发布文档

准备环境

> 1.配置可以共享session的tomcat
>
> 2.创建docker容器
>
> 3.上传war包 重启容器

配置可以共享session的tomcat

参考连接https://blog.csdn.net/lzc409973859/article/details/51981079

`本质为替换掉tomcat默认的session管理`

> 1. 下载RedisSessionManager 源代码 https://github.com/jcoleman/tomcat-redis-session-manager

> 2. 修改源代码 `修改RedisSessionManager类中的initializeSerializer()方法`

```
  private void initializeSerializer() throws ClassNotFoundException, IllegalAccessException,
InstantiationException {
    log.info("Attempting to use serializer :" + serializationStrategyClass);
    serializer = (Serializer) Class.forName(serializationStrategyClass).newInstance();
    Loader loader = null;
    if (getContainer() != null) {
      loader = getContainer().getLoader();
    }
    ClassLoader classLoader = null;
    if (loader != null) {
    serializer.setClassLoader(classLoader);
    }
  }
  private void initializeSerializer() throws ClassNotFoundException, IllegalAccessException,
InstantiationException {
    log.info("Attempting to use serializer :" + serializationStrategyClass);
    serializer = (Serializer) Class.forName(serializationStrategyClass).newInstance();
    Loader loader = null;
    Context context = this.getContext();
    if (context != null) {
      loader = context.getLoader();
    }
    ClassLoader classLoader = null;
    if (loader != null) {
      classLoader = loader.getClassLoader();
    }
    serializer.setClassLoader(classLoader);
  }
```

> 3. 使用idea maven 打包

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.yuncai</groupId>
    <artifactId>build_tomcat_session_redis</artifactId>
    <version>1.0-SNAPSHOT</version>

    <properties>
        <maven.compiler.source>8</maven.compiler.source>
        <maven.compiler.target>8</maven.compiler.target>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.apache.tomcat</groupId>
            <artifactId>tomcat-catalina</artifactId>
            <version>8.0.33</version>
        </dependency>
        <dependency>
            <groupId>redis.clients</groupId>
            <artifactId>jedis</artifactId>
            <version>2.7.2</version>
        </dependency>
    </dependencies>
    <build>
        <plugins>
            <plugin> <!-- 打jar包 -->
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-compiler-plugin</artifactId>
                <version>3.0</version>
                <configuration>
                    <!-- 源代码编译版本 -->
                    <source>1.8</source>
                    <!-- 目标平台编译版本 -->
                    <target>1.8</target>
                    <!-- 字符集编码 -->
                    <encoding>UTF-8</encoding>
                </configuration>
            </plugin>
        </plugins>
    </build>
</project>
```

4. 将 `build_tomcat_session_redis` `jedis-2.7.2.jar` `commons-pool2-2.3.jar` 拷贝到tomcat的lib文件夹下面

   这里可以把这些jar放到lib的自定义文件夹。以便与docker映射。

   下边是 将jar放入tomcat lib 文件夹下的 rslib文件夹的配置

- 修改tomcat `conf` 文件夹下 `catalina.properties` 配置文件

在common.loader后边添加"${catalina.home}/lib/rslib/*.jar"

```
common.loader="${catalina.base}/lib","${catalina.base}/lib/*.jar","${catalina.home}/lib",
"${catalina.home}/lib/*.jar","${catalina.home}/lib/rslib/*.jar"
```

> 5. 修改tomcat配置 使其使用自定义SessionManager 并配置redis

- 修改tomcat `conf` 文件夹下 `context.xml`

```
<Valve className="com.orangefunction.tomcat.redissessions.RedisSessionHandlerValve" />
<Manager className="com.orangefunction.tomcat.redissessions.RedisSessionManager"
    host="www.cyc-fund.com.cn"
    port="16379"
    password="******"
    database="0"
    maxInactiveInterval="300" />
```

创建docker容器

```
##创建卷 用于docker映射
docker volume create www_tomcat_bin
docker volume create www_tomcat_lib_rslib
docker volume create www_tomcat_conf
docker volume create wb_tomcat_webapp
docker volume create wb_tomcat_logs
docker volume create wb_tomcat_log4j_logs

docker run -di --name=wb_tomcat \
##时区文件映射
-v /etc/localtime:/etc/localtime \
##业务 服务上传文件路径映射
-v /home/upload:/home/upload \
##业务 服务上传文件路径映射
-v /home/virtualPath:/home/virtualPath \
-v www_tomcat_bin:/usr/local/tomcat/bin \
-v www_tomcat_lib_rslib:/usr/local/tomcat/lib/rslib \
-v www_tomcat_conf:/usr/local/tomcat/conf \
-v wb_tomcat_webapp:/usr/local/tomcat/webapps \
-v wb_tomcat_logs:/usr/local/tomcat/logs \
##log4j日志映射
-v wb_tomcat_log4j_logs:/usr/logs \
-p 8010:8080 tomcat:8.0.33-jre8
```

测试出现的问题

> 1.时区问题

- 把宿主机环境映射到容器中 -v /etc/localtime:/etc/localtime
- 修改tomcat `catalina.sh` 文件

```
JAVA_OPTS="$JAVA_OPTS -Duser.timezone=GMT+08"
```

> 2. 经nginx代理后tomcat获取 `scheme` `servername` `port` 为代理后的信息

- 修改nginx配置文件

```
location /
       {
        proxy_pass http://127.0.0.1:8010/;
        proxy_set_header Host $host:$server_port;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-Proto $scheme;
       }
```

- 修改tomcat `server.xml` 添加如下配置

```
<Valve className="org.apache.catalina.valves.RemoteIpValve"  remoteIpHeader="X-Forwarded-For"  protocolHeader="X-Forwarded-Proto"  protocolHeaderHttpsValue="https"/>
```

发布版本

> 1. 关闭docker容器并 删除 `/var/lib/docker/volumes/wb_tomcat_webapp/_data` 下 `ROOT` 文件夹

```
docker stop wb_tomcat
cd /var/lib/docker/volumes/wb_tomcat_webapp/_data
rm -rf *
```

> 2. 上传war包到 `/var/lib/docker/volumes/wb_tomcat_webapp/_data` 下

> 3. 重启docker 容器

```
docker start wb_tomcat
```

不停机发布

创建第二个容器 待容器启动成功后修改nginx 配置文件 并使其重新加载配置文件。

> 1. 创建容器

```
##创建卷 用于docker映射
docker volume create www_tomcat_webapp
docker volume create www_tomcat_logs
docker volume create www_tomcat_log4j_logs

docker run -di --name=www_tomcat \
```

```
##时区文件映射
-v /etc/localtime:/etc/localtime \
##业务 服务上传文件路径映射
-v /home/upload:/home/upload \
##业务 服务上传文件路径映射
-v /home/virtualPath:/home/virtualPath \
-v www_tomcat_bin:/usr/local/tomcat/bin \
-v www_tomcat_lib_rslib:/usr/local/tomcat/lib/rslib \
-v www_tomcat_conf:/usr/local/tomcat/conf \
-v www_tomcat_webapp:/usr/local/tomcat/webapps \
-v www_tomcat_logs:/usr/local/tomcat/logs \
##log4j日志映射
-v www_tomcat_log4j_logs:/usr/logs \
-p 8000:8080 tomcat:8.0.33-jre8
```

> 2. 上传war包到 `/var/lib/docker/volumes/www_tomcat_webapp/_data` 下

> 3. 启动容器 并测试是否启动成功

```
docker start www_tomcat
curl http://127.0.0.1:8000
```

> 4. 修改nginx配置文件 `proxy_pass`

```
location /
       {
        ######
        proxy_pass http://127.0.0.1:8000/;
        proxy_set_header Host $host:$server_port;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-Proto $scheme;
       }
```

> 5. nginx重新加载配置文件

```
/etc/init.d/nginx reload
```

日常发布

上述不停机发布为第一次时的操作

之后 切换 `www` 和 `wb` 两个容器， 操作 2 3 4 5步(注意对应容器目录)即可。

回滚方案

操作不停机发布中第 4 5 步即可