Setting up WWW

`httpd`: the daemon that provices WWW service.
Several free versions exist, apache is the most common.
Usually port 80, but URLs can include port number.
Port 8080: no need to be root.

Configuration and logs are sometimes kept in a directory subtree.
On ours the configuration directory is `/etc/httpd`
The configuration file is `httpd.conf`
The log files are kept where specified by the configuration file.

Command line options can set configuration file and some options.

Starting the server:
`httpd -f /usr/www/httpd.conf -X`
Use designated configuration file.
Use debug mode, server runs in current terminal.

Usually in non-debug cases the server is started with the `apachectl` command.
At boot the system starts it with `rc.httpd` (which calls `apachectl`); this script can be invoked by hand to start, stop or restart the server.

httpd.conf

```
#  (comment line)
Listen 80
User apache
Group apache
ServerRoot /usr
ErrorLog /var/log/httpd/error_log
ServerAdmin you@example.com
Include /etc/httpd/extra/httpd-userdir.conf
DocumentRoot /srv/http/htdocs
```

Server is running on port 80
Server runs as user:apache, group:apache
`ServerRoot`: prepend `/usr` to relative log and conf paths
Put errors in file: `error_log`
E-mail address of the www administrator
Supplemental features are added by includes
`DocumentRoot`: where the html files live (prepend).
The web page `/xx.html` is found at
`/srv/http/htdocs/xx.html`
(Our `DocumentRoot` is `/var/www/htdocs`)

Access Control (`httpd.conf`)

Principle: permissions are granted and retracted in the order given in the config file.
He who grants/denies last grants/denies best.

```
<Directory />
Options FollowSymLinks
AllowOverride None
Order allow,deny
Deny from all
</Directory>
```

Web access permissions may be specified for each directory Any directive applies to the named directory and any directories under it, unless overridden by a later (hopefully more specific) directive.

Symbolic links occuring in web directories may be followed through this directory. (Can bring directories not under the document root into the web area.)

Access control files are not allowed to do anything

Access denied to this directory and all subdirectories. Lower directories must specifically allow/enable things. Your password file and other files outside the web hierarchy are not web accessible.

`httpd.conf` (access control continued)

```
<Directory /var/www>
Options Indexes FollowSymLinks
AllowOverride FileInfo
Order allow,deny
allow from all
deny from .hacker.com
</Directory>
```

Index files are used, symbolic links are followed.

Access control files are allowed to used the `FileInfo` directive. Allowed overrides are `All`, `None` or any combination of `FileInfo`, `AuthConfig`, `Limit`.

Web access is allowed to this directory and all subdirectories not mentioned in other directives except for browsers from the `hacker.com` domain.
`Order` is important, reversing the order would allow from all because the allow would be processed last and "last is best".

Reminder: must follow the `<Directory />` directory to have any effect at all.

`httpd.conf` (more access control)

```
<Directory /home/*>
AllowOverride All
Options Indexes FollowSymLinks
Order deny,allow
Deny from all
Allow from .cecs.csulb.edu
</Directory>
```

`/home/*`: you can use wild cards to specify directories, in fact you can use extended regular expressions.

The access control file is allowed to contain any access control directive.

These directories are available only from machines within the department.

```
DirectoryIndex index.html index.shmtl index.cgi
```

If the browser requests a directory this is the file that will be used to answer that request.
If none of these exist you get a text listing of the files in the directory.

`httpd.conf` (more)

```
<FilesMatch "^\.ht">
Order allow,deny
Deny from all
Satisfy All
</FilesMatch>
```

Web access to specific files can be explicitly denied.
Here any file starting with `.ht` is disallowed.
This directive must be inside a directory directive (applies to the directory and subs) or and htaccess file (covered later).

`AddHandler cgi-script .cgi`
CGI scripts (actually any executable) can be placed in any web directory rather than just the script directory (automatically enabled for files in the script directory)
To be executed, they must be marked executable (`chmod`) and the file name must end in `.cgi`.
"`ExecCGI`" must be an "`Options`" for that directory

```
AddType text/html .shtml
AddOutputFilter INCLUDE .shtml
```

Server side scripts may be enabled.
Must have "`Options Includes`"

Extra Features

Extra features are accessed using the `Include` directive

`http-userdir.conf`

If this is present, access is allowed to users personal web pages (e.g. `http://server/~bob`)

```
UserDir public_html
```

Web access is not supplied to the users home directory, only to the users web directory. This directive gives the name of the users web directory.

Warning, the include file has a `Directory` directive, it works, but if you want to allow users to have scripts or server side includes, you will need to edit it.

`mod_php.conf`

Include this if you want to allow php scripts.

`httpd-ssl.conf`

Include this if you want to allow ssl (secure) access to your web pages.

Extra Features (more)

`http-vhosts.conf`

You are supplying web service for several domains (other than the name of your host).
A separate directory is designated for the web pages of each of the vitual hosts. Separate entries are designated for the logfiles and the server administrator.

`http-default.conf`

Every value given in here is a default value used by the web server.
The include won't modify anything, unless you edit the file.
(The file gives you the options to override the server defaults.)
There are several entries pertaining to the persistence of connections from clients.
There is an `AccessFileName` entry that allows you to change the name of the access file from `.htaccess`

`http-mpm.conf`

Lets you manage the behavior of the server.
Detail on next slide.

Server Behavior

```
StartServers 5
MaxClients 64
MinSpareServers 5
MaxSpareServers 10
MaxRequestsPerChild 16
```

This is a multi-process concurrent connection-oriented server (process based), one `httpd` process exists per client

For efficiency: after a client disconnects, the process may be reallocated to a new client (instead of killing the old one and starting a new one).

`StartServers`: fork this many at boot

`MaxClients`: limit the number of clients to prevent thrashing.

`MinSpare` if you don't have this many spares, fork new processes at the rate of 1/sec.

`MaxSpare`: if you have more than this many spares, kill the extras.

`MaxRequestsPerChild`: limits number of requests per client to prevent a few users from locking out others. 0 indicates the number of requests is unlimited.

Sample .htaccess

```
AuthName "MyCluster"
AuthType Basic
AuthUserFile /full/path/of/pwd/file/
require valid-user
```

An account and password is required for web access of this directory.
After entering a password any other access will be automatically granted (without asking for a password) to any other secured directory with an `AuthName` of `MyCluster`.

`Basic` implies simple password encrypting.

The name of the password file is defined by `AuthUserFile`.

The format of the password file is "username:crypt"
A non-rooted path can be given for the password file but may not work some servers (for example, `~bob/pwd` might not work).

`require` limits whose password will be checked.
`require sue bob` will only check the password for bob and sue; all other users will be automatically denied.
The reserved word `valid-user` will grant permission to anyone with a name and password in the specified password file.

htpasswd

The httpd password file is not the system password file.

The `htpasswd` command is used to create/add/modify passwords in this file.

Command Form:

```
htpasswd -c users bob
```

A password file called `users` is created and an entry for bob will be added. You will be prompted for the password bob will use.

The `-c` option causes a new password file to be created. Do not use this if the files already exists.

Bob does not need to be a user (`/etc/passwd`).

The password is independent of any other password for bob.

Cautions

Your htaccess and htpassword files should not be readable from the web.

This is done for the default file of `.htaccess` and `.htpasswd` by a `FilesMatch` directive in the default version of `httpd.conf`

Recommend use of the default names.

No way to prevent users from using another name and making their password file vulnerable.

Ubuntu

Include file are found in the `sites_available` directory.

Default `httpd.conf` includes everything found in the `sites_inuse` directory.

An "extra" is enabled by making a softlink in the "inuse" directory that refers to a file in the "available" directory.

(`ln -s ../...`)