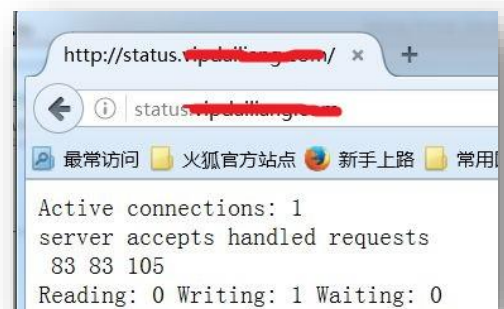


## 1. 克隆 4 台**新虚拟机**，IP 配置如下：

虚拟机 7	eth2:201.1.1.100
虚拟机 8	eth2:201.1.1.5
	eth3:201.1.2.5
虚拟机 9	eth3:201.1.2.100
虚拟机 10	eth3:201.1.2.200

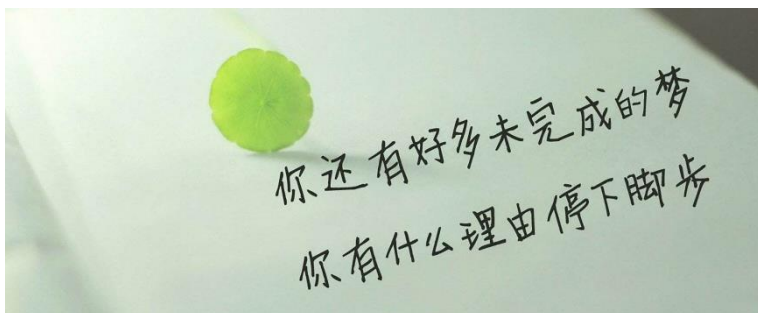


- a) 在虚拟机 8 上源码安装 Nginx 软件，要求如下：
- b) 要求安装 ssl 模块以及 stub\_status 模块
- c) 安装前创建用户 nginx，安装时指定安装用户为 nginx
- d) 该 nginx 配置为后端两台虚拟机的调度器



## 2. 在虚拟机 9 和 10 上部署 LNMP 环境：

- a) 部署 lnmp 环境（支持动静分离）
- b) 部署测试性页面（参考 lnmp\_soft/php\_scripts/php-memcached-demo.tar.gz）
- c) 测试登陆页面的效果
- d) 查看/etc/php-fpm.d/www.conf 配置文件，确认 session 会话信息保存位置
- e) 到/var/lib/php/session 目录下查看 session 文件



### 3. 使用 memcached 实现会话信息共享

- a) 虚拟机 8 安装部署 memcached 缓存数据库
- b) 修改虚拟机 9 和 10 的配置，让 PHP 保存会话信息到 memcached
- c) 默认 PHP 无法连接 memcached，需要安装对应的扩展模块
- d) 完成后再次通过集群调度器访问登陆页面，测试效果（为了查看效果，可将后台页面的背景颜色适当调整）

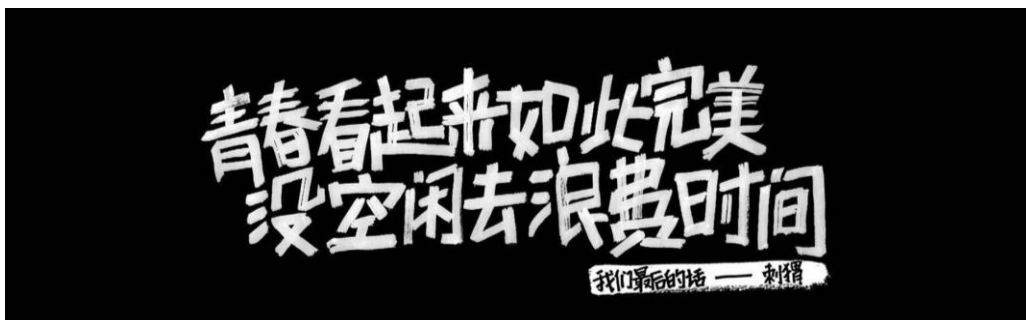
### 4. 配置 Tomcat 服务器，要求如下：

- a) 在虚拟机 9 和 10 上安装部署 tomcat
- b) 在虚拟机 8 上配置调度器，调度后台 2 个 tomcat 站点
- c) 测试查看负载均衡效果



### 5. 使用版本控制系统：

- a) 使用 subversion 创建版本仓库
- b) 将自己之前写的 shell 脚本导入到仓库中
- c) 配置 SVN 服务器，让所有客户端都可用访问 svn 服务器，下载上传代码到服务器
- d) 模拟多人协作修改 shell 脚本（修改不同文件，修改相同文件不同和，修改相同文件相同行）
- e) 登陆 github 网站，创建 note 仓库，将自己之前的笔记上传 github 仓库（再也不用担心笔记丢失）
- f) 在真实主机安装 git 软件，使用 git clone 将 github 仓库下载到本地
- g) 本地新建一个笔记文件，将本地笔记推送到 github 服务器
- h) git add .
- i) git commit -m "xx"
- j) git push



6. 将 nginx 源码包制作成 RPM 软件包, 支持%post 脚本

7. 创建三种 VPN 服务器,要求如下:

- a) 让虚拟机 7 和虚拟机 8 之间建立 gre VPN
- b) 让虚拟机 10 和 windows2008 之间建立 PPTP VPN
- c) 让虚拟机 9 和 windows2008 之间建立 Xl2tp+IPsec VPN

8. 使用 chrony 创建 NTP 服务器

- a) 将虚拟机 8 配置为 NTP 服务器
- b) 其他所有虚拟机与虚拟机 8 同步时间

9. 在虚拟机 8 上安装 pssh

- a) 使用 pssh 远程给其他虚拟机批量创建账户
- b) 使用 pssh 将本地文件批量传给其他所有虚拟机