|  |  |
| --- | --- |
| LIMS 产品容器化  版本 1.0 | 摘要  LIMS产品容器化, 快速发布,节省成本. 手把手带你开启Docker之旅  齐俊波  qijb@sunwayworld.com |

**目录**

[前言 1](#_Toc489960258)

[在宿主机器上安装Docker 2](#_Toc489960259)

[把数据库容器化 3](#_Toc489960260)

[MySQL 容器的初始化 3](#_Toc489960261)

[如何从外部访问MySQL 3](#_Toc489960262)

[如何从内部访问MySQL 3](#_Toc489960263)

[LIMS 数据库初始化 4](#_Toc489960264)

[把LIMS应用容器化 4](#_Toc489960265)

[创建Dockerfile. 4](#_Toc489960266)

[创建 Docker 镜像 4](#_Toc489960267)

[启动LIMS容器 4](#_Toc489960268)

[创建开发专用Tomcat容器 5](#_Toc489960269)

[特别说明: 5](#_Toc489960270)

[创建Dockerfile 5](#_Toc489960271)

[定制化Tomcat镜像 5](#_Toc489960272)

[启动Tomcat容器 5](#_Toc489960273)

[发布LIMS 6](#_Toc489960274)

[上传LIMS镜像 6](#_Toc489960275)

[下载LIMS镜像 6](#_Toc489960276)

特别说明： 随本文档一起发布的还包含一个压缩包lims.zip , 其中包含了部署LIMS所需的配置文件。

LIMS 产品的容器化

# 前言

和虚拟机相比，Dockre容器具有轻量级的优点，同样硬件配置的机器上，能够同时运行Docker 容器实例的数量要远大于能够运行Vmware 虚拟机实例的数量。 LIMS 产品容器化以后，能够简化部署过程，并能够保证环境的一致性。 运维人员从云端获取Docker镜像后，能够几分钟内完成部署，并且同一个镜像在各个机器上的运行时的行为高度一致， 大大的减少了配置不一致带来运行效果不一致带来的麻烦。

**主要步骤**

完成LIMS产品的容器化，主要需要完成如下几个过程。

1. 在宿主机器上安装Docker
2. 把数据库容器化
3. 把LIMS应用容器化
4. 发布LIMS

# 在宿主机器上安装Docker

下面是Docker在CentOS Linux 下面的安装步骤， 如果你用的是其它操作系统可以点击如下链接查看官方网站对其他操作系统安装步骤的说明：<https://docs.docker.com/engine/installation/linux/docker-ce/centos/#install-using-the-repository>

yum **install** -y yum-utils device-mapper-persistent-**data** lvm2

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

sudo yum **install** docker-ce

systemctl **start** docke

安装之后可以用下面的指令简单验证一下是否安装成功。

docker version

# 把数据库容器化

这进行这一步工作的时候，有一个非常终于的概念我们必须先弄清楚。 容器运行期间，所有发生在容器**内部**的改动，在容器关闭后都会丢失。 当你再次启动这个容器时，一切都回到了**出厂状态**。 正因为如此，我们必须把数据库文件和数据库配置文件**外部化**.

下面两个链接是MySQL官方提供的对镜像文件的详细说明，以及镜像是如何生成的。 虽然你不一定非要了解细节， 但是当你遇到困难时，了解这些内容有助于解决问题。

<https://github.com/docker-library/mysql/blob/7a850980c4b0d5fb5553986d280ebfb43230a6bb/8.0/Dockerfile>

<https://github.com/mysql/mysql-docker>

## MySQL 容器的初始化

创建用来保存数据库文件和数据库配置文件的目录

mkdir -p /root/mysql/user1/data

mkdir -p /root/mysql/user1/conf

copy the docker.cnf and mysql.cnf from source forder conf to /root/mysql/user1/conf 这些文件可以在说明文档一起发布的包里面找到。

docker run --name mysql -e MYSQL\_ROOT\_PASSWORD=sunway123*# -d -p 3306:3306 \*

-v /root/mysql/user1/data:/var/**lib**/**mysql** \

-v /root/mysql/user1/conf:/etc/mysql/conf.d mysql

**发生了什么？** 完成上面两个步骤后，外面的两个目录和容器内部的目录就完全共享了。 /root/mysql/user1/data => /var/lib/mysql

/root/mysql/user1/conf => /etc/mysql/conf.d

## 如何从外部访问MySQL

mysql 超级用户 root 密码： sunway123# ip地址可以到宿主机器上查看。

知道以上信息后，可以用任意第三方数据库客户端工具远程链接，并创建数据库。

## 如何从内部访问MySQL

执行如下指令，即可进入mysql容器内部。

docker container exec -**it** mysql bash

mysql -u root -p

密码： sunway123*#*

## LIMS 数据库初始化

(内容仅供参考，一切以LIMS实际发布的产品为准)

docker container exec -it mysql bash

mysql -u root -p

CREATE DATABASE IF NOT EXISTS dbs\_dev DEFAULT CHARSET utf8 COLLATE utf8\_general\_ci;

use dbs\_dev;

source /var/**lib**/**mysql**/**dbs\_dev**.**sql**

# 把LIMS应用容器化

## 创建Dockerfile.

这个是生产环境专用。LIMS打包到容器内部， 减少发布时间和与运维人员的沟通成本。

*#Create a LIMS image based on tomcat8*

FROM tomcat:latest

COPY iframework /usr/local/iframework

COPY server.xml /usr/local/tomcat/conf/server.xml

## 创建 Docker 镜像

在做这个工作之前，要把最新版的LIMS 发布文件(iframework)和tomcat配置文件server.xml 复制到Dockerfile同一个目录下。

docker image rm sunway/lims:1

docker image build -t sunway/lims:1 .

## 启动LIMS容器

docker container run --name lims -d -p 8080:8080 sunway/lims:1

点击如下链接尝试是否可以成功访问系统

<http://192.168.1.30:8080/iframework>

# 创建开发专用Tomcat容器

## 特别说明:

开发版容器使用步骤如下:

1. 把LIMS复制到：/root/docker/mylims/iframework

2. 启动定制化的tomcat容器。

docker container run --name mylims -d -P \

-v /root/docker/mylims/logs:/usr/local/tomcat/logs \

-v /root/docker/mylims/iframework:/usr/local/iframework sunway/mylims

3. 到这个目录查看日志：

/root/docker/mylims/logs

4. 查看运行在哪个端口：

docker port mylims

## 创建Dockerfile

*# Setup a tomcat 8 as a development environment.*

FROM tomcat:latest

VOLUME /usr/local/iframework /usr/local/tomcat/logs

COPY server.xml /usr/local/tomcat/conf/server.xml

## 定制化Tomcat镜像

docker **image** rm sunway/mylims

docker **image** build -t sunway/mylims .

## 启动Tomcat容器

docker container run --name mylims -d -P \

-v /root/docker/mylims/logs:/usr/local/tomcat/logs \

-v /root/docker/mylims/iframework:/usr/local/iframework sunway/mylims

docker exec -it mylims bash

http:*//192.168.1.30:<port>/iframework*

# 发布LIMS

当LIMS产品Docker容器化后，可以很方便的把LIMS产品发布到云端，部署时可以直接从云端下载。 由于Docker公司在美国，所以我们一般从国内的云平台购买Docker服务。

## 上传LIMS镜像

把我们前面创建好的LIMS镜像推送到云端。

docker push <registry-host>:5000/sunway/lims:1

## 下载LIMS镜像

从云端下载镜像，运行

docker pull <myregistry.local>:5000/sunway/lims:1

docker container run --name lims -d -p 8080:8080 sunway/lims:1

这样客户就可以直接访问LIMS服务器了。

<http://192.168.1.30:8080/iframework>

文档结束