实验人员： 3022207128 杨宇鑫
实验目的： 搭建微服务环境,启动微服务并使用 IDEA 插件进行端云联调

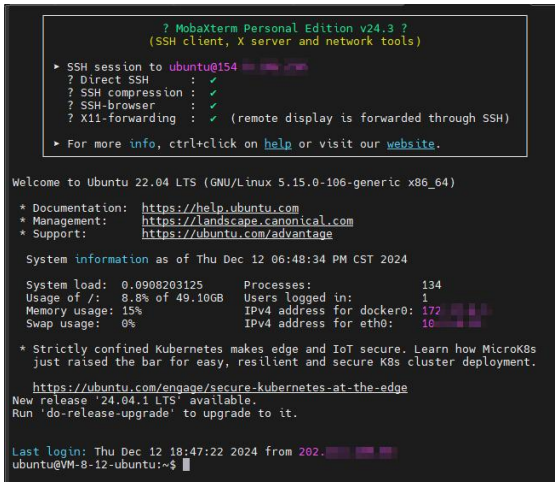
1. 前期准备工作

1.1 拉取课程代码

基于微服务电商网站设计和实现（购物车、商品、支付等）
https://github.com/aliyun/alibabacloud-microservice-demo

1.2 购买云服务器

1.3 通过 MobaXterm 连接到服务器



2. 搭建开源 MySQL - 数据存储

参考教程 Docker 部署 MySQL 8.3.0（保姆级图文教程） _docker 安装 mysql8.3-CSDN 博客

2.1 查找 Docker Hub 上的 MySQL 镜像并拉取镜像

```
ubuntu@VM-8-12-ubuntu:~$ docker search mysql
permission denied while trying to connect to the Docker daemon socket at unix:///var
/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.45/images/search?term=my
sql": dial unix /var/run/docker.sock: connect: permission denied
```

发生错误，在国内网络环境下，Docker 连接到 Docker Hub 可能会因网络延迟或连接超时而失败。在此处无需处理，只要能通过腾讯服务器默认的镜像源 pull 到最新的镜像就可以了

```
root@VM-8-12-ubuntu:/etc/docker# docker pull mysql
Using default tag: latest
latest: Pulling from library/mysql
2c0a233485c3: Pull complete
cb5a6a8519b2: Pull complete
570d30cf82c5: Pull complete
a841bff36f3c: Pull complete
80ba30c57782: Pull complete
5e49e1f26961: Pull complete
ced670fc7f1c: Pull complete
0b9dc7ad7f03: Pull complete
cd0d5df9937b: Pull complete
1f87d67b89c6: Pull complete
Digest: sha256:0255b469f0135a0236d672d60e3154ae2f4538b146744966d96440318cc822c6
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest
root@VM-8-12-ubuntu:/etc/docker# docker images mysql
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
mysql        latest    56a8c14e1404   8 weeks ago   603MB
root@VM-8-12-ubuntu:/etc/docker#
```

成功拉取到 mysql 最新的镜像

2.2 在宿主机创建目录

2.2.1 创建挂载目录

后面用于挂载 mysql 容器内目录，这里就放在 home 目录下

```
root@VM-8-12-ubuntu:/home/ubuntu# mkdir -p  /home/mysql/{conf,data,log}
```

2.2.2 创建配置文件

```
root@VM-8-12-ubuntu:/home/ubuntu# cd /home/mysql/conf
root@VM-8-12-ubuntu:/home/mysql/conf# vim my.cnf
```

```
[client]
#设置客户端默认字符集utf8mb4
default-character-set=utf8mb4
[mysql]
#设置服务器默认字符集为utf8mb4
default-character-set=utf8mb4
[mysqld]
#配置服务器的服务号，具备日后需要集群做准备
server-id = 1
#开启MySQL数据库的二进制日志，用于记录用户对数据库的操作SQL语句，具备日后需要集群做>
准备
log-bin=mysql-bin
#设置清理超过30天的日志，以免日志堆积造过多成服务器内存爆满。2592000秒等于30天的秒数
binlog_expire_logs_seconds = 2592000
#解决MySQL8.0版本GROUP BY问题
sql_mode='STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZER
O,NO_ENGINE_SUBSTITUTION'
#允许最大的连接数
max_connections=1000
# 禁用符号链接以防止各种安全风险
symbolic-links=0
# 设置东八区时区
default-time_zone = '+8:00'
```

2.3 启动 mysql 容器

`-p` 表示端口映射

`--restart=always` 表示容器退出时总是重启

`--name` 表示容器命名

`--privileged=true` 表示赋予容器权限修改宿主文件权利

`-v /home/mysql/log:/var/log/mysql` 表示容器日志挂载到宿主机

`-v /home/mysql/data:/var/lib/mysql` 表示容器存储文件挂载到宿主机

`-v /home/mysql/conf/my.cnf:/etc/mysql/my.cnf` 表示容器配置文件挂载到宿主机

`-e MYSQL_ROOT_PASSWORD=a12bCd3_W45pUq6` 表示设置 `mysql` 的 `root` 用户密码,建议用 `强密码`

`-d` 表示后台运行

docker 运行命令:

```
docker run \
-p 3306:3306 \
--restart=always \
--name mysql \
--privileged=true \
-v /home/mysql/log:/var/log/mysql \
-v /home/mysql/data:/var/lib/mysql \
-v /home/mysql/conf/my.cnf:/etc/mysql/my.cnf \
-e MYSQL_ROOT_PASSWORD=1q2w3e4razsxdcfv \
-d mysql
```

```
root@VM-8-12-ubuntu:/home/mysql/conf# docker run \
-p 3306:3306 \
--restart=always \
--name mysql \
--privileged=true \
-v /home/mysql/log:/var/log/mysql \
-v /home/mysql/data:/var/lib/mysql \
-v /home/mysql/conf/my.cnf:/etc/mysql/my.cnf \
-e MYSQL_ROOT_PASSWORD=1q2w3e4razsxdcfv \
-d mysql
5b63490c449ea3eb0574636ac50a90d098697a21b17b2f97c737befb7c9edbea
```

查看 docker 中运行中的容器

```
root@VM-8-12-ubuntu:/home/mysql/conf# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS         PORTS                                                  NAMES
5b63490c449e   mysql     "docker-entrypoint.s…"   24 seconds ago   Up 2 seconds   0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp   mysql
root@VM-8-12-ubuntu:/home/mysql/conf#
```

看到 MySQL 容器成功运行

3. 搭建开源 Redis - 数据缓存

参考教程 [Docker 大学生看了都会系列（四、常用命令实战）-CSDN 博客]

查看结果:

```
root@VM-8-12-ubuntu:/home/mysql/conf# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS                                 PORTS                                      NAMES
43c92adab074   redis     "docker-entrypoint.s…"   23 seconds ago   Up 22 seconds                          0.0.0.0:6379->6379/tcp, :::6379->6379/tcp  myredis
5b63490c449e   mysql     "docker-entrypoint.s…"   21 minutes ago   Restarting (1) 41 seconds ago                                             mysql
```

看到 Redis 容器成功运行

4. 搭建开源 RocketMQ - 异步解耦

参考教程 Docker 部署 RocketMQ（图文并茂超详细）_docker rocketmq-CSDN 博客

成果：

4.1 部署 NameServer 并查看 NameServer 启动日志

```
The Name Server boot success. serializeType=JSON, address 0.0.0.0:9876
```

这表示 NameServer 启动成功

4.2 部署 Broker+Proxy 并查看 Broker 启动日志

4.2.1 配置 broker.conf

```
vim /data/rocketmq/broker/conf/broker.conf
```

```
namesrvAddr = 127.0.0.1:9876
brokerClusterName = DefaultCluster
brokerName = broker-a
brokerId = 0
brokerIP1 = 127.0.0.1
brokerRole = ASYNC_MASTER
flushDiskType = ASYNC_FLUSH
deleteWhen = 04
fileReservedTime = 72
autoCreateTopicEnable=true
autoCreateSubscriptionGroup=true
tlsTestModeEnable = false
```

4.2.2 启动并查看 Broker 启动日志

```
root@VM-8-12-ubuntu:/home/mysql/conf# docker run -d --network rocketmq \
--restart=always --name rmqbroker --privileged=true \
-p 10911:10911 -p 10909:10909 \
-v /data/rocketmq/broker/logs:/root/logs \
-v /data/rocketmq/broker/store:/root/store \
-v /data/rocketmq/broker/conf/broker.conf:/home/rocketmq/broker.conf \
-v /data/rocketmq/broker/bin/runbroker.sh:/home/rocketmq/rocketmq-5.1.0/bin/runbroker.sh \
-e "NAMESRV_ADDR=rmqnamesrv:9876" \
apache/rocketmq:5.1.0 sh mqbroker --enable-proxy -c /home/rocketmq/broker.conf
9dd7454374d9d672215589b5de17aac1c6b667d0ee7d6a8377c663838c51bf56
root@VM-8-12-ubuntu:/home/mysql/conf# docker logs -f rmqbroker
OpenJDK 64-Bit Server VM warning: Using the DefNew young collector with the CMS collector is deprecated and wil
l likely be removed in a future release
OpenJDK 64-Bit Server VM warning: UseCMSCompactAtFullCollection is deprecated and will likely be removed in a f
uture release.
Thu Dec 12 13:04:15 UTC 2024 rocketmq-proxy startup successfully
```

这表示 Broker 启动成功

4.3 部署 RocketMQ 控制台 - rocketmq-dashboard

4.3.1 在安全组/防火墙放行 rocketmq-dashboard 运行在的端口

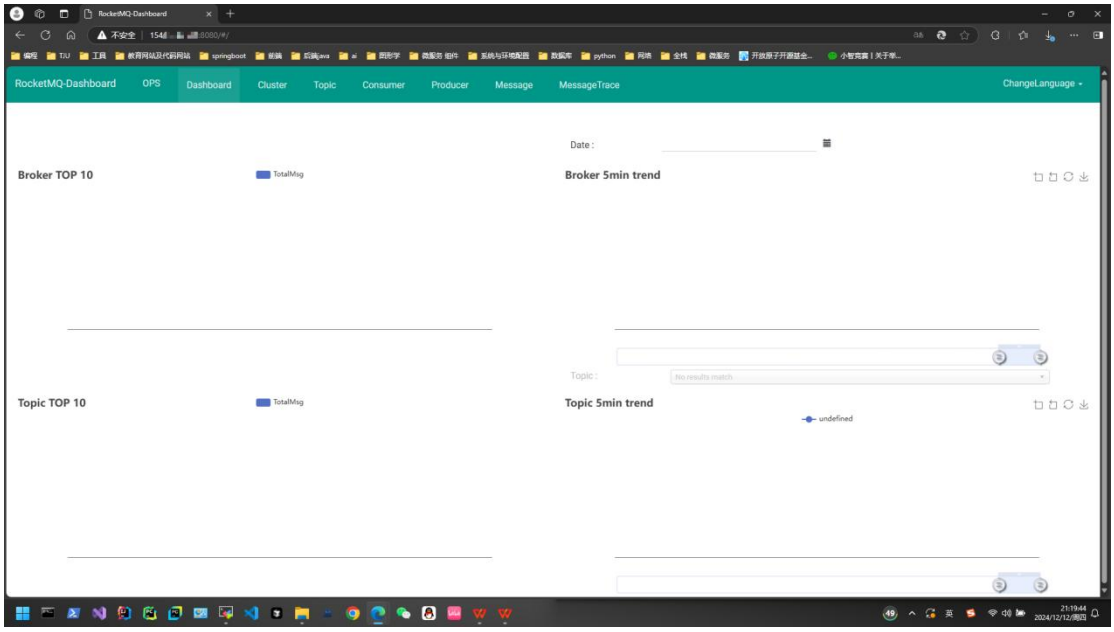
4.3.2 查看 dashboard 启动日志

```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v2.2.2.RELEASE)

[2024-12-12 13:11:18.812]  INFO Starting App v1.0.0 on 713fdb7328ef with PID 6 (/rocketmq-dashboard.jar started
 by root in /)
[2024-12-12 13:11:18.818]  INFO No active profile set, falling back to default profiles: default
[2024-12-12 13:11:21.803]  INFO Tomcat initialized with port(s): 8080 (http)
[2024-12-12 13:11:21.817]  INFO Initializing ProtocolHandler ["http-nio-0.0.0.0-8080"]
[2024-12-12 13:11:21.819]  INFO Starting service [Tomcat]
[2024-12-12 13:11:21.819]  INFO Starting Servlet engine: [Apache Tomcat/9.0.29]
[2024-12-12 13:11:22.040]  INFO Initializing Spring embedded WebApplicationContext
[2024-12-12 13:11:22.041]  INFO Root WebApplicationContext: initialization completed in 3051 ms
[2024-12-12 13:11:23.303]  INFO Initializing ExecutorService 'applicationTaskExecutor'
[2024-12-12 13:11:23.473]  INFO Adding welcome page: class path resource [static/index.html]
[2024-12-12 13:11:23.712]  INFO Initializing ExecutorService 'taskScheduler'
[2024-12-12 13:11:23.737]  INFO Exposing 2 endpoint(s) beneath base path '/actuator'
[2024-12-12 13:11:23.829]  INFO Starting ProtocolHandler ["http-nio-0.0.0.0-8080"]
[2024-12-12 13:11:23.858]  INFO Tomcat started on port(s): 8080 (http) with context path ''
[2024-12-12 13:11:23.862]  INFO Started App in 5.904 seconds (JVM running for 6.79)
```

成功在 8080 端口启动

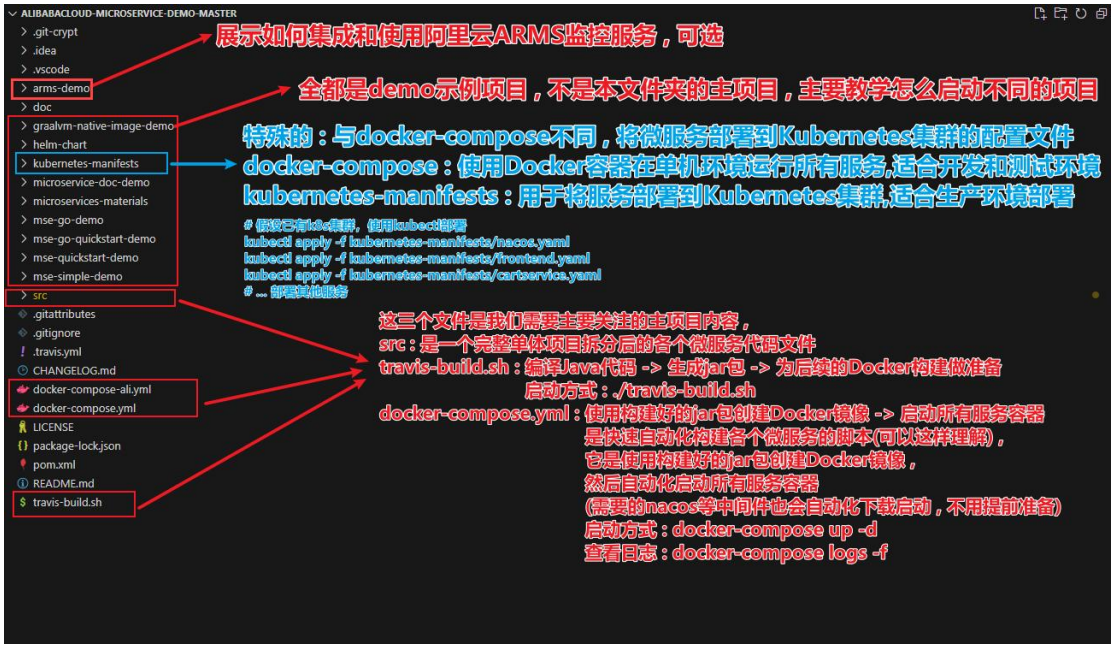
4.3.3 访问 RMQ 控制台



至此，成功搭建 RocketMQ

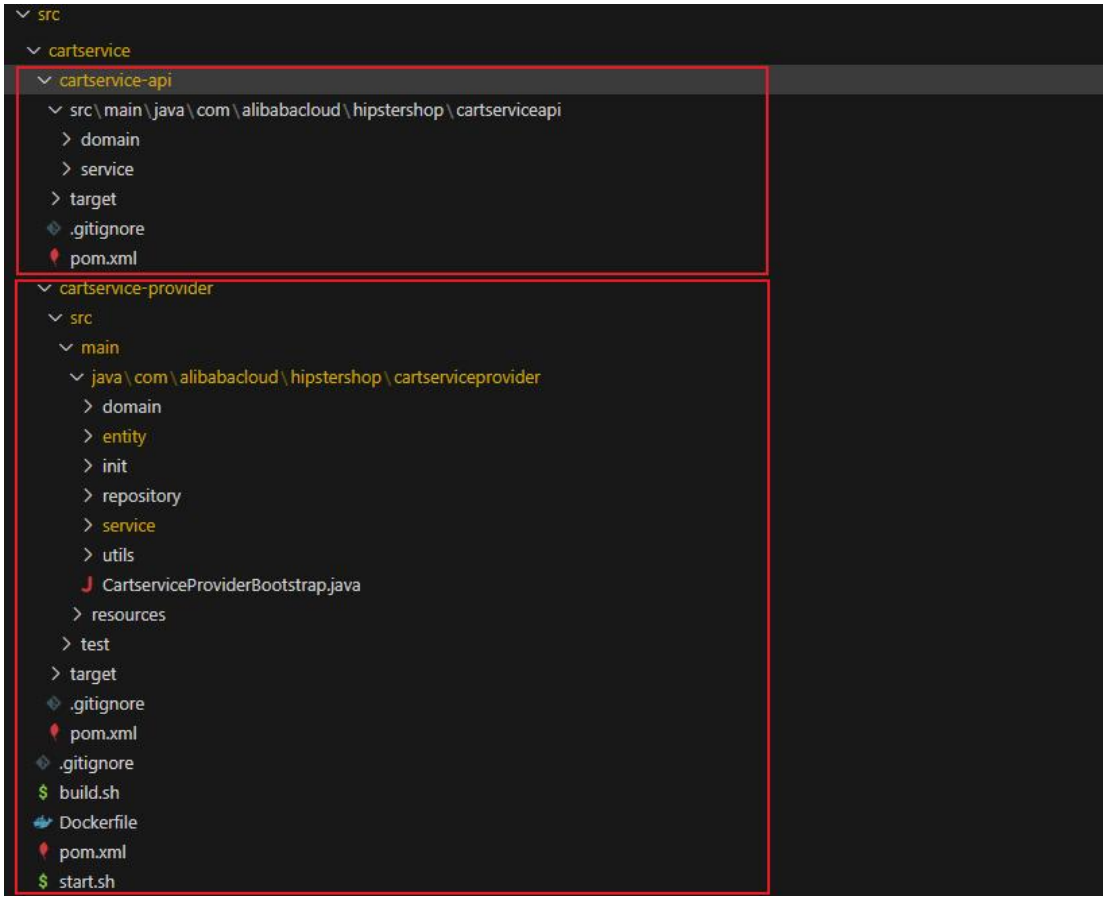
5. 剖析微服务代码结构并启动项目:

整体:



==========

Src 中具体微服务文件 -> 以 cartservice 为例 (其他微服务都类似)



分析 cartservice-api 文件夹的结构和功能：

```
cartservice-api/
├── src/main/java/com/alibaba/hipstershop/cartserviceapi/
│   ├── domain/          # API 相关的领域模型
│   └── service/         # 服务接口定义
├── target/              # 编译输出目录
├── .gitignore           # Git 忽略文件配置
└── pom.xml              # Maven 项目配置文件
```

详细功能分析：

- domain 目录
- 定义了购物车服务的数据传输对象(DTO)
- 包含接口间传递的请求和响应模型
- 这些对象在服务提供方和消费方之间共享
- 例如：
- CartDTO：购物车数据传输对象
- CartItemDTO：购物车商品项数据传输对象
- CartRequest：购物车操作请求对象
- CartResponse：购物车操作响应对象
- service 目录
- 定义了购物车服务的接口规范

- 包含所有对外暴露的服务方法定义
- 通常包括：

```
public interface CartService {
        // 添加商品到购物车
        CartResponse addItem(CartRequest request);

        // 从购物车移除商品
        CartResponse removeItem(CartRequest request);

        // 获取购物车内容
        CartDTO getCart(String userId);

        // 清空购物车
        void emptyCart(String userId);

        // 更新购物车商品数量
        CartResponse updateItemQuantity(CartRequest request);
    }
```

- pom.xml
- Maven 项目配置文件
- 定义项目依赖
- 包含构建配置
- 主要包括：
- 项目基本信息
- 依赖管理
- 编译配置
- 打包配置

重要特点：

- 接口设计
- 采用面向接口编程
- 定义清晰的服务边界
- 支持服务解耦
- 版本管理
- 独立的版本控制
- 支持接口的演进
- 向后兼容性维护
- 依赖隔离
- 最小化依赖
- 只包含必要的接口定义
- 避免实现细节泄露

- 跨服务通信
- 定义服务间通信契约
- 支持 RPC 调用
- 确保服务兼容性

这个 API 模块的主要作用是：

- 定义服务接口规范
- 提供数据传输对象
- 实现服务解耦
- 支持分布式部署
- 便于服务治理

通过将 API 单独抽取为一个模块，可以：

- 更好地管理服务版本
- 简化服务集成
- 提高代码复用性
- 降低服务间耦合

==========

分析 cartservice-provider 文件夹中的文件结构和功能：

```
cartservice-provider/
├── domain/          # 领域模型层
├── api/             # API 接口定义
├── repository/      # 数据访问层
├── service/         # 业务逻辑层
├── util/            # 工具类
├── CartServiceProviderBootstrap.java  # 启动类
└── test/            # 测试代码
```

详细解释每个部分的功能：

- domain 目录
- 包含购物车领域的核心业务实体
- 定义了购物车相关的数据模型和业务规则
- 例如 Cart、CartItem 等领域对象
- api 目录
- 定义了购物车服务对外暴露的接口
- 包含接口方法定义和请求/响应对象
- 其他服务通过这些接口与购物车服务交互
- repository 目录
- 负责数据持久化层
- 定义了与数据库交互的接口和实现
- 处理购物车数据的存储和查询操作
- service 目录
- 包含核心业务逻辑的实现
- 实现购物车的添加、删除、更新等操作

- 协调领域对象和数据访问层
- util 目录
- 包含通用工具类
- 可能包含日期处理、字符串处理等辅助功能
- 提供各种通用的工具方法
- CartServiceProviderBootstrap.java
- 服务的启动入口类
- 配置服务启动参数
- 初始化 Spring 容器和服务注册
- test 目录
- 包含单元测试和集成测试代码
- 验证各个组件的功能正确性
- 确保代码质量和可靠性

主要功能：

- 购物车数据管理
- 创建新购物车
- 添加/删除商品
- 更新商品数量
- 清空购物车
- 数据持久化
- 保存购物车状态
- 读取用户购物车信息
- 管理购物车数据的生命周期

服务集成

- 与商品服务集成
- 与用户服务集成
- 提供 RPC 接口供其他服务调用
- 性能和可靠性
- 缓存机制
- 异常处理
- 事务管理
- 并发控制

==========

下面实践如何部署+运行微服务代码:

(经过多次失败 ，得出经验 ：最好使用境外服务器 ，国内网络环境会导致很多地方失败 ，而且最好是弹性服务器 ，配置最少 4 核 8 G 并且 2 核 4 G 服务器也跑不了这个微服务项目)

我的服务器配置:

系统信息

系统名称	Ubuntu 重装系统
系统版本	Ubuntu-22.04-x64

实例规格

CPU	4核
内存	8G
系统盘	50G
网络类型	经典网络
上行宽带	10Mbps
下行宽带	10Mbps
IP数量	1IP
快照数量	1个
备份数量	1个
数据盘	0G数据盘

5.1 使用 MobaXterm 上传代码文件到 Ubuntu 服务器

可能会报错 ，原因是权限不够 ，可以执行

```
sudo chmod 777 /home
```

/home 换成自己上传代码的文件夹

再上传代码就可以了

附 ：常见文件权限

```
sudo chmod 600 xxx （只有所有者有读和写的权限）
```

```
sudo chmod 644 xxx （所有者有读和写的权限，组用户只有读的权限）
```

```
sudo chmod 700 xxx （只有所有者有读和写以及执行的权限）
```

```
sudo chmod 666 xxx （每个人都有读和写的权限）
```

```
sudo chmod 777 xxx （每个人都有读和写以及执行的权限）
```

5.1.1 上传可能比较慢 ，还需耐心等待 ，也可以先在 windows 上将代码文件打一个 tar 包 ，上传 tar 包到服务器上 ，在服务器上执行 tar 包解压命令即可

```
root@VM-8-12-ubuntu:/# sudo chmod 777 /home
root@VM-8-12-ubuntu:/# cd /home
root@VM-8-12-ubuntu:/home# l
alibabacloud-microservice-demo-master.tar  lighthouse/  mysql/  ubuntu/
root@VM-8-12-ubuntu:/home# tar -xvf alibabacloud-microservice-demo-master.tar
alibabacloud-microservice-demo-master/
alibabacloud-microservice-demo-master/.git-crypt/
alibabacloud-microservice-demo-master/.git-crypt/.gitattributes
alibabacloud-microservice-demo-master/.git-crypt/keys/
alibabacloud-microservice-demo-master/.git-crypt/keys/default/
alibabacloud-microservice-demo-master/.git-crypt/keys/default/0/
alibabacloud-microservice-demo-master/.git-crypt/keys/default/0/0ECB9E7547E841622DFB2273F73736F351109763.gpg
alibabacloud-microservice-demo-master/.gitattributes
alibabacloud-microservice-demo-master/.gitignore
alibabacloud-microservice-demo-master/.idea/
alibabacloud-microservice-demo-master/.idea/.gitignore
```

亲测 - 打 tar 包上传再解压的方式超级快 ！

5.2 搭建环境并启动微服务项目

使用 Docker Compose 方式（用于开发/测试）：

5.2.1 搭建环境

因为是新服务器 ，需要先执行一些命令 ，搭建好运行环境 (先使用 su 命令获取最高权限 ，

或者在每次使用命令的时候在命令前面加上 sudo)
下面是提前需要执行的命令和一些详细信息/注意事项:

apt-get update

apt-get install openjdk-21-jre：此处必须使用高版本的 java，亲测 openjdk-8-jdk 会在之后运行这个微服务项目的时候报错

apt-get install maven：参考教程 maven 的安装与配置（Command ‘mvn ‘ not found）修改配置文件后新终端依旧无法识别到 mvn 命令 command not found: mvn-CSDN 博客

apt-get install docker.io

apt-get install docker-compose

5.2.2 启动微服务项目

5.2.2.1 操作文件时可能会有权限问题

```
Permission denied
```

解决方法(递归设置文件权限):

```
sudo chmod -R 777 /home/alibabacloud-microservice-demo-master
```

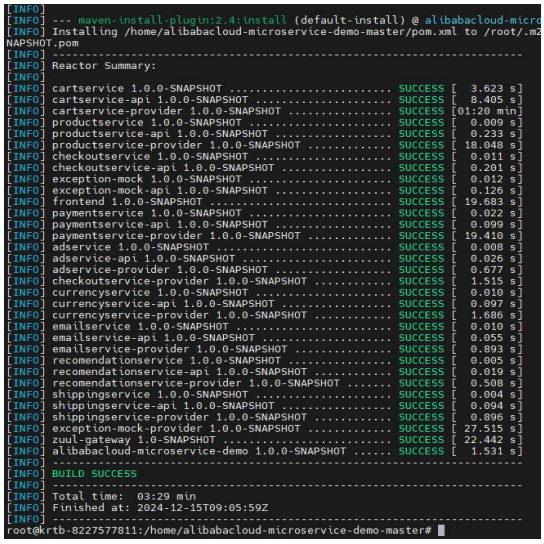
5.2.2.2 启动命令

依次执行下面两个命令

mvn clear：可能在运行后有一些红色 error，不用管 ，接着执行下面的命令

mvn install

执行 mvn install 后 ：



5.3. 启动所有服务

执行下面的启动命令:

docker-compose up --build

这种方式会在单机上启动所有服务，包括：

nacos（注册中心）

frontend（前端服务，端口 8080）

cartservice（购物车服务）

checkoutservice（结账服务）

productservice（产品服务）

相关的 MySQL 和 Redis 等基础服务

注意事项:

5.3.1. 成功自动化启动所有中间件与微服务:

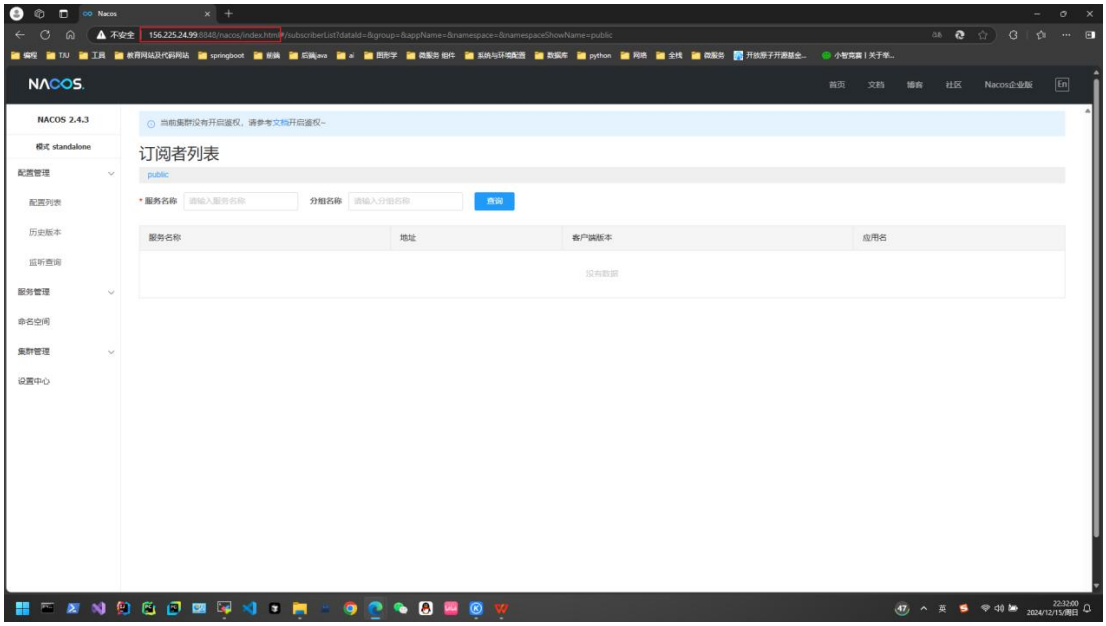
```
cartservice         |  :: Dubbo Spring Boot (v2.7.18) : https://github.com/apache/dubbo-spring-boot-project
cartservice         |  :: Dubbo (v2.7.18) : https://github.com/apache/dubbo
cartservice         |  :: Discuss group : dev@dubbo.apache.org
cartservice         |
cartservice         | 2024-12-15 09:20:10.392  INFO 7 --- [           main] e.OverrideDubboConfigApplicationListener : Dubbo Config was overridden by externalized configuration {dubbo.application.name=cartservice, dubbo.application.qos-accept-foreign-ip=false, dubbo.application.qos-enable=true, dubbo.cloud.subscribed-services=cartservice, dubbo.config.multiple=true, dubbo.protocol.dispatcher=message, dubbo.protocol.name=dubbo, dubbo.protocol.port=-1, dubbo.protocol.queues=6000, dubbo.protocol.threads=600, dubbo.provider.delay=5000, dubbo.provider.warmup=60000, dubbo.registry.address=nacos://nacos-server:8848, dubbo.scan.base-packages=com.alibabacloud.hipstershop.cartserviceprovider.service}
product-mysql_1     | 2024-12-15 09:20:10 93 [Note] InnoDB: Shutdown completed; log sequence number 1625997
product-mysql_1     | 2024-12-15 09:20:10 93 [Note] Shutting down plugin 'BLACKHOLE'
product-mysql_1     | 2024-12-15 09:20:10 93 [Note] Shutting down plugin 'ARCHIVE'
product-mysql_1     | 2024-12-15 09:20:10 93 [Note] Shutting down plugin 'MRG_MYISAM'
product-mysql_1     | 2024-12-15 09:20:10 93 [Note] Shutting down plugin 'MyISAM'
product-mysql_1     | 2024-12-15 09:20:10 93 [Note] Shutting down plugin 'MEMORY'
product-mysql_1     | 2024-12-15 09:20:10 93 [Note] Shutting down plugin 'CSV'
product-mysql_1     | 2024-12-15 09:20:10 93 [Note] Shutting down plugin 'sha256_password'
product-mysql_1     | 2024-12-15 09:20:10 93 [Note] Shutting down plugin 'mysql_old_password'
product-mysql_1     | 2024-12-15 09:20:10 93 [Note] Shutting down plugin 'mysql_native_password'
product-mysql_1     | 2024-12-15 09:20:10 93 [Note] Shutting down plugin 'binlog'
product-mysql_1     | 2024-12-15 09:20:10 93 [Note] mysqld: Shutdown complete
product-mysql_1     |
checkoutservice     | [15/12/24 09:20:10:998 UTC] main  INFO logger.LoggerFactory: using logger: org.apache.dubbo.common.logger.log4j.Log4jLoggerAdapter
productservice      | log4j:WARN No appenders could be found for logger (org.apache.dubbo.common.logger.LoggerFactory).
productservice      | log4j:WARN Please initialize the log4j system properly.
productservice      | log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
cartservice         |
cartservice         |   .   ____          _            __ _ _
cartservice         |  /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
cartservice         | ( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
cartservice         |  \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
cartservice         |   '  |____| .__|_| |_|_| |_\__, | / / / /
cartservice         |  =========|_|==============|___/=/_/_/_/
cartservice         |  :: Spring Boot ::        (v2.2.5.RELEASE)
cartservice         |
checkoutservice     | [15/12/24 09:20:11:359 UTC] main ERROR common.Version:  [DUBBO] Duplicate class org/apache/dubbo/common/Version.class in 2 jar [file:/app/checkoutservice-provider-1.0.0-SNAPSHOT.jar!/BOOT-INF/lib/dubbo-common-2.7.18.jar!/org/apache/dubbo/common/Version.class, file:/app/checkoutservice-provider-1.0.0-SNAPSHOT.jar!/BOOT-INF/lib/dubbo-2.7.18.jar!/org/apache/dubbo/common/Version.class], dubbo version: 2.7.18, current host: 172.18.0.6
productservice      | 2024-12-15 09:20:11.352  INFO 7 --- [           main] d.s.b.c.e.WelcomeLogoApplicationListener :
productservice      |
productservice      |  :: Dubbo Spring Boot (v2.7.18) : https://github.com/apache/dubbo-spring-boot-project
productservice      |  :: Dubbo (v2.7.18) : https://github.com/apache/dubbo
productservice      |  :: Discuss group : dev@dubbo.apache.org
productservice      |
checkoutservice     | 2024-12-15 09:20:11.410  INFO 6 --- [           main] d.s.b.c.e.WelcomeLogoApplicationListener :
checkoutservice     |
checkoutservice     |  :: Dubbo Spring Boot (v2.7.18) : https://github.com/apache/dubbo-spring-boot-project
checkoutservice     |  :: Dubbo (v2.7.18) : https://github.com/apache/dubbo
checkoutservice     |  :: Discuss group : dev@dubbo.apache.org
checkoutservice     |
checkout-mysql_1    | 2024-12-15 09:20:11 92 [Warning] 'proxies_priv' entry '@ root@4d7803456bd9' ignored in --skip-name-resolve mode.
checkout-mysql_1    | 2024-12-15 09:20:11+00:00 [Note] [Entrypoint]: Creating database checkout
checkout-mysql_1    |
checkout-mysql_1    | 2024-12-15 09:20:11+00:00 [Note] [Entrypoint]: Stopping temporary server
checkout-mysql_1    | 2024-12-15 09:20:11 92 [Note] mysqld: Normal shutdown
checkout-mysql_1    |
checkout-mysql_1    | 2024-12-15 09:20:11 92 [Note] Giving 0 client threads a chance to die gracefully
checkout-mysql_1    | 2024-12-15 09:20:11 92 [Note] Event Scheduler: Purging the queue. 0 events
checkout-mysql_1    | 2024-12-15 09:20:11 92 [Note] Shutting down slave threads
```

```
checkout-mysql_1    | 2024-12-15 09:21:04 1 [Warning] Insecure configuration for --pid-file: Location '/var/run/mysqld' in the path is accessible to all OS users. Consider choosing a different directory.
checkout-mysql_1    | 2024-12-15 09:21:04 1 [Warning] 'proxies_priv' entry '@ root@4d7803456bd9' ignored in --skip-name-resolve mode.
checkout-mysql_1    | 2024-12-15 09:21:04 1 [Note] Event Scheduler: Loaded 0 events
checkout-mysql_1    | 2024-12-15 09:21:04 1 [Note] mysqld: ready for connections.
checkout-mysql_1    | Version: '5.6.51'  socket: '/var/run/mysqld/mysqld.sock'  port: 3306  MySQL Community Server (GPL)
nacos-standalone    |
nacos-standalone    |                                        Nacos 2.4.3
nacos-standalone    |                                        Running in stand alone mode, All function modules
nacos-standalone    |                                        Port: 8848
nacos-standalone    |                                        Pid: 1
nacos-standalone    |                                        Console: http://nacos:8848/nacos/index.html
nacos-standalone    |
nacos-standalone    |                                           https://nacos.io
nacos-standalone    |
alibabacloud-microservice-demo-master_product-mysql_1 exited with code 137
productservice      | 2024-12-15 09:21:09.108  INFO 7 --- [pool-1-thread-1] .b.c.e.AwaitingNonWebApplicationListener :  [Dubbo] Current Spring Boot Application is await...
checkoutservice     | 2024-12-15 09:21:09.563  INFO 6 --- [pool-1-thread-1] .b.c.e.AwaitingNonWebApplicationListener :  [Dubbo] Current Spring Boot Application is await...
productservice      | 2024-12-15 09:21:09.882  INFO 7 --- [           main] e.OverrideDubboConfigApplicationListener : Dubbo Config was overridden by externalized configuration {dubbo.application.name=productservice, dubbo.application.qos-accept-foreign-ip=false, dubbo.application.qos-enable=true, dubbo.cloud.subscribed-services=productservice, dubbo.config.multiple=true, dubbo.protocol.dispatcher=message, dubbo.protocol.name=dubbo, dubbo.protocol.port=-1, dubbo.protocol.queues=6000, dubbo.protocol.threads=600, dubbo.provider.delay=5000, dubbo.provider.warmup=60000, dubbo.registry.address=nacos://nacos-server:8848, dubbo.scan.base-packages=com.alibabacloud.hipstershop.productservice.service}
productservice      |
productservice      |   .   ____          _            __ _ _
productservice      |  /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
productservice      | ( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
productservice      |  \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
productservice      |   '  |____| .__|_| |_|_| |_\__, | / / / /
productservice      |  =========|_|==============|___/=/_/_/_/
productservice      |  :: Spring Boot ::        (v2.2.5.RELEASE)
productservice      |
checkoutservice     | 2024-12-15 09:21:10.609  INFO 6 --- [           main] e.OverrideDubboConfigApplicationListener : Dubbo Config was overridden by externalized configuration {dubbo.application.name=checkoutservice, dubbo.application.qos-accept-foreign-ip=false, dubbo.application.qos-enable=true, dubbo.cloud.subscribed-services=checkoutservice,cartservice,productservice, dubbo.config.multiple=true, dubbo.consumer.init=true, dubbo.consumer.retries=0, dubbo.consumer.timeout=1000, dubbo.protocol.dispatcher=message, dubbo.protocol.name=dubbo, dubbo.protocol.port=-1, dubbo.protocol.queues=0, dubbo.protocol.threads=600, dubbo.provider.delay=5000, dubbo.provider.warmup=60000, dubbo.registry.address=nacos://nacos-server:8848, dubbo.scan.base-packages=com.alibabacloud.hipstershop.checkoutservice.service}
checkoutservice     |
checkoutservice     |   .   ____          _            __ _ _
checkoutservice     |  /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
checkoutservice     | ( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
checkoutservice     |  \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
checkoutservice     |   '  |____| .__|_| |_|_| |_\__, | / / / /
checkoutservice     |  =========|_|==============|___/=/_/_/_/
checkoutservice     |  :: Spring Boot ::        (v2.2.5.RELEASE)
checkoutservice     |
productservice      | 2024-12-15 09:21:11.545  INFO 7 --- [           main] org.reflections.Reflections              : Reflections took 789 ms to scan 1 urls, producing 3 keys and 6 values
cartservice         | 2024-12-15 09:21:12.048  INFO 7 --- [           main] com.alibaba.spring.util.BeanRegistrar    : The Infrastructure bean definition [Root bean: class [org.apache.dubbo.config.spring.beans.factory.annotation.ReferenceAnnotationBeanPostProcessor]; scope=; abstract=false; lazyInit=null; autowireMode=0; dependencyCheck=0; autowireCandidate=true; primary=false; factoryBeanName=null; factoryMethodName=null; initMethodName=null; destroyMethodName=nullwith name [referenceAnnotationBeanPostProcessor] has been registered.
cartservice         | 2024-12-15 09:21:12.050  INFO 7 --- [           main] com.alibaba.spring.util.BeanRegistrar    : The Infrastructure bean definition [Root bean: class [org.apache.dubbo.config.spring.beans.factory.annotation.DubboConfigAliasPostProcessor]; scope=; abstract=false; lazyInit=null; autowireMode=0; dependencyCheck=0; autowireCandidate=true; primary=false; factoryBeanName=null; factoryMethodName=null; initMethodName=null; destroyMethodName=nullwith name [dubboConfigAliasPostProcessor] has been registered.
cartservice         | 2024-12-15 09:21:12.060  INFO 7 --- [           main] com.alibaba.spring.util.BeanRegistrar    : The Infrastructure bean definition [Root bean: class [org.apache.dubbo.config.spring.con
```

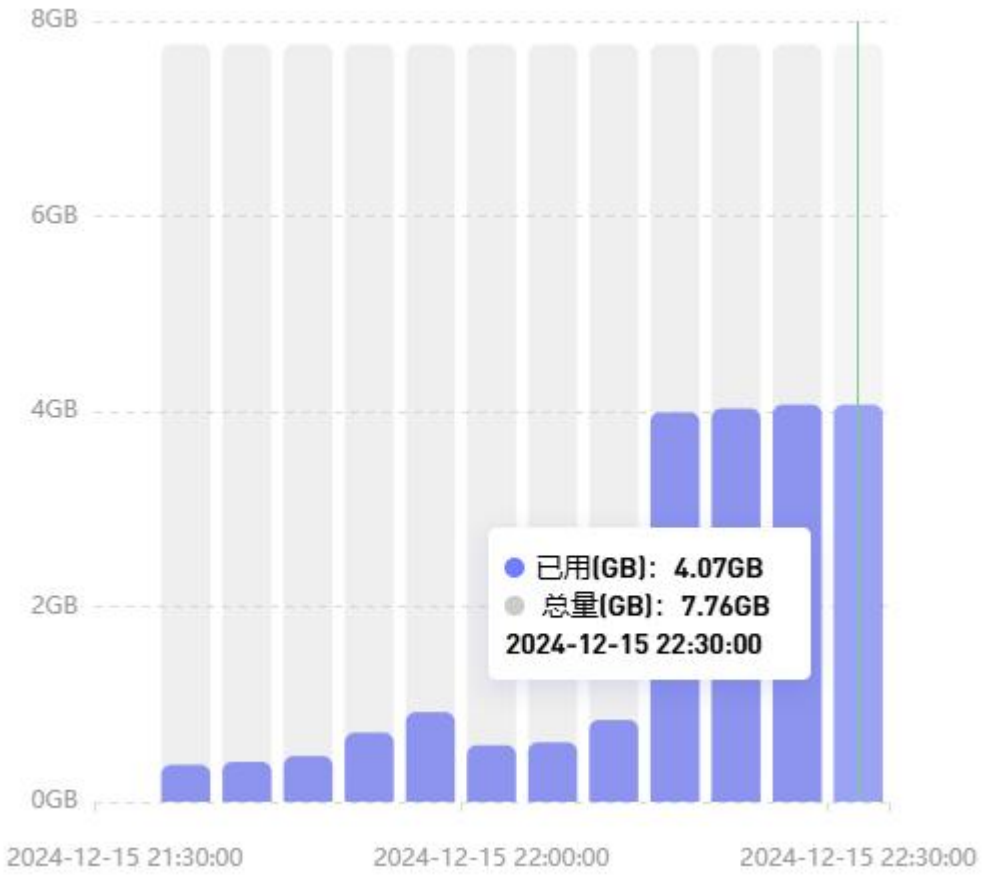
5.4 使用 docker ps 命令查看当前运行的容器

```
root@ser412579206836:~# docker ps
CONTAINER ID   IMAGE                          COMMAND                  CREATED         STATUS         PORTS                     NAMES
66d2dc7e789b   checkoutservice:1.0.0-SNAPSHOT "/app/start.sh"          4 minutes ago   Up 4 minutes   8080/tcp                  checkoutservice
b4a3b829882c   cartservice:1.0.0-SNAPSHOT     "/app/start.sh"          4 minutes ago   Up 4 minutes   8080/tcp                  cartservice
ae1576318d71   mysql:5.6                      "docker-entrypoint.s…"   4 minutes ago   Up 4 minutes   0.0.0.0:32768->3306/tcp   alibabacloud-microservice-demo-master_product-mysql_1
21779785b135   redis                          "docker-entrypoint.s…"   4 minutes ago   Up 4 minutes   0.0.0.0:6379->6379/tcp    alibabacloud-microservice-demo-master_cart-redis_1
8f16bc05cb99   mysql:5.6                      "docker-entrypoint.s…"   4 minutes ago   Up 4 minutes   0.0.0.0:32769->3306/tcp   alibabacloud-microservice-demo-master_checkout-mysql_1
bc90fb6ec393   nacos/nacos-server:latest      "sh bin/docker-start…"   4 minutes ago   Up 4 minutes   0.0.0.0:8848->8848/tcp    nacos-standalone
```

5.5 在浏览器可以看到部署在服务器上的 nacos 控制面板



可以看到 ，启动后 ubuntu 一共占用 4.07G 内存 ，这就是为什么刚才说 4G 系统跑不了项目(我也在这里踩了两次坑 ，内存从 2G 换到 4G，从 4G 换到 8G，才解决无法启动的问题)



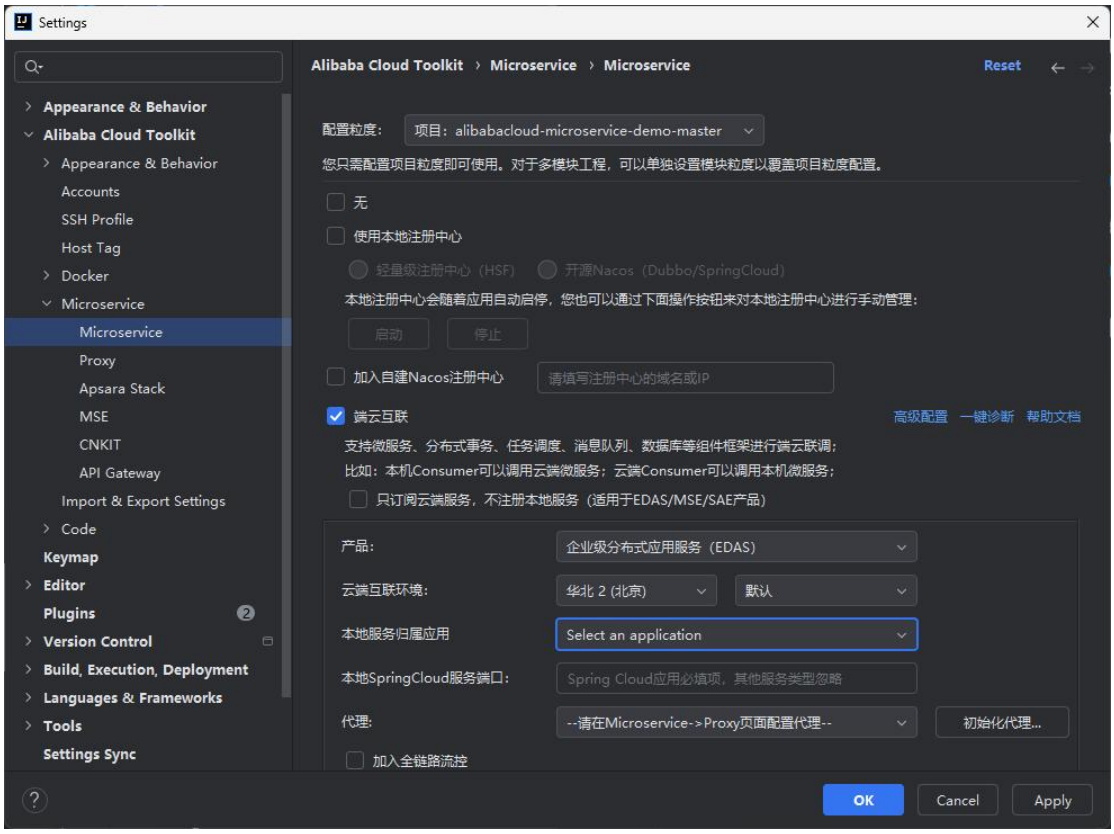
6. 使用 Cloud Toolkit 进行端云联调 - (IntelliJ IDEA)

参考教程[如何在 IntelliJ IDEA 中使用 Cloud Toolkit 实现端云互联_企业级分布式应用服务(EDAS)-阿里云帮助中心]

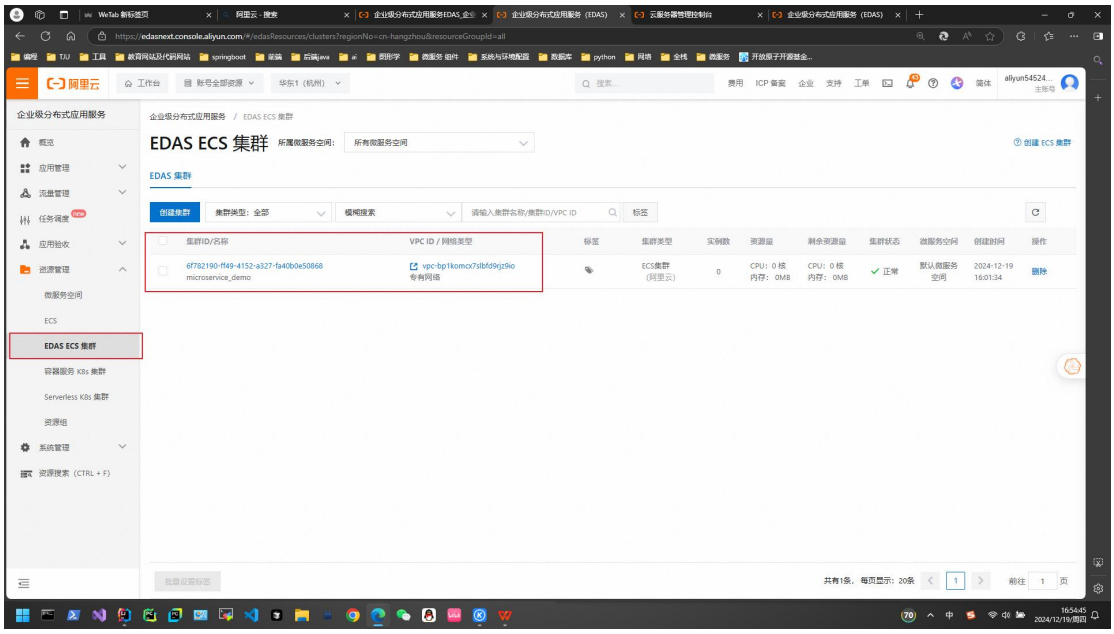
了解远程调试与端云联调 ， 参考 bilibili 视频 ： BV1sy4y1v78y

了解并使用 DEAS ： bilibili 视频 ： BV1dY411a7p1 第 16 课时+第 17 课时

6.1 配置端云互联:



6.1.1 创建一个 EDAS ECS 集群



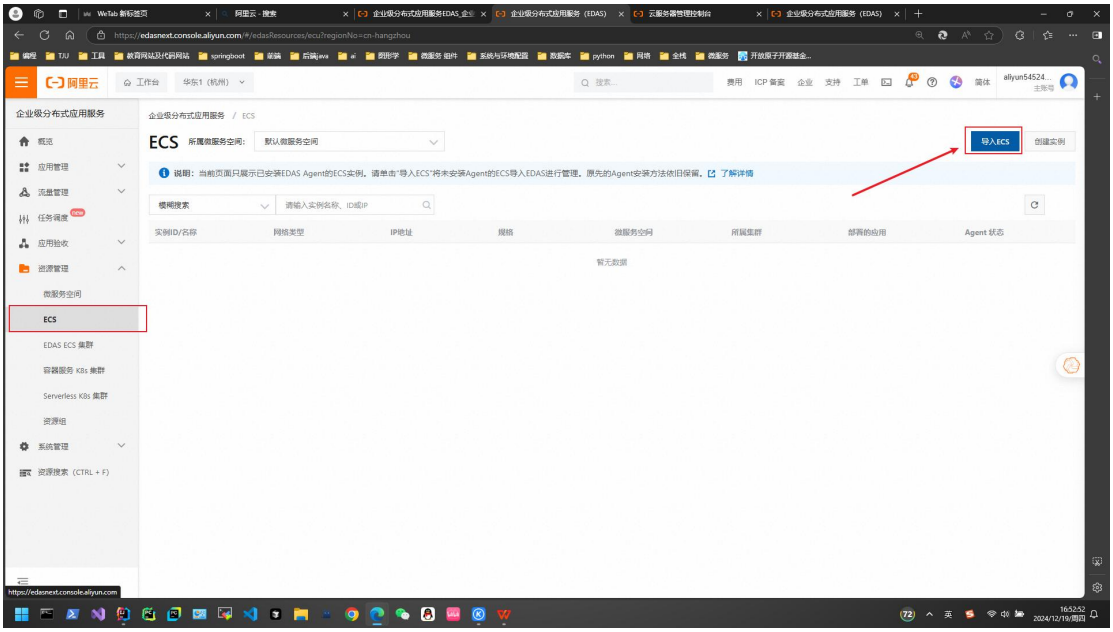
记住 vpc 网络(我的是华东 1(杭州)H)

6.1.2 安装 ECS 实例代理机

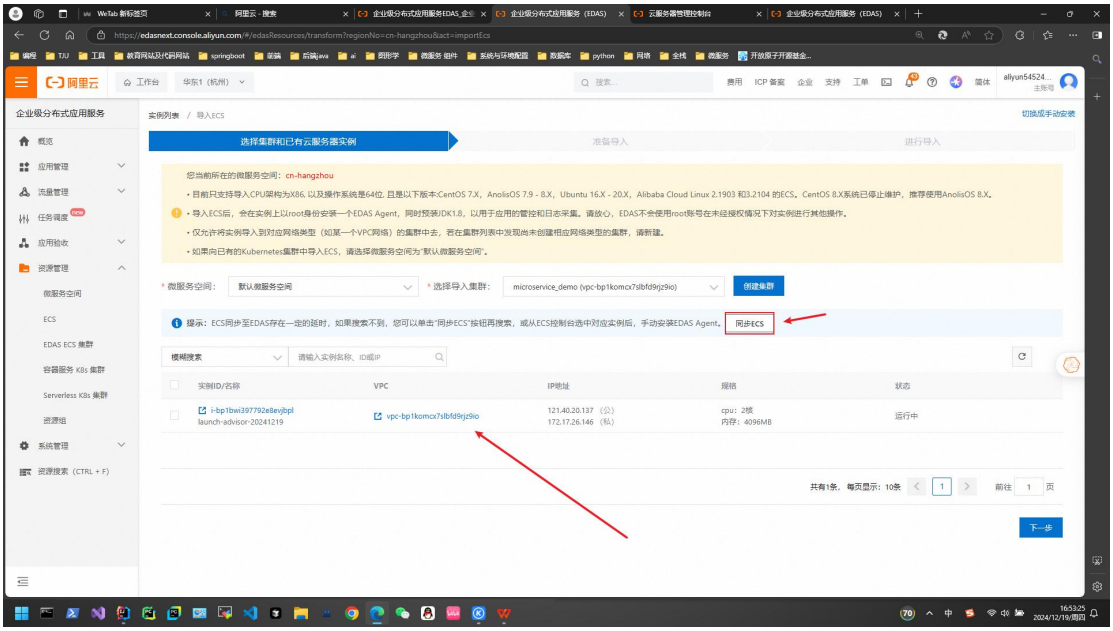
6.1.2.1 确保安装 EDAS Agent 镜像的 ECS 实例 ： 在 ECS 实例中安装 edas_agent_企业级分布式应用服务(EDAS)-阿里云帮助中心

6.1.2.2 安装的时候保证 ECS 实例与 EDAS ECS 集群在同一个 vpc 网络下

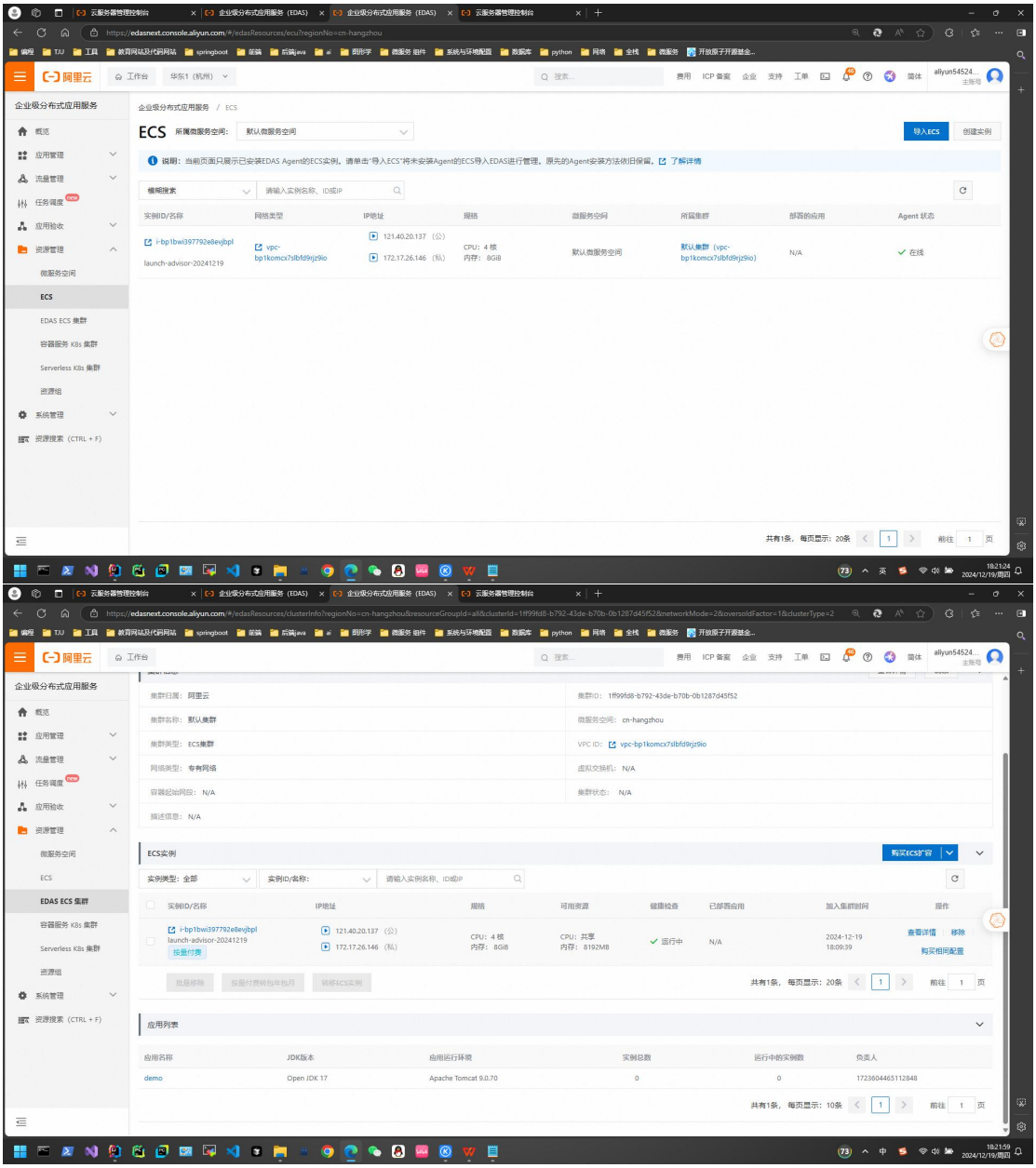
6.1.3 找到代理机



同步之后可以看到代理机

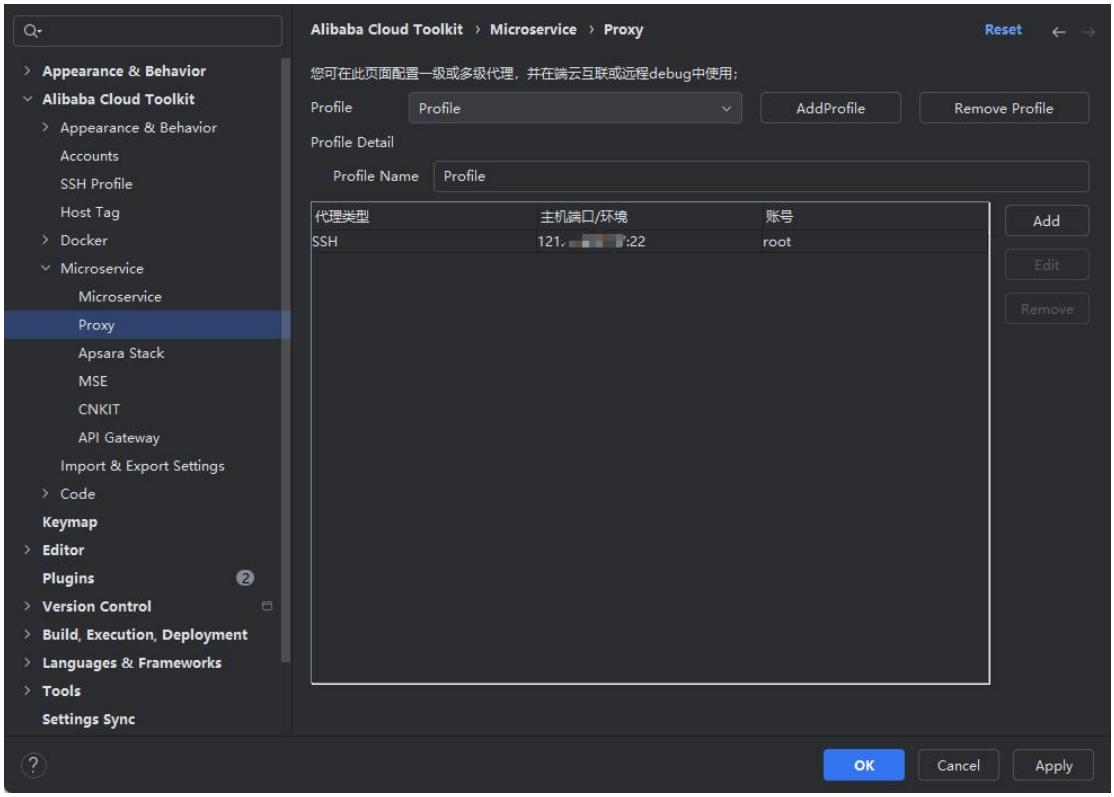


导入代理机:



6.1.4 配置 proxy:

使用代理机的 ip 与默认端口 22 (确保安全组中对端口 22 的进入策略是放行的)



6.1.5 创建应用

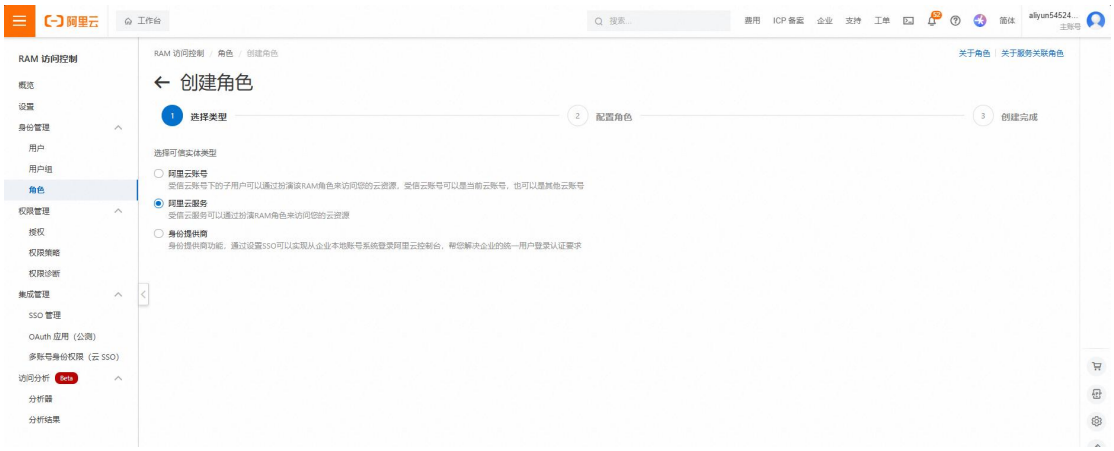
然而创建失败了，这里我整整研究了三天，问题出在权限上

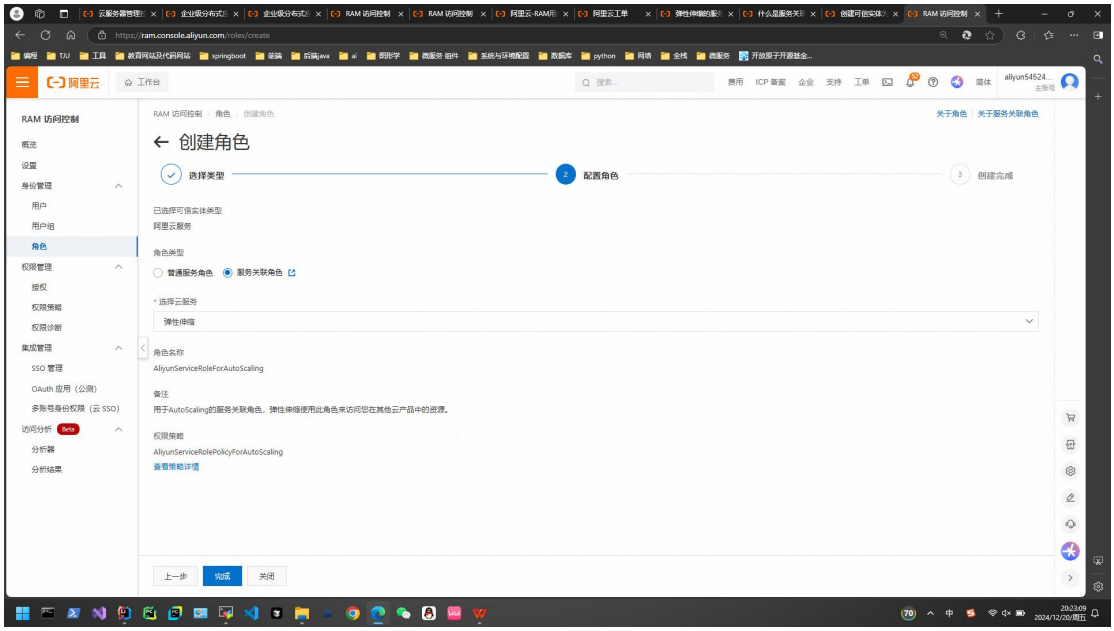
解决方案:

https://help.aliyun.com/zh/edas/user-guide/how-do-i-troubleshoot-issues-in-a-change-process#7cfa540074upg

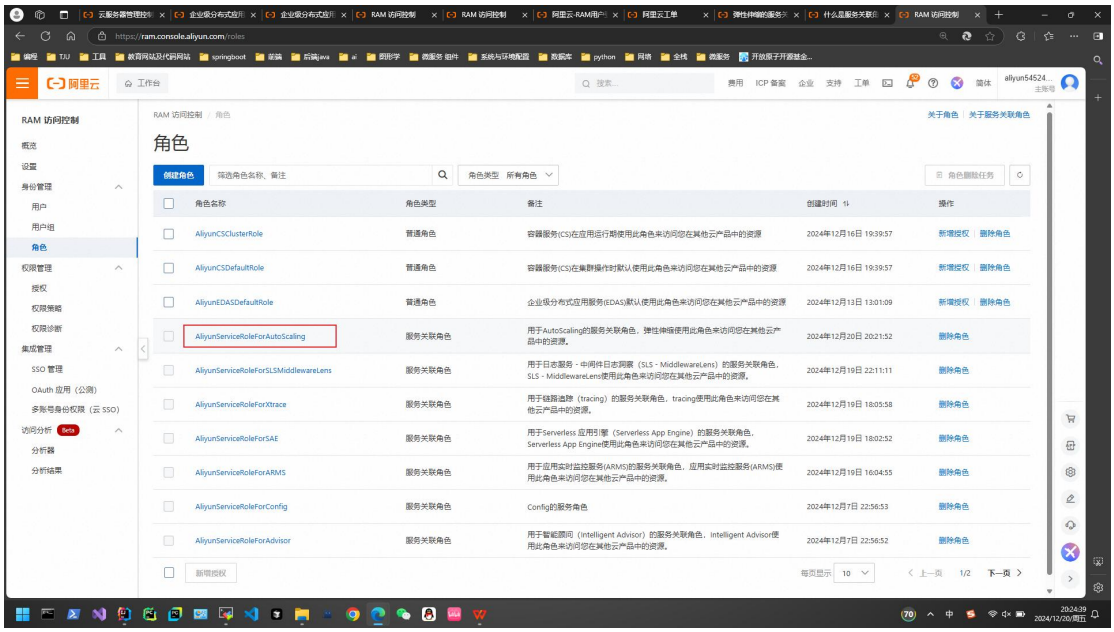
参考上面的教程创建服务关联角色

下面是具体操作:

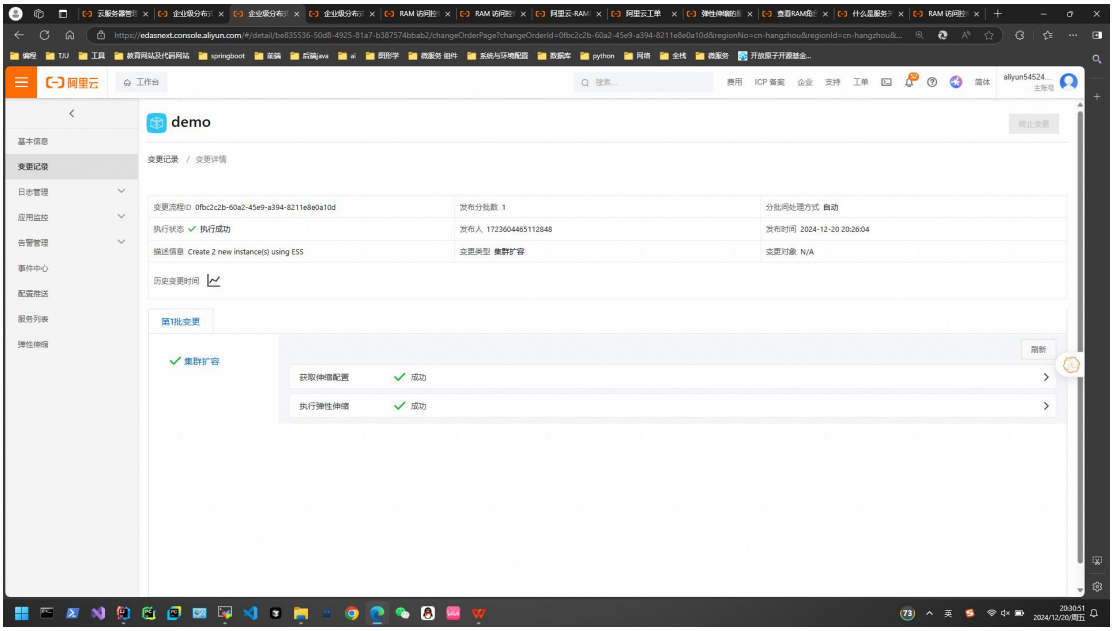




创建成功:



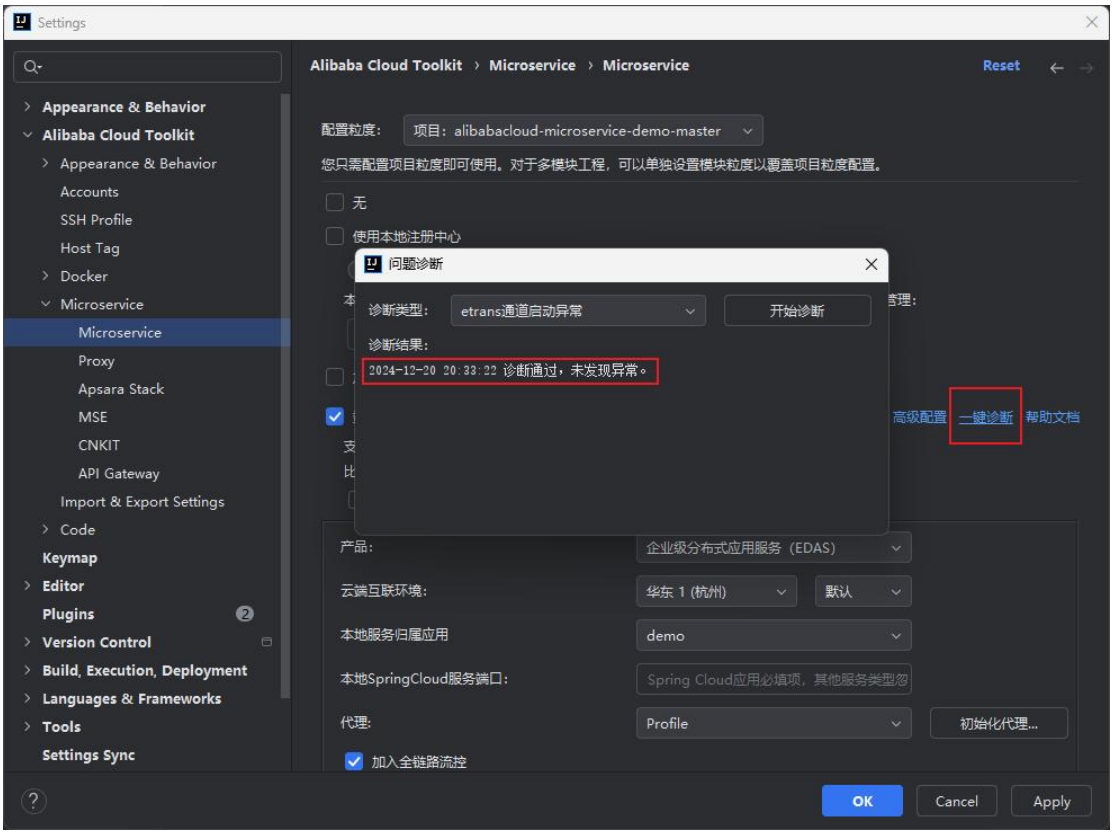
再次尝试创建应用:



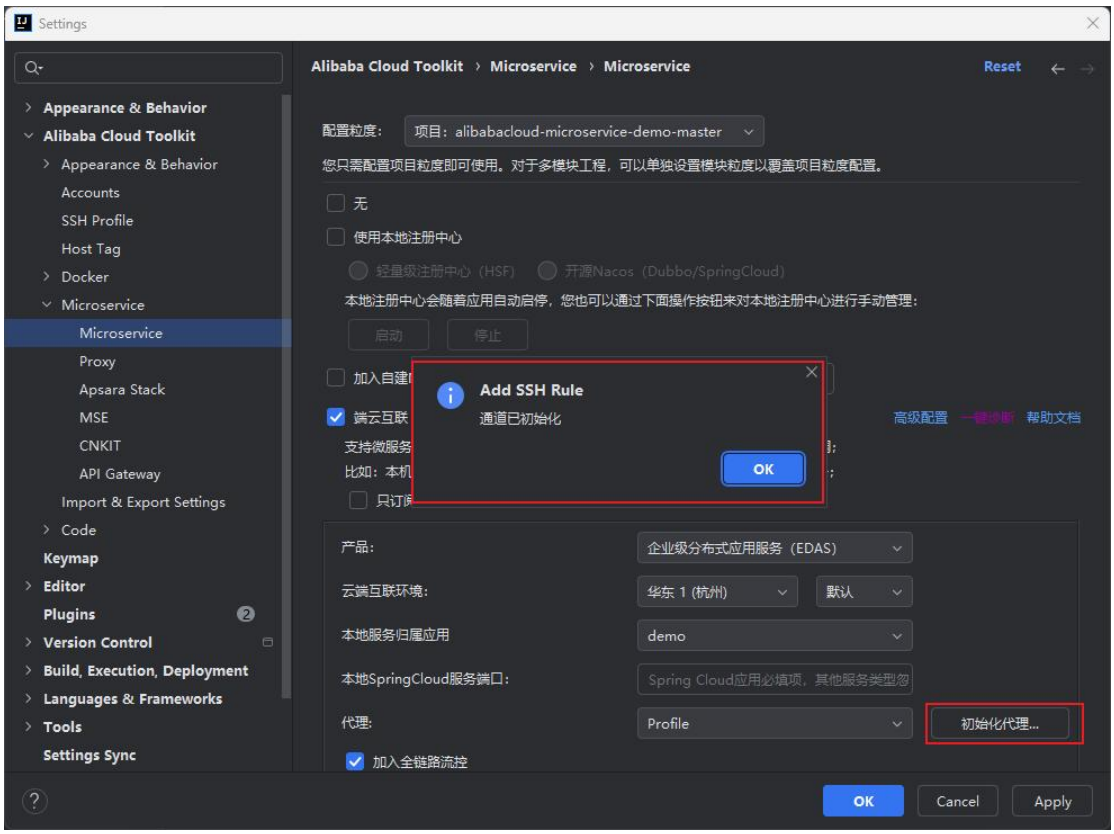
创建应用成功

6.2 启动端云互联

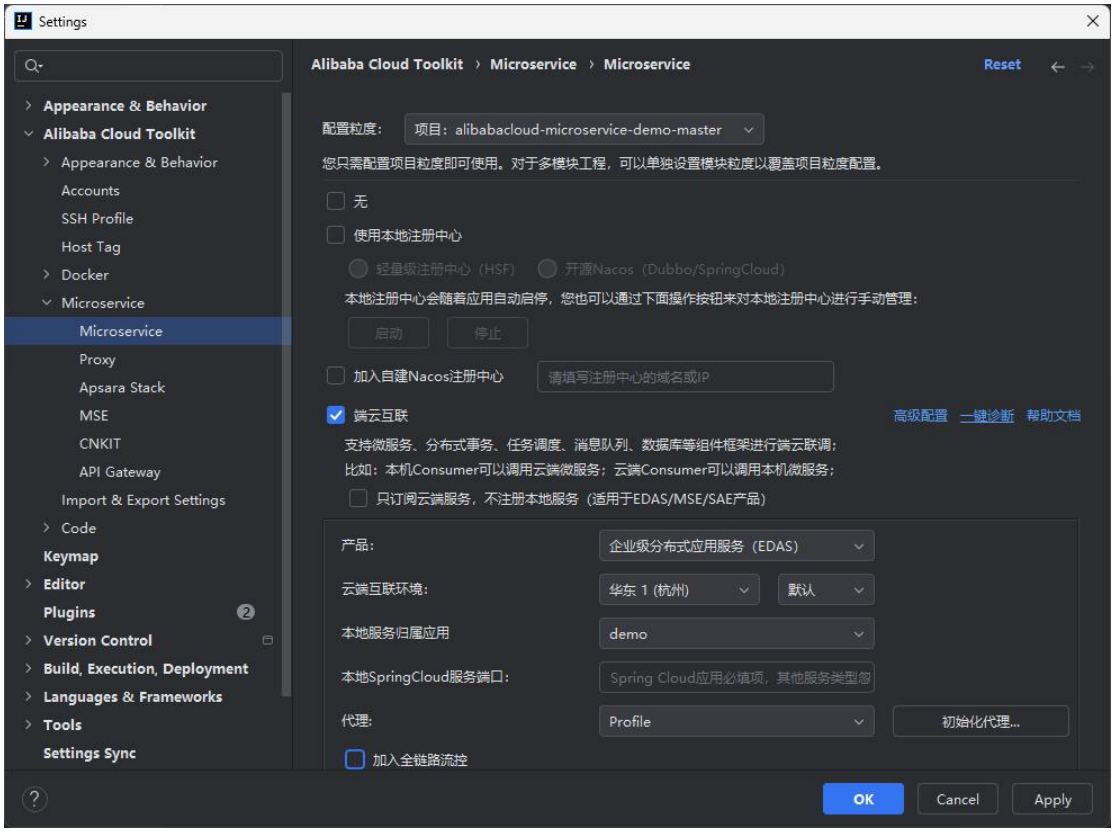
6.2.1 一键诊断



6.2.2 初始化代理

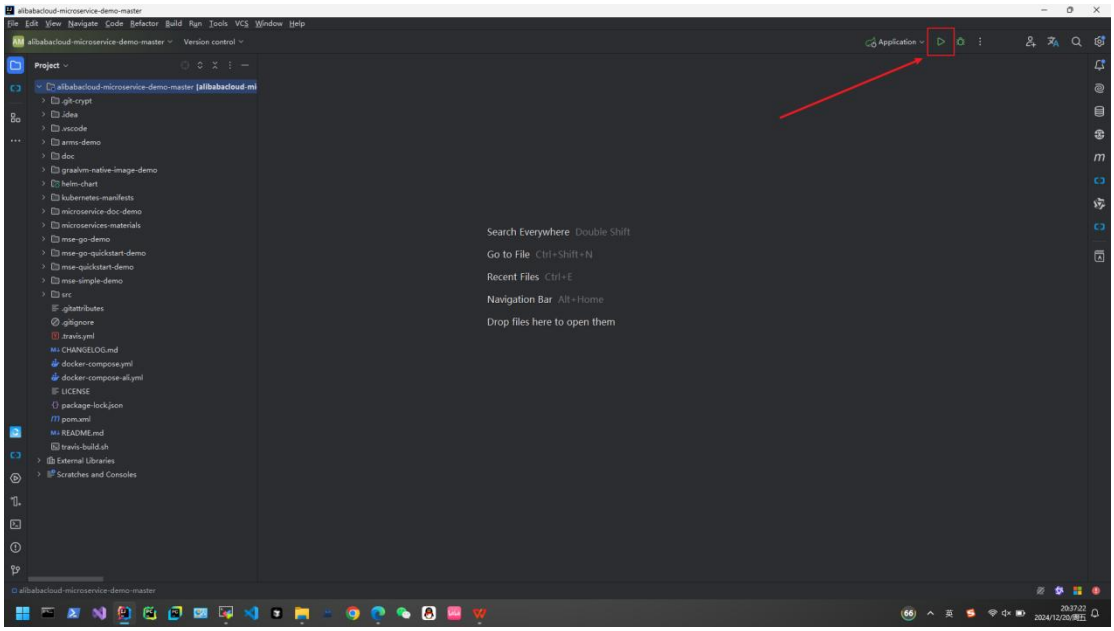


6.2.3 选择好云端互联环境与本地服务归属应用:



注意不要选择[加入全链路流控]的按钮 ， 否则还是失败

6.2.4 Apply -> OK -> 运行



6.2.5 成功启动端云互联