

基本工具配置

📅 创建时间: 2025-07-02

1. 比赛环境配置

参考**使用手册**(见第一部分的文档), 首先需要根据系统下载 **Docker**, 官网上下载即可。注意初次使用容器需要配置好 **WSL 2**, 整体流程:

1. 如果从来没有下载过 **WSL 2**, 先下载安装, [下载网址](#), 下载安装后**重启电脑**
2. 重启后, 以管理员身份打开 **Powershell**, 分别输入以下命令, 完成后重启电脑:

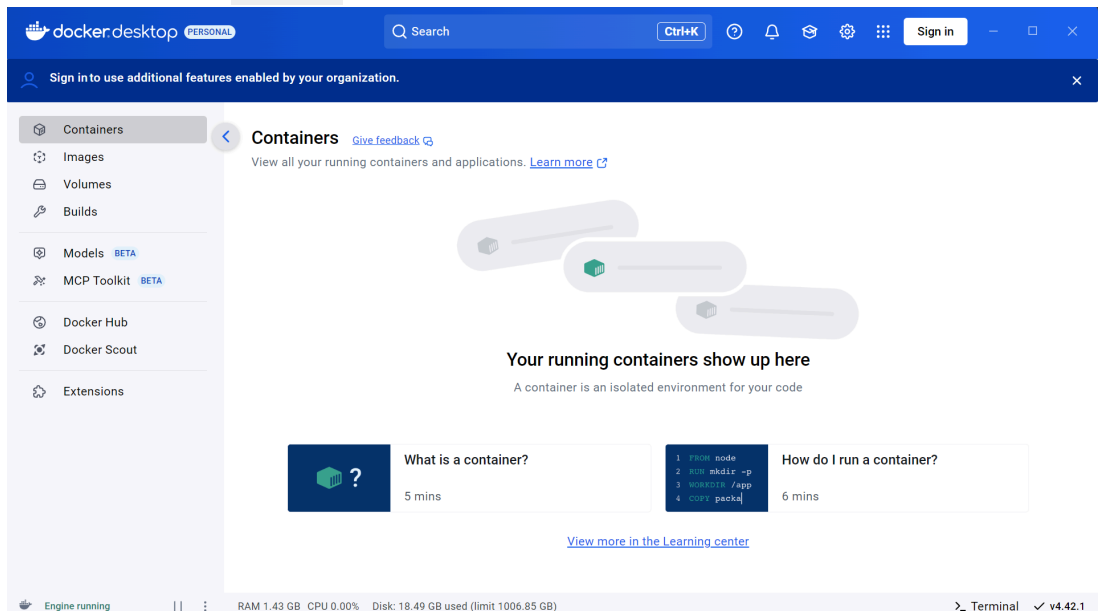
```
dism /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
dism /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart
```

3. 重启后, 同样管理员权限打开 **Powershell**, 输入以下命令:

```
wsl --version
wsl -l -v
```

如果出现正常的版号信息, 说明成功完成

4. 打开之前下载好的 **Docker**, 完成注册登录后, 点击运行, 应该会得到如下的情况:



, 注意在后序的开发训练中, **Docker** 需要挂在后台

5. 进入下载好的客户端, 按照使用手册配置好相应的工作路径, 随后启动开发环境进行初次配置, 注意在使用GPU的要下载好相应的驱动(一般是自带的)

2. 协作开发

由于本次比赛的平台无法做到共享开发代码，为了我们开发的协作性，同时也可以学习一下良好的团队开发流程，我们使用 `github` 进行协作开发，我创建了一个仓库，[仓库地址](#)，我们可以克隆，拉取仓库的代码，推送代码，同时发布相应的笔记内容与说明文档，具体操作指令如下。这个可能需要花费一定的时间去学习使用。**如果大家觉得有什么更好工具进行协作开发的可以提出来我们一起讨论一下**

一、项目初始化：首次下载（Clone）

命令：

```
git clone https://github.com/ming-lah/Optimal-Minds
cd Optimal-Minds
```

用途：将整个远程 GitHub 仓库复制到本地。

二、创建与切换分支

查看已有分支：

```
git branch          # 本地分支
git branch -r       # 远程分支
```

新建并切换分支：

```
git checkout -b feature/xxx
```

切换已有分支：

```
git checkout dev
```

首次追踪远程分支（如 dev）：

```
git checkout -b dev origin/dev
```

三、本地修改后推送

常规三步提交：

```
git add .
git commit -m "提交说明"
git push origin 分支名
```

示例：

```
git push origin feature/hello-demo
```

四、拉取远程内容同步

拉取最新内容：

```
git checkout dev          # 确保在正确分支
git pull origin dev
```

五、提交 Pull Request (合并请求)

1. 在 GitHub 页面点击 `Compare & Pull Request`
2. 选择:
 - **base 分支**: 目标 (如 dev)
 - **compare 分支**: 你的 feature/xxx 分支
3. 填写标题与描述 → 点击 `Create Pull Request`

六、团队协作开发建议

角色	操作流程
所有成员	clone 项目 → checkout dev → 新建 feature 分支
每个成员	在 feature/xxx 分支开发 → 提交并推送 → 创建 PR 到 dev
管理者/组长	审核合并 PR 到 dev → 定期将 dev 合并到 main (发布用)

七、冲突处理

处理流程:

```
git pull origin dev      # 拉取可能冲突内容
git status               # 查看冲突文件
# 手动编辑冲突文件
```

```
git add 冲突文件
git commit -m "解决冲突"
git push
```

八、其他常用命令

命令	说明
git status	查看当前更改状态
git log --oneline --graph	图形化提交历史
git remote -v	查看远程仓库地址
git stash / git stash pop	临时保存/恢复本地改动
git pull --rebase	整洁地同步远程更改

大家可以多使用一下这个功能，目前仓库里面的文件都是可以随意进行修改的，可以多练练手(其实我也太会)，同时可以探索一下Vscode中的方便的推送拉取方式，可能更方便。

3. 任务与建议

我们学习期的时间相对还是比较充裕的，而且大家考完试可能都比较累，也就不用那么着急去开始投入研究。个人觉得可以空闲下来时，或者觉得无聊了打开看一看，配置一下环境，熟练一下协作开发工具的使用。上面的笔记都是我先进行探索总结的，大家做个参考就好，可能存在错误的地方。如果大家有什么问题或者遇到什么困难都可以在群里提出来，我们一起解决就好。最后就是关于比赛的，这次比赛的想要获奖的难度可能还是有的，我觉得我们抱着一学习的心态去就好了，大家在实现的时候既可以练习开发的一些基础内容，同时开发的课题也是比较有趣的。拿不拿奖的其实不重要的，注重当下就好啦。