

Linux サーバーの導入

山内優弥

2024 年 12 月 3 日

目次

第 I 章	まえがき	5
I.1	この文書について	5
I.2	実行環境	5
第 II 章	Ubuntu の導入	7
II.1	Ubuntu のダウンロード	7
II.2	rufus	8
II.3	Ubuntu との邂逅	9
II.4	エディタ	9
第 III 章	Web サーバー	11
III.1	Apache	11
III.2	Apache の設定	11
III.2.1	仮想ホストの作成	13
III.2.2	ファイアウォールの設定	14
第 IV 章	Docker	17
IV.1	Docker を使う	18
IV.2	Docker を用いた Web サーバー	18
付録 A	コマンド一覧	21

第1章

まえがき

1.1 この文書について

Linux のディストリビューションの一つである Ubuntu の導入からサーバーとしての利用の手順を述べて文書である。また

ああああ

あああああ

や

いいいい

いいいいい

で書かれる文章は必ず読む必要はなく適当な補足であることにする。

1.2 実行環境

- 空っぽのパソコン
 - WSL2:
 - WSL バージョン:2.3.26.0
 - カーネルバージョン:5.15.167.4-1
 - WSLg バージョン:1.0.65
 - MSRDC バージョン:1.2.5620
 - Direct3D バージョン:1.611.1-81528511
 - DXCore バージョン:10.0.26100.1-240331-1435.ge-release
 - Windows バージョン:10.0.26100.2454
- WSL2 内の Ubuntu のバージョン:Ubuntu 22.04.5 LTS

第 II 章

Ubuntu の導入

II.1 Ubuntu のダウンロード

今回は Ubuntu Japanese Team から日本語環境の Ubuntu のダウンロードをこのページ <https://www.ubuntulinux.jp/japanese> からダウンロードします。どのミラーサイトを用いても構いません。

Ubuntu の日本語環境のバージョンについて

今現在 (2024/10/21) 最新の LTS(Long-Term Support) バージョンは Ubuntu 24.04.1 LTS ですが、2024/06/10 の記事 (<https://www.ubuntulinux.jp/News/ubuntu2404-ja-remix>) において Ubuntu Japanese Team が Ubuntu 24.04 LTS の日本語 Remix をリリースしないことを表明しています。

Ubuntu の歴史

Ubuntu は、2004 年にリリースされた Linux ディストリビューションで、Debian をベースにしています。開発元は Canonical という会社で、創設者の Mark Shuttleworth によって立ち上げられました。Ubuntu の名称は南アフリカのズールー語で「他者への思いやり」や「人間性」を意味し、オープンソースコミュニティやユーザー間での協力を象徴しています。

- 2004 年: 最初のバージョン「Ubuntu 4.10 "Warty Warthog"」がリリースされました。Debian ベースで使いやすさを重視し、デスクトップ Linux の普及を目指しました。
- 2006 年: LTS (Long Term Support) リリースが導入され、「Ubuntu 6.06 LTS "Dapper Drake"」が最初の LTS 版です。LTS 版は 5 年間の長期サポートが提供され、企業や組織での採用が進みました。
- 2010 年: Ubuntu はデフォルトのデスクトップ環境を GNOME から Unity に変更しました。Unity はユーザーインターフェースを刷新し、使いやすさを向上させることを目的としていましたが、一部のユーザーから批判もありました。

- 2017 年: Unity の開発が中止され, GNOME デスクトップに戻ることが発表されました。「Ubuntu 17.10 "Artful Aardvark"」で Unity から GNOME への移行が行われ, 従来のデスクトップ環境を採用する形に戻りました。
- 2020 年代以降: クラウドや IoT (モノのインターネット), サーバー市場でも Ubuntu は広く利用されています。また, Ubuntu ベースの派生ディストリビューション (Kubuntu, Xubuntu, Lubuntu など) がそれぞれのニーズに応じて発展してきました。

iso ファイルについて

気が向いたら書く。

II.2 rufus

rufus とはブータブル USB 作成するソフトウェアで次の URL からダウンロードできる。
https://rufus.ie/ja/#google_vignette
USB を挿して rufus を開くと次のようなウィンドウが出現する。

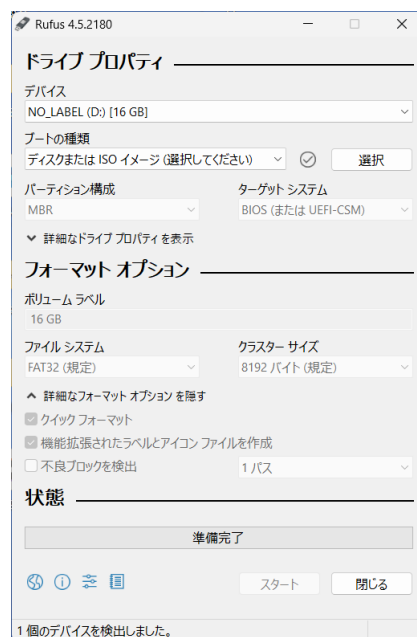


図 II.1 rufus のウィンドウ

「選択」から先程ダウンロードした Ubuntu の iso ファイルを選択して, スタートを選択する。
これで Ubuntu を起動する準備が整った。

II.3 Ubuntu との邂逅

iso ファイルを入れた USB を挿し込んで BIOS を起動する．そして，Boot の順位を変更して Ubuntu を一番上に変更する．これにより Ubuntu が起動する．

まず初めに現時点では，Ubuntu に IME が導入されていないため日本語が入力できない．これを解消するために Mozc をダウンロードしよう．

ターミナルを開いて以下を実行するとダウンロードが完了する．

```
$ sudo apt update
$ sudo apt -y install ibus-mozc
```

忘れないうちに，ここで超重要なことを教えよう．実行中のコマンドを強制終了するのは Ctrl+c で行うことができる．

Mozc について

-y について

II.4 エディタ

UNIX 系の OS では，vi エディタが標準で搭載されている．vi エディタの基本的な使い方を紹介しよう．

```
$ vi ファイル名
```

とすれば，エディタが起動する．

vi エディタは二つのモードが存在する．

コマンドモード ⇄ インサートモード

はじめに起動時にはコマンドモードになっている．この状態では，テキストファイル内の文字列の編集はできず，入力された文字はコマンドとして認識される．

とりあえず，覚えておくべきコマンドを紹介しよう．

1. カーソル移動

- h : カーソルを一つ左に

- j : カーソルを一つ下に
- k : カーソルを一つ上に
- l : カーソルを一つ右に
- Ctrl+f : 次のページへ
- Ctrl+b : 前のページへ

2. 切り取り/コピー/貼り付け

- x : カーソル位置の文字を切り取る (ほとんど Delete の意味)
- XX : カーソル位置の手前の文字を切り取る (ほとんど Backspace の意味)
- dd : カーソルのある行の切り取り
- yy : カーソルのある行のコピー
- p : カーソルのある行の下に貼り付け
- P : カーソルのある行に貼り付け

3. ファイルの保存や終了など

- :w : ファイルの保存
- :w ファイル名 : 名前をつけてファイルを保存
- :q : エディタの終了
- :wq : :w をして :q するコマンド (i.e. ファイルを保存して終了)
- ! : 強制するコマンド. :q! ← 強制終了, :wq! ← 強制的に保存して終了

もちろん**インサートモード**に移行するコマンドが存在する。インサートモードとはテキストを編集するモードのことである。

- i : カーソルの位置からインサートモードに移行する。
- a : カーソルの後ろからインサートモードに移行する。
- o : カーソルのある行の下に空白行を追加してインサートモードに移行する。

vi と vim

第 III 章

Web サーバー

III.1 Apache

本書では、Web サーバーとして Apache を紹介する。
まず以下のコマンドで Apache をインストールしよう。

```
$ sudo apt update $ sudo apt -y install apache2
```

インストールが完了したら早速 Apache を起動しよう！

```
$ sudo systemctl start apache2
```

これが起動コマンドである。

```
$ sudo systemctl enable apache2
```

これは Apache の自動起動の有効化を行うコマンドである。

```
$ sudo systemctl status apache2
```

これは Apache の現在の状態 (ステータス) を表示するコマンドである。ステータスの Active の部分が **active(running)** になっていれば Apache が動いている。

最後に、ブラウザから「http://localhost」と入力すると Apache のウェルカムページ (Apache2 Default Page と書かれたページ) が表示されたら成功である。

III.2 Apache の設定

Apache の設定ファイルはサーバー全体に適用されるグローバル設定ファイル「/etc/apache2/apache2.conf」やデフォルトでアクセスされる仮想ホストに適用される設定ファイル「/etc/apache2/sites-available/000-default.conf」で管理される。

仮想ホストに適応される設定ファイルについて変更を加えていこう。

まず、000-default.conf の中身を以下に示す。(ただしコメントアウトは削除している。)

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

上から見ていこう。

```
ServerAdmin webmaster@localhost
```

とは Apache がエラー表示を行う場合などに問い合わせ先となる連絡先メールアドレスまたは参照先 URL を設定するのに使用。

```
DocumentRoot /var/www/html
```

DocumentRoot は Web ページを配置するディレクトリである。この場合 Apache のウェルカムページは/var/www/html ディレクトリにあることがわかる。

```
ErrorLog ${APACHE_LOG_DIR}/error.log
```

はサーバーのエラーログをどこに保存するかを設定します。\${APACHE_LOG_DIR} は環境変数であって、Ubuntu においては/etc/apache2/envvars ファイル内で定義されています。私の実行環境では/var/log/apache2でした。

```
CustomLog ${APACHE_LOG_DIR}/access.log combined
```

はサーバーのアクセスログをどこに保存するかを設定します。combined は、アクセスログのフォーマットを指定するもの。

また、

```
DirectoryIndex ファイル名
```

で表示する html ファイルを指定することができる。

III.2.1 仮想ホストの作成

「/etc/apache2/sites-available/000-default.conf」が初期の仮想ホストとして設定されている。これを残したまま別の仮想ホストを作成する場合は、「/etc/apache2/sites-available/」直下に新たな設定ファイルを作成する。

簡単のために、もとあった 000-default.conf をコピーするとよい。

```
$ sudo cp コピーするファイル コピーした後のファイル名
```

なので、今回は test.conf という設定ファイルを作成しよう。

```
$ sudo cp /etc/apache2/sites-available/000-default.conf
/etc/apache2/sites-available/test.conf
```

次に、DocumentRoot と DirectoryIndex を変更しよう。今回は以下のように設定した。

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /hoge
    DirectoryIndex hoge.html      ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

ディレクトリやファイルを作成しよう。

```
$ sudo mkdir /hoge
```

```
$ sudo touch /hoge/hoge.html
$ sudo vi /hoge/hoge.html
```

で適当な html ファイルを作成する。

次に作成した仮想ホストを有効化したい有効化する前に今有効になっているデフォルトの仮想ホストを切断しよう。

```
$ sudo a2disssite 000-default.conf
```

次にこの設定をサーバーに反映させるためにリロードしよう。

```
$ sudo systemctl reload apache2
```

さて、先ほど作成した仮想ホストを有効化しよう。

```
$ sudo a2ensite test.conf
```

次に有効化した仮想ホストをサーバーに反映させよう。

```
$ sudo systemctl reload apache2
```

いま、「http://localhost」にアクセスしたら「403 Forbidden」が表示されるはずだ。これを解消するためにグローバル設定ファイル (/etc/apache2/apache2.conf) を編集しよう。以下のように書かれている部分をみよう。

```
<Directory /www/html>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```

これはデフォルトの DocumentRoot に対する設定であって、

```
<Directory /www/html>
```

を

```
<Directory /hoge>
```

に変更して

```
$ sudo systemctl reload apache2
```

設定を反映させよう。

403 Forbidden

今回なぜ 403 Forbidden が出たのかと言うと、Apache が /hoge にアクセスする権限がないためである。そのため /hoge にアクセスしてもいいですよ。ということを先ほどの設定でする必要がある。

III.2.2 ファイアーウォールの設定

Ubuntu では、ファイアーウォールは `ufw` コマンドを用いる。

```
$ sudo ufw enable
```

これでファイアーウォールを有効化

```
$ sudo ufw allow Apache
```

80 番ポートの開放

```
$ sudo ufw status
```

これで ufw の状態を確認することができる。
うまく行っていれば To の下に Apache,Apache(v6), Action の下に ALLOW, From の下に Anywhere,Anywhere(v6) と出るはずである。

第Ⅳ章

Docker

さて Docker のインストールから早速行う .

```
$ sudo apt update
```

お決まりのアップデート

```
$ sudo apt install ca-certificates curl
```

ca-certificates と curl のインストールである .

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

```
$ sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o  
/etc/apt/keyrings/docker.asc
```

```
$ sudo chmod a+r /etc/apt/keyrings/docker.asc
```

```
$ echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/d  
https://download.docker.com/linux/ubuntu $(./etc/os-release && echo  
"$VERSION_CODENAME") stable" | sudo tee /etc/apt/sources.list.d/docker.list  
> /dev/null
```

```
$ sudo apt update
```

```
$ sudo apt install docker-ce docker-ce-cli containerd.io  
docker-buildx-plugin docker-compose-plugin
```

Docker のインストールが完了したことを確認するためにバージョンと状態確認を行う。

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

IV.1 Docker を使う

以下のコマンドでコンテナを起動する。

```
$ docker container run IMAGE
```

さて Hello World に相当するものを実行してみよう。

```
$ docker container run hello-world
```

実行すれば、チュートリアルやドキュメントの URL が記載されてるものが出てくるだろう。

コンテナ一覧を確認するには

```
$ docker container ls
```

を実行する。

IV.2 Docker を用いた Web サーバー

httpd というイメージを用いてコンテナを起動することで Web サーバーを起動することができる。まずサーバー管理用のディレクトリを作成しよう。

```
$ mkdir -p ~/dockertest/htdocs ~/dockertest/conf
```

次に httpd コンテナを起動しよう。

```
$ cd ~/dockertest
$ docker run -dit --name my-apache-app -p 8080:80 -v
./htdocs:/usr/local/apache2/htdocs/ -v ./conf:/usr/apache2/conf
httpd:2.4
```

「-v ./htdocs:/usr/apache2/htdocs/」や「-v ./conf:/usr/apache2/conf」によって「~/dockertest/htdocs」や「~/dockertest/conf」に配置されたファイルがコンテナに反映されるようになる。例え

ば、「~/dockertest/htdocs」の直下に index.html を配置すると、その内容がアクセス時に表示される。

付録 A

コマンド一覧

気が向けばコマンドについての一覧を作る．