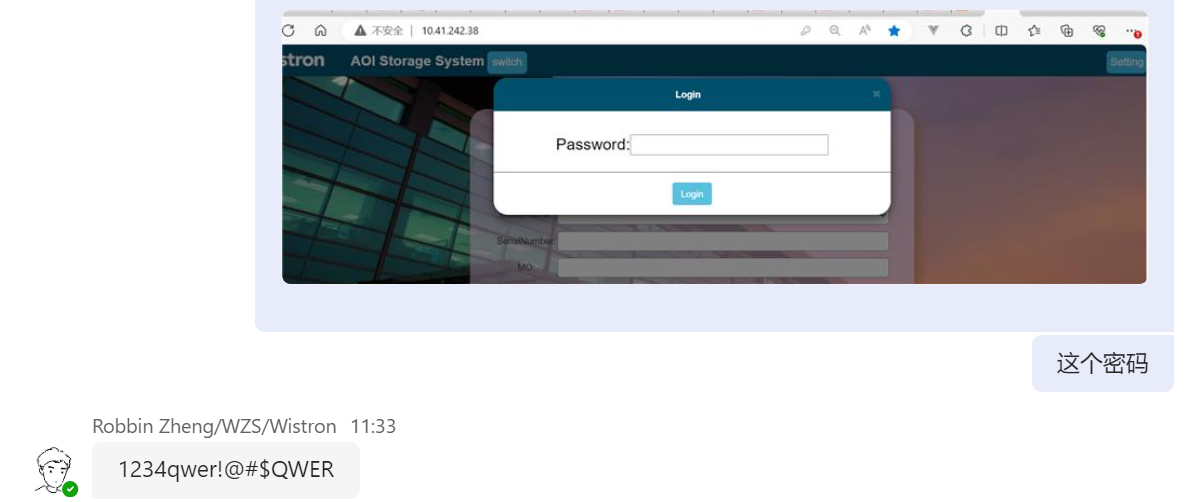


AIOSS

服务器：10.41.242.38 账号：root 密码：111111



// 前端ｖｉｔｅ．ｃｏｎｆｉｇ．ｊｓ文件配置ａｐｉ，可以直接用服务器后端 本地就可以不开后端服务了

      // '/api': {

      //   target: 'https://10.41.95.45:8510/api',

      //   changeOrigin: true,

      //   secure: false,

      //   rewrite: (path) => path.replace(/^\/api/, ''),

      // },

刚刚更新了我们**部门的docker私有镜像仓库harbo**r的域名为  <https://swharbor.wistron.com>

       后面需要使用到镜像仓库的话。可以使用docker login <https://swharbor.com>   账号:admin  密码1234qwer!@#$QWER 进行登录。

       也可以去到服务器 /etc/docker/下   修改或者新增文件daemon.json文件。新增一行  "insecure-registries":["<https://harbor.wistron.com>","https://swharbor.wistron.com "]  随后使用systemctl restart docker 生效。

       后续编写docker compose文件的话。需要使用到镜像仓库里面的镜像就需要修改为新的域名+项目名+镜像名称    已启动的docker compose如果是使用镜像仓库里面的镜像需要也请及时更新。

**这是WZS的Harbor相关信息：**

IP：<http://10.41.95.93/harbor/projects>

账号：admin

密码：1234qwer!@#$QWER

**安装依赖包出错解决方法:（也可能是网络问题）**

验证代理设置：如果你在使用代理服务器，请确保代理配置正确。检查当前的代理设置：

npm config get proxy

如果代理设置不正确或者需要更新，可以使用以下命令设置正确的代理：

npm config set proxy http://10.38.250.162:3128

npm config set https-proxy http://10.38.250.162:3128

清理npm缓存：有时，npm缓存的问题可能导致网络连接问题。清理npm缓存可能会解决问题。你可以使用以下命令清除缓存：

npm cache clean --force

使用其他npm源：尝试更换使用其他的npm源，如官方源或其他可靠的源。

npm config set registry https://registry.npmjs.org/

下载包requirements.txt：

python.exe -m pip install -r requirements.txt

接口文档

TSC：http://10.41.95.93:9877/web/#/item/index

门户系统：

测试服：http://10.41.95.88:18003

正式服：<https://wzsofe.wistron.com>

问题答疑助手GPT：

公司的GPT ：https://chat.wistron.com/app/chatRoom

<https://qithub.com/Yangjianxin1/GPT2-chitchat>

88服务器：

账号：root

密码：esrz30

启动服务器：docker start sns\_test\_db

查看redis任务进程：

查询进入redis 2库：select 2

查看所有任务：keys \*

查看单个任务执行情况：get 加查看任务进程的值

例子：

127.0.0.1:6379[2]> keys \*

1) "celery-task-meta-59116e06-e822-4a40-9245-3993d3d94cdd"

2) "celery-task-meta-61489130-7b32-4cd7-b350-776a1e60c60e"

127.0.0.1:6379[2]> get celery-task-meta-59116e06-e822-4a40-9245-3993d3d94cdd

"{\"status\": \"SUCCESS\", \"result\": {\"success\": true}, \"traceback\": null, \"children\": [], \"date\_done\": \"2023-08-14T00:50:23.023663\", \"task\_id\": \"59116e06-e822-4a40-9245-3993d3d94cdd\"}"

 Django基本操作：

创建项目：

Django-admin startproject 项目名

应用：py manage.py startapp 应用名

运行：py manage.py runserver

创建虚拟环境：

Py -m venv 名称

激活：名称/scripts/activate

创建项目后下载：

pip install Django

pip install djangorestFramework

pip install psycopg2

迁移：py manage.py makemigrations

同步：py manage.py migrate

部署操作：

1、生成requirements.txt文件：pip freeze > requirements.txt

2、配置各类后端文件：

（1）创建一个docker文件夹，存放新建的docker-compose.yml ; Dockerfile ; nginx.conf和生成的requirements.txt 文件

（2） 新建start.sh; uwsgi.ini文件，配置

（3）打包前端项目：npm run build (在后端项目新建一个文件夹，把前端打包生成好的dist文件放进去）

（4）打开FileZilla软件，连接要使用的服务器，要用到的文件上传到云端服务器。

3、打开PUTTY，登录服务器 : （输入服务器地址和登录的用户名密码）

进入docker目录：cd home// -> cd xue/ -> cd build// -> cd docker/

测试一下是否有网，要不然要离线下载包：ping [www.baidu.com](http://www.baidu.com)

**在vscode中下载离线包：**pip download -d [存放包文件的目录名] -r requirements.txt

（1）构建环境：docker compose build

（2）运行：docker compose up

运行成功后：在logs文件中查看问题

更新文件内容，直接在docker目录下：docker-compose restart运行

查看所有容器：docker ps

删除容器：docker compose down -v

1、build打包-

rm -rf \* 删除当前所有目录



**服务器Linux基本命令：**

**查容器名称**：docker ps

**删除文件**：rm -r 文件名

**删除某类文件**：rm -rf ./\*.whl

**复制文件**

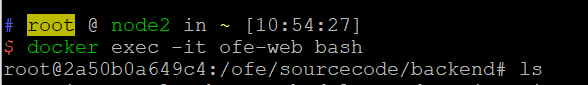
docker cp ofe-web:ofe/sourcecode/backend/media/whl ./whl

**容器中复制文件从某个目录（一般当前服务器下载的目录）复制到某个服务器目录**

scp -r /home/xue/whl [root@10.41.242.38:/home/aoiss/snimg](mailto:root@10.41.242.38:/home/aoiss/snimg)

**容器中下载离线包操作：**

**先进入容器**：

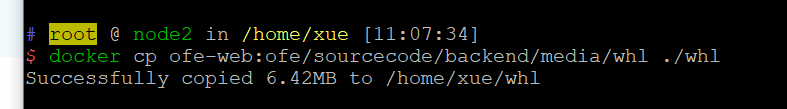


**创建一个文件夹装**：mkdir 包名

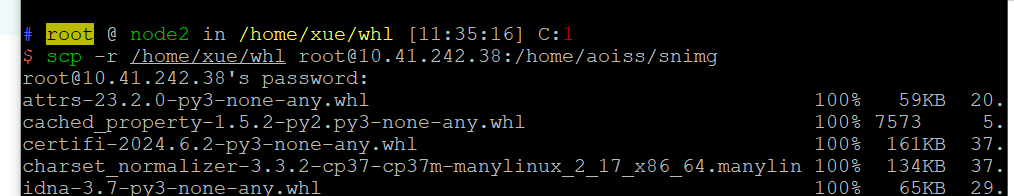
**下载**：pip3 download zeep

**退出容器**：exit

**先复制下载的文件**：



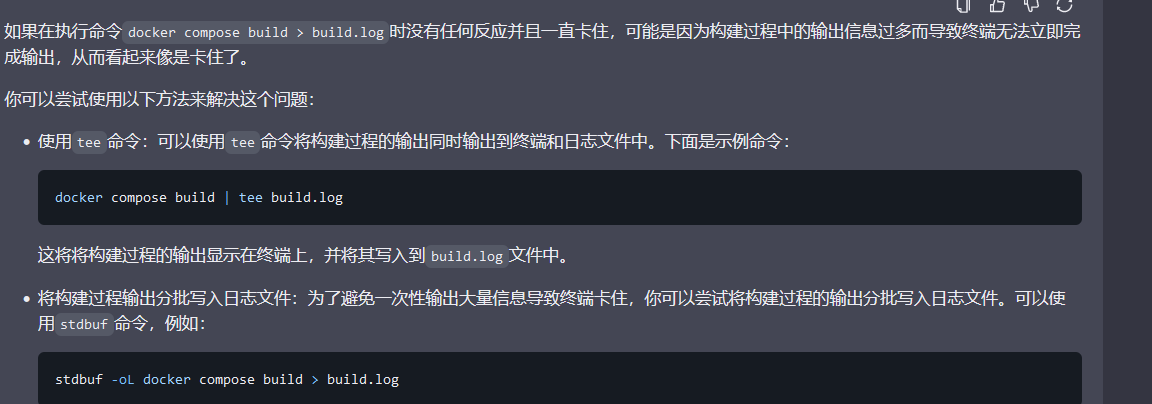
**进入目录，把容器中下载的文件复制到另一个要使用的容器中**：



**打开文件：**cat 文件名

docker compose build > build.log。然后，你可以使用cat命令或文本编辑器，如vi或nano，查看构建日志文件的内容，从中找到报错信息。

docker logs命令查看相关容器的日志。使用docker ps命令获取容器的ID或名称，然后使用docker logs <container\_id>或docker logs <container\_name>命令来查看容器的日志输出。



harbor地址：

http://10.41.95.93/harbor/projects/12/repositories

用户名：admin

密码：1234qwer!@#$QWER

加代理：

终端中执行：

$Env:http\_proxy="http://10.38.250.162:3128"

$Env:https\_proxy="10.38.250.162:3128"

PPT转编码：

把ppt传到88的ofe后端的media下面

PPT地址：https://wzsofe.wistron.com/api/media/ppt/666\_ODQ3R0r.pptx

把地址转成base64：

aHR0cHM6Ly93enNvZmUud2lzdHJvbi5jb20vYXBpL21lZGlhL3BwdC82NjZfT0RRM1Iwci5wcHR4

访问内容：

http://10.41.95.90:8500/pre/add\_task?url=

访问PPT：

http://10.41.95.90:8500/pre/view?url= aHR0cHM6Ly93enNvZmUud2lzdHJvbi5jb20vYXBpL21lZGlhL3BwdC82NjZfT0RRM1Iwci5wcHR4

转编码工具：https://www.toolhelper.cn/EncodeDecode/Base64EncodeDecode

Vue3：小兔鲜项目资料网站

资料：<https://xiaobaicai350.github.io/2023/06/18/%E5%B0%8F%E5%85%94%E9%B2%9C%E9%A1%B9%E7%9B%AE/>

git项目：https://github.com/Xiaobaicai350/vue-rabbit/blob/main/src/styles/common.scss

接口文件：https://apifox.com/apidoc/shared-c05cb8d7-e591-4d9c-aff8-11065a0ec1de/api-67132158

安装QtPy5：pip install PyQt5

需先cmd下载：wget https://aka.ms/vs/17/release/vs\_BuildTools.exe -o vs\_BuildTools.exe ; cmd /c vs\_BuildTools.exe

下载完成后，勾选下列内容点击安装

