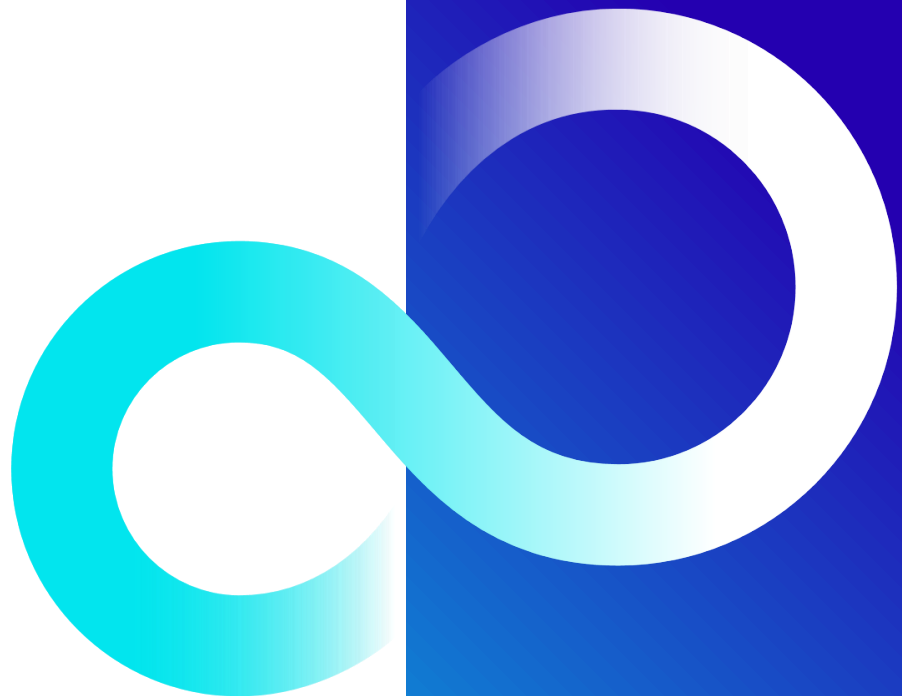


Oracle Solaris  
Webサーバ構築手順書  
-Apacheの利用-



2019年9月

第1.0版

富士通株式会社

## ■ 使用条件

- 著作権・商標権・その他の知的財産権について

コンテンツ(文書・画像・音声等)は、著作権・商標権・その他の知的財産権で保護されています。

本コンテンツは、個人的に使用する範囲でプリントアウトまたはダウンロードできます。ただし、これ以外の利用(ご自分のページへの再利用や他のサーバへのアップロード等)については、当社または権利者の許諾が必要となります。

- 保証の制限

本コンテンツについて、当社は、その正確性、商品性、ご利用目的への適合性等に関して保証するものではなく、そのご利用により生じた損害について、当社は法律上のいかなる責任も負いかねます。本コンテンツは、予告なく変更・廃止されることがあります。

- 輸出または提供

本製品を輸出又は提供する場合は、外国為替及び外国貿易法及び米国輸出管理関連法規等の規制をご確認の上、必要な手続きをお取りください。

## ■ 商標について

- UNIX は、米国およびその他の国におけるオープン・グループの登録商標です。
- SPARC Enterprise、SPARC64 およびすべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している、同社の米国およびその他の国における商標または登録商標です。
- Oracle と Java は、Oracle Corporation およびその子会社、関連会社の米国およびその他の国における登録商標です。
- その他各種製品名は、各社の製品名称、商標または登録商標です。

## はじめに

### 本書の内容

- SPARC/Solaris を使用される方を対象に、Oracle Solaris 11 で提供される Apache の導入手順について記載しています。
- Oracle Solaris 11 の詳細については、以下の URL を参照してください。  
Oracle Solaris 11.4 Infomation Library  
[https://docs.oracle.com/cd/E75431\\_01/](https://docs.oracle.com/cd/E75431_01/)

### 留意事項

- 本書は、Oracle Solaris 11.4 の機能を基に作成しています。
- 本書に記載の設定値(ホスト名、IP アドレスなど)は、参考例です。実際のシステム環境に応じて読み替えてください。
- 本書は、Apache 2.4.39 で環境を構築しています。Apache 2.4.39 よりも前の版数では、設定値および手順が異なる場合がありますので、注意してください。
- 本書は、Oracle Solaris を Solaris と記載することがあります。

### 検証環境

- 本書に記載している操作の実行例には、以下の検証環境を使用しています。

ハードウェア	
サーバ機種	SPARC M12-1
ソフトウェア	
OS	Oracle Solaris 11.4
SRU	SRU19051 (Oracle Solaris 11.4.9.5.0)
ESF (Enhanced Support Facility)	5.2.1
Apache	2.4.39

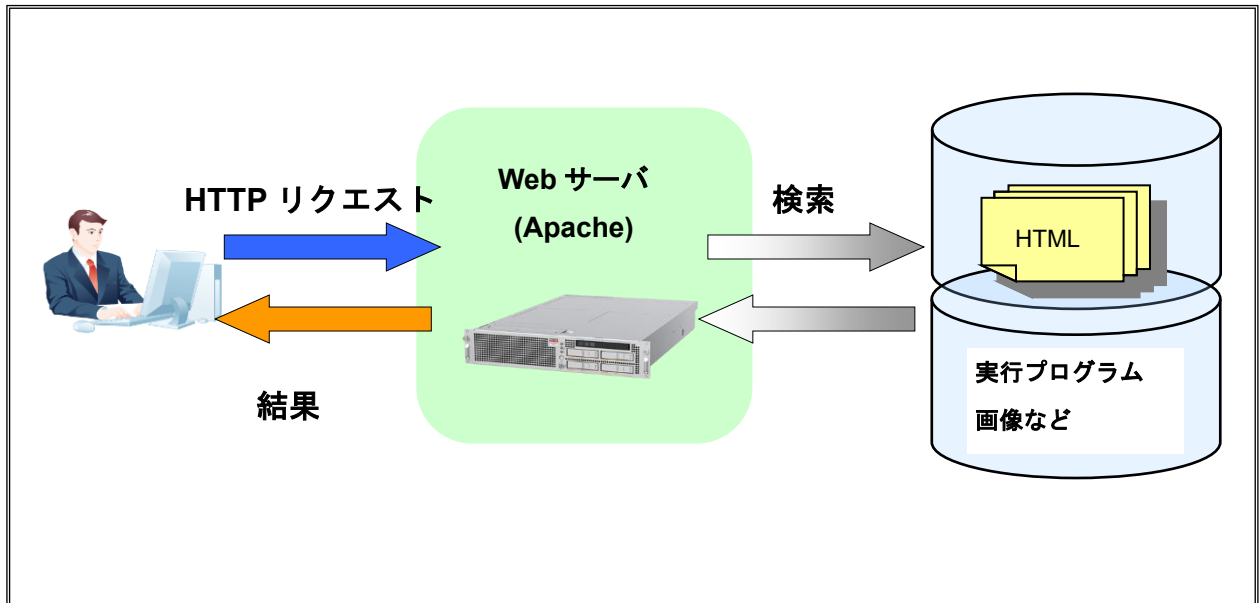
## 目次

1	Apache と Web サーバ.....	1
2	構築前の各種確認.....	2
3	Apache の構築 .....	4
3.1	構成ファイルの編集 .....	4
3.2	サービスの起動 .....	5
3.3	ブラウザからの確認.....	6
4	SMF サービスの管理方法.....	7
5	Apache 環境の詳細設定 .....	8
5.1	構成ファイルの設定 .....	8
5.2	httpd.conf ファイルの内容.....	9
5.3	サービスの再起動.....	14
6	アクセス制御情報ファイルの編集と Basic 認証.....	15
7	SSL/TLS 認証 (Secure Socket Layer/Transport Layer Security) .....	18
7.1	SSL/TLS 環境の設定 .....	18
7.2	ブラウザからの確認.....	22
8	バーチャルホストの設定 .....	23
8.1	NAME ベース.....	23
8.2	IP ベース.....	26
	改版履歴 .....	30

## 1 Apache と Web サーバ

Apache とは、Web サービスを提供するための HTTP(ハイパーテキスト転送プロトコル)サーバソフトウェアです。

Apache を使用して Web サーバを構築することで、HTML 文書や画像などの情報を蓄積しておき、Web ブラウザなどのクライアントからの HTTP リクエストに応じて、これらの情報を送信できます。



## 2 構築前の各種確認

Oracle Solaris 11 をインストールすると、標準で Apache がインストールされます。  
本書では、Apache の構築手順を解説します。

構築前に、本書で使用する環境を確認します。

- Solaris 11(本書では Solaris 11.4)のバージョン
- Solaris 11 の SRU(Support Repository Update)
- Apache のバージョン
- Apache サービスの状態

### 1) Solaris 11 のバージョンを確認します。

```
# cat /etc/release
Oracle Solaris 11.4 SPARC
Copyright (c) 1983, 2019, Oracle and/or its affiliates. All rights reserved.
Assembled 21 May 2019
```

### 2) SRU(Support Repository Update)のバージョンを確認します。

```
# pkg info entire
名前: entire
サマリー: entire incorporation including Support Repository Update
(Oracle Solaris 11.4.9.5.0).
説明: This package constrains system package versions to the same
build. WARNING: Proper system update and correct package
selection depend on the presence of this incorporation.
Removing this package will result in an unsupported system.
For more information see:
https://support.oracle.com/rs?type=doc&id=2433412.1
カテゴリ: Meta Packages/Incorporations
状態: インストール済み
パブリッシャー: solaris
バージョン: 11.4 (Oracle Solaris 11.4.9.5.0)
分岐: 11.4.9.0.1.5.0
パッケージ化の日付: 2019年05月22日 18時01分03秒
最終インストール時間: 2019年06月21日 05時30分18秒
サイズ: 2.52 kB
FMRI: pkg://solaris/entire@11.4-11.4.9.0.1.5.0:20190522T180103Z
```

**3) Apache のバージョンを確認します。**

```
# /usr/apache2/2.4/bin/httpd -v
Server version: Apache/2.4.39 (UNIX)
Server built: Apr 8 2019 05:14:49
```

**4) Apache のサービスの状態を確認します。**

```
# svcs svc:/network/http:apache24
STATE      STIME      FMRI
disabled   14:32:33   svc:/network/http:apache24
```

## 3 Apache の構築

### 3.1 構成ファイルの編集

Apache の構成ファイルは、`/etc/apache2/2.4/`の配下にサンプルが用意されていますので、サンプルファイルをコピーして編集します。

- 1) サンプルファイルのうち、構成ファイル(`httpd.conf`)のバックアップを作成します。

```
# cp /etc/apache2/2.4/httpd.conf /etc/apache2/2.4/httpd.conf_bak
```

- 2) 構成ファイルを編集します。

```
# vi /etc/apache2/2.4/httpd.conf
```

- ▶ 構成ファイルの設定項目(パラメータ)については「[5.2 httpd.conf ファイルの内容](#)」を参照してください。
- ▶ 以降、ファイル記述例の行頭に記述されている数字は、ファイルの行数を表します。vi エディタの「:set nu」コマンドで確認できます。

- i) `[ServerName]`を設定します。

自身のホスト名を指定します。ここでは例として、ホスト名を「`websv.example.co.jp`」とします。

【書式】 `ServerName` ホスト名[:ポート番号]

```
246 # ServerName gives the name and port that the server uses to identify itself.
247 # This can often be determined automatically, but we recommend you specify
248 # it explicitly to prevent problems during startup.
249 #
250 # If your host doesn't have a registered DNS name, enter its IP address here.
251 #
252 ServerName websv.example.co.jp:80
```

- ▶ DNS による名前解決ができない場合、IP アドレスを指定します。
- ▶ ポート番号を省略した場合は、80 番ポートが指定されます。

#### 《注意》

Apache を起動する際に、ローカルリポジトリとポート番号がバッティングするため、サービスが起動できない場合があります。publisher のポート番号を他の空きのポート番号に変更してください。ポート番号の変更手順に関しては、以下を参照してください。

- 『Oracle Solaris 11 を使ってみよう(構築・運用手順書)』  
<https://www.fujitsu.com/jp/sparc-technical/document/solaris/#os>



- ii) [DocumentRoot]を設定します。

HTMLドキュメントを置くトップディレクトリを指定します。

```
272 # DocumentRoot: The directory out of which you will serve your
273 # documents. By default, all requests are taken from this directory, but
274 # symbolic links and aliases may be used to point to other locations.
275 #
276 DocumentRoot "/var/apache2/2.4/htdocs"
277 <Directory "/var/apace2/2.4/htdocs">
```

### 3.2 サービスの起動

- 1) Apache のサービスを起動します。

```
# svcadm enable svc:/network/http:apache24
```

- 2) Apache のサービスを確認します。

```
# svcs svc:/network/http:apache24
STATE      STIME      FMRI
online     15:58:23   svc:/network/http:apache24
```

- 3) ホスト名を設定します。

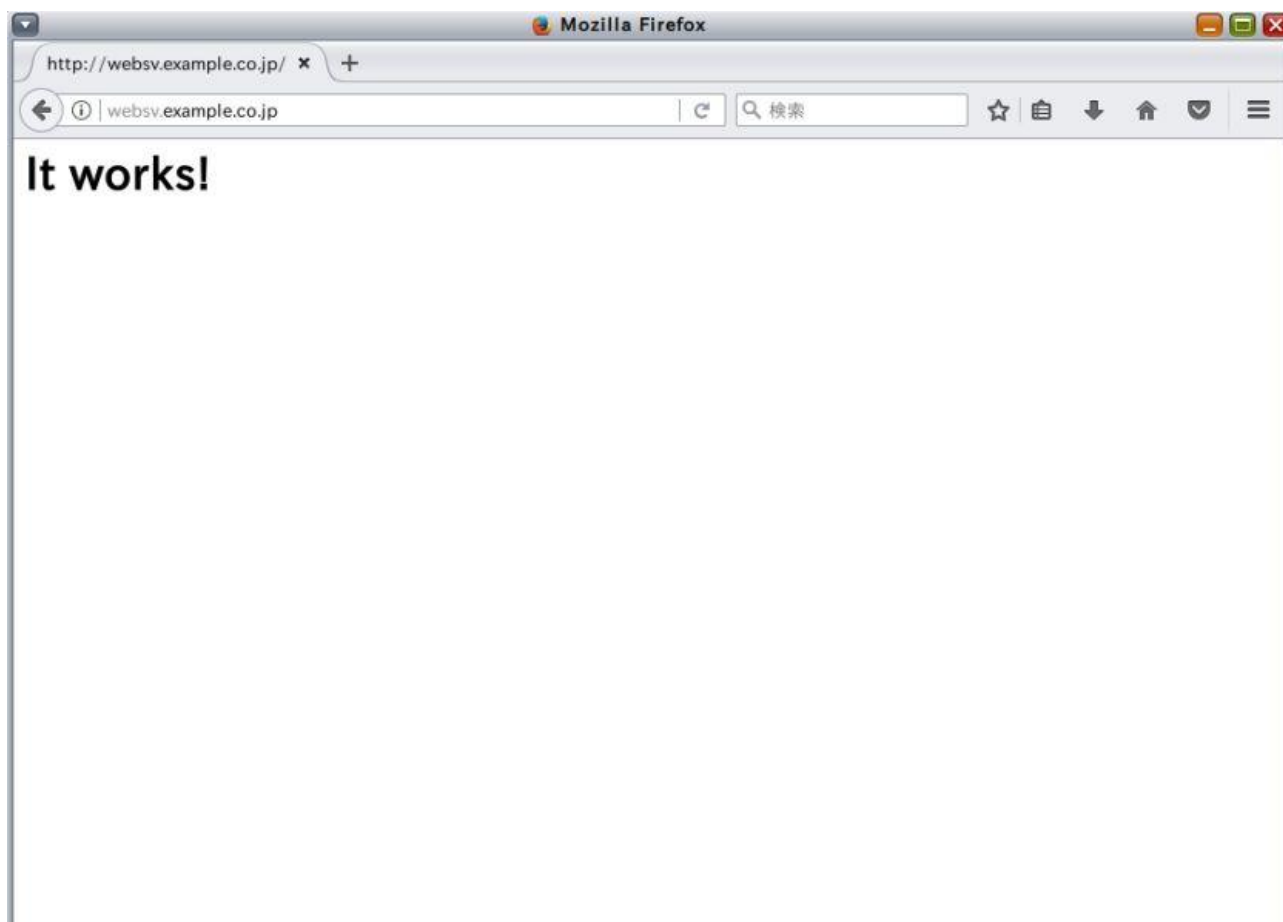
```
# vi /etc/hosts
:
10.20.112.192    websv.example.co.jp
:
```

### 3.3 ブラウザからの確認

Web ブラウザから Web サーバへ接続できることを確認します。以下の URL を入力します。

http://ホスト名<:ポート番号>

※ 「3.1 構成ファイルの編集」手順 2)の [i](#))で IP アドレスを設定した場合は、「ホスト名」に IP アドレスを指定してください



#### Point

DocumentRoot ディレクティブで指定したディレクトリ内の、DirectoryIndex ディレクティブで指定したファイルの内容が表示されます。

ディレクティブについては、「[5.2 httpd.conf ファイルの内容](#)」を参照してください。

## 4 SMF サービスの管理方法

ここでは、SMF 対応した Apache のサービス管理の方法についてまとめています。必要に応じて参照してください。

- サービスの状態確認

```
# svcs svc:/network/http:apache24
STATE      STIME      FMRI
online     16:27:01   svc:/network/http:apache24
```

- ▶ STATE の主な表示は、「online」(起動)と「disable」(停止)です。
- ▶ サービス名は、省略形「apache24」でも指定することができます。以下のコマンドも同様です。

- サービスの起動

```
# svcadm enable svc:/network/http:apache24
```

- サービスの停止

```
# svcadm disable svc:/network/http:apache24
```

- サービスの再起動

```
# svcadm restart svc:/network/http:apache24
```

- サービスの更新

```
# svcadm refresh svc:/network/http:apache24
```

## 5 Apache 環境の詳細設定

### 5.1 構成ファイルの設定

構成ファイル (/etc/apache2/2.4/httpd.conf) を編集することで、Web サーバの動作環境の詳細設定が可能です。

以下では、構成ファイルについて解説します。

ディレクティブ名	内容
ServerRoot	サーバのベースディレクトリ指定
IfModule	モジュールが存在するときに処理されるディレクティブを指定
Listen	リクエストを受け付ける IP アドレスやポート番号の設定
User/Group	httpd プロセスのユーザー名とグループ名の設定
ServerAdmin	通知先メールアドレスの設定
ServerName	自分自身を示すホスト名とポート番号の設定
DocumentRoot	HTML ドキュメントの格納先の設定
Directory	特定のディレクトリ配下に適用されるディレクティブを設定
AllowOverride	アクセス制御情報ファイルで利用できるディレクティブの指定
Options	ディレクトリに対して使用可能な機能を設定
DirectoryIndex	インデックス要求のデフォルトページの設定
ErrorLog	エラーログの格納先ファイルの設定
LogLevel	エラーログのレベルの設定
LogFormat	アクセスログのフォーマット指定
DefaultType	デフォルトの MIME タイプの設定
CustomLog	ログの格納ファイルと出力フォーマットの設定
ScriptAlias	CGI スクリプトのパス設定
RequestHeader	ヘッダカスタマイズ機能の設定
TypesConfig	MIME タイプ設定ファイルの位置を指定
IncludeOptional	サーバの設定ファイルから、取り込む他の設定ファイルの指定

## 5.2 httpd.conf ファイルの内容

ここでは、`/etc/apache2/2.4/httpd.conf` の内容をディレクティブごとに解説します。

### ● [ServerRoot]

インストールされたサーバのベースディレクトリを指定します。

```
31 # Do not add a slash at the end of the directory path.  If you point
32 # ServerRoot at a non-local disk, be sure to specify a local disk on the
33 # Mutex directive, if file-based mutexes are used.  If you wish to share the
34 # same ServerRoot for multiple httpd daemons, you will need to change at
35 # least PidFile.
36 #
37 ServerRoot "/usr/apache2/2.4"
```

▶ ディレクトリパスの最後に「/」(スラッシュ)を付加しないでください。

### ● [IfModule]

<IfModule>から</IfModule>に囲まれるセクションは、モジュールが存在するときに処理されるディレクティブを指定するために利用します。通常の運用では、変更する必要はありません。

### ● [Listen]

Apache が外部からリクエストを受け付けるポート番号を指定します。

特定のインターフェースからのみリクエストを受け付ける場合は、IP アドレスも指定してください。

```
50 # Listen: Allows you to bind Apache to specific IP addresses and/or
51 # ports, instead of the default.  See also the <VirtualHost>
52 # directive.
53 #
54 # Change this to Listen on specific IP addresses as shown below to
55 # prevent Apache from glomming onto all bound IP addresses (0.0.0.0)
56 #
57 #Listen 12.34.56.78:80
58 Listen 80
```

▶ 通常、ポート番号 80 を http プロトコルで使います。

### ● [User] と[Group]

http サーバプロセスの所有者となるユーザー名とグループ名を指定します。

```
214 # If you wish httpd to run as a different user or group, you must run
215 # httpd as root initially and it will switch.
216 #
217 # User/Group: The name (or #number) of the user/group to run httpd as.
218 # It is usually good practice to create a dedicated user and group for
219 # running httpd, as with most system services.
220 #
221 User webservd
222 Group webservd
```

- **[ServerAdmin]**

http サーバに障害が発生した場合の通知先メールアドレスを指定します。

```
239 # ServerAdmin: Your address, where problems with the server should be
240 # e-mailed. This address appears on some server-generated pages, such
241 # as error documents. e.g. admin@your-domain.com
242 #
243 ServerAdmin you@yourhost.com
```

- **[ServerName]**

サーバが自分自身を示すときに使うホスト名とポートを指定します。IP アドレスも指定できます。

```
246 # ServerName gives the name and port that the server uses to identify itself.
247 # This can often be determined automatically, but we recommend you specify
248 # it explicitly to prevent problems during startup.
249 #
250 # If your host doesn't have a registered DNS name, enter its IP address here.
251 #
252 ServerName 127.0.0.1
```

- **[DocumentRoot]**

HTMLドキュメントを置くためのトップディレクトリを指定します。

```
272 # DocumentRoot: The directory out of which you will serve your
273 # documents. By default, all requests are taken from this directory, but
274 # symbolic links and aliases may be used to point to other locations.
275 #
276 DocumentRoot "/var/apache2/2.4/htdocs"
```

- ▶ ディレクトリパスの最後に「/」(スラッシュ)を付加しないでください。
- ▶ ディレクトリを変更した場合、<directory> でアクセス許可を設定してください。

- **[Directory]**

特定のディレクトリと、そのサブディレクトリだけに適用するディレクティブを設定します。

- **[AllowOverride]**

アクセス制御情報ファイル(.htaccess)で使用できるディレクティブを指定します。

本ディレクティブに None を指定した場合、対象ディレクトリ配下でアクセス制御情報ファイルは無効になります。

```
293 # AllowOverride controls what directives may be placed in .htaccess files.
294 # It can be "All", "None", or any combination of the keywords:
295 # AllowOverride FileInfo AuthConfig Limit
296 #
297 AllowOverride None
```

- **[Options]**

ディレクトリに対して使用可能な機能を設定します。

- **[DirectoryIndex]**

クライアントが HTML ファイルを指定しないでアクセスした場合、ここで指定したファイルにアクセスします。複数指定する場合は、スペースで区切ります。優先順位は左からです。

```
306 # DirectoryIndex: sets the file that Apache will serve if a directory
307 # is requested.
308 #
309 <IfModule dir_module>
310     DirectoryIndex index.html
311 </IfModule>
```

- **[ErrorLog]**

エラーログの格納先ファイルを指定します。

```
322 # ErrorLog: The location of the error log file.
323 # If you do not specify an ErrorLog directive within a <VirtualHost>
324 # container, error messages relating to that virtual host will be
325 # logged here. If you *do* define an error logfile for a <VirtualHost>
326 # container, that host's errors will be logged there and not here.
327 #
328 ErrorLog /var/apache2/2.4/logs/error_log
```

- **[LogLevel]**

エラーログのレベルを指定します。

```
331 # LogLevel: Control the number of messages logged to the error_log.
332 # Possible values include: debug, info, notice, warn, error, crit,
333 # alert, emerg.
334 #
335 LogLevel warn
```

レベル	説明	備考
emerg	緊急 - システムが利用できない	
alert	直ちに対処が必要	
crit	致命的な状態	
error	エラー	
warn	警告	デフォルトレベル
notice	普通だが、重要な情報	
info	追加情報	
debug	デバッグメッセージ	

- **[LogFormat]**

アクセスログのフォーマットと、フォーマットに対するニックネームを指定します。

ニックネームは、CustomLog ディレクティブの引数として使用できます。

```
339 # The following directives define some format nicknames for use with
340 # a CustomLog directive (see below).
341 #
342 LogFormat "%h %l %u %t %r" %>s %b "%{Referer}i" "%{User-Agent}i" combined
343 LogFormat "%h %l %u %t %r" %>s %b "common"
```

フォーマットに指定できる主な "%" ディレクティブを、以下に示します。

設定値	説明
%h	リモートホストの情報
%l	クライアントから報告される識別情報(条件を満たすとき取得可能)
%u	ローカルで識別されたユーザー名(HTTP 基本認証のときに取得可能)
%t	時刻
%r	リクエストの内容(先頭のみ)
%s	サーバから返されるステータス(リダイレクト先のステータスを表示する場合には %>s)
%b	リクエストに対して送信されたバイト数

- **[CustomLog]**

アクセスログの格納先ファイルと出力フォーマット(または、LogFormat ディレクティブで設定したフォーマットに対するニックネーム)を指定します。

```
351 # The location and format of the access logfile (Common Logfile Format).
352 # If you do not define any access logfiles within a <VirtualHost>
353 # container, they will be logged here. Contrariwise, if you *do*
354 # define per-<VirtualHost> access logfiles, transactions will be
355 # logged therein and *not* in this file.
356 #
357 CustomLog "/var/apache2/2.4/logs/access_log" common
```

- **[ScriptAlias]**

CGI スクリプトのパスを指定します。CGI ファイルに対して要求があった場合には、ファイルの内容を返す代わりに要求があった CGI サーバ側で実行し、その結果をブラウザに返します。通常、このディレクトリには、CGI などの実行形式ファイルを配置します。

```
386 # ScriptAlias: This controls which directories contain server scripts.
387 # ScriptAliases are essentially the same as Aliases, except that
388 # documents in the realname directory are treated as applications and
389 # run by the server when requested rather than as documents sent to the
390 # client. The same rules about trailing "/" apply to ScriptAlias
391 # directives as to Alias.
392 #
393 ScriptAlias /cgi-bin/ "/var/apache2/2.4/cgi-bin/"
```



- **[RequestHeader]**

HTTP のリクエストヘッダと応答ヘッダを制御し、ヘッダの追加、書き換え、および削除などを行います。

設定値	説明
set	ヘッダを設定します。すでにヘッダが存在する場合は、置き換えます。
append	存在するヘッダに値を追加します。存在するヘッダ値との間は、(コンマ)で区切られます。
add	ヘッダがあっても、別の行にヘッダを設定します。同じヘッダを複数行設定する場合に使用します。
unset	指定した名前のヘッダが存在する場合、すべて削除します。

- **[TypesConfig]**

MIME タイプ設定ファイルの位置を設定します。

```
426 # TypesConfig points to the file containing the list of mappings from
427 # filename extension to MIME-type.
428 #
429 TypesConfig /etc/apache2/2.4/mime.types
```

- **[IncludeOptional]**

サーバの設定ファイルから、他の設定ファイルを取り込む際に使用します。

```
524 # Sample configuration files are in the
525 # /etc/apache2/2.4/samples-conf.d directory. To use a
526 # sample .conf file, copy it to /etc/apache2/2.4/conf.d
527 # directory and modify as required.
528 #
529 IncludeOptional /etc/apache2/2.4/conf.d/*.conf
```

httpd.conf のいくつかの設定が、/etc/apache2/2.4/samples-conf.d/ディレクトリ配下の別の\*.conf ファイルに分離しています。

IncludeOptional で設定したディレクトリ(上記の場合は/etc/apache2/2.4/conf.d/)に、\*.conf ファイルをコピーして使用します。

レベル	説明
autoindex.conf	ブラウザでのディレクトリ表示に関する設定ファイル
dav.conf	WebDAV の設定ファイル
default.conf	Apache サーバに関する基本設定ファイル
info.conf	Apache サーバの動作状況や環境を web 上で表示する設定ファイル
mpm.conf	PidFile, LockFile, 同時接続数の設定ファイル
languages.conf	文字コードの設定ファイル
manual.conf	ローカルに所有している Apache マニュアル公開の設定ファイル
multilang-errordoc.conf	エラー時に返すメッセージの設定ファイル
proxy_scgi.conf	プロキシサーバ設定時に使用するファイル
ssl.conf	SSL 対応の設定ファイル

レベル	説明
userdir.conf	ユーザーディレクトリを「http://aaa.com/~ユーザー名」で公開する設定ファイル
vhosts.conf	バーチャルホストの設定ファイル。詳細は「 <a href="#">8 バーチャルホストの設定</a> 」を参照してください。

### 5.3 サービスの再起動

- 1) 構成ファイルを変更した場合、サービスを再起動して変更を反映させます。

```
# svcadm restart svc:/network/http:apache24
```

- 2) サービスの状態を確認します。

STIME の値が更新されることを確認します。

```
# svcs svc:/network/http:apache24
STATE      STIME      FMRI
online     10:22:44   svc:/network/http:apache24
```

## 6 アクセス制御情報ファイルの編集と Basic 認証

Basic 認証とは、Web サイトの閲覧に使用する http プロトコルが備える、最も基本的なユーザー認証方式です。アクセスの制限された Web ページにアクセスしようとする、Web ブラウザでユーザー名とパスワードの入力を求め、サーバでアクセスを許可しているユーザーに一致すると、ページを閲覧することができる仕組みです。

Basic 認証は httpd.conf ファイルの <Directory> セクションに必要な設定情報を記載します。特定のディレクトリ配下のみ設定することもできます。

従来はアクセス制御情報ファイル(.htaccess)を用いた認証が主流でしたが、Apache チュートリアルではサーバの性能やセキュリティを考え、httpd.conf ファイルで設定する方法を推奨しています。

### 1) パスワードファイルを作成します。

```
# mkdir -p /usr/local/apache/passwd
# touch /usr/local/apache/passwd/passwords
```

### 2) .htpasswd ファイルに secret ユーザーを登録します。

**[書式]** /usr/apache2/2.4/bin/htpasswd [-c] <パスワードファイル名> <登録するユーザー名>

```
# /usr/apache2/2.4/bin/htpasswd -c /usr/local/apache/passwd/passwords secret
New password:
Re-type new password:
Adding password for user secret
```

▶ 初回作成時のみ、「-c」オプションを付けて実行します。

### 3) guest ユーザーを追加登録します。

```
# /usr/apache2/2.4/bin/htpasswd /usr/local/apache/passwd/passwords guest
New password:
Re-type new password:
Adding password for user guest
```

### 4) パスワードファイルを確認します。

```
# ls -la /usr/local/apache/passwd/passwords
-rw-r--r-- 1 root root 89 6月 25日 14:07 /usr/local/apache/passwd/passwords

# cat /usr/local/apache/passwd/passwords
secret:$apr1$527AL2xa$PPf0coNRJgMNUsHxY4FLv/
guest:$apr1$EOAy3txF$SPwjvhAn1vyuBuZKbXFeE1
```

▶ パスワードは暗号化されて表示されます。

## 5) グループ単位での認証の設定をします。

グループファイルを作成し、グループファイルに以下の書式で記載します。

【書式】 [グループ名]: [ユーザー名 1] [ユーザー名 2] ...

```
# vi /usr/local/apache/passwd/group
test: secret guest
```

## 6) httpd.conf ファイルを設定します。

以下の設定を追記します。

```
# vi /etc/apache2/2.4/httpd.conf
:
<Directory "/var/apache2/2.4/htdocs">
:
AuthType Basic
AuthName "Please enter your ID & password"
AuthUserFile /usr/local/apache/passwd/passwords
AuthGroupFile /usr/local/apache/passwd/group
Require group test
</Directory>
```

- **AuthType**  
ユーザー認証機構の種別を指定します。type には、Basic (基本認証) と Digest (MD5 ダイジェスト認証) の設定が可能です。上記の例では、Basic 認証を設定しています。
- **AuthName**  
ユーザー認証の際に、クライアント側に表示させる文字列を設定します。上記の例では、"Please enter your ID & password" の部分です。任意に設定することが可能です。
- **AuthUserFile**  
ユーザー認証のための、ユーザーとパスワードの一覧ファイル (上記の例では、/usr/local/apache/passwd/passwords) を指定します。ファイル名を絶対パスで指定しない場合、ServerRoot ディレクティブで指定したディレクトリからの相対パスとして扱われます。Apache チュートリアルでは、他の人がパスワードファイルをダウンロードできないように、ウェブからアクセスできない場所にパスワードファイルを置くことを推奨しています。
- **AuthGroupFile**  
ユーザー認証のための、グループファイル (上記の例では、/usr/local/apache/passwd/group) を指定します。Require ディレクティブで、group を宣言したときに必要です。ファイル名を絶対パスで指定しない場合、ServerRoot ディレクティブで指定したディレクトリからの相対パスとして扱われます。

- Require

サーバのディレクトリに接続するクライアントの接続を許可、または拒否する条件を設定します。

設定値	説明	備考
all granted	すべてのクライアントからの接続を許可します	デフォルト値
all denied	すべてのクライアントからの接続を拒否します	
not [ip.address]	特定のホストやアドレスからの接続を拒否します	
ip [ip]	特定の IP アドレスからの接続を許可します	
group [グループ名]	特定のグループからの通信を許可します	
user [ユーザー名]	特定のユーザーからの通信を許可します	
valid-user	すべてのユーザーからの通信を許可します	
local	ローカルホストからの接続を許可します	

### 7) ブラウザから接続確認をします。

Web ブラウザから Web サーバへ接続し、認証ダイアログが表示されることを確認します。

接続方法は、「[3.3 ブラウザからの確認](#)」と同じです。



ユーザー名とパスワードを入力し、「OK」ボタンをクリックします。

### 8) 作成したユーザーでアクセスできることを確認してください。

## 7 SSL/TLS 認証 (Secure Socket Layer/Transport Layer Security)

SSL/TLS は、HTTP や FTP など送受信するネットワーク上のデータを暗号化し、安全な通信を行うためのプロトコルです。

### 7.1 SSL/TLS 環境の設定

Solaris 11 の標準機能である openssl を使用し、SSL/TLS 環境を設定する手順を説明します。本書では、自己署名の手順を解説します。

#### 1) 環境変数 PATH を設定します。

```
# PATH=$PATH:/usr/sfw/bin
# export PATH
```

#### 2) Apache が読み込むサーバ証明書や秘密鍵を格納するためのディレクトリを作成します。

```
# mkdir /etc/apache2/2.4/ssl.crt
# mkdir /etc/apache2/2.4/ssl.key
```

#### 3) SSL/TLS の環境定義ファイルをコピーします。

```
# cp /etc/apache2/2.4/samples-conf.d/ssl.conf /etc/apache2/2.4/conf.d/.
```

#### 4) パスフレーズを作成します。

パスフレーズは任意の文字列で、4 文字以上で設定してください。

```
# openssl genrsa -des3 -out /etc/apache2/2.4/ssl.key/server.key 1024
Generating RSA private key, 1024 bit long modulus
.....+++++
.....+++++
e is 65537 (0x10001)
Enter pass phrase for /etc/apache2/2.4/ssl.key/server.key:*****
Verifying - Enter pass phrase for /etc/apache2/2.4/ssl.key/server.key:*****
```

## 5) CSR を発行します。

```
# openssl req -new -x509 -days 365 -key /etc/apache2/2.4/ssl.key/server.key -out
/etc/apache2/2.4/ssl.crt/server.crt
Enter pass phrase for /etc/apache2/ssl.key/server.key:***** ←手順 4) で設定した
                                     パスフレーズを入力

You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) []:JP                                     ←国コードを入力
State or Province Name (full name) []:TOKYO                         ←都道府県名を入力
Locality Name (eg, city) []:SHINAGAWA                             ←都市名を入力
Organization Name (eg, company) []:FUJITSU                         ←組織名を入力
Organizational Unit Name (eg, section) []:INFRA                    ←部署名を入力
Common Name (eg, YOUR name) []:websv.example.co.jp                ←ホスト名を入力
Email Address []:                                                  ←メールアドレスを入力 (任意)
```

## 《注意》

パスフレーズ以外の項目をすべて空白にした場合、エラーが発生します。  
また、日本語は使用せず、半角英数字で入力してください。

## 6) RSA パラメータ生成機能を使用し、公開鍵を作成します。

```
# openssl rsa -in /etc/apache2/2.4/ssl.key/server.key -out
/etc/apache2/2.4/ssl.key/server.key
Enter pass phrase for /etc/apache2/2.4/ssl.key/server.key:*****
writing RSA key
```

## 7) Apache 用認証鍵を確認します。

```
# ls -l /etc/apache2/2.4/ssl.key/server.key
-rw-r--r-- 1 root root 887 6月 25日 14:53 /etc/apache2/2.4/ssl.key/server.key
# cat /etc/apache2/2.4/ssl.key/server.key
-----BEGIN RSA PRIVATE KEY-----
MIICXgIBAAKBgQCkbDKH9LTkVyAyVUnweAH3tzCNwYvUWseiUcuudAvZBXk3XZLf
Bd3uzsElyun9rNW53qAHm7AAGhu6uu0N2iV6v2UqCJ2xA/+QqBoqf/uvP+7LIP5
JlnSv+jckqFNI8idFglFQiy/JHutqQpm9PS0jfkX/ptVI7gNvtQS00yL4wIDAQAB
AoGAPT8AY5xHdXbuy1Dxxd0t/nQCZgteZq+aBWCrmkGn1KMxT+pPNzwc0ORQq1HK
b2/d/OpseekIWKfFatCONIK5yrIgstLEXevLh5zRIUE+APx1TCL5VDFvZRmh0Bi2
NwFkUzxY00G8pf2DdEvWEbhHRQP8uw5fw13CrP9u1KEh6CECQQDagAGypT5HAIVm
QPPvDsPpD3CenCmfZundv0E9sFA9tpp+D/J4IIGg6jvzZjxQJjH9VIYFnzW8KFZE
s9ZuZpDVakEAwKRAvH5rGi4T6JvWWi3XM9i2BMRQ1V8/MuKoMGxwJLBvkFBizCVD
1QNk4Z5R1tdtEI/GVM79DAe9K0CPcIHF1wJBAJTSp1Vkk5yODFjPWNao+d9eMtj1
hd65PAVdZTZ9Rsoh++xtNDt5+JD/ucG5Gs+EIHv1/CsMG++Cke/NhfpARdkCQQCB
WMguyCNZvL6ZXKmce1MX+NaedMJhwkZJAG81tFoN3RnLqG3t7NkIAfNmmksRGauy
hxcAMsssnLSiEcW+ISSZAKEAorenm+8PAAu3bUI7FLEhgYSDomfspXG/Tw1jT7pP
FzoPz6kRDwrz2TskCuSRyeZjDk2la0LV1CDzEEHW4WZc6Q==
-----END RSA PRIVATE KEY-----
```

## 8) ssl.conf ファイルを編集します。

```
# vi /etc/apache2/2.4/conf.d/ssl.conf
:
# General setup for the virtual host
DocumentRoot "/var/apache2/2.4/htdocs"
ServerName webserv.example.co.jp:443
ServerAdmin webserv@example.co.jp
ErrorLog /var/apache2/2.4/logs/error_log
TransferLog /var/apache2/2.4/logs/access_log
:
:
# Server Certificate:
:
SSLCertificateFile "/etc/apache2/2.4/ssl.crt/server.crt"
:
# Server Private Key:
:
SSLCertificateKeyFile "/etc/apache2/2.4/ssl.key/server.key"
:
```

←[ホスト名]: 443 を設定  
←管理者メールアドレスを設定

←ディレクトリを修正

←ディレクトリを修正



**9) Apache のサービス構成 httpd/ssl プロパティを true に設定します。**

```
# svccfg -s apache24
svc:/network/http:apache24> setprop httpd/ssl = boolean: true
svc:/network/http:apache24> listprop httpd/ssl
httpd/ssl    boolean    true
svc:/network/http:apache24> end
```

**10) SSL/TLS モジュールを適用します。**

Apache2.4 から SSL/TLS モジュールはデフォルトでコメントアウトされているため、行頭の#を削除してコメントアウトを解除します。

```
# vi /etc/apache2/2.4/httpd.conf
:
#LoadModule ssl_module libexec/mod_ssl.so
:
#LoadModule socache_shmcb_module libexec/mod_socache_redis.so
```

**11) サービス構成を再読み込みします。**

```
# svcadm refresh svc:/network/http:apache24
```

**12) サービスを再起動します。**

```
# svcadm restart svc:/network/http:apache24
```

**13) サービスの状態を確認します。**

```
# svcs apache24
STATE      STIME      FMRI
online     16:32:59  svc:/network/http:apache24
```

**14) Apache の SSL/TLS 受付を確認します。**

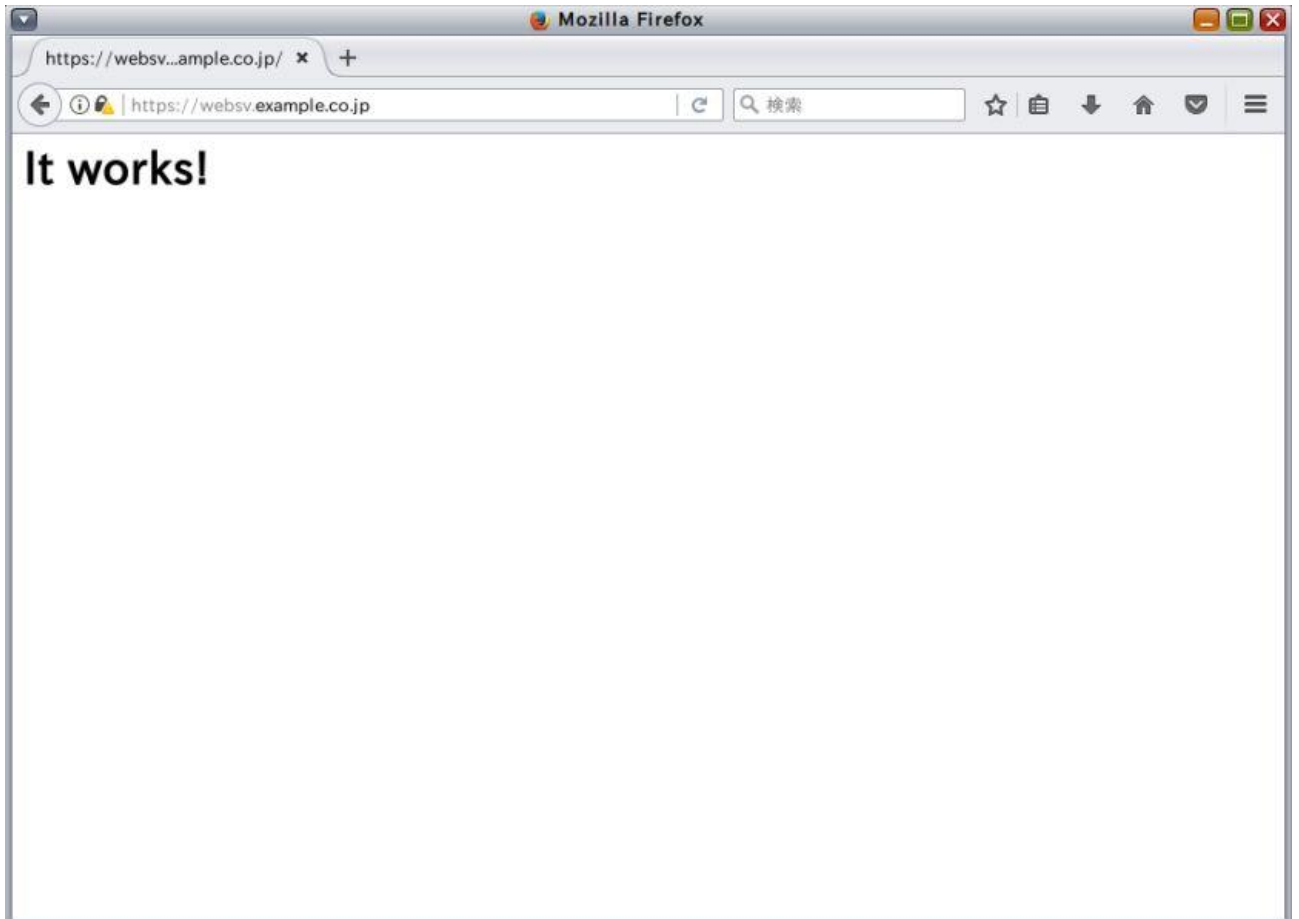
```
# openssl s_client -connect localhost:443 -brief
depth=0 C = JP, ST = TOKYO, L = SHINAGAWA, O = FUJITSU, OU = INFRA, CN =
websv.example.co.jp, emailAddress =
verify error:num=18:self signed certificate
CONNECTION ESTABLISHED
Protocol version: TLSv1.2
Cipher suite: ECDHE-RSA-AES256-GCM-SHA384
Peer certificate: C = JP, ST = TOKYO, L = SHINAGAWA, O = FUJITSU, OU = INFRA, CN =
websv.example.co.jp, emailAddress =
Hash used: SHA512
Supported Elliptic Curve Point Formats:
uncompressed:ansiX962_compressed_prime:ansiX962_compressed_char2
Server Temp Key: ECDH, P-256, 256 bits
```

## 7.2 ブラウザからの確認

Web ブラウザから Web サーバへ、SSL/TLS を使用して接続できることを確認します。

以下の URL を入力します。

<https://ホスト名<:ポート番号>>



## 8 バーチャルホストの設定

バーチャルホスト(Virtualhost)とは、1 台のサーバ上に複数の Web サイトを構成する機能です。バーチャルホストの運用方法には、NAME ベースと IP ベースの 2 種類の方法があります。両方式とも、DNS サーバの設定が必要です。

- NAME ベース  
NAME ベースのバーチャルホストは、Web ブラウザが Web サーバに対して送るホスト名を基にし、応答するホストを決定する方式です。1 つの IP アドレスで、複数の Web サイトを構成することが可能です。
- IP ベース  
IP ベースのバーチャルホストは、IP アドレスでホストを区別する方式です。1 つのサーバ上に複数の IP アドレスを設定し、複数の Web サイトを構成することが可能です。

### 8.1 NAME ベース

設定例)

	ドメイン 1	ドメイン 2
ドメイン名	example.co.jp	sparc.co.jp
IP アドレス	10.20.112.192	10.20.112.192
DocumentRoot	/var/apache2/2.4/htdocs	/var/apache2/2.4/htdocs/dom2
ServerAdmin	websv@example.co.jp	websv2@sparc.co.jp

- 1) バーチャルホストの環境定義ファイルをコピーします。

```
# cp /etc/apache2/2.4/samples-conf.d/vhosts.conf /etc/apache2/2.4/conf.d/.
```

## 2) vhosts.conf ファイルを編集します。

サーバ名 (ServerName) とドキュメントルート (DocumentRoot) を編集します。

```
# vi /etc/apache2/2.4/conf.d/vhosts.conf
:
# Use name-based virtual hosting.
#
NameVirtualHost 10.20.112.192:80      ←ホスト名[または IP アドレス]:ポート番号を指定

#
# VirtualHost example:
# Almost any Apache directive may go into a VirtualHost container.
# The first VirtualHost section is used for all requests that do not
# match a ServerName or ServerAlias in any <VirtualHost> block.
#
<VirtualHost 10.20.112.192:80>
    ServerAdmin websv@example.co.jp    ←ドメイン 1 の管理者メールアドレスを指定
    ServerName websv.example.co.jp     ←ドメイン 1 のホスト名を指定
    DocumentRoot /var/apache2/2.4/htdocs ←ドメイン 1 のドキュメントルートを指定
</VirtualHost>
<VirtualHost 10.20.112.192:80>
    ServerAdmin websv2@sparc.co.jp     ←ドメイン 2 の管理者メールアドレスを指定
    ServerName websv2.sparc.co.jp      ←ドメイン 2 のホスト名 (別名) を指定
    DocumentRoot /var/apache2/2.4/htdocs/dom2 ←ドメイン 2 のドキュメントルートを指定
</VirtualHost>
```

## 3) Apache のサービスを再起動します。

```
# svcadm restart svc:/network/http:apache24
```

## 4) サービスの状態を確認します。

```
# svcs svc:/network/http:apache24
STATE      STIME      FMRI
online     15:00:03   svc:/network/http:apache24
```

## 5) DNS サーバを設定します。

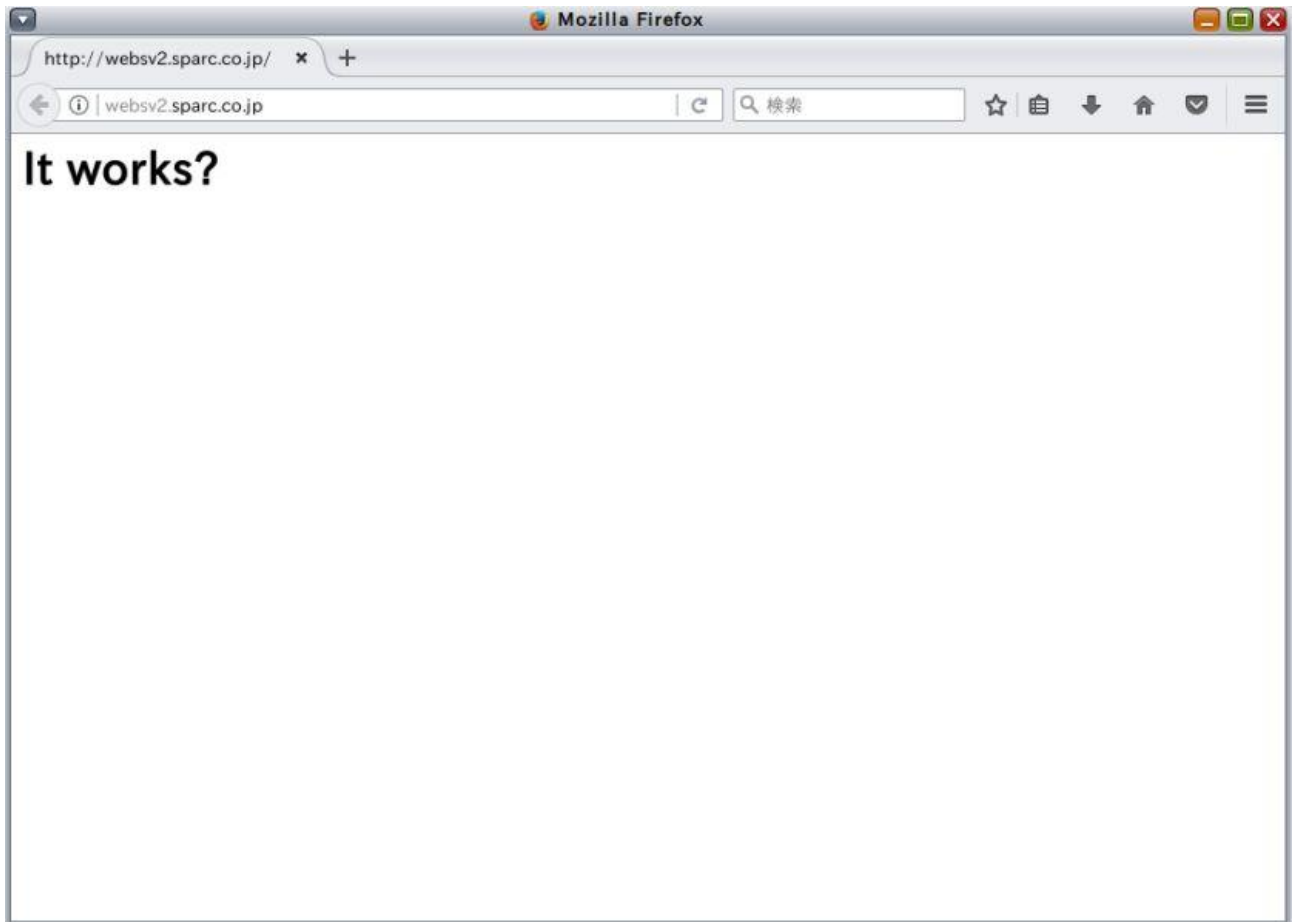
以下の設定を追記します。

```
# vi /etc/hosts
:
10.20.112.192    websv2.sparc.co.jp
```

6) 設定したサーバ名で Web サーバに接続できることを確認します。

以下の URL を入力します。

<http://新規ホスト名<:ポート番号>>



**Point**

ドメイン 2 に設定した、`/var/apache2/2.4/htdocs/dom2/index.html` の内容が表示されます。

## 8.2 IP ベース

設定例)

	ドメイン 1	ドメイン 2
ドメイン名	example.co.jp	sparc.co.jp
IP アドレス	10.20.112.192	10.20.112.193
DocumentRoot	/var/apache2/htdocs	/var/apache2/htdocs/dom2
ServerAdmin	websv@example.co.jp	websv2@sparc.co.jp

- 1) バーチャルホストの環境定義ファイルをコピーします。

```
# cp /etc/apache2/2.4/samples-conf.d/vhosts.conf /etc/apache2/2.4/conf.d/.
```

- 2) サーバに設定されている IP を確認します。

```
# ipadm show-addr
ADDROBJ      TYPE      STATE      ADDR
lo0/v4       static    ok         127.0.0.1/8
net0/v4       static    ok         10.19.112.192/24
net1/v4       static    ok         10.19.112.193/24
lo0/v6       static    ok         ::1/128
net0/v6       addrconf ok         fd80::c199:2ff4:fc9d:ff90/10
```

### 3) vhosts.conf ファイルを編集します。

サーバ名 (ServerName) とドキュメントルート (DocumentRoot) を編集します。

```
# vi /etc/apache2/2.4/conf.d/vhosts.conf
:
NameVirtualHost 10.20.112.192:80
#
# VirtualHost example:
# Almost any Apache directive may go into a VirtualHost container.
# The first VirtualHost section is used for all requests that do not
# match a ServerName or ServerAlias in any <VirtualHost> block.

<VirtualHost 10.20.112.192:80>
    ServerAdmin websv@example.co.jp
    ServerName websv.example.co.jp
    DocumentRoot /var/apache2/2.4/htdocs
</VirtualHost>

<VirtualHost 10.20.112.193:80>
    ServerAdmin websv2@sparc.co.jp
    ServerName websv2.sparc.co.jp
    DocumentRoot /var/apache2/2.4/htdocs/dom2
</VirtualHost>
```

← 「8.1 NAME ベース」で指定済みのため、  
ここでは修正の必要はありません

←ドメイン1の管理者メールアドレスを指定  
←ドメイン1のホスト名を指定  
←ドメイン1のドキュメントルートを指定

←ドメイン2の管理者メールアドレスを指定  
←ドメイン2のホスト名を指定  
←ドメイン2のドキュメントルートを指定

### 4) サービスを再起動します。

```
# svcadm restart svc:/network/http:apache24
```

### 5) サービスの状態を確認します。

```
# svcs svc:/network/http:apache24
STATE      STIME      FMRI
online     15:00:05  svc:/network/http:apache24
```

### 6) DNS サーバを設定します。

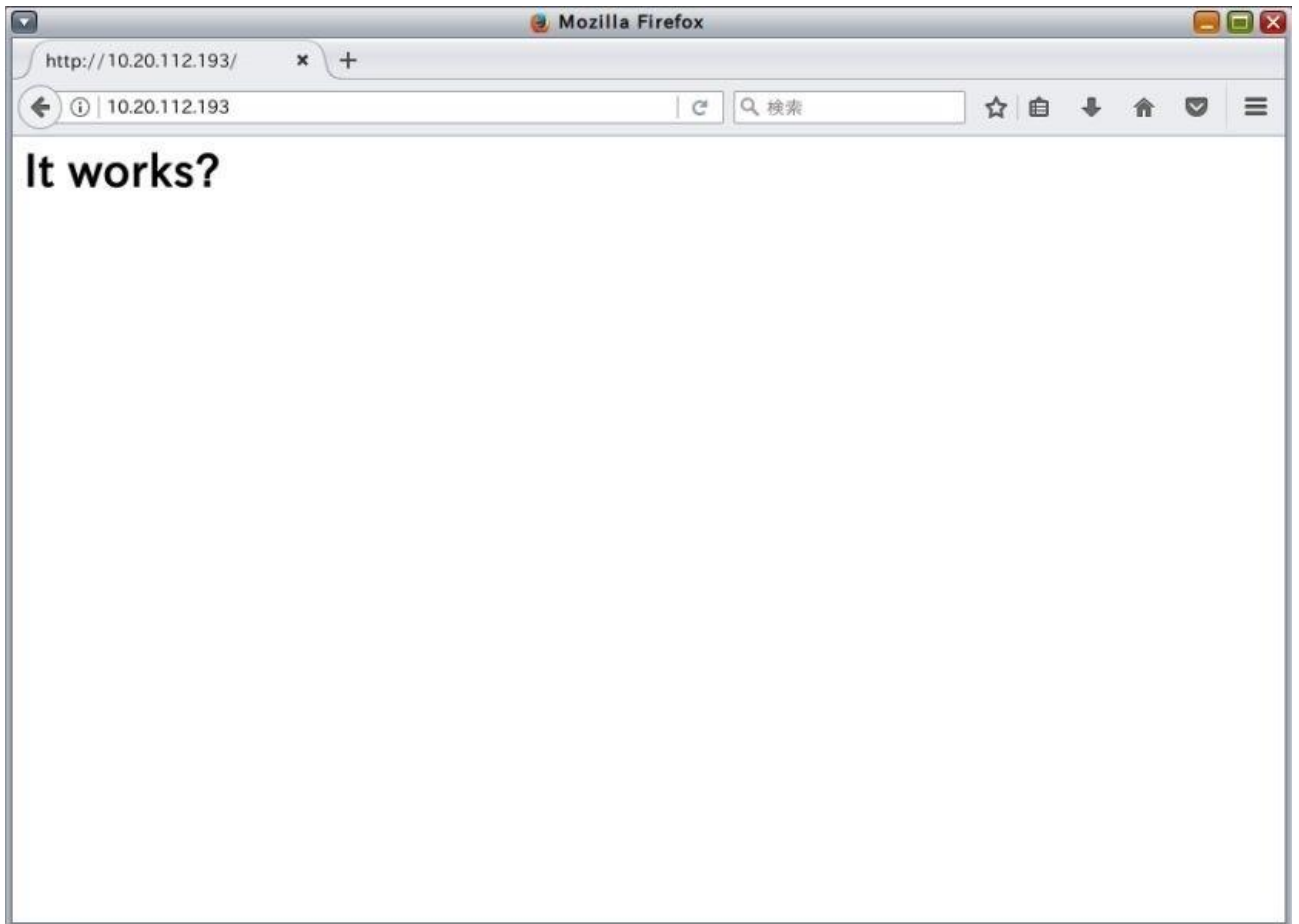
以下の設定を追記します。

```
# vi /etc/hosts
:
10.20.112.192 websv.example.co.jp
10.20.112.193 websv2.sparc.co.jp
```

7) 設定した IP アドレスで Web サーバに接続できることを確認します。

以下の URL を入力します。

<http://IP アドレス<:ポート番号>>





---

**参考文献**

- Apache HTTP サーバ バージョン 2.4 ドキュメント  
<http://httpd.apache.org/docs/2.4/>

## 改版履歴

改版年月	版数	改版内容
2019年9月	1.0版	新規作成

