execute();

    header('Location: index.php');
}

?>
<!DOCTYPE html>
<html lang="ja">

<head>
```



```

<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>売店 | 管理画面</title>
</head>

<body>

  <form method="post" action="index.php">
    <input name="method" type="hidden" value="post">
    商品名
    <input type="text" name="name" size="15">
    価格(円)
    <input type="number" name="price">
    <input type="submit" value="送信">
  </form>

  <?php
  // GET
  $sql = 'select * from product';
  print('<p>商品名 価格(円)</p>');
  foreach ($dbh->query($sql) as $row) {
    ?>
    <form method="post" action="index.php">
      <?php print($row['name'] . ' ' . $row['price'] . '円 '); ?>
      <input name="method" type="hidden" value="delete">
      <input name="name" type="hidden" value="<?php print($row['name']); ?>">
      <input name="price" type="hidden" value="<?php print($row['price']); ?>">
      <input type="submit" value="削除">
    </form>
  <?php
  }
  ?>

</body>

</html>

```

編集が完了したら、SSH接続した際に使ったIP アドレスとIPアドレスの後に/index.phpとブラウザに入力します。正しく編集できている場合は以下のような画面が出ます。

例

12.34.567.890/index.php

商品名 価格(円)

商品名 価格(円)

ノート 100円

送信ボタン、削除ボタンが正常に動作するか確認してください。正常に、商品の追加や削除ができたなら完成です。

アプリケーションからデータの削除や追加を行い、mysqlコンソールからデータの確認を行ってみましょう。データが変化していることが確認できます。

ソースコードの解説

最後にPHPのソースコードの解説をします。

このアプリケーションでは、PHP からデータベースにアクセスし、データの作成、データの削除、データの取得をしています。それぞれ解説します。

データベースに接続する

```
$dsn = 'mysql:dbname=webapp;host=localhost';
$user = 'webapp';
$password = 'webappのパスワード';

try {
    $dbh = new PDO($dsn, $user, $password);
} catch (PDOException $e) {
    print('Error: ' . $e->getMessage());
    die();
}
```

ここでデータベースに接続しています。接続するためには、PDOにデータの名前、ホスト名、MySQL のユーザーのパスワードを指定します。

データを取得する

```
<?php
// GET
$sql = 'select * from product';
print('<p>商品名 価格(円)</p>');
foreach ($dbh->query($sql) as $row) {
?>
    <form method="post" action="index.php">
        <?php print($row['name'] . ' ' . $row['price'] . '円 '); ?>
    </form>
<?php
}
?>
```

SQL をPDOクラスのqueryメソッドの引数に渡します。そうすることで、SQL によりデータを取得することができます。

SQLのselect * from product;でproductテーブルから商品データをすべて取得し、foreachで商品を表示しています。

データを追加する

```
<form method="post" action="index.php">
    <input name="method" type="hidden" value="post">
    商品名
    <input type="text" name="name" size="15">
    価格(円)
    <input type="number" name="price">
    <input type="submit" value="送信">
</form>
```

HTML のフォーム²⁶からデータをPOSTします。

1. <form method="post" action="index.php">
method にはpostを指定します。
action にはindex.phpを指定します。
(データを自分自身(index.php)に POST します)
2. <input type="text" name="name" size="15">
name には\$_POST['name']で受け取るための名前を指定します。

name をpriceにして POST すると、PHP では\$_POST['price']²⁷で値を取得することができます。

```
// POST
```

²⁶ フォーム要素

<https://developer.mozilla.org/ja/docs/Web/HTML/Element/form>

²⁷ \$_POST

<https://www.php.net/manual/ja/reserved.variables.post.php>

```

if ($_POST['method'] == 'post') {
    $stmt = $dbh->prepare("INSERT INTO product (name, price) VALUES (:name, :price)");
    $stmt->bindParam(':name', $_POST['name'], PDO::PARAM_STR);
    $stmt->bindParam(':price', $_POST['price'], PDO::PARAM_INT);
    $res = $stmt->execute();

    header('Location: index.php');
}

```

1. `$stmt = $dbh->prepare("INSERT INTO product (name, price) VALUES (:name, :price)");`
不完全な SQL をPDOクラスのprepareメソッドの引数に渡します。
2. `$stmt->bindParam(':name', $_POST['name'], PDO::PARAM_STR);`
不完全な SQL の `:name` に `$_POST['name']` を入れます。
(PDO::PARAM_STRは`$_POST['name']`が string(文字)であることをしめています。)
3. `$res = $stmt->execute();`
完全な SQL になったので実行します。

データを削除する

HTML のフォームからデータをPOSTします。

```

<form method="post" action="index.php">
    <input name="method" type="hidden" value="delete">
    <input name="name" type="hidden" value="<?php print($row['name']); ?>">
    <input name="price" type="hidden" value="<?php print($row['price']); ?>">
    <input type="submit" value="削除">
</form>

```

`<input name="method" type="hidden" value="post">`
`type="hidden"`を指定することで、`<input>`タグを非表示にすることができます。

```

// DELETE
if ($_POST['method'] == 'delete') {
    $stmt = $dbh->prepare("DELETE FROM product WHERE name = :name AND
price=:price");
    $stmt->bindParam(':name', $_POST['name'], PDO::PARAM_STR);
    $stmt->bindParam(':price', $_POST['price'], PDO::PARAM_INT);
    $res = $stmt->execute();

    header('Location: index.php');
}

```

データを追加するときと同様に、SQLのDELETEでデータを削除しています。

ソースコードをGitHubにpushする

作成したアプリケーションのソースコードを GitHub にアップロードします。GitHubにソースコードをアップロードする際にGit²⁸コマンドを用います。

GitHubのアカウントがない方は、アカウントを作成してください。

▼アカウントを作成

<https://github.com/>

レポジトリを作成する

以下のサイトを参考にレポジトリを作成してください。

<https://docs.github.com/ja/get-started/quickstart/create-a-repo>

注意

- レポジトリを作成する際に[Initialize this repository with a README] は選択しないでください。

作成したレポジトリにアプリケーションのソースコードをアップロードします。

EC2 から GitHub に SSH 接続する

EC2 内で SSH 鍵を作成します。GitHubにソースコードをアップロードする際にSSHを用いてアップロードします。

/home/ubuntu/.sshディレクトリに移動します。

.sshディレクトリがない時は、mkdirコマンドでディレクトリを作成してください。

```
ubuntu@ip-172-31-85-199:~$ mkdir .ssh
ubuntu@ip-172-31-85-199:~$ ls -a
. .. .ssh
```

/home/ubuntu/.sshでディレクトリ内でSSH鍵を作成しましょう。

```
ubuntu@ip-172-31-85-199:~/.ssh$ ssh-keygen -f github
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in github
Your public key has been saved in github.pub
The key fingerprint is:
SHA256:71RR1riBd30VdHBsZHT2QZZ/aUH2JROibGScSkPH
ubuntu@ip-172-31-85-199
The key's randomart image is:
+---[RSA 3072]-----+
|    ..o+...@^/|
```

²⁸ Git

<https://git-scm.com/>

```
|      E=..+*O#|
|      . ++o.. =B|
|      ..o  ..oo|
|      S.  ...|
|      ..      |
|      o      |
|      o      |
|      .      |
|+----[SHA256]-----+
```

ssh-keygenコマンドでSSHの鍵を作成します。

-fで SSH 鍵のファイル名を指定します。今回はgithub という名前で作成します。

- Enter passphrase (empty for no passphrase):
パスワードの設定をすることができます。パスワードを設定すると、この鍵を用いてSSH
する際にパスワードを求められます。忘れないようにしましょう。

lsコマンドでファイルを確認すると、github、github.pubが作成できていることが確認できます。

- github
SSHの秘密鍵です。公開ないようにしましょう。
- github.pub
SSHの公開鍵です。GitHubに設定します。

SSHでは公開鍵と秘密鍵を用いて認証を行います。

公開鍵をGitHubに設定する

SSH 鍵の公開鍵であるgithub.pubの中身をコピーし、GitHub に登録してください。

▼登録方法

<https://docs.github.com/ja/enterprise-server@3.2/authentication/connecting-to-github-with-ssh/adding-a-new-ssh-key-to-your-github-account>

```
ubuntu@ip-172-31-85-199:~/.ssh$ cat github.pub
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQGC61wt66jVW24odHtZMIJJeAc/2ZrLqKiYi/kk0y
+u5jEk0nR+g7bZsJtaBC8sJPYBtkJB/EgTM8c8Clnzhw4Gt2uD3/3VaLOhuZTta9C0NtisEZ
WXJ...= ubuntu@ip-172-31-85-199
```

EC2内の秘密鍵とGitHubに設定した公開鍵を用いて認証を行います。公開鍵と秘密鍵の認証が通過するとGitHubにソースコードをアップロードすることができます。

SSHの設定ファイルを書く

SSH の設定を書きます。/home/ubuntu/.sshディレクトリにconfigというファイルを作成し、以下のように入力してください。

```
Host github.com
  HostName github.com
  User git
  IdentityFile /home/ubuntu/.ssh/github
```

configファイルの説明をします。

1. Hostはなんでもかまいません。
(次で説明するssh github.comでgithub.comにあたる名前です。今回は、github.comを指定します。)
2. HostNameは、ドメイン名や IP アドレスを書きます。
(今回はGitHubなのでgithub.comと書きます。)
3. Userは、SSH 接続する際のユーザー名を記載します。
(今回はgitを指定します。)
4. IdentityFileは、SSH 接続する際の SSH 鍵の場所を書きます。
(今回は/home/ubuntu/.ssh/githubを指定します。)

設定ファイルを書くことで、長いコマンドを入力する必要がなくなります。

▼以下のような長いコマンド

```
$ ssh -i C:\Users\1234\Documents\labsuser.pem ubuntu@12.34.567.890
```

接続をテストする²⁹

GitHubとSSHができるかを確認します。確認できればソースコードをアップロードすることができます。

```
ubuntu@ip-172-31-85-199:~/.ssh$ ssh -T github.com
```

自分のユーザー名が出力されたら成功です。他のメッセージが出力された場合は、「SSHの設定ファイルを書く」を見直してください。

configファイルを書いたため、SSHの鍵の場所やユーザー名などを指定することなく、短いコマンドでSSHの確認ができました。

GitHubにソースコードをアップロードする

ubuntuユーザーの home ディレクトリ(/home/ubuntu)に/var/www/htmlをコピーしましょう。linuxコマンドで紹介したcpコマンドを使います。

```
ubuntu@ip-172-31-85-199:/var/www/html$ cp -r /var/www/html/ /home/ubuntu/
```

-rをつけることで、コピー元にディレクトリを指定した場合、再帰的にコピーすることができます。

²⁹SSH 接続をテストする

<https://docs.github.com/ja/authentication/connecting-to-github-with-ssh/testing-your-ssh-connection>

lsコマンドでindex.phpとindex.htmlが/home/ubuntu/html内にコピーされていることが確認できます。

Gitコマンド

/home/ubuntu/html内のファイルをGitHubにアップロードします。Gitコマンドを使っていきます。
/home/ubuntu/htmlディレクトリでコマンドを入力してください。

git init

```
ubuntu@ip-172-31-85-199:~/html$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint: git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint: git branch -m <name>
Initialized empty Git repository in /home/ubuntu/html/.git/
```

gitの初期化を行っています。

git add

```
ubuntu@ip-172-31-85-199:~/html$ git add index.html index.php
```

アップロード(push)する準備をindex.htmlとindex.phpファイルに行っています。

git commit

```
ubuntu@ip-172-31-85-199:~/html$ git commit -m "first commit"
[master (root-commit) 2bda837] first commit
Committer: Ubuntu <ubuntu@ip-172-31-85-199.ec2.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:
    git config --global user.name "Your Name"
    git config --global user.email you@example.com
After doing this, you may fix the identity used for this commit with:
    git commit --amend --reset-author
...
```

pushする前にaddしたファイル(今回はindex.html、index.php)にどのような変更を加えたかメッセージを残します。今回は最初のコミットということで、メッセージを「first commit」としています。

git remote add

```
ubuntu@ip-172-31-85-199:~/html$ git remote add origin git@github.com:ユーザー名/レポ  
ジトリ名.git
```

push先のレポジトリを登録します。GitHubで作成したアカウント名、リポジトリ名に差し替えて実行してください。

git push

```
ubuntu@ip-172-31-85-199:~/html$ git push origin master  
Enumerating objects: 12, done.  
Counting objects: 100% (12/12), done.  
Compressing objects: 100% (11/11), done.  
Writing objects: 100% (12/12), 3.14 KiB | 1.57 MiB/s, done.  
Total 12 (delta 1), reused 0 (delta 0), pack-reused 0  
remote: Resolving deltas: 100% (1/1), done.  
To github.com:OHMORIYUSUKE/ec2_test.git  
* [new branch]    master -> master
```

GitHubにアップロード(Push)するコマンドです。

最後に作成したレポジトリを確認し、ソースコードがアップロードされているか確認してください。これで、SSHを用いてGitHubにソースコードをアップロードすることができました。

Gitについての詳細は以下のサイトを参考にしてください。

▼サル先生のGit入門

<https://backlog.com/ja/git-tutorial/>

おわりに

この本でコマンドラインでの操作に慣れ、楽しいと思ってもらえたら幸いです。また、紹介したコマンドを駆使してShellスクリプトを書いて自動化してみると楽しいかもしれません。

この本では扱っていませんが、データベースにもMySQLの他にPostgreSQLなどのデータベースもあるので、他のデータベースに変えて構築してみたり、WEBサーバーをApacheからNginxなどに変えてみると楽しいかもしれません。また、ApacheとMySQLを別なインスタンスにするのも良いです。