

### 3.Webサーバ(apatche)の構築

---

#### 【事前準備】

---

①下記のとおりネットワーク設定が行われている

(受講者1)

```
仮想マシン名      : student1
ドメイン名        : alpha.jp
ホスト名          : host1.alpha.jp
IPアドレス        : 192.168.1.101
サブネットマスク  : 255.255.255.0 (/24)
デフォルトゲートウェイ : 192.168.1.1
DNSサーバアドレス : 192.168.1.10
```

(受講者2)

```
仮想マシン名      : student2
ドメイン名        : beta.jp
ホスト名          : host2.beta.jp
IPアドレス        : 192.168.1.102
サブネットマスク  : 255.255.255.0 (/24)
デフォルトゲートウェイ : 192.168.1.1
DNSサーバアドレス : 192.168.1.10
```

#### 【事前準備2】

それぞれの学生用PC(仮想マシンstudent)でDNSコンテンツサーバが稼働している

教師用PC(仮想マシンteacher)でDNSキャッシュサーバが稼働している

以降、管理者アカウントで作業をおこなう

**sudo su -**

【必要なパッケージをインストール】

```
[root@host2 admin]# yum install -y httpd bind-utils
```

## 設定ファイルの修正

apacheの設定ファイルは `/etc/httpd/conf/httpd.conf`

今回設定する部分がどうなっているか確認する

```
~# grep "Listen" /etc/httpd/conf/httpd.conf
# Listen: Allows you to bind Apache to specific IP addresses and/or
# Change this to Listen on specific IP addresses as shown below to
#Listen 12.34.56.78:80
Listen 80

~# grep "DocumentRoot" /etc/httpd/conf/httpd.conf
# DocumentRoot: The directory out of which you will serve your
DocumentRoot "/var/www/html"
# access content that does not live under the DocumentRoot.

~# grep "ServerName" /etc/httpd/conf/httpd.conf
# ServerName gives the name and port that the server uses to identify itself.
#ServerName www.example.com:80

~# grep "DirectoryIndex" /etc/httpd/conf/httpd.conf
# DirectoryIndex: sets the file that Apache will serve if a directory
DirectoryIndex index.html

~# grep "AddDefaultCharset" /etc/httpd/conf/httpd.conf
AddDefaultCharset UTF-8
root@host2.beta.jp ~#
```

確認したところ、ServerNameがコメントアウトされているので、設定を変更する

```
#ServerName www.example.com:80
```

↓

```
ServerName www.alpha.jp
```

viエディタで変更してもよいが、今回は文字列の置換コマンドsedを用いる

# 元々コメントアウトされている場合

```
# sed -i -e "/^ *#ServerName www.example.com:80/c\ ServerName www.alpha.jp" /etc/httpd/conf/httpd.conf
```

設定の確認

```
# apachectl configtest
Syntax OK
```

参考 # コメントアウトされていない場合

例 abcde の文字列を hello taro を hey hanakoに変更する

```
sudo sed -i -e "/^ * hello taro/c\    hey hanako" 【対象ファイル】
```

### 公開するindex.htmlを編集する

```
~# echo "This is Test Page on host1.alpha.jp" > /var/www/html/index.html

~# cat /var/www/html/index.html
This is Test Page on host1.alpha.jp
```

※仮想マシンstudent2 では host1.alpha.jp は、host2.beta.jp に変更する

### ファイアウォールの設定

```
~# firewall-cmd --add-service=http
Warning: ALREADY_ENABLED: 'http' already in 'public'
success
```

### 設定の保存

```
~# firewall-cmd --runtime-to-permanent
success
```

### 設定の反映

```
~# firewall-cmd --reload
success
```

### Apacheの起動と確認

```
~# systemctl start httpd
```

### 確認

```
~# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since 金 2023-05-26 15:14:46 JST; 1h 3min ago
     Docs: man:httpd(8)
           man:apachectl(8)
  Main PID: 1172 (httpd)
    Status: "Total requests: 0; Current requests/sec: 0; Current traffic:  0 B/sec"
    CGroup: /system.slice/httpd.service
            └─1172 /usr/sbin/httpd -DFOREGROUND
              └─1265 /usr/sbin/httpd -DFOREGROUND
                └─1266 /usr/sbin/httpd -DFOREGROUND
```

```
|—1267 /usr/sbin/httpd -DFOREGROUND
|—1268 /usr/sbin/httpd -DFOREGROUND
|—1269 /usr/sbin/httpd -DFOREGROUND
```

```
5月 26 15:14:46 host2.beta.jp systemd[1]: Starting The Apache HTTP Server...
5月 26 15:14:46 host2.beta.jp systemd[1]: Started The Apache HTTP Server.
root@host2.beta.jp ~#
```

## 自動起動の設定

```
~# systemctl enable httpd
~# systemctl is-enabled httpd
enabled
```

## 接続の確認

### ①Windowsホストから確認

WindowsからHyper-Vの仮想マシンのIPアドレスを指定して、ブラウザで確認する

ブラウザのURL欄に 学生マシンのIPアドレスを入力 「192.168.1.101」

Enter押下でページが表示されることを確認する

### ②テキストベースブラウザで確認

Linuxで利用できるテキストベースのブラウザLynxを利用するとCUI上でも確認ができる

Lynxのインストール

```
~# yum install -y lynx
```

lynxで表示確認

```
~# lynx 192.168.1.101
~# lynx host1.alpha.jp
~# lynx www.alpha.jp
```

### ③プロトコル転送コマンドcurlを用いた確認

curlは様々なプロトコルに対応したデータ転送コマンドである

・curlコマンドの基本形

**curl {URL}**

例 curl <http://www.google.com/>

- ・curlコマンドでファイルをダウンロード

**curl -O {URL}**

例 curl -O <http://172.16.1.2/index.html>

- ・curlコマンドでDigest認証

**curl --digest -u {ユーザID:パスワード} {URL}**

例 curl --digest -u linuc:linuc <http://172.16.1.2/secret/index.html>

```
~# curl http://www.alpha.jp
This is TEST Page on host1.alpha.jp
```

④windows10よりcurlコマンドが実装された。(正式にはCURLプログラムの仕様からMicrosoftが作った独自のソフトウェア)

### Apacheのログファイルを確認する

```
~# tail /var/log/httpd/access_log
```

## 特定の権限がある人だけ閲覧できるようにする (アクセス制御)

通常、多くの人から自由に見てもらいたい情報をWeb サービスで公開する。その一方で、特定の権限がある人だけに見せる会員制のホームページを作りたいときがある。

特定の個人だけアクセスできるように限定する手段としてApache にはBASIC 認証と**DIGEST** 認証(ダイジェスト認証)という機能が備えられている。ダイジェスト認証は、ユーザ名とパスワードを暗号学的ハッシュ関数でハッシュ(ダイジェスト)化して送る。これによりBasic認証では防げなかった盗聴や改竄を防ぐことができる。

### アクセス制御の設定

/var/www/html ディレクトリの直下にsecret ディレクトリを作成し、セキュリティ上で安心なDIGEST 認証を設定する。

作業1) /var/www/beta.jp直下にsecretフォルダを作成する

```
# mkdir /var/www/html/secret
```

作業2) /var/www/beta.jp/secret 直下に index.htmlを作る

```
# echo -n 'This is Secret Page on ' | sudo tee /var/www/html/secret/index.html > /dev/null
# hostname | sudo tee -a /var/www/html/secret/index.html > /dev/null
```

作業3) ホストPCのWebブラウザで <http://192.168.1.101/secret> にアクセスできることを確認する。

(下記の例はWebサーバ内でCUI版のブラウザを利用した例)

```
# curl http://localhost/secret/index.html
This is Secret Page on host2.beta.jp
```

作業4) /etc/httpd/conf/httpd.confにDigest認証の設定を追加する

```
<Directory /var/www/html/secret>
    AuthType Digest
    AuthName "Secret Zone"
    AuthUserFile /etc/httpd/.htdigest
    Require user linuc
</Directory>
```

教科書p65参照

作業5) パスワードファイルを作成し、ユーザlinucを追加する

```
# htdigest -c /etc/httpd/.htdigest 'Secret Zone' linuc
Adding password for linuc in realm Secret Zone.
New password:(linuc)
Re-type new password:(linuc)
```

確認

```
# ls -l /etc/httpd/.htdigest
-rw-r--r--. 1 root root 51  5月 29 15:45 /etc/httpd/.htdigest
# cat /etc/httpd/.htdigest
linuc:Secret Zone:eec778bafdac9de277650f169b5156a5
```

作業6) 設定の再読み込みと確認

```
# systemctl restart httpd
```

curlで確認

通常のアクセスでは、接続ができないことを確認する

```
# curl http://www.alpha.jp/secret/index.html

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>401 Unauthorized</title>
</head><body>
<h1>Unauthorized</h1>
<p>This server could not verify that you
are authorized to access the document
requested. Either you supplied the wrong
credentials (e.g., bad password), or your
browser doesn't understand how to supply
the credentials required.</p>
</body></html>
root@host1.alpha.jp /home/admin#
```

ユーザーIDとパスワード込みの接続の場合、アクセスが可能となる

```
$ curl --digest -u linuc:linuc http://www.alpha.jp/secret/index.html
```

アクセスログファイルを確認する

```
$ tail /var/log/httpd/access_log
```

エラーログファイルを確認する

```
$ tail /var/log/httpd/access_log
```

ここまですべてbeta.jpでも行ってください。

#### バーチャルホストの設定

バーチャルホストとは、1台のWebサーバに異なるURL(別々のホスト)でアクセスが可能とすること

http://www.alpha.jp ➡ /var/www/html/index.htmlにアクセス

http://vhost1.alpha.jp ➡ /var/www/html/vhost1/index.htmlにアクセス

http://vhost2.alpha.jp ➡ /var/www/html/vhost2/index.htmlにアクセス

#### ipアドレスレスの名前解決の確認

```
PS C:\Users\hosokawa> ssh teacher
```

```
[admin@teacher ~]$ host vhost1.alpha.jp
vhost1.alpha.jp has address 192.168.1.101
[admin@teacher ~]$ host vhost2.alpha.jp
vhost2.alpha.jp has address 192.168.1.101
[admin@teacher ~]$ host vhost1.beta.jp
vhost1.beta.jp has address 192.168.1.102
[admin@teacher ~]$ host vhost2.beta.jp
vhost2.beta.jp has address 192.168.1.102
[admin@teacher ~]$
```

## VirtualHostディレクティブの設定

su

vi /etc/httpd/conf.d/virtualhost.conf

```
<VirtualHost *:80>
    ServerName      www.alpha.jp
    DocumentRoot    /var/www/html
    ServerAdmin      webmaster@www.alpha.jp
    ErrorLog         /var/log/httpd/error_log
    CustomLog        /var/log/httpd/access_log common
</VirtualHost>

<VirtualHost *:80>
    ServerName      vhost1.alpha.jp
    DocumentRoot    /var/www/vhost1.alpha.jp
    ServerAdmin      webmaster@vhost1.alpha.jp
    ErrorLog         /var/log/httpd/vhost1.alpha.jp-error_log
    CustomLog        /var/log/httpd/vhost1.alpha.jp-access_log common
</VirtualHost>

<VirtualHost *:80>
    ServerName      vhost2.alpha.jp
    DocumentRoot    /var/www/vhost2.alpha.jp
    ServerAdmin      webmaster@vhost2.alpha.jp
    ErrorLog         /var/log/httpd/vhost2.alpha.jp-error_log
    CustomLog        /var/log/httpd/vhost2.alpha.jp-access_log common
</VirtualHost>
```

## 設定の確認

```
# apachectl configtest
AH00112: Warning: DocumentRoot [/var/www/vhost1.alpha.jp] does not exist
AH00112: Warning: DocumentRoot [/var/www/vhost2.alpha.jp] does not exist
Syntax OK
```



## htmlファイルの作成

```
# cd /var/www
# mkdir vhost1.alpha.jp
# mkdir vhost2.alpha.jp
# ls
cgi-bin  html  vhost1.alpha.jp  vhost2.alpha.jp
# echo -n 'This is Virtual Page 1 on ' > /var/www/vhost1.alpha.jp/index.html
# hostname >> /var/www/vhost1.alpha.jp/index.html
# echo -n 'This is Virtual Page 2 on ' > /var/www/vhost2.alpha.jp/index.html
# hostname >> /var/www/vhost2.alpha.jp/index.html
```

## 設定ファイルの読み込み

### 設定の確認

```
# apachectl configtest
Syntax OK
# systemctl reload httpd
```

## ブラウザ経由でのアクセス

今回は、ブラウザ経由ではなく、様々な通信プロトコルでデータの送受信できるcurlコマンドを用いる

```
# curl http://vhost1.alpha.jp
This is Virtual Page 1 on host1.alpha.jp
# curl http://vhost2.alpha.jp
This is Virtual Page 1 on host1.alpha.jp
```

コマンドラインブラウザをlynxを利用して確認してもよい

## ログファイルの確認

```
[root@host1 ~]# ls -l /var/log/httpd/
合計 36
-rw-r--r--. 1 root root 12241 6月 12 16:50 access_log
-rw-r--r--. 1 root root 13278 6月 20 16:56 error_log
-rw-r--r--. 1 root root    71 6月 20 16:57 vhost1.alpha.jp-access_log
-rw-r--r--. 1 root root    0 6月 20 16:56 vhost1.alpha.jp-error_log
-rw-r--r--. 1 root root   142 6月 20 16:59 vhost2.alpha.jp-access_log
-rw-r--r--. 1 root root    0 6月 20 16:56 vhost2.alpha.jp-error_log
[root@host1 ~]#
```