

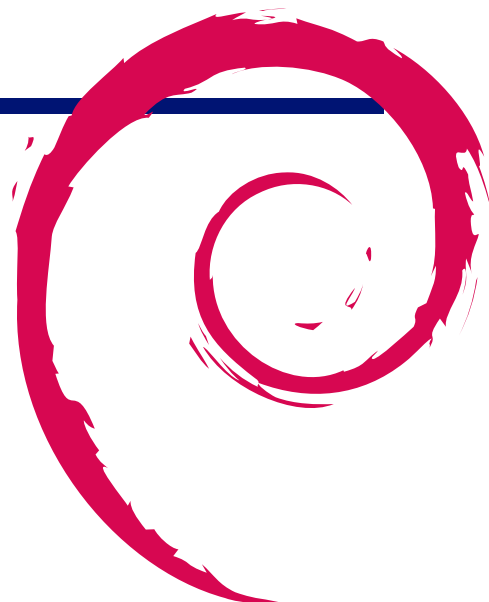
.Debian

銀河系唯一のDebian専門誌

2023年4月15日

PHP特集





1 最近の Debian 関連のミーティング報告

杉本 典充

1.1 2023 年 2 月度 東京エリア・関西合同 Debian 勉強会

2023 年 2 月 18 日 (土) に東京エリア Debian 勉強会と関西 Debian 勉強会の合同でオンラインによる Debian 勉強会を開催しました。参加者は 8 名でした。

セミナー発表は、dictoss さんによる「DDTP 及び DDTSS の紹介」、ハンズオン「DDTSS を使ってパッケージ説明文を翻訳してみよう」を行いました。

勉強会の終了後、参加者同士で Debian や OSS に関する話の情報交換を行いました。

1.2 オープンソースカンファレンス 2023 Online/Spring (兼 2023 年 3 月勉強会)

2023 年 3 月 11 日 (土) にオンラインで「オープンソースカンファレンス 2023 Online/Spring」^{*1}が開催されました。東京エリア Debian 勉強会ではセミナーを出展し、dictoss さんが「Debian Update」を発表しました^{*2}。発表では Debian、Debian JP Project、bookworm の概要を説明しました。

なお、セミナーの参加者は 17 名でした。

1.3 オープンソースカンファレンス 2023 Tokyo/Spring

2023 年 4 月 1 日 (土) に東京都立産業貿易センター台東館 7 階で「オープンソースカンファレンス 2023 Tokyo/Spring」^{*3} が開催されました。東京エリア Debian 勉強会ではブースを出展しました。

なお、イベントの来場者数は以下のとおりでした。

- イベント全体の来場者 約 450 名 (そのうち、出展者・関係者 約 190 名)
- Debian ブースへの来場者 87 名



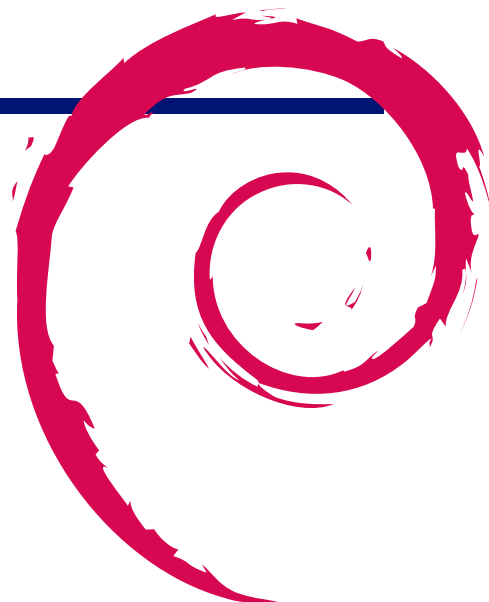
^{*1} <https://event.ospn.jp/osc2023-online-spring/>

^{*2} <https://tokyodebian-team.pages.debian.net/pdf2023/debianmeetingresume202303-osc2023online-presentation.pdf>

^{*3} <https://event.ospn.jp/osc2023-spring/>

2 事前課題

杉本 典充



今回の事前課題は以下です。

1. PHP で Web ページを作ったことはありますか。
2. mod_php という言葉は聞いたことがありますか。
3. php-fpm という言葉を聞いたことがありますか。

2.1 dictoss

1. はい
2. はい
3. はい

2.2 yy-y-ja-jp

1. いいえ
2. はい
3. はい

2.3 NOKUBI Takatsugu (knok)

1. いいえ
2. はい
3. はい

2.4 ysaito

1. はい
2. はい
3. はい

2.5 hatta0713

1. はい
2. はい
3. はい

2.6 Hiroyuki Yamamoto (yama1066)

1. いいえ
2. いいえ
3. いいえ

2.7 su_do

1. いいえ
2. いいえ
3. いいえ

2.8 hihitani

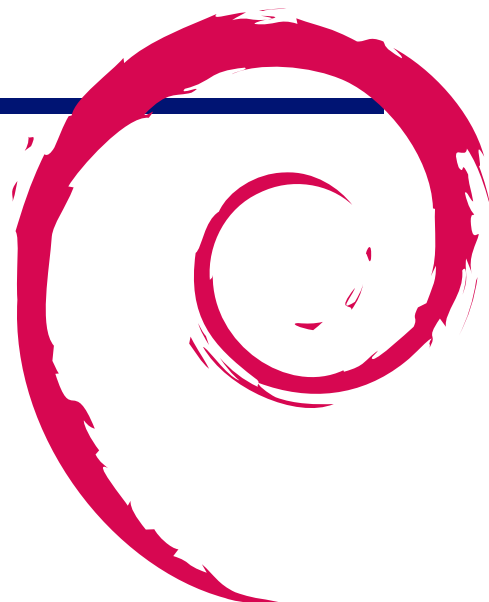
1. はい
2. いいえ
3. いいえ

2.9 kenhys

1. いいえ
2. はい
3. いいえ

2.10 あおき (sambar_gamer)

1. いいえ
2. いいえ
3. いいえ



3 Debian で PHP が動く Web サーバを構築してみる (apache2 / nginx)

杉本 典充

3.1 はじめに

最近 PHP で Web アプリの開発を少ししているため、Debian で PHP の Web アプリ開発の方法とサーバを構築する設定方法を調べてみました。なお、本記事における実行環境は Debian 12 bookworm とします。

3.2 Debian における PHP

3.2.1 PHP のインストール

Debian 12 bookworm では php というパッケージ名で PHP-8.2.4 を提供しています。

PHP を利用するには、以下の apt-get コマンドでインストールします。PHP は Web サーバ向けのスクリプト言語のため、依存関係の都合で apache2 が一緒にインストールされます。

```
# apt-get install php
(中略)
以下のパッケージが新たにインストールされます:
apache2 apache2-bin apache2-data apache2-utils libapache2-mod-php8.2 libapr1
libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libcurl4 liblua5.3-0
php php-common php8.2 php8.2-cli php8.2-common php8.2-opcache
php8.2-readline psmisc ssl-cert
アップグレード: 0 個、新規インストール: 20 個、削除: 0 個、保留: 0 個。
7,408 kB のアーカイブを取得する必要があります。
この操作後に追加で 31.2 MB のディスク容量が消費されます。
続行しますか? [Y/n]
```

インストールした php コマンドを実行してバージョンを確認してみます。

```
# php -v
PHP 8.2.4 (cli) (built: Mar 16 2023 14:24:40) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.2.4, Copyright (c) Zend Technologies
with Zend OPcache v8.2.4, Copyright (c), by Zend Technologie
```

3.2.2 PHP のスクリプト開発 最初の一步

ファイルをまとめるディレクトリを作成

PHP のスクリプト開発を行うにあたり、まずはファイルを保存するディレクトリを作成します。

```
$ cd
$ mkdir phpapp
$ cd phpapp
```

phpinfo()

まずは PHP のスクリプト開発を始めるときのお約束である、phpinfo() を呼び出すファイルを作成します。

```
$ vi phpinfo.php
<?php
    phpinfo();
?>
```

php コマンドで phpinfo.php を実行すると、PHP の実行環境について詳細な情報が出力されます。

```
$ php phpinfo.php | head
phpinfo()
PHP Version => 8.2.4

System => Linux deb12-webapp-test 6.1.0-7-amd64 #1 SMP PREEMPT_DYNAMIC
Debian 6.1.20-1 (2023-03-19) x86_64
Build Date => Mar 16 2023 14:24:40
Build System => Linux
Server API => Command Line Interface
Virtual Directory Support => disabled
Configuration File (php.ini) Path => /etc/php/8.2/cli
Loaded Configuration File => /etc/php/8.2/cli/php.ini
(省略)
```

PHP の簡単な処理

PHP で for を使って、箇条書きの HTML を出力する printli.php を作成してみます。

```
$ vi printli.php
<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="UTF-8">
<title>test printli</title>
</head>
<body>
<ul>
<?php
for($i = 0; $i < 5; $i++){
    $s_link = sprintf("./detail/%d/", $i);
    $s_str = sprintf("article %d", $i);
    print "<li><a href=\"$s_link\">$s_str</li>\n";
}
?>
</ul>
</body>
</html>
```

Web ブラウザからディレクトリ直下にアクセスした場合に表示するトップページとして index.php を作成します。ページ内には、先ほど作成した phpinfo.php と printli.php へのリンクがある箇条書きの HTML が作成しておきます。

```
$ vi index.php
<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="UTF-8">
<title>test page</title>
</head>
<body>
<ul>
<li><a href="./phpinfo.php">phpinfo</li>
<li><a href="./printli.php">printli</li>
</ul>
</body>
</html>
```

PHP のビルトインウェブサーバーでの Web 画面表示

PHP には「ビルトインウェブサーバー」という機能があります*4。この機能を使うと PHP を動くように設定した Web サーバ を用意せずとも、ユーザのホームディレクトリ内にある PHP スクリプトを実行して Web 画面の表示と動作を確認することができます。

```
$ php -S 0.0.0.0:8000
[Mon Apr 10 22:56:08 2023] PHP 8.2.4 Development Server (http://0.0.0.0:8000) started
```

PHP を開発しているサーバの IP アドレスとポート番号を含めた "http://192.168.1.2:8000/" などの URL を Web ブラウザで開くと index.php の Web 画面が表示されます。Web ページには phpinfophp と printli.php へのリンクが表示されますので、クリックして画面表示を確認してみましょう。

| PHP Version 8.2.4 | |
|---|--|
| System | Linux deb12-webapp-test 6.1.0-7-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.20-1 (2023-03-19) x86_64 |
| Build Date | Mar 16 2023 14:24:40 |
| Build System | Linux |
| Server API | Built-in HTTP server |
| Virtual Directory Support | disabled |
| Configuration File (php.ini) Path | /etc/php/8.2/cli |
| Loaded Configuration File | /etc/php/8.2/cli/php.ini |
| Scan this dir for additional .ini files | /etc/php/8.2/cli/conf.d |

図 1 index.php、phpinfo.php、printli.php

PHP パッケージマネージャ「composer」

PHP のパッケージマネージャに「composer」というコマンドがあります*5。composer を使うと PHP 向けのパッケージを配布する Web サイト*6 から、アプリの動作に必要な PHP のパッケージを一括でインストールやアップグレードすることができます。

composer は以下の apt-get コマンドでインストールできます。

```
# apt-get install composer
```

composer の使い方は、まずユーザ定義の設定ファイル "composer.json" を作成します。次に "composer install" コマンドを実行すると、composer.json の定義にしたがって PHP のパッケージをインストールします。コマンドの実行後は vendor ディレクトリ と composer.lock ファイルが生成され、インストールしたファイルと処理結果が保存されます。

```
$ vi composer.json
{
  "require": {
    "phpmailer/phpmailer": "~6.8.0"
  }
}
```

```
$ composer install
$ ls -l
composer.json composer.lock index.php phpinfophp printli.php vendor
合計 24
-rw-r--r-- 1 dictoss dictoss 67 4月 10 23:47 composer.json
-rw-r--r-- 1 dictoss dictoss 4039 4月 10 23:47 composer.lock
-rw-r--r-- 1 dictoss dictoss 216 4月 10 22:51 index.php
-rw-r--r-- 1 dictoss dictoss 24 4月 9 14:36 phpinfophp
-rw-r--r-- 1 dictoss dictoss 299 4月 10 23:25 printli.php
drwxr-xr-x 4 dictoss dictoss 4096 4月 10 23:47 vendor
```

*4 <https://www.php.net/manual/ja/features.commandline.websserver.php>

*5 <https://getcomposer.org/>

*6 規定では <https://packagist.org/> からダウンロードします。

3.2.3 PHP アプリケーションサーバ

mod_php

mod_php^{*7}とは apache で PHP を実行できるようにする組み込みモジュールであり、apache-1 時代から存在している PHP アプリケーションサーバの歴史ある実装です。Debian 12 bookworm では apache-2.4 向けに libapache2-mod-php8.2 パッケージで mod_php を提供しています。

linux 向けの apache-2.4 では MPM (マルチプロセッシングモジュール) が 3 つ含まれています (prefork、worker、event)。mod_php の実運用環境では apache を MPM マルチスレッドモードで使用することは推奨されていないため^{*8}、prefork MPM を選択する必要があります。

prefork MPM は 1 つの apache プロセスで同時に 1 つのクライアントのみを処理する方式であり、マルチスレッドの機能は実装されていないため動作がかなり安定しています。しかし、クライアント数が増えると PHP を処理する apache2 プロセスも連動して増えるためメモリの使用量がとても大きくなるデメリットがあります。

php-fpm

php-fpm^{*9}とは PHP における FastCGI^{*10} 実装です。なお、fpm とは "FastCGI Process Manager" の単語の頭文字をとった略称です。php-fpm には 高負荷のサイトで有用な機能が含まれています。Debian 12 bookworm では php-fpm というパッケージ名で php-fpm の機能を提供しています。

php-fpm は web サーバとは別の独立したプロセスで php を処理できるようバックグラウンドで待機し、web サーバから FastCGI プロトコルでの接続要求を受け入れて PHP スクリプトの実行結果を応答します。このとき、php-fpm のプロセスはクライアントの同時接続数に応じてプロセス数を自動で増減する仕組みがあります。

PHP の Web アプリケーションを提供するサーバでは多数のクライアントを処理するため、mod_php よりプロセス数が少なくメモリの利用効率が良い php-fpm を選択することを筆者はお勧めします^{*11}。

3.3 Debian で PHP が動く Web サーバの構築手順

3.3.1 apache2 + mod_php

apache2、php、libapache2-mod-php8.2 パッケージをインストールします。apache2 パッケージを単独でインストールすると MPM は event が有効になるのですが、mod_php である libapache2-mod-php8.2 パッケージのインストール処理で MPM が prefork へ切り替わります。

```
# apt-get install apache2 php libapache2-mod-php8.2
(中略)
Module mpm_event disabled.
Enabling module mpm_prefork.
apache2_switch_mpm Switch to prefork
apache2_invoke: Enable module php8.2
(省略)
```

有効になっている apache2 の MPM を確認すると prefork になっています。

^{*7} <https://www.php.net/manual/ja/install.unix.apache2.php>

^{*8} <https://www.php.net/manual/ja/install.unix.apache2.php>

^{*9} <https://www.php.net/manual/ja/install.fpm.php>

^{*10} 仕様は元々 fastcgi.com で公開していたが今は閉鎖されているため、コピーサイトの <https://fast-cgi.github.io/> で情報を確認できます。

^{*11} php-fpm で PHP を実行する場合は、apache は event MPM を利用することができます。event MPM は linux では epoll を使ったイベント駆動な実装になっており、少ないプロセス数とスレッド数で多数のクライアントを処理することができます。

```
# cd /etc/apache2/mods-enabled
# ls -l mpm*
lrwxrwxrwx 1 root root 34  4月  9 14:34 mpm_prefork.conf -> ../mods-available/mpm_prefork.conf
lrwxrwxrwx 1 root root 34  4月  9 14:34 mpm_prefork.load -> ../mods-available/mpm_prefork.load
```

もし、MPM が prefork になっていない場合は、a2dismod で mpm_event を無効化し、a2enmod で mpm_prefork を有効化してください。


```
# a2dismod mpm_event
Module mpm_event disabled.
To activate the new configuration, you need to run:
systemctl restart apache2

# a2enmod mpm_prefork
Considering conflict mpm_event for mpm_prefork:
Considering conflict mpm_worker for mpm_prefork:
Enabling module mpm_prefork.
To activate the new configuration, you need to run:
systemctl restart apache2
```

apache2 を再起動します。

```
# systemctl restart apache2
```

Web ブラウザで phpinfo を表示する画面を開きます。一番上部にある表の ”Server API” 行の設定値が ”Apache 2.0 Handler” になっていれば mod_php で動作しています。

PHP Version 8.2.4


| | |
|---|--|
| System | Linux deb12-webapp-test 6.1.0-7-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.20-1 (2023-03-19) x86_64 |
| Build Date | Mar 16 2023 14:24:40 |
| Build System | Linux |
| Server API | Apache 2.0 Handler |
| Virtual Directory Support | disabled |
| Configuration File (php.ini) Path | /etc/php/8.2/apache2 |
| Loaded Configuration File | /etc/php/8.2/apache2/php.ini |
| Scan this dir for additional .ini files | /etc/php/8.2/apache2/conf.d |

図 2 phpinfo 画面 (apache2 + mod_php)

3.3.2 apache2 + php-fpm

apache2、php、php-fpm パッケージをインストールします。php パッケージの依存関係の都合で mod_php である libapache2-mod-php8.2 がインストールされてしまいます。

```
# apt-get install apache2 php php-fpm

(中略)
下のパッケージが新たにインストールされます:
apache2 apache2-bin apache2-data apache2-utils libapache2-mod-php8.2
libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libcurl4
liblua5.3-0 php php-common php-fpm php8.2 php8.2-cli php8.2-common
php8.2-fpm php8.2-opcache php8.2-readline psmisc ssl-cert

(中略)
php8.2-fpm (8.2.4-1) のトリガを処理しています ...
NOTICE: Not enabling PHP 8.2 FPM by default.
NOTICE: To enable PHP 8.2 FPM in Apache2 do:
NOTICE: a2enmod proxy_fcgi setenvif
NOTICE: a2enconf php8.2-fpm
NOTICE: You are seeing this message because you have apache2 package installed.
libapache2-mod-php8.2 (8.2.4-1) のトリガを処理しています ...
```

php-fpm で動作させたい場合は mod_php をインストールしておくことと設定のコンフリクトを起こすため libapache2-mod-php8.2 パッケージを削除しておきます。

```
# apt-get purge libapache2-mod-php8.2
```

パッケージのインストール時に表示された NOTICE に従い、apache2 のモジュールと設定を有効化します。

```
# a2enmod proxy_fcgi setenvif
Considering dependency proxy for proxy_fcgi:
Enabling module proxy.
Enabling module proxy_fcgi.
Module setenvif already enabled
To activate the new configuration, you need to run:
    systemctl restart apache2

# a2enconf php8.2-fpm
Enabling conf php8.2-fpm.
To activate the new configuration, you need to run:
    systemctl reload apache2
```

apache2 の MPM を prefork から event へ切り替えます。

```
# a2dismod mpm_prefork
Module mpm_prefork disabled.
To activate the new configuration, you need to run:
    systemctl restart apache2

# a2enmod mpm_event
Considering conflict mpm_worker for mpm_event:
Considering conflict mpm_prefork for mpm_event:
Enabling module mpm_event.
To activate the new configuration, you need to run:
    systemctl restart apache2
```

php8.2-fpm、apache2 を再起動します。

```
# systemctl restart php8.2-fpm
# systemctl restart apache2
```

Web ブラウザで phpinfo を表示する画面を開きます。一番上部にある表の "Server API" 行の設定値が "FPM/FastCGI" になっていれば php-fpm で動作しています。

3.3.3 nginx + php-fpm

nginx、php-fpm パッケージをインストールします。

```
# apt-get install nginx php-fpm

(中略)
下のパッケージが新たにインストールされます:
nginx nginx-common php-common php-fpm php8.2-cli php8.2-common php8.2-fpm php8.2-opcache php8.2-readline
```

php8.2-fpm、nginx を再起動します。

| | |
|---|--|
| System | Linux deb12-webapp-test 6.1.0-7-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.20-1 (2023-03-19) x86_64 |
| Build Date | Mar 16 2023 14:24:40 |
| Build System | Linux |
| Server API | FPM/FastCGI |
| Virtual Directory Support | disabled |
| Configuration File (php.ini) Path | /etc/php/8.2/fpm |
| Loaded Configuration File | /etc/php/8.2/fpm/php.ini |
| Scan this dir for additional .ini files | /etc/php/8.2/fpm/conf.d |

図 3 phpinfo 画面 (apache2 + php-fpm)

```
# systemctl restart php8.2-fpm
# systemctl restart nginx
```

Web ブラウザで phpinfo を表示する画面を開きます。なぜか phpinfo.php ファイルがサーバで実行されず、ダウンロードされてしまいました。

nginx の設定を確認すると、`/etc/nginx/sites-available/default` の `fastcgi` の中継先となる UNIX ドメインソケットのパスがコメントで無効になっていました。

```
# grep -nr fastcgi_pass /etc/nginx
/etc/nginx/sites-available/default:60: #       fastcgi_pass unix:/run/php/php7.4-fpm.sock;
/etc/nginx/sites-available/default:62: #       fastcgi_pass 127.0.0.1:9000;
```

`/etc/nginx/sites-available/default` を変更します。

```
# vi /etc/nginx/sites-available/default
- #location ~ /\.php$ {
- #       include snippets/fastcgi-php.conf;
- #
- #       # With php-fpm (or other unix sockets):
- #       fastcgi_pass unix:/run/php/php7.4-fpm.sock;
- #       # With php-cgi (or other tcp sockets):
- #       fastcgi_pass 127.0.0.1:9000;
- #}
+ location ~ /\.php$ {
+       include snippets/fastcgi-php.conf;
+
+       # With php-fpm (or other unix sockets):
+       fastcgi_pass unix:/run/php/php8.2-fpm.sock;
+       # With php-cgi (or other tcp sockets):
+       #fastcgi_pass 127.0.0.1:9000;
+ }
```

変更した設定を反映するため、nginx を再起動します。

```
# systemctl restart nginx
```

Web ブラウザで phpinfo を表示する画面を開きます。一番上部にある表の”Server API” 行の設定値が”FPM/FastCGI” になっていれば php-fpm で動作しています。

| | |
|---|--|
| System | Linux deb12-webapp-test 6.1.0-7-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.20-1 (2023-03-19) x86_64 |
| Build Date | Mar 16 2023 14:24:40 |
| Build System | Linux |
| Server API | FPM/FastCGI |
| Virtual Directory Support | disabled |
| Configuration File (php.ini) Path | /etc/php/8.2/fpm |
| Loaded Configuration File | /etc/php/8.2/fpm/php.ini |
| Scan this dir for additional .ini files | /etc/php/8.2/fpm/conf.d |

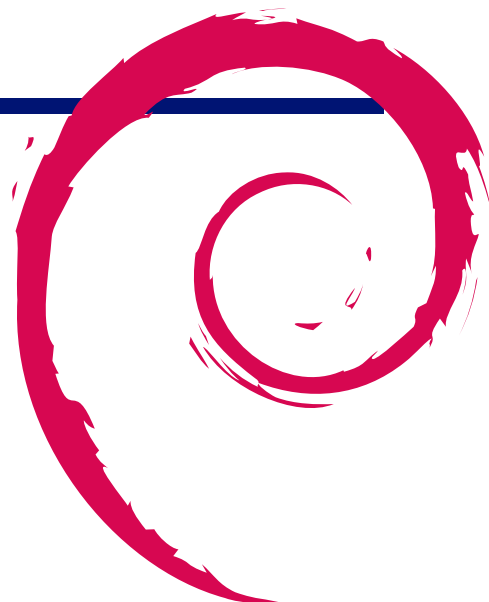
図 4 phpinfo 画面 (nginx + php-fpm)

3.4 終わりに

Debian 12 bookworm で PHP を実行できる Web サーバを構築する設定方法を説明しました。apache2 のモジュールや設定を有効化・無効化する `a2enmod`、`a2dismod`、`a2enconf`、`a2disconf` コマンドや、nginx の `php-fpm` の中継先の設定を正しく行わないと PHP をうまく実行できないため、注意が必要です。

そのほか、apache2 の MPM 設定、nginx の worker 設定、`php-fpm` のプロセス数設定をチューニングするとサーバの応答性能をさらに引き出すことができるため、こだわりがある人は挑戦してみてください。

4 メモ





Debian 勉強会資料

2023年4月15日 初版第1刷発行

東京エリア Debian 勉強会（編集・印刷・発行）
