

实验报告

221900110 黄扬昊

实验步骤

任务一

安装vscode



```
nju@nju: ~/dl
获取并检查 snap "code" (184) 的ass
获取并检查 snap "code" (184)
获取并检查 snap "code"
获取并检查 snap
挂载 snap
挂载
snap "code" (184)
挂载 snap "code" (184)
挂载 snap "code" (184)
设置 snap "code" (184) 的安全配置
设置 snap "code" (184) 的安全配置
设置 snap "code" (184) 的安全配置
设置 snap "code" (184) 的安全配置
设置 snap "code" (184) 的安全配置
设置 snap "code" (184) 的安全配置
设置 snap "code" (184) 的安全配置
配置
的安全配置
(184) 的安全配置
使 snap "co
如果
存在，则运行 "code" snap 的配置钩子
code e54c774e from Visual Studio Code (vscode✓) installed
nju@nju:~/dl$
```

安装docker并测试docker成功安装

```
nju@nju: /etc/apt/sources.list.d
正在设置 libnvidia-container-tools (1.13.5-1) ...
正在设置 nvidia-container-toolkit (1.13.5-1) ...
正在处理用于 libc-bin (2.35-0ubuntu3.9) 的触发器 ...
{
  "registry-mirrors": [
    "https://docker.ins.run",
    "https://docker.xuanyuan.me",
    "https://docker.nju.edu.cn",
    "https://do.nark.eu.org",
    "https://dc.j8.work",
    "https://docker.m.daocloud.io",
    "https://dockerproxy.com",
    "https://docker.mirrors.ustc.edu.cn"
  ],
  "default-runtime": "nvidia",
  "runtimes": {
    "nvidia": {
      "path": "nvidia-container-runtime",
      "runtimeArgs": []
    }
  }
}
nju@nju: /etc/apt/sources.list.d$ docker run hello-world
docker: permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Head "http://%2Fvar%2Frun%2Fdocker.sock/_ping": dial unix /var/run/docker.sock: connect: permission denied

Run 'docker run --help' for more information
nju@nju: /etc/apt/sources.list.d$ newgrp docker
nju@nju: /etc/apt/sources.list.d$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
e6590344b1a5: Pull complete
Digest: sha256:e0b59a5163a5e6be84e210a2587e7d447e08f07a0e90798363fa4a046401e8
Status: Downloaded newer image for hello-world:latest

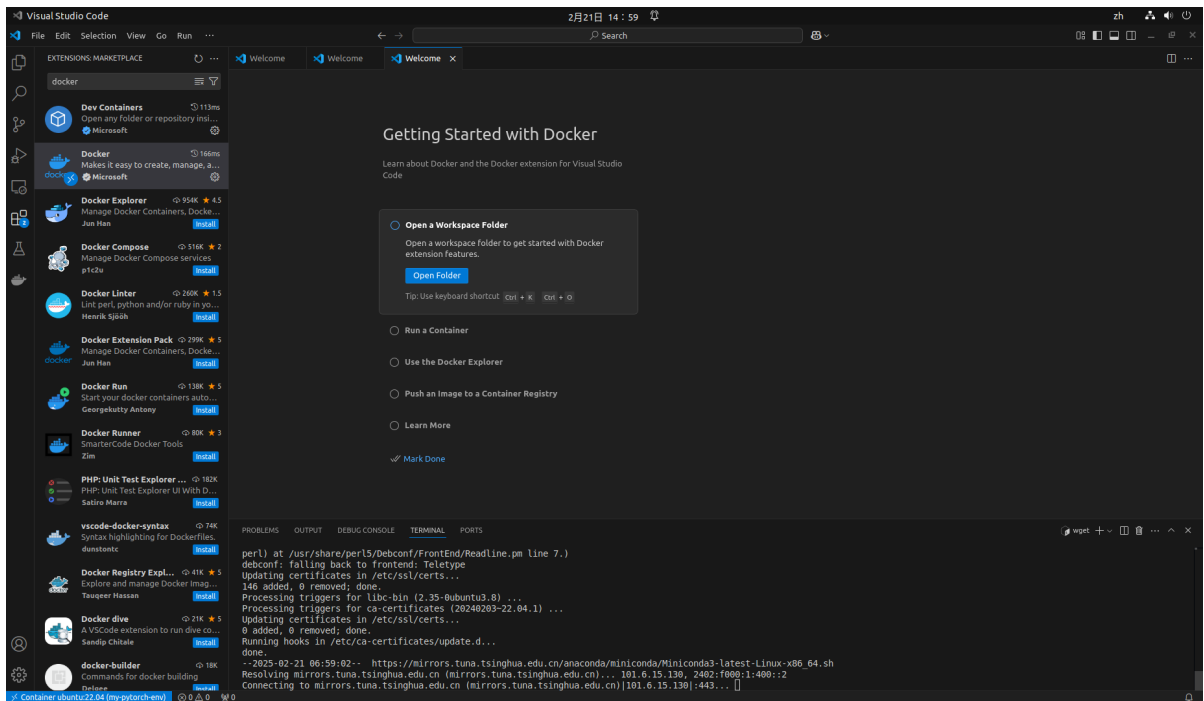
Hello from Docker!
This message shows that your installation appears to be working correctly.

To operate this program, Docker took the following steps:
```

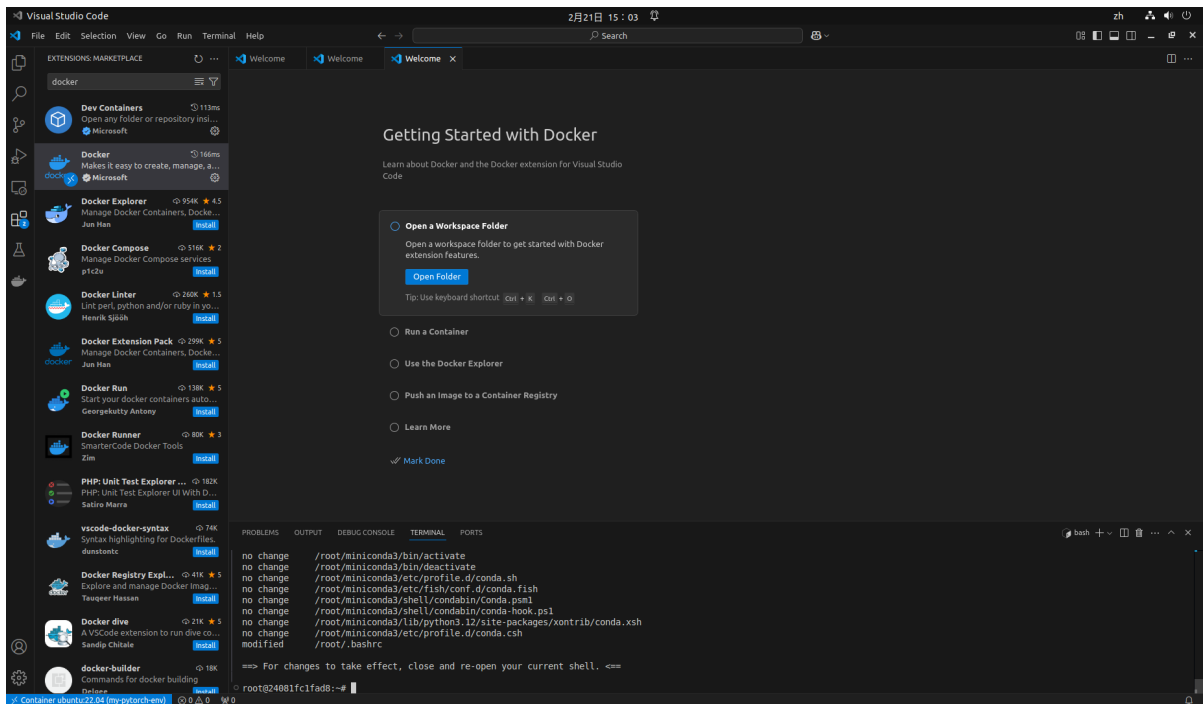
创建并运行基础容器

```
nju@nju: /etc/apt/sources.list.d$ docker run -it --gpus all --name my-pytorch-env -v ~/./.workspace ubuntu:22.04 /bin/bash
Unable to find image 'ubuntu:22.04' locally
22.04: Pulling from library/ubuntu
9cb31e2e37ea: Pull complete
Digest: sha256:ed154e454989878f5dec1bdfadb8c5cc9c48e0705d07b678ab6ae3fb61952d2
Status: Downloaded newer image for ubuntu:22.04
root@24081fc1fad8:/# 123456
bash: 123456: command not found
root@24081fc1fad8:/#
```

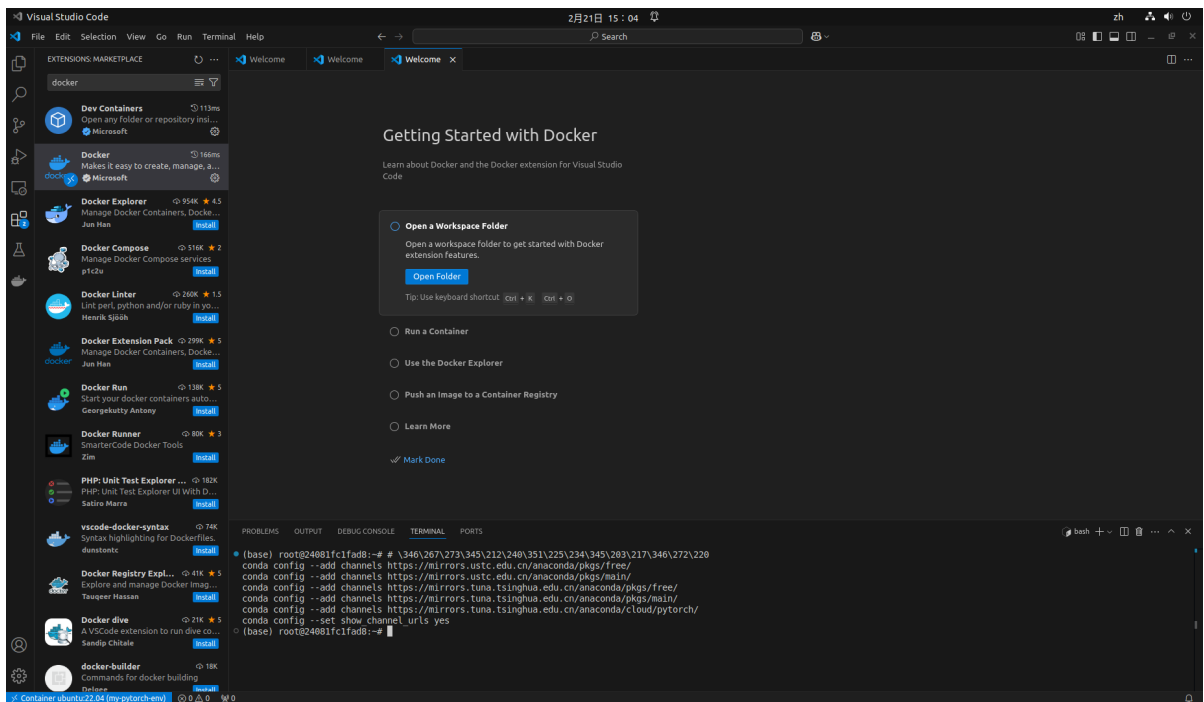
在vscode中配置插件并连接容器



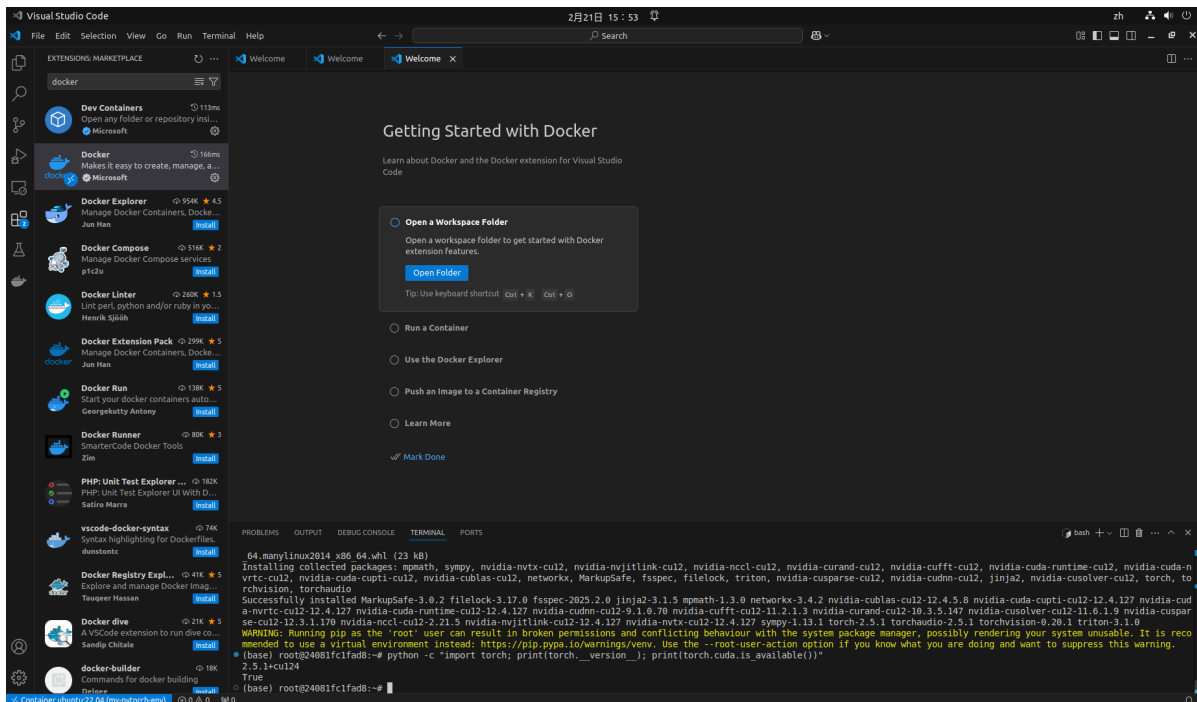
在vscode中加载docker环境



安装miniconda，添加镜像源



安装pytorch环境



任务二

完成第五节，输出如图

```
"""构造一个PyTorch数据迭代器。"""
dataset = data.TensorDataset(*data_arrays)
return data.DataLoader(dataset, batch_size, shuffle=is_train)

batch_size = 10
data_iter = load_array((features, labels), batch_size)

net = nn.Sequential(nn.Linear(2, 1)) #线性层
net[0].weight.data.normal_(0, 0.01)
net[0].bias.data.fill_(0)
loss = nn.MSELoss()
trainer = torch.optim.SGD(net.parameters(), lr=0.03)

num_epochs = 3
for epoch in range(num_epochs):
    for X, y in data_iter:
        #TODO:计算损失
        l = loss(net(X), y)
        #TODO:清空梯度
        trainer.zero_grad()
        #TODO:反向传播
        l.backward()
        #TODO:优化器更新参数
        trainer.step()
    l = loss(net(features), labels)
    print(f'epoch {epoch + 1}, loss {l:f}')

epoch 1, loss 0.000166
epoch 2, loss 0.000098
epoch 3, loss 0.000098
```