

# 安装部署

## 目录:

使用语言:	php
反向代理服务器:	Nginx
数据库:	mysql
数据缓存:	memcache
Web 框架:	YII
Web 前端 CSS 框架:	bootstrap
模板引擎:	Smarty

## Windows 下 Nginx 配置

### 1. 安装 Nginx

在其官方网站上面下载: <https://www.nginx.com/>  
下载后直接解压即可,这里解压缩到 c:\nginx 目录。

### 2. 启动 Nginx

命令行进入 c:\nginx 目录,运行 `nginx.exe`,启动控制台窗口。默认启用 80 端口。  
用过 Tomcat 的人都希望能在控制台看到启动日志,nginx 的日志却不得不查看 logs 目录下的相应 log 文件。

### 3. 访问欢迎 html 页

在浏览器中访问 `http://localhost`,可以看到默认的欢迎页。

### 4. 停止 Nginx

`Ctrl+C` 没反应。于是关闭控制台窗口。可是再访问 `http://localhost` 依然有效。查看进程,发现 nginx 根本没有被关闭。因此如果想彻底关闭 nginx,应该是

*Command* 代码

```
1. nginx -s stop
```

或者使用 windows 的 `taskkill` 命令:

Command 代码

```
1. taskkill /F /IM nginx.exe > nul
```

## 5. Nginx 常用配置

Nginx 的所有配置都默认使用 `conf/nginx.conf` 文件，其地位相当于 apache 的 `httpd.conf` 文件。当运行 `nginx.exe` 暗含运行了 `nginx -c conf/nginx.conf`。如果想使用自己定义的 `conf` 文件如 `my.conf`，命令为 `nginx -c conf/my.conf`。

# Windows 下安装配置 MySQL

1. 下载 MySQL 的包：<http://mysql.com> 并解压至根目录。

## 2. 创建配置文件

打开 MySQL 安装目录，在该目录下有一系列的 `ini` 文件，我们可以任选一个文件作为 MySQL 服务每次启动都用到的配置文件，以后我们可以通过该文件来指定 MySQL 配置。

## 3. 选择 MySQL 服务类型

`mysqld-nt` 二进制优化并支持命名管道

`mysqld` 二进制优化但不支持命名管道

`mysqld-debug` 和 `mysqld-nt` 一样，但已经编译过，可以进行完全调试和自动对内存的分配进行检查

## 4. 启动 MySQL

从 Windows 命令行启动 MySQL 服务器，启动控制台窗口并输入命令：

```
D:/MySQL/bin>mysqld-nt
```

从 Windows 命令行停止 MySQL 服务器，启动控制台窗口并输入命令：

```
D:/MySQL/bin>mysqladmin -u root shutdown
```

根据系统中 MySQL 安装位置的不同，使用的路径也不同。

启动服务器后，服务器则在前台运行，在服务器退出前，不会出现命令提示。因此，当服务器运行时，应当打开另一个控制台窗口来运行客户端程序，停止服务器也是一样的。

以 Windows 服务方式启动 MySQL 服务器，安装 MySQL 服务，启动控制台窗口并输入命令：

```
D:/MySQL/bin>mysqld-nt --install-manual MySQL  
--defaults-file=D:/MySQL/my.ini
```

--install 选项, 给出了默认服务名(MySQL), 使用--install-manual 选项是将服务设置为手动启动模式, 如果想开机自动启动服务只需将--install-manual 替换为--install。

--defaults-file 选项, 服务器只从指定文件的[mysqld]组读取配置

使用命令 NET START MySQL 启动服务, NET STOP MySQL 停止服务。

要想卸载服务, 如果服务正运行则应首先使用命令 NET STOP MySQL 停止服务。

启动控制台窗口并输入命令:

```
D:/MySQL/bin>mysqld-nt --remove
```

## 5. 连接与断开 MySQL

启动控制台窗口并输入命令:

```
D:/MySQL/bin>mysql -u root
```

```
mysql> quit
```

```
Bye
```

```
D:/MySQL/bin>
```

默认 root 用户的密码为空, 只需回车一次, 使用命令 quit 就可以断开 MySQL 连接。

## 6. 使默认用户账户安全

初始 root 账户的密码为空, 因此任何人可以用 root 账户不用任何密码来连接 MySQL 服务器, 并具有所有权限。

```
mysql> select host,user,password from mysql.user;
```

host	user	password
localhost	root	
127.0.0.1	root	
localhost		

如果不想让人通过匿名用户登录的话可以将匿名用户删除, 建议删除或为其设置密码, 因为该匿名用户的权限和 root 用户是一样的。

```
mysql> drop user '@'localhost' ;
```

使用 SET PASSWORD 指定密码, 用 root 连接服务器并执行两个 SET PASSWORD 语句。一定要使用 PASSWORD()函数来加密密码。

```
mysql> set password for 'root'@'localhost' = password('root');
```

```
mysql> set password for 'root'@'127.0.0.1' = password('root');
```

设置完密码之后在下次重新连接服务器时就必须要提供相应的密码。

```
D:/MySQL/bin>mysql -u root -p
```

```
Enter password: ****
```

## Memecache 安装

1.将 memcache 服务器安装包解压到 C:\memcached 文件夹后，使用 cmd 命令窗口安装。

2>开始>运行:CMD(确定)

3>cd C:\memcached (回车)

4>memcached -d install(回车 这步执行安装)

5>memcached -d start(回车 这步执行启动 memcache 服务器，默认分配 64M 内存，使用 11211 端口)

此时 memcache 服务器已经可以正常使用了。

## Yii 框架部署

1.到 yii 官方网站下载 yii framework: <http://www.yiiframework.com>

2.将下载下来的文件夹解压，放置你的 php 环境目录里面；这里我们需要的只有两个文件夹 framework 和 requirements: framework 是框架文件,这个是必须的；requirements 是环境检测文件；demos 是示例。

3.然后检测一下当前的环境是否符合 yii framework 的要求，访问 <http://192.168.0.112:81/yii/requirements/> 将给出我们建议的环境配置。

4.使用命令行工具 yiic 创建 web 应用: yiic 文件 D:\webSite\yii\framework 下。

目录 1>yiic.bat webapp 目录 2

目录 1 位 yiic 所在目录，目录 2 为刚创建的 webapp 所在目录。

## bootstrap 部署

1.到 bootstrap 官方网站下载 bootstrap: <http://www.bootcss.com>

2.解压到，然后添加进项目之后，在用到的 html 或者 jsp 中引用他的文件即可，不过必须注意的是，在引入 js 之前，一定要先把 jquery 引入进去，因为它依赖于 jquery。

## Smarty 安装配置

1. 下载 Smarty: <http://smarty.php.net/>

2. 将压缩包解压, 得到一个 libs 目录, 其中包括了 smarty 类库的核心文件, 包括 smarty.class.php、smarty\_Compiler.class.php、config\_File.class.php 和 debug\_tpl 4 个文件, 还有 internals 和 plug-ins 两个目录

复制 libs 目录到服务器根目录下, 并为其重命名为 smarty

将 smarty 文件夹复制到需要的项目中(例 aaa), 在 Smarty 目录下再新建 4 个文件夹, 分别是

- template(模板目录),
- templates\_c(编译目录),
- configs(config 文件目录),
- cache(模板缓存目录).

创建一个 php 文件, 将配置代码写到这里面,

需要用的时候只要每一个页面包含过来就可以了, 而不需要每一个页面都写。html 代码都放在 template 文件夹下面, php 程序直接放在 aaa 项目中就可以了。