1. 硬件要求：CPU 2核以上， 硬盘100G 以上， 内存4G以上，如果要标注超大图像，则需要16G内存。
2. 宿主机安装Linux Ubuntu 16版本操作系统

查看ubuntu操作系统版本：

lsb\_release -a

uname -a

1. 在宿主机上安装mysql数据库

sudo apt-get install mysql-server

sudo apt install mysql-client

sudo apt install libmysqlclient-dev

找到默认用户及密码：

sudo cat /etc/mysql/debian.cnf

登录：

mysql -u debian-sys-maint -p

更改root密码及访问策略：

use mysql;

update user set authentication\_string=password("pcl123456") where user="root";

grant all privileges on \*.\* to 'root'@'%' identified by 'pcl123456' with grant option;

update user set plugin="mysql\_native\_password";

flush privileges;

设置mysql允许远程访问：

sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf

注释掉 bind-address = 127.0.0.1

然后保存。

重新启动mysql

sudo service mysql restart

连接上mysql数据库，创建labelsystem库（CREATE SCHEMA `labelsystem` DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci ;如果执行出现：

Unknown collation: 'utf8mb4\_0900\_ai\_ci'，

则修改成：CREATE SCHEMA `labelsystem` DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci），一定要使用utf8mb4字符集，并执行db目录下的labelsystem\_2.sql脚本。

1. 从jar目录中，拷贝linux-minio目录，运行minio（先赋权限 chmod 755 minio），其中/data/minio/data/表示minio数据存储地址，执行以下命令：

export MINIO\_ACCESS\_KEY=minio

export MINIO\_SECRET\_KEY=miniominio

nohup ./minio server /data/minio/data/ &

1. 安装jdk，并运行标注系统的后端程序。

将jar目录下的linux-jdk 拷贝到宿主机指定目录，比如：

/home/linux-jdk

设置环境变量：

vi /etc/profile

在内容的末尾增加：

export JAVA\_HOME=/home/linux-jdk/jdk1.8.0\_161

export JRE\_HOME=${JAVA\_HOME}/jre

export CLASSPATH=.:$JAVA\_HOME/jre/lib/rt.jar:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

export PATH=$PATH:$JAVA\_HOME/bin

在宿主机中，创建一个javaapp目录，将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.username=root

spring.datasource.password=pcl123456

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

这个配置文件中下面这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：

nohup java -jar labelSystem-0.0.1-SNAPSHOT.jar &

1. 在宿主机上安装nginx反向代理

sudo apt-get install nginx

配置nginx:

cd /etc/nginx/

sudo vim nginx.conf

在此文件中的http节点下添加：

server {

listen 80;

server\_name ip;

client\_max\_body\_size 10000m;

client\_body\_buffer\_size 8m;

location ^~ /minio/ {

proxy\_pass <https://ip:8000/minio/;>

proxy\_set\_header Host $host;

}

location ^~ /dcm/ {

proxy\_pass <https://ip:8000/dcm/;>

proxy\_set\_header Host $host;

}

location ^~ /api/ {

proxy\_pass <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 /home/label/javaapp/web/labeling/web-dl;

index index.html;

}

}

（1）将server\_name中的ip地址修改成正确的地址，此地址为前端访问的IP地址，第7步将要使用。

（2）修改proxy\_pass中的ip及端口，此IP及端口为标注系统后端运行的IP及端口，如下图为标注系统运行成功之后，查看labelsystem.log，显示的端口及http的访问方式。

7086833b9ae0de63fc2e238864c4c9d

proxy\_pass的代表中前端web发送过来的请求，将会被nginx转发到标注系统后端进行处理。

（3）root 节点中为web前端的页面，地址填写第7步中web页面文件目录路径，路径到web-dl为止。

7、在宿主机上安装前端页面

（1）将web目录下文件拷贝到宿主机的某个目录下（如：/home/pcl/web），然后在nginx.conf配置文件root节点中将此目录配置上。

1. 修改 web\labeling\web-dl\js\func.js 文件中getIp方法返回的IP地址，与nginx.conf中配置的server\_name及端口一致，即等于：server\_name:listen。(nginx对外提供的IP及端口)
2. 重新启动nginx服务。ubuntu的命令为：sudo service nignx restart

（4）使用浏览器，用 http://宿主机的IP地址 访问，并进行测试。默认用户名：

LabelSystem01 / pcl123456

8、如果要操作视频，则需要安装ffmpeg，可以自行搜索安装。

1. 如果要标注超大图像（内存最好16G以上），需要安装python 3.6，并需要将jar目录下的image2dzi拷贝到宿主机中，比如安装目录为/home/image2dzi/，下面将会用到这个安装路径，安装对应的python库，如下：
2. 安装python3.6

add-apt-repository ppa:deadsnakes/ppa

apt-get update

apt-get install python3.6

--切换3.6为默认python3

update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.5 1

update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.6 1

update-alternatives --config python3

--安装pip3

apt-get install python3-pip

1. 安装image2dzi需要的依赖库

pip3 install Flask

--安装 医学图像处理库pyvips

apt-get install libvips

apt-get install libvips-dev

apt-get install libxml2-dev libxslt1-dev python-dev

apt-get install zlib1g-dev

apt-get install libevent-dev

apt-get install build-essential libssl-dev libffi-dev

apt-get install python3.6-dev

pip3 install lxml

pip3 install --user pyvips

pip3 install flask

pip3 install requests

用后台命令执行（执行之后，会占用8020端口，如果端口冲突，请修改同目录下config.ini中端口配置）：

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

执行完成后，使用以下命令检查进程是否正常启动：

ps -ef | grep image2dzi

如果出现如下所示结果，表示运行成功。

root 29615 1 2 08:05 pts/0 00:00:00 python3 image2dzi.py

root 29619 1 0 08:06 pts/0 00:00:00 grep --color=auto image2dzi

另外还需要修改 application-runtime.properties ，设置如下两个属性，一个是开关，打开大图标注功能，另外一个是配置image2dzi的安装路径。

server.enable.bigimg=true

server.dzi.install.path=/home/image2dzi/