Guide for Setting Up Nginx

12/2/2017

# 硬件参数

|  |  |  |
| --- | --- | --- |
| **Item** | **Specification** | **Comments** |
| RAM | 64GB |  |
| CPU | 8 Cores |  |
| OS Disk | 20 GB |  |
| Log Disk | 127 GB |  |
| Data Disk | 200 GB |  |
| Swap Disk | 128 GB |  |

# 开始之前

## 所需文件

请确保你获得以下文件:

* nginx-1.10.2.tar.gz
* pcre-8.39.tar.gz

## 所需工具

请准备好以下工具:

* WinSCP (<http://winscp.net/> )
* PuTTY (<http://www.putty.org/>)
* Notepad ++ (可选, <https://notepad-plus-plus.org/>) (WinSCP和Notepad++ 的集成可参见 <https://winscp.net/eng/docs/integration_editor>)

# 安装前置

本说明以以下假设为前提: 服务器满足硬件参数条件; Linux HyperV虚拟机集成服务或者VmWare虚拟机的VmWare Tools已安装.

关于Linux集成服务的更多信息请参阅:

<http://blogs.technet.com/b/virtualization/archive/2015/05/01/linux-integration-services-4-0-announcement.aspx>

## 更新Hostname

hostnamectl set-hostname **NginXName**

例如:

hostnamectl set-hostname NginX

## 安装前置

安装 open ssh：sudo apt-get install openssh-server

编辑配置文件，允许以 root 用户通过 ssh 登录：

sudo vi /etc/ssh/sshd\_config

找到：PermitRootLogin prohibit-password禁用

添加：PermitRootLogin yes

sudo service ssh restart

# 安装PCRE

## 准备 Tarball

使用WinSCP上传“pcre-8.39.tar.gz” 到路径**~**

## 解压Tarball

cd

tar zxvf pcre-8.39.tar.gz

## 编译PCRE前准备

如果系统没安装build-essential：

sudo apt-get install build-essential

虚拟机上的Ubuntu已经安装过openssl但是仍然不能进行openssl编程  
上网查找资料得到是由于未安装openssl-devel包，执行以下两条命令更新后就可以了  
sudo apt-get install openssl   
sudo apt-get install libssl-dev

## 编译，安装PCRE可执行文件

参考文档：<http://www.runoob.com/linux/nginx-install-setup.html>

<http://www.jb51.net/os/Ubuntu/64596.html>

cd

tar zxvf sudo tar -xzvf pcre-8.39.tar.gz

cd pcre-8.39.tar.gz

cd deps

cd /usr/local/src/pcre-8.39   
./configure   
make   
make install

验证：pcre-config --version

## 编译，安装Nginx可执行文件

tar zxvf nginx-1.10.2.tar.gz

cd nginx-1.10.2

./configure --prefix=/usr/local/webserver/nginx --with-http\_stub\_status\_module --with-http\_ssl\_module --with-pcre=/opt/pcre-8.39

make

make install

查看版本：

/usr/local/webserver/nginx/sbin/nginx –v

# Nginx配置

## nginx.conf脚本

编辑 “/usr/local/webserver/nginx/conf/nginx.conf”, 替换如下:

user nginx nginx;

worker\_processes 2;

error\_log /usr/local/webserver/nginx/logs/nginx\_error.log crit;

#error\_log logs/error.log;

#error\_log logs/error.log notice;

#error\_log logs/error.log info;

pid /usr/local/webserver/nginx/nginx.pid;

worker\_rlimit\_nofile 65535;

events {

use epoll;

worker\_connections 65535;

}

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"';

server\_names\_hash\_bucket\_size 128;

client\_header\_buffer\_size 32k;

large\_client\_header\_buffers 4 32k;

client\_max\_body\_size 8m;

sendfile on;

tcp\_nopush on;

keepalive\_timeout 60;

tcp\_nodelay on;

fastcgi\_connect\_timeout 300;

fastcgi\_send\_timeout 300;

fastcgi\_read\_timeout 300;

fastcgi\_buffer\_size 64k;

fastcgi\_buffers 4 64k;

fastcgi\_busy\_buffers\_size 128k;

fastcgi\_temp\_file\_write\_size 128k;

gzip on;

gzip\_min\_length 1k;

gzip\_buffers 4 16k;

gzip\_http\_version 1.0;

gzip\_comp\_level 2;

gzip\_types text/plain application/x-javascript text/css application/xml;

gzip\_vary on;

server {

listen 80;

#ssl on;

listen 443 ssl;

ssl\_certificate /usr/local/webserver/nginx/cert.crt;

ssl\_certificate\_key /usr/local/webserver/nginx/cert.key;

# create crt&key from .pfx: https://wiki.cac.washington.edu/display/infra/Extracting+Certificate+and+Private+Key+Files+from+a+.pfx+File

# directly create crt&key: https://www.digitalocean.com/community/tutorials/how-to-configure-nginx-with-ssl-as-a-reverse-proxy-for-jenkins

# directly create crt&key: cd /etc/nginx && sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/nginx/cert.key -out /etc/nginx/cert.crt

server\_name localhost;

index index.html index.htm index.php;

root /usr/local/webserver/nginx/html;

#charset koi8-r;

#access\_log logs/host.access.log main;

location /site/1/ {

proxy\_pass http://172.16.1.1/;

#proxy\_redirect default;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection "upgrade";

proxy\_set\_header X-real-ip $remote\_addr;

proxy\_set\_header X-Forwarded-For $remote\_addr;

#proxy\_http\_version 1.1;

#proxy\_set\_header Upgrade $http\_upgrade;

#proxy\_set\_header Connection "upgrade";

#proxy\_set\_header Host $host;

}

location /site/2/ {

proxy\_pass http://172.16.1.2/;

#proxy\_redirect default;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection "upgrade";

proxy\_set\_header X-real-ip $remote\_addr;

proxy\_set\_header X-Forwarded-For $remote\_addr;

#proxy\_http\_version 1.1;

#proxy\_set\_header Upgrade $http\_upgrade;

#proxy\_set\_header Connection "upgrade";

#proxy\_set\_header Host $host;

}

location /res/s {

proxy\_pass http://172.16.0.1:8800/;

proxy\_redirect default;

}

location /res/d {

proxy\_pass http://172.16.0.101:8800/;

proxy\_redirect default;

}

#location / {

# root html;

# index index.html index.htm;

#}

#location ~ .\*\.(gif|jpg|jpeg|png|bmp|swf|ico)$

#{

# expires 30d;

## access\_log off;

#}

location ~ .\*\.(php|php5)?$ {

#fastcgi\_pass unix:/tmp/php-cgi.sock;

fastcgi\_pass 127.0.0.1:9000;

fastcgi\_index index.php;

include fastcgi.conf;

}

#location ~ .\*\.(js|css)?$

#{

# alias http://172.16.1.1/bps-app-web/;

# expires 15d;

## access\_log off;

#}

#error\_page 404 /404.html;

# redirect server error pages to the static page /50x.html

#

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

location = /50x.html {

root html;

}

# proxy the PHP scripts to Apache listening on 127.0.0.1:80

#

#location ~ \.php$ {

# proxy\_pass http://127.0.0.1;

#}

# pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000

#

#location ~ \.php$ {

# root html;

# fastcgi\_pass 127.0.0.1:9000;

# fastcgi\_index index.php;

# fastcgi\_param SCRIPT\_FILENAME /scripts$fastcgi\_script\_name;

# include fastcgi\_params;

#}

# deny access to .htaccess files, if Apache's document root

# concurs with nginx's one

#

#location ~ /\.ht {

# deny all;

#}

}

# another virtual host using mix of IP-, name-, and port-based configuration

#

#server {

# listen 8000;

# listen somename:8080;

# server\_name somename alias another.alias;

# location / {

# root html;

# index index.html index.htm;

# }

#}

# HTTPS server

#server {

# listen 9292;

# server\_name 172.16.7.1;

# location / {

# proxy\_pass http://172.16.1.1:9292/;

# proxy\_http\_version 1.1;

# proxy\_set\_header Upgrade $http\_upgrade;

# proxy\_set\_header Connection "upgrade";

#

# }

# }

server {

listen 9292;

ssl on;

ssl\_certificate /usr/local/webserver/nginx/cert.crt;

ssl\_certificate\_key /usr/local/webserver/nginx/cert.key;

server\_name 172.16.7.1;

location /site/1/ {

proxy\_pass http://172.16.1.1:9292/;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection "upgrade";

proxy\_set\_header X-real-ip $remote\_addr;

proxy\_set\_header X-Forwarded-For $remote\_addr;

}

location /site/2/ {

proxy\_pass http://172.16.1.2:9292/;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection "upgrade";

proxy\_set\_header X-real-ip $remote\_addr;

proxy\_set\_header X-Forwarded-For $remote\_addr;

}

}

# HTTPS server

#

#server {

# listen 443 ssl;

# server\_name localhost;

# ssl\_certificate cert.pem;

# ssl\_certificate\_key cert.key;

# ssl\_session\_cache shared:SSL:1m;

# ssl\_session\_timeout 5m;

# ssl\_ciphers HIGH:!aNULL:!MD5;

# ssl\_prefer\_server\_ciphers on;

# location / {

# root html;

# index index.html index.htm;

# }

#}

}

检查配置文件ngnix.conf的正确性命令:

/usr/local/webserver/nginx/sbin/nginx -t

# 建立防火墙

## 更新防火墙配置

apt install firewalld

firewall-cmd --permanent --zone=internal --add-port=6379/tcp

firewall-cmd --permanent --zone=internal --add-port=16379/tcp

firewall-cmd --permanent --zone=internal --add-port=26379/tcp

firewall-cmd --permanent --zone=internal --add-source=**redis\_node\_1\_ip\_addr**

firewall-cmd --permanent --zone=internal --add-source=**redis\_node\_2\_ip\_addr**

firewall-cmd --permanent --zone=internal --add-source=**redis\_node\_3\_ip\_addrk**

firewall-cmd --permanent --zone=internal --add-source=**app\_server\_ip1**

firewall-cmd --permanent --zone=internal --add-source=**app\_server\_ip2**

firewall-cmd --reload

## 检查防火墙状态

firewall-cmd --zone=internal --list-all

输出应该如下:

internal

interfaces:

sources: ***ip\_addresses***

services: dhcpv6-client ipp-client mdns samba-client ssh

ports: **16379/tcp 6379/tcp 26379/tcp**

masquerade: no

forward-ports:

icmp-blocks:

rich rules:

appendonly yes

## 将NginX添加到开机启动服务

## 启动 NginX服务

/usr/local/webserver/nginx/sbin/nginx

## 检查NginX状态

从浏览器访问我们配置的站点ip：

如：http://172.16.7.1/

输入应该如下, 并且没有错误和警告.

# Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org/).  
Commercial support is available at [nginx.com](http://nginx.com/).

Thank you for using nginx.