阿凡搭企业数字化底座平台

部署手册

深圳市非凡信息技术有限公司

本文档若未及时更新或对软件说明有不完整之处，则以实际交付的软件产品所具备的功能为准。本文档的版权和最终解释权归深圳市非凡信息技术有限公司所有。

目录

1 部署架构 4
1.1 单服务器架构 4
1.1.1 部署架构图 4
1.1.2 服务器清单 5
1.2 多服务器架构 5
1.2.1 部署架构图 6
1.2.2 服务器清单 6
1.3 软件负载均衡架构 7
1.3.1 部署架构图 8
1.3.2 服务器清单 8
1.4 硬件负载均衡架构 9
1.4.1 部署架构图 10
1.4.2 服务器清单 10
2 部署方式 11
2.1 私有化本地部署 11
2.1.1 服务器目录结构（建议） 11
2.1.2 应用组件安装 12
2.1.3 平台部署 27
2.1.4 在线部署 ..... **错误！未定义书签。**
2.1.5 离线部署 ..... **错误！未定义书签。**

3 系统初始化 ..... 错误！未定义书签。

3.1 设置系统超级管理员 ..... 错误！未定义书签。

3.2 创建组织机构及用户 ..... 错误！未定义书签。

3.3 创建系统功能模块 ..... 错误！未定义书签。

3.4 测试应用转生产环境 ..... 错误！未定义书签。

1 部署架构

1.1 单服务器架构

阿凡搭平台单服务器部署架构模式是将平台所有的应用程序、组件和服务都部署在同一个服务器上。服务器上需完成以下工作：

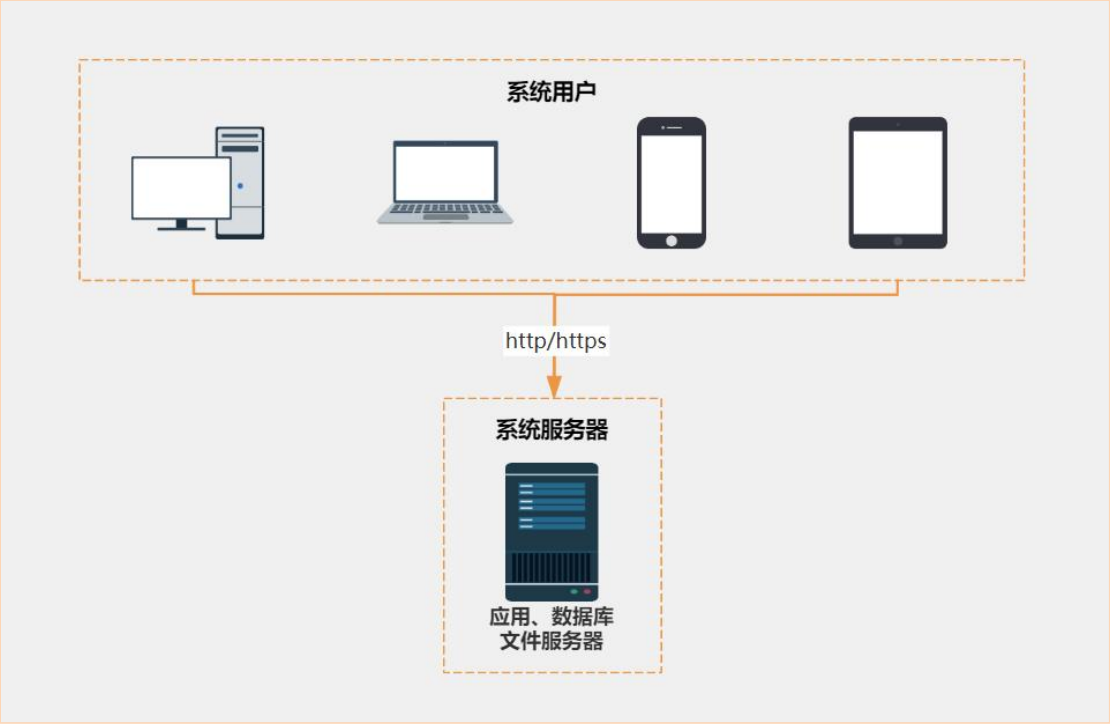
安装阿凡搭平台部署所需的所有组件，包括数据库和文件存储服务；

部署阿凡搭平台引擎服务；

部署阿凡搭平台 PC 端应用前后端；

部署阿凡搭平台移动端应用前后端；

1.1.1 部署架构图



1.1.2 服务器清单

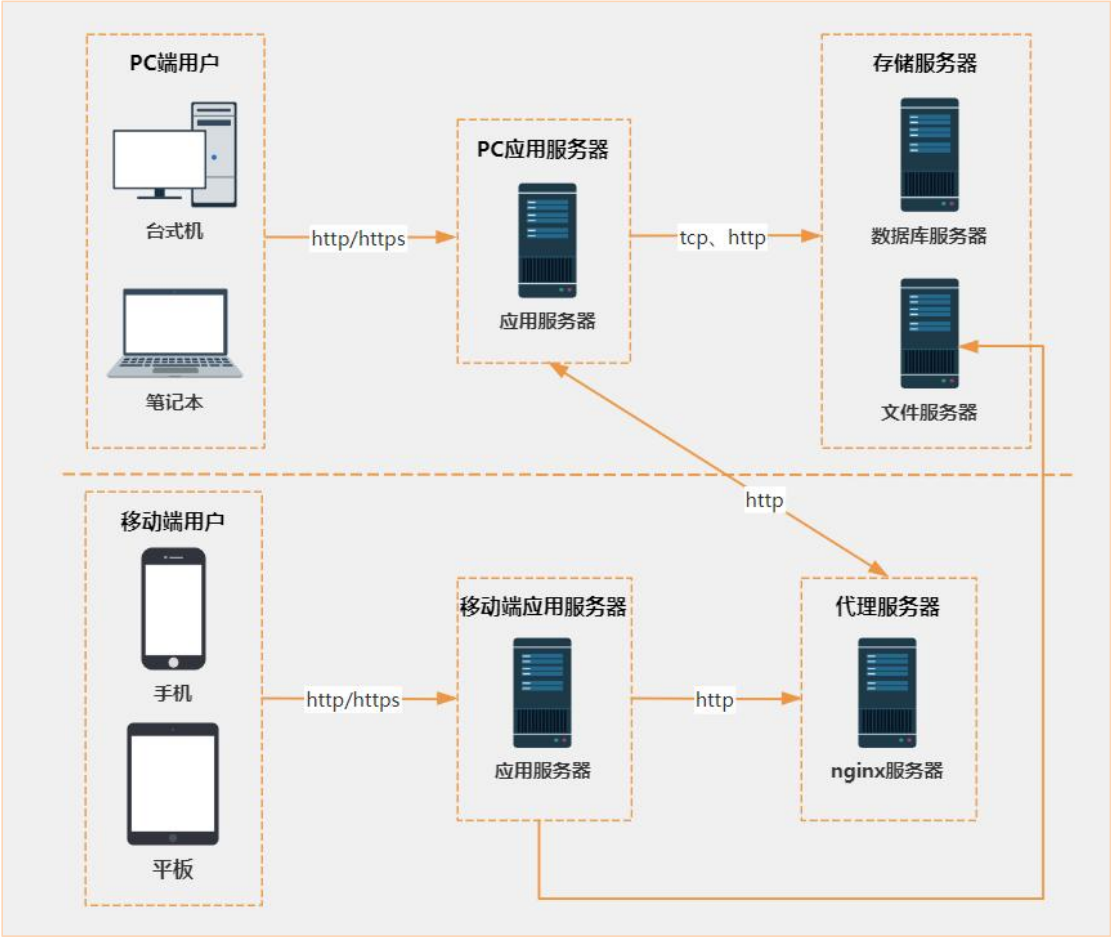
服务器名称	操作系统	建议配置		
		CPU	内存	存储
阿凡搭平台服务器	CentOS7+/Redhat8+/银河麒麟 v10	16C	32G	1T

1.2 多服务器架构

阿凡搭平台多服务器部署架构模式是指将平台应用程序、数据及文件存储分不同的服务器来部署。服务器上需执行以下操作：

- PC 端应用服务器上部署阿凡搭平台 PC 端应用前后端和引擎服务；
- 移动端应用服务器上需部署阿凡搭平台移动端应用前后端；
- 数据库服务器上安装关系数据库 mysql 并存储阿凡搭平台数据；
- 文件服务器上安装文件存储服务并存储阿凡搭平台的文件；

1.2.1 部署架构图



1.2.2 服务器清单

服务器名称	操作系统	建议配置		
		CPU	内存	存储
PC 应用服务器	CentOS7+/Redhat8+/银河麒麟 v10	8C	16G	200G
数据库服务器	CentOS7+/Redhat8+/银河麒麟 v10	8C	16G	500G
文件服务器	CentOS7+/Redhat8+/银河麒麟 v10	8C	16G	500G
移动端应用服务器	CentOS7+/Redhat8+/银河麒麟 v10	8C	16G	200G
代理服务器	CentOS7+/Redhat8+/银河麒麟 v10	4C	8G	100G

说明：

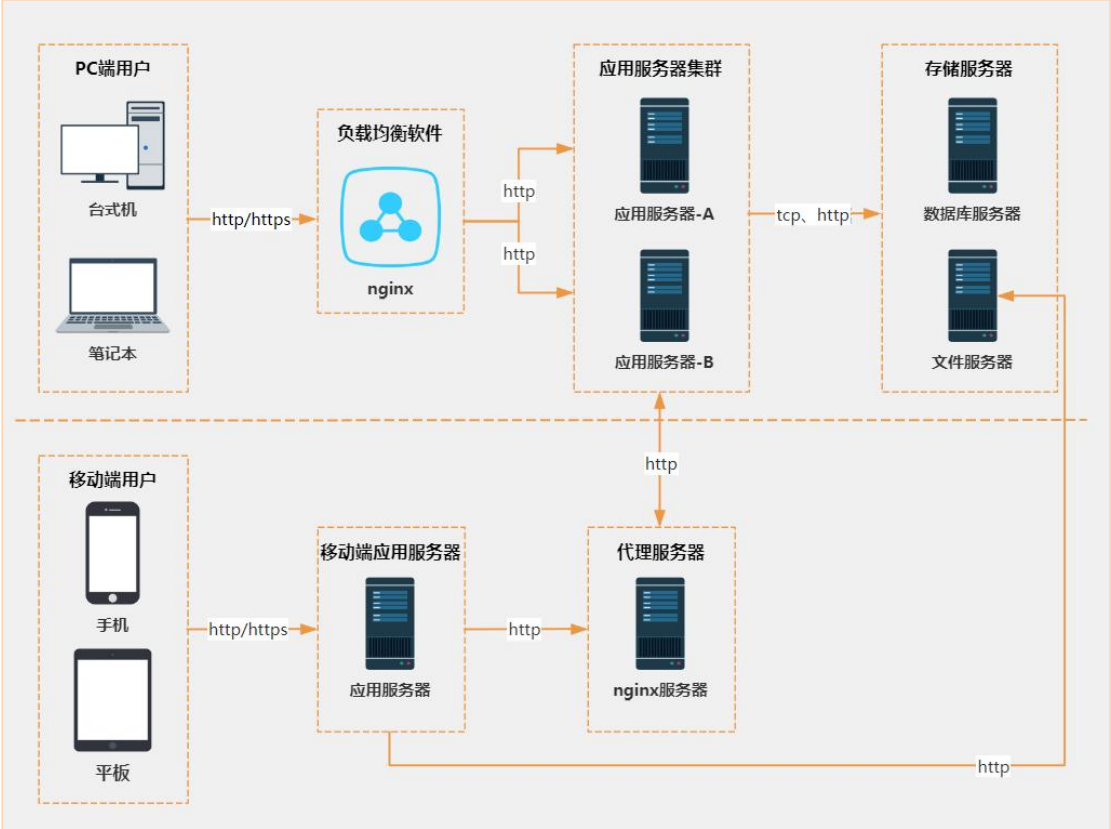
1. 文件服务器为非必须配备，根据实际情况配备，文件存储服务可安装在应用服务器或数据库服务器上。
2. 移动端应用服务器的配备是根据是否有移动端实施需求确定的，移动端应用服务器需开通外网访问策略，如果移动端接入的是微信公众号则需开通 80 端口；
3. 代理服务器为非必须配备，代理服务器的作用是：当企业有内外网络隔离的策略时，代理服务器可用来建立内外网络之间的通信，移动端与 PC 端之间的交互及存取数据。

1.3 软件负载均衡架构

阿凡搭平台软件负载均衡架构模式是指将平台应用程序、数据及文件存储分不同的服务器来部署，部署多套应用程序在不同应用服务器上，应用服务器支持弹性扩展，由 nginx 软件来实现负载均衡。服务器上需执行以下操作：

- PC 端应用服务器上部署阿凡搭平台 PC 端应用前后端和引擎服务；
- 移动端应用服务器上需部署阿凡搭平台移动端应用前后端；
- 数据库服务器上安装关系数据库 mysql 并存储阿凡搭平台数据；
- 文件服务器上安装文件存储服务并存储阿凡搭平台的文件；

1.3.1 部署架构图



1.3.2 服务器清单

服务器名称	操作系统	建议配置		
		CPU	内存	存储
负载均衡服务器	CentOS7+/Redhat8+/银河麒麟 v10	8C	16G	200G
应用服务器-A	CentOS7+/Redhat8+/银河麒麟 v10	8C	16G	200G
应用服务器-B	CentOS7+/Redhat8+/银河麒麟 v10	8C	16G	200G
数据库服务器	CentOS7+/Redhat8+/银河麒麟 v10	8C	16G	500G
文件服务器	CentOS7+/Redhat8+/银河麒麟 v10	8C	16G	500G
移动端应用服务器	CentOS7+/Redhat8+/银河麒麟 v10	8C	16G	200G
代理服务器	CentOS7+/Redhat8+/银河麒麟 v10	4C	8G	100G

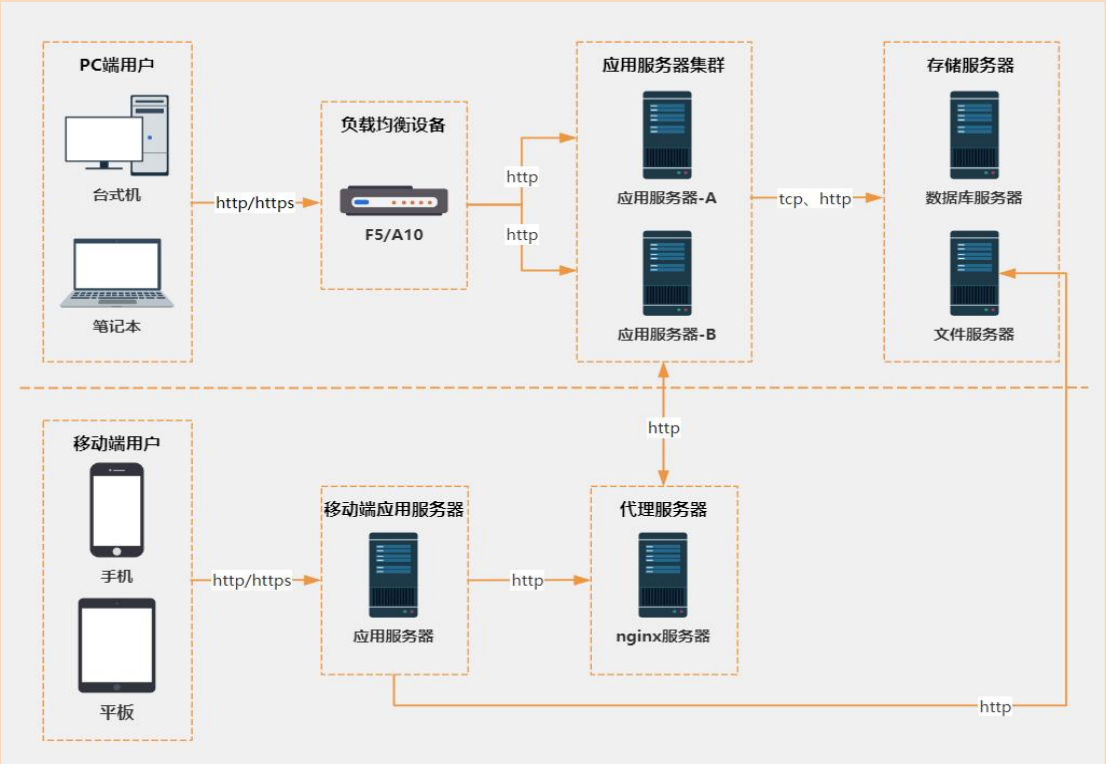
1. 负载均衡服务器为非必须配备，负载均衡服务器主要是用来安装负载均衡软件 nginx，负载均衡软件可安装在应用服务器上；
2. 应用服务器数量至少两台，可支持弹性扩展；
3. 文件服务器为非必须配备，根据实际情况配备，文件存储服务可安装在应用服务器或数据库服务器上；
4. 移动端应用服务器的配备是根据是否有移动端实施需求确定的，移动端应用服务器需开通外网访问策略，如果移动端接入的是微信公众号则需开通 80 端口；
5. 代理服务器为非必须配备，代理服务器的作用是：当企业有内外网络隔离的策略时，代理服务器可用来建立内外网络之间的通信，移动端与 PC 端之间的交互及存取数据。

1.4 硬件负载均衡架构

阿凡搭平台硬件负载均衡架构模式是指将平台应用程序、数据及文件存储分不同的服务器来部署，部署多套应用程序在不同应用服务器上，应用服务器支持弹性扩展，由负载均衡设备 F5 或 A10 来实现负载均衡。服务器上需执行以下操作：

- PC 端应用服务器上部署阿凡搭平台 PC 端应用前后端和引擎服务；
- 移动端应用服务器上需部署阿凡搭平台移动端应用前后端；
- 数据库服务器上安装关系数据库 mysql 并存储阿凡搭平台数据；
- 文件服务器上安装文件存储服务并存储阿凡搭平台的文件；

1.4.1 部署架构图



1.4.2 服务器清单

服务器名称	操作系统	建议配置		
		CPU	内存	存储
应用服务器-A	CentOS7+/Redhat8+/银河麒麟 v10	8C	16G	200G
应用服务器-B	CentOS7+/Redhat8+/银河麒麟 v10	8C	16G	200G
数据库服务器	CentOS7+/Redhat8+/银河麒麟 v10	8C	16G	500G
文件服务器	CentOS7+/Redhat8+/银河麒麟 v10	8C	16G	500G
移动端应用服务器	CentOS7+/Redhat8+/银河麒麟 v10	8C	16G	200G
代理服务器	CentOS7+/Redhat8+/银河麒麟 v10	4C	8G	100G

说明：

1. 应用服务器数量至少两台，可支持弹性扩展；

2. 文件服务器为非必须配备，根据实际情况配备，文件存储服务可安装在应用服务器或数据库服务器上；
3. 移动端应用服务器的配备是根据是否有移动端实施需求确定的，移动端应用服务器需开通外网访问策略，如果移动端接入的是微信公众号则需开通 80 端口；
4. 代理服务器为非必须配备，代理服务器的作用是：当企业有内外网络隔离的策略时，代理服务器可用来建立内外网络之间的通信，移动端与 PC 端之间的交互及存取数据。

2 部署方式

阿凡搭平台可基于两种方式部署：本地服务器直接部署和本地服务器容器化部署。

2.1 私有化本地部署

2.1.1 服务器目录结构（建议）

一级目录	二级目录	三级目录	四级目录
home	ffapaas	web	
		back	jdk
			tomcat
		engine	jdk
			tomcat
		upload	
	mobile	web	
		back	jdk
			tomcat
	software		

2.1.2 应用组件安装

将部署阿凡搭平台所需的所有应用组件、数据库和文件存储服务安装包上传到服务器/home/software 目录下。

2.1.2.1 组件清单

资源类型	资源名称	版本	说明	资源包或下载地址
运行环境	JDK	1.8	jdk-8u201-linux-x64.tar.gz	链接：https://pan.baidu.com/s/1ckNXAjLU6L-meFHI7J6iDg?pwd=o28a
中间件	Tomcat	8.5	apache-tomcat-8.5.57.tar.gz	
Web 服务器	Nginx	1.18	nginx-1.18.0.tar.gz	
缓存存储数据库	Redis	6.26	redis-6.2.6.tar.gz	
文件存储服务	Minio	6.0	minio	
附件在线预览	KkFileView	4.3.0	kkFileView-4.3.0.tar.gz	
			LibreOffice_7.5.3.2_Linux_x86-64_rpm.tar.gz	
			fonts.zip	
			font_rpm.zip	
文件转换	unoconv	0.8.2	unoconv-0.8.2.zip	

关系数据库	MySQL	8.0.28	mysql-8.0.28-el7-x86_64.tar.gz	

2.1.2.2 安装 JDK

1. 在 software 目录下解压 JDK 压缩包并重命名；

   目录：cd /home/software

   命令：tar -zxvf jdk-8u201-linux-x64.tar.gz

   命令：mv jdk1.8.0_201 jdk

2. 配置 JDK 环境变量，在 profile 文件中文本最后添加以下信息:

   export JAVA_HOME=/home/software/jdk

   export JRE_HOME=${JAVA_HOME}/jre

   export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib

   export PATH=${JAVA_HOME}/bin:$PATH

   命令：vim /etc/profile

```
67          if [ "${-#*i}" != "$-" ]; then
68              . "$i"
69          else
70              . "$i" >/dev/null
71          fi
72      fi
73  done
74
75  unset i
76  unset -f pathmunge
77
78  export JAVA_HOME=/home/software/jdk
79  export JRE_HOME=${JAVA_HOME}/jre
80  export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib
81  export PATH=${JAVA_HOME}/bin:$PATH
82
```

3. 保存后执行命令让 profile 文件立马生效

   命令：source /etc/profile

4. 查看安装的版本

命令：java -version

显示版本号信息则表示安装成功。

```
[root@localhost ~]# java -version
java version "1.8.0_201"
Java(TM) SE Runtime Environment (build 1.8.0_201-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.201-b09, mixed mode)
[root@localhost ~]#
```

2.1.2.3 安装 Nginx

1. 首先安装依赖包

命令：yum -y install gcc zlib zlib-devel pcre-devel openssl openssl-devel

出现选择时，输入 y，一直等待安装完成

2. 在 software 目录下解压 nginx 压缩包

目录：cd /home/software

命令：tar -zxvf nginx-1.18.0.tar.gz

3. 在 nginx 目录下创建日志目录

目录：cd /home/software/nginx-1.18.0

命令：mkdir logs

4. 进入 nginx 目录，执行安装

目录：cd /home/software/nginx-1.18.0

命令：./configure --prefix=/home/software/nginx-1.18.0

--with-http_stub_status_module --with-http_ssl_module

5. 在 nginx 目录执行编译命令

目录：cd /home/software/nginx-1.18.0

命令：make

6. 在 nginx 目录执行安装命令

目录：cd /home/software/nginx-1.18.0

命令：make install

7. 在 nginx 目录 sbin 目录下执行启动命令

目录：cd /home/software/nginx-1.18.0/sbin

命令：./nginx -c /home/software/nginx-1.18.0/conf/nginx.conf

8. 浏览器访问 IP:80，验证 nginx 安装和启动状态

出现 Welcome to nginx! 表示启动成功

修改 nginx 配置文件参考如下：

nginx.conf

查看/修改配置命令：vim /home/software/nginx-1.18.0/conf/nginx.conf

重启 nginx 命令：/home/software/nginx-1.18.0/sbin/nginx -s reload -c /home/software/nginx-1.18.0/conf/nginx.conf

2.1.2.4 安装 Redis

1. 在 softwware 目录下解压 Redis 压缩包

目录：cd /home/software

命令：tar -zxvf redis-6.2.6.tar.gz

2. 修改 redis 相关配置

目录：cd /home/software/redis-6.2.6

命令：vim redis.conf

1) 将 redis.conf 文件中 daemonize 的值修改为 yes;

2) 将 redis.conf 文件中 requirepass 前面的注释去掉，并将 redis 密码值改为：123456

/+关键字 ，回车即可。

此为从文档当前位置向下查找关键字，按 n 键查找关键字下一个位置；

?+关键字，回车即可。

此为从文档当前位置向上查找关键字，按 n 键向上查找关键字；

3. 在 redis 目录下执行编译

目录：cd /home/software/redis-6.2.6

命令：make

4. 在 redis 目录下执行启动 redis 命令

目录：cd /home/software/redis-6.2.6

命令：./src/redis-server /home/software/redis-6.2.6/redis.conf

默认情况下，redis 只允许本机访问。

如果需要外部访问，需要修改配置文件。

修改配置文件 redis.conf：

修改第 70 行，将 bind 127.0.0.1 注释

修改第 90 行，将 protected-mode 修改为 no

2.1.2.5 安装 Minio

1. 创建 minio 文件夹，并在 minio 文件夹下上传 minio 文件

目录：cd /home/software

命令：mkdir minio

2. 创建文件存放目录和日志存放目录

目录：cd /home/software/minio

命令：mkdir minio_data

命令：mkdir minio_log

3. 创建 minio 启动脚本

目录：cd /home/software/minio

命令：vim minio_start.sh

复制以下脚本代码作为 minio_start.sh 的内容：

#!/bin/sh

echo "minio 开始启动....."

export MINIO_ROOT_USER=minioadmin

export MINIO_ROOT_PASSWORD=minioadmin

./minio server --address 0.0.0.0:9000 --console-address 0.0.0.0:9001 /home/software/minio/minio_data > /home/software/minio/minio_log/minio.log 2>&1 &

echo "minio 启动成功....."

4. 授权 minio 及 minio_start 启动脚本

目录：cd /home/software/minio

命令：chmod +x minio

命令：chmod +x minio_start.sh

5. 启动 minio

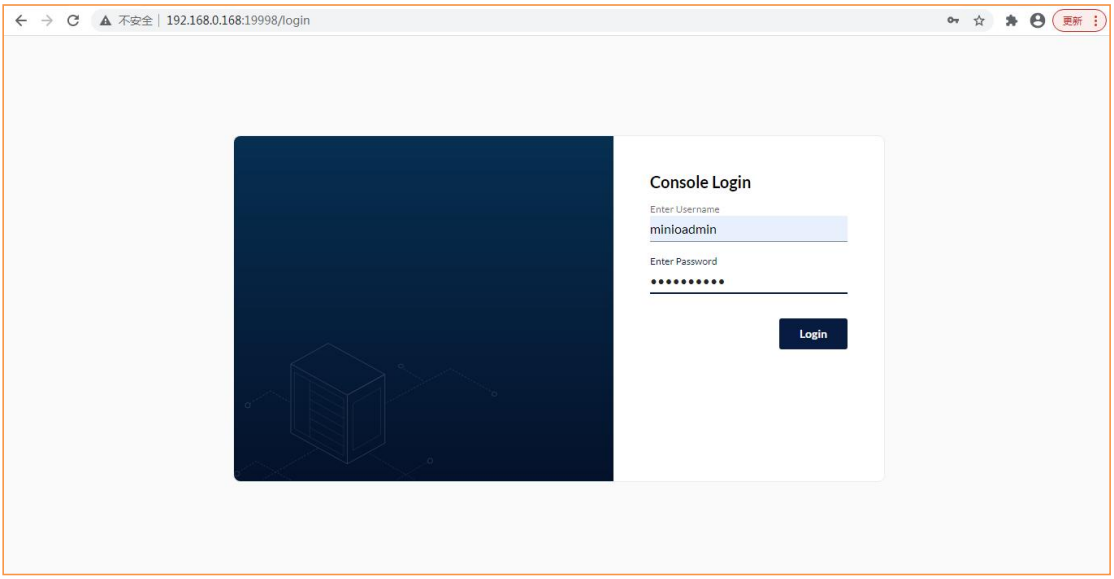
目录：cd /home/software/minio

命令：./minio_start.sh

查看 minio 进程命令：ps -ef | grep minio

```
[1] 13079
[root@192 local]# ps -ef | grep minio
root     13079 12890  7 14:49 pts/0    00:00:00 /root/local/minio server /home/itsm/upload
root     13089 12890  0 14:49 pts/0    00:00:00 grep --color=auto minio
[root@192 local]#
```

6. 浏览器访问 IP:9000，验证 minio 安装和启动状态



如上图所示：minio 初始登录用户和密码都为 minioadmin

2.1.2.6 安装 KkFileView

1. 在 software 目录下解压 KkFileView 压缩包。

目录：cd /home/software

命令：tar -zxvf kkFileView-4.3.0.tar.gz

2. 在 KkFileView 解压包 bin 目录下执行安装命令。

目录：cd /home/software/kkFileView-4.3.0/bin

命令：./install.sh

3. 进入 LibreOffice 目录，执行安装命令。

命令：cd /tmp/LibreOffice_7.5.3.2_Linux_x86-64_rpm/RPMS

命令：yum localinstall *.rpm

如果离线状态下，可以不用执行./install.sh 命令，直接将 LibreOffice_7.5.3.2_Linux_x86-64_rpm.tar.gz 包解压至 tmp 目录，然后按上述步骤直接执行 yum localinstall *.rpm 进行安装

4. 配置 kkFileView。

命令：cd /home/software/kkFileView-4.3.0/config

命令：vim application.properties

将 cache.type 的值改为 redis；

将 redis 连接密码改为 123456；

禁止首页上传文件

file.upload.disable = ${KK_FILE_UPLOAD_ENABLED:true}

```
spring.freemarker.check-template-location = true
spring.freemarker.content-type = text/html
spring.freemarker.expose-request-attributes = true
spring.freemarker.expose-session-attributes = true
spring.freemarker.request-context-attribute = request
spring.freemarker.suffix = .ftl

# office-plugin
## office转换服务的进程数，默认开启两个进程
office.plugin.server.ports = 2001,2002
## office 转换服务 task 超时时间，默认五分钟
office.plugin.task.timeout = 5m

#文件资源路径（默认为打包根路径下的file目录下）
#file.dir = D:\\kkFileview\\
file.dir = ${KK_FILE_DIR:default}
#openoffice home路径
#office.home = C:\\Program Files (x86)\\OpenOffice 4
office.home = ${KK_OFFICE_HOME:default}

#缓存实现类型，不配默认为内嵌RocksDB(type = default)实现，可配置为redis(type = redis)实现（需要配置spring.redisson.address等参数）和 JDK
cache.type = ${KK_CACHE_TYPE:redis}
#redis连接，只有当cache.type = redis时才有用
spring.redisson.address = ${KK_SPRING_REDISSON_ADDRESS:127.0.0.1:6379}
spring.redisson.password = ${KK_SPRING_REDISSON_PASSWORD:123456}
#缓存是否自动清理 true 为开启，注释掉或其他值都为关闭
cache.clean.enabled = ${KK_CACHE_CLEAN_ENABLED:true}
#缓存自动清理时间，cache.clean.enabled = true时才有用，cron表达式，基于Quartz cron
cache.clean.cron = ${KK_CACHE_CLEAN_CRON:0 0 3 * * ?}

###################################可在运行时动态配置#####################################
#提供预览服务的地址，默认从请求url读，如果使用nginx等反向代理，需要手动设置
#base.url = https://file.keking.cn
base.url = ${KK_BASE_URL:default}

#信任站点，多个用','隔开，设置了之后，会限制只能预览来自信任站点列表的文件，默认不限制
#trust.host = file.keking.cn,kkfileview.keking.cn
trust.host = ${KK_TRUST_HOST:default}

#是否启用缓存
cache.enabled = ${KK_CACHE_ENABLED:true}
```

修改缓存方式为redis

配置对应redis地址和密码

修改配置文件参考如下：

application.properties

5. 启动 kkFileView。

   目录：cd /home/software/kkFileView-4.3.0/bin

   命令：./startup.sh

6. 验证 kkFileView 启动状态。

   浏览器访问 IP:8012

如果在线预览 word 文件出现乱码

(1) 上传 fonts.zip 至 /usr/share/fonts 目录下并解压

(2) 依次执行 mkfontscale 、mkfontdir 、fc-cache 使字体生效

(3) 如果执行 mkfontscale 失败，使用 yum install -y mkfontscale 安装

如果执行 mkfontscale 失败，可以使用 yum install -y mkfontscale 安装（离线环境时，可上传依赖包 font_rpm.zip 至/usr/share/fonts 目录并解压，然后进入解压后目录，执行安装命令 rpm -ivh ./*.rpm --nodeps --force）

(4) 授权字体访问权限，进入解压后的目录，执行命令：

chmod -R 755 *.TTF

chmod -R 755 *.ttf

chmod -R 755 *.ttc

(5) 刷新配置 source /etc/profile

(6) 重启 kkFileView

7. 清空 kkFileView 缓存

连接 redis，找到对应的 kkFileView 存储的 key 值进行删除

目录：cd /home/software/redis-6.2.6/src

命令：./redis-cli

命令：auth 填写 redis 密码进行登录

```
[root@localhost src]# ./redis-cli
127.0.0.1:6379> auth 123456
OK
```

命令：keys *

```
127.0.0.1:6379> keys *
 1) "nonce:evlykr"
 2) "nonce:5w6fhi"
 3) "nonce:3st4ku"
 4) "nonce:jy4exb"
 5) "nonce:2dm6ju"
 6) "nonce:d9fezc"
 7) "nonce:o6o1ce"
 8) "converted-preview-pdf-file"
 9) "nonce:2my2tp"
```

删除对应 key

命令：del converted-preview-pdfimgs-file

命令：del converted-preview-pdf-file

```
127.0.0.1:6379> del converted-preview-pdfimgs-file
(integer) 1
127.0.0.1:6379> del converted-preview-pdf-file
(integer) 1
```

2.1.2.7 安装 Unoconv

基于已经安装 Libreoffice 且部署好中文字体包的前提下，安装 UNOVOCN，具体 office 安装可以参考 kkFileView 里 office 安装步骤。

1. 在 software 目录下解压 unoconv 压缩包

目录：cd /home/software

命令：unzip unoconv-0.8.2.zip

2. 进入 unoconv 目录，执行安装

目录：cd /home/software/unoconv

命令：make install

3. 验证，将 md 文件转换为 pdf

命令：unoconv -f pdf CHANGELOG.md

2.1.2.8 安装 Mysql

1. 将 software 目录下 mysql 压缩包解压并重命名。

目录：cd /home/software

命令：tar -zxvf mysql-8.0.28-el7-x86_64.tar.gz

命令：mv mysql-8.0.28-el7-x86_64 mysql

2. 进入 mysql 目录下创建 data 目录。

目录：cd /home/software/mysql

命令：mkdir data

3. 创建 mysql 用户组、用户密码并授权

命令：groupadd mysql

命令：useradd -g mysql mysql

命令：chown -R mysql.mysql /home/software/mysql

4. 初始化数据库

命令：./bin/mysqld --user=mysql --basedir=/home/software/mysql/

--datadir=/home/software/mysql/data/ --initialize

--lower-case-table-names=1;

```
[root@rzitsm-host mysql]# ./bin/mysqld --user=mysql --basedir=/usr/local/mysql/ --datadir=/usr/local/mysql/data/ --initialize  --lo
wer-case-table-names=1;
2022-09-02T14:26:49.615042Z 0 [Warning] [MY-011070] [Server] 'Disabling symbolic links using --skip-symbolic-links (or equivalent)
is the default. Consider not using this option as it' is deprecated and will be removed in a future release.
2022-09-02T14:26:49.615133Z 0 [System] [MY-013169] [Server] /usr/local/mysql/bin/mysqld (mysqld 8.0.28) initializing of server in p
rogress as process 82045
2022-09-02T14:26:49.623677Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
2022-09-02T14:26:49.806363Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2022-09-02T14:26:51.447342Z 6 [Note] [MY-010454] [Server] A temporary password is generated for root@localhost: TjA7Xp;J2ku%
[root@rzitsm-host mysql]#
```

TjA7Xp;J2ku%为初始化密码。

5. 修改 my.cnf 文件，配置 mysql 连接相关值

命令：vim /etc/my.cnf

增加或修改内容：port、basedir、datadir、socket、lower_case_table_names、skip-grant-tables

注意：skip-grant-tables 这一行配置，需要先开放，修改完密码后再回到这个文件注释这一行配置。

参考如下：

my.cnf

```
[mysqld]
port=3306
basedir=/home/mysql
datadir=/home/mysql/data
socket=/tmp/mysql.sock
lower_case_table_names=1
# Disabling symbolic-links is recommended to prevent assorted security risks
character-set-server=UTF8MB4
default-storage-engine=INNODB
default_authentication_plugin=mysql_native_password
skip-grant-tables
# Settings user and group are ignored when systemd is used.
# If you need to run mysqld under a different user or group,
# customize your systemd unit file for mariadb according to the
# instructions in http://fedoraproject.org/wiki/S
# include all files from the config directory
~
```

6. 添加 mysqld 服务到系统并授权

目录：cd /home/soft/mysql

命令：cp -a ./support-files/mysql.server /etc/init.d/mysqld

命令：chmod +x /etc/init.d/mysqld

命令：chkconfig --add mysqld

7. 检查服务是否生效

命令：chkconfig --list mysqld

```
[root@localhost support-files]# cp -a ./mysql.server /etc/init.d/mysqld
[root@localhost support-files]# cd /etc/in
init.d/  inittab  inputrc
[root@localhost support-files]# cd /etc/init.d/
[root@localhost init.d]# ls
functions  mysqld  netconsole  network  README
[root@localhost init.d]# chmod +x /etc/rc.d/init.d/mysqld
[root@localhost init.d]# chkconfig --add mysqld
[root@localhost init.d]# chkconfig --list mysqld

Note: This output shows SysV services only and does not include native
      systemd services. SysV configuration data might be overridden by native
      systemd configuration.

      If you want to list systemd services use 'systemctl list-unit-files'.
      To see services enabled on particular target use
      'systemctl list-dependencies [target]'.

mysqld          0:off   1:off   2:on    3:on    4:on    5:on    6:off
[root@localhost init.d]#
```

https://blog.csdn.net/oschina_41140683

8. 将 mysql 命令添加到服务，用于客户端登录

命令：ln -s /home/software/mysql/bin/mysql /usr/bin

如果存在则使用下面命令覆盖

命令：ln -sf /home/software/mysql/bin/mysql /usr/bin

9. 启动 MySQL 服务

命令：service mysqld start

10. 查看启动状态

命令：service mysqld status

11. 登录 mysql

命令：mysql -uroot -p

提示输入密码，由于当前/etc/my.cnf 里配置了免密登录 skip-grant-tables

则直接回车就可以进入 mysql

12. 进入 mysql 数据库，刷新权限

命令：use mysql;

命令：flush privileges;

13. 清空 root 密码

命令：update user set authentication_string='' where user='root';

命令：flush privileges;

14. 修改 root 密码及密码验证方式

命令：alter user 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY '123456';

命令：flush privileges;

执行命令后，123456 就是 root 账号设置的新密码

15. 修改 root 账号允许远程访问（如果需要远程访问可以设置）

命令：update user set host='%' where user='root';

命令：flush privileges;

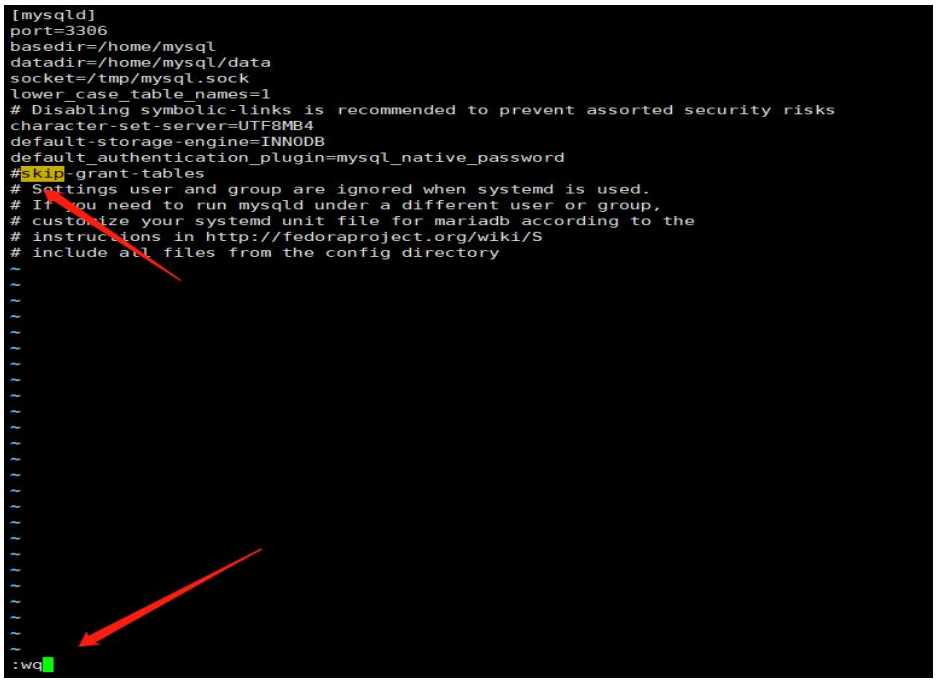
16. 退出 mysql

命令：quit;

17. 取消免密登录

编辑 my.cnf 文件，将 skip-grant-tables 注释然后保存文件退出

命令：vim /etc/my.cnf



18. 重启 mysql 服务

命令：service mysqld restart

2.1.3 平台部署

2.1.3.1 部署文件清单

类型	文件	说明
引擎部署文件	config 文件夹	配置文件夹，包含两个配置文
	sjt_agent.jar	
	setenv.sh	
	ff-apaas-engine.war	引擎 war 包
后端部署文件	config 文件夹	配置文件夹，包含两个配置文
	ffapaas.war	后端 war 包
前端部署文件	web.tar.gz	前端部署压缩包
移动端部署文件	mobileweb.tar.gz	移动端部署压缩包

数据库文件	ffapaas.sql	初始化数据库备份文件
授权文件及配置参数文件	https://app.afanbuild.com/externalLink/dataAddition?linkType=2&refAppid=877463D66D7241A08AB79DD4028A0D11&createrId=204305C1B332454D8AE6D9DC52F6E98C&flowmodelid=26CE5D70CA484C08B87C19D754FF5D44	访问链接填写表单获取授权文件及配置参数文件

2.1.3.2 初始化数据库

1. 将数据库备份文件 ffapaas.sql 文件上传到目录 home/ffapaas 下
2. 登录 mysql 创建 ffapaas 数据库并设置字符集；

   命令：mysql -uroot -p

   命令：create database ffapaas;

   命令：show databases;

   命令：use ffapaas;

   命令：set names utf8mb4;
3. 导入数据库；

   命令：source /home/ffapaas/ffapaas.sql;

2.1.3.3 系统引擎部署

1. 将 Tomcat 压缩包上传至 engine 目录下解压并重命名（如已完成则忽略此步骤）

   进入目录：cd /home/ffapaas/engine

   解压：tar -zxvf apache-tomcat-8.5.57.tar.gz

重命名：mv apache-tomcat-8.5.57 tomcat

删除压缩包：rm -f apache-tomcat-8.5.57.tar.gz

2. 将 JDK 压缩包上传至 engine 目录下解压并重命名

进入目录：cd /home/ffapaas/engine

解压：tar -zxvf jdk-8u201-linux-x64.tar.gz

重命名：mv jdk1.8.0_201 jdk

删除压缩包：rm -rf jdk-8u201-linux-x64.tar.gz

3. Tomcat 监听端口配置（此步骤可根据实际端口情况灵活调整）

进入目录：cd /home/ffapaas/engine/tomcat/conf/

编辑文件：vim server.xml

将 <Server port="8005" shutdown="SHUTDOWN"> 中的 8005 改为 6005；

将<Connector port="8080" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443" />中的 8080 改为 6161、8443 改成 6443

4. Tomcat 配置指定 JDK

修改 bin 目录下 setclasspath.sh 文件内容 ， 指定 JDK 目录；

JDK 安装目录/home/software/jdk

进入目录：cd /home/ffapaas/engine/tomcat/bin/

拷贝文件夹：cp /home/software/jdk/ -a /home/ffapaas/engine

编辑文件：vim setclasspath.sh

在文件中顶部添加如下内容：

export JAVA_HOME=/home/ffapaas/engine/jdk

export JRE_HOME=/home/ffapaas/engine/jdk/jre

```
24 # Make sure prerequisite environment variables are set
25 export JAVA_HOME=/home/ffapaas/engine/jdk
26 export JRE_HOME=/home/ffapaas/engine/jdk/jre
27
28 if [ -z "$JAVA_HOME" ] && [ -z "$JRE_HOME" ]; then
29   if $darwin; then
```

5. 上传相关文件

(1) 将授权文件和 sjt_agent.jar 文件上传到/home/ffapaas/engine/目录下；

(2) 将 setenv.sh 文件上传到/home/ffapaas/engine/tomcat/bin/目录下；

setenv.sh 文件中内容设置环境变量如：

export CATALINA_OPTS="$CATALINA_OPTS

-javaagent:/home/ffapaas/engine/sjt_agent.jar"；

(3) 将配置文件夹 config 上传至 tomcat 目录 conf 文件夹下，config 中包含文件：

application.yml 和 application-prod.yml 文件；

修改 application.yml 文件内容：

- minioEndpoint：minio 服务地址及端口；
- accessKey：启动 minio 服务时的 MINIO_ACCESS_KEY 的值，默认 minioadmin；
- secretKey：启动 minio 服务时设置的 MINIO_SECRET_KEY 值，默认 minioadmin；
- fileFullPath：授权文件完整路径加文件名及文件后缀；
- platformPrivateKey：授权文件私钥；
- platformUrl：平台后台接口地址；
- dbType: 数据库类型；

修改 application-prod.yml 文件内容：

- driverClassName：数据库驱动；
- master 下 url：数据库配置连接地址及相关参数；
- master 下 username：数据库登录账号；
- master 下 password：数据库登录密码（加密后的字符串）；
- privateKey：密码解密私钥；
- redis 下 host：redis 地址；
- redis 下 port：redis 地址端口；
- redis 下 password：redis 登录密码；

6. 部署引擎包

将引擎包文件 ff-apaas-engine.war 上传/home/ffapaas/engine/tomcat/webapps/目录下；

7. 验证

启动 tomcat

目录：/home/ffapaas/engine/tomcat/bin

命令：./startup.sh

浏览器访问 http://IP:6161/ff-apaas-engine/doc.html，显示如下图表示部署成功！

2.1.3.4 应用后端部署

1. 将 Tomcat 压缩包上传到 back 目录下解压并重命名（如已完成则忽略此步骤）

进入目录：cd /home/ffapaas/back

解压：tar -zxvf apache-tomcat-8.5.57.tar.gz

重命名：mv apache-tomcat-8.5.57 tomcat

删除压缩包：rm -f apache-tomcat-8.5.57.tar.gz

2. 将 JDK 压缩包上传至 back 目录下解压并重命名

进入目录：cd /home/ffapaas/back

解压：tar -zxvf jdk-8u201-linux-x64.tar.gz

重命名：mv jdk1.8.0_201 jdk

删除压缩包：rm -rf jdk-8u201-linux-x64.tar.gz

3. Tomcat 配置指定 JDK

修改 bin 目录下 setclasspath.sh 文件内容 ， 指定 JDK 目录；

JDK 安装目录/home/software/jdk

进入目录：cd /home/ffapaas/back/tomcat/bin/

拷贝文件夹：cp /home/software/jdk/ -a /home/ffapaas/back

编辑文件：vim setclasspath.sh

在文件中顶部添加如下内容：

export JAVA_HOME=/home/ffapaas/back/jdk

export JRE_HOME=/home/ffapaas/back/jdk/jre

```
24 # Make sure prerequisite environment variables are set
25 export JAVA_HOME=/home/ffapaas/back/jdk
26 export JRE_HOME=/home/ffapaas/back/jdk/jre
27
28 if [ -z "$JAVA_HOME" ] && [ -z "$JRE_HOME" ]; then
```

4. 上传相关文件

将配置文件夹 config 上传至 tomcat 目录 conf 文件夹下，config 中包含文件：

application.yml 和 application-prod.yml 文件；

修改 application.yml 文件内容：

- profile: 上传文件临时路径 默认/home/ffapaas/upload；
- minioEndpoint：minio 服务地址及端口；

- accessKey：启动 minio 服务时的 MINIO_ACCESS_KEY 的值，默认 minioadmin；
- secretKey：启动 minio 服务时设置的 MINIO_SECRET_KEY 值，默认 minioadmin；
- engineUrl：引擎后台接口地址；
- mobileBackEnd：移动端后台地址(暂时不变，后续配置完移动端在调整)
- dbType：数据库类型；
- appKey：签名验证编号；
- appSecret：签名验证秘钥；
- signSource：签名来源；

修改 application-prod.yml 文件内容：

- driverClassName：数据库驱动；
- master 下 url：数据库配置连接地址及相关参数；
- master 下 username：数据库登录账号；
- master 下 password：数据库登录密码（加密后的字符串）；
- privateKey：密码解密私钥；
- redis 下 host：redis 地址；
- redis 下 port：redis 地址端口；
- redis 下 password：redis 登录密码；

5. 部署后台包

将后台包文件 ffapaas.war 上传到/home/ffapaas/back/tomcat/webapps/目录下；

6. 验证

启动 tomcat

目录：/home/ffapaas/back/tomcat/bin

命令：./startup.sh

浏览器访问 http://IP:8080/ffapaas/doc.html，显示如下图表示部署成功！

2.1.3.5 应用前端部署

1. 上传 web 部署压缩包到 ffapaas 目录下解压；

   进入目录：cd /home/ffapaas

   解压：tar -zxvf web.tar.gz

   如果压缩包是 web.zip，则执行 unzip web.zip 进行解压。

2. 修改前端配置

   进入目录：cd /home/ffapaas/web

   编辑文件：vim config.js

   - window.BASEURL：后台 API 接口地址；
   - window.ENGINEBASEURL：流程引擎接口地址；
   - window.PLATFORMURL：管理后台地址；
   - window.FILEVIEW：在线预览地址；
   - window.APPKEY：签名验证编号；
   - window.APPSECRET：签名验证秘钥；
   - window.SIGNSOURCE：签名来源；

3. 修改 nginx 配置并启动 nginx

   编辑 nginx 配置: vim /home/software/nginx-1.18.0/conf/nginx.conf

nginx.conf

进入目录：cd /home/software/nginx-1.18.0/sbin/

重启 nginx：./nginx -s reload -c /home/software/nginx-1.18.0/conf/nginx.conf

2.1.3.6 移动端应用部署（微信公众号）

2.1.3.6.1 移动后端部署

1. 登录 PC 端网页进入微信公众平台管理中心->服务连接->公众号填写相关公众号配置值，如下图：

2. 在 ffapaas 目录下创建 mobile 目录，在 mobile 目录下创建 mobile_back 目录

3. 将 Tomcat 压缩包上传到 mobile_back 目录下解压并重命名

进入目录：cd /home/ffapaas/mobile/mobile_back

解压：tar -zxvf apache-tomcat-8.5.57.tar.gz

重命名：mv apache-tomcat-8.5.57 tomcat

删除压缩包：rm -f apache-tomcat-8.5.57.tar.gz

4. 将 JDK 压缩包上传至 mobile_back 目录下解压并重命名

进入目录：cd /home/ffapaas/mobile/mobile_back

解压：tar -zxvf jdk-8u201-linux-x64.tar.gz

重命名：mv jdk1.8.0_201 jdk

删除压缩包：rm -rf jdk-8u201-linux-x64.tar.gz

5. Tomcat 监听端口配置

进入目录：cd /home/ffapaas/mobile/mobile_back/tomcat/conf/

编辑文件：vim server.xml

将<Server port="8005" shutdown="SHUTDOWN">中的 8005 改为 7005；

将<Connector port="8080" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443" />中的 8080 改为 7171、8443 改成 7443

6. Tomcat 配置指定 JDK

修改 bin 目录下 setclasspath.sh 文件内容 ， 指定 JDK 目录；

JDK 安装目录/home/software/jdk

进入目录：cd /home/ffapaas/mobile/mobile_back/tomcat/bin/

拷贝文件夹：cp /home/software/jdk/ -a /home/ffapaas/mobile/mobile_back

编辑文件：vim setclasspath.sh

在文件中顶部添加如下内容：

```
export JAVA_HOME=/home/ffapaas/mobile/mobile_back/jdk
export JRE_HOME=/home/ffapaas/mobile/mobile_back/jdk /jre
```

7. 上传相关文件

将配置文件夹 config_mobile 上传至 tomcat 目录 conf 文件夹下，config 中包含文件：

application.yml 和 application-prod.yml 文件；

修改 application.yml 文件内容：

- sourceArray：支持的移动端部署类型；
- backUrl：后台接口地址；

修改 application-prod.yml 文件内容：

- redis 下 host：redis 地址；
- redis 下 port：redis 地址端口；
- redis 下 password：redis 登录密码；

8. 部署后台包

将移动后台包文件 ff-apaas-mobile.war 上传到

/home/ffapaas/mobile/mobile_back/tomcat/webapps/目录下

9. 验证

启动 tomcat

目录：/home/ffapaas/mobile/mobile_back /tomcat/bin

命令：./startup.sh

浏览器访问 http://IP:6161/ff-apaas-mobile/doc.html

10. 修改 ffapaas 后台

进入目录：cd /home/ffapaas/back/tomcat/conf/config

编辑文件：vim application.yml

修改 application.yml 文件内容：

- mobileBackEnd：移动端后台地址

进入目录：cd /home/ffapaas/back/tomcat/bin

关闭 tomcat: ./shutdown.sh

启动 tomcat: ./startup.sh

2.1.3.6.2 移动前端部署

1. 上传 web 部署压缩包到 mobile 目录下解压；

   进入目录：cd /home/ffapaas/mobile

   解压：tar -zxvf mobileweb.tar.gz

   如果压缩包是 mobileweb.zip，则执行 unzip mobileweb.zip 进行解压。

2. 修改前端配置

   进入目录：cd /home/ffapaas/mobile/mobile_web/static/config

   编辑文件：vim url.config.js

   - BASEURL：后台 API 接口地址；
   - ENGINEBASEURL：流程引擎接口地址；
   - FILEVIEW：在线预览地址；
   - PLATFORM：移动端类型,默认值 MP-WEIXIN；
   - BASEURLMOBILE: 移动后台接口地址；
   - APPKEY：签名验证编号；
   - APPSECRET: 签名验证秘钥；
   - SIGNSOURCE: 签名来源；
   - Invitetoregister：邀请注册地址；

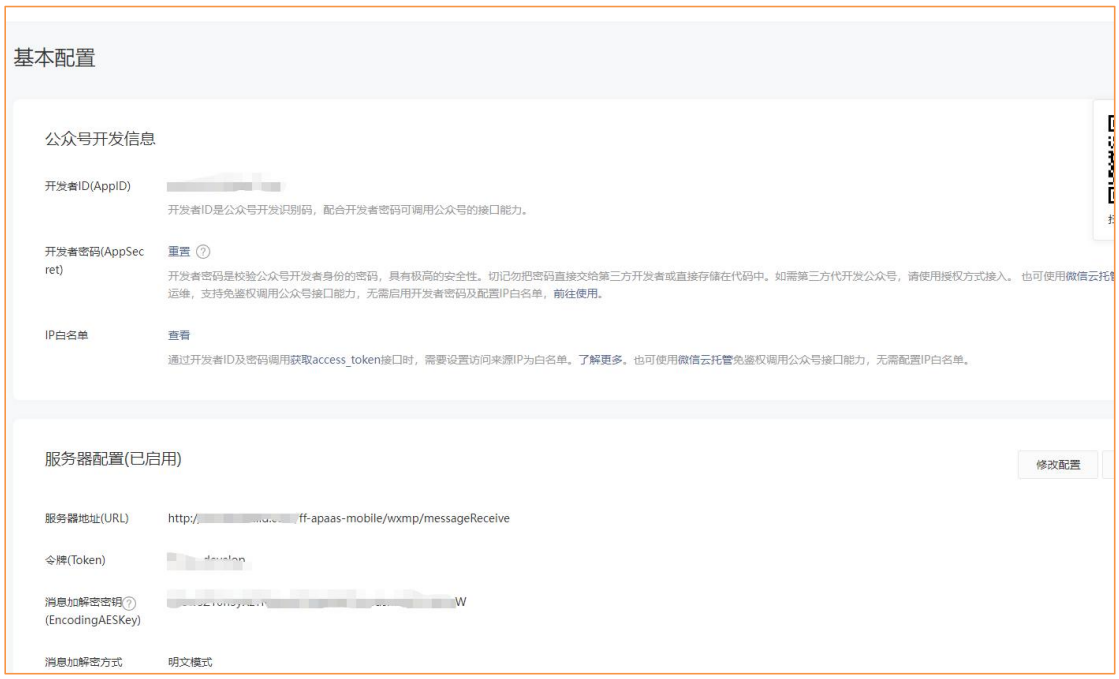
3. 配置并重新启动 nginx

   进入目录：cd /home/software/nginx-1.18.0/conf 修改 nginx.conf 增加移动端代理

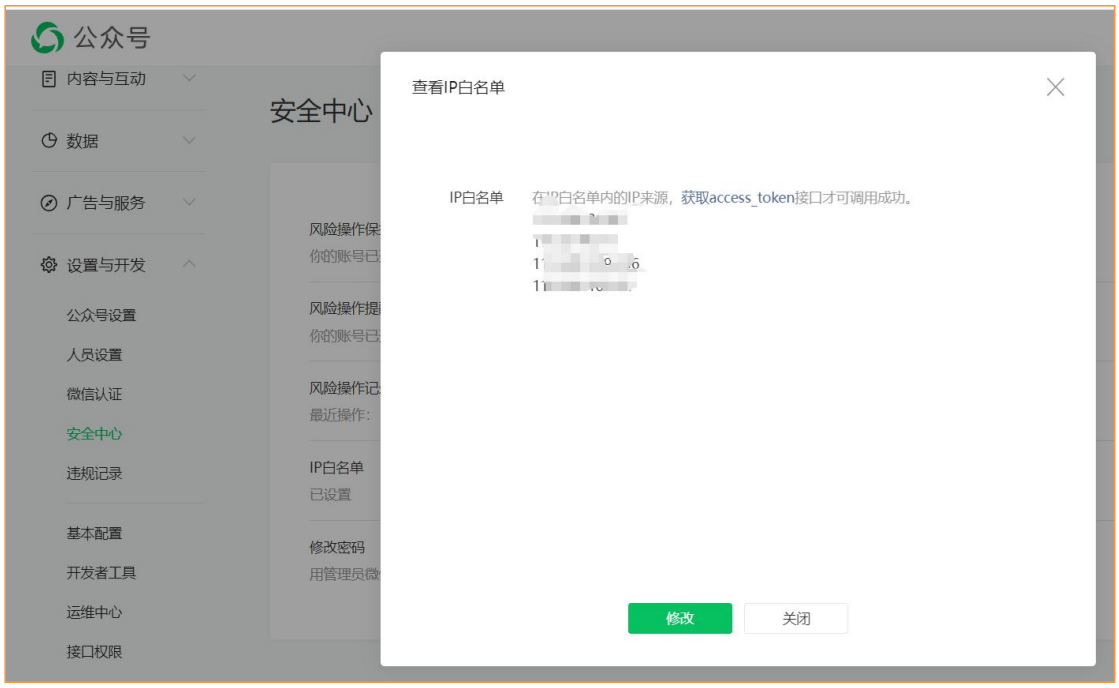
   进入目录：cd /home/software/nginx-1.18.0/sbin/

   重启 nginx：./nginx -s reload -c /home/software/nginx-1.18.0/conf/nginx.conf

4. 进入微信公众平台修改相关配置，及配置访问白名单


基本配置
公众号开发信息
开发者ID(AppID)
开发者ID是公众号开发识别码，配合开发者密码可调用公众号的接口能力。
开发者密码(AppSecret)
重置
开发者密码是校验公众号开发者身份的密码，具有极高的安全性。切记勿把密码直接交给第三方开发者或直接存储在代码中。如需第三方代开发公众号，请使用授权方式接入。 也可使用微信云托管
运维，支持免鉴权调用公众号接口能力，无需启用开发者密码及配置IP白名单，前往使用。
IP白名单
查看
通过开发者ID及密码调用获取access_token接口时，需要设置访问来源IP为白名单。了解更多。也可使用微信云托管免鉴权调用公众号接口能力，无需配置IP白名单。
服务器配置(已启用)
修改配置
服务器地址(URL)
ff-apaas-mobile/wxmp/messageReceive
令牌(Token)
消息加解密密钥(EncodingAESKey)
消息加解密方式
明文模式

首页
通知中心
内容与互动
数据
广告与服务
设置与开发
公众号设置
人员设置
微信认证
安全中心
违规记录
基本配置
开发者工具
公众号设置
账号详情
功能设置
授权管理
功能设置
微信音频播放器
允许微信音频播放器音乐故事场景引用公众号文章片段
隐私设置
已允许 通过名称搜索到本账号
图片水印
使用名称作为水印
业务域名
JS接口安全域名
网页授权域名



2.2 私有化容器部署

2.2.1 离线部署

2.2.1.1 环境资源清单

资源类型	资源名称	镜像/版本	说明
运行环境	JDK	1.8	
容器引擎	Docker	24.0.2（docker-ce-24.0.2-1.el7.x86_64.tar.gz）	

容器编排服务	Docker Compose	2.21.0	
中间件	Tomcat	tomcat-8.5.47.tar、ff-tomcat-fonts-8.5.47.tar	
Web 服务器	Nginx	nginx-1.20.2.tar	
缓存存储数据	Redis	redis-6.2.6.tar	
文件存储服务	Minio	minio-RELEASE.2022-03-03T21-21-16Z.tar	
附件在线预览	KkFileView	kkfileview-v4.0.0.tar	
关系数据库	MySQL	mysql-8.0.28.tar	

2.2.1.2 部署文件清单

文件	说明
docker-deploy-files.tar.gz	下载链接： https://pan.baidu.com/s/1NNFpaJwq8hK9l6X07-QA2w?pwd=szff
阿凡搭平台部署参数提供表.docx	访问以下链接填写表单获取： https://app.afanbuild.com/externalLink/dataAddition?linkType=2&refAppid=877463D66D7241A08AB79DD4028A0D11&createrId=204305C1B332454D8AE6D9DC52F6E98C&flowmodelid=26CE5D70CA484C08B87C19D754FF5D44

2.2.1.3 Docker 安装

1. 将文件 docker-deploy-files.tar.gz 上传到服务器/home 目录下；

   执行解压命令：tar -zxvf /home/docker-deploy-files.tar.gz

2. 安装 rpm 软件包；

   执行命令：cd /home/docker-deploy-files/

   执行解压命令：tar -zxvf ffapaas-rpm-packages.tar.gz

   成功解压后，进入解压目录：cd ffapaas-rpm-packages/

   执行授权命令：chmod +x install_rpm_packages.sh

   执行安装命令：./install_rpm_packages.sh

3. 安装 docker 及相关依赖；

   执行命令：cd /home/docker-deploy-files/

   执行解压命令：tar -zxvf docker-ce-24.0.2-1.el7.x86_64.tar.gz

   成功解压后，进入解压目录：cd docker/

   执行安装命令：rpm -ivh *.rpm --force --nodeps

4. 启动 docker；

   查看安装的 docker 信息，命令：docker info

   将 docker-compose 复制到/usr/local/bin 目录下

命令：cp /usr/libexec/docker/cli-plugins/docker-compose /usr/local/bin/

启动 docker，命令：systemctl start docker

2.2.1.4 镜像加载

1. 执行加载镜像命令；

执行命令：cd /home/docker-deploy-files/docker-images

加载 redis 镜像命令：docker load -i redis-6.2.6.tar

加载 minio 镜像命令：docker load -i minio-RELEASE.2022-03-03T21-21-16Z.tar

加载 kkfileview 镜像命令：docker load -i kkfileview-v4.0.0.tar

加载 tomcat 镜像命令：

docker load -i tomcat-8.5.47.tar

docker load -i ff-tomcat-fonts-8.5.47.tar

加载 nginx 镜像命令：docker load -i nginx-1.20.2.tar

加载 mysql 镜像命令：docker load -i mysql-8.0.28.tar

查看所有镜像：docker images

2.2.1.5 平台部署

2.2.1.5.1 修改配置参数

编辑/home/docker-deploy-files/ffapaas/install.sh 脚本文件，配置以下 5 个参数：

```
# 授权文件私钥
export PLATFORM_PK="706b9783744bbdd07265026cedc56bd78bfe2f35424b0fc66e5c4a79b78732d5"

# 数据库登录加密密码
export DATABASE_PWD="043d11cd209f18974117719c84ba44c5660f4c755509d013bec228b569bc8b64a1fabbf2d5fca2bad14a3d59b304c

# 数据库密码解密私钥
export DATABASE_PK="c107e156b8e4cb5177e9b1205bea914c25836a8fba4739e4524d97c8eea0c207"

# 签名验证编号
export APPKEY="1UE6VTLG"

# 签名验证密钥
export APPSECRET="BAA15CDC430A35DFD6B877FFE092C612"

# 签名来源
export SIGNSOURCE="E40580BFDC4DB90C64DC1C4941254D60"
```

以上参数均来自《阿凡搭平台部署参数提供表.docx》，请参照对应的参数进行修改。

2.2.1.5.2 替换授权文件

从《阿凡搭平台部署参数提供表.docx》获取授权文件，覆盖替换服务器上的 /home/docker-deploy-files/ffapaas/engine/authorizationKey.txt 文件。

2.2.1.5.3 授权并启动部署

命令：cd /home/docker-deploy-files/ffapaas

命令：chmod +x install.sh firewalld-centos.sh

命令：dos2unix firewalld-centos.sh install.sh

命令：./install.sh

2.2.1.5.4 验证部署结果

查看部署情况，执行命令：docker ps

浏览器直接访问 web 应用部署服务器 IP 地址

3 系统初始化

系统部署完成后，建议按照以下几步对系统进行初始化工作。

3.1 设置系统超级管理员

系统部署完成后，通过 URL 地址访问系统登录界面，点击如下图所示“账号重置”按钮即可初始化超级管理员账号。（账号重置完成后，该账号重置按钮会自动消失）

3.2 创建组织机构及用户

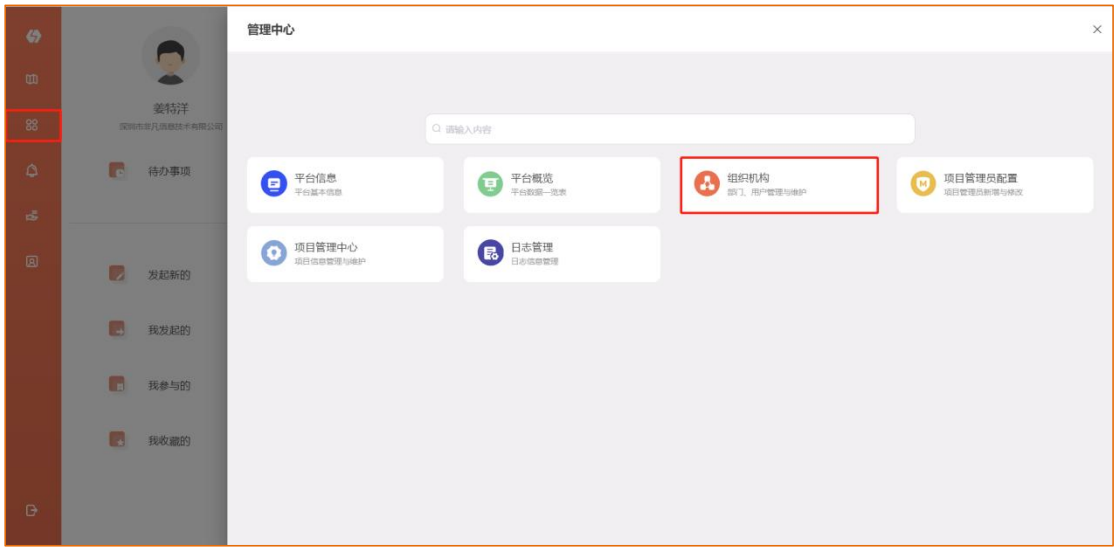
当我们在创建系统功能模块时，常常会关联到组织架构及用户的相关数据（例如权限分配、业务流程审批、表单数据填写），所以为了更贴切真实的业务场景，我们建议在创建系统功能模块之前，先完成组织机构及用户的创建。

阿凡搭平台支持三种方式（单个录入、excel 批量导入、同步其他系统数据）来完成组织机构及用户的创建。

第一种：部门及用户单个录入

1．单个录入：用超级管理员账号登录到系统，在最左侧点击“管理中心”菜单，然后

点击管理中心面板中的”组织机构“。（操作如下图）



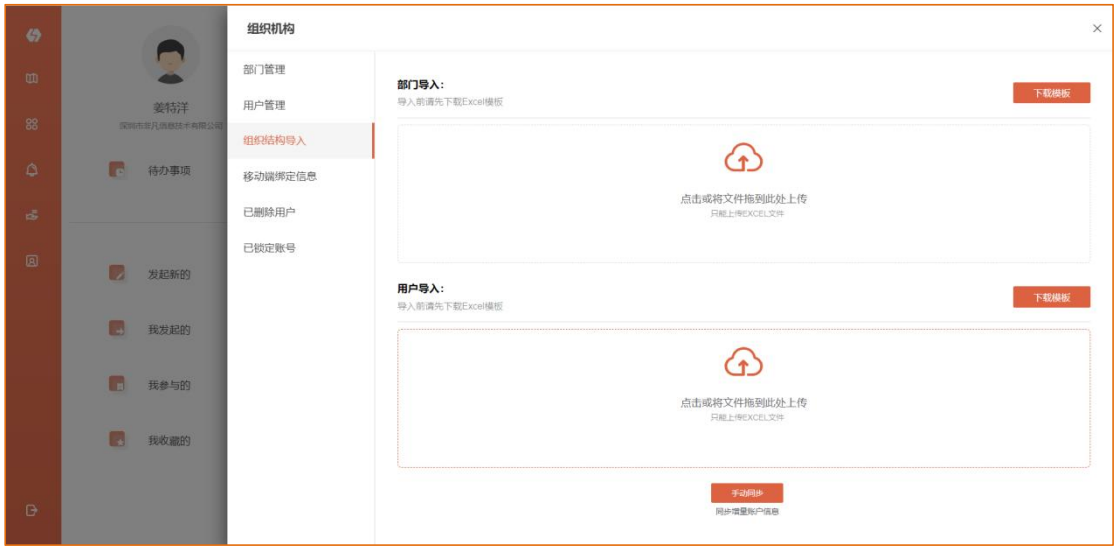
2．在弹出的页面中，通过部门管理及用户管理对企业部门及企业用户进行创建与维护。（操作如下图）

（部门维护）

（用户维护）

第二种：excel 批量导入组织结构

前面步骤参考上述步骤，选择”组织机构导入“菜单，下载相关模板，在模板中填充相关数据，然后按照提示进行文件上传，点击导入，组织结构则导入成功。（操作如下图所示）



（组织机构导入）

备注：用户导入时需填写部门编码，查看部门编码可通过点击部门管理，然后在部门树形结构中点击具体的部门名称，在右边属性栏中可看到对应部门的部门编码。若无部门编码可自定义一个（需确保部门编码的唯一性）。

（查看部门编码）

第三种：同步其他系统数据

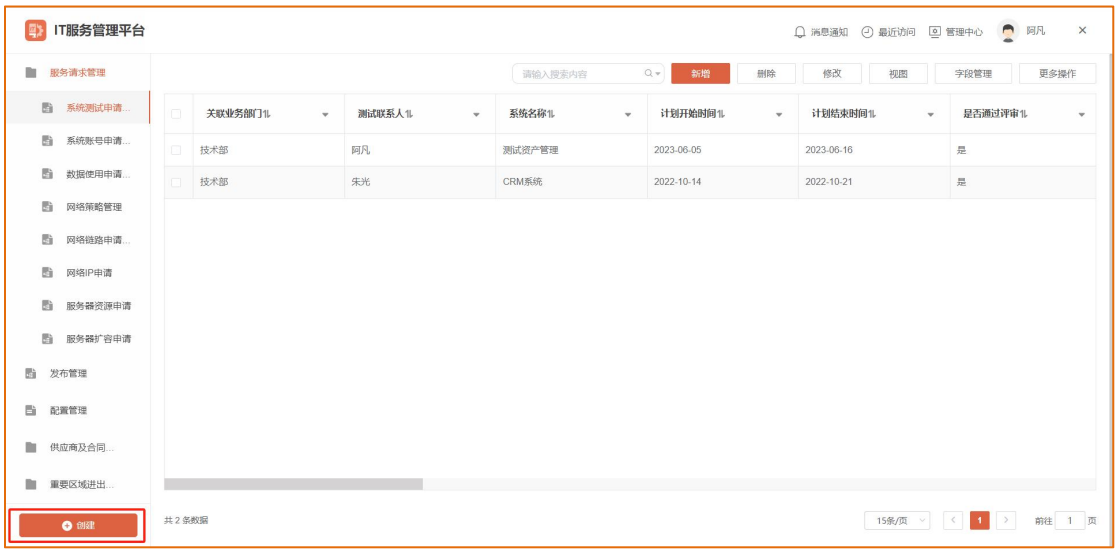
同步其他系统来完成组织机构及用户的创建一般需要根据每个企业的实际环境做少量二次开发才能完成，因为不同的企业主数据来源（OA 、HR 等）存在差异。

3.3 创建系统功能模块

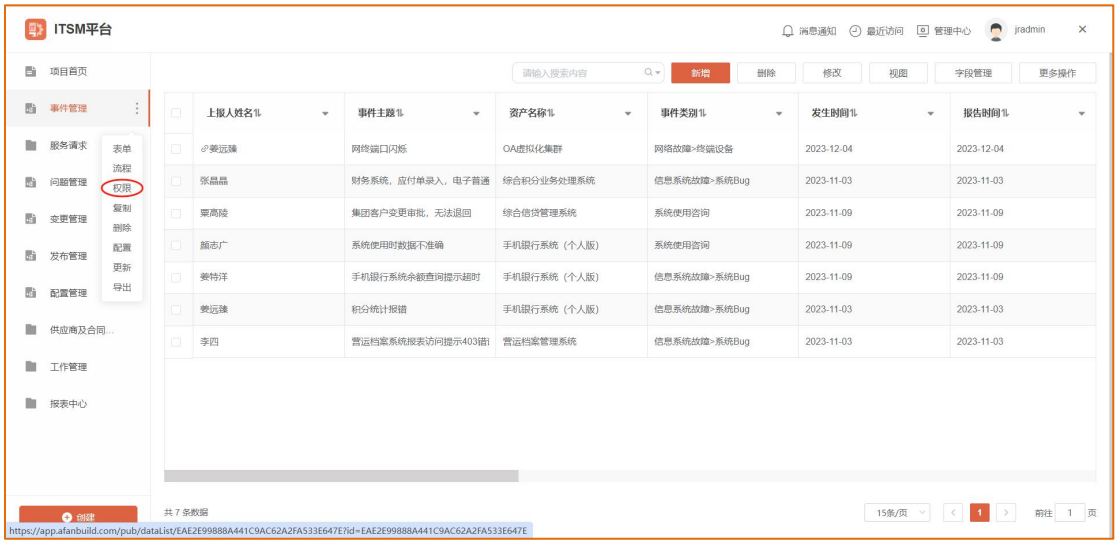
在阿凡搭平台中，通过”创建应用“的方式来完成一个新功能模块的搭建，例如：客户信息管理、合同管理、费用报销管理、请假管理等等。但每一个功能模块必须归属于某个项目，不能是独立的个体。所以我们在**创建新功能模块之前必须先创建一个项目**。

（项目创建示意图）

项目创建完成后，点击该项目即可进入项目空间，在项目空间的左侧底部有一个“创建”按钮，点击该按钮即可按照提示完成功能模块的新增。功能模块新增完成后通过权限分配即可授权他人访问该功能模块了。（详细介绍可参考阿凡搭管理员操作手册）



（功能模块创建按钮）

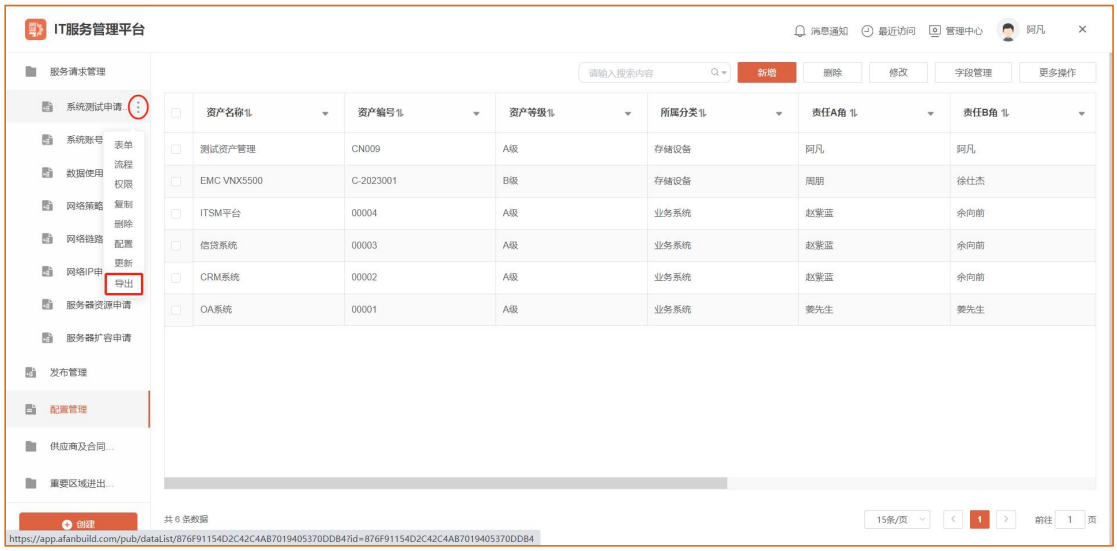


（功能模块权限分配按钮）

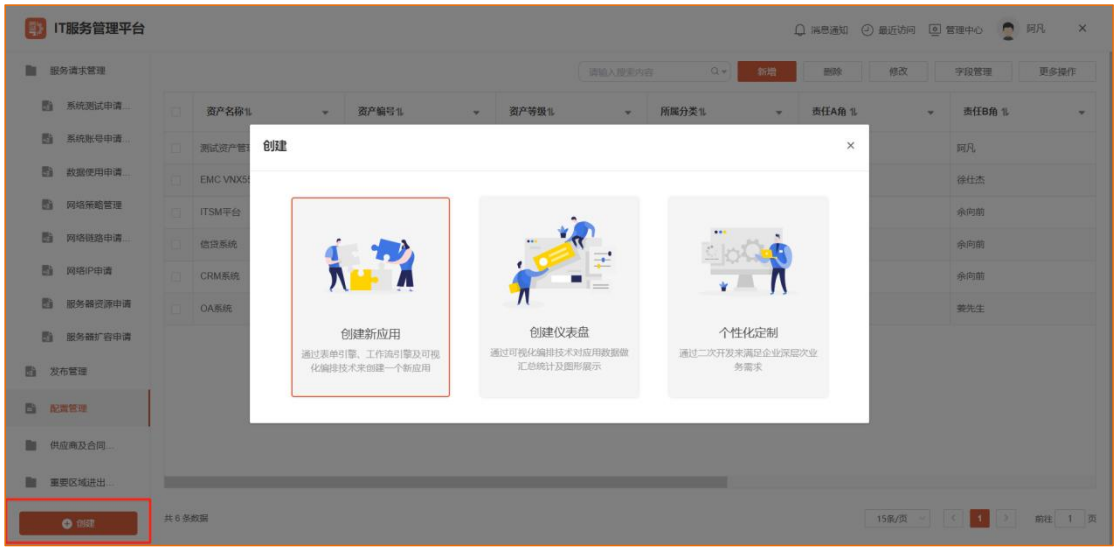
3.4 测试应用转生产环境

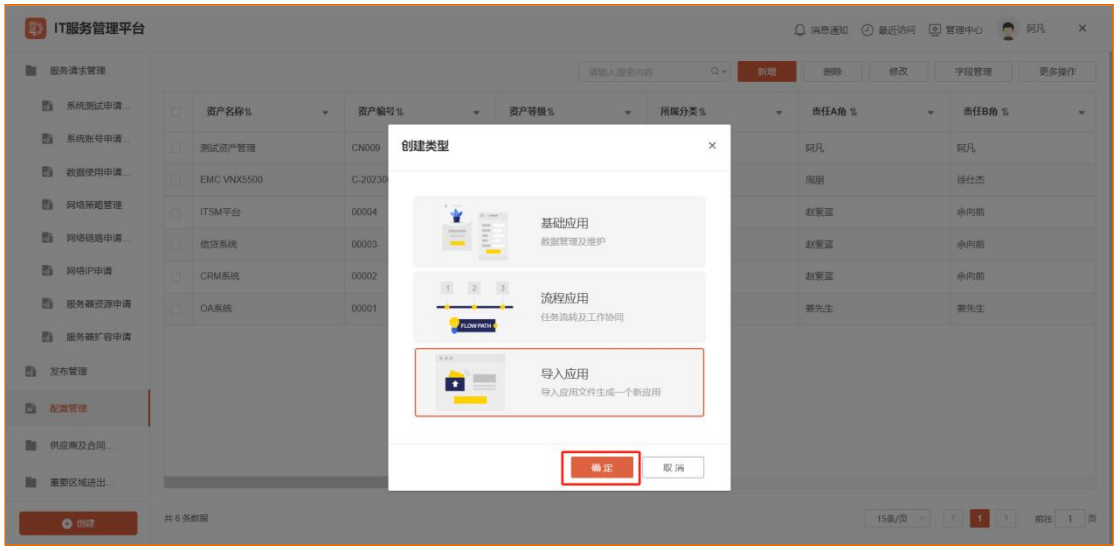
在阿凡搭平台中支持将测试环境的应用导入至生产环境，降低企业应用发布复杂度。导入应用操作步骤如下：

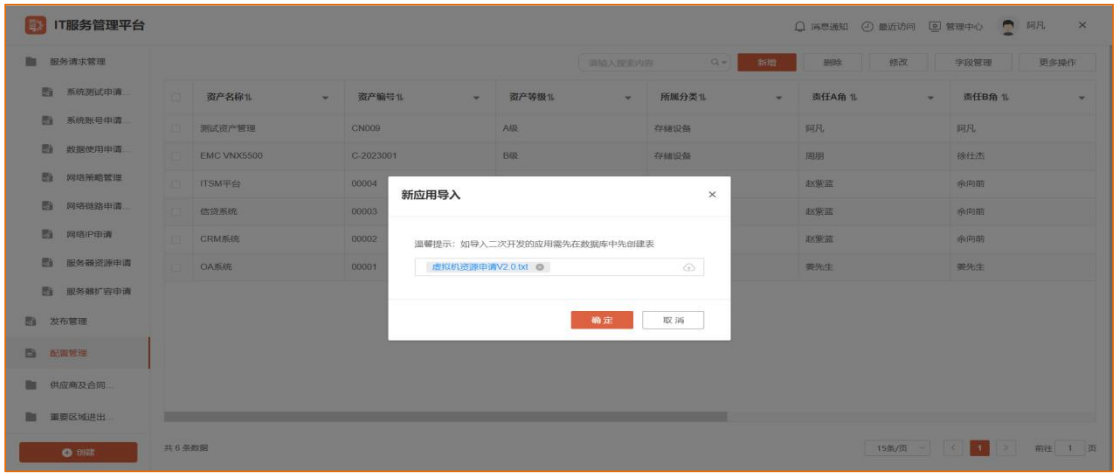
1．将鼠标移动至应用菜单，点击右侧“三个点”，在下拉菜单中点击“导出”。



2．点击底部“创建”按钮，然后点击”创建新应用”，然后选中“导入应用”选项并点击“确定”按钮，在弹出框中将刚才导出的文件上传，然后点击“确定”即导入完成。


IT服务管理平台
创建
创建新应用
通过表单引擎、工作流引擎及可视化编排技术来创建一个新应用
创建仪表盘
通过可视化编排技术对应用数据做汇总统计及图形展示
个性化定制
通过二次开发来满足企业深层次业务需求


IT服务管理平台
创建类型
基础应用
数据管理及维护
流程应用
任务流转及工作协同
导入应用
导入应用文件生成一个新应用
确定
取消


IT服务管理平台
新应用导入
温馨提示：如导入二次开发的应用需先在数据库中先创建表
确定
取消

（完成）。