

畅写在线 V6 安装部署手册

【2024 年版】



编号		版本	v6.7.5
日期	2024 年 11 月 27 日	密级	公开
适用范围	外部公开		

目录

- 1.服务器要求 4
 - 1.1.硬件配置 4
 - 1.2.操作系统 4
 - 1.3.依赖组件说明 4
 - 1.4.支持浏览器 5
- 2.部署包 5
 - 2.1.部署必读 5
 - 2.2.Docker 部署包 5
 - 2.3.部署架构 6
- 3.部署方式 6
 - 3.1.Docker 部署 6
 - 3.1.1.优化系统 6
 - 3.1.1.1.关闭防火墙 6
 - 3.1.1.1.1.CentOS 系统 7
 - 3.1.1.1.2.Ubuntu 系统 7
 - 3.1.1.2.关闭 selinux 7
 - 3.1.1.2.1.CentOS 系统 7
 - 3.1.1.2.2.Ubutun 系统 7
 - 3.1.2.部署 Docker 服务 7
 - 3.1.2.1.在线部署 7
 - 3.1.2.1.1.Ubutun 系统-apt 安装 7
 - 3.1.2.1.2.CentOS 系统-yum 安装 8
 - 3.1.2.2.离线部署 9
 - 3.1.3.部署 Docker-compose 10
 - 3.1.3.1.在线部署 10
 - 3.1.4.部署编辑器服务 10
 - 3.1.4.1.单机部署 10
 - 3.1.4.1.1.在线部署 12

3.1.4.1.2.离线部署	12
3.1.4.1.3.验证服务是否正常	13
3.1.4.1.4.获取机器码	13
3.1.4.1.5.做授权激活	13
3.1.4.2.集群部署	13
3.1.4.2.1.畅写编辑器依赖的组件说明	13
3.1.4.2.2.docker-compose.yaml 中关于依赖组件的配置信息	13
3.1.4.2.3.畅写编辑器依赖的组件说明	15
3.1.4.2.3.1.数据库	15
3.1.4.2.3.2.缓存	16
3.1.4.2.3.3.消息对列	16
3.1.4.2.3.4.存储 (NFS)	17
3.1.4.2.3.5.负载均衡和反向代理 (Nginx)	17
3.2.K8s 部署	18
3.3.原生部署	23
3.3.1.关闭防火墙或者开放端口	23
3.3.2.关闭 selinux	23
3.3.3.解压部署包	23
3.3.4.配置环境变量	24
3.3.5.修改配置文件	24
3.3.5.1.创建数据库	25
3.3.5.2.数据库配置	26
3.3.5.3.redis 配置	26
3.3.5.4.配置缓存本机目录	27
3.3.6.配置运行环境	28
3.3.7.授权页面	29
3.3.8.演示程序配置	29
3.3.9.演示程序访问	29
4.授权页面	30
4.1.授权页面	30
4.2.演示示例	30

1.服务器要求

1.1.硬件配置

序号	名称	最小配置	配置说明	网络环境
1	畅写应用服务器	CPU: 4C+ 内存: 8G+ 硬盘: 200GB+ 网卡: 千兆	用于部署畅写在线应用程序 用于部署数据库、文档缓存等	支持内网和公网部署

1.2.操作系统

- 统信服务器操作系统 V20
- Ubuntu 16.04 或以上
- CentOS7.4 或以上
- Redhat7.4 或以上
- 中标麒麟服务器操作系统 V7.0
- 银河麒麟 V10
- 湖南麒麟 V3.3-6A
- 湖南麒麟 V3.3-6B
- EulerOS 2.8

1.3.依赖组件说明

注：以下组件单机模式容器中会内置第三方组件，集群模式必须连接外部同一个第三方组件。我们建议自行准备第三方组件，这样会更方便您的数据维护。

以下组件需自行部署成单机环境或集群环境（Redis 需要为单机或主从模式）

- 数据库：MySQL 5.7、PostgreSQL 9.5 及以上、oracle19c 、dmdb 7.6(arm64)、kingbase(v8)

- 缓存: Redis 4.0 以上
- 消息队列: rabbitmq 3.8、 kafka 2.13 (支持 SASL 认证方式)
- 存储: NFS
- 负载均衡和反向代理: Nginx 1.18.0

1.4.支持浏览器

浏览器名称	低版本号
Google-Chrome	63.0.3239.132(2018-01-05)
Edge	42.17134.1.0(2018-04-30)
360 极速浏览器	9.0.1.1562018-01-08)
Firefox	60

2.部署包

2.1.部署必读

畅写产品支持 3 种部署方式: docker 部署、k8s 部署、原生部署。默认推荐使用 docker 部署

1. docker 部署包含两大类部署:

- ✓ 在宿主机上部署 docker
- ✓ 基于 docker 部署畅写服务

2. k8s 部署:

- ✓ 自行部署 k8s 基础环境, 准备第三方组件依赖

3. 原生部署:

- ✓ 自行准备第三方组件依赖, 下载原生部署包。

2.2.Docker 部署包

官方下载地址: <https://download.docker.com/linux/static/stable/>

X86 体系 Docker 离线部署包

链接: <https://pan.baidu.com/s/1moout9vxJF4tTIUJt9PF4Q>

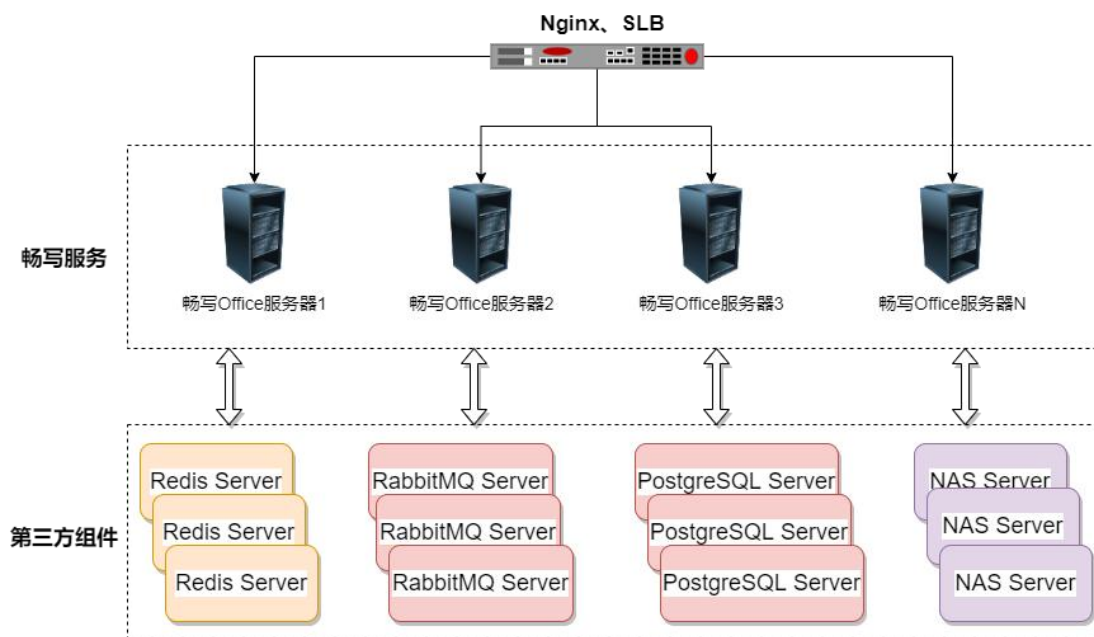
提取码: lktn

ARM 体系 Docker 离线部署包

链接: <https://pan.baidu.com/s/16MKMm1BCF5MO8RnwsUnkuQ>

提取码: 3bjb

2.3.部署架构



3.部署方式

3.1.Docker 部署

3.1.1.优化系统

```
sysctl -w net.core.somaxconn=65535
sysctl -w net.ipv4.ip_local_port_range="1024 65535"
sysctl -w net.ipv4.tcp_max_tw_buckets=55000
sysctl -w net.ipv4.tcp_tw_reuse=1
sysctl -w fs.file-max=1048576
sysctl -w net.ipv4.tcp_fin_timeout=15
sysctl -w net.netfilter.nf_conntrack_tcp_timeout_time_wait=30
```

3.1.1.1.关闭防火墙

3.1.1.1.1.CentOS 系统

```
# systemctl disable --now firewalld
```

3.1.1.1.2.Ubuntu 系统

```
# ufw disable
```

3.1.1.2.关闭 selinux

3.1.1.2.1.CentOS 系统

```
# setenforce 0  
# sed -i 's/^SELINUX=enforcing$/SELINUX=disabled/' /etc/selinux/config
```

3.1.1.2.2.Ubutun 系统

不需要做这一步

3.1.2.部署 Docker 服务

3.1.2.1.在线部署

3.1.2.1.1.Ubutun 系统-apt 安装

安装必要的一些系统工具

```
sudo apt-get update  
sudo apt-get -y install apt-transport-https ca-certificates curl  
software-properties-common
```

安装 GPG 证书

```
curl -fsSL http://mirrors.aliyun.com/docker-ce/linux/ubuntu/gpg | sudo apt-key add -
```

写入软件源信息

```
sudo add-apt-repository "deb [arch=amd64]  
http://mirrors.aliyun.com/docker-ce/linux/ubuntu $(lsb_release -cs) stable"
```

更新并安装 Docker CE

```
sudo apt-get -y update
sudo apt-get -y install docker-ce
```

建立 docker 用户组

默认情况下，docker 命令会使用 Unix socket 与 Docker 引擎通讯。而只有 root 用户和 docker 组的用户才可以访问 Docker 引擎的 Unix socket。出于安全考虑，一般 Linux 系统上不会直接使用 root 用户。因此，更好地做法是将需要使用 docker 的用户加入 docker 用户组。

建立 docker 组:

```
sudo groupadd docker
```

将当前用户加入 docker 组:

```
sudo usermod -aG docker $USER
```

修改 docker 默认存储位置

使用 ln 软连接: 停止 docker 服务后, 把 docker 中的内容全部拷贝到指定的目录, 然后删除 docker 目录, 最后使用 ln 软连接到指定的目录

#移动文件位置

```
cp -a /var/lib/docker /data/
```

#创建软连接

```
sudo ln -fs /data/docker /var/lib/docker
```

#重新加载配置&查看位置

```
systemctl daemon-reload
```

```
systemctl restart docker
```

重新启动 docker 服务

```
sudo systemctl restart docker
```

3.1.2.1.2.CentOS 系统-yum 安装

执行以下命令安装依赖包:

```
sudo yum install -y yum-utils \
    device-mapper-persistent-data \
    lvm2
```

鉴于国内网络问题, 强烈建议使用国内源, 官方源请在注释中查看。

执行下面的命令添加 yum 软件源:

```
sudo yum-config-manager \  
  --add-repo \  
  https://mirrors.usc.edu.cn/docker-ce/linux/centos/docker-ce.repo
```

安装指定版本 docker-ce-edge (默认为最新版本)

```
sudo yum-config-manager --enable docker-ce-edge
```

更新 yum 软件源缓存, 并安装 docker-ce

```
sudo yum makecache fast  
sudo yum install docker-ce
```

启动 docker-ce 并设置开机自启动

```
sudo systemctl enable docker  
sudo systemctl start docker
```

建立 docker 用户组

默认情况下, docker 命令会使用 Unix socket 与 Docker 引擎通讯。而只有 root 用户和 docker 组的用户才可以访问 Docker 引擎的 Unix socket。出于安全考虑, 一般 Linux 系统上不会直接使用 root 用户。因此, 更好地做法是将需要使用 docker 的用户加入 docker 用户组。

建立 docker 组:

```
sudo groupadd docker
```

将当前用户加入 docker 组:

```
sudo usermod -aG docker $USER
```

重新启动 docker 服务

```
sudo systemctl restart docker
```

3.1.2.2. 离线部署

#1、将 docker 部署包上传到系统中的目录, 例如/usr/local/src, 目录可自定义

```
cd /usr/local/src
```

#2、进入指定目录

```
cd /usr/local/src/docker
```

统信软件技术有限公司 ©版权所有

#4、执行脚本安装 docker 和 docker-compose

```
source install.sh
```

3.1.3.部署 Docker-compose

3.1.3.1.在线部署

1.下载最新版的 docker-compose 文件

```
sudo curl -L
```

```
https://github.com/docker/compose/releases/download/1.16.1/docker-compose-`uname`  
-s-`uname -m` -o /usr/local/bin/docker-compose
```

若是 github 访问太慢，可以用 daocloud 下载

```
sudo curl -L
```

```
https://get.daocloud.io/docker/compose/releases/download/1.25.1/docker-compose-`una`  
me -s-`uname -m` -o /usr/local/bin/docker-compose
```

添加可执行权限

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

测试安装结果

```
docker-compose --version
```

```
docker-compose version 1.16.1, build 1719ceb
```

3.1.4.部署编辑器服务

3.1.4.1.单机部署

注：默认情况下，`docker-compose.yml` 文件中定义的镜像地址是畅写在阿里云上的镜像仓库地址，如果网络环境可以访问公网，则直接在线部署即可，反之选择离线部署。

```
version: '3'
```

```
services:
```

```
  changxieoffice-documentserverkeep:
```

```
    container_name: changxieoffice-documentserverkeep
```

```
    image: registry.cn-beijing.aliyuncs.com/changxie/changxie:6.1.52
```

```
    environment:
```

```
      # postgres mysql dmdb oracle kingbase golden(v6.7.5)
```

```
      #- DB_TYPE=mysql
```

```
      #- DB_HOST=10.1.11.108
```

```
##- DB_NAME=changxieoffice
##- DB_USER=root
##- DB_PWD=1234@1
##- DB_PORT=3309
##- REDIS_SERVER_HOST=10.1.11.213
##- REDIS_SERVER_PORT=6379
##- REDIS_SERVER_PASS=qwe123
##- AMQP_TYPE=rabbitmq
##- AMQP_URI=amqp://wan:qwe123@10.1.11.213:5672
##- RABBITMQ_ENABLE=true
##- RABBITMQ_URL=amqp://guest:guest@localhost:5673
##- RABBITMQ_EXCHANGE_PUBSUB=ds.pubsub1
##- RABBITMQ_QUEUE_CONVERTTASK=ds.converttask1
##- RABBITMQ_QUEUE_CONVERTRESPONSE=ds.convertresponse1
##- RABBITMQ_EXCHANGE_CONVERTDEAD=ds.exchangeconvertdead1
##- RABBITMQ_QUEUE_CONVERTDEAD=ds.convertdead1
##- RABBITMQ_QUEUE_DELAYED=ds.delayed1
##- KAFKA_ENABLED = true
##- KAFKA_AUTOCONNECT = true
##- KAFKA_HOST = 192.168.100.80:9092
##- KAFKA_ENABLESASL = true
##- KAFKA_MECHANISM = plain
##- KAFKA_USERNAME = ecb
##- KAFKA_PASSWORD = E_b63a2A
##- KAFKA_CONVERTTASK = converttask
##- KAFKA_CONVERTRESPONSE = convertresponse
##- KAFKA_CONVERTDEAD = convertdead
##- KAFKA_DELAYED = delayed
##- KAFKA_TOPIC_PUBSUB = pubsub
##- KAFKA_CONSUMER_CONVERTTASK = gid_ecb_converttask
##- KAFKA_CONSUMER_CONVERTRESPONSE = gid_ecb_convertresponse
##- KAFKA_CONSUMER_TOPIC_PUBSUB = gid_ecb_pubsub_001
- EXPIRE_SESSION_IDLE=30m #长时间不操作踢人的时长(v6.7.5)
- EXPIRE_MAX_CHANGE_NUM=15000 #长时间不操作踢人的操作记录条数(v6.7.5)
- JWT_ENABLED=false
- JWT_SECRET=ds@changxieoffice
- JWT_IN_BODY=false
- POSTFILE_ENABLED=false
- POSTFILE_RETRY=3
- DELAY_SAVE=true
- AUTO_SAVE_SERVER=false
- TZ=Asia/Shanghai
```

```
stdin_open: true
restart: always
privileged: true
networks:
  - changxieofficekeep
volumes:
  - ./customfonts:/usr/share/fonts/truetype/custom
  - /vol/var/www/changxieoffice/Data:/var/www/changxieoffice/Data
  - /vol/appdata/var/log/changxieoffice:/var/log/changxieoffice
  -
/vol/appdata/var/lib/changxieoffice/documentserver/App_Data/cache/files:/var/lib/changxieoffice
/documentserver/App_Data/cache/files
  - /vol/appdata/var/lib/postgresql:/var/lib/postgresql
  - /vol/appdata/var/lib/rabbitmq:/var/lib/rabbitmq
  - /vol/appdata/var/lib/redis:/var/lib/redis
expose:
  - '80'
  - '443'
ports:
  - '80:80'
networks:
  changxieofficekeep:
    driver: 'bridge'
```

3.1.4.1.1.在线部署

进入指定目录, 然后需要上传或者创建 docker-compose.yml 配置文件, 如上配置。

```
cd /usr/local/src
```

启动服务

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

3.1.4.1.2.离线部署

注: 离线部署需要先将畅写人员发送的离线镜像包上传到服务器, 然后导入到本地, 再部署编辑器服务。离线镜像包的名称为 changxie5016.tar

例如, 将离线镜像包上传到了/usr/local/src 目录下, 执行如下命令导入镜像到本地

```
docker load -i /usr/local/src/changxie5016.tar
```

导入之后启动编辑器服务(需要上传或者创建 docker-compose.yml 配置文件)

```
cd /usr/local/src/
```

启动服务

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

3.1.4.1.3.验证服务是否正常

打开浏览器，输入 <http://yourip/welcome>

注：yourip 表示部署编辑器服务的服务器 IP

3.1.4.1.4.获取机器码

打开浏览器，输入 <http://yourip/info/register.html>

注：yourip 表示部署编辑器服务的服务器 IP

3.1.4.1.5.做授权激活

将上一步获取到的机器码给到畅写商务人员，然后商务人员会返回一个授权文件给您，在授权页面上传即可完成授权操作。

3.1.4.2.集群部署

注：集群部署与单机部署的操作步骤是一样的，服务器由单台扩容为 2 台或以上。由于集群环境涉及到数据同步、统一访问入口等问题，所以需要将依赖的组件都连接外部的。具体配置方式如下：

3.1.4.2.1.畅写编辑器依赖的组件说明

注：以下组件需自行部署成单机环境或集群环境（Redis 需要为单机或主从模式）

- 数据库：MySQL 5.7、PostgreSQL 9.5 及以上、oracle19c 、dmdb 7.6
- 缓存：Redis 4.0 以上
- 消息队列：rabbitmq 3.8、kafka 2.13 (支持 SASL 认证方式)
- 存储：NFS
- 负载均衡和反向代理：Nginx

3.1.4.2.2.docker-compose.yaml 中关于依赖组件的配置信息

```
version: '3'
```

```
services:
```

```
  changxieoffice-documentserverkeep:
```

```
    container_name: changxieoffice-documentserverkeep
```

```
    image: registry.cn-beijing.aliyuncs.com/changxie/changxie:6.1.52
```

environment:

```
# postgres mysql oracle dmdb golden(v6.7.5)
#- DB_TYPE=mysql
#- DB_HOST=10.1.11.108
#- DB_NAME=changxieoffice
#- DB_USER=root
#- DB_PWD=1234@1
#- DB_PORT=3309
#- REDIS_SERVER_HOST=10.1.11.213
#- REDIS_SERVER_PORT=6379
#- REDIS_SERVER_PASS=qwe123
#- AMQP_TYPE=rabbitmq
#- AMQP_URI=amqp://wan:qwe123@10.1.11.213:5672
#- RABBITMQ_ENABLE=true
#- RABBITMQ_URL=amqp://guest:guest@localhost:5673
#- RABBITMQ_EXCHANGE_PUBSUB=ds.pubsub1
#- RABBITMQ_QUEUE_CONVERTTASK=ds.converttask1
#- RABBITMQ_QUEUE_CONVERTRESPONSE=ds.convertresponse1
#- RABBITMQ_EXCHANGE_CONVERTDEAD=ds.exchangeconvertdead1
#- RABBITMQ_QUEUE_CONVERTDEAD=ds.convertdead1
#- RABBITMQ_QUEUE_DELAYED=ds.delayed1
#- KAFKA_ENABLED = true
#- KAFKA_AUTOCONNECT = true
#- KAFKA_HOST = 192.168.100.80:9092
#- KAFKA_ENABLESASL = true
#- KAFKA_MECHANISM = plain
#- KAFKA_USERNAME = ecb
#- KAFKA_PASSWORD = E_b63a2A
#- KAFKA_CONVERTTASK = converttask
#- KAFKA_CONVERTRESPONSE = convertresponse
#- KAFKA_CONVERTDEAD = convertdead
#- KAFKA_DELAYED = delayed
#- KAFKA_TOPIC_PUBSUB = pubsub
#- KAFKA_CONSUMER_CONVERTTASK = gid_ecb_converttask
#- KAFKA_CONSUMER_CONVERTRESPONSE = gid_ecb_convertresponse
#- KAFKA_CONSUMER_TOPIC_PUBSUB = gid_ecb_pubsub_001
- EXPIRE_SESSIONIDLE=30m #长时间不操作踢人的时长(v6.7.5)
- EXPIRE_MAXCHANGESNUM=15000 #长时间不操作踢人的操作记录条数
(v6.7.5)
- JWT_ENABLED=false
- JWT_SECRET=ds@changxieoffice
- JWT_IN_BODY=false
```

```
- POSTFILE_ENABLED=false
- POSTFILE_RETRY=3
- DELAY_SAVE=true
- AUTO_SAVE_SERVER=false
- TZ=Asia/Shanghai
stdin_open: true
restart: always
privileged: true
networks:
  - changxieofficekeep
volumes:
  - ./customfonts:/usr/share/fonts/truetype/custom
  - /vol/var/www/changxieoffice/Data:/var/www/changxieoffice/Data
  - /data/appdata/var/log/changxieoffice:/var/log/changxieoffice
  -
/vol/appdata/var/lib/changxieoffice/documentserver/App_Data/cache/files:/var/lib/chang
xieoffice/documentserver/App_Data/cache/files
  - /vol/appdata/var/lib/postgresql:/var/lib/postgresql
  - /vol/appdata/var/lib/rabbitmq:/var/lib/rabbitmq
  - /vol/appdata/var/lib/redis:/var/lib/redis
expose:
  - '80'
  - '443'
ports:
  - '80:80'
networks:
  changxieofficekeep:
    driver: 'bridge'
```

3.1.4.2.3.畅写编辑器依赖的组件说明

3.1.4.2.3.1.数据库

- DB_TYPE=mysql # 数据库类型, 可以是 mysql、postgresql、oracle、dmdb、kingbase、golden(v6.7.5)
- DB_HOST=10.1.11.108 # 数据库服务器地址
- DB_NAME=changxieoffice # 数据库名称
- DB_USER=root # 数据库用户
- DB_PWD=1234@1 # 数据库密码
- DB_PORT=3309 # 数据库端口

3.1.4.2.3.2.缓存

- REDIS_SERVER_HOST=10.1.11.213 # Redis 服务器地址
- REDIS_SERVER_PORT=6379 # Redis 服务端口
- REDIS_SERVER_PASS=qwe123 # Redis 服务密码

3.1.4.2.3.3.消息对列

rabbitmq

- AMQP_TYPE=rabbitmq # 消息队列类型
- AMQP_URI=amqp://wan:qwe123@10.1.11.213:5672 # 消息队列连接信息，其中 amqp 表示协议，wan 表示用户名，qwe123 表示密码，10.1.11.213 表示消息队列服务器地址，5672 表示消息队列服务端口

rabbitmq 细化

- RABBITMQ_ENABLE=true #开启 rabbitmq
- RABBITMQ_URL=amqp://guest:guest@localhost:5673 #RabbitMQ 服务器的 URL
- RABBITMQ_EXCHANGE_PUBSUB=ds.pubsub1 #发布/订阅模式的交换机名称
- RABBITMQ_QUEUE_CONVERTTASK=ds.converttask1 #用于转换任务的队列名称
- RABBITMQ_QUEUE_CONVERTRESPONSE=ds.convertresponse1 #用于转换响应的队列名称
- RABBITMQ_EXCHANGE_CONVERTDEAD=ds.exchangeconvertdead1 #死信交换机名称
- RABBITMQ_QUEUE_CONVERTDEAD=ds.convertdead1 #死信队列名称
- RABBITMQ_QUEUE_DELAYED=ds.delayed1 #延迟队列名称

注：rabbitmq 的用户密码中不允许带@符号，例如 123@abc 是不允许的

KAFKA

- KAFKA_ENABLED = true #开启 kafka
- KAFKA_AUTOCONNECT = true #自动连接

- KAFKA_HOST = 192.168.100.80:9092 #kafka Url
- KAFKA_ENABLESASL = true #启用 SASL 认证
- KAFKA_MECHANISM = plain #SASL 认证机制
- KAFKA_USERNAME = ecb #SASL 认证的用户名
- KAFKA_PASSWORD = E_b63a2A #SASL 认证的密码
- KAFKA_CONVERTTASK = converttask #转换任务的名称
- KAFKA_CONVERTRESPONSE = convertresponse #转换响应的名称
- KAFKA_CONVERTDEAD = convertdead #死信主题的名称
- KAFKA_DELAYED = delayed #延迟处理主题的名称
- KAFKA_TOPICPUBSUB = pubsub #发布/订阅模式的名称
- KAFKA_CONSUMERCONVERTTASK = gid_ecb_converttask #消费者用于处理转换任务的组 ID
- KAFKA_CONSUMERCONVERTRESPONSE = gid_ecb_convertresponse #消费者用于处理转换响应的组 ID
- KAFKA_CONSUMERTOPICPUBSUB = gid_ecb_pubsub_001 #消费者用于处理发布/订阅模式消息的组 ID

3.1.4.2.3.4.存储 (NFS)

注: 挂载文件共享服务器的配置需要在每台部署了畅写编辑器服务的服务器上都要做

用文本编辑工具打开/etc/fstab 配置文件, 添加以下配置

NFS 服务器地址:NFS 共享目录 编辑器服务所在服务器上的挂载点 nfs defaults,_net dev 0 0

执行以下命令使上面的配置生效

```
mount -a
```

3.1.4.2.3.5.负载均衡和反向代理 (Nginx)

```
http {  
    upstream changxieoffice-document-server {  
        server app1:80;  
        server app2:80;
```

```
...
}
server {
    server_name IP;
    listen IP:POST;
    location /doc { #该配置为 websocket 配置, 该 URI 不能改
        proxy_pass http://changxieoffice-document-server;
        proxy_redirect off;
        client_max_body_size 100m;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_set_header Host $http_host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Host $host/ds-vpath; #如果编辑器服
务本身的 URI 未配置二级目录, 只是/的话, 这里也不需要带二级目录
        proxy_set_header X-Forwarded-Proto $scheme;
    }

    location ~* ^/ds-vpath { #该配置为畅写编辑器服务本身配置
        proxy_pass http://changxieoffice-document-server;
    }
}
}
```

3.2.K8s 部署

K8s 部署参考配置文件 changxie.yaml:

```
---
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: editor
  namespace: default
  labels:
    namespace: default
    deployment: editor
spec:
  #replicas: 2
  selector:
    matchLabels:
```

```
namespace: default
deployment: editor
template:
  metadata:
    labels:
      namespace: default
      deployment: editor
  spec:
    affinity:
      nodeAffinity:
        requiredDuringSchedulingIgnoredDuringExecution:
          nodeSelectorTerms:
            - matchExpressions:
                - key: kubernetes.io/hostname
                  operator: In
                  values:
                    - work3
                    - work2
      #imagePullSecrets:
      #- name: harbor-secret
    volumes:
      - name: editor-files
        persistentVolumeClaim:
          #claimName: longhorn-editor-pvc
          claimName: nfs-editor-files
      - name: editor-log
        persistentVolumeClaim:
          #claimName: longhorn-editor-log
          claimName: nfs-editor-log
      - name: editor-config
        persistentVolumeClaim:
          #claimName: longhorn-editor-config
          claimName: nfs-editor-config
    containers:
      - name: editor
        image: registry.cn-beijing.aliyuncs.com/changxie/changxie:6.1.52
        env:
          - name: DB_TYPE
            value: "mysql"
          - name: DB_HOST
            value: "mysql-mysqlha-0.mysql-mysqlha"
          - name: DB_NAME
```

```
    value: "changxieoffice"
- name: DB_USER
  value: "root"
- name: DB_PWD
  value: "changxieoffice"
- name: DB_PORT
  value: "3306"
- name: REDIS_SERVER_HOST
  value: "redis-master"
- name: REDIS_SERVER_PORT
  value: "6379"
- name: REDIS_SERVER_PASS
  value: "changxieoffice"
- name: AMQP_TYPE
  value: "rabbitmq"
- name: AMQP_URI
  value: "amqp://guest:guest@rabbitmq-rabbitmq-ha"
- name: RABBITMQ_ENABLE
  value: "true"
- name: RABBITMQ_URL
  value: "amqp://guest:guest@localhost:5673"
- name: RABBITMQ_EXCHANGE_PUBSUB
  value: "ds.pubsub1"
- name: RABBITMQ_QUEUE_CONVERTTASK
  value: "ds.converttask1"
- name: RABBITMQ_QUEUE_CONVERTRESPONSE
  value: "ds.convertresponse1"
- name: RABBITMQ_EXCHANGE_CONVERTDEAD
  value: "ds.exchangeconvertdead1"
- name: RABBITMQ_QUEUE_CONVERTDEAD
  value: "ds.convertdead1"
- name: RABBITMQ_QUEUE_DELAYED
  value: "ds.delayed1"
- name: KAFKA_ENABLED
  value: "true"
- name: KAFKA_AUTOCONNECT
  value: "true"
- name: KAFKA_HOST
  value: "192.168.100.80:9092"
- name: KAFKA_ENABLESASL
  value: "true"
- name: KAFKA_MECHANISM
```

```
value: "plain"
- name: KAFKA_USERNAME
value: "ecb"
- name: KAFKA_PASSWORD
value: "E_b63a2A"
- name: KAFKA_CONVERTTASK
value: "converttask"
- name: KAFKA_CONVERTRESPONSE
value: "convertresponse"
- name: KAFKA_CONVERTDEAD
value: "convertdead"
- name: KAFKA_DELAYED
value: "delayed"
- name: KAFKA_TOPICPUBSUB
value: "pubsub"
- name: KAFKA_CONSUMERCONVERTTASK
value: "gid_ecb_converttask"
- name: KAFKA_CONSUMERCONVERTRESPONSE
value: "gid_ecb_convertresponse"
- name: KAFKA_CONSUMERTOPICPUBSUB
value: "gid_ecb_pubsub_001"
- name: EXPIRE_SESSIONIDLE
value: "30m"
- name: EXPIRE_MAXCHANGESNUM
value: "15000"
- name: JWT_ENABLED
value: "false"
- name: JWT_SECRET
value: "ds@changxieoffice"
- name: POSTFILE_ENABLED
value: "false"
- name: POSTFILE_RETRY
value: "3"
- name: DELAY_SAVE
value: "true"
- name: AUTO_SAVE_SERVER
value: "false"
- name: TZ
value: "Asia/Shanghai"
ports:
- name: liveness-port
containerPort: 80
```

```
#livenessProbe:
# httpGet:
#   path: /healthcheck
#   port: liveness-port
#   initialDelaySeconds: 65
#   periodSeconds: 5
#readinessProbe:
# httpGet:
#   path: /healthcheck
#   port: liveness-port
#   initialDelaySeconds: 65
#   periodSeconds: 5
#securityContext:
# privileged: true
volumeMounts:
#- name: editor-pvc
#   mountPath: /var/www/changxieoffice/Data
#- name: editor-log
#   mountPath: /var/log/changxieoffice/documentserver
- name: editor-files
  mountPath: /var/lib/changxieoffice
#- name: editor-config
#   mountPath: /etc/changxieoffice/documentserver
---
```

```
apiVersion: v1
kind: Service
metadata:
  labels:
    namespace: default
    deployment: editor
  name: editorsvc
  namespace: default
spec:
  type: NodePort
  ports:
    - port: 80
      protocol: TCP
      targetPort: 80
      nodePort: 30090
  selector:
    namespace: default
    deployment: editor
```

3.3.原生部署

使用原生部署强烈建议使用 root 用户，使用非 root 用户需要给 `/sys/class/dmi/id/` 目录只读权限。

```
sudo chmod 666 /sys/class/dmi/id/
```

3.3.1.关闭防火墙或者开放端口

关闭防火墙

```
systemctl stop firewalld
systemctl disable firewalld
```

或者开放端口

添加

```
firewall-cmd --zone=public --add-port=80/tcp --permanent    (--permanent 永久生效，没有此参数重启后失效 端口根据实际情况修改，默认编辑器端口)
```

```
firewall-cmd --zone=public --add-port=3000/tcp --permanent    (--permanent 永久生效，没有此参数重启后失效 端口根据实际情况修改，默认演示示例端口，如不开启可不操作)
```

重新载入

```
firewall-cmd --reload
```

查看

```
firewall-cmd --zone=public --query-port=80/tcp    //端口根据实际情况修改
firewall-cmd --zone=public --query-port=3000/tcp    //端口根据实际情况修改，如不开启可不操作
```

3.3.2.关闭 selinux

```
# setenforce 0
# sed -i 's/^SELINUX=enforcing$/SELINUX=disabled/' /etc/selinux/config
```

3.3.3.解压部署包

```
cd /home
mkdir -p ecb/work
```

将部署包 documentserver.tar.gz 解压

```
tar zxvf documentserver.tar.gz
```

3.3.4.配置环境变量

```
vim /etc/profile 或者 vim ~/.bash_profile (非 root 用户)
```

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/ecb/work/documentserver/var/www/changxieoffice/documentserver/server/FileConverter/bin
```

```
source /etc/profile 或者 source ~/.bash_profile
```

3.3.5.修改配置文件

```
cd /home/ecb/work/documentserver
```

```
vim etc/changxieoffice/documentserver/default.json
```

```
"queue": {
    "type": "rabbitmq", // 修改消息使用的消息队列 rabbitmq 或者 kafkamq
    "visibilityTimeout": 300,
    "retentionPeriod": 900
},
###如果使用 rabbitmq 修改如下
"rabbitmq": {
    "url": "amqp://guest:guest@localhost:5672", //修改 rabbitmq 对应的用户、
    密码及 ip、端口
    "socketOptions": {},
    "exchangepubsub": "ds.pubsub",
    "queueconverttask": "ds.converttask",
    "queueconvertresponse": "ds.convertresponse",
    "exchangeconvertdead": "ds.exchangeconvertdead",
    "queueconvertdead": "ds.convertdead",
    "queuedelayed": "ds.delayed"
},
###如果使用 kafka 修改如下
"kafkamq": {
    "connectOptions": {
        "autoConnect": true,
        //kafka 的 ip 和端口
        "kafkaHost": "192.168.100.80:9092,192.168.100.81:9092,19
2.168.100.83:9092"
    },
    "enableSasl":false, //是否开启验证 false 关闭, true 开启
    "sasl":{
```



```

        "mechanism": "plain",
        "username": "ecb",
        "password": "E_b63a2A"
    },
    "converttask": "converttask",
    "convertresponse": "convertresponse",
    "convertdead": "convertdead",
    "delayed": "delayed",
    "topicpubsub": "pubsub",
    "consumerconverttask": "gid_ecb_converttask",
    "consumerconvertresponse": "gid_ecb_convertresponse",
    "consumertopicpubsub": "gid_ecb_pubsub_001"
},

```

###配置编辑器启动端口

```
server": {
```

未被占用及端口开放了

```

        "port": 80, //修改成需要的端口号, 请确保端口
        "workerpercpu": 1,
        "mode": "development",
        "limits_tempfile_upload": 1048576000,
        "limits_image_size": 26214400,
        "limits_image_download_timeout": {
            "connectionAndInactivity": "10s",
            "wholeCycle": "2m"
        },
        "callbackRequestTimeout": {
            "wholeCycle": "2m"
        },
    },

```

3.3.5.1.创建数据库

安装包目录 documentserver/var/www/changxieoffice/documentserver/server/schema/ 下存放了 mysql、postgres、oracle sql 语句。使用对应的数据库 sql 语句创建数据库。

以 mysql 为例:

登录 mysql

```
mysql -u root -p
```

```
CREATE DATABASE IF NOT EXISTS changxieoffice DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
```

```
CREATE USER 'changxieoffice'@'localhost' IDENTIFIED BY 'changxieoffice'; #创建用户
CREATE USER 'changxieoffice'@'127.0.0.1' IDENTIFIED BY 'changxieoffice';
GRANT ALL PRIVILEGES ON changxieoffice.* TO changxieoffice@'localhost' IDENTIFIED BY 'changxieoffice' WITH GRANT OPTION; #授权用户
GRANT ALL PRIVILEGES ON changxieoffice.* TO changxieoffice@'127.0.0.1' IDENTIFIED BY 'changxieoffice' WITH GRANT OPTION; #授权用户
```

#MySQL5.7 版本中如果对用户授权后只能连接数据库并无任何操作权限，请设置用户对所有数据库有 SUPER 权限

```
grant SUPER on *.* to 'changxieoffice'@'localhost' ; #授予用户对所有数据库有 SUPER 权限
grant SUPER on *.* to 'changxieoffice'@'127.0.0.1' ;
flush privileges; #刷新系统授权表
mysql -q -h127.0.0.1 -P3306 -uchangxieoffice -pchangxieoffice -w changxieoffice < "/home/ecb/work/documentserver/var/www/changxieoffice/documentserver/server/schema/mysql/createdb.sql"
```

至此，MySQL5.7 版本下创建数据库添加用户并对用户进行授权完成

3.3.5.2.数据库配置

```
"sql": {
    "type": "postgres", // 填写 mysql、postgres、oracle、kingbase、dmdb (达梦数据库)、golden(金篆 V6.7.5)之一
    "tableChanges": "doc_changes",
    "tableResult": "task_result",
    "dbHost": "localhost",
    "dbPort": 5432, //数据库端口 postgres 默认 5432 mysql 默认 3306 oracle 默认 1521
    "dbName": "changxieoffice", //数据库名称
    "dbUser": "changxieoffice", //数据库的用户名
    "dbPass": "changxieoffice", //数据库的密码
    "charset": "utf8",
    "connectionlimit": 20, //连接数
    "max_allowed_packet": 60000,
    "pgPoolExtraOptions": {}
}
```

3.3.5.3.redis 配置

```

"redis": {
    "mode": "single", //single (单机) 、sentinel (哨兵) 、cluster (集群)
    "name": "redis",
    "prefix": "ds:",
    "host": "localhost",单机的 ip
    "port": 6379, //单机的端口
    "password": "", //单机的密码
    "options": {},
    "sentinel": {
        "name": "mymaster",
        "prefix": "ds:",
        "password": "", //哨兵密码
        "db": 0,
        "family": 4,
        "slave": [
            ],
        "sentinels": [
            {"host": "127.0.0.1", "port": 26482, "password": ""}, //sentinel 的 ip、端口及密
            {"host": "127.0.0.1", "port": 26480, "password": ""},
            {"host": "127.0.0.1", "port": 26481, "password": ""}
        ]
    },
    "cluster": {
        "masters": [
            {"host": "127.0.0.1", "port": 7000, "password": ""},
            {"host": "127.0.0.1", "port": 7001, "password": ""},
            {"host": "127.0.0.1", "port": 7002, "password": ""}
        ]
    }
}

```

3.3.5.4.配置缓存本机目录

```

"usenas": {
    "enable" : false //false 使用 http,true 使用本机目录
},

```

prod-linux.json 文件指定缓存目录

```

"storage": {
    "fs": {
        "folderPath": "../files/documentserver/App_Data"
    }
}

```

```
    }  
  }  
  
  "FileStorage": {  
    "directory": "../../files/documentserver/App_Data"  
  },  
}
```

在 FileStorage.directory 指定目录下创建 docbuilder 目录。如:

```
mkdir -p ../../files/documentserver/App_Data/docbuilder
```

3.3.5.5.配置日志

修改../../etc/changxieoffice/documentserver/log4js/prod.json

```
{  
  "appenders": {  
    "default": {  
      "type": "file",  
      "filename": "../../var/log/changxieoffice/documentserver/app.log",//日志存  
      放位置  
      "maxLogSize":1024000,  
      "backups":3,  
      "layout": {  
        "type": "pattern",  
        "pattern": "[%d] [%p] %c - %.10000m"  
      }  
    }  
  },  
  "categories": {  
    "default": {  
      "appenders": [  
        "default"  
      ],  
      "level": "ALL"  
    }  
  }  
}
```

3.3.6.配置运行环境

进入到指定目录

```
cd /home/ecb/work/documentserver/usr/bin
```

生成字体

```
sh ./generate-allfonts.sh
```

启动编辑器服务

```
sh ./start.sh
```

停止编辑器服务

```
sh ./stop.sh
```

3.3.7.授权页面

访问 <http://部署编辑器服务的 ip:端口/info/register.html>,将授权码发给畅写工作人员

3.3.8.演示程序配置

修改 `vim ../../etc/changxieoffice/documentserver-example/prod-linux.json`

```
{
  "server": {
    "siteUrl": "http://192.168.100.68:80", //编辑器 ip 和端口.
    "maxFileSize": 104857600,
    "storageFolder": "../../files/documentserver-example/files", //演示程序文档保存位置
    "static":[
      {
        "name": "/files",
        "path": "../../files/documentserver-example/files"
      }
    ]
  }
}
```

启动演示程序

```
sh startdemo.sh
```

停止演示程序

```
sh ./stopdemo.sh
```

3.3.9.演示程序访问

访问 <http://部署编辑器服务的 ip:3000/> 3000 是默认访问端口

documentserver 修改成畅写在线部署的服务器 ip 或者域名

4. 授权页面

4.1. 授权页面

<http://documentserver/info/register.html>

取到的机器码给到畅写商务人员，然后商务人员会返回一个授权文件给您，在授权页面上上传即可完成授权操作。



4.2. 演示示例

