**OceanBase从入门到实战**

# 实战1： OceanBase Docker 体验

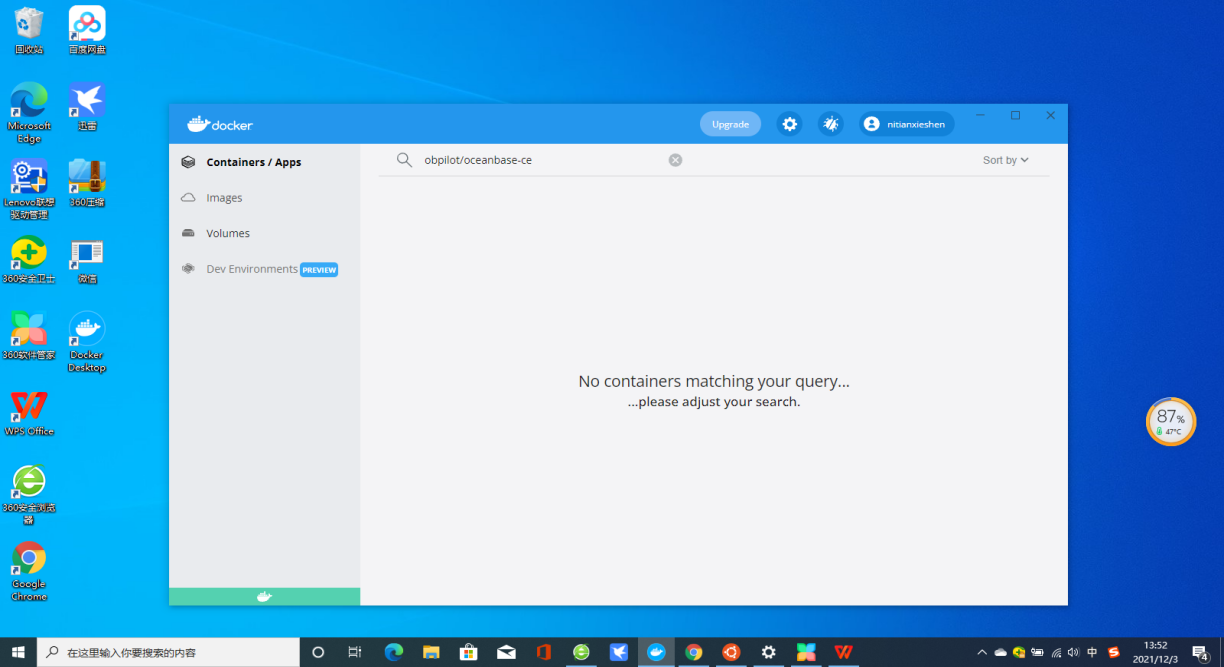
## 环境准备(Win10 8G内存 256G存储SSD)

1.1.1 安装linux的Windows子系统(Ubuntu 18.04 LTS)

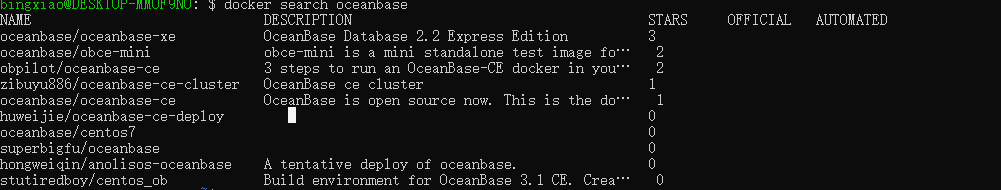
详见：https://docs.microsoft.com/zh-cn/windows/wsl/install-manual



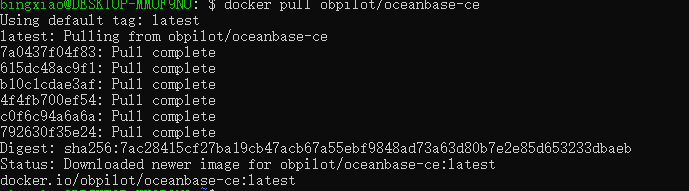
1.1.2 安装docker。



## 下载并运行Docker 镜像

1.2.1 搜索镜像资源 docker search oceanbase 

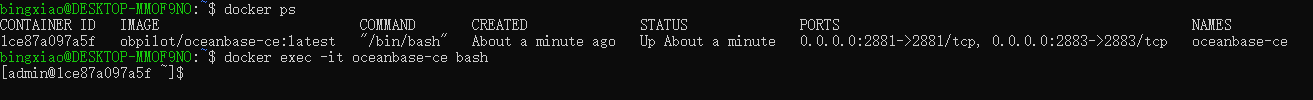
1.2.2 拉取ob镜像 docker pull obpilot/oceanbase-ce



1.2.3 运行镜像docker run -itd -m 10G -p 2881:2881 -p 2883:2883 --name oceanbase-ce obpilot/oceanbase-ce:latest

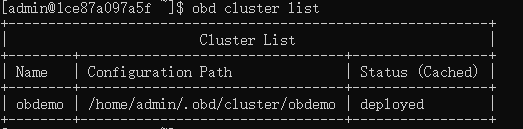


1.2.4 以bash方式进入容器中 docker exec -it oceanbase-ce bash

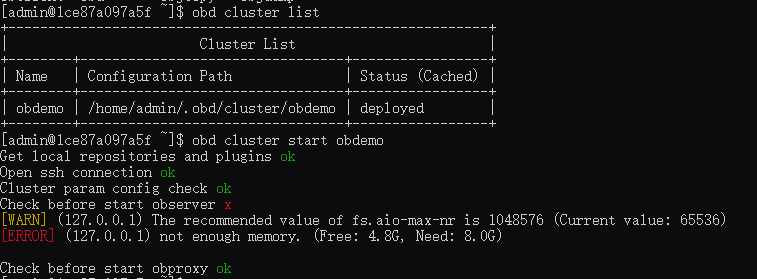


## OBD 命令部署OB集群

1.3.1 obd cluster list



1.3.2 obd cluster start obdemo

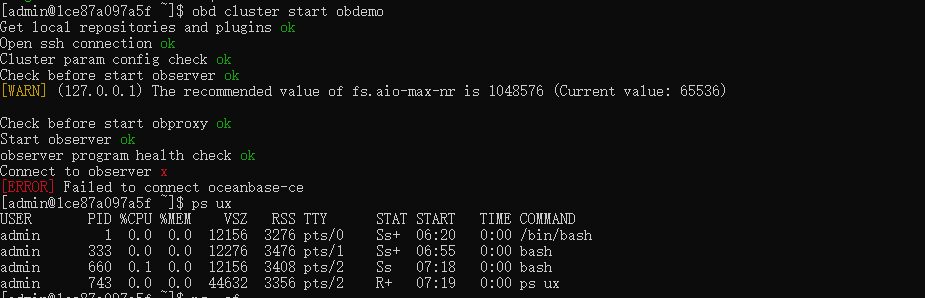


obd cluster edit-config obdemo

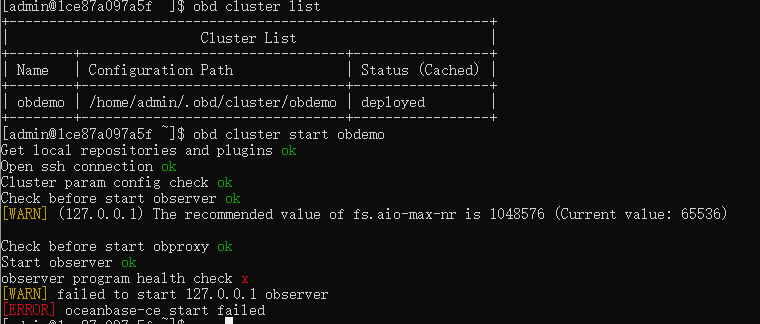
修改：



重启之后，安装ob，obproxy完成，但是连接不上ob数据库。



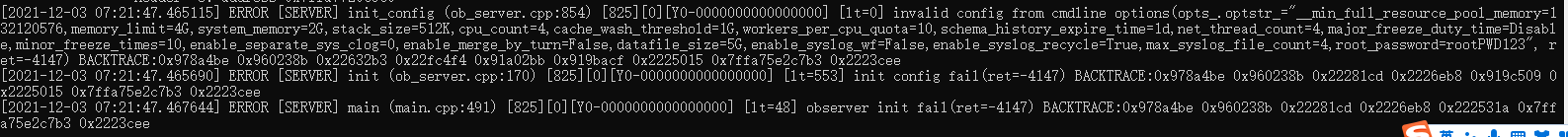
再次重启，health check报错。



尝试修改告警项 fs.aio-max-nr = 1048576，重启服务



报错依旧，health check报错（没有具体报错项），observer日志如下。



初步判断应该是当前win10笔记本硬件资源或系统变量设置存在问题，没有符合ob启动参数校验机制，虽然尝试调整了obd cluster edit-config的config.yaml文件中的cpu,内存，存储，工作线程等参数变量，但依然报错。同时obd的observer program health 信息较少，没有输出详细的报错信息，无法继续排查。暂时，只能换成服务器资源进行后续练习。

## 创建业务租户、数据库、表

OceanBase 数据库是多租户的分布式数据库，租户（tenant）使用的资源建立在资源池(resource pool)上。资源池包含了资源单元(resource unit)，而资源单元规定了具体资源的量化（如 CPU、Memory、 IOPS 、session数量、Disk\_Size）。创建租户前，必须规定租户使用的资源范围，资源池和资源单元就是为了满足租户资源隔离和负载均衡而存在的。租户的概念类似于传统数据库的数据库实例。租户下可以建立数据库，在租户的数据库下可以建立表。

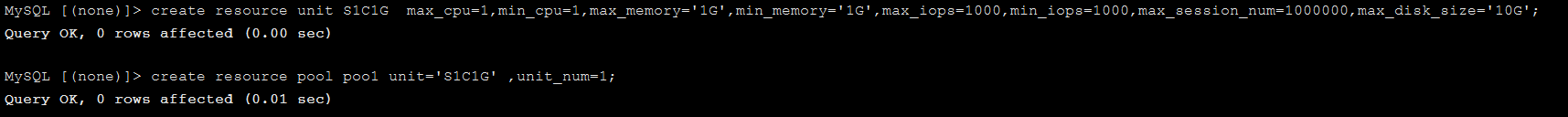
* + 1. 创建租户

A、创建资源单元1C1G

create resource unit S1C1G max\_cpu=1,min\_cpu=1,max\_memory='1G',min\_memory='1G',max\_iops=1000,min\_iops=1000,max\_session\_num=1000000,max\_disk\_size='10G';

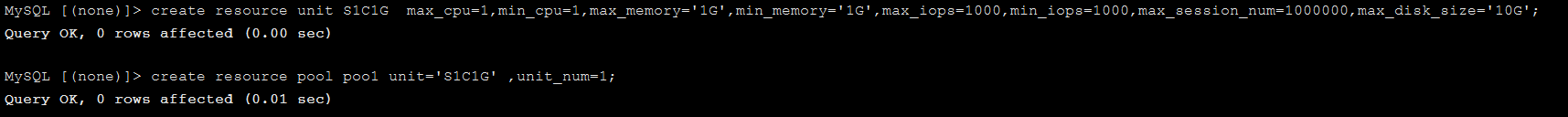
B、创建资源池

create resource pool poo1 unit='S1C1G' ,unit\_num=1;



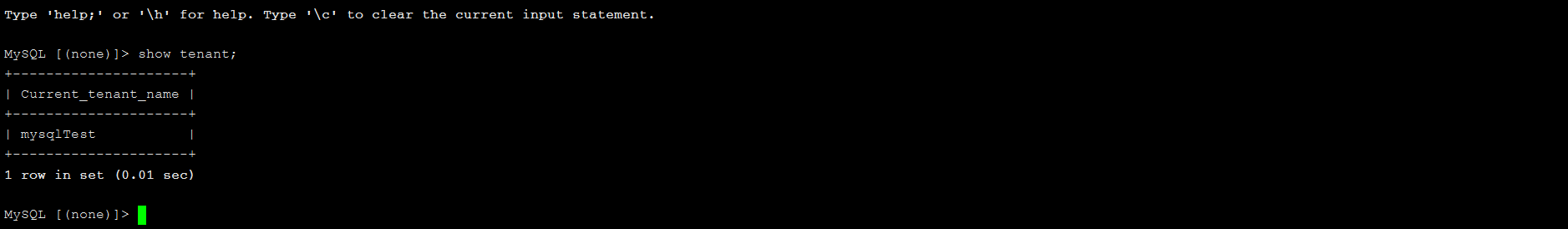
C、创建租户

create tenant mysqlTest resource\_pool\_list=('poo1'),primary\_zone='RANDOM',charset='utf8' set ob\_tcp\_invited\_nodes='%',ob\_compatibility\_mode='mysql';



1.4.2 创建数据库、表

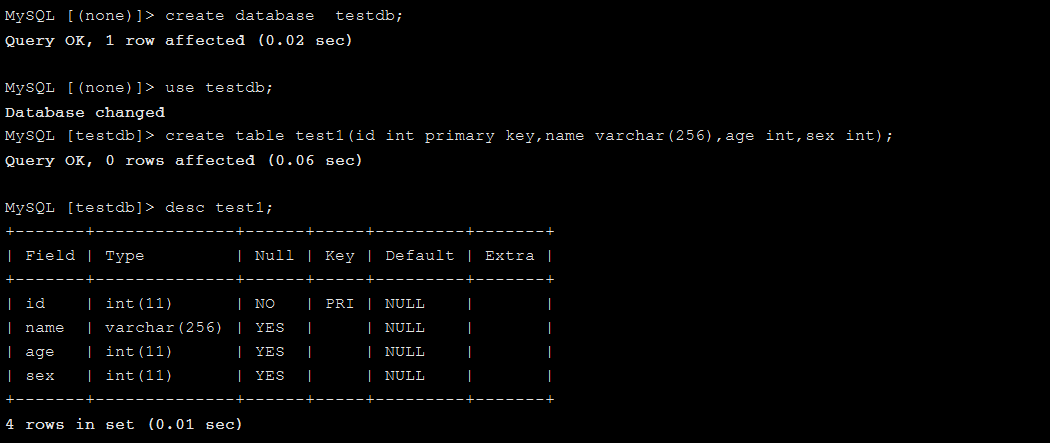
mysql -h xxxx -P 2883 -uroot@mysqlTest



create database testdb;

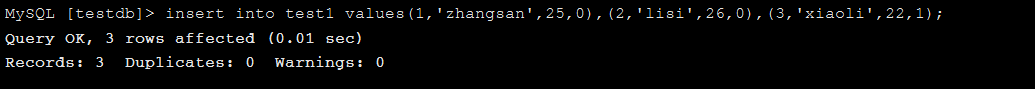
use testdb;

create table test1(id int primary key,name varchar(256),age int,sex int);

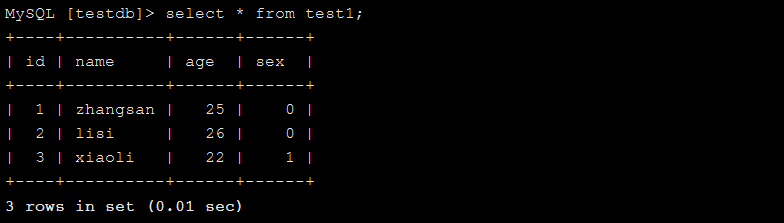


desc test1;

insert into test1 values(1,'zhangsan',25,0),(2,'lisi',26,0),(3,'xiaoli',22,1);



select \* from test1;



## 总结

本次主要体验的是使用docker镜像通过obd快速部署ob数据库，虽然镜像屏蔽了系统软件参数对ob的影响，但受限于硬件资源无法达到obd默认配置要求，在win10笔记本上启动失败，异常日志信息暴露较少，对于排错来说不是很方便。更换成服务器之后，启动配置还是挺顺利的，但内网连接不上外网，文件传输也挺麻烦的。