**Linux下Nginx的安装**

在linux 中新建一个目录，用于保存ngnix压缩包，一般在/usr/local

安装依赖库： yum install gcc gcc-c++ zlib-devel pcre-devel openssl-devel openssl-libs openssl -y

cd到当前目录，下载 源码包 wget http://nginx.org/download/nginx-1.12.2.tar.gz （可下载最新的版本）

解压： tar -xzvf nginx-1.12.2.tar.gz （解压到非/usr/local/nginx目录）

配置： cd到解压后的目录　 ./configure --prefix=/usr/local/nginx 用来配置安装的目录

编译： make

安装：　　　　make install

查看安装路径： whereis nginx

将nginx安装目录添加到环境变量中：

vi /etc/profile

在最后添加　export PATH=/usr/local/nginx/sbin:$PATH

source !$ 生效

修改配置文件 参考 下节 **Nginx的基本配置**

启动nginx：　　　　　　　　 控制台运行　nginx

查看是否启动成功：　　　　　ps -ef | grep ngnix 默认会有两个nginx相关进程

查看ngnix端口：　　　　　 netstat -ntlp | grep nginx

开启防火墙的对应端口： firewall-cmd --zone=public --add-port=xxx/tcp --permanent

重启防火墙: systemctl restart firewalld.service

在浏览器中验证：　　　　　 浏览器中输入 ip和对应的端口号 ，如果有网页显示表示安装成功

常用控制命令

启动　　　　　　　　　　　　nginx

停止　　　　　　　　　　　　nginx -s stop

从容停止　　　　　　　　　　nginx -s quit

重新加载配置　　　　　　　　nginx -s reload

注册服务随系统自动启动

在 /usr/lib/systemd/system 中 建立nginx.service 的文件

　[Unit]

Description=nginxservice

After=network.target

[Service]

Type=forking

PIDFile=

ExecStart=/usr/local/nginx/sbin/nginx

ExecReload=

ExecStop=/usr/local/nginx/sbin/nginx -s stop

PrivateTmp=true

[Install]

WantedBy=multi-user.target

通过 systemctl 进行管理， 使其开机启动: systemctl enable nginx

**Windows下Nginx的安装**

**下载nginx windows安装包**

使用 [Xiaolei.ServicePacker](https://github.com/xl711436/Xiaolei.ServicePacker) 进行包装

使用[说明文档](http://www.cnblogs.com/xxssdxxs/articles/7192494.html)

当无法联网是参考如下：

将Xiaolei.ServicePacker编译后的bin 目录下的所有文件 copy到nginx目录。

其中 InstallService.bat（servicename变量）  ，UninstallService.bat（servicename变量），Xiaolei.ServicePacker.exe.config（MyServiceName配置项）  三个文件中配置服务名称的三个地方，名称必须一样 ，这个名称就是 最终部署的服务的名称

在启动参数模式，只需要配置Xiaolei.ServicePacker.exe.config 文件中  ExecuteFileName ，StartPara  ，StopPara  三个配置项，分别配置 为服务对应的可执行程序，启动时的参数和结束时的参数，对于ngnix而言，启动时不需要参数，结束时传-s stop 即可。（ExecuteFileName 配置为nginx.exe StartPara 为空字符串 StopPara 为 -s stop）

执行  InstallService.bat  文件 安装服务。

当需要卸载时，执行 UninstallService.bat 即可

**Nginx的基本配置**

配置文件的地址： conf/nginx.conf

配置示例：（linux下 删除stream 这个节点）

#user nobody;

worker\_processes 1;

#error\_log logs/error.log;

#error\_log logs/error.log notice;

#error\_log logs/error.log info;

#pid logs/nginx.pid;

events {

worker\_connections 1024;

}

stream{

upstream tcptest{

server 192.168.11.175:3389;

server 192.168.11.154:3389 backup;

}

server{

listen 2211;

proxy\_connect\_timeout 1s;

proxy\_timeout 3s;

proxy\_pass tcptest;

}

}

http{

include mime.types;

default\_type application/octet-stream;

sendfile on;

keepalive\_timeout 60;

upstream webtest{

server 192.168.11.153:80 weight=10;

server 192.168.11.154:80 weight=20;

}

server {

listen 8000;

server\_name localhost;

location / {

proxy\_pass http://webtest;

proxy\_set\_header Host $host:80;

proxy\_set\_header Cookie $http\_cookie;

proxy\_connect\_timeout 1;

proxy\_send\_timeout 30;

proxy\_read\_timeout 60;

client\_max\_body\_size 20m;

}

error\_page 500 502 503 504 /50x.html;

location = /50x.html {

root html;

}

}

}

**负载均衡配置**

其中 stream 用来 配置 tcp服务器 ，http 用来配置http服务器， upstream 用来配置负载均衡

upstream webtest{

server 192.168.11.153:80;

server 192.168.11.154:80;

}

轮询访问

upstream webtest{

server 192.168.11.153:80 weight=10;

server 192.168.11.154:80 weight=20;

}

weight 表示权重 基于权重访问

upstream webtest {

server 192.168.11.175:3389;

server 192.168.11.154:3389 backup;

}

backup 表示备份，主挂了才启用备

upstream webtest {

ip\_hash;

server 192.168.11.175:3389;

server 192.168.11.154:3389;

}

同一个ip访问同一个服务器

upstream webtest {

least\_conn;

server 192.168.11.175:3389;

server 192.168.11.154:3389 backup;

}

转发到连接数最少的服务器上

**和keeepalived的配合使用**

**环境设置，主备两台机都需要设置**

# vi /etc/sysconfig/selinux

#SELINUX=enforcing #注释掉

#SELINUXTYPE=targeted #注释掉

SELINUX=disabled #增加

setenforce 0 生效

**配置iptables**，添加VRRP通讯支持 iptables -A INPUT -d 224.0.0.18 -j ACCEPT （经对比，此步不需要）

**yum安装**：

推荐直接使用 yum -y install keepalived

**修改配置文件** /etc/keepalived/keepalived.conf

Master:

! Configuration File for keepalived

global\_defs {

router\_id master

}

vrrp\_script chk\_nginx {

script "/etc/keepalived/nginx\_check.sh"

interval 2

weight -20

}

vrrp\_instance VI\_1 {

state MASTER

interface eth0 # ifconfig 查看IP对应的网卡

virtual\_router\_id 220

mcast\_src\_ip 192.168.11.163 #localIP

priority 100

nopreempt

advert\_int 1

authentication {

auth\_type PASS

auth\_pass 1111

}

track\_script {

chk\_nginx

}

virtual\_ipaddress {

192.168.11.220 #vip

}

}

Slave 配置文件和Master的基本一致，不一致的地方有如下3点：

　　　　 01. router\_id 信息不一致

　　　　02. state 状态描述信息不一致 MASTER 和 BACKUP

　　　　 03. priority 主备竞选优先级数值不一致 主的要大于备的

添加检查脚本 vi /etc/keepalived/nginx\_check.sh

脚本内容：

#!/bin/bash

A=`ps -C nginx –no-header |wc -l`

if [ $A -eq 0 ];then

/usr/local/nginx/sbin/nginx

sleep 2

if [ `ps -C nginx --no-header |wc -l` -eq 0 ];then

killall keepalived

fi

fi

设置脚本权限 chmod 777 /etc/keepalived/nginx\_check.sh

设置 keepalived 开机自动启动 systemctl enable keepalived.service

启动 keepalived systemctl enable keepalived.service

查看启动状态 systemctl status keepalived.service

在浏览器中输入vip 来验证