

学習目標

この授業を受講すると，前回に引き続きApacheを用いて基本的なWebサーバの構築が出来るようになります．具体的には次のようなことが出来るように学習しましょう．

- ApacheによりWebページにパスワードによるアクセス制限を設定できるようになる
- Apacheにより仮想サーバを構築できるようになる

パスワード認証によるアクセス制限

Apacheでは，特定のディレクトリへのアクセスを制限する機能が用意されています．1つはホストごとに制限する方法で，アクセスしてくるホストのIPアドレスでアクセスを許可するか否かを判断します．もう1つは，ユーザで制限する方法で，アクセスする際にユーザ名とパスワードの入力を求め，正しく認証できた場合のみアクセスを許可する方法です．この授業では，後者のユーザ名とパスワードでアクセスを制限する方法を見ていきます．

Webminによる設定

Webminで特定のディレクトリへのパスワード認証によるアクセス制限を設定するには，まずパスワード認証を設定したいディレクトリに対する，オプション設定の画面を作成します．図28「仮想サーバのオプション デフォルトのサーバ」の設定画面の下の方にある「ディレクトリ/ファイル/ロケーション毎のオプションの作成」で，次のように入力します．ここでは，/var/www/html/ ディレクトリの中に himitsu ディレクトリを作成して，このディレクトリに対してパスワード認証によるアクセス制限を設定しています．

- タイプ：「ディレクトリ」
- 正規表現：「正確に一致」
- パス：パスワードを設定するディレクトリ(図28の例では /var/www/html/himitsu)

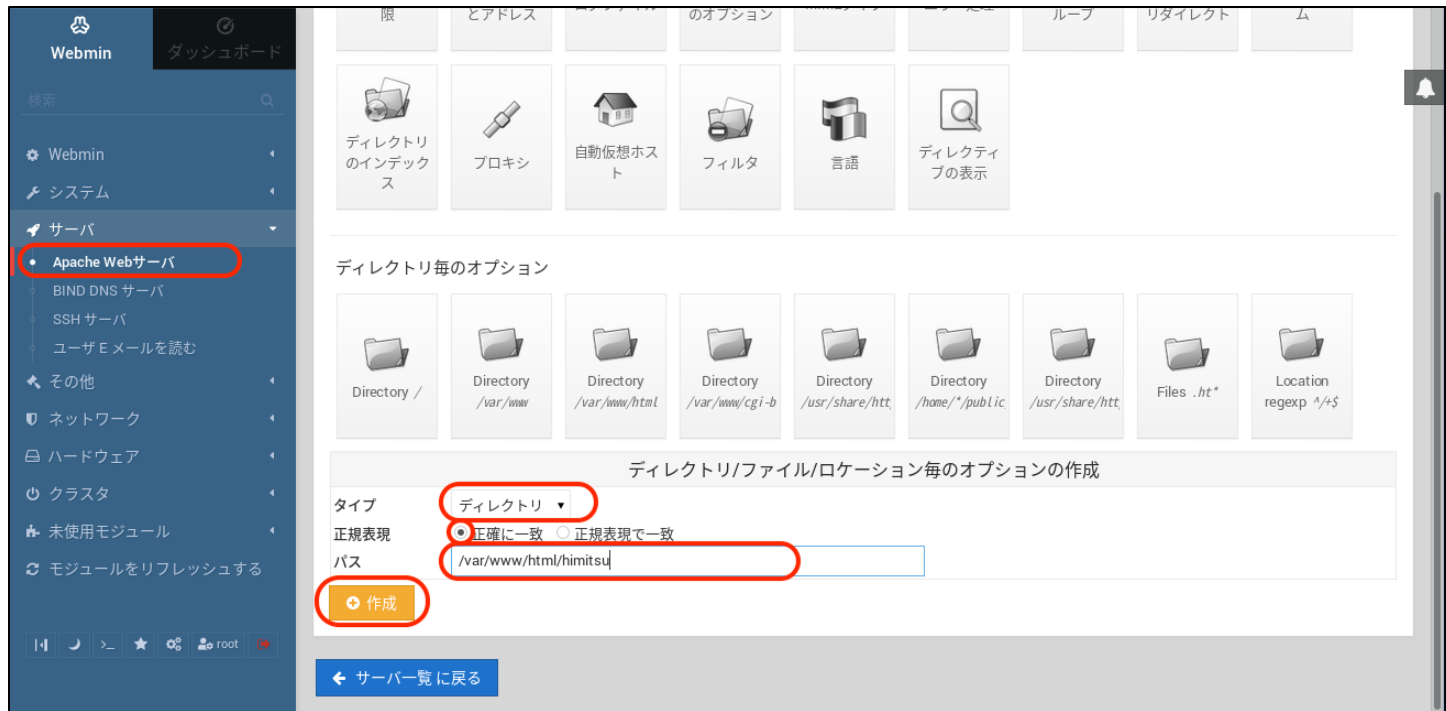


図28 ディレクトリ毎のオプションを作成する

入力が終わったら「作成」ボタンを押します．すると，図29のように，特定のディレクトリのオプション(図29の例では「Directory /var/www/html/himitsu」)が作成されますので，このディレクトリの設定画面を開きます．



図29 ディレクトリ毎の設定画面を開く

すると，図30の「ディレクトリ毎のオプション」の設定画面が開きますので，その中の「アクセス制御」の設定画面を開きます．

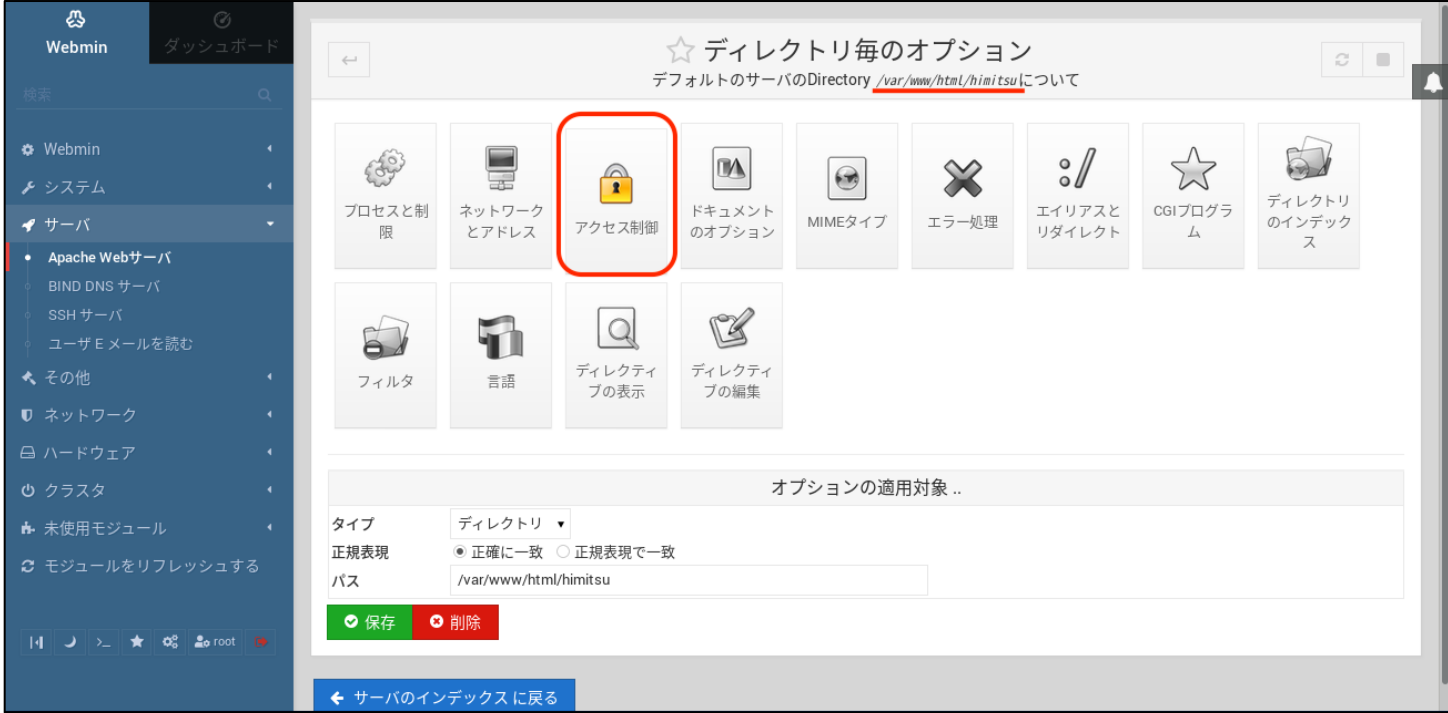


図30 「アクセス制御」を開く

図31のような「アクセス制御」の設定画面が開きます．この画面で次の項目を設定します．

☆ アクセス制御

デフォルトのサーバのDirectory /var/www/html/himitsuについて

Directory /var/www/html/himitsuのアクセス制御

認証領域の名前

☐ デフォルト

☒ himitsu no directory

☐ デフォルト

☒ これらのユーザのみ: himiko

☐ これらのグループのみ:

☐ ログインによるアクセス制限

☐ 有効なユーザすべて

☐ File owner matches

☐ Group owner matches

認証のタイプ

Basic

▼

クライアントの条件

☐ デフォルト

☐ すべてのアクセス制御

☒ 任意のアクセス制御

Pass basic login failures to next module?

☐ はい

☐ いいえ

☒ デフォルト

Basic login user file types

Text file

DBM database

Hashing algorithm

☐ apache

☐ apache

☒ デフォルト

Pass digest login failures to next module?

☐ はい

☐ いいえ

☒ デフォルト

Digest login user file types

Text file

DBM database

ユーザのDBMファイル

☒ デフォルト

☐ /etc/httpd/.htpasswd

ユーザのテキストファイル

☐ デフォルト

☒ /etc/httpd/.htpasswd

グループのDBMファイル

☒ デフォルト

☐

Pass group failures to next module?

☐ はい

☐ いいえ

☒ デフォルト

グループのテキストファイル

☒ デフォルト

☐

アクセスのチェック順

☒ 拒否の後に許可

☐ 許可の後に拒否

☐ 許可と拒否の両方

☐ デフォルト

アクセス制限

動作

条件

拒否

▼

すべての要求

▼

DBMファイルの種類(type)

デフォルト

▼

DBM group database type

デフォルト

▼

保存

ディレクトリのインデックスに戻る

図31 「アクセス制御」の設定画面

- ・ 認証領域の名前：「**himitsu no directory**」
ユーザ名とパスワードの入力画面に表示される、認証領域の名前を設定します。
- ・ 認証のタイプ：「**Basic**」
認証のタイプはベーシック認証とダイジェスト認証があります。ここではベーシック認証を選択しています。
- ・ ログインによるアクセス制限：「これらのユーザのみ」をチェックし、ユーザ名を入力
このディレクトリにアクセスできるユーザ名を設定します。ここで設定するユーザは、Linuxに登録されているユーザアカウントとは関係ありません。パスワードは後で設定します。図31の例では、ユーザ名として「himiko」と入力しています。
- ・ ユーザのテキストファイル：「**/etc/httpd/.htpasswd**」
このディレクトリにアクセスできるユーザとパスワードを記憶するファイルを設定します。
図31では /etc/httpd/.htpasswd を指定しています。このファイルはWebで公開するディレクトリには保存しないようにします。「ユーザのテキストファイル」は、ユーザ名とパスワードのリストです。一方「ログインによるアクセス制限」で設定するのは、ユーザのテキストファイルに登録されているどのユーザがアクセスできるか、です。

- ・ アクセス制限 - アクセスのチェック順：「拒否の後に許可」
 - ・ アクセス制限 - 動作：「拒否」
 - ・ アクセス制限 - 条件：「すべての要求」
- このディレクトリへのアクセス制限を適用する順番を設定します。いまは「すべての要求」を「拒否」する設定のみですが、複数のアクセス制限を設定する場合は、アクセスのチェック順が重要になります。たとえば、すべてを拒否して特定のホストだけに公開するような制限を設ける場合は、「拒否の後に許可」なら、すべてを拒否して例外的に許可することになりますが、「許可の後に拒否」では、例外的な許可のあとにすべての拒否を設定することになるため、結局すべて拒否となってしまいます。

設定が終わったら「保存」ボタンを押します。設定を有効にするため、「変更を適用」のリンクをクリックします(図32)。

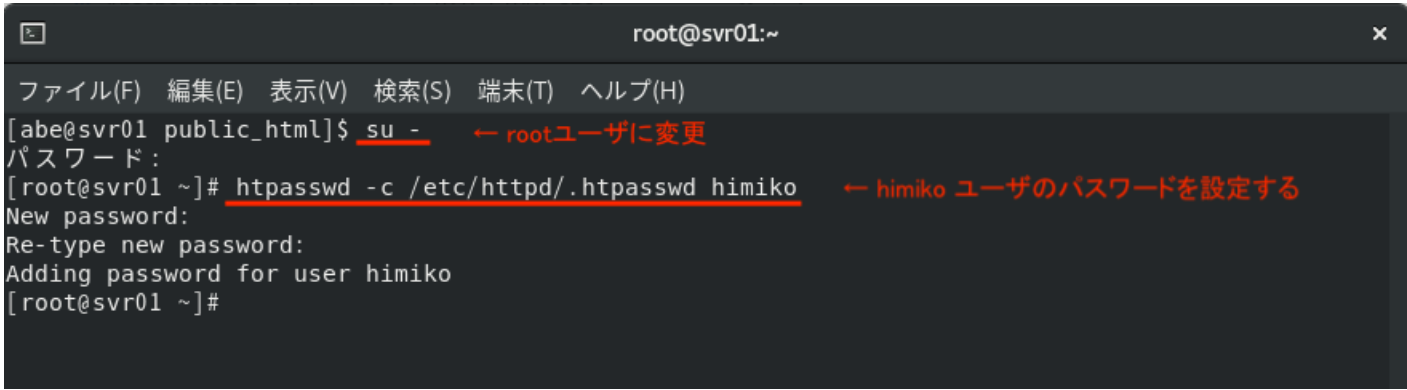


図32 設定が終わったら「変更を適用」する

次に、パスワードファイルを作成します。この作業はWebminからではなくプロンプトから行います。ユーザ名とパスワードを登録するには「htpasswd」コマンドを用いて次のように入力します。

```
htpasswd -c /etc/httpd/.htpasswd himiko
```

htpasswdコマンドを実行すると、ユーザのパスワードを聞いてきますので、パスワードを設定します(図33)。(htpasswdコマンドはrootユーザで実行します)



```
root@svr01:~
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)
[abe@svr01 public_html]$ su -      ← rootユーザに変更
パスワード:
[root@svr01 ~]# htpasswd -c /etc/httpd/.htpasswd himiko  ← himiko ユーザのパスワードを設定する
New password:
Re-type new password:
Adding password for user himiko
[root@svr01 ~]#
```

図33 htpasswdコマンドでユーザ名とパスワードを設定する

htpasswdに与えた3つの引数の意味は次のとおりです。

-cオプションは、パスワードファイルを作成するときに指定します。既にあるパスワードファイルにユーザを追加する場合は-cは不要です。(cを付けるとファイルを上書きしてしまいます。)

2つめの引数/etc/httpd/.htpasswd では、図32の「ユーザのテキストファイル」で設定したファイルを指定しています。

3つめの引数himiko では、図32の「ログインによるアクセス制限」で設定したユーザ名を指定しています。

※htpasswdコマンドでユーザとパスワードを設定しない場合、パスワードなしでディレクトリにアクセスできてしまいますので注意してください。

設定内容の確認

パスワード認証によるアクセス制限の設定を httpd.conf ファイルで確認してみましょう。
図34にhttpd.confファイルの一部を示します。

```

#
# Some examples:
#ErrorDocument 500 "The server made a boo boo."
#ErrorDocument 404 /missing.html
#ErrorDocument 404 "/cgi-bin/missing_handler.pl"
#ErrorDocument 402 http://www.example.com/subscription_info.html
#
#
# EnableMMAP and EnableSendfile: On systems that support it,
# memory-mapping or the sendfile syscall may be used to deliver
# files. This usually improves server performance, but must
# be turned off when serving from networked-mounted
# filesystems or if support for these functions is otherwise
# broken on your system.
# Defaults if commented: EnableMMAP On, EnableSendfile Off
#
#EnableMMAP off
EnableSendfile on
#
# Supplemental configuration
#
# Load config files in the "/etc/httpd/conf.d" directory, if any.
IncludeOptional conf.d/*.conf
<Directory "/var/www/html/himitsu">
AuthName "himitsu no directory"
AuthType Basic
require user himiko
Satisfy any
AuthUserFile /etc/httpd/.htpasswd
deny from all
order deny,allow
</Directory>

```

図34 httpd.confファイルの「パスワード認証」に関する設定項目

「<Directory "/var/www/html/himitsu">」から「</Directory>」までが、/var/www/html/himitsuディレクトリに関する設定項目です。図31で設定した内容が記述されています。

「AuthName」では「認証領域の名前」を設定しています。

「AuthType」では「認証のタイプ」を設定しています。

「require user」では「ログインによるアクセス制御」で設定したユーザ名を設定しています。

「AuthUserFile」では「ユーザのテキストファイル」で設定した、パスワードファイルの位置を指定しています。

「deny from all」では、「アクセス制限」の「動作」と「条件」を設定しています。

「order deny,allow」では、「アクセス制限」の「アクセスのチェック順」を設定しています。

動作確認

それではパスワード認証によるアクセス制限の動作確認をしてみましょう。まず、/var/www/html/ ディレクトリに himitsu という名前のディレクトリを作成します。このフォルダのURLは http://10.0.2.100/himitsu/ となります。このフォルダの中に、リスト3のような index.html を作成します。

リスト3 himitsuディレクトリのindex.html

```
<html>
<head></head>
<body>
Authentication success!<br>
(/var/www/html/himitsu/index.html)
</body>
</html>
```

Webブラウザに 上のURL 「http://10.0.2.100/himitsu」 を入力して、このページのインデックスページを開こうとすると、図35のようにユーザ名とパスワードを入力するダイアログボックスが表示されるので、ユーザ名とパスワードを入力します。入力したユーザ名とパスワードが正しければ himitsu ディレクトリにアクセスできるようになります(図36)。

※himitsu ディレクトリにはすべてのユーザが「読み出し」「書き込み」「実行」できる権限が設定されていることを確認してください。またindex.html ファイルにはすべてのユーザが「読み出し」できる権限が設定されていることを確認してください。

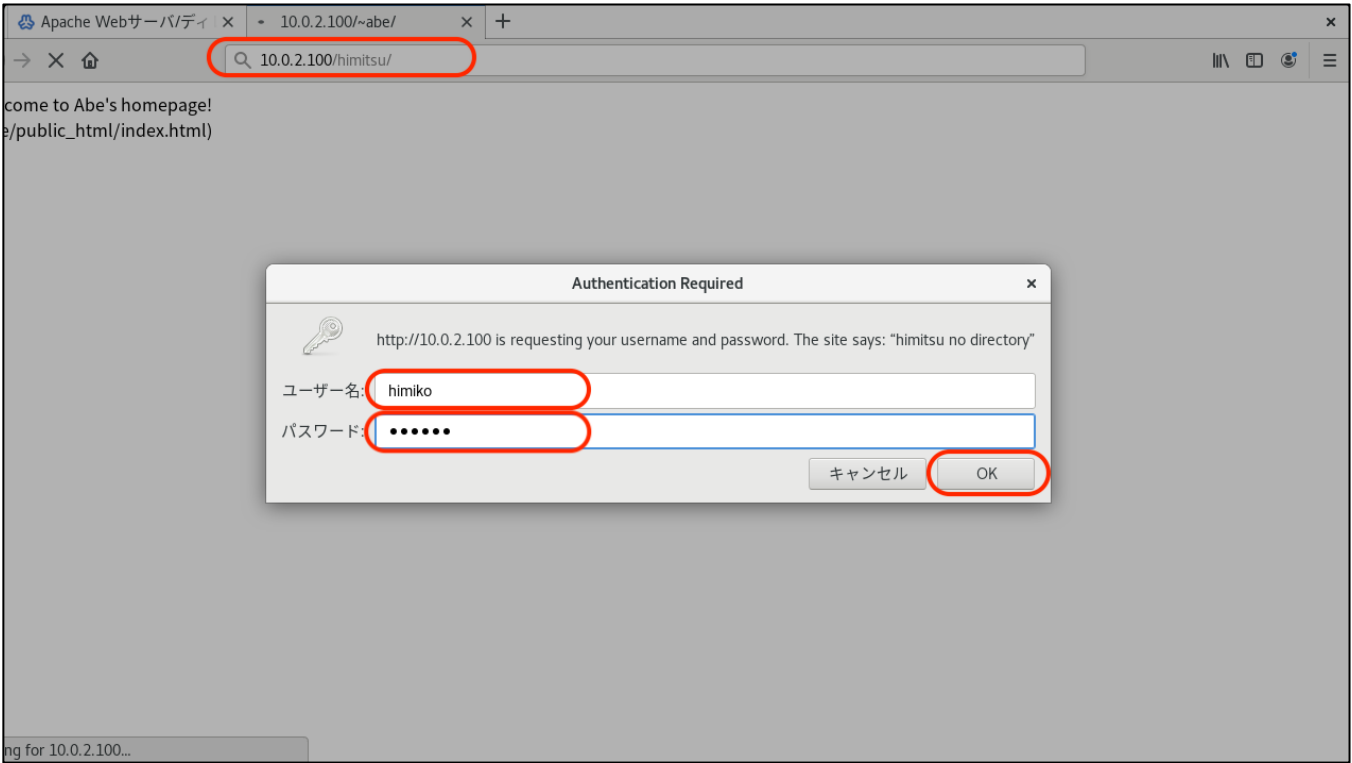


図35 himitsuディレクトリをユーザとパスワードで認証

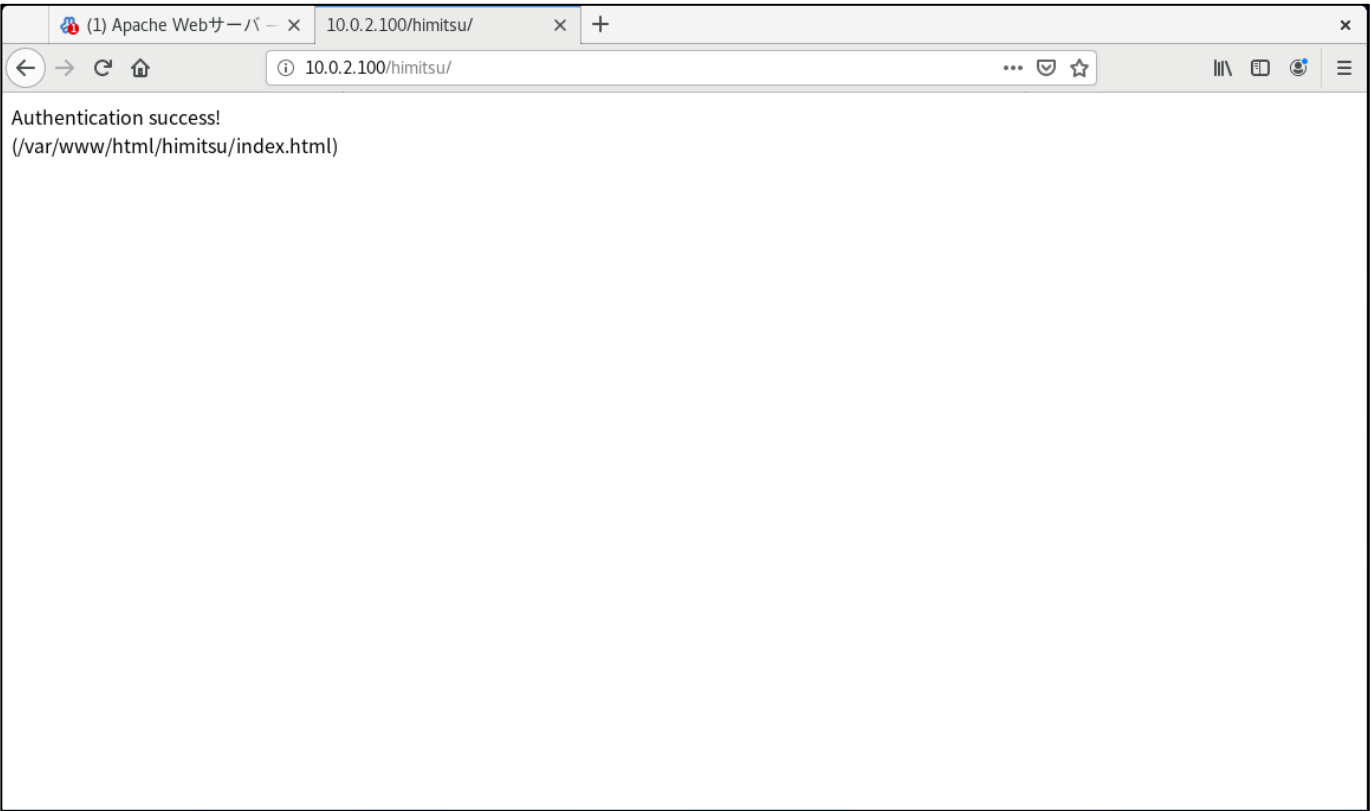


図36 himitsuディレクトリにアクセスできた

仮想ホストの設定

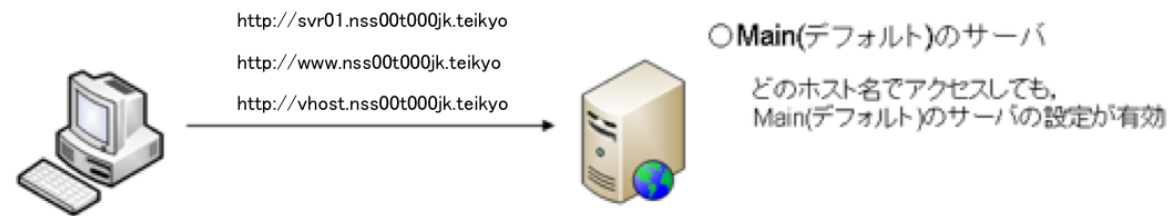
仮想ホスト(Virtual host)とは、1台のWebサーバで複数のドメイン名のWebサイトを利用できるようにすることです。たとえば、<http://www.nss00t000.teikyo/> と <http://vhost.nss00t000.teikyo/> という2つのホストを構築する場合、通常は www と vhost の2つのWebサーバが必要となりますが、これを1つのWebサーバ上で構築するのが仮想ホストです。異なるホスト www と vhost では、通常別々のWebページを表示するようにしますが、仮想ホストでもホストごとに表示するWebページを設定できるようになっています。(単に異なるドメイン名でもアクセスできるWebページというわけではありません。)

仮想ホストの実現方法には、仮想ホストで用いる複数のドメイン名に対して、それぞれ別のIPアドレスを振る方法(IPベース)と、複数のドメインで同じIPアドレスを用いる方法(NAMEベース)があります。この授業では、後者の1つのIPアドレスを用いる方法(NAMEベース)により、仮想ホストを構築します。

ドメイン名とIPアドレスの対応を管理するのは「DNSの設定」で扱ったDNSサーバでした。BINDによるDNSサーバでは、ゾーンファイルのAレコードでゾーンのホストのIPアドレスを設定していました。BINDでは、Aレコードに同じIPアドレスを複数登録することはできませんが、既に登録されているホスト名に、別名(エイリアス)を設定することができます。別名はCNAMEレコードで登録します。

仮想ホストを設定した場合のWebサーバの動作の仕組みを図48で説明します。

仮想ホストを設定していない場合



仮想ホストを設定している場合

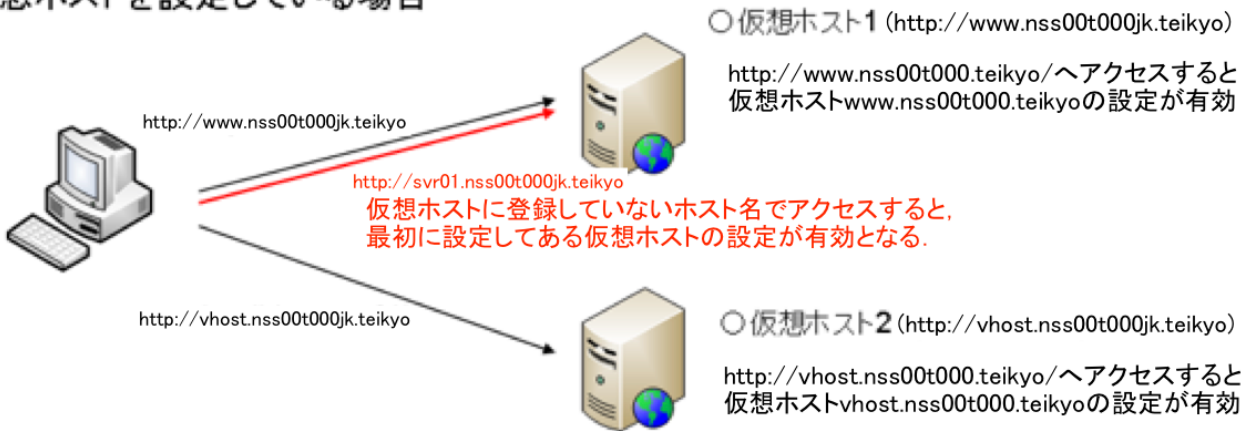


図48 仮想ホストの動作の仕組み

いま、svr01.nss00t000.teikyoというサーバにApacheを導入してWebサーバとして利用することにします。クライアントのWebブラウザのURLに <http://svr01.nss00t000.teikyo/> と入力すると、Webサーバは Main(デフォルト)の設定で応答します。

Webサーバには www というホスト名でアクセスさせたいので、DNSで www という別名を付けます。これで

www.nss00t000.teikyo でWebサーバにアクセスできます。(svr01.nss00t000.teikyoでもアクセスできます。) クライアントのWebブラウザのURLに http://www.nss00t000.teikyo/ と入力すると、Webサーバは svr01 の時と同様に Main(デフォルト)の設定で応答します。この設定は、Webサーバに別名を付けただけなので www でアクセスしても svr01 でアクセスしても、Webサーバは同じホストとして動作します。(表示されるWebページはMain(デフォルト)のページです。)

次に、仮想ホストを設定する場合を考えます。その前に、DNSに vhost という別名を登録しておきます。これで、svr01.nss00t000.teikyo、www.nss00t000.teikyo、vhost.nss00t000.teikyo の名前解決をすると、DNSサーバはどの場合もsvr01.nss00t000.teikyoのIPアドレス(10.0.2.100)を返すようになります。

Webブラウザに仮想ホスト「vhost.nss00t000j.teikyo」の設定を行い、クライアントのWebブラウザのURLに http://vhost.nss00t000.teikyo/ と入力すると、仮想ホストvhostの設定で応答します。仮想ホストがvhost 1つだけの場合は http://svr01.nss00t000.teikyo/ あるいは http://www.nss00t000.teikyo/ でWebサーバにアクセスしたときも、仮想ホストvhost.nss00t000.teikyo の設定で動作します。(仮想ホストを1つでも設定すると、Main(デフォルト)の設定は利用されず、仮想ホストの設定が有効になります。)

例えばwww.nss00t000.teiokyo と vhost.nss00t000.teikhyo の2つのドメイン名を持つサーバを構築する場合は、www とvhostという2つの仮想ホストを構築します。仮想ホストごとに、表示されるドキュメントの場所(DocumentRoot)を設定できるため、仮想ホストごとに別の内容を表示するように設定することができます。このように、仮想ホストを使うと、1台のWebサーバでwwwと、vhost という、異なる2台のWebサーバがあるかのように設定することができます。

仮想ホストを設定したWebサーバに、仮想ホストで登録したホスト名以外のホスト名(ここではsvr01)や、IPアドレスでアクセスした場合には、最初に登録した仮想ホストのWebページが表示されます。これは、httpd.confの中で一番上に記述してある仮想サーバがデフォルトのサーバとしてふるまうからです。Webminで設定する際には仮想サーバの記述を後から入れ替えることはできないようですので、デフォルトとしてふるまうサーバを、最初に仮想サーバとして登録するようにします。この授業では、既に構築したMain(デフォルト)のサーバの設定で、仮想ホストwwwを構築し、その後で別のドキュメントルートを持つ仮想サーバvhostを構築することにします。こうすると、wwwがデフォルトのサーバとなるため、svr01やIPアドレスでアクセスした際にはwww としてふるまうようになります。

DNSの設定

現在，サーバには「DNSの設定」で設定したホスト名(svr01)がついています．Webサーバは通常wwwというホスト名で運用されますので，このサーバをwwwというホスト名でもアクセスできるようにします．DNSのAレコードには同じIPアドレスを登録することはできないため，既に登録されているホスト名に別名を登録します．こうすることで，別名でもこのホストのIPアドレスが引けるようになります．

このサーバに次のような別名を設定します．

www

Webサーバとしてのホスト名を「www」とします．すなわちドメイン名は「www.nss00t000.teikyo」となります．(ドメインnss00t000.teikyo は説明のための例です．実際には学生ごとに指定されたドメイン(自分の学籍番号を入れる)を利用してください．)

vhost

仮想ホストとして利用するホスト名を「vhost」とします．(ドメイン名はvhost.nss00t000.teikyo となります．)

Webminによる設定

WebブラウザからWebmin(<https://localhost:10000/>)を起動します。Webminのメニューから、「サーバ」を開き、その中の「BIND DNSサーバ」の設定画面を開きます(図3)。



図3 Webminの「サーバ」メニューから「**BIND DNS**サーバ」を開く

「既存のDNSゾーン」にある、ゾーンの設定画面を開きます。ここでは例として「nss00t000.teikyo」ドメインの設定画面を開きます(図4)。



図4 「既存のDNSゾーン」にある，ゾーンの設定画面を開く

図5のような「マスターゾーンの編集」画面が開いたら，既に登録されているホストに別名を設定するため「ネームエイリアス」の設定画面を開きます．



図5 「ネーム エイリアス」の設定画面を開く

図6のような「ネーム エイリアス レコード」の設定画面が開きますので，「レコード名」の欄に，新たに別名として設定する「www」を入力します．また，「実名」の欄には，既にAレコードでホスト名と関連付け てIPアドレスが設定してある，このサーバのホスト名(この例ではsvr01)を入力します．入力が終わったら，「作成」ボタンを押します．



図6 「www」を既存のホスト「svr01」の別名として登録する

画面が 図7のように変わり，svr01の別名として www.nss00t000.teikyo が登録されたことが分かります。

「www」と同様に，「vhost」もこのサーバのホスト名の別名として登録してください。



図7 「www.nss00t000.teiokyo」と「vhost.nss00t000.teiokyo」が「svr01」の別名として登録された

別名の登録が終わったら「変更を適用」ボタンを押します(図8)．これで登録した別名が有効になります．



図8 「変更を適用」して登録した別名を有効にする

登録した別名はゾーンファイルに登録されています。viエディタなどでゾーンファイル(ここでは /var/named/nss00t000.teikyo.hosts)を開くと、図9のような内容となっています。CNAMEレコードで「www.nss00t000.teikyo.」が「svr01」の別名として登録されています。同様に、「vhost.nss00t000.teikyo.」も「svr01」の別名として登録されています。

[illegible]

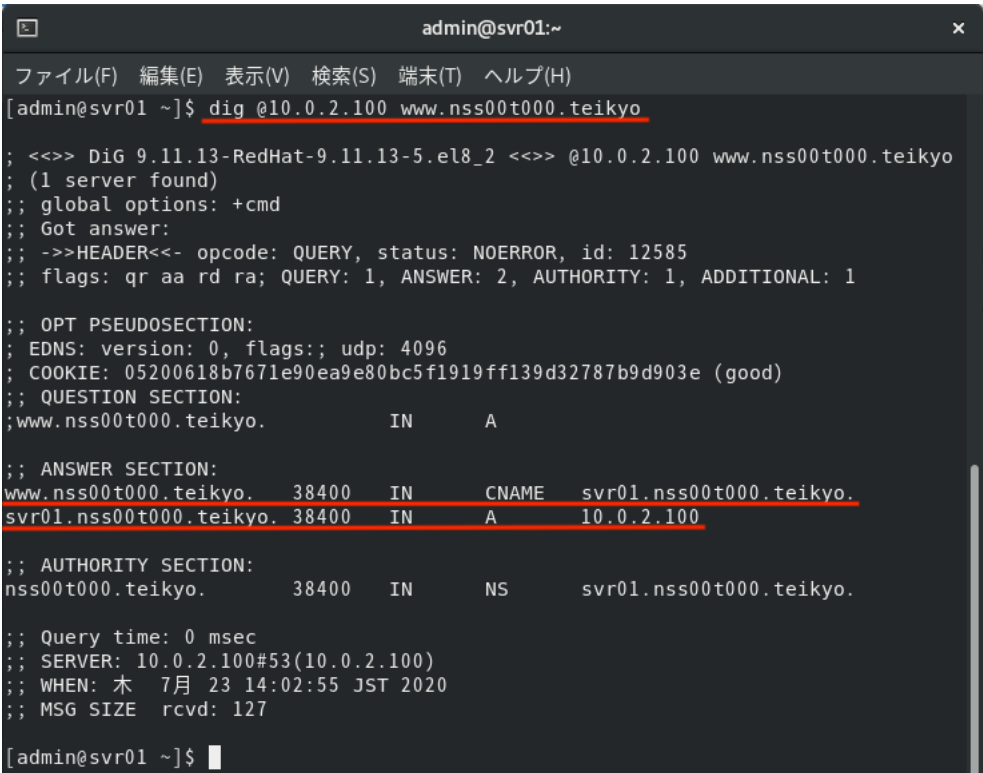
図9 ゾーンファイル(nss00t000.teikyo.hosts)に登録された別名(wwwとvhost)

動作確認

登録した別名で名前解決ができるかどうかを、digコマンドで確認してみます。プロンプトから次のようにdigコマンドを実行します。

```
dig @10.0.2.100 www.nss00t000.teikyo
```

digコマンドの実行結果を図10に示します。「www.nss00t000.teikyo.」が「svr01.nss00t000.teikyo.」の別名であることがCNAMEレコードから得られ、「svr01.nss00t000.teikyo.」のIPアドレスが「10.0.2.100」であることがAレコードから得られています。



```
admin@svr01:~
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)
[admin@svr01 ~]$ dig @10.0.2.100 www.nss00t000.teikyo
; <<>> DiG 9.11.13-RedHat-9.11.13-5.el8_2 <<>> @10.0.2.100 www.nss00t000.teikyo
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 12585
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 05200618b7671e90ea9e80bc5f1919ff139d32787b9d903e (good)
;; QUESTION SECTION:
;www.nss00t000.teikyo.      IN      A

;; ANSWER SECTION:
www.nss00t000.teikyo.     38400   IN      CNAME   svr01.nss00t000.teikyo.
svr01.nss00t000.teikyo.  38400   IN      A       10.0.2.100

;; AUTHORITY SECTION:
nss00t000.teikyo.        38400   IN      NS      svr01.nss00t000.teikyo.

;; Query time: 0 msec
;; SERVER: 10.0.2.100#53(10.0.2.100)
;; WHEN: 木 7月 23 14:02:55 JST 2020
;; MSG SIZE rcvd: 127

[admin@svr01 ~]$
```

図10 digコマンドで「www.nss00t000.teikyo」の名前解決を行った結果

vhost.nss00t000.teikyo についても同様に確認してみましょう。

仮想ホストの設定 (Webminによる設定)

Webminで仮想ホストを設定するには、「Apache Webサーバ」の設定画面から、「仮想ホストの作成」タブを開きます。すると図49の「新しい仮想サーバの作成」画面が表示されます。ここでは www と vhost の2つの仮想ホストを作成します。wwwがデフォルトのホストとなるように、wwwを先に作成します。



図49 「仮想ホストの作成」タブで新しい仮想サーバを作成する

「アドレスへの接続を取り扱う」には、「特定のアドレス」として、サーバのIPアドレス(図49の例では10.0.2.100)を入力します。NAMEベースの設定とする場合は、ここにはすべて同じIPアドレスを設定します。

「ポート」には、この仮想サーバにアクセスするポート番号を設定します。図49では「80」番ポートを設定しています。

「ドキュメントのルート」には、この仮想ホストのルートディレクトリを設定します。

「サーバ名」には、この仮想ホストのホスト名(ドメイン名)を設定します。図49では、Main(デフォルト)の設定と同じドキュメントルートとサーバ名を設定しています。

設定が終わったら「作成する」ボタンを押して、仮想ホストを作成します。すると、図50のように、仮想ホストが1つ追加されます。



図50 仮想ホストが追加された

同様に、仮想ホストvhostも作成します。

図51のように仮想ホストvhostを作成します。

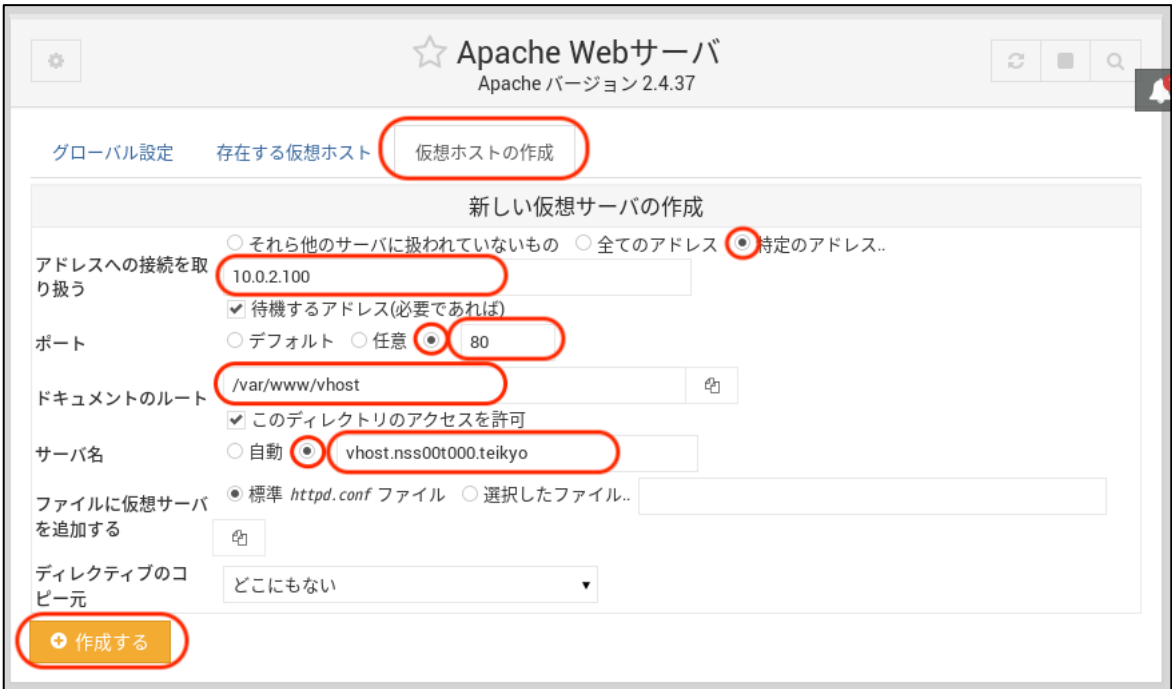


図51 新しい仮想ホストvhostを作成する

すると、図52のように、仮想ホストvhostが作成されます。

仮想ホストを作成したら、設定を有効にするため「変更を適用」ボタンを押します。



図52 仮想ホストvhostが追加された

これで仮想ホストの作成が終わりました。仮想ホストwww.nss00t000.teikyo はすでに作成したドキュメントルートを 使いますのでそのままアクセスできます。vhost.nss00t000.teikyo のドキュメントルートにはまだファイルがないため、リスト5のようなファイルを作成します。

リスト5 /var/www/vhost の index.html

```
<html>
<head></head>
<body>
Welcome to Virtual Host
(vhost)<br>
(/var/www/vhost/inde.html)
</body>
</html>
```

WebブラウザからURL「http://vhost.nss00t000.teikyo/」にアクセスすると、図53のよう
に、/var/www/vhost/index.html が表示されます。このように、仮想ホストが有効になっていることが分かり
ます。

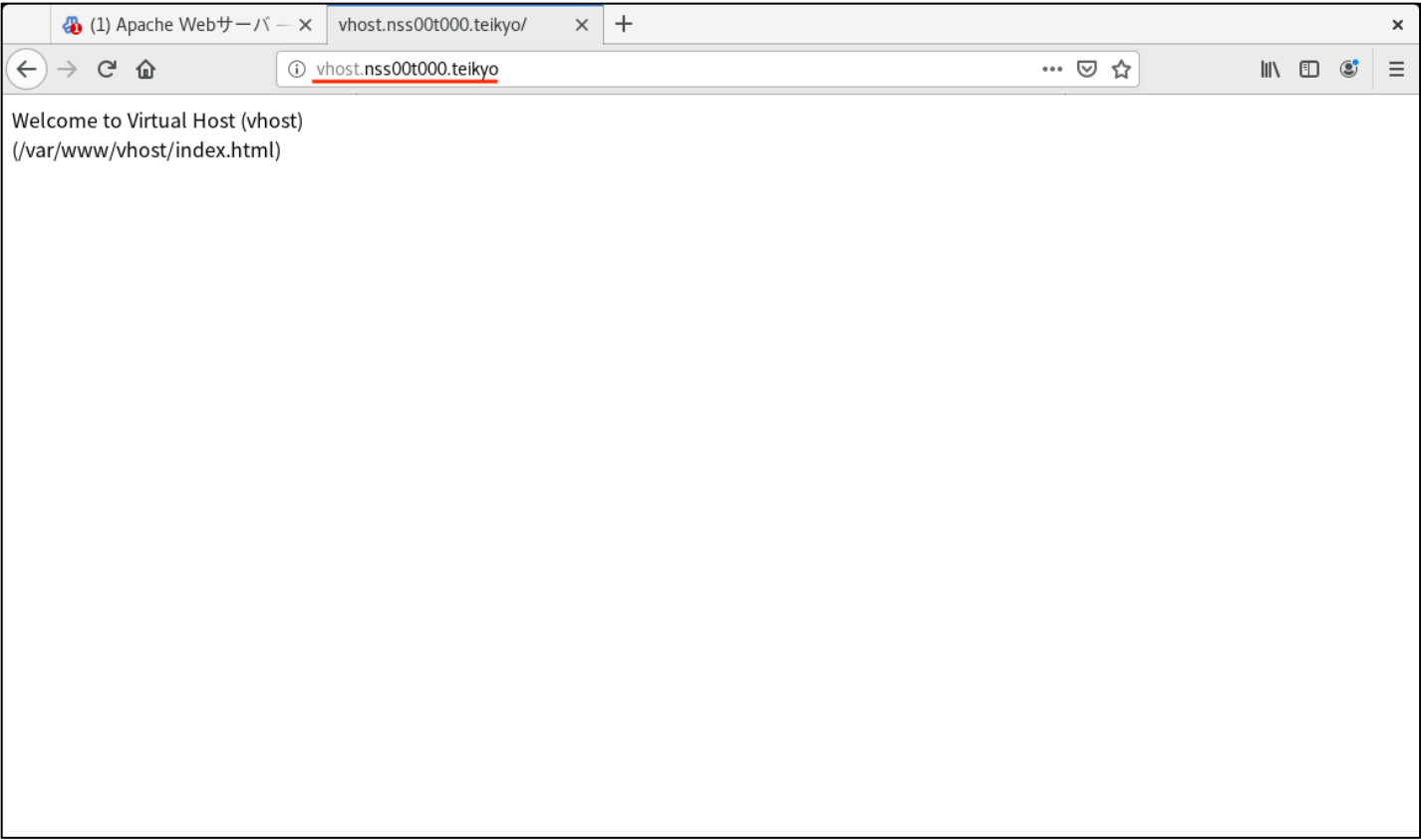
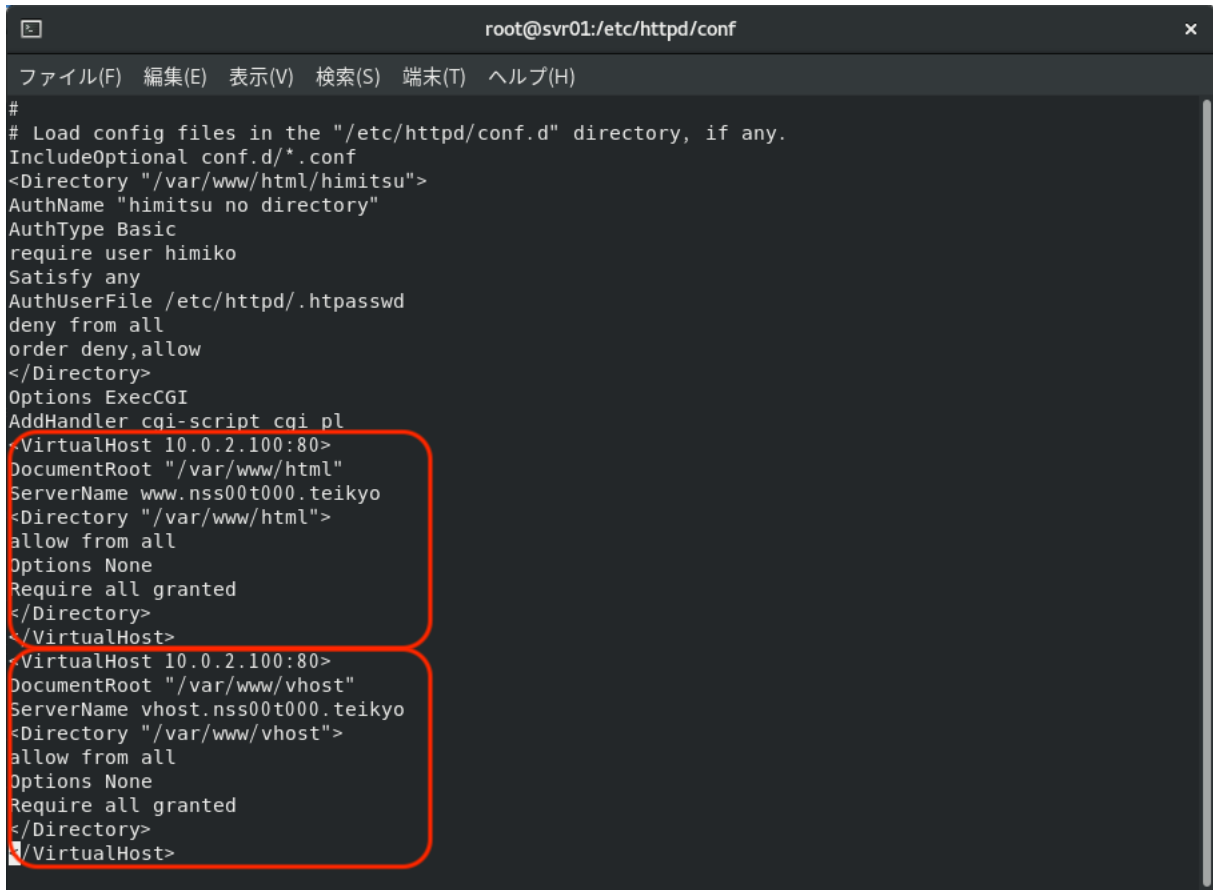


図53 仮想ホスト **http://vhost.nss00t000.teikyo/** にアクセスした結果

問3 Webブラウザに次のURLを入力して，どのページが表示されるか確認してみよう．

- (1) http://www.nss00t000.teikyo/ （学生ごとに割り振られたドメイン）
- (2) http://svr01.nss00t000.teikyo/ （学生ごとに割り振られたドメイン）
- (3) http://10.0.2.100/ （サーバのIPアドレス）

NAMEベースによる仮想ホストwwwとvhostの設定を，httpd.confファイルで確認してみましょう．
図54にhttpd.confファイルの一部を示します．（図54は httpd.conf ファイルの最後の部分）



```
#
# Load config files in the "/etc/httpd/conf.d" directory, if any.
IncludeOptional conf.d/*.conf
<Directory "/var/www/html/himitsu">
AuthName "himitsu no directory"
AuthType Basic
require user himiko
Satisfy any
AuthUserFile /etc/httpd/.htpasswd
deny from all
order deny,allow
</Directory>
Options ExecCGI
AddHandler cgi-script cgi pl
<VirtualHost 10.0.2.100:80>
DocumentRoot "/var/www/html"
ServerName www.nss00t000.teikyo
<Directory "/var/www/html">
allow from all
Options None
Require all granted
</Directory>
</VirtualHost>
<VirtualHost 10.0.2.100:80>
DocumentRoot "/var/www/vhost"
ServerName vhost.nss00t000.teikyo
<Directory "/var/www/vhost">
allow from all
Options None
Require all granted
</Directory>
</VirtualHost>
```

図54 httpd.confファイルの仮想ホストに関する設定項目

Httpd.confファイルの最後に，<VirutalHost 10.0.2.100:80> タグの中に仮想ホストの設定が記述されています．

「DocumentRoot」は，この仮想ホストの「ドキュメントのルートディレクトリ」です．

「ServerName」は，この仮想ホストの「サーバ名」です．このサーバ名でアクセスすると，Webサーバはこの仮想ホストとしてふるまいます．

<Directory>タグの中では，ドキュメントのルートディレクトリの設定をしています．

「allow from all」ではすべてのアクセスを許可しています．