

# Ubuntu 22以上版本全私有化部署流程

## 第一步：准备部署资料

1.1从配置好的linux服务器拷贝配置文件

/docker目录

1.2从配置好的linux服务器拷贝docker镜像

```
docker save onlyoffice/documentserver:latest > /backup/onlyoffice.tar  
docker save 13671354640/zxcm:0.9.1 > /backup/zxcm.tar  
docker save 13671354640/zxcm-xxl-job-admin:0.9.1 > /backup/xxljob.tar  
docker save redis:6.2.6 > /backup/redis.tar  
docker save mysql:8.0.27 > /backup/mysql.tar  
docker save nginx:1.21.3 > /backup/nginx.tar
```

将save的tar文件全部拷贝出来做部署准备

1.3准备好docker和docker-compose安装包

1.4准备好部署用的html文件和jar文件

文件已经准备好在html和jar目录下

## 第二步：Docker离线部署

2.1下载docker镜像

Docker deb 包下载地址：

<https://download.docker.com/linux/ubuntu/dists/bionic/pool/stable/amd64/>

2.2下载三个文件：

containerd.io\_1.6.9-1\_amd64.deb

docker-ce\_24.0.2-1\_ubuntu.18.04\_bionic\_amd64.deb

docker-ce-cli\_24.0.2-1\_ubuntu.18.04\_bionic\_amd64.deb

2.3 sudo dpkg -i /文件目录/\*.deb 进行安装

2.4启动服务

重启守护进程

systemctl daemon-reload

systemctl start docker

查看docker状态

systemctl status docker

设置开机启动

```
systemctl enable docker
```

2.5查看docker启动状态是否正确

```
docker -v
```

## 第三步：安装docker-compose

3.1 上传docker-compose-linux-x86\_64

```
3.2 sudo cp /文件目录/docker-compose-linux-x86_64 /usr/local/bin/docker-compose
```

```
3.3 sudo chmod +x /usr/local/bin/docker-compose
```

3.4 测试是否安装成功 docker-compose version

## 第四步：拷贝部署文件

上传整个docker文件夹到服务器 /docker目录下

```
sudo cp -r /文件目录/docker /docker
```

## 第五步：安装镜像

```
sudo docker load -i /文件目录/onlyoffice.tar
```

```
sudo docker load -i /文件目录/zxcm.tar
```

```
sudo docker load -i /文件目录/xxljob.tar
```

```
sudo docker load -i /文件目录/redis.tar
```

```
sudo docker load -i /文件目录/mysql.tar
```

```
sudo docker load -i /文件目录/nginx.tar
```

```
sudo docker images 查看是否镜像是否都成功恢复
```

## 第六步：修改文件权限和配置文件

docker子目录都给到写文件的权限

```
chmod -R 777 /docker
```

```
chmod -R 777 /docker/server1/runner
```

修改/docker/docker-compose文件，修改mysql默认密码，请使用复杂度足够的密码export

```
TERM=linux #设置linux环境
```

如果不擅长linux，可以先改好

```

version: '3'

services:
  mysql:
    image: mysql:8.0.27
    container_name: mysql
    environment:
      # 时区上海
      TZ: Asia/Shanghai
      # root 密码
      MYSQL_ROOT_PASSWORD: root
      # 初始化数据库(后续的初始化操作会在这个库执行)
      MYSQL_DATABASE: zxcm
    ports:
      - "13306:3306"
    volumes:
      # 数据挂载
      - /docker/mysql/data/:/var/lib/mysql/
      # 配置挂载
      - /docker/mysql/conf/:/etc/mysql/conf.d/
    command:
      # 将mysql8.0默认密码策略 修改为 原先 策略 (mysql8.0对其默认策略做了更改 会导致密码无法匹配)
      --default-authentication-plugin=mysql_native_password
      --character-set-server=utf8mb4
      --collation-server=utf8mb4_general_ci
      --explicit_defaults_for_timestamp=true
      --lower_case_table_names=1
    privileged: true
    restart: always
    networks:
      ruoyi_net:
        ipv4_address: 172.30.0.36

  nginx-web:
    image: nginx:1.21.3

```

## 修改java程序的数据库配置 /docker/server1/application-prod.yml中

```

--- # 数据源配置
spring:
  datasource:
    type: com.alibaba.druid.pool.DruidDataSource
    # 动态数据源文档 https://www.kancloud.cn/tracy5546/dynamic-datasource/content
    dynamic:
      # 性能分析插件 (有性能损耗 不建议生产环境使用)
      p6spy: false
      # 设置默认的数据源或者数据源组,默认值即为 master
      primary: master
      datasource:
        # 王库数据源
        master:
          # jdbc 所有参数配置参考 https://lionli.blog.csdn.net/article/details/122018562
          # rewriteBatchedStatements=true 批处理优化 大幅提升批量插入更新删除性能 (对数据库有性能损耗 使用批量操作应考虑性能问题)
          driverClassName: com.mysql.cj.jdbc.Driver
          url: jdbc:mysql://172.30.0.36:3306/zxcm?useUnicode=true&characterEncoding=utf8&zeroDateTimeBehavior=convertToNull
          username: zxcm
          password: 123456 # 修改为刚才新设置的密码
        # 从库数据源
        slave:
          lazy: true
          driverClassName: oracle.jdbc.driver.OracleDriver
          url: jdbc:oracle:thin:@121.5.233.137:1521:helowin
          username: test
          password: test
    druid:
      # 初始连接数
      initialSize: 5
      # 最小连接池数量
      minIdle: 10

```

## 修改xxljob数据库配置

```

--- # 客户端配置
spring:
  boot:
    admin:
      # Spring Boot Admin Client 客户端的相关配置
      client:
        # 增加客户端开关
        enabled: true
        # 设置 Spring Boot Admin Server 地址
        url: http://172.30.0.90:9090/admin
      instances:
        service-host-type: IP
        username: ruoyi
        password: 123456

--- # 数据库配置
spring:
  datasource:
    type: com.zaxxer.hikaricp.HikariDataSource
    driver-class-name: com.mysql.cj.jdbc.Driver
    url: jdbc:mysql://172.30.0.90:3306/test?characterEncoding=UTF-8&autoReconnect=true&serverTimezone=Asia/Shanghai
    username: root
    password: lifanhuibin
    hikari:
      auto-commit: true
      connection-test-query: SELECT 1
      connection-timeout: 10000
      idle-timeout: 300000
      max-lifetime: 900000
      maximum-pool-size: 30
      minimum-idle: 10

```

修改onlyoffice和java程序连接的设置/docker/server1/application-office.yml

查看服务器 ip

ip addr show

```

#wps或者onlyoffice
onlineEditor: onlyoffice
wps:
  appid: 7c1815c1893740ab9cfac9628595e8c3
  secret: b6ea538b9d5d4819bd21c6cab0c67515
  wpsdomain: https://wwo.wps.cn
  downloadurl: http://proxy.zhaoxinms.com:9100
onlyoffice:
  domain: http://121.5.233.137
  port: 8082
  callbackUrl: http://143.143.173.96/api-prod/onlyoffice/callback
  #生成jwt密钥,需要修改onlyoffice的配置文件
  secret: zxcm
  plugin: http://121.5.233.137:8082/plugin/html/config.json
compare:
  create: https://doc-compare.intsig.com/doc_compare/create
  app-key: 7aa4d00ac591630006eff1189c73dd10
  app-secret:
    authorize: https://contracts.textin.com/contracts_backend_api/oidc/authorization_code
#阿里云发票识别
invoice:
  access-key-id: LTAI5tCP8ey86Dr3bhSkFwu3
  access-key-secret:
    url: ocr-api.cn-hangzhou.aliyuncs.com
#天眼查
tianyancha:
  url: http://open.api.tianyancha.com/services/open/
  token:

```

java程序访问的ip，这个是回调

onlyoffice的访问的ip,如果是同  
一台服务器和上一个ip一致

## 第五步：启动服务

基于docker-compose安装并启动服务

#启动mysql、nginx、redis等基础服务

cd /docker

sudo docker-compose up -d mysql nginx-web redis

#将sql文件放入容器并导入

sudo docker cp /路径/1.3.sql mysql:/opt/1.3.sql #文件传到容器

sudo vi /docker/mysql/conf/my.cnf #创建一个cnf, 内容如下：

```
[mysqld]
skip-grant-tables
sudo docker-compose stop mysql #重启mysql
sudo docker-compose up -d mysql
sudo docker exec -it mysql bash #进入容器
mysql -u root
FLUSH PRIVILEGES;
ALTER USER 'root'@'localhost' IDENTIFIED BY '新密码';
exit; #退出mysql
exit; #退出容器
sudo vi /docker/mysql/conf/my.cnf #将内容全部删除掉保存
sudo docker-compose stop mysql
sudo docker-compose up -d mysql
sudo docker exec -it mysql bash #进入容器
mysql -u root -p 输入刚才设置的密码
use jdy-dzht;
source /opt/1.3.sql; #导入数据库
exit;
exit;
#启动mysql服务
sudo docker-compose stop mysql
sudo docker-compose up -d mysql
#启动onlyoffice服务
sudo docker-compose up -d docServer
#查看redis启动状态，如果
#启动web服务、计划任务
sudo docker-compose up -d zxcm-server1 zxcm-xxl-job-admin
#访问 http://服务器ip:80/ 进入项目界面,
账号：zxcmAdmin 密码：默认密码
#访问 http://服务器ip:80/xxl-job-admin
账号：admin 密码：123456
#访问文档服务http://服务器ip:8082
```