**Docker部署文档（虚拟机）**

组长：XXX XXXXXXXX

组员：XXX XXXXXXXX

XXX XXXXXXXX

XXX XXXXXXXX

病虫害识别系统

2021

目录

[1 环境配置 2](#_Toc91173397)

[1.1 docker环境安装 2](#_Toc91173398)

[1.2 配置专有网络 2](#_Toc91173399)

[2 微服务部署 2](#_Toc91173400)

[2.1 Nacos安装 2](#_Toc91173401)

[2.2 病虫识别模型部署 3](#_Toc91173402)

[2.3 使用docker-compose部署微服务 5](#_Toc91173403)

[3 前端部署 9](#_Toc91173404)

# 环境配置

## docker环境安装

操作系统为Ununtu20.04，docker安装采用脚本安装的方式。执行shell指令：

|  |
| --- |
| curl -fsSL https://get.docker.com -o get-docker.sh  sudo sh get-docker.sh |

docker安装完成后，配置镜像加速：

|  |
| --- |
| sudo mkdir -p /etc/docker  sudo tee /etc/docker/daemon.json <<-'EOF'  {    "registry-mirrors": ["https://95rosw2w.mirror.aliyuncs.com"]  }  EOF  sudo systemctl daemon-reload  sudo systemctl restart docker |

安装docker-compose，执行指令

|  |
| --- |
| pip install docker-compose |

至此服务器基础环境配置成功。

## 配置专有网络

由于docker默认使用bridge网络，会随机给容器分配IP地址，但是nacos的ip地址写在java微服务的配置文件中，不能改动，所以需要指定nacos的IP地址，需要创建一个新的mynet网络，输入指令。

|  |
| --- |
| docker network create --driver bridge --subnet=172.19.12.0/16 --gateway=172.19.1.1 mynet |

# 微服务部署

## Nacos安装

1. 拉取nacos镜像

|  |
| --- |
| docker pull nacos/nacos-server |

1. 启动nacos容器

|  |
| --- |
| docker run --env MODE=standalone --name nacos -d --network=mynet --ip=172.19.12.1 -p 8848:8848 nacos/nacos-server |

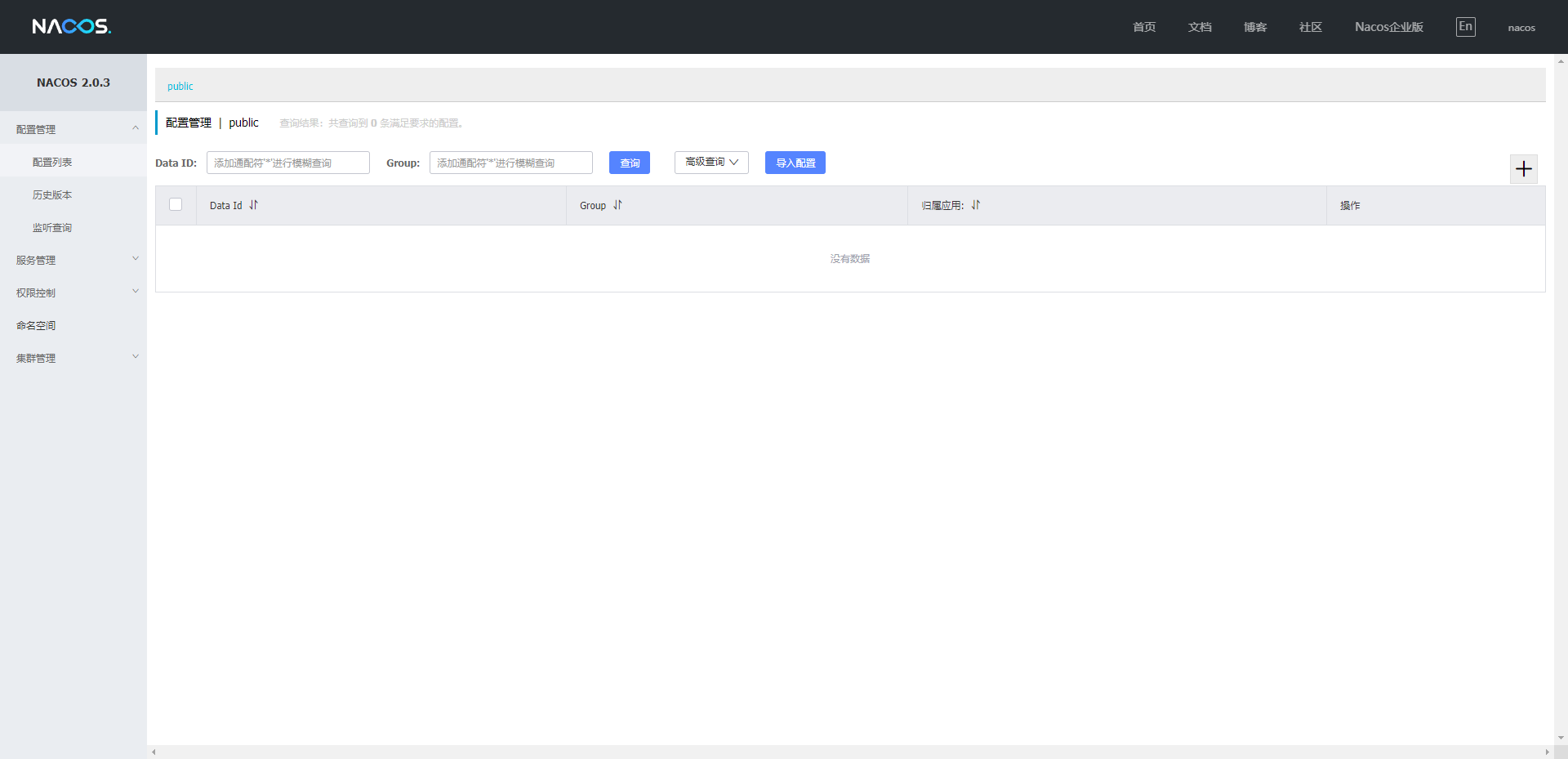
1. nacos的默认端口是8848，默认账号密码是nacos/nacos
2. 查看nacos容器日志，执行（e15761426db5为nacos容器id）

|  |
| --- |
| docker logs -f e15761426db5 |

1. 获得nacos图形化界面地址

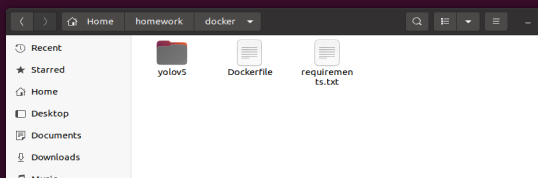


1. 至此nacos已配置成功，访问http:// 172.19.12.1:8848/nacos/index.html，如图

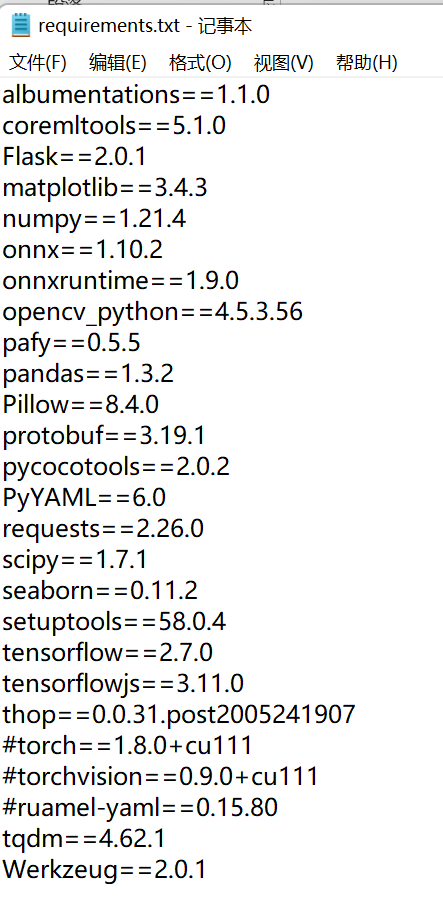


## 病虫识别模型部署

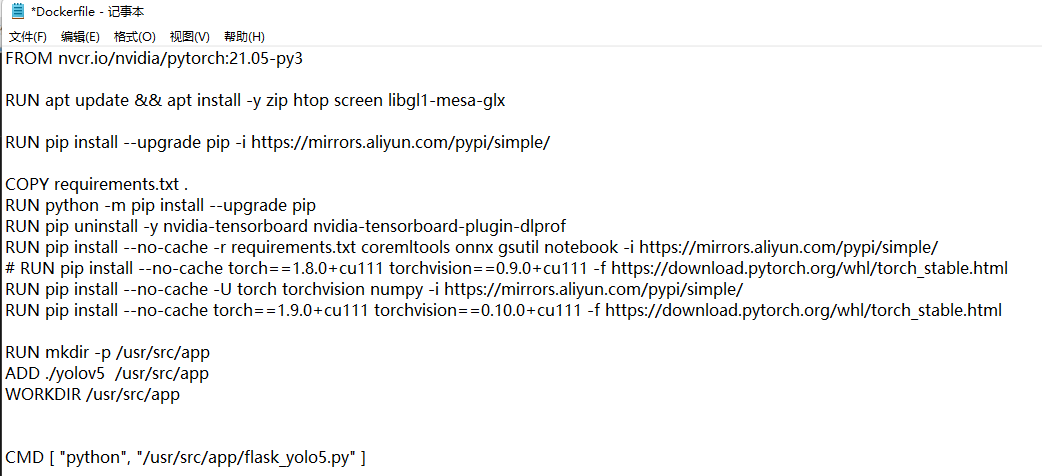
1. 将yolov5模型代码、Dockerfile以及requirenmrnts.txt文件导入服务器/home/dilidili131/homework/docker目录下，如图



1. Requirement.txt文件内记录模型需要的环境，如图：



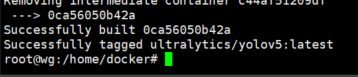
1. Dockerfile文件如图：



该dockerfile，首先拉取pytorch镜像作为依赖环境，然后升级apt及pip，并安装requirements.txt记载的python包，以及其他需要依赖。

1. 切换到/home/dilidili131/homework/docker目录，运行dockerfile，执行：

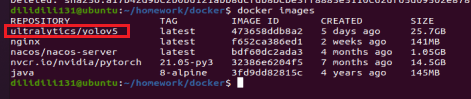
|  |
| --- |
| t=ultralytics/yolov5:latest && sudo docker build -t $t . |



1. 查看docker容器中的镜像

|  |
| --- |
| docker images |

如图：

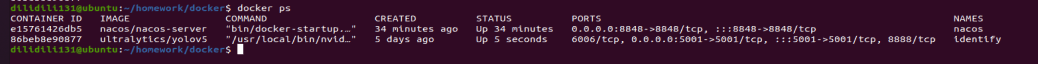


模型镜像搭建成功。

1. 启动容器，并暴露5001端口，执行

|  |
| --- |
| sudo docker run --name identify -p 5001:5001 -d ultralytics/yolov5 |

1. 查看已运行容器，执行docker ps，如图

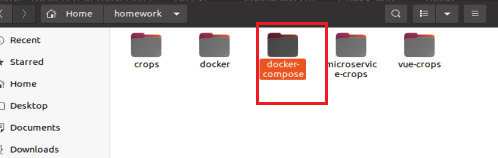


可见容器已运行成功。

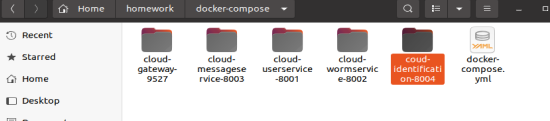
1. 至此，模型已部署完成

## 使用docker-compose部署微服务

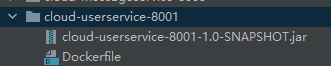
1. 将docker-compose文件夹复制到/home/dilidili131/homework目录下，如图



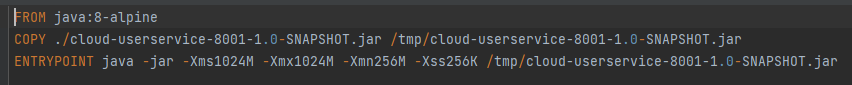
目录结构如下图：



1. 用户服务目录结构



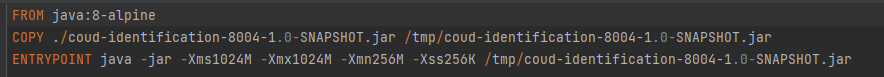
dockerfile如下



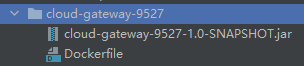
1. 识别服务目录结构如下



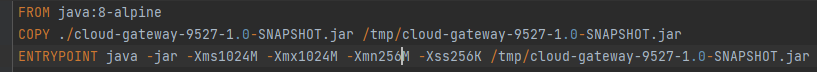
Dockerfile如下：



1. 网关服务目录结构如下



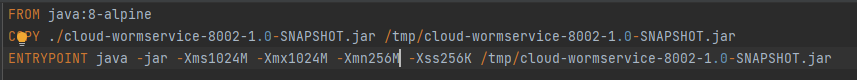
Dockerfile如下：



1. 病虫信息管理服务目录结构



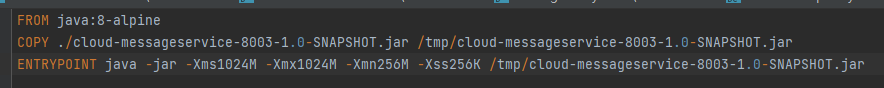
dockerfile如下



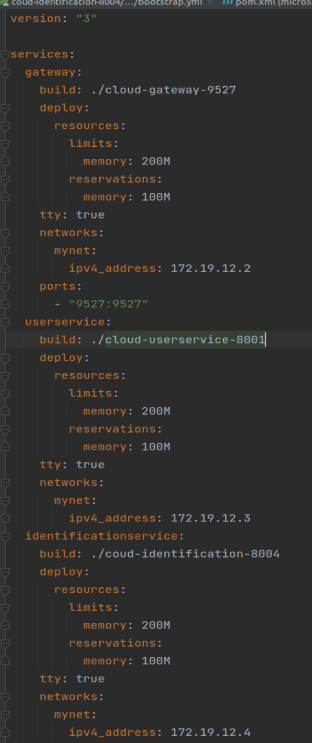
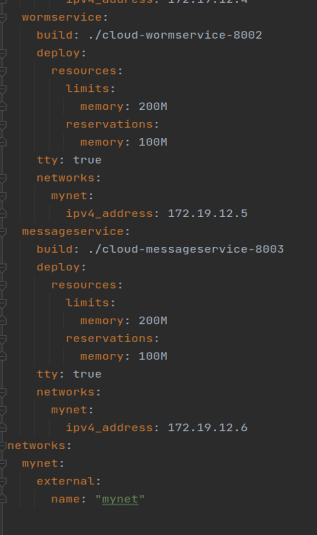
1. 报警信息管理服务服务目录结构如下



Dockerfile如下：



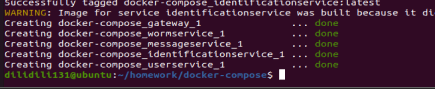
1. docker-compose.yml如下

1. 切换到/home/dilidili131/homework/docker-compose目录下，开始打包进行，执行指令

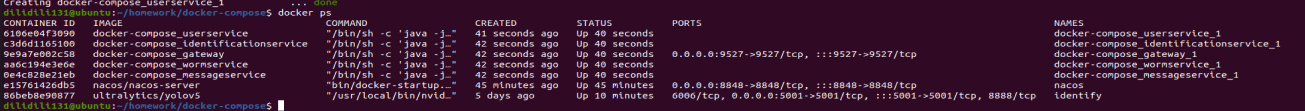
|  |
| --- |
| docker-compose up -d |

如图，说明镜像构建成功



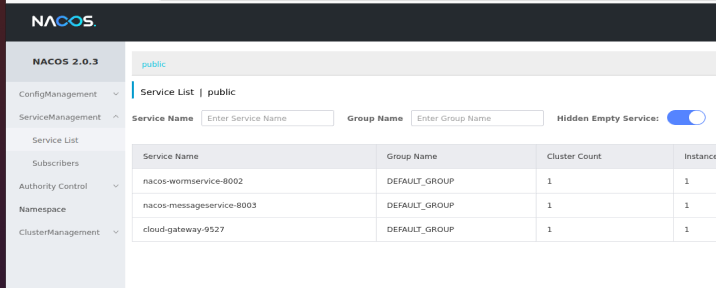
1. 查看容器是否启动成功执行

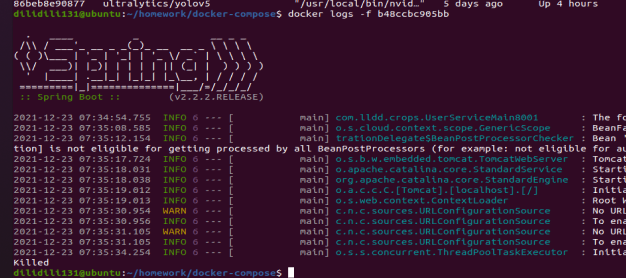
|  |
| --- |
| docker ps |



发现容器成功启动

1. 此时发现部分微服务并没有注册进nacos，通过日志查看原因，发现容器被kill，分析原因可能是设备配置导致同时启动多个容器部分失败。





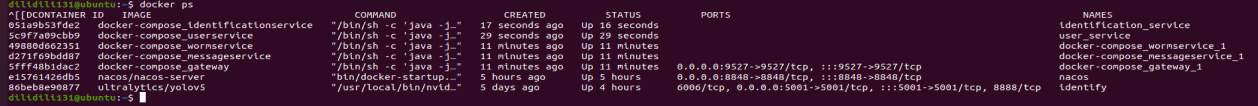
1. 删除启动失败微服务相关容器

执行docker rm -f 容器id

1. 手动启动之前启动失败的容器，执行指令

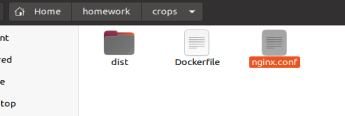
|  |  |
| --- | --- |
| Gateway | docker run --name gateway --network=mynet --ip=172.19.12.2 -p 9527:9527 -d docker-compose\_gateway |
| Identification | docker run --name identification\_service --network=mynet --ip=172.19.12.4 -d docker-compose\_identificationservice |
| Userservice | docker run --name user\_service --network=mynet --ip=172.19.12.3 -d docker-compose\_userservice |
| Wormservice | docker run --name worm\_service --network=mynet --ip=172.19.12.5 -d docker-compose\_wormservice |
| Messagesrvice | docker run --name message\_service --network=mynet --ip=172.19.12.6 -p 8003:8003 -d -v /home/dilidili131/Document:/home/dilidili131/Document |

1. 再次查看容器，发现均已启动成功



# 前端部署

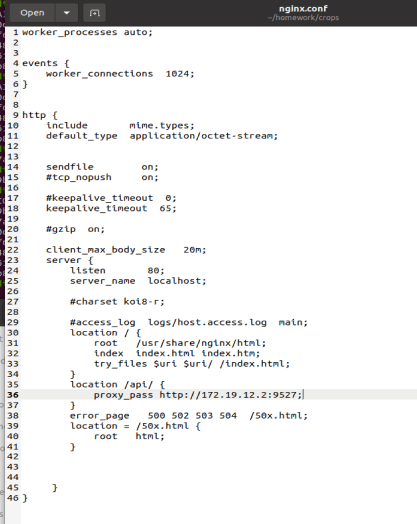
1. 将前端打包文件dist、dockerfile文件、nginx.conf配置文件上传至服务器/home/dilidili131/homework/crops目录下，如图



1. Dockerfile如下



1. nginx.conf如下



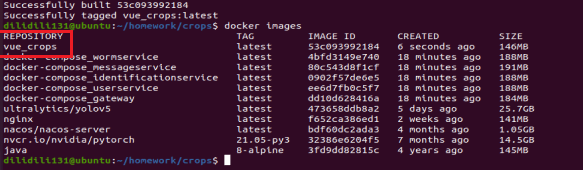
1. 构建镜像，在文件目录下执行指令

|  |
| --- |
| docker build -t vue\_crops . |

1. 查询镜像，输入指令

|  |
| --- |
| docker images |

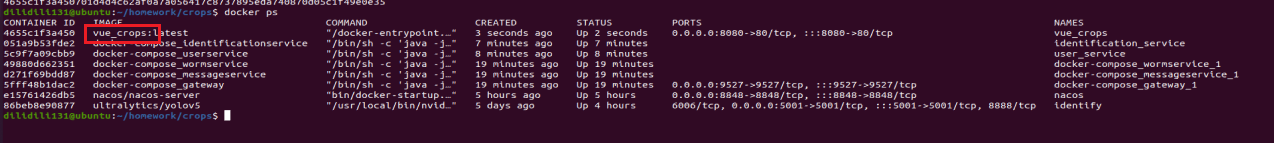
如图：镜像搭建成功



1. 启动容器，执行指令：

|  |
| --- |
| docker run -di --network=mynet --ip=172.19.12.7 -p 8080:80 --name vue\_crops vue\_crops:latest |

1. 输入docker ps查看容器，如图



至此全部部署完成