

# 部署Perplexity r1-1776模型指南

## 前置条件

- 为了部署相关模型，您需要开通相关的ECS机型购买选项，并申请到相关的资源quota配置
- 完成VPC，EIP等服务开通和配置

## 部署步骤

### 1. 创建GPU ECS实例

- 选择合适的region/az，以及相应的机型，配置合适大小的云盘空间

模型名称	参数量	云盘大小推荐
r1-1776	671B	建议FlexPL,2TB

- 选择ubuntu 22.04 配置gpu driver 535版本镜像：



- 为GPU ECS配置公网访问方式，选择绑定公网IP或者使用NAT网关均可。后续自动拉取镜像和权重需要公网访问。

### 2. 部署容器环境

登录2台H20 ECS 分别安装容器环境，具体安装方法参考如下：

方法一：安装nvidia-docker2

```
1 curl -s https://mirrors.ivolces.com/nvidia_all/ubuntu2204/x86_64/3bf863cc.pub | sudo apt-key add -
2 cat <<EOF >/etc/apt/sources.list.d/nvidia.list
3 deb http://mirrors.ivolces.com/nvidia_all/ubuntu2204/x86_64/ /
4 EOF
5 apt update
```

6 apt install nvidia-docker2

方法二：或者使用docker + nvidia container toolkit，具体如下：

- 安装docker:

```
1 # Update the apt package index and install packages to allow apt to use a
  repository over HTTPS
2 sudo apt update
3 sudo apt install ca-certificates curl gnupg lsb-release
4 # Add Docker's official GPG key
5 sudo mkdir -p /etc/apt/keyrings
6 curl -fsSL https://mirrors.ivoles.com/docker/linux/ubuntu/gpg | sudo gpg --
  dearmor -o /etc/apt/keyrings/docker.gpg
7 # Use the following command to set up the repository
8 echo "deb [arch=$(dpkg --print-architecture) signed-
  by=/etc/apt/keyrings/docker.gpg]
  https://mirrors.ivoles.com/docker/linux/ubuntu $(lsb_release -cs) stable" |
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
9 # update package index
10 sudo apt update
11 # Install docker-ce
12 sudo apt install docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

- 安装nvidia-container-toolkit:

```
1 curl -s https://mirrors.ivoles.com/nvidia_all/ubuntu2204/x86_64/3bf863cc.pub
  | sudo apt-key add -
2 cat <<EOF >/etc/apt/sources.list.d/nvidia.list
3 deb http://mirrors.ivoles.com/nvidia_all/ubuntu2204/x86_64/ /
4 EOF
5 apt update
6 apt install nvidia-container-toolkit
7 sudo nvidia-ctl runtime configure --runtime=docker
8 sudo systemctl restart docker
```

### 3. 准备模型文件：

建立路径

```
1 # 模型权重存放路径
2 mkdir -p /data01/models
```

```
3 cd /data01/models
```

安装huggingface cli并拉取模型文件

```
1 # 安装HF CLI
2 pip install huggingface_hub[cli]
3 # 拉取模型文件, 支持续传
4 huggingface-cli download perplexity-ai/r1-1776 --local-dir r1-1776
```

## 4. 启动模型

1. 请采用下面命令完成相关配置, 启动docker和模型服务

a. 登录节点0, 执行如下docker run命令

```
1 ##total ranks = 2
2 # rank 0
3 docker run -d --network host --privileged --gpus=all --ipc=host -v
  /data01:/data -v /var/run/nvidia-topologyd/:/var/run/nvidia-topologyd/ -e
  MODEL_NAME=r1-1776 -e MODEL_LENGTH=8192 -e TP=16 -e TOTAL_RANKS=2 -e
  RANKS=0 -e RANK0_ADDR=192.168.0.2:10240 -e PORT=8080 -e CMD_ARGS="--mem-
  fraction-static 0.95 --disable-cuda-graph" ai-containers-cn-
  beijing.cr.volces.com/deeplearning/sglang:0.4.2.iaas
4
```

b. 登录节点1, 执行如下docker run命令

```
1 ## total ranks = 2
2
3 # rank 1cd
4 docker run -d --network host --privileged --gpus=all --ipc=host -v
  /data01:/data -v /var/run/nvidia-topologyd/:/var/run/nvidia-topologyd/ -e
  MODEL_NAME=r1-1776 -e MODEL_LENGTH=8192 -e TP=16 -e TOTAL_RANKS=2 -e
  RANKS=1 -e RANK0_ADDR=192.168.0.2:10240 -e PORT=8080 -e CMD_ARGS="--mem-
  fraction-static 0.95 --disable-cuda-graph" ai-containers-cn-
  beijing.cr.volces.com/deeplearning/sglang:0.4.2.iaas
```

上述命令中 标黄字段的相关环境变量说明如下, 需要按照实际情况修改:

环境变量	默认值	描述
------	-----	----

MODEL_PATH	/data/models	容器内模型存储路径
MODEL_NAME	DeepSeek-R1	模型名称
MODEL_LENGTH	131072	模型的最大长度（token 数）
TP	16	Tensor Parallelism 并行度
TOTAL_RANKS	1	总节点数
RANKS	0	当前节点的rank号
RANK0_ADDR	需要指定	rank0节点的IP和端口，一般为node0 所在的内网ip
PORT	8080	服务监听的端口号

高级配置：如果要修改模型其他启动参数（如GPU\_MEM\_UTIL，PREFIX\_CACHE等）的话，请在容器中修改模型启动脚本 “/entrypoint.sh” 的参数

两个节点docker分别启动以后，会自动拉取镜像和对应的权重文件，权重存放在/data/models目录下。两个节点的运行状态可以通过 docker logs查看，当节点0的docker logs显示如下时，代表模型服务已经启动成功，可以进行下一步测试操作。

```
[2025-02-06 09:01:08 TP5] max_total_num_tokens=506285, chunked_prefill_size=8192, max_prefill_tokens=16384, max_running_requests=2049, context_len=163840
[2025-02-06 09:01:08 TP7] max_total_num_tokens=506285, chunked_prefill_size=8192, max_prefill_tokens=16384, max_running_requests=2049, context_len=163840
[2025-02-06 09:01:08 TP6] max_total_num_tokens=506285, chunked_prefill_size=8192, max_prefill_tokens=16384, max_running_requests=2049, context_len=163840
[2025-02-06 09:01:08 TP0] max_total_num_tokens=506285, chunked_prefill_size=8192, max_prefill_tokens=16384, max_running_requests=2049, context_len=163840
[2025-02-06 09:01:08 TP4] max_total_num_tokens=506285, chunked_prefill_size=8192, max_prefill_tokens=16384, max_running_requests=2049, context_len=163840
[2025-02-06 09:01:08 TP2] max_total_num_tokens=506285, chunked_prefill_size=8192, max_prefill_tokens=16384, max_running_requests=2049, context_len=163840
[2025-02-06 09:01:08 TP3] max_total_num_tokens=506285, chunked_prefill_size=8192, max_prefill_tokens=16384, max_running_requests=2049, context_len=163840
[2025-02-06 09:01:08 TP1] max_total_num_tokens=506285, chunked_prefill_size=8192, max_prefill_tokens=16384, max_running_requests=2049, context_len=163840
[2025-02-06 09:01:08] INFO: Started server process [30]
[2025-02-06 09:01:08] INFO: Waiting for application startup.
[2025-02-06 09:01:08] INFO: Application startup complete.
[2025-02-06 09:01:08] INFO: Uvicorn running on http://0.0.0.0:8080 (Press CTRL+C to quit)
[2025-02-06 09:01:09] INFO: 127.0.0.1:34254 - "GET /get_model_info HTTP/1.1" 200 OK
[2025-02-06 09:01:09 TP0] Prefill batch. #new-seq: 1, #new-token: 7, #cached-token: 0, cache hit rate: 0.00%, token usage: 0.00, #running-req: 0, #queue-req: 0
[2025-02-06 09:01:11 TP4] Using default W8A8 Block FP8 kernel config. Performance might be sub-optimal! Config file not found at /usr/local/lib/python3.10/dist-packages/sglang/srt/layers/g
ization/configs/N=2048,K=512,device_name=NVIDIA H20,dtype=fp8 w8a8,block_shape=[128, 128].json
[2025-02-06 09:01:11 TP7] Using default W8A8 Block FP8 kernel config. Performance might be sub-optimal! Config file not found at /usr/local/lib/python3.10/dist-packages/sglang/srt/layers/g
ization/configs/N=2048,K=512,device_name=NVIDIA H20,dtype=fp8 w8a8,block_shape=[128, 128].json
[2025-02-06 09:01:11 TP6] Using default W8A8 Block FP8 kernel config. Performance might be sub-optimal! Config file not found at /usr/local/lib/python3.10/dist-packages/sglang/srt/layers/g
ization/configs/N=2048,K=512,device_name=NVIDIA H20,dtype=fp8 w8a8,block_shape=[128, 128].json
[2025-02-06 09:01:11 TP0] Using default W8A8 Block FP8 kernel config. Performance might be sub-optimal! Config file not found at /usr/local/lib/python3.10/dist-packages/sglang/srt/layers/g
ization/configs/N=2048,K=512,device_name=NVIDIA H20,dtype=fp8 w8a8,block_shape=[128, 128].json
[2025-02-06 09:01:11 TP2] Using default W8A8 Block FP8 kernel config. Performance might be sub-optimal! Config file not found at /usr/local/lib/python3.10/dist-packages/sglang/srt/layers/g
ization/configs/N=2048,K=512,device_name=NVIDIA H20,dtype=fp8 w8a8,block_shape=[128, 128].json
[2025-02-06 09:01:11 TP3] Using default W8A8 Block FP8 kernel config. Performance might be sub-optimal! Config file not found at /usr/local/lib/python3.10/dist-packages/sglang/srt/layers/g
ization/configs/N=2048,K=512,device_name=NVIDIA H20,dtype=fp8 w8a8,block_shape=[128, 128].json
[2025-02-06 09:01:11 TP1] Using default W8A8 Block FP8 kernel config. Performance might be sub-optimal! Config file not found at /usr/local/lib/python3.10/dist-packages/sglang/srt/layers/g
ization/configs/N=2048,K=512,device_name=NVIDIA H20,dtype=fp8 w8a8,block_shape=[128, 128].json
[2025-02-06 09:01:11 TP5] Using default W8A8 Block FP8 kernel config. Performance might be sub-optimal! Config file not found at /usr/local/lib/python3.10/dist-packages/sglang/srt/layers/g
ization/configs/N=2048,K=512,device_name=NVIDIA H20,dtype=fp8 w8a8,block_shape=[128, 128].json
[2025-02-06 09:01:15] INFO: 127.0.0.1:34258 - "POST /generate HTTP/1.1" 200 OK
[2025-02-06 09:01:15] INFO: The server is fired up and ready to roll!
```

2. 登录节点0，执行以下curl prompt，观察到流式生成为模型正常运行，可以进行下一步的模型调用。执行docker logs 看到模型端口是否成功启动日志。

```
1 curl -X POST http://0.0.0.0:8080/v1/chat/completions -H "Content-Type: application/json" -d '{
2   "model": "/data/models/DeepSeek-R1",
3   "messages": [
```

```
4      {
5          "role": "user",
6          "content": "请证明一下黎曼猜想"
7      }
8  ],
9  "stream": true,
10 "max_tokens": 100,
11 "temperature": 0.7
12 }'
```

## 5. 【可选】部署NGINX

生成API Key (using python3):

```
1 import secrets
2 api_key = secrets.token_hex(16)
3 print(api_key)
```

安装:

```
1 sudo apt update
2 sudo apt install nginx -y
3
4 # 创建配置文件
5 sudo nano /etc/nginx/sites-available/llm-proxy
6
```

配置文件:

```
1 server {
2     listen 80;
3     server_name [公网IP];
4
5     set $valid_api_key 0;
6     if ($http_x_api_key = "YOUR_API_KEY_1") {
7         set $valid_api_key 1;
8     }
9     if ($http_x_api_key = "YOUR_API_KEY_2") {
10         set $valid_api_key 1;
11     }
12 }
```

```

13     location / {
14         if ($valid_api_key = 0) {
15             return 403 "Forbidden: Missing or invalid API key";
16         }
17
18         proxy_pass http://[服务ip]:8080;
19         proxy_set_header Host $host;
20         proxy_set_header X-Real-IP $remote_addr;
21         proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
22         proxy_set_header X-Forwarded-Proto $scheme;
23     }
24 }
25

```

## 开启服务:

```

1  sudo ln -s /etc/nginx/sites-available/llm-proxy /etc/nginx/sites-enabled/
2  sudo nginx -t
3  sudo systemctl restart nginx

```

## 排查日志:

```

1  # 查看日志
2  tail /var/log/nginx/error.log

```

## 测试:

```

1  curl -X POST http://45.78.210.224/v1/chat/completions -H "Content-Type:
2  application/json" -H "X-API-Key: [您的API key]" -d '{
3      "model": "/data/models/DeepSeek-R1",
4      "messages": [
5          {
6              "role": "user",
7              "content": "hello who are you?"
8          }
9      ],
10     "stream": true,
11     "max_tokens": 100,
12     "temperature": 0.7
13 }'

```

## 6. 【可选】部署Streamlit简单UI

安装依赖:

```
1 apt install python3-venv
2
3 python3 -m venv chatbot-venv
4 source chatbot-venv/bin/activate
5
6 pip install streamlit requests
```

界面主文件:

```
1 import streamlit as st
2 import requests
3 import json
4
5 st.title("DeepSeek-R1 Chatbot")
6
7 # Initialize chat history
8 if "messages" not in st.session_state:
9     st.session_state.messages = []
10
11 # Display chat messages from history on app rerun
12 for message in st.session_state.messages:
13     with st.chat_message(message["role"]):
14         st.markdown(message["content"])
15
16 # React to user input
17 if prompt := st.chat_input("What is your question?"):
18     # Display user message in chat message container
19     st.chat_message("user").markdown(prompt)
20     # Add user message to chat history
21     st.session_state.messages.append({"role": "user", "content": prompt})
22
23 # Call the API
24 response = requests.post(
25     "http://45.78.228.109/v1/chat/completions",
26     headers={"Content-Type": "application/json"},
27     data=json.dumps({
28         "model": "/data/models/DeepSeek-R1",
```

```

29         "messages": st.session_state.messages
30     })
31 )
32
33 if response.status_code == 200:
34     assistant_response = response.json()["choices"][0]["message"]
35     ["content"]
36     # Display assistant response in chat message container
37     with st.chat_message("assistant"):
38         st.markdown(assistant_response)
39         # Add assistant response to chat history
40         st.session_state.messages.append({"role": "assistant", "content":
41         assistant_response})
42 else:
43     st.error(f"Error: {response.status_code} - {response.text}")

```

## 附录

- Dockerfile中的引擎启动脚本 entrypoint.sh

```

1  #!/bin/bash
2
3  MODEL_PATH=${MODEL_PATH:-"/data/models"}
4  MODEL_NAME=${MODEL_NAME:-"DeepSeek-R1"}
5  MODEL_LENGTH=${MODEL_LENGTH:-131072}
6  TP=${TP:-8}
7  RANK0_ADDR=${RANK0_ADDR:-""}
8  RANKS=${RANKS:-0}
9  TOTAL_RANKS=${TOTAL_RANKS:-1}
10 PORT=${PORT:-8080}
11
12 # check if MODEL_PATH and MODEL_NAME are set
13 if [ -z "$MODEL_PATH" ] || [ -z "$MODEL_NAME" ]; then
14     echo "MODEL_PATH and MODEL_NAME must be set"
15     exit 1
16 fi
17
18 # check if MODEL_PATH not exists, create it
19 if [ ! -d "$MODEL_PATH" ]; then
20     mkdir -p $MODEL_PATH
21 fi
22

```



```
23 # check if MODE_PATH/MODEL_NAME not exists, download it
24 if [ ! -d "$MODEL_PATH/$MODEL_NAME" ]; then
25     cd $MODEL_PATH
26     oniond download model $MODEL_NAME
27     if [ $? -ne 0 ]; then
28         echo "Failed to download model $MODEL_NAME"
29         exit 1
30     fi
31     cd -
32 fi
33
34 # check if it is multiple ranks
35 if [ $TOTAL_RANKS -gt 1 ]; then
36     # check if RANK0_ADDR is set
37     if [ -z "$RANK0_ADDR" ]; then
38         echo "RANK0_ADDR must be set"
39         exit 1
40     fi
41
42     GLOO_SOCKET_IFNAME=eth0 NCCL_IB_HCA=mlx5 NCCL_IB_DISABLE=0
43     NCCL_SOCKET_IFNAME=eth0 NCCL_IB_GID_INDEX=3 python3 -m
44     sglang.launch_server --model-path $MODEL_PATH/$MODEL_NAME --tp $TP --dist-
45     init-addr $RANK0_ADDR --nnodes $TOTAL_RANKS --node-rank $RANKS --trust-
46     remote-code --host 0.0.0.0 --port $PORT
47 else
48
49     if [ $TP -gt 8 ]; then
50         TP=8
51     fi
52
53     python3 -m sglang.launch_server --model-path $MODEL_PATH/$MODEL_NAME --
54     context-length $MODEL_LENGTH --tp $TP --trust-remote-code --host 0.0.0.0 --
55     port $PORT --mem-fraction-static 0.95
56 fi
```