

知能情報実験実習

3 年 I コース

福田捷人

Web サーバー構築手順書

目次

1.1 環境構築とセットアップ	1
1.2 Linux のインストール	1
1.3 設定に必要な知識	3
2.1 Apache の設定	7
2.2 インストール	7
2.2 ファイアウォールの設定	8
3.1 Docker のインストールと設定	9
4.1 Docker Compose について	11
4.2 Docker Compose とは	11
4.3 実用的な例	11
参考 URL	14

1.1 環境構築とセットアップ

使用 os : asus ノート pc

インストール後の Ubuntu のバージョン: 20.04

1.2 Linux のインストール

Ubuntu のインストールと Rufus のダウンロードを行かない選択からインストールし

たブートを選ぶその後 USB をさし、スタートを押す。(図 1.1 参照)

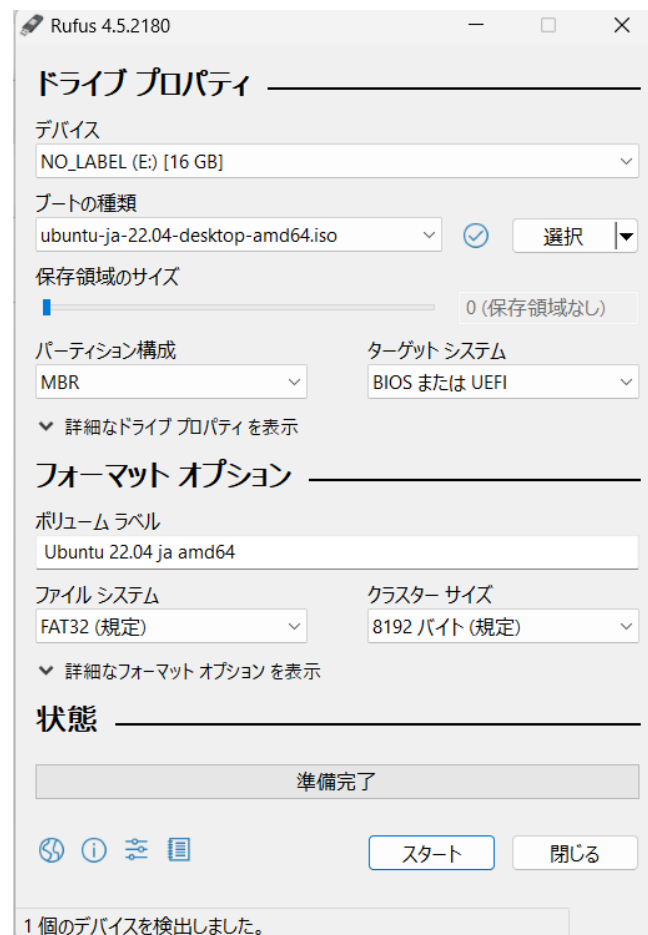


図 1.1

その後ノート pc を起動し電源を押したのちに F2 キーを長押しする, そうすること

よって BIOS を起動する(図 1.2) Boot Priority 欄に Ubunt の表記があるためそれを最も上に持ってくる.

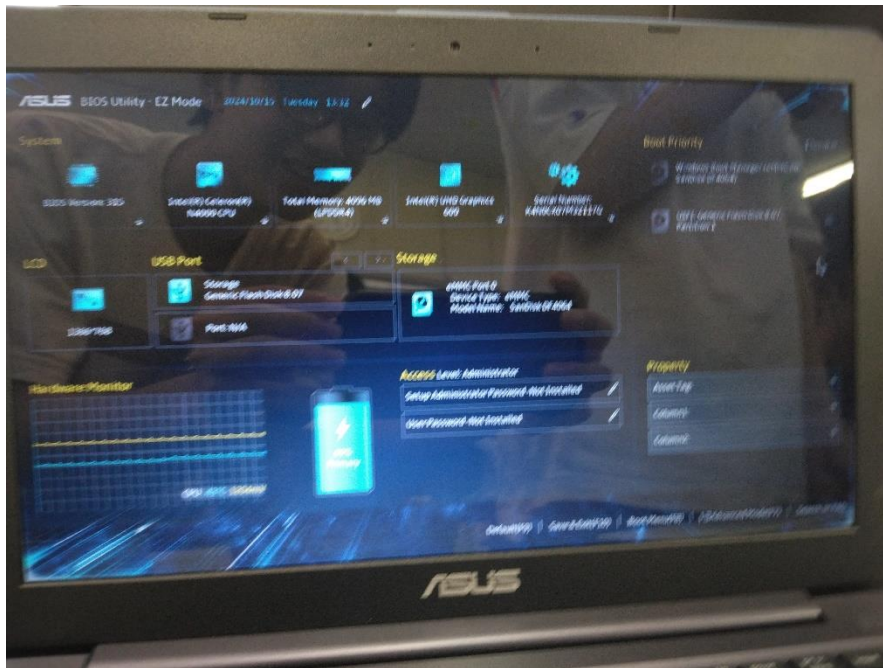


図 1.2

Ubunt をインストールを選択(図 1.3)



図 1.3

表示される画面のままに進め、(図 1.4)の画面になったら Ubuntu を選択する.

なお wifi の接続はどちらでもよい.

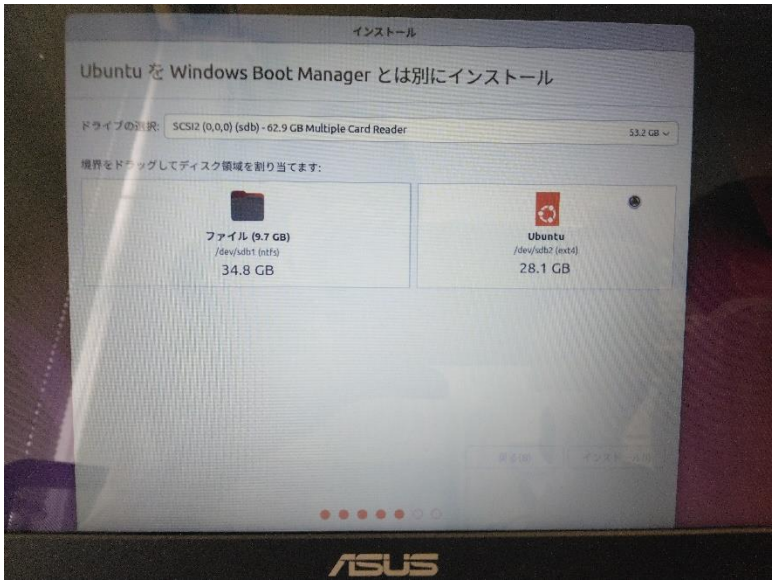


図 1.4

1.3 設定に必要な知識

①Linux におけるアクセス権変更に必要なコマンド

対象	操作	権限
● u:所有者	● +:権限の追加	● r: 読み取り権
● g:所有グループ	● -:権限の削除	● w:書き込み権
● o:その他ユーザ	● =:権限の指定	● x:実行権
● a:全てのユーザ		

コマンドの書式

chmod 対象+権限 ファイル名(ディレクトリ名)

②Linux について

- ・ コマンドを書く際に[sudo]と先頭につけることがあるこれはユーザー権限で実行する際に書かれることが多く、筆者の経験では動かない際にこれを先頭に着けることを忘れているということが多々ある.

- ・ Linux のファイルシステム構造は(図 1.5)のようになっている bash を開いた際にデフォルトでカレントディレクトリになっており var ディレクトリや usr ディレクトリにアクセスするには以下のように入力する必要がある.

```
cd ../..
```

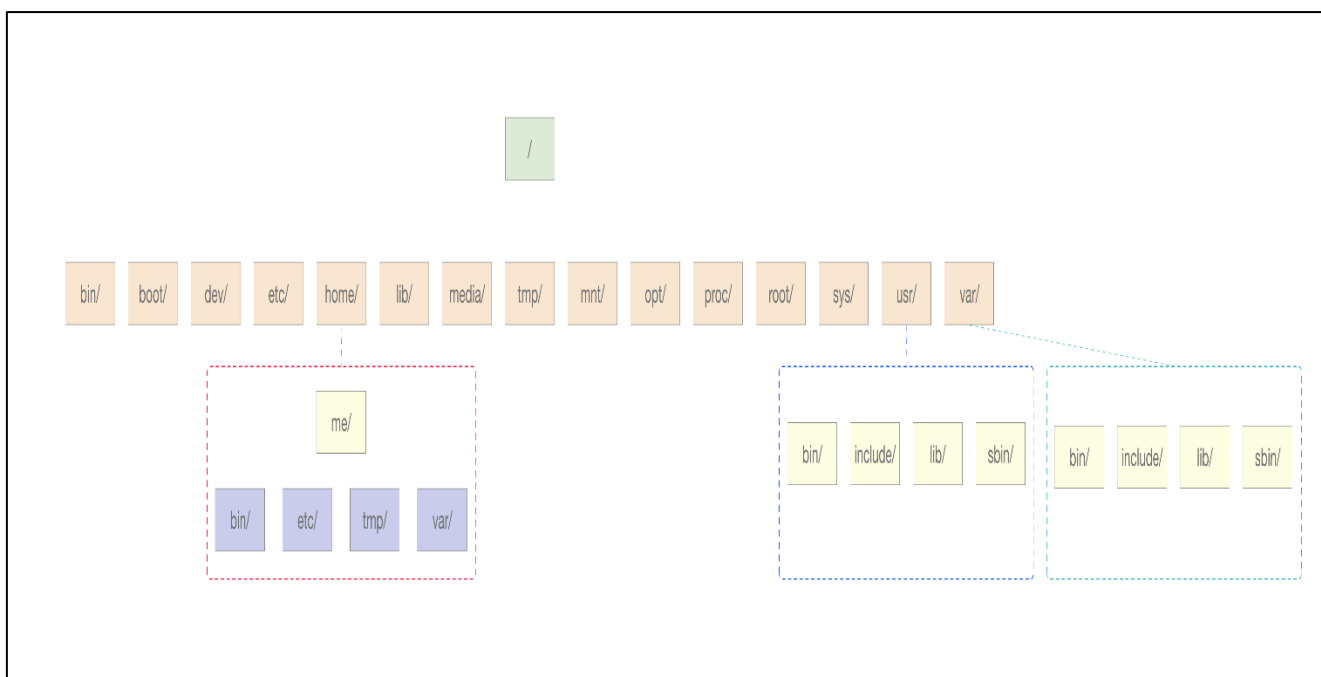


図 1.5

③Docker とは

- ・ Docker とは軽量で高速に動作するコンテナ型仮想環境用のプラットフォームである。従来の仮想化技術と異なり、ホスト OS のカーネルを共有することでゲスト OS を必要とせず、軽快に動作する。Docker を使用することで、1 台のサーバー上で複数のアプリケーション実行環境を作成・利用でき、環境の移転や共有も容易に行える。

- ・ インストールについて

Docker は、軽量で効率的なコンテナ型仮想化プラットフォームであり、アプリケーションの開発、配布、実行を容易にする技術である。Ubuntu に Docker をインストールする方法は主に Docker Engine と Docker Desktop の 2 つに大別される。

Docker Engine は、Ubuntu のターミナルで Docker を利用するための標準的な方法であり、インストール手順には古いバージョンの削除、必要なパッケージのインストール、GPG 鍵の取得、Docker リポジトリの追加、Docker CE パッケージのインストールが含まれる。インストール後は、Docker デーモンの起動と一般ユーザーの権限設定が必要である。

Docker Desktop は、グラフィカルインターフェースを提供し、統合

Kubernetes クラスタやサードパーティ拡張機能をサポートする。ただし、特

定の Ubuntu バージョンと AMD64 プロセッサを必要とする。

インストール方法の選択は、ユーザーのニーズと環境に依存する。ターミナ

ル作業に慣れた軽量環境を求める場合は Docker Engine、グラフィカルインタ

ーフェイスと多機能性を望む場合は Docker Desktop が適している。

Docker の導入により、開発環境の構築と共有が容易となり、アプリケーション

ンの移植性が向上し、リソースの効率的な利用が可能となる。

2.1 Apache の設定

Apache という web サーバソフトをインストールする。

Apache の特徴として OS を選ばずに動作することやオープンソースなので無料で使用可能である点が挙げられる。

2.2 インストール

①bash に以下のコマンドを入力する。これによって Apache のインストールされる

```
sudo apt update
```

```
sudo apt -y install apache2
```

②サービスを起動しステータスを確認する。

```
sudo systemctl start apache2
```

```
sudo systemctl enable apache2
```

```
sudo systemctl status apache2 #このコマンドがステータスの確認
```

③ブラウザを開き、以下のいずれかの URL を入力:

- ローカル環境の場合: `http://localhost`
- 別 PC からアクセスする場合: `http://[サーバの IP アドレス]`

2.2 ファイアウォールの設定

①ファイアウォールで Apache の許可を行う.

```
sudo ufw allow 'Apache'
```

②ファイアウォールの設定の確認するコマンドを実行し以下のような結果を得られる.

```
sudo ufw status
```

To	Action	From

Apache	ALLOW	Anywhere
Apache (v6)	ALLOW	Anywhere (v6)

3.1 Docker のインストールと設定

①Docker のインストールのために以下のコマンドを実行する。なお1つのコマンドごとに・で区切っており実際に実行する際には必要ない。

```
• sudo apt update

• sudo apt install -y apt-transport-https ca-certificates curl software-properties-

common

• curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o

/usr/share/keyrings/docker-archive-keyring.gpg

• echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]

https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee

/etc/apt/sources.list.d/docker.list > /dev/null

• sudo apt update

• sudo apt install -y docker-ce
```

コマンドを上から1つずつ解説すると、

- 最新のソフトウェアリストを取得する。
- Docker のインストールに必要な追加ツールをインストール

- ・ダウンロード元の信頼性を確保する
- ・ Docker パッケージをインストールするためのリポジトリを追加
- ・最後の2つで Docker エンジンを実インストール

という手順になっている。

4.1 Docker Compose について

4.2 Docker Compose とは

Docker Compose は、複数のコンテナで構成される Docker アプリケーションの設定を記述し、実行するツールである。

複数のコンテナ（データベース、キャッシュ、Web API など）で構成されたシステムを `docker` コマンドだけで起動する場合、個別コンテナの設定をパラメタで指定しながら、コンテナの依存関係の順にコマンドで起動していく必要があるが、

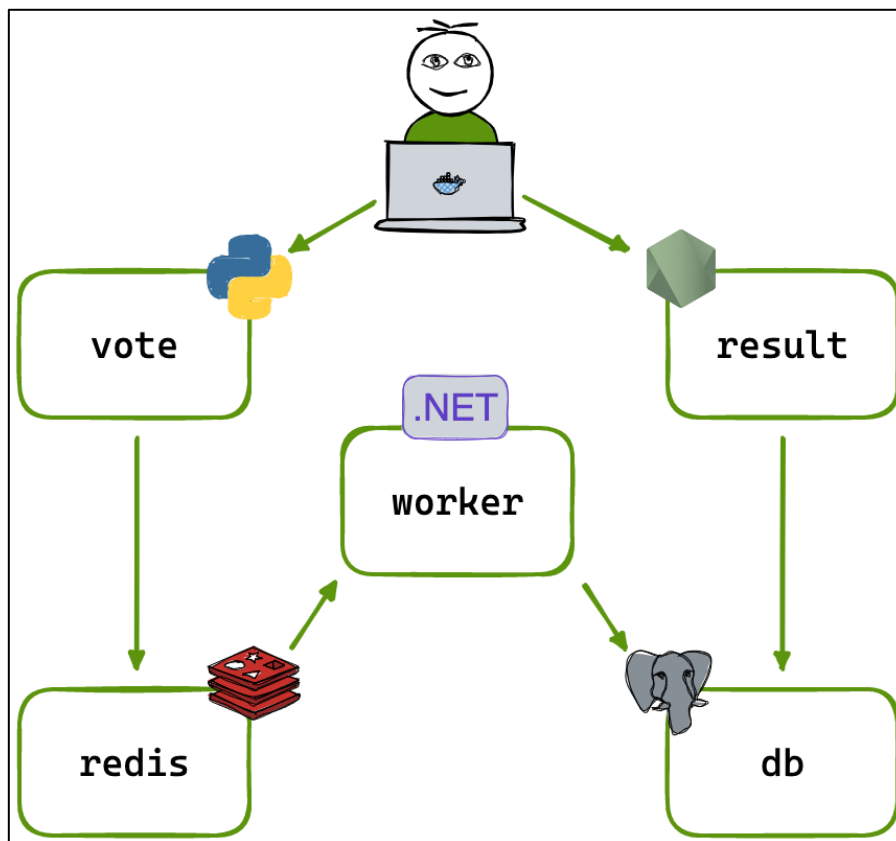
Docker Compose を使うと複数コンテナの設定や依存関係を設定ファイルに記述し、単一のコマンドで複数コンテナを一度に起動できる。

4.3 実用的な例

本記事は実際に動かすことではなく仕組みの理解を目的としているためサンプルは興味のあるかたは [Github リポジトリ](#)を参照してください。

Github 上に公開されている Docker 公式サンプルの[voting app]についてみていく

本アプリは以下のような構成で構築されている(図 4.1)



<https://github.com/dockersamples/example-voting-app> より図 4.1

アプリケーションの構成要素は以下の通りで

- Python で書かれたフロントエンドウェブアプリ（投票インターフェース）
- 新しい投票を収集する Redis インスタンス
- 投票を処理する .NET ワーカー
- 投票データを保存する PostgreSQL データベース
- リアルタイムで投票結果を表示する Node.js ウェブアプリ

なおこの文章は ai によって書かせたものである。

このアプリケーション自体が voting-app（投票アプリ）であり、複数の Docker コンテナ上で動作する単純な投票システムを実装している。

参考 URL

(出展: <https://hnavi.co.jp/knowledge/blog/apache/>

<https://qiita.com/nys9302/items/a8ddeedd3cd9d0deb332>

<https://www.kagoya.jp/howto/cloud/container/docker/>

<https://www.rstone-jp.com/column/107612/>

<https://qiita.com/yoshiyasu1111/items/f2cab116d68ed1a0ce13>

<https://kinsta.com/jp/blog/install-docker-ubuntu/>

https://qiita.com/ohhara_shiojiri/items/486a54ad895d6bb3144e

<https://www.ogis-ri.co.jp/otc/hiroba/technical/docker/part4.html>

<https://qiita.com/prgseek/items/e557a371d7bd1f57b9b1>

<https://blog.denet.co.jp/centrally-manage-multiple-containers-with-docker->

[compose/](#)

<https://github.com/dockersamples/example-voting-app>

)