

文书生成系统部署文档

版本: v20180803

目录

- 1. 操作系统版本要求
- 2. 软件环境
- 3. 部署前
 - 3.1 端口说明
 - 3.2 安装 docker
- 4. 外网环境下部署
 - 4.1 编译 docker 镜像
 - 4.2 运行各服务
- 5. 内网环境下部署
 - 5.1 外网环境下拉取/编译 docker 镜像
 - 5.2 保存本地镜像
 - 5.3 加载本地镜像
 - 5.4 运行服务
- 6. 测试是否部署成功

1. 操作系统版本要求

CentOS 7, 或Ubuntu 14.04, 或 CentOS 6.5

2. 软件环境

1. CentOS 7, 或Ubuntu 14.04

docker 最新版

2. CentOS 6.5

Docker (docker.io) version 1.7.1 或更高版本

3. 部署前

3.1 端口说明

本系统使用的端口有：

1. 7099 : supervisor
2. 8009 : 文书生成服务, 需要保证该端口可用
3. 8019 : OnlyOffice 服务, 需要保证该端口可用
4. 8029 : 对象存储服务, 需要保证该端口可用
5. 13306 : MySQL 5.7

建议部署前首先确保以上端口无占用。

3.2 安装 docker

2.1 内外网环境下 CentOS 7 安装 docker

参阅: [安装文档 \(https://docs.docker.com/install/linux/docker-ce/centos/\)](https://docs.docker.com/install/linux/docker-ce/centos/)

2.2 内外网环境下 Ubuntu 14.04 安装 docker

参阅: [安装文档 \(https://docs.docker.com/install/linux/docker-ce/ubuntu/\)](https://docs.docker.com/install/linux/docker-ce/ubuntu/)

2.3 内外网环境下 CentOS 6.5 安装 docker

参阅: [安装文档 \(https://docs.docker.com/v1.7/docker/installation/centos/\)](https://docs.docker.com/v1.7/docker/installation/centos/)

4. 外网环境下部署

4.1 编译 docker 镜像

```
# cd 到文书生成应用服务器 Dockerfile 文件所在的路径
cd ../wssc/docker/

# tag: wssc:1.0
docker build -t wssc:1.0 .
```

4.2 运行各服务

按照下列顺序依次运行镜像

4.2.1 运行 minio 对象存储服务

```
sudo docker run -d -p 8029:9000 --name=minio -e  
MINIO_ACCESS_KEY=minio -e MINIO_SECRET_KEY=minio123 -v /var/www  
/minio/data:/data -v /var/www/minio/config:/config -v /var/www  
/minio/exchange:/exchange minio/minio server -C /config /data
```

说明:

1. 该服务映射端口至 8029 , 容器名为 minio
2. 该服务用来存储文件, 文件在服务器上的保存位置是 /var/www/minio/data
3. 对象存储服务 web 页面访问端口是 8029 , 用户名和密码是 minio , minio123

4.2.2 运行 MySQL 5.7 数据库服务

```
docker run --name mysql57 -v /var/www/mysql:/var/lib/mysql -e  
MYSQL_ROOT_PASSWORD=root -p 13306:3306 -d mysql:5.7  
  
docker exec -it mysql57 mysql -uroot -proot -e "create database  
wssc charset='utf8';"
```

说明:

1. 该服务映射端口至 13306 , 容器名为 mysql57
2. 数据库在服务器上的保存位置是 /var/www/mysql
3. 数据库 root 用户密码是 root

4.2.3 运行 OnlyOffice 文档编辑服务

```
sudo docker run -i -t -d -p 8019:80 --name=onlyoffice
--restart=always -v /var/www/onlyoffice/documentserver/logs:
/var/log/onlyoffice -v /var/www/onlyoffice/documentserver/data:
/var/www/onlyoffice/Data -v /var/www/onlyoffice/documentserver
/lib:/var/lib/onlyoffice -v /var/www/onlyoffice/documentserver
/db:/var/lib/postgresql -v /var/www/onlyoffice/documentserver
/fonts:/usr/share/fonts -v /var/www/onlyoffice/documentserver
/exchange:/exchange onlyoffice/documentserver
```

说明:

1. 该服务映射端口至 8019 , 容器名为 onlyoffice

4.2.4 修改文档编辑服务配置

```
# 首先 cd 到应用服务器 dockerfile 所在的目录
cd ../wssc/docker/

# 复制字体文件到指定目录
cp ../zh-fonts -r /var/www/onlyoffice/documentserver/fonts

docker exec -it onlyoffice /bin/bash documentserver-generate-
allfonts.sh
```

4.2.5 运行文书生成应用服务器

注意:

复制该命令后需要修改各环境变量的值, 各环境变量的含义如下:

环境变量	含义	可选值
SERVER_ENV_MODE	服务器运行环境	dev - 开发, stage - 过渡, production - 生产
SERVER_ENV_DB_HOST	数据库 ip	填写服务器 ip, 例如: 112.74.130.243
SERVER_ENV_DB_PORT	数据库端口	例如: 13306

环境变量	含义	可选值
SERVER_ENV_DB_NAME	数据库名称	如果运行 mysql 服务时, 没有修改数据库名称, 默认为 wssc
SERVER_ENV_DB_USER	数据库用户	默认是 root
SERVER_ENV_DB_PASS	数据库用户密码	默认是 root
SERVER_ENV_LOCAL_HOST	应用服务器后端 ip 和端口	填写本服务器 ip + 8009 端口, 例如: 112.74.130.243:800
SERVER_ENV_ONLY_OFFICE_HOST	文档编辑服务 ip 和端口	填写本服务器 ip + onlyoffice 文档编辑服务的端口, 例如: 112.74.130.243:801
SERVER_ENV_MINIO_HOST	对象存储服务 ip 和端口	填写本服务器 ip + 对象存储服务的端口, 例如: 112.74.130.243:802
SERVER_ENV_DOSSIER_HOST	电子卷宗服务的 ip 和端口	填写电子卷宗服务器 和端口, 例如: 172.29.226.90:8080

运行命令:

```
docker run -it --name=wssc -d -p 7079:7079 -p 7089:7089 -p 7099:7099 -p 8009:8009 -v /var/www/WenShuShengCheng/exchange:/var/www/WenShuShengCheng/exchange -e SERVER_ENV_MODE=dev -e SERVER_ENV_DB_HOST=112.74.130.243 -e SERVER_ENV_DB_PORT=13306 -e SERVER_ENV_DB_NAME=wssc -e SERVER_ENV_DB_PASS=root -e SERVER_ENV_DB_USER=root -e SERVER_ENV_LOCAL_HOST=112.74.130.243:8009 -e SERVER_ENV_ONLY_OFFICE_HOST=112.74.130.243:8019 -e SERVER_ENV_MINIO_HOST=112.74.130.243:8029 -e SERVER_ENV_DOSSIER_HOST=172.29.226.90:8080 wssc:1.0
```

说明:

1. 该服务映射端口至 7079 7089 7099 8009 , 容器名为 wssc .
2. 映射端口中需要确保 8009 , 其他的端口可以修改.

5. 内网环境下部署

5.1 外网环境下拉取/编译 docker 镜像

1. 本步骤在内外网环境下安装时都需要执行.
2. 本步骤需要在连接外网的电脑上执行.
3. 镜像文件总大小约 3GB

5.1.1 拉取对象存储 minio/minio , OnlyOffice onlyoffice/documentserver , MySQL 5.7 mysql:5.7 镜像

```
docker pull minio/minio:latest  
  
docker pull onlyoffice/documentserver:latest  
  
docker pull mysql:5.7
```

5.1.2 编译文书生成应用服务器镜像

```
# cd 到文书生成应用服务器 Dockerfile 文件所在的路径
cd ../wssc/docker/

# tag: wssc:1.0
docker build -t wssc:1.0 .
```

5.2 保存本地镜像

1. 本步骤仅在内网环境下安装时需要执行, 目的是将从外网拉取的镜像保存为压缩包, 稍后上传至内网环境.
2. 外网安装时跳过.

5.2.1 CentOS 7 或 Ubuntu 14.04

```
# minio
docker image save minio/minio:latest -o ../minio_latest.tar.gz

# mysql
docker image save mysql:5.7 -o ../mysql_5_7.tar.gz

# onlyoffice
docker image save onlyoffice:documentserver -o
../onlyoffice_latest.tar.gz

# 文书生成应用服务器
docker image save wssc:1.0 -o ../wssc_1_0.tar.gz
```

5.2.2 CentOS 6.5

```
# minio
docker save minio/minio:latest -o ./minio_latest.tar.gz

# mysql
docker save mysql:5.7 -o ./mysql_5_7.tar.gz

# onlyoffice
docker save onlyoffice:documentserver -o
./onlyoffice_latest.tar.gz

# 文书生成应用服务器
docker save wssc:1.0 -o ./wssc_1_0.tar.gz
```

5.3 加载本地镜像

1. 本步骤仅在内网环境下安装时需要执行. 目的是加载复制到内网环境内的 docker 镜像, 需要提前将镜像压缩包复制到内网环境中.
2. 外网安装时跳过.

5.3.1 复制镜像到内网机器

5.3.2 CentOS 7 或 Ubuntu 14.04

```
# minio
docker image load -i ./minio_latest.tar.gz

# mysql
docker image load -i ./mysql_5_7.tar.gz

# onlyoffice
docker image load -i ./onlyoffice_latest.tar.gz

# 文书生成应用服务器
docker image load -i ./wssc_1_0.tar.gz
```

5.3.3 CentOS 6.5


```
# minio
docker load -i ./minio_latest.tar.gz

# mysql
docker load -i ./mysql_5_7.tar.gz

# onlyoffice
docker load -i ./onlyoffice_latest.tar.gz

# 文书生成应用服务器
docker load -i ./wssc_1_0.tar.gz
```

5.4 运行服务

5.4.1 检查各镜像是否加载

```
# CentOS 7, Ubuntu 14.04
docker image ls

# CentOS 6.5
docker images
```

5.4.2 运行各服务

运行方式参阅 [4.2 运行各服务](#)，但需要注意的是，如果镜像加载后没有识别到 repository 和 tag，运行时需要使用 image id 替换，例如将下列命令中的 repository:tag 替换为 image id:

```
docker run --name mysql57 -v /var/www/mysql:/var/lib/mysql -e
MYSQL_ROOT_PASSWORD=root -p 13306:3306 -d mysql:5.7
```

```
docker run --name mysql57 -v /var/www/mysql:/var/lib/mysql -e
MYSQL_ROOT_PASSWORD=root -p 13306:3306 -d <IMAGE ID>
```

6. 测试是否部署成功

使用 360 浏览器或 chrome 浏览器访问 `http://<ip>:8009/welcome/`，如果看到 `Server is up and running...`，系统即部署完成。