链滴

Docker 环境下使用 SpringBootAdmin2.x 教程

作者：liumapp

原文链接：https://ld246.com/article/1536894516521

来源网站：链滴

Spring Boot Admin 2.x是一个用于监测管理微服务应用的程序，通过Eureka注册中心发现服务，然获取服务的运行状态、日志信息、并提供一套UI界面供系统运维人员检查，同时结合Spring Security行登录权限认证

首先上完整的Github项目代码： liumapp/spring-boot-admin-in-docker

然后是完整的Wiki文档：使用文档

所需依赖

- spring boot 2.0.2
- spring cloud Finchley.RELEASE
- codecentric的spring-boot-admin 2.0.2
- docker及docker-compose
- maven

快速启动

更新admin-server和admin-client的application.yml配置文件

```
spring:
    profiles:
        active: docker
```

使用以下命令安装docker镜像

```
./build-image.sh
```

使用以下命令启动docker容器

```
docker-compose up -d
```

等待几十秒后，用浏览器访问 http://localhost:8766

登录用户名是 : admin

登录密码是 : adminadmin

你可以在admin-server的application.yml中更改账号密码

使用以下命令停止容器：

```
docker-compose down
```

使用以下命令删除docker镜像：

```
./rm-image.sh
```

日志管理

首先请clone项目到本地

如果您只希望了解跟日志管理相关的内容，那么请使用git命令切换到v1.2.0的tag

```
git checkout v1.2.0
```

在这个版本下，我们基于v1.0.0的版本，对admin-client进行了一些配置上的改动：

```
logging:
  file: /usr/local/tomcat/project/spring-boot-admin-in-docker/log/admin-client.log
  pattern:
    file: "%clr(%d{yyyy-MM-dd HH:mm:ss.SSS}){faint} %clr(%5p) %clr(${PID}){magenta} %clr(---)
faint} %clr([%15.15t]){faint} %clr(%-40.40logger{39}){cyan} %clr(:){faint} %m%n%wEx"
management:
  endpoints:
    web:
      exposure:
        include: "*"
```

上面的改动中，我们通过management开放了所有的actuator节点信息，因为在springboot2.x系列，默认只开放了三个，如果您希望在admin-server中查询到详细信息，您需要全部开启他们

然后logging.file指向日志文件的存放地址，请确保该目录具有可写权限

之后再依次运行admin-eureka、admin-client跟admin-server

访问浏览器的 http://localhost:8766/ ，在点击admin-client进入详情页，我们可以发现所有的配置息包括日志信息将会罗列出来

配置Spring Security

如果您只希望了解跟Spring Security相关的内容首先请将项目切换到v1.3.0版本

```
git checkout v1.3.0
```

在之前的版本中，我们并没有引入spring security

这意味着admin-server管理控制台随便是谁都可以登录，这在本地开放环境下是没有什么影响的

但是如果发布到线上呢?

所以接下来要实现的功能，就是给admin-server添加一个登录登出的界面跟按钮

首先我们要对admin-server引入spring security

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-security</artifactId>
```

```
</dependency>
```

随后进行配置：

```
spring:
  security:
    user:
      name: "admin"
      password: "adminadmin"

eureka:
    metadata-map:
      user.name: "admin"
      user.password: "adminadmin"
```

user.name与user.password便是登录的账号与密码

接下来修改启动类的代码：

```
@Configuration
@EnableAutoConfiguration
@EnableDiscoveryClient
@EnableAdminServer
public class AdminServerMain {

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

    @Configuration
    public static class SecuritySecureConfig extends WebSecurityConfigurerAdapter {

        private final String adminContextPath;

        public SecuritySecureConfig(AdminServerProperties adminServerProperties) {
            this.adminContextPath = adminServerProperties.getContextPath();
        }

        @Override
        protected void configure(HttpSecurity http) throws Exception {
            // @formatter:off
            SavedRequestAwareAuthenticationSuccessHandler successHandler = new SavedRequ
stAwareAuthenticationSuccessHandler();
            successHandler.setTargetUrlParameter("redirectTo");
            successHandler.setDefaultTargetUrl(adminContextPath + "/");

            http.authorizeRequests()
                    .antMatchers(adminContextPath + "/assets/**").permitAll()
                    .antMatchers(adminContextPath + "/login").permitAll()
                    .anyRequest().authenticated()
                    .and()
                    .formLogin().loginPage(adminContextPath + "/login").successHandler(successHa
dler).and()
```

```
                .logout().logoutUrl(adminContextPath + "/logout").and()
                .httpBasic().and()
                .csrf()
                .csrfTokenRepository(CookieCsrfTokenRepository.withHttpOnlyFalse())
                .ignoringAntMatchers(
                        adminContextPath + "/instances",
                        adminContextPath + "/actuator/**",
                        adminContextPath + "/logout"
                );
            // @formatter:on
        }

    }

}
```

启动类代码主要是参考spring boot admin官方手册上的

但是他们的官方手册有一个bug：

当你完全按照官方手册上来的做，你会发现点击"log out"按钮的时候，会报403异常

我附上的代码把这个bug解决掉了

接下来配置admin-client端，只需要在其注册到eureka的时候，附上admin-server配置的账号密码可：

```
eureka:
  instance:
    metadata-map:
      user.name: "admin"
      user.password: "adminadmin"
```

Docker环境下运行

接下来我们利用docker-compose将admin-client、admin-server以及admin-eureka部署到docke环境下面运行

如果您只希望了解跟Docker环境下运行的内容，请利用git命令，将项目代码切换到v2.1.0版本

```
git checkout v2.1.0
```

版本切换后，项目目录下面会多出三个文件：build-image.sh、rm-image.sh和docker-compose.yml

- build-image.sh

脚本文件，用于安装三个微服务(admin-client、admin-server和admin-eureka)的docker镜像

- rm-image.sh

脚本文件，用于删除三个微服务的docker镜像

ps：要删除镜像，必须在镜像生成的容器处于stop状态下才可以执行

- docker-compose.yml

在执行完build-image.sh之后，通过docker-compose编排工具，启动容器的配置文件
具体启动命令为：

```
docker-compose up -d
```

停止命令为：

```
docker-compose down
```

利用docker-compose up -d命令启动成功后，我们可以访问浏览器的admin-server界面

您也可以通过Docker的容器工具：kitmatic来查看容器的运行状态