

一、服务器连接

- 1 - 推荐一个云服务器平台：[恒源云](#)

M40-24G

4090-24G

4080-16G

4070 Ti-12G

3090 Ti-24G

4070-12G

4060 Ti-16G

3090-24G

3080 Ti-12G

3080-10G

3080-12G

4060-8G

3070-8G

3060 Ti-8G

3060Ti-8G

3060-12G

2080 Ti-11G

2070 SUPER-8G

2060 SUPER-8G

1080 Ti-12G

V100-32G

P40-24G

T4-16G

P4-8G

TITAN V-12G

TITAN X-12G

TITAN XP-12G

GPU数量

1

2

3

4

5

6

7

8

9

10

多机多卡需求请联系客服开通

GPU类型	地区	数量	显存	显卡驱动版本	最高CUDA版本	CPU 型号	CPU配置	内存	实例硬盘	网络	价格	到期时间	可用券
3090-24G	华北	1	24 GB	535.161.08	12.2	AMD EPYC 7542 32-Core Processor	16核	31.5G	系统盘: 20G 数据盘: 50GB SSD 不可扩容	U: 80 Mbps/s D: 480 Mbps/s	¥0.9/小时	2024-12-21	-

数据盘: 免费 50GB

可以看到 24G 显存的 3090 只需要 0.9 元/小时
不需要大显存的组也可以换成 3060 or 3080，也都很便宜

- 2 - 镜像的选择

PyTorch > 2.2.1 > 11.8.0 > 3.8

TensorFlow > 2.0.0 > 11.7.0 >

PaddlePaddle > 1.13.1 >

OneFlow > 1.13.0 >

OpenCV > 1.12.1 >

MXNet > 1.12.0 >

KataGo > 1.11.0 >

实例镜像

请选择框架名称/框架版本/Cuda版本/Python版本

没有需要的镜像?

CPU配置	内存	实例硬盘	网络	价格	到期时间	可用券
2 32-Core	16核	31.5G	系统盘: 20G 数据盘: 50GB SSD 不可扩容	U: 80 Mbps/s D: 480 Mbps/s	¥0.9/小时	2024-12-21

费用金额 ¥0.90/小时 计费说明

账户可用余额: ¥1.58

创建实例

BERT 是基于 PyTorch 构建的，所以我们在创建服务器时就可直接选择，推荐 PyTorch==2.0.0
Cuda==11.7.0 Python==3.8，比较 Stable
然后点击 创建实例 就可以了

- 3 - 服务器的连接

华东 Linux

创建中

3080-12G * 1 卡

系统磁盘

按量付费

2025-12-21

1fd1d1c298100c01ab5

查看详情

正常

等待创建完成，=>

华东 Linux

运行中

3080-12G * 1 卡

系统磁盘

按量付费 余额不足72小时

创建时间 2024-10-18 14:00:36

2025-12-21

1fd1d1c298100c01ab5

查看详情

数据盘 hy-tmp

0/50GB

登录指令

ssh***** 复制

密码 ***** 复制

JupyterLab

Tensorboard

实例监控

开启自定义服务?

实例管理

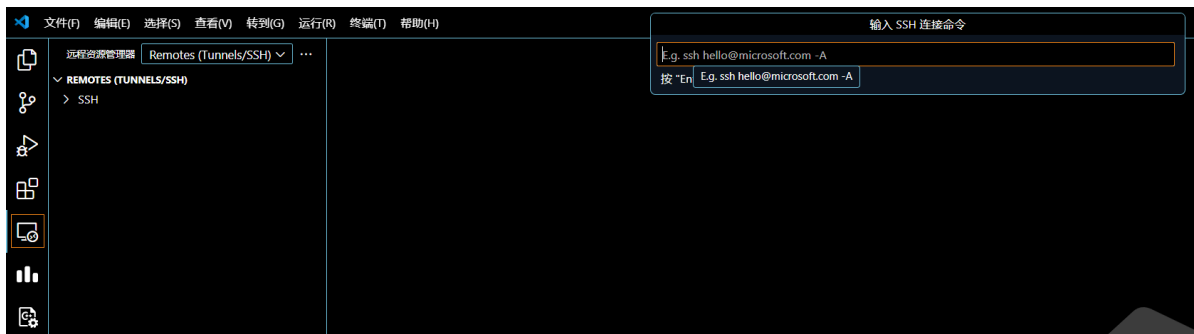
打开连接工具，首推VSCode，再推荐一个MobaXterm



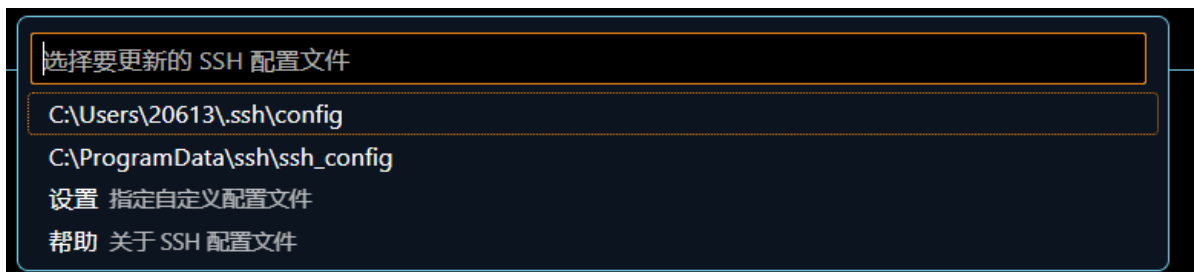
找到左侧的扩展，搜索 `remote ssh`，安装完成之后左侧会有一个新的图标，点击 `SSH` 右侧的 `+`



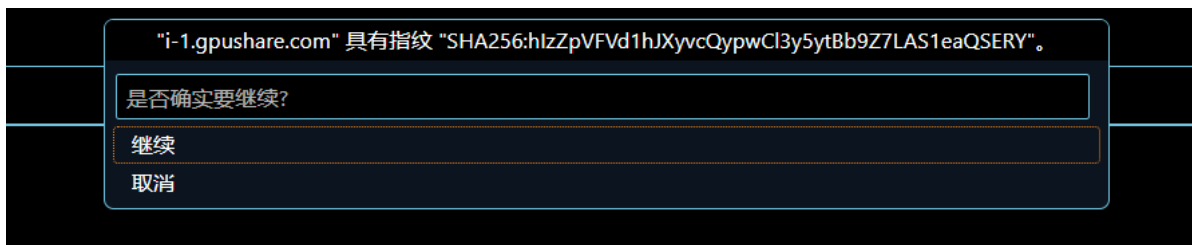
=>



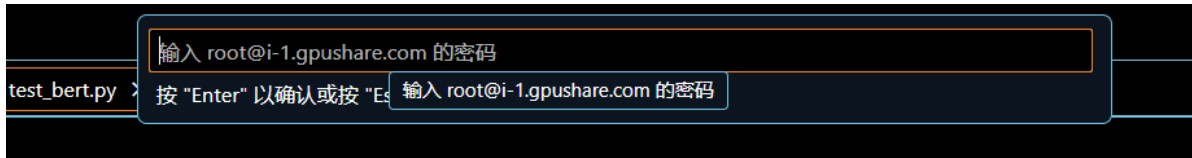
然后点击 `登陆指令` 的第一行 `ssh*****` 右侧的 `复制` 按钮，输入到VSCode的输入框中，按回车



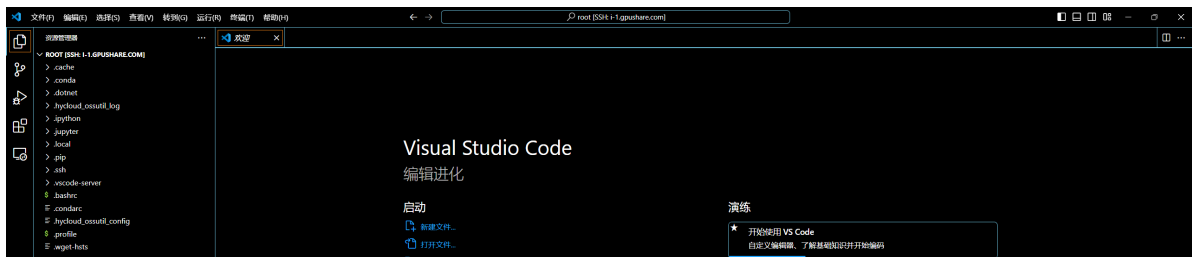
配置文件选第一个，右下角会出现 `已添加主机`，点击连接



有些可能会出现指纹提示，点 **继续**，有的可能会出现选平台，选 **Linux**

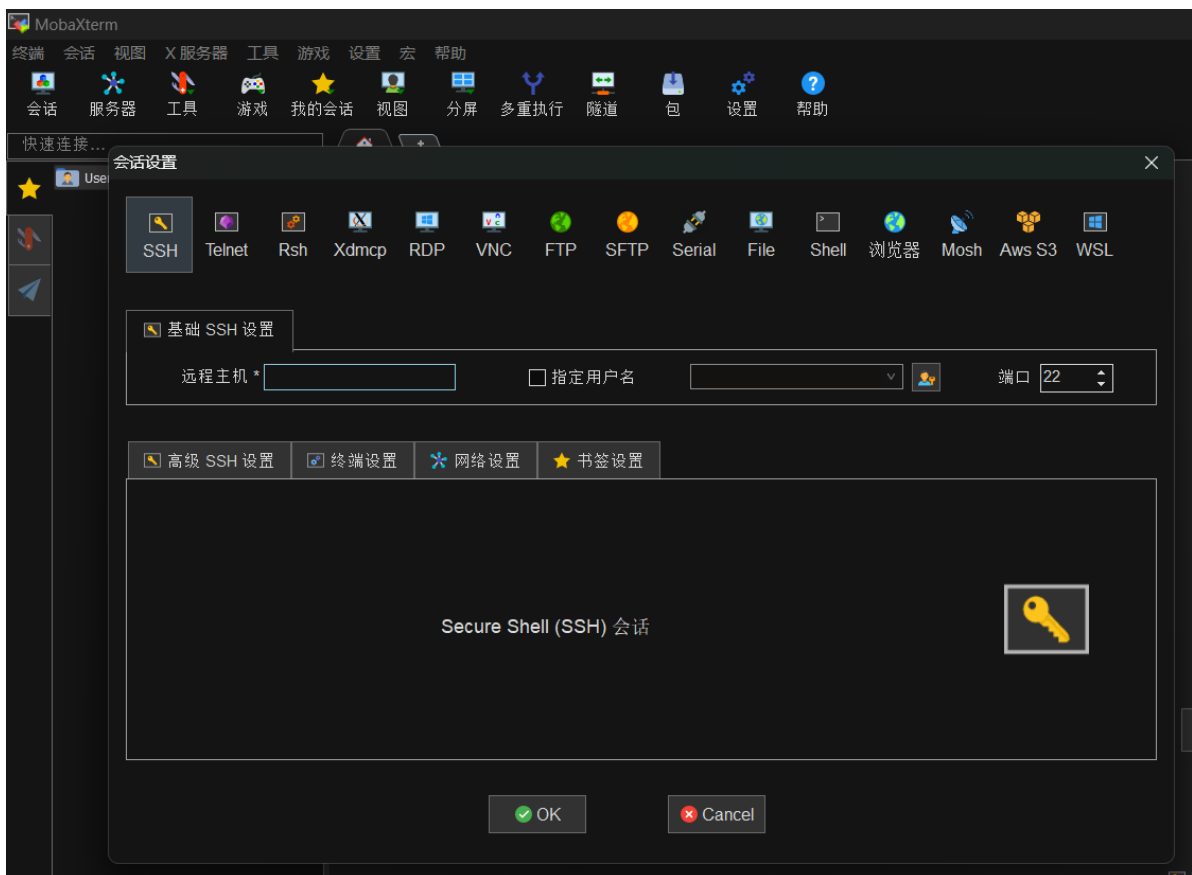


然后再将密码复制过来输入

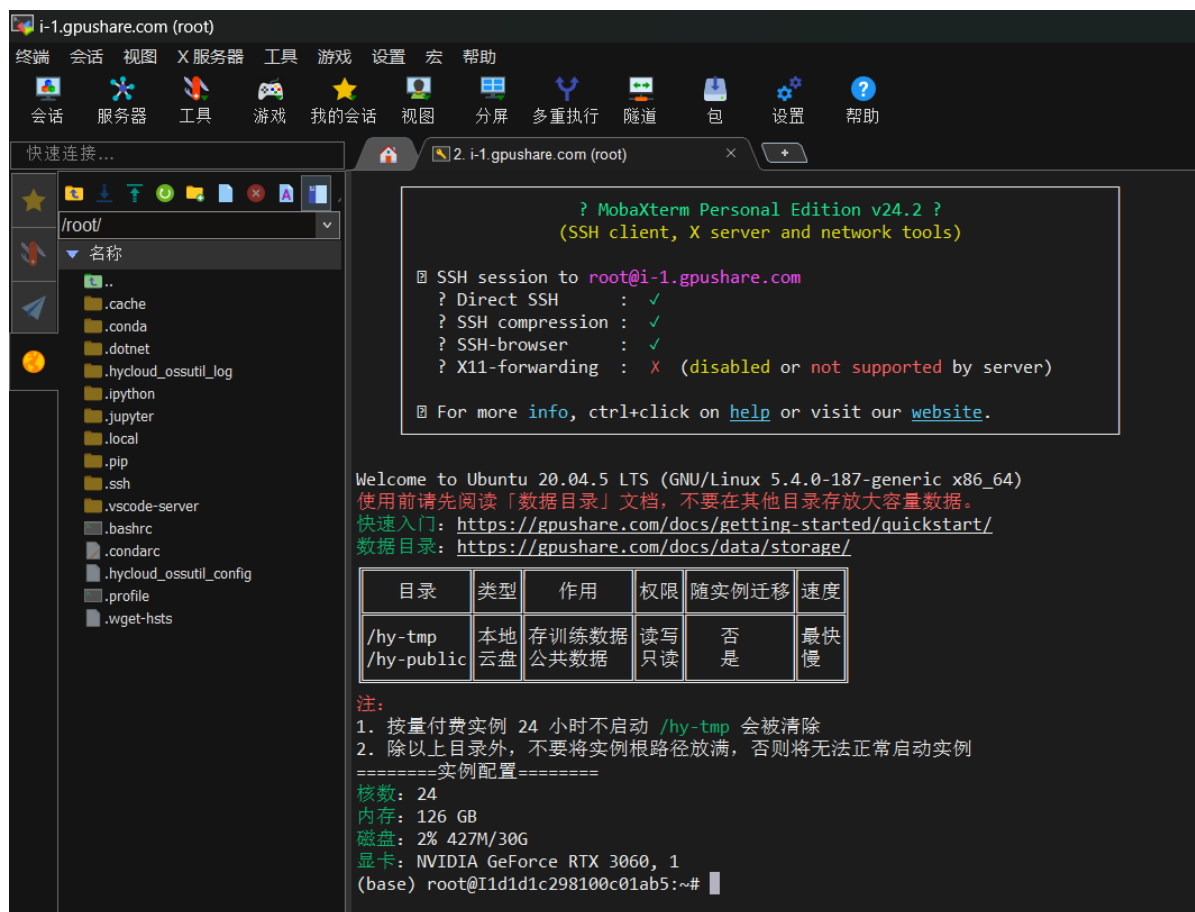


这样就是连接成功！

MobaXterm连接（自行下载）



点击 **会话** 的 **SSH**，将登录指令复制下来，然后分别输入 **远程主机**，**指定用户名** 和 **端口号**，例如我的命令是 `ssh -p 34552 root@i-1.gpushare.com`，那我就应该分别填 **i-1.gpushare.com**，**root**，**34552**，然后点击 **OK**，然后按照提示输入密码。注意，MobaXterm的默认粘贴命令是 **Shift+Insert** !!! (可改)



这样就是连接成功了！

PS: VSCode那么好用，为什么要MobaXterm?

因为远程服务器不带显示器，VSCode默认没有X-sever (一种远程显示技术)，但是MobaXterm自带，在一些需要显示图片或者其他任务时，VSCode只能保存下来看，但是MobaXterm可以实时观看。

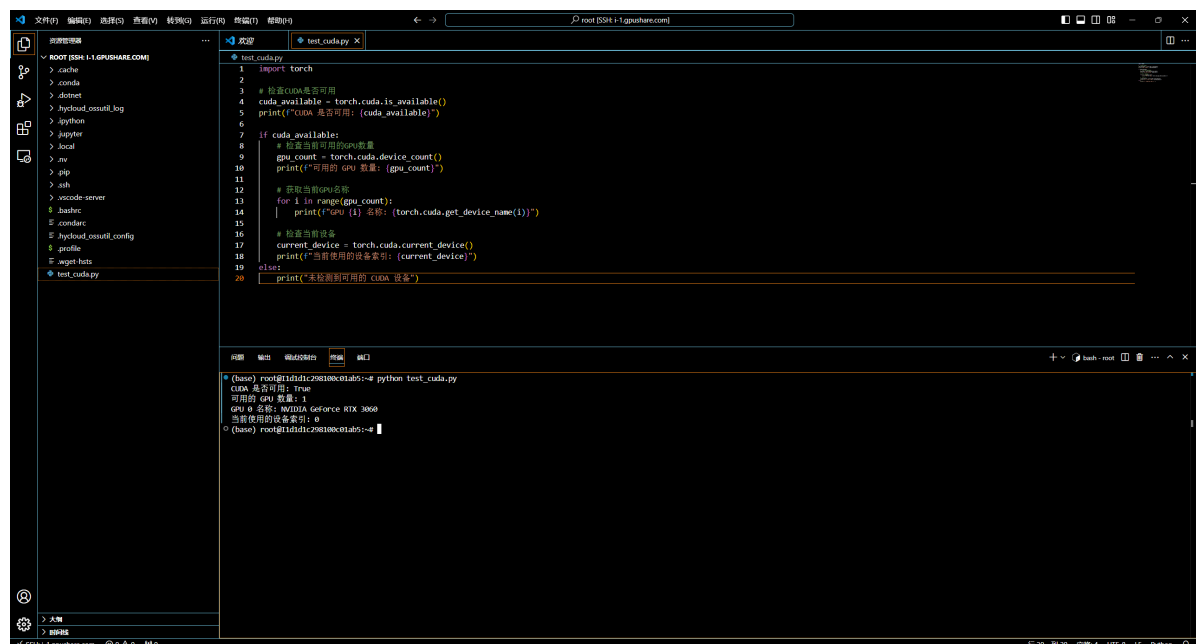
二、BERT环境配置

因为在创建服务器时就已经装好了 `PyTorch`，那么我们配环境中最困难的一步已经让别人帮我们做好了，可以运行以下测试代码：`test_cuda.py`

```
import torch

# 检查CUDA是否可用
cuda_available = torch.cuda.is_available()
print(f"CUDA 是否可用: {cuda_available}")

if cuda_available:
    # 检查当前可用的GPU数量
    gpu_count = torch.cuda.device_count()
    print(f"可用的 GPU 数量: {gpu_count}")
    # 获取当前GPU名称
    for i in range(gpu_count):
        print(f"GPU {i} 名称: {torch.cuda.get_device_name(i)}")
    # 检查当前设备
    current_device = torch.cuda.current_device()
    print(f"当前使用的设备索引: {current_device}")
else:
    print("未检测到可用的 CUDA 设备")
```



```
root@101d1c29818ec1ab5:~# python test_cuda.py
CUDA 是否可用: True
可用的 GPU 数量: 1
GPU 0 名称: NVIDIA GeForce RTX 3090
当前使用的设备索引: 0
root@101d1c29818ec1ab5:~#
```

得到如图输出，说明 `torch`包的 `CUDA` 和 `GPU` 都是可用的

但是想要运行 `BERT`，还要 `pip install transformers`，因为他是基于 `Transformer` 架构的
安装好后，可以运行以下测试代码：`test_bert.py`

```
from transformers import BertTokenizer, BertForSequenceClassification
import torch

model_name = "roberta-base-finetuned-jd-binary-chinese" # 加载预训练的 tokenizer
和 微调好的情感分类模型(bert-base-chinese)
tokenizer = BertTokenizer.from_pretrained(model_name)
model = BertForSequenceClassification.from_pretrained(model_name)
```

```

# 移动模型到 GPU
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model.to(device)

# 准备输入的文本（中文情感分析）
def predict_sentiment(text):
    # 编码输入
    inputs = tokenizer(text, return_tensors='pt', padding=True, truncation=True,
max_length=512)
    # 将输入移动到 GPU
    inputs = {key: value.to(device) for key, value in inputs.items()}
    # 模型推理（不计算梯度）
    with torch.no_grad():
        outputs = model(**inputs)
    # 获取模型的输出 logits 并将其转换为概率
    logits = outputs.logits
    probs = torch.softmax(logits, dim=-1)
    # 获取最大概率的类别（0：消极，1：积极）
    sentiment = torch.argmax(probs, dim=-1).item()
    # 返回情感类别和相应的概率
    return "积极" if sentiment == 1 else "消极", probs

# 测试文本
text = "今天的天气真好，我感到非常开心！"
# text = input("请输入测试文本：")

# 预测情感
sentiment, probability = predict_sentiment(text)
print(f"文本：{text}")
print(f"预测的情感：{sentiment}")
print(f"概率：{probability}")

```

```

(base) root@ididic2888x0ab5:~# python test_bert.py
/usr/local/miniconda3/lib/python3.8/site-packages/torch/_utils.py:726: UserWarning: TypedStorage is deprecated. It will be removed in the future and TypedStorage will be the only storage class. This should only matter to you if you are using storages directly. To access UntypedStorage directly, use tensor.untyped_storage() instead of tensor.storage()
  return self._get_(instance, owner)()
文本: 今天的天气真好，我感到非常开心！
预测的情感: 积极
概率: tensor([0.0110, 0.9889]), device='cuda:8'
(base) root@ididic2888x0ab5:~#

```

得到如图输出，就说明 BERT 的基础环境也已经配好了！

着手开始构建属于自己的模型吧！

附录

1. 如果服务器连接不上，请核对要填的各个信息是否正确，也可以在 `Windows Terminal` 中直接输入连接命令以及密码，测试命令以及密码是否正确，若否，请重启服务器。
2. 运行 `test_bert.py` 时若报错显示没有或者无法下载 `tokenizer` 以及模型，可以前往 [uer/roberta-base-finetuned-jd-binary-chinese-Hugging_Face](#) 手动下载，其他模型 (大量) 也可在 [Hugging_Face](#) 找到并手动下载。

Mail : `jiuri.z.0796@gmail.com`

Date : `2024/10/18`