Cola-Admin 部署文档

序

项目地址

- 后端地址：https://gitee.com/xiaolifeizei/cola-admin
- 前端地址：https://gitee.com/xiaolifeizei/cola-ui

在线演示

- 演示地址：http://www.cola-admin.vip
- 默认用户：admin
- 默认密码：123123

部署说明

本部署文档是在Linux系统下使用docker部署的。本部署文档操作环境均在本地虚拟机上搭建。

部署环境

- CenterOS：CentOS Linux release 7.9.2009 (Core)
- IP地址：192.168.230.128
- docker：20.10.9-3.el7
- docker-compose：v2.6.0

安装部署依赖

安装docker

```
# 更新软件源
yum -y update
# 安装需要的软件包
yum install -y yum-utils device-mapper-persistent-data lvm2
# 设置yum源
yum-config-manager --add-repo http://mirrors.aliyun.com/docker-
ce/linux/centos/docker-ce.repo
# 查询版本号
yum list docker-ce --showduplicates | sort -r
# 安装指定版本
yum -y install docker-ce-20.10.9-3.el7
# 启动并设置开机自动启动
systemctl start docker & systemctl enable docker
```

安装docker-compose

创建docker文件夹

```
mkdir ~/docker
```

进入~/docker文件夹

```
cd ~/docker
# 下载并安装
curl -L "https://get.daocloud.io/docker/compose/releases/download/v2.6.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
# 添加执行权限
chmod +x /usr/local/bin/docker-compose
# 测试是否安装成功
docker-compose -v
```

推送镜像到远程仓库

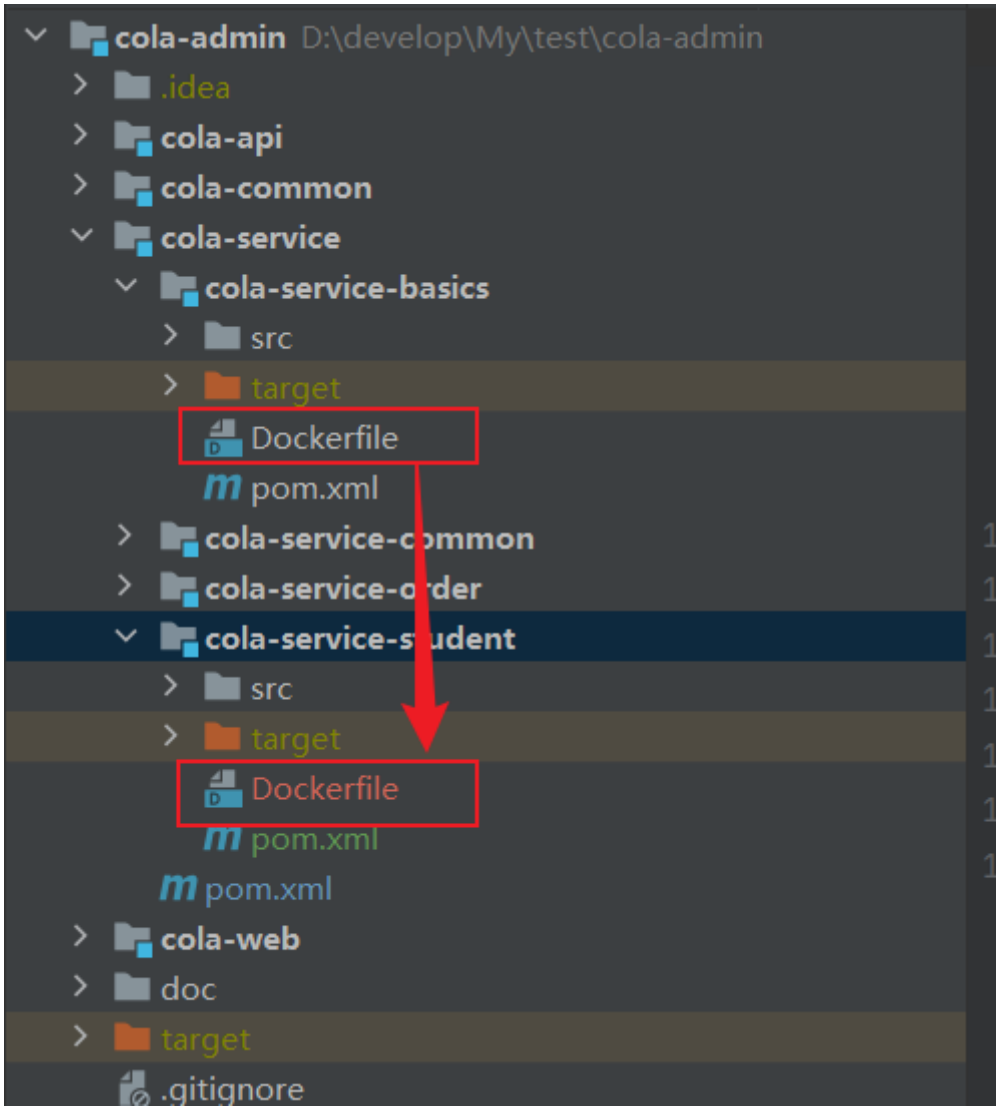
准备镜像仓库

这里以阿里云为例（其他云平台请参考其文档），登录阿里云控制台，搜索容器镜像服务

创建命名空间

编写Dockerfile

复制cola-service-basics下的Dockerfile到cola-service-student服务的根目录下



修改Dockerfile，主要修改工作目录及要导出的端口号

```
FROM openjdk:8u292-jdk-slim
MAINTAINER xiaolifeizei@163.com

RUN mkdir -p /cola-service-student
WORKDIR /cola-service-student

EXPOSE 8084
EXPOSE 20883

ADD ./target/*.jar ./app.jar
ENV PARAMS=""

ENTRYPOINT ["sh","-c","java -Djava.security.egd=file:/dev/./urandom $PARAMS -jar
app.jar"]
```

修改服务pom.xml文件

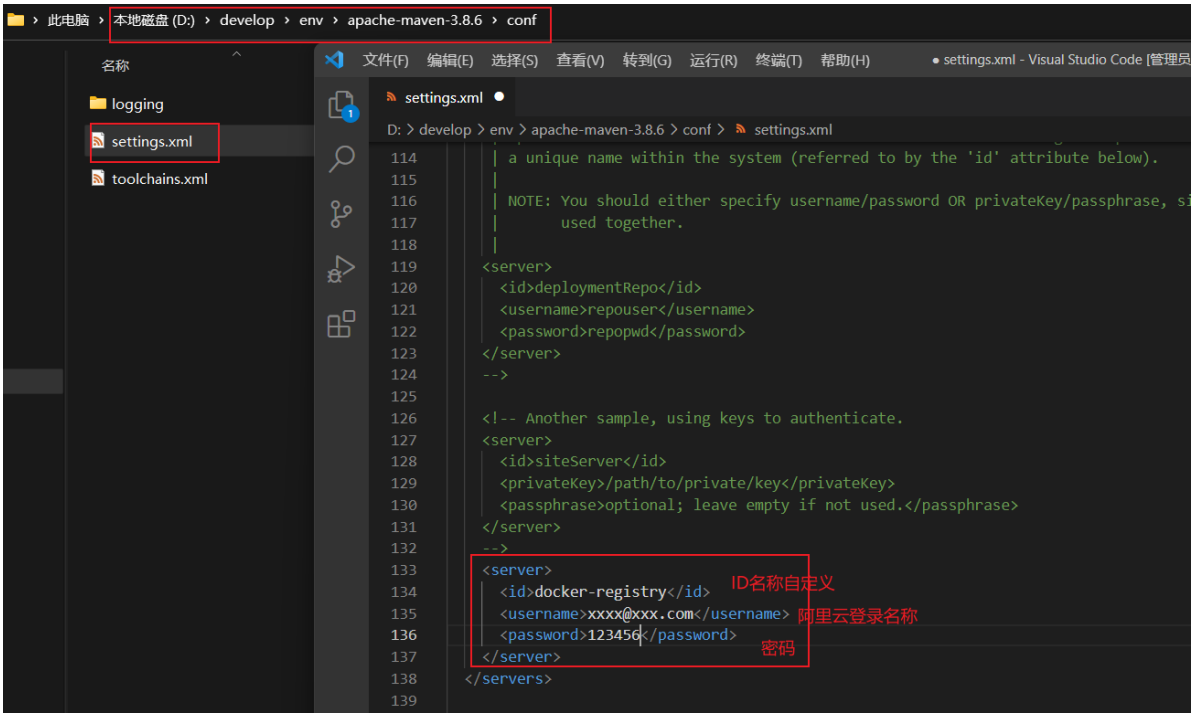
修改cola-service-student服务的pom.xml文件，添加 docker打包配置

```
    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
```

```
                <artifactId>spring-boot-maven-plugin</artifactId>
                <version>${springboot.version}</version>
                <executions>
                    <execution>
                        <phase>package</phase>
                        <goals>
                            <goal>repackage</goal>
                        </goals>
                    </execution>
                </executions>
            </plugin>
            <!--docker maven插件-->
            <plugin>
                <groupId>com.spotify</groupId>
                <artifactId>docker-maven-plugin</artifactId>
                <version>${docker.plugin.version}</version>
                <configuration>

 <imageName>${cola.registry.url}/${cola.registry.name}/${project.artifactId}:${p
roject.version}</imageName>
                    <dockerDirectory>${project.basedir}</dockerDirectory>
                    <resources>
                        <resource>
                            <targetPath>/</targetPath>
                            <directory>${project.build.directory}</directory>
                            <include>${project.build.finalName}.jar</include>
                        </resource>
                    </resources>
                    <registryUrl>${cola.registry.url}</registryUrl>
                    <serverId>${docker.server.id}</serverId>
                    <pushImage>true</pushImage>
                </configuration>
            </plugin>
        </plugins>
    </build>
```

修改maven配置

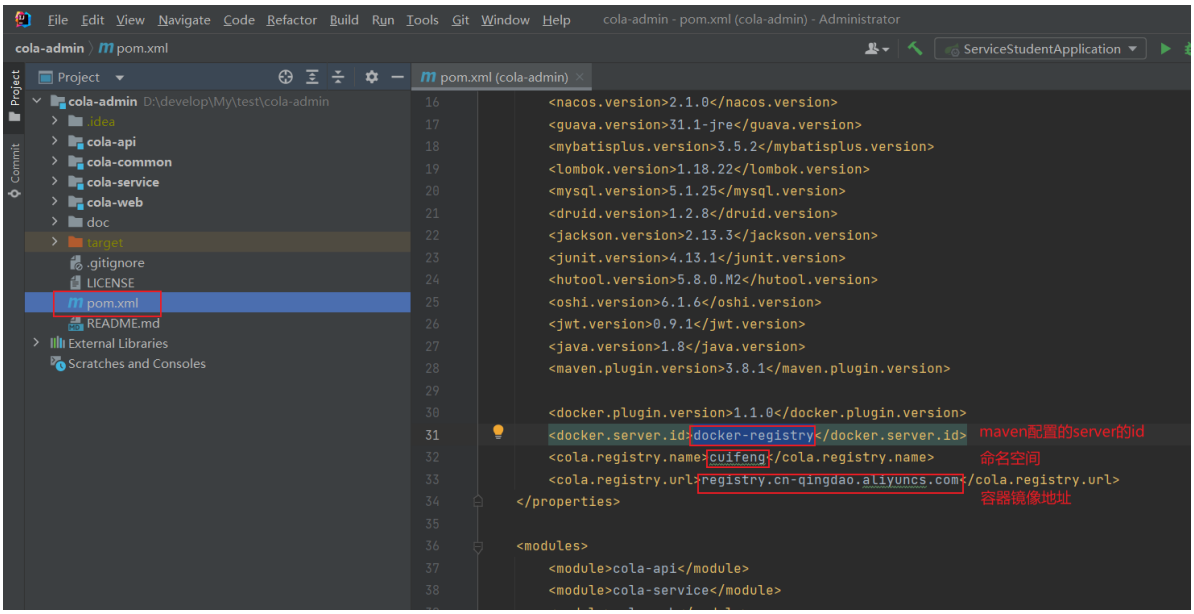
打开本机maven中的settings.xml添加配置项



修改主pom.xml文件

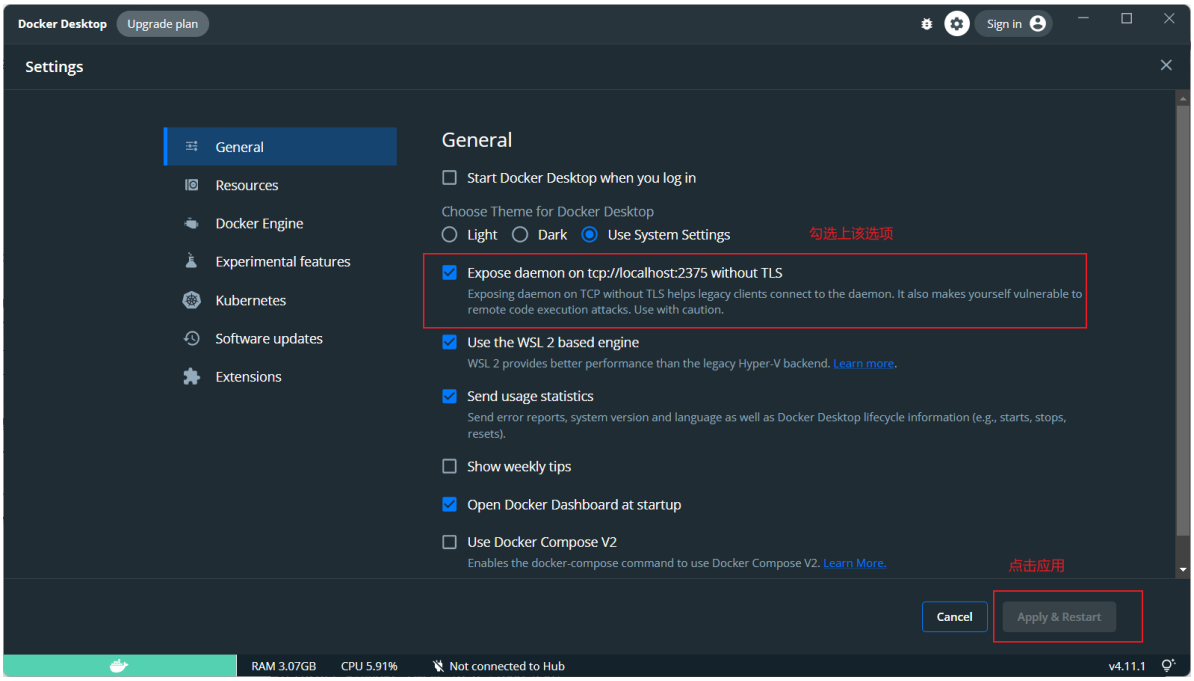
打开cola-admin下的pom.xml文件，修改如下配置项

```
<docker.server.id>docker-registry</docker.server.id>
<cola.registry.name>cuifeng</cola.registry.name>
<cola.registry.url>registry.cn-qingdao.aliyuncs.com</cola.registry.url>
```



本机安装docker desktop

安装完成后打开2375端口

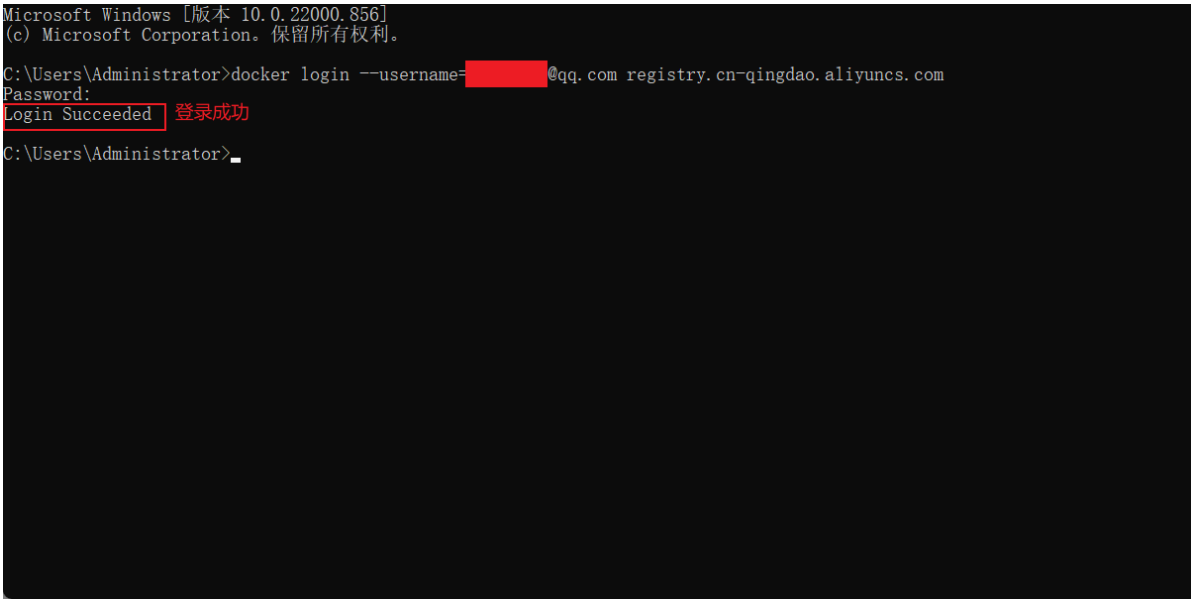


登录到镜像仓库

打开cmd，输入如下命令登录容器镜像服务

```
docker login --username=*******@qq.com registry.cn-qingdao.aliyuncs.com
```

执行后输入密码，显示Login Succeeded即为登录成功



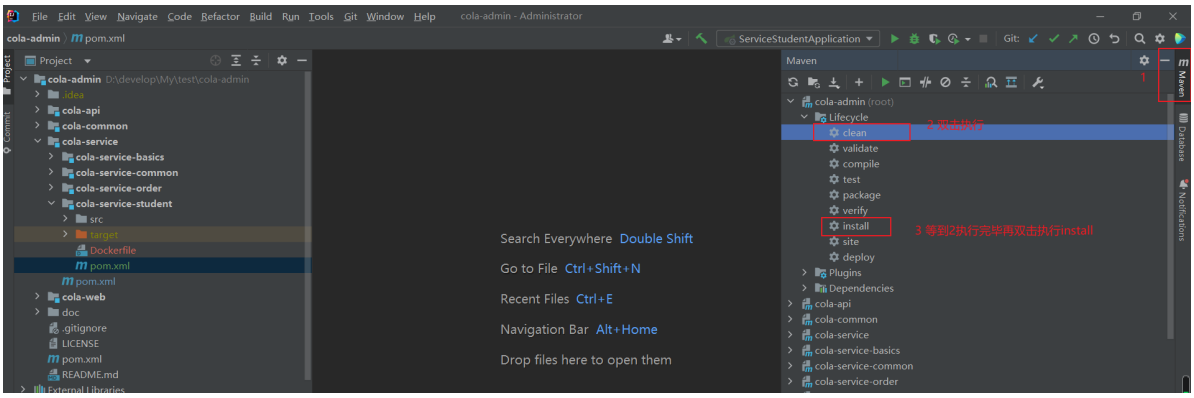
推送到镜像仓库

进入打开cmd进入cola-service-student服务的根目录，执行如下命令

```
mvn clean package docker:build -DpushImage -Dmaven.test.skip=true
```

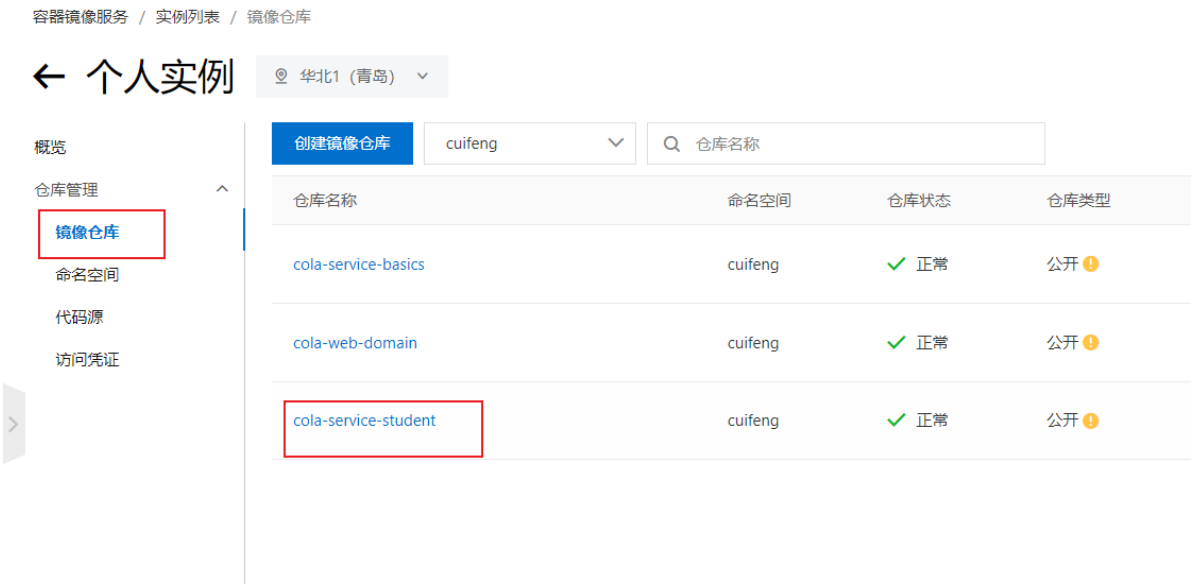
```
 ---> Running in c1ba65c05c64
Removing intermediate container c1ba65c05c64
 ---> d964b7896d8d
Step 7/9 : ADD ./target/*.jar ./app.jar

 ---> 76036e86dc7b
Step 8/9 : ENV PARAMS=""

 ---> Running in bd102cbe16b3
Removing intermediate container bd102cbe16b3
 ---> 25b3ed8802eb
Step 9/9 : ENTRYPOINT ["sh","-c","java -Djava.security.egd=file:/dev/./urandom $PARAMS -jar  app.jar"]

 ---> Running in 9e7da7c45d40
Removing intermediate container 9e7da7c45d40
 ---> b8a9f57734ee
ProgressMessage{id=null, status=null, stream=null, error=null, progress=null, progressDetail=null}
Successfully built b8a9f57734ee
Successfully tagged registry.cn-qingdao.aliyuncs.com/cuifeng/cola-service-student:2.1
[INFO] Built registry.cn-qingdao.aliyuncs.com/cuifeng/cola-service-student:2.1
[INFO] Pushing registry.cn-qingdao.aliyuncs.com/cuifeng/cola-service-student:2.1
The push refers to repository [registry.cn-qingdao.aliyuncs.com/cuifeng/cola-service-student]
e37ad3c37092: Pushed
140878ea7b86: Pushed
220d02abd73d: Pushed
eb4d0db4a187: Pushed
aa7d6f60014a: Pushed
764055ebc9a7: Pushed
2.1: digest: sha256:e93d3105da43d2aa60c98286a7e5a9a16ce7fcbbf2345565f754da42133538e1 size: 1579
null: null
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  03:36 min
[INFO] Finished at: 2022-08-15T10:40:53+08:00
```

如果执行过程有错误，请执行clean后再执行install，然后再执行上面的命令即可



查看仓库

进入容器镜像服务中的镜像仓库，即可查询到刚刚推送的镜像



用同样的方法将cola-basics、cola-domain服务发布到镜像仓库

单机编排

编写docker-compose.yml

```yaml
version: '3'
services:
  nginx:
    image: nginx
    container_name: nginx
    environment:
      TZ: Asia/Shanghai
    volumes:
      - /root/docker/nginx:/usr/share/nginx/html
    ports:
      - "80:80"
    restart: always
    networks:
      - bridge

  nacos:
    image: nacos/nacos-server:v2.1.0
    container_name: nacos
    links:
      - "mysql"
    ports:
      - "8848:8848"
      - "9848:9848"
      - "9849:9849"
    environment:
      MODE: standalone
      SPRING_DATASOURCE_PLATFORM: mysql
      MYSQL_SERVICE_HOST: mysql
      MYSQL_SERVICE_DB_NAME: nacos
      MYSQL_SERVICE_USER: root
      MYSQL_SERVICE_PASSWORD: root
    restart: always
    networks:
      - bridge

  redis:
    image: redis
    container_name: redis
    ports:
      - "6379:6379"
    restart: always
    networks:
      - bridge

  mysql:
    image: mysql:5.7
    container_name: mysql
    ports:
      - "3306:3306"
    environment:
      MYSQL_ROOT_PASSWORD: root  #数据库root密码
    restart: always
    networks:
      - bridge
```

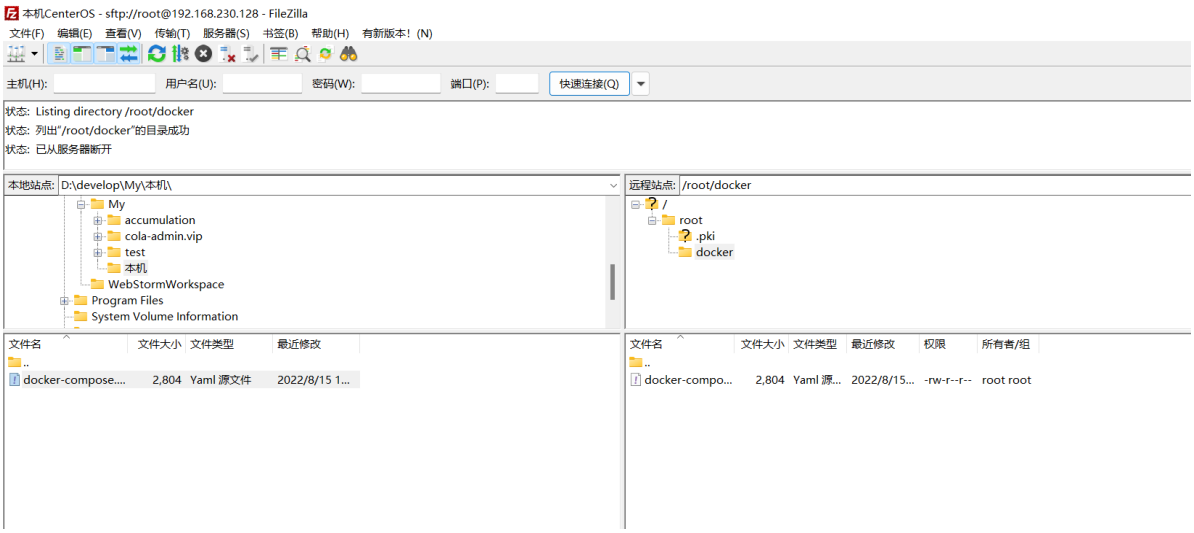
```
  cola-basice:
    image: aliyuncs.com/namespace/cola-service-basics:2.1
    container_name: cola-basice
    ports:
      - "8081:8081"
      - "20881:20881"
    links:
      - "nacos"
      - "redis"
      - "mysql"
    environment:
      TZ: Asia/Shanghai
      DUBBO_IP_TO_REGISTRY: cola-basice
      PARAMS: -Xmx512m -Xms128m -Ddubbo.registry.host=cola-basice -
Ddubbo.registry.address=nacos://nacos:8848 -Dspring.redis.host=redis -
Dspring.datasource.url=jdbc:mysql://mysql:3306/cola?
serverTimezone=UTC&autoReconnect=true&useUnicode=true&characterEncoding=UTF-8 -
Dspring.datasource.password=root
    restart: always
    networks:
      - bridge

  cola-student:
    image: aliyuncs.com/namespace/cola-service-student:2.1
    container_name: cola-student
    ports:
      - "8084:8084"
      - "20883:20883"
    links:
      - "nacos"
      - "redis"
      - "mysql"
      - "cola-basice"
    environment:
      TZ: Asia/Shanghai
      DUBBO_IP_TO_REGISTRY: cola-student
      PARAMS: -Xmx512m -Xms128m -Ddubbo.protocol.host=cola-student -
Ddubbo.registry.address=nacos://nacos:8848 -Dspring.redis.host=redis -
Dspring.datasource.url=jdbc:mysql://mysql:3306/cola?
serverTimezone=UTC&autoReconnect=true&useUnicode=true&characterEncoding=UTF-8 -
Dspring.datasource.password=root
    restart: always
    networks:
      - bridge

  cola-domain:
    image: aliyuncs.com/namespace/cola-web-domain:2.1
    container_name: cola-domain
    ports:
      - "8085:8085"
      - "20889:20889"
    links:
      - "nacos"
      - "redis"
      - "cola-basice"
      - "cola-student"
    environment:
      TZ: Asia/Shanghai
```

```
      DUBBO_IP_TO_REGISTRY: cola-domain
      PARAMS: -Xmx512m -Xms128m -Ddubbo.protocol.host=cola-domain -
Ddubbo.registry.address=nacos://nacos:8848 -Dspring.redis.host=redis
    restart: always
    networks:
      - bridge

networks:
  bridge:
    driver: bridge
```

修改cola-basics、cola-student、cola-domain中的image属性，改成自己的镜像地址

上传到服务器

将docker-compose.yml上传到服务器的~/docker目录下，没有就创建一个



启动服务

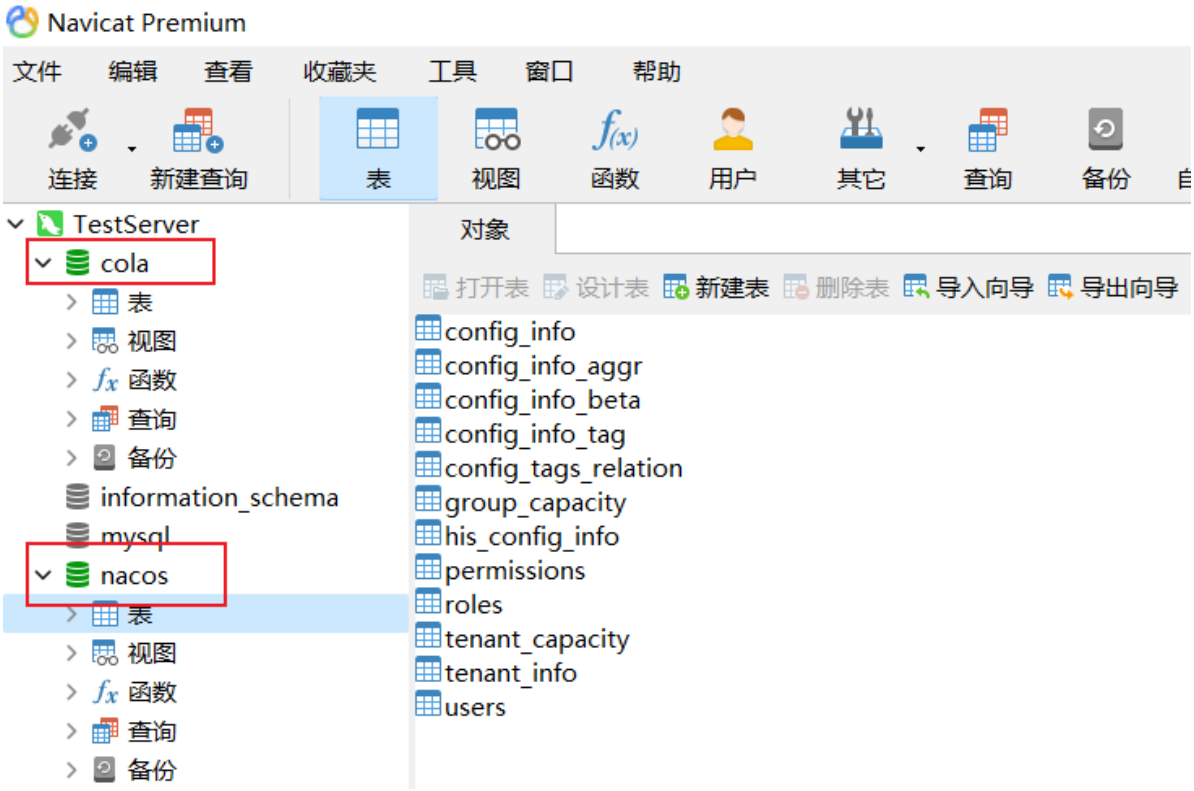
启动mysql

在服务器上执行如下命令

```
cd ~/docker
docker-compose up -d mysql
```

```
[root@localhost docker]# docker-compose up -d mysql
[+] Running 12/12
 # mysql Pulled
   # 72a69066d2fe Pull complete
   # 93619dbc5b36 Pull complete
   # 99da31dd6142 Pull complete
   # 626033c43d70 Pull complete
   # 37d5d7efb64e Pull complete
   # ac563158d721 Pull complete
   # d2ba16033dad Pull complete
   # 0ceb82207cd7 Pull complete
   # 37f2405cae96 Pull complete
   # e2482e017e53 Pull complete
   # 70deed891d42 Pull complete
[+] Running 2/2
 # Network docker_bridge  Created
 # Container mysql        Started
[root@localhost docker]#
```

导入cola.sql和nacos.sql



启动redis

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

```
[root@localhost docker]# docker-compose up -d redis
[+] Running 7/7
 # redis Pulled
   # a2abf6c4d29d Pull complete
   # c7a4e4382001 Pull complete
   # 4044b9ba67c9 Pull complete
   # c8388a79482f Pull complete
   # 413c8bb60be2 Pull complete
   # 1abfd3011519 Pull complete
[+] Running 1/1
 # Container redis  Started
[root@localhost docker]#
```

启动nacos

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

```
[root@localhost docker]# docker-compose up -d nacos
[+] Running 11/11
 # nacos Pulled
   # 2d473b07cdd5 Pull complete
   # 815fc4d5db6e Pull complete
   # 7128bc78cbbc Pull complete
   # 7d739257ae96 Pull complete
   # 9b8d32566260 Pull complete
   # a6223e2872bd Pull complete
   # 267d1d69b4e3 Pull complete
   # 98fbeaf82461 Pull complete
   # 8b43f8d72154 Pull complete
   # 4f4fb700ef54 Pull complete
[+] Running 2/2
 # Container mysql  Started
 # Container nacos  Started
[root@localhost docker]#
```

启动nginx

在~/docker目录下新建nginx文件夹

```
mkdir nginx
```

执行如下命令运行nginx

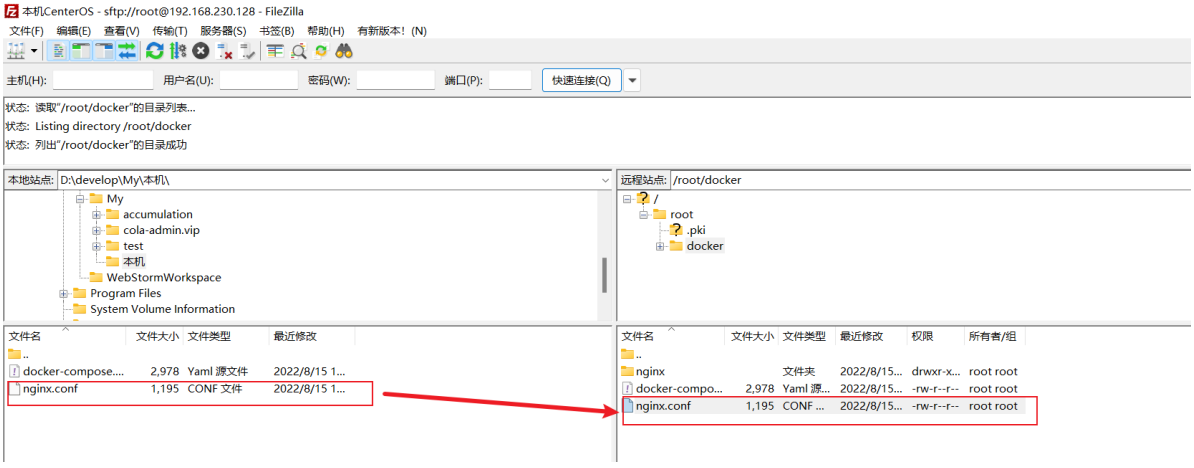
```
docker-compose up -d nginx
```

```
[root@localhost docker]# docker-compose up -d nginx
[+] Running 7/7
 # nginx Pulled
   # a2abf6c4d29d Already exists
   # a9edb18cadd1 Pull complete
   # 589b7251471a Pull complete
   # 186b1aaa4aa6 Pull complete
   # b4df32aa5a72 Pull complete
   # a0bcbecc962e Pull complete
[+] Running 1/1
 # Container nginx  Started
[root@localhost docker]#
```

复制以下内容（注意修改ip地址），保存为nginx.conf文件，并上传到~/docker目录中

```
user  nginx;
worker_processes  auto;

error_log  /var/log/nginx/error.log notice;
pid        /var/run/nginx.pid;


events {
    worker_connections  1024;
}


http {
    include       /etc/nginx/mime.types;
```

```
    default_type  application/octet-stream;

    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile        on;
    #tcp_nopush     on;

    keepalive_timeout  65;

    #gzip  on;

    #include /etc/nginx/conf.d/*.conf;
    server {
        listen       80;
        server_name  192.168.230.128; #
        location / {
            root   /usr/share/nginx/html;
            index  index.html index.htm;
        }
        location ^~ /api/ {
            proxy_redirect off;
            proxy_set_header Host $host:$server_port;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header x-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_pass  http://192.168.230.128:8085/;
        }
    }
}
```



复制~/docker目录下的nginx.conf到nginx容器中

```
docker cp ~/docker/nginx.conf nginx:/etc/nginx
```

```
[root@localhost docker]# docker cp ~/docker/nginx.conf nginx:/etc/nginx
[root@localhost docker]#
```

重启nginx

```
docker-compose restart nginx
```

```
[root@localhost docker]# docker-compose restart nginx
[+] Running 1/1
 ⠿ Container nginx  Started
[root@localhost docker]# 
```

启动其他服务

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

```
[root@localhost docker]# docker-compose up -d
[+] Running 13/13
 ⠿ cola-domain Pulled
   ⠿ 2e675ca2543a Pull complete
   ⠿ 2790c0b6511c Pull complete
 ⠿ cola-basice Pulled
   ⠿ 2ae00df93add Pull complete
   ⠿ 72be3388a82e Pull complete
 ⠿ cola-student Pulled
   ⠿ b4d181a07f80 Pull complete
   ⠿ 3ee45ae97306 Pull complete
   ⠿ 3346229aaa80 Pull complete
   ⠿ f9b8db9a69df Pull complete
   ⠿ 45ffece1083e Pull complete
   ⠿ 8a650d89a182 Pull complete
[+] Running 7/7
 ⠿ Container mysql         Running
 ⠿ Container nginx         Started
 ⠿ Container nacos         Started
 ⠿ Container redis         Started
 ⠿ Container cola-basice   Started
 ⠿ Container cola-student  Started
 ⠿ Container cola-domain   Started
```

查询服务运行情况

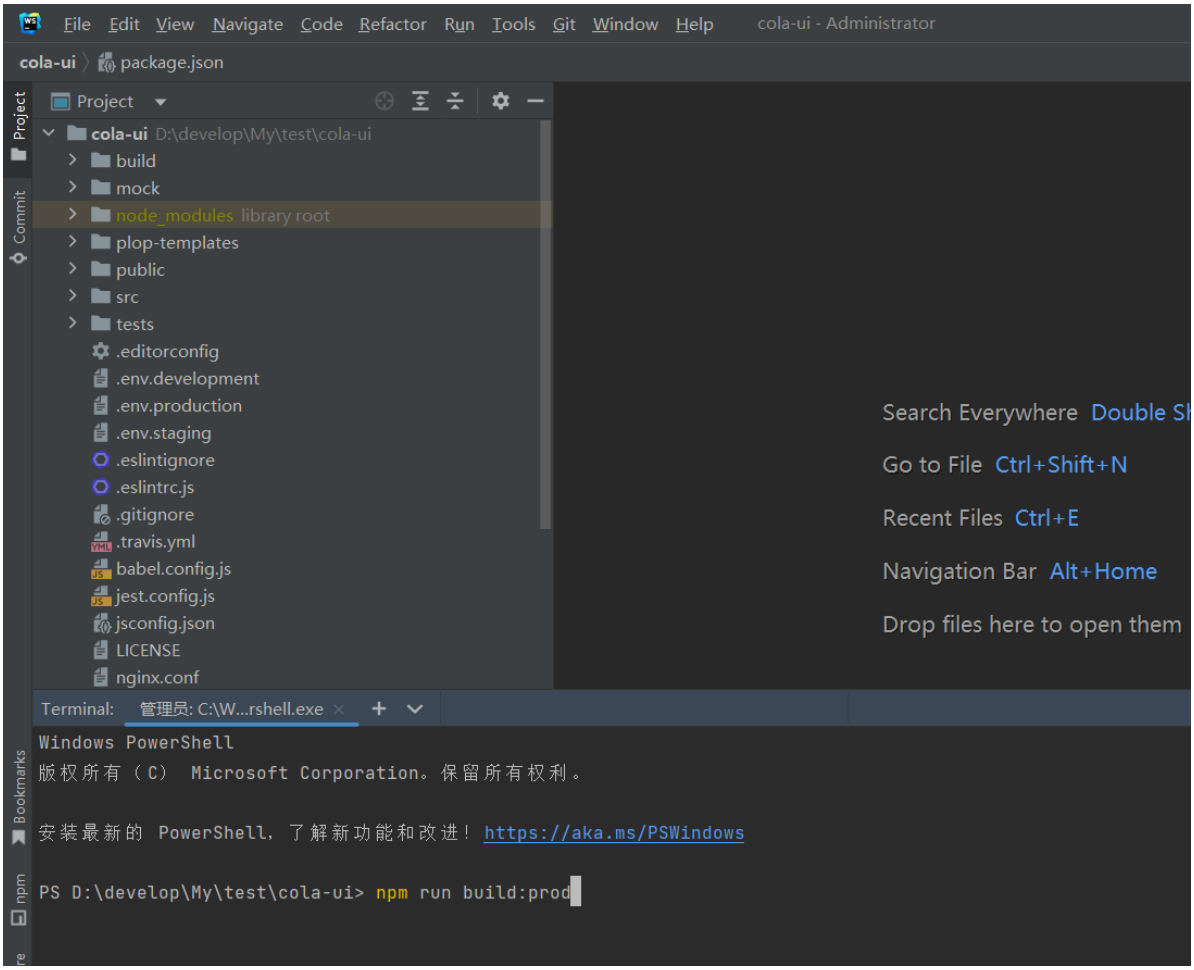
```
docker ps
```

```
[root@localhost docker]# docker ps
CONTAINER ID   IMAGE                                                         COMMAND                  CREATED          STATUS           PORTS
                                                             NAMES
09fa436363be   registry.cn-qingdao.aliyuncs.com/cuifeng/cola-web-domain:2.1        "sh -c 'java -Djava.…"   3 minutes ago    Up 2 minutes     0.0.0.0:8085->8085/tcp, :::8085->8085/t
.0.0:20889->20889/tcp, :::20889->20889/tcp                   cola-domain
4b348f46f5f1   registry.cn-qingdao.aliyuncs.com/cuifeng/cola-service-student:2.1   "sh -c 'java -Djava.…"   3 minutes ago    Up 22 seconds    0.0.0.0:8084->8084/tcp, :::8084->8084/t
.0.0:20883->20883/tcp, :::20883->20883/tcp                   cola-student
7a5e41b8b659   registry.cn-qingdao.aliyuncs.com/cuifeng/cola-service-basics:2.1    "sh -c 'java -Djava.…"   3 minutes ago    Up 3 minutes     0.0.0.0:8081->8081/tcp, :::8081->8081/t
.0.0:20881->20881/tcp, :::20881->20881/tcp                   cola-basice
e4c7496ccce6   nacos/nacos-server:v2.1.0                                     "bin/docker-startup.…"   3 minutes ago    Up 3 minutes     0.0.0.0:8848->8848/tcp, :::8848->8848/t
.0.0:9848-9849->9848-9849/tcp, :::9848-9849->9848-9849/tcp   nacos
a2356b0ffecc   redis                                                         "docker-entrypoint.s…"   3 minutes ago    Up 3 minutes     0.0.0.0:6379->6379/tcp, :::6379->6379/t
                                                             redis
ae36c14be03b   nginx                                                         "/docker-entrypoint.…"   3 minutes ago    Up 3 minutes     0.0.0.0:80->80/tcp, :::80->80/tcp
                                                             nginx
c6f14a94acc8   mysql:5.7                                                     "docker-entrypoint.s…"   40 minutes ago   Up 40 minutes    0.0.0.0:3306->3306/tcp, :::3306->3306/t
60/tcp                                                       mysql
[root@localhost docker]# 
```

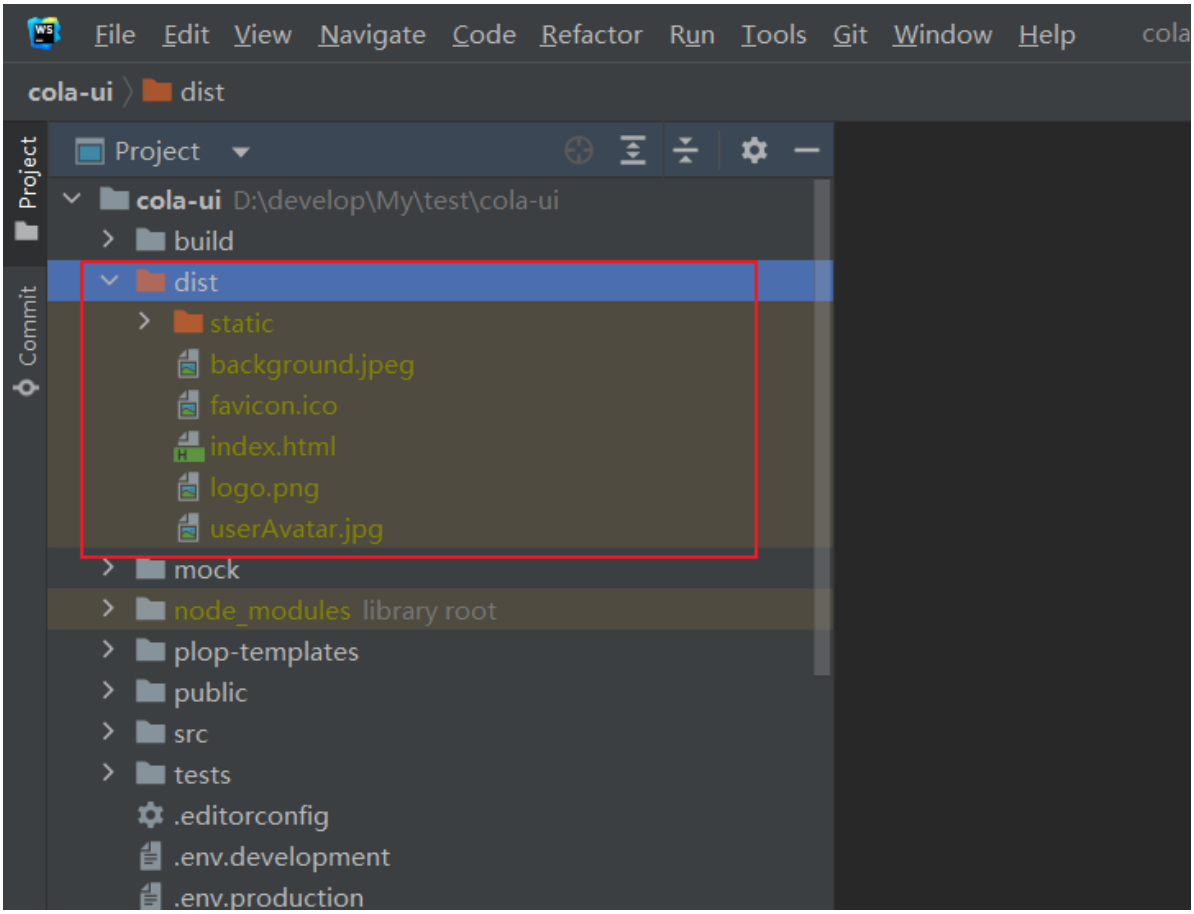
编译代码

打开WebStorm，运行如下命令

```
npm run build:prod
```

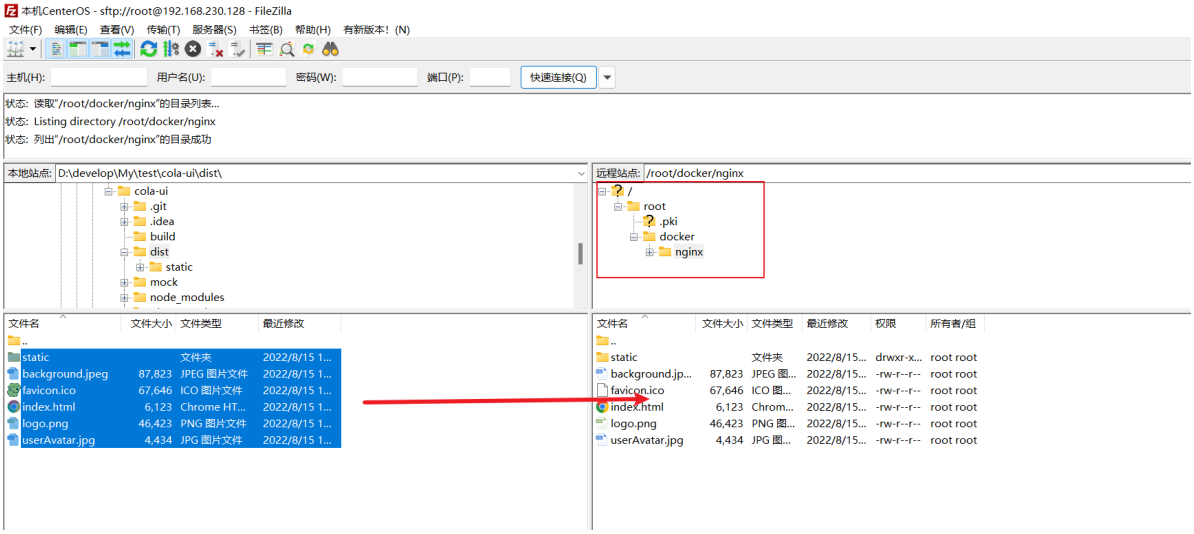


执行成功后会自动生成dist目录



上传前端代码

将cola-ui\dist目录下的所有文件上传到服务器的~/docker/nginx/目录下



访问