

Apache

Apache Structure

- Parent service with multiple child processes
 - Each child can serve different pages at the same time
- Control is like a manager with a team
 - Interactions through the parent
 - Parent manages children
- Fault tolerant, high performance

Interacting with Apache

- **apachectl** - script to control service
 - Some distributions provide another layer on top of apachectl

Apache Actions

- **status**
 - Show what Apache is doing
 - May need to be enabled manually
- **start**
 - Starts parent web server and spawns children
- **stop**
 - Parent tells children to shut down right now
 - Parent process shuts down
 - Connections terminate
- **graceful-stop**
 - Requests in progress are completed
 - Parent process shut down

- **restart**

- Parent does not fully exit
- Children are stopped
- Configuration re-read
- Children spawned with new config
- Required for low-level configuration changes

- **graceful - AKA reload**

- Children finish what they're doing
- Graceful shutdown
- Config re-read
- Children spawned
- Required for minor configuration changes

Control Requires Elevated Privileges

- **root**

- Linux super user

- **sudo**

- provides elevated privileges to regular accounts

Configuration Overview

- <http://httpd.apache.org/docs/current/configuring.html>
- **Stored in plain text files**
- **Typically named httpd.conf**
- **Location set at compile time**
 - Possible to override
- **Changes only recognized when started**

Apache Config by Distribution

- **Fedora and CentOS** - /etc/httpd/conf/httpd.conf
 - One monolithic file
- **Debian and Ubuntu** - /etc/apache2/apache2.conf
 - Related, smaller configuration files
 - One per site
 - One per group of configuration

```
$ apachectl -V
```

Finding the Apache Configuration

- Start with default location in distribution
- apachectl
- Search entire file system

Directory-Level Configuration

- **.htaccess files**
 - Plain text
 - Similar to Apache configuration
- **Intended to provide security for directories**
 - Evolved to extending configurations
- **Main configuration can restrict overrides.**
 - Useful for shared hosts, untrusted users

.htaccess Disadvantages

- Increase security risks
- Slower than just server config

Read every page load

Searched for each time

Apache httpd is Modular

- System of plugins that add functionality
- Static modules included upon compilation

Loaded every time Apache is started

- Shared modules added without compilation

Dynamically loaded

Can be turned on and off

Slower than static

Modules Have Directives

- Configure additional functionality
- If config has module directives and module is missing...
Server won't start!
- Wrap module configuration in IfModule
`<IfModule mod_ssl>`

Debian Module Configuration

- **mods-enabled**

Symbolic link to mods-available

- **a2enmod / a2dismod**

Enables/disables Apache modules

Quick Commands

\$ apachectl -V	List configurations and version
\$ apachectl -t	<i>Test the configurations</i>
\$ apachectl -t -D DUMP_VHOSTS	List of the virtual hosts and definition file names
\$ apachectl -t -D DUMP_MODULES	List the all modules of apache configuration

STEP # 1 - Working on Configurations

- Discover 'HTTP_ROOT' -> where http config files present
- Discover "DocumentRoot" for html files are staying

Find the httpd/apache2 - Configuration in Linux

```
$ sudo find / | grep "httpd\conf"
```

```
$ sudo find / | grep "apache2\conf"
```

Search recursively for text 'DocumentRoot'

```
$ grep -Ri "DocumentRoot" .
```

Find the list of httpd module

```
$ apachectl -t -D DUMP_MODULES
```

Identify modules Load/Unload . Eg: status module

```
$ cd /etc/apache2/mods-available
```

```
$ cat status.conf
```

...

```
$ cat status.load
```

```
LoadModule status_module /usr/lib/apache2/modules/mod_status.so
```

```
$ ls -la /etc/apache2/mods-enabled
```

```
$ sudo a2dismod status
```

```
$ sudo service apache2 graceful
```

```
$ sudo a2enmod status
```

STEP # 2 - Debugging Log files and Analyzing

- Discover 'ErrorLog' inside '/etc/apache2'

```
$ grep -Ri "ErrorLog" /etc/apache2
```

```
$ grep -Ri "export APACHE_LOG_DIR" /etc/apache2
```

```
$ cd /var/log/apache2
```

```
<VirtualHost 10.0.2.15:80>
    ServerAdmin username@example.com
    ServerName alice.example.com

    <Directory /srv/web/>
        Order allow,deny
        Allow from all
        Require all granted
    </Directory>

    DocumentRoot /srv/web

    ErrorLog ${APACHE_LOG_DIR}/alice-error.log
    LogLevel info
    CustomLog ${APACHE_LOG_DIR}/alice-access.log combined
</VirtualHost>
```

Analyzing - Log files

AWStats

- Real-time access log analyzer
- <http://www.awstats.org/>
- Available as a package on most Linux distributions
- Web interface with graphs and summaries
- Can be a security risk if available publicly

Restrict access to only those who should see it.

GoAccess

- Real-time access log analyzer
- Terminal application
 - Easier to secure
- Lightweight and fast
- Many features
- <http://goaccess.prosoftcorp.com/>

Log Aggregators

- graylog2.org
- logstash.net
- Combine log files; interface for sharing reports
- Suited for multiserver environments

Virtual Hosting

- Serve content for multiple domains from same server
- Commonly used by shared web hosts
- Cost-effective

Name-Based Virtual Hosting

- Routes requests based on domain name
- Easiest to implement
 - SSL challenges

IP-Based Virtual Hosting

- Separate IP per site
- Solves SSL challenges
- Costlier
- Greater technical overhead

Configurations

```
user@apache:/etc/apache2$ cat /etc/apache2/sites-available/alice.conf
<VirtualHost 10.0.2.15:80>
    ServerAdmin username@example.com
    ServerName alice.example.com

    <Directory /srv/web/>
        Order allow,deny
        Allow from all
        Require all granted
    </Directory>

    DocumentRoot /srv/web
</VirtualHost>
```