APACHE

Isabel Odida
2019

About this presentation

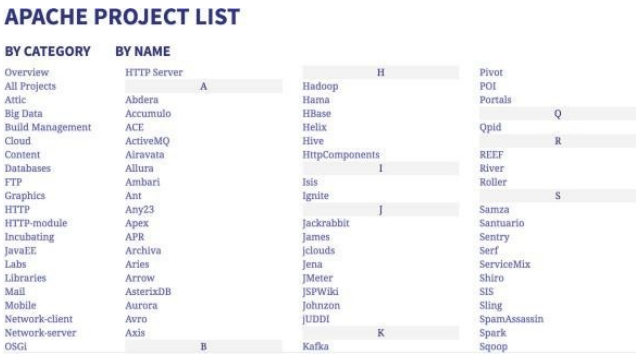
Based on a previous talk by Joel Jaeggli with thanks!

You can access this presentation at:

- Online: http://afnog.github.io/sse/apache/
  Local: http://www.ws.afnog.org/afnog2019/sse/apache/Apachepresentation.pdf
- Github: https://github.com/afnog/sse/blob/master/apache/presentation.md
- Download PDF:
- http://www.ws.afnog.org/afnog2018/sse/apache/Apachepresentation.pdf

What is Apache?

- An HTTP server (web server)
-



APACHE PROJECT LIST

BY CATEGORY **BY NAME**

Overview
All Projects
Attic
Big Data
Build Management
Cloud
Content
Databases
FTP
Graphics
HTTP
HTTP-module
Incubating
JavaEE
Labs
Libraries
Mail
Mobile
Network-client
Network-server
OSGi

HTTP Server

A
Abdera
Accumulo
ACE
ActiveMQ
Airavata
Allura
Ambari
Ant
Any23
Apex
APR
Archiva
Aries
Arrow
AsterixDB
Aurora
Avro
Axis

B

H
Hadoop
Hama
HBase
Helix
Hive
HttpComponents

I
Isis
Ignite

J
Jackrabbit
James
jclouds
Jena
JMeter
JSPWiki
Johnzon
jUDDI

K
Kafka

Pivot
POI
Portals

Q
Qpid

R
REEF
River
Roller

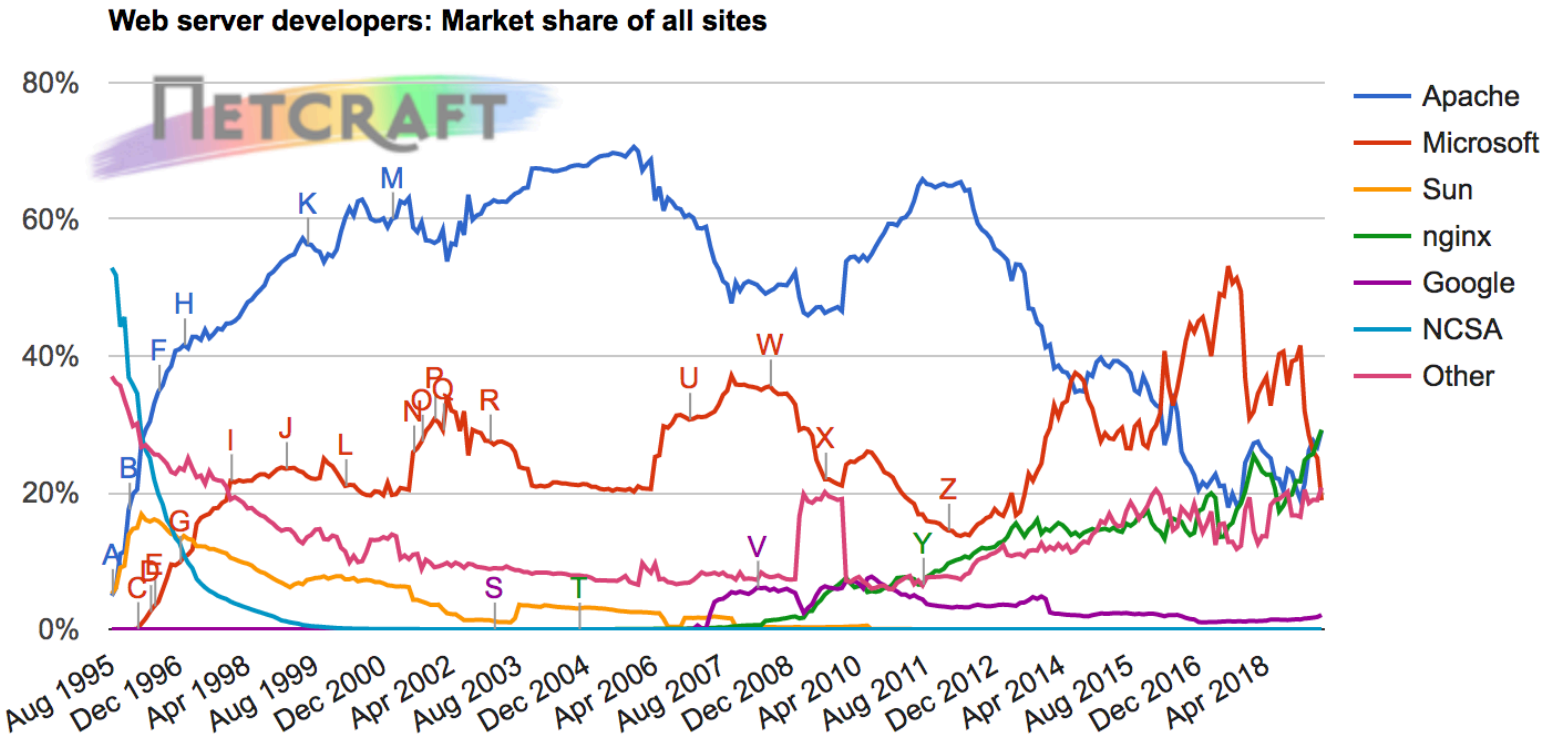
S
Samza
Santuario
Sentry
Serf
ServiceMix
Shiro
SIS
Sling
SpamAssassin
Spark
Sqoop

A foundation supporting several web-related software projects

For clarity it might help to talk about "Apache Server" to mean the HTTPD server.

Other HTTP servers

What other HTTP (web) servers are commonly used?



Which one to use?

- Apache
  - Popular, well-documented, flexible, secure, big, slow, heavy, SSL support, PHP integration.
- Nginx
  - Increasingly popular, quite well-documented, very fast, reverse proxy, SSL support, no PHP.
- Lighttpd
  - Simple, fast, no PHP.
- Thttpd
  - Tiny, fast, no PHP.

Apache Features

Server Side Programming Language Support

- Apache supports some common language interfaces which include Perl, Python, Tcl, and PHP.
  - It also supports a variety of popular authentication modules like mod_auth, mod_access, mod_digest and many others.

IPv6 Support

- On systems where IPv6 is supported by the underlying Apache Portable Runtime library, Apache
  - gets IPv6 listening sockets by default.

Virtual Hosting

- Apache will allow one installation instance to serve multiple websites. For instance one Apache installation can serve sse.afnog.org, ws.afnog.org etc
  - Simplified configuration (!)

More at: http://httpd.apache.org/

Virtual Hosting

What does it mean?
Apache support virtual hosting (deciding which website to display) using:

- Name based virtual hosts
- IP based virtual hosts
-

PHP and MySQL

- Many web applications written in PHP and using a MySQL database.
- Relatively easy to deploy under Apache (and most web hosting).
- We will install the necessary software shortly.

Install Apache

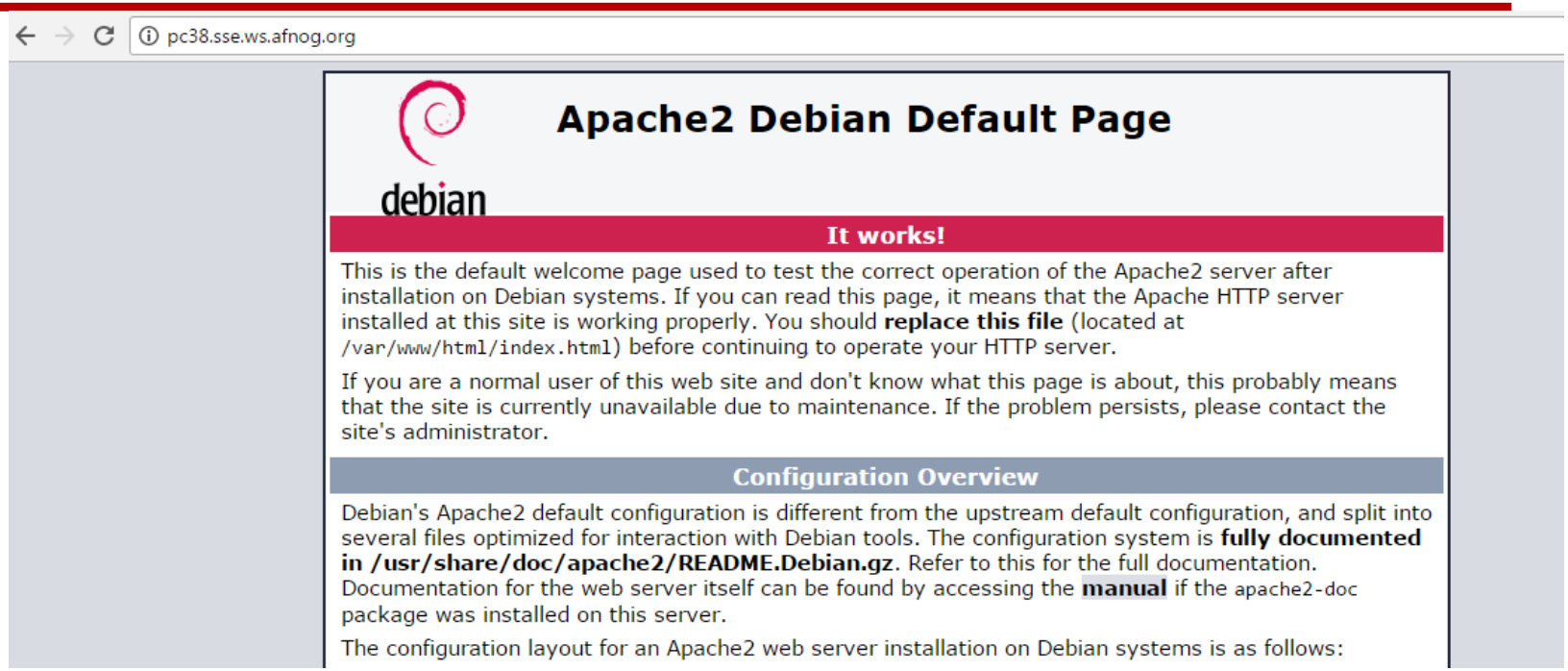
```
sudo apt install apache2
```

How do you test that it worked?

```
telnet localhost 80
```

```
root@pc38:/home/afnog# telnet localhost 80
Trying ::1...
Connected to localhost.
```

And visit http://pcXX.sse.ws.afnog.org in your browser.



What content is it serving? How do we change it?

`/var/www/html/index.html`

Enable and test IPv6

Set your IPv6 address to match your IPv4 address (replace `XX` with your PC number plus 100):

```
$ sudo ip -6 addr add 2001:43f8:220:219::XX/64 dev ens3
```

Then add your default route for IPv6:

```
$ sudo ip -6 route add default via 2001:43f8:220:219::1
```

On the above if you get the error message `RTNETLINK answers: File exists`, it means that the gateway is already in place, as it was auto-configured.

Test your IPv6 connectivity:

```
$ ping6 www.google.com
```

Then browse your IPv6 address at http://[2001:43f8:220:219::XX] (the square brackets are deliberate and essential!).

Apache configuration files

```
* /etc
* /apache2
* apache2.conf
* ports.conf
* conf-available
* *.conf
* conf-enabled
* symlinks to mods-available for services which are enabled
* mods-available (and mods-enabled)
* *.load
* *.conf
* sites-available (and sites-enabled)
* 000-default.conf
* default-ssl.conf
* /var/www/html (content)   * index.html (the test page)
```

https://httpd.apache.org/docs/2.4/configuring.html

Starting Apache

- Startup scripts are located in `/etc/init.d/`
  - `/etc/init.d/apache2 start`
  - `Service apache2 start`

- Other useful commands:
  - `/etc/init.d/apache2 stop`
  - `/etc/init.d/apache2 restart` (stop+start)
  - `/etc/init.d/apache2 reload` (graceful reload config)

Install MySQL

```
$ wget http://repo.mysql.com/mysql-apt-config_0.8.9-1_all.deb
```

```
$ sudo dpkg -i mysql-apt-config_0.8.9-1_all.deb
```

During installation of MySQL apt config package, It will prompt to select MySQL version to install. Select the MySQL 5.7 or 5.6 option to install on your system.

```
$ sudo apt update
```

```
$ sudo apt install mysql-server
```

When the mysql-server prompts for a password to be entered use 'afnog' as the password. If not prompted, don't worry, we will set it later.

Install PHP

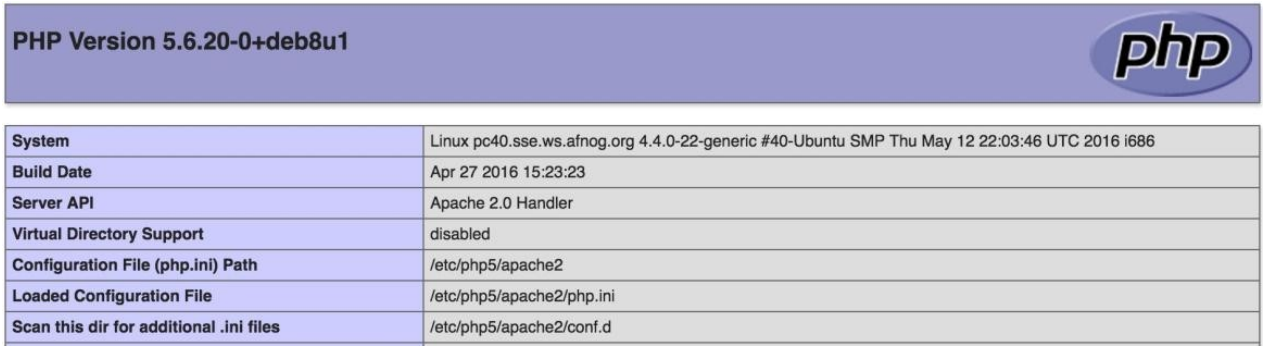
```
$ sudo apt install php7.0 libapache2-mod-php7.0 php7.0-mysql php7.0-gd
php7.0-opcache
```

Testing PHP

Create the file `/var/www/html/test.php` with the following contents:

```
<?php echo phpinfo(); ?>
```

Load it in your browser at http://pcXX.sse.ws.afnog.org/test.php. You should see this:



PHP Version 5.6.20-0+deb8u1

System	Linux pc40.sse.ws.afnog.org 4.4.0-22-generic #40-Ubuntu SMP Thu May 12 22:03:46 UTC 2016 i686
Build Date	Apr 27 2016 15:23:23
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d

Securing MySQL

Please read the instructions and use the letters "y" or “n” on the keyboard.

```
$ sudo mysql_secure_installation
```

The password for MySQL is probably `afnog` (unless you entered a different password during the installation above).

```
Enter current password for root (enter for none):
OK, successfully used password, moving on...

Remove anonymous users? [Y/n] y
... Success!
Disallow root login remotely? [Y/n] n
... Success!
Remove test database and access to it? [Y/n] y
Reload privilege tables now? [Y/n] y

... Success!


Cleaning up...
```

Testing MySQL

Log in to mysql console to check if the password was set properly using command below.

```
$ mysql -u root -p
Password:
```

Type the password at the prompt. Then you should see a `mysql>` prompt, which means that you authenticated successfully and can enter SQL commands.

FIN

Any questions?

(yeah, right!)