

虚拟化开发环境

刘梓懿

进化路线图

1. 手工配置开发环境
2. 开发环境虚拟机镜像
3. 自动化构建开发环境镜像
4. 持续集成/构建流水线
5. 云端虚拟化开发机实例
6. 全环境自动化构建部署

手工构建开发环境

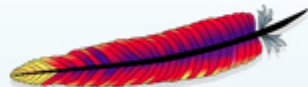
以典型的 PHP 应用架构为例

Linux + Apache + MySQL + PHP

LAMP:



Linux



Apache



MySQL

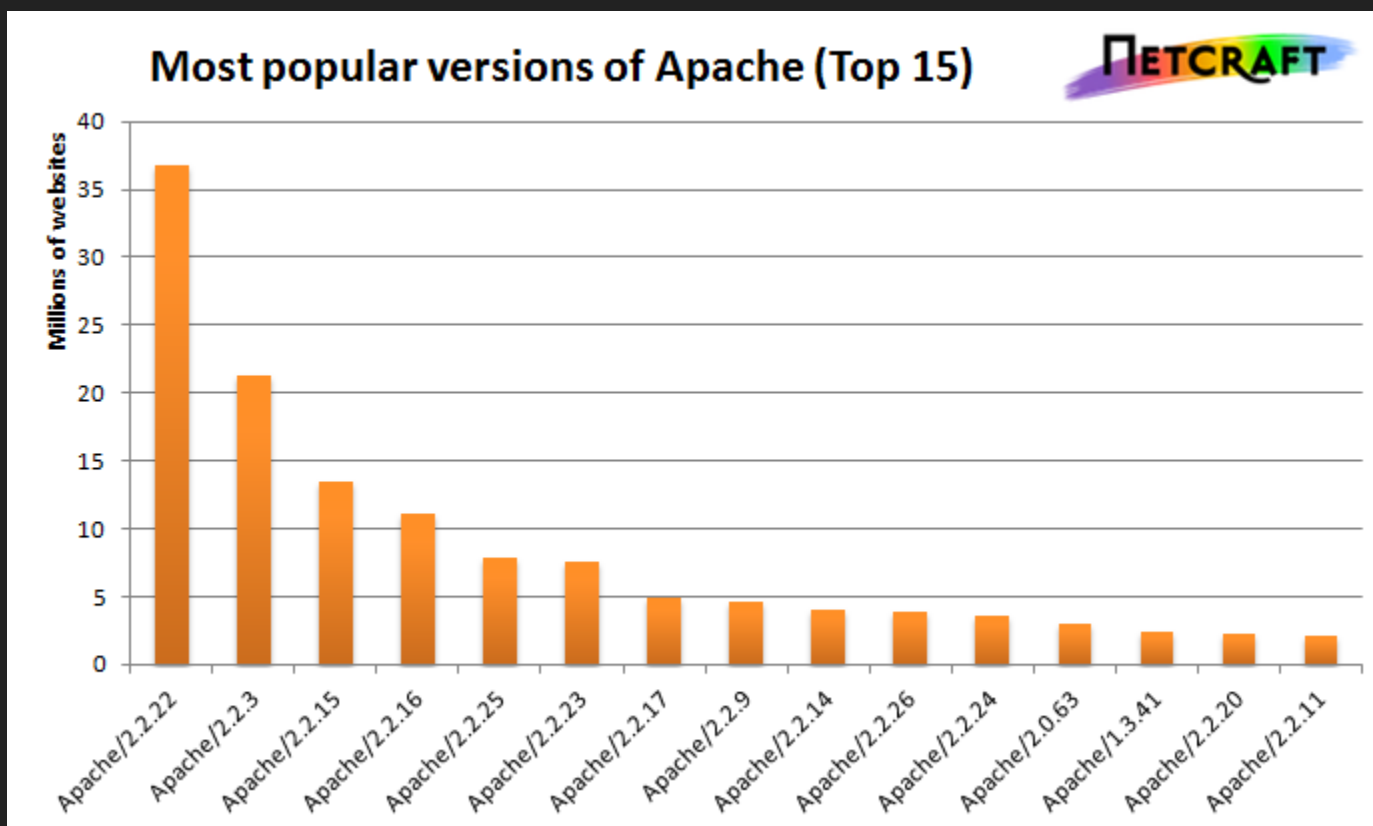


PHP

如何手工配置开发环境

- 安装 Apache，并配置 Apache 设置
- 安装 PHP
- 安装 MySQL，并配置 MySQL 设置
- 配置其他依赖
- 安装 Git/SVN 等版本控制工具
- 下载代码库
- 下载并导入 MySQL 数据
- 运行起来看看是否工作

版本问题



开发环境与产品环境不一致是万恶之源



繁琐的安装文档

安装

- # 安装 **Composer**
 - # 安装 **Laravel**
 - # 服务器环境需求
 - # 设定
 - # 优雅链接
-

安装 Composer

Laravel 框架使用 [Composer](#) 来管理其相依性。首先，下载一份 `composer.phar` 下来。之后，您可以把它放在本地端的项目目录，或者是移至 `/usr/local/bin` 让全站皆可使用。在 Windows 下，您可以使用 Composer [Windows 安装工具](#)。

安装 Laravel

通过 Laravel 安装工具

首先, 使用 `Composer` 全局下载并安装 `Laravel/installer` :

繁琐的安装文档

服务器环境需求

Laravel 框架有一些系统需求：

- PHP >= 5.4
- MCrypt PHP 扩展包

PHP 5.5 之后，一些发行版本需要手动安装 PHP JSON 扩展包。如果您使用的是 Ubuntu，可以通过 `apt-get install php5-json` 来直接安装。

设定

安装完 Laravel 的第一件事就是为你的应用设定一个随机的 key。如果你是通过 Composer 安装的 Laravel，这个 key 已经通过 `key:generate` 命令为你自动设置了。一般来说，这个字符串应该是 32 位长度的字符。这个 key 应该设定在 `app.php` 文件中。如果这个应用的 **key** 没有设置，你的用户的 **sessions** 以及其他加密数据就不能保证安全。

Laravel 几乎无需设定即可马上使用。您可以自由的开始开发。然而，您可以查看 `app/config/app.php` 文件和他的文件。它包含了数个您的应用程序所想要更动的选项如 `时区 (timezone)` 和 `语系 (locale)`。

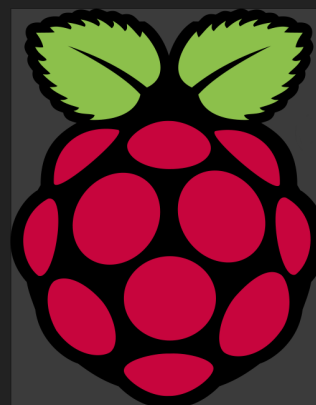
一旦 Laravel 安装完成，您应该同时 设定本地环境。当您在您的本机上部署时，可以让您得到更详细的错误信息。默认在您的正式环境里详细的错误信息是被关掉的。

附注：您不应该在正式环境中将 `app.debug` 设为 `true`。绝对！千万不要！

安装遇到问题怎么办



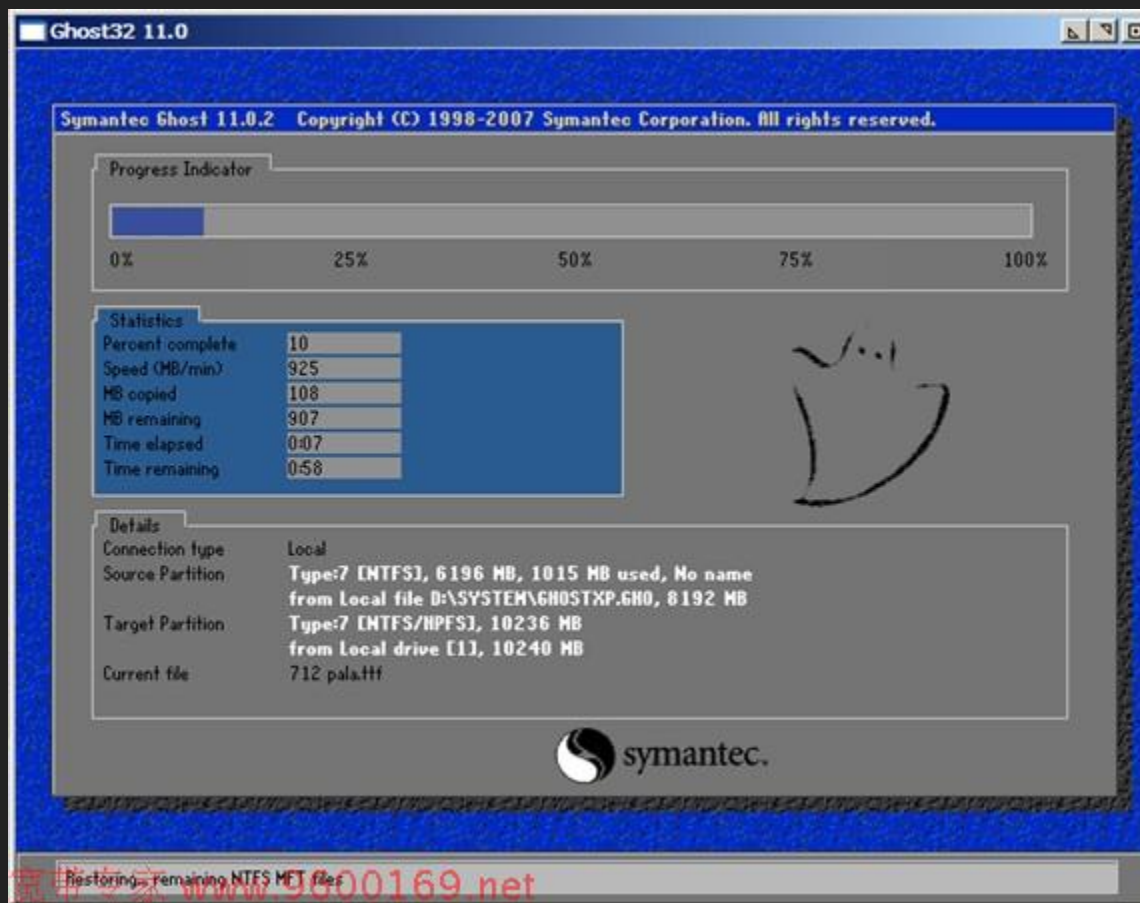
安装问题无可避免，这是因为...



开发机系统环境不一致

其他的问题？

镜像技术



镜像技术



+

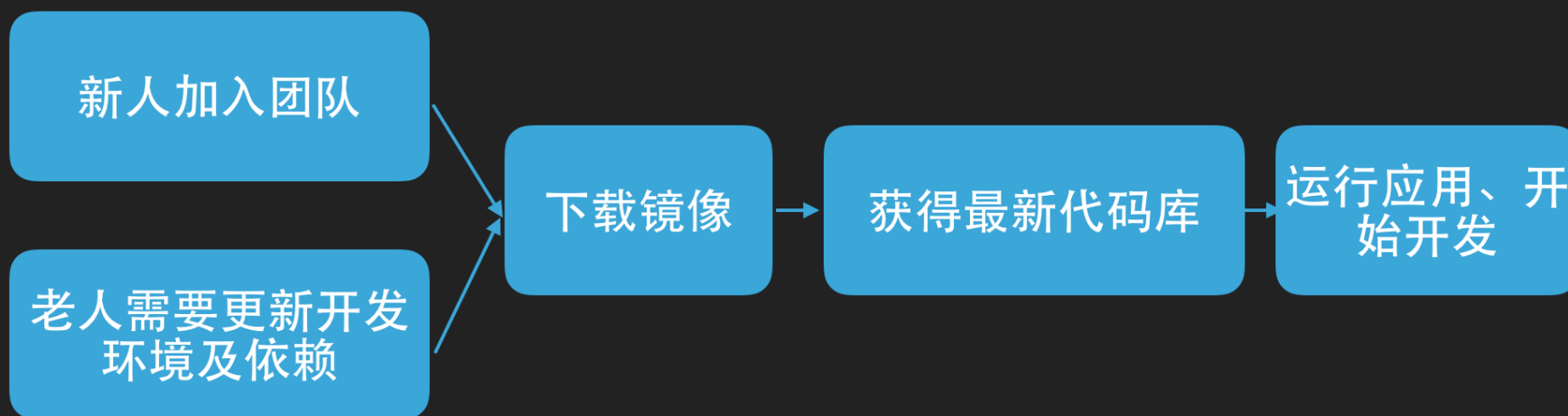


镜像技术

通过虚拟化技术，构建统一的、接近于产品环境的应用运行环境，同时开发者仍然拥有选择自己喜欢的开发工具的自由。



拥有镜像之后的开发流程



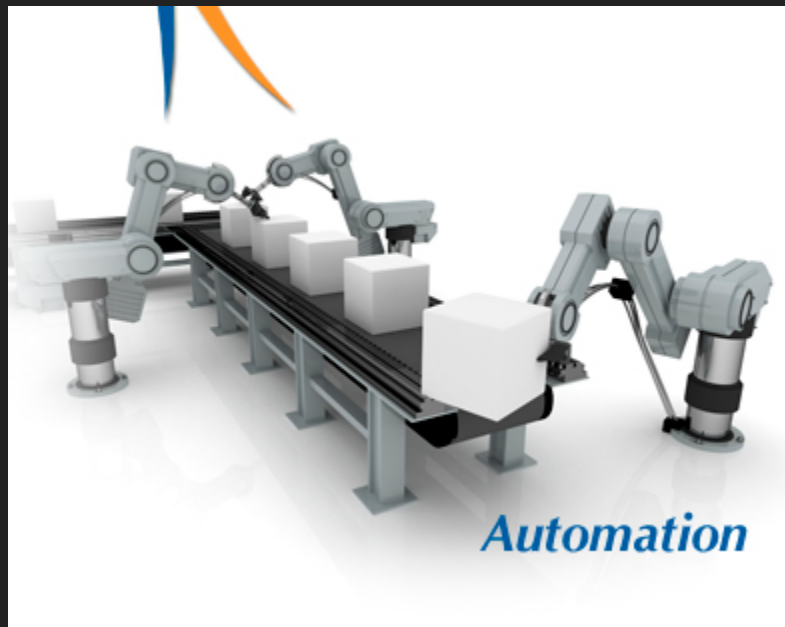
DEMO

手工管理镜像存在的问题

- 镜像配置仍然需要文档
- 镜像本身没有版本控制
- 升级依赖、更新环境仍然需要手工操作，繁琐易出错

自动化开发环境构建

- 自动化环境构建
- 基础设施即代码



如何把手工搭建环境的操作代码化



Ansible is Pythonic Simple IT Automation

自动化搭建 RELAX 开发环境

Relax 是基于 Node.js 技术栈的 CMS

- 安装并配置 Git、C++ 编译器
- 安装 Node.js 4.2.2，配置使用国内 npm 镜像源
- 安装 Mongodb 3.0 并配置
- 安装 Sharp 图形处理库
 - 构建 Sharp 所需依赖：libvips
- 拉取代码库，运行 node 构建
- 拉取开发环境测试数据库，并导入

DEMO

持续集成/构建流水线



DEMO

云端虚拟化



BUILD VIRTUALBOX/CLOUD IMAGES AT ONCE

```
{  
  "provisioners": [  
    {  
      "type": "ansible-local",  
      "playbook_dir": "stack-provision",  
      "playbook_file": "stack-provision/dev-stack.yml"  
    }  
  ]  
}
```

DEMO

全环境自动化构建部署



持续交付流水线

凤凰式部署

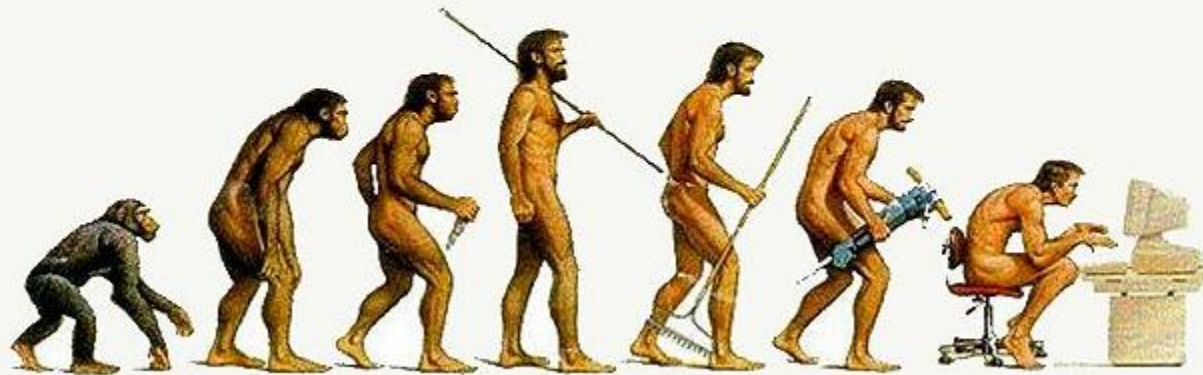


每次部署时，服务器毁灭并重建

进化路线图

1. 手工配置开发环境
2. 开发环境虚拟机镜像
3. 自动化构建开发环境镜像
4. 持续集成/构建流水线
5. 云端虚拟化开发机实例
6. 全环境自动化构建部署

Evolution





简单可依赖的环境搭建

Let's Automate Everything

