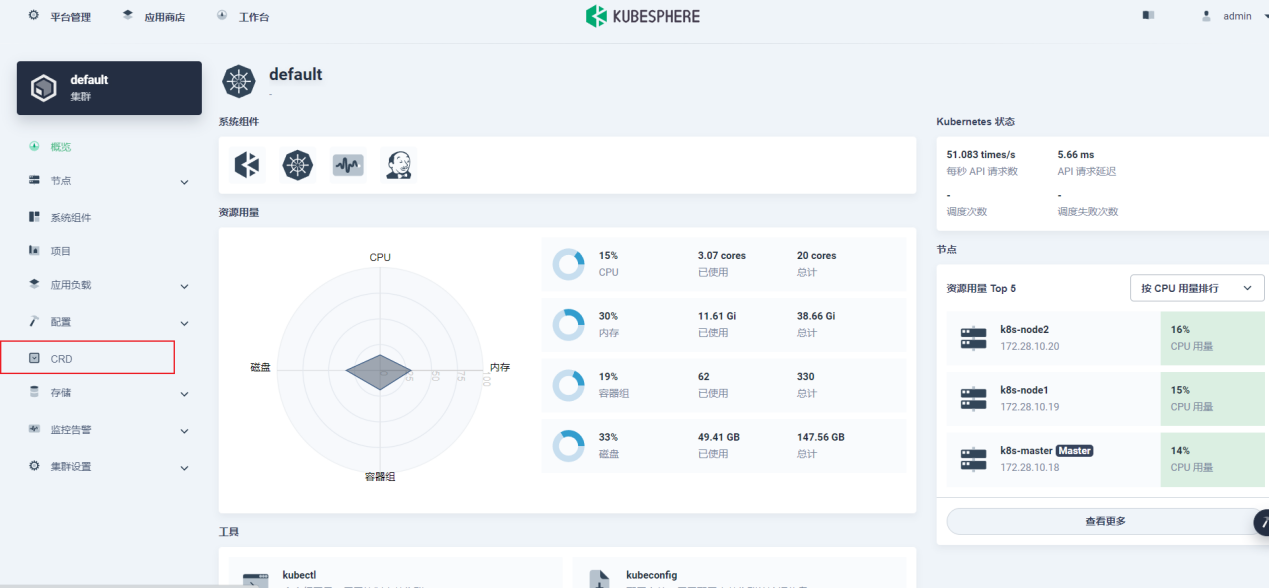
**3 kubesphere 无法运行devops流水线**

**没有安装openldap导致**

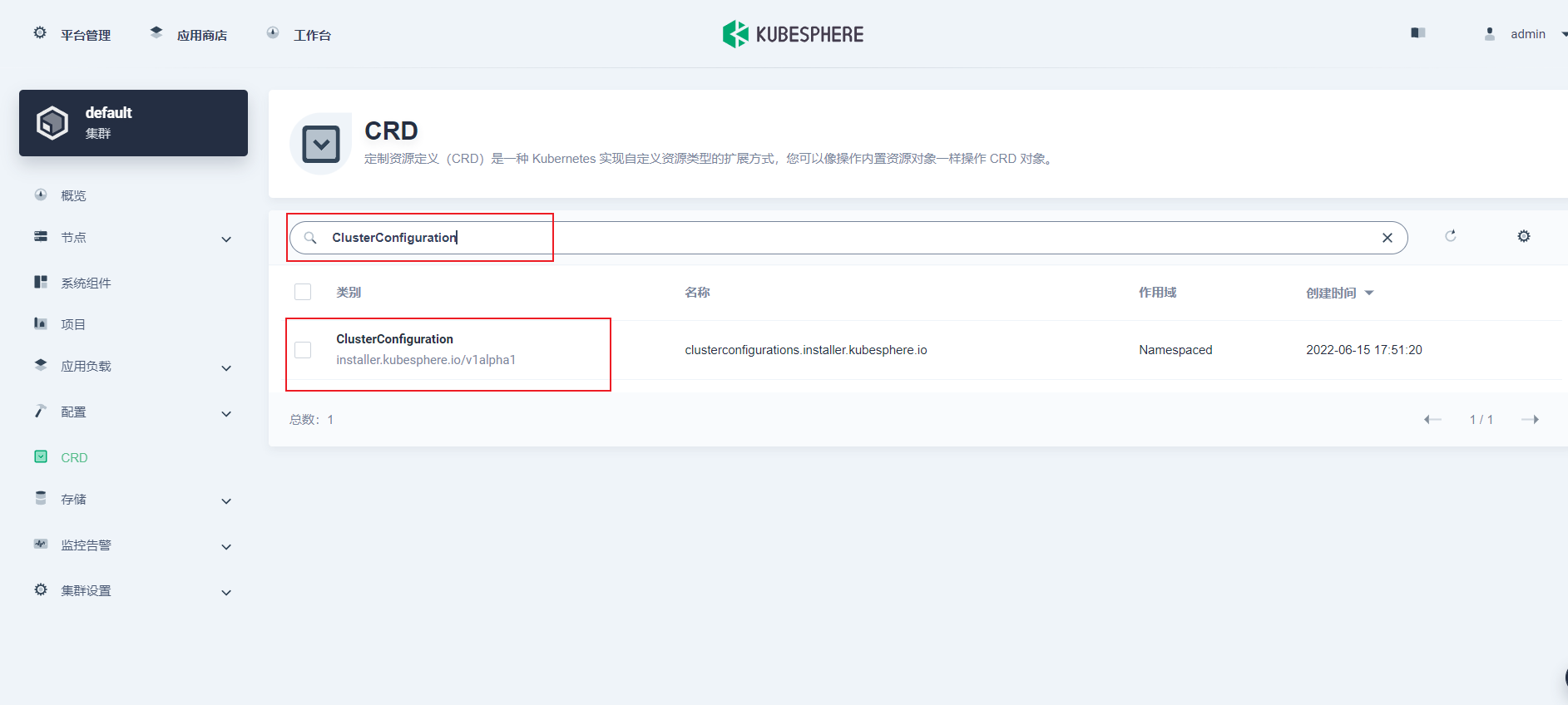
|  |
| --- |
| 2022-06-18 03:07:13.662+0000 [id=245] WARNING i.k.j.d.a.KubesphereApiTokenAuthenticator#authenticate: API token matched for user dev-zhao but the impersonation failed  org.acegisecurity.userdetails.UsernameNotFoundException: User dev-zhao not found in directory.  at org.acegisecurity.ldap.search.FilterBasedLdapUserSearch.searchForUser(FilterBasedLdapUserSearch.java:126)  at hudson.security.LDAPSecurityRealm$LDAPUserDetailsService.loadUserByUsername(LDAPSecurityRealm.java:1320)  at hudson.security.LDAPSecurityRealm$LDAPUserDetailsService.loadUserByUsername(LDAPSecurityRealm.java:1273)  at jenkins.security.ImpersonatingUserDetailsService.loadUserByUsername(ImpersonatingUserDetailsService.java:30)  at hudson.model.User.getUserDetailsForImpersonation(User.java:387)  at io.kubesphere.jenkins.devops.auth.KubesphereApiTokenAuthenticator.authenticate(KubesphereApiTokenAuthenticator.java:47)  at jenkins.security.BasicHeaderProcessor.doFilter(BasicHeaderProcessor.java:79)  at hudson.security.ChainedServletFilter$1.doFilter(ChainedServletFilter.java:87)  at org.acegisecurity.context.HttpSessionContextIntegrationFilter.doFilter(HttpSessionContextIntegrationFilter.java:249)  at hudson.security.HttpSessionContextIntegrationFilter2.doFilter(HttpSessionContextIntegrationFilter2.java:67)  at hudson.security.ChainedServletFilter$1.doFilter(ChainedServletFilter.java:87)  at hudson.security.ChainedServletFilter.doFilter(ChainedServletFilter.java:90)  at hudson.security.HudsonFilter.doFilter(HudsonFilter.java:171)  at org.eclipse.jetty.servlet.ServletHandler$CachedChain.doFilter(ServletHandler.java:1618)  at org.kohsuke.stapler.compression.CompressionFilter.doFilter(CompressionFilter.java:51)  at org.eclipse.jetty.servlet.ServletHandler$CachedChain.doFilter(ServletHandler.java:1618)  at hudson.util.CharacterEncodingFilter.doFilter(CharacterEncodingFilter.java:82)  at org.eclipse.jetty.servlet.ServletHandler$CachedChain.doFilter(ServletHandler.java:1618)  at org.kohsuke.stapler.DiagnosticThreadNameFilter.doFilter(DiagnosticThreadNameFilter.java:30)  at org.eclipse.jetty.servlet.ServletHandler$CachedChain.doFilter(ServletHandler.java:1618)  at jenkins.security.SuspiciousRequestFilter.doFilter(SuspiciousRequestFilter.java:36)  at org.eclipse.jetty.servlet.ServletHandler$CachedChain.doFilter(ServletHandler.java:1618)  at org.eclipse.jetty.servlet.ServletHandler.doHandle(ServletHandler.java:549)  at org.eclipse.jetty.server.handler.ScopedHandler.handle(ScopedHandler.java:143)  at org.eclipse.jetty.security.SecurityHandler.handle(SecurityHandler.java:578)  at org.eclipse.jetty.server.handler.HandlerWrapper.handle(HandlerWrapper.java:127)  at org.eclipse.jetty.server.handler.ScopedHandler.nextHandle(ScopedHandler.java:235)  at org.eclipse.jetty.server.session.SessionHandler.doHandle(SessionHandler.java:1610)  at org.eclipse.jetty.server.handler.ScopedHandler.nextHandle(ScopedHandler.java:233)  at org.eclipse.jetty.server.handler.ContextHandler.doHandle(ContextHandler.java:1369)  at org.eclipse.jetty.server.handler.ScopedHandler.nextScope(ScopedHandler.java:188)  at org.eclipse.jetty.servlet.ServletHandler.doScope(ServletHandler.java:489)  at org.eclipse.jetty.server.session.SessionHandler.doScope(SessionHandler.java:1580)  at org.eclipse.jetty.server.handler.ScopedHandler.nextScope(ScopedHandler.java:186)  at org.eclipse.jetty.server.handler.ContextHandler.doScope(ContextHandler.java:1284)  at org.eclipse.jetty.server.handler.ScopedHandler.handle(ScopedHandler.java:141)  at org.eclipse.jetty.server.handler.HandlerWrapper.handle(HandlerWrapper.java:127)  at org.eclipse.jetty.server.Server.handle(Server.java:501)  at org.eclipse.jetty.server.HttpChannel.lambda$handle$1(HttpChannel.java:383)  at org.eclipse.jetty.server.HttpChannel.dispatch(HttpChannel.java:556)  at org.eclipse.jetty.server.HttpChannel.handle(HttpChannel.java:375)  at org.eclipse.jetty.server.HttpConnection.onFillable(HttpConnection.java:272)  at org.eclipse.jetty.io.AbstractConnection$ReadCallback.succeeded(AbstractConnection.java:311)  at org.eclipse.jetty.io.FillInterest.fillable(FillInterest.java:103)  at org.eclipse.jetty.io.ChannelEndPoint$1.run(ChannelEndPoint.java:104)  at org.eclipse.jetty.util.thread.strategy.EatWhatYouKill.runTask(EatWhatYouKill.java:336)  at org.eclipse.jetty.util.thread.strategy.EatWhatYouKill.doProduce(EatWhatYouKill.java:313)  at org.eclipse.jetty.util.thread.strategy.EatWhatYouKill.tryProduce(EatWhatYouKill.java:171)  at org.eclipse.jetty.util.thread.strategy.EatWhatYouKill.run(EatWhatYouKill.java:129)  at org.eclipse.jetty.util.thread.ReservedThreadExecutor$ReservedThread.run(ReservedThreadExecutor.java:375)  at org.eclipse.jetty.util.thread.QueuedThreadPool.runJob(QueuedThreadPool.java:806)  at org.eclipse.jetty.util.thread.QueuedThreadPool$Runner.run(QueuedThreadPool.java:938)  at java.lang.Thread.run(Thread.java:748)  2022-06-18 03:07:13.662+0000 [id=245] WARNING h.i.i.InstallUncaughtExceptionHandler#handleException: Caught unhandled exception with ID 2f192d9a-c1d9-4f27-9e89-4bf64ea1fe0e  org.acegisecurity.userdetails.UsernameNotFoundException: User dev-zhao not found in directory.  at org.acegisecurity.ldap.search.FilterBasedLdapUserSearch.searchForUser(FilterBasedLdapUserSearch.java:126)  at hudson.security.LDAPSecurityRealm$LDAPUserDetailsService.loadUserByUsername(LDAPSecurityRealm.java:1320)  at hudson.security.LDAPSecurityRealm$LDAPUserDetailsService.loadUserByUsername(LDAPSecurityRealm.java:1273)  at jenkins.security.ImpersonatingUserDetailsService.loadUserByUsername(ImpersonatingUserDetailsService.java:30)  at hudson.model.User.getUserDetailsForImpersonation(User.java:387)  at io.kubesphere.jenkins.devops.auth.KubesphereApiTokenAuthenticator.authenticate(KubesphereApiTokenAuthenticator.java:47)  Caused: javax.servlet.ServletException  at io.kubesphere.jenkins.devops.auth.KubesphereApiTokenAuthenticator.authenticate(KubesphereApiTokenAuthenticator.java:54)  at jenkins.security.BasicHeaderProcessor.doFilter(BasicHeaderProcessor.java:79)  at hudson.security.ChainedServletFilter$1.doFilter(ChainedServletFilter.java:87)  at org.acegisecurity.context.HttpSessionContextIntegrationFilter.doFilter(HttpSessionContextIntegrationFilter.java:249)  at hudson.security.HttpSessionContextIntegrationFilter2.doFilter(HttpSessionContextIntegrationFilter2.java:67)  at hudson.security.ChainedServletFilter$1.doFilter(ChainedServletFilter.java:87)  at hudson.security.ChainedServletFilter.doFilter(ChainedServletFilter.java:90)  at hudson.security.HudsonFilter.doFilter(HudsonFilter.java:171)  at org.eclipse.jetty.servlet.ServletHandler$CachedChain.doFilter(ServletHandler.java:1618)  at org.kohsuke.stapler.compression.CompressionFilter.doFilter(CompressionFilter.java:51)  at org.eclipse.jetty.servlet.ServletHandler$CachedChain.doFilter(ServletHandler.java:1618)  at hudson.util.CharacterEncodingFilter.doFilter(CharacterEncodingFilter.java:82)  at org.eclipse.jetty.servlet.ServletHandler$CachedChain.doFilter(ServletHandler.java:1618)  at org.kohsuke.stapler.DiagnosticThreadNameFilter.doFilter(DiagnosticThreadNameFilter.java:30)  at org.eclipse.jetty.servlet.ServletHandler$CachedChain.doFilter(ServletHandler.java:1618)  at jenkins.security.SuspiciousRequestFilter.doFilter(SuspiciousRequestFilter.java:36)  at org.eclipse.jetty.servlet.ServletHandler$CachedChain.doFilter(ServletHandler.java:1618)  at org.eclipse.jetty.servlet.ServletHandler.doHandle(ServletHandler.java:549)  at org.eclipse.jetty.server.handler.ScopedHandler.handle(ScopedHandler.java:143)  at org.eclipse.jetty.security.SecurityHandler.handle(SecurityHandler.java:578)  at org.eclipse.jetty.server.handler.HandlerWrapper.handle(HandlerWrapper.java:127)  at org.eclipse.jetty.server.handler.ScopedHandler.nextHandle(ScopedHandler.java:235)  at org.eclipse.jetty.server.session.SessionHandler.doHandle(SessionHandler.java:1610)  at org.eclipse.jetty.server.handler.ScopedHandler.nextHandle(ScopedHandler.java:233)  at org.eclipse.jetty.server.handler.ContextHandler.doHandle(ContextHandler.java:1369)  at org.eclipse.jetty.server.handler.ScopedHandler.nextScope(ScopedHandler.java:188)  at org.eclipse.jetty.servlet.ServletHandler.doScope(ServletHandler.java:489)  at org.eclipse.jetty.server.session.SessionHandler.doScope(SessionHandler.java:1580)  at org.eclipse.jetty.server.handler.ScopedHandler.nextScope(ScopedHandler.java:186)  at org.eclipse.jetty.server.handler.ContextHandler.doScope(ContextHandler.java:1284)  at org.eclipse.jetty.server.handler.ScopedHandler.handle(ScopedHandler.java:141)  at org.eclipse.jetty.server.handler.HandlerWrapper.handle(HandlerWrapper.java:127)  at org.eclipse.jetty.server.Server.handle(Server.java:501)  at org.eclipse.jetty.server.HttpChannel.lambda$handle$1(HttpChannel.java:383)  at org.eclipse.jetty.server.HttpChannel.dispatch(HttpChannel.java:556)  at org.eclipse.jetty.server.HttpChannel.handle(HttpChannel.java:375)  at org.eclipse.jetty.server.HttpConnection.onFillable(HttpConnection.java:272)  at org.eclipse.jetty.io.AbstractConnection$ReadCallback.succeeded(AbstractConnection.java:311)  at org.eclipse.jetty.io.FillInterest.fillable(FillInterest.java:103)  at org.eclipse.jetty.io.ChannelEndPoint$1.run(ChannelEndPoint.java:104)  at org.eclipse.jetty.util.thread.strategy.EatWhatYouKill.runTask(EatWhatYouKill.java:336)  at org.eclipse.jetty.util.thread.strategy.EatWhatYouKill.doProduce(EatWhatYouKill.java:313)  at org.eclipse.jetty.util.thread.strategy.EatWhatYouKill.tryProduce(EatWhatYouKill.java:171)  at org.eclipse.jetty.util.thread.strategy.EatWhatYouKill.run(EatWhatYouKill.java:129)  at org.eclipse.jetty.util.thread.ReservedThreadExecutor$ReservedThread.run(ReservedThreadExecutor.java:375)  at org.eclipse.jetty.util.thread.QueuedThreadPool.runJob(QueuedThreadPool.java:806)  at org.eclipse.jetty.util.thread.QueuedThreadPool$Runner.run(QueuedThreadPool.java:938)  at java.lang.Thread.run(Thread.java:748) |

**3.1 Kubesphere开启openldap插件**

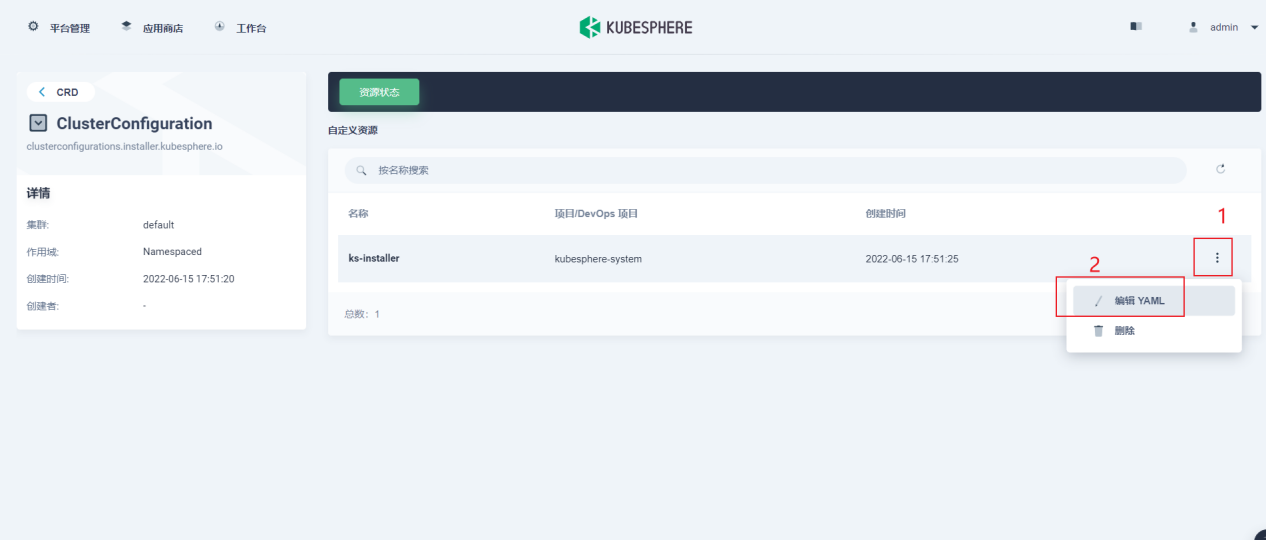
**3.2点击左侧CRD**

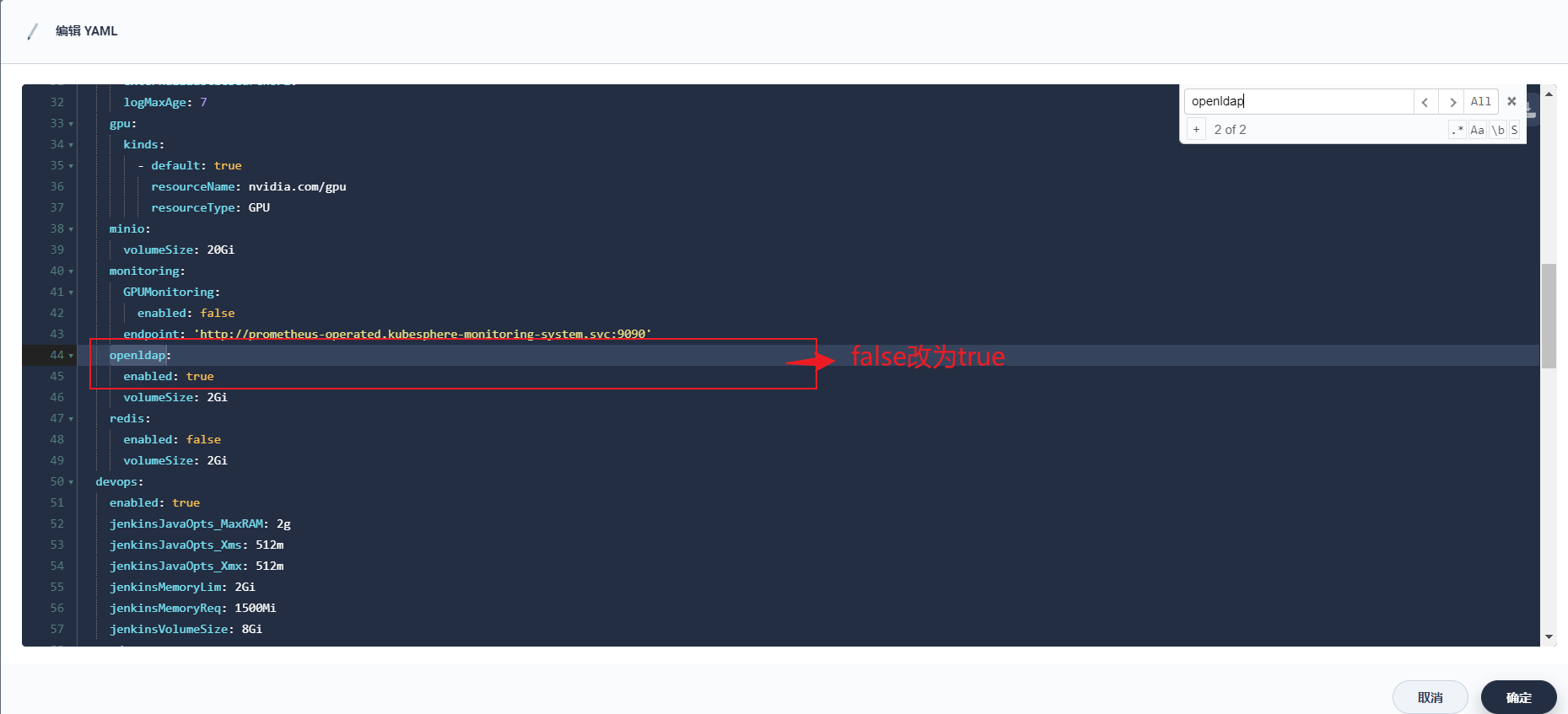


**3.3搜索ClusterConfiguration**

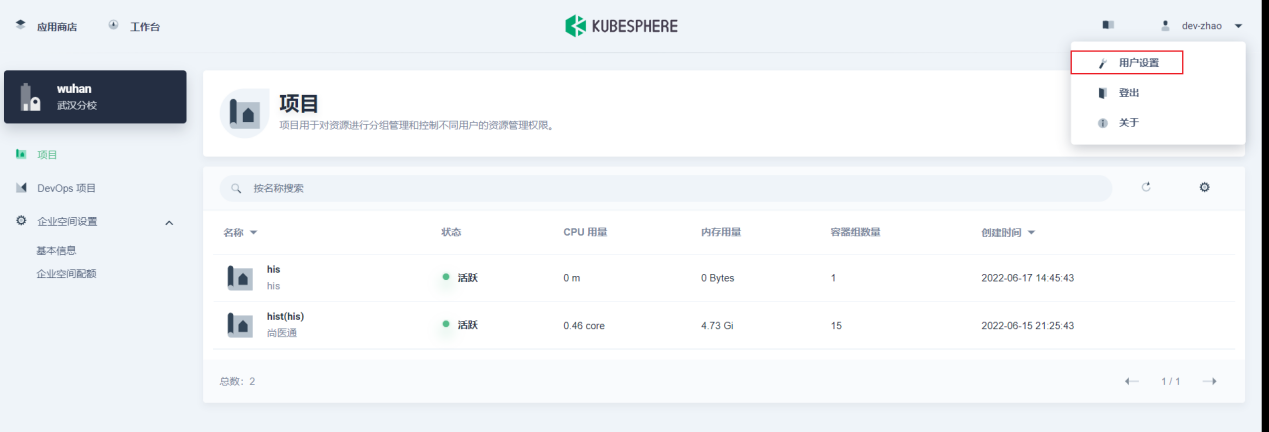


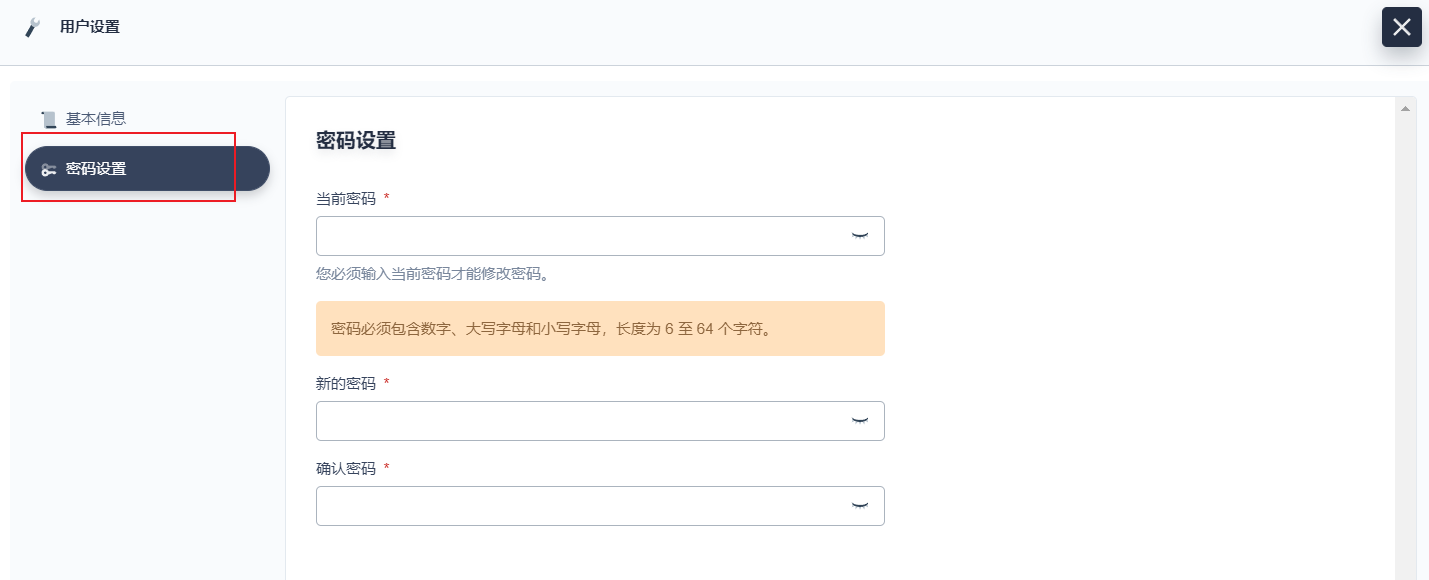
**3.4 编辑yaml**





**3.5开启后还得重置用户密码 才能同步账号密码到openldap**





**3.6参考URL**

<https://github.com/kubesphere/ks-devops/blob/master/docs/faq.md>

**4 kubesphere默认用户密码**

**Ks Jenkins也是一样**

|  |
| --- |
| 账号是：admin  密码是：P@88w0rd |

**5 部署报错**

|  |
| --- |
| Starting Kubernetes deployment  Loading configuration: /home/jenkins/agent/workspace/his-devopshmhfk/yygh-parent-devops/service/service-statistics/deploy/deploy.yml  ERROR: ERROR: java.lang.RuntimeException: io.kubernetes.client.openapi.ApiException: Bad Request  hudson.remoting.ProxyException: java.lang.RuntimeException: io.kubernetes.client.openapi.ApiException: Bad Request |

|  |
| --- |
| stage('service-user 部署到dev环境') {  agent none  steps {  kubernetesDeploy(configs: 'service/service-user/deploy/\*\*', enableConfigSubstitution: true, kubeconfigId: "$KUBECONFIG\_CREDENTIAL\_ID")  }  } |

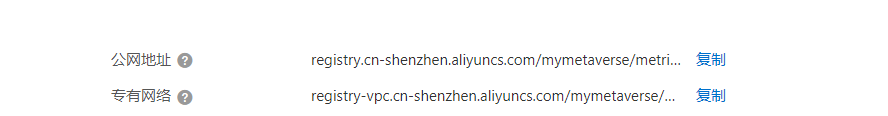
改为下面形式

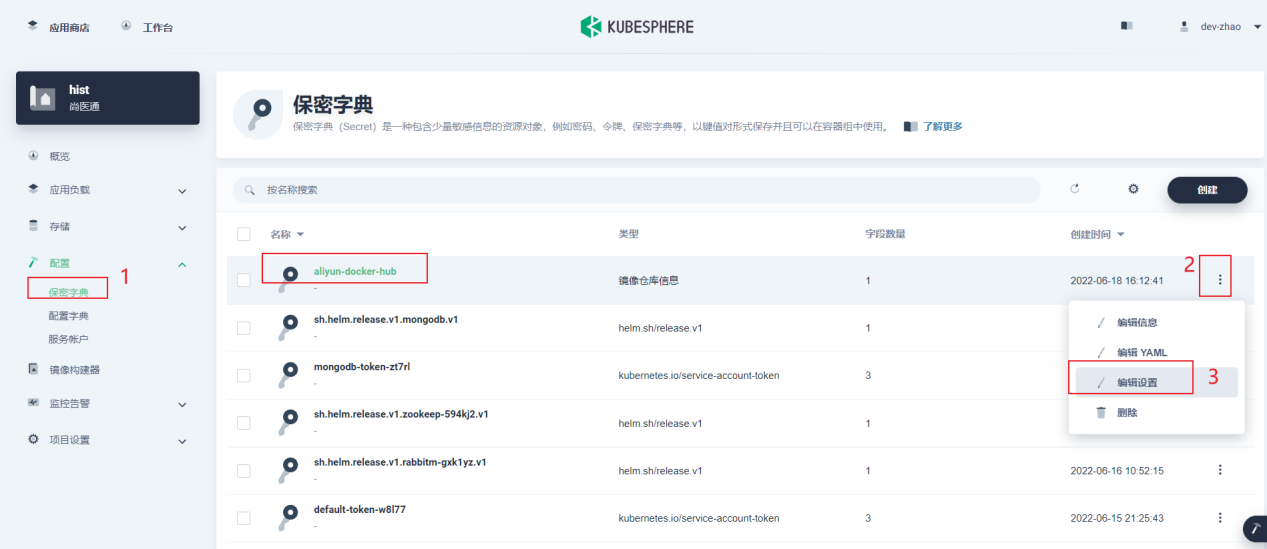
|  |
| --- |
| stage('service-user 部署到dev环境') {  agent none  steps {  container ('maven') {  withCredentials([  kubeconfigFile(  credentialsId: env.KUBECONFIG\_CREDENTIAL\_ID,  variable: 'KUBECONFIG')  ]) {  sh 'envsubst < service/service-user/deploy/deploy.yml | kubectl apply -f -'  }  }  }  } |

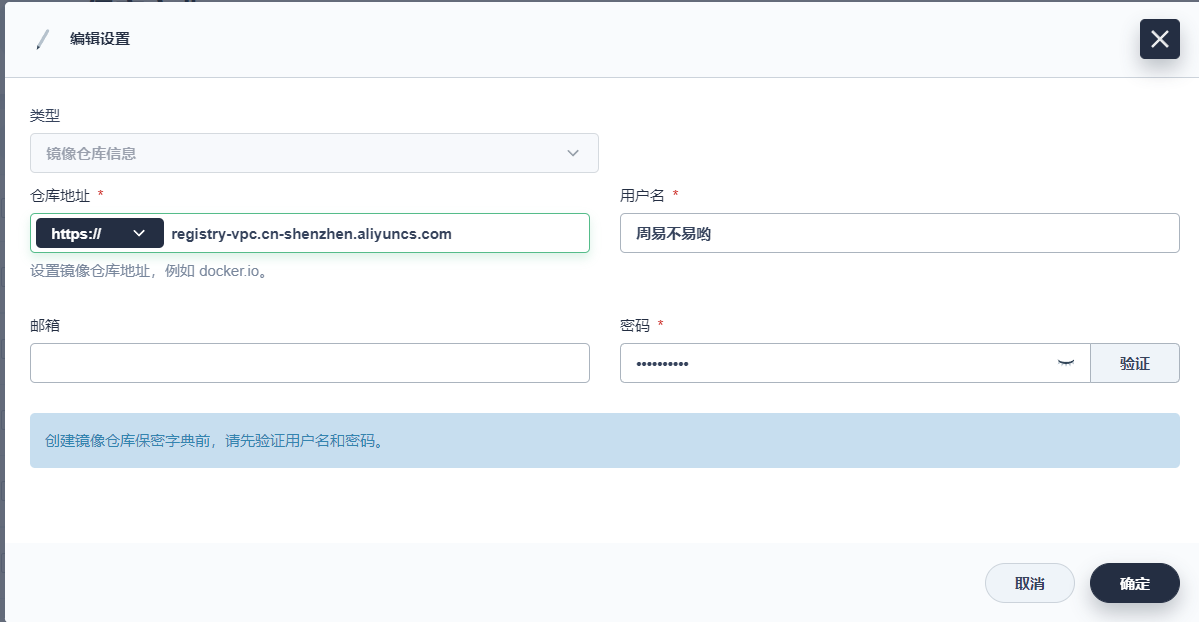
#### **5.2参考url 构建和部署 Maven 项目**

<https://kubesphere.io/zh/docs/devops-user-guide/examples/a-maven-project/>

**6 推送拉取镜像注意仓库url**

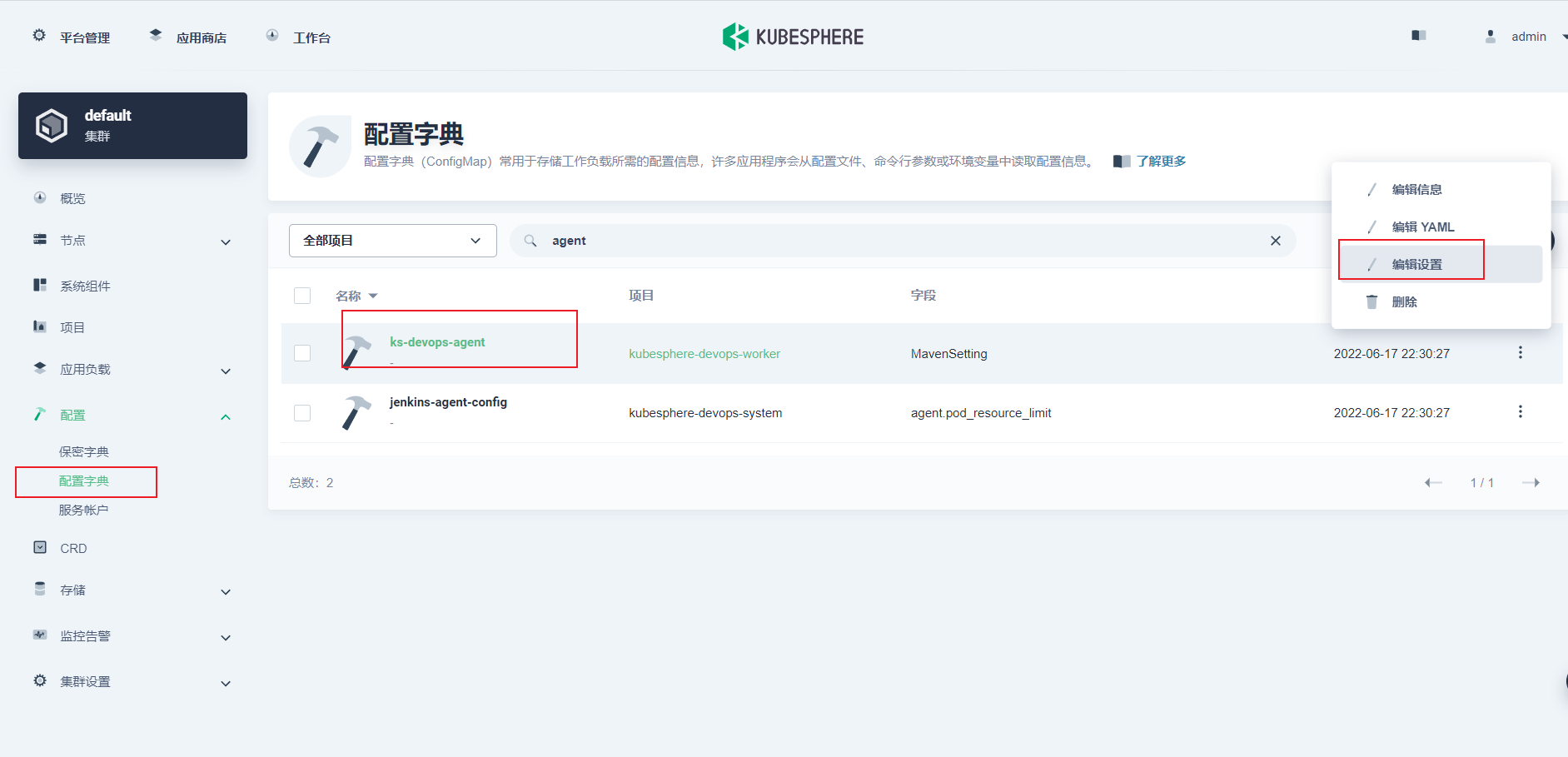


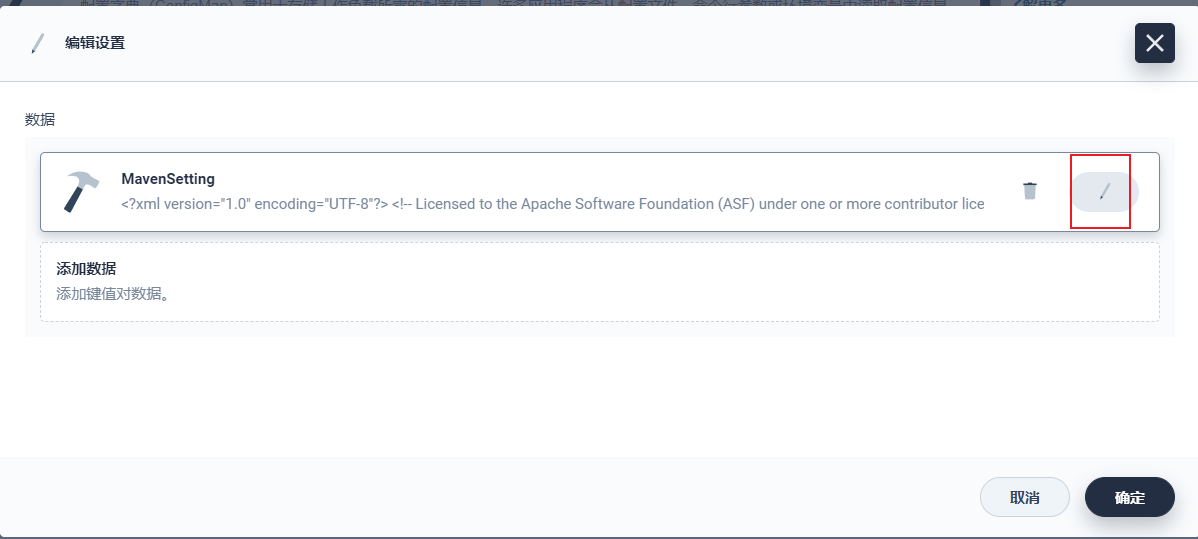






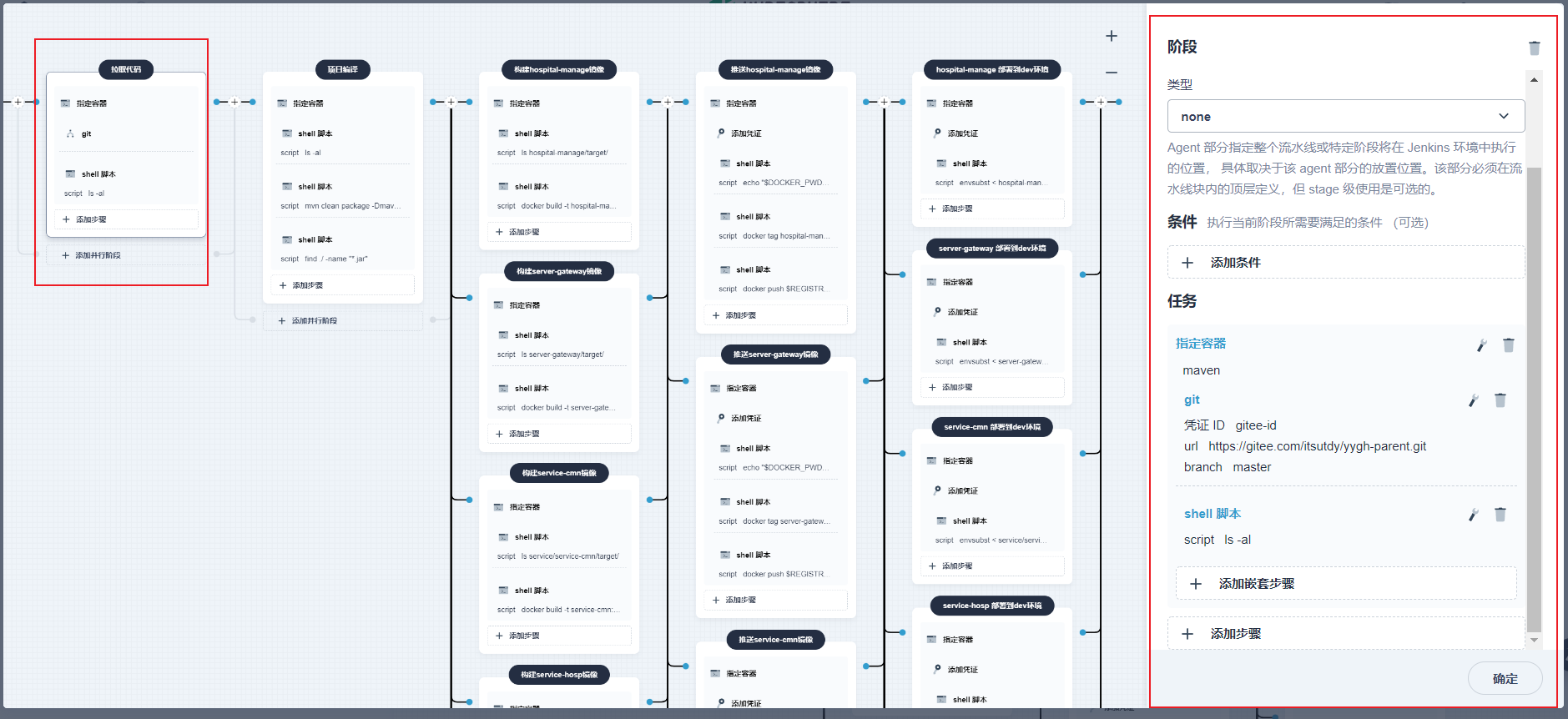
**7 Ks修改阿里mvn仓库**





**8 Devops模版**

**1拉取代码**



**2编译代码maven**

mvn clean package -Dmaven.test.skip=true



**3 构建镜像**

docker build -t hospital-manage:latest -f hospital-manage/Dockerfile hospital-manage/



|  |
| --- |
| stage('default-2') {  parallel {  stage('构建hospital-manage镜像') {  agent none  steps {  container('maven') {  sh 'ls hospital-manage/target'  sh 'docker build -t hospital-manage:latest -f hospital-manage/Dockerfile ./hospital-manage/'  }   }  }  stage('构建service-user镜像') {  agent none  steps {  container('maven') {  sh 'ls service/service-user/target'  sh 'docker build -t service-user:latest -f service/service-user/Dockerfile ./service/service-user/'  }   }  }   } } |

**4推送镜像**

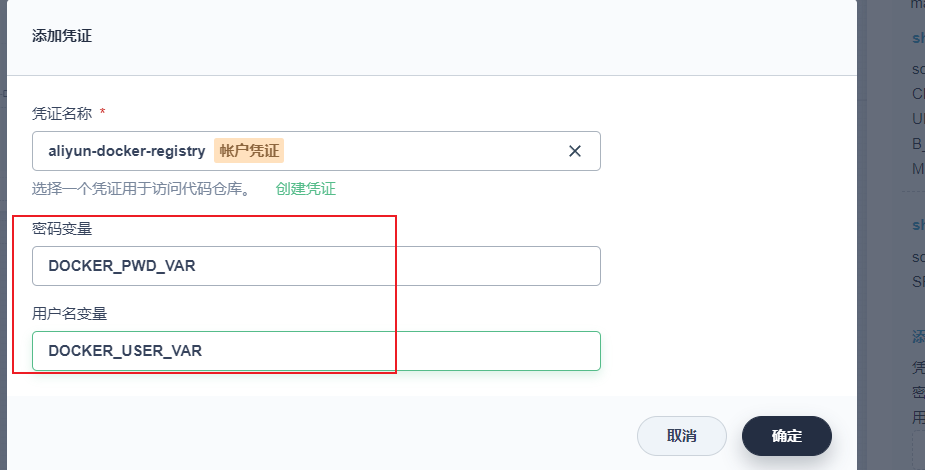
|  |
| --- |
| echo "$DOCKER\_PWD\_VAR" | docker login $REGISTRY -u "$DOCKER\_USER\_VAR" --password-stdin  docker tag hospital-manage:latest $REGISTRY/$DOCKERHUB\_NAMESPACE/hospital-manage:SNAPSHOT-$BUILD\_NUMBER  docker push $REGISTRY/$DOCKERHUB\_NAMESPACE/hospital-manage:SNAPSHOT-$BUILD\_NUMBER |







在添加凭证会跳出设置变量

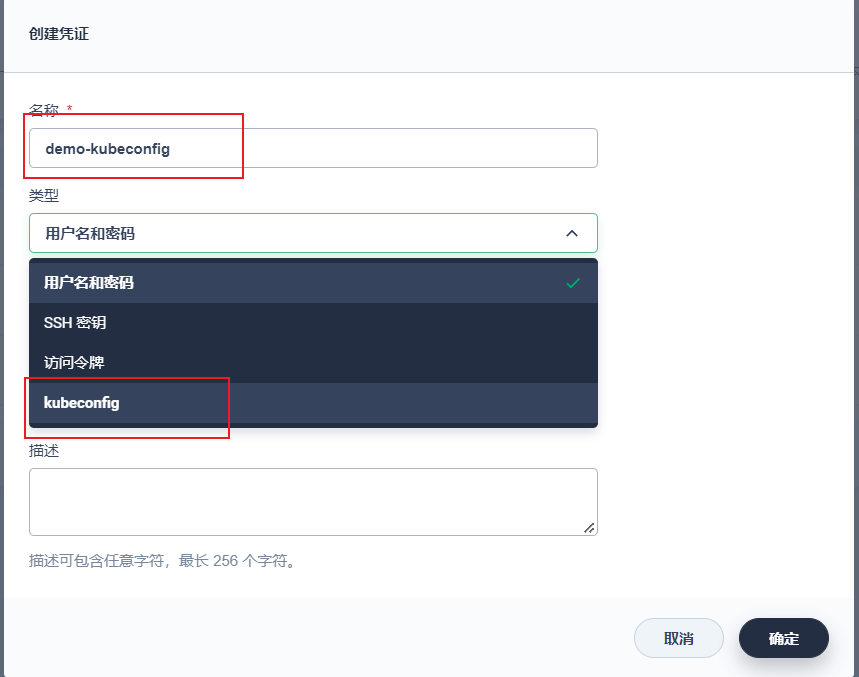


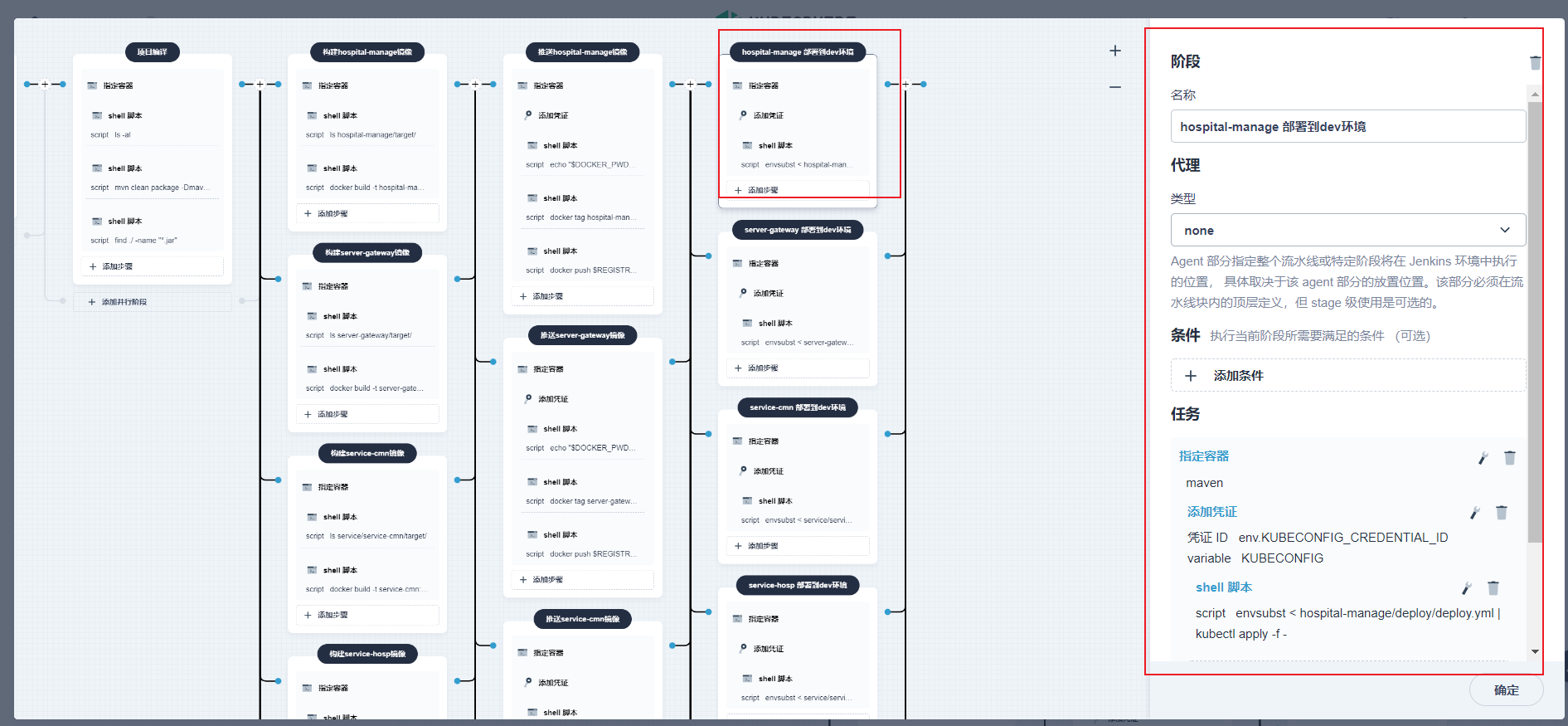
|  |
| --- |
| stage('default-3') {  parallel {  stage('推送hospital-manage镜像') {  agent none  steps {  container('maven') {  withCredentials([usernamePassword(credentialsId : 'aliyun-docker-registry' ,usernameVariable : 'DOCKER\_USER\_VAR' ,passwordVariable : 'DOCKER\_PWD\_VAR' ,)]) {  sh 'echo "$DOCKER\_PWD\_VAR" | docker login $REGISTRY -u "$DOCKER\_USER\_VAR" --password-stdin'  sh 'docker tag hospital-manage:latest $REGISTRY/$DOCKERHUB\_NAMESPACE/hospital-manage:SNAPSHOT-$BUILD\_NUMBER'  sh 'docker push $REGISTRY/$DOCKERHUB\_NAMESPACE/hospital-manage:SNAPSHOT-$BUILD\_NUMBER'  }   }   }  }  stage('推送service-user镜像') {  agent none  steps {  container('maven') {  withCredentials([usernamePassword(credentialsId : 'aliyun-docker-registry' ,usernameVariable : 'DOCKER\_USER\_VAR' ,passwordVariable : 'DOCKER\_PWD\_VAR' ,)]) {  sh 'echo "$DOCKER\_PWD\_VAR" | docker login $REGISTRY -u "$DOCKER\_USER\_VAR" --password-stdin'  sh 'docker tag service-user:latest $REGISTRY/$DOCKERHUB\_NAMESPACE/service-user:SNAPSHOT-$BUILD\_NUMBER'  sh 'docker push $REGISTRY/$DOCKERHUB\_NAMESPACE/service-user:SNAPSHOT-$BUILD\_NUMBER'  }   }   }  }   } } |

**5 部署deployment**

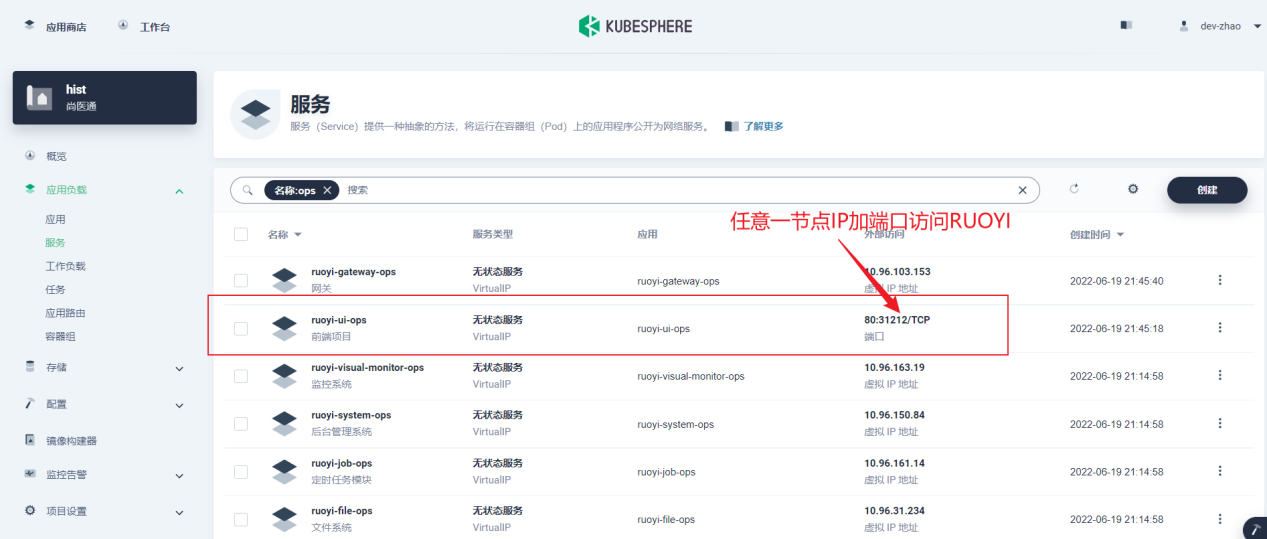
**envsubst < hospital-manage/deploy/deploy.yml | kubectl apply -f -**

**创建[demo-kubeconfig](http://47.112.174.183:30880/wuhan/clusters/default/devops/his-ruoyi-uih7zpn/credentials/demo-kubeconfig)凭证**

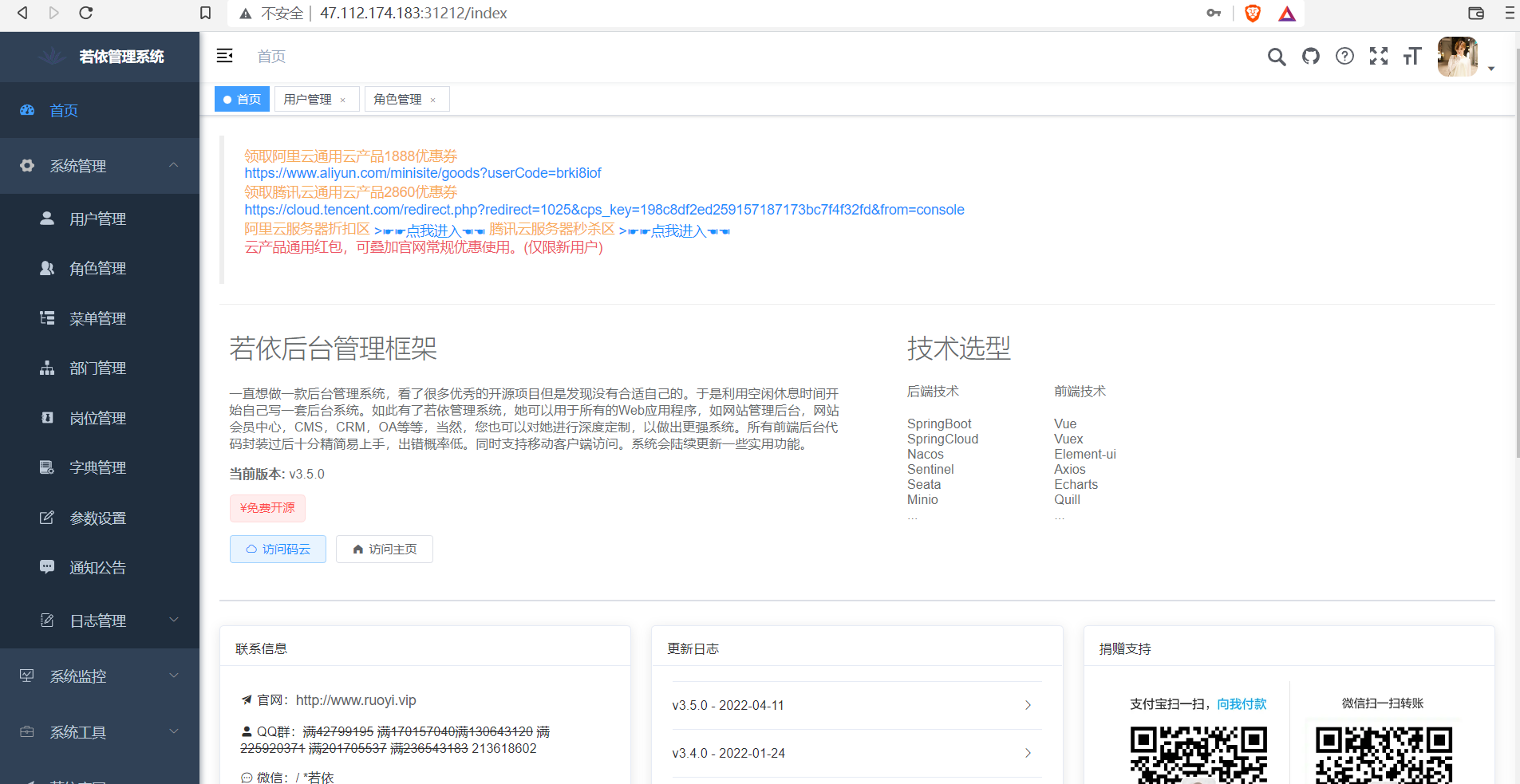




|  |
| --- |
| environment {  //定义的所有变量可以在流水线的任意位置 $变量名  DOCKER\_CREDENTIAL\_ID = 'dockerhub-id'  GITHUB\_CREDENTIAL\_ID = 'github-id'  KUBECONFIG\_CREDENTIAL\_ID = 'demo-kubeconfig'  REGISTRY = 'registry.cn-shenzhen.aliyuncs.com' //镜像要推送的仓库地址  //registry.cn-shenzhen.aliyuncs.com/mymetaverse/metrics-server:[镜像版本号]  DOCKERHUB\_NAMESPACE = 'syt2022' //推送空间  GITHUB\_ACCOUNT = 'kubesphere'  APP\_NAME = 'devops-java-sample'  ALIYUNHUB\_NAMESPACE = 'syt2022' //deploy.yaml 中引用  } |

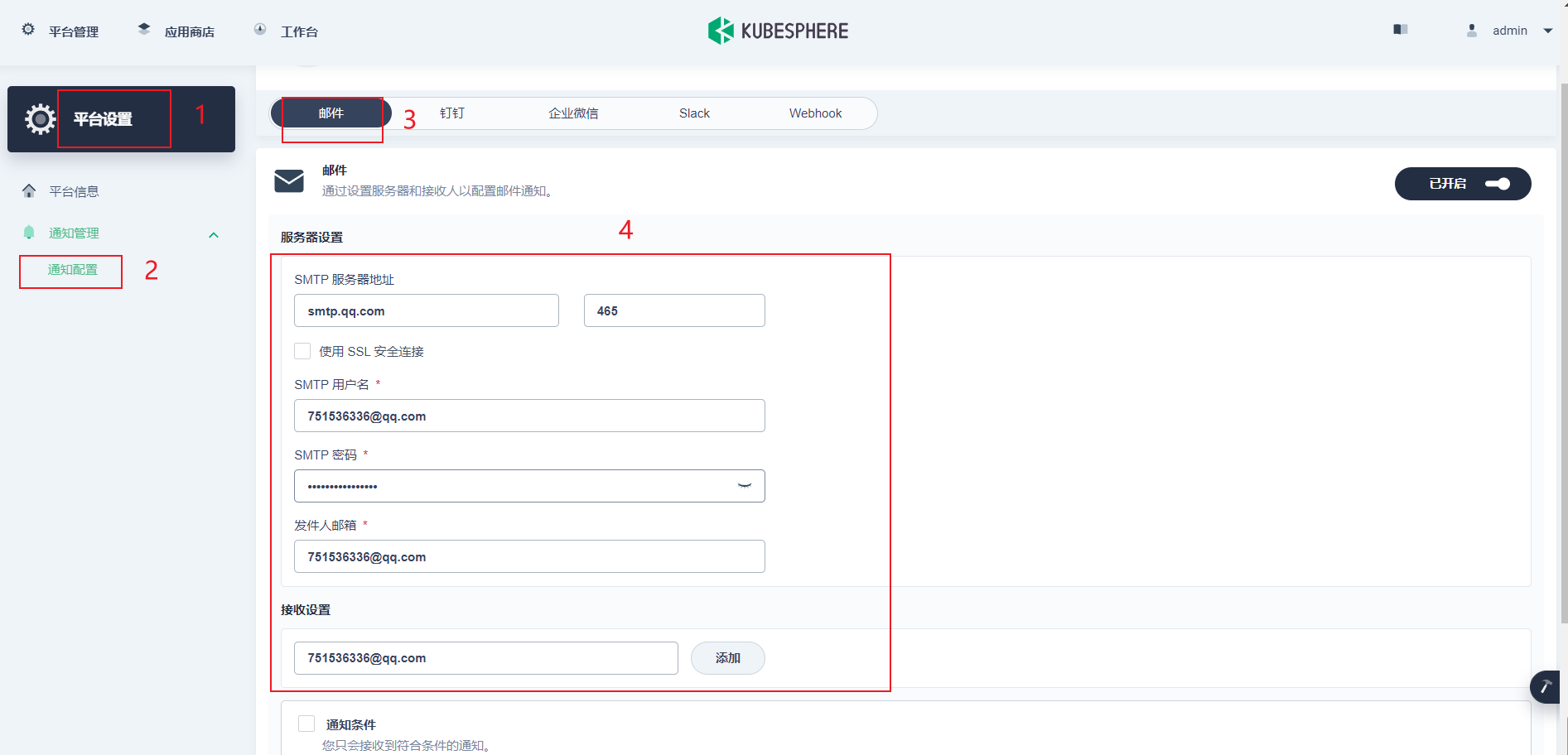


<http://47.112.174.183:31212/>



**邮件**

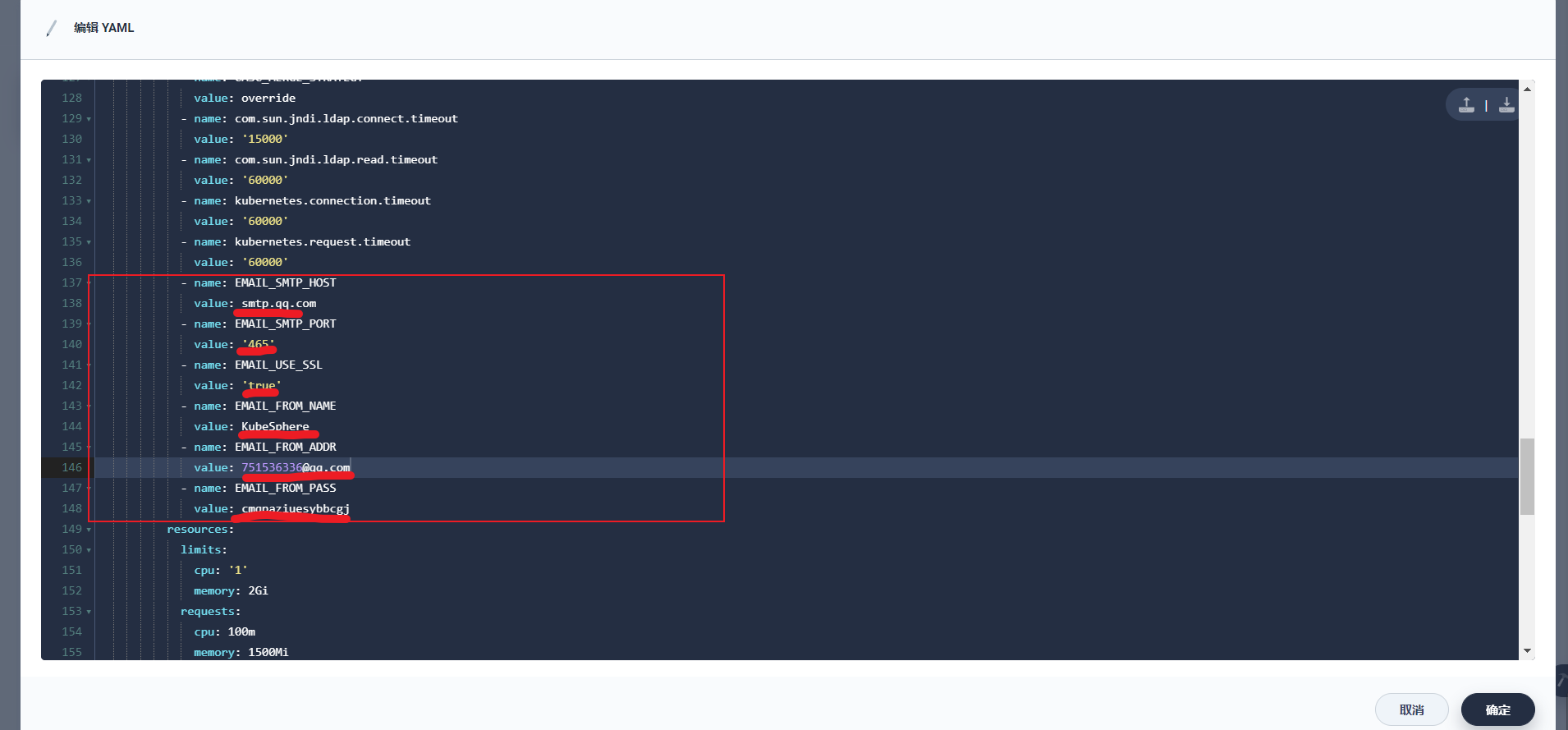
**1、配置全系统的邮件**

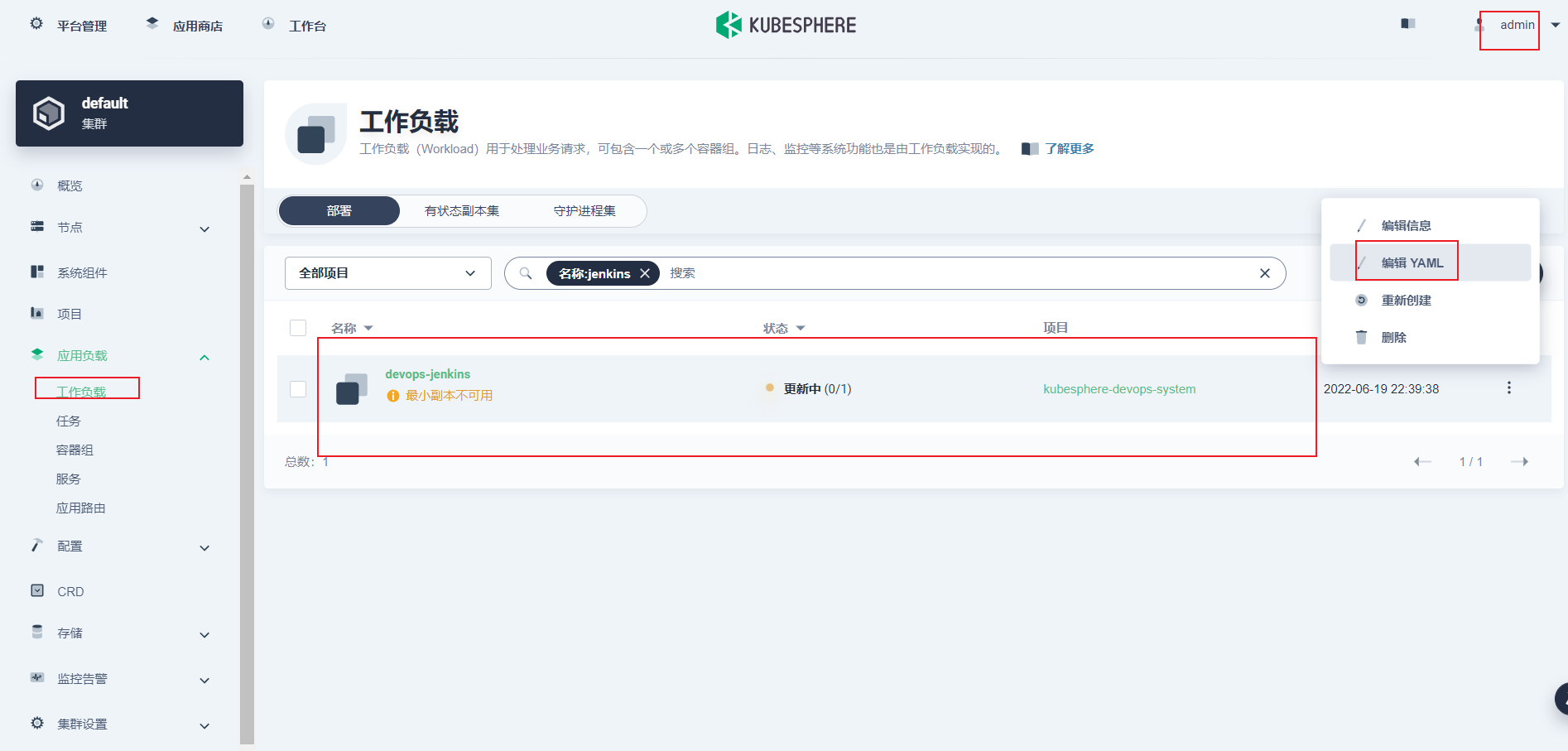


**2、修改ks-jenkins的配置里面的邮件：流水线（devops）发邮件**

**为 KubeSphere 流水线设置电子邮件服务器**

admin用户登陆

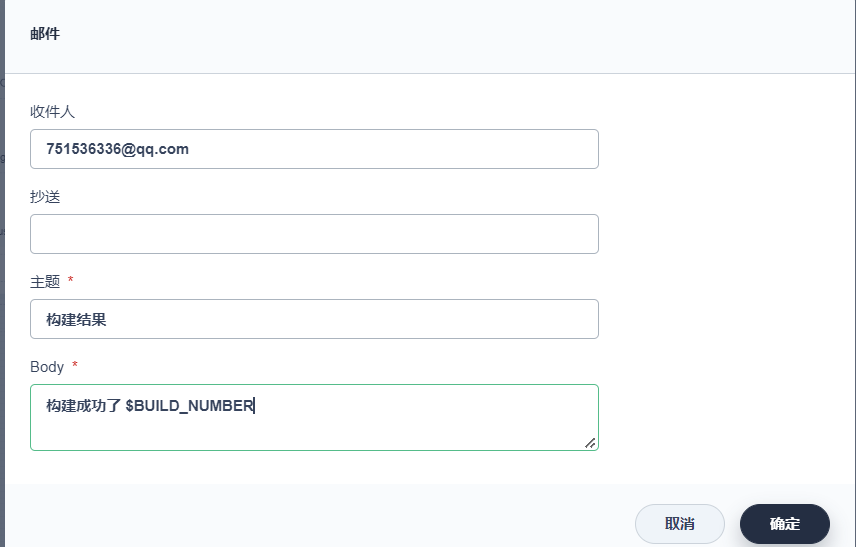




<https://kubesphere.io/zh/docs/devops-user-guide/how-to-use/jenkins-email/>

**3、测试**







**Webhook创建**

之前的流水线创建没有选择代码仓库没有生成webhook地址

要重新创建流水线项目并且选择代码仓库

