1. 硬件要求：两台机器，后端在Linux，前端在Windows，CPU 2核以上， 硬盘100G 以上， 内存4G，如果要标注超大图像，则需要16G内存。
2. 软件要求：Windows操作系统及Linux Ubuntu操作系统

下面安装Windows及Linux程序，可以在链接：https://pan.baidu.com/s/19q3t97Vv7ZMqHRO05IE-jg

提取码：1234

上下载，或者代码中的jar目录下就有。

1. 后端在linux ubuntu上安装mysql数据库，参见linux环境搭建手册。
2. 安装linux minio，并启动。参见linux环境搭建手册。
3. 安装linux的jdk1.8，参见linux环境搭建手册。

在ubuntu的某个 目录下创建一个javaapp目录，将代码中Jar目录下的 labelSystem-0.0.1-SNAPSHOT.jar及

application-runtime.properties 拷贝到此目录。

修改application-runtime.properties中下面标红的参数为自己设置的参数：

spring.datasource.url=jdbc:mysql://192.168.62.129:3306/labelsystem?serverTimezone=CTT&useUnicode=true&characterEncoding=utf8&useSSL=false

#spring.datasource.url，数据库的IP地址修改对应的就行。

spring.datasource.username=root

#spring.datasource.username数据库的用户名

spring.datasource.password=pcl123456

#spring.datasource.password数据库的密码

server.port=8000

#后端访问的端口，如果8000被占用，可以修改。

minio.url= <http://192.168.62.129:9000/>

# minio.url 这个使用安装minio中的 browser access中的一个，能访问就行。

minio.username=minioadmin

# username 对应启动minio后，显示的ACCESS-KEY

minio.password=minioadmin

# password对应启动minio后，显示的SECRET-KEY

这个配置文件中这4行用来决定服务对外提供https访问，如果注释掉这4行，则是http访问。

server.ssl.key-store=classpath:keystore.p12

server.ssl.key-store-password=Pcl123456

server.ssl.keyStoreType=PKCS12

server.ssl.keyAlias=pcl

然后在javaapp目录下运行：

java -jar labelSystem-0.0.1-SNAPSHOT.jar

1. 安装windows的nginx反向代理，百度一下，或者参见：https://www.cnblogs.com/jiangwangxiang/p/8481661.html

在安装nginx目录中，修改conf目录下的nginx.conf

在此文件中的http节点下添加：

server {

listen 80;

server\_name ip1;

client\_max\_body\_size 10000m;

client\_body\_buffer\_size 8m;

location ^~ /minio/ {

proxy\_pass [https://ip2:8000/minio/;](https://ip:8000/minio/;)

proxy\_set\_header Host $host;

}

location ^~ /dcm/ {

proxy\_pass [https://ip2:8000/dcm/;](https://ip:8000/dcm/;)

proxy\_set\_header Host $host;

}

location ^~ /api/ {

proxy\_pass [https://ip2:8000/api/;](https://ip:8000/api/;)

proxy\_set\_header Host $host;

}

location / {

add\_header Access-Control-Allow-Origin \*;

add\_header Access-Control-Allow-Methods 'GET, POST, OPTIONS, PATCH';

add\_header Access-Control-Allow-Headers 'DNT,X-Mx-ReqToken,Keep-Alive,User-Agent,X-Requested-With,If-Modified-Since,Cache-Control,Content-Type,Authorization';

root d:/pcl/web/labeling/web-dl;

index index.html;

}

}

修改ip1为nginx对外访问的windows的IP地址。

修改其中的ip2为正确的的linux后端IP地址;8000为 server.port对应的值，

7086833b9ae0de63fc2e238864c4c9d

其中，在第5步正常启动之后，查看labelsystem.log，检查tomcat的启动方式是http还是https，并查看端口是否8000。 然后根据显示的信息修改proxy\_pass中的ip2及8000。

7、在windows下安装前端页面

（1）将web目录下文件拷贝到宿主机的某个目录下（如：d:/pcl/），然后在nginx.conf配置文件中增加服务器配置，可以参见nginx搭建一节。

（2）查看宿主机的IP地址，修改 web\labeling\web-dl\js\func.js 文件中getIp方法返回的IP地址，与nginx中配置的ip1地址一致。

（3）重新启动nginx服务。

（4）使用浏览器，用 http://ip1 访问，并进行测试。默认用户名：：

LabelSystem01 / pcl123456

8、如果要操作视频，则需要在linux后端上安装ffmpeg，可以自行搜索安装。

1. 如果要标注超大图像，需要安装python 3.6.5，并需要将jar目录下的image2dzi拷贝到宿主机中，在该目录下运行

python3 image2dzi.py （或者： python image2dzi.py）

，需要先安装对应的python库，如下：

from flask import Flask, abort, make\_response, render\_template, url\_for, request, Response

from io import BytesIO

import os

import pyvips

import json

import requests

from logging.handlers import TimedRotatingFileHandler

import logging

import time

from requests.packages.urllib3.exceptions import InsecureRequestWarning

requests.packages.urllib3.disable\_warnings(InsecureRequestWarning)

如果运行成功后，可以结束掉此进程，并转用后台命令执行：

nohup python3 image2dzi.py & 或者（nohup python image2dzi.py &）