###1、简述一下什么是Nginx,它有什么优势和功能？ HTTP、反向代理服务器和负载均衡服务器，占用内存小并发能力强，并发能力在2同类型的网页服务器中表现较好，用户很多

Nginx ("engine x") 是一个高性能的 HTTP 和反向代理服务器,特点是占有内存少，并发能力强，
它的并发能力确实在同类型的网页服务器中表现较好，中国大陆使用 nginx网站用户有很多，例如百度、京东、新浪、网易、腾讯、淘宝等
Nginx 可以作为静态页面的 web 服务器，可以进行正向代理、反向代理、负载均衡

###2、Nginx是如何处理一个HTTP请求的呢？ 多进程机制和异步机制
Nginx 是一个高性能的 Web 服务器，能够同时处理大量的并发请求。它结合多进程机制和异步非阻塞机制 ， 异步机制使用的是异步非阻塞方式 ，
Nginx 的多线程机制和异步非阻塞 机制 。

1、多进程机制

服务器每当收到一个客户端时，就有 服务器主进程 （ master process ）生成一个 子进程（ worker process ）出来和客户端建立连接进行交互，
直到连接断开，该子进程就结束了。
使用进程的好处是各个进程之间相互独立，不需要加锁，减少了使用锁对性能造成影响，同时降低编程的复杂度，降低开发成本。
其次，**采用独立的进程，可以让进程互相之间不会影响 ，如果一个进程发生异常退出时，其它进程正常工作**，
master 进程则很快启动新的 工作（worker）进程，确保服务不会中断，从而将风险降到最低。
缺点是操作系统生成一个子进程需要进行 内存复制等操作，在资源和时间上会产生一定的开销。当有大量请求时，会导致系统性能下降 。

2、异步非阻塞机制

每个工作进程 使用 异步非阻塞方式 ，可以处理 多个客户端请求 。
当某个 工作进程 接收到客户端的请求以后，调用 IO 进行处理，如果不能立即得到结果，就去 处理其他请求 （即为 非阻塞 ）；
而 由于异步非阻塞机制，客户端 在此期间也 无需等待响应 ， 可以去处理其他事情~~（即为 异 步 ）~~。
当 IO 返回时，就会通知此 工作进程 ； 该进程得到通知，暂时 挂起 当前处理的事务去 响应客户端请求 。

###3、列举一些Nginx的特性

Nginx服务器的特性包括：

1. 反向代理/L7负载均衡器
2. 嵌入式 Perl 解释器
3. 动态二进制升级
4. 可用于重新编写URL，具有非常好的 PCRE 支持

###4、请列举Nginx和Apache之间的不同点
考到再说

Nginx	Apache
• Nginx是一个基于事件的web服务器 • 所有请求都由一个线程处理 • Nginx避免子进程的概念 • Nginx类似于速度 • Nginx在内存消耗和连接方面比较好 • Nginx在负载均衡方面表现较好 • 对于PHP来说，Nginx可能更可取，因为它支持PHP • Nginx不支持像IBMi和OpenVMS一样的OS。 • Nginx只具有核心功能 • Nginx的性能和可伸缩性不依赖于硬件	• Apache是一个基于流程的服务器 • 单个线程处理单个请求 • Apache是基于子进程的 • Apache类似于功率 • Apache在内存消耗和连接上并没有提高 • 当流量到达进程的极限时，Apache将拒绝新的连接 • Apache支持的PHP、Python、Perl和其他语言使用插件，当应用程序基于Python或Ruby时，它非常有用 • Apache支持更多的OS • Apache提供了比Nginx更多的功能 • Apache依赖于CPU和内存等硬件组件

###5、在Ngix中，如何使用未定义的服务器名称来阻止处理请求?

```
Server {
    listen 80;
    server_name "" ;
    return 444;
}
```

服务器名被保留为一个空字符串，它将在没有“主机”头字段的情况下匹配请求，而一个 特殊的Nginx的非标准代码444被返回，从而终止连接。

###6、请解释Nginx服务器上的 Master 和 Worker 进程分别是什么?

- 主程序 Master process 启动后，通过一个 for 循环来 接收 和 处理外部信号；
- 主进程通过 fork() 函数产生 worker 子进程 ， 每个子进程执行一个 for循环来实现Nginx服务器 对事件的接收和处理 。

###7、请解释代理中的正向代理和反向代理
正向代理：如果把局域网外的 Internet 想象成一个巨大的资源库，则局域网中的客户端要访问 Internet，则需要通过代理服务器来访问，这种代理服务就称为正向代理

> 【百度百科】意思是一个位于客户端和原始服务器(origin server)之间的服务器，为了从原始服务器取得内容，客户端向代理发送一个请求并指定目标(原始服务器)，
> 然后代理向原始服务器转交请求并将获得的内容返回给客户端。客户端才能使用正向代理。

反向代理: 其实客户端对反向代理是无感知的，因为客户端不需要任何配置就可以访问，我们只需要将请求发送到反向代理服务器，
由反向代理服务器去选择目标服务器获取数据后，在返回给客户端，此时反向代理服务器和目标服务器对外就是一个服务器，暴露的是代理服务器地址，
隐藏了真实服务器 IP 地址。

###8、解释Nginx用途

- 正向代理
- 反向代理
- 负载均衡
- 动静分离

~~Nginx服务器的最佳用法是在网络上部署动态HTTP内容，使用SCGI、WSGI应用程序服务器、用于脚本的FastCGI处理程序。它还可以作为负载均衡器。~~

9.一个 master 和多个 woker 有好处？可以进行热部署，每个work进程独立，一个出现问题不会造成服务器终端

（1）可以使用 nginx –s reload 热部署，利用 nginx 进行热部署操作
（2）每个 woker 是独立的进程，如果有其中的一个 woker 出现问题，其他 woker 独立的，继续进行争抢，实现请求过程，不会造成服务中断、

10.设置多少个 woker 合适？ 服务器的 cpu 数相等是最为适宜的

worker 数和服务器的 cpu 数相等是最为适宜的

11.发送请求，占用了 worker 的几个连接数?

答案：2 或者 4 个

12。nginx 有一个 master，有四个 woker，每个 woker 支持最大的连接数 1024，支持的最大并发数是多少?

- 普通的静态访问最大并发数是： worker_connections * worker_processes /2，
- 而如果是 HTTP 作 为反向代理来说，最大并发数量应该是 worker_connections * worker_processes/4。
  4* 1024 / 2 = 2048

4* 1024 / 4 = 1024
所以支持的最大并发数是 1024 或 2048

以上 11 和 12 问题是 连接数 worker_connection相关问题

13 nginx 配置文件

nginx.conf 配置文件分为三部分：
全局快/events块/http块

第一部分：全局块

从配置文件开始到 events 块之间的内容，主要会设置一些影响 nginx 服务器整体运行的配置指令，主要包括配置运行 Nginx 服务器的用户（组）、
允许生成的 worker process 数，进程 PID 存放路径、日志存放路径和类型以
及配置文件的引入等。
比如上面第一行配置的：
这是 Nginx 服务器并发处理服务的关键配置，worker_processes 值越大，可以支持的并发处理量也越多，但是
会受到硬件、软件等设备的制约

第二部分：events 块

events 块涉及的指令主要影响 Nginx 服务器与用户的网络连接，常用的设置包括是否开启对多 work process
下的网络连接进行序列化，是否允许同时接收多个网络连接，选取哪种事件驱动模型来处理连接请求，每个 word
process 可以同时支持的最大连接数等。
上述例子就表示每个 work process 支持的最大连接数为 1024.
这部分的配置对 Nginx 的性能影响较大，在实际中应该灵活配置。

第三部分：http 块

这算是 Nginx 服务器配置中最频繁的部分，代理、缓存和日志定义等绝大多数功能和第三方模块的配置都在这里。需要注意的是：http 块也可以包括 http 全局块、server 块。
①、http 全局块
http 全局块配置的指令包括文件引入、MIME-TYPE 定义、日志自定义、连接超时时间、单链接请求数上限等。
②、server 块
这块和虚拟主机有密切关系，虚拟主机从用户角度看，和一台独立的硬件主机是完全一样的，该技术的产生是为了
节省互联网服务器硬件成本。
每个 http 块可以包括多个 server 块，而每个 server 块就相当于一个虚拟主机。
而每个 server 块也分为全局 server 块，以及可以同时包含多个 locaton 块。
1、全局 server 块
最常见的配置是本虚拟机主机的监听配置和本虚拟主机的名称或 IP 配置。
2、location 块
一个 server 块可以配置多个 location 块。
这块的主要作用是基于 Nginx 服务器接收到的请求字符串（例如 server_name/uri-string），对虚拟主机名称
（也可以是 IP 别名）之外的字符串（例如 前面的 /uri-string）进行匹配，对特定的请求进行处理。地址定向、数据缓
存和应答控制等功能，还有许多第三方模块的配置也在这里进行。