文档中心

GPU云主机 UHost

产品文档

目录

动态与公告

本章节主要介绍GPU产品功能和对应文档动态。

2023年4月

功能名称	功能描述	发布时间	发布地域	相关文档
AI绘画Web UI镜像上线	支持客户通过可视化页面进行绘画	2023-4-28	全部	文档链接

2022年12月

功能名称	功能描述	发布时间	发布地域	相关文档
抢占式GPU云主机上线	支持客户低折扣购买抢占式GPU云主机	2022-12-30	全部	文档链接

2022年11月

功能名称	功能描述	发布时间	发布地域	相关文档

AI绘画镜像优化	支持Jupyter可视化编程界面以及“太乙”中文模型	2022-11-8	全部	文档链接

2022年10月

功能名称	功能描述	发布时间	发布地域	相关文档
镜像市场功能发布	发布一系列预装GPU驱动及CUDA的镜像	2022-10-17	全部	
AI绘画stable diffusion平台	发布预装stable diffusion模型的镜像	2022-10-17	全部	文档链接

概览

产品简介

GPU云主机是基于UCloud成熟的云计算技术，专享高性能GPU硬件的云主机服务。大幅提升图形图像处理和高性能计算能力，并具备弹性、低成本、易于使用等特性。有效提升图形处理、科学计算等领域的计算处理效率，降低IT成本投入。目前提供P40、V100、T4、T4S、V100S机型。

产品优势

GPU+SSD 优秀处理性能

Tesla K80计算卡拥有4992个CUDA核心，显存12G，可提供1.87 TFlops的双精度性能和5.6 TFlops的单精度性能;

Telsa P40计算卡拥有3840个CUDA核心，显存24G，可提供12 TFlops的单精度性能和47 TOPS的INT8性能;

Telsa V100计算卡拥有5120个CUDA核心，显存16G，可提供14TFlops的单精度性能和28 TFlops的FP16性能;

Telsa T4计算卡拥有2560个CUDA核心，显存16G，可提供8.1TFlops的单精度性能和65.13 TFlops的FP16性能;

Telsa V100S计算卡拥有5120个CUDA核心，显存32G,可提供16TFlops的单精度性能和32 TFlops的FP16性能；

UCloud GPU主机在科学计算表现中比传统架构性能提高数十倍。同时其搭配SSD固态磁盘，IO性能亦在普通磁盘的数十倍以上。

与UCloud云产品联通

UCloud GPU云主机和标准云主机采用一致的管理方式，包括内外网IP分配、防火墙、子网管理等，只需按现在的标准流程即可实现和标准云主机、路由器、LB、UDB等的业务对接，不增加额外的管理和运维成本。

灵活配置 便捷管理

UCloud GPU云主机与普通UHost配置方式并无差别，用户可在控制台和标准云主机一样创建、扩展和管理。另外，用户可以在云主机控制台自定义需要的GPU、CPU、内存、硬盘数量，让企业以更具性价比的方式享受GPU带来的高性能计算。

支持多种操作系统

UCloud GPU云主机支持多种操作系统，如CentOS、Ubuntu、Windows等，以适应不同行业的专业软件及建模需求。

机型与性能

K80（售罄）

GPU可选:1-2颗 Tesla K80。

CPU可选:4核/8核/16核

内存可选:8G/16G/32G/64G

数据盘可选:100-1000G SSD本地盘

P40

GPU可选:1-4颗 Tesla P40。

CPU可选:4核-32核

内存可选:8G-128G

数据盘可选:20-1000G SSD本地盘 or SSD云盘(不限容量)

可选可用区:上海二可用区B,华北一B

V100（售罄）

GPU可选:1-4颗 Tesla V100

CPU可选:4核-32核

内存可选:8G-128G

数据盘可选:20-1000G SSD本地盘 or SSD云盘(不限容量)

已下线

T4

GPU可选:1-4颗 Tesla T4

CPU可选:4核-32核

内存可选:8G-128G

数据盘可选:20-1000G SSD本地盘 or SSD云盘(不限容量)

可选可用区:华北一可用区E

T4S

CPU可选:4核-92核

内存可选:8G-320G

数据盘可选:SSD云盘(不限容量)

可选可用区:华北一可用区B,广州B,华北一E

V100S

CPU可选:4核-92核

内存可选:8G-576G

数据盘可选:SSD云盘(不限容量)

可选可用区:华北一可用区B,广州B,上海二A

A100

CPU可选:16核-124核

内存可选:64G-1024G

数据盘可选:SSD云盘(不限容量)

可选可用区:华北二可用区A

性能参数

参数	Tesla K80	Tesla P40	Telsa V100	Tesla T4/T4S	Tesla V100S	Tesla A100
CUDA核心数	2496	3840	5120	2560	5120	6912
单精度浮点性能	8.7 TFLOPS	12 TFLOPS	14 TFOPS	8.1 TFLOPS	16.35 TFLOPS	19.49 TFLOPS
INT8性能	N/A	47 TOPS	N/A	130 TOPS	N/A	N/A
Tensor性能	N/A	N/A	112 TFLOPS	N/A	130 TFLOPS	312 TFLOPS
显存容量	12GB	24GB	16GB	16GB	32GB	80GB
功耗	300W	250W	250W	70W	300W	300W
架构	Kepler	Pascal	Volta	Turing	Volta	Ampere

抢占式GPU云主机

产品简介

抢占式GPU云主机相较于按时付费的GPU云主机具有超低的折扣，旨在为客户在部分场景下提供更加便捷低成本的GPU算力资源。

抢占式GPU云主机的列表价会跟随市场价格和库存余量的情况，发生变动。如后台资源不足，或者客户的出价低于当前列表价，则会邮件通知客户，将在**30min**之后回收资源。

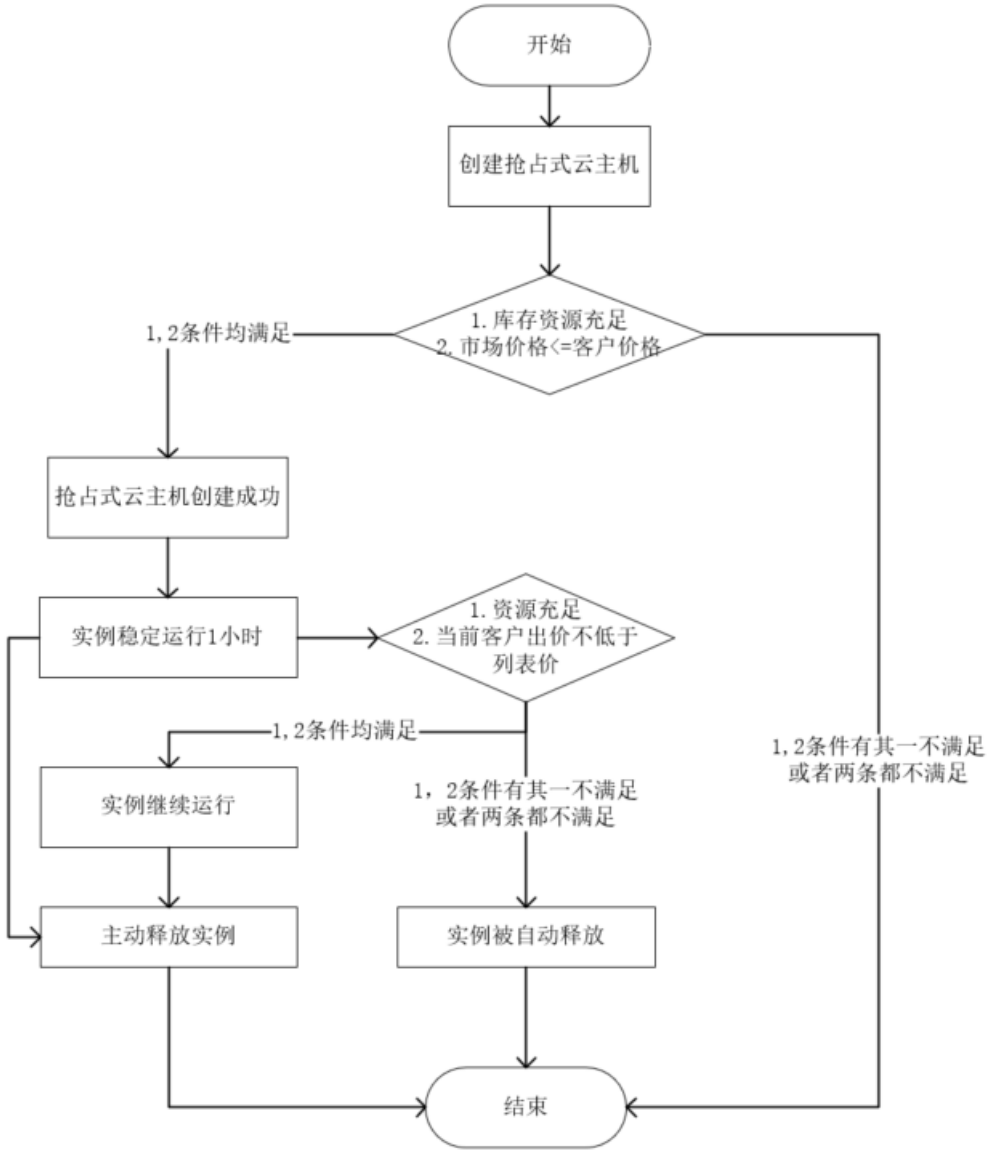
适用场景

抢占式GPU云主机适用于无状态、支持断点续算的应用场景，例如图像渲染、高性能并行计算、大数据业务等。应用程序的分布度、可扩展性和容错性越高，越适合使用抢占式GPU云主机节省资源成本，提升吞吐量。

抢占式GPU云主机上不宜运行对稳定性要求较高的服务，因为系统资源不足或者竞价失败等因素会导致抢占式云主机被释放，任务运行中断，可能会影响正常业务运行。

生命周期

抢占式GPU云主机生命周期如下图所示：



?> 抢占式GPU云主机创建之后可以平稳运行一个小时，在这一小时之内不受系统资源不足或者价格调整的影响，均能平稳使用。

如后台资源不充裕或市场价格大于客户的出价，系统将会邮件通知客户，抢占式GPU云主机将在**30min**之后被释放，请提前做好数据备份。

您可以在创建抢占式GPU云主机时，同步开启快照服务。如仍需继续使用，可以通过快照恢复GPU云主机。

计费说明

目前仅云主机支持抢占式，即CPU、内存和GPU卡可以享受超低折扣，云主机关联的EIP、磁盘仍按照原价进行售卖；

如抢占式GPU云主机资源欠费，会立即进入回收列表，邮件通知客户立即续费，5min之后如还未正常续费，则资源将被关机回收；

抢占式GPU云主机不支持做计费方式变更，不支持改配升级；

抢占式GPU云主机不受客户销售折扣影响，不可享受叠加折扣优惠。

支持机型配置&地域

地域	机型&卡型	配置
华北二A	GPU型 高性价比显卡5	16核32G GPU* 1
华北二A	GPU型 高性价比显卡5	16核64G GPU* 1
华北二A	GPU型 高性价比显卡5	32核64G GPU* 2
华北二A	GPU型 高性价比显卡5	32核128G GPU* 2
华北一B/上海二A	GPU型 V100S	10核32G GPU* 1
华北一B/上海二A	GPU型 V100S	10核64G GPU* 1

华北一B/上海二A	GPU型 V100S	10核96G GPU* 1
华北一B/上海二A	GPU型 V100S	20核64G GPU* 2
华北一B/上海二A	GPU型 V100S	20核96G GPU* 2
华北一B/上海二A	GPU型 V100S	20核128G GPU* 2
华北一B/上海二A	GPU型 V100S	20核160G GPU* 2
华北一B/上海二A	GPU型 V100S	20核160G GPU* 2
华北一B/上海二B	GPU型 P40	4核8G GPU* 1
华北一B/上海二B	GPU型 P40	4核16G GPU* 1
华北一B/上海二B	GPU型 P40	8核16G GPU* 1
华北一B/上海二B	GPU型 P40	8核32G GPU* 1
华北一B/上海二B	GPU型 P40	8核64G GPU* 1
华北一B/上海二B	GPU型 P40	8核16G GPU* 2
华北一B/上海二B	GPU型 P40	8核32G GPU* 2
华北一B/上海二B	GPU型 P40	16核32G GPU* 2
华北一B/上海二B	GPU型 P40	16核64G GPU* 2
华北一B	GPU型 T4S	4核8G GPU* 1
华北一B	GPU型 T4S	4核16G GPU* 1
华北一B	GPU型 T4S	4核32G GPU* 1

华北一B	GPU型 T4S	8核16G GPU* 1
华北一B	GPU型 T4S	8核32G GPU* 1
华北一B	GPU型 T4S	8核64G GPU* 1
华北一B	GPU型 T4S	8核16G GPU* 2
华北一B	GPU型 T4S	8核32G GPU* 2
华北一B	GPU型 T4S	8核64G GPU* 2
华北一B	GPU型 T4S	8核128G GPU* 2

NCCL性能调优

NVIDIA 集合通信库(NCCL) 可实现针对NVIDIA GPU 和网络进行性能优化的多GPU 和多节点通信基元。

GPU云主机内NCCL测试步骤

确认环境:需在8卡高性价比显卡6/高性价比显卡6pro/A800云主机下

确认依赖:nvidia驱动,cuda,gcc依赖:版本>=8,NCCL依赖包,topo文件

依赖不满足

若依赖满足,直接跳至本文NCCL-Test部分

这部分为基础环境搭建指导,以Ubuntu系统为例

基础环境搭建

```
## make
```

```
sudo apt update
sudo apt-get install make


## g++ 、gcc
sudo apt install build-essential
sudo apt install g++
sudo apt install gcc
```

NVIDIA驱动和CUDA安装

NVIDIA

1. 控制台创建云主机，选择想要安装驱动的基础镜像版本，创建虚机
2. 访问NVIDIA官网获取下载地址
    - 产品家族选择具体机型
    - CUDA ToolKit选择驱动支持的CUDA版本，没有选择则是默认版本
3. 选择完成后，点击搜索→下载，复制链接地址（下图以A100为例：）

NVIDIA 驱动程序下载

在下方的下拉列表中进行选择，针对您的 NVIDIA 产品确定合适的驱动。

产品类型:	Data Center / Tesla	产品类型
产品系列:	A-Series	产品系列
产品家族:	NVIDIA A100	具体产品
操作系统:	Linux 64-bit	系统
CUDA Toolkit:	Any	语言
语言:	Chinese (Simplified)	

搜索

点击搜索

Data Center Driver For Linux X64

版本: 550.90.07
发布日期: 2024.6.6
操作系统: CBL Mariner, Linux 64-bit
CUDA Toolkit: 12.4
语言: Chinese (Simplified)
文件大小: 293.33 MB

下载 点击按钮

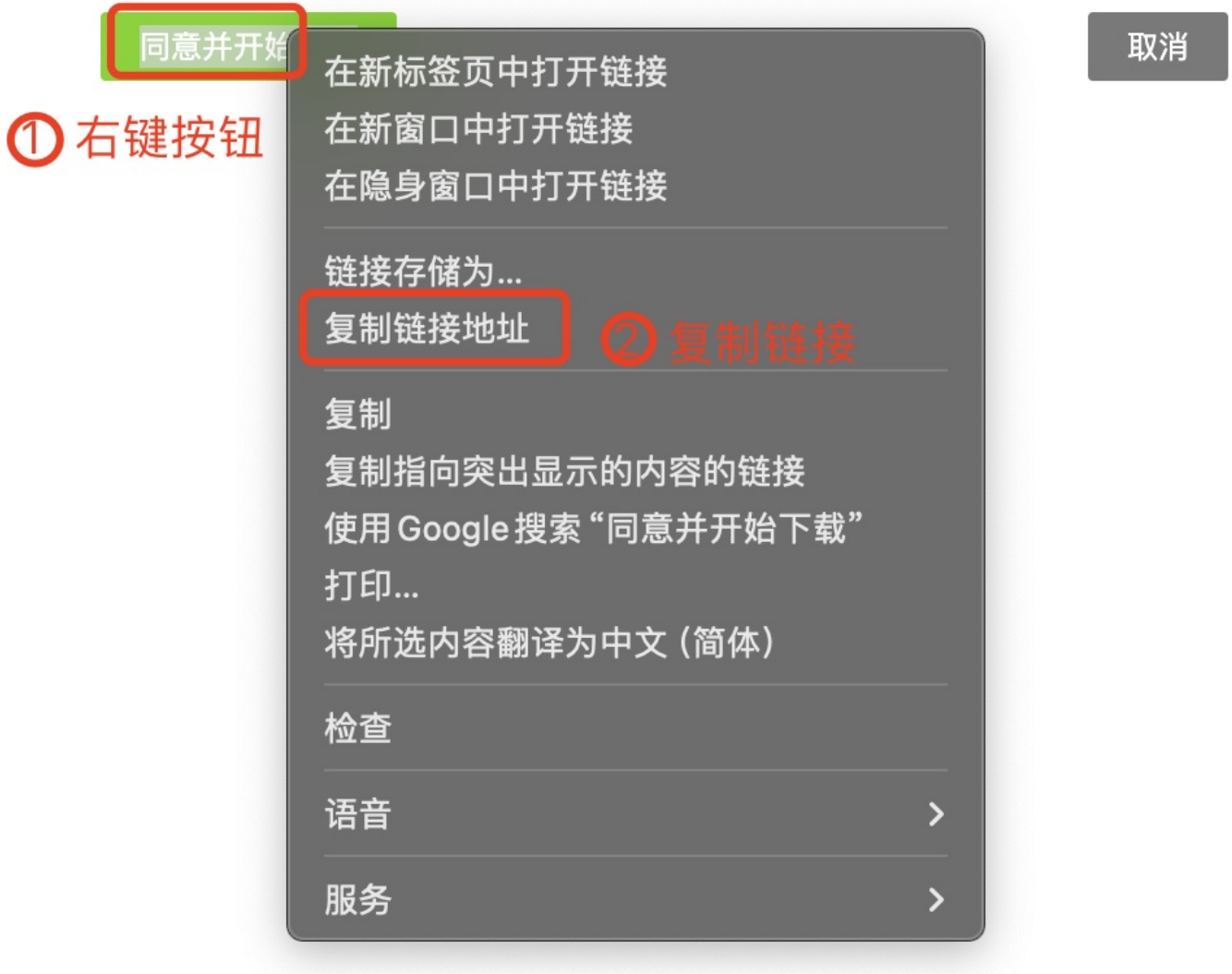
发布重点	产品支持列表	其他信息

Release notes, supported GPUs and other documentation can be found at:
https://docs.nvidia.com/datacenter/tesla/index.html

下载

单击"同意并开始下载"按钮，即表示您确认已阅读并同意 NVIDIA 软件用户使用许可，单击下方"同意并开始下载"按钮后

驱动程序将立即开始下载。NVIDIA 建议用户更新到最新的驱动版本。如需了解更多信息，请查看 NVIDIA 产品安全。



4. 进入GPU云主机
   - 执行命令下载驱动(wget后面是复制的链接地址):`wget {上一步复制的链接地址}`

- 检查gcc、make软件库是否安装，及安装gcc 和 make

```
## which make 检测make是否安装, 安装命令 # sudo apt-get install make
## gcc --version 检测gcc是否安装, 安装命令 # sudo apt-get install gcc
```

5. 开始安装：执行sh NVIDIA-xxxxxxx.run ，即开始安装驱动，注：遇到权限问题命令前添加sudo即可
6. 验证nvidia驱动：执行 nvidia-smi

CUDA

1. 官网下载CUDA
   选择对应系统和CUDA版本（执行 nvidia-smi 可查看驱动适配的最高cuda版本，小于等于Nvidia驱动的cuda版本）
2. wget下载到虚机本地，执行 sudo sh cuda_xxxxxxx_linux.run 进行安装
3. 配置环境变量，添加软链接

```
## 添加环境变量：
sudo vim /etc/profile //编辑文件，在末尾添加,保存退出:
export CUDA_HOME=/usr/local/cuda
export PATH=$PATH:/usr/local/cuda/bin
export LD_LIBRARY_PATH=/usr/local/cuda/lib64:$LD_LIBRARY_PATH
## 添加软链接：
sudo ln -s /usr/local/cuda-{cuda版本号} /usr/local/cuda（修改版本号即可，eg：cuda-10.1）
## 重启
sudo reboot
```

4. 验证cuda环境是否配置完成：nvcc -V

NCCL环境准备

```
## ubuntu 18.04
wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/cuda-keyring_1.0-1_all.deb
## ubuntu 20.04
wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2004/x86_64/cuda-keyring_1.0-1_all.deb
## ubuntu 22.04
wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64/cuda-keyring_1.0-1_all.deb
sudo dpkg -i cuda-keyring_1.0-1_all.deb
sudo apt update
## 下面软件包要和cuda版本匹配(具体版本可查询官方信息(举例ubuntu22.4):https://developer.download.nvidia.cn/compute/cuda/repos/ubuntu2204/x86_64/)
sudo apt install libnccl2=2.18.3-1+cuda12.2
sudo apt install libnccl-dev=2.18.3-1+cuda12.2
```

NCCL-Test

1. 下载NCCL-Test

```
## 下载并编译nccl-test
git clone https://github.com/NVIDIA/nccl-tests.git
cd nccl-tests

## 高性价比显卡6/6增强型
make CUDA_HOME=/usr/local/cuda -j
```

```
## A800
make MPI=1 MPI_HOME=/usr/mpi/gcc/openmpi-4.1.5a1 CUDA_HOME=/usr/local/cuda -j
```

2. 指定拓扑文件

```
## 高性价比显卡6/6增强型
cd nccl-tests/build
NCCL_MIN_NCHANNELS=32 NCCL_MAX_NCHANNELS=32 NCCL_NTHREADS=256 NCCL_BUFFSIZE=2097152 NCCL_P2P_DISABLE=1 ./all_reduce_perf -b 8 -e 8G -f 2 -g 8

## A800;注意:下面命令中有三处需要注意手动更改
## 1. NCCL_TOPO_FILE 对应path换成topo xml 文件的path
## 2. PATH 需要换成nccl-test对应的路径
## 3. numa -H 后面地址需要换成内网ip地址

mpirun --allow-run-as-root --oversubscribe -np 8 --bind-to numa -H {内网IP地址} -mca plm_rsh_args "-p 22 -q -o StrictHostKeyChecking=no" -mca coll_hcoll_enable 0 -mca pml ob1 -mca btl ^openib -mca btl_openib_if_include mlx5_0:1,mlx5_1:1,mlx5_2:1,mlx5_3:1 -mca btl_openib_cpc_include rdmacm -mca btl_openib_rroce_enable 1 -x NCCL_IB_DISABLE=0 -x NCCL_SOCKET_IFNAME=eth0 -x NCCL_IB_GID_INDEX=3 -x NCCL_IB_TC=184 -x NCCL_IB_TIMEOUT=23 -x NCCL_IB_RETRY_CNT=7 -x NCCL_IB_PCI_RELAXED_ORDERING=1 -x NCCL_IB_HCA=mlx5_0,mlx5_1,mlx5_2,mlx5_3 -x CUDA_VISIBLE_DEVICES=0,1,2,3,4,5,6,7 -x NCCL_TOPO_FILE={nccl_topo xml文件地址} -x NCCL_NET_GDR_LEVEL=1 -x CUDA_DEVICE_ORDER=PCI_BUS_ID -x NCCL_ALGO=Ring -x LD_LIBRARY_PATH -x PATH {对应nccl-test目录path} -b 8 -e 8G -f 2 -g 1

# 上述是单机测试命令,多机测试只需要 -H 后面添加测试的虚机eth0对应的IP即可(多IP之间英文逗号相隔)
# np后面的参数:测试几台虚机则是(需要测试虚机个数*8)
```

```
# -H 后面参数:虚机eth0对应的ip地址,多个之间英文都好隔开(ip addr 可查看,必须包含程序运行的虚机eth0的ip)
# -x NCCL_TOPO_FILE=指定nccl_topo.xml所在的绝对路径
# -x PATH 后面参数: 绝对路径下nccl-test下all_reduce_perf的路径
```

GPU云主机NCCL TOPO文件透传至容器

当您在8卡高性价比显卡6/高性价比显卡6pro/A800等类型的GPU云主机内自行使用docker时,可根据下列步骤完成topo透传:

1. 确认GPU云主机内是否存在该文件/var/run/nvidia-topologyd/virtualTopology.xml

- 若存在执行第2步
- 若不存在请咨询技术支持,将提供您该文件,拷贝文件保存至GPU node的 /var/run/nvidia-topologyd/virtualTopology.xml后执行第2步

2. 在GPU云主机内执行下列操作

```
docker run -it -e NVIDIA_VISIBLE_DEVICES=all -v /var/run/nvidia-topologyd/virtualTopology.xml:/var/run/nvidia-topologyd/virtualTopology.xml ubuntu /bin/bash
```

NCCL性能优化配置

```
NCCL_MIN_NCHANNELS=32 //nccl可以使用的最小通道数
NCCL_MAX_NCHANNELS=32 //nccl可以使用的最大通道数,增加通道数也会增加nccl使用的cuda块数,这可能有助于提高性能,2.5以上nccl版本最大值为32
NCCL_NTHREADS=256 //设置每个cuda块的cuda线程数,gpu时钟频率较低并且想要增加线程数量,可调整此参数;新一代gpu,默认值是512
NCCL_BUFFSIZE=2097152 //nccl在gpu对之间传递数据时使用的缓冲区的大小,默认4194304(4MB),值是整数,以字节为单位
```

A800 NCCL多机通信

gpu环境准备

1. 驱动和cuda

用行业镜像创建云主机默认带gpu驱动和cuda，

如果想自行安装gpu驱动和cuda，请参考网址：https://developer.nvidia.com/cuda-toolkit-archive

2. 加载nvidia_peermem内核mod

cuda11.4以上版本自带此内核mod，低于此版本则需要额外安装nv_peer_mem，建议加入开机自启动

```
sudo modprobe nvidia_peermem
```

3. 安装nvidia-fabricmanager

```
# ubuntu
# version为nvidia driver版本
version=535.54.03
main_version=$(echo $version | awk -F '.' '{print $1}')
```

```
sudo apt update
sudo apt -y install nvidia-fabricmanager-${main_version}=${version}-*
sudo systemctl start nvidia-fabricmanager
sudo systemctl status nvidia-fabricmanager
sudo systemctl enable nvidia-fabricmanager
```

A800 虚拟网卡配置

A800云主机创建结束，默认有一个eth0，此网络一般用于管理面流量转发。对于多机GPU通信场景，需要每台云主机创建4张虚拟网卡来进行数据面流量转发。以下是网卡配置的最佳实践

1. 创建子网

在私有网络VPC界面创建新的VPC，网段为192.168.0.0/16，初始子网subnet-eth1网段为192.168.1.0/24

然后切换到子网tab栏，选中刚才创建的VPC，

新增子网subnet-eth2，网段为192.168.2.0/24

新增子网subnet-eth3，网段为192.168.3.0/24

新增子网subnet-eth4，网段为192.168.4.0/24

2. 虚拟网卡创建和挂载

4个子网创建完成后，切换到虚拟网卡tab栏，每台A800云主机都需要4张虚拟网卡，4张虚拟网卡分别对应4个子网

第1张虚拟网卡对应子网subnet-eth1

第2张虚拟网卡对应子网subnet-eth2

第3张虚拟网卡对应子网subnet-eth3

第4张虚拟网卡对应子网subnet-eth4

虚拟网卡创建完成后，按照相同的顺序将虚拟网卡挂载到每台云主机上

例如挂载到每台云主机的第一张网卡都是从子网subnet-eth1申请的虚拟网卡

3. 虚拟网卡初始化

虚拟网卡配置完成，需要在云主机内部初始化网络配置，每次开机都需执行，建议加入开机自启动

```
# ubuntu
mtu=4200
devs="eth1 eth2 eth3 eth4"
for dev in $devs
do
sudo ip link set $dev mtu $mtu
sudo ip link set $dev up
sudo dhclient $dev
done
devs="mlx5_0 mlx5_1 mlx5_2 mlx5_3"
for dev in $devs
do
sudo cma_roce_tos -d $dev -t 184
sudo cma_roce_mode -d $dev -m 2
sudo echo 184 > /sys/class/infiniband/$dev/tc/1/traffic_class
done
sudo pkill dhclient
sudo modprobe rdma_cm
```

nccl多机通信验证

1. 下载并编译nccl-test

```
git clone https://github.com/NVIDIA/nccl-tests.git
cd nccl-tests
make MPI=1 MPI_HOME=/usr/mpi/gcc/openmpi-4.1.5a1 CUDA_HOME=/usr/local/cuda -j
```

2. nccl多机测试命令

```
mpirun --allow-run-as-root --oversubscribe -np {gpu卡数量} --bind-to numa -H {内网ip地址} -mca plm_rsh_args "-p 22 -q -o StrictHostKeyChecking=no" -mca coll_hcoll_enable 0 \
-mca pml ob1 -mca btl_tcp_if_include eth0 -mca btl ^openib -mca btl_openib_if_include mlx5_0:1,mlx5_1:1,mlx5_2:1,mlx5_3:1 -mca btl_openib_cpc_include rdmacm -mca btl_openib_rroce_enable 1 -x NCCL_IB_DISABLE=0 \
-x NCCL_SOCKET_IFNAME=eth0 -x NCCL_IB_GID_INDEX=3 -x NCCL_IB_TC=184 -x NCCL_IB_TIMEOUT=23 -x NCCL_IB_RETRY_CNT=7 -x NCCL_IB_PCI_RELAXED_ORDERING=1 \
-x NCCL_IB_HCA=mlx5_0,mlx5_1,mlx5_2,mlx5_3 -x CUDA_VISIBLE_DEVICES=0,1,2,3,4,5,6,7 -x NCCL_TOPO_FILE={nccl_topo_file文件} -x NCCL_NET_GDR_LEVEL=1 \
-x CUDA_DEVICE_ORDER=PCI_BUS_ID -x NCCL_ALGO=Ring -x LD_LIBRARY_PATH -x PATH {all_reduce_perf文件} -b 8 -e 8G -f 2 -g 1
# gpu卡数量:云主机机个数*8(每台云主机8张gpu卡)
# 内网ip地址:云主机eth0对应的ip地址,多个ip之间之间英文,隔开
# nccl_topo_file文件: 指定nccl_topo.xml文件绝对路径
# all_reduce_perf文件:指定all_reduce_perf文件绝对路径,例如:$HOME/nccl-tests/build/all_reduce_perf
```

常见问题

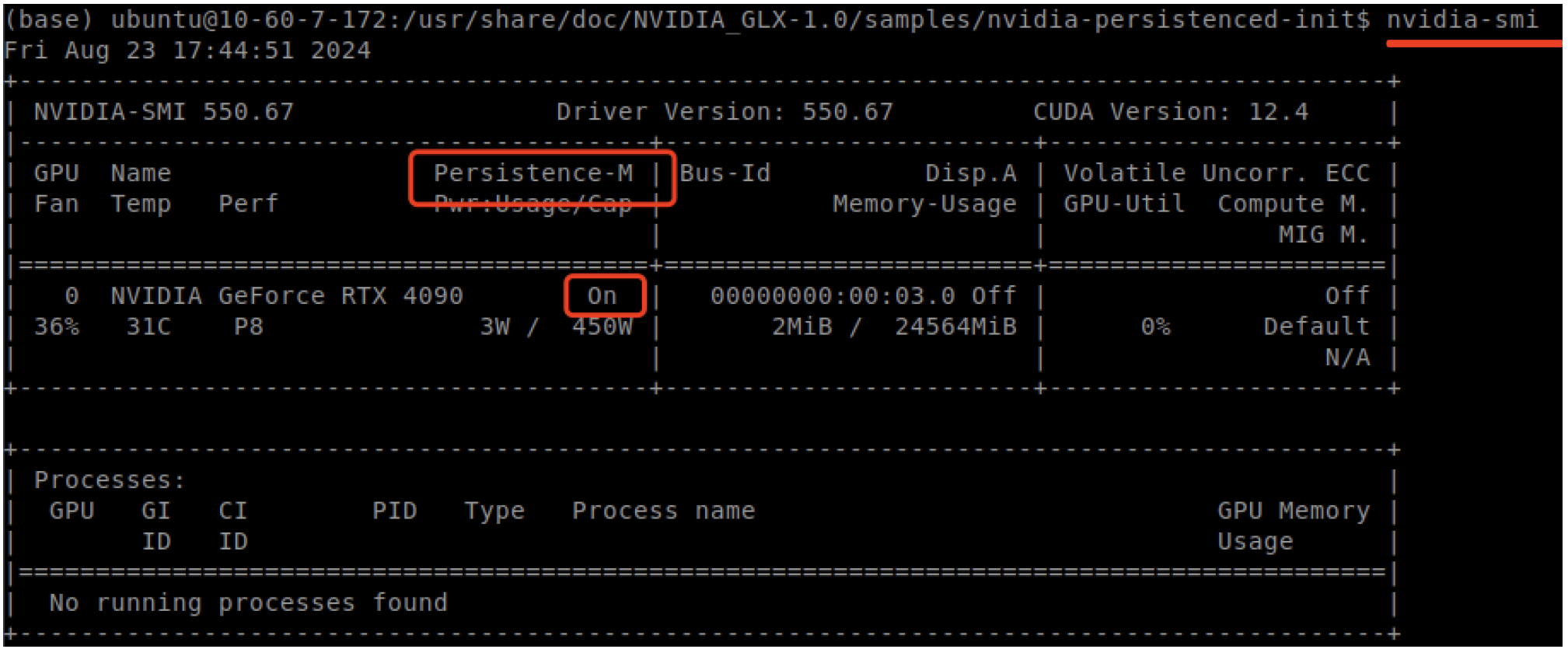
如何打开pm模式

1. 进入主机,执行以下操作

```
cd /usr/share/doc/NVIDIA_GLX-1.0/samples
sudo tar -xvjf nvidia-persistenced-init.tar.bz2
sudo bash ./nvidia-persistenced-init/install.sh
```

2. 验证是否开启成功

```
nvidia-smi
```



深度学习指南

机器学习进阶笔记系列

1. 机器学习进阶笔记之一 |TensorFlow安装与入门
2. 机器学习进阶笔记之二 |深入理解Neural Style
3. 机器学习进阶笔记之三 |深入理解Alexnet
4. 机器学习进阶笔记之四 |深入理解GoogleNet
5. 机器学习进阶笔记之五 |深入理解VGG\Residual Network
6. 机器学习进阶笔记之六 |深入理解Fast Neural Style
7. 机器学习进阶笔记之七 |MXnet初体验
8. 机器学习进阶笔记之八 |TensorFlow与中文手写汉字识别
9. 机器学习进阶笔记之九 |利用TensorFlow搞定「倒字验证码」
10. 机器学习进阶笔记之十 |那些TensorFlow上好玩的黑科技

驱动安装指南

- CentOS7环境配置
- Ubuntu14.04环境配置
- Ubuntu16.04环境配置
- Ubuntu18.04环境配置
- Windows环境配置

CentOS 7 环境配置

1. 检查GPU设备识别

```
$ yum install pciutils
$ sudo lspci | grep NVIDIA
3D controller: NVIDIA Corporation GK210GL [Tesla K80] 表示识别为K80
3D controller: NVIDIA Corporation Device 1b38 (rev a1) 表示为P40
```

2. 获取cuda网络源，并配置

NVidia官方源地址http://developer.download.nvidia.com/compute/cuda/repos/rhel7/x86_64/

```
$ wget http://developer.download.nvidia.com/compute/cuda/repos/rhel7/x86_64/cuda-repo-rhel7-8.0.61-1.x86_64.rpm
$ rpm -Uvh cuda-repo-rhel7-8.0.61-1.x86_64.rpm
```

注:安装nvidia驱动需要kernel-devel包,安装方法如下:

```
$ wget http://vault.centos.org/7.0.1406/updates/x86_64/Packages/kernel-devel-3.10.0-123.4.4.el7.x86_64.rpm
$ wget http://vault.centos.org/7.0.1406/updates/x86_64/Packages/kernel-headers-3.10.0-123.4.4.el7.x86_64.rpm
```

```
$ rpm -Uvh kernel-devel-3.10.0-123.4.4.el7.x86_64.rpm
$ rpm -Uvh kernel-headers-3.10.0-123.4.4.el7.x86_64.rpm
```

3. 安装cuda 8.0

$ yum install cuda-8-0

3.1 查看驱动状态

```
$ sudo nvidia-smi
```

看到如下输出表示GPU驱动正常：

4. 测试GPU基本功能（可选）

4.1 增加LD path

```
$ export LD_LIBRARY_PATH="/usr/local/cuda-7.5/lib64:/usr/lib64/:$LD_LIBRARY_PATH"
```

4.2 安装cuda examples

```
$ cd /usr/local/cuda/bin
$ sh cuda-install-samples-8.0.sh ~/cuda-test/
$ cd ~/cuda-test/NVIDIA_CUDA-8.0_Samples
$ make
$ ./bin/x86_64/linux/release/deviceQuery 获取设备状态
$ ./bin/x86_64/linux/release/bandwidthTest 测试设备带宽
```

Note: 如果编译过程发现Invcuvid的错误,可以执行:

```
$ find . -type f -execdir sed -i 's/UBUNTU_PKG_NAME = "nvidia-367"/UBUNTU_PKG_NAME = "nvidia-375"/g' '{}' \
```

其中nvidia-375是当前安装的驱动的版本

5. 安装cudnn

选装,注:不同AI框架对cudnn的版本支持不同

5.1 下载cudnn软件包

,需要注册nvidia账号后才能下载。

注意:CentOS 下载 cuDNN v5.1 Library for Linux

5.2 安装

案例使用cudnn5.1, 因为TensorFlow目前仅支持5.1 `$ tar -zxf cudnn-8.0-linux-x64-v5.1.tgz`

解压的路径可以自由选择，一般是/usr/lib下面，这边假设为<CUDNN_INSTALL_PATH> `$ export LD_LIBRARY_PATH=:$LD_LIBRARY_PATH`

FAQ

1. nvidia-smi 发现 GPU使用率100%，为什么?

这个问题是系统读取gpu状态信息不准确导致，执行下列命令可更正，让系统读取命令正确。`# nvidia-smi -pm 1`

Ubuntu 14.04 环境配置

1. 检查GPU设备识别

```
$ sudo lspci | grep NVIDIA
3D controller: NVIDIA Corporation GK210GL [Tesla K80] 表示识别为K80
3D controller: NVIDIA Corporation GP102GL [Tesla P40] (rev a1) 表示为P40
```

2. 获取cuda网络源，并配置：

NVidia官方源地址 http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1404/x86_64/

```
$ wget http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1404/x86_64/cuda-repo-ubuntu1404_8.0.44-1_amd64.deb
$ sudo dpkg -i cuda-repo-ubuntu1404_8.0.44-1_amd64.deb
$ sudo apt-get update
```

3. 安装cuda 8.0

在安装前请uname -a 检测当前内核版本，然后确保对应版本的kernel-header包已经安装，否则无法正常编译驱动。

```
$ uname -a
$ Linux X-X-X-X 3.13.0-123-generic #172-Ubuntu SMP Mon
$ sudo apt search 3.13.0-123-generic
$ p linux-cloud-tools-3.13.0-123-generic - Linux kernel version specific cloud tools for version 3.13.0-123
$ p linux-headers-3.13.0-123-generic - Linux kernel headers for version 3.13.0 on 64 bit x86 SMP
$ p linux-headers-3.13.0-123-generic:i386 - Linux kernel headers for version 3.13.0 on 32 bit x86 SMP
$ sudo apt-get install linux-headers-3.13.0-123-generic
```

安装cuda

```
$ sudo apt-get install cuda-8.0
```

3.1 查看驱动状态

`$ sudo nvidia-smi`
看到如下输出表示GPU驱动正常：

```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 375.66                 Driver Version: 375.66                    |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla P40           Off  | 0000:00:08.0     Off |                    0 |
| N/A   29C    P0    48W / 250W |      0MiB / 22912MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   1  Tesla P40           Off  | 0000:00:09.0     Off |                    0 |
| N/A   34C    P0    49W / 250W |      0MiB / 22912MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID  Type  Process name                               Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

4. 测试GPU基本功能（可选）

4.1 增加LD path

```
$ export LD_LIBRARY_PATH="/usr/local/cuda-7.5/lib64:/usr/lib64/:$LD_LIBRARY_PATH"
```

4.2 安装cuda examples

```
$ cd /usr/local/cuda/bin
$ sh cuda-install-samples-8.0.sh ~/cuda-test/
$ cd ~/cuda-test/NVIDIA_CUDA-8.0_Samples
$ make
$ ./bin/x86_64/linux/release/deviceQuery 获取设备状态
$ ./bin/x86_64/linux/release/bandwidthTest 测试设备带宽
```

如果编译过程发现lnvcuvid的错误,可以执行:

```
$ find . -type f -execdir sed -i 's/UBUNTU_PKG_NAME = "nvidia-367"/UBUNTU_PKG_NAME = "nvidia-375"/g' '{}' \
```

其中nvidia-375是当前安装的驱动的版本

5. 安装cudnn

选装,注:不同AI框架对cudnn的版本支持不同

5.1 下载cudnn软件包

https://developer.nvidia.com/cudnn ,需要注册nvidia账号后才能下载

5.2 安装

案例使用cudnn5.1, 因为TensorFlow目前仅支持5.1

ubuntu 可以选择 cuDNN v5.1 Runtime Library for Ubuntu14.04 (Deb) `$ sudo dpkg -i libcudnn5_5.1.10-1+cuda8.0_amd64.deb`

5. 关闭ubuntu自动更新内核及NVidia Tools

建议操作 `$ sudo vim /etc/apt/apt.conf.d/10periodic` 将 APT::Periodic::Update-Package-Lists "1"; 修改为 APT::Periodic::Update-Package-Lists "0"; 以禁止ubuntu自动更新软件包

FAQ

1. nvidia-smi 发现 GPU使用率100%，为什么?

这个问题是系统读取gpu状态信息不准确导致,执行下列命令可更正,让系统读取命令正确。`# nvidia-smi -pm 1`

2. 除自行安装外，是否有其它可获得驱动镜像的方法?

可提交工单,或联系工作人员,获得UCloud制作的包含GPU驱动和Cuda环境的镜像,节省人工安装的时间。

Ubuntu 16.04 环境配置

1. 检查GPU设备识别

```
$ sudo lspci | grep NVIDIA
3D controller: NVIDIA Corporation GK210GL [Tesla K80] 表示识别为K80
3D controller: NVIDIA Corporation GP102GL [Tesla P40] (rev a1) 表示为P40
```

2. 屏蔽开源驱动nouveau

编辑如下文件：

```
sudo vim /etc/modprobe.d/blacklist-nouveau.conf
```

写入下列内容：

```
blacklist nouveau
blacklist lbm-nouveau
options nouveau modeset=0
alias nouveau off
```

```
alias lbm-nouveau off
```

更新并重启：

```
sudo update-initramfs -u
sudo reboot
sudo apt-get install build-essential pkg-config linux-headers-`uname -r`
```

3. 安装nvidia驱动

3.1 下载

到nvidia官网下载合适的驱动（目前版本418.126.02），地址https://www.nvidia.com/Download/index.aspx?lang=en-us

也可从UFile下载，速度更快 http://gpu.cn-bj.ufileos.com/NVIDIA-Linux-x86_64-418.126.02.run

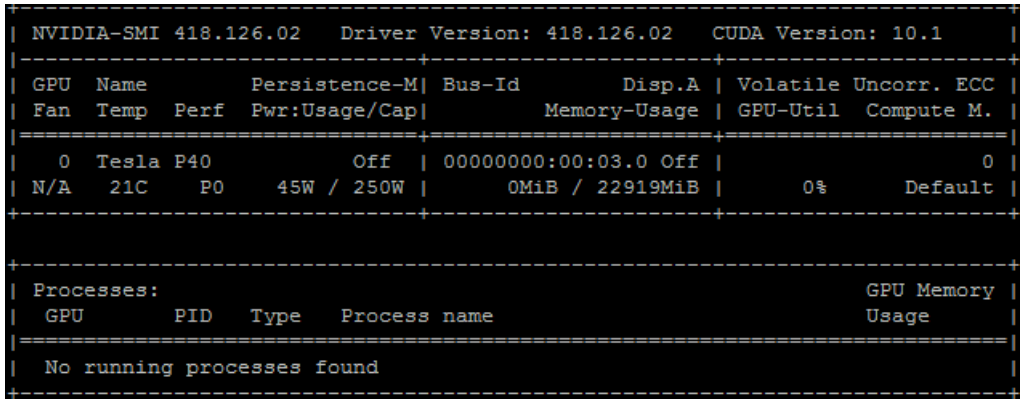
3.2 安装

```
sudo chmod +x NVIDIA-Linux-x86_64-418.126.02.run
sudo ./NVIDIA-Linux-x86_64-418.126.02.run
```

3.3 查看驱动状态

`$ sudo nvidia-smi`

看到如下输出表示GPU驱动正常：



4. 安装cuda库

4.1 网络安装

```
sudo wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64/cuda-ubuntu1604.pin

sudo mv cuda-ubuntu1604.pin /etc/apt/preferences.d/cuda-repository-pin-600

sudo apt-key adv --fetch-keys http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64/7fa2af80.pub

sudo add-apt-repository "deb http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64/ /"

sudo apt-get update
```

```
sudo apt-get -y install cuda
```

FAQ

1. nvidia-smi 发现 GPU使用率100%，为什么?

这个问题是系统读取gpu状态信息不准确导致,执行下列命令可更正,让系统读取命令正确。`#sudo nvidia-smi -pm 1`

2. 除自行安装外，是否有其它可获得驱动镜像的方法?

可提交工单,或联系工作人员,获得UCloud制作的包含GPU驱动和Cuda环境的镜像,节省人工安装的时间。

Ubuntu 18.04 环境配置

1. 检查GPU设备识别

```
$ sudo lspci | grep NVIDIA
3D controller: NVIDIA Corporation GK210GL [Tesla K80] 表示识别为K80
3D controller: NVIDIA Corporation GP102GL [Tesla P40] (rev a1) 表示为P40
```

2. 屏蔽开源驱动nouveau

编辑如下文件：

```
sudo vim /etc/modprobe.d/blacklist-nouveau.conf
```

写入下列内容：

```
blacklist nouveau
blacklist lbm-nouveau
options nouveau modeset=0
alias nouveau off
```

```
alias lbm-nouveau off
```

更新并重启：

```
sudo update-initramfs -u
sudo reboot
sudo apt-get install build-essential pkg-config
```

控制台Ubuntu 18.04镜像的内核为4.15.0-68-generic，该版本 linux-headers-4.15.0-68在Ubuntu官方已无法下载（状态deleted），此为安装驱动所必需，建议先升级内核至后续版本。

可从官方 https://kernel.ubuntu.com/~kernel-ppa/mainline/ 下载内核，例如4.15.1

也可从UFile下载，速度更快

```
http://gpu.cn-bj.ufileos.com/linux-headers-4.15.1-041501-generic_4.15.1-041501.201802031831_amd64.deb
http://gpu.cn-bj.ufileos.com/linux-headers-4.15.1-041501_4.15.1-041501.201802031831_all.deb
http://gpu.cn-bj.ufileos.com/linux-image-4.15.1-041501-generic_4.15.1-041501.201802031831_amd64.deb
```

安装内核，重启并查看版本：

```
sudo dpkg -i *.deb
sudo reboot
uname -r
```

3. 安装nvidia驱动

3.1 下载

到nvidia官网下载合适的驱动(目前版本418.126.02),地址https://www.nvidia.com/Download/index.aspx?lang=en-us

也可从UFile下载,速度更快 http://gpu.cn-bj.ufileos.com/NVIDIA-Linux-x86_64-418.126.02.run

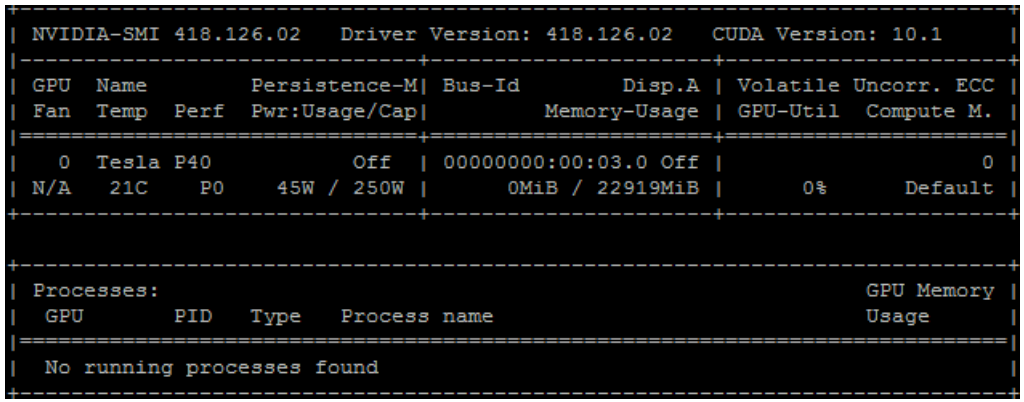
3.2 安装

```
sudo chmod +x NVIDIA-Linux-x86_64-418.126.02.run
sudo ./NVIDIA-Linux-x86_64-418.126.02.run
```

3.3 查看驱动状态

`$ sudo nvidia-smi`

看到如下输出表示GPU驱动正常:



4. 安装cuda库

4.1 网络安装

```
sudo wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/cuda-ubuntu1804.pin

sudo mv cuda-ubuntu1804.pin /etc/apt/preferences.d/cuda-repository-pin-600

sudo apt-key adv --fetch-keys http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/7fa2af80.pub

sudo add-apt-repository "deb http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/ /"

sudo apt-get update

sudo apt-get -y install cuda
```

4.2 本地安装

```
wget http://developer.download.nvidia.com/compute/cuda/10.2/Prod/local_installers/cuda_10.2.89_440.33.01_linux.run
sudo sh cuda_10.2.89_440.33.01_linux.run
```

FAQ

1. nvidia-smi 发现 GPU使用率100%，为什么?

这个问题是系统读取gpu状态信息不准确导致,执行下列命令可更正,让系统读取命令正确。`#sudo nvidia-smi -pm 1`

2. 除自行安装外，是否有其它可获得驱动镜像的方法?

可提交工单,或联系工作人员,获得UCloud制作的包含GPU驱动和Cuda环境的镜像,节省人工安装的时间。

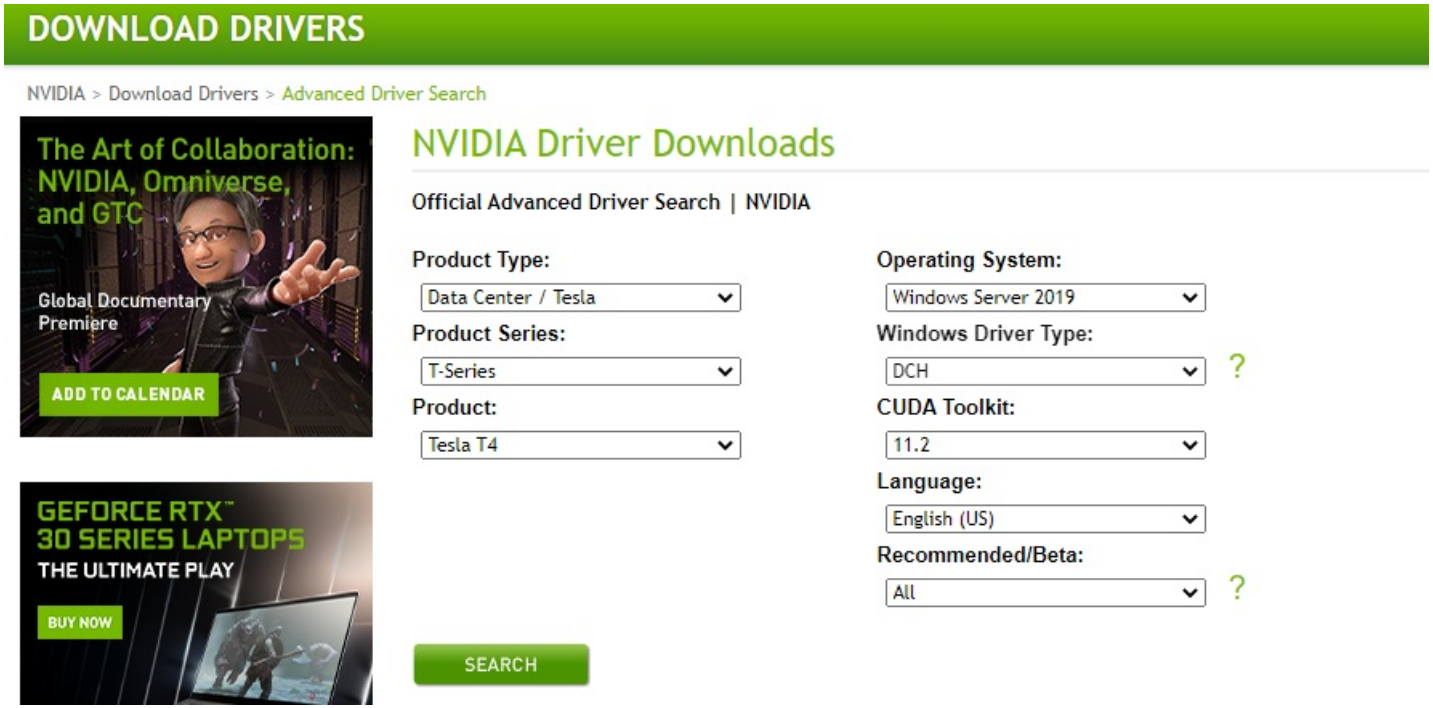
Windows 环境配置

Windows 驱动安装

1.登录GPU云主机；

2.访问NVIDIA驱动下载 官网；

3.根据GPU云主机操作系统及GPU规格，选择操作系统和安装包，本文以T4为例，如下图所示：



4.打开下载驱动程序所在的文件夹，双击安装文件开始安装，按照界面上的提示安装驱动程序并根据需要重启实例。安装完成后，如需验证 GPU 是否正常工作，请查看设备管理器。

常见报错

1. windows安装驱动报错缺少wlanapi.dll

解决方式：打开【服务器管理器】窗口，选择【管理】菜单中的【添加角色和功能】选项，启动【添加角色和功能向导】，勾选【无线LAN服务】，也可以直接点击【仪表板】中【快速启动】的【添加角色和功能】链接来启动。


服务器管理器
服务器管理器 ▸ 仪表板
管理(M)
工具(T)
视图(V)
帮助(H)
添加角色和功能
删除角色和功能
添加服务器
创建服务器组
服务器管理器属性
仪表板
本地服务器
所有服务器
文件和存储服务
欢迎使用服务器管理器
快速启动(Q)
新增功能(W)
了解详细信息(L)
1 配置此本地服务器
2 添加角色和功能
3 添加要管理的其他服务器
4 创建服务器组
5 将此服务器连接到云服务
隐藏
角色和服务器组
角色: 1 | 服务器组: 1 | 服务器总数: 1

!> 该步骤需要重启系统后才能生效。

Rocky Linux 8环境配置

1.检查GPU设备识别

```
# yum install pciutils
# sudo lspci | grep NVIDIA
3D controller: NVIDIA Corporation GV100GL [Tesla V100S PCIe 32GB] (rev a1) 表示识别为V100S
```

2.将 Nvidia驱动程序添加到软件包管理器列表中

```
sudo dnf config-manager --add-repo https://developer.download.nvidia.com/compute/cuda/repos/rhel8/x86_64/cuda-rhel8.repo
```

3.安装NVIDIA驱动和配置

```
sudo dnf install nvidia-driver nvidia-settings
```

4.安装CUDA

```
sudo dnf install cuda-driver
```

5.重启系统并验证

重启操作系统之后，通过"nvidia-smi"命令来查看显卡是否正常工作。

```
[root@10-13-47-75 ~]# nvidia-smi
Thu Apr 6 17:10:52 2023
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 530.30.02 Driver Version: 530.30.02 CUDA Version: 12.1 |
|-----------------------------------------+----------------------+----------------------+
| GPU Name Persistence-M| Bus-Id Disp.A | Volatile Uncorr. ECC |
| Fan Temp Perf Pwr:Usage/Cap| Memory-Usage | GPU-Util Compute M. |
| | | MIG M. |
|=========================================+======================+======================|
| 0 Tesla V100S-PCIE-32GB Off| 00000000:00:03.0 Off | 0 |
| N/A 26C P0 35W / 250W| 0MiB / 32768MiB | 0% Default |
| | | N/A |
+-----------------------------------------+----------------------+----------------------+
```

```
+-----------------------------------------------------------------------------+
| Processes:                                                                  |
| GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|       ID   ID                                                   Usage      |
|=============================================================================|
| No running processes found                                                  |
+-----------------------------------------------------------------------------+
```

Redhat 6.6环境配置

1.检查GPU设备识别

```
# yum install pciutils
# sudo lspci | grep NVIDIA
3D controller: NVIDIA Corporation Device 1df6 (rev a1) 表示识别为V100S
```

运行"yum install pciutils"提示"This system is not registered with an entitlement server. You can use subscription-manager to register."

请运行以下命令启动Redhat账号登录：

`subscription-manager register`

按照提示输入 Redhat 帐户的用户名和密码。

确认系统已成功注册，并启用订阅:

`subscription-manager list --consumed`

运行以下命令以更新系统：

`yum update`

2.下载GPU驱动

```
wget https://cn.download.nvidia.com/tesla/460.106.00/NVIDIA-Linux-x86_64-460.106.00.run
```

驱动版本可根据业务需求从nvidia官方链接下载，https://www.nvidia.cn/Download/index.aspx?lang=cn。

3.禁用nouveau

因部分linux系统安装的nouveau驱动与nvidia驱动有冲突，因此需先禁用。输入"lsmod |grep nouveau"，如果有返回，则需禁用，禁用方式如下：

```
# lsmod | grep nouveau
nouveau 1514531 0
ttm 89568 1 nouveau
drm_kms_helper 127731 1 nouveau
drm 355270 3 nouveau,ttm,drm_kms_helper
i2c_algo_bit 5903 1 nouveau
i2c_core 29164 5 i2c_piix4,nouveau,drm_kms_helper,drm,i2c_algo_bit
mxm_wmi 1967 1 nouveau
video 21686 1 nouveau
wmi 6287 2 nouveau,mxm_wmi
```

```
# tail -1 /etc/modprobe.d/blacklist.conf
blacklist nouveau
```

4.安装驱动

```
# sudo sh <driver_installer>.run --kernel-source-path=/usr/src/kernels/<kernel_version>
```

其中"driver_installer"是驱动程序的安装程序文件名，"kernel_version"是当前系统检查搭到的内核版本号。检查当前正在运行的内核版本，可以通过以下命令来查看：

```
uname -r
```

5.验证GPU卡是否正常工作

使用nvidia-smi来验证，如果能正常显示出卡型，即可正常使用。

```
# nvidia-smi
Fri Apr 7 15:02:14 2023
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 460.106.00 Driver Version: 460.106.00 CUDA Version: 11.2 |
|-------------------------------+----------------------+----------------------+
| GPU Name Persistence-M| Bus-Id Disp.A | Volatile Uncorr. ECC |
| Fan Temp Perf Pwr:Usage/Cap| Memory-Usage | GPU-Util Compute M. |
| | | MIG M. |
|===============================+======================+======================|
| 0 Tesla V100S-PCI... Off | 00000000:00:03.0 Off | 0 |
```

```
| N/A 26C P0 35W / 250W | 0MiB / 32510MiB | 0% Default |
| | | N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes: |
| GPU GI CI PID Type Process name GPU Memory |
| ID ID Usage |
|=============================================================================|
| No running processes found |
```

AI绘画Web UI使用指南

1.创建一台GPU云主机

创建流程参考创建第一台云主机

创建GPU云主机时，镜像选择“AI绘画Web UI”，操作路径：镜像市场——>AI绘图 Web UI，便捷安装AI绘画Web UI页面，镜像内置环境：CentOS 7.8。

推荐机型及配置：

GPU型云主机 V100S 最低配置：10核32G一颗GPU

?> 内存请选择32GB及以上，否则模型加载时可能会触发OOM。

绑定EIP并在外网防火墙放行TCP 7860端口。

2.登录GPU云主机

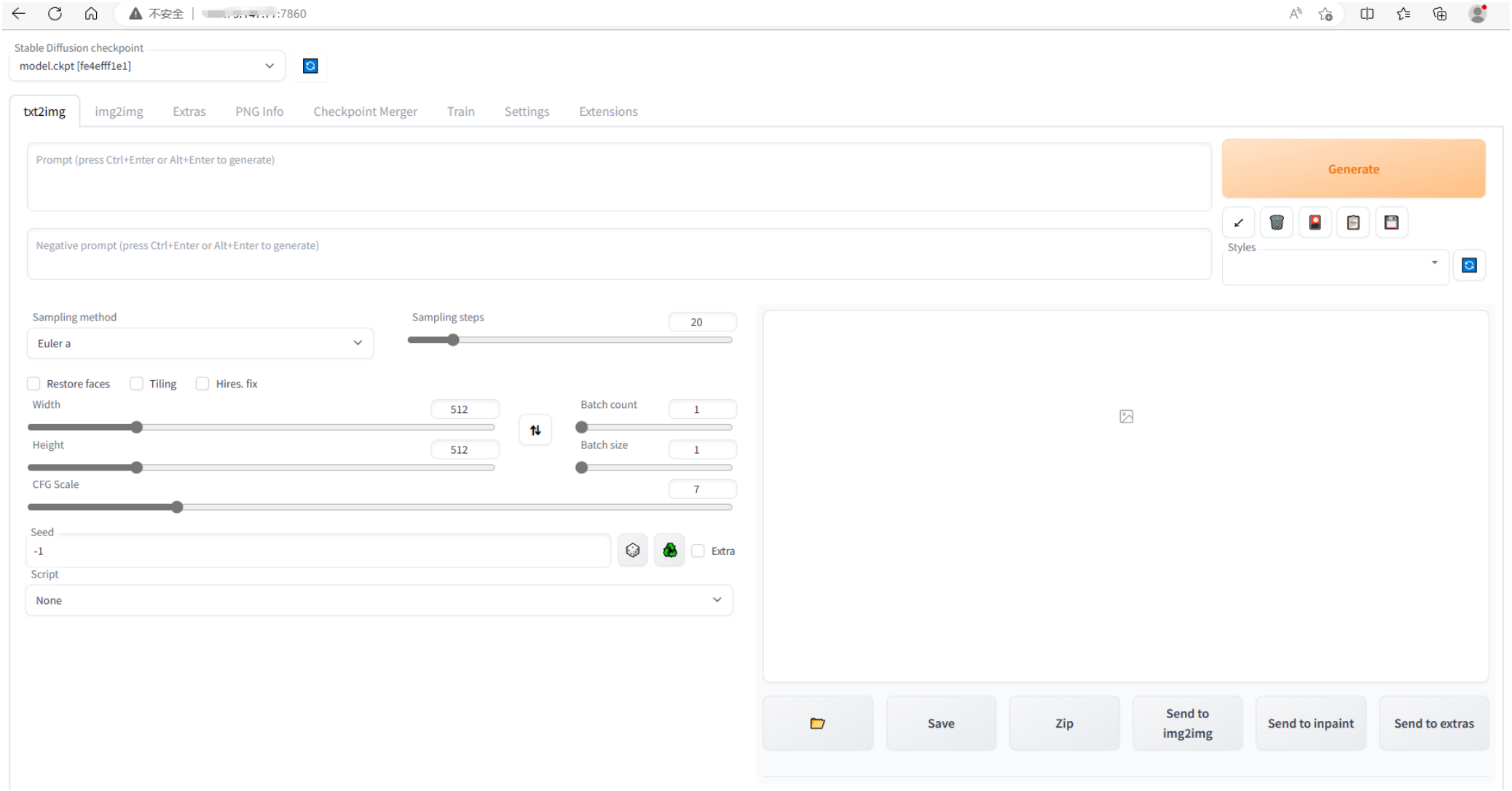
登录GPU云主机，按照系统首页提示，输入如下命令，启动stable diffusion Web UI。

```
su ucloud
cd /home/ucloud/stable-diffusion-webui
export GRADIO_SERVER_NAME=0.0.0.0
export GRADIO_SERVER_PORT=7860
python ./launch.py --xformers
```

```
WARNING! The remote SSH server rejected X11 forwarding request.
Last login: Thu Mar 23 09:44:48 2023 from 106.75.220.2
--------------------------------------------------------------------------------
温馨提示:
当前镜像已预装nvidia-driver,cuda,conda,stable-diffusion-webui。
stable-diffusion-webui启动方式:
su ucloud
cd /home/ucloud/stable-diffusion-webui
export GRADIO_SERVER_NAME=0.0.0.0
export GRADIO_SERVER_PORT=7860
python ./launch.py --xformers

启动好之后，通过浏览器访问: http://ip:7860即可，注意ip需要替换为云主机的外网ip,
如果无法访问请在控制台检查安全规则是否放行上述端口。
--------------------------------------------------------------------------------

(base) [root@10-13-163-123 ~]# su ucloud
(base) [ucloud@10-13-163-123 root]$ cd /home/ucloud/stable-diffusion-webui
(base) [ucloud@10-13-163-123 stable-diffusion-webui]$ export GRADIO_SERVER_NAME=0.0.0.0
(base) [ucloud@10-13-163-123 stable-diffusion-webui]$ export GRADIO_SERVER_PORT=7860
(base) [ucloud@10-13-163-123 stable-diffusion-webui]$ python ./launch.py --xformers
Python 3.10.9 (main, Jan 11 2023, 15:21:40) [GCC 11.2.0]
Commit hash: a9fed7c364061ae6efb37f797b6b522cb3cf7aa2
Installing requirements for Web UI
Launching Web UI with arguments: --xformers
Loading weights [fe4efff1e1] from /home/ucloud/stable-diffusion-webui/models/Stable-diffusion/model.ckpt
Creating model from config: /home/ucloud/stable-diffusion-webui/configs/v1-inference.yaml
LatentDiffusion: Running in eps-prediction mode
DiffusionWrapper has 859.52 M params.
Applying xformers cross attention optimization.
Textual inversion embeddings loaded(0):
Model loaded in 50.2s (load weights from disk: 41.0s, create model: 0.5s, apply weights to model: 0.5s, apply half(): 0.3s, load VAE: 7.3s, move model to device: 0.6s).
Running on local URL:  http://0.0.0.0:7860

To create a public link, set `share=True` in `launch()`.
Startup time: 59.8s (import gradio: 3.7s, import ldm: 1.4s, other imports: 3.7s, load scripts: 0.4s, load SD checkpoint: 50.2s, create ui: 0.2s).
```

3.登录Web UI页面进行绘图

登录http://eip:7860即可访问stable diffusion Web UI页面。


不安全
:7860
Stable Diffusion checkpoint
model.ckpt [fe4efff1e1]
txt2img
img2img
Extras
PNG Info
Checkpoint Merger
Train
Settings
Extensions
Prompt (press Ctrl+Enter or Alt+Enter to generate)
Negative prompt (press Ctrl+Enter or Alt+Enter to generate)
Generate
Styles
Sampling method
Euler a
Sampling steps
20
Restore faces
Tiling
Hires. fix
Width
512
Height
512
Batch count
1
Batch size
1
CFG Scale
7
Seed
-1
Extra
Script
None
Save
Zip
Send to img2img
Send to inpaint
Send to extras

AI绘画stable diffusion 实践

1.创建一台GPU云主机

创建流程参考创建第一台云主机

创建GPU云主机时，镜像选择“AI绘画stable diffusion平台”，操作路径：镜像市场——>AI绘画stable diffusion平台，便捷安装stable diffusion，镜像内置环境：CentOS 7.8。

推荐机型及配置：

GPU型云主机 T4/T4S 最低配置：8核32G一颗GPU

GPU型云主机 V100S 最低配置：10核32G一颗GPU

GPU型云主机 P40 最低配置：8核32G 一颗GPU

?> 内存请选择32GB及以上，否则模型加载时可能会触发OOM。

绑定EIP并在外网防火墙放行TCP 8888端口。

2.使用stable diffusion

2.1 方式一：使用jupyter nodebook新建ldm环境（推荐）

根据外网ip地址，访问jupyter：http://EIP:8888

Token authentication is enabled

If no password has been configured, you need to open the notebook server with its login token in the URL, or paste it above. This requirement will be lifted if you enable a password.

The command:

```
jupyter notebook list
```

will show you the URLs of running servers with their tokens, which you can copy and paste into your browser. For example:

```
Currently running servers:
http://localhost:8888/?token=c8de56fa... :: /Users/yo
u/notebooks
```

or you can paste just the token value into the password field on this page.

See the documentation on how to enable a password in place of token authentication, if you would like to avoid dealing with random tokens.

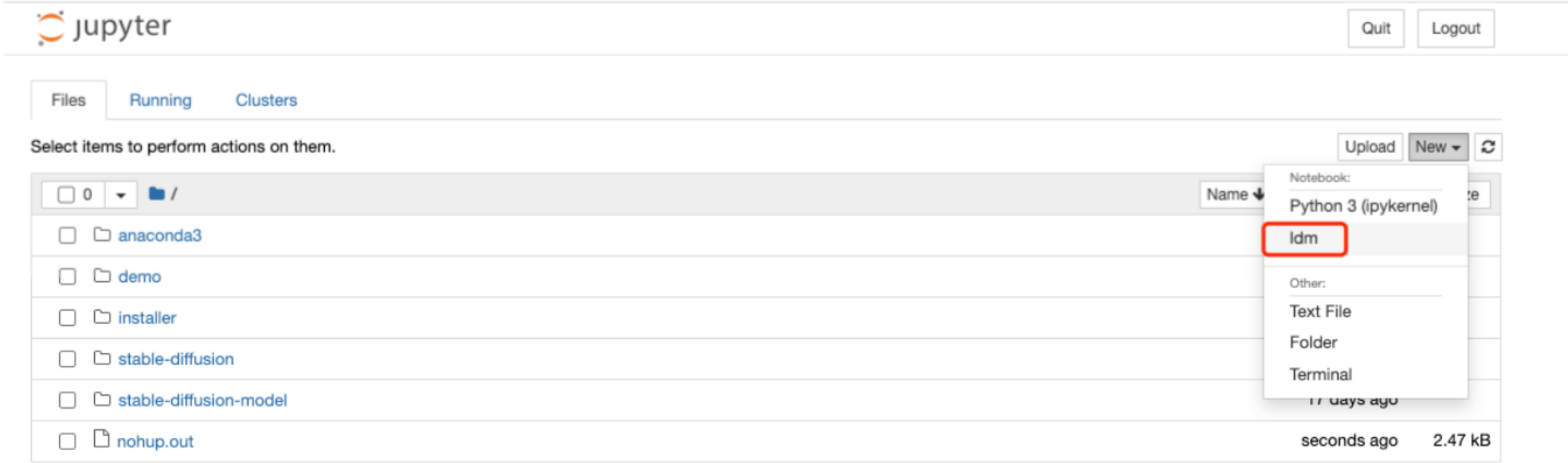
Cookies are required for authenticated access to notebooks.

Setup a Password

You can also setup a password by entering your token and a new password on the fields below:

输入token（在/root/.jupyter/jupyter_notebook_config.py中查看c.NotebookApp.token的配置，可自行修改。）

新建ldm



如您使用英文版描述，可参考以下示例：

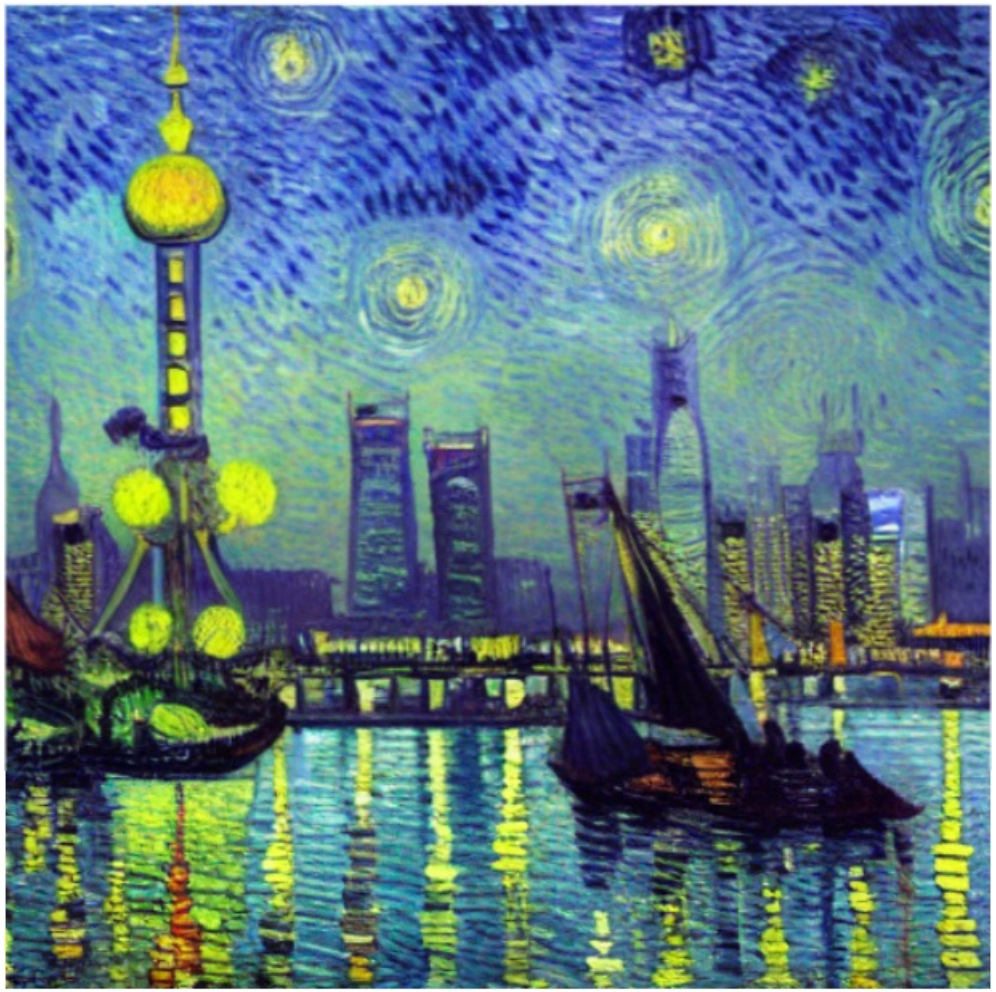
添加如下示例代码后运行（以"A thriving view alongside Pearl of the Orient in Shanghai , by Van Gogh, oil painting trending on artstation HQ"为示例）

```
from torch import autocast
from diffusers import StableDiffusionPipeline
from IPython.display import Image

width = 512
```



```
height = 512

pipe = StableDiffusionPipeline.from_pretrained(
"/root/demo/stable-diffusion-v1-4").to("cuda")

prompt = "A thriving view alongside Pearl of the Orient in Shanghai , by Van Gogh, oil painting trending on artstation HQ"
with autocast("cuda"):
image = pipe(prompt, height, width)["sample"][0]

image.save("Van_Gogh_Style_Shanghai.png")

# show the image in web
Image(filename = 'Van_Gogh_Style_Shanghai.png', width=width, height=height)
```

生成图片即可立即查看。



如您使用中文版描述，可参考以下示例：

添加如下示例代码后运行（以"大漠孤烟直，长河落日圆，油画"为示例）

```
from diffusers import StableDiffusionPipeline
from IPython.display import Image
```

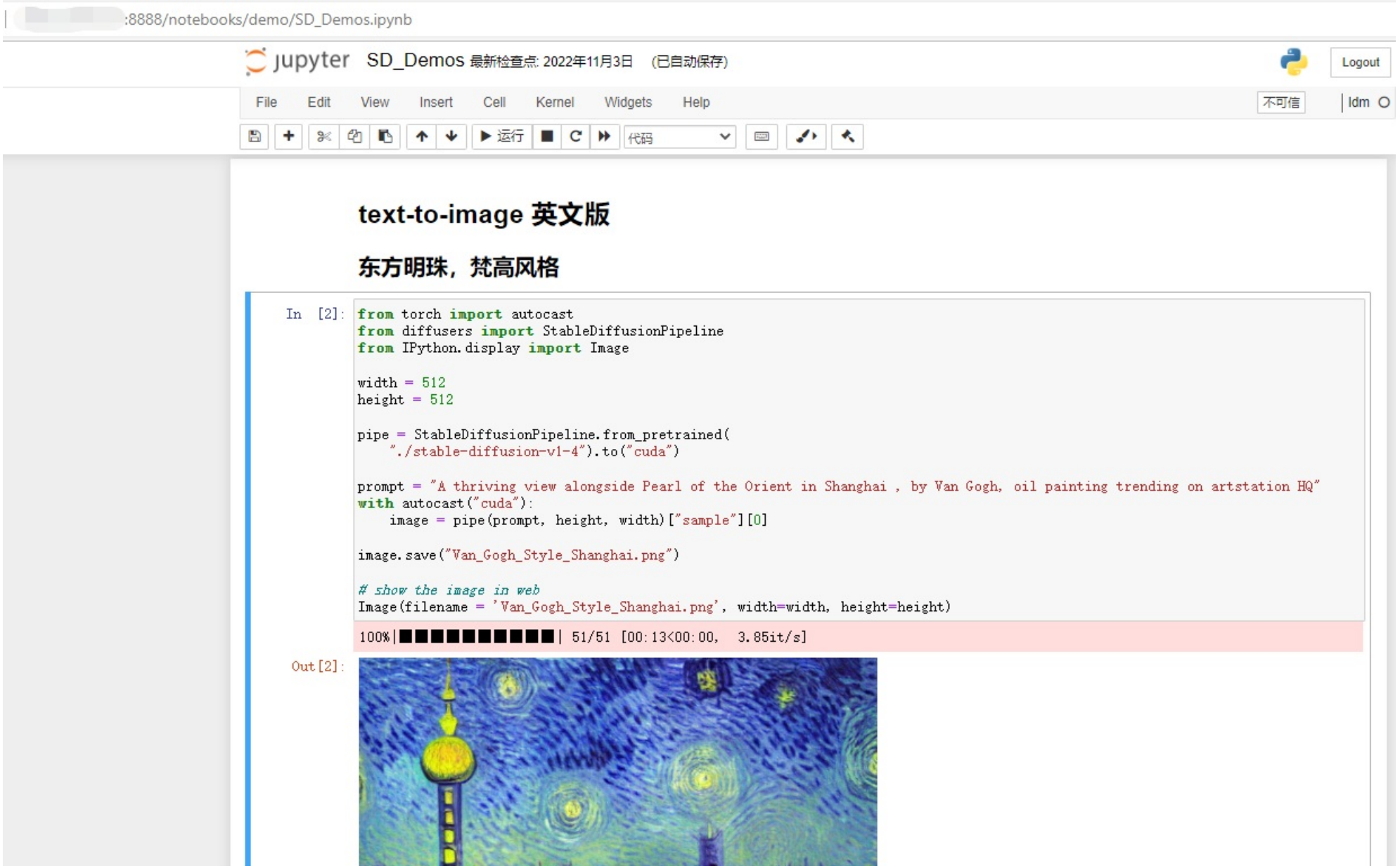
```
width = 512
height = 512

pipe = StableDiffusionPipeline.from_pretrained("IDEA-CCNL/Taiyi-Stable-Diffusion-1B-Chinese-v0.1").to("cuda")

prompt = '大漠孤烟直,长河落日圆,油画'
image = pipe(prompt, guidance_scale=7.5).images[0]
image.save("油画.png")

# show the image in web
Image(filename = '油画.png', width=width, height=height)
```

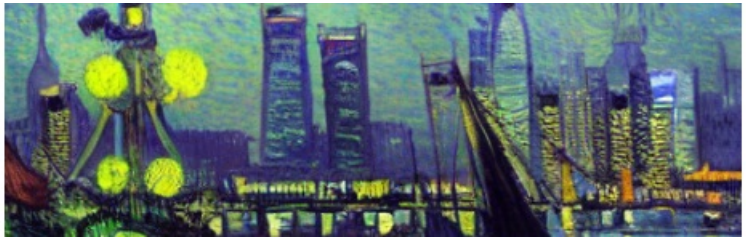
生成图片即可立即查看。

?> 1. 选中代码分区，点击【运行】，如出现In[*]，则表示代码运行中，静等出图结果即可；

2. 如您需要调整画布尺寸，需保证调整后长宽的均为8的整数倍。

如您想快速尝试，还可直接使用demo/路径下的SD_Demos.ipynb，内置英文版、中文版模型，如下图所示：





2.2 方式二：使用stable diffusion的sample script

2.2.1 切换conda环境

```
conda activate ldm
```

2.2.2 执行sample脚本

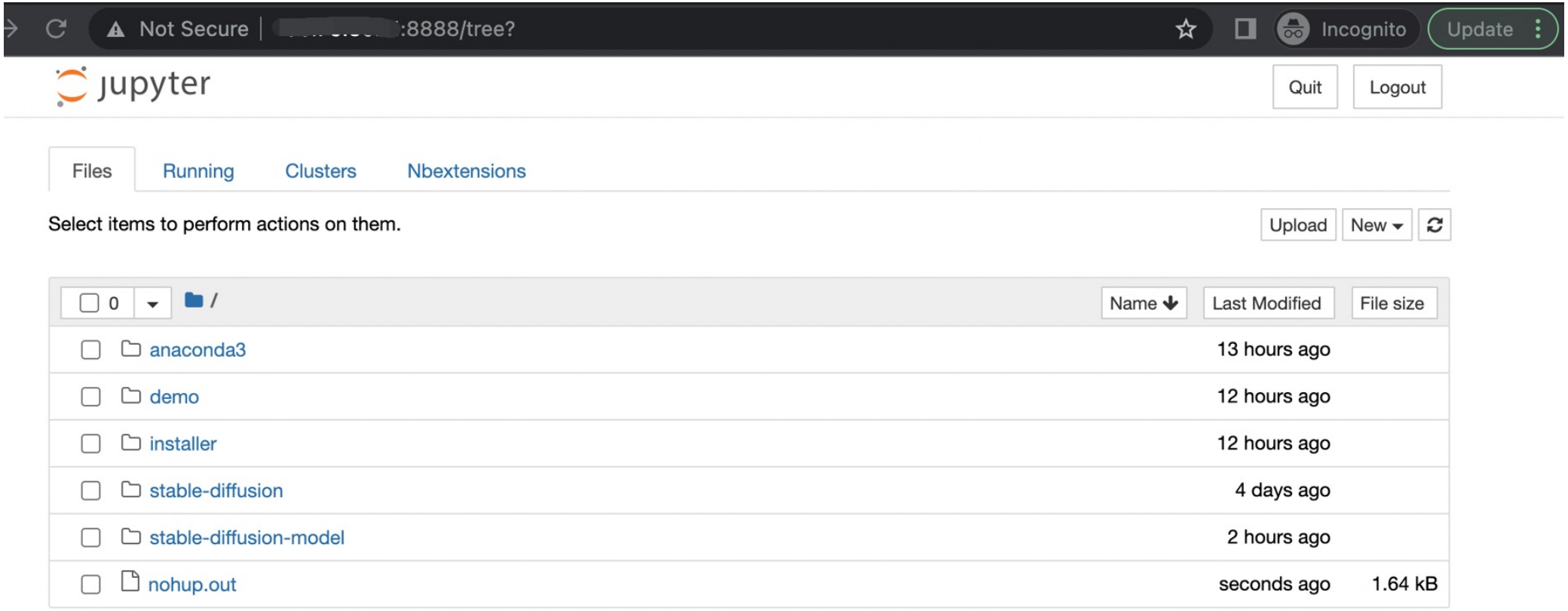
执行脚本，输入您预想图画的描述，即可得到图片（以下以“A valley full of flowers , by Van Gogh, oil painting trending on artstation HQ”为例），生成的图片在 /root/stable-diffusion/outputs/txt2img-samples/目录下。

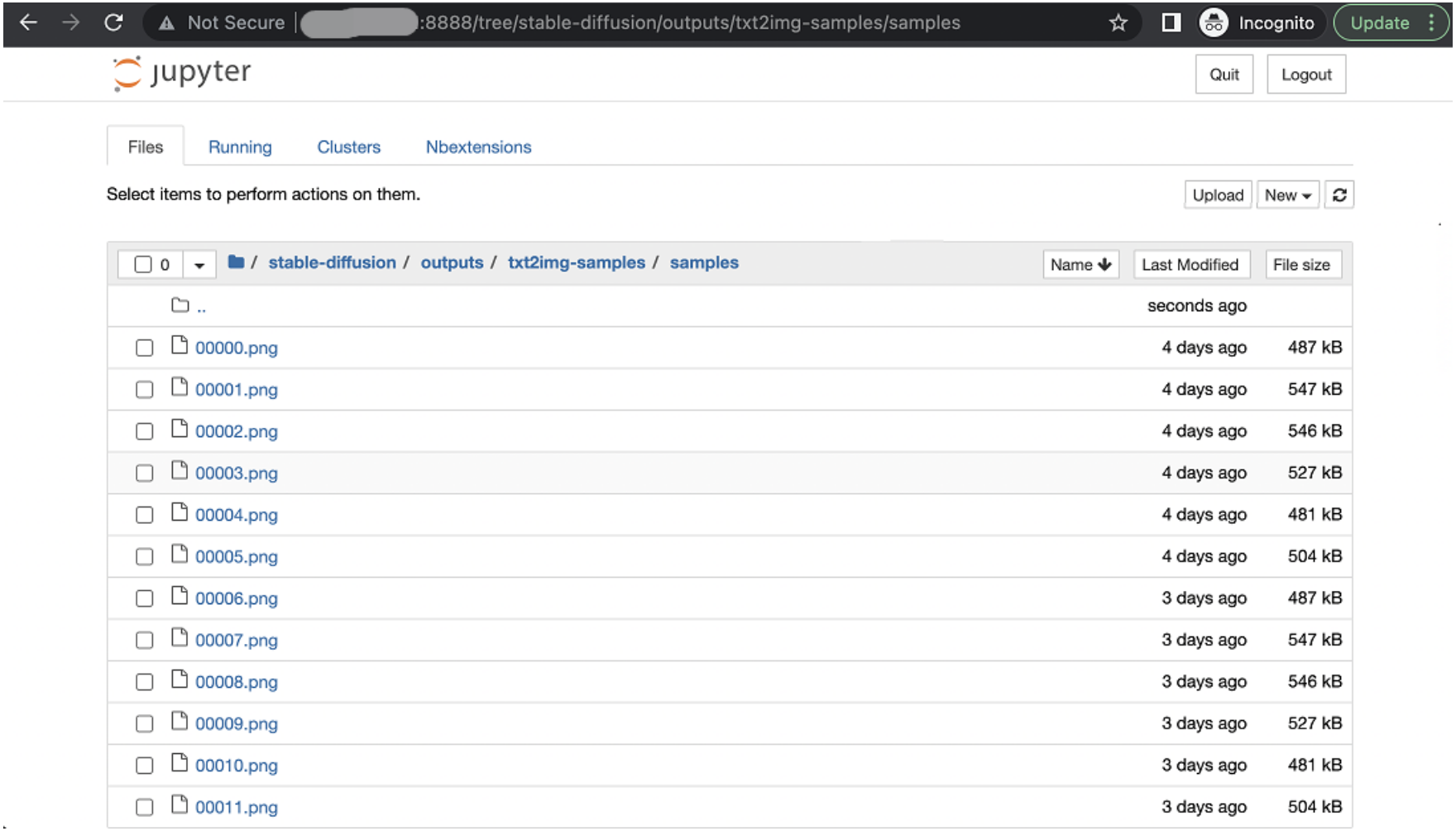
```
cd stable-diffusion
python scripts/txt2img.py --prompt "A valley full of flowers , by Van Gogh, oil painting trending on artstation HQ"
```

2.2.3 使用jupyter页面查看

根据外网ip地址，访问jupyter：http://EIP:8888

输入token（在/root/.jupyter/jupyter_notebook_config.py中查看c.NotebookApp.token的配置，可自行修改。）





根据导航点击预览图片

Alpaca-LoRA模型

简介

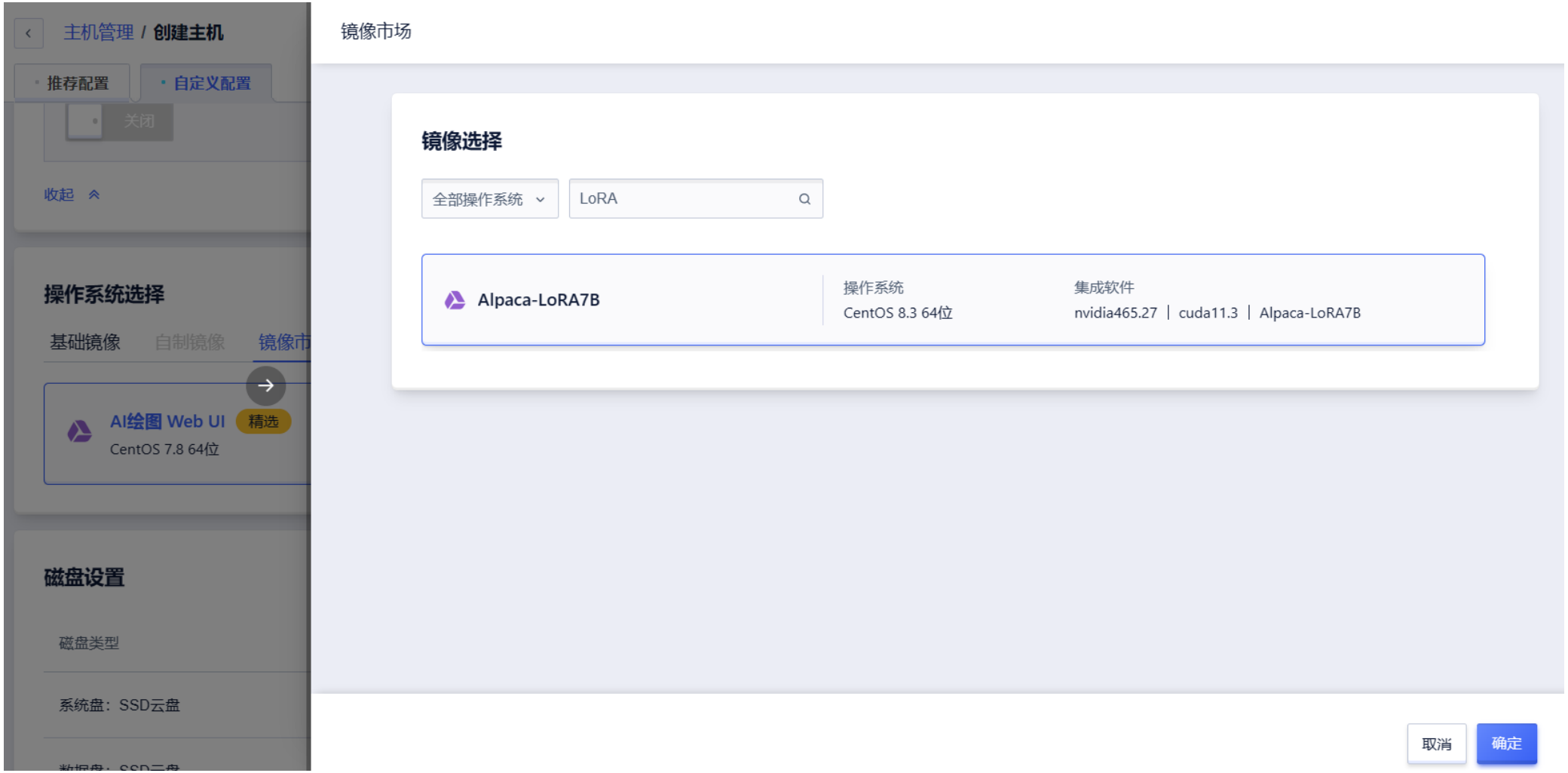
Alpaca LoRA是使用Lora(Low-rank Adaptation)技术在Meta的LLaMA 7B模型上微调，只需要训练很小一部分参数就可以获得媲美 Standford Alpaca 模型的效果。可以认为是ChatGPT轻量级的开源版本。

快速部署

登录UCloud控制台(https://console.ucloud.cn/uhost/uhost/create)，机型选择“GPU型”，“V100S”，CPU及GPU颗数等详细配置按需选择。最低推荐配置：10核CPU 64G内存 1颗V100S。

镜像选择“镜像市场”，镜像名称搜索“Alpaca-LoRA7B”，选择该镜像创建GPU云主机即可。

GPU云主机创建成功之后，登录GPU云主机。



登录页面如下所示：

```
运行前请初始化GPT环境：conda activate gpt
微调：cd /opt/alpaca-lora-main && python finetune.py \
    --base_model '/opt/llama-7b-hf' \
    --data_path 'yahma/alpaca-cleaned' \
    --output_dir './lora-alpaca'
推理：cd /opt/alpaca-lora-main && python generate.py \
    --base_model '/opt/llama-7b-hf' \
    --lora_weights 'tloen/alpaca-lora-7b' 并用网页浏览终端显示的Public URL进行交互
<<<<<<<<<<<<<<<<<<<<<<<<<

Activate the web console with: systemctl enable --now cockpit.socket

Last failed login: Thu May 18 21:57:06 CST 2023 from 20.189.122.249 on ssh:notty
There were 6 failed login attempts since the last successful login.
Last login: Thu May 18 21:56:47 2023 from 10.13.253.51
```

我们预装的镜像提供如下信息：

1.微调

2.推理

以推理为例，展示页面如下：

Alpaca-LoRA

Alpaca-LoRA is a 7B-parameter LLaMA model finetuned to follow instructions. It is trained on the Stanford Alpaca dataset and makes use of the Huggingface LLaMA implementation. For more information, please visit the project's website.

Instruction

how to reduce the air pollution

Input

none

Temperature 0.1

Top p 0.75

Top k 40

Beams 4

Output

The best way to reduce air pollution is to reduce the use of fossil fuels. This can be done by switching to renewable sources of energy, such as solar and wind power. Additionally, it is important to reduce emissions from transportation, such as by using public transportation or carpooling. Finally, it is important to reduce emissions from industrial processes, such as by investing in cleaner technologies.

Instruction:
how to reduce the water pollution

Flag

LLaMA2 模型快速部署

简介

LLaMA2模型是经过了更广泛和深入的训练，具有更多的标记数量和更长的上下文长度。相较于LLaMA 1，LLaMA2 接受了 2 万亿个标记的训练，并且其上下文长度是LLaMA1 的两倍。除此之外，LLaMA-2-chat 模型还经过了超过 100 万个新的人类注释的训练。LLaMA2模型的训练语料比LLaMA 1 更为丰富，多出了40%的数据。其上下文长度由之前的2048升级到4096，这使得它可以理解和生成更长的文本，为更复杂的任务提供了更好的支持。预训练数据中的语言分布，百分比大于等于 0.005%，但其中大部分数据都是英文，因此 LLaMA2 在英语用例中表现最佳。如果想要在中文场景中使用LLaMA2 进行文案策划，还需要进行中文增强训练，以使其在处理中文文本时表现更加出色。

快速部署

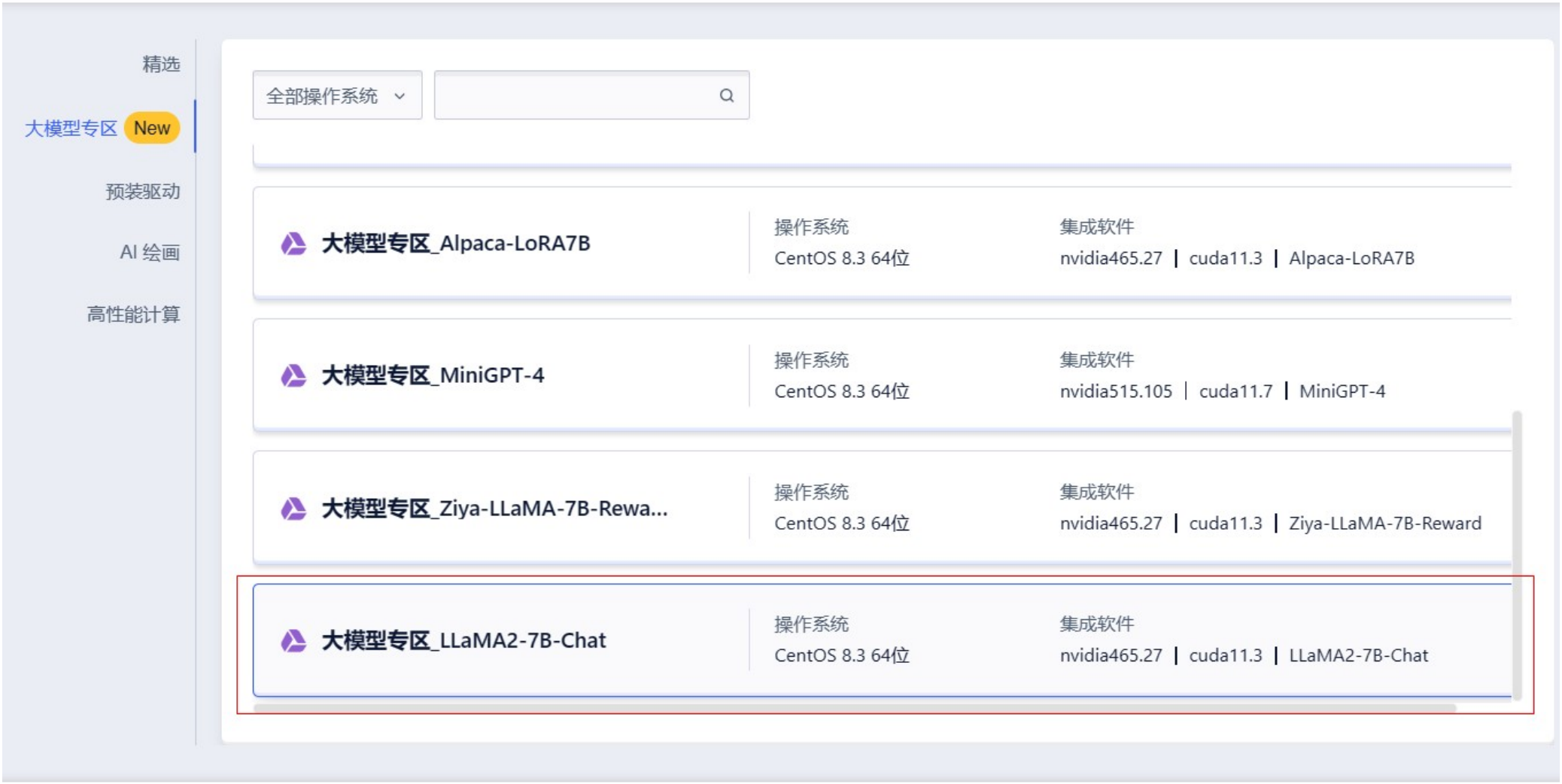
登录UCloud控制台（https://console.ucloud.cn/uhost/uhost/create ），机型选择“GPU型”，“V100S”，CPU及GPU颗数等详细配置按需选择。

最低推荐配置：10核CPU 64G内存 1颗V100S。

镜像选择“镜像市场”，镜像名称搜索“LLaMA2”，选择该镜像创建GPU云主机即可。

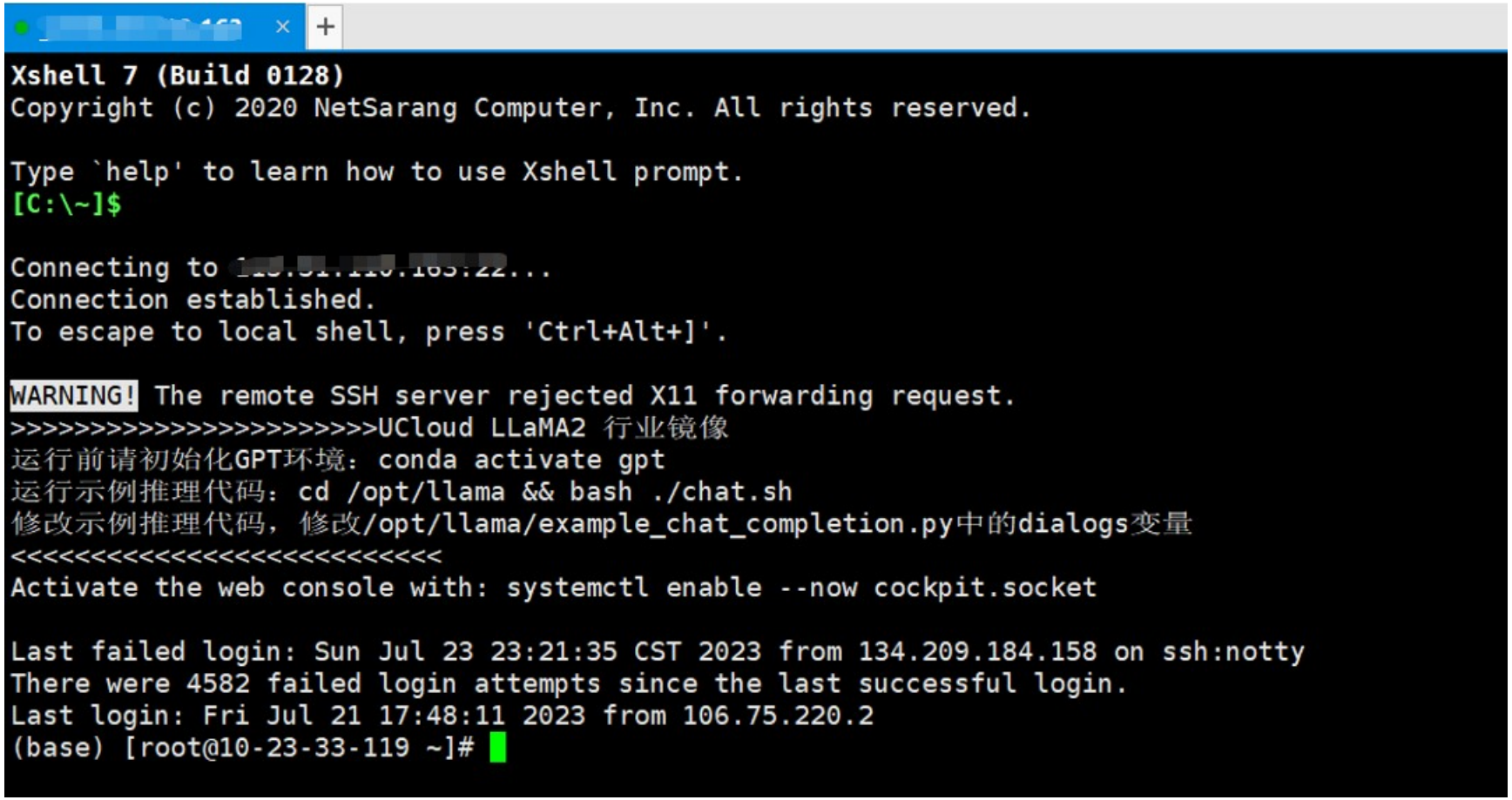
GPU云主机创建成功之后，登录GPU云主机。

镜像市场



取消 确定

登录页面如下所示：



T5

简介

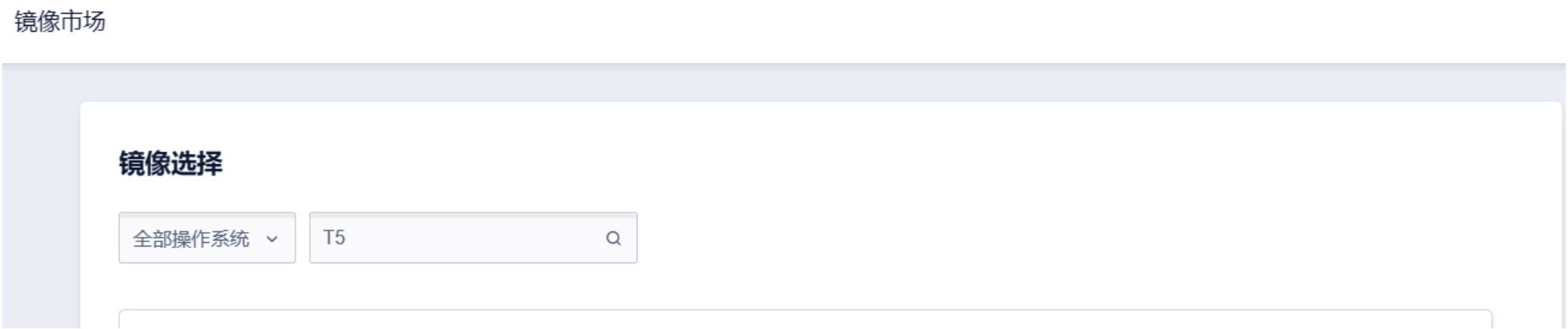
T5是谷歌发布的NLP 大语言模型，即Text-To-Text Transfer Transformer。UCloud镜像市场分别提供了T5-Base，T5-3B两个模型的镜像。

快速部署

登录UCloud控制台（https://console.ucloud.cn/uhost/uhost/create ），机型选择“GPU型”，“V100S”，CPU及GPU颗数等详细配置按需选择。最低推荐配置：T5-Base 10核CPU 64G内存 1颗V100S。

T5-3B 20核CPU 128G内存 2颗V100S 镜像选择“镜像市场”，镜像名称搜索“T5”，选择该镜像创建GPU云主机即可。

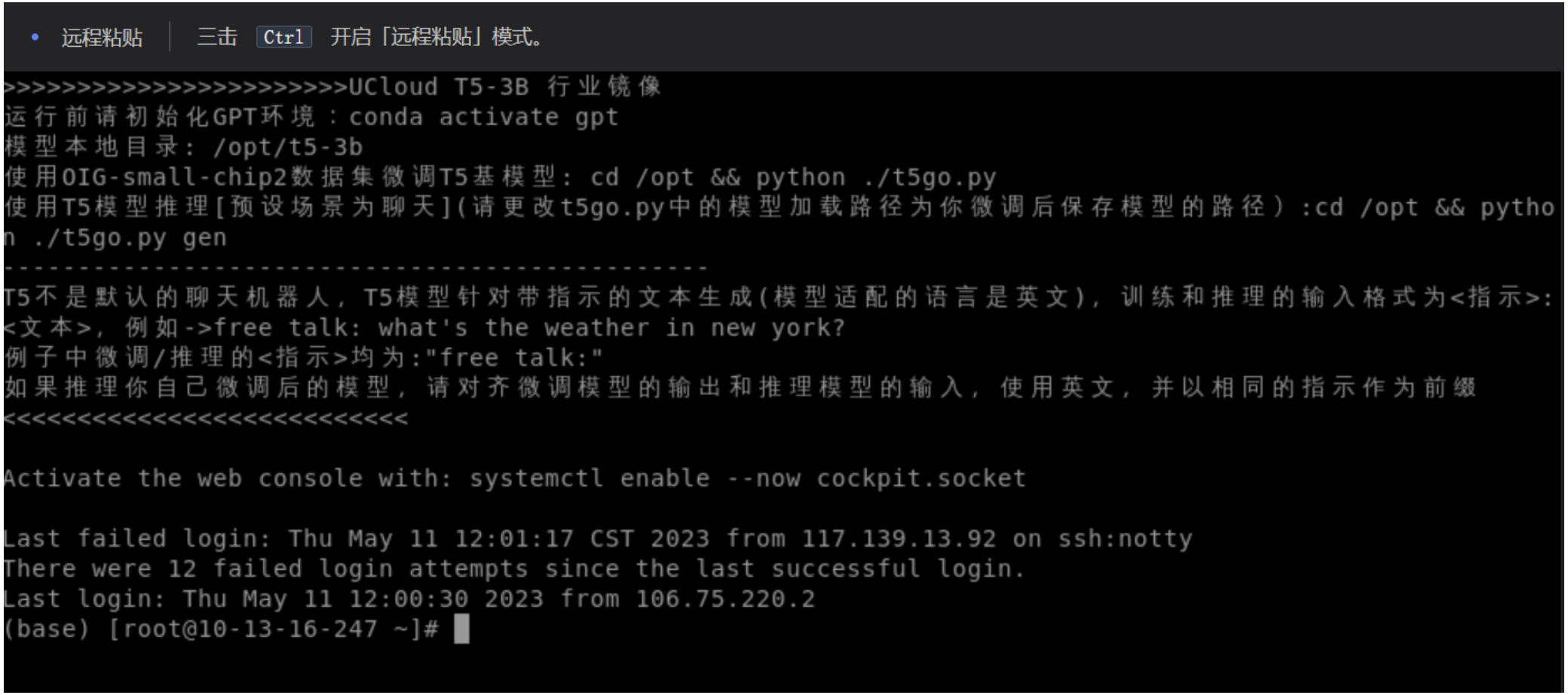
GPU云主机创建成功之后，登录GPU云主机。



	操作系统	集成软件
T5-Base	CentOS 8.3 64位	nvidia465.27 | cuda11.3 | T5-Base
T5-3B	CentOS 8.3 64位	nvidia465.27 | cuda11.3 | T5-3B

取消 确定

登录页面如下所示：



我们预装的镜像提供如下信息：

1.微调

2.推理

MiniGPT-4模型快速部署

简介

MiniGPT-4是基于GPT-3.5的小型语言模型，在多个领域展现了其强大的潜力。作为多模态模型，MiniGPT-4能够理解和处理不同模态之间的关联性，从而为更丰富的创作和应用提供支持。通过将图像、文本和音频等多种形式的数据结合在一起，MiniGPT-4可以生成与输入数据相关的多模态输出。无论是创意写作、故事构思、诗歌创作还是市场营销文案，MiniGPT-4都能为您提供灵感和支持。

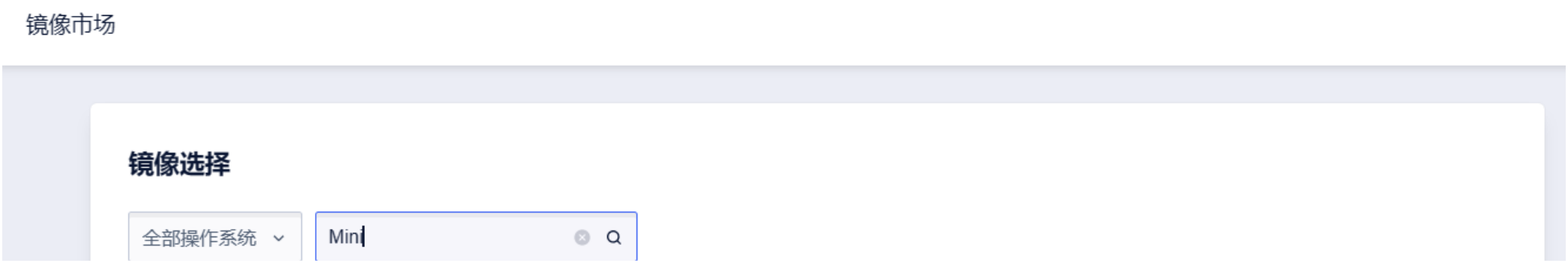
快速部署

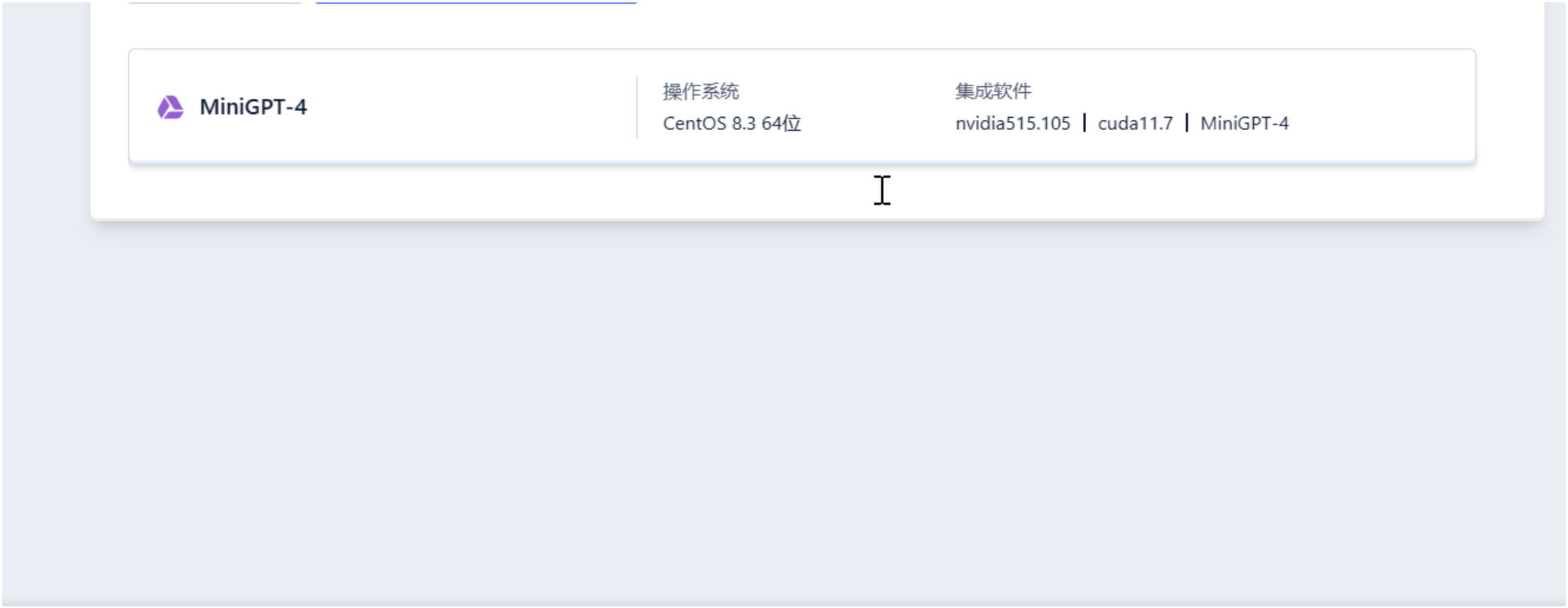
登录UCloud控制台（https://console.ucloud.cn/uhost/uhost/create ），机型选择“GPU型”，“V100S”，CPU及GPU颗数等详细配置按需选择。

最低推荐配置：10核CPU 64G内存 1颗V100S。

镜像选择“镜像市场”，镜像名称搜索“MiniGPT-4”，选择该镜像创建GPU云主机即可。

GPU云主机创建成功之后，登录GPU云主机。





登录页面如下所示：

```
WARNING! The remote SSH server rejected X11 forwarding request.
------------------------------------------------------------------------------
温馨提示：
当前镜像已预装nvidia-driver,cuda,conda,MiniGPT-4。
MiniGPT-4启动方式：
cd /MiniGPT-4/MiniGPT-4
conda activate minigpt4
python demo.py --cfg-path eval_configs/minigpt4_eval.yaml  --gpu-id 0
启动好之后，通过浏览器访问给出的地址即可。
------------------------------------------------------------------------------
Activate the web console with: systemctl enable --now cockpit.socket

Last failed login: Wed May 24 19:43:04 CST 2023 from 221.215.21.91 on ssh:notty
There were 2 failed login attempts since the last successful login.
Last login: Wed May 24 19:40:18 2023 from 10.13.253.52
(base) [root@10-13-79-50 ~]#
```

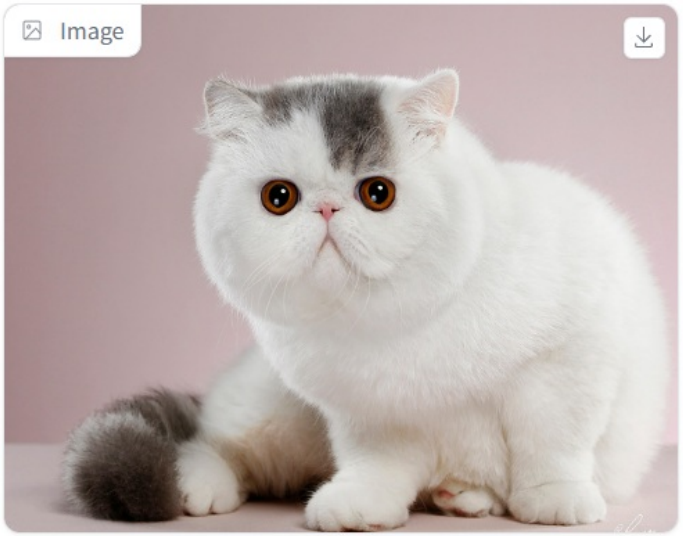
执行进程之后，Demo展示如下：

Demo of MiniGPT-4

This is the demo of MiniGPT-4. Upload your images and start chatting!

Project Page
Github Code
Paper PDF

Image



Start Chatting

Restart

beam search numbers) 1

MiniGPT-4

wirte a story based on the picture， approximately 150 words.

The white cat has a calm and serene expression on its face. Its eyes are brown and wide open, looking directly at the camera. The cat's fur is fluffy and well-groomed, and it looks healthy and well-fed. The cat's nose is pink, and its paws are small and neat. The background is a soft pink shade.

User

Type and press Enter

Ziya模型快速部署

简介

Ziya模型是由IDEA开发的一款领先的自然语言处理模型。基于LLaMa的130亿参数的大规模预训练模型，具备翻译、编程、文本分类、信息抽取、摘要、文案生成、常识问答和数学计算等能力。目前“Ziya”模型已完成大规模预训练、多任务有监督微调和人类反馈学习三阶段的训练过程。

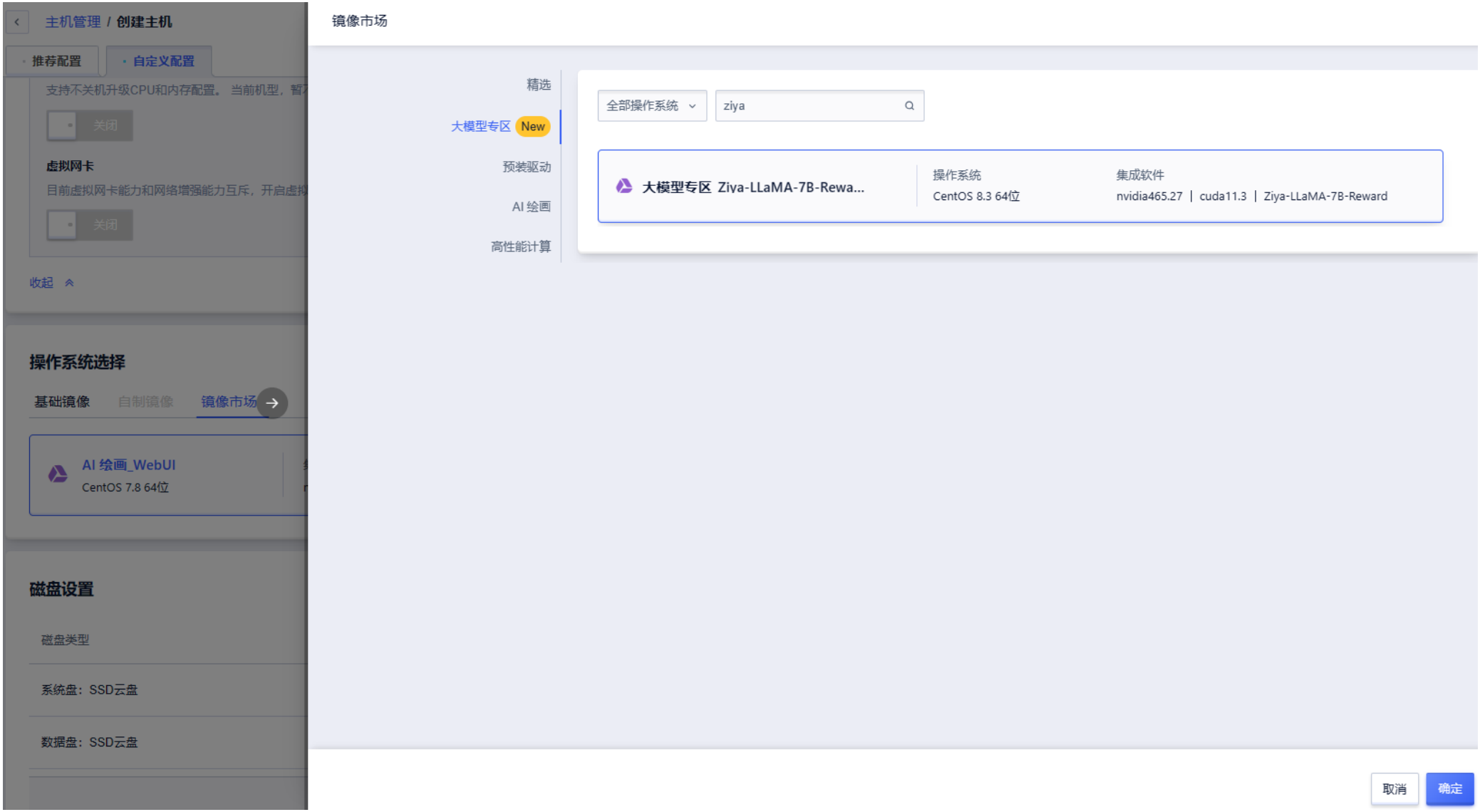
快速部署

登录UCloud控制台（https://console.ucloud.cn/uhost/uhost/create ），机型选择“GPU型”，“V100S”，CPU及GPU颗数等详细配置按需选择。

最低推荐配置：10核CPU 64G内存 1颗V100S。

镜像选择“镜像市场”，镜像名称搜索“Ziya”，选择该镜像创建GPU云主机即可。

GPU云主机创建成功之后，登录GPU云主机。



登录页面如下所示：

LLaMA-Factory快速部署

简介

LLaMA-Factory 是一个涵盖预训练、指令微调到 RLHF 阶段的开源全栈大模型微调框架，具备高效、易用、可扩展的优点，配备有零代码可视化的一站式网页微调界面 LLaMA Board。同时包含预训练、监督微调、RLHF等多种训练方法，支持 0-1 复现 ChatGPT 训练流程，丰富的中英文参数提示，实时的状态监控和简洁的模型断点管理，支持网页重连和刷新。

快速部署

登录UCloud控制台（https://console.ucloud.cn/uhost/uhost/create ），机型选择“GPU型”，“高性价比显卡6”，CPU及GPU颗数等详细配置按需选择。

最低推荐配置：16核CPU 64G内存 1颗GPU。

镜像选择“镜像市场”，镜像名称搜索“LLaMA-Factory”，选择该镜像创建GPU云主机即可。

GPU云主机创建成功之后，登录GPU云主机。

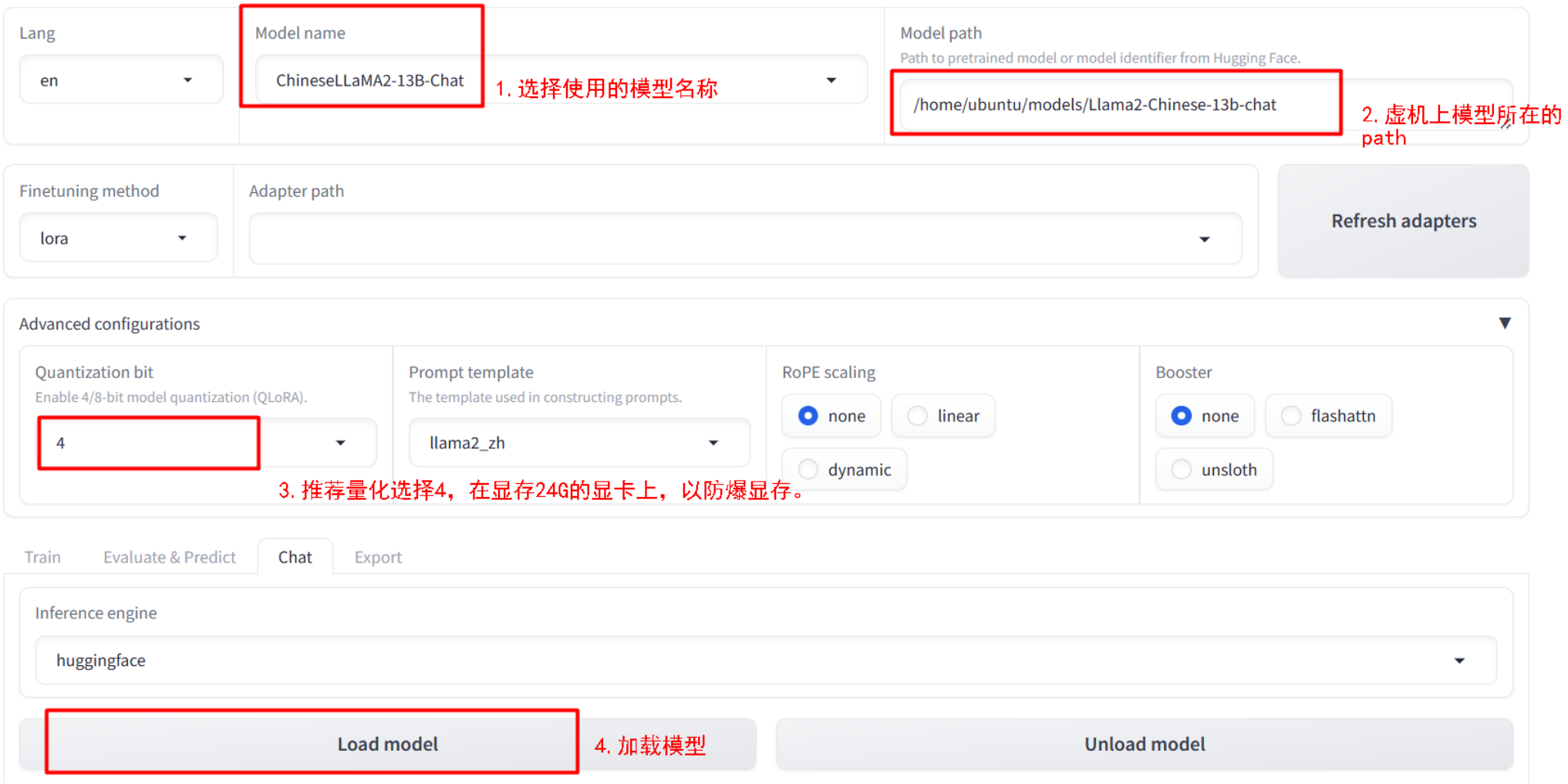
登录页面如下所示：

```
--------------------------------------------------------------------------------
温馨提示：
当前镜像已预装nvidia-driver,cuda,conda,LLaMA-Factory。
LLaMA-Factory启动方式：
conda activate llama_factory
cd /home/ubuntu/LLaMA-Factory
export CUDA_VISIBLE_DEVICES=0
nohup python src/train_web.py > train_web.log 2>&1 &
启动好之后，通过浏览器访问：http://ip:7860即可，注意ip需要替换为云主机的外网ip，
如果无法访问请在控制台检查安全规则是否放行上述端口。
--------------------------------------------------------------------------------
/usr/bin/xauth:  file /home/ubuntu/.Xauthority does not exist
(base) ubuntu@192-168-1-83:~$
```

操作实践

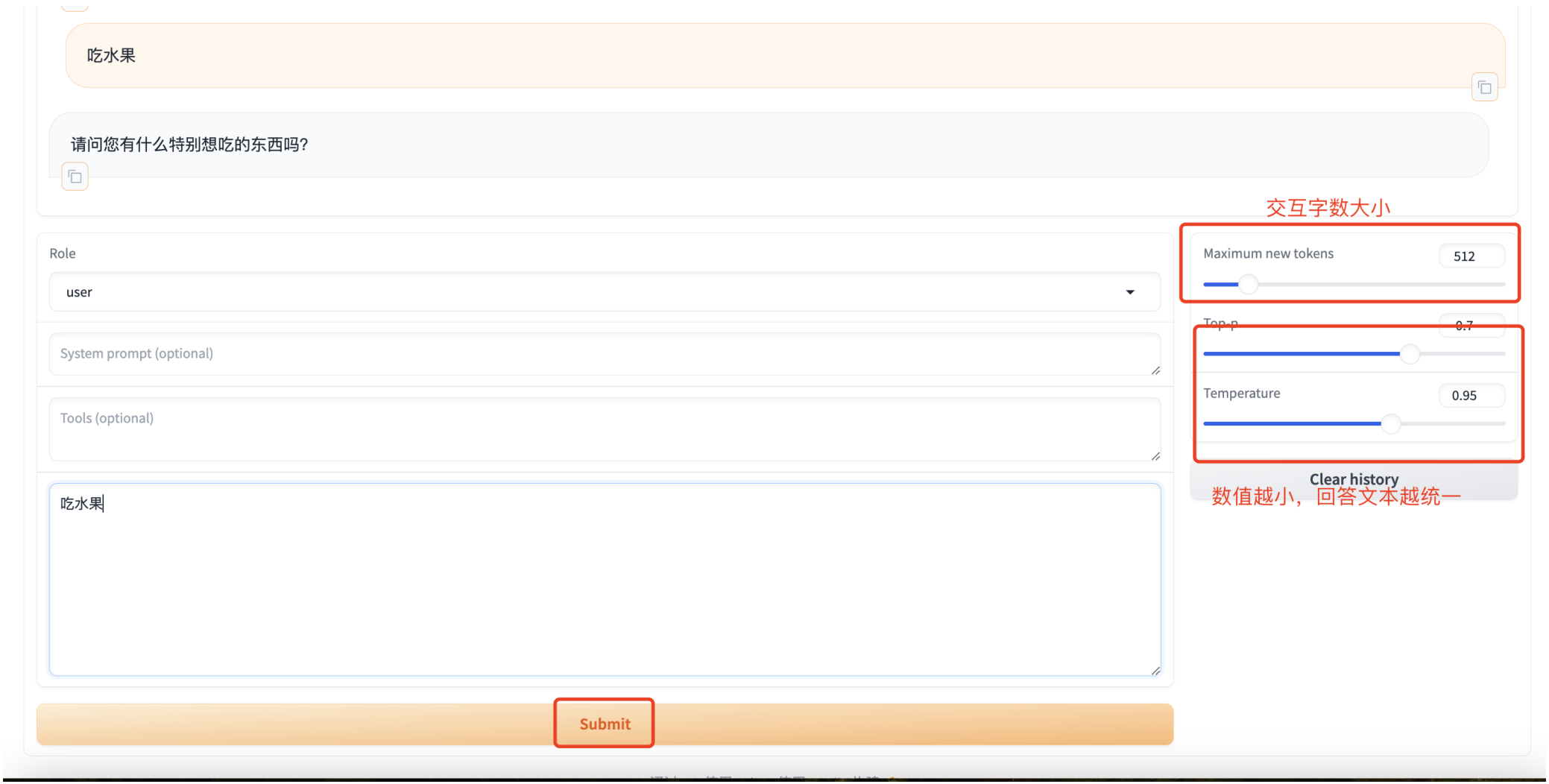
通过浏览器访问：http://ip:7860 即可，注意ip需要替换为云主机的外网ip，如果无法访问请在控制台检查安全规则是否放行上述端口

1.加载装置

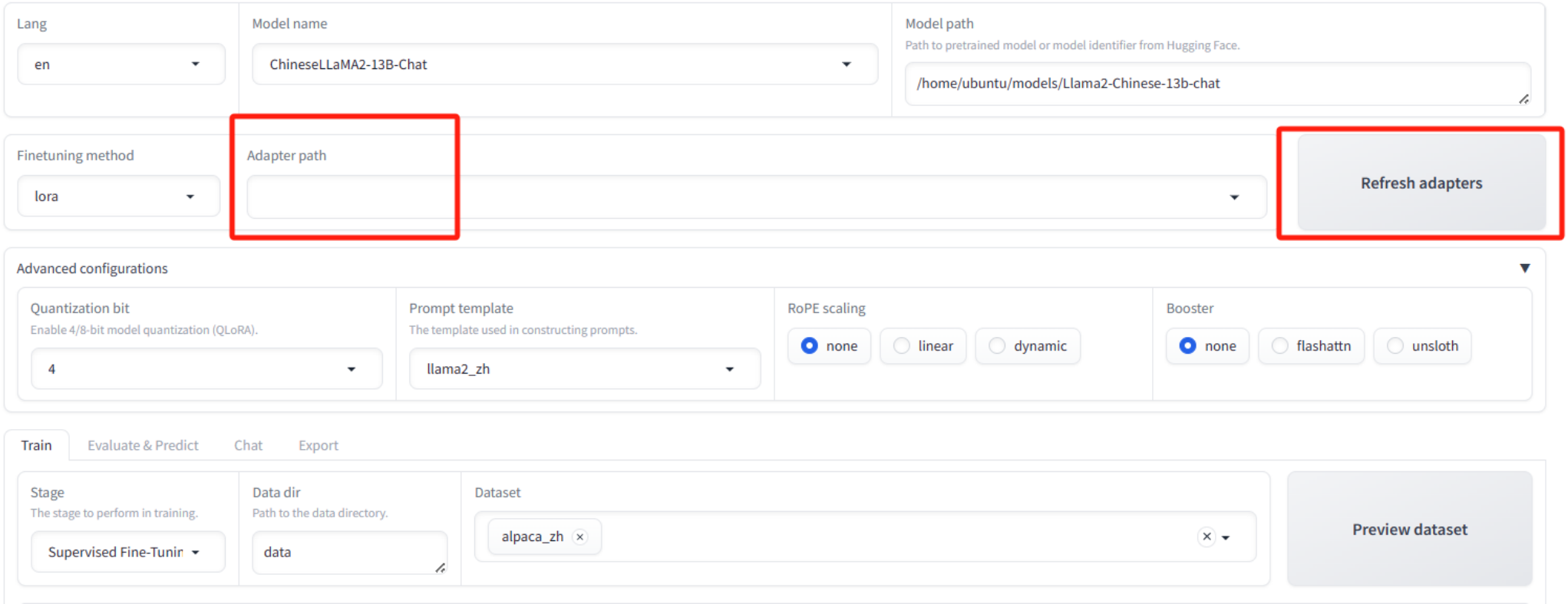


注:1、2 的模型必须一一对应,2的path为虚机本地模型下载的path。

2.操作



3.训练参数

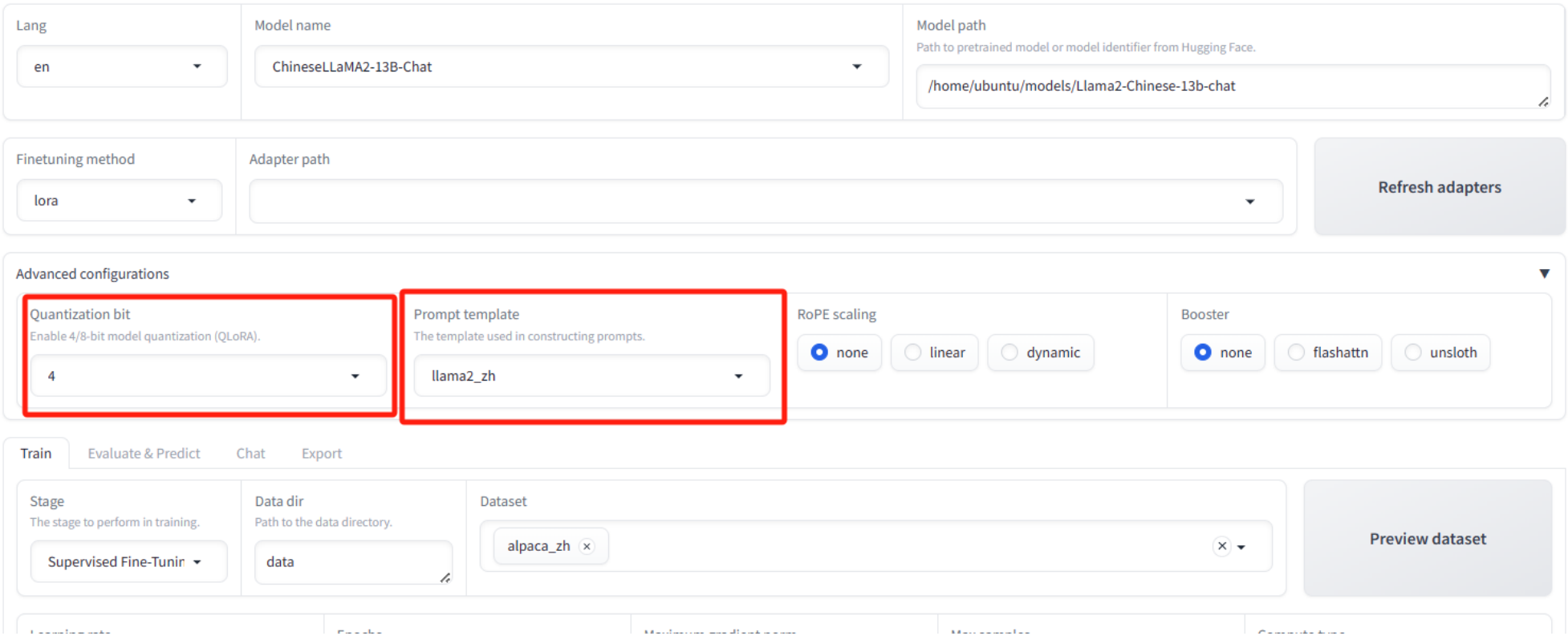


使用web ui训练后可以点refresh adapters刷新本地Lora权重，从adapter path指定训练好的Lora权重。

选定权重后，训练会从训练好的Lora权重继续训练。

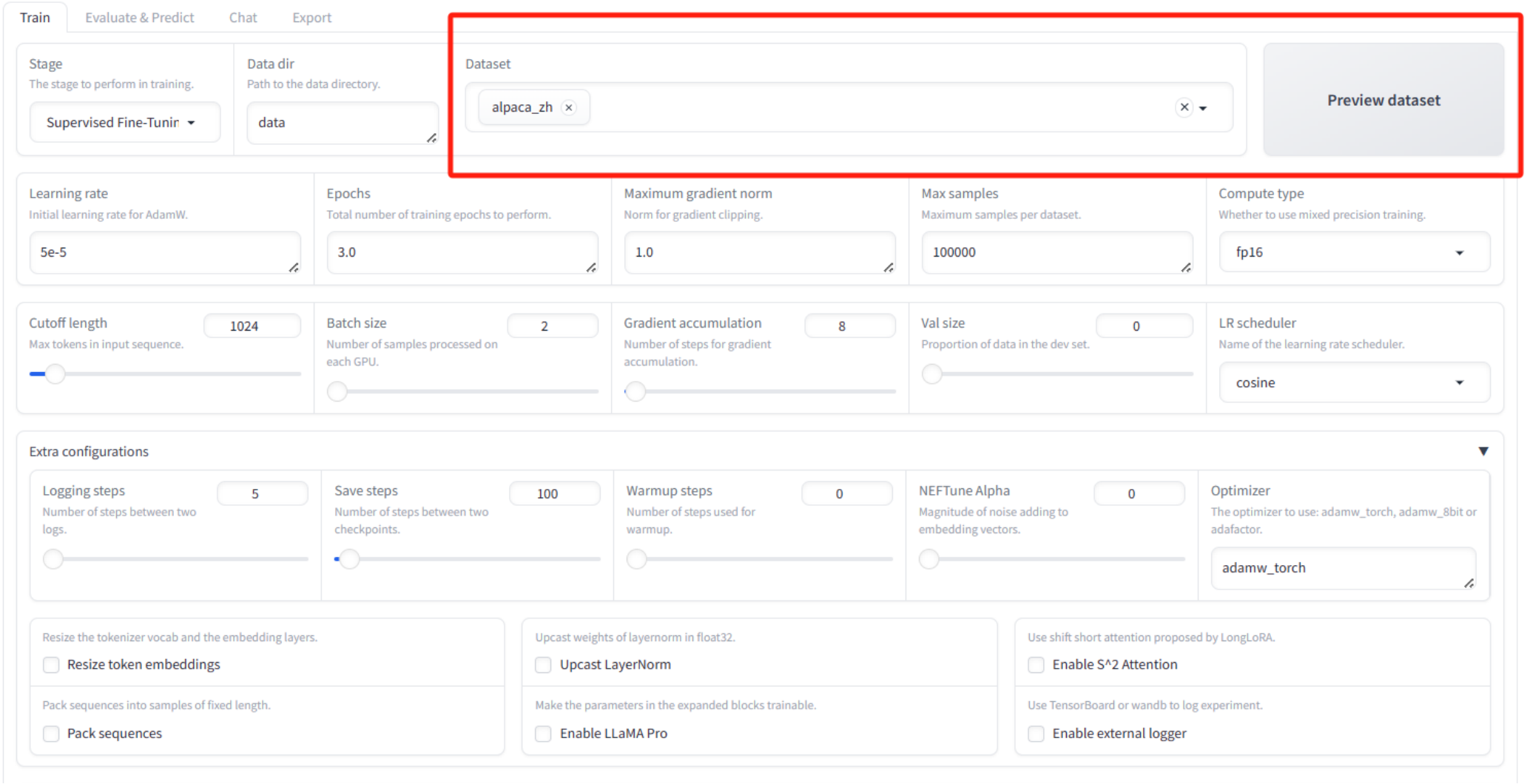
选定权重后，对话会和训练后的Lora模型进行对话。

注：由于webui有路径规则，所以只能识别由webui训练的Lora权重，无法识别命令行训练的权重。



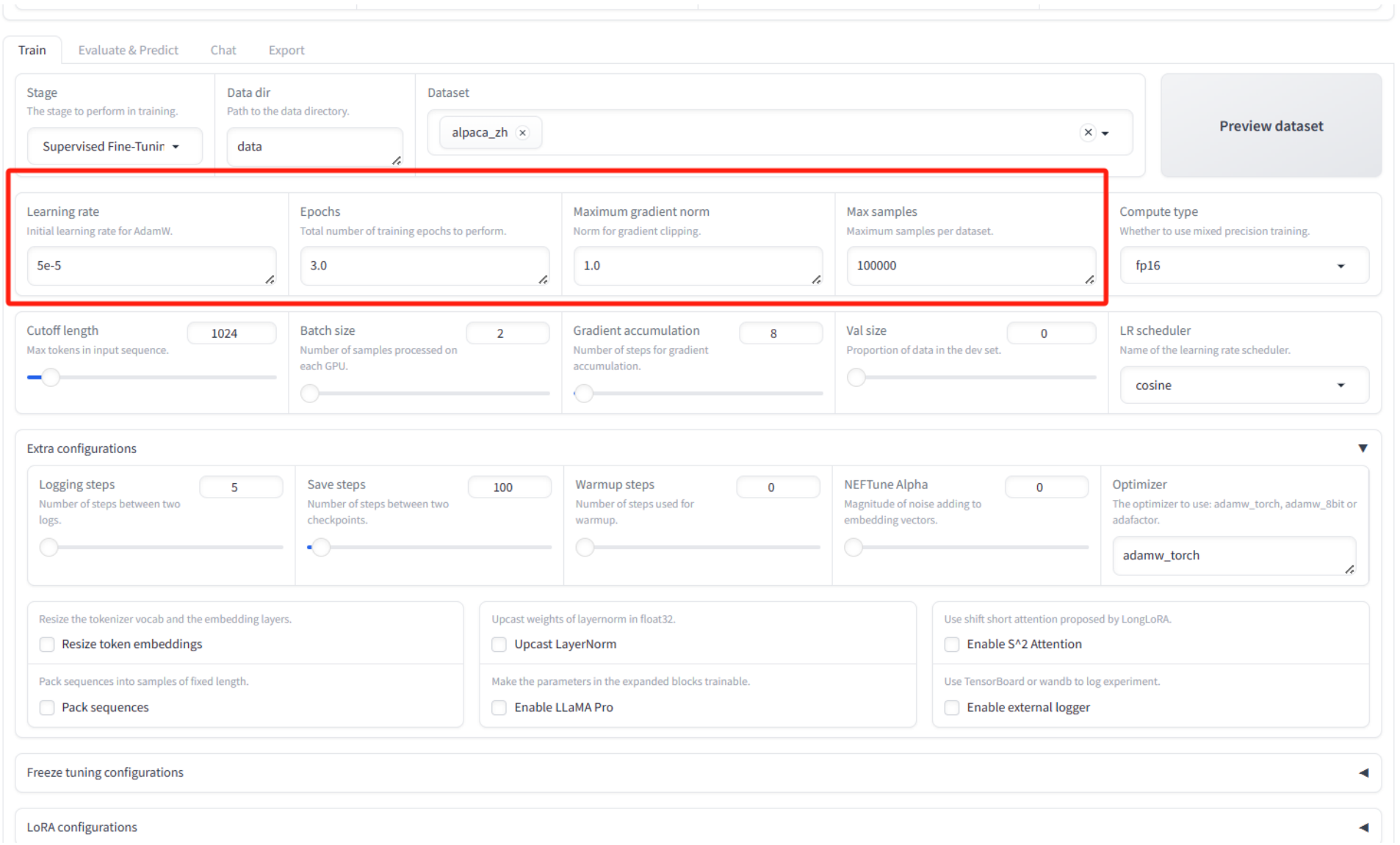
Quantization bit 需要选择4，否则当前推荐机型的24G显存吃紧。

Prompt template 是和对应模型绑定的，在上面选择model name后会自动带出，不用调整。



在图中位置选择训练数据集，点击preview dataset可以预览训练数据。

如果需要添加自定义数据集，请参考 /home/ubuntu/LLaMA-Factory/data/README.md

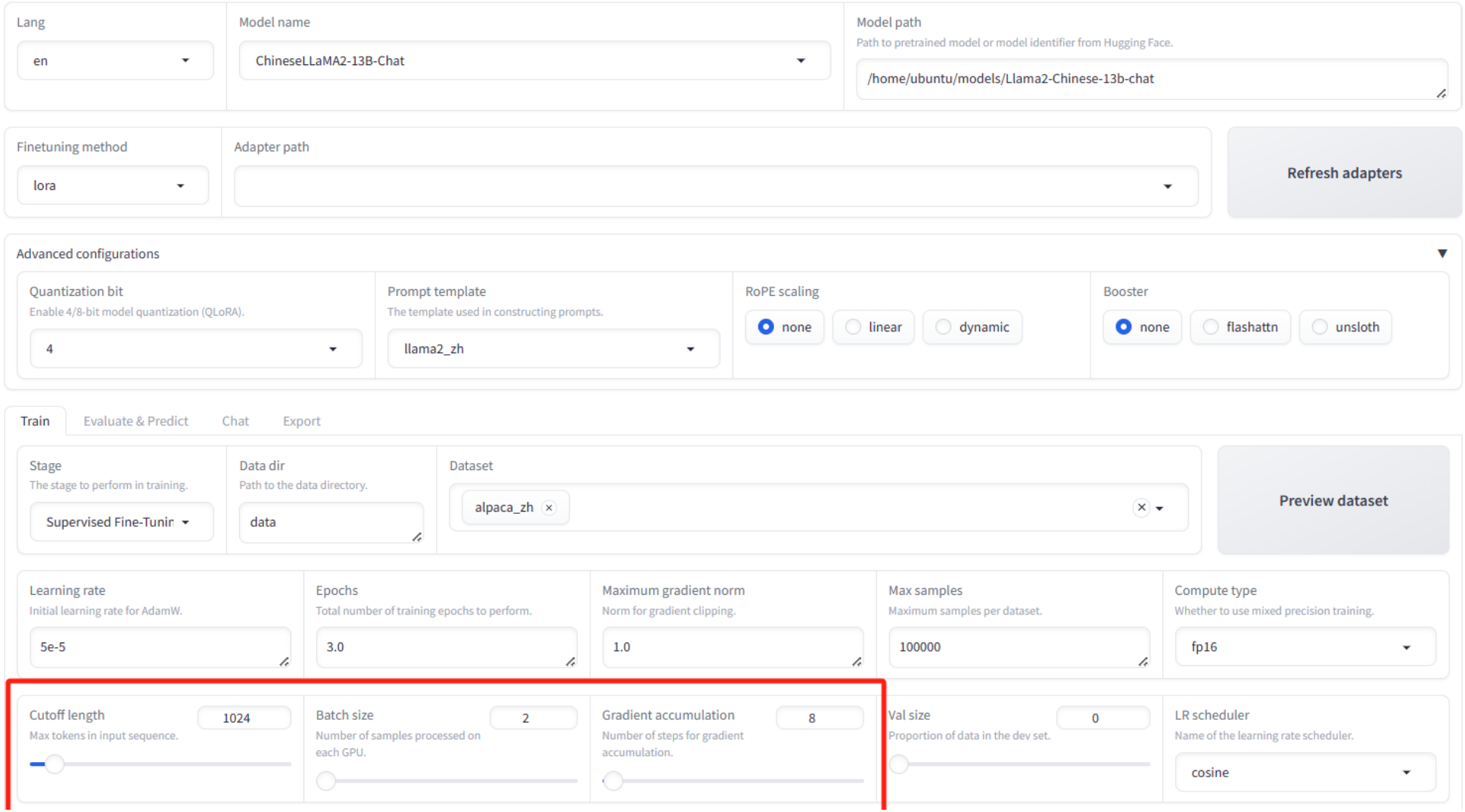


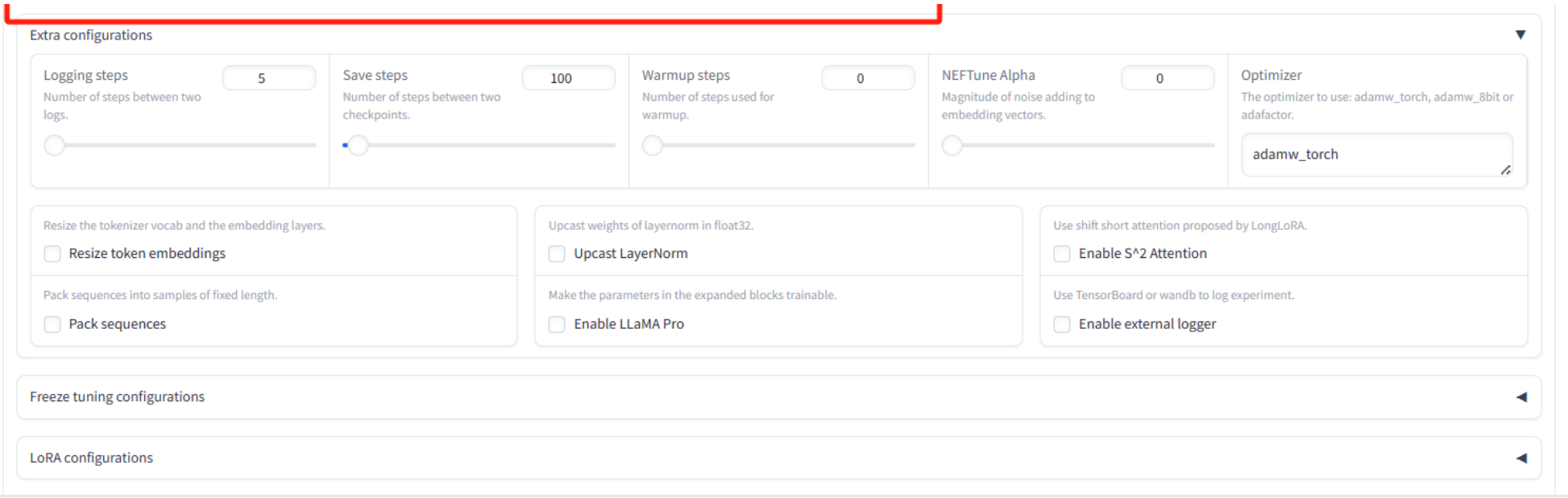
Learning rate 在 1e-5到5e-4间(不改optimizer的情况)。带有玄学性质,在刷分时可以多试几个学习率。

Epochs 根据loss下降图像调整，小于1万条的数据集可以从epochs=3开始试，如果loss图像尾部没变平缓就往上调。超过6万条文本（1条平均20字以上）的数据集在epochs=1至3之间。

Maximum gradient norm 推荐值在1-10之间。

Max samples 是限定使用数据集的前多少条，一般是全量训练，测试时可以设值为10避免验证时间过长。

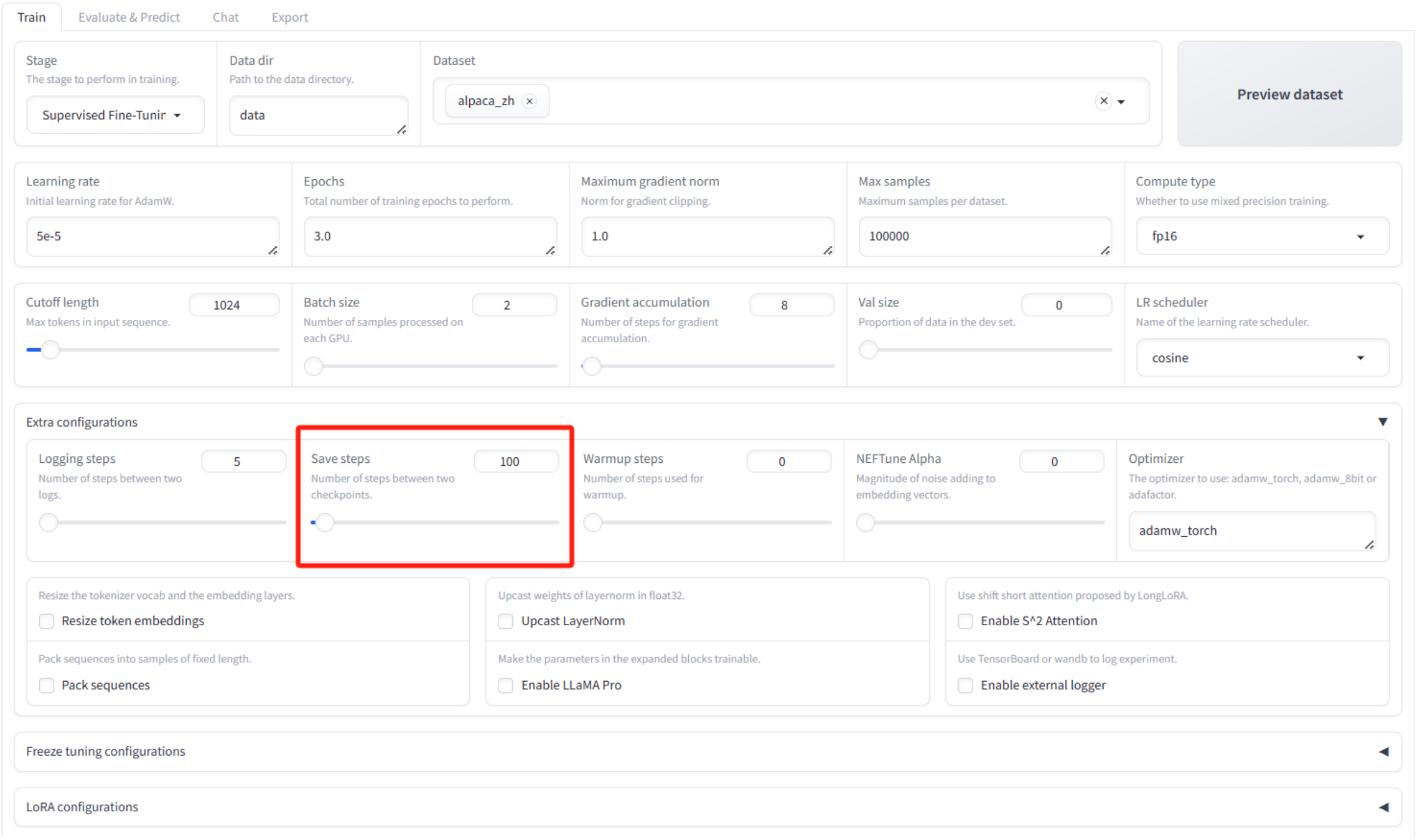




Cutoff length 是训练句子截断长度，句子越长，显存占用越多，如果显存不够可以考虑降低到512甚至256。可以根据微调目标需要的长度进行设置。微调后，模型处理长度大于cut off length的句子的能力会下降。

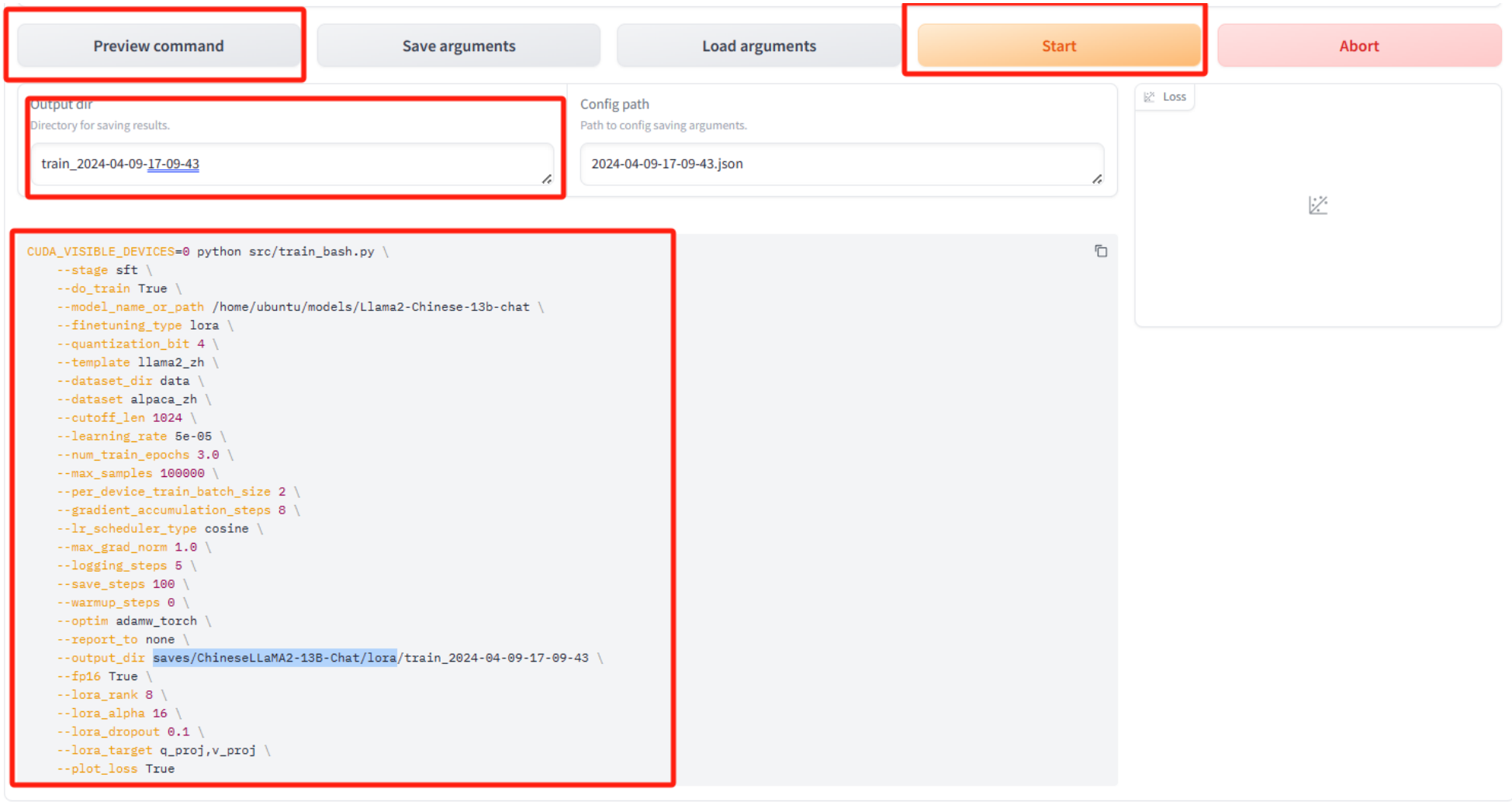
Batch size 会影响模型的训练质量，训练速度以及显存。考虑训练质量方面，一般Batch size * Gradient accumulation 的数值在16到64之间，这个乘积是最终的等效batch size。考虑训练速度，batch size在4090的可观测范围内越大越快。考虑显存，batch size =4 是极限，放不下就调低。

句子很长的情况下，如2048，可以batch size=1, gradient accumulation=16。gradient accumulation不影响显存，但是影响等效batch size，进而影响训练质量。



Save steps 是每多少step存一次模型断点，方便在训练意外断掉的情况继续训练。总step数=epochs*数据集条数/batch size / gradient accumulation，这在训练时有个进度条也会显示。由于每次储存的断点都是模型的全部权重，所以一个断点就是存几十个G。

建议调高至1000以上，否则硬盘容量会很容易不足。每次训练完后，可以到/home/ubuntu/LLaMA-Factory/saves/ChineseLLaMA2-13B-Chat/lora/... 下进行删除。



可以通过Preview command 得到可以在命令行运行的等效命令，Output dir 是训练结果以及断点的存放路径。设置好参数后可以直接点start进行训练。也可以复制命令在/home/ubuntu/LLaMA-Factory 下，conda的llama_factory环境下运行。

Llama3-8B-Instruct-Chinese 快速部署

简介

Llama3 由 Meta 是在15万亿tokens数据集上训练的，是 Llama2的7倍，包括4倍的代码数据。其中预训练数据集中还有5%的非英文数据集，总共支持的语言高达30种，在做其他语言能力对齐方面也会更有优势。Llama 3 Instruct 更是针对对话应用进行了优化，结合了超过 1000 万的人工标注数据，通过监督式微调（SFT）、拒绝采样、邻近策略优化（PPO）和直接策略优化（DPO）进行训练。本文提及的模型即为基于中文语料指令微调之后的模型（Llama3-8B-Instruct-Chinese），在中文表现上有相对不错的效果。

快速部署

登录UCloud控制台（https://console.ucloud.cn/uhost/uhost/create ），机型选择“GPU型”，“高性价比显卡6”，CPU及GPU颗数等详细配置按需选择。

最低推荐配置：16核CPU 32G内存 1颗高性价比显卡6 GPU。

镜像选择“镜像市场”，镜像名称搜索“Llama3”，选择该镜像创建GPU云主机即可。

GPU云主机创建成功之后，登录GPU云主机。

	CPU平台	CPU型号	GPU	CPU	内存	显存	性能
◉	AMD (x86_64)	自动分配	1颗	16核	32 G	24 G	83 TFlops
○	AMD (x86_64)	自动分配	1颗	16核	64 G	24 G	83 TFlops
○	AMD (x86_64)	自动分配	2颗	32核	64 G	24 G	83 TFlops
○	AMD (x86_64)	自动分配	2颗	32核	128 G	24 G	83 TFlops
○	AMD (x86_64)	自动分配	4颗	64核	128 G	24 G	83 TFlops
○	AMD (x86_64)	自动分配	4颗	64核	256 G	24 G	83 TFlops
○	AMD (x86_64)	自动分配	4颗	92核	470 G	24 G	83 TFlops
○	AMD (x86_64)	自动分配	8颗	92核	440 G	24 G	83 TFlops
○	AMD (x86_64)	自动分配	8颗	124核	440 G	24 G	83 TFlops
○	AMD (x86_64)	自动分配	8颗	124核	512 G	24 G	83 TFlops

地域可用区

可用区根据所选的机型、镜像、配置进行过滤。

华北二可用区A

操作系统选择

镜像市场 基础镜像 自制镜像

大模型专区_Llama3-8B-In...
Ubuntu 22.04 64位

集成软件
nvidia535.146.02 | cuda12.2 | Llama3-8B-Instn

更新时间
2024-04-23 05:25:15

更多选择

登录页面如下所示：

```
Connection established.
To escape to local shell, press 'Ctrl+Alt+]'.

Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-43-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Tue Apr 23 03:13:00 PM CST 2024

  System load:  0.2099609375      Processes:             240
  Usage of /:   56.6% of 98.25GB  Users logged in:       1
  Memory usage: 1%                IPv4 address for eth0: 192.168.2.133
  Swap usage:   0%


227 updates can be applied immediately.
148 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable


--------------------------------------------------------------------------------
温馨提示：
当前镜像已预装nvidia-driver,cuda,conda,Llama3-8B-Instruct-Chinese。
Llama3-8B-Instruct-Chinese启动方式：
cd /home/ubuntu/llama3-Chinese-chat
conda activate llama3-chinese
nohup streamlit run deploy/web_streamlit_for_instruct.py model/llama-3-8b-Instruct-chinese --theme.base="light" 2>&1 &
启动好之后，通过浏览器访问：http://ip:8501
如果无法访问请在控制台检查安全规则是否放行上述端口。
--------------------------------------------------------------------------------
Last login: Tue Apr 23 14:59:14 2024 from 106.75.220.2
(base) ubuntu@192-168-2-133:~$
```

web操作页面如下所示：