

[年]

サーバー構築 手順書

初心者にもわかりやすく簡単

3-I-28 藤枝 昂汰

内容

準備物.....	2
環境.....	2
Ubuntu の USB インストーラを作成	3
Ubuntu をインストールする方法	3
Ubuntu をインストールする PC の設定方法	4
重要！テキストエディタの利用方法	6
VI テキストエディタの操作コマンド	7
サーバー構築の手順.....	8
今回紹介するオープンソースソフトウェアの紹介.....	8
Apache HTTP Server.....	8
Docker.....	8
インストールから起動までの手順(Apache HTTP Server 編).....	9
Apache のプロセス管理コマンド	11
仮想ホストの作成.....	11
仮想ホストの作成コマンド	11
Docker で WEB ページを作る方法.....	12
Docker のインストール	12
インストールが完了して、起動しているかの確認.....	13
Docker コマンドをユーザーで実行したい場合.....	14
Web サーバの起動.....	15
SSH で手持ちの BWOD と接続する方法.....	15
OpenSSH をインストールする	15
任意(推奨) セキュリティ観点における SSH の設定の変更.....	16
設定ファイルのバックアップ.....	16
ログイン制限のコマンド	17
SSH へのログイン	17
参考文献.....	17
Ubuntu インストール方法参考資料	17
vi テキストエディタコマンド参考資料(再記).....	17
Web サーバ構築参考資料(Docker).....	18
SSH 参考資料.....	18

準備物

- ・ ノートパソコン
- ・ USB
- ・ デスクトップパソコン

環境

- ・ デスクトップパソコン
- ・ Ubuntu

OS 名 Ubuntu 22.04 LTS

OS 種類 64 ビット

Ubuntu の USB インストーラを作成

Ubuntu をインストールする方法

1. ノートパソコンで Ubuntu の公式ホームページにアクセスする

• <https://www.ubuntulinux.jp/download/ja-remix>

2. 日本語版インストールをクリックしてページからインストールする。

⚠ 容量が大きいのインストールする際は注意する

3. USB インストーラを作成するための準備

• Rufus をインストールする

• <https://rufus.ie/ja/>

• Rufus を起動する(図 1.1)

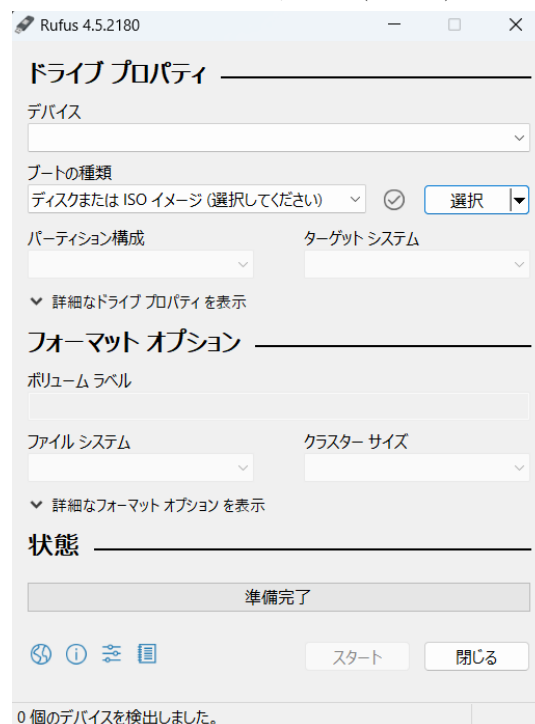


図 1.1 Rufus 起動画面

- ・ Rufus のブートの種類に先ほどインストールしたファイルをドラッグアンドドロップ
→スタートボタンを押下

ダウンロード途中図を図 1.2 に示す。

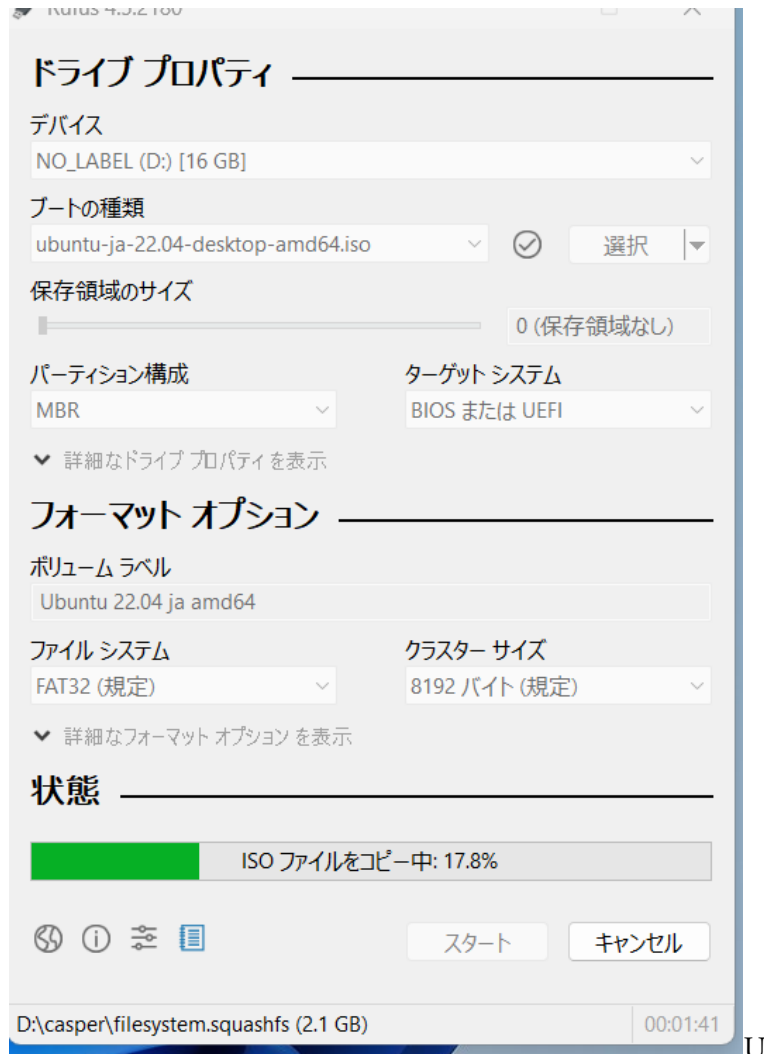


図 1.2 ダウンロード途中の様子

Ubuntu をインストールする PC の設定方法

まっさらなデスクトップパソコンに USB を挿す。

電源をつけた直後に F12 キーを押下することで Boot メニューを直接立ち上げることができる。

Boot メニューの一例の画面を図 1.3 に示す。

(ただし機種によって Boot メニューの画面は違うので注意すること)

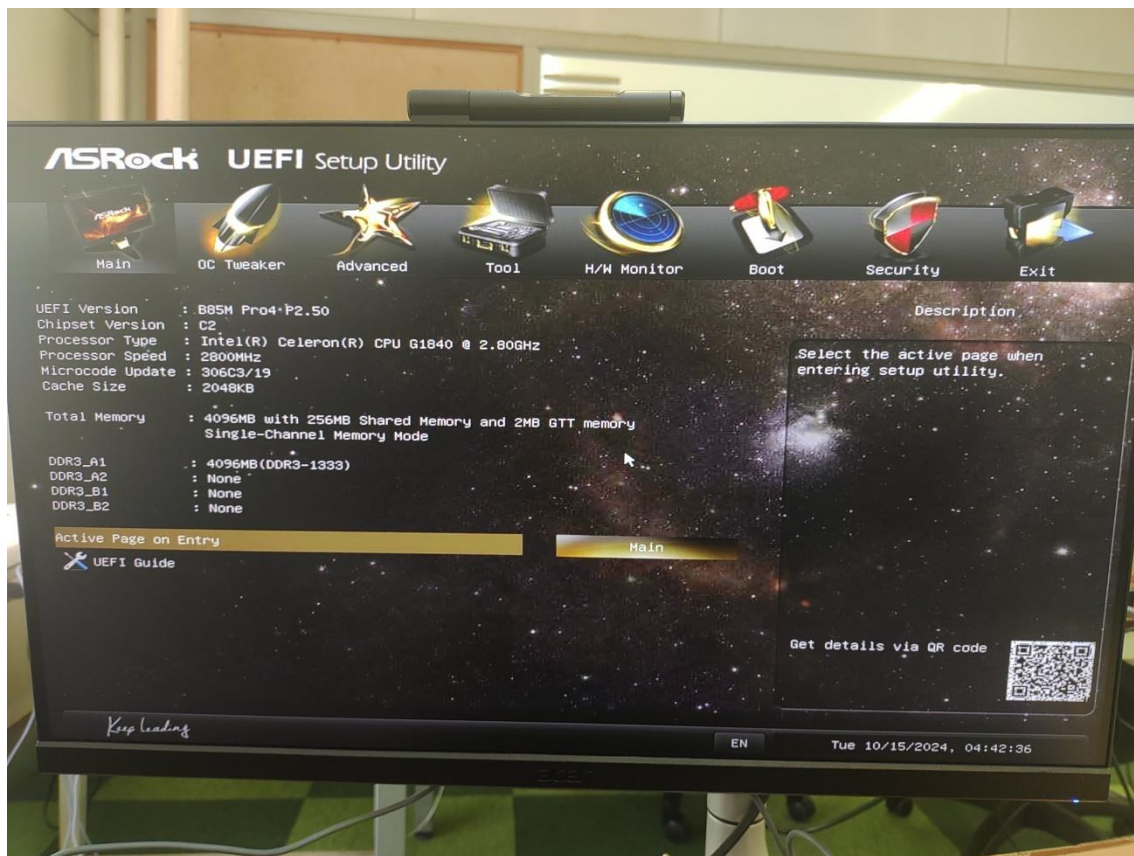


図 1.3 Setup メニューの画面の一例

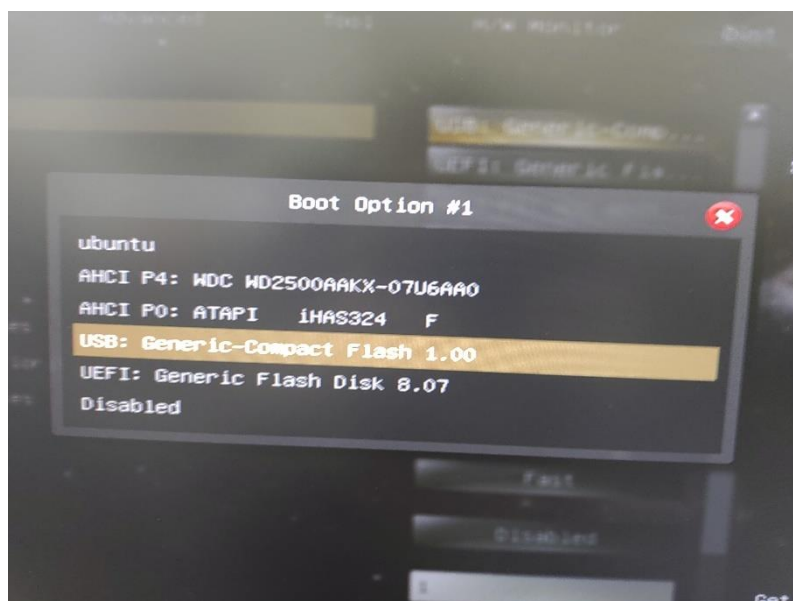


図 1.4 BootOption の一例の画面

BootOption で USB を優先に起動するようにしてから電源を入れなおすと以下のような画面が表示される。2 つ目の Try Ubuntu with Japanese… は日本語対応しているのでおすすめである。

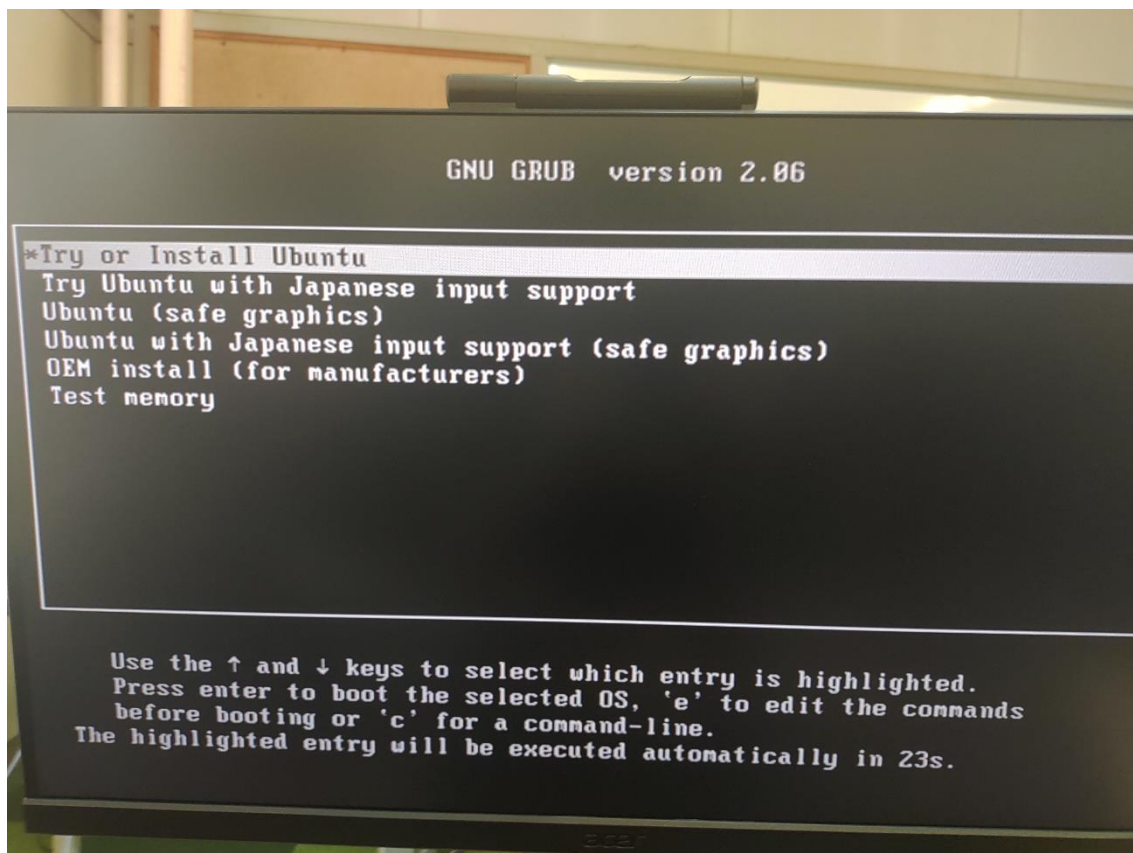


図 1.4 GNU GRUB の画面の一例

押下すると無事に Ubuntu の画面に入れる。

重要！テキストエディタの利用方法

テキストエディタと一口にいっても様々な種類がある。

今回は vi エディタを紹介しますが自分であったのを見つけてみる等もおすすめである。

以下のリンクで様々なフリーテキストエディタを紹介する。(参考文献にものせている)

https://eng-entrance.com/free_texteditor_linux …(1)

VI テキストエディタの操作コマンド

まずは vi テキストエディタを開くコマンドを紹介する。

『\$ vi {ファイル名}』

Vi テキストエディタを開いた際はコマンドモードと呼ばれるモードにいます
コマンドモードは「保存」「終了」「カーソル移動」「削除」と言った操作を行うための
モードであり、これは vi エディタがマウスを使わずにキーボードのみで操作するためです。

実際に編集するためには i キーもしくは a キーを押して入力モードに入る
その他コマンドモードで主に使用するコマンドを記載する。(表 2.1)

コマンド名	説明
:w	上書き保存
:wq	保存して終了
:q!	保存しないで終了
:q	終了

表 2.1：コマンドモードで主に使用するコマンド

VI テキストエディタは独自のカーソル移動を採用している。
以下にカーソル移動のコマンドを示すのでカーソル移動の際は活用すること(表 2.2)

コマンド名	説明
h	←
j	↓
k	↑
L	→

表 2.2：カーソルの移動コマンド

削除コマンドも必須なので記載する。切り取りと記載しているが貼り付けを無視すれば
削除と同義であるのでこちらを紹介する。(表 2.3)

コマンド名	説明
x	カーソルの文字の切り取り
dd	カーソルの行の切り取り

表 2.3：削除コマンド

その他便利なコマンドも紹介する。(表 2.4)

コマンド名	説明
u	直前の動作の取り消し
Ctrl+R	上記の作業の取り消し
.	直前の操作の繰り返し

その他コマンドは様々ありここに記載したのはほんの一例です。

以下に vi の基本コマンドの一覧の参考ページのリンクも記載するので確認してください。

<https://docs.oracle.com/cd/E19253-01/816-3946/editorvi-tbl-83/index.html> …(2)

サーバー構築の手順

ついにメイン！サーバー構築の手順について説明していく。

オープンソースソフトウェアも多くの種類がありますが今回は Apache HTTP Server と Docker を紹介する。

他にも様々なオープンソースがあるので調べてみることをおすすめする。

今回紹介するオープンソースソフトウェアの紹介

Apache HTTP Server

今回紹介する Apache HTTP Server（アパッチ）は Apache Software Foundation が開発しているウェブページをインターネット上で公開するための無料でオープンソースのウェブサーバーソフトウェアです。リクエストを受け取り、HTML ファイルや画像などのコンテンツをユーザーに配信する役割を果たします。

Docker

Docker は、アプリケーションを動作環境ごと「コンテナ」という単位でパッケージ化して管理・実行するプラットフォームだ。コンテナは軽量で、OS の仮想化を行わずにアプリケーションを独立して動作させることができる。ただし、ホスト OS が Linux である

が要求されるため、Windows では Linux の仮想環境を用意する必要がある。
その場合は Windows Subsystem for Linux(WSL)などを利用する。

インストールから起動までの手順(Apache HTTP Server 編)

ソフトウェアをインストールするコマンドを以下に記載する。

```
『$ sudo apt update # インストール前にパッケージの更新』
```

```
『$ sudo apt -y install apache2 # Apache HTTP Server のインストール』
```

次にソフトウェアを起動する。起動後に確認することがあるが後に記載する。

```
『$ sudo systemctl start apache2 # Apache の起動』
```

```
$ sudo systemctl enable apache2 # Apache の自動起動の有効化
```

```
$ sudo systemctl status apache2 # Apache のステータス』
```

以上のコマンドを実行し、成功すると以下のような画面が出てくる。特に赤丸で囲った
Active:欄が active (running)になっていることを確認しておくこと！

(画像は一例なので必ずしも同じとは限らない)(図 2.1)

```
$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Wed 2024-11-27 12:34:56 UTC; 1h 15min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 12345 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 67890 (apache2)
    Tasks: 6 (limit: 12345)
   Memory: 12.3M
    CGroup: /system.slice/apache2.service
            └─67890 /usr/sbin/apache2 -k start
              └─67891 /usr/sbin/apache2 -k start
                └─67892 /usr/sbin/apache2 -k start

Nov 27 12:34:56 hostname systemd[1]: Starting The Apache HTTP Server...
Nov 27 12:34:56 hostname systemd[1]: Started The Apache HTTP Server.
```

最後に起動できているかどうかを WEB ページで確認したい。以下の URL にアクセスし、
デフォルトページが表示されるかを確認すること！

URL

「http://<サーバの IP アドレス>」にアクセス

「http://localhost」サーバー上でWEBサーバーにアクセスする場合に有効

デフォルトページの画像を以下に記載する(図 2.2)

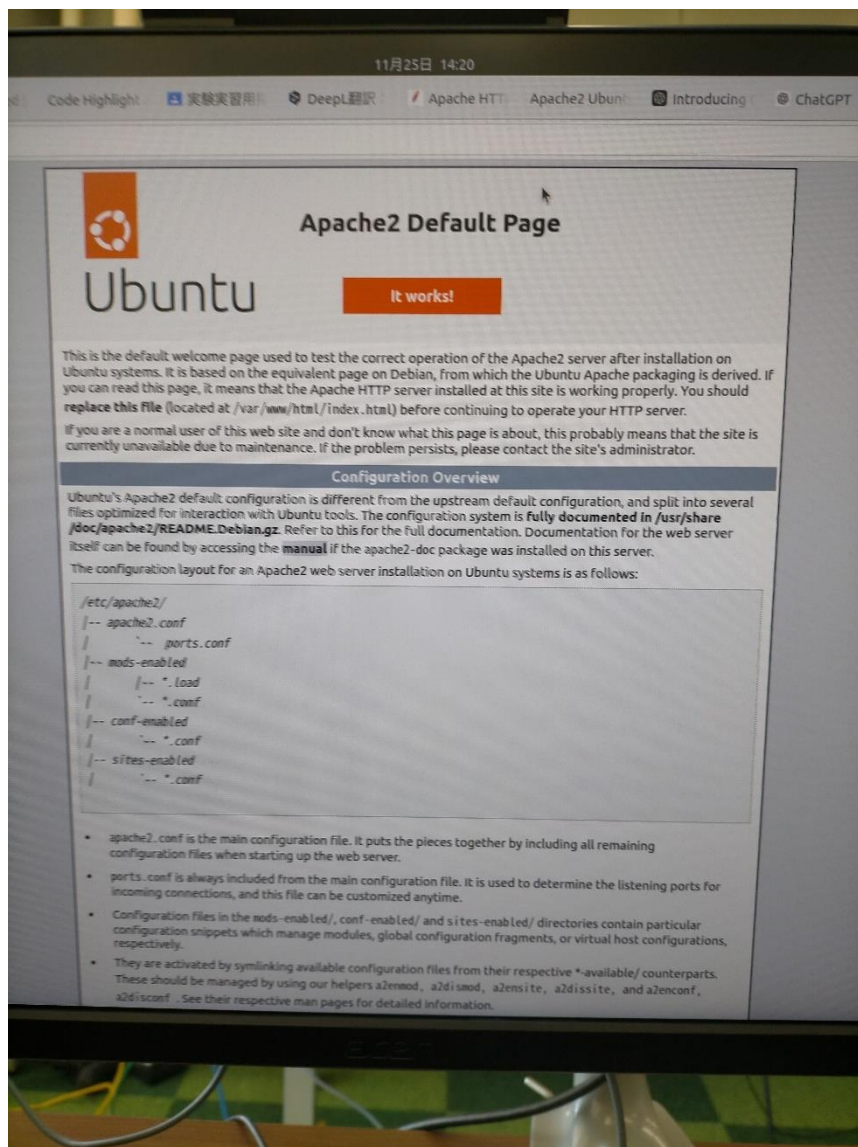


図 2.2 デフォルト画面

Apache のプロセス管理コマンド

ここで先に Apache のプロセス管理のコマンドを紹介しておく。

以下このコマンドを使用する際は操作しか記載しないので注意すること。(表 2.5)

コマンド名	説明
\$ sudo systemctl stop apache2	Web サーバー停止
\$ sudo systemctl start apache2	Web サーバー起動
\$ sudo systemctl restart apache2	Web サーバー再起動
\$ sudo systemctl reload apache2	設定のリロード
\$ sudo systemctl enable apache2	自動起動の有効化
\$ sudo systemctl disable apache2	自動起動の無効化

表 2.5 : Apache のプロセス管理のコマンド

仮想ホストの作成

ここからは実際に仮想ホストを作成し、その中に HTML を記述し WEB ページにあげていきたいと思う。

仮想ホストの作成コマンド

まずは仮想ホストの設定ファイルを作成する

以下のコマンドで設定ファイルを作成する(ファイル名は任意とする)

```
『$ sudo touch /etc/apache2/sites-available/ファイル名.conf』
```

次に、ディレクトリを作成する

```
『$ sudo mkdir -p ディレクトリパス』
```

作成したディレクトリパスのファイルを新規作成します

```
『$ sudo touch ディレクトリパス/ファイル名』
```

(touch コマンドはファイルを作成または更新します)

先ほど作成したファイルを編集します

```
『$ sudo vi ディレクトリパス/ファイル名』
```

VI テキストエディタの操作は上記のトピックに記載しているので省略する。
ここではファイルに記述する例を示す。HTML 形式なら任意のコードが記述可能です。

```
『<html>
<body>
<h1>Hello word </h1>
</body>
</html>』
```

次に仮想ホストを有効化します。

```
『$ sudo a2ensite ファイル名.conf # 仮想ホストの有効化
$ sudo systemctl reload apache2 # 有効化した仮想ホストをサーバに反映』
```

反映されているかどうか確かめるためデフォルトの仮想ポストを無効化します。

```
『$ sudo a2dissite 000-default.conf # 仮想ホストの無効化
$ sudo systemctl reload apache2 # 有効化した仮想ホストをサーバに反映』
```

最後にファイアウォールの有効化コマンドを示す。

```
『$ sudo ufw enable # ファイアウォールの有効化
$ sudo ufw allow 'Apache' # 80 番ポートの開放
$ sudo ufw status』
```

Docker で WEB ページを作る方法

ここからは Docker での WEB ページ構築の手順を示す。
Docker の説明は上記に記載している

Docker のインストール

Docker のインストールコマンドを以下に示す

```
『$ sudo apt-get update』
```

システム管理データベースの更新

```
『$ sudo apt-get install ca-certificates curl』
```

必要なツールや証明書のインストール

```
『$ sudo install -m 0755 -d /etc/apt/keyrings』
```

/etc/apt/keyrings にディレクトリを作成

```
『$ echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu
```

```
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | sudo tee  
/etc/apt/sources.list.d/docker.list > /dev/null』
```

⚠ このコマンドは詳細説明が長いので細かな意味はぜひ生成系 AI 等を活用して調べてみてください

全体的なコードの流れ

現在のシステムアーキテクチャ（例: amd64）と Ubuntu のバージョンコード名（例: jammy）に応じて適切な Docker リポジトリを指定。

リポジトリ設定をファイル /etc/apt/sources.list.d/docker.list に保存。

最終的な効果

Docker の公式パッケージを APT（Advanced Package Tool）を使ってインストールおよび更新できるようになる。次に `sudo apt-get update` を実行することで、このリポジトリから Docker 関連のパッケージ情報が取得される。

インストールが完了して、起動しているかの確認

Version と状態を確認するコマンドを以下に示す

```
『$ sudo docker --version  
$ sudo systemctl status docker』
```

特に、systemctl を起動した際の結果の active の部分は Apache と同様に注意して確認してほしい。参考までに実行結果の一例を図 3.1 に示す。

```
$ sudo systemctl status docker  
● docker.service - Docker Application Container Engine  
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)  
   Active: active (running) since Mon 2024-11-27 10:30:12 UTC; 5min ago  
 TriggeredBy: ● docker.socket  
     Docs: https://docs.docker.com  
    Main PID: 1234 (dockerd)  
      Tasks: 8  
     Memory: 35.5M  
        CPU: 1.234s  
    CGroup: /system.slice/docker.service  
            └─1234 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock  
  
Nov 27 10:30:12 myhostname dockerd[1234]: time="2024-11-27T10:30:12.123456789Z" level=info  
Nov 27 10:30:12 myhostname dockerd[1234]: time="2024-11-27T10:30:12.234567890Z" level=info
```

Docker コマンドをユーザーで実行したい場合

以下のコマンドを実行することで可能になる。

```
『$ sudo usermod -aG docker $USER』
```

Docker の group に USER を追加する

⚠ ここでの \$ USER は環境変数と呼ばれ、プログラムの挙動を調整するためのパラメータの一種です。ここではコマンドを実行している現在のログインユーザー名を表すため、わざわざユーザー名を打つ必要はありません。環境変数について知りたい方は解説リンクを以下に記載するので参考にしてみてください。

https://qiita.com/angel_p_57/items/480e3fd4552e52199835

Web サーバの起動

最後に Docker を使って Web サーバを起動させる。まずは Web サーバ管理用のディレクトリを作成する。

```
『$ mkdir -p ~/ディレクトリ名/htdocs ~/ディレクトリ名/conf』
```

ファイルの作成。ディレクトリ名は任意とする。

次に Web サーバのコンテンツ管理などのディレクトリを作成し、その直下に HTML ファイルを管理するディレクトリを作成する。

```
『$ cd ~/ディレクトリ名
```

```
$ docker run -dit --name my-apache-app -p 8080:80 -
```

```
v ./htdocs:/usr/local/apache2/htdocs/ -v ./conf:/usr/apache2/conf httpd:2.4』
```

cd ファイルに移動

docker run 新しい Docker コンテナを作成し実行する

その他詳しいコマンドの解説は

SSH で手持ちの BWOD と接続する方法

OpenSSH をインストールする

以下のコマンドを打ち込む

- `sudo apt install openssh-server`

(ただの SSH のインストールコマンドです。)

その後、OpenSSH の起動をコマンドから実行する

```
$ sudo systemctl start ssh
```


(SSH起動)

```
$ sudo systemctl enable ssh
```

(次回から自動で起動するようにする)

```
$ sudo systemctl status ssh
```

(SSH の現在の状態を確認)

Active:の欄が active (running)になっていれば成功！

任意(推奨) セキュリティ観点における SSH の設定の変更

設定ファイルのバックアップ

設定の変更時に設定ファイルを間違えて削除しないようにバックアップを作成する

```
『$ sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config_backup』
```

設定ファイルの変更

Vi コマンドで/etc/ssh/sshd_config を開き、編集モードで Port の番号を任意のポート番号に変更する。ただし、1023 番以下は既に使われていることが多く、通信がバッティングしてしまうとよくないのでできるだけ避けましょう。

詳しくは以下の参考リンクを参照してください。

<https://www.braveryk7.com/change-22-port/>

SSH で利用するポート番号を変更したら、次にファイアウォールに SSH のポート利用許可のルールを追加する。

```
『$ sudo ufw allow 任意のポート番号/tcp』
```

ログイン制限のコマンド

ログインを禁止する際は設定ファイルに以下を追加する

『PermitRootLogin no』

特定ユーザーのみログイン可能にするには以下のコマンドを使用する。

『AllowUsers ユーザ名』

SSH へのログイン

他のパソコンからログインするときは以下のコマンドを打つ

『\$ ssh ユーザ名@ホスト名(もしくは IP アドレス) -p ポート番号』

参考文献

Ubuntu インストール方法参考資料

<https://rtc-fukushima.jp/technical/6671/>

vi テキストエディタコマンド参考資料(再記)

https://eng-entrance.com/free_texteditor_linux …(1)

<https://docs.oracle.com/cd/E19253-01/816-3946/editorvi-tbl-83/index.html> …(2)

Web サーバ構築参考資料(Docker)

https://qiita.com/angel_p_57/items/480e3fd4552e52199835

SSH 参考資料

<https://www.braveryk7.com/change-22-port/>