

需要环境

node 版本 $\geq 16.14.2$

golang 版本 $\geq v1.20$

Mysql 引擎需要是 innoDB

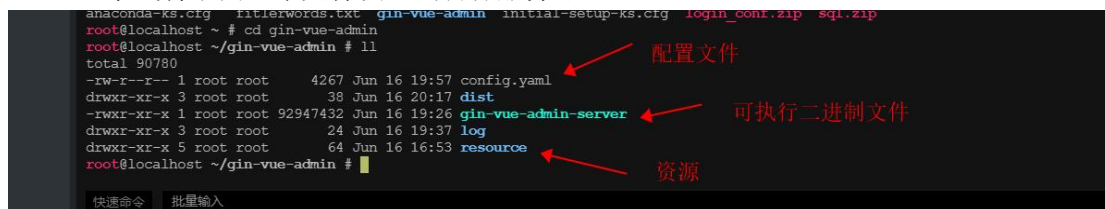
Redis 7.0.9

Elasticsearch 8.8.1

RocketMQ 5.102

后台服务端（端口默认 8888，需要安装好 mysql 和 redis）

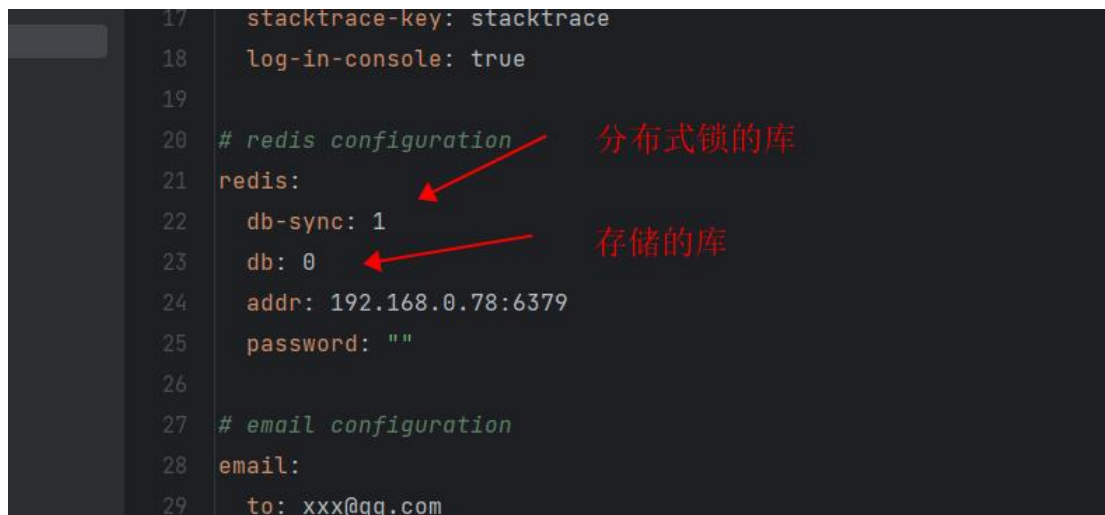
1. 跑流水线将下面三个文件发送到目标服务器



A terminal window showing a file transfer process. At the top, a list of files is shown: `anaconda-ks.cfg`, `litterwords.txt`, `gin-vue-admin`, `initial-setup-ks.cfg`, `login_conf.zip`, and `sql.zip`. Below this, the user runs `cd gin-vue-admin` and `ls -l`. The output shows a directory listing with permissions, owner, size, date, and filename. Red arrows point from Chinese labels to specific files: `config.yaml` is labeled '配置文件' (configuration file), `gin-vue-admin-server` is labeled '可执行二进制文件' (executable binary file), and `resource` is labeled '资源' (resource).

```
root@localhost ~ # cd gin-vue-admin
root@localhost ~/gin-vue-admin # ls -l
total 90780
-rw-r--r-- 1 root root 4267 Jun 16 19:57 config.yaml
drwxr-xr-x 3 root root 38 Jun 16 20:17 dist
-rwxr-xr-x 1 root root 92947432 Jun 16 19:26 gin-vue-admin-server
drwxr-xr-x 3 root root 24 Jun 16 19:37 log
drwxr-xr-x 5 root root 64 Jun 16 16:53 resource
root@localhost ~/gin-vue-admin #
```

2. Config.yaml 文件需填写 redis 的配置



A screenshot of a `config.yaml` file. The file contains configuration for `stacktrace-key`, `log-in-console`, `redis`, and `email`. Red arrows point from Chinese labels to specific configuration values: `db-sync: 1` is labeled '分布式锁的库' (distributed lock database), and `db: 0` is labeled '存储的库' (storage database).

```
17 stacktrace-key: stacktrace
18 log-in-console: true
19
20 # redis configuration
21 redis:
22   db-sync: 1
23   db: 0
24   addr: 192.168.0.78:6379
25   password: ""
26
27 # email configuration
28 email:
29   to: xxx@qq.com
```

需要填上日志库，不然启动不了

```

52         Content-Type
53         allow-credentials: true
54     db-list:
55         #游戏日志数据库
56         - disable: false # 是否禁用
57           type: "mysql" # 数据库的类型,目前支持mysql、pgsql、mssql、oracle
58           alias-name: "logsrv" # 数据库的名称,注意: alias-name 需要在db-list中唯一
59           path: "127.0.0.1"
60           port: "3306"
61           config: "charset=utf8mb4&parseTime=True&loc=Local"
62           db-name: "logsrv"
63           username: "root"
64           password: "123456"
65           max-idle-conns: 10
66           max-open-conns: 30
67           log-mode: "error"
68           log-zap: false
69     email:

```

3. 使用启动脚本

```
#!/bin/bash
```

```
# 定义变量
```

```
APP_PATH="/gin-vue-admin"    # Go 二进制文件的路径
```

```
APP_NAME="gin-vue-admin-server"    # Go 二进制文件的名称
```

```
# 检查命令行参数
```

```
if [ $# -ne 1 ]; then
    echo "Usage: $0 [start|stop]"
    exit 1
fi
```

```
# 根据命令行参数执行对应操作
```

```
if [ "$1" = "start" ]; then
    # 检查进程是否已经在运行
    PID=$(pgrep -f "$APP_NAME")
    if [[ -n "$PID" ]]; then
        echo "Error: $APP_NAME is already running with PID $PID"
        exit 1
    fi

```

```
# 启动 Go 二进制文件
```

```
cd "$APP_PATH" || exit
nohup "./$APP_NAME" > /dev/null 2>&1 &
echo "$APP_NAME started successfully"
```

```
elif [ "$1" = "stop" ]; then
```

```
# 检查进程是否在运行
```

```
PID=$(pgrep -f "$APP_NAME")
```

```

if [[ -z "$PID" ]]; then
    echo "Error: $APP_NAME is not running"
    exit 1
fi

# 终止进程
kill "$PID"
echo "$APP_NAME stopped successfully"

else
    echo "Invalid command: $1"
    echo "Usage: $0 [start|stop]"
    exit 1
fi
sh 文件名.sh start 启动

```

前端页面

1. 可以使用 nginx 的方式去部署，跑完流水线后将 **dist** 文件夹发送到目标服务器
添加 nginx 配置文件（按需修改）

```

server {
    listen      8081;
    server_name 192.168.0.78;

    location / {
        root /root/gin-vue-admin/dist;
        add_header Cache-Control 'no-store, no-cache, must-revalidate, proxy-revalidate,
max-age=0';
        try_files $uri $uri/ /index.html;
    }

    location /api {
        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-Proto $scheme;
        rewrite ^/api/(.*)$ /$1 break; #重写
        proxy_pass http://192.168.0.78:8881; # 设置代理服务器的协议和地址
    }

    location /api/swagger/index.html {
        proxy_pass http://192.168.0.78:8881/swagger/index.html;
    }
}

```

成功后访问 <http://192.168.0.78:8081/index.html>

GIN-VUE-ADMIN

初始化须知

- 1.您需有用一定的VUE和GOLANG基础
 - 2.请您确认是否已经阅读过官方文档
 - 3.请您确认是否了解后续的配置流程
 - 4.如果您使用mysql数据库，请确认数据库引擎为**innnoDB**
- 注：开发组不为文档中书写过的内容提供无偿服务

阅读文档

我已确认

点我已确认，然后按需填写数据库名（不可以输入已存在的数据库）。完成后点击初始化

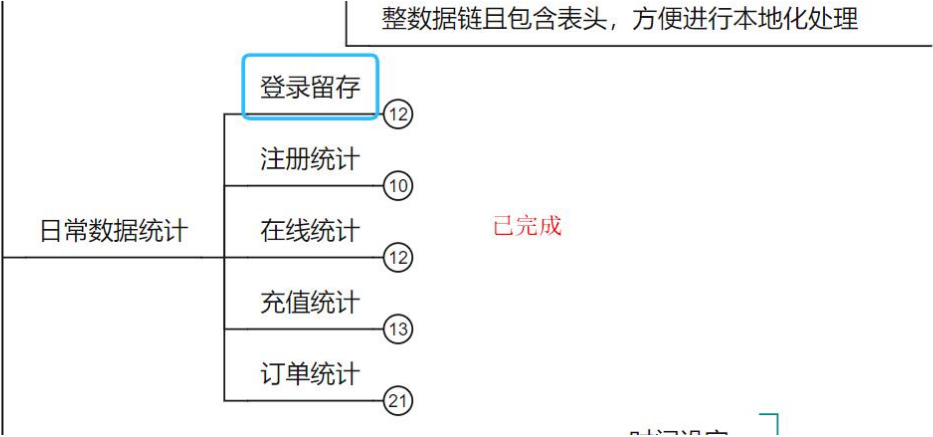
数据库类型	<input type="text" value="mysql"/>
host	<input type="text" value="127.0.0.1"/>
port	<input type="text" value="3306"/>
userName	<input type="text" value="root"/>
password	<input type="text" value="请输入数据库密码（没有则为空）"/>
dbName	<input type="text" value="gva"/>
<input type="button" value="立即初始化"/>	

日志消费

游戏日志是消费 MQ

但是打点需要 http 建议使用 nginx 代理

```
server {  
    listen      8081;  
    server_name 192.168.0.78;  
  
    location /log {  
        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-Proto $scheme;  
        proxy_connect_timeout 15s;  
        rewrite ^/log/(.*)$ /$1 break; #重写  
        proxy_pass http://server_list;  
    }  
}  
  
upstream server_list{  
    # 这个是 tomcat 的访问路径  
    server 192.168.0.52:8886;  
    server 192.168.0.78:8886;  
}
```



测试管理后台 Makefile 文件 docker-compose 部署
管理后台初始化，自动添加菜单权限等