

# Step5: Webアプリ構築 (AppServer)

# 1. 目的

---

- AppServerに簡単なWebアプリを導入し、ブラウザからデータベースを参照できるようにする
- 「Web + DB連携」を実務を意識した形で再現する

## 2. 環境準備 (AppServer)

### 1.PHPと必要モジュールをインストール

```
sudo dnf install -y php php-mysqlnd php-fpm
```

### 2.NginxとPHP-FPMを連携

/etc/nginx/conf.d/app.conf を作成：コードは次ページに記載

配置用コマンド：scp -P 2221 app.conf admin@localhost:/etc/nginx/conf.d

### 3.サービス起動

```
sudo systemctl enable --now php-fpm
```

```
sudo systemctl restart nginx
```

```
Installed:
  apr-1.7.0-12.el9_3.x86_64
  apr-util-openssl-1.6.1-23.el9.x86_64
  httpd-filesystem-2.4.62-4.el9.noarch
  mailcap-2.1.49-5.el9.noarch
  php-8.0.30-3.el9_6.x86_64
  php-fpm-8.0.30-3.el9_6.x86_64
  php-opcache-8.0.30-3.el9_6.x86_64
  apr-util-1.6.1-23.el9.x86_64
  httpd-2.4.62-4.el9.x86_64
  httpd-tools-2.4.62-4.el9.x86_64
  mod_http2-2.0.26-4.el9.x86_64
  php-cli-8.0.30-3.el9_6.x86_64
  php-mbstring-8.0.30-3.el9_6.x86_64
  php-pdo-8.0.30-3.el9_6.x86_64
  apr-util-bdb-1.6.1-23.el9.x86_64
  httpd-core-2.4.62-4.el9.x86_64
  libxslt-1.1.34-13.el9_6.x86_64
  mod_lua-2.4.62-4.el9.x86_64
  php-common-8.0.30-3.el9_6.x86_64
  php-mysqlnd-8.0.30-3.el9_6.x86_64
  php-xml-8.0.30-3.el9_6.x86_64

Complete!
```

## 2. 環境準備 (AppServer)

---

```
server {  
    listen      80;  
    server_name localhost;  
  
    root /var/www/html;  
    index index.php index.html;  
  
    location / {  
        try_files $uri $uri/ =404;  
    }  
  
    location ~ \.php$ {  
        include fastcgi_params;  
        fastcgi_pass   unix:/run/php-fpm/www.sock;  
        fastcgi_param  SCRIPT_FILENAME $document_root$fastcgi_script_name;  
    }  
}
```

# 3. Webアプリの配置

---

## 1. サンプルPHPを作成

コードは次ページに記載

## 2. 権限調整

```
sudo chown -R nginx:nginx /var/www/html
```

# 3. Webアプリの配置

---

```
sudo tee /var/www/html/index.php <<'EOF'
<?php
$servername = "192.168.56.102"; // DBServerのHost-Only IP
$username = "appuser";
$password = "StrongPassw0rd!";
$dbname = "appdb";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("DB接続失敗: " . $conn->connect_error);
}

$result = $conn->query("SELECT * FROM test");
```

```
echo "<h1>DB接続テスト</h1>";
if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        echo "ID: " . $row["id"]. " - Name: " . $row["name"]. "<br>";
    }
} else {
    echo "データなし";
}
$conn->close();
?>
EOF
```

# 4. 動作確認

---

## 1. ブラウザでアクセス

http://127.0.0.1:8080

## 2. 期待される結果

- ページタイトル: **DB接続テスト**
- 内容: **DBServer**の **test** テーブルに入れたデータ (例: ID:1 - Name:Hello DB)

---

## **DB接続テスト**

ID: 1 - Name: Hello DB

# 補足資料:DB接続エラーの場合の対処

---

## 権限・コンテキスト

- `sudo chown -R nginx:nginx /var/www/html`
- `sudo restorecon -Rv /var/www/html`

## SELinux 外向き通信を許可（汎用）

- `sudo setsebool -P httpd_can_network_connect on`

## サービス再起動

- `sudo systemctl restart php-fpm nginx`