

月刊

Debian 専

日本唯一のDebian専門月刊誌

2012年3月17日

Debian 勉強会 - ユーザサイド -

Apache2 / HTTP サーバから始める Debian



1 Introduction

上川 純一

今月の Debian 勉強会へようこそ。これから Debian の世界にあしを踏み入れるという方も、すでにどっぷりとつかっているという方も、月に一回 Debian について語りませんか？

Debian 勉強会の目的は下記です。

- Debian Developer (開発者) の育成。
- 日本語での「開発に関する情報」を整理してまとめ、アップデートする。
- 場 の提供。
 - 普段ばらばらな場所にいる人々が face-to-

face で出会える場を提供する。

- Debian のためになることを語る場を提供する。
- Debian について語る場を提供する。

Debian の勉強会ということで究極的には参加者全員が Debian Package をがりがりと作るスーパーハッカーになった姿を妄想しています。情報の共有・活用を通して Debian の今後の能動的な展開への土台として、「場」としての空間を提供するのが目的です。

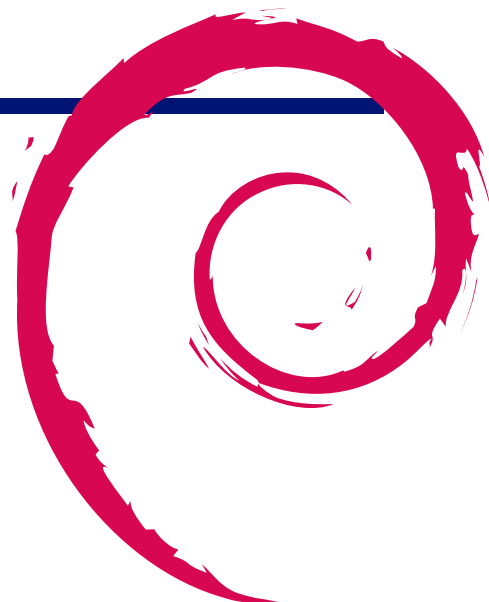
会 勉 強 会 の ア ー キ ブ ト

目次

1	Introduction	1	ド - Apache2 / HTTP サーバ から始める Debian	4
2	最近の Debian 関連のミーテ ィング報告	3	3.1 はじめに	4
2.1	東京エリア Debian 勉強会 85 回目報告	3	3.2 Debian の Apache2 バージョン	4
2.2	福岡 Debian 勉強会 0 回目報告	3	3.3 Debian のパッケージ構成と パッケージのインストール . .	4
3	Debian 勉強会 - ユーザサイ		3.4 Apache2 の設定ファイル . .	7
			3.5 サイトを設定する	7
			3.6 モジュールを有効/無効にする	9
			3.7 その他	10
			3.8 まとめ	10

2 最近の Debian 関連のミーティング報告

岩松 信洋



2.1 東京エリア Debian 勉強会 85 回目報告

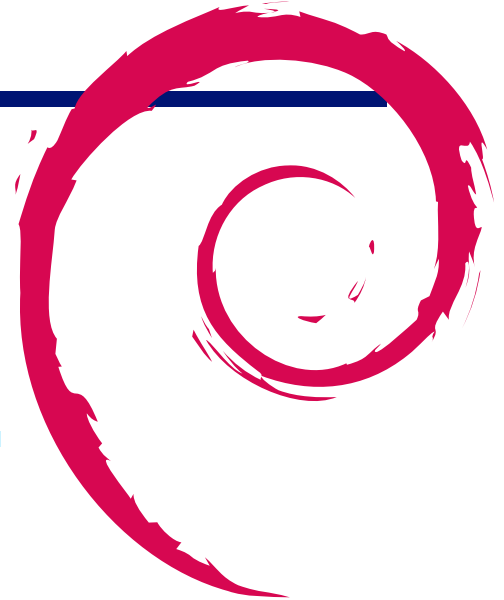
2.2 福岡 Debian 勉強会 0 回目報告

たまたま 福岡に行く機会がありましたので、2012 年 2 月 18 日に 福岡で 第 0 回 福岡 Debian 勉強会を行いました。岩松が DKMS の仕組みと Debian での対応状況、パッケージ作成方法について説明しました。いきなりカーネルの話をするので、参加者はちょっと引いていました。DKMS へサポート体制はほとんど完了しており、ユーザは容易に使えることが分かりました。Debian パッケージ化をサポートするツールもあるので、まだ DKMS をサポートしていないメンテナはサポートしましょう。

また、山田さんが Debian サーバを容易に量産する仕組みについて紹介しました。いかに速く「構築」できるか、楽に「構築」できるか いかに楽に「再現」できるかについて紹介し、Debian にはこれらを満たすツールがたくさんあることが分かりました。ユーザランドイメージ作るなら、debootstrap ではなく、multistrap を使うのがよいそうです。第 1 回は未定ですが、今後も福岡での Debian 勉強会開催が期待されます。

3 Debian 勉強会 - ユーザサイド - Apache2 / HTTP サーバから始める Debian

岩松 信洋



普段はちょっと開発者寄りな話をしている Debian 勉強会ですが、今回は OSC 出張企画として、ユーザー視点の勉強会を開催します。今回はよく使われていると思われる Apache2 / HTTP サーバ に焦点を当ててみます。

3.1 はじめに

Debian は 日本では HTTP サーバとして利用されているように見えませんが、世界では一番採用されている Linux ディストリビューションになったようです。^{*1} この記事によると、利用されている理由は HTTP サーバパッケージの種類が多くある事が理由の一つに挙げられています。Debian を HTTP サーバとして利用している理由を実際に使っている方に聞いてみたところ、理由はこれだけではないことが分かりました。Debian のパッケージングシステム、APT、Apache モジュールパッケージの多さ、Web アプリケーションで採用される P 言語 (Perl, Python, PHP) のサポートなどがあり、一番良い点として挙げられたのは設定ファイルの柔軟性についてでした。

Debian の Apache2 / HTTP サーバ は Red Hat 系 と違い、Debian 特有の構成になっています。これは他のディストリビューションしか知らない人にとっては難しいかもしれませんが。しかし Debian 特有の構成を理解すると、他のディストリビューションとのメリット、デメリットが見えてくると思います。というわけで今回は、Debian の Apache2 / HTTP サーバ (以下、Apache2) について勉強していこうと思います。

3.2 Debian の Apache2 バージョン

まず、Debian で提供されている Apache2 のバージョンを見てみます。表 1 にまとめました。Upstream と比べると少し古いですが、機能的には問題ないでしょう。RHEL、CentOS (バージョン 2.2.15-15) と比べても特にバージョンが古いというわけでもありません。

ディストリビューション	stable	testing	unstable	experimental	upstream
バージョン	2.2.16-6+squeeze6	2.2.22-1	2.2.22-1	-	2.4.1

表 1 Debian ディストリビューションと Apache2 のバージョン

3.3 Debian のパッケージ構成とパッケージのインストール

次に Apache2 のパッケージ構成とインストール方法について説明します。

^{*1} http://w3techs.com/blog/entry/debian_is_now_the_most_popular_linux_distribution_on_web_servers

3.3.1 パッケージ構成

Debian の Apache2 で提供されているパッケージは以下の通りです。HTTP サーバの処理モデルごとにパッケージ (apache2-mpm-worker、apache2-mpm-prefork、apache2-mpm-event、apache2-mpm-itk) が分離されていることがわかります。これにより自分の用途に合わせたパッケージをインストールできます。Red Hat 系は一つのパッケージに纏まっていて、処理モデル毎にサフィックスをつけています (例: httpd.worker)。

パッケージ名	パッケージの説明
apache2	Apache HTTP サーバメタパッケージ
apache2-mpm-worker	スレッドモデル Apache HTTP サーバ
apache2-mpm-prefork	非スレッドモデル Apache HTTP サーバ
apache2-mpm-event	イベントドリブンモデル Apache HTTP サーバ
apache2-mpm-itk	マルチユーザ環境 Apache HTTP サーバ
apache2.2-common	Apache HTTP サーバ 共通ファイル
apache2.2-bin	Apache HTTP サーバの共通バイナリファイル
apache2-utils	ウェブサーバ用ユーティリティプログラム
apache2-suexec	Apache2 mod-suexec 用 基本 suexec プログラム
apache2-suexec-custom	Apache2 mod-suexec 用 設定可能 suexec プログラム
apache2-dbg	Apache HTTP サーバ デバッグシンボルファイル
apache2-prefork-dev	非スレッドモデル Apache HTTP サーバ 開発用ヘッダファイル
apache2-threaded-dev	マルチスレッドモデル Apache HTTP サーバ 開発用ヘッダファイル
apache2-doc	Apache HTTP サーバドキュメント

表 2 Debian で提供される Apache2 パッケージ

次にパッケージの依存関係図を図 3.3.1 に示します。依存関係が複雑なのでユーザは不安になるかもしれませんが。しかし Debian では強力なパッケージ管理ツール APT によって気にする事なくインストールできます。

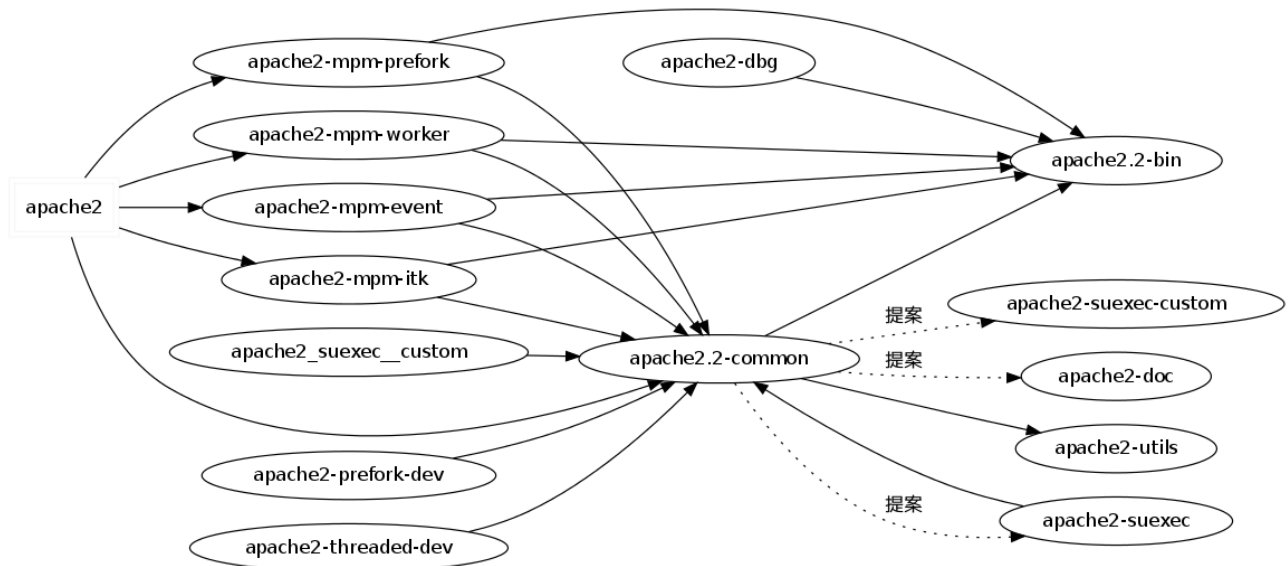


図 1 Debian でのパッケージ依存関係

3.3.2 インストール

Debian で Apache2 をインストールする場合は `apt-get install` コマンドを使います (図 3.3.2)。

Debian では `apache2` というメタパッケージを使ってインストールすることが多いです。 `apache2` をインストールすると、 `apache2-mpm-worker` がインストールされます。他の HTTP サーバパッケージをインストールしたい場合は、各々のパッケージを指定してインストールする必要があります。

また CentOS などでは、「`httpd`」パッケージとして提供されているのでパッケージ名が異なります。普段は他のディストリビューションを使っている人は注意しましょう。

```
$ sudo apt-get update ← リポジトリを更新
$ sudo apt-get install apache2 ← apache2 パッケージをインストール
```

図 2 Debian で Apache2 をインストールする

3.3.3 Apache HTTP サーバの起動と停止

Debian は「インストールしたものは使う」というポリシーなので、インストール完了の時点で既に Apache HTTP サーバは起動しています。停止したい場合には root 権限で「`/etc/init.d/apache2 stop`」を実行します。起動したい場合は「`/etc/init.d/apache2 start`」、再起動したい場合には「`/etc/init.d/apache2 restart`」を実行します。図 3.3.3 に例を示します。

```
$ ps ax | grep apache2 ← apache2 のプロセスを確認
10034 ?      Ss      0:05 /usr/sbin/apache2 -k start
13008 ?      S       0:00 /usr/sbin/apache2 -k start
....
$ sudo /etc/init.d/apache2 stop ← apache2 を停止
$ ps ax | grep apache2 ← apache2 のプロセスを確認
16833 pts/1    S+      0:00 grep apache2
$ sudo /etc/init.d/apache2 start ← apache2 を開始
10048 ?      Ss      0:05 /usr/sbin/apache2 -k start
13024 ?      S       0:00 /usr/sbin/apache2 -k start
....
```

図 3 Apache2 の起動と停止

デフォルトの状態では、マシンを立ち上げ時に HTTP サーバが起動するようになっています。マシン立ち上げ時に HTTP サーバの起動しないようにするには、ランレベル毎のサービス起動スクリプトを制御するツール `update-rc.d` を使います。

全てのランレベルで `apache2` を起動させないようにするには、コマンドにサービス名と `remove` を指定して実行します。

またインストール直後のデフォルトの状態に戻したい場合には、コマンドにサービス名と `default` を指定して実行します。

実行例を図 3.3.3 に示します。

Red Hat 系では `chkconfig` を使いますが、Debian でも提供されています。しかし、`chkconfig` は RedHat 系のサービス管理ツールなので Debian ではうまく動作しないことがあるようです。同様のツールとして `sysv-rc-conf` があるのでこちらを使ったほうがいいでしょう。図 3.3.3 に簡単な使い方を説明します。

```

$ sudo update-rc.d -f apache2 remove ← 全てのランレベルで apache2 を起動させないようにする。
$ sudo update-rc.d -f apache2 default ← サーバ起動をデフォルトの状態に戻す

```

図 4 ランレベルの制御

```

$ sudo apt-get install sysv-rc-conf ← sysv-rc-conf パッケージをインストール
$ sudo sysv-rc-conf --list ← 現在の状態を出力
apache2      0:off1:off2:on3:on4:on5:on6:off
bootlogd     S:on
(中略)
$ sudo sysv-rc-conf --level 2 apache2 off ← ランレベル 2 の apache2 を無効にする
$ sudo sysv-rc-conf --list | head -1 ← 現在の状態を出力
apache2      0:off1:off2:off3:off4:off5:off6:off
$ sudo sysv-rc-conf --level 2 apache2 on ← ランレベル 2 の apache2 を有効にする
$ sudo sysv-rc-conf --list | head -1 ← 現在の状態を出力
apache2      0:off1:off2:on3:off4:off5:off6:off

```

図 5 Apache2 の起動と停止

3.4 Apache2 の設定ファイル

Red Hat 系の場合、主な設定は `/etc/httpd/conf/httpd.conf` で行い、include されるファイルは `/etc/httpd/conf.d/` ディレクトリに格納しますが、Debian の場合は表 3 のようになっています。

設定ファイル	内容
<code>/etc/apache2/apache2.conf</code>	基本設定
<code>/etc/apache2/httpd.conf</code>	オーバーライドする設定
<code>/etc/apache2/conf.d/</code>	基本設定の中で Include するファイルを格納する
<code>/etc/apache2/ports.conf</code>	ポートの設定
<code>/etc/apache2/envvars</code>	環境変数の設定
<code>/etc/apache2/mods-available/</code>	利用可能なモジュール設定
<code>/etc/apache2/mods-enabled/</code>	利用中のモジュール設定
<code>/etc/apache2/sites-available/</code>	利用可能なサイト設定
<code>/etc/apache2/sites-enabled/</code>	利用中のサイト設定
<code>/var/www</code>	ドキュメントルート
<code>/usr/lib/cgi-bin</code>	cgi-bin
<code>/var/log/apache2</code>	Apache2 ログ

表 3 Debian の Apache2 設定ファイル群

`apache2.conf` には図 3.4 のような行があり、`apache2.conf` から各設定が読み込まれるようになっています。Apache2 の設定を変更する場合、`apache2.conf` を変更せず、`httpd.conf` や `ports.conf` を変更します。

3.5 サイトを設定する

Debian は `/etc/apache2/sites-available/default` に `apache2` のデフォルトのサイト設定を格納しています。サイトを一つだけ構築する場合はこのファイルを変更し、`apache2` を再起動すれば設定された内容で `apache2` が立ち上がります。再起動する方法は図 3.5 の通りです。

```

(省略)
# Include module configuration:
Include /etc/apache2/mods-enabled/*.load
Include /etc/apache2/mods-enabled/*.conf

# Include all the user configurations:
Include /etc/apache2/httpd.conf

# Include ports listing
Include /etc/apache2/ports.conf
(中略)
# Include generic snippets of statements
Include /etc/apache2/conf.d/

# Include the virtual host configurations:
Include /etc/apache2/sites-enabled/

```

図 6 apache2.conf の内容

```
$ sudo /etc/init.d/apache2 restart
```

図 7 Apache2 の再起動

Debian の apache2 で複数のサイトを立ち上げる場合、httpd.conf や apache2.conf は編集しません。サイト別に設定を記述し、/etc/apache2/sites-available/ディレクトリに格納します。そして、そのサイトを設定を有効にするコマンド「a2ensite」実行し、apache2 を再起動します。

簡単な手順を説明します。例えば、test.example.org というサイトを立ち上げるとします。内容は図 3.5 のようになるでしょう。

```

<VirtualHost *>
  ServerAdmin admin-test@example.org
  ServerName test.example.org
  DocumentRoot /home/test/public_html/
  <Directory />
    Options FollowSymLinks ExecCGI Includes
    AllowOverride None
  </Directory>
</VirtualHost>

```

図 8 サイトの設定例

そしてこのサイト設定を/etc/apache2/sites-available/test に格納します。格納した後、サイトを有効にする「a2ensite コマンド」に有効にしたいサイトの設定ファイル名を指定して実行します。実行すると/etc/apache2/sites-enabled/ にシンボリックリンクが張られ設定が有効になります。有効にただけでは、稼働している httpd サーバには設定が反映されていないため、httpd サーバを再起動します。

図 3.5 に例を示します。

サイトの設定を無効にする場合には、サイトを有効にする「a2dissite コマンド」に無効にしたいサイトの設定ファイル名を指定して実行します。実行すると/etc/apache2/sites-enabled/ からシンボリックリンクが削除されます。サイト設定を無効にした後は、有効時と同様に httpd サーバを再起動する必要があります図 3.5 に例を示します。

このように Debian ではサイトの設定を分離し、サイト毎に状態を管理することができます。他のディストリ

```

$ ls -l /etc/apache2/sites-enabled/
合計 0
lrwxrwxrwx 1 root root 26 2011-03-20 08:23 000-default -> ../sites-available/default
lrwxrwxrwx 1 root root 30 2011-03-20 08:23 default-ssl.old -> ../sites-available/default-ssl
$ sudo a2ensite test ← test を有効にする
Enabling site test.
Run '/etc/init.d/apache2 reload' to activate new configuration!
$ ls -l /etc/apache2/sites-enabled/
合計 0
lrwxrwxrwx 1 root root 26 2011-03-20 08:23 000-default -> ../sites-available/default
lrwxrwxrwx 1 root root 29 2012-03-10 06:24 test -> ../sites-available/test
lrwxrwxrwx 1 root root 30 2011-03-20 08:23 default-ssl.old -> ../sites-available/default-ssl
$ sudo /etc/init.d/apache2 restart ← Apache2 を再起動

```

図 9 サイトを有効にする

```

$ sudo a2dissite test ← test を無効にする
Site test disabled.
Run '/etc/init.d/apache2 reload' to activate new configuration!
$ ls -l /etc/apache2/sites-enabled/
合計 0
lrwxrwxrwx 1 root root 26 2011-03-20 08:23 000-default -> ../sites-available/default
lrwxrwxrwx 1 root root 30 2011-03-20 08:23 default-ssl.old -> ../sites-available/default-ssl

```

図 10 サイトを無効にする

ビューションでは include 等を使って管理することができますが、ファイル内容を変更する必要があり非常に手間です。Debian はシンボリックリンクを使うことによって Apache2 の設定ファイルを変更せずにサイト設定の有効・無効ができるようになっています。

3.6 モジュールを有効/無効にする

Debian のモジュールに関する設定はモジュール毎の設定ファイルとして mods-available ディレクトリに格納されています。それらのうち、実際に有効にするものがシンボリックリンクとして mods-enabled ディレクトリに張られます。シンボリックリンクは手動で行わず、モジュールを有効にする場合には a2enmod コマンド、無効にする場合には a2dismod コマンドを使います。

図 3.6 に mod_info を有効にする例を、図 3.6 に mod_info を有効にする例を示します。

```

$
sudo a2enmod info

```

図 11 mod_info を有効にする

```

$ sudo a2dismod info

```

図 12 mod_info を無効にする

3.7 その他

その他、注意すべき点をいくつか教えてもらったので紹介します。

3.7.1 libapache2-mod-php5 と apache2-mpm-prefork

Apache2 上で mod-php5 を使いたい場合、apache2-mpm-worker は使えない点に注意してください。これは PHP5 (mod-php5) の制限で、スレッドで動作することができないためです。libapache2-mod-php5 をインストールすると、使いたい場合、apache2-mpm-worker が削除され、apache2-mpm-prefork がインストールされます。PHP ユーザの方は注意しましょう。

3.7.2 再起動確認について

その他、Debian で Apache2 を使う理由として、再起動確認を行うという点があります。例えば glibc が更新されたとき、サーバ系は再起動する必要があるのですが、Red Hat 系では再起動してくれず、管理者が手動で行う必要があります。CentOS を使っている会社ではデーモンを再起動しないとしないアップデートがあったかどうかチェックするツールをわざわざ作って管理していたりするようです。しかし Debian では再起動の確認が行われる（設定によって自動再起動も可能）ので、管理者の手を煩わせません。このような細かいところに気を使ってくれるのも Debian の良い所です。

3.8 まとめ

Debian のパッケージ古いというのは昔の話。セキュリティ対応あるし、早い。Debian の設定ファイルが独自の理由は理由がある。専用のツールもあるので、慣れるとメンテナンスが容易。パッケージによる、細かいところへの気配り。とりあえず、Debian 使え。



Debian 勉強会資料

2012年3月17日 初版第1刷発行

東京エリア Debian 勉強会（編集・印刷・発行）
