项目上线和Nginx负载均衡

一、项目日志

1.1 日志

常见的日志处理：

1、记录日志到数据库

2、使用日志框架 记录文件中

3、ELK平台 日志采集

4、云解决方案

特点：

多 周期短

SpringBoot 推荐的日志 LogBack(Slf4j（Log4j2）)

支持的日志配置：

```
logback-spring.xml`, `logback-spring.groovy
```

日志级别：`ERROR`，`WARN`，`INFO`，`DEBUG`，或`TRACE`

二、项目上线

Java项目上线：war包

SpringBoot打成war步骤：

1、修改打包方式

在pom.xml中

使用

```
<packaging>war</packaging>
```

2、在开关类直接继承SpringBootServletInitializer并重写configure

``

```
@SpringBootApplication
@MapperScan("com.feri.mybatisplus_study.dao")
@EnableSwagger2
@ServletComponentScan //自定义Servlet或Filter生效
public class MybatisplusStudyApplication extends SpringBootServletInitializer {

    public static void main(String[] args) {
        SpringApplication.run(MybatisplusStudyApplication.class, args);
    }

    /**
     * 分页插件
     */
    @Bean
    public PaginationInterceptor paginationInterceptor() {
        PaginationInterceptor paginationInterceptor = new PaginationInterceptor();
        // paginationInterceptor.setLimit(你的最大单页限制数量，默认 500 条，小于 0 如 -1 不受限
制);
        return paginationInterceptor;
    }

    @Override
    protected SpringApplicationBuilder configure(SpringApplicationBuilder builder) {
        return builder.sources(MybatisplusStudyApplication.class);
    }
}
```

3、配置Tomcat运行

上线前必须确保本地测试没问题

上线步骤：

1、数据库同步到线上

2、项目打包

注意数据库连接更改为线上地址

3、项目打包之后进行测试

4、上传到线上的Tomcat发布路径 webapps

必须上传完整

5、访问测试

http://39.105.189.141:8081/mps/swagger-ui.html

压力测试：Jemeter

云测试：

阿里云

腾讯云

百度云

三、Nginx简介

http://nginx.org/en/

Nginx的功能包括基本HTTP功能和扩展功能。和Apache服务器一样，Nginx服务器为了提供更多的功能并且能够有效地扩展这些功能。每一个模块都提供了一个功能，通过编译这些功能模块来实现功能的扩展

可以实现代理、静态资源服务器、负载均衡（实现Tomcat集群）、邮件服务器等

四、Nginx实现负载均衡

4.1 准备工作

拥有Nginx服务器

http://note.youdao.com/noteshare?id=a0191b020585dcdd510280f91f77b421&sub=F3D71179E55643038720C162363593FD

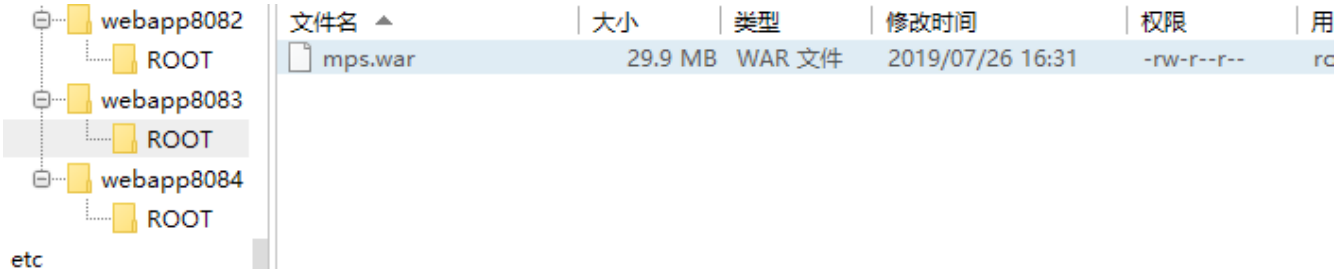
准备Tomcat服务器

三台Tomcat

```
[root@iz2ze9d7x8qidi9uuthpghz webapp8081]# docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS                      PORTS               NAMES
867ae2863491        tomcat              "catalina.sh run"        2 weeks ago         Exited (143) 2 weeks ago                        tomcat8082
24e387fe797f        tomcat              "catalina.sh run"        2 weeks ago         Exited (143) 2 weeks ago                        tomcat8084
4c8f078c8d3d        tomcat              "catalina.sh run"        2 weeks ago         Exited (143) 2 weeks ago                        tomcat8083
58f658f79363        mysql:5.7           "docker-entrypoint..."   4 weeks ago         Exited (0) 2 weeks ago                          mysql3312
```

4.2 基于Nginx实现Tomcat集群 负载均衡

1、上传war到3台Tomcat



2、启动3台Tomcat

```
[root@iz2ze9d7x8qidi9uuthpghz tomcat]# docker ps
CONTAINER ID        IMAGE                     COMMAND                  CREATED             STATUS              PORTS
                    NAMES
867ae2863491        tomcat                    "catalina.sh run"        2 weeks ago         Up 6 seconds        0.0.0.0:8082->8080/tcp
                    tomcat8082
24e387fe797f        tomcat                    "catalina.sh run"        2 weeks ago         Up 2 seconds        0.0.0.0:8084->8080/tcp
                    tomcat8084
4c8f078c8d3d        tomcat                    "catalina.sh run"        2 weeks ago         Up 4 seconds        0.0.0.0:8083->8080/tcp
                    tomcat8083
a54bdb290d05        mysql:5.7                 "docker-entrypoint..."   4 weeks ago         Up About an hour    33060/tcp, 0.0.0.0:3307->3306/tcp
                    mysqlmaster
6f7befe6e894        nginx                     "nginx -g 'daemon ..."   2 months ago        Up 2 weeks          0.0.0.0:81->80/tcp
                    nginx81
037356299b91        rabbitmq:management       "docker-entrypoint..."   7 months ago        Up 2 months         0.0.0.0:5671-5672->5671-5672/tcp,
-15672->15671-15672/tcp   rabbitmq
4cceeed56a83        redis                     "docker-entrypoint..."   7 months ago        Up 3 months         0.0.0.0:6379->6379/tcp
```

3、查看3台Tomcat的Ip地址

```
        "bridge": {
            "IPAMConfig": null,
            "Links": null,
            "Aliases": null,
            "NetworkID": "14a4626a284a292b4e8fde957a1254b3357a4309c8ed34d9c1323f31edc58eff",
            "EndpointID": "96019475eedd8346ee654555fec44f63b0390735b8f6bd4d587f9088cb7a24e4",
            "Gateway": "172.18.0.1",
            "IPAddress": "172.18.0.11",
            "IPPrefixLen": 16,
            "IPv6Gateway": "",
            "GlobalIPv6Address": "",
            "GlobalIPv6PrefixLen": 0,
            "MacAddress": "02:42:ac:12:00:0b"
        }
    }
```

三台容器的ip:

172.18.0.5 Tomcat8082

172.18.0.10 Tomcat8083

172.18.0.11 Tomcat8084

4、配置Nginx

编辑nginx.conf

修改:

`user root; 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;

```
upstream mytomcat{
    server 172.18.0.5:8080 weight=10;
    server 172.18.0.10:8080 weight=50;
    server 172.18.0.11:8080 weight=10;
}
#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  localtomcat;

    #charset koi8-r;

    #access_log  logs/host.access.log  main;

    location / {
       # root   html;
       #index  index.html index.htm;
        proxy_connect_timeout 50;
        proxy_read_timeout 10;
        proxy_send_timeout 20;
        proxy_pass http://mytomcat;
    }

    #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$ {
```

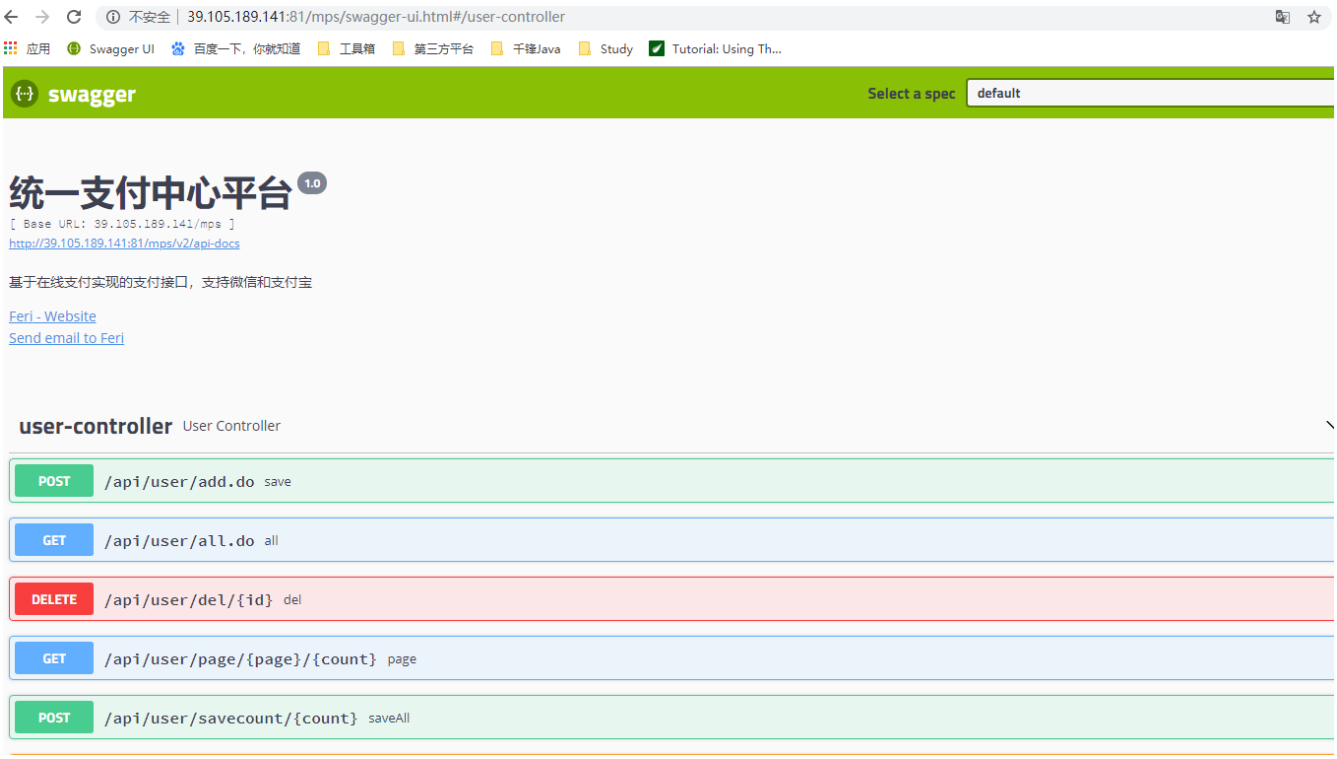
```
    #    proxy_pass   http://127.0.0.1;
    #}

    # 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;
```

```
#     }
#}
}
```

5、启动Nginx并测试



MIME类型：标记文件的类型

格式：大类型/小类型

eg:

text/html

image/jpg

image/*