

Apache HTTP Server

Getting start with Apache HTTP Server

1) install apache2

```
$ sudo apt-get install apache2
```

2) test apache2

```
$ apache2 -v
$ sudo /etc/init.d/apache2 stop
$ sudo /etc/init.d/apache2 start
$ sudo /etc/init.d/apache2 restart
$ sudo /etc/init.d/apache2 reload          #reloading web server configurations
```

3) configurations files

- config files “/etc/apache2” (httpd.conf, ports.conf, sites-enabled/ mods-enabled/)
- the structures of configuration files

4) directives of configuration files

```
$ cat /etc/apache2/sites-enabled/000-default #see sample code: apache default file
```

NameVirtualHost IP:port	Normally the IP of the host, and Must exactly match <VirtualHost> directives
<VirtualHost IP:port> <directive a> </VirtualHost/>	Contains directives only for the specified virtual host
ServerName name_of_server	Must be set also in DNS system
ServerAlias *.name_of_server	Specifies aliases for the servername
DocumentRoot /path/to/htdocs	Path to the htdocs
<Directory /any/path> <directive b> </Directory>	Contains directives only for the specified directories
AllowOverride keyword	All: evaluated and every directive is allowed None: Not evaluated Limit: evaluated and allows use of the directives controlling host access
Alias /your_alias/ “real name”	In order to save web files anywhere on the system

5) Virtual Hosts

```
$ cat /etc/apache2/sites-enabled/000-default #see sample code: apache default file
```

Virtual host is a concept to apply different URLs to the same IP and let the same Apache deliver different websites distinguished by the URL. To configure Apache for virtual host, it is common to create for each website a directory /var/www/<website>/htdocs. The directives are setup in /etc/apache2/sites-available/<pagename> need the VirtualHost directive. Example:

```
/var/www/
```

This directory is the document root for the web server. This is where your web sites go.

```
/var/www/uobabylon.org/
```

A directory you create to contain the publicly accessible files for uobabylon.org.

```
$ mkdir /var/www/uobabylon.org
```

```
/var/www/uobabylon.org/index.php
```

An html file you create to test your configuration.

```
$ vi /var/www/uobabylon.org/index.php # you can enter some simple php code for testing i.e. <?php echo “hello world.”?>
```

```
/etc/apache2/sites-available/uobabylon.org
```

This file you create containing the Apache configuration directives to define the virtual domain uobabylon.org. Each file in the sites-available directory defines a different virtual domain.

```
$ cd /etc/apache2/sites-available/
$ sudo su
$ cp default uobabylon.org
```

Update the default configuration file according to your system

```
.....
ServerName tuberlin.org
    DocumentRoot /var/www/uobabylon.org/
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /var/www/uobabylon.org>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>
.....
```

```
/etc/apache2/sites-enabled/uobabylon.org
```

This is a symlink you create with the handy little command `a2ensite uobabylon.org`. The presence of a symlink in the sites-enabled directory has the effect of "enabling" a site defined in the sites-available directory.

```
$ a2ensite uobabylon.org
```

edit file /etc/hosts

```
$ sudo vi /etc/hosts

127.0.0.1    localhost
127.0.0.1    tuberlin.org
127.0.1.1    servername
```

Restart apache server

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

6) basic authentication to web pages with .htaccess
- create .htaccess file in the directory you want to protect

```
$ cd /var/www/php/
$ vi .htaccess
```

```
AuthType Basic
AuthName "this is basic auth system for site PHP"
AuthUserFile /var/www/php/.htusers
Require user ihab
```

```
$ sudo su
$ htpasswd -c /var/www/htdocs/.htusers <testuser>
```

Realtime Online editor

<http://piratenpad.de/>

mediawiki

<http://www.mediawiki.org/wiki/Download>

wordpress

<http://wordpress.org/download/>

Enter the following lines in your terminal, to update the MySQL root password on Linux(Ubuntu).

1) Stop mysql:

```
$ /etc/init.d/mysql stop
```

2) Make sure that all of the MySQL processes are actually stopped or killed. You can check for running MySQL processes with this command:

```
$ ps waux | grep mysql
```

3) If any MySQL processes are still running, they will look similar to the following:

```
11207 32523 ... /bin/sh /usr/local/mysql/bin/safe_mysql ...  
2102 32557 ... /usr/local/mysql/libexec/mysqld --basedir= ...
```

You can kill the processes with the kill command(Note: The second column is the Process ID or PID):

```
kill -9 32523
```

4) Restart mysql with --skip-grant-tables:

```
$ /usr/bin/mysqld_safe --skip-grant-tables
```

5) Open a new session, leave the other open as well, in the new session:

```
$ /usr/bin/mysql
```

6) This should get you in, now:

```
use mysql;
```

7) Once you are using the mysql database, run the following:

```
UPDATE user SET Password=PASSWORD('YOUR_PASSWORD_HERE')  
WHERE Host='localhost' AND User='root';
```

now quit out with quit; and then restart mysql:

```
$ /etc/init.d/mysql restart
```