

# dev服务器部署位置 (mxdemo2.qunl.com)

服务器用户名: root

服务器密码: 7r(hf1\$T

部署位置 /data/app/zhijian

```
[root@k8s-master zhijian]# ll
total 40
drwxr-xr-x  3 root root 4096 Nov 27 16:27 backend #后端
drwxr-xr-x  4 root root 4096 Nov 28 09:39 bge-service
drwxr-xr-x  7 root root 4096 Nov 27 10:52 data #数据挂载
-rw-r--r--  1 root root 3610 Nov 27 17:51 docker-compose.yml
drwxr-xr-x  3 root root 4096 Nov 27 11:21 frontend #前端
drwxr-xr-x  2 root root 4096 Nov 27 17:58 images
drwxr-xr-x 10 root root 4096 Nov 27 13:10 logs #日志
drwxr-xr-x  3 root root 4096 Nov 27 10:52 nginx # nginx
drwxr-xr-x  2 root root 4096 Nov 27 17:05 scripts #启动/更新/停止脚本
drwxr-xr-x  2 root root 4096 Nov 27 14:25 xxl-init
```

## 📋 服务信息汇总表 (含 Web 地址)

服务名称	用户名	密码	主机端口	Web/API 访问地址 (通过你的域名)	用途说明
MySQL	root / model	Hangju@2025@root / Hangju@2025	3307	✗ 不直接对外提供 Web 访问	数据库服务, 供后端和 XXL-Job 使用
Redis	(无用户名)	Redis@Hangju@2025	6389	✗ 无 Web 界面	缓存服务
ChromaDB	(无认证)	(无密码)	8000	<a href="http://mxdemo2.qunl.com:8000">http://mxdemo2.qunl.com:8000</a>	向量数据库, 提供 Embedding 存储与检索
MinIO	minioadmin	minioadmin@Hangju@2025	19000 19001	API: <a href="http://mxdemo2.qunl.com:19000">http://mxdemo2.qunl.com:19000</a> 控制台: <a href="http://mxdemo2.qunl.com:19001">http://mxdemo2.qunl.com:19001</a>	对象存储服务, 用于文件/模型存储
XXL-Job	admin	xxljob@Hangju@2025	9888	<a href="http://mxdemo2.qunl.com:9888/xxl-job-admin/">http://mxdemo2.qunl.com:9888/xxl-job-admin/</a>	分布式任务调度平台 (首次登录需用默认账号)
后端	(由应用逻辑控制)	(如 JWT、OAuth 等)	10023	<a href="http://mxdemo2.qunl.com:10023">http://mxdemo2.qunl.com:10023</a>	后端 API 服务 (Spring Boot 应用)
前端	郭新雨	Gxy132119..	10022	<a href="http://mxdemo2.qunl.com:10022">http://mxdemo2.qunl.com:10022</a>	前端静态资源托管 + 可能的反向代理入口

服务名称	用户名	密码	主机端口	Web/API 访问地址 (通过你的域名)	用途说明
向量化服务	(无认证)	(无密码)	9099	<a href="https://modest-mudfish-freely.ngrok-free.app/embed">https://modest-mudfish-freely.ngrok-free.app/embed</a>	暂用内网穿透

## 后续维护命令

```
# 更新后端
# 无需手动备份，只需要将新的jar包命名为非启动jar名称，脚本会自行根据时间备份重启
./scripts/update-backend.sh

# 更新前端 需手动备份

# 重启所有服务
./scripts/restart.sh

# 查看特定服务日志
docker-compose logs -f backend
docker-compose logs -f mysql

# 进入容器调试
docker-compose exec backend sh

#重启mysql
docker-compose up -d mysql
```

## test服务器部署位置 ([mxdemo1.qunl.com](http://mxdemo1.qunl.com))

服务器用户名: root

服务器密码: 7r(hf1\$T

部署位置/home/data/app/zhijian

```
[root@k8s-master zhijian]# ll
total 40
drwxr-xr-x  3 root root 4096 Nov 27 16:27 backend #后端
drwxr-xr-x  4 root root 4096 Nov 28 09:39 bge-service
drwxr-xr-x  7 root root 4096 Nov 27 10:52 data  #数据挂载
-rw-r--r--  1 root root 3610 Nov 27 17:51 docker-compose.yml
drwxr-xr-x  3 root root 4096 Nov 27 11:21 frontend #前端
drwxr-xr-x  2 root root 4096 Nov 27 17:58 images
drwxr-xr-x 10 root root 4096 Nov 27 13:10 logs      #日志
drwxr-xr-x  3 root root 4096 Nov 27 10:52 nginx    # nginx
drwxr-xr-x  2 root root 4096 Nov 27 17:05 scripts  #启动/更新/停止脚本
drwxr-xr-x  2 root root 4096 Nov 27 14:25 xxl-init
```

## 📋 服务信息汇总表 (含 Web 地址)

服务名称	用户名	密码	主机端口	Web/API 访问地址 (通过你的域名)	用途说明
MySQL	root / model	Hangju@2025@root / Hangju@2025	3307	✗ 不直接对外提供 Web 访问	数据库服务, 供后端和 XXL-Job 使用
Redis	(无用户名)	Redis@Hangju@2025	6389	✗ 无 Web 界面	缓存服务
ChromaDB	(无认证)	(无密码)	8000	<a href="http://mxdemo1.qunl.com:8000">http://mxdemo1.qunl.com:8000</a>	向量数据库, 提供 Embedding 存储与检索
MinIO	minioadmin	minioadmin@Hangju@2025	19000 19001	API: <a href="http://mxdemo1.qunl.com:19000">http://mxdemo1.qunl.com:19000</a> 控制台: <a href="http://mxdemo1.qunl.com:19001">http://mxdemo1.qunl.com:19001</a>	对象存储服务, 用于文件/模型存储
XXL-Job	admin	xxljob@Hangju@2025	9888	<a href="http://mxdemo1.qunl.com:9888/xxl-job-admin/">http://mxdemo1.qunl.com:9888/xxl-job-admin/</a>	分布式任务调度平台 (首次登录需用默认账号)
后端	(由应用逻辑控制)	(如 JWT、OAuth 等)	10023	<a href="http://mxdemo1.qunl.com:10023">http://mxdemo1.qunl.com:10023</a>	后端 API 服务 (Spring Boot 应用)
前端	郭新雨	Gxy132119..	10022	<a href="http://mxdemo1.qunl.com:10022">http://mxdemo1.qunl.com:10022</a>	前端静态资源托管 + 可能的反向代理入口
向量化服务	(无认证)	(无密码)	9099	<a href="https://modest-mudfish-freely.ngrok-free.app/embed">https://modest-mudfish-freely.ngrok-free.app/embed</a>	暂用内网穿透

## ⌚ 后续维护命令

```
# 更新后端
# 无需手动备份, 只需要将新的jar包命名为非启动jar名称, 脚本会自行根据时间备份重启
./scripts/update-backend.sh

# 更新前端 需手动备份

# 重启所有服务
./scripts/restart.sh

# 查看特定服务日志
docker-compose logs -f backend
docker-compose logs -f mysql

# 进入容器调试
docker-compose exec backend sh
```

#重启mysql

```
docker-compose up -d mysql
```