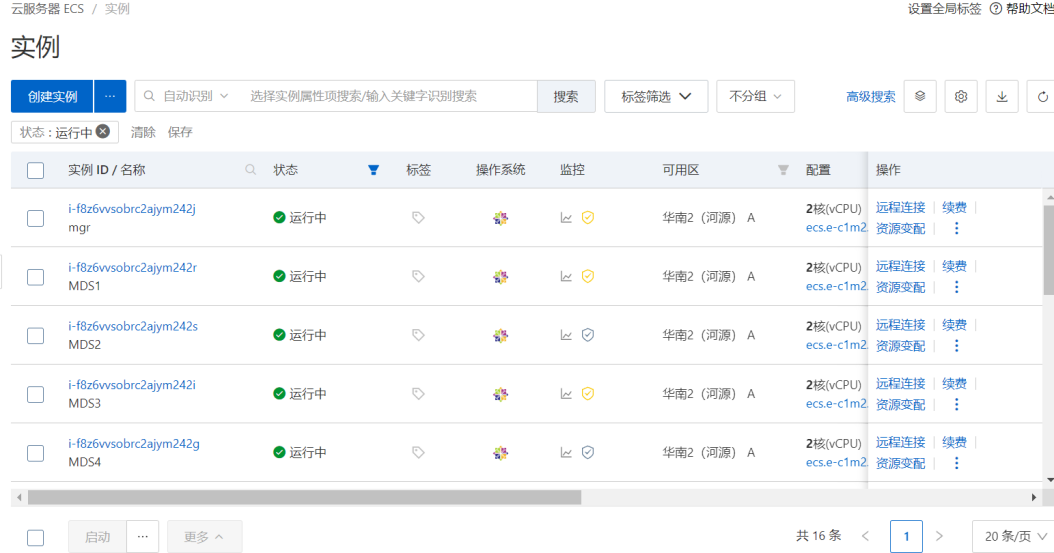
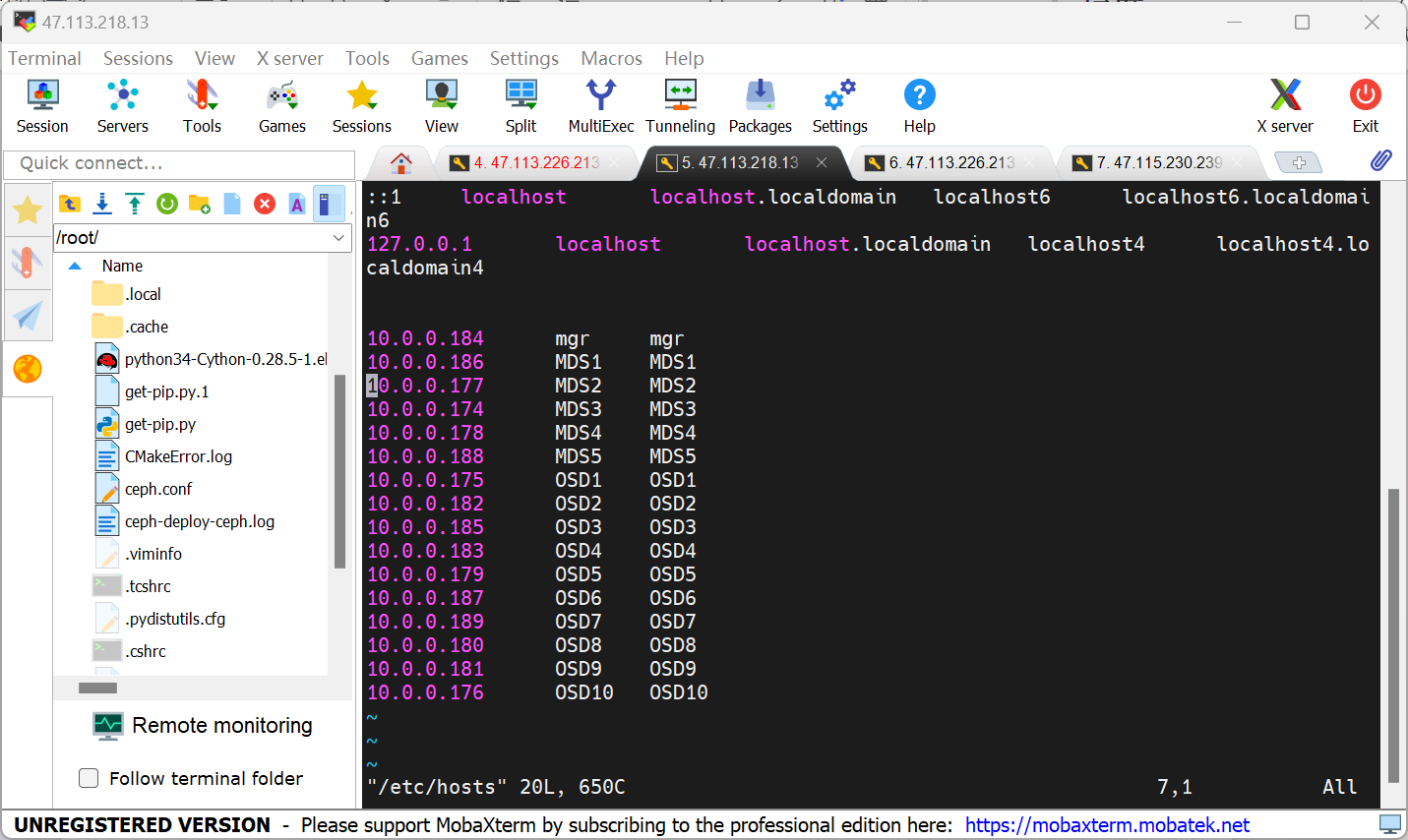
首先在阿里云购买16个ecs.c7.2xlarge的ECS云服务器，花费400多元。其中其中 1 个节点安装 CephFS，5 个节点作为 MDS 存储元数据，9 个 节点作为 OSD 存储实验数据集。最终购买结果如下，因使用了学生优惠，最终数额与图片有稍许差异：



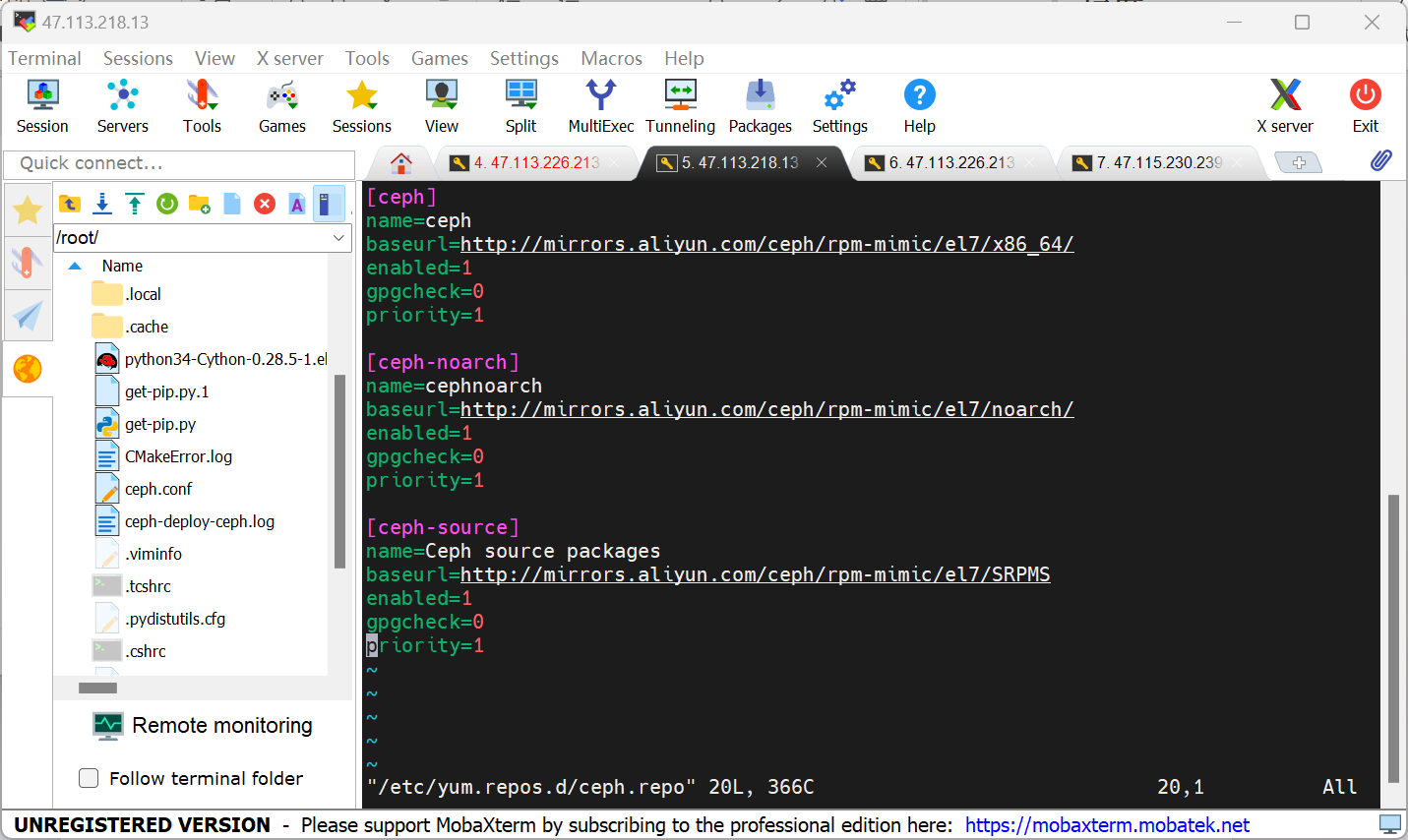
我们采用了华南2河源地区的服务器来降低成本，并采用包年包月的形式，购买了一周的短期服务；并为每个服务器搭载了150GB的ESSD云盘，并创建了专有网络VPC，网段10.0.0.0/24。购买完毕后为其更改实例名称与主机名，方便后续为其配置ceph。

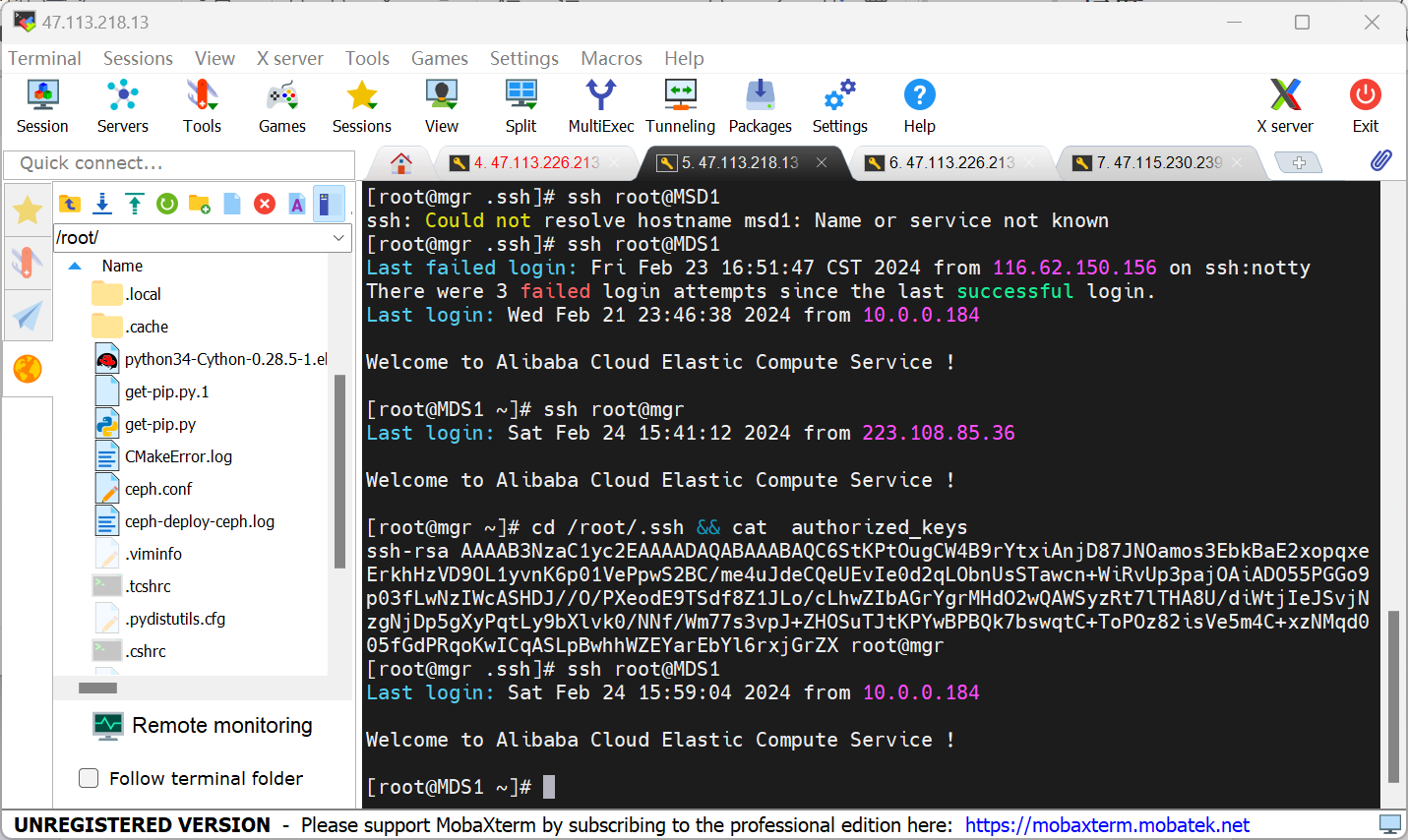


在配置ceph前，需要更改manager主机的hosts。需要将其的专有网络ip与localhost对应。如10.0.0.184与mgr对应，10.0.0.186与msd1对应等。

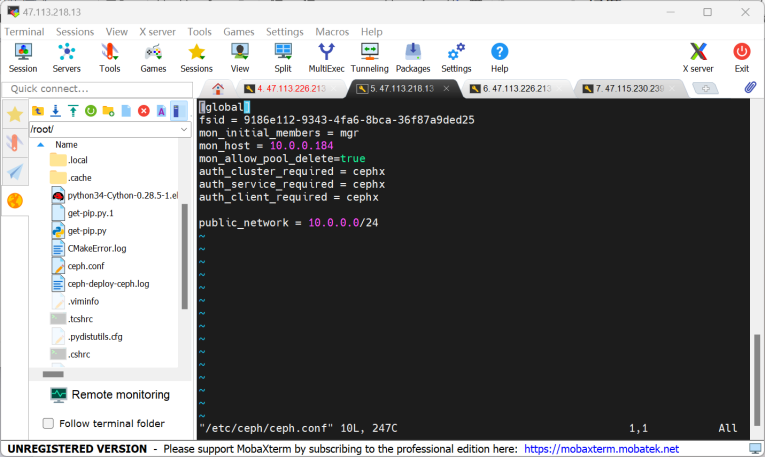


随后关闭防火墙，安装配置epl源与ceph源，加快下载速度。并为manager配置免登录，方便从manager主机到其他服务器的切换。

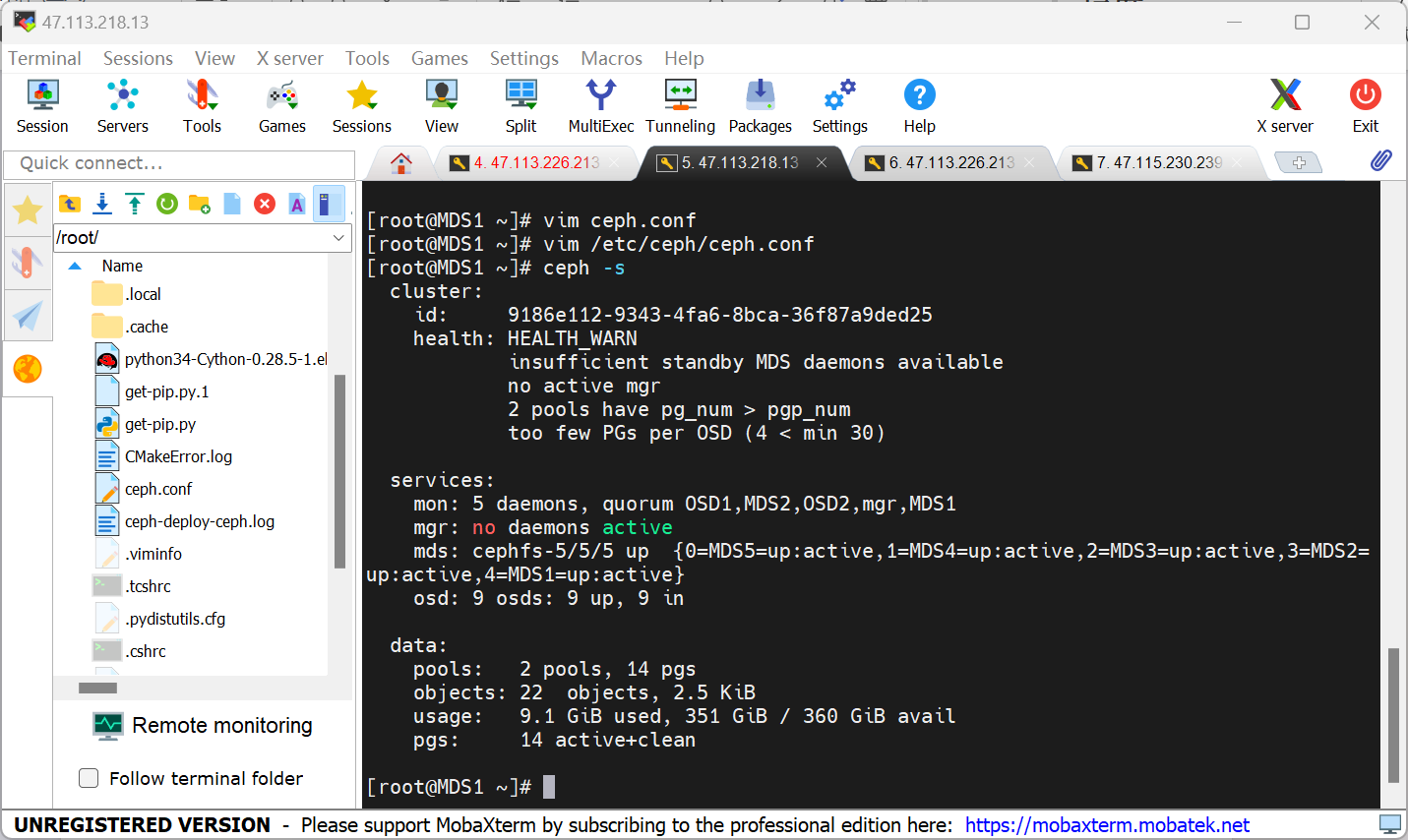




随后部署安装工具ceph-deloy，创建集群，为集群配置公共网络与mon监视节点。



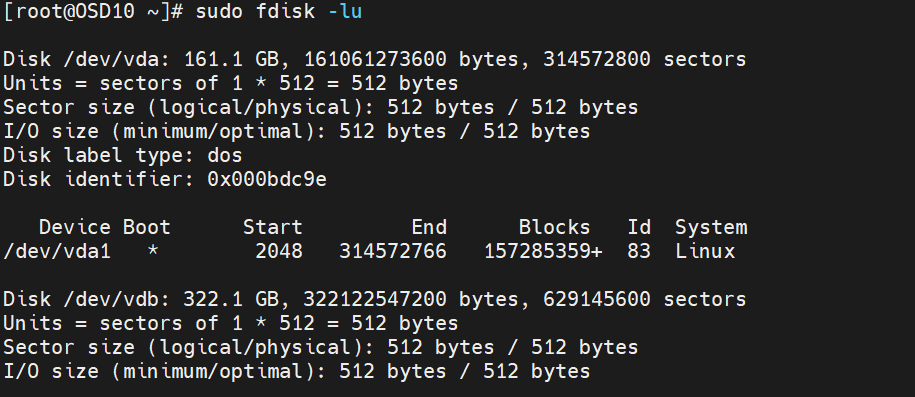
安装后使用ceph -s查看集群状态。



随后需要创建OSD节点，osd 必须是磁盘设备、分区、逻辑卷lv 不能是普通目录，普通目录创建的时候会报错,好像lv挂载的目录也可以，但是这引入一个思考。如果是lv挂载的目录，那么当该lv扩容之后，osd并不会扩容，所以这样做的目的是什么，还不如直接是磁盘或分区，osd扩容直接扩一个磁盘或分区即可。

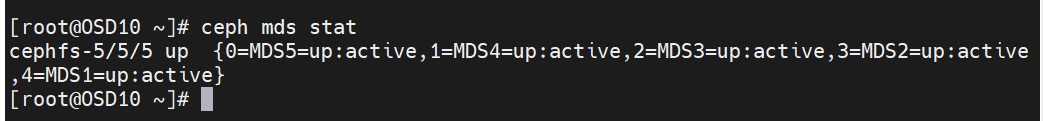
为此我们选择购买额外云盘，在阿里云实例控制台中，选择实例进行存储配置，进行创建云盘。在创建云盘后需要对其进行格式化。使用sudo fdisk -lu

，获取数据盘的设备名称，vdb即是我们新加的额外云盘。随后我们为其创建分区，创建文件系统，配置开机自动挂载分区即可使用。

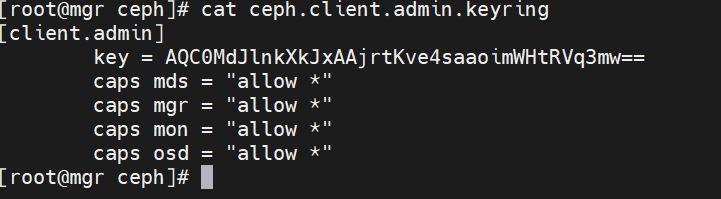


在配置完OSD后，需要为其配置MDS。在 linux 系统使用 ls 等操作查看某个目录下的文件的时候， 会有保存在磁盘上的分区表记录文件的名称、 创建日期、 大小、 inode 及存储位置等元数据信息， 在 cephfs 由于数据是被打散为若干个离散的 object 进行分布式存储， 因此并没有统一保存文件的元数据， 而且将文件的元数据保存到一个单独的存储出 matedata pool， 但是客户端并不能直接访问matedata pool 中的元数据信息， 而是在读写数的时候有 MDS(matadata server)进行处理，读数据的时候有 MDS从 matedata pool加载元数据然后缓存在内存(用于后期快速响应其它客户端的请求)并返回给客户端， 写数据的时候有 MDS 缓存在内存并同步到 matedata pool。

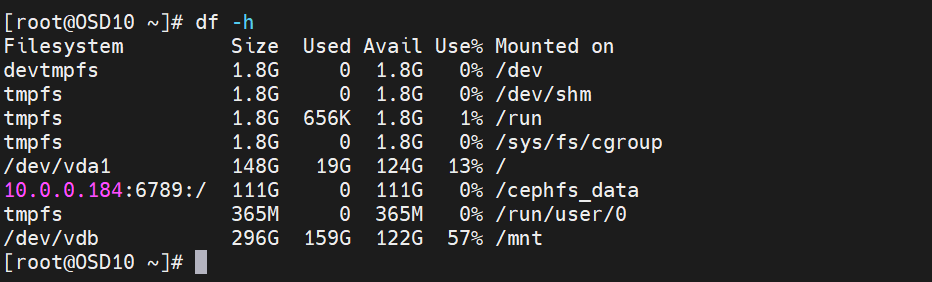
在配置完后，使用ceph mds stat查看。



最终配置client，使用cat ceph.client.admin.keyring查看客户机配置，并依据情况修改相应的key。



在完成所有的ceph配置后，可以使用df-h查看整体情况。



安装lunule

Linux 平台上安装

Git 的工作需要调用 curl，zlib，openssl，expat，libiconv 等库的代码，所以需要先安装这些依赖工具。

在有 yum 的系统上（比如 Fedora）或者有 apt-get 的系统上（比如 Debian 体系），可以用下面的命令安装：

各 Linux 系统可以使用其安装包管理工具（apt-get、yum 等）进行安装：

$ yum install curl-devel expat-devel gettext-devel \

openssl-devel zlib-devel

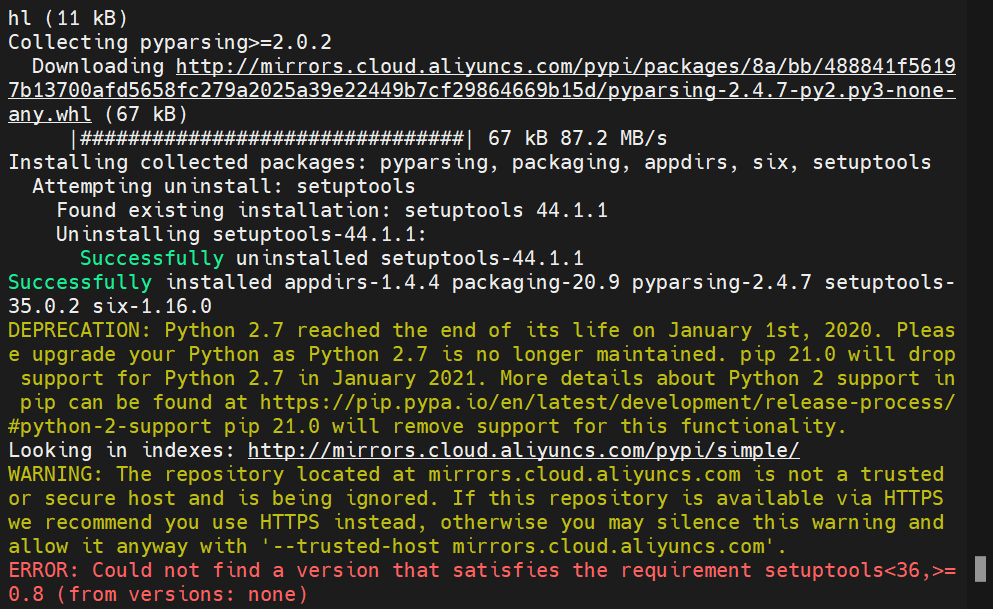
$ yum -y install git-core

$ git --version

git version 1.7.1

随后使用git clone https://github.com/mdbal-lunule/lunule.git 进行下载。

随后需要对项目进行安装依赖，/install-deps.sh 后，会出现这样的错误，因为项目使用的python为2，而我们目前常用的依赖常常依靠python3运行，故而需要我们满足安装工具的条件。



在centos编译时报错：error: bzlib.h: No such file or directory

yum search bzip2

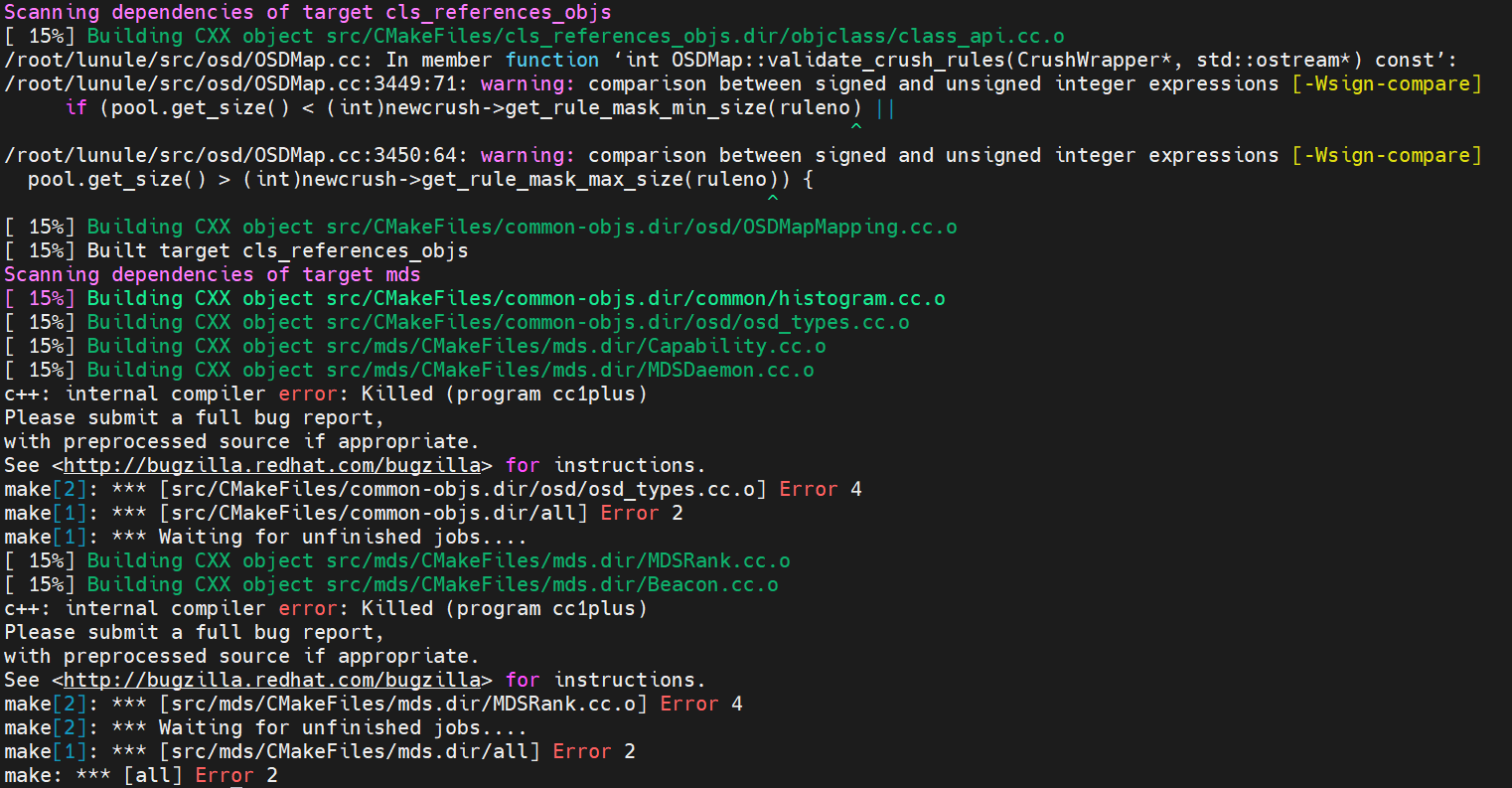
yum install bzip2-devel.x86\_64

解决c++: internal compiler error: 已杀死 (program cc1plus)，这个报错意味着服务器的内存不足，需要使用交换内存来进行解决。

sudo dd if=/dev/zero of=/swapfile bs=1M count=2048 # 1 \* 2048 = 2048 创建 1 g 的内存分区

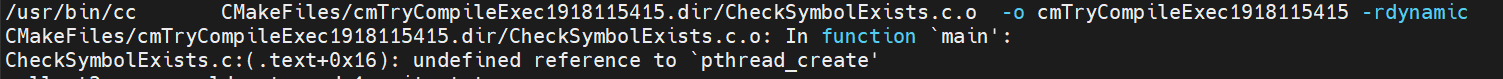
sudo mkswap /swapfile

sudo swapon /swapfile



undefined reference to `pthread\_create'

由于 pthread 库不是 Linux 系统默认的库，连接时需要使用静态库 libpthread.a，所以在使用 pthread\_create () 创建线程，以及调用 pthread\_atfork () 函数建立 fork 处理程序时，在编译中要加 -lpthread 参数。



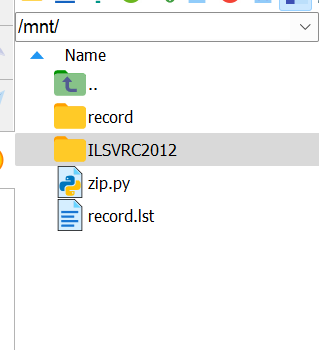
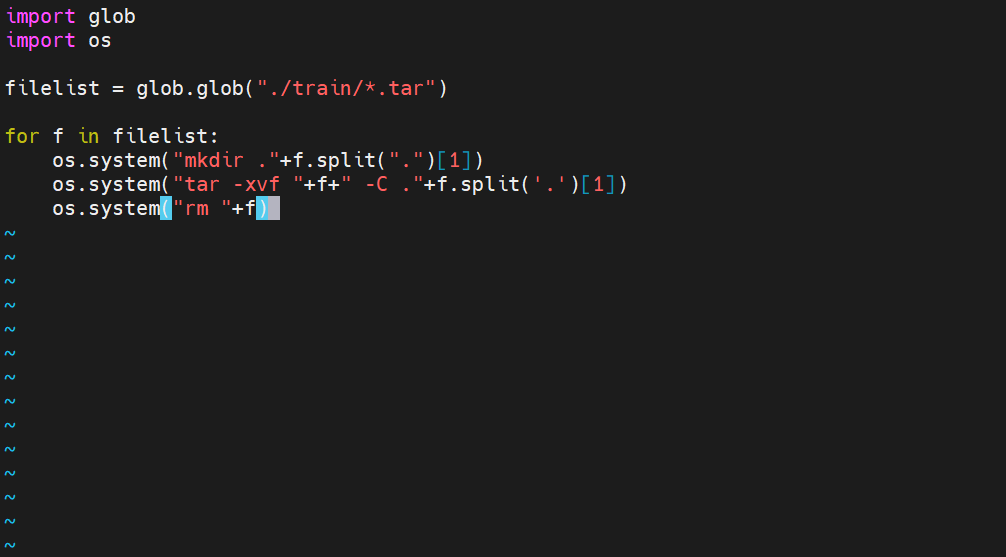
例如：在加了头文件 #include <pthread.h> 之后执行 pthread.c 文件，需要使用如下命令： gcc thread.c -o thread -lpthread

这种情况类似于 <math.h> 的使用，需在编译时加 -m 参数。

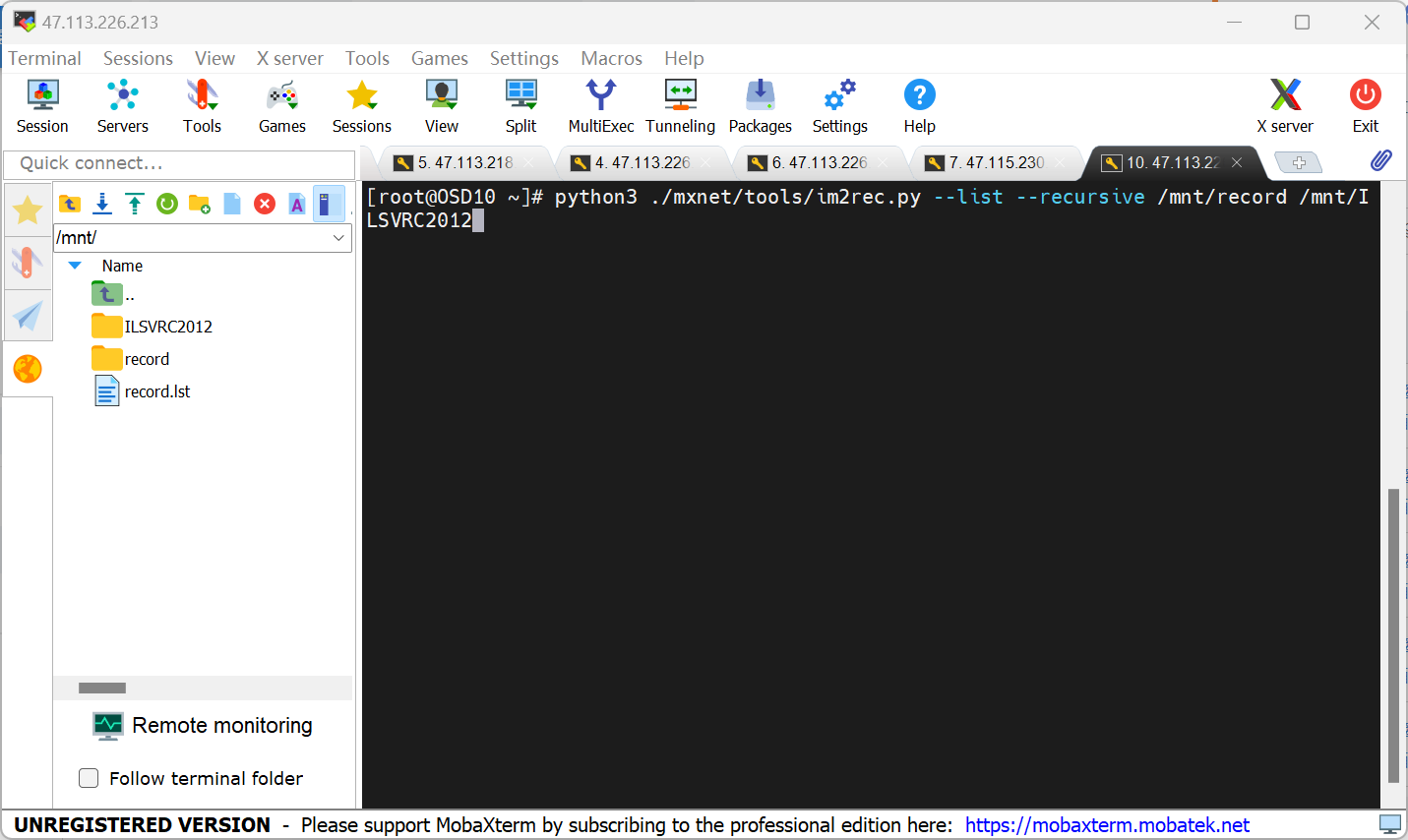
在运行CNN项目时，因为imagenet庞杂的数据集，所以我们在客户机上额外增加了300GB的云盘，在进行格式化后，因为数据集已经下载在物理硬盘中。使用临时升级带宽的方法增加传输文件的速率。



随后编写zip.py的python代码，以此来批量解压数据集压缩包。



最终结果



存在的问题1、opencv-python无法安装

原因：当前opencv版本不支持python3.6

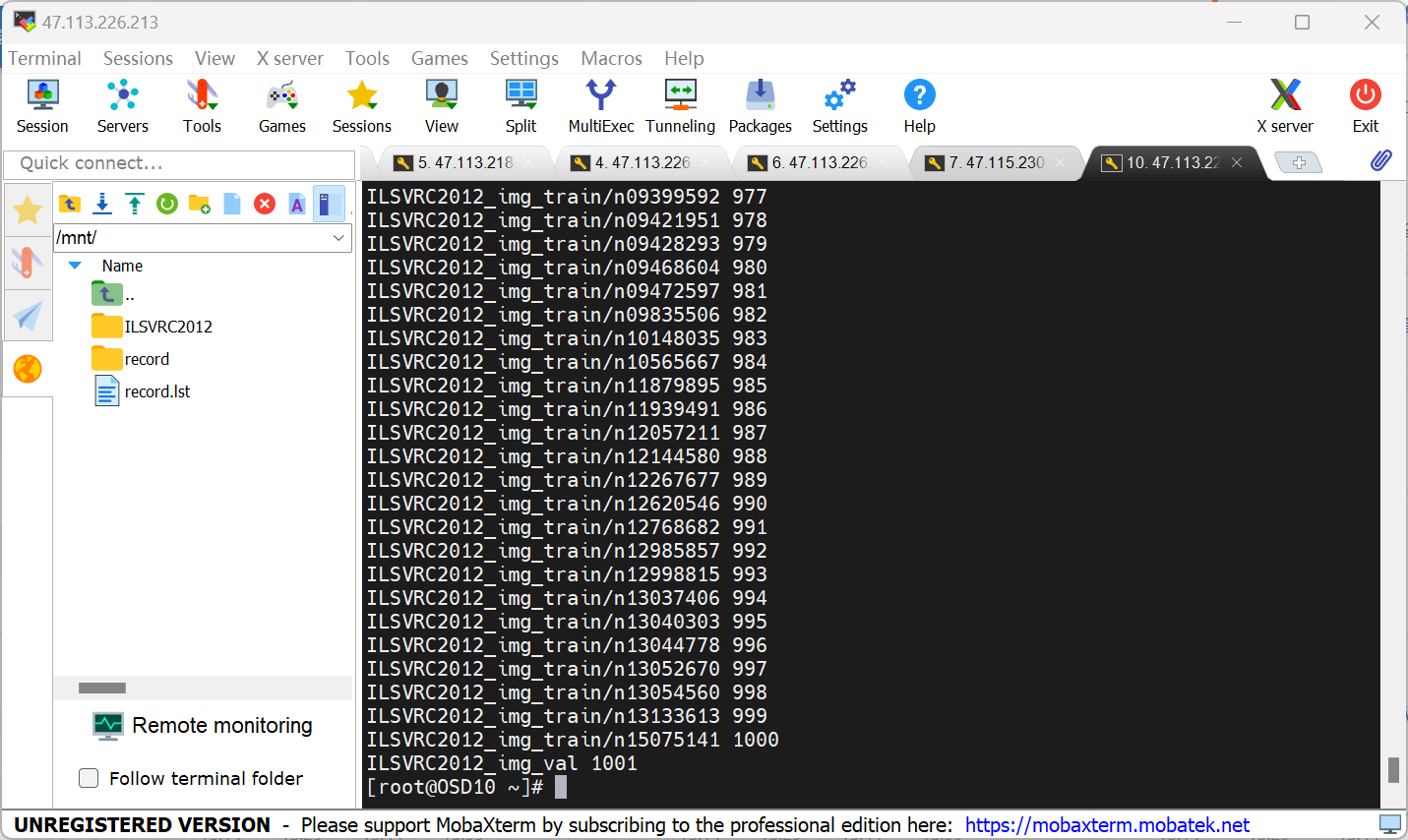
解决方法：opencv-python==4.3.0.38

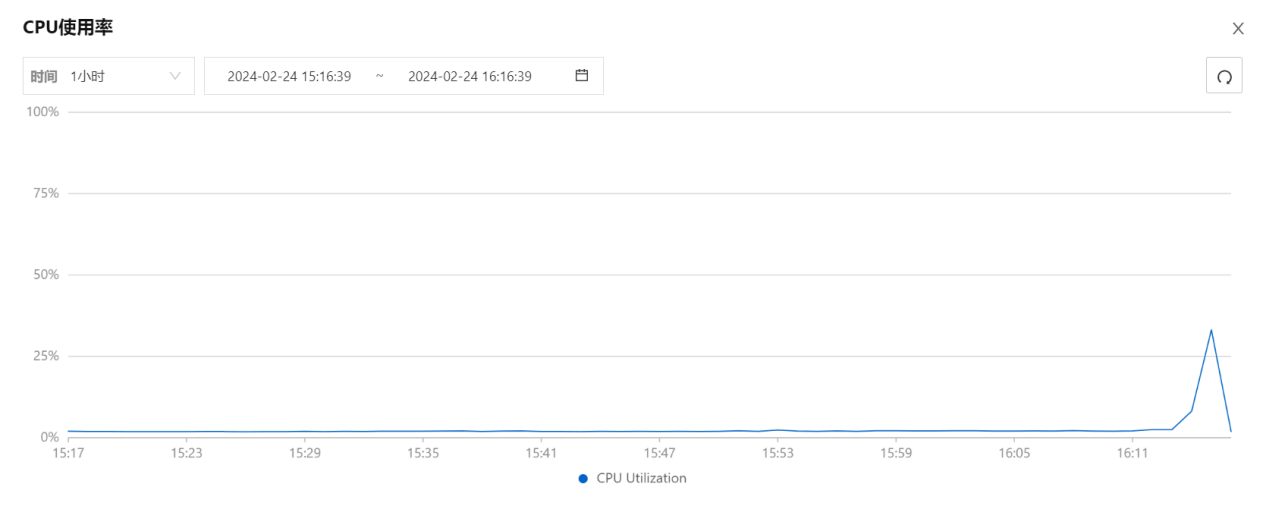
2、ImportError: libGL.so.1: cannot open shared object file: No such file or directory

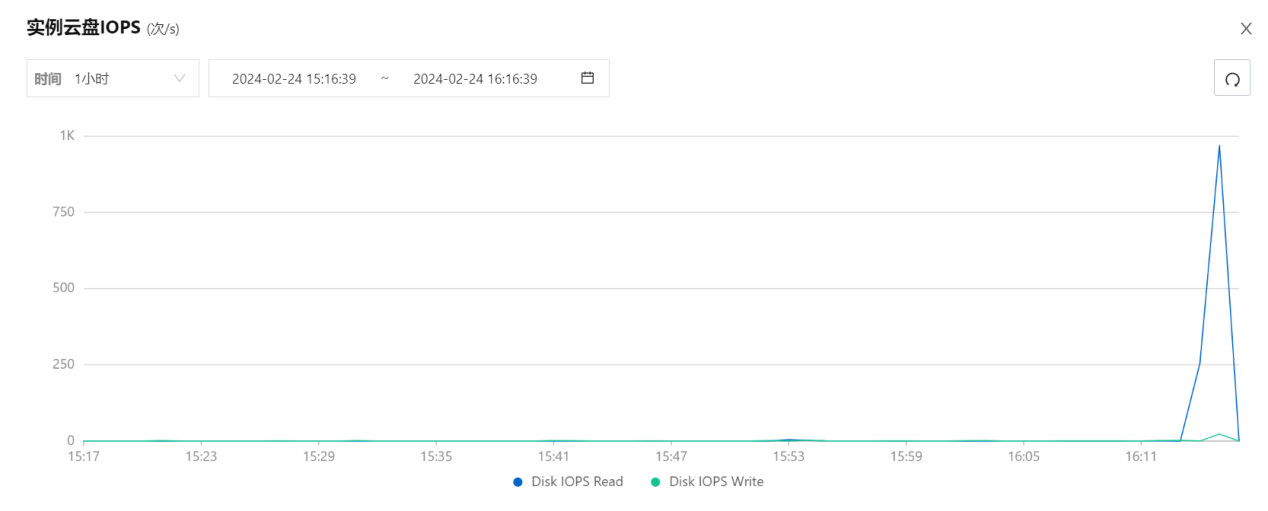
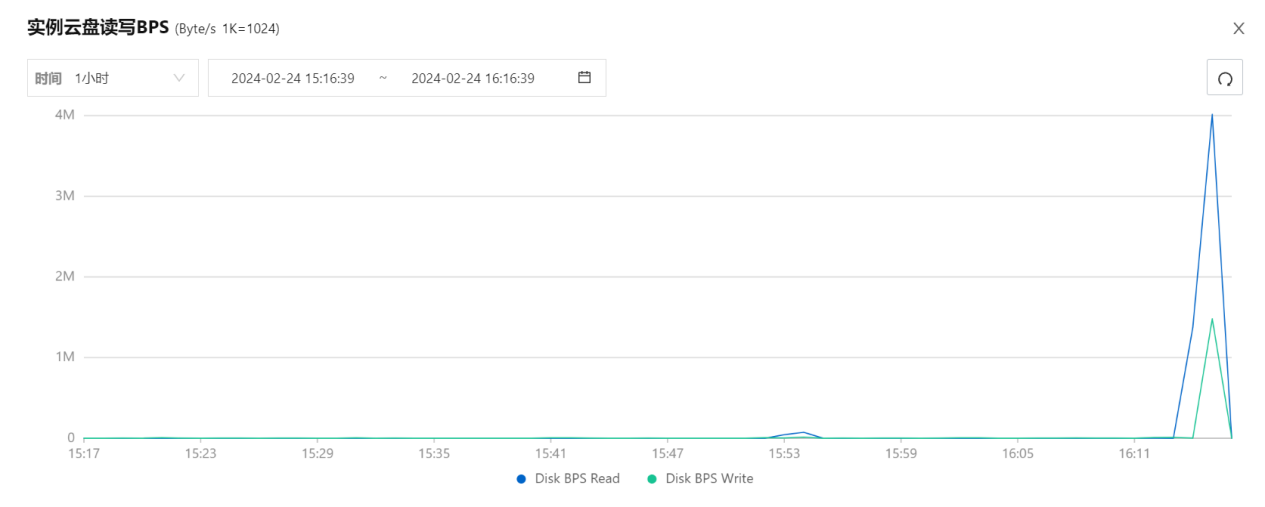
原因：缺少opencv-python依赖项libGL.so.1

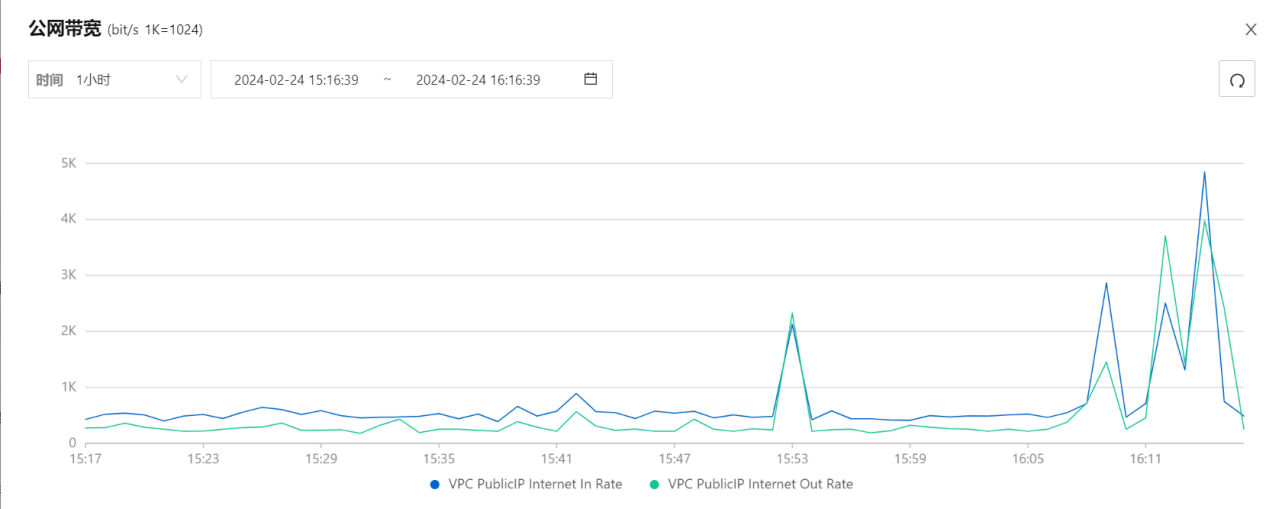
解决方法：1/ yum install libGL.so.1

2/ 安装无依赖opencv即opencv-headless











WSL本地结果

