软件系统分析与设计

**IShare-学术成果分享平台**

**部署安装文档**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **项目组成员信息** | | | | |
| **小组编号** | **3** | | | |
| **小组名称** | **就要整整齐齐队** | | | |
| **小组联系人** | **唐熙程** | | | |
| **学号** | **姓名** | **联系电话** | **本次实践中主要承担的工作内容** | **本次作业占比**  **(95%-105%之间)** |
| 20373809 | 周骁睿 | 18322377517 |  | 100% |
| 20373677 | 陈楚凡 | 18159211139 |  | 100% |
| 20373481 | 刘谕笑眉 | 18811596392 |  | 100% |
| 20373114 | 唐熙程 | 18873578949 |  | 100% |
| 20373571 | 吕新月 | 13305226276 |  | 100% |
| 20373565 | 黄瑞 | 15828398946 | 编写文档 | 100% |
| 20373661 | 徐亦佳 | 13857760431 |  | 100% |
| 20373721 | 洪秋訸 | 17398568289 |  | 100% |
| 20373420 | 张铭轩 | 18601000101 |  | 100% |
| 20373460 | 张博皓 | 17704295626 |  | 100% |
| 20372008 | 刘汉尧 | 18830788186 |  | 100% |

2022

版本变更历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 提交日期 | 主要编制人 | 审核人 | 版本说明 |
| 1.0 | 2022/12/17 | 黄瑞 |  | 完成了文档的初版 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目 录

[IShare-学术成果分享平台 1](#_Toc3475)

[1. 部署环境需求 1](#_Toc28481)

[1.1 最低配置 1](#_Toc20887)

[1.2 适宜配置 1](#_Toc23973)

[1.2.1 后端部署步骤 3](#_Toc1215)

[1.2.2 前端部署步骤 6](#_Toc16466)

# 部署环境需求

## 最低配置

本项目可以使用**较低**的配置进行部署，部署本项目的最低配置如下：

1. 一台4核16G内存，700G硬盘的云服务器。
2. 一个域名。
3. 一个开通了SMTP服务的邮箱。

需要注意的是，使用最低配置部署本项目时，服务器的初始文献资源为零，服务器响应时间可能较长，且不保证服务的可靠性等非功能性需求。

使用最低配置部署可以体验本项目的完整功能，但是在生产环境下**极力不推荐**使用最低配置进行部署。

在下文的部署步骤中，**将使用一台崭新的CentOS 7云服务器部署本项目，仅给出在最低配置下部署本项目的步骤**，若要在生产环境下部署本项目，请咨询专业的运维人员，或者发邮件联系我们。

另外，服务器的操作系统建议使用**CentOS 7**，服务器的443、8000端口需要配置为允许外网访问的状态。

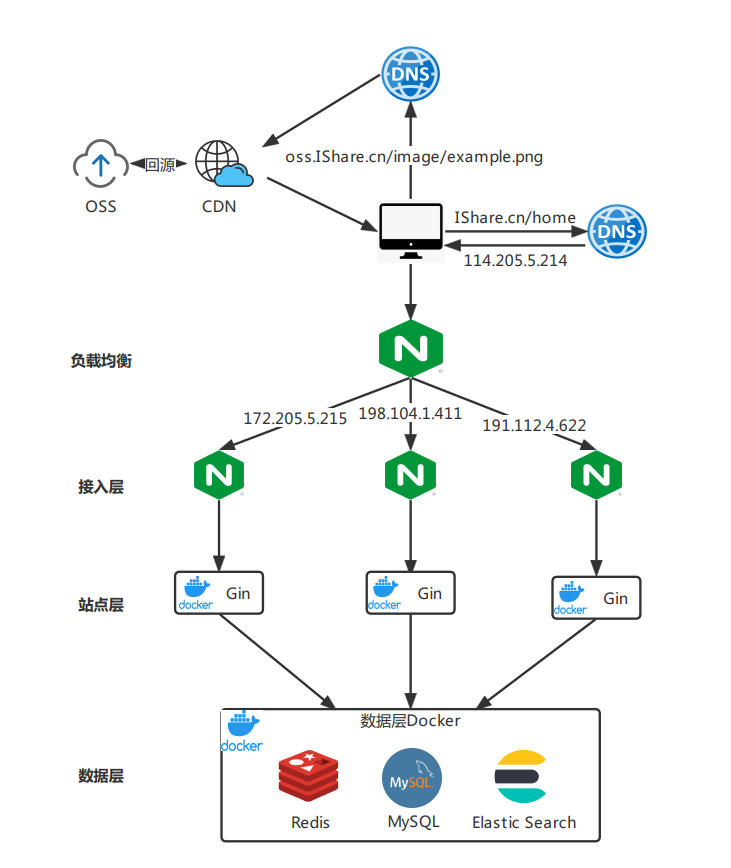
## 适宜配置

本项目的设计定位较高，于是本项目的适宜配置也较高，具体配置如下：

1. 云服务商的CDN服务或者全站加速服务，即动态加速、TCP加速、分发加速等。
2. 一个前端服务，使用Nginx进行均衡负载、反向代理等工作。
3. 一个后端服务，直接运行Gin应用，并记录日志等。
4. Elastic Search服务，负责向后端服务提供搜索引擎服务。
5. MySQL服务，负责向后端服务提供基础的数据存储服务。

上述服务最好都部署在同一个网络内，这样可以降低网络传输的时延。在本项目的开发环境中，我们使用部署在同一服务器上，并进行代码的编写与测试。

下面是本项目在适宜配置下的部署架构图，帮助读者更好地理解本项目的部署策略：



**图1-1 部署架构图**

### 后端部署步骤

**在CentOS 7服务器安装MySQL数据库**

[参考此文](https://www.cnblogs.com/smiles365/p/16296227.html)

**在CentOS 7服务器安装ElasticSearch引擎**

[参考此文](https://blog.csdn.net/wwwwwww31311/article/details/118342330)

**在 CentOS 7服务器安装Go环境**

* 安装wget工具

yum install wget -y

* 下载安装包

wget https://golang.google.cn/dl/go1.18.3.linux-amd64.tar.gz

* 解压安装包

tar -xzvf go1.18.3.linux-amd64.tar.gz

* 移动安装文件到自己指定的安装目录

mv go /usr/local/

* 设置环境变量

vim ~/.bashrc #这一步添加了安装目录和代码目录以及将可执行文件目录添加到环境变量

.bashrc

# .bashrc
  
  
# User specific aliases and functions
  
  
alias rm='rm -i'
  
alias cp='cp -i'
  
alias mv='mv -i'
  
  
# Source global definitions
  
if [ -f /etc/bashrc ]; then
  
 . /etc/bashrc
  
fi
  
PATH=/usr/local/mysql/bin:$PATH
  
PATH=/usr/local/mysql/bin:$PATH
  
PATH=/usr/local/mysql/bin:$PATH
  
  
# 安装目录
  
export GOROOT=/usr/local/go
  
# 代码目录
  
export GOPATH=~/code
  
export PATH=$PATH:$GOPATH:$GOROOT/bin

添加完后， 执行

source ~/.bashrc

* 检查 => 查看安装的go的版本

go version

**修改config.yml配置文件**

db:
  
 addr:
  
 port:
  
 user:
  
 password:
  
 dbname:
  
  
jwt:
  
 signature:
  
 expiresHours:
  
  
es:
  
 host:
  
 username:
  
 password:(如果没有配置密码可以不填)
  
 logpath:
  
 timeout:
  
  
avatar\_dir: "./media/avatars"
  
image\_dir: "./media/images"
  
temp\_dir: "./media/temp"
  
  
port: 8000

**fork仓库，添加部署脚本所需的仓库信息**

fork我们的[后端仓库](https://github.com/2022autumn/backend)，自行在仓库秘密中添加脚本需要的信息

name: deploy backend to HUAWEIYUN and runserver
  
on:
  
 #监听push操作
  
 push:
  
 branches:
  
 - main
  
 - dev
  
jobs:
  
 # 任务ID
  
 deploy:
  
 # 运行环境
  
 runs-on: ubuntu-latest
  
 # 步骤
  
 steps:
  
 # 使用别人的action
  
 - uses: actions/checkout@v2
  
 # 步骤名称
  
 # 命名这个任务为发布Deploy
  
 - name: build
  
 # 运行命令
  
 run: |
  
 go build -o IShare
  
 - name: executing remote ssh commands using password[delete old file]
  
 uses: appleboy/ssh-action@master
  
 with:
  
 host: ${{ secrets.HUAWEIYUN\_SERVER\_HOST }}
  
 username: ${{ secrets.HUAWEIYUN\_REMOTE\_USER }}
  
 key: ${{ secrets.HUAWEIYUN\_SERVER\_ACCESS\_TOKEN }}
  
 port: 22
  
 script: bash ${{ secrets.HUAWEIYUN\_SHELL\_SCRIPT\_ADDRESS }} delete
  
 - name: Deploy to HUAWEIYUN Server
  
 uses: easingthemes/ssh-deploy@v2.2.11
  
 with:
  
 # Private Key
  
 SSH\_PRIVATE\_KEY: ${{ secrets.HUAWEIYUN\_SERVER\_ACCESS\_TOKEN }}
  
 # Remote host
  
 REMOTE\_HOST: ${{ secrets.HUAWEIYUN\_SERVER\_HOST }}
  
 # Remote user
  
 REMOTE\_USER: ${{ secrets.HUAWEIYUN\_REMOTE\_USER }}
  
 # Remote port
  
 REMOTE\_PORT: 22
  
 # local directory
  
 SOURCE: "./IShare"
  
 # Target directory
  
 TARGET: ${{ secrets.HUAWEIYUN\_TARGET }}
  
 - name: executing remote ssh commands using password
  
 uses: appleboy/ssh-action@master
  
 with:
  
 host: ${{ secrets.HUAWEIYUN\_SERVER\_HOST }}
  
 username: ${{ secrets.HUAWEIYUN\_REMOTE\_USER }}
  
 key: ${{ secrets.HUAWEIYUN\_SERVER\_ACCESS\_TOKEN }}
  
 port: 22
  
 script: bash ${{ secrets.HUAWEIYUN\_SHELL\_SCRIPT\_ADDRESS }} start

**在服务器添加运行脚本和上文所述的config.yml配置文件**

restartGin.sh

# 脚本使用方法 bash restartGin.sh delete || bash restartGin.sh start
  
# 参数数目检查
  
if [ $# -ne 1 ]; then
  
 echo "Usage: $0 || $0 start"
  
 exit 1
  
fi
  
if [ $1 == "delete" ]; then
  
 rm -rf /root/backend/IShare
  
 pids=$(ps -ef | grep IShare | grep -v grep | awk '{print $2}')
  
 for pid in $pids; do
  
 kill -9 $pid
  
 done
  
 # docker rm -f gin
  
fi
  
if [ $1 == "start" ]; then
  
 cd /root/backend
  
 ./IShare >> ./IShare.log 2>&1 &
  
 if [ $? -eq 0 ]; then
  
 echo "IShare start success"
  
 else
  
 echo "IShare start failed"
  
 fi
  
 # docker run -d --name gin -p 8080:8080 -v /home/centos/gin:/gin gin
  
fi
  
exit 0

**在本地提交，完成自动部署**

git add .
  
git commit -m "deploy"
  
git push

### 前端部署步骤

**配置nginx**

下载nginx[参考此文](https://blog.csdn.net/Lee_yc/article/details/127867291)

修改default.conf相关配置

server {
  
 listen 443 ssl;
  
 server\_name ishare.horik.cn;
  
 client\_max\_body\_size 50m;
  
  
 ssl\_certificate /ssl/ishare.horik.cn.pem; # pem文件的路径
  
 ssl\_certificate\_key /ssl/ishare.horik.cn.key; # key文件的路径
  
 ssl\_session\_timeout 5m; #缓存有效期
  
 ssl\_ciphers ECDHE-RSA-AES128-GCM-SHA256:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL:!MD5:!ADH:!RC4; #加密算法
  
  
 ssl\_protocols TLSv1 TLSv1.1 TLSv1.2; #安全链接可选的加密协议
  
 ssl\_prefer\_server\_ciphers on; #使用服务器端的首选算法
  
  
 location / {
  
 root /root/frontend/dist;
  
 index index.htm index.html;
  
 try\_files $uri $uri/ /index.html; # 解决Vue单页面刷新404问题
  
 }
  
  
 location /api {
  
 proxy\_pass http://127.0.0.1:8000;
  
 proxy\_set\_header Host $host;
  
 proxy\_set\_header X-Real-IP $remote\_addr;
  
 proxy\_set\_header REMOTE-HOST $remote\_addr;
  
 proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;
  
 }
  
  
 error\_page 500 502 503 504 /50x.html;
  
 location = /50x.html {
  
 root /usr/local/nginx/html/50x.html;
  
 }
  
}
  
  
server {
  
 # HTTP 重定向（redireciton）。当用户以 HTTP 协议访问的时候，自动导向 HTTPS 保证安全。
  
 listen 80;
  
 server\_name ishare.horik.cn;
  
  
 location /api {
  
 proxy\_pass http://127.0.0.1:8000;
  
 proxy\_set\_header Host $host;
  
 proxy\_set\_header X-Real-IP $remote\_addr;
  
 proxy\_set\_header REMOTE-HOST $remote\_addr;
  
 proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;
  
 }
  
 #fastcgi\_intercept\_errors on;
  
 #return 301 https://$server\_name$request\_uri;
  
}
  
  
~
  
"default.conf" 49L, 1643C 32,1 All

**fork仓库，添加部署脚本所需的仓库信息**

fork我们的[前端仓库](https://github.com/2022autumn/frontend)，自行在仓库秘密中添加脚本需要的信息

name: Build app and deploy to HUAWEIYUN
  
on:
  
 #监听push操作
  
 push:
  
 branches:
  
 # mian分支，你也可以改成其他分支
  
 - master
  
 - dev
  
jobs:
  
 # 任务ID
  
 build:
  
 # 运行环境
  
 runs-on: ubuntu-latest
  
 # 步骤
  
 steps:
  
 # 使用别人的action
  
 - uses: actions/checkout@v2
  
 # 步骤名称
  
 - name: npm install
  
 # 步骤执行指令
  
 run: npm install
  
 - name: npm run build
  
 run: npm run build
  
 # 命名这个任务为发布Deploy
  
 - name: Deploy to HUAWEIYUN Server
  
 uses: easingthemes/ssh-deploy@v2.2.11
  
 with:
  
 # Private Key [public key should be add to the list of known hosts in .ssh of ALIYUN\_REMOTE\_USER] [pem format] [尝试放到root中]
  
 SSH\_PRIVATE\_KEY: ${{ secrets.HUAWEIYUN\_SERVER\_ACCESS\_TOKEN }}
  
 # Remote host
  
 REMOTE\_HOST: ${{ secrets.HUAWEIYUN\_SERVER\_HOST }}
  
 # Remote user
  
 REMOTE\_USER: ${{ secrets.HUAWEIYUN\_REMOTE\_USER }}
  
 # Remote port
  
 REMOTE\_PORT: 22
  
 # Source directory
  
 SOURCE: "dist/"
  
 # Target directory
  
 TARGET: ${{ secrets.HUAWEIYUN\_TARGET }}

**在本地提交，完成自动部署**

git add .
  
git commit -m "deploy"
  
git push