**Ruoyi-AI平台（后端服务、管理端界面前端、用户端界面前端）生产环境Linux Server容器化部署手册**

1. 准备AlmaLinux / RockyLinux 9.x 操作系统基础环境
2. 禁用SELinux及防火墙

**setenforce 0**

**firewall-cmd --set-default-zone=trusted**

**sed --follow-symlinks -i "s/SELINUX=enforcing/SELINUX=disabled/g" /etc/selinux/config**

1. 安装docker服务软件及docker-compose命令

**dnf config-manager --add-repo=https://download.docker.com/linux/centos/docker-ce.repo**

国内网络可以：

**dnf config-manager --add-repo=http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo**

然后：

**dnf install docker-ce docker-ce-cli**

**systemctl enable --now docker**

最后安装docker-compose命令：

**curl -L -o /usr/local/bin/docker-compose** [**https://github.com/docker/compose/releases/download/v2.36.0/docker-compose-linux-x86\_64**](https://github.com/docker/compose/releases/download/v2.36.0/docker-compose-linux-x86_64)

**chmod +x /usr/local/bin/docker-compose**

1. 安装git命令

**dnf install git**

1. 下载源码
2. 创建工作目录

**mkdir /root/ruoyi-ai-docker**

**mkdir /root/ruoyi-ai-docker/source-code**

**mkdir /root/ruoyi-ai-docker/build-docker-images**

**mkdir /root/ruoyi-ai-docker/deploy**

**cd /root/ruoyi-ai-docker/source-code**

1. 下载源码

**git clone** [**https://github.com/ageerle/ruoyi-ai**](https://github.com/ageerle/ruoyi-ai)

**git clone** [**https://github.com/ageerle/ruoyi-admin**](https://github.com/ageerle/ruoyi-admin)

**git clone** [**https://github.com/ageerle/ruoyi-web**](https://github.com/ageerle/ruoyi-web)

1. 修改源码目录名

**mv ruoyi-ai ruoyi-ai-backend**

**mv ruoyi-admin ruoyi-ai-admin**

**mv ruoyi-web ruoyi-ai-web**

1. 切换后端服务源码分支

**cd ruoyi-ai-backend**

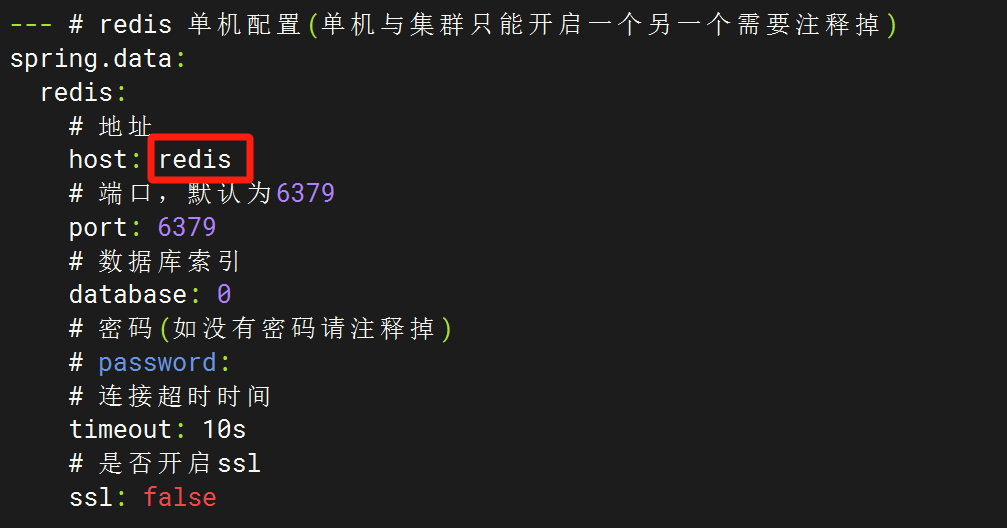
**git checkout v2.0.4**

**cd ..**

1. 应用构建编译打包
2. Java后端服务ruoyi-ai.jar包文件构建
3. 修改 /root/ruoyi-ai-docker/source-code/ruoyi-ai-backend/ruoyi-admin/src/main/resources/application-prod.yml 内容：



需要修改MySQL数据库服务器地址（这里改为名称mysql）、数据库名（这里改为ruoyi-ai），用户名root密码root，可根据实际需求改为复杂值。



需要修改Redis服务器地址（这里改为名称redis）

1. 用maven容器构建应用jar包：

**docker run --rm --name build-ruoyi-ai -v /root/ruoyi-ai-docker/source-code/ruoyi-ai-backend:/app -w /app maven:3.9.9-eclipse-temurin-17-alpine bash -c "mvn clean package -Pprod"**

1. 如果网络正常，代码没有错误，耐心等待编译构建过程直至结束，得到生成的jar包文件：

**/root/ruoyi-ai-docker/source-code/ruoyi-ai-backend/ruoyi-admin/target/ruoyi-admin.jar**

1. 管理端界面前端程序JS目录编译打包（.env.\*文件内容可选择性修改，也可保持不变，按实际安全要求决定）

**rm -f /root/ruoyi-ai-docker/source-code/ruoyi-ai-admin/apps/web-antd/.env.analyze**

**rm -f /root/ruoyi-ai-docker/source-code/ruoyi-ai-admin/apps/web-antd/.env.development**

**rm -f /root/ruoyi-ai-docker/source-code/ruoyi-ai-admin/apps/web-antd/.env.production**

**rm -f /root/ruoyi-ai-docker/source-code/ruoyi-ai-admin/apps/web-antd/.env.test**

**rm -f /root/ruoyi-ai-docker/source-code/ruoyi-ai-admin/apps/web-antd/vite.config.mts**

按需对上述文件进行修改后覆盖到源码目录位置：

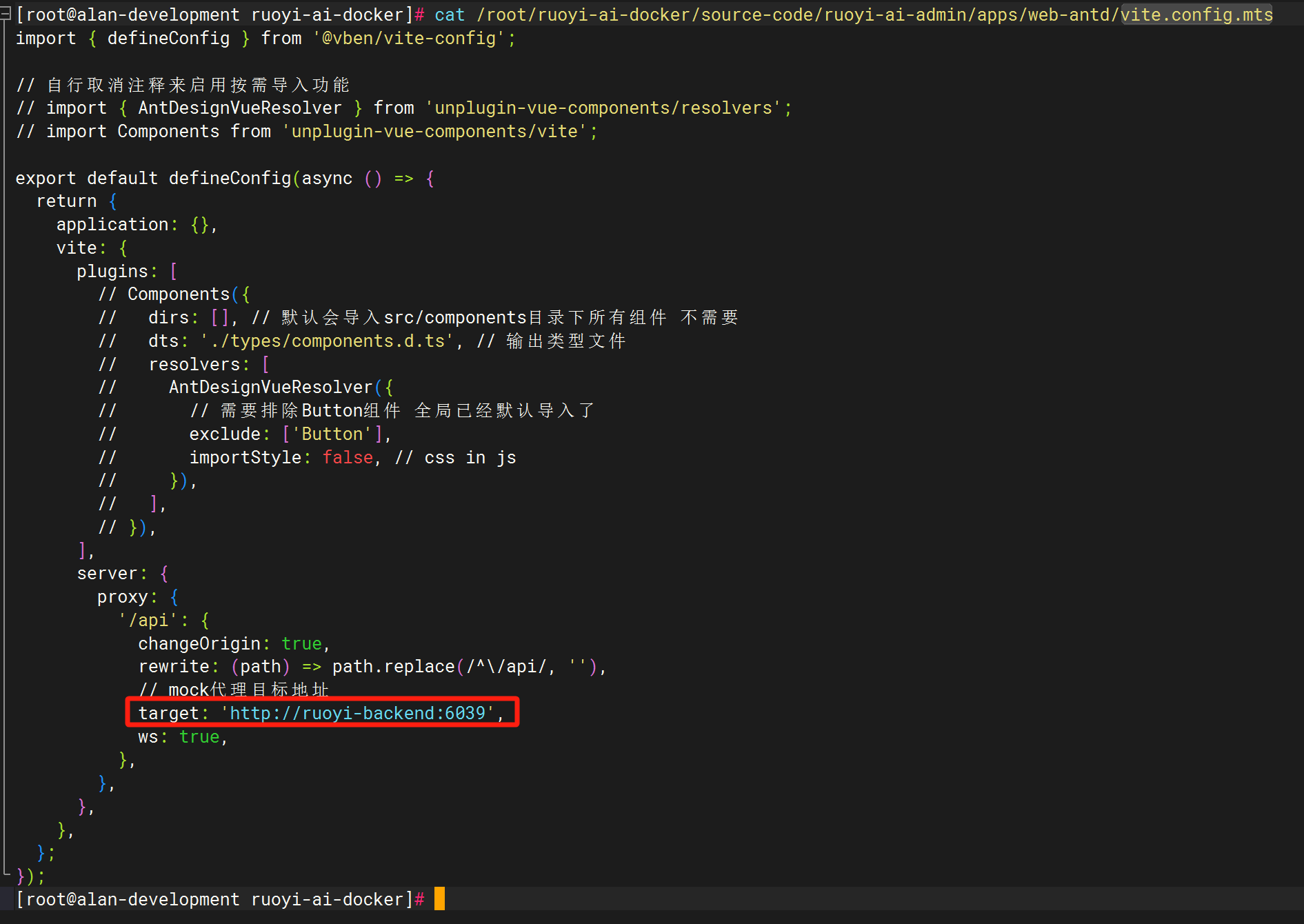
**cp /root/ruoyi-ai-docker/build-docker-images/modify-source-code/ruoyi-ai-admin/.env.analyze /root/ruoyi-ai-docker/source-code/ruoyi-ai-admin/apps/web-antd/**

**cp /root/ruoyi-ai-docker/build-docker-images/modify-source-code/ruoyi-ai-admin/.env.development /root/ruoyi-ai-docker/source-code/ruoyi-ai-admin/apps/web-antd/**

**cp /root/ruoyi-ai-docker/build-docker-images/modify-source-code/ruoyi-ai-admin/.env.production /root/ruoyi-ai-docker/source-code/ruoyi-ai-admin/apps/web-antd/**

**cp /root/ruoyi-ai-docker/build-docker-images/modify-source-code/ruoyi-ai-admin/.env.test /root/ruoyi-ai-docker/source-code/ruoyi-ai-admin/apps/web-antd/**

**cp /root/ruoyi-ai-docker/build-docker-images/modify-source-code/ruoyi-ai-admin/vite.config.mts /root/ruoyi-ai-docker/source-code/ruoyi-ai-admin/apps/web-antd/**



将vite.config.mts 文件里将target的地址修改为：'http://ruoyi-backend:6039', 然后进行编译打包：

**docker run --rm --name build-ruoyi-ai-admin -v /root/ruoyi-ai-docker/source-code/ruoyi-ai-admin:/app -w /app node:20 bash -c "npm install -g pnpm && pnpm install && pnpm build"**

如果网络正常，代码没有错误，耐心等待编译构建过程直至结束，得到生成的dist.zip文件：

**/root/ruoyi-ai-docker/source-code/ruoyi-ai-admin/apps/web-antd/dist.zip**

1. 用户端界面前端程序JS目录编译打包

**docker run --rm --name build-ruoyi-ai-web -v /root/ruoyi-ai-docker/source-code/ruoyi-ai-web:/app -w /app node:20 bash -c "npm install -g pnpm && pnpm install && pnpm build"**

如果网络正常，代码没有错误，耐心等待编译构建过程直至结束，得到生成的dist文件夹：

**/root/ruoyi-ai-docker/source-code/ruoyi-ai-web/dist**

1. 构建容器镜像
2. 构建Java后端服务容器镜像

**cd /root/ruoyi-ai-docker/build-docker-images/Dockerfile-Resources/ruoyi-ai-backend/**

在当前目录创建Dockerfile文件内容如下：

**FROM openjdk:17-jdk**

**WORKDIR /app**

**COPY ruoyi-admin.jar /app/ruoyi-admin.jar**

**EXPOSE 8080**

**ENTRYPOINT ["java","-jar","ruoyi-admin.jar"]**

拷贝前阶段得到的ruoyi-admin.jar文件到当前目录，即可开始构建容器镜像：

**cp /root/ruoyi-ai-docker/source-code/ruoyi-ai-backend/ruoyi-admin/target/ruoyi-admin.jar ./**

**docker build -t ruoyi-ai-backend:v2.0.4 .**

1. 构建管理端界面前端服务容器镜像

**cd /root/ruoyi-ai-docker/build-docker-images/Dockerfile-Resources/ruoyi-ai-admin/**

在当前目录创建Dockerfile内容如下：

**FROM nginx:1.25-alpine**

**COPY dist/ /usr/share/nginx/html/**

**COPY nginx.conf /etc/nginx/conf.d/default.conf**

**EXPOSE 80**

**CMD ["nginx", "-g", "daemon off;"]**

在当前目录创建nginx软件的配置文件default.conf如下：

**server {**

**listen 80;**

**server\_name localhost;**

**location / {**

**root /usr/share/nginx/html;**

**index index.html index.htm;**

**try\_files $uri $uri/ /index.html;**

**}**

**location /prod-api/ {**

**proxy\_pass http://ruoyi-backend:6039/;**

**proxy\_set\_header Host $host;**

**proxy\_set\_header X-Real-IP $remote\_addr;**

**proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;**

**}**

**error\_page 500 502 503 504 /50x.html;**

**location = /50x.html {**

**root /usr/share/nginx/html;**

**}**

**}**

拷贝前阶段得到的dist.zip文件夹到当前目录，即可开始构建容器镜像：

**cp /root/ruoyi-ai-docker/source-code/ruoyi-ai-admin/apps/web-antd/dist.zip ./**

**unzip dist.zip -d dist**

**rm -f dist.zip**

**docker build -t ruoyi-ai-admin:v2.0.4 .**

1. 构建用户端界面前端服务容器镜像

**cd /root/ruoyi-ai-docker/build-docker-images/Dockerfile-Resources/ruoyi-ai-web/**

在当前目录创建Dockerfile内容如下：

**FROM nginx:1.25-alpine**

**COPY dist/ /usr/share/nginx/html/**

**COPY nginx.conf /etc/nginx/conf.d/default.conf**

**EXPOSE 80**

**CMD ["nginx", "-g", "daemon off;"]**

在当前目录创建nginx软件的配置文件default.conf如下：

**server {**

**listen 80;**

**server\_name localhost;**

**location / {**

**root /usr/share/nginx/html;**

**index index.html index.htm;**

**try\_files $uri $uri/ /index.html;**

**}**

**location /api/ {**

**proxy\_pass http://ruoyi-backend:6039/;**

**proxy\_set\_header Host $host;**

**proxy\_set\_header X-Real-IP $remote\_addr;**

**proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;**

**}**

**error\_page 500 502 503 504 /50x.html;**

**location = /50x.html {**

**root /usr/share/nginx/html;**

**}**

**}**

拷贝前阶段得到的dist.zip文件夹到当前目录，即可开始构建容器镜像：

**cp -pr /root/ruoyi-ai-docker/source-code/ruoyi-ai-web/dist ./**

**docker build -t ruoyi-ai-web:v2.0.4 .**

1. 编写环境变量文件.env和docker-compose.yaml容器应用栈单机编排文件
2. 拷贝数据库初始化文件到MySQL容器需要挂载调用的目录

**mkdir -p /root/ruoyi-ai-docker/deploy/mysql-init**

**cd /root/ruoyi-ai-docker/deploy/mysql-init/**

**cp /root/ruoyi-ai-docker/source-code/ruoyi-ai-backend/script/sql/ruoyi-ai.sql ./01\_ruoyi-ai.sql**

**cp /root/ruoyi-ai-docker/source-code/ruoyi-ai-backend/script/sql/update/20250407.sql ./02\_update\_20250407.sql**

**cp /root/ruoyi-ai-docker/source-code/ruoyi-ai-backend/script/sql/update/20250505.sql ./03\_update\_20250505.sql**

**cp /root/ruoyi-ai-docker/source-code/ruoyi-ai-backend/script/sql/update/20250509.sql ./04\_update\_20250509.sql**

**cd /root/ruoyi-ai-docker/deploy/**

1. 在当前目录创建环境变量文件.env

**# Timezone**

**TZ=Asia/Shanghai**

**# MySQL Configuration**

**MYSQL\_ROOT\_PASSWORD=root**

**MYSQL\_DATABASE=ruoyi-ai**

**MYSQL\_PORT=3306**

**# Redis Configuration**

**REDIS\_PORT=6379**

**REDIS\_PASSWORD=**

**REDIS\_DATABASE=0**

**REDIS\_TIMEOUT=10s**

**# Backend Configuration**

**BACKEND\_SERVER\_PORT=6039**

**DB\_URL=jdbc:mysql://mysql:3306/ruoyi-ai?useUnicode=true&characterEncoding=utf8&zeroDateTimeBehavior=convertToNull&useSSL=true&serverTimezone=GMT%2B8&autoReconnect=true&rewriteBatchedStatements=true**

**DB\_USERNAME=root**

**DB\_PASSWORD=root**

**REDIS\_HOST=redis**

**# Admin Configuration**

**ADMIN\_SERVER\_PORT=8082**

**# Web Configuration**

**WEB\_SERVER\_PORT=8081**

1. 在当前目录创建docker-compose.yaml文件

**version: '3'**

**services:**

**mysql:**

**image: mysql:8.0.33**

**container\_name: ruoyi-ai-mysql**

**env\_file:**

**- ./.env**

**environment:**

**- MYSQL\_ROOT\_PASSWORD=${MYSQL\_ROOT\_PASSWORD}**

**- MYSQL\_DATABASE=${MYSQL\_DATABASE}**

**#ports:**

**# - "${MYSQL\_PORT}:3306"**

**volumes:**

**- ./mysql-init:/docker-entrypoint-initdb.d**

**- ./data/mysql:/var/lib/mysql**

**command:**

**--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**

**restart: always**

**networks:**

**- ruoyi-net**

**redis:**

**image: redis:6.2**

**container\_name: ruoyi-ai-redis**

**env\_file:**

**- ./.env**

**#ports:**

**# - "${REDIS\_PORT}:6379"**

**volumes:**

**- ./data/redis:/data**

**command: redis-server --appendonly yes ${REDIS\_PASSWORD:+--requirepass ${REDIS\_PASSWORD}}**

**restart: always**

**networks:**

**- ruoyi-net**

**ruoyi-backend:**

**image: ruoyi-ai-backend:v2.0.4**

**container\_name: ruoyi-ai-backend**

**env\_file:**

**- ./.env**

**ports:**

**- "${BACKEND\_SERVER\_PORT}:${BACKEND\_SERVER\_PORT}"**

**volumes:**

**- ./data/logs:/ruoyi/server/logs**

**restart: always**

**depends\_on:**

**- mysql**

**- redis**

**networks:**

**- ruoyi-net**

**ruoyi-admin:**

**image: ruoyi-ai-admin:v2.0.4**

**container\_name: ruoyi-ai-admin**

**ports:**

**#- "8082:80"**

**- "${ADMIN\_SERVER\_PORT}:80"**

**restart: always**

**depends\_on:**

**- ruoyi-backend**

**networks:**

**- ruoyi-net**

**ruoyi-web:**

**image: ruoyi-ai-web:v2.0.4**

**container\_name: ruoyi-ai-web**

**ports:**

**#- "8081:80"**

**- "${WEB\_SERVER\_PORT}:80"**

**restart: always**

**depends\_on:**

**- ruoyi-backend**

**networks:**

**- ruoyi-net**

**networks:**

**ruoyi-net:**

**driver: bridge**

1. 启动全套服务

**cd /root/ruoyi-ai-docker/deploy/**

**docker-compose up -d**

**docker-compose logs -f**