本地模型部署_交接文档

一、交接模型总览

模型类型	模型路径	服务地址（内网）
通用模型	`/Qwen/Qwen3-14B`	`http://172.32.1.163:8000/v1` `http://gpu3.xx`
涉台模型	`/model/model_taiwan_14B`	`http://172.32.1.162:8000/v1` `http://gpu2.xx`
涉藏模型	`/model/model_xizang_14B`	`http://172.32.1.161:8000/v1` `http://gpu1.xx`
翻译模型	`/Qwen/Qwen3-8B`（用于团源数据翻译任务）	`http://192.168.48.25:8000/v1`

二、容器部署与运行细节

✅ 通用模型（Qwen3-14B）部署命令示例：

```shell
sudo docker run -d --runtime nvidia --gpus all \
-v /home/ubuntu/.cache/modelscope/hub/models/Qwen/Qwen3-14B:/Qwen/Qwen3-14B \
-v /home/ubuntu/code/vllm/chat_template:/chat_template \
-p 8000:8000 \
--ipc=host \
--log-driver json-file \
--log-opt max-size-10m \
--log-opt max-file=3 \
vllm/vllm-openai:latest \
--model /Qwen/Qwen3-14B \
--tensor-parallel-size 4 \
--api-key token-abc123 \
--dtype 'float16' \
--port 8000 \
--chat-template /chat_template/template_custom.jinja \
--no-enable-chunked-prefill \
--rope-scaling '{"rope_type":"yarn","factor":4.0,"original_max_position_embeddings":32768}' \
--max-model-len 131072
```

> 说明：其余模型部署命令与之类似，仅需更换 `-v 模型挂载路径` 与 `--model` 指定的模型路径。

> 🔗 更多参数说明与定制部署参考：

- vLLM 推理引擎完整参数列表文档：https://vllm.hyper.ai/docs/inference-and-serving/engine_args
- OpenAI 兼容接口说明文档：https://vllm.hyper.ai/docs/inference-and-serving/openai_compatible_server

---

三、认证方式与调用说明

- 模型统一采用 OpenAI 接口标准（`/v1/completions`）
- 使用 API-Key（如：`token-abc123`）进行简单认证，建议交接后进行轮换
- 接口部署端口均为 `8000`
- 支持内网调用（如 curl 或本地服务）

四、注意事项与日志处理

- 所有模型容器均通过 `Docker` 启动，日志配置为 `json-file` 驱动，单文件最大 10MB，最多 3 个轮换。
- 日志查看示例命令：

```shell
docker logs -f 容器名称     # 通用模型
docker logs -f 容器名称     # 涉台模型
docker logs -f 容器名称     # 涉藏模型
docker logs -f 容器名称     # 翻译模型
```

- 如需更新容器配置，请更新脚本并执行 `docker restart [容器名]`。

五、模型挂载路径一览

模型类型	实际挂载路径（宿主机）	容器内路径
通用模型	`/home/ubuntu/.cache/modelscope/hub/models/Qwen/Qwen3-14B`	`/Qwen/Qwen3-`
涉台模型	`/home/ubuntu/code/vllm/model/model_taiwan_14B`	`/model/model`
涉藏模型	`/home/ubuntu/code/vllm/model/model_xizang_14B`	`/model/model`
翻译模型	`/home/ubuntu/.cache/modelscope/hub/models/Qwen/Qwen3-8B`	`/Qwen/Qwen3-`

六. vLLM 定制化开发路径说明

如需定制模型推理行为（如动态路由、多任务指令解析、Token 重写、预处理优化等），请修改以下路径中的源码文件：

```python
源码路径：/home/ubuntu/code/vllm/vllm/engine/
```

建议优先修改的关键文件模块说明如下：

文件/模块名称	功能说明
`vllm/engine/llm_engine.py`	核心推理引擎类，模型加载、请求分发入口
`vllm/engine/async_llm_engine.py`	异步推理核心，支持多请求异步调度
`vllm/engine/arg_utils.py`	CLI 参数解析模块，可添加自定义启动参数
`vllm/engine/openai_server.py`	OpenAI 接口兼容层，修改接口逻辑或增加日志审计点
`vllm/engine/routing/`	各类请求的分发、调度逻辑模块

开发建议：

- 修改后使用 `docker build` 自行构建定制镜像；
- 如仅调试可挂载本地代码目录进行热更新；
- 推荐建立 Git 分支记录开发变更，避免污染官方分支。

八. 配套服务 `new-api` 部署信息

该服务作为模型 API 的配套调用服务，负责数据库连接、接口转发、日志写入等功能，已通过 Docker 容器方式长期运行。

📌 基本信息

- 服务名称：new-api（模型统一接口服务）
- 部署机器：172.32.1.163
- 部署路径（宿主机）：`/home/ubuntu/code/newapi`
- 容器镜像：`calciumion/new-api:latest`
- 挂载目录：`/home/ubuntu/code/newapi/data` → `/data`
- 监听端口：3000

✅ 部署命令如下：

```shell
sudo docker run --name new-api -d --restart always \
  -p 3000:3000 \
  -e SQL_DSN="admin:admin@tcp(172.32.1.163:3306)/model" \
  -e TZ=Asia/Shanghai \
  -v /home/ubuntu/code/newapi/data:/data \
  calciumion/new-api:latest
```

✅ 说明要点：

项目	内容
容器名称	`new-api`
镜像名	`calciumion/new-api:latest`
服务端口	`3000`（HTTP服务）
数据挂载路径	`/home/ubuntu/code/newapi/data`（宿主机） → `/data`（容器内）
数据库配置	`172.32.1.163:3306`，数据库名：`model`，用户：admin 密码：admin（建议后续更换）
时区设置	`Asia/Shanghai`
启动策略	`--restart always`（系统重启后自动启动）

✅ 数据与代码位置：

- 启动脚本或部署配置已内嵌于 Docker 启动命令中
- 程序文件与配置位于：

```
/home/ubuntu/code/newapi/
├── data/                  # 数据存储目录（容器挂载）
├── logs/（如有）           # 建议配置挂载输出日志
```

✅ 建议事项：

- 若进行代码级定制或日志路径扩展，建议基于该路径下源码构建自定义镜像；
- SQL 连接信息目前为明文，建议迁移至 `.env` 文件或容器密钥管理策略；
- 日志建议统一挂载至 `/var/log/newapi/`，便于与模型服务日志统一管理。