

基于头歌平台的 kairosh 实验使用引导

基于 C 语言的 OS 实验

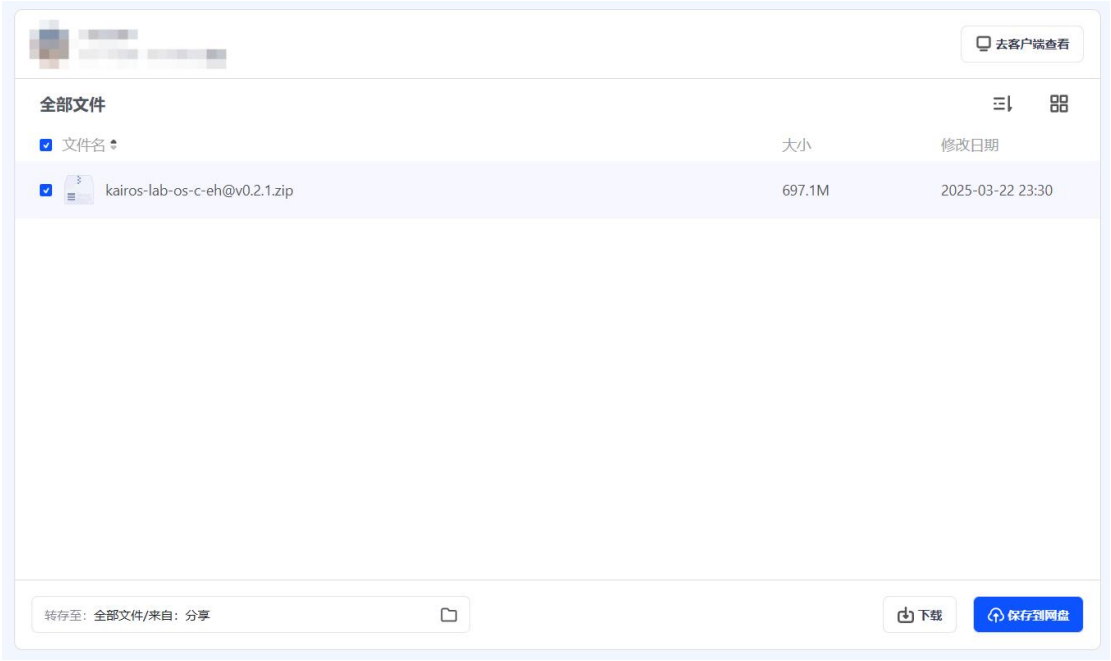
- 基于头歌平台的 kairosh 实验使用引导1
 - 资源下载 1
 - 本地搭建实验环境 3
 - 头歌平台使用 9
 - 完整实验过程 14

资源下载

实验容器镜像

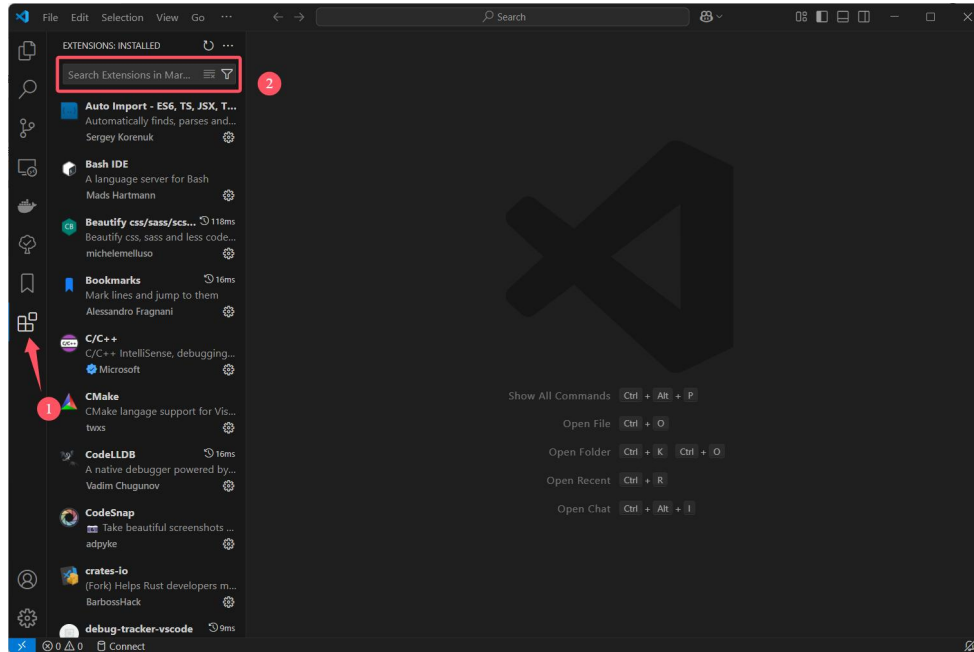
夸克网盘「kairosh-lab-os-c-eh@v0.2.1.zip」

<https://pan.quark.cn/s/fc25ed1da7b7>

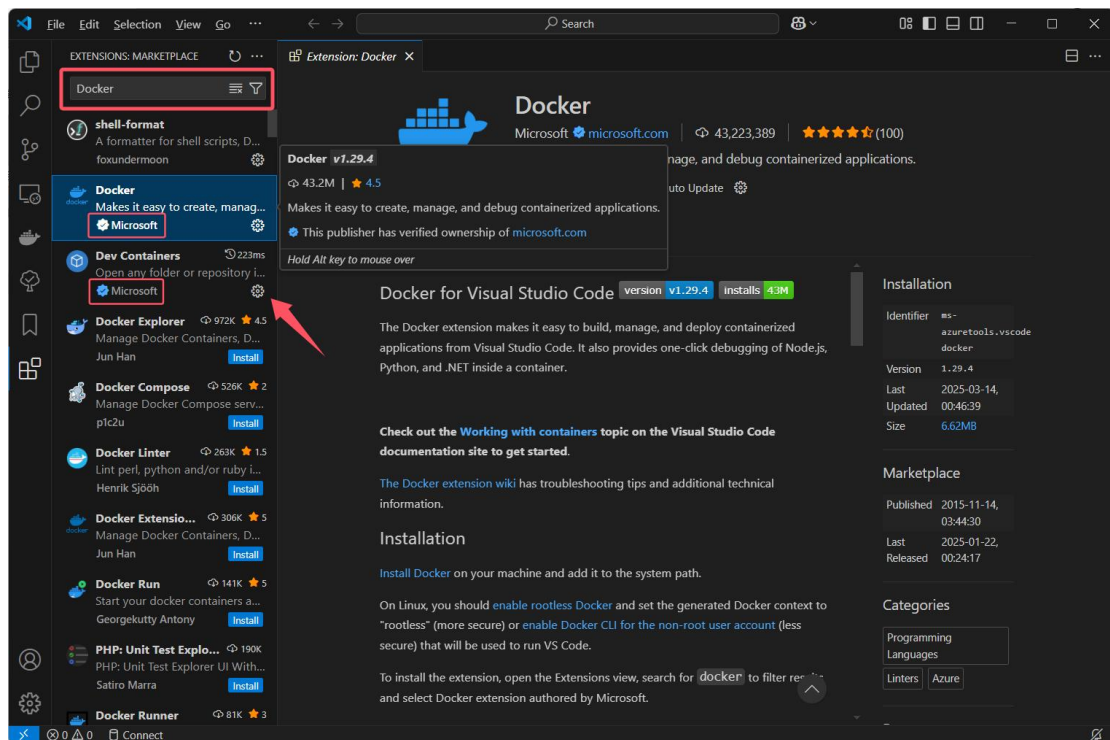


VSCode

[Download Visual Studio Code – Mac, Linux, Windows](#)



1. 按照上图找到插件搜索界面
2. 在搜索栏输入 “Docker”
3. 选择下载下图标出的两款插件 “Docker” 与 “Dev Containers”



注意需要有“Microsoft”认证：

安装完成后可以在左侧看到两款插件的图标：



Docker Desktop

Windows 系统：[Windows | Docker Docs](#)

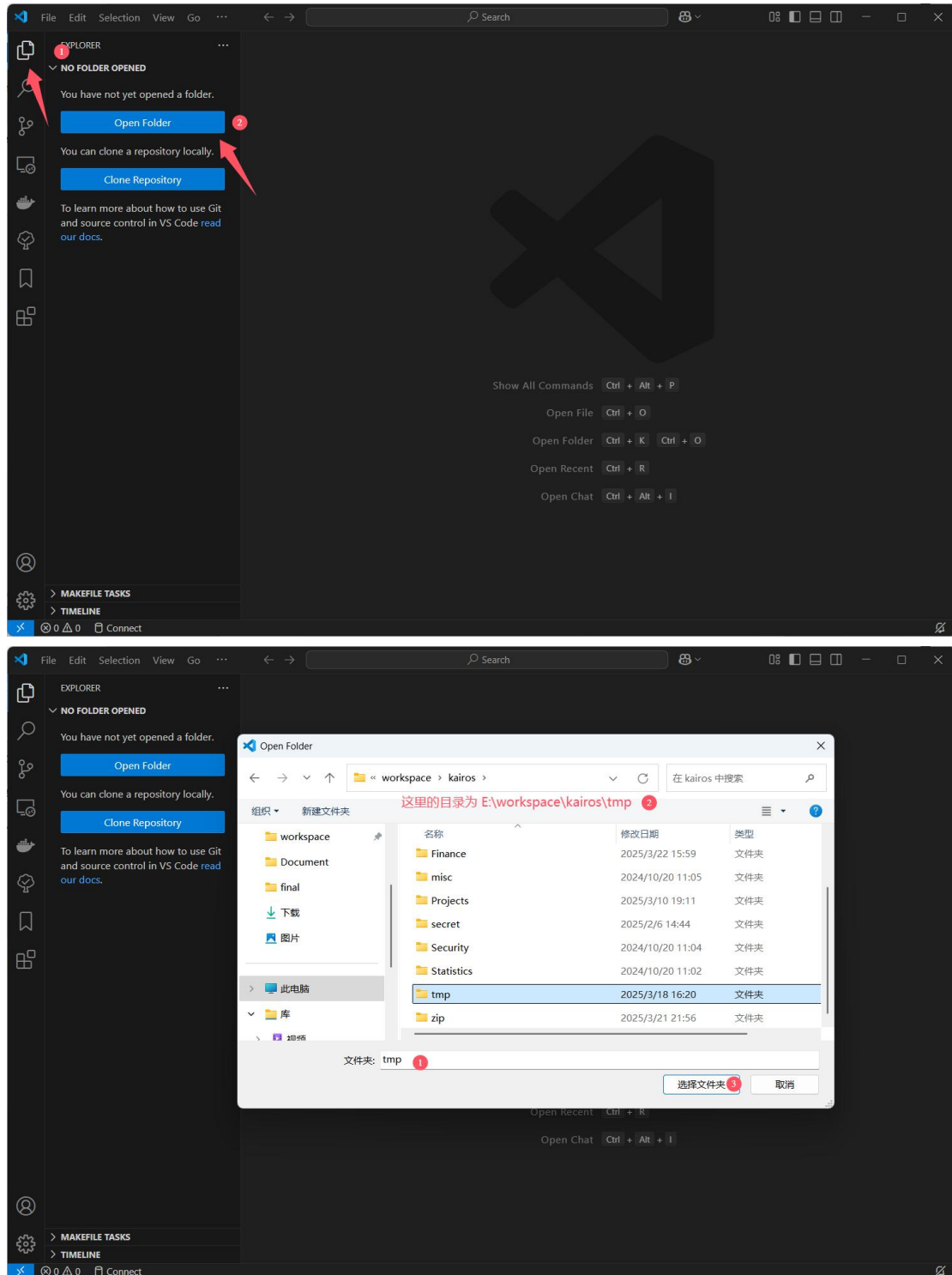
MacOS：[Mac | Docker Docs](#)

Linux：[Linux | Docker Docs](#)

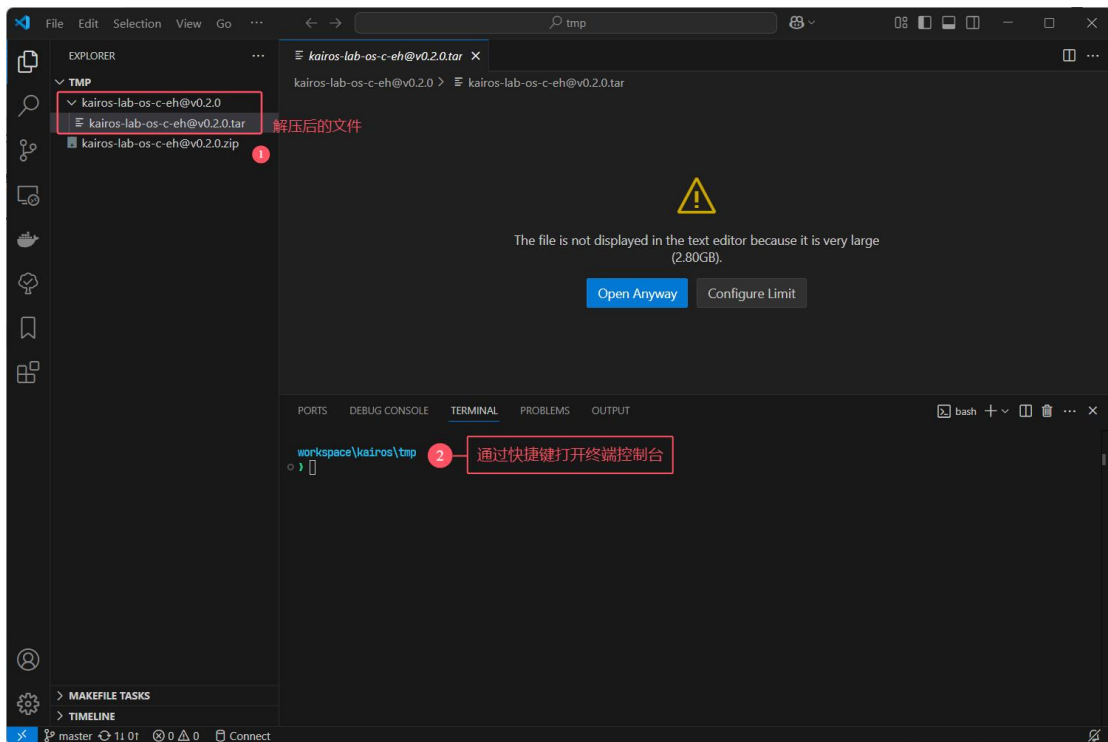
本地搭建实验环境

1. 打开安装好的“Docker Desktop”，这会同步启动“Docker Engine”，开启后放置在后台即可

2. 打开“VSCode”，选择一个没有中文，文件夹名不带空格的目录



3. 将下载好的镜像解压到 VSCode 打开的目录中，并通过快捷键 **Ctrl + `** 打开 VSCode 终端界面



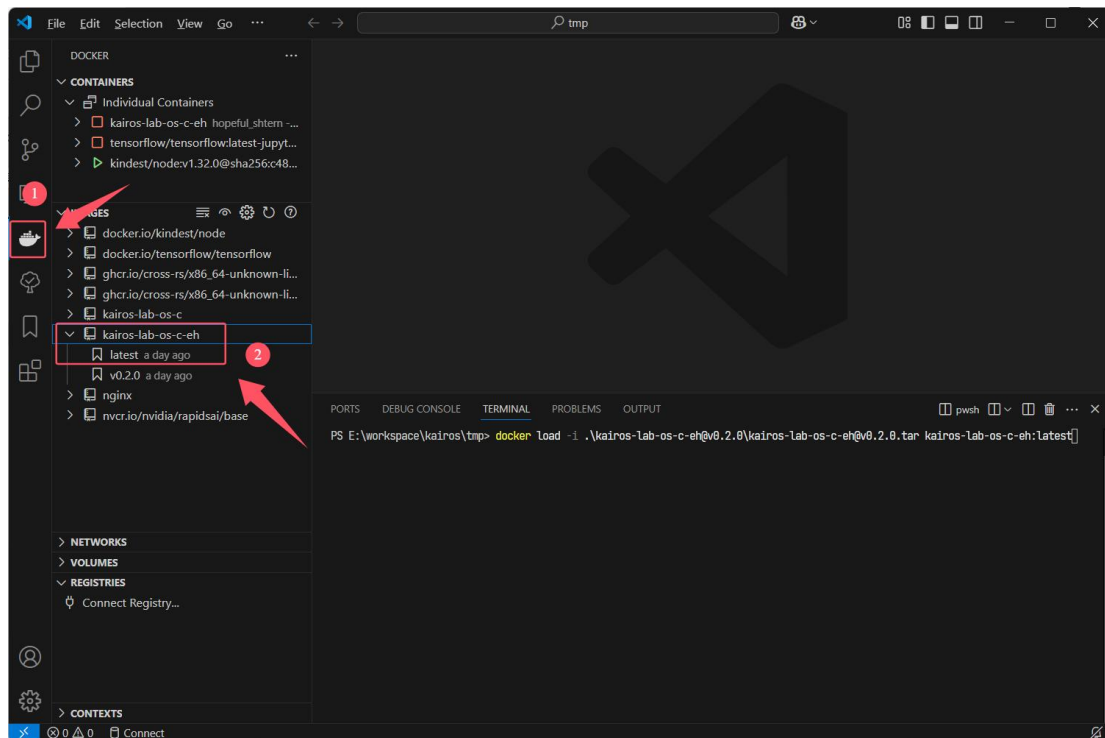
4. 终端输入命令导入容器镜像：

```
docker load -i .\kairos-lab-os-c-eh@v0.2.0\kairos-lab-os-c-eh@v0.2.0.tar
```

```
kairos-lab-os-c-eh:latest
```

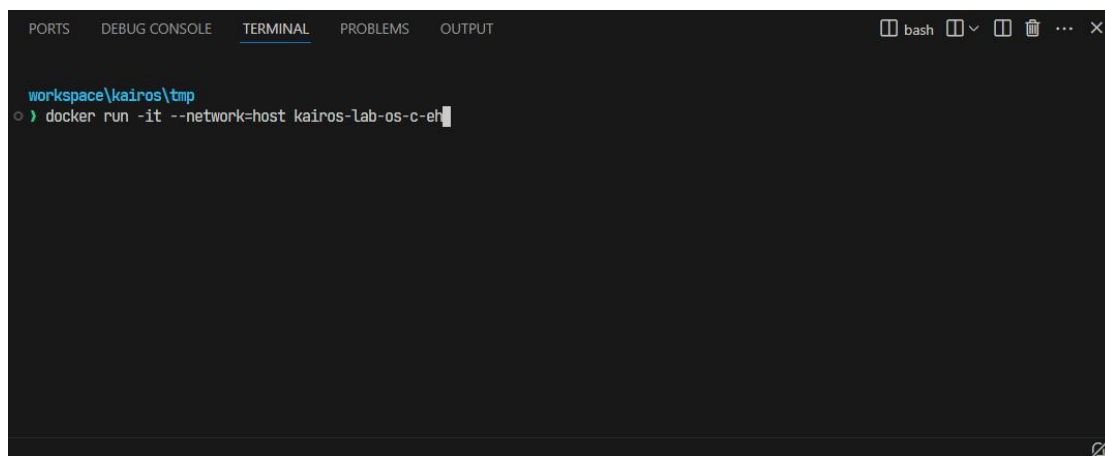
时间会比较长，请耐心等待

5. 点击左侧 VSCode Docker 插件，查看已经导入的镜像

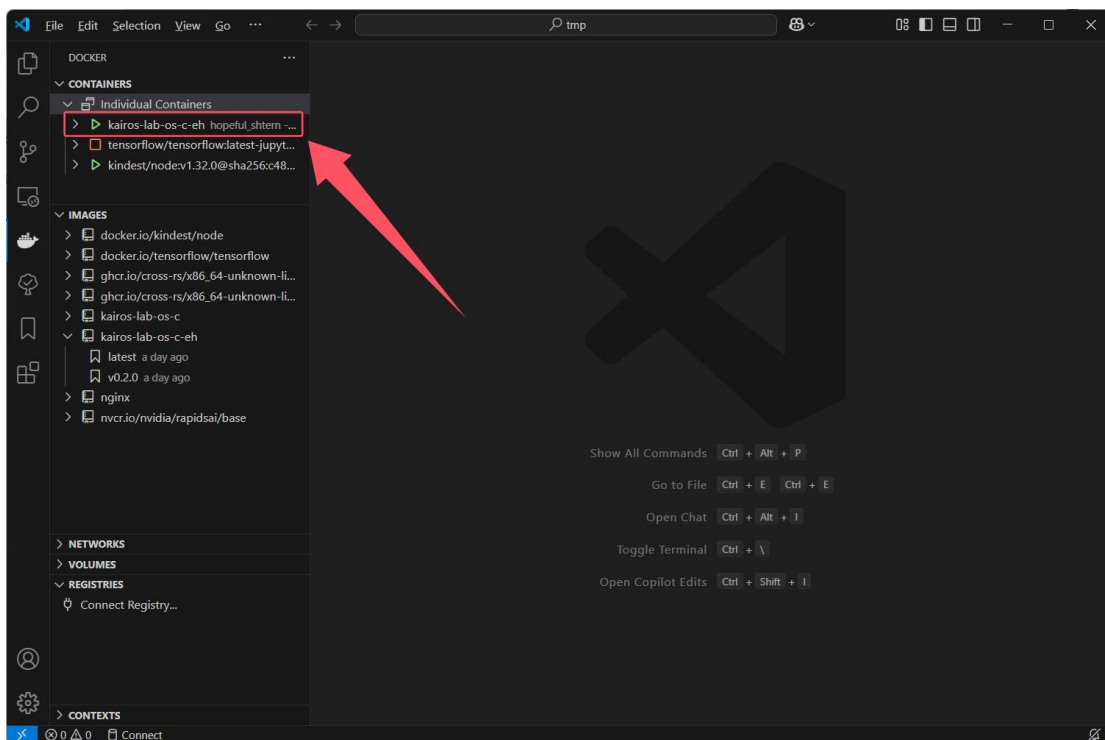


6. 在控制台输入：

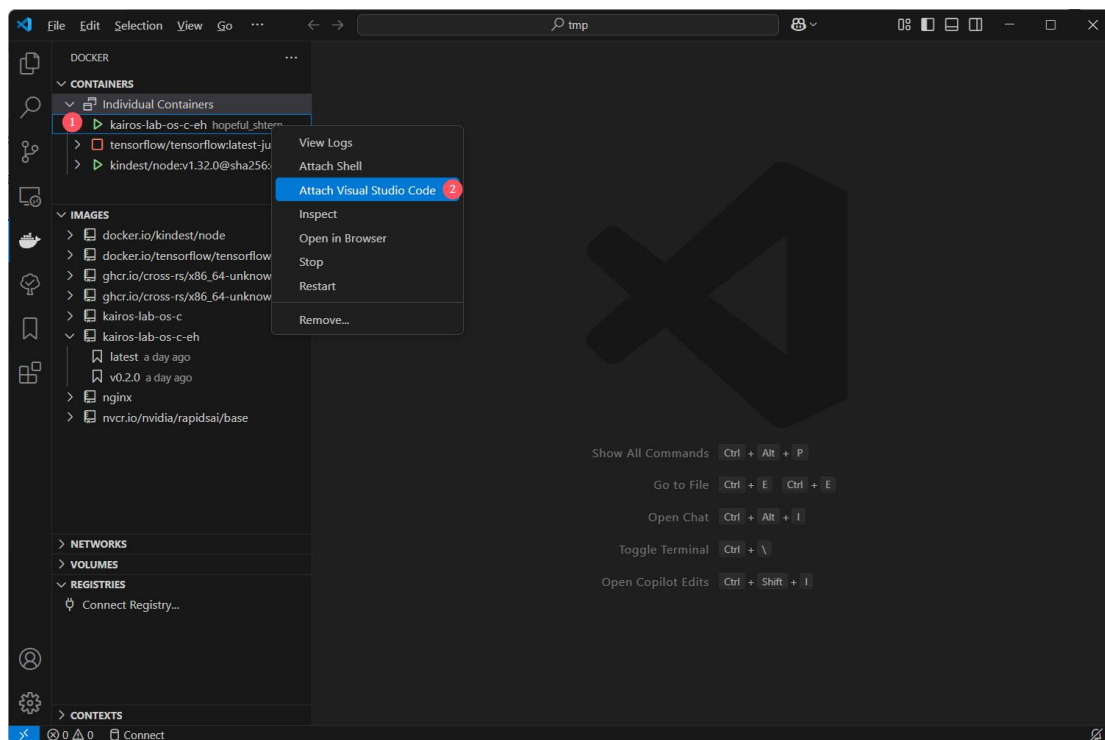
```
docker run -it --network=host kairos-lab-os-c-eh
```



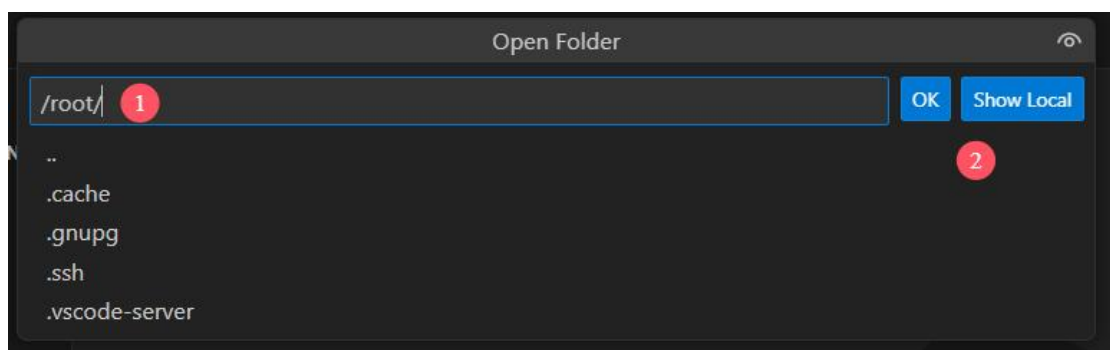
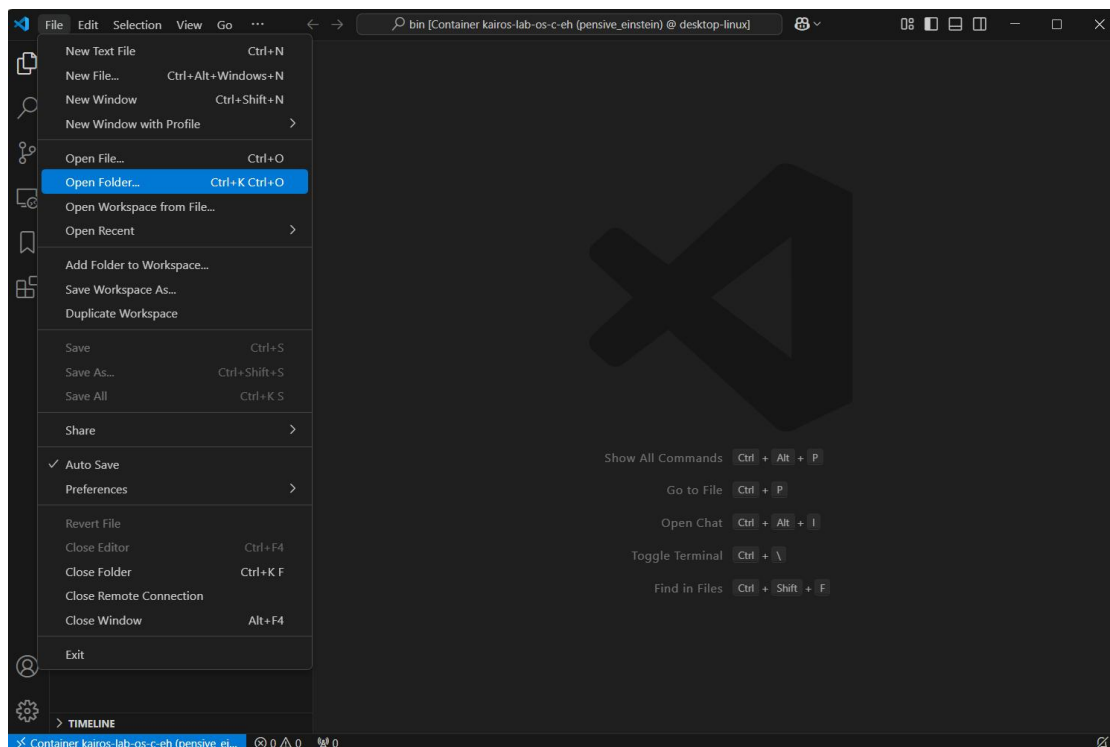
运行成功后可以在插件界面看到正在运行的容器：



7. 右键正在运行的“kairos-lab-os-c-eh”，选择“Attach Visual Studio Code”，等待一会后会弹出一个新 VSCode 窗口，该窗口即是容器内 VSCode 环境

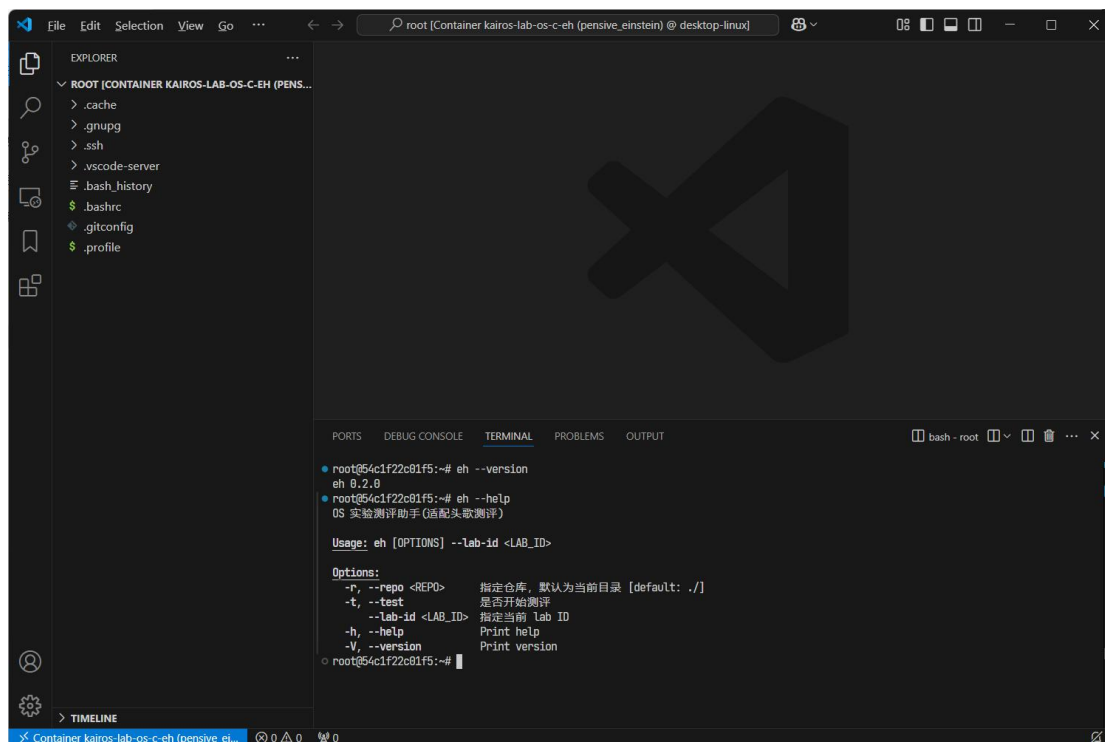


通过左上角的“File”->“Open Folder...”，回到 root 用户根目录：（输入 “/root/”，单击“OK”）



8. 到此，本地环境基本搭建完成，可以使用下面的命令查看当前环境中用于测评的工具版本（使用 **Ctrl + `** 打开控制台）

```
eh --verison
```

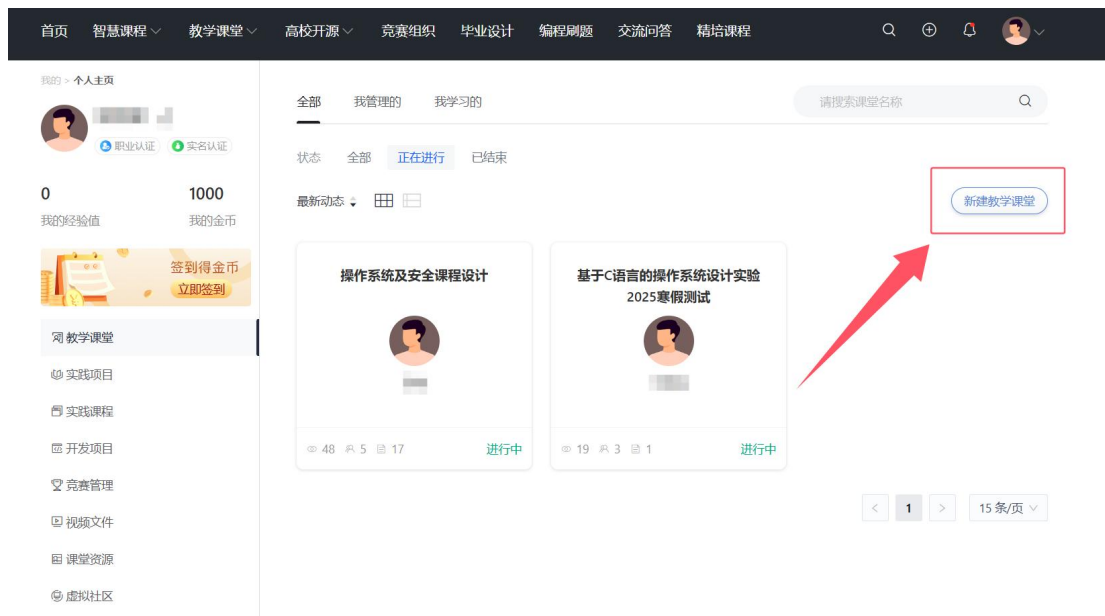



头歌平台使用

1. 鼠标移动到右上角头像处，选择“我的教学课堂”



2. 选择右侧“新建课堂”



3. 完成教学课堂的创建

教学课堂 > 创建课堂

创建课堂 [返回](#)

* 课程名称

例如: 数据结构 0/60

正确示例: 数据结构
错误示例: 数据结构2025春

* 课堂名称

例如: 数据结构2016秋季班级 0/60

正确示例: 数据结构2025春季班级
错误示例: 2025春季班级数据结构

总学时

例如: 30 0/5

学分

例如: 3 0/5

结束时间

请选择结束时间

课堂模块

4. 课堂添加实验，选择从“实践课程添加”，选择“基于 C 语言的 OS 实验”，而**不是**“基于 C 语言的 OS **设计**实验”

首页

智慧课程

教学课堂

高校开源

竞赛组织

毕业设计

编程刷题

交流问答

精培课程

基于C语言的操作系统设计实验2025寒假测试

私有

邀请码:

分享课堂

当前身份: 老师 | 课堂管理

(杭州电子科技大学)

成员管理 | 学生 0 | 教师 3

添加老师 | 添加助教 | 添加学生

通知公告

课堂实验 1

图文作业

分组作业

编程作业

在线考试

知识图谱

学习路径

全部 0

未发布 0

未开始 0

进行中 0

已截止 0

+ 选用实验 2

默认排序

请输入实验名称

添加老师 | 添加助教 | 添加学生

0

已截止 0

+

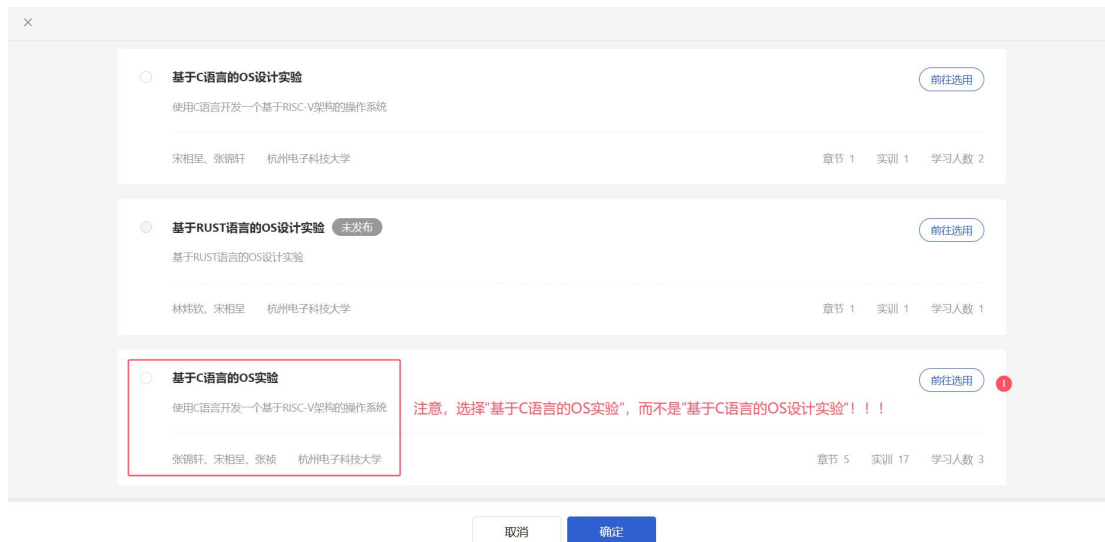
选用实验

请输入实验名称

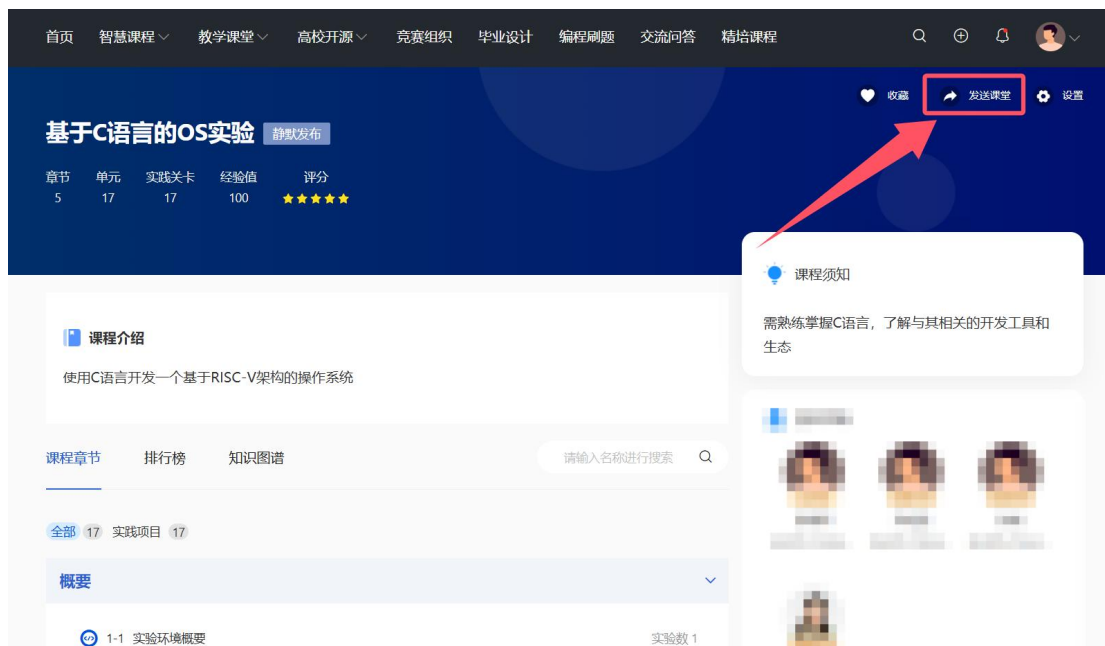
从实践项目添加

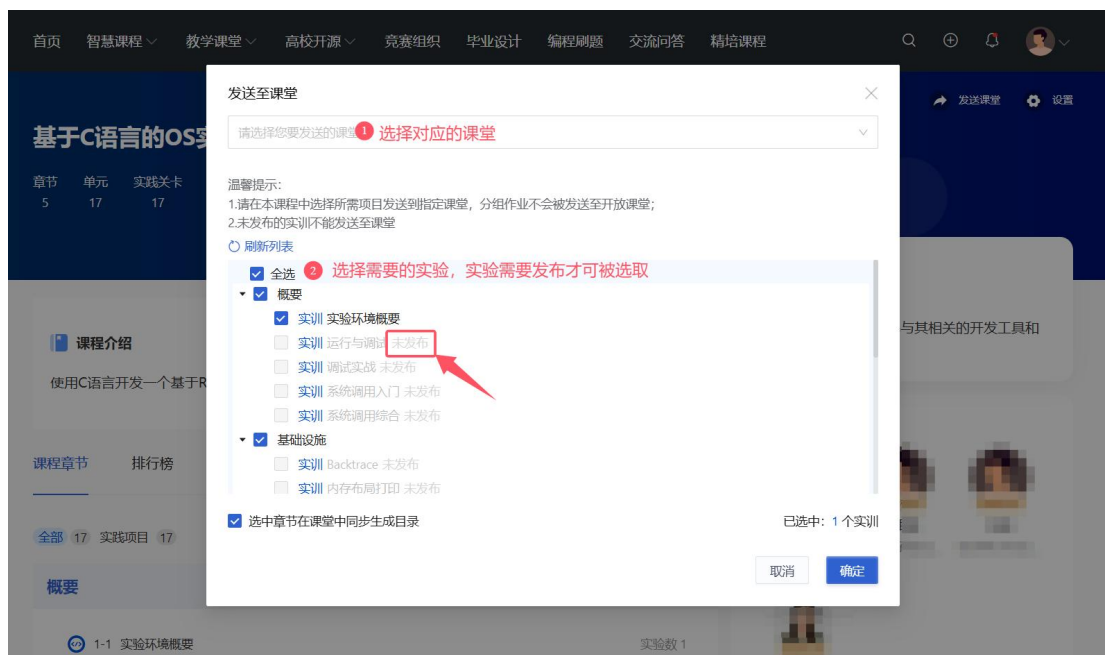
从实践课程添加

导入外部成绩

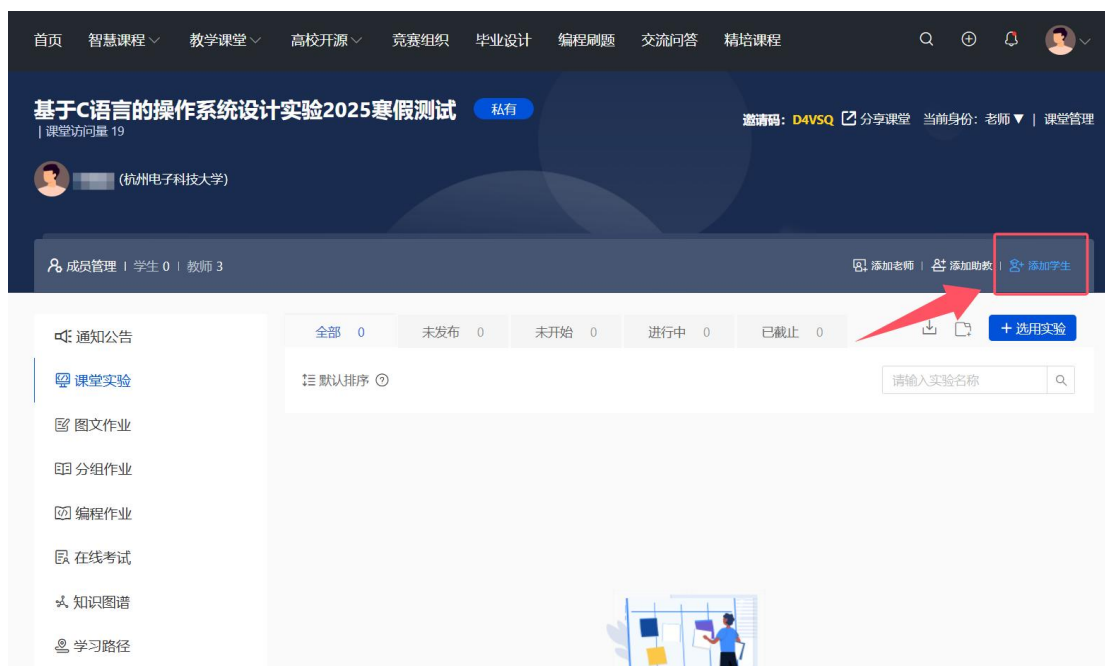


点击右上角的“发送课堂”





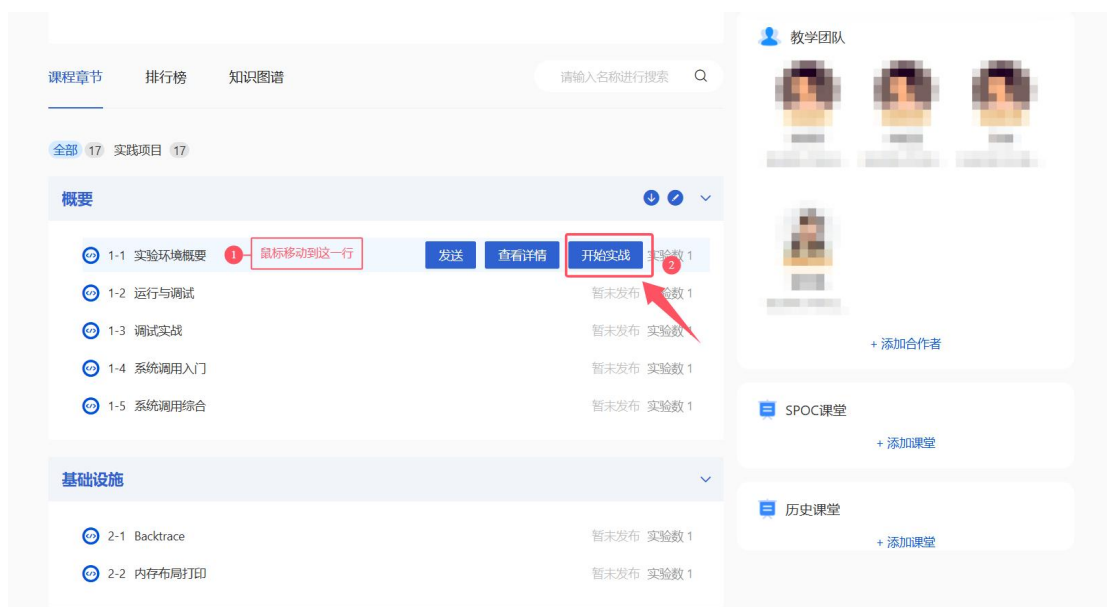
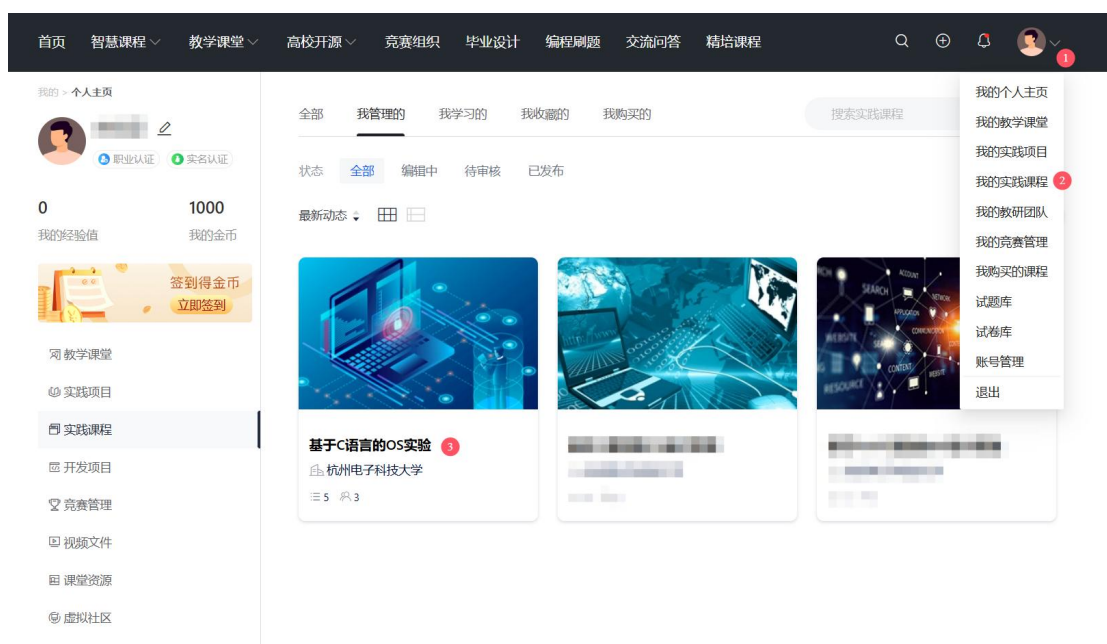
5. 课堂批量添加学生：回到先前课堂界面，点击右侧“添加学生”，在弹出的窗口中选择“批量导入”





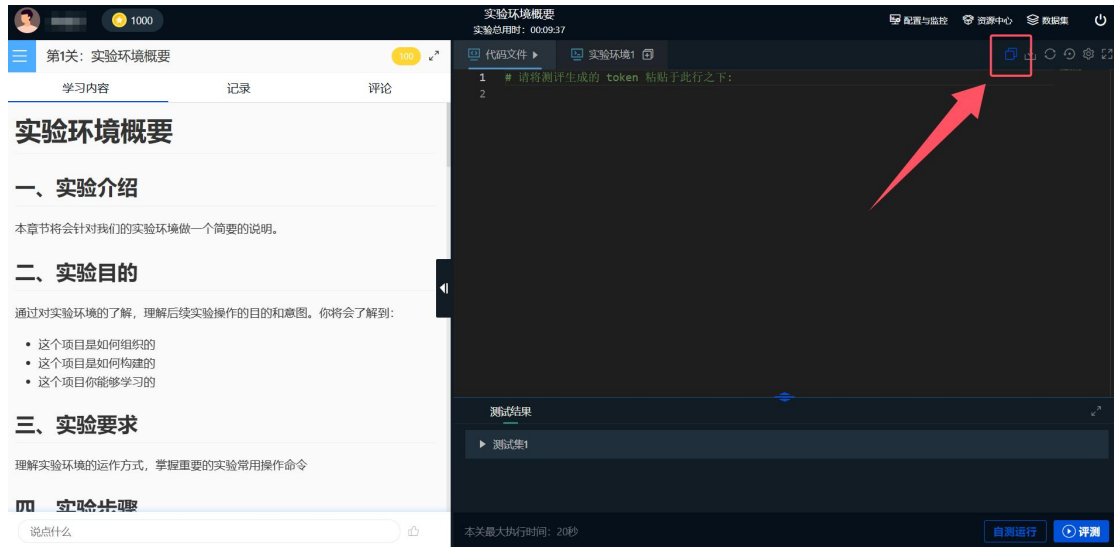
完整实验过程

1. 根据之前的章节，搭建好本地环境
2. 通过头歌平台获取实验代码
 - (1) 找到课堂对应的实验



根据身份和权限的不同以及实验是否发布，鼠标移动上去后显示的内容大同小异

(2) 点击开始实战，通过右上角的按钮获取当前实验的 Git 仓库地址



(3) 回到 VSCode 的本地容器环境中，使用 Git 拉取代码到本地

`git clone <获取到的 Git 仓库地址> os-lab`

```
PORTS  DEBUG CONSOLE  TERMINAL  PROBLEMS  OUTPUT
root@docker-desktop:~# git clone https://git.educoder.net/... .git os-lab
Cloning into 'os-lab'...
remote: Enumerating objects: 287, done.
remote: Counting objects: 100% (287/287), done.
remote: Compressing objects: 100% (270/270), done.
remote: Total 287 (delta 5), reused 280 (delta 2), pack-reused 0
Receiving objects: 100% (287/287), 398.98 KiB | 10.23 MiB/s, done.
Resolving deltas: 100% (5/5), done.
root@docker-desktop:~#
```

(4) 执行下面的命令，让 VSCode 位于新拉取的 Git 仓库中：

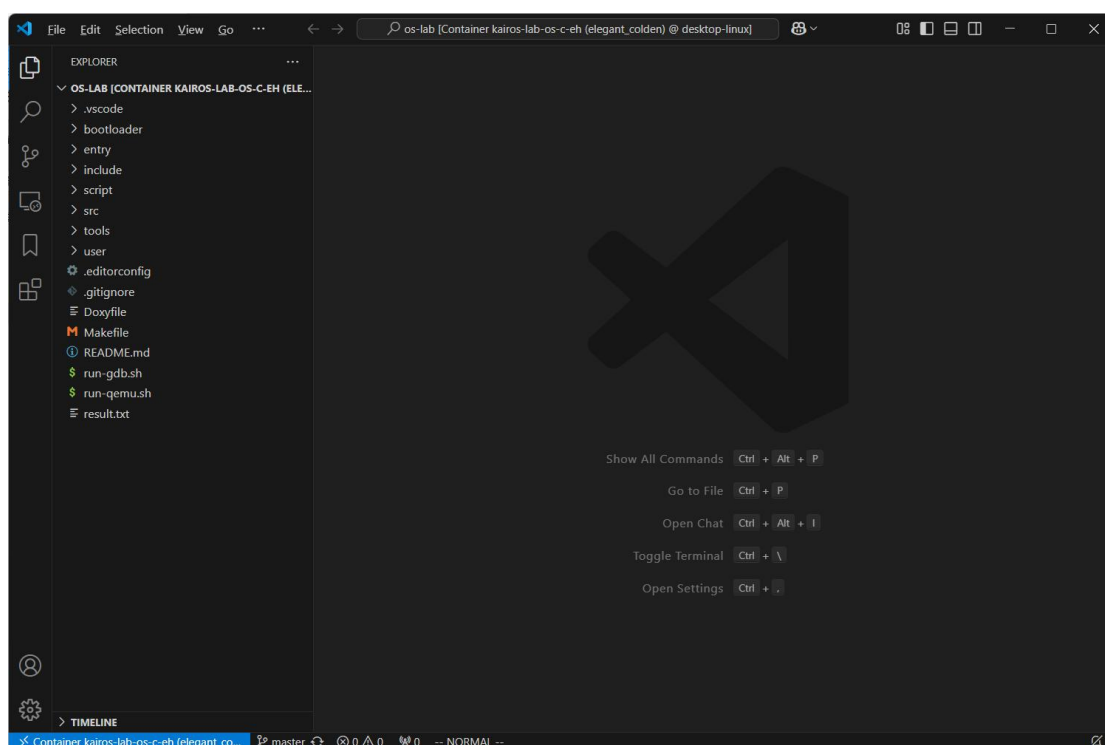
`cd os-lab`

`code .`

注意上一行有一个“.”，英文输入状态下的句号

```
root@docker-desktop:~# git clone https://git.educoder.net/pprq4s586/efvx6b5t8720250321152905.git os-lab
Cloning into 'os-lab'...
remote: Enumerating objects: 287, done.
remote: Counting objects: 100% (287/287), done.
remote: Compressing objects: 100% (270/270), done.
remote: Total 287 (delta 5), reused 280 (delta 2), pack-reused 0
Receiving objects: 100% (287/287), 398.98 KiB | 10.23 MiB/s, done.
Resolving deltas: 100% (5/5), done.
root@docker-desktop:~# cd os-lab/
root@docker-desktop:~/os-lab# code .
root@docker-desktop:~/os-lab#
```

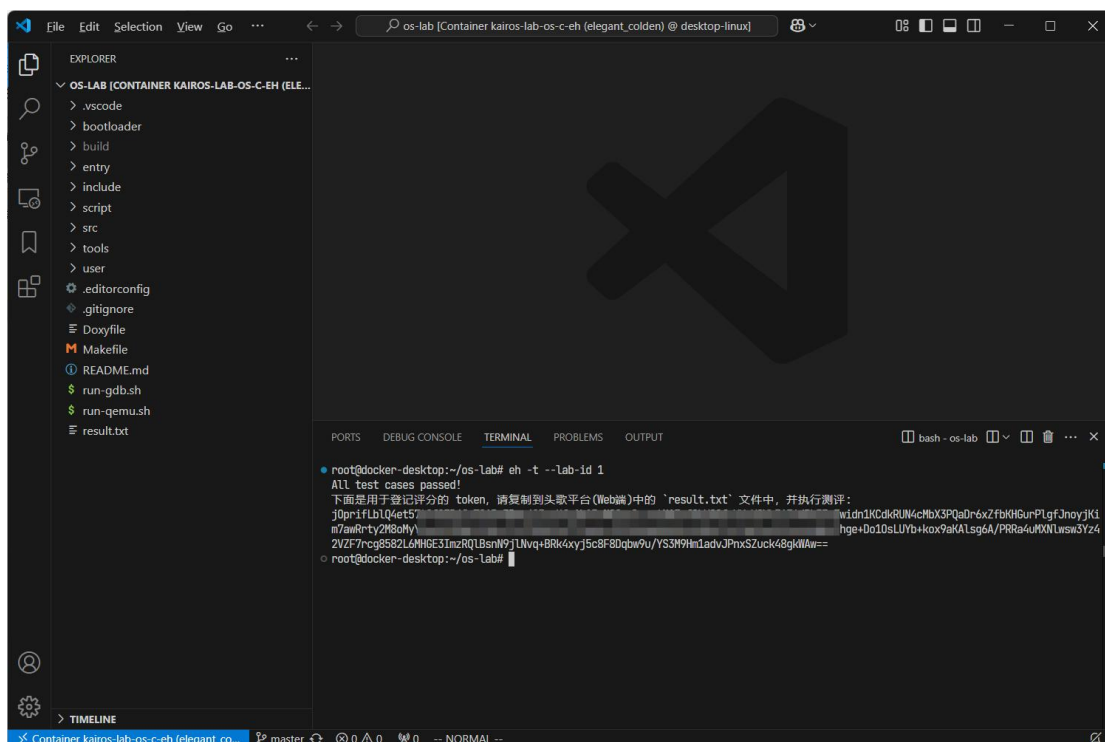

上述操作后会打开一个新的 VSCode 界面，后续实验过程均在这个界面中

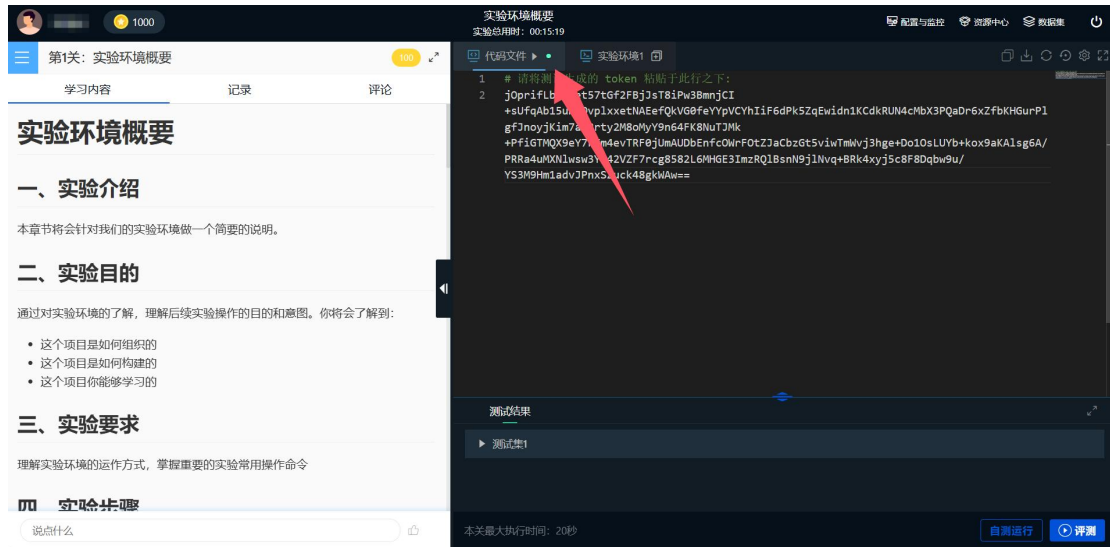


3. 完成实验，使用“eh”完成打分，输入指令：

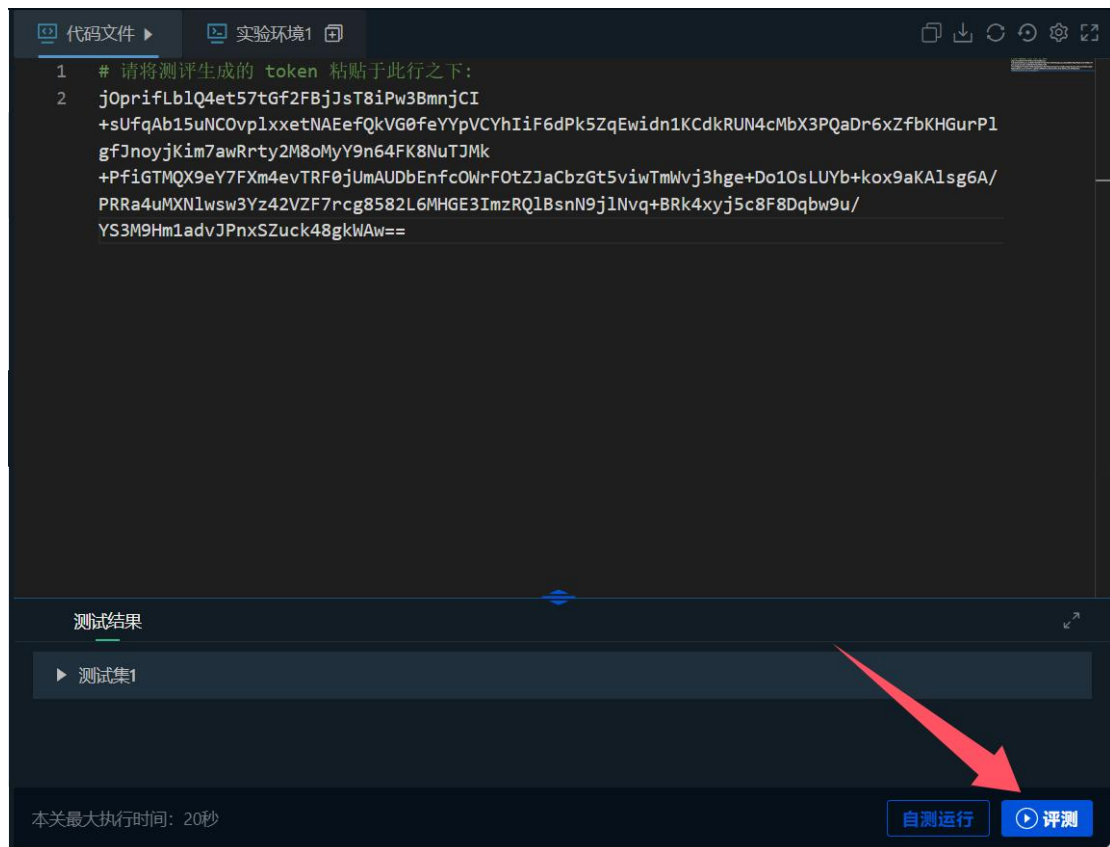
```
eh -t --lab-id 1
```

这会生成一个 token，复制该 token 到头歌平台，提交实验，即可完成打分





注意这里的绿点，等其消失后则说明文件已经保存，点击右下角的测评



代码文件

实验环境1

📄

📄

🔄

🔄

⚙️

🔍

1

请将测评生成的 token 粘贴于此行之下:

2

j0prifLb1Q4et57t6f2FBjJsT8iPw3BmnjCI

+sUfqAb15uNCOvplxxetNAEefQkVG0feYYpVCYhIiF6dPk5ZqEwidn1KCdkRUN4cMbX3PQaDr6xZfbKHGurP1

gfJnoyjKim7awRrty2M8oMyY9n64FK8NuTJMk

+Pfi6TMQX9eY7FXm4evTRF0jUmAUDbEnfcOWrF0tZJaCbzGt5viwTmWvj3hge+Do10sLUYb+kox9aKA1sg6A/

PRRa4uMXN1wsw3Yz42VZF7rcg8582L6MHGE3ImzRQ1BsnN9j1Nvq+BRk4xyj5c8F8DqbW9u/

YS3M9Hm1advJPnxSZuck48gkWAw==

测试结果

📄

🟢 1/1 全部通过

▶ 测试集1

消耗内存84.52MB 代码执行时长: 0.41秒

🟢

本关最大执行时间: 20秒 本次评测耗时(编译、运行总时间): 0.642 秒

自测运行

6S