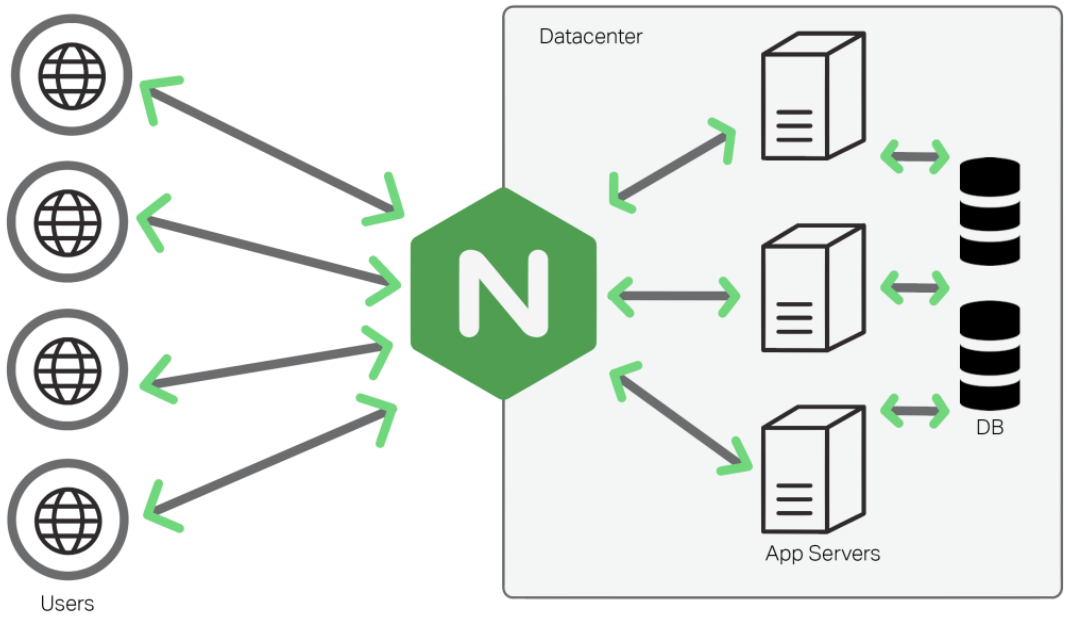
Nginx

凯盛软件

Nginx

凯盛软件

- https://nginx.org
- Nginx（发音同engine x）是一个Http服务器，它能反向代理HTTP, HTTPS, SMTP, POP3, IMAP的协议链接，以及一个负载均衡器和一个HTTP缓存。



安装

凯盛软件

- 下载Nginx
  - wget https://nginx.org/download/nginx-1.12.2.tar.gz
- 安装依赖库
  - apt-get install make
  - apt-get install gcc
  - apt-get install libpcre3 libpcre3-dev
  - apt-get install zlib1g-dev
- 解压Nginx，并在目录中执行
  - ./configure
- 编译并安装
  - make
  - make install
- 安装后目录在/usr/local/nginx中

```
Configuration summary
  + using system PCRE library
  + OpenSSL library is not used
  + using system zlib library

  nginx path prefix: "/usr/local/nginx"
  nginx binary file: "/usr/local/nginx/sbin/nginx"
  nginx modules path: "/usr/local/nginx/modules"
  nginx configuration prefix: "/usr/local/nginx/conf"
  nginx configuration file: "/usr/local/nginx/conf/nginx.conf"
  nginx pid file: "/usr/local/nginx/logs/nginx.pid"
  nginx error log file: "/usr/local/nginx/logs/error.log"
  nginx http access log file: "/usr/local/nginx/logs/access.log"
  nginx http client request body temporary files: "client_body_temp"
  nginx http proxy temporary files: "proxy_temp"
  nginx http fastcgi temporary files: "fastcgi_temp"
  nginx http uwsgi temporary files: "uwsgi_temp"
  nginx http scgi temporary files: "scgi_temp"
```

启动 停止 重启

凯盛软件

- 启动
  - /usr/local/nginx/sbin/nginx
- 重新加载配置文件
  - /usr/local/nginx/sbin/nginx –s reload
- 重启
  - /usr/local/nginx/sbin/nginx –s reopen
- 停止
  - /usr/local/nginx/sbin/nginx –s stop
- 检测配置文件是否存在语法错误
  - /usr/local/nginx/sbin/nginx -t

配置文件

凯盛软件

- 配置文件路径
  - /usr/local/nginx/conf/nginx.conf
- 组成部分
  - 全局配置：从配置文件开始到events块之间的部分，主要设置一些影响Nginx服务器整体运行的配置指令
  - events块：events模块来用指定nginx的工作模式和工作模式及连接数上限
  - http块：是Nginx中配置的重要部分，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放在这个模块中
  - server块：sever 模块是http的子模块，它用来定一个虚拟主机
  - location块：是nginx中用的最多的，也是最重要的模块了，负载均衡啊、反向代理、虚拟域名等都与它相关
  - upstream块：upstream 模块负债负载均衡模块，通过一个简单的调度算法来实现客户端IP到后端服务器的负载均衡

全局配置

凯盛软件

- 配置运行nginx服务器的用户/组
  - user user [group]； user为运行Nginx服务器的用户，group可选项，指定运行Nginx服务器的用户组
  - user root;
- 配置允许生成的work process数量
  - worker_processes auto | num;
  - 指定Nginx要开启的子进程数。每个Nginx进程平均耗费10M~12M内存
  - 一般1个进程就足够了，如果是多核CPU，建议指定和CPU的数量一样的进程数即可。例如配置为2，就是除了主进程外，再运行两个work process进程
- 配置Nginx进程pid文件的存放路径
  - pid file;
  - 默认存放在nginx的安装目录logs下，名字为nginx.pid
- 配置错误日志存放路径
  - error_log file | level;
  - 用来定义全局错误日志文件
  - 日志输出级别有debug、info、notice、warn、error、crit可供选择，其中，debug输出日志最为最详细，而crit输出日志最少

events块配置

凯盛软件

```
events {
    worker_connections  1024;
}
```

- worker_connections用于定义Nginx每个进程的最大连接数，即接收前端的最大请求数，默认是1024

http块配置

凯盛软件

```
http {
    include       mime.types;
    default_type  application/octet-stream;

    #log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
    #                  '$status $body_bytes_sent "$http_referer" '
    #                  '"$http_user_agent" "$http_x_forwarded_for"';

    #access_log  logs/access.log  main;

    sendfile        on;
    #tcp_nopush     on;

    #keepalive_timeout  0;
    keepalive_timeout  65;

    #gzip  on;
```

- include 来用设定文件的mime类型,类型在配置文件目录下的mime.type文件定义，来告诉nginx来识别文件类型
- default_type设定了默认的类型为二进制流，也就是当文件类型未定义时使用这种方式
- log_format name format；name为格式字符串的名字，format为日志的记录格式
- access_log path formatname; path为存放输出日志的路径及文件名称 formatname对应的是log_format的name值
- sendfile参数用于开启高效文件传输模式。将tcp_nopush和tcp_nodelay两个指令设置为on用于防止网络阻塞
- keepalive_timeout设置客户端连接保持活动的超时时间。在超过这个时间之后，服务器会关闭该连接
- gzip 配置结果的gzip压缩

server块配置

凯盛软件

```
server {
    listen       80;
    server_name  localhost;

    #root /var/html

    #charset koi8-r;

    #access_log  logs/host.access.log  main;
```

- listen用于指定虚拟主机的服务端口
- server_name用来指定IP地址或者域名，多个域名之间用空格分开
- root 表示在这整个server虚拟主机内，全部的root web根目录
- charset用于设置网页的默认编码格式
- access_log用来指定此虚拟主机的访问日志存放路径，main用于指定访问日志的输出格式
- index 全局定义访问的默认首页地址

location块的配置

凯盛软件

```
location / {
    root   html;
    index  index.html index.htm;
}
```

- root指令用于指定访问根目录时，虚拟主机的web目录，这个目录可以是相对路径（相对路径是相对于nginx的安装目录）。也可以是绝对路径
- index用于设定我们只输入域名后访问的默认首页地址，有个先后顺序：index.php index.html index.htm，如果没有开启目录浏览权限，又找不到这些默认首页，就会报403错误

location配置示例

凯盛软件

```
server {
    listen 80;
    server_name localhost;
    #当用户访问/server1时，将在/var/html/server1目录中查找index.html响应请求
    #当用户访问/server1/product.html时，将在/var/html/server1目录中查找product.html响应请求
    location /server1 {
        root /var/html;
        index index.html;
    }
    #当前用户访问/image/logo.png时，将在/var/html/image目录中查找logo.png响应请求
    location /image {
        root /var/html;
    }
}
```

- location中的路径分类
  - 标准URL，例如 /server1
  - 正则URL，例如 \.php$ 表示以php结尾的URL
- URL的匹配模式
  - = 用于标准URL前，要求请求字符串与URL严格匹配。如果匹配成功，就停止继续向下搜索并立即处理此请求
  - ~ 用于表示URL包含正则表达式，并区分大小写
  - ~* 用于表示URL包含正则表达式，并不区分大小写
  - ^~ 用于标准的URL前，要求服务器找到标识URL和请求字符串匹配度最高的location后,立即使用该location处理请求，而不再使用location块中的正则URL和请求字符串匹配
  - 如果URL包含有正则表达式，就必须使用~或~*开头

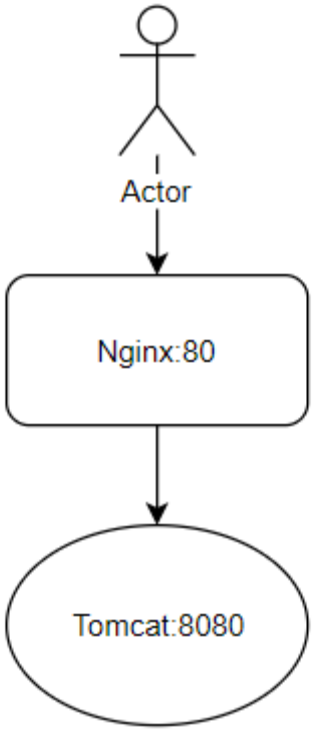
- 处理静态资源

```
server {
    listen 80;
    server_name localhost;
    #访问以image、css、js开头的URL时去/var/html/static路径中查找image、css、js子文件夹的资源响应请求
    location ~ ^/(image|css|js)/ {
        root /var/html/static;
        #关闭访问日志的记录
        access_log off;
        #缓存30天
        expires 30d;
    }
}
```

反向代理

凯盛软件

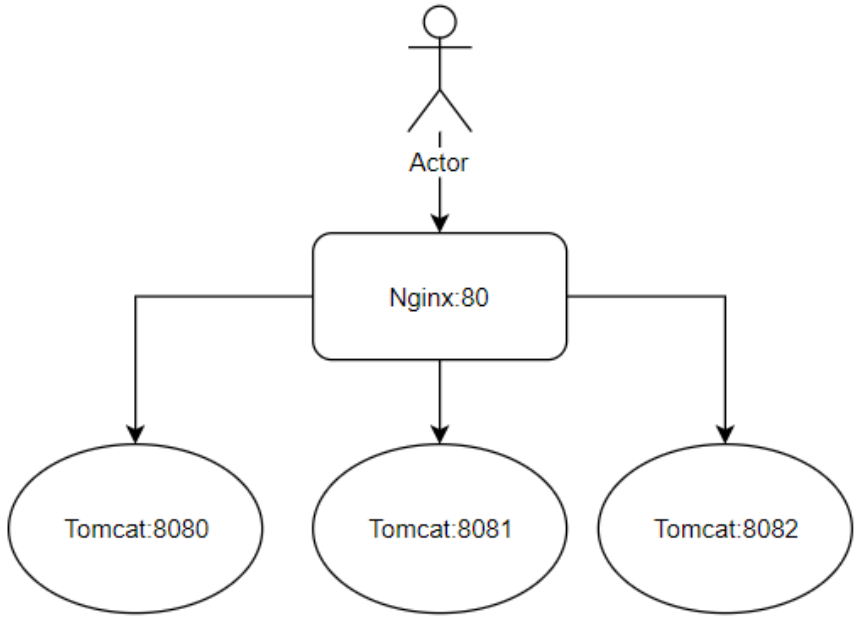
```
server {
    listen 80;
    server_name localhost;
    location / {
        #被代理服务器的地址
       proxy_pass http://localhost:8080;
        #nginx服务器与后端服务器建立连接的超时时间，默认为60s
        proxy_connect_timeout 60;
        #nginx服务器向后端服务器发出read请求后的超时时间，默认为60s
        proxy_read_timeout 60;
        #nginx服务器向后端服务器发出write请求后的超时时间，默认为60
        proxy_send_timeout 60;
        #nginx获取服务器接收到客户端请求的请求头信息，将新的请求头发送给代理服务器
       proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }
}
```



负载均衡

凯盛软件

```
upstream springboot {
    server 127.0.0.1:8080;
    server localhost:8081;
    server localhost:8082;
}
server {
    listen 80;
    server_name localhost;
    location / {
        proxy_pass http://springboot;
    }
}
```



- 默认情况下，服务器组收到请求后，按照轮询(Round-Robin,RR)的方式选择组内服务器响应请求。如果一个服务器在响应过程中出现错误，请求会交个组内下一个服务器进行响应，以此类推，直到响应正常返回。如果所有的服务器都出现错误，则返回最后一个服务器的处理结果
- upstream中的server指令可以设置组内服务器，除了配置服务器地址和端口外，还可以配置参数
    - weight=number，为服务器设置权重，权重值高的服务器优先处理请求，此时组内服务器的选择策略为加权轮询策略，默认值为1
    - backup 将服务器标记为备份服务器，只有当正常的服务器处于down状态或繁忙状态时，该服务器才会被用来处理请求
    - down 将服务器标记为无效状态，无效后将不会用于处理请求，通常与ip_hash指令配合使用
    - max_fails=number 设置请求失败的次数。在一定时间内，当对组内某台服务器请求失败次数超过配置值时，将服务器标记为down状态
    - fail_timeout=number 第一可以设置max_fails参数一定时间内的值；第二当服务器设置为down时，在该时间内不再检查该服务器状态，默认为10s

```
upstream springboot {
    server 127.0.0.1:8080 weight=5;
    server localhost:8081 weight=10 fail_timeout=5s max_fails=10;
    server localhost:8082 backup;
}
```

- ip_hash 指令用于实现保持会话功能，将某个客户端的多次请求定向到组内同一个服务器上，保证session的不丢失，只有该服务器down时，才会将请求转交给组内其他的服务器
- ip_hash 指令不能与server 中的 weight 参数一起使用

```
upstream springboot {
    ip_hash;
    server 127.0.0.1:8080;
    server localhost:8081;
    server localhost:8082;
}
```