

```
1 FROM openjdk:8u342-jre
2 ENV LANG C.UTF-8
3 VOLUME /tmp
4 COPY PKU_GeekGame_Ant_Web-1.0.0-SNAPSHOT.jar web.jar
5 COPY PKU_GeekGame_Ant_Backend-1.0.0-SNAPSHOT.jar backend.jar
6 COPY PKU_GeekGame_Ant_Bonus-1.0.0-SNAPSHOT.jar bonus.jar
7 COPY flag1.txt /root/flag1.txt
8 COPY flag2.txt /root/flag2.txt
9 COPY flag3.txt /root/flag3.txt
10
11 RUN echo \
12 "#!/bin/sh\n"\
13 "nohup java -jar /backend.jar --server.port=8079 &\n"\
14 "nohup java -jar /bonus.jar --server.port=8080 &"\
15 >> /start.sh
16
17 RUN chmod +x /start.sh
18
19 CMD nohup sh -c "/start.sh && java -jar /web.jar --server.port=80"
20 EXPOSE 80
```

```
1 package com.alipay.pkuctf;
2
3 import org.springframework.context.annotation.Bean;
4 import org.springframework.context.annotation.Configuration;
5 import org.springframework.security.config.annotation.web.builders.HttpSecurity;
6 import org.springframework.security.config.annotation.web.builders.WebSecurity;
7 import org.springframework.security.config.annotation.web.configuration.
  WebSecurityConfigurerAdapter;
8 import org.springframework.security.web.firewall.HttpFirewall;
9 import org.springframework.security.web.firewall.StrictHttpFirewall;
10
11 @Configuration
12 public class WebSecurityConfig extends WebSecurityConfigurerAdapter {
13     @Override
14     public void configure(WebSecurity web) throws Exception {
15         super.configure(web);
16     }
17
18
19     @Override
20     protected void configure(HttpSecurity http) throws Exception {
21         http.formLogin()
22             .successForwardUrl("/admin")
23             .and()
24             .authorizeRequests()
25             .antMatchers("/admin").hasAnyRole()
26             .antMatchers("/admin/query").hasAnyRole()
27             .antMatchers("/admin/source_bak").hasAnyRole()
28             .antMatchers("/login", "/*", "/favicon.ico").permitAll()
29             .anyRequest()
30             .authenticated()
31             .and()
32             .csrf()
33             .disable();
34     }
35
36     @Bean
37     HttpFirewall httpFirewall() {
38         StrictHttpFirewall strictHttpFirewall = new StrictHttpFirewall();
39         strictHttpFirewall.setUnsafeAllowAnyHttpMethod(true);
40         strictHttpFirewall.setAllowBackSlash(true);
41         strictHttpFirewall.setAllowUrlEncodedSlash(true);
42         strictHttpFirewall.setAllowUrlEncodedDoubleSlash(true);
43         strictHttpFirewall.setAllowUrlEncodedPeriod(true);
44         strictHttpFirewall.setAllowUrlEncodedPercent(true);
45         return strictHttpFirewall;
46     }
47 }
48
49
```