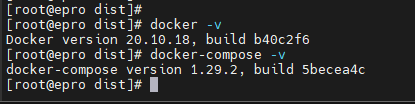
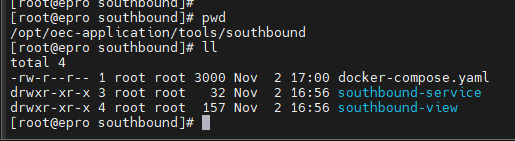
# **说明**

部署之前需要在主机上安装docker以及docker-compose，可使用docker -v以及docker-compose -v查看是否安装



# **下载项目文件**

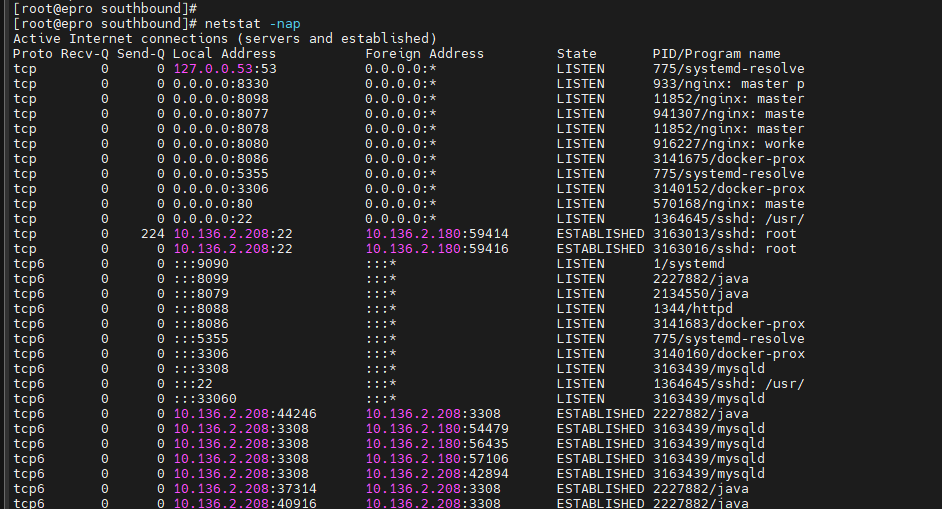
下载项目代码，执行命令git clone https://gitee.com/openeuler/oec-application.git，进入项目/tools/southbound目录下，目录结构如下所示



# **修改代码配置**

## 查看端口是否占用

当前设置前台端口9527、后台端口8888、数据库端口3309，执行netstat -nap（netstat -nap|grep 9527）命令查看端口是否被占用

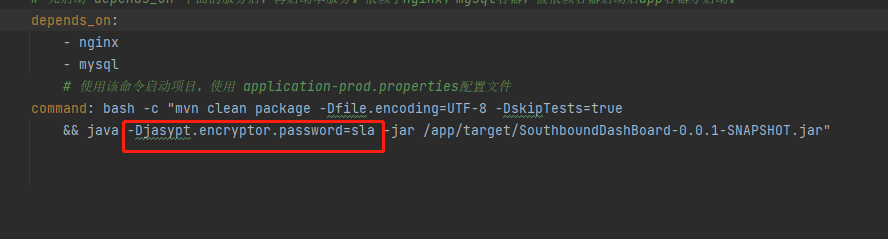


## **修改docker-compose配置**

docker-compose.yaml文件中是对应的前、后台、数据库镜像配置，如果端口被占用，做如下修改，如果不想使用mysql容器，可以删除mysql相关配置



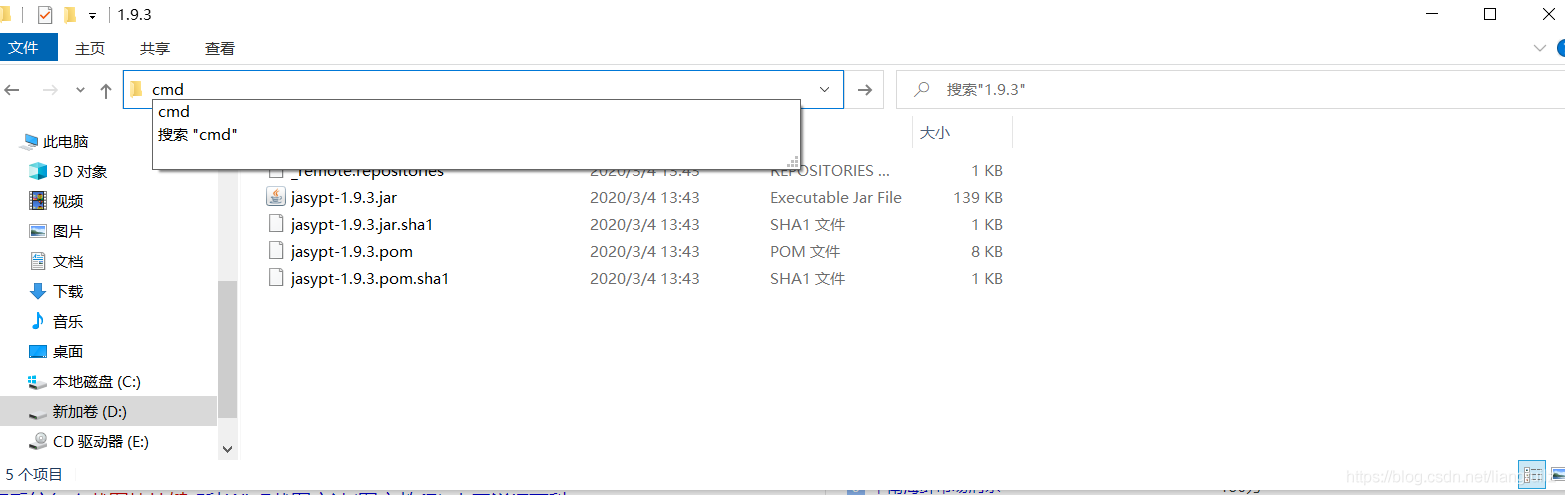
将salt改为数据库连接的加密密钥，如southbound-dashboard



## **生成加密数据库账号和密码**

### 打开找到jasypt-1.9.3.jar所在位置，如图打开输入cmd，然后Enter



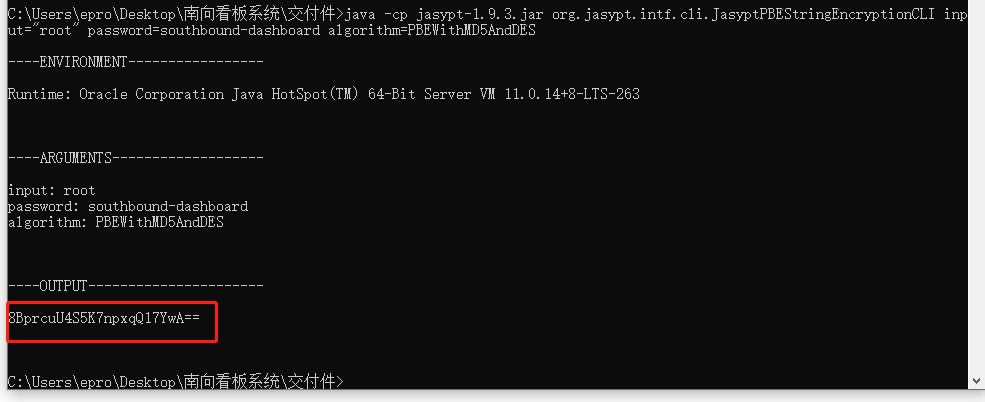


### 输入如下命令执行，生成数据库加密后的账号密码，加密密钥、加密字符串与docker-compose.yml配置保持一致

Java -cp jasypt-1.9.3.jar org.jasypt.intf.cli.JasyptPBEStringEncryptionCLI input=你要加密的字符串 password=加密秘钥 algorithm=PBEWithMD5AndDES

例如：

java -cp jasypt-1.9.3.jar org.jasypt.intf.cli.JasyptPBEStringEncryptionCLI input=root password= southbound-dashboard algorithm=PBEWithMD5AndDES

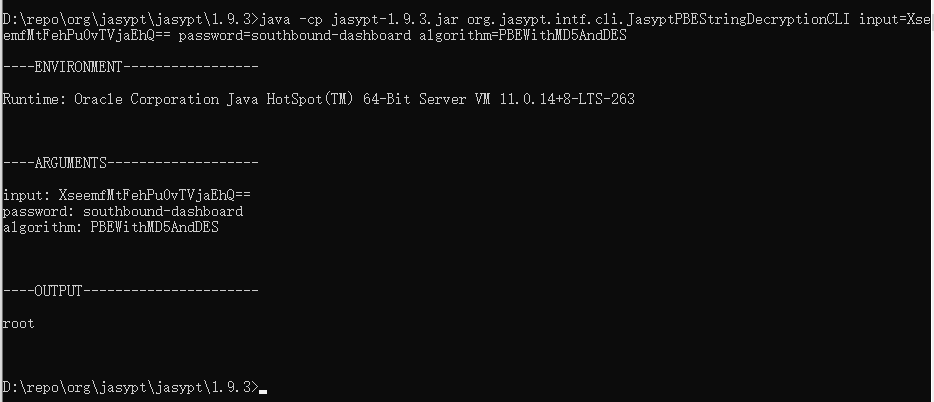


### 检查生成的加密文是否正确

执行命令：java -cp jasypt-1.9.3.jar org.jasypt.intf.cli.JasyptPBEStringDecryptionCLI input=加密后内容 password=加密秘钥 algorithm=PBEWithMD5AndDES

例如：

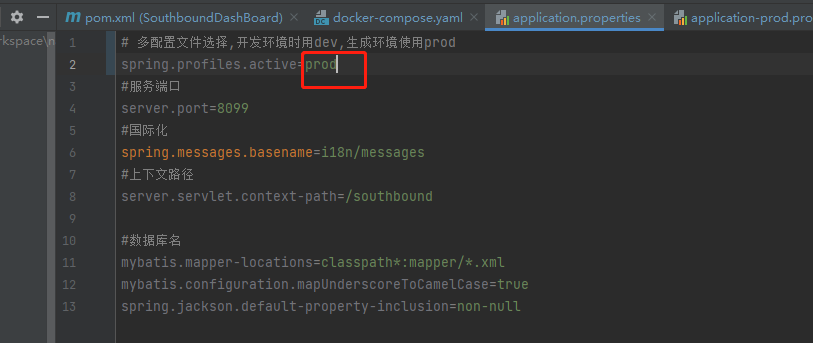
java -cp jasypt-1.9.3.jar org.jasypt.intf.cli.JasyptPBEStringDecryptionCLI input=XseemfMtFehPu0vTVjaEhQ== password=southbound-dashboard algorithm=PBEWithMD5AndDES



## **修改数据库连接配置**

### 将对应应用的数据库配置改为prod（默认prod不用修改）

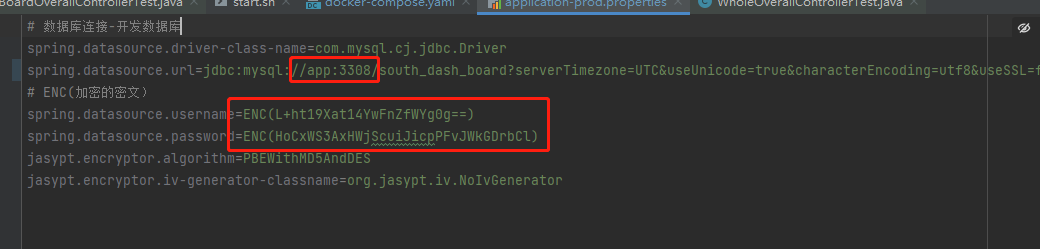
在shouthbound目录下执行vim southbound-service /src/main/resources /application.properties



### 修改数据库连接

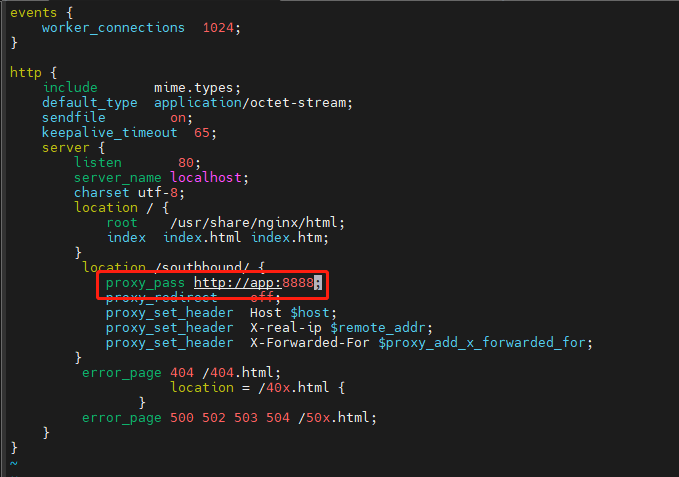
在shouthbound目录下执行vim southbound-service /src/main/resources /application-prod.properties

* 将mysql改为部署服务器IP
* 3309与docker-compose.yaml中mysql配置保持一致
* spring.datasource.username改为ENC(之前生成的账号)
* spring.datasource.password改为ENC(之前生成的密码)"



## **修改前台服务配置**

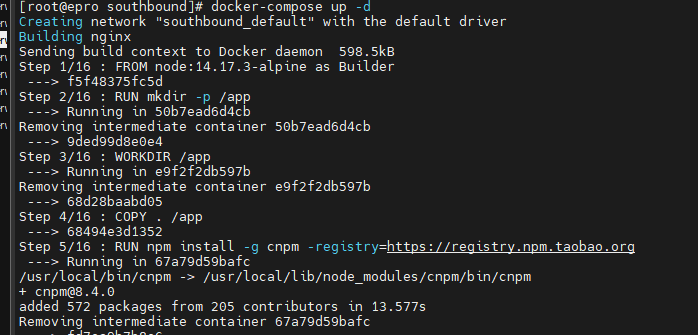
在shouthbound目录下执行vim southbound-view/ nginx.conf ，将app改为部署后台服务器IP，8888与docker-compose.yaml中后台配置保持一致



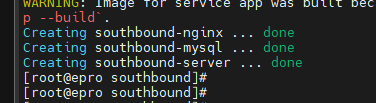
# **启动docker-compose**

## **启动docker-compose**

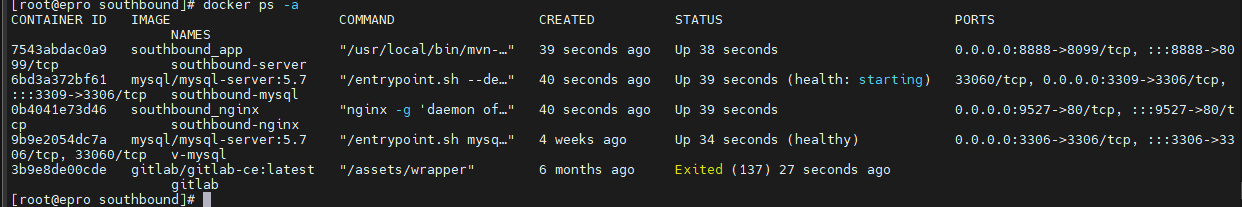
在southbound目录下执行docker-compose up -d，第一次运行时制作镜像会比较慢



出现下图所示，执行成功

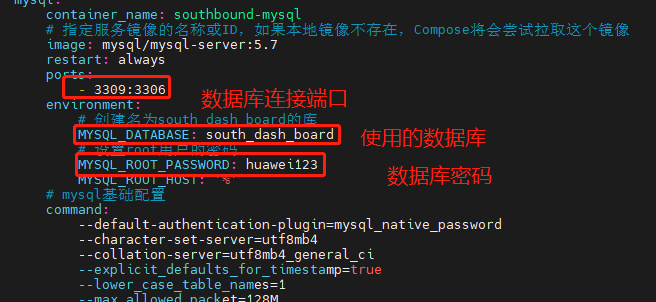


使用docker ps -a查看启动的容器，如下图所示



## **导入数据库脚本**

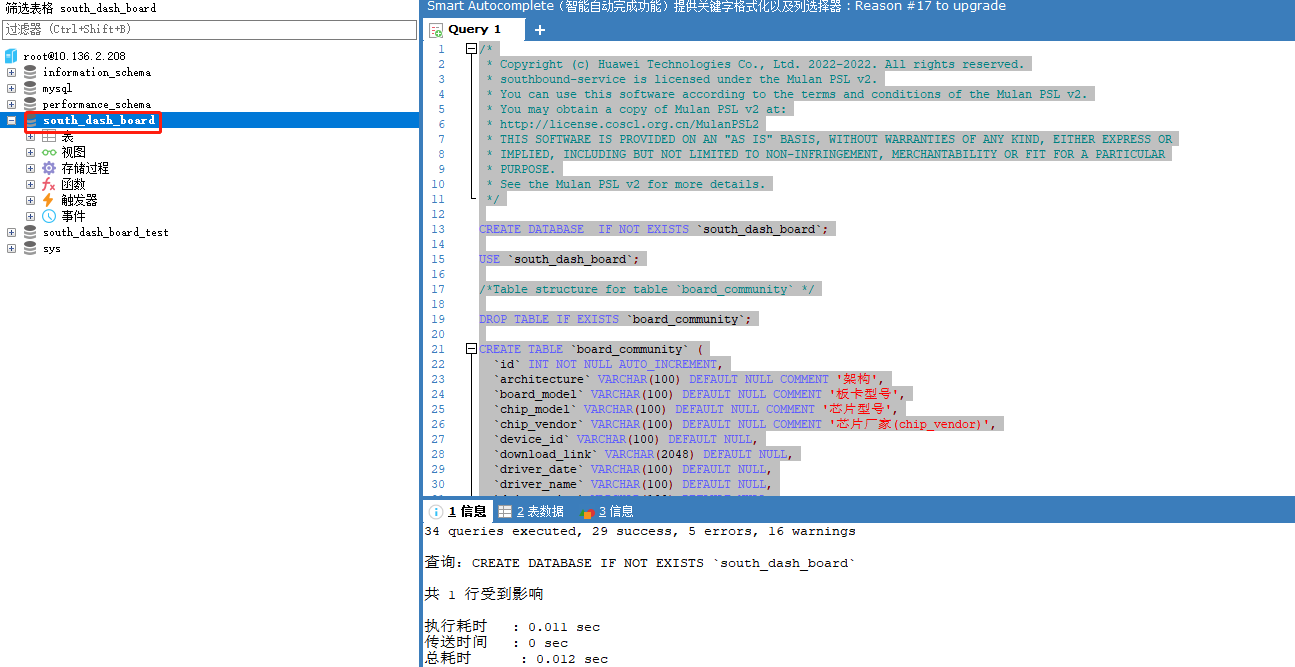
待mysql容器状态为Up，在数据库连接工具中连接数据库服务，配置为**docker-compose.yaml**中mysql的配置。



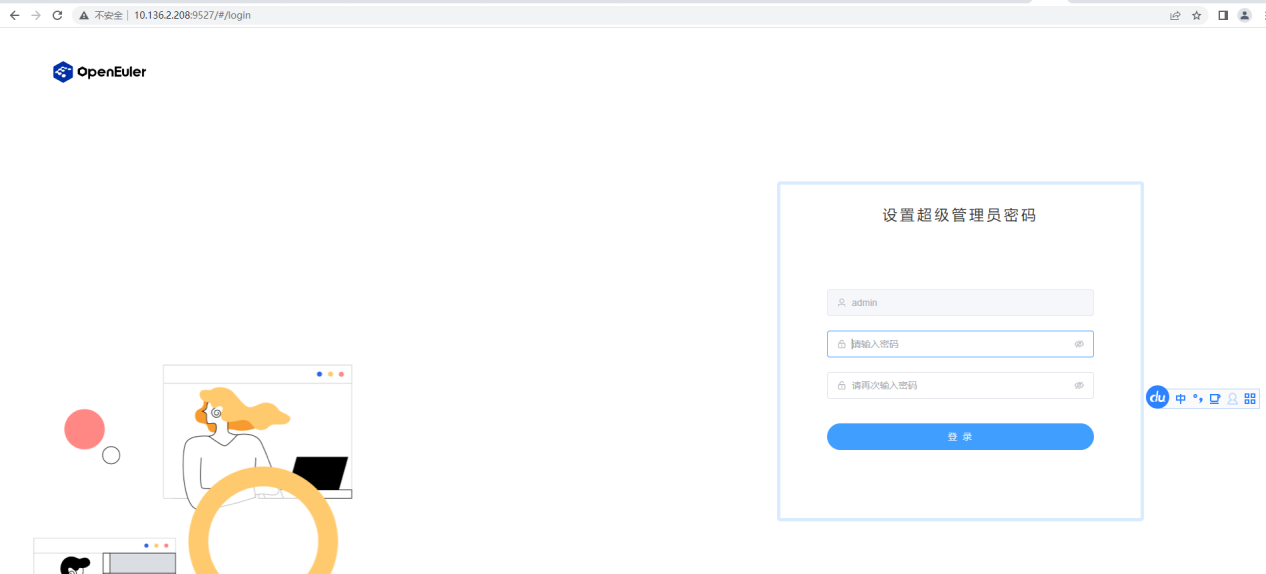
## **执行建库、建表语句**

执行[/southbound/southbound-service/src/main/resources/sql/](javascript:void(0);)

[south\_dash\_board.sql](javascript:void(0);)文件内容

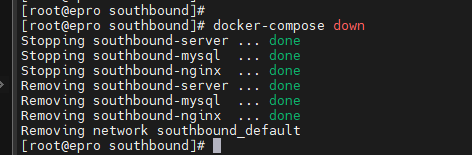


## **访问前台页面，设置管理员密码**



# **结束docker-compose**

在southbound目录下执行命令docker-compose down，docker-compose结束后，所有服务将被销毁



使用命令docker ps -a查看容器是否被删除