静态服务和动态服务

静态服务一般返回静态资源，不同用户，无论何时，只要访问同一个服务，得到的内容都是一样的。

动态服务一般返回的就是所谓的动态资源，典型的场景：比如用户登录一个网站的后台，会根据不同的用户身份和权限信息返回不同的内容和资源。所有看到的页面、内容都不一样。

Web服务器、HTTP服务器、应用服务器

Web服务器：

一般指网站服务器，是指驻留于Internet上某种类型计算机的程序，可以向浏览器等Web客户端提供文档，也可以放置网站文件，让全世界浏览；可以放置数据文件，让全世界下载。目前最主流的三个Web服务器是Apache、Nginx、IIS。

从广义上来说，就是响应用户的需求，然后提供服务。几乎所有的服务器软件都可以称为Web服务器。

HTTP服务器：

当今Web服务器应用层通讯协议主要就是HTTP协议。基本上可以将HTTP服务器理解为和Web服务器对等。

HTTP服务器只是把服务器上的资源通过HTTP协议形式传输给客户端。侧重于它对静态资源传输的支持。

应用服务器：

应用服务器通过各种协议把商业逻辑暴露给客户端的程序。它提供了访问商业逻辑的途径以供客户端应用程序使用。应用服务器使用此商业逻辑就像调用对象的一个方法一样。

可以理解为一个特定应用的承载容器，需要有运行时环境的支持。（比如Java领域常用的应用服务器Tomcat，它就必须要Java的runtime运行时环境的支持）因此应用服务器也常被称作动态服务器，动态容器，应用容器，Web容器等。

既然属于动态服务器，它就支持动态响应。可以根据不同的用户请求，能够动态地生成资源并且返回。

一般来说，在流量不大的情况下，应用服务器可以用来做HTTP服务器。

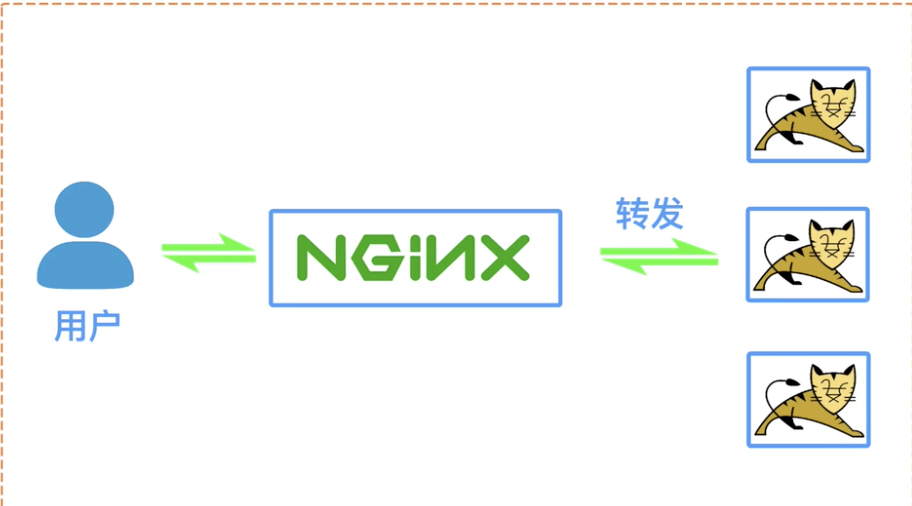
Nginx（HTTP服务器）

将服务端的静态内容或资源，通过HTTP协议的形式传输给客户端。典型的静态服务器。

常用功能：

* 反向代理：用来代理服务器，代理用户要访问的目标服务器。代理服务器接收请求，然后将请求转发给内部网络的服务器（服务集群模式），并将从服务器上得到的结果返回给客户端，此时代理服务器对外就表现为一个服务器。
* 负载均衡：将数据流量分摊到多个服务器执行，减轻每台服务器的压力，多态服务器（集群）共同完成工作任务，从而提高了数据的吞吐量。（策略：轮询、权重、ip\_hash、url\_hash、fair）
* 动静分离：常用于前后端分离，把动态请求和静态请求分离开，合适的服务器处理相应的请求，使整个服务器系统的性能、效率更高。

在现实的应用部署场景中，Nginx一般是与后面真正的动态应用服务器配合，把收到的用户请求转发给后面真正的Tomcat应用服务器，从而提高灵活稳定的Web服务。



Tengine（Nginx的改造）

Tengine是由淘宝网发起的Web服务器项目。它在Nginx的基础上，针对大访问量网站的需求，添加了很多高级功能和特性。Tengine的性能和稳定性已经在大型的网站如淘宝网，天猫商城等得到了很好的检验。它的最终目标是打造一个高效、稳定、安全、易用的Web平台。

Apache HTTP Server

IIS

微软阵营的Web服务器，只能运行于windows系统环境下，非开源。HTTP静态服务器。

Tomcat（Java应用服务器软件，SpringBoot默认的内嵌应用服务器）

提高动态服务，需要Java运行时环境的支持。

一般会和高性能的静态服务器（如Nginx）配合使用。

Jetty（SpringBoot支持的内嵌应用服务器）

Jetty是一个开源的HTTP服务器+应用容器。

与Tomcat类似，都是做动态应用服务器，都符合Servlet标准，Jetty更加轻量级。

Undertow（SpringBoot支持的内嵌应用服务器）

红帽公司开发的一款基于NIO的高性能Web嵌入式服务器。

JBoss

Wildfly

Weblogic（Oracle）

WebSphere（IBM）