数据库服务地址:192.168.52.130

FastDFS,Mysql，Redis，Zookeeper ,Elasticsearch,ActiveMq=>

Manager 服务，Search 服务=>front-web，ucenter-web,manager-web=>

nginx\_front\_web,Nginx\_manager\_web

FastDFS:192.168.52.128

Mysql,Redis,Zookeeper,Elasticsearch,ActiveMq:192.168.52.130

zmw-search-service,zmw-manager-service,Xxl-Job:192.168.52.132

nginx\_front\_web,nginx\_manager\_web,front-web，ucenter-web,manager-web:

192.168.52.131

#### Xxl-Job部署

将xxl-job数据库导入，并将xxl-job中 xxl-job-admin 中的application.properties 文件中的spring.datasource.url=jdbc:mysql://192.168.52.130:3306/xxl-job?Unicode=true&characterEncoding=UTF-8

servlet.port=8280

cd xxl-job-admin

通过maven clean package对项目进行打包

将打包好jar 发布到服务器/opt目录

cd /opt

mkdir -p /usr/local/software/xxlJob

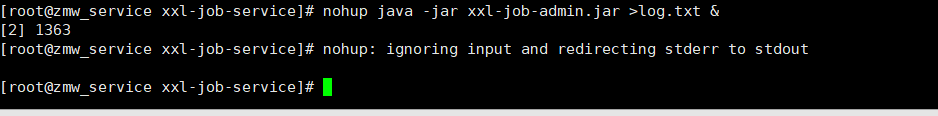
cp xxl-job-admin-2.0.2-SNAPSHOT.jar /usr/local/software/xxlJob/xxl-job-admin.jar

cd /usr/local/software/xxlJob

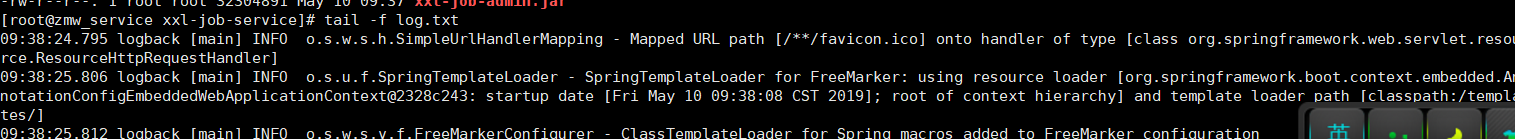
firewall-cmd --zone=public --add-port=8280/tcp --permanent 开启8280端口

firewall-cmd --reload 重启防火墙

nohup java -jar xxl-job-admin.jar >log.txt & 运行xxl-job-admin



tail -f log.txt 查看启动日志



<http://192.168.52.132:8280/xxl-job-admin/> 访问管理界面

#### Search服务部署

将项目通过maven clean package 进行打包

将生成的 zmw-my-search-service.jar 传输到 centos 的/opt目录下

启动 elasticsearch 服务和logstash 服务

开启响应的端口:

firewall-cmd --zone=public --add-port=33333/tcp --permanent search.qos.port

firewall-cmd --zone=public --add-port=20881/tcp --permanent search.dubbo.protocol.port

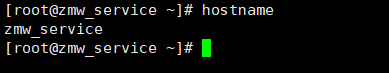
firewall-cmd --reload

cd /opt

mkdir -p /usr/local/software/zmw\_search

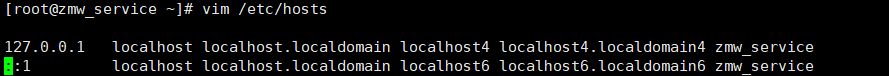
cp zmw-my-search-service.jar /usr/local/software/zmw\_search/zmw-search-service.jar

hostname查看主机名

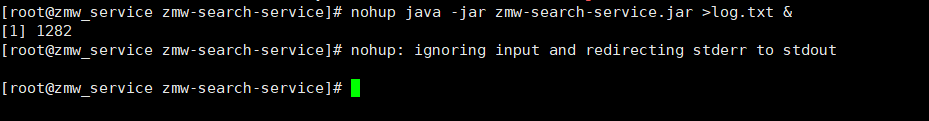


vim /etc/hosts

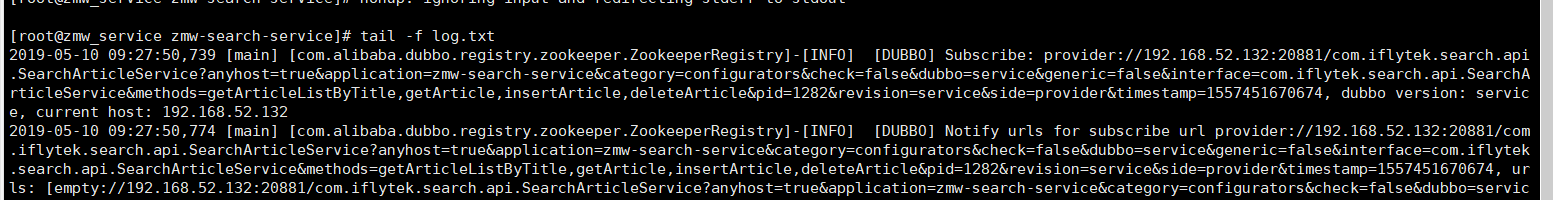
将主机名(zmw\_service)添加到末尾



nohup java -jar zmw-search-service.jar >log.txt &



tail -f log.txt 查看日志



#### 业务逻辑服务

将zmw.sql 导入数据库

将打包后生成的 zmw-manager-service.jar 传输到 centos 的/opt目录下

开启响应的端口:

firewall-cmd --zone=public --add-port=33330/tcp --permanent user.qos,port

firewall-cmd --zone=public --add-port=20880/tcp --permanent user.dubbo-protocol.port

firewall-cmd --reload

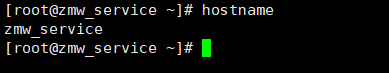
cd /opt

mkdir -p /usr/local/software/zmw\_manager\_service

cp zmw-manager-service.jar /usr/local/software/zmw\_manager\_service/zmw-manager-service.jar

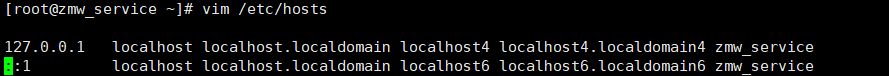
cd /usr/local/software/zmw\_manager\_service

hostname查看主机名

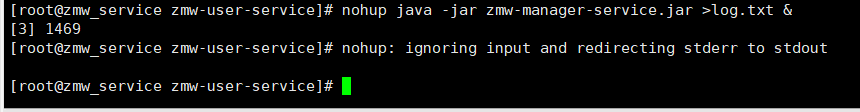


vim /etc/hosts

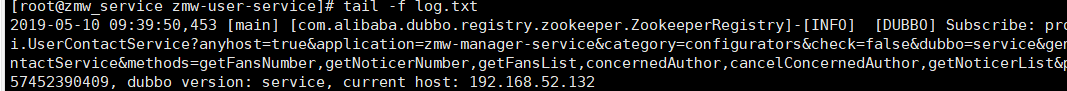
将主机名(zmw\_service)添加到末尾



nohup java -jar zmw-manager-service.jar >log.txt & 启动服务



tail -f log.txt 查看日志



#### Ucenter 部署

firewall-cmd --zone=public --add-port=8082/tcp --permanent 开启web访问端口

firewall-cmd --zone=public --add-port=33332/tcp --permanent ucenter-web.qos.port

firewall-cmd --reload 重启防火墙

将打包后的zmw-ucenter-web.war复制到Centos /opt目录下

mkdir -p /usr/local/software

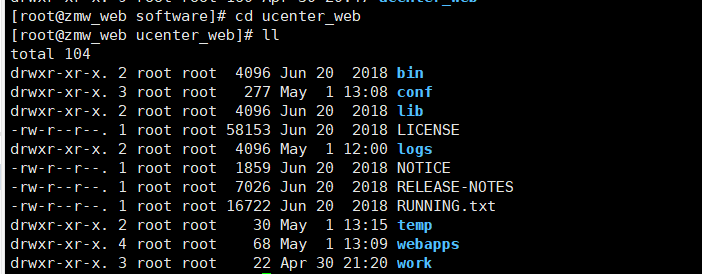
cp apache-tomcat-9.0.10.zip /usr/local/software

cd /usr/local/software

unzip apache-tomcat-9.0.10.zip

mv apache-tomcat-9.0.10 ucenter\_web

cd ucenter\_web



rm -rf webapps

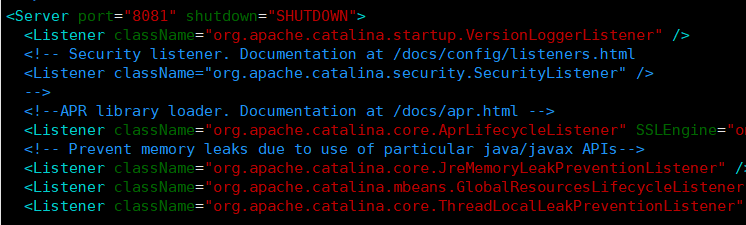
mkdir webapps

cp /opt/zmw-ucenter-web.war /usr/local/software/ucenter\_web/webapps/zmw-ucenter-web.war

vim /usr/local/software/ucenter\_web/conf/server.xml

对下列字段进行修改

<Server port="8081" shutdown="SHUTDOWN">



<Connector port="8082" protocol="HTTP/1.1"

connectionTimeout="20000"

redirectPort="8443" />

<Connector port="8083" protocol="AJP/1.3" redirectPort="8443" />

对Host进行修改

<Host name="localhost" appBase="webapps"

unpackWARs="true" autoDeploy="true">

<!-- SingleSignOn valve, share authentication between web applications

Documentation at: /docs/config/valve.html -->

<!--

<Valve className="org.apache.catalina.authenticator.SingleSignOn" />

-->

<!-- Access log processes all example.

Documentation at: /docs/config/valve.html

Note: The pattern used is equivalent to using pattern="common" -->

<Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"

prefix="localhost\_access\_log" suffix=".txt"

pattern="%h %l %u %t &quot;%r&quot; %s %b" />

<Context path ="/"

reloadable ="false"

docBase ="/usr/local/software/ucenter\_web/webapps/zmw-ucenter-web"/>

</Host>

cd /usr/local/software/ucenter\_web/bin

chmod u+x \*.sh 授权

./startup.sh 启动tomcat

tail -f /usr/local/software/ucenter\_web/logs/catalina.out 查看启动日志

启动成功 访问<http://192.168.52.131:8082/getVerificationCode> 返回验证码

#### Front-Web 部署

firewall-cmd --zone=public --add-port=8182/tcp --permanent 开启web访问端口

firewall-cmd --zone=public --add-port=33531/tcp --permanent ucenter-web.qos.port

firewall-cmd --reload 重启防火墙

将打包后的zmw-front-web.war复制到Centos /opt目录下

mkdir -p /usr/local/software

cp apache-tomcat-9.0.10.zip /usr/local/software

cd /usr/local/software

unzip apache-tomcat-9.0.10.zip

mv apache-tomcat-9.0.10 front\_web

cd front\_web

rm -rf webapps

mkdir webapps

cp /opt/zmw-front-web.war /usr/local/software/front\_web/webapps/zmw-front-web.war

vim /usr/local/software/front\_web/conf/server.xml

对下列字段进行修改

<Server port="8181" shutdown="SHUTDOWN">

<Connector port="8182" protocol="HTTP/1.1"

connectionTimeout="20000"

redirectPort="8443" />

<Connector port="8183" protocol="AJP/1.3" redirectPort="8443" />

对Host进行修改

<Host name="localhost" appBase="webapps"

unpackWARs="true" autoDeploy="true">

<!-- SingleSignOn valve, share authentication between web applications

Documentation at: /docs/config/valve.html -->

<!--

<Valve className="org.apache.catalina.authenticator.SingleSignOn" />

-->

<!-- Access log processes all example.

Documentation at: /docs/config/valve.html

Note: The pattern used is equivalent to using pattern="common" -->

<Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"

prefix="localhost\_access\_log" suffix=".txt"

pattern="%h %l %u %t &quot;%r&quot; %s %b" />

<Context path ="/"

reloadable ="false"

docBase ="/usr/local/software/front\_web/webapps/zmw-front-web"/>

</Host>

cd /usr/local/software/front\_web/bin

chmod u+x \*.sh 授权

./startup.sh 启动tomcat

tail -f /usr/local/software/front\_web/logs/catalina.out 查看启动日志

#### Manager-Web 部署

firewall-cmd --zone=public --add-port=8282/tcp --permanent 开启web访问端口

firewall-cmd --zone=public --add-port=34333/tcp --permanent ucenter-web.qos.port

firewall-cmd --reload 重启防火墙

将打包后的zmw-manager-web.war复制到Centos /opt目录下

mkdir -p /usr/local/software

cp apache-tomcat-9.0.10.zip /usr/local/software

cd /usr/local/software

unzip apache-tomcat-9.0.10.zip

mv apache-tomcat-9.0.10 manager\_web

cd manager\_web

rm -rf webapps

mkdir webapps

cp /opt/zmw-manager-web.war /usr/local/software/manager\_web/webapps/zmw-manager-web.war

vim /usr/local/software/manager\_web/conf/server.xml

对下列字段进行修改

<Server port="8281" shutdown="SHUTDOWN"

<Connector port="8282" protocol="HTTP/1.1"

connectionTimeout="20000"

redirectPort="8443" />

<Connector port="8283" protocol="AJP/1.3" redirectPort="8443" />

对Host进行修改

<Host name="localhost" appBase="webapps"

unpackWARs="true" autoDeploy="true">

<!-- SingleSignOn valve, share authentication between web applications

Documentation at: /docs/config/valve.html -->

<!--

<Valve className="org.apache.catalina.authenticator.SingleSignOn" />

-->

<!-- Access log processes all example.

Documentation at: /docs/config/valve.html

Note: The pattern used is equivalent to using pattern="common" -->

<Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"

prefix="localhost\_access\_log" suffix=".txt"

pattern="%h %l %u %t &quot;%r&quot; %s %b" />

<Context path ="/"

reloadable ="false"

docBase ="/usr/local/software/manager\_web/webapps/zmw-manager-web"/>

</Host>

cd /usr/local/software/manager\_web/bin

chmod u+x \*.sh 授权

./startup.sh 启动tomcat

tail -f /usr/local/software/manager\_web/logs/catalina.out 查看启动日志

#### Nginx下载编译

yum install -y pcre pcre-devel 使用pcre库中的正则表达式

yum install -y gcc-c++ 需要gcc对源码进行编译

yum install -y zlib zlib-devel 主要用于对http包的压缩

yum install -y openssl openssl-devel 密码算法库，主要用于https

cd /opt 下载到opt目录

wget [http://nginx.org/download/nginx-1.14.2.tar.gz](https://nginx.org/download/nginx-1.14.2.tar.gz) 下载nginx包

tar -zxvf nginx-1.14.2.tar.gz 解压

cd nginx-1.14.2

#### Front静态页面部署

mkdir /usr/local/software/nginx\_front\_web

修改zmw-front项目下的src/utils/myGlobal.js,修改正确的服务地址

const myBaseUrl = {

ucenter: 'http://192.168.52.131:8082',

front: 'http://192.168.52.131:8182'

};

通过npm run build 进行打包

./configure --prefix=/usr/local/software/nginx\_front\_web/ 选择安装目录

make & make install 编译安装

将打包好的dist文件复制到/usr/local/software/nginx\_front\_web/ 目录下

vim /usr/local/software/nginx\_front\_web/conf/nginx.conf

events

{

use epoll;

worker\_connections 1024; #单个后台worker process进程的最大并发链接数

}

#设定http服务器,利用它的反向代理功能提供负载均衡支持

http

{

include mime.types;

default\_type application/octet-stream;

#设定请求缓冲

server\_names\_hash\_bucket\_size 128;

client\_header\_buffer\_size 32K;

large\_client\_header\_buffers 4 32k;

# client\_max\_body\_size 8m;

sendfile on;

tcp\_nopush on;

tcp\_nodelay on;

#连接超时时间

keepalive\_timeout 65;

#开启gzip压缩，降低传输流量

gzip on;

gzip\_min\_length 1k;

gzip\_buffers 4 16k;

gzip\_http\_version 1.1;

gzip\_comp\_level 2;

gzip\_types text/plain application/x-javascript text/css application/xml;

gzip\_vary on;

#添加tomcat列表，真实应用服务器都放在这

upstream front\_web

{

server 192.168.52.202:80 weight=4 max\_fails=2 fail\_timeout=30s;server 192.168.52.203:80 weight=4 max\_fails=2 fail\_timeout=30s;

}

server

{

listen 9090; #监听端口

server\_name localhost;

#默认请求设置

location / {

proxy\_pass http://front\_web;

root /usr/local/software/nginx\_front\_web/dist;

index index.html index.htm;

try\_files $uri $uri/ /index.html;

}

}

}

192.168.52.202和 192.168.52.203 是tomcat

cd /usr/local/software/nginx\_front\_web/sbin

firewall-cmd --zone=public --add-port=9090/tcp --permanent

firewall-cmd --reload

chmod u+x nginx 授予可执行权限

./nginx 执行

tail -f /usr/local/software/nginx\_front\_web/logs/access.log 可查看访问日志

访问http://192.168.52.131:9090/

#### Manager静态页面部署

mkdir /usr/local/software/nginx\_manager\_web

修改zmw-front项目下的src/utils/myGlobal.js,修改正确的服务地址

const myBaseUrl = {

ucenter: 'http://192.168.52.131:8082',

front: 'http://192.168.52.131:8182',

manager: 'http://192.168.52.131:8282'

};

通过npm run build 进行打包

将打包好的dist文件复制到/usr/local/software/nginx\_manager\_web/ 目录下

./configure --prefix=/usr/local/software/nginx\_manager\_web/ 选择安装目录

make & make install 编译安装

vim /usr/local/software/nginx\_manager\_web/conf/nginx.conf

events

{

use epoll;

worker\_connections 1024; #单个后台worker process进程的最大并发链接数

}

#设定http服务器,利用它的反向代理功能提供负载均衡支持

http

{

include mime.types;

default\_type application/octet-stream;

#设定请求缓冲

server\_names\_hash\_bucket\_size 128;

client\_header\_buffer\_size 32K;

large\_client\_header\_buffers 4 32k;

# client\_max\_body\_size 8m;

sendfile on;

tcp\_nopush on;

tcp\_nodelay on;

#连接超时时间

keepalive\_timeout 65;

#开启gzip压缩，降低传输流量

gzip on;

gzip\_min\_length 1k;

gzip\_buffers 4 16k;

gzip\_http\_version 1.1;

gzip\_comp\_level 2;

gzip\_types text/plain application/x-javascript text/css application/xml;

gzip\_vary on;

#添加tomcat列表，真实应用服务器都放在这

upstream front\_web

{

server 192.168.52.202:80 weight=4 max\_fails=2 fail\_timeout=30s;server 192.168.52.203:80 weight=4 max\_fails=2 fail\_timeout=30s;

}

server

{

listen 9091; #监听端口

server\_name localhost;

#默认请求设置

location / {

proxy\_pass http://front\_web;

root /usr/local/software/nginx\_manager\_web/dist;

index index.html index.htm;

try\_files $uri $uri/ /index.html;

}

}

}

192.168.52.202和 192.168.52.203 是tomcat

cd /usr/local/software/nginx\_manager\_web/sbin

firewall-cmd --zone=public --add-port=9091/tcp --permanent

firewall-cmd --reload

chmod u+x nginx 授予可执行权限

./nginx 执行

tail -f /usr/local/software/nginx\_manager\_web/logs/access.log 可查看访问日志

访问http://192.168.52.131:9091/