## **nginx实现最简单的直播**

https://blog.csdn.net/leifukes/article/details/73244012

**我的环境**

[root@controller ~]# cat /etc/redhat-release

CentOS Linux release 7.2.1511 (Core)

[root@controller ~]# getenforce

Disabled

[root@controller ~]# systemctl stop firewalld

[root@controller ~]# hostname -I

10.0.0.11

##### 1：git拉取nginx-rtmp插件

mkdir -p /server/tools

cd /server/tools

git clone <https://github.com/arut/nginx-rtmp-module.git>

##### 2：编译安装nginx

useradd -s /sbin/nologin -M nginx

wget <http://nginx.org/download/nginx-1.14.0.tar.gz>

tar xf nginx-1.14.0.tar.gz

ls

cd nginx-1.14.0/

yum install pcre-devel -y

yum install openssl-devel.x86\_64 -y

./configure --user=nginx --group=nginx --with-http\_ssl\_module --prefix=/application/nginx --add-module=../nginx-rtmp-module

make

make install

##### 3:配置启动nginx

cd /application/nginx/conf/

grep -Ev '^$|#' nginx.conf.default >nginx.conf

vim nginx.conf

worker\_processes 1;

events {

worker\_connections 1024;

}

rtmp {

server {

listen 1935;

chunk\_size 4096;

application live {

     live on;

hls on;

hls\_path /application/nginx/html/live;

hls\_fragment 5s;

}

}

}

http {

include mime.types;

default\_type application/octet-stream;

sendfile on;

keepalive\_timeout 65;

server {

listen 80;

server\_name localhost;

location /live {

types {

application/vnd.apple.mpegurl m3u8;

video/mp2t ts;

}

alias /application/nginx/html/live;

expires -1;

add\_header Cache-Control no-cache;

}

location / {

root html;

index index.html index.htm;

}

}

}

#测试nginx语法

../sbin/nginx –t

启动nginx

../sbin/nginx

##### 4:使用EV录屏实现推流



串流地址: rtmp://10.0.0.11:1935/live

地址密钥随便填，我这里是test

如果你的地址密钥填写和一样的话，

开启直播之后，测试能否下载test.m3u8文件

<http://10.0.0.11/live/test.m3u8>

如果能够成功下载，恭喜你离成功很近了！

##### 5：配置站点首页

vi /application/nginx/html/index.html

<!DOCTYPE html>

<html lang="zh-CN">

<head>

<meta charset="UTF-8">

<title>前端播放m3u8格式视频</title>

<link rel="stylesheet" href="http://vjs.zencdn.net/5.5.3/video-js.css">

<script src="http://vjs.zencdn.net/5.5.3/video.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/videojs-contrib-hls/5.12.2/videojs-contrib-hls.js"></script>

</head>

<body>

<video id="myVideo" class="video-js vjs-default-skin vjs-big-play-centered" controls preload="auto" width="1080" height="708" data-setup='{}'>

<source id="source" src="http://10.0.0.11/live/test.m3u8" type="application/x-mpegURL">

</video>

</body>

<script>

// videojs 简单使用

var myVideo = videojs('myVideo',{

bigPlayButton : true,

textTrackDisplay : false,

posterImage: false,

errorDisplay : false,

})

myVideo.play() // 视频播放

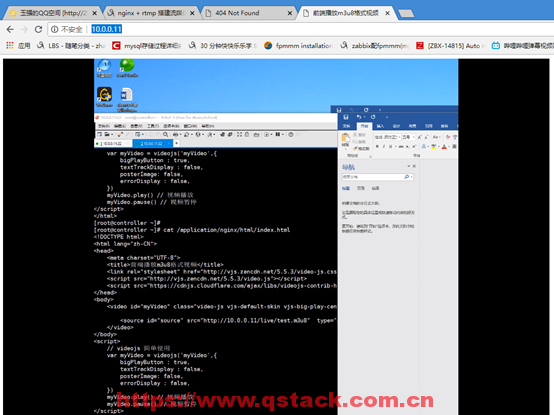
myVideo.pause() // 视频暂停

</script>

</html>

打开浏览器测试：

http://10.0.0.11/



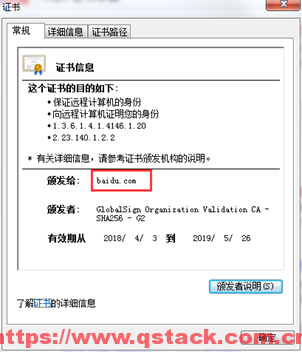
至此成功！！！

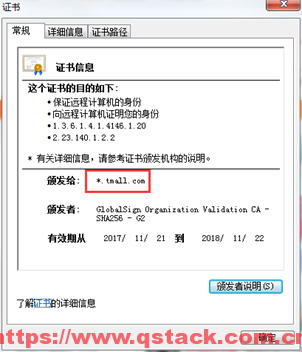
## **网站配置https过程**

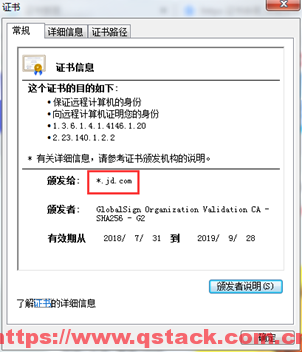
#### 一：申请https证书

如果你所在的是大公司，建议去https://www.globalsign.com申请https证书，为什么？

国内的大公司的https证书都是这家颁发的



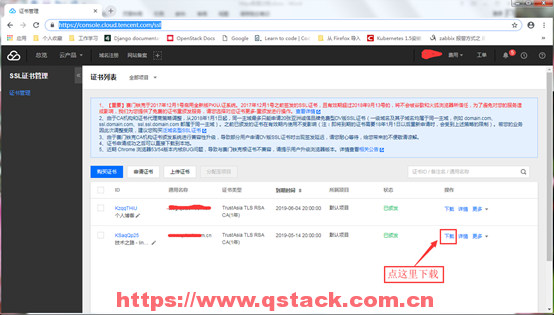




如果你和我一样使用的云主机，申请证书特别简单，腾讯云直接打开SSL证书管理，就可以申请，免费的，阿里云也有证书管理的接口，自己找一下咯



#### 二：下载证书



#### 三：配置web服务器，使支持https

这一步，其实很简单，因为你申请https的地方就有详细的教程

例如腾讯云的证书安装指引链接是https://cloud.tencent.com/document/product/400/4143

1.apache证书配置



2.nginx证书配置



3.IIS略

4.tomcat证书配置



#### 四：配置http——>https

很多时候用户还是会用http协议去访问，如果你只监听443，使用http就打不开网站了，so，这一步，我们需要配置http到https的跳转

1.apache的配置方法：

如果要针对整个网站全部执行HTTP自动转为HTTPS的功能，可以在httpd.conf文件中，增加:

RewriteEngine on

RewriteCond   %{HTTPS} !=on

RewriteRule   ^(.\*)  https://%{SERVER\_NAME}$1 [L,R]

2.nginx配置例子

server {

listen 80;

server\_name www.test.com;

rewrite ^(.\*)$ https://$host$1 permanent;

}

3.IIS略

4:tomcat

在web.xml末尾加上如下配置：

<security-constraint>

    <web-resource-collection >

              <web-resource-name >SSL</web-resource-name>

              <url-pattern>/\*</url-pattern>

       </web-resource-collection>

       <user-data-constraint>

       <transport-guarantee>CONFIDENTIAL</transport-guarantee>

       </user-data-constraint>

</security-constraint>

然后重启tomcat

#### 五：注意事项

https配置好之后，还会面临红绿的问题，绿的是最好的



证书无效或者过期都会显示下图



有时候证书有效也不一定是绿的



出现这种情况的原因是，当前页面中有部分url连接是http不是https，所以才会有这个提示。

总结：如果你的网站https需要显示是绿的，安全的，那么需要你所有的url地址都是https！

## **演示僵尸进程的形成**

僵尸进程：一个进程使用fork创建子进程，如果子进程退出，而父进程并没有调用wait或waitpid获取子进程的状态信息，那么子进程的进程描述符仍然保存在系统中。这种进程称之为僵死进程。在每个进程退出的时候,内核释放该进程所有的资源,包括打开的文件,占用的内存等。 但是仍然为其保留一定的信息(包括进程号the process ID,退出状态the termination status of the process,运行时间the amount of CPU time taken by the process等)。直到父进程通过wait / waitpid来取时才释放。 但这样就导致了问题，如果进程不调用wait / waitpid的话， 那么保留的那段信息就不会释放，其进程号就会一直被占用，但是系统所能使用的进程号是有限的，如果大量的产生僵死进程，将因为没有可用的进程号而导致系统不能产生新的进程. 此即为僵尸进程的危害，应当避免。

从系统角度来说，处理僵尸进程有两种方法：

1 找到僵死进程的父进程，kill掉父进程，那么僵死进程将变为孤儿进程，孤儿进程在系统中由init进程接管，init进程将回收僵死进程的资源

2 reboot系统，因为僵死进程是不可以被kill掉

为了模拟僵尸进程，特找到如下C语言的源代码

vi zombia.c

#include <sys/types.h>

#include <sys/wait.h>

#include <errno.h>

#include <unistd.h>

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

int main(int argc, char \*argv[])

{

pid\_t pid;

pid = fork();

if (pid == 0) {

int iPid = (int)getpid();

fprintf(stderr,"I am child,%d\n",iPid);

sleep(1);

fprintf(stderr, "Child exits\n");

return EXIT\_SUCCESS;

}

int iPid = (int)getpid();

fprintf(stderr,"I am parent,%d\n",iPid);

fprintf(stderr, "sleep....\n");

sleep(120);

fprintf(stderr, "parent exits\n");

return EXIT\_SUCCESS;

}

保存退出:wq!

使用gcc编译(没有gcc命令，自行安装)

gcc zombia.c

编译完成可以看到有一个a.out

执行./aout输出如下

[root@m03 ~]# ./a.out

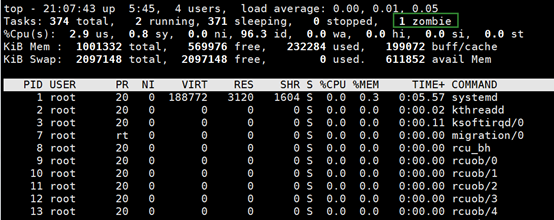
I am parent,23217

sleep....

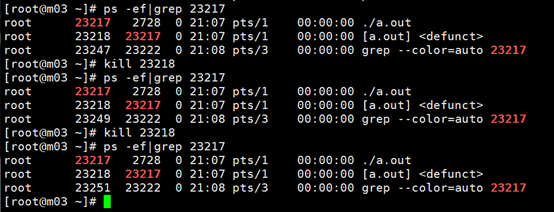
I am child,23218

Child exits

这时候top查看存在一个僵尸进程



ps -ef找到父进程23217



从上图可以看出无论怎么kill子进程23218，都已经杀不死了。

僵尸进程测试完毕！

## **rsync+nginx实现公网yum源**

2018-05-03 分类：[linux服务](https://oldqiang.com/archives/category/service), [实用技能](https://oldqiang.com/archives/category/skill) 阅读(1821) 评论(0)

rsync+nginx实现公网yum源

整个过程分3步：

1：nginx提供目录浏览功能

nginx直接yum安装，不废话，直接贴配置文件

[root@oldboyedu ~]# cat /etc/nginx/nginx.conf

worker\_processes auto;

events {

worker\_connections 1024;

}

http {

include mime.types;

charset utf-8;

default\_type application/octet-stream;

sendfile on;

autoindex on; #开启目录浏览功能

keepalive\_timeout 65;

server {

listen 80;

listen [::]:80;

server\_name localhost;

location / {

root html;

index index.html index.htm;

}

}

}

2：从上游yum源同步yum源到本地

直接贴定时任务

# rsync centos6 repos

30 21 \* \* \* /usr/bin/rsync -zaP --exclude-from /usr/share/nginx/html/rsync\_exclude2.txt rsync://rsync.mirrors.ustc.edu.cn/centos/7.4.1708 /usr/share/nginx/html/centos

00 22 \* \* \* /usr/bin/rsync -zaP --exclude-from /usr/share/nginx/html/rsync\_exclude.txt rsync://rsync.mirrors.ustc.edu.cn/centos/6.9 /usr/share/nginx/html/centos

00 21 \* \* \* /usr/bin/rsync -zaP --exclude-from /usr/share/nginx/html/rsync\_exclude.txt rsync://rsync.mirrors.ustc.edu.cn/epel/7/x86\_64 /usr/share/nginx/html/epel/7/

30 20 \* \* \* /usr/bin/rsync -zaP --exclude-from /usr/share/nginx/html/rsync\_exclude.txt rsync://rsync.mirrors.ustc.edu.cn/epel/6/x86\_64 /usr/share/nginx/html/epel/6/

从定时任务的配置，可以看出，我同步了centos6、7基础源和epel源，有的同学担心，这样会特别占用空间！是的，如果不启用过滤，全部同步，确实很占用空间！

下面我把rsync里面的--exclude-from文件贴出来，centos6和7稍微不同

centos6

[root@oldboyedu ~]# cat /usr/share/nginx/html/rsync\_exclude.txt

centosplus/

cloud/

contrib/

cr/

fasttrack/

isos/

sclo/

storage/

virt/

i386/

debug/

drpms/

centos7

[root@oldboyedu ~]# cat /usr/share/nginx/html/rsync\_exclude2.txt

atomic/

centosplus/

cloud/

configmanagement/

cr/

dotnet/

fasttrack/

isos/

nfv/

opstools/

paas/

rt/

sclo/

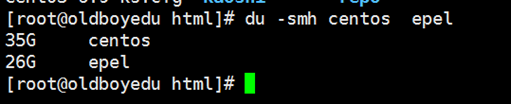
storage/

virt/

debug/

drpms/

最终4个源全部同步完，并且可用只占了60G左右



到这里已经能提供yum服务了，但是无法为下游提供同步服务，于是有了第三步

3：开启rsync --daemon模式

[root@oldboyedu ~]# cat /etc/rsyncd.conf

#rsync server

uid = nginx

gid = nginx

use chroot = no

max connections = 2000

timeout = 600

pid file = /var/run/rsyncd.pid

lock file = /var/run/rsync.lock

log file = /var/log/rsyncd.log

ignore errors

read only = true #只提供同步，只读足够

list = true             #允许查看列表，认证的什么的不需要配置

hosts allow = 0.0.0.0/0

#####################################

[centos]

path = /usr/share/nginx/html/centos

[epel]

path = /usr/share/nginx/html/epel

到这里，一个公网yum该有的功能都有了！

zabbix监控tomcat

2018-08-10 分类：[监控](https://oldqiang.com/archives/category/zabbix) 阅读(1584) 评论(0)

1.环境准备

首先准备两台机器

192.168.56.25 zabbix

192.168.56.22 tomcat

保证zabbix能够使用Template OS Linux监控tomcat

zabbix监控tomcat使用的是zabbix-java-gateway，zabbix-java-gateway需要java环境

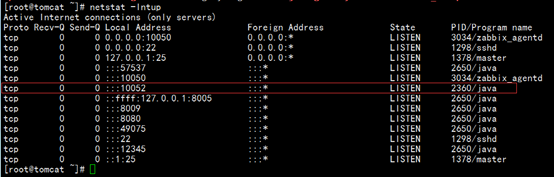
这里，我的zabbix-java-gateway就安装在tomcat上

2.在tomcat上安装zabbix-java-gateway-

[root@tomcat ~]# rpm -ivh zabbix-java-gateway-3.0.5-1.el6.x86\_64.rpm

[root@tomcat ~]# /etc/init.d/zabbix-java-gateway start

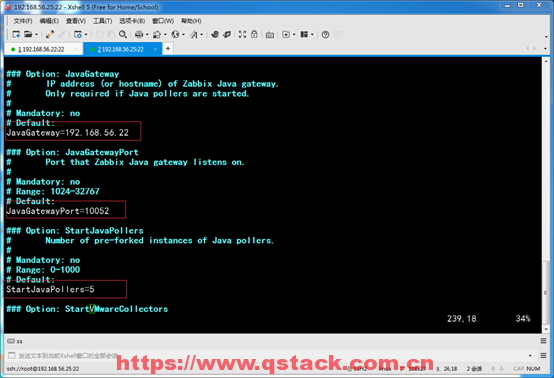
检查java-gateway启动情况，10052端口，代表正常



3.在zabbix server上修改主配置文件

[root@zabbix ~]# vim /etc/zabbix/zabbix\_server.conf

主要修改下面三个地方



然后重启zabbix-server

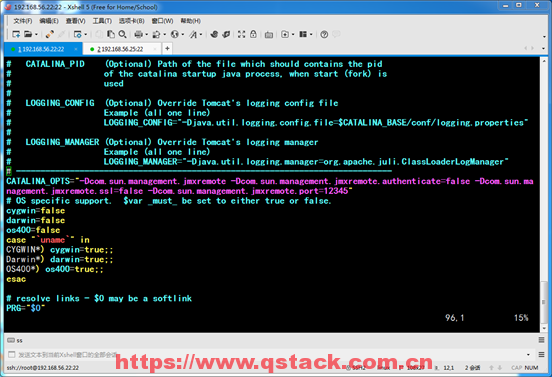
[root@zabbix ~]# /etc/init.d/zabbix-server restart

4.监控tomcat，需要开启jmx

[root@tomcat bin]# vim catalina.sh

CATALINA\_OPTS="-Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.authenticate=false -Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.port=12345"

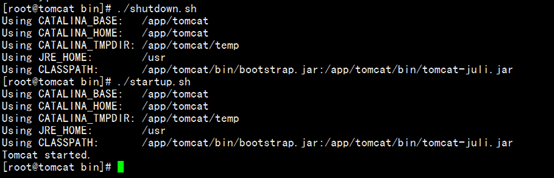
修改效果如下



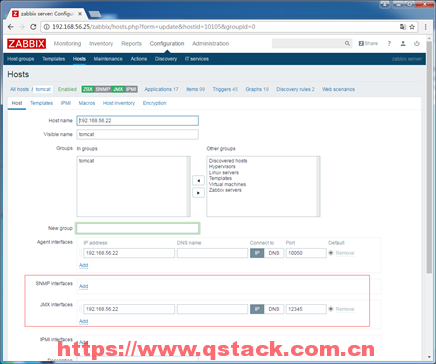
重启tomcat

[root@tomcat bin]# ./shutdown.sh

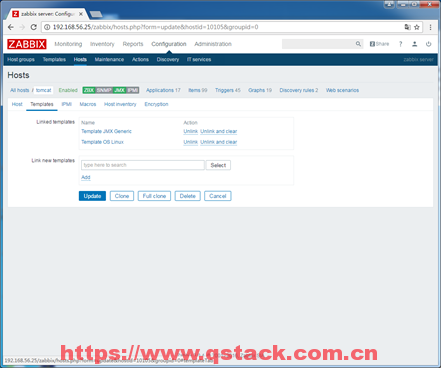
[root@tomcat bin]# ./startup.sh



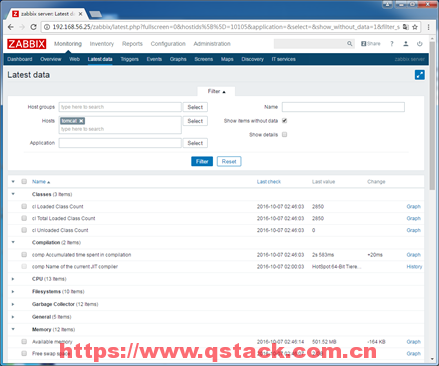
5.在zabbix web上添加监控



关联Template JMX Generic模板



一段时间之后，就有数据了



至此监控tomcat完成