# Nginx

# 作用：

1. 反向代理 2.负载均衡 3.动静分离

# Docker方式安装

docker run -d --restart=always --privileged=true --name nginx -p 80:80 -v /home/docker/nginx/nginx.conf:/etc/nginx/nginx.conf -v /home/docker/nginx/conf.d:/etc/nginx/conf.d nginx:latest

# 配置文件：

# Nginx负载均衡分配服务器策略：

## 轮询

轮询是默认的分配策略。每个请求按照时间顺序轮流分配给服务器，某个服务器宕机也能自动剔除。

## 权重

在配置服务器时增加权重配置，权重也是按照比例去轮询分配服务器。

配置样例：

如下所示，服务器一会分配五次后才会分配一次给服务器二

upstream myserver {

server 192.168.62.130:8080 weight=10;

server 192.168.62.132:8080 weight=2;

}

location / {

proxy\_pass http://myserver;

root /usr/share/nginx/html;

index index.html index.htm;

}

## ip\_hash

根据客户端ip，取余后分配服务器，保证每个ip固定访问一个服务器。可以解决session问题。

upstream myserver {

ip\_hash;

server 192.168.62.130:8080;

server 192.168.62.132:8080;

}

location / {

proxy\_pass http://myserver;

root /usr/share/nginx/html;

index index.html index.htm;

}

## Fair（第三方）

根据相应时间分配客户端请求

upstream myserver {

server 192.168.62.130:8080;

server 192.168.62.132:8080;

fair;

}

location / {

proxy\_pass http://myserver;

root /usr/share/nginx/html;

index index.html index.htm;

}

## url\_hash（第三方）

按访问url的hash结果来分配请求，使每个url定向到同一个后端服务器，后端服务器为缓存时比较有效。

upstream backserver {

server squid1:3128;

server squid2:3128;

hash $request\_uri;

hash\_method crc32;

}

location / {

proxy\_pass http://myserver;

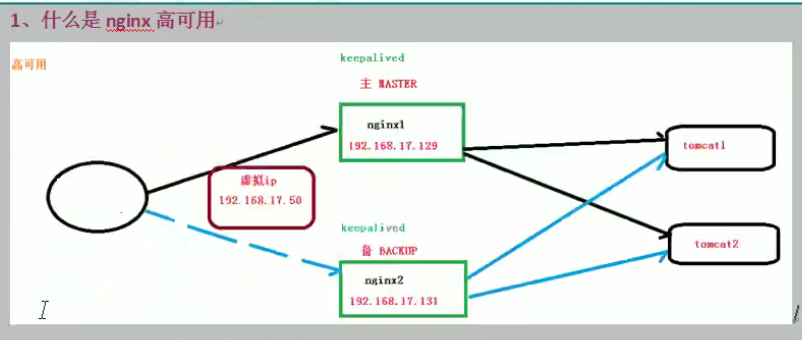
root /usr/share/nginx/html;

index index.html index.htm;

}

# Nginx高可用

使用keepalived服务给nginx服务器绑定一个虚拟ip，客户端通过虚拟ip访问时每个节点都能接收到请求，主节点会通过心跳机制告诉备份节点自己还活着，备份节点不会处理请求。nginx主服务器挂掉后，从服务器依然可以用这个虚拟ip访问。



# Nginx工作原理

Nginx服务工作时为多进程单线程，io多路复用技术。

进程分为master和worker进程。Master负责接收客户端请求，然后worker进程会进行争抢任务，并通过互斥锁保证一个任务只有一个进程获得。

好处：

1. 热部署：配置修改好后不需要重启启动即可生效（配置更新时因为是多进程的，没有工作任务的进程可以进行重新加载）
2. 降低服务器出问题的风险（多进程工作模式，使得单个进程挂掉后服务还能正常使用）

# Nginx参数设置：

1. 工作进程数量一般为cpu的 核数（由于多路复用技术，每个线程都能发挥最大的性能）

2.每个请求会占用work进程2（请求的静态资源）或者4（需要请求后台）个连接数据

Nginx auth模块

Nginx可以提供简单的的访问权限控制，访问nginx代理转发的资源时需要登录。

实现：

1.首先要安装httpd-tools

2.创建/etc/nginx/nginx\_auth\_conf 文件

3.配置auth认证的文件路径：

server {

listen 80;

server\_name example.com;

access\_log logs/access.log main;

location / {

root /data/www/project;

auth\_basic "auth.......";

auth\_basic\_user\_file /etc/nginx/nginx\_auth\_conf;

index index.php index.html;

}

}