Ollama的概念，安装，部署，推理

讲义地址

https://docs.qq.com/doc/p/8f80d461e5097f1a57e5d5fbe61b9174c1794972

概念

Ollama是一个开源的人工智能模型管理和推理工具项目。其简单、易用、多平台兼容是最主要的特点。它涵盖了大模型的下载，部署，推理，修改配置等操作，提供ui操作界面，简答好用，兼容win，mac，linux等多个平台。和vLLM相比，它使用门槛更低。

官网：https://ollama.com/

之前尝试过用mac直接安装ollama客户端并且下载部署大模型并进行推力测试，结果由于配置不足，机器很卡。

安装

在线安装

在官网可以找到linux环境下的安装方式。直接拷贝，在linux中运行。

```
root@a892b4dac720494f923c9986c3c71f2e-task0-0:/code# curl -fsSL https://ollama.com/install.sh | sh
>>> Installing ollama to /usr/local
```

有可能运行不了，是因为服务器无法访问国外网络。

离线安装

汇视威的机器无法完成试验，因为它无法访问外网。

离线安装就是先下载安装必须的文件，上传到 我们自己的linux服务器，然后执行安装脚本。

步骤如下： 1、 去 https://github.com/ollama/ollama/releases/tag/v0.4.2 找到安装包。

Contributors

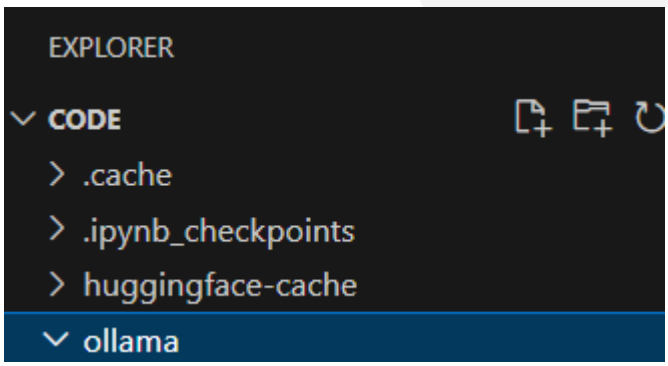
neomantra, ivostoykov, and 4 other contributors

▾ Assets 13

ollama-darwin	62.9 MB	2 days ago
Ollama-darwin.zip	185 MB	2 days ago
ollama-linux-amd64-rocm.tgz	1.13 GB	2 days ago
ollama-linux-amd64.tgz	1.71 GB	2 days ago
ollama-linux-arm64-jetpack5.tgz	452 MB	2 days ago
ollama-linux-arm64-jetpack6.tgz	415 MB	2 days ago
ollama-linux-arm64.tgz	1.44 GB	2 days ago
ollama-windows-amd64.zip	1.79 GB	2 days ago
ollama-windows-arm64.zip	22.5 MB	2 days ago
OllamaSetup.exe	747 MB	2 days ago
sha256sum.txt	916 Bytes	2 days ago
Source code (zip)		2 days ago
Source code (tar.gz)		2 days ago

- darwin对应MacOS
- linux对应linux系统，其中rocm表示AMD显卡，arm表示arm架构芯片
- widows对应windows系统

按照我机器的配置，我下载其中的 `ollama-linux-amd64.tgz`，将它上传到我自创的ollama目录中。



下一步，用浏览器获得 https://ollama.com/install.sh 这个sh文件的内容，并修改其中这一部分。

```
76      curl --fail --show-error --location --progress-bar \
77          "https://ollama.com/download/ollama-linux-${ARCH}.tgz${VER_PARAM}" | \
78          $SUDO tar -xzf - -C "$OLLAMA_INSTALL_DIR"
```

将76到78行注释，改成这一行：

```
$SUDO tar -xzf /code/ollama/ollama-linux-amd64.tgz - -C "$OLLAMA_INSTALL_DIR"
```

这里其实就是将 用解压的过程替代 下载再解压的过程，两者的结果是一样，都是得到一个ollama解压包的文件,上面的 `/code/ollama/ollama-linux-amd64.tgz` 是压缩包的绝对路径。

将改过之后的 install.sh 文件 保存到刚刚的ollama目录下。

并执行`sh install.sh`

测试是否安装成功

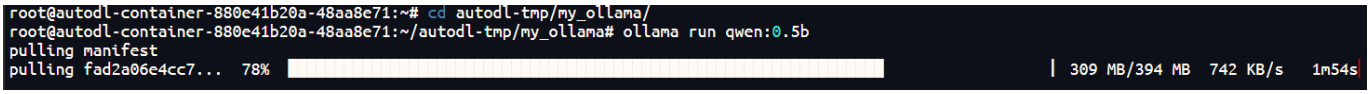
如果是刚刚安装成功，那必须先启动服务

```
ollama serve
```

启动之后，执行:

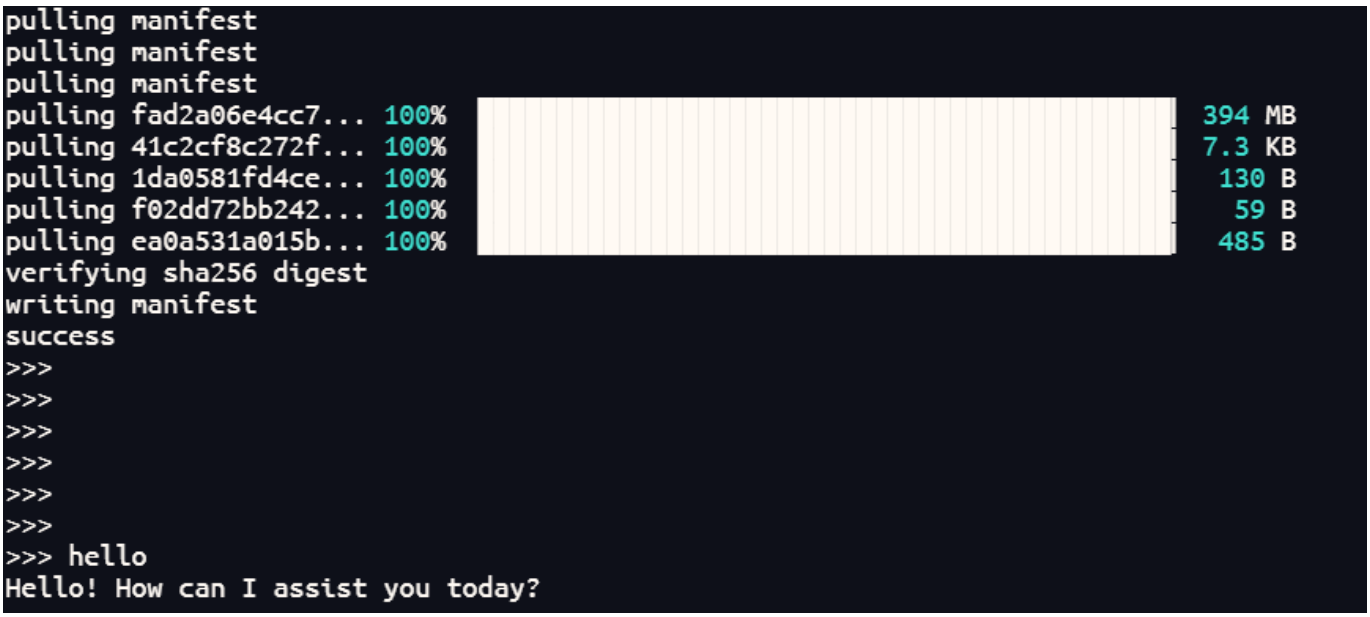
```
ollama run qwen:0.5b
```

用国产最小的一个模型来测试ollama是否正常。



看上去一切正常，等待下载完成。

启动之后可以直接对话：



测试成功。

至于docker方式的安装和启动，以及多卡推理，现在我用的autoDL租用的服务器没有发现支持多显卡的，并且aotoDL的机器本身就是用docker容器实现的，无法嵌套docker，所以安装部署的试验先做到这里。

接下来试试怎么用我本地的 FastGpt，oneAPI来接入这个。

接入oneapi和fastGpt

从外部接入ollama提供的http接口，需要设置 OLLAMA_HOST参数为 0.0.0.0，方法为：

先创建配置的目录

```
mkdir -p /etc/systemd/system/ollama.service.d
```

并且在其中新建一个 environment.conf文件，并保存如下内容：

```
[Service]
Environment="OLLAMA_HOST=0.0.0.0:11434"
Environment="CUDA_VISIBLE_DEVICES=0"
Environment="OLLAMA_SCHED_SPREAD=0"
```

由于修改了配置，所以必须重启服务。

```
systemctl daemon-reload  # 重新加载启动文件
systemctl restart ollama # 重启ollama serve
```

（尴尬的是，autoDL的机器是基于docker容器的，所以无法执行上面的命令，我只能手动重启机器）

在这台机器上安装 oneapi，并且将ollama发布的http服务添加到渠道中去。

安装 oneapi

官网 https://github.com/songquanpeng/one-api

同理，autoDL的机器是基于docker容器的，无法执行最简易的 docker-compose方式的oneapi方式的部署，但是可以尝试手动部署。

手动部署

1. 从 GitHub Releases 下载可执行文件或者从源码编译：

```
git clone https://github.com/songquanpeng/one-api.git

# 构建前端
cd one-api/web/default
npm install
npm run build

# 构建后端
cd ../..
go mod download
go build -ldflags "-s -w" -o one-api
```

2. 运行：

```
chmod u+x one-api
./one-api --port 3000 --log-dir ./logs
```

3. 访问 http://localhost:3000/ 并登录。初始账号用户名为 `root`，密码为 `123456`。

更加详细的部署教程参见此处。

并且，我发现，one-api似乎支持mac，linux，win三个平台。

在 https://github.com/songquanpeng/one-api/releases 中找到exe：

同时，又发现了先知老师的一篇文章：`https://www.cnblogs.com/Listener-wy/p/18422770`

似乎支持通过docker来启动one-api。

三步走， 第一：安装docker的win客户端。 第二：命令行拉取one-api镜像 `docker pull justsong/one-api` 第三：启动one-api: `run --name one-api -d --restart always -p 3002:3000 -e TZ=Asia/Shanghai -v C:/LLM/OneApi-V-Data:/data justsong/one-api`

接下来，浏览器打开 `http://localhost:3002/` 就能顺利进入到oneAPI的操作界面了。

今天的收获是：**能够在我本机安装one-api。但是如何去访问到我在AutuDL上部署的ollama服务，如何做?**

有大佬提示可以：`ssh -p 53407 root@connect.yza1.seetacloud.com -L 0.0.0.0:8080:0.0.0.0:11434 -N`

可以用这种ssh隧道的方式建立端口映射。

算了，我总结出一点吧。

老师用的机器，和我们autoDL租的机器，在试验流程上有很大区别，如果要做作业的话，最好选阿里云那种正常机器，比autuDL稍贵一些，但是实验流程可以保持一致。

本次实验我放弃了。没法玩。

`ssh -p 30524 root@connect.nma1.seetacloud.com -L 0.0.0.0:11434:0.0.0.0:11434 -N`