

2025年度

ソフトウェアシステム開発

第11,12回Webプラットフォーム1（余力ある人のみ、本番環境設定編②）

本番環境設定編②

本番環境設定編①では、「サーバ・ネットワーク」非履修者を対象に、Linuxサーバーのインストールおよび設定について取り扱ってきた。

本番環境設定編②は、「サーバ・ネットワーク」履修者も非履修者も対象とする本番環境の設置を取り扱います。

VirtualBox上にあるLinuxをほかのPCからアクセスできるようにする設定①

※「サーバー・ネットワーク」を履修していない人がサーバーを立ち上げる場合は、「(参考) VirtualBox上にAlmaLinuxのインストールする方法」というPPT記載の作業をあらかじめ行ってください。

① まず正常にLinux が動くかVirtualBox から Linuxを起動してみてください。

Windows Update などされている場合、ネットワークアダプターがクリアされている場合があります。

その時は、スタート→設定→ネットワークとインターネット より、

VirtualBox Host-Only Ethernet Adapterを有効化してください。

② 正常に起動できることを確認したら、ターミナルから次のコマンドを実行

```
shutdown -h -t 0 now
```

VirtualBox上にあるLinuxをほかのPCからアクセスできるようにする設定②

- ① VirtualBox にて、AlmaLinuxが起動していないことを確認。
- ② ネットワークをクリック。「アダプター1」の「割り当て」を「NAT」から「ブリッジアダプター」に変更する。
- ③ AlmaLinuxを起動する。
- ④ AlmaLinuxにログイン後、ターミナルを起動。
- ⑤ ターミナルにて、`hostname -I` と入力して実行
最初に出てくるIP アドレスを確認する。(仮に10.65.70.99とする。)
- ⑥ 違うPCのコマンドプロンプトにおいて、下記を実行
`ssh username@10.65.70.99`
※ username は適宜自分のユーザー名に置き換えて下さい。
- ⑦ ログインできたことを確認。

Node.js v18 のインストール

ターミナルで下記を実行

`su root`

`dnf module reset nodejs` → 既存のNode.jsモジュールの設定をリセット

`dnf module enable nodejs:18` → v18モジュールを有効化

`dnf install -y nodejs` → Node.jsをインストール

Express.js のインストール

以下をユーザーで実行して下さい。

(username は自分のユーザ名、passwordは自分のパスワードに置き換えて下さい。)

```
npm init -y
```

```
npm config set proxy http://username:password@proxy01.osaka.hal.ac.jp:8080
```

```
npm config set https-proxy http://username:password@proxy01.osaka.hal.ac.jp:8080
```

```
npm install express
```

vi エディタ

Expressが動くか試すためにサンプルコードを書きますが、その前にLinuxの軽量なエディタについて紹介します。

ターミナルにて

```
vi app.js
```

と打つと、テキストファイルapp.jsの編集画面になります。

編集する場合は a を押した後、編集することができます。

保存 + 終了は Esc 押した後、:wq と押してください。

詳しくviエディタを学びたい人は様々な解説のホームページがありますので随時参照してください。

Express 動作確認

vi を用いて、下記のファイルを作成してください。ファイル名 app.js

保存が終わったら node app.js と実行し、Firefox で <http://localhost:3000> にアクセスしてください。

```
const express = require('express');
const app = express();
app.get('/', (req, res) => `
  res.send('Hello Express');
});
app.listen(3000, () => {
  console.log('Server is running on http://localhost:3000');
});
```


他のPCからアクセスするための設定と確認

AlmaLinuxの3000番ポートを開けます。

```
su root
```

```
firewall-cmd --list-ports
```

```
firewall-cmd --permanent --add-port=3000/tcp
```

```
firewall-cmd --reload
```

その後、AlmaLinuxから `node app.js` を実行し、

ほかのコンピュータから `http://10.65.70.99:3000/` を参照して下さい。

SQLiteのインストール

以下をユーザーで実行して下さい。(先ほどのターミナルで行います)

```
npm install sqlite3
```

※ npm warn deprecated ...というワーニングが出ますが、無視してください。

SQLite の動作確認

vi を用いて、下記のファイルを作成してください。ファイル名 test-sqlite.js

保存が終わったら node test-sqlite.js と実行し、ほかのPCのブラウザから接続してみてください。

```
const sqlite3 = require('sqlite3').verbose();
const db = new sqlite3.Database(':memory:');
db.serialize(() => {
  db.run("CREATE TABLE users (id INT, name TEXT)");
  db.run("INSERT INTO users VALUES (1, 'Taro')");
  db.each("SELECT id, name FROM users", (err, row) => {
    console.log(row.id + ": " + row.name);
  });
});
db.close();
```

test-sqlite.js の解説

`sqlite3.Database(':memory:');` → `:memory:` とはメモリ空間上にデータベースを作成するという意味になります。終了後破棄されます。

`db.serialize(() => {` → 以下逐次実行します。

`db.run("CREATE TABLE users (id INT, name TEXT)");` → `users` というテーブルを作るSQL

`db.run("INSERT INTO users VALUES (1, 'Taro')");` → `1, Taro` をテーブルに挿入

`db.each("SELECT id, name FROM users", (err, row) => {` → `users` テーブルのデータをすべて取り出します。

`console.log(row.id + ": " + row.name);` → `SELECT`文の結果をコンソールに表示

`});`