**操作系统教学实践云平台**

**操作系统教学实践云平台部署文档**

**北京航空航天大学**

**操作系统教学实践云平台部署文档**

**文档变更记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **时间** | **版本** | **变更内容简述** | **变更人** | **审核人** |
| 1 | 2022/8/15 | V0.1 | 文档初稿撰写 | 王廉杰 | 王雷 |
| 2 | 2022/9/12 | V0.2 | 增加静态扫描结果 | 王廉杰 | 王雷 |
| 3 | 2022/9/15 | V0.3 | 增加第三方软件 | 王廉杰 | 王雷 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**目录**

[1. 系统部署 4](#_Toc114425330)

[1.1. 系统需求 4](#_Toc114425331)

[1.2. 部署准备 4](#_Toc114425332)

[1.3. 部署流程 4](#_Toc114425333)

[2. 学生环境管理 8](#_Toc114425334)

[3. 评测程序部署 9](#_Toc114425335)

[4. 静态扫描结果 10](#_Toc114425336)

[5. 第三方软件列表 10](#_Toc114425337)

# 系统部署

## 系统需求

本项目支持在arm64架构的主流Linux发行版下运行，并在openEuler 20.03 LTS环境下经过测试。运行本项目时，系统需要至少4GB的物理内存，且需要安装有以下依赖包：

* docker-engine（版本至少为20.10）
* python3（版本至少为3.7）
* curl，sqlite，openssh

除docker-engine之外的依赖在大多数环境中应当已预先安装，若未安装则可直接使用系统包管理器安装，如在openEuler中使用dnf：

$ dnf install python3 curl sqlite openssh

若系统源未提供Docker Engine或提供的版本过低，可从<https://docs.docker.com/engine/install/binaries>下载并安装其二进制程序。

## 部署准备

为了加速从Docker Hub拉取镜像的过程，可在系统中创建并编辑 /etc/docker/daemon.json文件，在其中配置国内镜像源：

{

"registry-mirrors": [

"https://hub-mirror.c.163.com",

"https://mirror.baidubce.com"

]

}

## 部署流程

下载打包好的平台程序和实验工具链，将oslab.tar.gz解压至 /opt/oslab目录，实验工具链toolchain.tar.gz解压至 /OSLAB目录：

$ mkdir /opt/oslab

$ tar xzf oslab.tar.gz -C /opt/oslab

$ mkdir /OSLAB

$ tar xzf toolchain.tar.gz -C /OSLAB

切换到项目目录 /opt/oslab，运行install/build.sh，开始构建镜像：

$ cd /opt/oslab

$ install/build.sh

构建镜像的过程需要拉取基础镜像并安装必要的软件包，耗时可能较长。构建完成后，编辑项目目录下的环境配置文件 /opt/oslab/config.ini，修改其中的前三行配置：

PUBLIC\_HOST=*122.9.36.214*

PUBLIC\_HOST指定服务被外部访问时使用的IP地址或域名，一般为本机的公网IP，仅面向内网师生时也可为本机的内网IP。

DOCKER\_GLOBAL\_HOST=*192.168.0.138*

DOCKER\_GLOBAL\_HOST用于指定学生容器访问主机时使用的IP地址，一般为本机的内网IP。

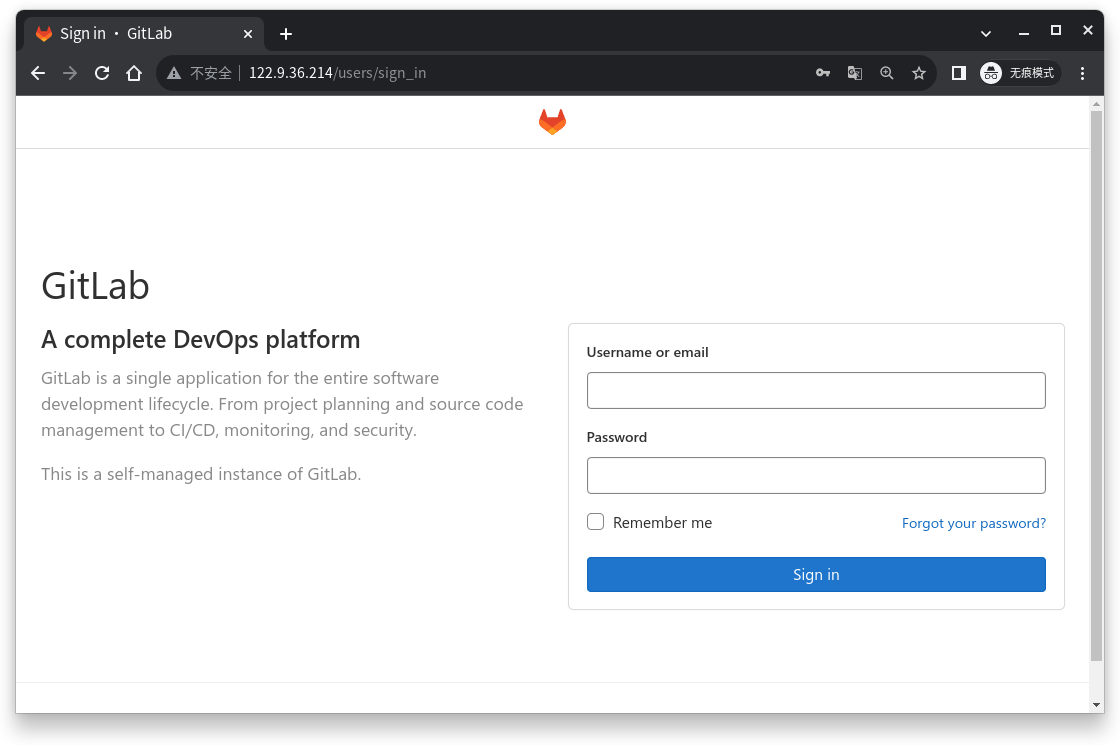
GITLAB\_SSH\_PORT=*22*

GITLAB\_SSH\_PORT指定Git服务器需要监听的SSH端口，用于学生容器访问代码仓库。若本机已运行了监听22端口的SSH服务器，可以在 /etc/ssh/sshd\_config 中将本机的sshd服务器配置为监听其他端口，或修改此处的配置，让Git服务器监听其他端口。

配置完成后，运行install/start.sh，启动服务容器：

$ install/start.sh

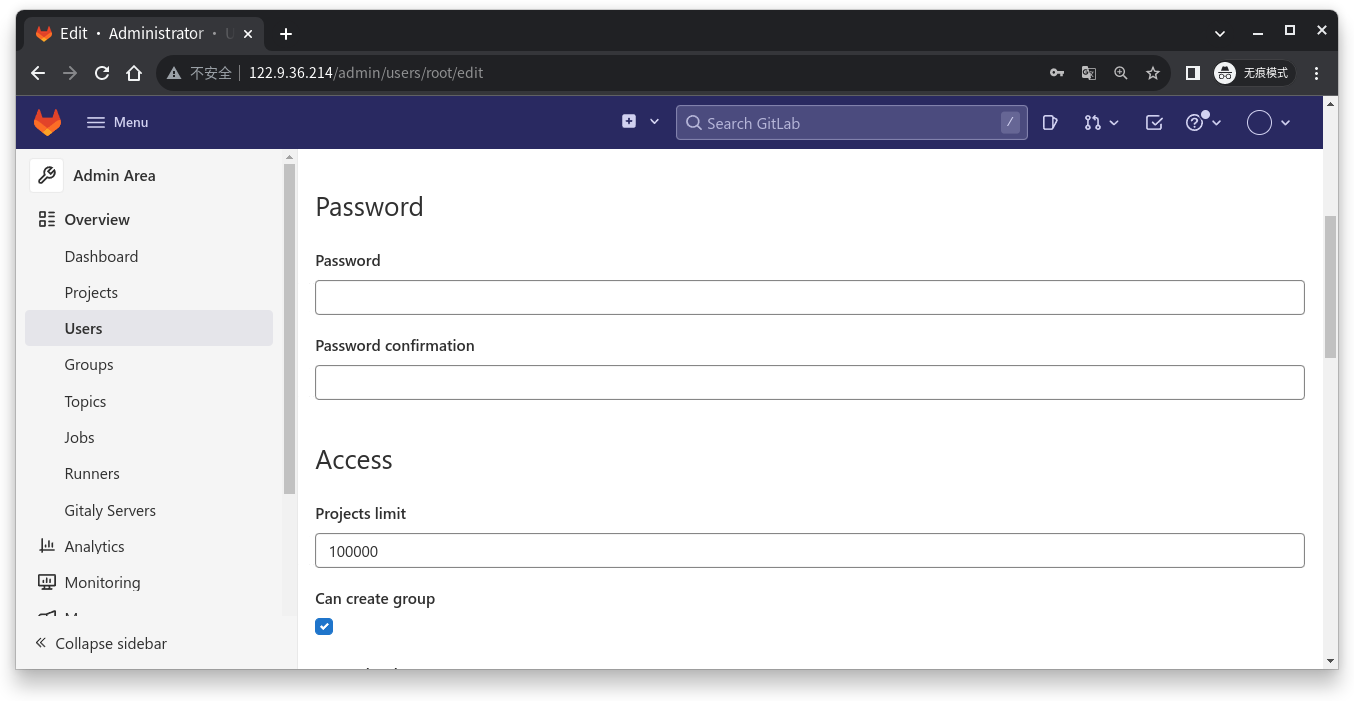
该过程会拉取必要的服务镜像，耗时可能较长。服务启动后，外部通过浏览器访问该机器的IP地址（http://*<PUBLIC\_HOST>*），应当能够看到GitLab的登录页面：



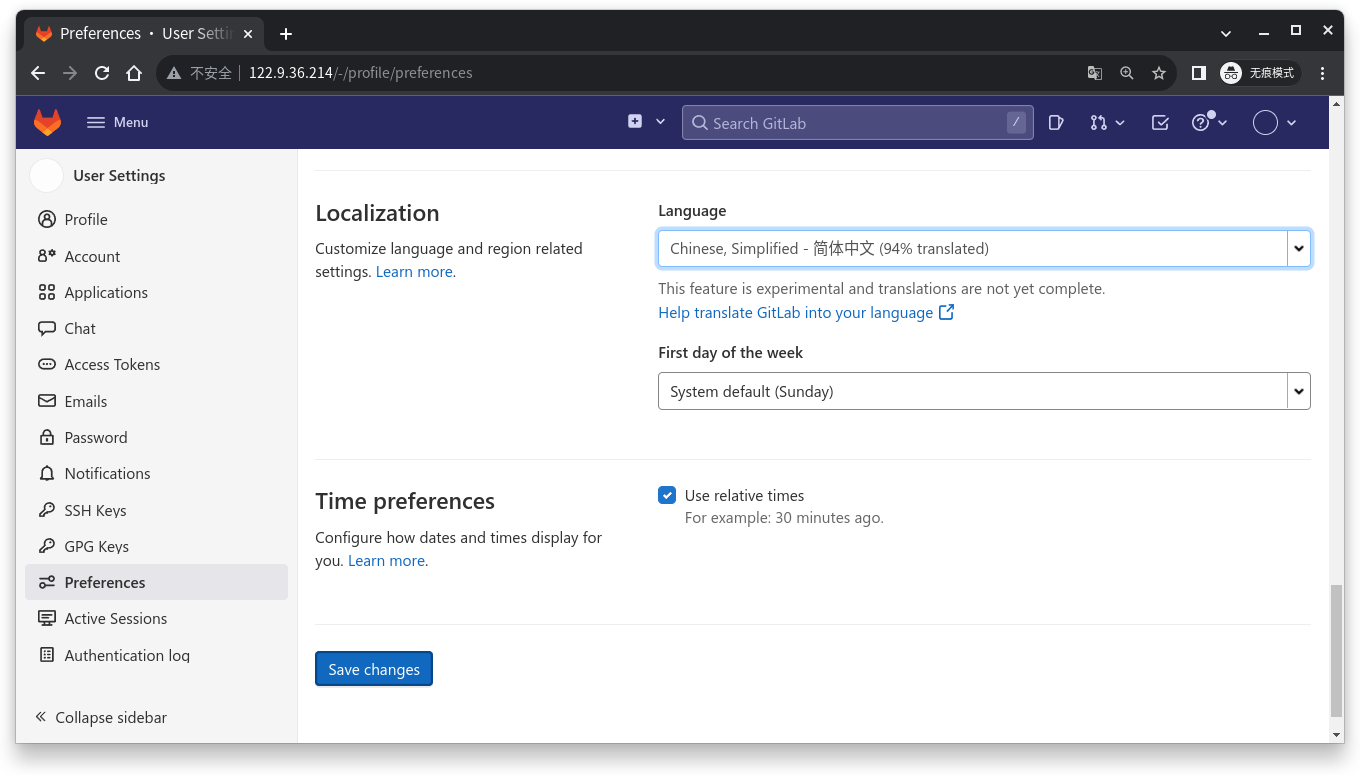
若服务启动数分钟后仍不能正常访问此页面，可能为arm64平台下低版本Linux内核存在的漏洞引起，此时可尝试升级内核，或运行以下命令来临时绕过此问题（可能需要几分钟时间生效）：

$ echo ignore-warnings ARM64-COW-BUG >> /srv/gitlab-ce/data/redis/redis.conf

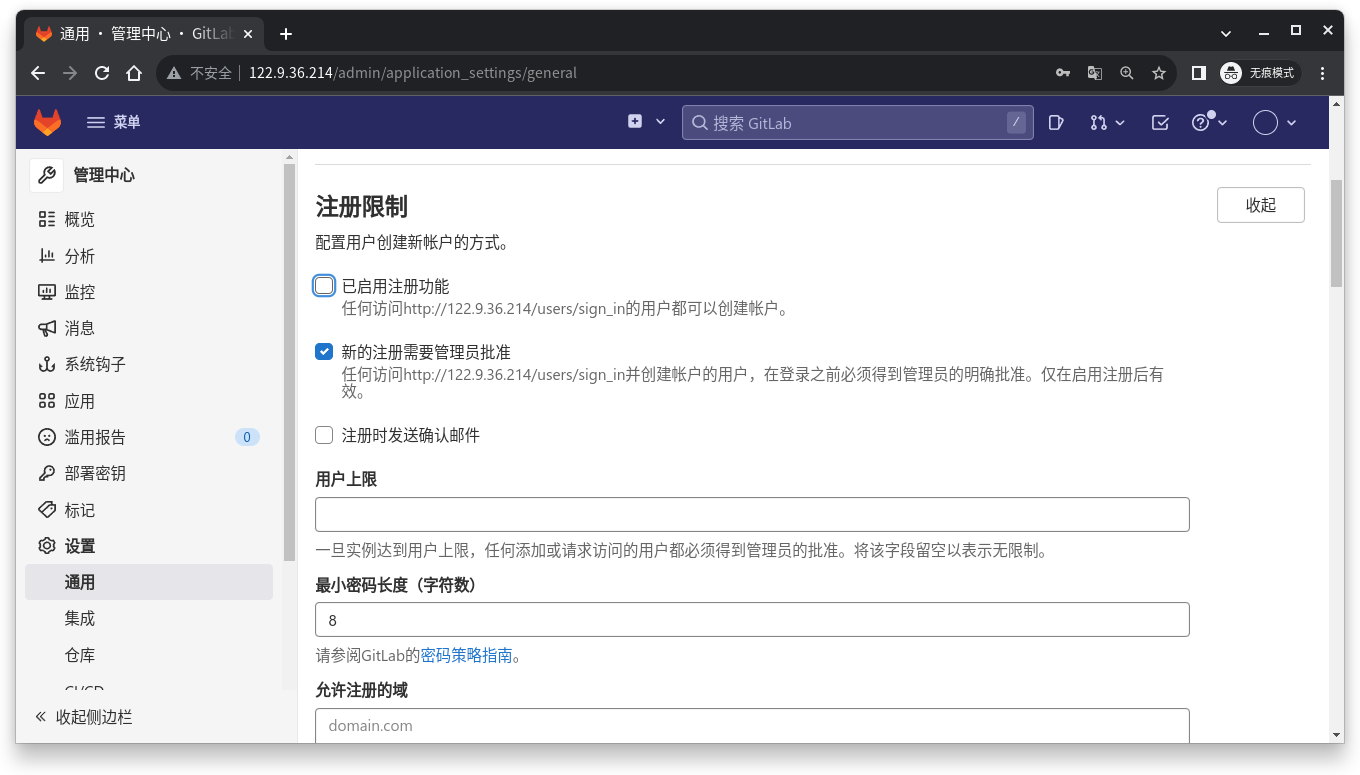
服务启动后，在浏览器中使用初始用户名root和密码w9uSjydOC1IIl5irALckxC8hrF登录到GitLab，再进入http://*<PUBLIC\_HOST>*/admin/users/root/edit页面，设置一个新的管理员密码并保存：



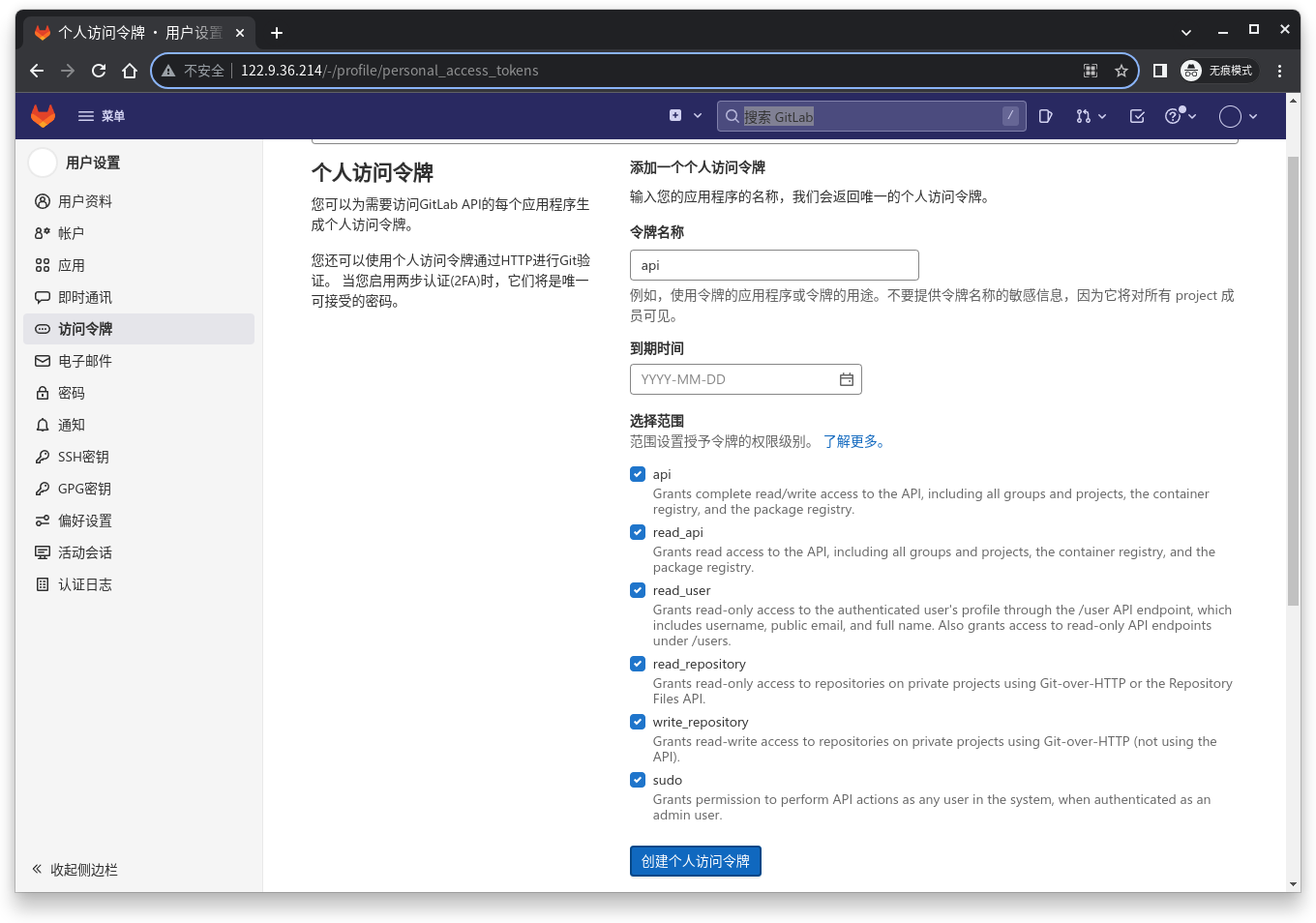
进入http://*<PUBLIC\_HOST>*/-/profile/preferences页面，可设置环境语言为中文：



进入http://*<PUBLIC\_HOST>*/admin/application\_settings/general页面，禁用直接注册功能，并保存更改：



进入http://*<PUBLIC\_HOST>*/-/profile/personal\_access\_tokens页面，将令牌名称设置为api，勾选所有范围，点击创建令牌，并将生成的令牌复制到剪切板：



编辑项目目录下的配置文件 /opt/oslab/config.ini，将复制的访问令牌设置在第四行：

GITLAB\_PRIVATE\_TOKEN=*nuzxoVMpwna6hBB6GypE*

编辑信息统计服务的配置文件 /opt/oslab/board.json，在以下行处设置信息统计平台中教师用户的密码：

"password": "inTDPC8oHCxvK2S7TmJv"

运行项目目录下的install/initialize.sh，启动评测服务：

$ cd /opt/oslab

$ install/initialize.sh

所有服务启动后，用浏览器访问信息统计平台http://*<PUBLIC\_HOST>*:81，使用刚才设置的密码登录后，应当能够看到统计页面：



可以运行docker ps -a ，检查各服务容器的状态：



目前该平台已在华为云提供的弹性云服务器上部署为测试环境，可通过ssh root@122.9.36.214 -p 19222登录该服务器进行配置，密码为uzAu3V55suKVL9yuCJLT。

# 学生环境管理

使用 /opt/oslab下的管理程序，可以快速为学生创建或删除实验环境，其使用方式为：

$ /opt/oslab/create-student <学生ID> [学生姓名] [学生密码]

$ /opt/oslab/delete-student <学生ID>

未指定学生姓名时，默认与学号相同；未指定学生密码时，将为学生自动生成随机口令。

例如，为学号为21060001的学生创建环境：

$ /opt/oslab/create-student 21060001

为学号为21060001的学生张三创建环境：

$ /opt/oslab/create-student 21060001 张三

删除学号为21060001的学生：

$ /opt/oslab/delete-student 21060001

成功创建学生后，可以在输出中看到为学生分配的SSH和Web IDE端口及相应的登录方式，以及生成的登录密码，例如：

SSH port: 13008 (ssh git@122.9.36.214 -p 13008)

Web IDE port: 23008 (http://122.9.36.214:23008)

Password: 4MoyiNUeBY8rVWJz

运行cat /opt/oslab/data/students命令，可以查看已添加的所有学生的登录信息。

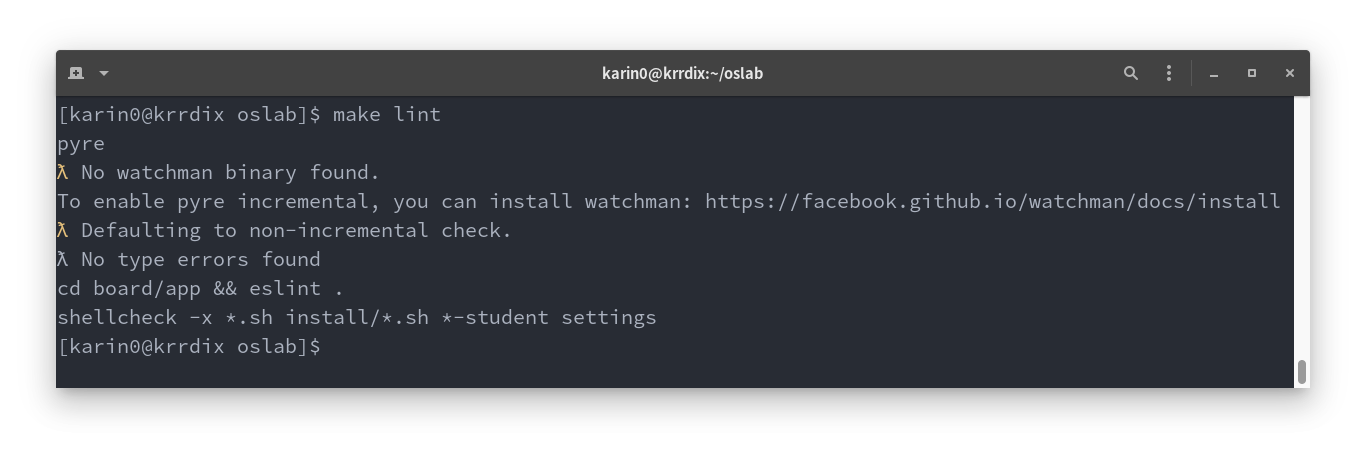
# 评测程序部署

本系统在学生仓库中使用Git分支模型管理学生在各个实验中的代码，例如学生在学生仓库的*lab-name*分支推送代码（git push）后，平台会自动调用位于 /opt/oslab/ostest/*lab-name*/judge.sh的评测程序，将其输出实时反馈给推送代码的学生，并将其返回值作为本次评测的得分进行记录，在信息统计平台中进行展示。教学团队可为每个实验在 /opt/oslab/ostest/ 目录下创建一个实验目录，在其中放置评测程序judge.sh以及评测依赖的其他文件。

评测程序可参考 /opt/oslab/ostest/ 目录下系统预设的lab-qmos、lab-c和lab-rs几个测试实验，以Shell脚本的形式进行编写。实验目录下的judge.sh应具有可执行权限（chmod +x），它被调用时工作目录仍为实验目录，参数为本次评测中学生提交的代码所在目录，退出时的返回值则被记录为本次评测的得分。

# 静态扫描结果

对于本项目使用的语言Python、JavaScript和Shell，分别使用Pyre、ESLint和ShellCheck三种分析工具对代码进行了静态扫描，并修正了发现的所有问题：



# 第三方软件列表

本项目使用了以下开源软件的代码：

* Axios（[MIT License](https://github.com/axios/axios/blob/v0.x/LICENSE)）
* Bootstrap（[MIT License](https://github.com/twbs/bootstrap/blob/main/LICENSE)）
* code-server（[MIT License](https://github.com/coder/code-server/blob/main/LICENSE)）
* dateutil（[Apache License 2.0](https://github.com/dateutil/dateutil/blob/master/LICENSE)）
* dhtmlxGantt（[GPL 2.0](https://github.com/DHTMLX/gantt/blob/master/license.txt)）
* ECharts（[Apache License 2.0](https://github.com/apache/echarts/blob/master/LICENSE)）
* Element（[MIT License](https://github.com/ElemeFE/element/blob/dev/LICENSE)）
* Font Awesome（[MIT License](https://github.com/FortAwesome/Font-Awesome/blob/6.x/LICENSE.txt)）
* format-duration（[ISC License](https://github.com/ungoldman/format-duration/blob/main/LICENSE.md)）
* Flask（[3-Clause BSD License](https://flask.palletsprojects.com/en/2.2.x/license/)）
* Flask-Caching（[3-Clause BSD License](https://flask-caching.readthedocs.io/en/latest/license.html)）
* Flask-HTTPAuth（[MIT License](https://github.com/miguelgrinberg/Flask-HTTPAuth/blob/main/LICENSE)）
* Gunicorn（[MIT License](https://github.com/benoitc/gunicorn/blob/master/LICENSE)）
* python-gitlab（[LGPL 3.0](https://github.com/python-gitlab/python-gitlab/blob/main/COPYING)）
* vscode-wakatime（[3-Clause BSD License](https://github.com/wakatime/vscode-wakatime/blob/master/LICENSE)）
* Vue.js（[MIT License](https://github.com/vuejs/vue/blob/main/LICENSE)）
* Vue Router（[MIT License](https://github.com/vuejs/vue-router/blob/dev/LICENSE)）