链滴

java 开源博客 solo 实现 linux 服务器通过 nginx 安装并且实现域名转发

作者：441985745

原文链接：https://ld246.com/article/1516892494490

来源网站：链滴

许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

首先说说想要实现的效果,我有一个项目部署在服务器的tomcat上,假如我服务器的公网ip为 40.12.12.2 ,

我项目名称是 solo ,我还有一个域名是 javaweb.55555.io

那么,我想实现通过 javaweb.55555.io 就能访问到我的项目,要怎么实现呢

首先是将域名解析到公网ip上,这个自行百度,我是用的花生壳,然后就是剩下的两种方式

- 第一种方式: **修改tomcat**

* 将tomcat的server.xml中的端口修改为80

```
<Connector port="80" protocol="HTTP/1.1"
       connectionTimeout="20000"
       redirectPort="8443" />
```

* 在<Host>标签中添加你自己的项目

```
<Host name="localhost"  appBase="webapps"
     unpackWARs="true" autoDeploy="true">

  <Context path="" docBase="solo" reloadable="true" />
  <!-- SingleSignOn valve, share authentication between web applications
       Documentation at: /docs/config/valve.html -->
  <!--
  <Valve className="org.apache.catalina.authenticator.SingleSignOn" />
  -->

  <!-- Access log processes all example.
       Documentation at: /docs/config/valve.html
       Note: The pattern used is equivalent to using pattern="common" -->
  <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
       prefix="localhost_access_log" suffix=".txt"
       pattern="%h %l %u %t &quot;%r&quot; %s %b" />

</Host>
```

其中的 ```<Context>``` 标签就是你要进行处理的,将path设成空字符串即可

- 第二种方式 **利用nginx实现转发**

  - 首先是安装nginx,可以参考 **Centos安装nginx** ,这里不再赘述

不过要强调一点,如果之前tomcat占用了80端口,启动nginx会报错 端口已被占用 或者类似 # 98: Address already in use

你需要把tomcat端口改回8080或者随意什么值,因为默认安装nginx配置的是80端口

  - 好了下面是 **最重要的nginx配置**

```
user root;
```

```
#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;
}


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;

    server {
        listen       80;
        server_name  javaweb.55555.io;

        #charset koi8-r;

        #access_log  logs/host.access.log  main;

        location / {
            proxy_pass http://localhost:8080/;
        }

        #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       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;
    #    }
    #}

}
```

上面是完整的配置文件,最主的配置在这里

```
    server {
        listen       80;
        server_name  javaweb.55555.io;

        #charset koi8-r;

        #access_log  logs/host.access.log  main;

        location / {
            proxy_pass http://localhost:8080/;
        }
```

当外网访问域名javaweb.55555.io时,其实访问的是公网的80端口,此时nginx监听到了这个端口,则会将这个请求转到http://localhost:8080/上,那么我们的目的就实现了

同样的tomcat里同样需要<Context path="" docBase="solo" reloadable="true" />配置,否则就会以这个javaweb.55555.io/solo网址访问

如果是solo这个开源项目别忘了在latke.properties中进行如下配置

```
# Browser visit protocol
 serverScheme=http
 # Browser visit domain name
 serverHost=javaweb.55555.io
 # Browser visit port, 80 as usual, THIS IS NOT SERVER LISTEN PORT!
 serverPort=
```