

# FAMP

---

FreeBSD/Apache/MySQL/PHP

# Outline

---

- ❑ Introduction
  - Apache
  - MySQL
  - PHP
- ❑ Installation and Administration
  - MySQL
  - Apache
  - PHP
- ❑ Appendix
  - phpMyAdmin
  - lighttpd
  - FastCGI

# OpenSource Web Server

---

- Lighttpd
- Apache
- Nginx

# Lighttpd

---

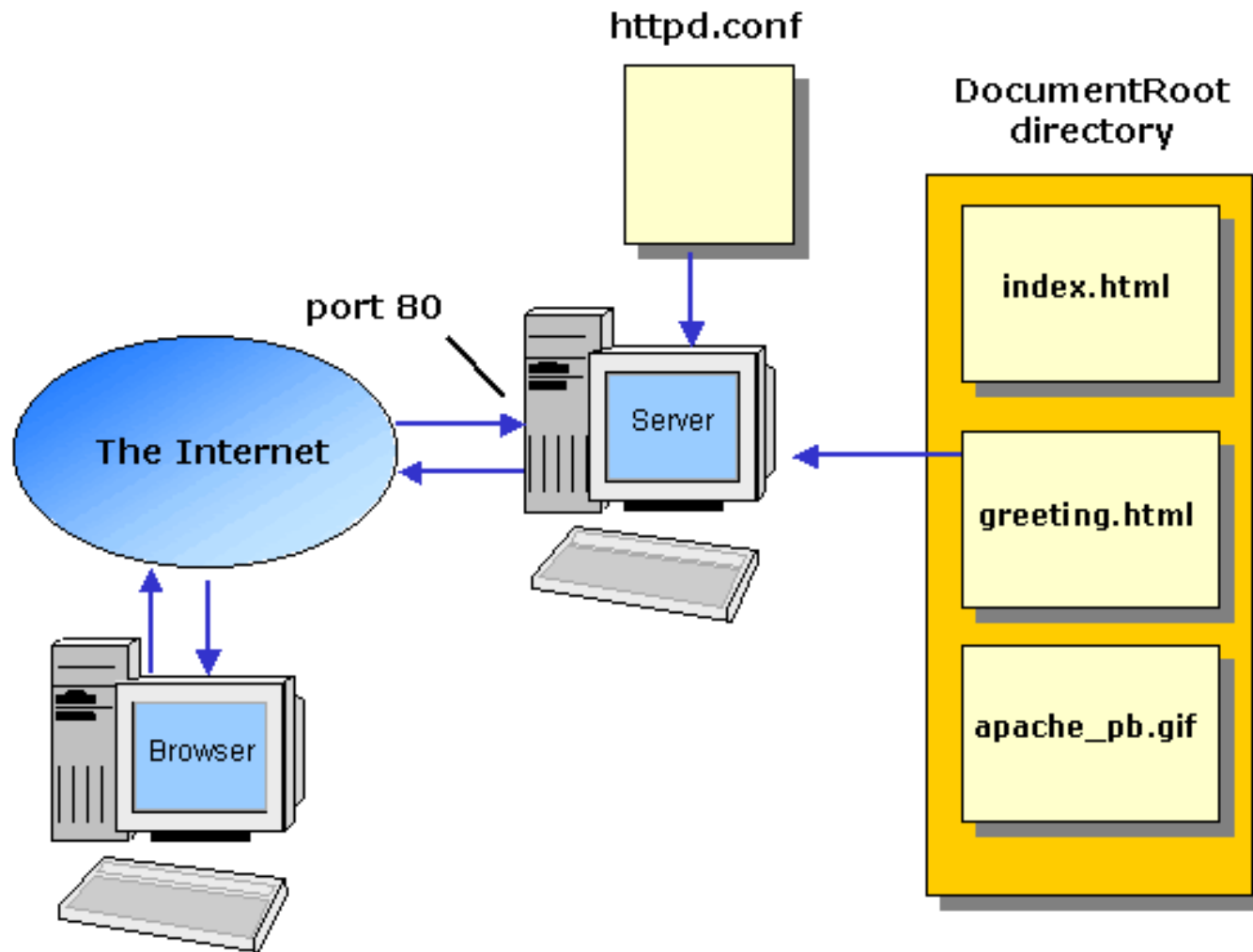
- ❑ Low resource (cpu & memory) usage
- ❑ Supported
  - FastCGI, CGI, Auth, Compress, URL Rewrite, Alias, chroot, vhost
- ❑ <https://www.lighttpd.net/>

# Apache

---

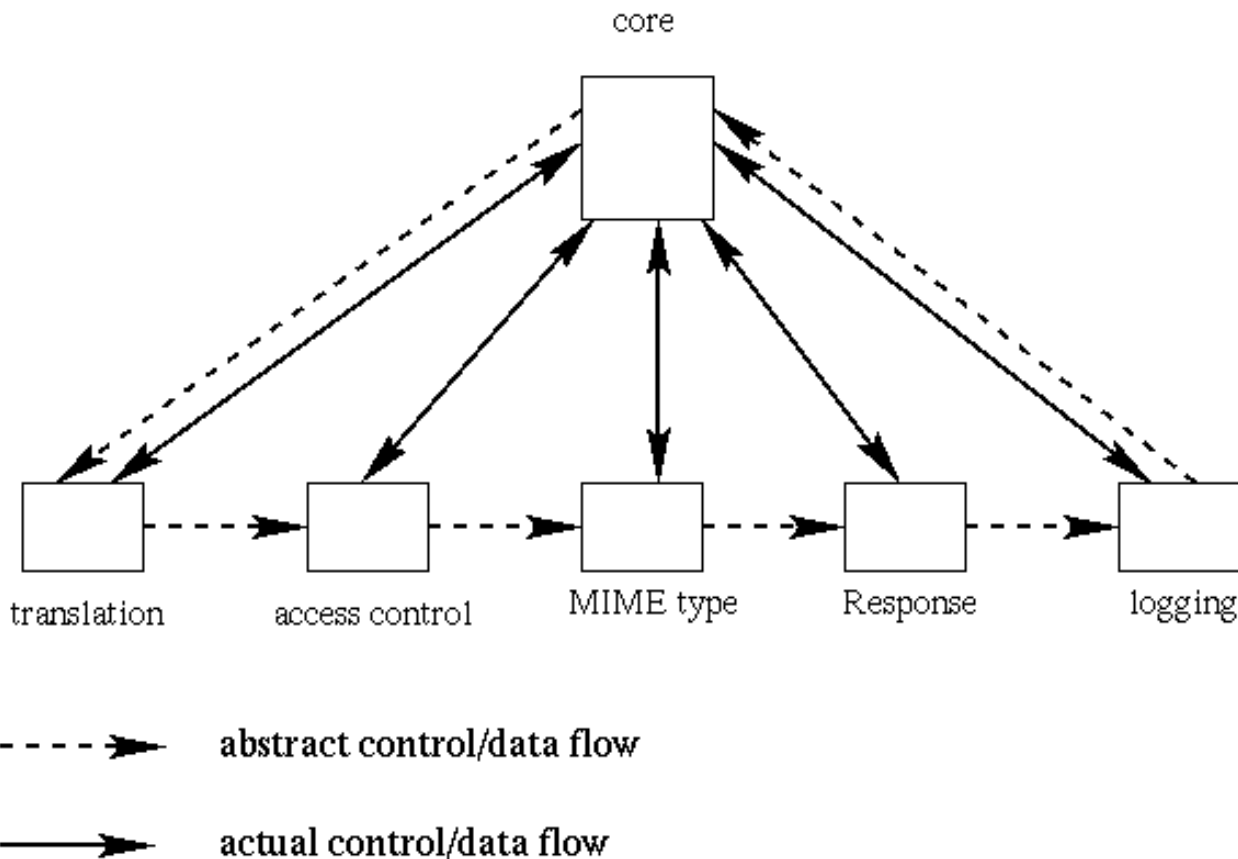
- ❑ Apache Software Foundation:  
<https://www.apache.org/>
- ❑ Apache HTTP Server Project:  
<https://httpd.apache.org/>
- ❑ Web httpd server that
  - HTTP/1.1
  - Modular design
  - Can be customized by writing modules using Apache module API
  - Freely available cross many platforms
- ❑ Two main parts
  - Core: implement basic functions and provide the interface for Apache modules
  - Modules: extend or override the function of Core
    - Example: Access control, logging, CGI, proxy, cache control, PHP...

# How Apache Works – request and response

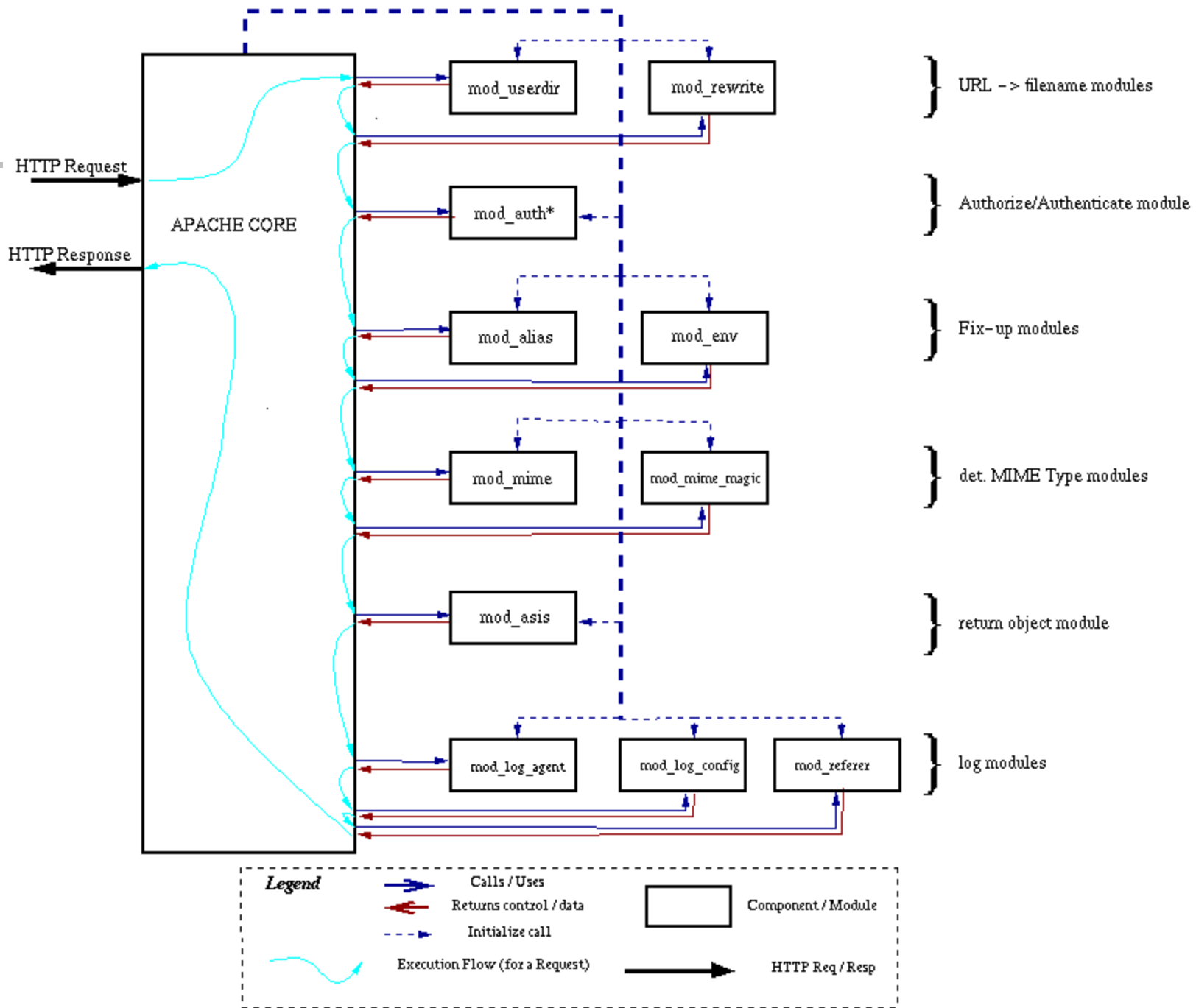


# How Apache Works – Each request-response

- Apache breaks client request into several steps which are implemented as modules

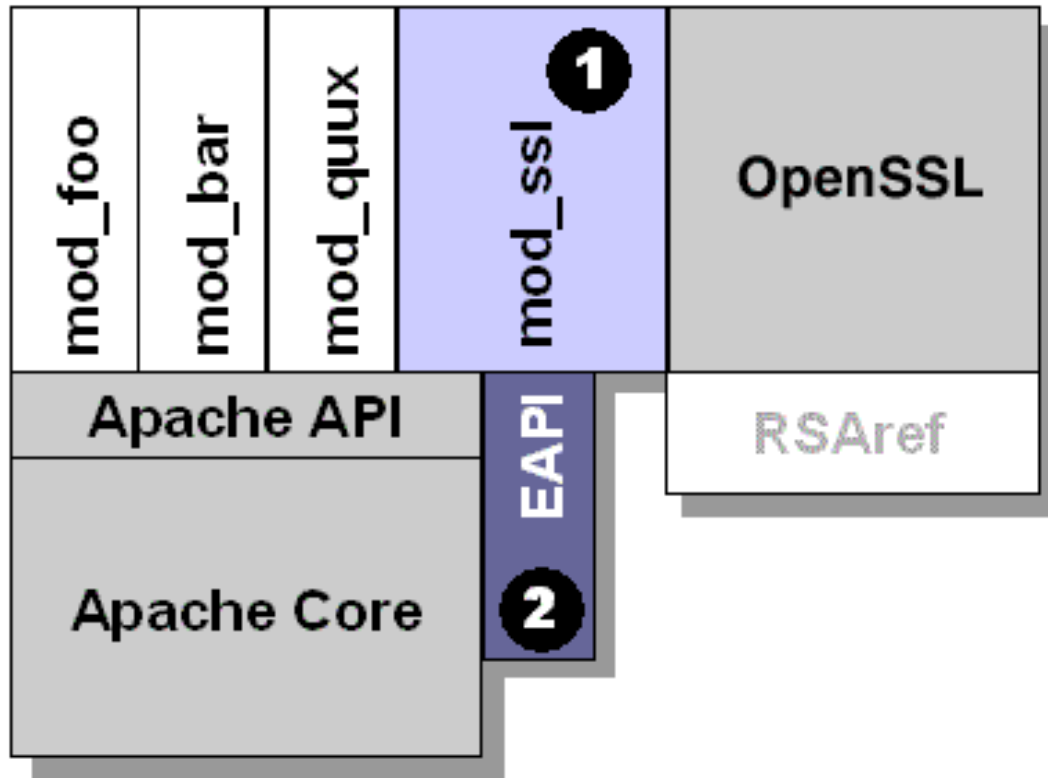


# A p a c h e D e t a i l





# Apache with mod\_ssl



# Nginx –

the High-Performance Web Server and Reverse Proxy

---

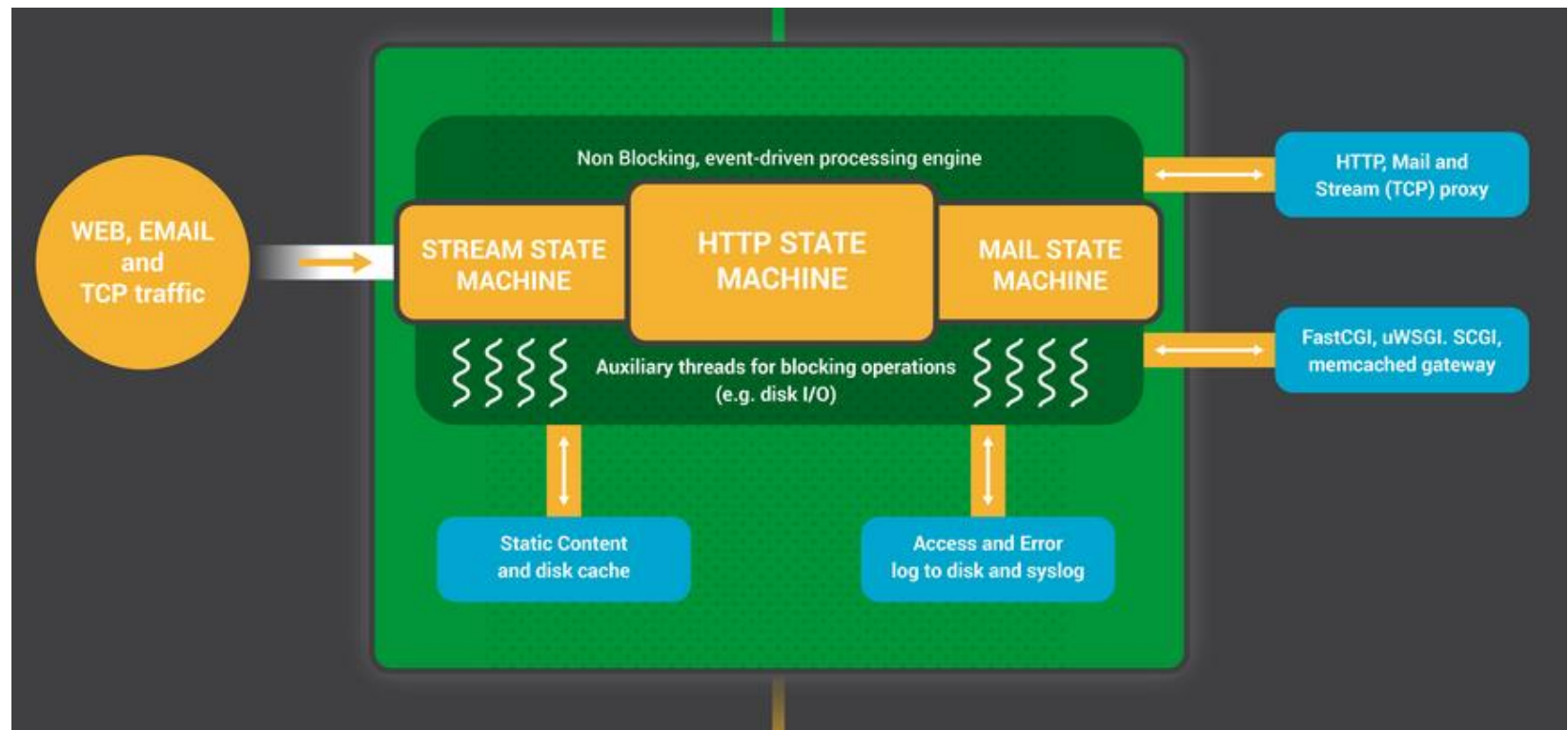
- ❑ From Russia
- ❑ HTTP/2 Supported
- ❑ Ability to handle 10,000 simultaneous connections with a low memory footprint

# The Nginx Process Model

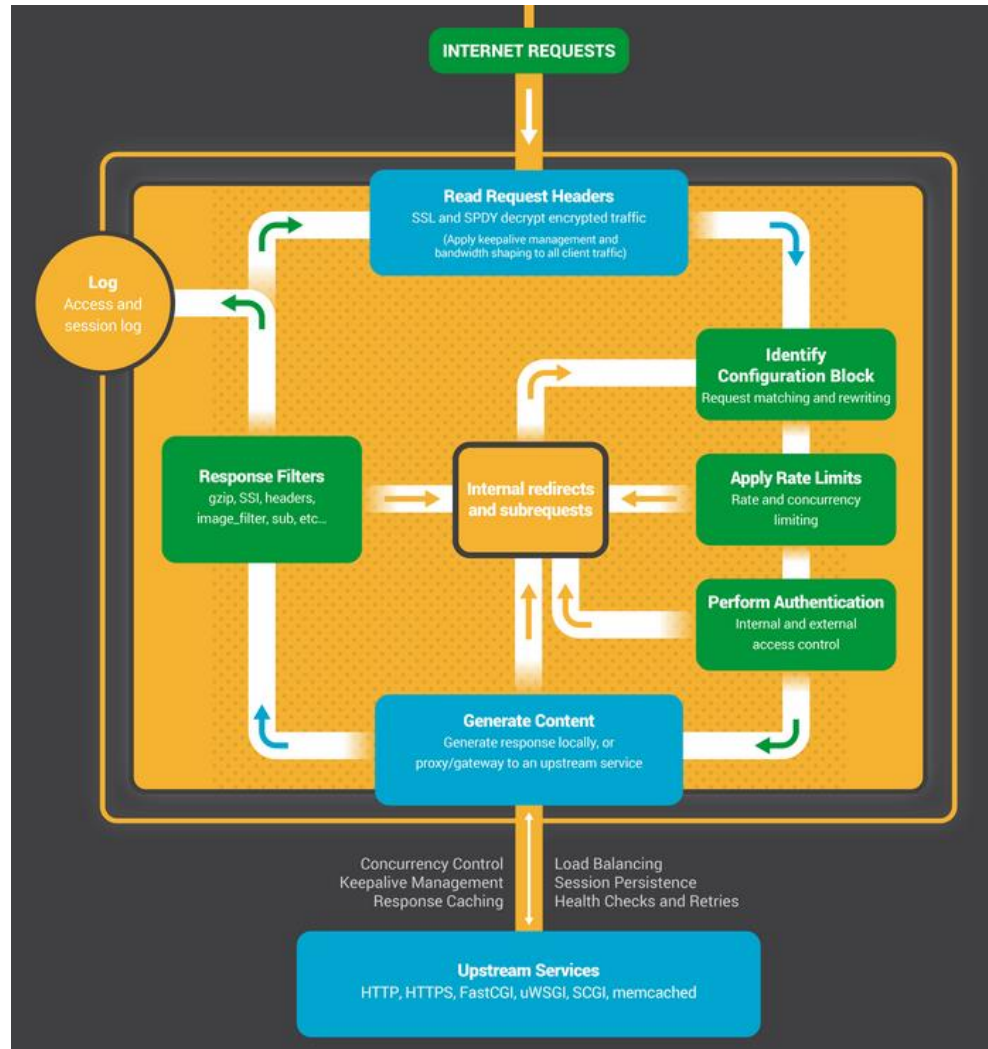


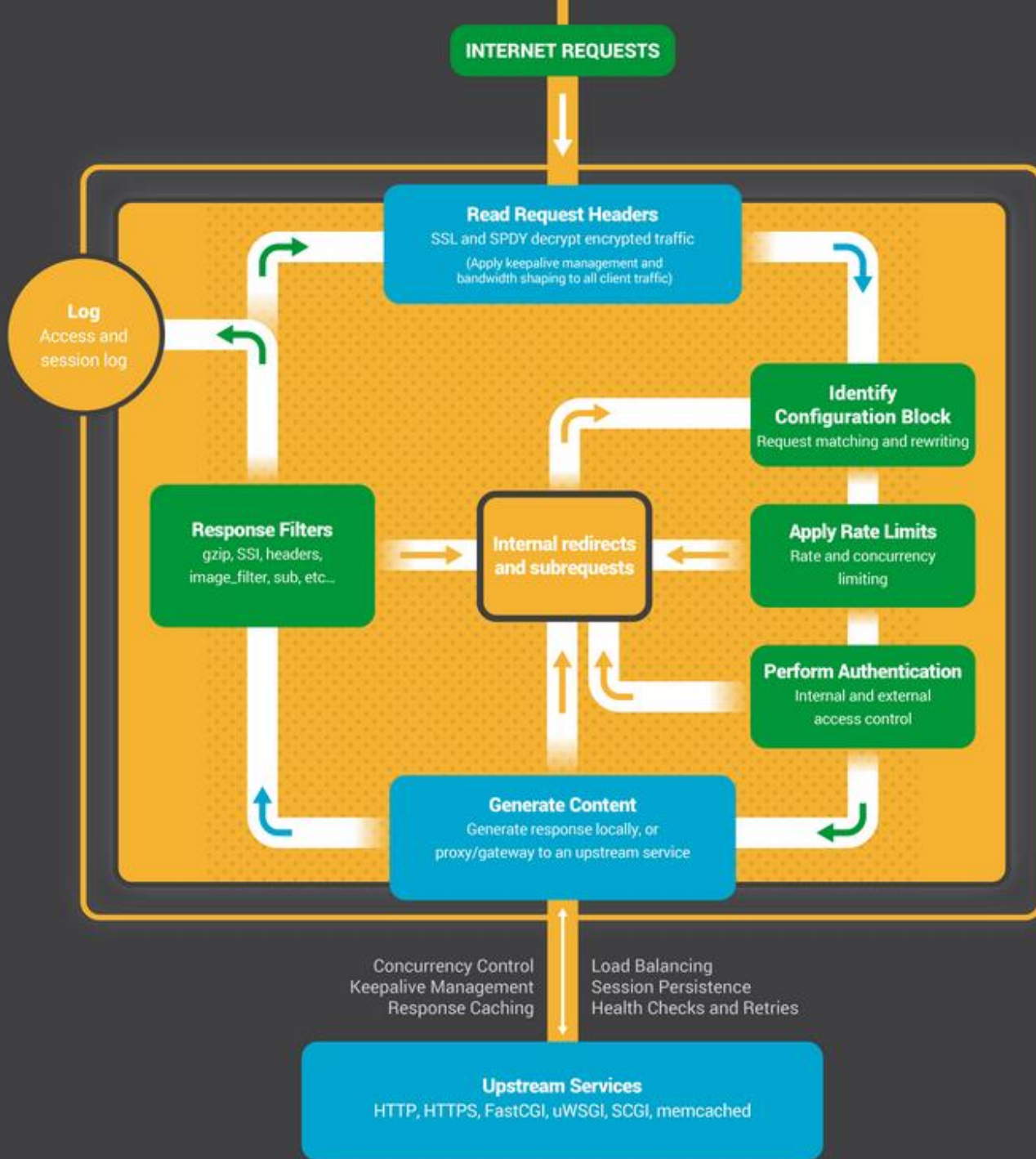
```
# service nginx restart
* Restarting nginx
# ps -ef --forest | grep nginx
root    32475      1  0 13:36 ?        00:00:00 nginx: master process /usr/sbin/nginx \
        -c /etc/nginx/nginx.conf
nginx   32476 32475  0 13:36 ?        00:00:00 \_ nginx: worker process
nginx   32477 32475  0 13:36 ?        00:00:00 \_ nginx: worker process
nginx   32479 32475  0 13:36 ?        00:00:00 \_ nginx: worker process
nginx   32480 32475  0 13:36 ?        00:00:00 \_ nginx: worker process
nginx   32481 32475  0 13:36 ?        00:00:00 \_ nginx: cache manager process
nginx   32482 32475  0 13:36 ?        00:00:00 \_ nginx: cache loader process
```

# Inside the Nginx Worker Process



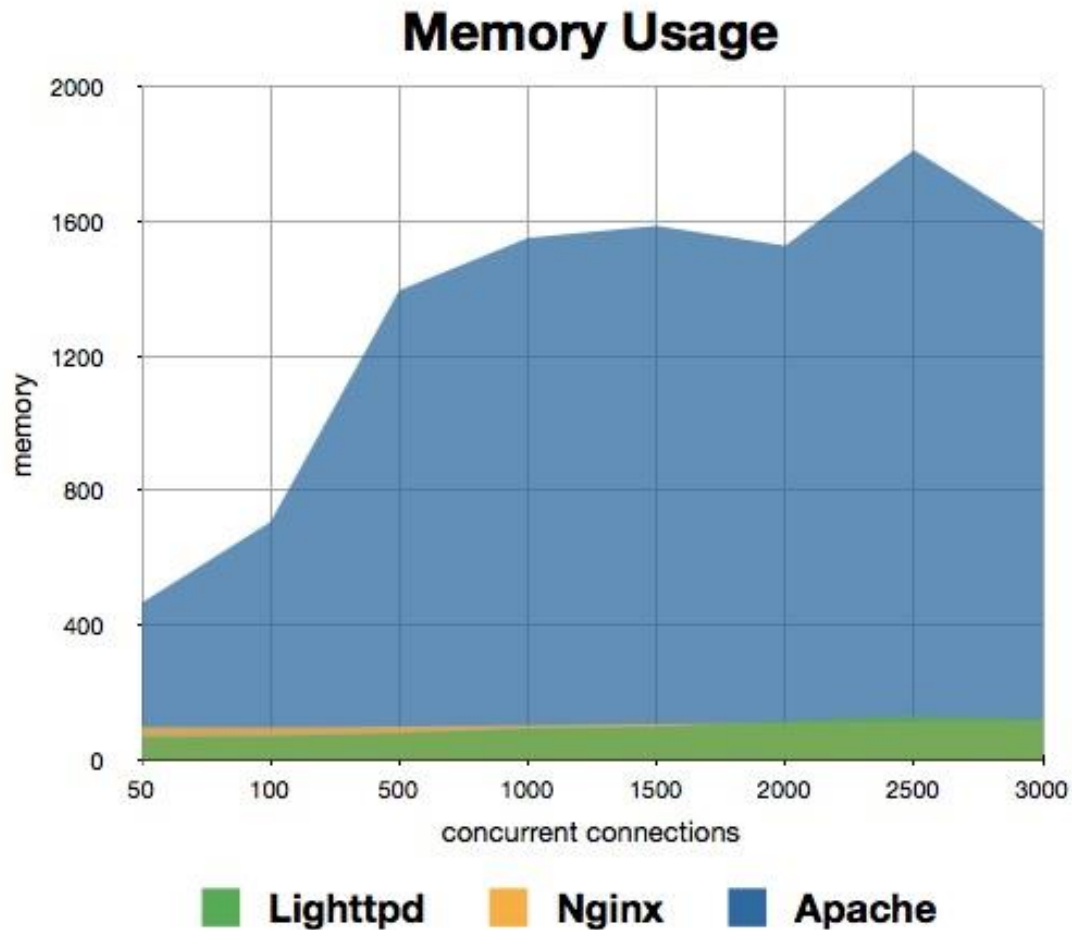
# How Nginx Works





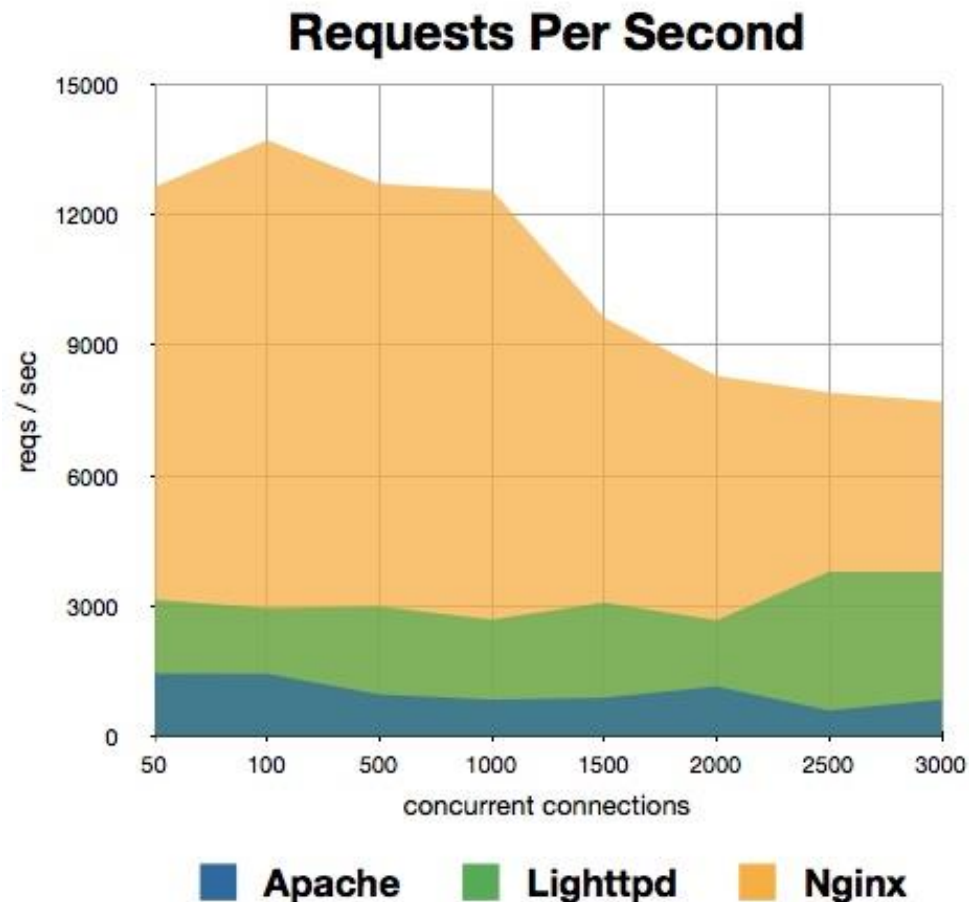
# Performance Comparison

## ❑ Memory usage



# Performance Comparison

## ❑ Requests per second





# Some of benchmark

---

## ❑ Throughput

- Layer 4 Throughput
- Layer 7 Throughput

## ❑ Concurrent Sessions

## ❑ Connections per second(CPS)

- Layer 4 CPS
- SSL CPS
  - SSL key 1024/2048/4096
- Layer 7 CPS (1 HTTP/con)

## ❑ Requests per second (RPS)

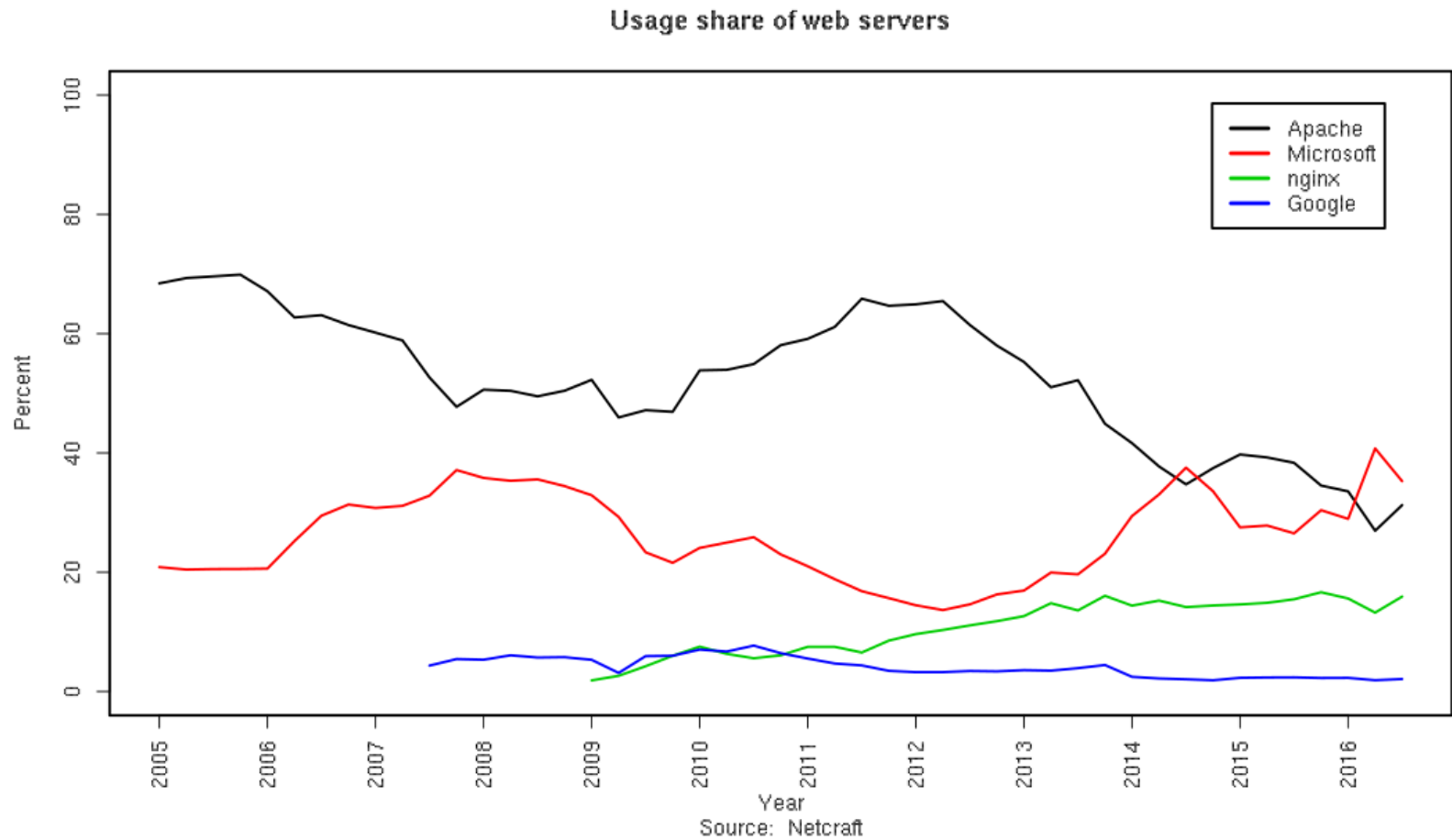
- Layer 4 HTTP RPS
- Layer 7 HTTP RPS

# Others Web Server

---

- IBM WebSphere Application Server (WAS)
- SAP EAServer (Sybase)
- Microsoft IIS
- Oracle WebLogic
  
- Google Servers
- Tomcat
- Node.js
- IdeaWebServer
- Tengine
- Cowboy
- LiteSpeed

# Usage share of web servers



# MySQL (1)

---

- ❑ SQL (Structured Query Language)
  - The most popular computer language used to create, modify, retrieve and manipulate data from **relational database** management systems.
  - Introduction to SQL:  
<http://www.1keydata.com/tw/sql/sql.html>
    - In Chinese.
- ❑ A **multithreaded, multi-user, SQL** Database Management System.
- ❑ Owned and sponsored by a Swedish company MySQL AB, acquired by Sun Microsystems 2008.
- ❑ Acquired by Oracle Corporation 2009.
- ❑ Before acquire Monty Widenius fork GPL-only to MariaDB
- ❑ Official Site: <https://www.mysql.com>
- ❑ Documentation: <https://dev.mysql.com/doc>

# MySQL (2)

---

## □ Features:

- Writing in C/C++, tested by many compilers, **portable to many platforms**.
  - AIX, FreeBSD, HP-UX, Linux, Mac OS, Solaris, Windows, ...etc.
- Providing APIs for C/C++, Java, Perl, PHP, Python, Ruby, Tcl, ...etc.
- **Multi-threaded** kernel, supporting systems with multiple CPUs.
- Optimized algorithm for **SQL** Query.
- Multi-Language (coding) Supports.
- Lots of connecting method: TCP/IP, ODBC, JDBC, Unix domain socket.
- **Free Software** (GNU General Public License version 2)
- Popular for web applications

# PHP

---

## ❑ PHP: Hypertext Preprocessor

- A widely-used Open Source general-purpose scripting language.
- Originally designed to create dynamic web pages, PHP's principal focus is server-side scripting.
- PHP scripts can be embedded into HTML.
- The LAMP architecture has become popular in the Web industry as a way of deploying inexpensive, reliable, scalable, secure web applications.

❑ Official Site: <https://secure.php.net/>



# Installation and Administration

---

MySQL

Apache

PHP

phpMyAdmin

# Installing MySQL / MariaDB(1)

---

## □ Steps (on FreeBSD)

- #pkg install mysql55-server
- #pkg install mysql56-server
- #pkg install mysql57-server
- #pkg install mariadb100-server
- #pkg install mariadb101-server
- #pkg install mariadb55-server
  
- Client
  - mariadb100-client
  - mariadb101-client
  - mysql55-client
  - mysql56-client

```
#yum install mariadb  
#apt-get install mariadb
```



# Installing MySQL (2)

---

## ❑ After install

- #mysql\_secure\_installation

# mysql\_secure\_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MySQL SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MySQL to secure it, we'll need the current password for the root user. If you've just installed MySQL, and you haven't set the root password yet, the password will be blank, so you should just press enter here.

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

Setting the root password ensures that nobody can log into the MySQL root user without the proper authorisation.

```
Set root password? [Y/n] y  
New password: *****  
Re-enter new password: *****  
Password updated successfully!  
Reloading privilege tables..  
... Success!
```

# mysql\_secure\_installation

By default, a MySQL installation has an anonymous user, allowing anyone to log into MySQL without having to have a user account created for them. This is intended only for testing, and to make the installation go a bit smoother. You should remove them before moving into a production environment.

```
# 移除 anonymous 使用者
Remove anonymous users? [Y/n] y
... Success!
```

Normally, root should only be allowed to connect from 'localhost'. This ensures that someone cannot guess at the root password from the network.

```
# 取消 root 遠端登入
Disallow root login remotely? [Y/n] y
... Success!
```

# mysql\_secure\_installation

By default, MySQL comes with a database named 'test' that anyone can access. This is also intended only for testing, and should be removed before moving into a production environment.

```
# 移除 test 資料表
```

```
Remove test database and access to it? [Y/n] y
```

```
- Dropping test database...
```

```
... Success!
```

```
- Removing privileges on test database...
```

```
... Success!
```

Reloading the privilege tables will ensure that all changes made so far will take effect immediately.

```
# 重新載入資料表權限
```

```
Reload privilege tables now? [Y/n] y
```

```
... Success!
```

```
Cleaning up...
```

# mysql\_secure\_installation

---

All done! If you've completed all of the above steps, your MySQL installation should now be secure.

Thanks for using MySQL!

# Installing MySQL (3)

---

## ❑ Startup script...

```
#
# Add the following line to /etc/rc.conf to enable mysql:
# mysql_enable (bool): Set to "NO" by default.
#                               Set it to "YES" to enable MySQL.
# mysql_limits (bool): Set to "NO" by default.
#                               Set it to yes to run `limits -e -U mysql`
#                               just before mysql starts.
# mysql_dbdir (str):   Default to "/var/db/mysql"
#                               Base database directory.
# mysql_args (str):   Custom additional arguments to be passed
#                               to mysqld_safe (default empty).
#
```

# Administering MySQL (1)

---

- ❑ Configuration file
  - Edit `/usr/local/etc/my.cnf`
  
- ❑ Start mysql daemon
  - Using startup script
    - `# /usr/local/etc/rc.d/mysql-server start`

# Administering MySQL (2)

## □ Test

- % `mysql -u root -p`
  - The initial password for root is empty

```
nasa [/usr/local/etc] -randy- mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.1.41-log FreeBSD port: mysql-server-5.1.41

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database                |
+-----+
| information_schema      |
| mysql                   |
| test                    |
+-----+
3 rows in set (0.06 sec)
```



# Administrating MySQL (3)

## ❑ Securing initial accounts

- Two initial accounts

- root
- anonymous

```
mysql> SELECT Host, User From mysql.user;
+-----+-----+
| Host          | User  |
+-----+-----+
| 127.0.0.1     | root  |
| nasa.cs.nctu.edu.tw |      |
| nasa.cs.nctu.edu.tw | root  |
| localhost     |      |
| localhost     | root  |
+-----+-----+
```

```
mysql> UPDATE mysql.user SET Password = PASSWORD('test123') WHERE User = 'root';
Query OK, 3 rows affected (0.08 sec)
Rows matched: 3  Changed: 3  Warnings: 0
```

```
mysql> FLUSH PRIVILEGES;           # Reload the grant tables
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> SET PASSWORD FOR 'root'@'localhost' = PASSWORD('ttt123');
Query OK, 0 rows affected (0.02 sec)
```

# Installing Apache (1)

---

## ❑ Steps

- # pkg install apache24
- # make install clean

## ❑ Options

- A lot of options for modules
- WITH\_SSL (default)
- WITH\_MPM=worker
- WITH\_THREADS=yes
- WITH\_SUEXEC=yes

# Installing Apache (2)

## ❑ Installed...

To run apache www server from startup, add **apache24\_enable="YES"** in your `/etc/rc.conf`. Extra options can be found in startup script.

Your **hostname** must be resolvable using at least 1 mechanism in `/etc/nsswitch` typically DNS or `/etc/hosts` or apache might have issues starting depending on the modules you are using.

## ❑ Startup script

- `/usr/local/etc/rc.d/apache24`
- `apache24_http_accept_enable`
  - `accf_http`
  - `/boot/loader.conf`
    - `accf_http_load="YES"`
    - `#kldload accf_http` (Prevent Slowloris attack)
    - [https://en.wikipedia.org/wiki/Slowloris\\_\(computer\\_security\)](https://en.wikipedia.org/wiki/Slowloris_(computer_security))

# Apache configuration – Configuration files

---

## ❑ Location

- The default location of apache (in ports) is /usr/local/etc/apache24
- Major configuration file: httpd.conf
  - Other configuration files could be included. (setting in httpd.conf)
  - extra/httpd-\*.conf, Includes/\*.conf

## ❑ Two types

- Global settings
  - Server configurations
  - Options of modules
- Directory Configuration
  - Local setting for certain directory

# Apache configuration – Global Settings (httpd.conf)

---

## ❑ Server configuration

- Listen 80
- ServerAdmin liuyh@cs.nctu.edu.tw
- ServerName nasa.cs.nctu.edu.tw
- DocumentRoot "/home/wwwadm/data"
  - Remember create DocumentRoot directory if you modify it

## ❑ Options of modules

## ❑ Include supplemental configuration files

- Include etc/apache24/extra/httpd-\*.conf
- Include etc/apache24/Includes/\*.conf

# Apache configuration – Directory Configuration (1)

## □ Configuration parameters

- Options
  - All
  - ExecCGI
  - FollowSymLinks
  - Indexs
  - MultiViews
  - SymLinksIfOwnerMatch
- <https://httpd.apache.org/docs/2.4/mod/core.html#options>

```
<Directory "/home/hosts/ftp">
  Options Indexes FollowSymlinks
  AllowOverride None
  <IfModule !mod_authz_core.c>
    Order allow,deny
    Allow from all
  </IfModule>
  <IfModule mod_authz_core.c>
    Require all granted
    Require method GET POST HEAD
  </IfModule>
```

# Apache configuration – Directory Configuration for 1.3 (2)

## □ Configuration parameters

- AllowOverride
  - All (Read .htaccess)
  - None (ignoring .htaccess)
- Order
  - Solve collision of deny and allow rules
- Deny/Allow
  - IP/DN (control access to this directory)

```
<Directory "/home/hosts/ftp">
  Options Indexes FollowSymlinks
  AllowOverride None
  <IfModule !mod_authz_core.c>
    Order allow, deny
    Allow from all
  </IfModule>
  <IfModule mod_authz_core.c>
    Require all granted
    Require method GET POST HEAD
  </IfModule>
</Directory>
```

```
<Directory "/home/hosts/ftp/none_pub">
  Options -Indexes -FollowSymlinks
  AllowOverride None
  <IfModule !mod_authz_core.c>
    Order allow, deny
    Allow from all
  </IfModule>
  <IfModule mod_authz_core.c>
    Require all granted
    Require method GET POST HEAD
  </IfModule>
</Directory>
```

# Apache configuration – Directory Configuration for 2.x (3)

## □ Configuration parameters

- AllowOverride
  - All (Read .htaccess)
  - None (ignoring .htaccess)
- Order
  - Solve collision of deny and allow rules
- Deny/Allow
  - IP/DN (control access to this directory)

```
<Directory "/home/hosts/ftp">
  Options Indexes FollowSymlinks
  AllowOverride None
  <IfModule !mod_authz_core.c>
    Order allow, deny
    Allow from all
  </IfModule>
  <IfModule mod_authz_core.c>
    Require all granted
    Require method GET HEAD
  </IfModule>
</Directory>
```

```
<Directory "/home/hosts/ftp/none_pub">
  Options -Indexes -FollowSymlinks
  AllowOverride None
  <IfModule !mod_authz_core.c>
    Order allow,deny
    Allow from all
  </IfModule>
  <IfModule mod_authz_core.c>
    Require all granted
    Require method GET HEAD
  </IfModule>
</Directory>
```



# Apache configuration – Directory Configuration for 2.x (3)

## ❑ Comparison

```
Deny All
  2.2 configuration:
    Order deny,allow
    Deny from all
  2.4 configuration:
    Require all denied

Allow All
  2.2 configuration:
    Order allow,deny
    Allow from all
  2.4 configuration:
    Require all granted

Allow Host
  2.2 configuration:
    Order Deny,Allow
    Deny from all
    Allow from example.org
  2.4 configuration:
    Require host example.org
```

# Apache configuration – Options of Modules

## ❑ dir\_module

```
<IfModule dir_module>
    DirectoryIndex index.html
</IfModule>
```

## ❑ alias\_module

([http://httpd.apache.org/docs/2.4/mod/mod\\_alias.html](http://httpd.apache.org/docs/2.4/mod/mod_alias.html))

```
<IfModule alias_module>
    Redirect /foo http://www.example.com/bar
    Alias /webpath /full/filesystem/path
    ScriptAlias /cgi-bin/ "/usr/local/www/apache24/cgi-bin/"
</IfModule>
```

## ❑ mime\_module

```
DefaultType text/plain
<IfModule mime_module>
    TypesConfig etc/apache24/mime.types
    AddType application/x-compress .Z
    AddHandler cgi-script .cgi
</IfModule>
```

## Supplemental configuration – httpd-mpm.conf (Multi-Processing Module)

---

- ❑ Server-pool management (MPM specific)
  - Include etc/apache24/extra/httpd-mpm.conf
- ❑ WITH\_MPM
  - prefork: non-threaded, pre-forking
  - worker: hybrid multi-process multi-threaded

```
<IfModule mpm_worker_module>
    StartServers          3
    MinSpareThreads      75
    MaxSpareThreads      250
    ThreadsPerChild      25
    MaxRequestWorkers    400
    MaxConnectionsPerChild 0
</IfModule>
```

# Supplemental configuration – httpd-userdir.conf

## ❑ User home directories

- Include etc/apache24/extra/httpd-userdir.conf

```
UserDir public_html
UserDir disabled root toor daemon operator bin tty kmem games news man
sshd bind proxy _pflogd _dhcp uucp pop www nobody mailnull smmsp

<Directory "/home/*/public_html">
    AllowOverride FileInfo AuthConfig Limit Indexes
    Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
    <Limit GET POST OPTIONS>
        Order allow,deny
        Allow from all
    </Limit>
    <LimitExcept GET POST OPTIONS>
        Order deny,allow
        Deny from all
    </LimitExcept>
</Directory>
```

- Methods: <http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html>

# Supplemental configuration – httpd-vhosts.conf

## □ Virtual hosts

- Include etc/apache24/extra/httpd-vhosts.conf
- Name-based
  - NameVirtualHost
  - <VirtualHost>
- IP-based
  - <VirtualHost>
- ServerName
- DocumentRoot
- Ref: <http://httpd.apache.org/docs/2.4/vhosts/>

```
Listen 80
Listen 8080

NameVirtualHost 172.20.30.40:80
NameVirtualHost 172.20.30.40:8080
<VirtualHost 172.20.30.40:80>
  ServerName www.example.com
  DocumentRoot /www/domain-80
</VirtualHost>
<VirtualHost 172.20.30.40:8080>
  ServerName www.example.com
  DocumentRoot /www/domain-8080
</VirtualHost>
<VirtualHost 172.20.30.40:80>
  ServerName www.example.org
  DocumentRoot /www/otherdomain-80
</VirtualHost>
<VirtualHost 172.20.30.40:8080>
  ServerName www.example.org
  DocumentRoot /www/otherdomain-8080
</VirtualHost>
```

# Supplemental configuration – More...

---

- ❑ Multi-language error messages
  - `httpd-multilang-errordoc.conf`
- ❑ Fancy directory listings
  - `httpd-autoindex.conf`
- ❑ Language settings
  - `httpd-languages.conf`
- ❑ Real-time info on requests and configuration
  - `httpd-info.conf`
- ❑ Local access to the Apache HTTP Server Manual
  - `httpd-manual.conf`
- ❑ Various default settings
  - `httpd-default.conf`

# Other configuration for Apache – log

- ❑ Rotate your log using newsyslog
- ❑ In httpd config
  - ErrorLog "/var/log/httpd-error.log"
  - TransferLog "/var/log/httpd-access.log"

```

/var/log/httpd-access.log      640  5  * @T00  Z      /var/run/httpd.pid
/var/log/httpd-error.log      640  5  * @T00  Z      /var/run/httpd.pid

/www/jal.tw/logs/access.log    644  5  10240  *  JC /var/run/httpd.pid
/www/jal.tw/logs/ssl-access.log 644  5  10240  *  JC /var/run/httpd.pid
/www/jal.tw/logs/error.log     644  5  10240  *  JC /var/run/httpd.pid
/www/140.131.150.111/logs/access.log 644  5  10240  *  JC /var/run/httpd.pid
/www/140.131.150.111/logs/error.log 644  5  10240  *  JC /var/run/httpd.pid

```

- ❑ In startup script
  - \_pidprefix="/var/run/httpd"
  - pidfile="\${\_pidprefix}.pid"

# Other configuration for Apache – Secure your Server

---

- ❑ Prevents git file leak

```
# jal.20150317: protect git file
<Directorymatch "^/.* /\.git+/">
    Require all denied
</Directorymatch>

<Files ~ "^\.git">
    Require all denied
</Files>
```



# .htaccess (1)

---

## ❑ .htaccess

- Allow admin or users to control access to certain directory

## ❑ Usage

- Modify httpd.conf
- Create .htaccess file
- Generate password database
- Test

# .htaccess (2)

## □ Example

- Modify httpd.conf
- Create .htaccess file
- Generate password file

```
<Directory
"/home/wwwadm/data/test1">
  Options None
  AllowOverride All
  Order allow,deny
  Allow from all
</Directory>
```

```
liuyh@nasa /home/wwwadm/data/test1> cat .htaccess
AuthName "SA-test1"
AuthType "Basic"
AuthUserFile "/home/wwwadm/data/test1/.htpasswd"
Require valid-user
Options Indexes
```

```
liuyh@nasa /home/wwwadm/data/test1> htpasswd -c ../.htpasswd SA-user1
New password:
Re-type new password:
Adding password for user SA-user1
```

# .htaccess (3)



# Installing PHP (1)

---

## □ Steps

- #pkg install php70
- #pkg install mod\_php70

# Installing PHP (2)

## ❑ Installed...

```
Make sure index.php is part of your DirectoryIndex.
```

```
You should add the following to your Apache configuration file:
```

```
AddType application/x-httpd-php .php
AddType application/x-httpd-php-source .phps
```

- For use of Apache, you should restart apache to load php7\_module

```
LoadModule php7_module          libexec/apache24/libphp7.so

<IfModule dir_module>
    DirectoryIndex index.php index.html
</IfModule>
```

## ❑ php.conf

# Test PHP in apache (1)

## ❑ Edit httpd.conf

- % mkdir -p /home/wwwadm/data
- % cd /usr/local/etc/apache24/
- Edit httpd.conf

```
<IfModule mime_module>
...
AddType application/x-httpd-php .php .phtml .php5
AddType application/x-httpd-php-source .phps
...
</IfModule>
```

```
<IfModule dir_module>
  DirectoryIndex index.php index.html
</IfModule>
```

# Test PHP in apache (2)

## ❑ Start apache

- /usr/local/etc/rc.d/apache24 start

## ❑ Test PHP

- % Edit /home/wwwadm/data/index.php

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



<b>System</b>	FreeBSD evilbig5.math.nctu.edu.tw 8.0-STABLE FreeBSD 8.0-STABLE #3: Fri Dec 4 04:28:07 CST 2009 root@evilbig5.math.nctu.edu.tw:/usr/obj/usr/src/sys/EVILBIG5 i386
<b>Build Date</b>	Dec 22 2009 13:05:05
<b>Configure Command</b>	'./configure' '--with-layout=GNU' '--with-config-file-scan-dir=/usr/local/etc/php' '--disable-all' '--enable-libxml' '--with-libxml-dir=/usr/local' '--enable-reflection' '--program-prefix=' '--enable-fastcgi' '--with-apxs2=/usr/local/sbin/apxs' '--with-regex=php' '--with-zend-vm=CALL' '--disable-ipv6' '--prefix=/usr/local' '--mandir=/usr/local/man' '--infodir=/usr/local/info' '--build=i386-portbld-freebsd8.0'
<b>Server API</b>	Apache 2.0 Handler
<b>Virtual Directory Support</b>	enabled
<b>Configuration File (php.ini) Path</b>	/usr/local/etc
<b>Loaded</b>	(none)

# Appendix

---

phpMyAdmin

lighttpd

FastCGI



# phpMyAdmin

---

- ❑ phpMyAdmin can manage a whole MySQL server as well as a single database over the World Wide Web.
- ❑ Official Site: <http://www.phpmyadmin.net/>
- ❑ Documentation: <http://www.phpmyadmin.net/documentation/>
- ❑ Features
  - Browser-based, Supporting PHP5.3+, MySQL 5.0+, Open Source
- ❑ There are four authentication modes offered:
  - http
  - cookie
  - signon
  - config(the less secure one, not recommended).

# Installing phpMyAdmin (1)

## ❑ databases/phpmyadmin

- # make install clean

## ❑ Installed...

phpMyAdmin-3.2.4 has been installed into:  
/usr/local/www/phpMyAdmin

Please edit **config.inc.php** to suit your needs.

To make phpMyAdmin available through your web site, I suggest that you add something like the following to httpd.conf:

```
Alias /phpmyadmin/ "/usr/local/www/phpMyAdmin/"
```

```
<Directory "/usr/local/www/phpMyAdmin/">
```

```
Options none
```

```
AllowOverride Limit
```

```
Order Deny,Allow
```

```
Deny from all
```

```
Allow from 127.0.0.1 .example.com
```

```
</Directory>
```

# Installing phpMyAdmin (2)

---

- ❑ config.inc.php
  - Override libraries/config.default.php
  
- ❑ config.sample.inc.php
  - `$cfg['blowfish_secret']`

# Adminstrating MySQL – Using phpMyAdmin (2)



歡迎使用 phpMyAdmin

語系 - *Language*

中文 - Chinese traditional

登入 

使用者名稱:

jal

密碼:

執行

# Administrating MySQL – Using phpMyAdmin (3)

The screenshot displays the phpMyAdmin interface for a MySQL server on localhost. The interface is in Chinese and shows various configuration panels.

**phpMyAdmin**

伺服器: localhost

資料庫 | SQL | 狀態 | 使用者帳號 | 匯出 | 匯入 | 設定 | 備援 | 更多

最近使用 | 我的最愛

新增

- information\_schema
- mysql
- performance\_schema
- phpmyadmin
- postfix

**一般設定**

- 修改密碼
- 伺服器連線編碼與排序: utf8mb4\_unicode\_ci

**外觀設定**

- 語系 - Language: 中文 - Chinese traditional
- 主題: pmahomme
- 字體大小: 82%
- 更多設定

**資料庫伺服器**

- 伺服器: Localhost via UNIX socket
- 伺服器類別: MySQL
- 伺服器版本: 5.6.35 - MySQL Community Server (GPL)
- 協定版本: 10
- 使用者: jal@localhost
- 伺服器字元集: UTF-8 Unicode (utf8)

**網頁伺服器**

- Apache
- 資料庫用戶端版本: libmysql - mysqlnd 5.0.11-dev - 20120503 - \$Id: 76b08b24596e12d4553bd41fc93cccd5bac: \$
- PHP 擴充套件: mysqli curl mbstring
- PHP 版本: 5.6.30

**phpMyAdmin**

- 版本資訊: 4.6.6 (最新)
- 說明文件
- 官方首頁
- 貢獻
- 技術支援
- 版本沿革
- 授權

主控台

# Administrating MySQL – Using phpMyAdmin (4)

- ❑ Create another user with limited privilege

The screenshot shows the '新增使用者帳號' (Add User) form in phpMyAdmin. The form is divided into three main sections: '登入資訊' (Login Information), '使用者帳號的資料庫' (Database for the user account), and '全域權限' (Global Privileges). The '登入資訊' section includes fields for '帳號' (Username), '主機名稱' (Host Name), '密碼' (Password), '重新輸入' (Re-enter), '認證外掛程式' (Authentication Plugin), and '產生密碼' (Generate Password). The '使用者帳號的資料庫' section has two checkboxes for granting all privileges. The '全域權限' section has a '全選' (Select All) checkbox. The browser address bar shows 'localhost' and the navigation menu includes '資料庫', 'SQL', '狀態', '使用者帳號', '匯出', '匯入', '設定', '備援', and '更多'.

新增使用者帳號

登入資訊

帳號: 使用文字方塊:

主機名稱: 任意主機 %

密碼: 使用文字方塊:

重新輸入:

認證外掛程式: 原生 MySQL 認證

產生密碼:

使用者帳號的資料庫

建立與使用者同名的資料庫並授予所有權限。

給以 帳號\_ 開頭的資料庫 (username\\_%) 授予所有權限。

全域權限  全選

注意: MySQL 權限名稱會以英文表示。

主控台

# Installing lighttpd

---

- ❑ `www/lighttpd`
  - Official: <http://www.lighttpd.net/>
- ❑ Configuration files
  - `/usr/local/etc/lighttpd/{lighttpd,modules}.conf`
  - `/usr/local/etc/lighttpd/{vhosts,conf}.d/`
- ❑ Startup script
  - `/usr/local/etc/rc.d/lighttpd`
- ❑ Documentation:
  - <https://redmine.lighttpd.net/projects/1/wiki/Docs>
  - alias, cgi, dirlisting, fastcgi, ssl, userdir
  - Virtual hosts: evhost, mysqlvhost, simple-vhost

# FastCGI

---

- ❑ FastCGI is actually CGI with only a few extensions.
  - FastCGI is language-independent.
  - FastCGI run applications in processes isolated from the core Web server, which provides greater security than APIs.
  - FastCGI developers are committed to propagating FastCGI as an open standard. (C/C++, Java, Perl, Tcl)
  - FastCGI is not tied to the internal architecture of any Web server and is therefore stable even when server technology changes.
- ❑ Benefits:
  - Distributed computing
  - Multiple and extensible roles