

手順書

Web サーバを構築する方法

目次

1.準備.....	2
2.コマンド実行の練習(コマンドリスト)	3
3.サーバの実行	4
4.Web サーバの構築	9
5.SSH	17
6.Docker	23
7.参考サイト一覧.....	24

1. 準備

1.1 開発環境

今回開発及び検証に使う PC の情報を以下に示す

リスト 1.1 サーバ開発用 PC

OS 名	Ubuntu 22.04 LTS
OS の種類	64 ビット
GNOME のバージョン	42.0
ウィンドウシステム	Wayland
CPU	Intel Celeron G1840
メモリ	4GB
ディスク容量	250GB

リスト 1.2 SSH の際に使用した PC

OS 名	Windows 11 24H2
種類	64 ビット
CPU	Intel Core i7-7500U @ 1.80 GHz
RAM	16.0GB

1.2 Ubuntu のインストール

- I. Ubuntu Japanese Team のサイト等から ubuntu-ja-22.04-desktop-amd64.iso をダウンロードする。
- II. Rufus をダウンロードして起動する。デバイスの項目は PC に差し込んだ USB を選択する。図 1-1 の選択から ubuntu-ja-22.04-desktop-amd64.iso を選びスタートを押す。

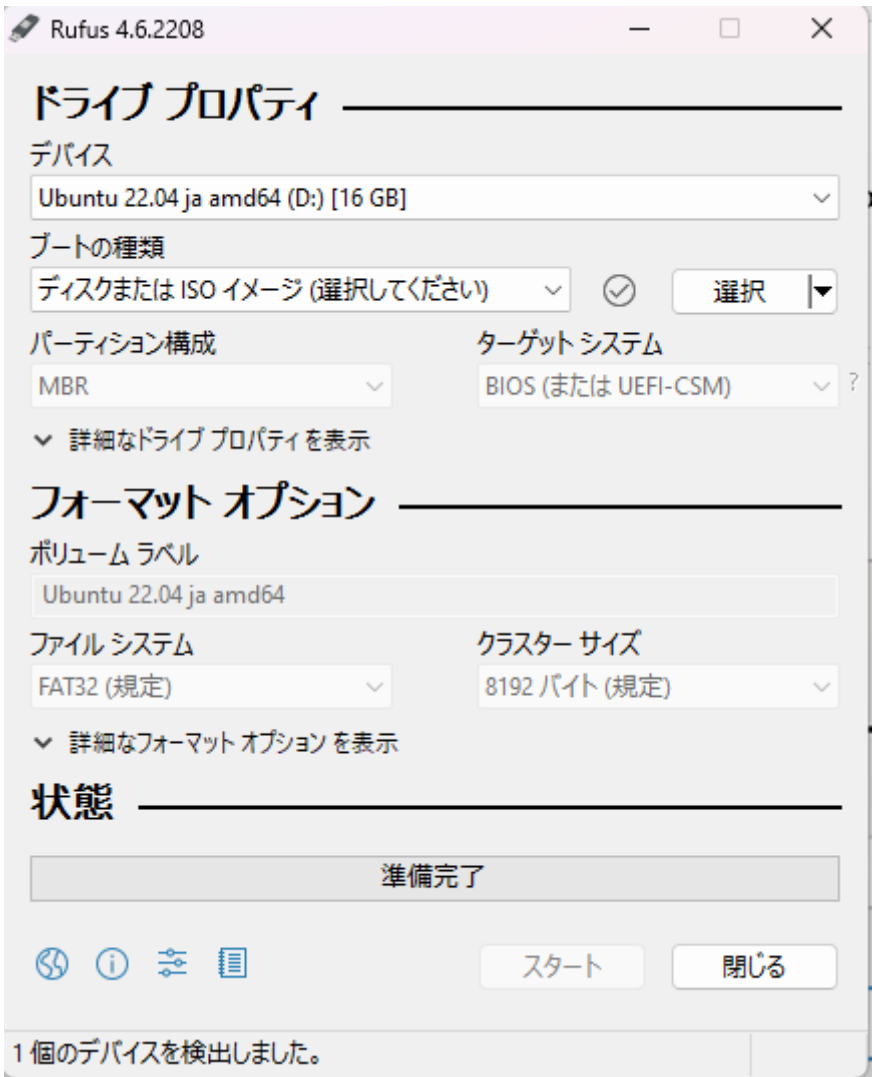


図 1-1 Rufus 設定

- III. 作成した USB を PC に差し込み起動する。そのあと BIOS を起動するために F2 キーを連打する。
- IV. BIOS のブートタブの Boot Option #1 を先ほどの USB にする。
- V. Exit で Bios を閉じると起動する。

2.コマンド実行の練習(コマンドリスト)

今回の資料において使用されるコマンドの中で知っておくとよいものを以下に示す。

リスト 2.1 コマンドリスト

mkdir	新しいディレクトリの作成
-------	--------------

cp	コピー
cat	ファイル内容の表示
vi	テキストエディタの起動

次にテキストエディタの簡単な操作法を以下に示す。

リスト 2.2 テキストエディタの使い方

i	入力モードに変更
Esc	コマンドモードに変更
x	デリート
Ctrl+shift+v	貼り付け
:w	ファイルの保存
:q	エディタの終了(:wq で保存して終了)
:q!	変更を保存せずに終了

3.サーバの実行

3.1 ユーザの管理

Linux には root ユーザ、一般ユーザ、システムユーザという 3 つユーザの種類がある。root ユーザはシステムの管理者であり、すべてのファイルの読み書き、作成、消去、実行などあらゆることができる。しかしそのアカウントのパスワードが外部に漏れたり、root ユーザでのコマンド実行時にミスを行ったりすると大惨事につながりかねない。そのため基本的に一般ユーザを使用し必要に応じて root 権限を借るようにすることが一般的である。(システムユーザは直接ログインしないため割愛)。現在ログインしているユーザが何かというのはターミナルのユーザ名の右が「\$」なら一般ユーザ、「#」なら root ユーザである。図 3.1 は\$なので一般ユーザ。



図 3.1 判別する例

I. root ユーザのパスワード設定

Ubuntu のインストール直後は root ユーザにはパスワードが設定されていない。先述した通り root ユーザの管理は非常に重要であるためパスワードを設定するととも

に厳重に管理する。以下のコマンドを用いることで root ユーザのパスワードを設定できる。また再設定を行う際も同じコマンドを用いるが、その際は現在のユーザのパスワードが要求される。

リスト 3.1 ルートユーザのパスワード設定

```
$ sudo passwd root
```

II. ユーザの追加

組織でサーバを利用する場合を含め複数の一般ユーザを使用したい時があるだろう。ユーザを追加するには以下のコマンドを入力する。今回はユーザ名を test としているがそこは任意のものに読み替えてよい。わかりやすいように test を赤字で表記する。

リスト 3.2 ユーザの追加

```
$ sudo adduser test
```

実行すると図 3-2 のようになる。

```
desktop:~$ sudo adduser test
[sudo] のパスワード:
ユーザー `test' を追加しています...
新しいグループ `test' (1001) を追加しています...
新しいユーザー `test' (1001) をグループ `test' に追加しています...
ホームディレクトリ `/home/test' を作成しています...
`/etc/skel' からファイルをコピーしています...
新しいパスワード:
正しくないパスワード: このパスワードは辞書チェックに失敗しました - 辞書の単語に基づいています
新しいパスワードを再入力してください:
パスワードが一致しません。
新しいパスワード:
正しくないパスワード: このパスワードは辞書チェックに失敗しました - 辞書の単語に基づいています
新しいパスワードを再入力してください:
パスワードが一致しません。
新しいパスワード:
正しくないパスワード: このパスワードは辞書チェックに失敗しました - 辞書の単語に基づいています
新しいパスワードを再入力してください:
パスワードが一致しません。
パスワード: サービスの最大試行回数に達しました
passwd: パスワードは変更されませんでした
もう一度試みますか? [y/N] N
test のユーザ情報を変更中
新しい値を入力してください。標準設定値を使うならリターンを押してください
フルネーム []:
部屋番号 []:
職場電話番号 []:
自宅電話番号 []:
その他 []:
以上で正しいですか? [Y/n] Y
```

図 3.2 ユーザ追加の実行結果

III. ユーザの消去

作成されたユーザを消去したいときは次のコマンドを使用する。ユーザ名の処理はユーザの追加手順の時と同様に赤字で test としてある。

リスト 3.2 ユーザの消去

```
$ sudo userdel test
```

実行すると以下のようなになる。何も表示はされないが正しく消去されている。確認したい場合はリスト 3.3 のコマンドを使うなどすれば確認することができる。また cut -d: -f1 とつけることで user 名のみを表示できる。

```

desktop:~$ sudo userdel test
desktop:~$

```

図 3.3 ユーザの消去結果

リスト 3.3 ユーザー一覧を表示

```
$ cat /etc/passwd
```

IV. ユーザのグループ追加

Linux にはグループというものがある。グループは所属しているすべてのユーザに対してまとめてアクセス権を与えるために使われる。例えば wheel グループに属する root ユーザ以外のユーザは root パスワードを変更またはリセットすることができ。ユーザは必ず 1 つのグループに属し、複数のサブグループに所属することができる。

ユーザをグループに追加するには以下のコマンドを使用する。個別のユーザ名はユーザの追加手順の時と同様に示す。またグループ名は group として赤色にする。

リスト 3.4 ユーザをグループに追加

```
$ sudo usermod -aG group test
```

これによりグループの追加ができる。以下にグループに関する便利なコマンドとグループへ追加した実行結果を示す。なおコマンドを入力しても特別に何かが表示されるわけではない。

リスト 3.5 グループに関するコマンド

グループ一覧を表示	\$ cat /etc/group
どのグループに属するか表示	\$ id

```

:x:1000:  :x:1000:test

```

図 3.4 ←実行前 | 実行後→

V. ユーザのグループ消去

ユーザをグループから消去する方法も紹介する。グループから消去する際は追加する際と異なりユーザネーム、グループ名という順番であるので注意。

リスト 3.6 ユーザをグループから消去するコマンド

```
$ sudo gpasswd -d test group
```

3.2 サーバの負荷を確認

サーバの負荷が大きすぎると十分に稼働できない。例えばオンラインゲームなら、サーバの許容量以上にアクセスが来てしまうとその間にゲームが遊べないなどのトラブルが起こってしまう。そのために uptime コマンドを使用すると稼働時間と左から 1 分、5 分、15 分間の負荷を確認することができる。この値が CPU のコア数を下回れば問題はない。

A terminal window with a dark background. The prompt is 'desktop:~\$'. The command 'uptime' has been entered. The output is '13:07:57 up 2:22, 1 user, load average: 0.77, 0.55, 0.33'.

```
desktop:~$ uptime
13:07:57 up 2:22, 1 user, load average: 0.77, 0.55, 0.33
```

図 3.5 負荷の確認

uptime コマンドを用いてサーバの負荷を確認できるが、リアルタイムで確認したい場合もある。そのような時は top コマンドを使用する。

リスト 3.7 top コマンド

```
$ top
```

```

~desktop:~$ top

top - 14:21:48 up 3:36, 1 user, load average: 0.54, 0.49, 0.79
Tasks: 217 total, 2 running, 215 sleeping, 0 stopped, 0 zombie
%Cpu(s): 20.2 us, 2.7 sy, 0.0 ni, 76.1 id, 1.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3571.2 total, 348.0 free, 960.2 used, 2263.0 buff/cache
MiB Swap: 2048.0 total, 1632.4 free, 415.6 used. 1909.9 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
  574 avahi     20   0   11620    7936   3584 R   33.1   0.2   49:02.37 avahi-daemon
 1446 isao     20   0 4682352 225860 92576 S    9.3   6.2   4:10.01 gnome-shell
 1706 isao     20   0 353764    15692 10772 S    0.7   0.4   1:04.68 Xwayland
 3751 isao     20   0 644212    55348 42384 S    0.7   1.5   0:40.22 gnome-terminal-
15209 isao     20   0  14256     3968   3200 R    0.7   0.1   0:00.07 top
   578 root      20   0 336448    14828 13036 S    0.3   0.4   0:32.24 NetworkManager
 1561 isao     20   0 316156     8740  7040 S    0.3   0.2   0:09.12 ibus-daemon
11485 root      20   0      0      0      0 I    0.3   0.0   0:00.45 kworker/u4:5-i915
13733 root      20   0      0      0      0 I    0.3   0.0   0:00.20 kworker/1:1-events
15196 root       0 -20      0      0      0 D    0.3   0.0   0:00.05 kworker/u5:2+i915_flip
     1 root      20   0 168004    12632 8152 S    0.0   0.3   0:06.82 systemd
     2 root      20   0      0      0      0 S    0.0   0.0   0:00.01 kthreadd
     3 root      20   0      0      0      0 S    0.0   0.0   0:00.00 pool_workqueue_release
     4 root       0 -20      0      0      0 I    0.0   0.0   0:00.00 kworker/R-rcu_g
     5 root       0 -20      0      0      0 I    0.0   0.0   0:00.00 kworker/R-rcu_p
     6 root       0 -20      0      0      0 I    0.0   0.0   0:00.00 kworker/R-slub_
     7 root       0 -20      0      0      0 I    0.0   0.0   0:00.00 kworker/R-netns
    12 root       0 -20      0      0      0 I    0.0   0.0   0:00.00 kworker/R-mm_pe
    13 root      20   0      0      0      0 I    0.0   0.0   0:00.00 rcu_tasks_kthread
    14 root      20   0      0      0      0 I    0.0   0.0   0:00.00 rcu_tasks_rude_kthread
    15 root      20   0      0      0      0 I    0.0   0.0   0:00.00 rcu_tasks_trace_kthread
    16 root      20   0      0      0      0 S    0.0   0.0   0:00.43 ksoftirqd/0

```

図 3.6 top コマンドの実行結果

4.Web サーバの構築

4.1 Apache HTTP Server のインストール

パッケージを更新したり、インストールしたりするためには、パッケージ管理ツールである apt コマンドを使用する。

Apache をインストールするためには以下のコマンドを実行する。インストールする前にパッケージの更新を行う。

リスト 4.1 更新とインストール

\$ sudo apt update	パッケージの更新
\$ sudo apt -y install apache2	Apache HTTP Server のインストール

```
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/systemd/system/apache-htcacheclean.service.
ufw (0.36.1-4build1) のトリガを処理しています ...
man-db (2.10.2-1) のトリガを処理しています ...
libc-bin (2.35-0ubuntu3) のトリガを処理しています ...
```

図 4.1 Apache HTTP Server をインストールするコマンドを打った結果

4.2 Apache HTTP Server の起動

Apache を起動するには以下のコマンドを入力する。実行しても特に何も表示されることはない。

リスト 4.2 Apache の起動

```
$ sudo systemctl start apache2
```

次に Apache の自動起動及びステータスを確認していく。ChatGPT で「apache の自動起動の有効化やステータスのコマンドを使う利点を教えてください」と聞いてみるとサーバのダウンタイムを最小限にできることやシステムの簡略化を行えること、ステータスに関してはエラーチェックやデバックがしやすいとの返答が得られた。また便利な Apache 管理コマンドが出力されたため自動起動、ステータスの確認のコマンドとともに以下に示す。

リスト 4.3 便利なコマンド一覧

\$ 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
\$ sudo systemctl stop apache2	Apache を停止
\$ sudo systemctl restart apache2	Apache を再起動
\$ sudo systemctl reload apache2	設定を変更した後、Apache をリロード

\$ sudo journalctl -u apache2 --no-pager	Apache のログを表示
sudo apachectl configtest	Apache の設定ファイルの構文チェック

最終的にステータスを確認すると以下のようになる。

```
desktop:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2024-12-23 13:46:16 JST; 1min 19s ago
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 11818 (apache2)
      Tasks: 55 (limit: 4198)
     Memory: 5.4M
        CPU: 40ms
    CGroup: /system.slice/apache2.service
            └─11818 /usr/sbin/apache2 -k start
              └─11819 /usr/sbin/apache2 -k start
                └─11820 /usr/sbin/apache2 -k start

12月 23 13:46:16 isao-desktop systemd[1]: Starting The Apache HTTP Server...
12月 23 13:46:16 isao-desktop apachectl[11811]: AH00558: apache2: Could not reliably determine the s
12月 23 13:46:16 isao-desktop systemd[1]: Started The Apache HTTP Server.
lines 1-16/16 (END)
```

図 4.2 ステータスを確認

Active の箇所が active(running)となっているので起動できているとわかる。ブラウザを立ち上げて URL の箇所に `http://localhost` と入力すると Web サーバにアクセスできる。アクセスすると以下のような画面が表示される。

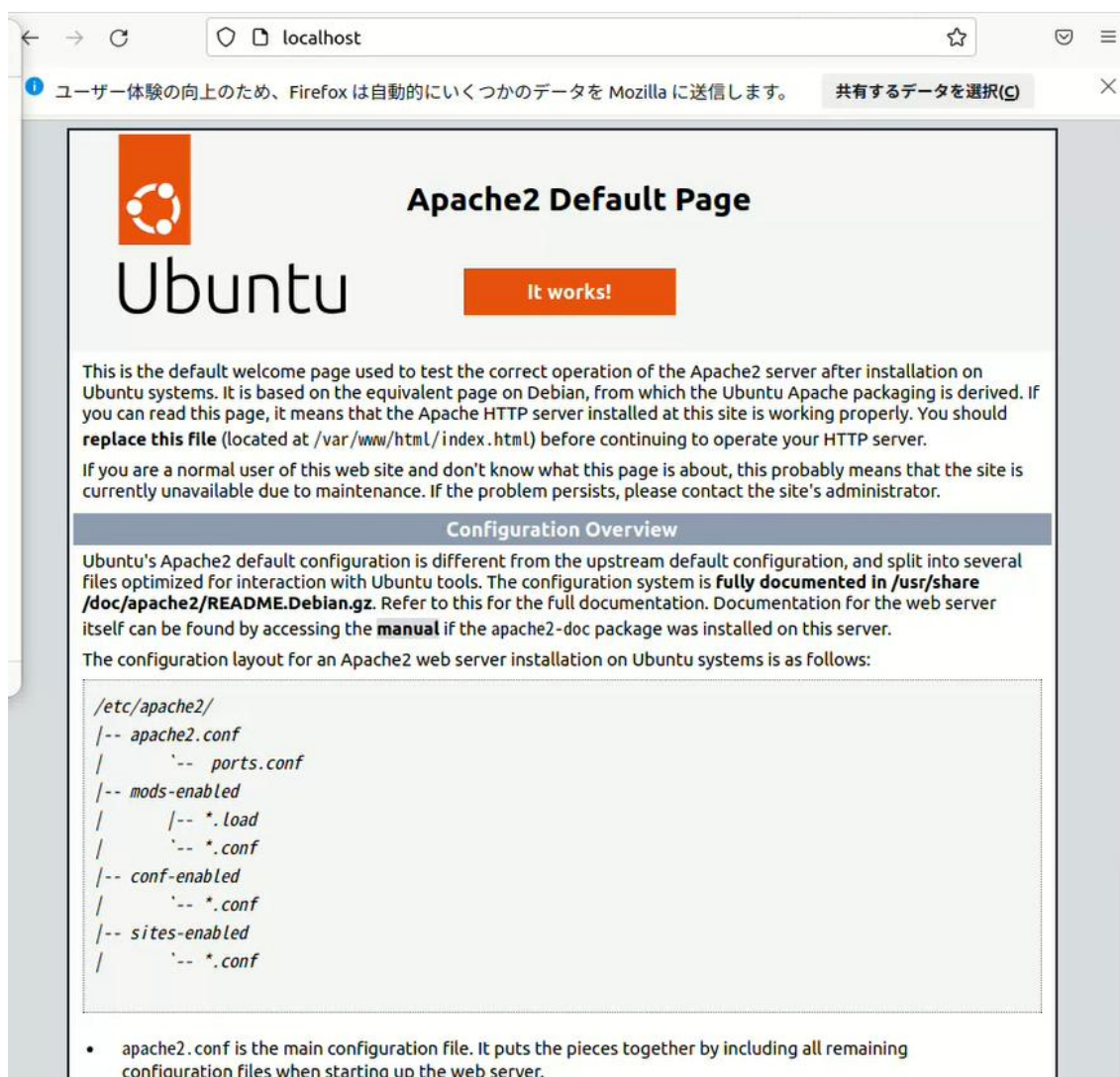


図 4.3 ウェルカムページ

4.3 仮想ホストの作成

仮想ホストを作成する。仮想ホストを作成することで1つのサーバで複数のドメインを運用することができる。ChatGPTで「仮想ホストを作成するメリットは何ですか？」と聞くとコスト削減のために1台のサーバで複数サイトを運用できるほか、ドメインごとに異なる設定が可能で異なるサービスを提供できる。本番環境・開発環境を整理しやすい、ログ管理がしやすく、負荷分散が可能。との返答が得られた。そのため、これを行うことは有用であるといえる。

I. 設定ファイルの作成

今回はデフォルトの仮想ホストの設定ファイルを残したまま異なる仮想ホストを作る。ファイルの作成場所は`/etc/apache2/sites-available/`で拡張子は`.conf`とする。設

定ファイルを作成するコマンドを以下に示す。またこれまでと同様に任意の名前を付けてよいところを赤色としている。

リスト 4.4 設定ファイルの作成

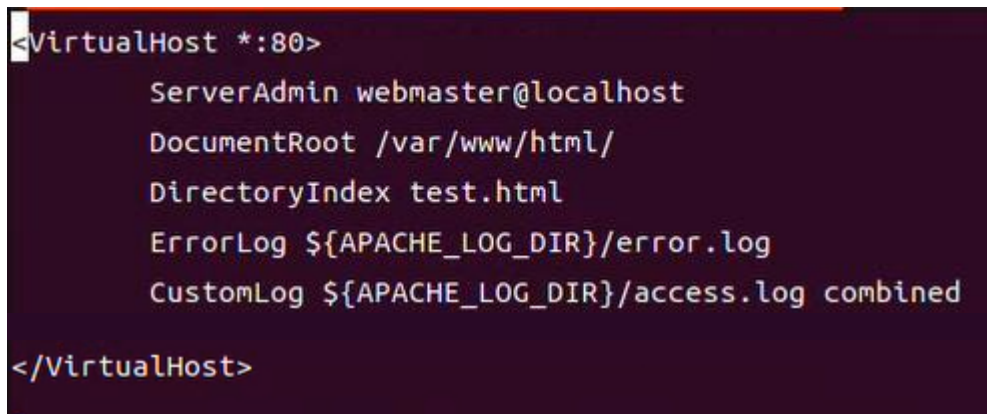
```
$ sudo touch /etc/apache2/sites-available/test.conf
```

II. 設定ファイルの編集

先ほど作成したファイルを vim コマンドで編集する。以下に先ほど作成した test.conf ファイルを編集するコマンドとどこを編集するかを示す。図 4.4 の 3 行目と 4 行目をそれぞれ、コンテンツをデプロイするディレクトリを指定、トップページとして表示したいファイルを指定している。

リスト 4.5 設定ファイルの編集

```
$ sudo vi /etc/apache2/sites-available/test.conf
```



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

図 4.4 設定ファイルの変更

III. ディレクトリの作成

フォルダを整理、管理しやすいようにディレクトリを作成する。またトップページに当たるファイルも作成する。コマンドは以下に示し今までと同様に任意の名前やパスを赤字で記述する。

リスト 4.6 ディレクトリ及びファイルの作成

\$ sudo mkdir -p /var/www/html/test	ディレクトリの作成
\$ sudo touch /var/www/html/test.html	ファイルの作成

IV. ページを作成

先ほど作成した test.html にページのレイアウトを記述していく。

リスト 4.7 test.html の編集

```
$ sudo vi /var/www/html/test.html
```

リスト 4.8 test.html に書き込み

```
<html>
  <body>
    <h1>Welcome to my page</h1>
  </body>
</html>
```

V. 仮想ホストの有効化

次に先ほどの仮想ホストを Web サーバに反映させるために仮想ホストを有効化し、サーバに反映する。以下にそのコマンドを示す。

リスト 4.9 仮想ホストを有効化及び反映

\$ sudo a2ensite test.conf	仮想ホストの有効化
\$ sudo systemctl reload apache2	仮想ホストをサーバに反映

VI. デフォルトの仮想ホストの無効化

先ほどは仮想ホストを有効化したので、次は仮想ホストを無効化していく。今回はデフォルトのページを無効化する。

リスト 4.10 仮想ホストを無効化及び反映

\$ sudo a2dissite 000-default.conf	仮想ホストの無効化
\$ sudo systemctl reload apache2	仮想ホストをサーバに反映

4.4 ファイアウォールの設定

セキュリティのためにファイアウォールを設定していく。ファイアウォールを設定するためには ufw コマンドを使用する。以下にファイアウォールの代表的なコマンドを示す。また今回はファイアウォールの有効化と 80 番ポートの開放を行い正しく行えているかを確認する(上三行を実行する)。

リスト 4.10 ファイアウォールの設定

\$ 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	状態とルールの確認
\$ sudo ufw disable	ファイアウォールの無効化
\$ sudo ufw status verbose	状態とルールに加えて各種デフォルトポリシーが表示される
\$ sudo ufw status numbered	ルール番号を確認できる
\$ sudo ufw delete 番号	ルールを消去する
\$ sudo ufw allow 'Apache Secure'	443 番ポートの開放

4.5 パスワード認証の設定

特定のユーザのみが閲覧できるページを作成するためにパスワードを設定するコマンドがある。今回は認証方法として基本認証を用いる。まずは以下のコマンドを入力し apache2-utils パッケージをインストールする。

リスト 4.11 インストールコマンド

\$ sudo apt install apache2-utils

Htpasswd コマンドを使用してユーザ登録を行う。ユーザ登録を行ったことがあるか無いかで少しコマンドが変わるので注意。username の箇所は任意の名前。

リスト 4.12 ユーザの登録

\$ sudo htpasswd -c /etc/apache2/.htpasswd username	初回(-c を付ける)
\$ sudo htpasswd /etc/apache2/.htpasswd username	二回目以降

実行すると以下ようになる。

```
desktop:~$ sudo htpasswd -c /etc/apache2/.htpasswd username
New password:
Re-type new password:
Adding password for user username
```

図 4.5 ユーザの登録

次に設定ファイルに移動(\$ sudo vi /ect/apache2/stites-available/test.conf)し以下の
ように編集する。<Directory>の箇所が追記箇所であり、"var/www/html"は認証を利用
するディレクトリパスである。

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

    <Directory "var/www/html/">
        AuthType Basic
        AuthName "Restricted Content"
        AuthUserFile /etc/apache2/.htpasswd
        Require valid-user
    </Directory>
</VirtualHost>
```

図 4.6 設定ファイルの編集

以下に記述した意味及びほかの設定をする際の書き方を示す。Require の箇所は
ChatGPT を用いてディレクティブを探している。

リスト 4.13 設定の意味

AuthType Basic	認証の種類指定	None, Basic, Digest, Form
AuthName	認証時に表示されるポップアップタイトル	
AuthFile	ユーザを登録したファイルのパス	
Require	アクセス可能なユーザを定義する	valid-user：全ユーザ可 all granted：全ユーザ可 all denied：全ユーザ不可 ip xxx.xxx.x.x/xx：指定 IP 可

変更が完了したら Apache を再起動する。

リスト 4.14 Apache の再起動

```
$ sudo systemctl reload apache2
```

4.6 アクセスログ

デフォルトでは、アクセスログは/var/log/apache2/access.log に記録される。確認方法は cat コマンドを使う。

リスト 4.15 アクセスログの確認

```
$ cat /var/log/apache2/access.log
```

5.SSH

5.1 ssh のサーバ構築

SSH はリモートホスト間の通信を安全に行うための仕組み。認証機能や暗号化を用いてリモート操作やファイル転送ができる。

I. OpenSSH のインストールと起動

SSH サーバを構築する際、OpenSSH を使用する。以下に OpenSSH のインストールとその起動方法を示す。

リスト 5.1 OpenSSH

\$ sudo apt install openssh-server	OpenSSH のインストール
\$ sudo systemctl start ssh	起動
\$ sudo systemctl enable ssh	自動起動の有効化
\$ sudo systemctl status ssh	ステータスの確認

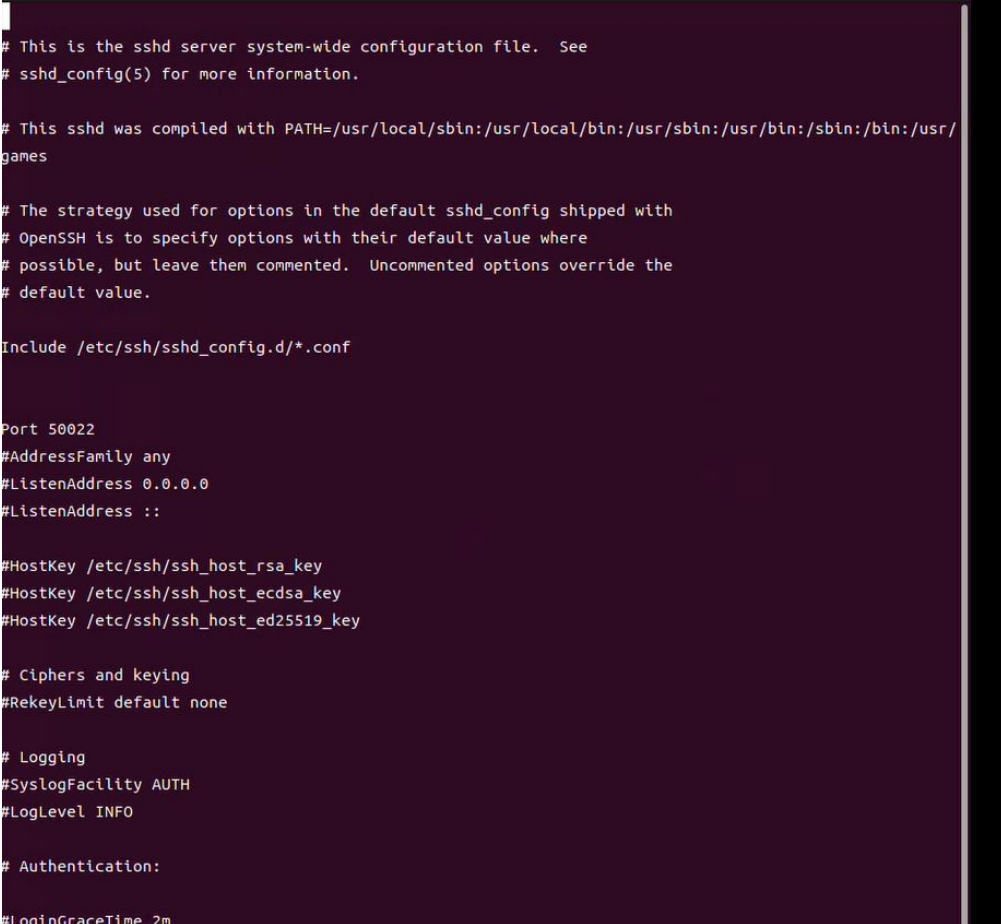
II. ポート番号の設定

OpenSSH はポート番号がデフォルトで決定されているがセキュリティの観点から別の番号に変更する方が良い。デフォルトで 22 番であることから下二桁を 22 にすることが多い。設定ファイルに変更を加える前に念のためバックアップを取る。

リスト 5.2 バックアップを取る

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

次にポート番号を設定する。vi コマンドで/etc/ssh/sshd_config を編集する。
Port22 というコメントアウトされた箇所があるためそこを好きな数に変える(今回は
50022 とした。)

A screenshot of a terminal window displaying the contents of the /etc/ssh/sshd_config file. The file is a configuration file for the SSH daemon. The 'Port' line is set to '50022'. Other visible settings include 'AddressFamily any', 'ListenAddress 0.0.0.0', 'HostKey' paths for rsa, ecdsa, and ed25519 keys, 'Ciphers and keying' section with 'RekeyLimit default none', 'Logging' section with 'SyslogFacility AUTH' and 'LogLevel INFO', and 'Authentication' section with 'LoginGraceTime 2m'.

```
# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/
games

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

Port 50022
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
```

図 5.1 ポート設定

ファイアウォールに SSH のポート利用許可ルールを追加するため、以下のコマン
ドを入力する。50022 は先ほど設定した Port の番号である。

リスト 5.3 ファイアウォールの設定

```
$ sudo ufw allow 50022/tcp
```

III. ログインの管理

ログインの制限や禁止、許可等の設定パラメータを以下に示す。

リスト 5.4 ユーザ管理

PermitRootLogin no	root ログインの設定
AllowUsers ユーザ名	許可ユーザの設定

5.2 SSH ログイン

別 PC(windows)で接続可能かテストする。まずは IP アドレスを確認する。IP アドレスの見方を以下に示す。したがって 4 行目の 10.133.6.207 を覚えておく。

リスト 5.5

```
$ ip a
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s25: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 70:85:c2:03:98:6b brd ff:ff:ff:ff:ff:ff
    inet 10.133.6.207/21 brd 10.133.7.255 scope global dynamic noprefixroute enp0s25
        valid_lft 2547sec preferred_lft 2547sec
    inet6 fe80::e2a3:1743:ba38:1704/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

図 5.2 IP アドレス確認

コマンドはリスト 5.6、実行結果は図 5.3 に示す。赤字の箇所は設定したものを使用する、また IP アドレスはホスト名でもよい。

リスト 5.6 コマンド

```
$ ssh ユーザ名@ip アドレス -p 50022
```

```
PS C:\Users\iyukk> ssh      @10.133.6.207 -p 50022
@10.133.6.207's password:
Welcome to Ubuntu 22.04 LTS (GNU/Linux 6.8.0-49-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

619のアップデートはすぐに適用されます。
これらの更新の415は、標準のセキュリティ更新です。
これらの追加アップデートを確認するには次を実行してください: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Last login: Mon Jan 20 14:41:39 2025 from 10.133.2.64
@: ~desktop:~$
```

図 5.3 SSH ログイン

5.2 2 段階認証

今回はパスワード認証と公開鍵認証を両方用いて 2 段階認証の設定を行う。

I. 公開鍵、秘密鍵を生成する

サーバを作成していない方の PC（今回は先ほどログインした WindowsPC を用いる）以下のコマンドを入力し、鍵を生成する。暗号化アルゴリズムが複数あるが主流である ed25519 を使用する。また -f の右側は任意のファイル名

リスト 5.7 鍵生成

\$ ssh-keygen -t ed25519 -f ./ssh/id_ed25519	dsa, rsa, ecdsa, ed25519 がある dsa は脆弱のため消去予定
--	--

II. 公開鍵を登録する

Linux や Mac は ssh-copy-id コマンドを用いて行うことができるが、Windows ではできないので生成したカギをクリップして直接サーバにファイルとして書き出す。使用するコマンドを以下に示す。赤字は設定したものを使用する。

リスト 5.8 公開鍵の登録

<pre>cat ~/.ssh/id_ed25519.pub ssh ユーザ名@ip アドレス -p 50022 "mkdir -p ~/.ssh && chmod 700 ~/.ssh && cat >> ~/.ssh/authorized_keys && chmod 600 ~/.ssh/authorized_keys"</pre>	Id_ed25519.pub : 公開鍵のファイル 名 ip アドレス : ホス ト名でも可 50022 : ポート番 号
---	---

III. 設定ファイルの編集

2 段階認証を有効化するために設定ファイルを編集する。最初の PC に戻って `sudo vi /etc/ssh/sshd_config` からファイルを図 5.4 のように変える
(AuthenticationMethods publickey,password と追記する。)

```
# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/
games

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

Port 50022
PermitRootLogin no

AuthenticationMethods publickey,password

#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
```

図 5.4 ファイルの編集

VI. 確認

まず秘密鍵のファイル（ホーム/.sshに入っている）を別の場所に移動させるか消去する。すると図 5.5 のようになり、入ることができない。

```
PS C:\Users\iyukk> ssh @10.133.6.207 -p 50022
@10.133.6.207: Permission denied (publickey).
```

図 5.5 鍵がない場合

次に鍵を元に戻して実行する。すると図 5.6 のように Password が聞かれた上で入ることができる。

```
PS C:\Users\iyukk> ssh @10.133.6.207 -p 50022
@10.133.6.207's password:
Welcome to Ubuntu 22.04 LTS (GNU/Linux 6.8.0-49-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

619のアップデートはすぐに適用されます。
これらの更新の415は、標準のセキュリティ更新です。
これらの追加アップデートを確認するには次を実行してください: apt list --upgradable

The list of available updates is more than a week old.
```

図 5.6 鍵がある場合

6.Docker

Docker のインストール、コンテナの起動及び停止を行う。Docker は OS やバージョンの違いの動作に素早く対応でき、デプロイの速度を上げることができるというメリットがある。また簡単に環境構築を行うこともできる。

6.1 Docker のインストール

リスト 6.1 を使用することで Docker をインストールすることができる。

リスト 6.1 Docker のインストール

```
# Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl
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

# Add the repository to Apt sources:
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
```



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

6.2 コンテナの起動

コンテナを起動するためにはイメージが必要である。今回は hello-world コンテナを使用する。以下にコンテナを起動するコマンドを示す。

リスト 6.2 コンテナの起動

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

次にコンテナが起動しているか確認するために、コンテナの一覧を確認する。以下にコンテナ一覧およびイメージやステータス等を表示するコマンドを示す。

リスト 6.3 コンテナ一覧

\$ docker container ls -a	-a の箇所は表示内容を変更するといったオプションを指定できる -a は停止中のコンテナも表示するオプション
---------------------------	---

6.3 コンテナの停止

コンテナを停止するコマンドを以下に示す。またコンテナを削除するコマンドも示す。CONTAINER の箇所にはコンテナ ID かコンテナ名を指定する。

\$ docker container stop CONTAINER	コンテナを停止する。CONTAINER の前には OPTIONS を、後ろには複数の名前を書きまとめてコンテナを停止できる
\$ docker container rm CONTAINER	コンテナを削除する。CONTAINER の前後は停止時と同じように記述できる

7.参考サイト一覧

ChatGPT : <https://chatgpt.com/>

【完全初心者向け】Linux の root ユーザとは? : <https://eng-entrance.com/linux-root>

root 権限の使い方 : <https://envader.plus/course/1/scenario/1004>

5.2 アカウントの種類：<https://l-study.arcjp.com/lpic/linux-essentials/linux-essentials-online-text/le-5-2/>

ubuntu の root パスワード設定、変更(CLI)：

<https://qiita.com/htako/items/cf84dc40f65a1af30923>

Linux グループ一覧の確認と/etc/group ファイル：

https://kazmax.zpp.jp/linux_beginner/etc_group.html

グループとは何かを理解する：<https://linuc.org/study/column/4077/>

apt コマンドチートシート：

https://qiita.com/SUZUKI_Masaya/items/1fd9489e631c78e5b007

Linux - パッケージ管理ツール dpkg と apt の使い方：[Linux - パッケージ管理ツール dpkg と apt の使い方 | pystyle](#)

ufw の基本操作：https://qiita.com/RyoMa_0923/items/681f86196997bea236f0

Basic 認証を使ったアクセス制限を行う：

<https://www.javadrive.jp/apache/allow/index3.html>

Basic 認証のつけかた：<https://qiita.com/yuwaita/items/e406c645ae8d0cc3c3fa>

Linux でアクセスログを確認する方法：完全ガイド：<https://ittrip.xyz/soft/linux/linux-access-log-guide>

Docker とは？使い方やメリット・デメリットを徹底解説！：

https://www.geekly.co.jp/column/cat-technology/1902_047/