

# 客服系统搭建运维手册

本系统采用**Golang Gin框架+GORM+MySQL+Vue+ElementUI**开发的独立高性能在线客服系统。

客服系统访客端支持PC端、移动端、小程序、公众号中接入客服，利用超链接、网页内嵌、二维码、定制对接等方式让网上所有通道都可以快速通过本系统联系到商家。

服务端可编译为二进制程序包，无需搭建环境即可运行，搭建过程比同类型PHP或Java客服系统要简单的多。

**自己域名 + 自己服务器 + 客服程序 = 私有云客服系统**

golang是编译型语言，最终是把代码编译为二进制可执行程序（类比windows下的exe），我们客服系统部署是不需要了解后端语言本身的，编译部署也不需要安装golang环境，常见的对接需求基本都可以实现。前端部分是非编译的vue+elementui，随便修改

## 1. 系统技术栈

Web服务：Nginx

数据库：MySQL + Qdrant(向量数据库，非必须)

前端：VueJS2 + ElementUI (CDN引入形式)

后端：Golang + Gin

## 2. 服务器配置

正是因为采用golang开发，我们系统对服务器配置没有任何要求，可以最低配服务器1核cpu 2G内存下高性能运行，我在线上就是1h 2G 5M，可支持几千客服账号，几万访客消息/日，并发上百。

日常情况服务器负载cpu和内存占用率均低于2%

查看cpu信息，1核

```
1 lscpu
```

```

Architecture:          x86_64
CPU op-mode(s):       32-bit, 64-bit
Byte Order:           Little Endian
Address sizes:        46 bits physical, 48 bits virtual
CPU(s):               1
On-line CPU(s) list: 0
Thread(s) per core:  1
Core(s) per socket:  1
Socket(s):            1
NUMA node(s):        1
Vendor ID:            GenuineIntel
CPU family:           6
Model:                94
Model name:           Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz
Stepping:             3
CPU MHz:              2394.374
BogoMIPS:             4788.74
Hypervisor vendor:    KVM
Virtualization type:  full
L1d cache:           32 KiB
L1i cache:           32 KiB
L2 cache:             4 MiB
L3 cache:             27.5 MiB
NUMA node0 CPU(s):   0
Vulnerability Itlb multihit: KVM: Vulnerable
Vulnerability L1tf:   Mitigation; PTE Inversion
Vulnerability Mds:    Vulnerable: Clear CPU buffers attempted, no microcode; SMT Host state unknown
Vulnerability Meltdown: Mitigation; PTI
Vulnerability Spec store bypass: Vulnerable
Vulnerability Spectre v1: Mitigation; usercopy/swapgs barriers and __user pointer sanitization
Vulnerability Spectre v2: Mitigation; Full generic retpoline, STIBP disabled, RSB filling
Vulnerability Srbds:  Unknown: Dependent on hypervisor status
Vulnerability Tsx async abort: Vulnerable: Clear CPU buffers attempted, no microcode; SMT Host state unknown
Flags:                fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse
                    opl cpuid tsc_known_freq pni pclmulqdq ssse3 fma cx16 pcid sse4_1 sse4_2
                    _lm abm 3dnowprefetch invpcid_single pti fsgsbase bmi1 hle avx2 smep
                    2bw avx512vl xsaveopt xsavec xgetbv1 arat

```

- 1 - \*\*CPU信息\*\*:
- 2 - 厂商: GenuineIntel
- 3 - CPU家族: 6
- 4 - 型号: 94
- 5 - 型号名称: Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz
- 6 - 步进: 3
- 7 - CPU频率: 2394.374 MHz
- 8 - 核心数: 1
- 9 - 线程数: 1
- 10 - 虚拟化类型: full (完全虚拟化)
- 11 - \*\*CPU缓存\*\*:
- 12 - L1数据缓存: 32 KiB
- 13 - L1指令缓存: 32 KiB
- 14 - L2缓存: 4 MiB
- 15 - L3缓存: 27.5 MiB
- 16 - \*\*系统信息\*\*:
- 17 - 架构: x86\_64
- 18 - 操作模式: 32位、64位
- 19 - 字节序: Little Endian
- 20 - 地址大小: 物理地址46位, 虚拟地址48位
- 21 - \*\*漏洞和脆弱性信息\*\*:
- 22 - 该服务器对一些漏洞和脆弱性有一定的防范措施和处理机制, 比如对Meltdown和Spectre漏洞的缓解措施。

查看内存，2G，程序运行占用的内存非常少

```
1 free -h
2
3          total        used        free        shared  buff/cache   available
4 Mem:    1.9Gi         799Mi        103Mi         11Mi         1.1Gi         1.0Gi
5 Swap:    1.0Gi         519Mi         505Mi
6 系统的物理内存总量为1.9GB，其中大约有1.1GB被用作缓存和缓冲，可用内存为1.0GB。交换空间总量为1.0GB，当前使用了约519MB。
```

请确认好自己Linux服务器镜像是 CentOS还是Ubuntu，这两个发行版下的客服程序依赖的系统库不兼容的。

最好使用ubuntu，因为ubuntu的仓库源软件都比较新，CentOS已经停止维护。

尽量不要使用阿里云或腾讯云自己改造的镜像版本，大部分是基于CentOS改造的，都缺斤少两，或者架构不兼容

编译环境和运行环境需要一致，这样才能把编译好的程序直接拿去运行。

区分CentOS还是Ubuntu以及具体发行版本

```
1 lsb_release -a
2
3 Distributor ID: Ubuntu
4 Description:    Ubuntu 20.04 LTS
5 Release:       20.04
6 Codename:      focal
```

运行占用CPU情况，cpu：1.8% 内存：7%，占用都非常低

```
1 htop
```

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
3128767	www	20	0	1050M	237M	14760	S	0.7	11.9	4h39:46	nginx: worker process
1766893	root	20	0	1151M	138M	22528	S	1.8	7.0	10:38.94	./kefu server -d
1218989	root	20	0	620M	15460	4748	S	0.0	0.0	7h19:24	barad_agent
4081557	root	20	0	982M	57964	15744	S	0.4	2.8	1h00:14	/usr/local/qcloud/YunJing/YDEyes/Y
4081520	root	20	0	982M	57964	15744	S	0.7	2.8	3h51:24	/usr/local/qcloud/YunJing/YDEyes/Y
2475546	root	20	0	9220	5076	3688	R	0.4	0.2	0:00.32	htop
1766895	root	20	0	1151M	138M	22528	S	1.1	7.0	2:05.56	./kefu server -d
839	bind	20	0	211M	5912	3184	S	0.0	0.3	15:11.50	/usr/sbin/named -f -u bind
715	bind	20	0	211M	5912	3184	S	0.0	0.3	16:22.05	/usr/sbin/named -f -u bind
1365	root	20	0	1253M	8680	928	S	0.0	0.4	15:03.02	/usr/bin/containerd
2718647	root	20	0	3650M	40604	2988	S	0.0	2.0	3h17:52	./qdrant
1219020	root	20	0	620M	15460	4748	S	0.0	0.8	2h13:52	barad_agent
4081522	root	20	0	982M	57964	15744	S	0.0	2.8	39:16.21	/usr/local/qcloud/YunJing/YDEyes/Y
4081617	root	20	0	982M	57964	15744	S	0.0	2.8	5:05.50	/usr/local/qcloud/YunJing/YDEyes/Y
2316939	root	20	0	982M	57964	15744	S	0.0	2.8	3:22.45	/usr/local/qcloud/YunJing/YDEyes/Y
751	root	20	0	1253M	8680	928	S	0.0	0.4	1h07:26	/usr/bin/containerd
2718689	root	20	0	3650M	40604	2988	S	0.0	2.0	1h14:57	./qdrant
1218988	root	20	0	46576	11016	3652	S	0.0	0.5	58:33.99	barad_agent
978	root	20	0	1253M	8680	928	S	0.0	0.4	25:32.52	/usr/bin/containerd
4081601	root	20	0	982M	57964	15744	S	0.7	2.8	12:31.05	/usr/local/qcloud/YunJing/YDEyes/Y
1230951	35002	20	0	2536M	110M	5112	S	0.0	5.6	3h46:33	java -Djava.awt.headless=true -cp

带宽情况，带宽可以适当高一些。

因为如果带宽太低，并发大时，网络请求会阻塞，进入不到系统内，并不是系统支持不了。

## 3. 如何购买服务器

因为我本身服务是运行在腾讯云中，所以这里以腾讯云服务器为例，演示如何购买一台云服务

### 3.1 注册账号

访问腾讯云官网以后，在登录地址处，可以扫码使用微信直接注册一个账号，并且完成实名认证。

这里要注意，想要有新用户优惠，实名认证的人也不能是老用户。

### 3.2 寻找产品

新用户会有一个购买一年或三年打2折的优惠，几百块买三年，优惠产品购买地址是

[https://cloud.tencent.com/act/pro/seckill\\_season?from=20210](https://cloud.tencent.com/act/pro/seckill_season?from=20210)

我选择的是两个三年的产品。具体轻量应用服务器和云服务器cvm的区别，问腾讯云客服。

配置	3年轻量 2核2G4M	3年轻量 2核4G5M
带宽	4M	5M
系统盘	50GB SSD盘	60GB SSD盘
月流量	300GB	500GB
时长	自购买3年内有效	自购买3年内有效
折扣	2.3折	2.5折
专享	个人专享	个人专享
限购	限1个	限1个
价格 (3年)	408元	628元
原价 (3年)	1800元	2520元
月租 (约)	11.33元/月	17.44元/月
抢购进度	已抢 20%	已抢 24.3%

比如选择：2核4G5M，系统盘60G，月流量500G

### 3.3 初始配置

需要先重置一下ssh密码

在通知信里可以看到重置密码的介绍

充值完成就可以使用ssh连接登录了，使用ssh密码进行登录。

### 3.4 使用ssh密钥登录 (免密，密钥登录，不会的忽略)

如果我们自己电脑window系统，可以使用git bash生成一下公钥

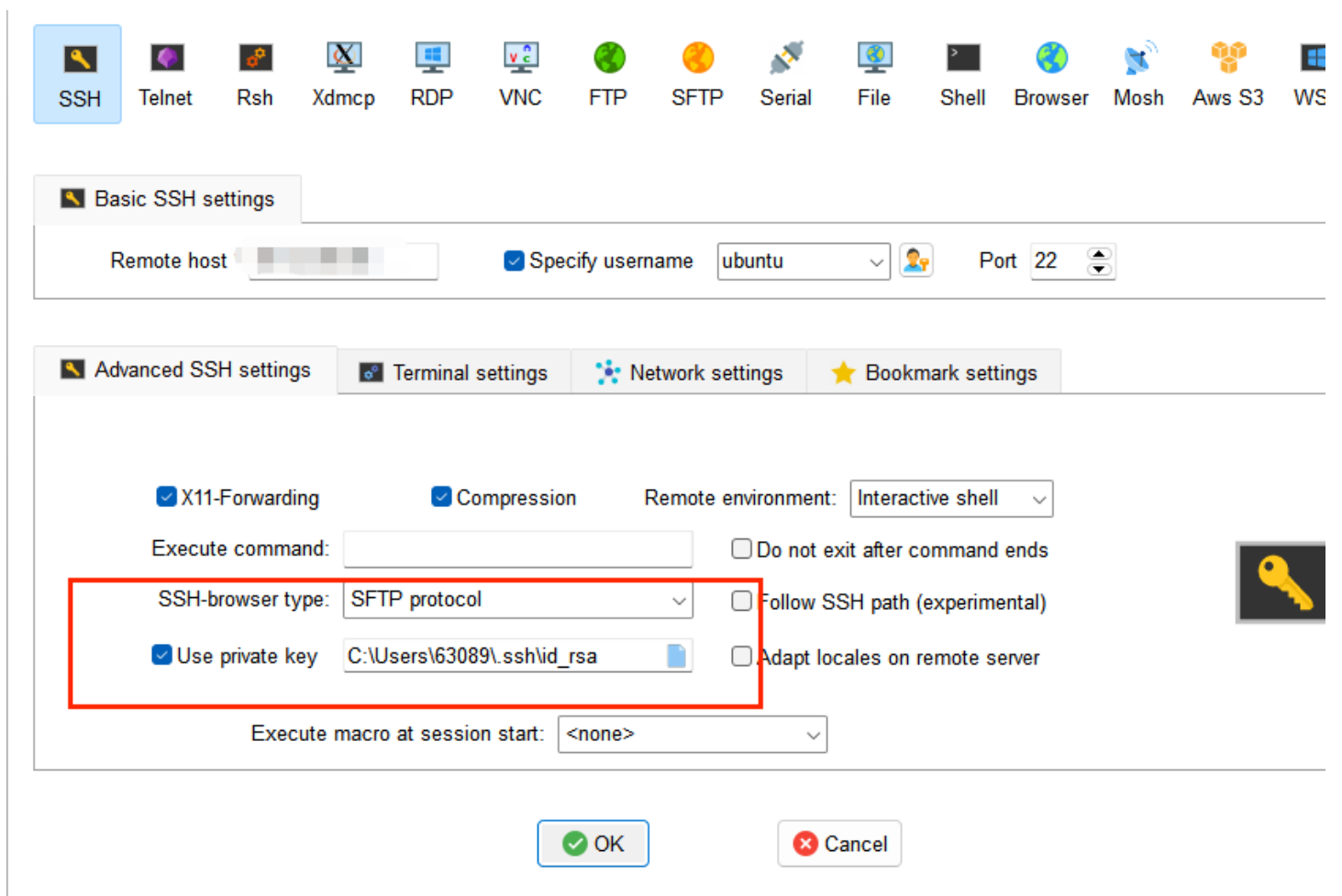
使用git bash打开后，执行以下命令，一路回车，就能生成自己的公钥

```
1 ssh-keygen
```

```
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/63089/.ssh/id_rsa):
Created directory '/c/Users/63089/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/63089/.ssh/id_rsa
Your public key has been saved in /c/Users/63089/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:ix910GxJLHHULocCF0mZmv1QbFeoLnE61xB3keYu2E8 63089@DESKTOP-UUKMP1E
The key's randomart image is:
+----[RSA 3072]-----+
|
|.oo.==+
|+ +=*..
|o B.*=0
|..@.oo+ o
|.@5* ooo
|O.+ +oE
|* .+
|..
+-----[SHA256]-----+
```

比如我的公钥的地址就位于 `/c/Users/63089/.ssh/id_rsa.pub`，需要把公钥内容配置到管理后台指定的地方。

我们的ssh客户端需要配置，使用私钥登录。每个人使用的ssh客户端都不一样，以自己的为准  
我的客户端配置如下，这个地方需要配置的是私钥文件的地址，这样就能免密码登录服务器了。

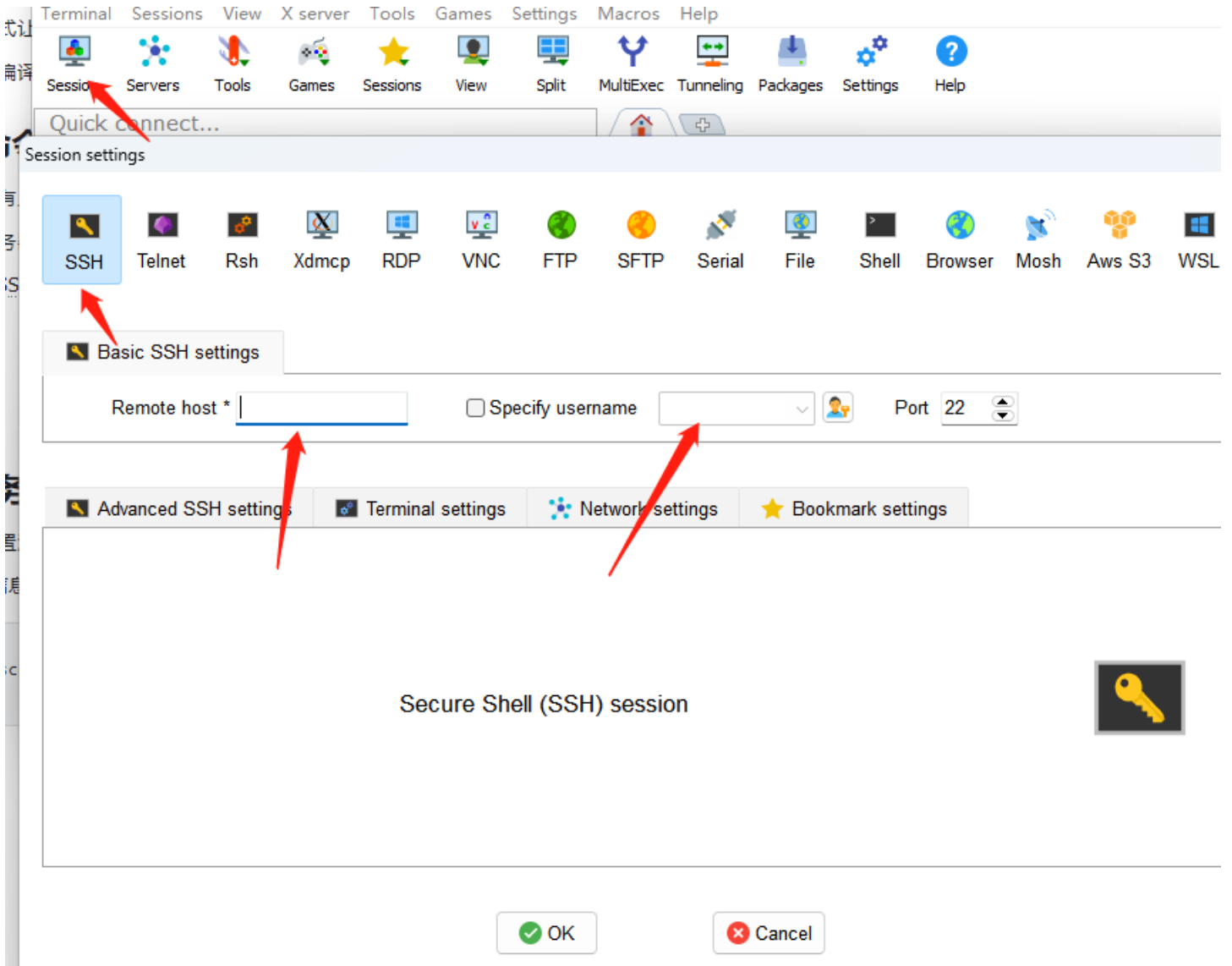


### 3.5 SSH命令行终端（密码登录）

购买完服务器后，会得到服务器IP，服务器SSH用户名和SSH密码

下载一个SSH终端，比如<https://mobaxterm.mobatek.net/download.html> 下载免费版就行

按下面的操作，输入IP，用户名，点OK进入终端后输入密码，点保存。



登录到服务器默认用户如果不是root，安装会有权限问题，比如ubuntu默认登录就不是root用户，执行下 `sudo -s` 可以切换为root

## 4. 如何安装宝塔面板

不用宝塔可以忽略这里，以及所有和宝塔相关的操作。只需要看如何导入数据库，如何启动服务那俩命令。以及如何配置nginx反向代理就行了。

### 4.1 ubuntu系统

直接在纯净新服务器上执行以下命令

```
1 wget -O install.sh https://download.bt.cn/install/install-ubuntu_6.0.sh &&  
sudo bash install.sh ed8484bec
```

## 4.2 CentOS系统

直接在纯净新服务器上执行以下命令

```
1 yum install -y wget && wget -O install.sh  
https://download.bt.cn/install/install_6.0.sh && sh install.sh ed8484bec
```

## 4.3 配置端口

外网面板地址: <https://服务器ip/42cc3158>

内网面板地址: <https://服务器ip/42cc3158>

username: xxxx

password: xxxx

If you cannot access the panel,

release the following panel port [31138] in the security group

若无法访问面板, 请检查防火墙/安全组是否有放行面板[31138]端口

因已开启面板自签证书, 访问面板会提示不匹配证书, 请参考以下链接配置证书

<https://www.bt.cn/bbs/thread-105443-1-1.html>

https自签证书, 可以视情况, 自己去导入。【面板设置】 【面板ssl】 【下载根证书】

## 4.4 放行端口

先前往腾讯云服务器后台, 放行端口, 【防火墙】 【添加规则】 端口号, 允许

**宝塔端口访问不了的, 一定要看看服务器安全组, 是否放行访问的端口号**

## 4.5 安装Nginx

部署web网站的时候, 必须安装web服务器, 我们选用的是nginx

直接在软件商店里找nginx就可以, 或者点网站的时候, 会自动出来让安装

## 4.6 安装数据库

数据库安装MySQL, 版本先不选8.0, 因为8.0的密码验证机制是不一样的, 担心各种语言的客户端没有更新验证机制

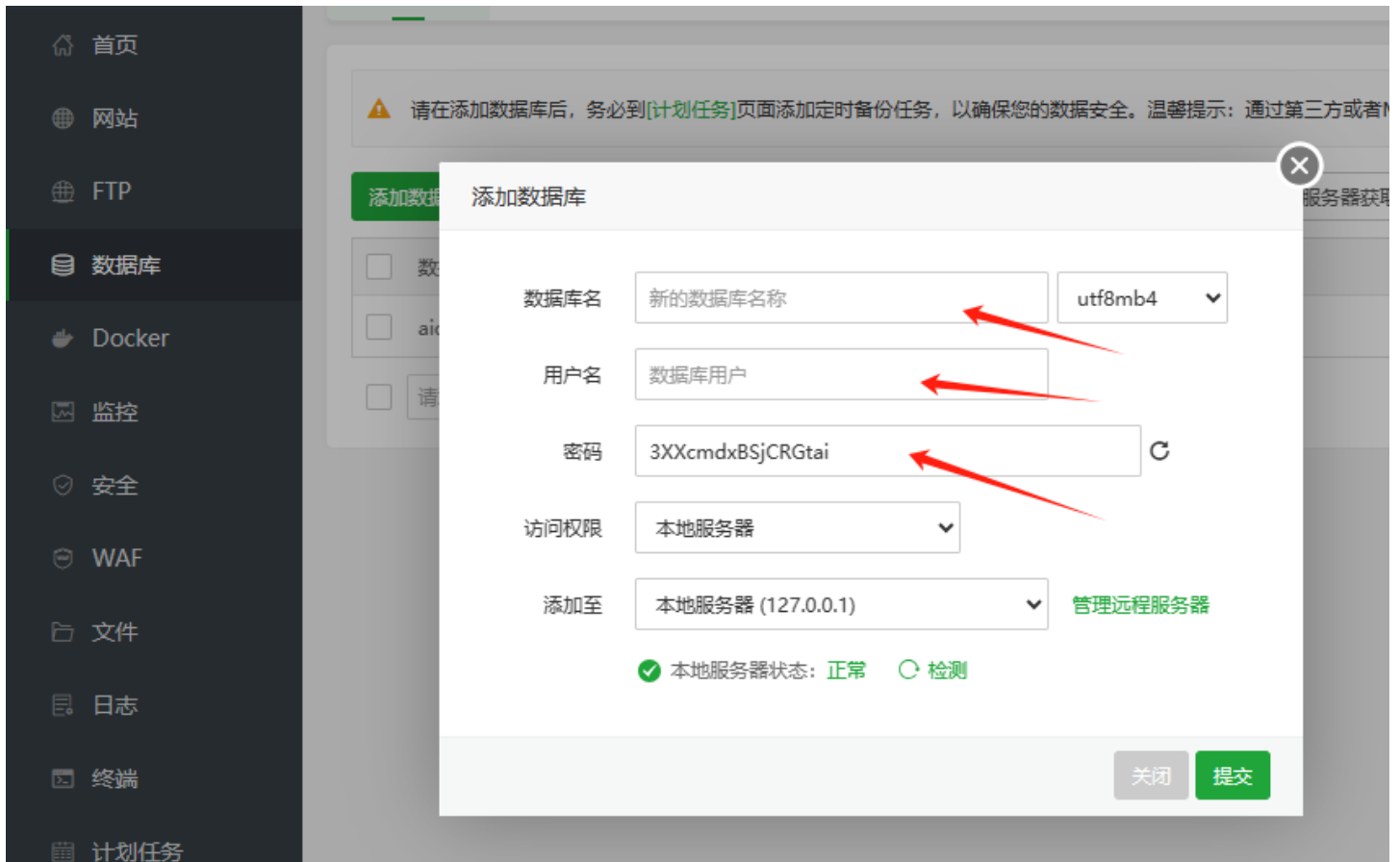
安装5.7版本就可以了

## 5. 客服部署流程



## 5.1 创建数据库

写上自己的数据库名称，用户名，密码。记住这几个参数



## 5.2 新建站点

进入宝塔面板，创建一个新的站点，新建站点的数据库。

可以选用默认的PHP站点，系统并没有使用PHP功能，无所谓PHP版本，仅仅只需要站点的Nginx域名绑定功能。

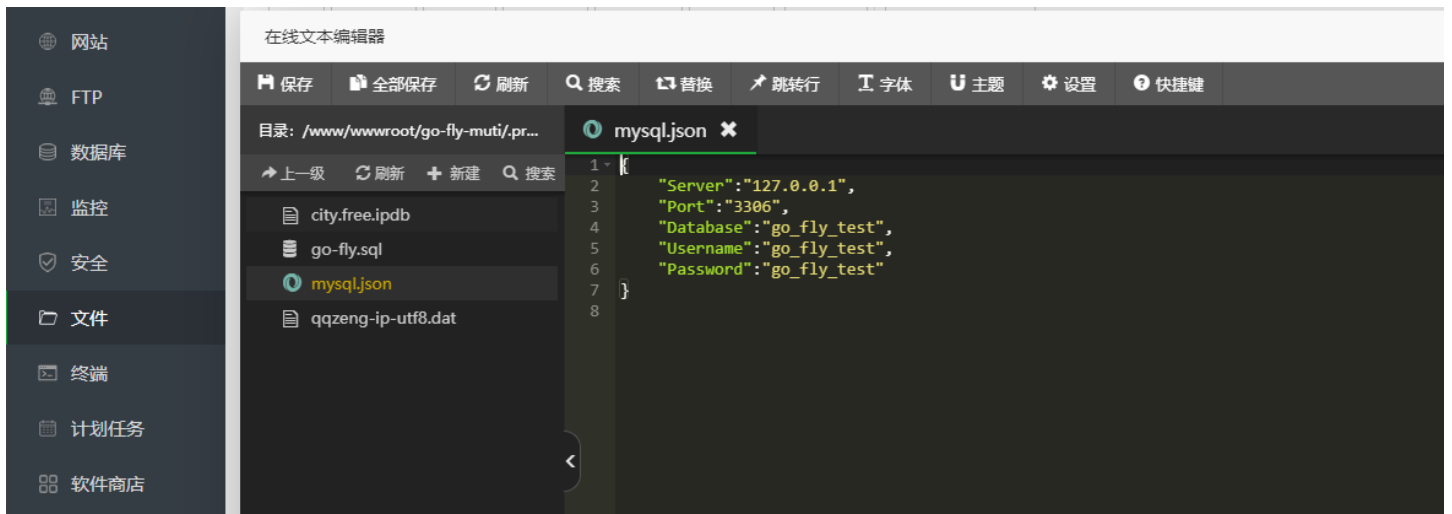
## 5.3 上传解压

通过宝塔的文件管理功能，进入站点的文件目录。

上传 最新版HelpDesk客服完整包 并解压到站点的当前目录，比如：/www/wwwroot/kefu1.v1kf.com，给此目录的所有文件赋权限777

## 5.4 配置并安装

编辑config/mysql.json，修改为创建数据库时设置的数据库名，用户名，密码



## 5.5 导入数据库

在终端切换目录到 /www/wwwroot/kefu.xxxx.com

```
cd /www/wwwroot/kefu.xxxx.com
```

执行导入数据库命令,确保数据库配置信息正确,自动创建导入数据库相关表

```
1 ./kefu install
```

启动服务,执行开启服务命令,默认监听8081端口,-d为守护进程模式 -p 可以指定监听的端口

```
1 ./kefu server -d
```

关闭服务

```
1 killall kefu
```

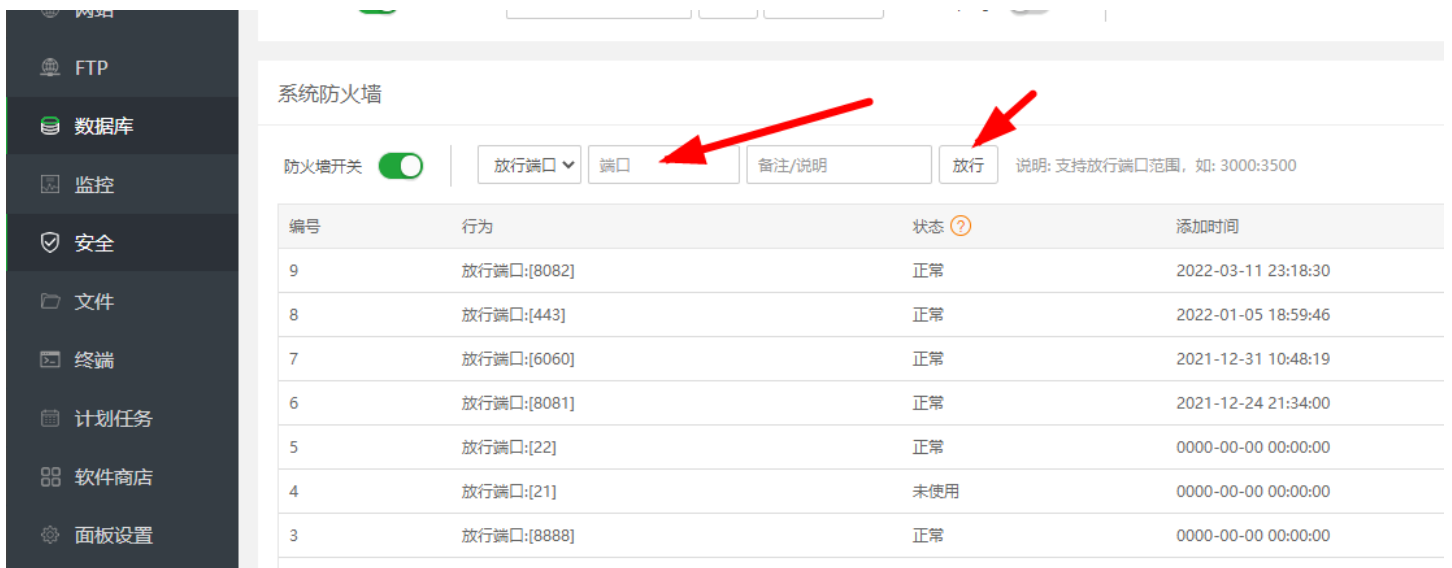
使用ps命令查看进程,确保所有进程已经关闭了。例如: ps aux|grep kefu

重启服务

```
1 killall kefu;killall kefu;./kefu server -d
```

## 5.6 开放端口

到宝塔安全页面将8081端口开放

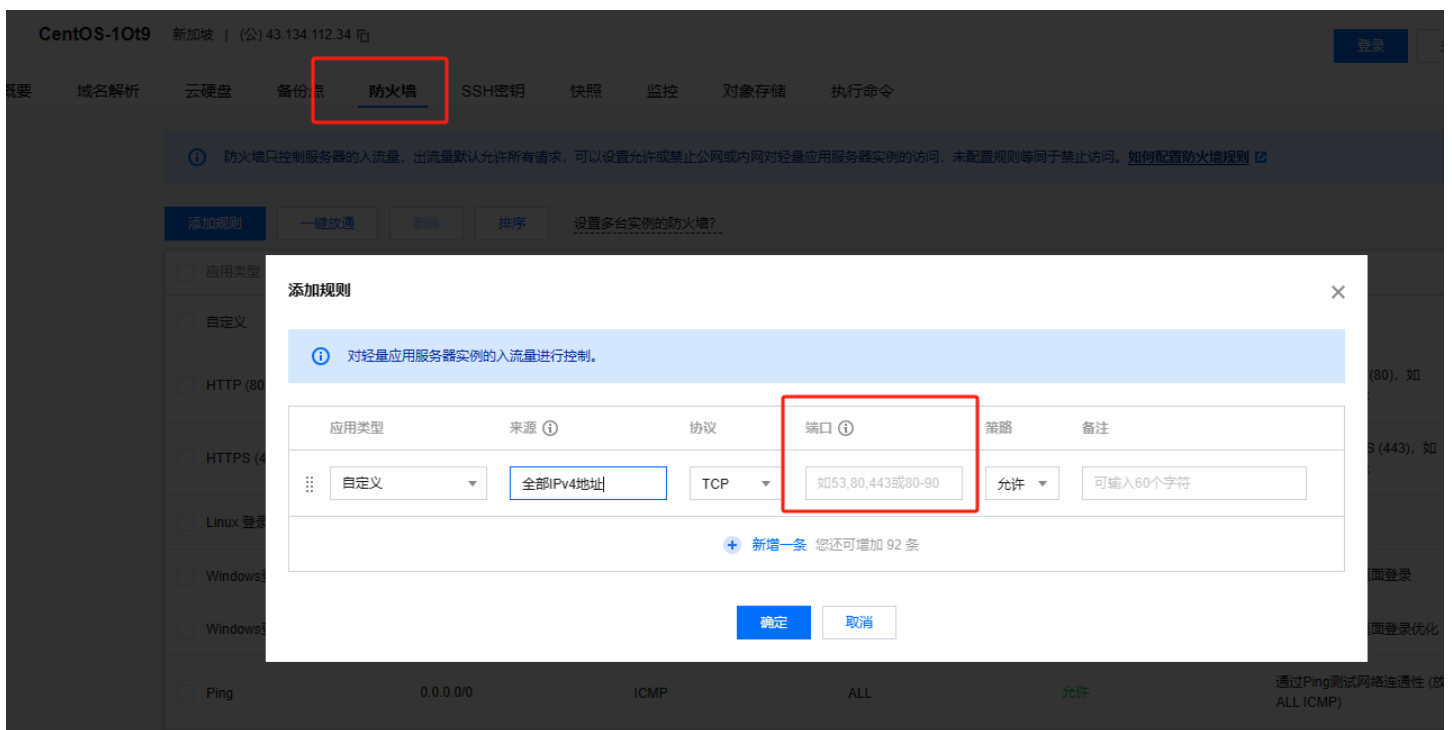


如果是阿里云/腾讯云等云服务器，则安全组也需要设置下让8081端口可以被外网访问。

此时可以访问以下地址，检测服务是否启动成功，端口是否可访问

<http://公网ip:8081/main admin 123>

例如下面是腾讯云轻量应用服务器，防火墙开放端口位置



## 5.7 配置域名

下面是nginx的配置文件，如果用的是宝塔前往宝塔后台【网站】==》【站点】==》【设置】===》【伪静态】，复制粘贴下面的配置

配置完成后可以访问项目网址 <http://域名>

```
1 #PROXY-START/
```

```

2
3 location /{
4     try_files $uri @kefu;
5 }
6 location @kefu {
7     proxy_read_timeout 1200s;
8     proxy_buffering off;
9     proxy_pass http://127.0.0.1:8081;
10    proxy_set_header X-Forwarded-Proto $scheme;
11    proxy_set_header Host $host;
12    proxy_set_header X-Real-IP $remote_addr;
13    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
14    proxy_set_header REMOTE-HOST $remote_addr;
15
16    add_header X-Cache $upstream_cache_status;
17    proxy_set_header Upgrade $http_upgrade;
18    proxy_set_header Connection "upgrade";
19
20    #Set Nginx Cache
21    add_header Cache-Control no-cache;
22 }
23 #PROXY-END/

```

1. 如果请求的是静态资源（例如文件），则尝试Nginx直接返回该资源。
2. 如果文件不存在，则将请求转发到命名位置 `@kefu`，从而进行反向代理到后端服务或者处理其他逻辑。这样可以减轻后端反代服务的压力。

### 3. 宝塔面板的伪静态

宝塔的伪静态功能

最终会把这段配置引入到Nginx每个站点vhost的配置中，那么我们也可以借助这个来写反向代理。

这样就不需要去反向代理那里配置了，省的麻烦。

**反向代理一定要支持websocket，很多自行配置nginx不看文档的，注意要传递两个反代头的**

```

1     proxy_set_header Upgrade $http_upgrade;
2     proxy_set_header Connection "upgrade";

```

域名管理

子目录绑定

网站目录

访问限制

流量限制

伪静态

默认文档

配置文件

SSL

PHP版本

Composer

Tomcat

重定向

反向代理

0.当前

规则转换工具: Apache转Nginx

```
1 # PROXY_PASS
2 location /
3 {
4     try_files $uri @kefu;
5 }
6
7 location @kefu {
8     proxy_buffering off;
9     proxy_read_timeout 1200s;
10    proxy_pass http://127.0.0.1:8081;
11    proxy_set_header X-Forwarded-Proto $scheme;
12    proxy_set_header Host $host;
13    proxy_set_header X-Real-IP $remote_addr;
14    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
15    proxy_set_header REMOTE-HOST $remote_addr;
16
17    add_header X-Cache $upstream_cache_status;
```

保存

另存为模板

- 请选择您的应用，若设置伪静态后，网站无法正常访问，请尝试设置回default
- 您可以对伪静态规则进行修改，修改完后保存即可。

## 5.8 配置HTTPS

前往宝塔面板后台 【站点】 ==> 【SSL】 ==> 【Let's Encrypt】 可以申请免费的ssl证书

配置完成后可以访问https项目网址 <https://域名>

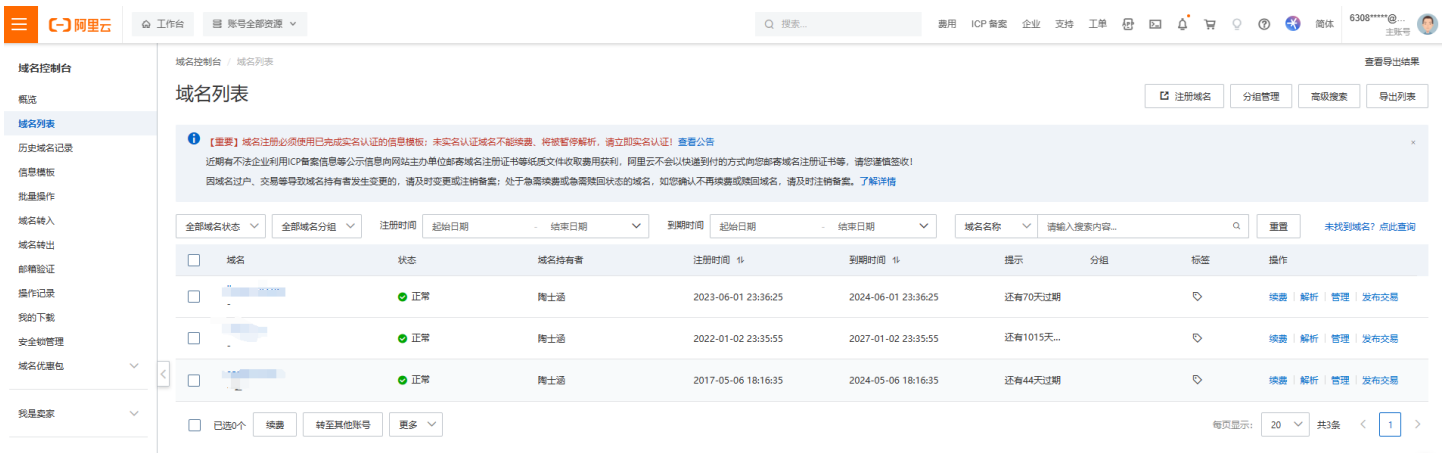
<https://域名/main> admin 123

阿里云数字证书管理服务，可以购买付费的SSL证书，也可以申请短期有效的免费证书



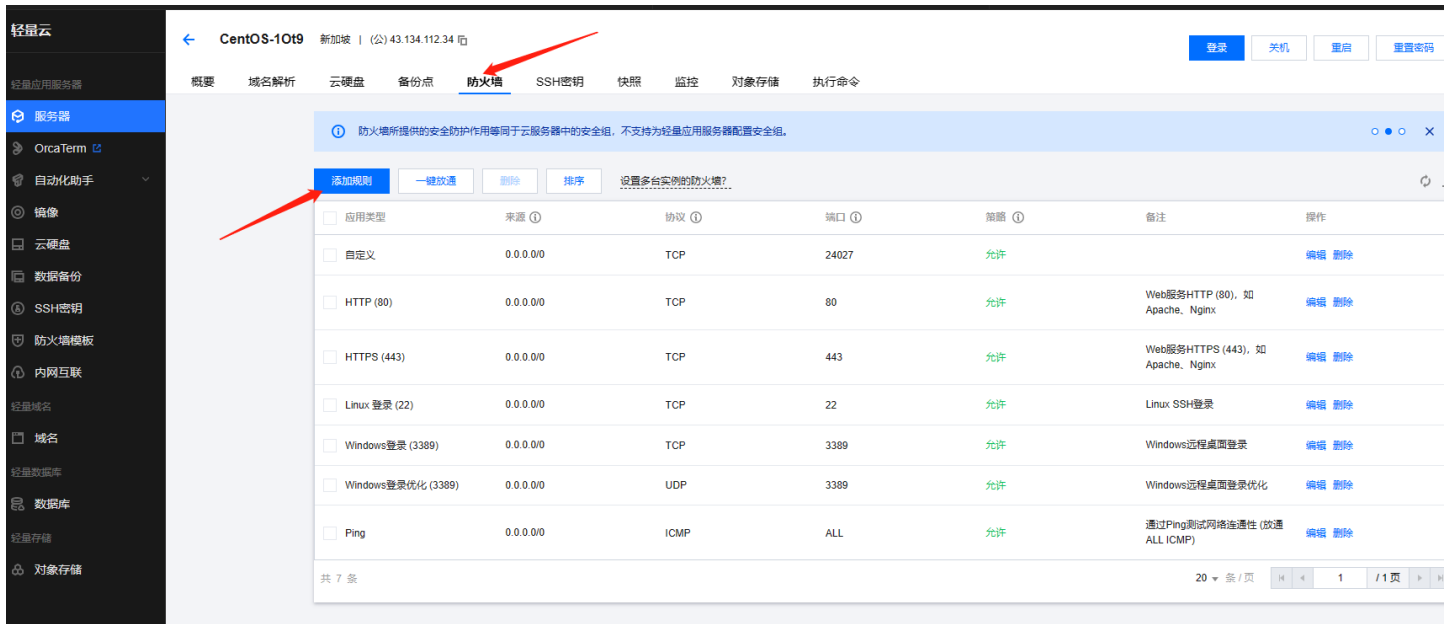
## 6. 域名解析流程

登录阿里云，找到域名控制台，域名列表，点后面的解析



## 7. 腾讯云轻量服务器放行端口

腾讯云服务器防火墙或安全组放行端口设置，当你需要使用IP+端口形式访问服务，那么需要在下面云服务器放行端口



## 8. 服务器空间清理

当系统运行时间长了以后，服务器的空间存储可能会占用过多，可以使用下面的命令逐步排查空间存储问题

### 8.1 查看当前存储大小

```
1 df -h
2
3 主要关注的是根目录 / 的存储剩余情况
4 下面的表示，总存储50G，已使用45G，剩余3.1G
5 /dev/vda2      50G   45G   3.1G   94% /
```

### 8.2 查看具体目录存储大小

比如：查看/www目录的存储大小

```
1 du -h --max-depth=1 /www | sort -hr | head -n 20
2
3 /www目录总共占用31G
4 /www/wwwroot 程序目录占用15G
5 /www/wwwlogs 日志目录占用13G，可以清理一下
6 31G      /www
7 15G      /www/wwwroot
8 13G      /www/wwwlogs
9
```

清理 /www/wwwlogs 日志目录，下面都是nginx的access日志和error日志

- 1 查看这个目录下文件和，文件的大小
- 2 -l：表示以长格式列出信息，包括文件的详细信息。
- 3 -S：表示按文件大小排序。
- 4 -h：表示以易读的格式显示文件大小（例如 KB、MB、GB）。
- 5
- 6 `ls -lSh`
- 7
- 8 清空文件
- 9 `echo "" > xxxx.log`

### 8.3 清理客服系统目录

客服系统程序上传目录为：程序目录/static/upload/，可以适当清理

客服系统程序日志：程序目录/logs `echo "" > kefu.log`

## 9. Golang环境安装（编译版客户可以忽略此处）

针对客服系统全源码的客户，源码开发或运行，需要配置Golang运行环境

Golang语言为跨平台的开发语言，当您在本地运行客服系统，或者在服务器运行编译客服系统源码，则需要搭建Golang运行环境，并且开启go module依赖管理

### Windows系统

首先下载golang压缩包，在下面这个地址下载

<https://studygolang.com/dl>

找到windows的安装包，msi的就可以，这样下一步下一步的直接就把环境变量配好了

在cmd里输入

```
go version
```

### Linux系统

下面的脚本是amd64架构的Golang包，如果你是arm架构或32位的服务器，需要选择arm架构的Golang包

<https://studygolang.com/dl>

解压源码压缩包目录，里面有个install\_go.sh的shell脚本，这是我写好的安装golang环境的脚本。

- 1 `wget https://studygolang.com/dl/golang/go1.20.2.linux-amd64.tar.gz`



```
2 tar -C /usr/local -xvf go1.20.2.linux-amd64.tar.gz
3 mv go1.20.2.linux-amd64.tar.gz /tmp
4 echo "PATH=\$PATH:/usr/local/go/bin" >> /etc/profile
5 echo "PATH=\$PATH:/usr/local/go/bin" >> ~/.bashrc
6 source /etc/profile
7 go version
8 go env -w GO111MODULE=on
9 go env -w GOPROXY=https://goproxy.cn,direct
```

直接执行 `sh install_go.sh`

检查golang环境是否安装成功，执行 `go version`

使环境变量生效

`source /etc/profile`

## 开启go module依赖管理

`go env -w GO111MODULE=on`

`go env -w GOPROXY=https://goproxy.cn,direct`

## Mac环境

参照golang中文网提供的golang安装包进行配置

## 查看CPU的系统架构

根据自己服务器的系统架构选择对应的Golang程序包

```
1 uname -m
```

或

```
1 lscpu
```

386的是32位的，需要选386的包

x86\_64的是 64位的，amd64和x86\_64 含义可认为相同，需要选amd64的包

arm64 或 aarch64 是arm体系的64位的，需要选arm64的包

```
root@VM-16-3-ubuntu:/home/ubuntu# uname -m
x86_64
root@VM-16-3-ubuntu:/home/ubuntu# lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Address sizes:         46 bits physical, 48 bits virtual
Byte Order:            Little Endian
CPU(s):                2
On-line CPU(s) list:   0,1
Vendor ID:             GenuineIntel
Model name:            Intel(R) Xeon(R) Gold 6133 CPU @ 2.50GHz
CPU family:            6
Model:                 94
Thread(s) per core:    1
Core(s) per socket:    2
Socket(s):             1
Stepping:              3
BogoMIPS:              4988.26
Flags:                 fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge
                      i pclmulqdq ssse3 fma cx16 pcid sse4_1 sse4_2 x2apic
                      e avx2 smep bmi2 erms invpcid rtm mpx rdseed adx sm
```

## 编译源码

在程序目录执行

go build -o kefu 可以在当前程序目录生成出可执行二进制文件 kefu

其他操作参照上一节中内容

也可以不生成二进制文件，直接运行源码

go run kefu.go install 导入数据库

go run kefu.go server 启动服务

## 10. Docker服务安装配置（仅客服系统的客户可以忽略此处）

部署知识库服务时，需要有Docker服务安装向量数据库等软件

### 10.1 一键安装docker

```
1 curl -fsSL https://get.docker.com -o get-docker.sh
2 sh get-docker.sh --mirror=Aliyun
```

### 10.2 centos安装

```
1 sudo yum remove docker \
2 docker-client \
```

```
3         docker-client-latest \  
4         docker-common \  
5         docker-latest \  
6         docker-latest-logrotate \  
7         docker-logrotate \  
8         docker-engine  
9 sudo yum install -y yum-utils  
10 sudo yum-config-manager --add-repo  
    https://download.docker.com/linux/centos/docker-ce.repo  
11 sudo yum install docker-ce docker-ce-cli containerd.io docker-buildx-plugin  
    docker-compose-plugin
```

## 10.3 ubuntu安装

```
1 # Add Docker's official GPG key:  
2 sudo apt-get update  
3 sudo apt-get install ca-certificates curl  
4 sudo install -m 0755 -d /etc/apt/keyrings  
5 sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o  
    /etc/apt/keyrings/docker.asc  
6 sudo chmod a+r /etc/apt/keyrings/docker.asc  
7  
8 # Add the repository to Apt sources:  
9 echo \  
10 "deb [arch=$(dpkg --print-architecture) signed-  
    by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \  
11 $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \  
12 sudo tee /etc/apt/sources.list.d/docker.list > /dev/null  
13 sudo apt-get update  
14 sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-  
    plugin docker-compose-plugin
```

### 配置阿里云镜像

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

## 10.4 Docker Compose V2 安装

- 官方文档: [Install using the repository](#)
- 当前版本: v2.21.0

- 使用的语法变了，用 docker compose 来运行命令，去掉了 -，详情可以看[迁移指南](#)

- 安装

- Ubuntu and Debian

```
1 $ sudo apt-get update
2 $ sudo apt-get install docker-compose-plugin
```

- RPM-based distros

```
1 $ sudo yum update
2 $ sudo yum install docker-compose-plugin
```

- `docker compose xxxx` 同之前的docker-compose
- `docker-compose scale` 需要修改 `docker compose up --scale`
- `docker-compose config` 命令用于验证和查看 Docker Compose 文件的配置
- 使用示例：[官方示例](#)

## 10.5 docker的基本命令

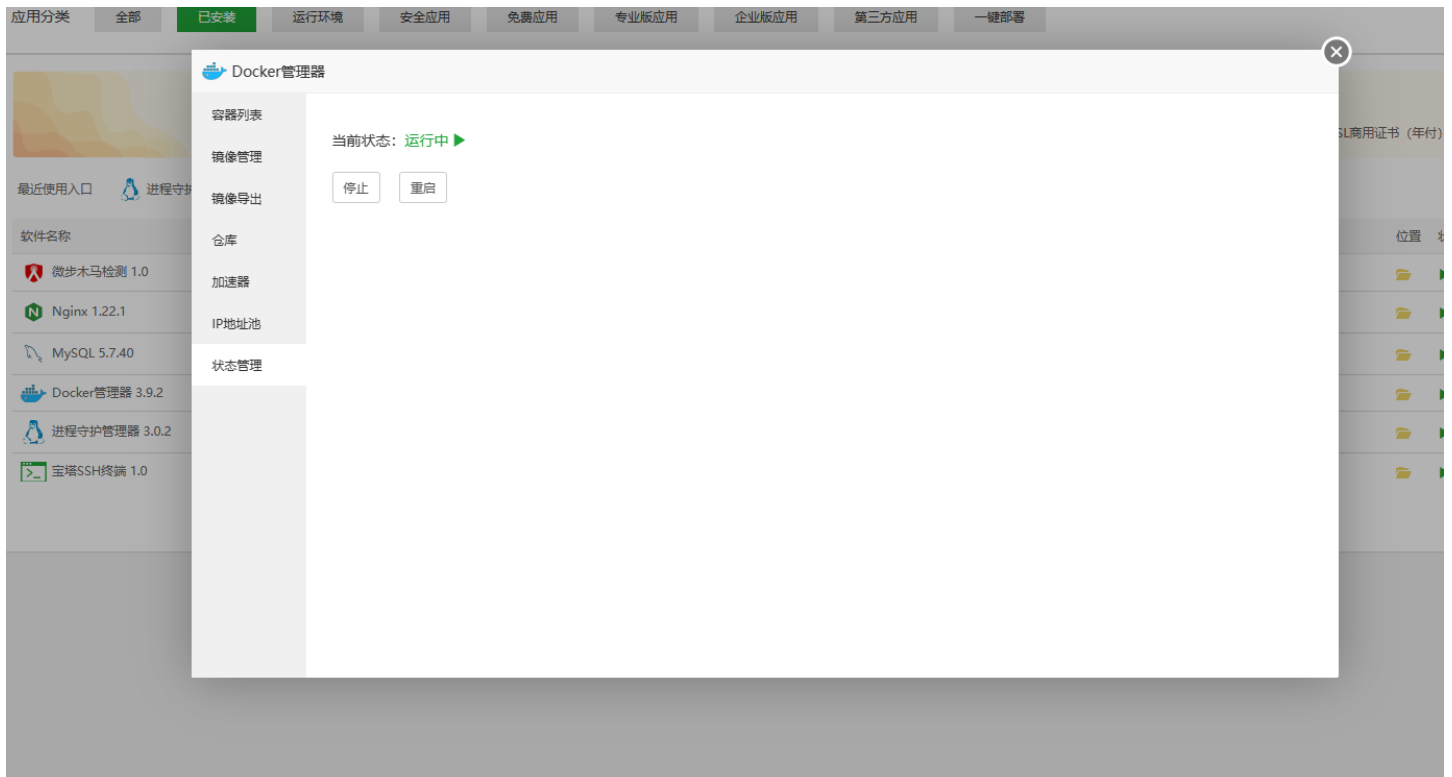
```
1 //列出镜像
2 docker images -a
3 //列出容器
4 docker ps -a
5 //查看容器日志
6 docker logs -f 容器ID
```

## 10.6 宝塔面板下安装docker

如果你使用宝塔面板，那么就简单了，提供可视化的界面进行管理



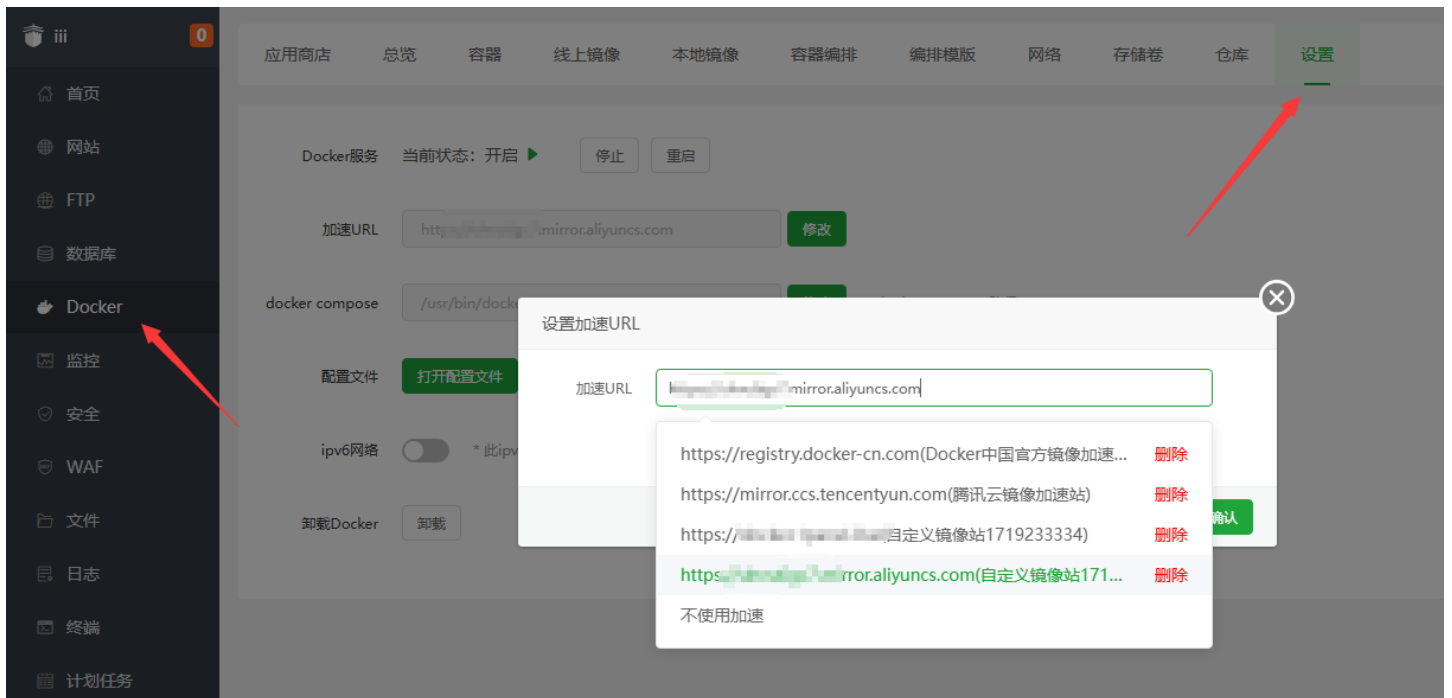
在【软件商店中】安装docker管理器即可



## 10.7 宝塔面板Docker仓库加速

由于国内网络原因，docker在使用官方仓库的时候，会连不上，拉取镜像失败。

可以在【设置】那里填写加速URL <https://docker.1ms.run>



## 11. OpenAI接口相关

开发知识库GPT必须要了解的概念，以及必须了解的OpenAI的两个接口，文本转向量和文本生成

文本转向量：将一个文本字符串转换成向量数据（浮点数的数组）

文本生成：我们常说的大模型聊天接口

### 11.1 文本转向量接口

Embedding models 嵌入模型，OpenAI下面三个模型

text-embedding-3-small

text-embedding-3-large

text-embedding-ada-002

接口使用案例

```
1 curl https://api.openai.com/v1/embeddings \
2   -H "Content-Type: application/json" \
3   -H "Authorization: Bearer $OPENAI_API_KEY" \
4   -d '{
5     "input": "Your text string goes here",
6     "model": "text-embedding-3-small"
7   }'
```

### 11.2 文本生成接口

```
1 curl https://api.openai.com/v1/chat/completions \  
2 -H "Content-Type: application/json" \  
3 -H "Authorization: Bearer $OPENAI_API_KEY" \  
4 -d '{  
5   "model": "gpt-3.5-turbo",  
6   "messages": [  
7     {  
8       "role": "system",  
9       "content": "You are a helpful assistant. 【知识库】 "  
10    },  
11    {  
12      "role": "user",  
13      "content": "Who won the world series in 2020?"  
14    },  
15    {  
16      "role": "assistant",  
17      "content": "The Los Angeles Dodgers won the World Series in 2020."  
18    },  
19    {  
20      "role": "user",  
21      "content": "Where was it played?"  
22    }  
23  ]  
24  }'
```

## 12. Qdrant向量数据库安装配置

知识库服务依赖该数据库，Embedding形式个性化训练ChatGPT，必不可少的就是向量数据库  
因为qdrant向量数据库只支持Docker部署，所以需要先安装好Docker服务。

(仅客服系统的客户可以忽略此处)

### 命令行安装

拉取镜像

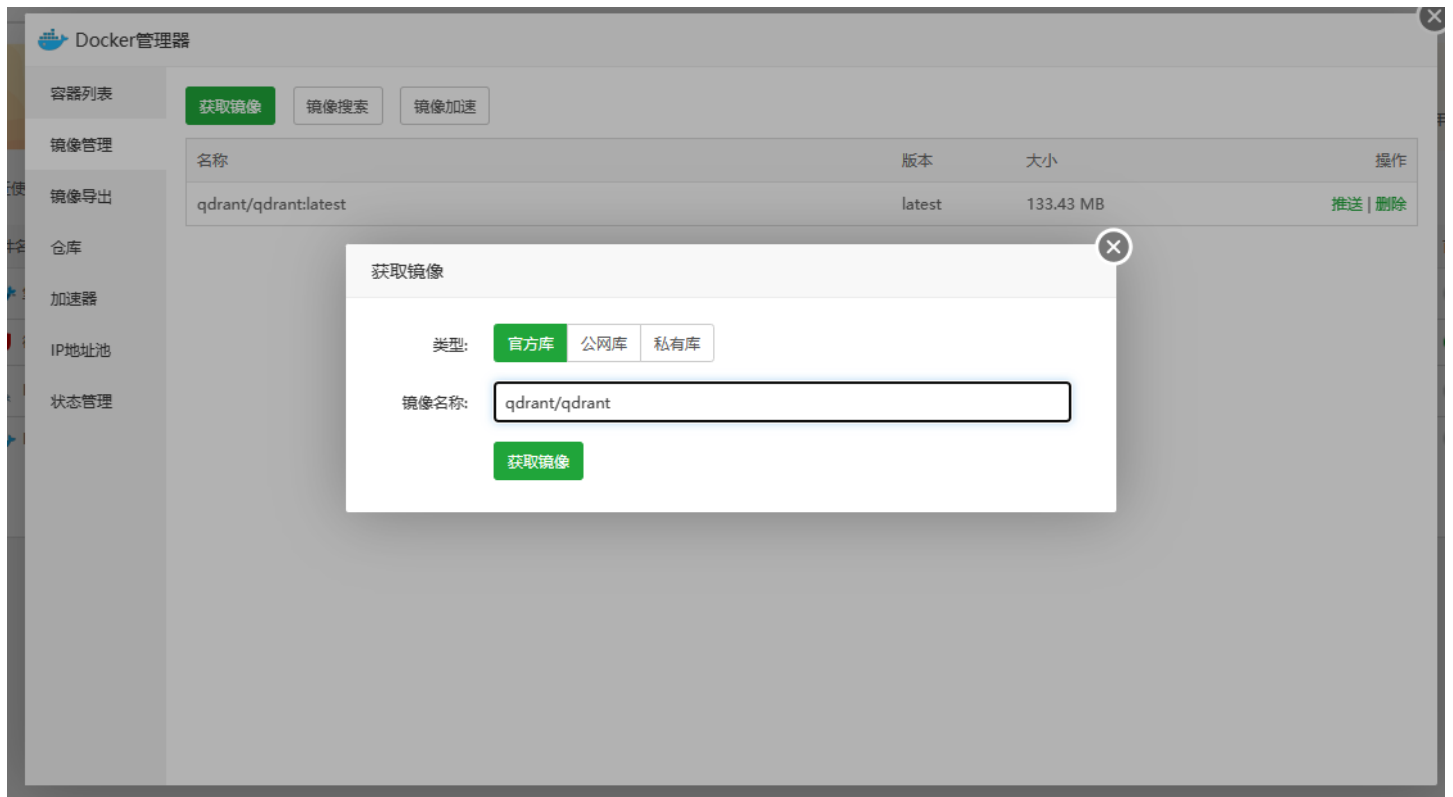
```
1 docker pull qdrant/qdrant
```

运行服务

```
1 docker run -d -p 6333:6333 qdrant/qdrant
```

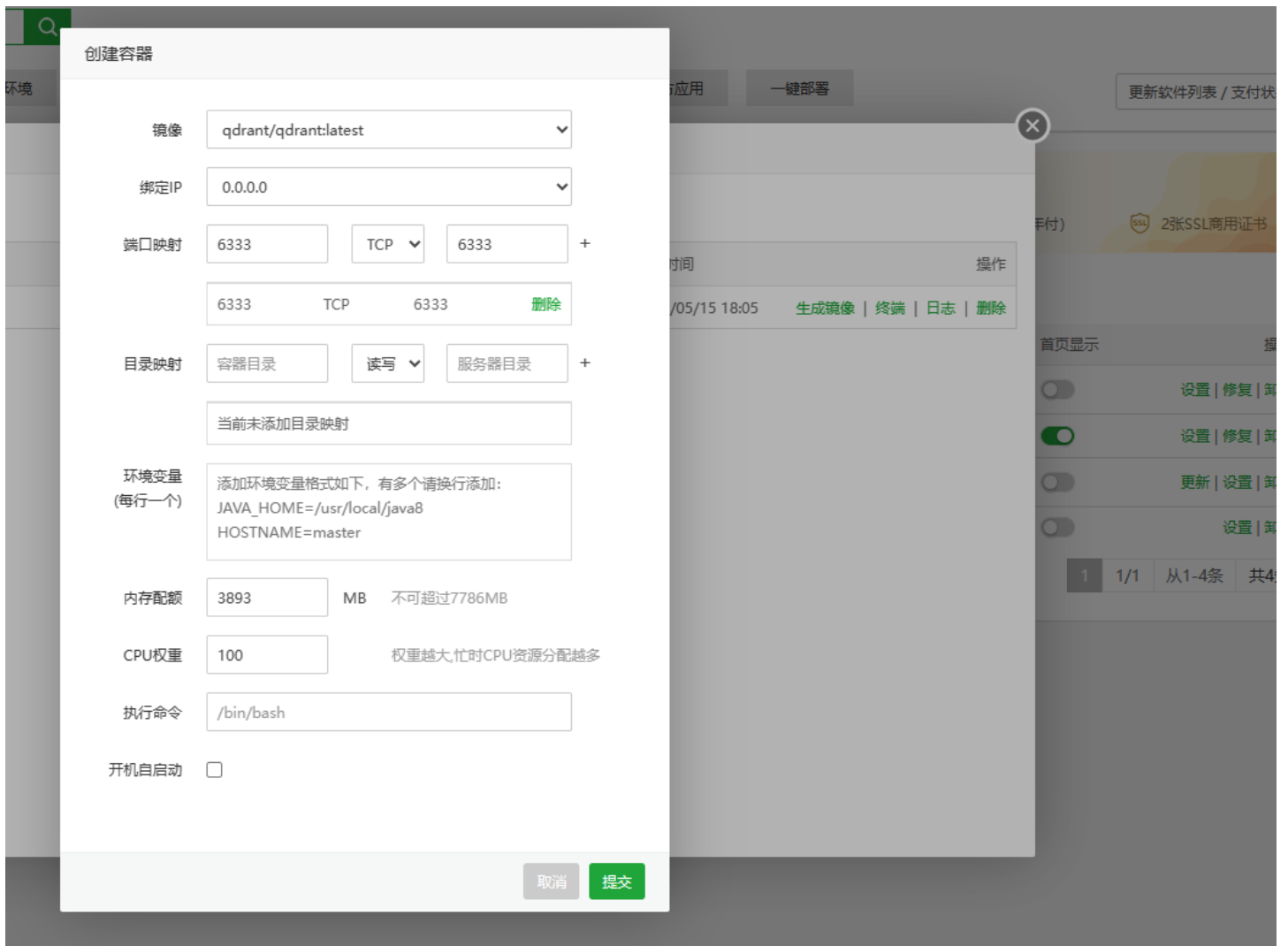
## 宝塔面板下安装

docker管理器的镜像管理里，拉取qdrant/qdrant



根据镜像创建容器，注意端口映射那里，填完以后一定要点那个+号，其他的是默认的





这样就能安装成功了。

## 启动容器

当服务器重启以后，需要重新启动容器

先查看容器ID `docker ps -a`

然后启动容器 `docker start 容器ID`

## 13. Apache Tika 读取复杂PDF服务安装

部署知识库服务时，如果有上传PDF文档的需求，并且PDF的格式复杂非标准格式，需要安装此依赖，该服务也是以Docker形式运行。

(无复杂PDF上传需求时，可忽略)

### 命令行安装

```
1 docker run -d -p 9998:9998 apache/tika:latest
```

# 宝塔面板下安装

略

## 14. 华为云服务器ECS

如果使用的是华为云ECS，那么可以参照这个进行购买，新用户都有优惠。我创建一个按量付费的进行下演示，这个是关机不计费的。

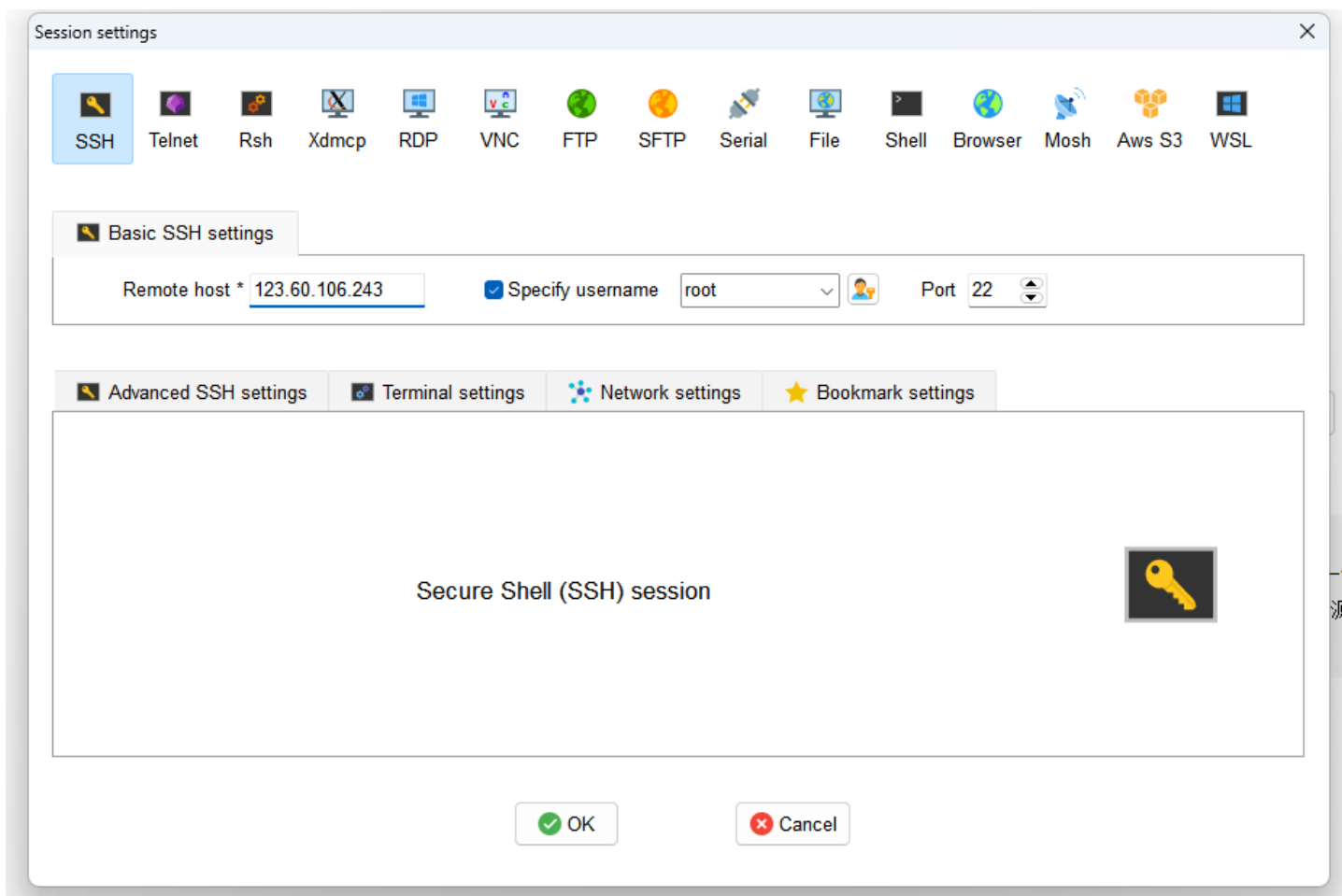
购买ECS地址：<https://www.huaweicloud.com/product/ecs.html>

选了一个CentOS系统，最低配 1核心 1G进行测试



记住自己设置的密码

使用一个SSH客户端连接进行就可以了，比如我使用的是MobaXterm



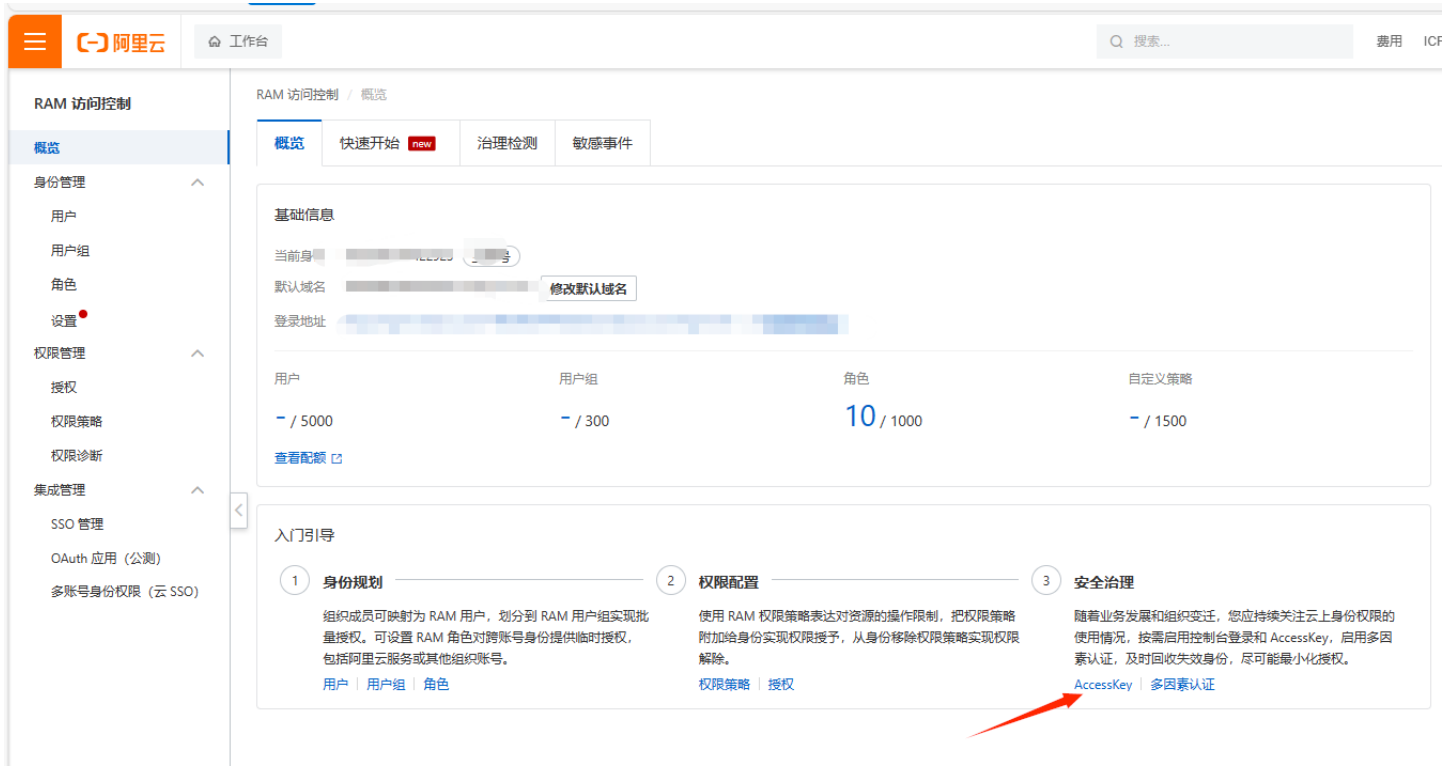
## 15. 阿里云短信服务配置

使用阿里云短信服务，必须有以下参数。在阿里云后台开通服务后，找全以下参数

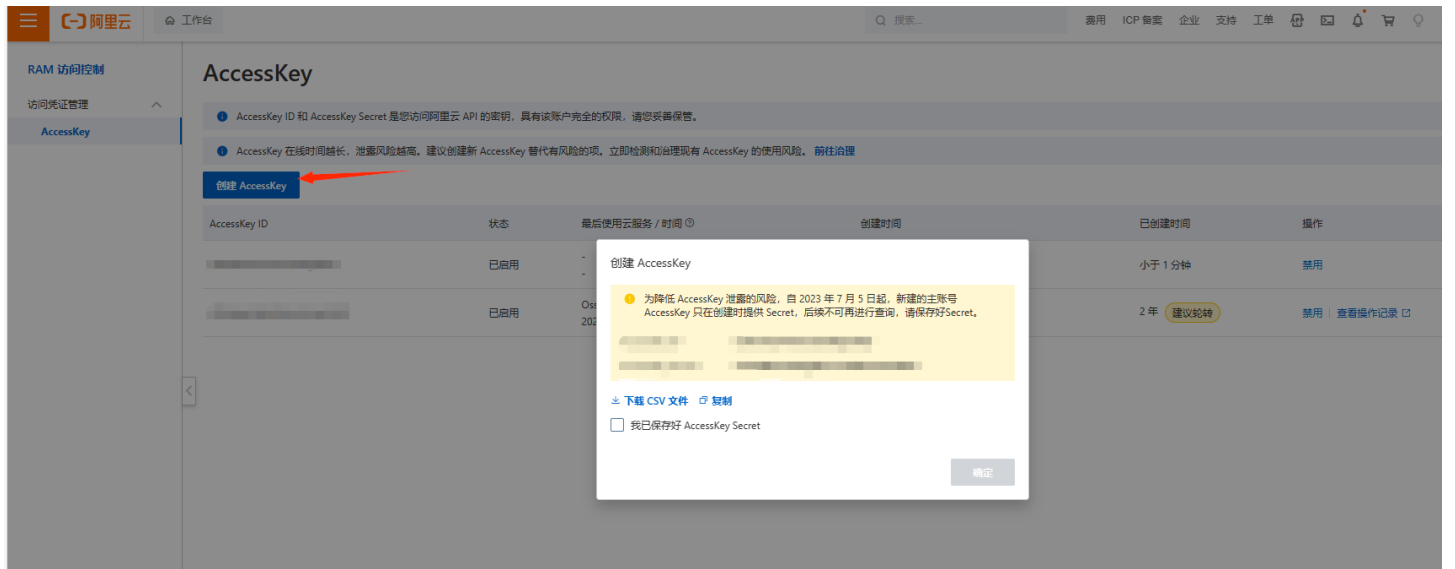
- 1 AccessKey ID
- 2 AccessKey Secret
- 3 模板Code
- 4 短信前缀前面

### 开启阿里云访问控制RAM

在左侧的产品列表找【访问控制】



## 创建AccessKey



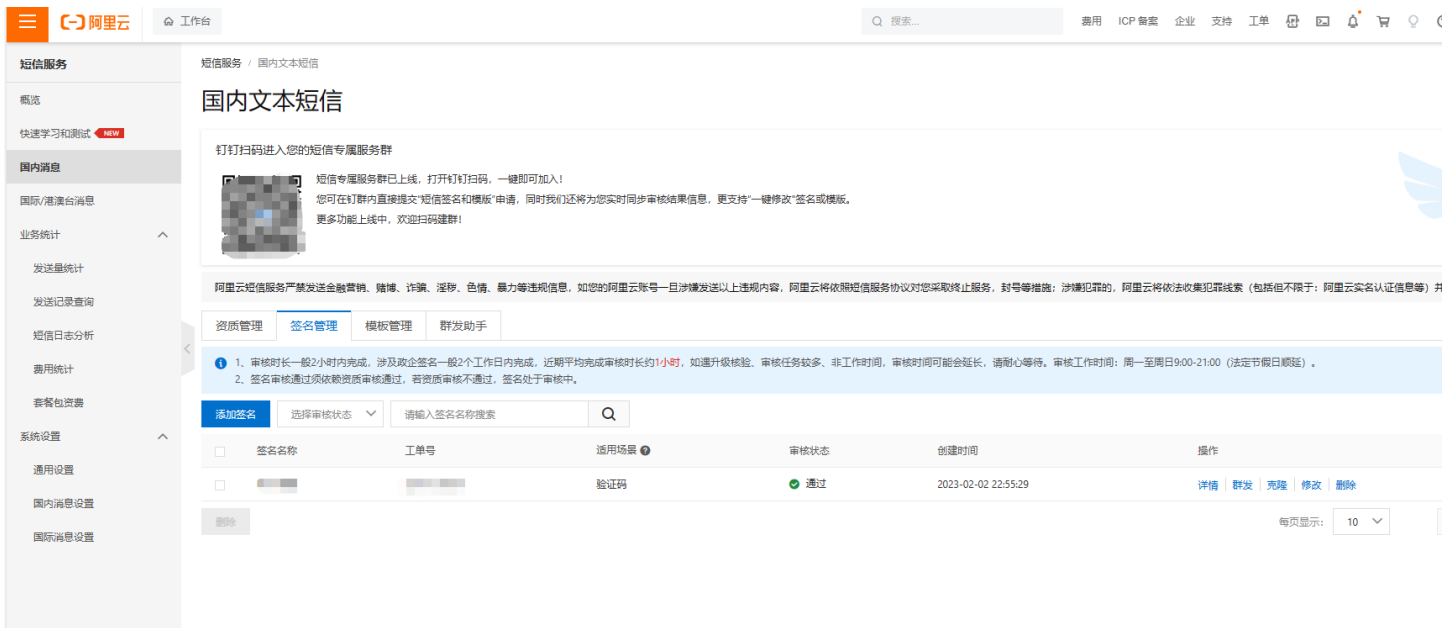
获取到AccessKey ID, AccessKeySecret

## 开启阿里云短信服务

在左侧的产品列表找【短信服务】

添加一个签名, 并选择一个短信模板, 比如我选择一个验证码的短信模板, 这样可以获取到短信内容前缀签名和模板CODE。

签名和短信模板都是需要审核的, 不能随意取。



请参考自己语言的SDK包进行开发

<https://api.aliyun.com/api-tools/sdk/Dysmsapi?version=2017-05-25&language=go-tea>

## 16. 阿里云OSS

客服系统上传图片或文件时，可以配置同步到阿里云的OSS中

首先需要先开通好oss，获取到以下几个参数：

阿里云OSS(EndPoint);

阿里云OSS(AccessKeyId)阿里云OSS(AccessKeySecret);

阿里云OSS(OssBucket);

阿里云OSS(Bucket域名);

因为上传完成以后会直接使用阿里云的外链地址，所以要把权限打开，创建好bucket后，权限管理里把Bucket ACL改成公共读

对象存储 / Bucket列表 / q01lychat / 权限管理

goflychat / 华东1 (杭州) ▾



## EndPoint和bucket域名在这里就能看到

用量查询 >  
文件管理 >  
权限管理 >  
基础设置 >  
冗余与容错 >  
传输管理 >  
日志管理 >  
数据管理 >  
数据安全 >

基础数据统计平均延迟 1-2 小时，不作为计量数据，仅作参考。不同存储类型的存储用量计费规则，请参考帮助文档。

存储用量 总用量 (不含 ECS 快照) 本月流量 外网流出流量 本月请求次数 读请求 文件数量 文件碎片

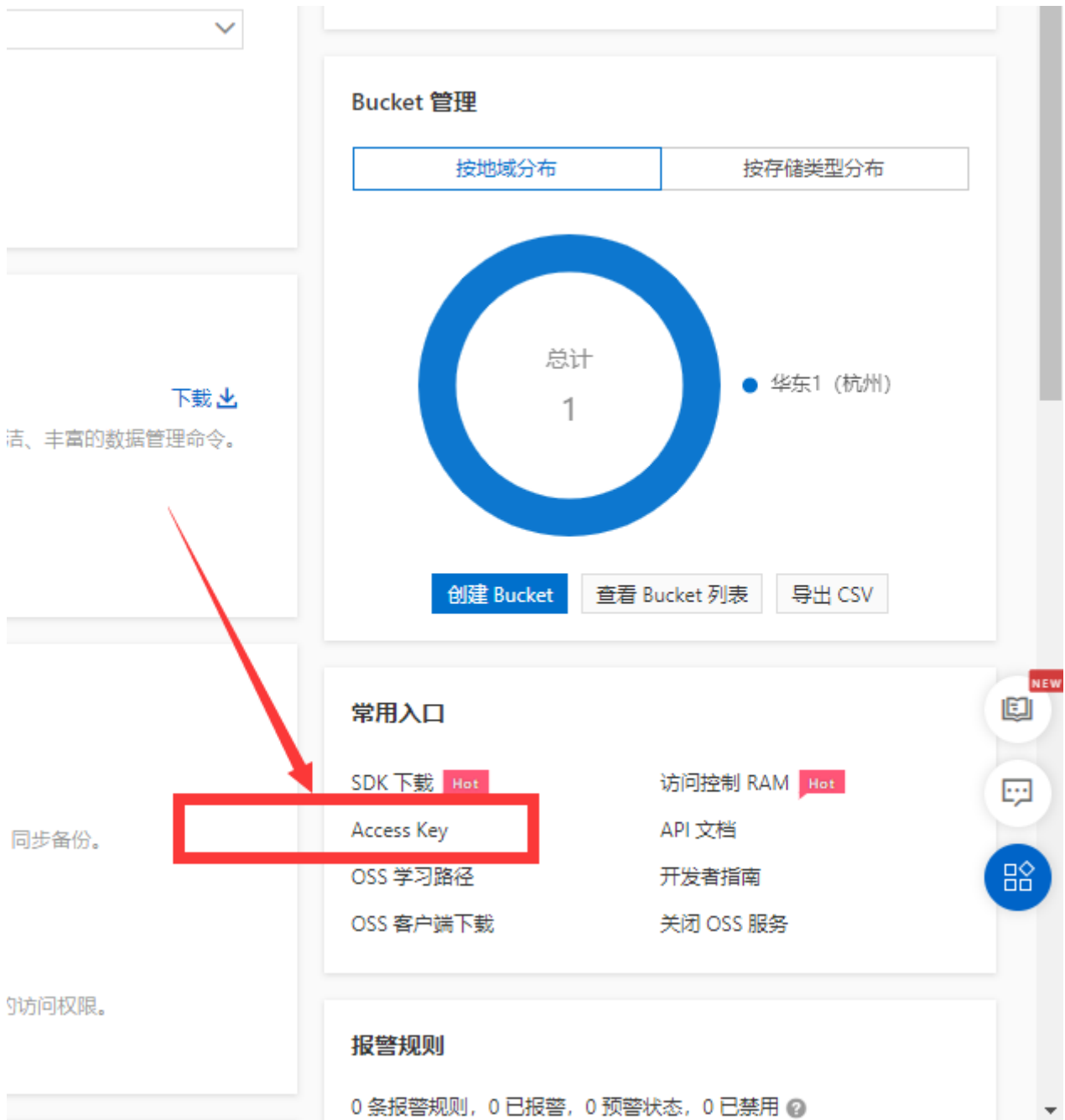
原始容量 1.24 MB 计算容量 1.24 MB 4.84 MB 381 30 0

月同比 -- 月同比 -- 上月外网流出流量: 0 Byte 上月请求次数: 0

访问域名

	Endpoint (地域节点)	Bucket 域名	HTTPS
外网访问	oss-cn-hangzhou.aliyuncs.com	goffychat-oss-cn-hangzhou.aliyuncs.com	支持
ECS 的经典网络访问 (内网)	oss-cn-hangzhou-internal.aliyuncs.com	goffychat-oss-cn-hangzhou-internal.aliyuncs.com	支持
ECS 的 VPC 网络访问 (内网)	oss-cn-hangzhou-internal.aliyuncs.com	goffychat-oss-cn-hangzhou-internal.aliyuncs.com	支持
传输加速域名 (全地域上传下载加速)	未开启	<a href="#">开启</a>	支持

## AccessId 和 AccessSecret在这里找



在GOFly客服系统后台，管理员身份登进去，配置参数都添加好，这时候上传就能上传到oss了

配置参数	配置项	配置值
IP黑名单	访客ID生成规则 (ip:是通过客户ip生成, 留空是uuid)	VisitorIdRule
重启服务	阿里云OSS(EndPoint)	OSS_ENDPOINT
	阿里云OSS(AccessKeyId)	OSS_ACCESS_KEY_ID
	阿里云OSS(AccessKeySecret)	OSS_ACCESS_KEY_SECRET
	阿里云OSS(OssBucket)	OSS_BUCKET
	阿里云OSS(Bucket域名)	OSS_BUCKET_DOMAIN

## 17. 腾讯云云数据库MySQL

项目运行一段时间后，一般会选择数据库服务器和程序应用服务器分离，下面的是腾讯云云数据库MySQL

新建实例可以根据自己需求选择配置

**使用说明** 单节点云盘实例，目前已支持上海、北京、广州、成都、中国香港、新加坡、法兰克福地域，具备高性价比，满足测试、开发、个人学习等场景使用，最大支持30T存储空间，欢迎选购。

**基础配置**

计费模式  包年包月 适用需求量长期稳定的业务  按量计费 适用需求量大波动场景

地域  中国  亚太  欧洲和美洲

广州 上海 南京 北京 成都 重庆 中国香港

处于不同地域的云产品内网不通，购买后不能更换，请您谨慎选择；例如，广州地域的云服务器无法通过内网访问上海地域的MySQL，若需要跨地域内网通信，请查阅：[对等连接](#)

数据库版本 MySQL5.6 MySQL5.7 **MySQL8.0** MySQL5.7(TDSQL-C) 了解更多

推荐使用新一代云数据库TDSQL-C，100%兼容MySQL，秒级添加只读实例和原地升降级，快照备份回档，海量智能存储自动扩容，按使用量计费。

引擎  InnoDB 了解更多

最常用的OLTP存储引擎，拥有完整的事务支持与强大的读写高并发能力

架构  双节点  三节点  单节点 了解更多

基础版不承诺 SLA，故障恢复时间较长，生产环境推荐使用双节点或三节点版本，提供最高99.99%可用性保障。

硬盘类型  云盘 了解硬盘类型

可用区  上海一区  上海二区  上海三区  上海四区  上海五区  上海八区

处于同一私有网络下不同可用区的云产品内网互通；例如，相同私有网络下的广州二区的云服务器可以通过内网访问广州三区的MySQL

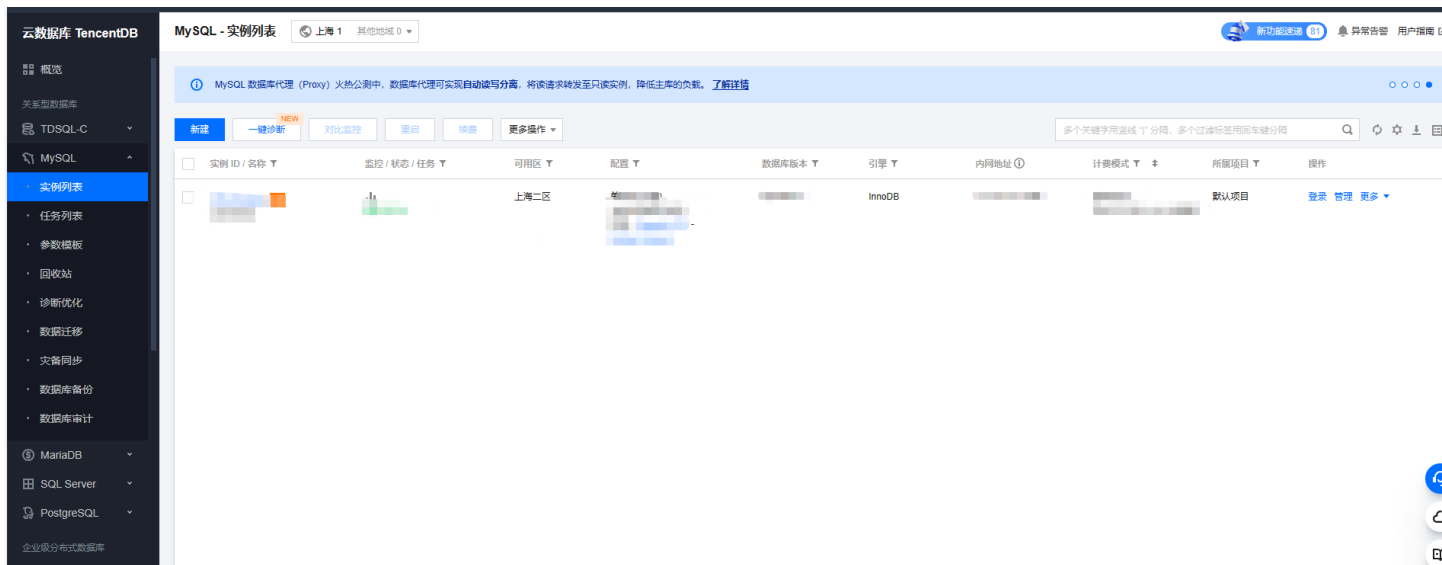
已选 基础版-1核1000MB内存 时长 1年 (7.5折) 数量 1

配置费用 495.00元 660.00元 备份费用 0.00025元/GB/小时 流量费用 0.00元/GB

[下一步: 设置网络和数据库](#)

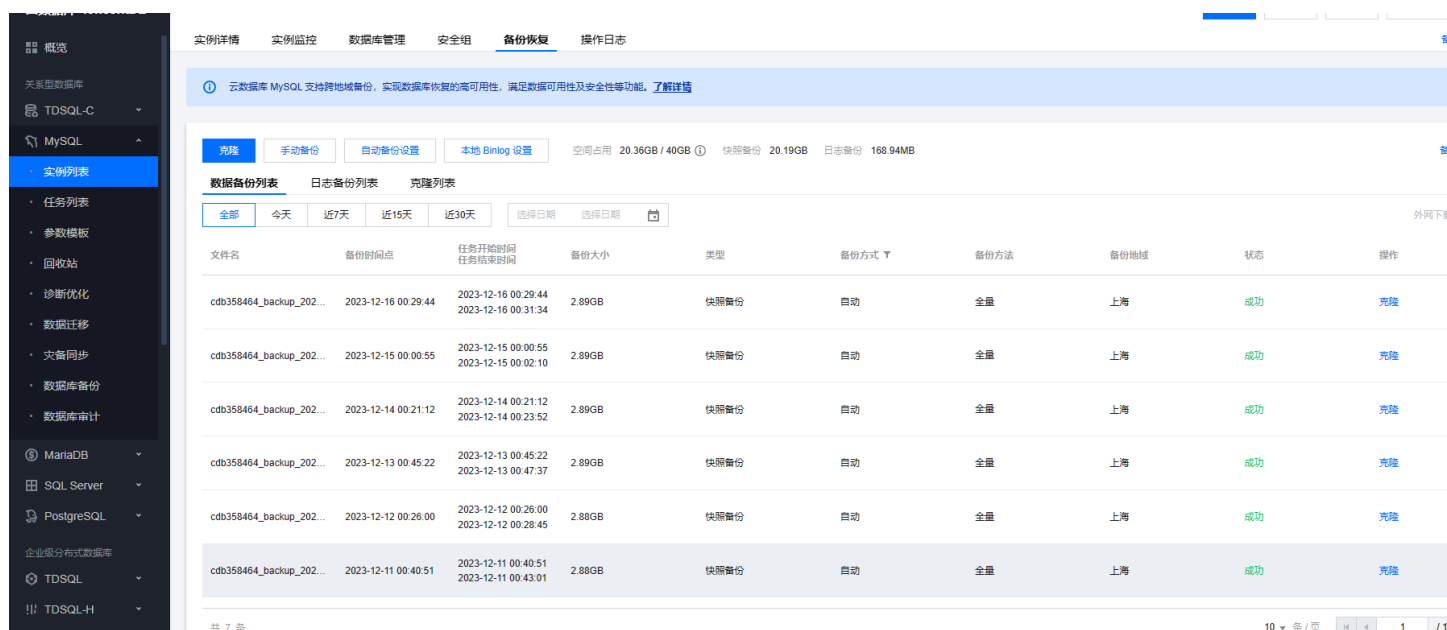
实例创建完成后会看到实例列表





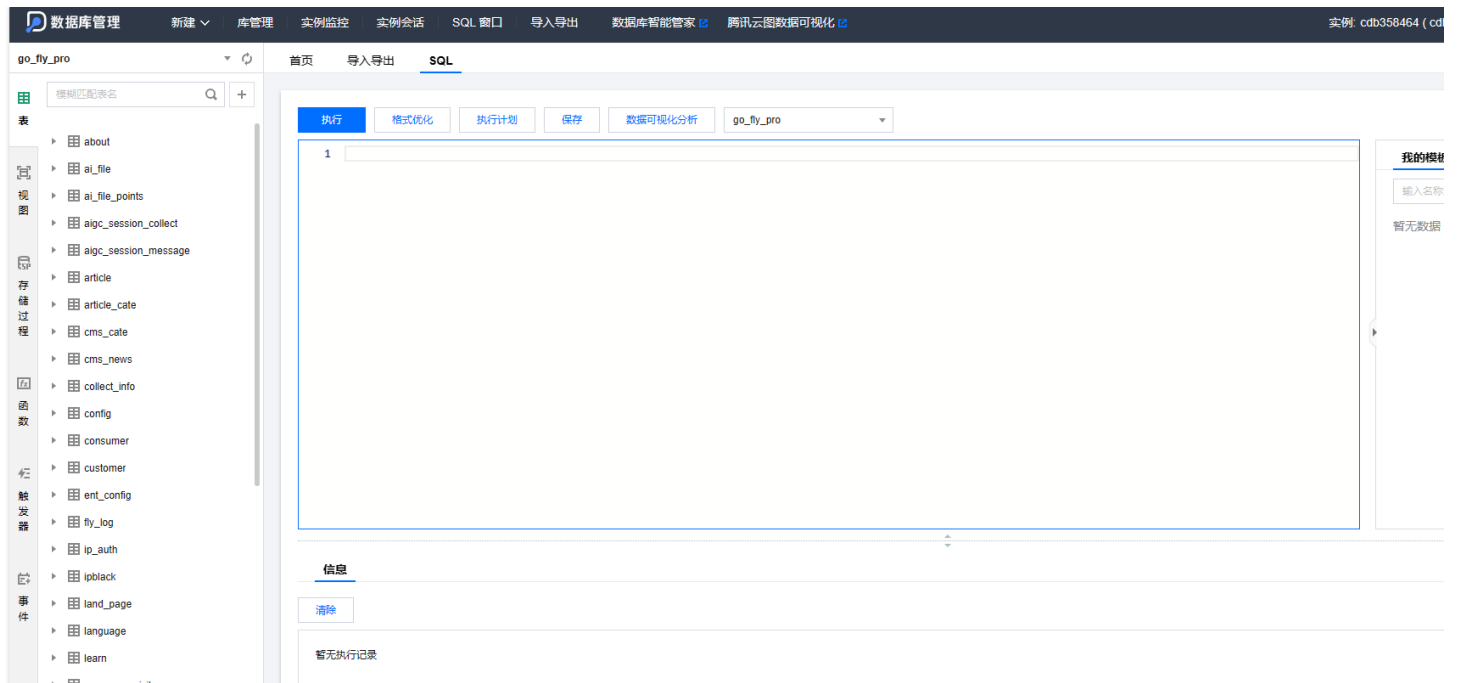
腾讯云数据库的逻辑备份是收费的

腾讯云数据库的快照备份有一定的免费额度，可以实现克隆恢复到新的实例



点击管理，可以登录到数据库管理后台，在这里进行数据库操作

导入可以把自己的SQL导入进来，导出可以输出SQL文件



## 18. 性能监控

Go 语言提供的性能剖析工具 (pprof)

访问 /debug/pprof/ 可以查看性能使用情况

## 19. Python环境搭建

当前，不管是RPA自动化类业务需求，还是使用AI实现变现或解决问题，都离不开python语言。

我举例子：八爪鱼RPA或影刀都有执行python代码功能。coze智能体平台的代码节点。调用GPT或各种指纹浏览器接口。AI生成python代码生成的做好。所以我们需要会搭建python环境

### 19.1 Windows下Python环境搭建

安装python3.10版本，不要安装最新版，有些依赖库不兼容

<https://www.python.org/ftp/python/3.10.7/python-3.10.7-amd64.exe>

一直下一步就可以了，注意要勾上pip和加入到环境变量选项

配置清华大学源

```
1 pip config set global.index-url https://pypi.tuna.tsinghua.edu.cn/simple
```

配置阿里云源

```
1 pip config set global.index-url https://mirrors.aliyun.com/pypi/simple/
```

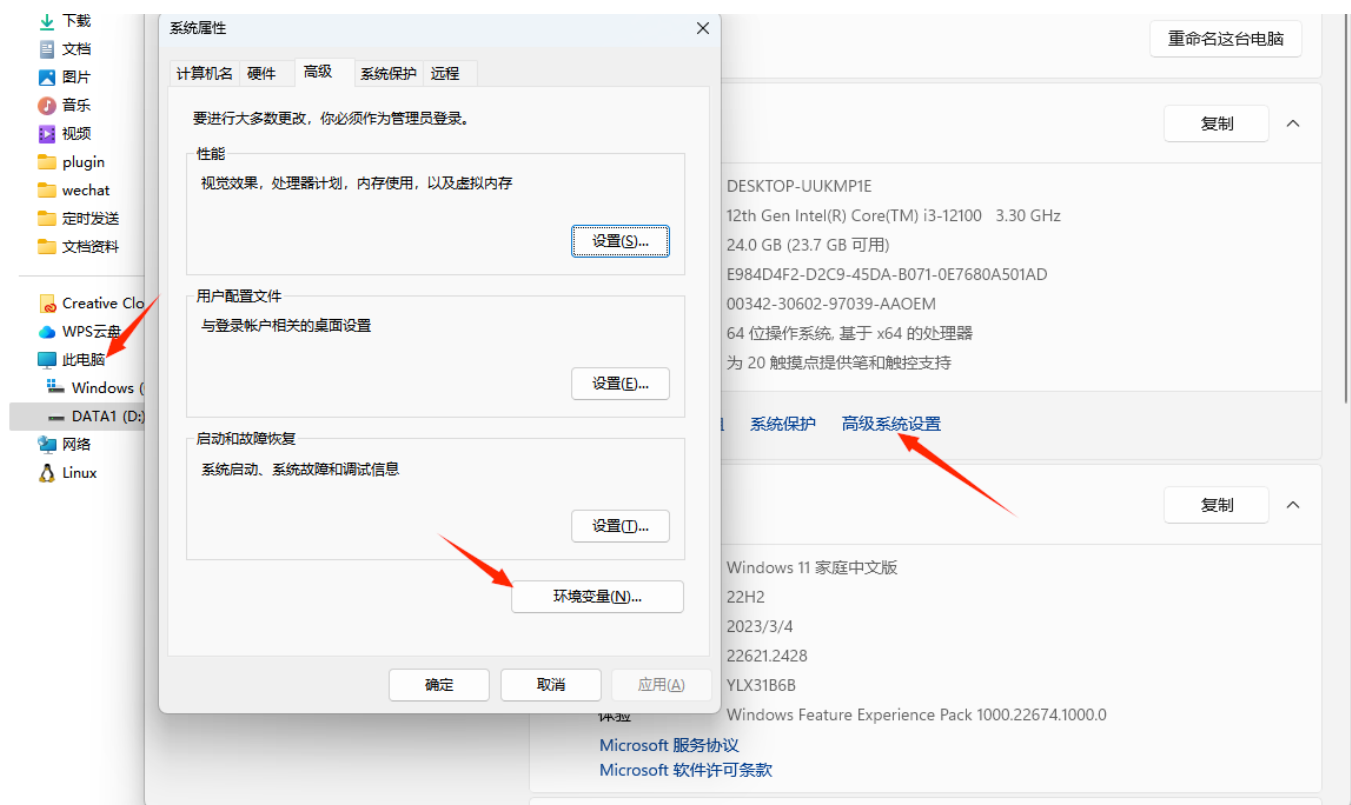
如果安装过程中没有选添加到环境变量，那么直接运行python会说找不到命令  
需要按照下面的方式设置环境变量

【此电脑】 【右键属性】 【高级系统设置】 【环境变量】

Path里增加上python目录的路径，多个路径分号分隔，例如：

C:\Users\63089\AppData\Local\Programs\Python\Python310\

C:\Users\63089\AppData\Local\Programs\Python\Python310\Scripts\



## 19.2 pip用法

pip是Python的包管理器，可以安装第三方的扩展库

pip更新特定库： pip install --upgrade 库名

查看单个库的版本： pip show 库名

安装指定版本的库： pip install openai==0.28

## 19.3 python的虚拟环境

Python虚拟环境是一种用于创建隔离的Python运行环境的方法。这样做的好处是，你可以为不同的项目设置不同的环境，每个环境都有自己的Python版本和一套独立的第三方库，这样可以避免不同项目之间的依赖冲突。

```
1 pip install virtualenv
```

创建虚拟环境

```
1 virtualenv myenv
```

激活虚拟环境

```
1 myenv\Scripts\activate
```

这个时候安装的依赖，是保存在了项目目录下了

Lib 目录，Scripts 目录

```
1 pip install package_name
```

## 20. 向量库Qdrant运维

Qdrant向量数据库运维方法

列出所有集合

查询集合中条数

按照指定条件查询向量条数

按照指定条件查询向量

按条件删除向量

集合快照与恢复

```
1 from qdrant_client import QdrantClient
2 from qdrant_client.http.models import models
3
```

```

4 # 文档地址
5 # https://qdrant.tech/documentation/concepts/points/
6
7 client = QdrantClient("ip", port=6333)
8 # 列出所有集合
9 collects = client.get_collections()
10 print(collects, "\n-----")
11 # 查询集合信息
12 collectInfo = client.get_collection(collection_name="qingfeideyi0926")
13 print(collectInfo, "\n-----")
14
15 # 按条件查询集合点个数
16 nums = client.count(
17     collection_name="qingfeideyi0926",
18     count_filter=models.Filter(
19         must=[
20             models.FieldCondition(
21                 key="fileid",
22                 match=models.MatchValue(value="453")
23             ),
24         ]
25     ),
26     exact=True,
27 )
28 print(nums, "\n-----")
29 # 按条件查询集合点
30 points = client.scroll(
31     collection_name="qingfeideyi0926",
32     scroll_filter=models.Filter(
33         must=[
34             models.FieldCondition(
35                 key="fileid",
36                 match=models.MatchValue(value="453")
37             ),
38         ]
39     ),
40     limit=1,
41     with_payload=True,
42     with_vectors=False,
43 )
44 print(points, "\n-----")
45
46 # 按条件删除
47 res = client.delete(
48     collection_name="qingfeideyi0926",
49     points_selector=models.FilterSelector(
50         filter=models.Filter(

```

```

51         must=[
52             models.FieldCondition(
53                 key="color",
54                 match=models.MatchValue(value="red"),
55             ),
56         ],
57     )
58 ),
59 )
60 print(res, "\n-----")
61
62 # 快照
63 # 快照存储在宿主机的位置是
64 # docker inspect 容器ID
65 # /var/lib/docker/overlay2/容器哈希/diff/qdrant/snapshots/集合
66 # 进容器终端 docker exec -it 容器ID /bin/bash
67 # 恢复快照 ./qdrant --snapshot ./snapshots/快照路径:集合名称
68 # 如果显示出错,就重启容器docker restart 容器ID
69 # client.create_snapshot(
70 #     collection_name="lrtfp0609"
71 # )
72 #api恢复快照
73 # res=client.recover_snapshot("taoshihan",
74 #     "http://81.69.45.114:6333/collections/taoshihan/snapshots/taoshihan-
75 #     3917799556161435487-2023-10-09-15-21-10.snapshot")
76 # print(res, "\n-----")
77 # 全部快照
78 # client.create_full_snapshot()

```

## 21. windows开发环境相关问题

### 21.1 windows查看网络端口占用情况

查看端口

```
1 netstat -ano | findstr :端口号
```

查看进程

```
1 tasklist |findstr 38664
```

## 21.2 windows下安装docker

Docker Desktop 是 Docker 官方提供的用于 Windows 的桌面应用程序，它包含了运行 Docker 所需的所有组件，包括 Docker 引擎、Docker CLI 和 Docker Compose。

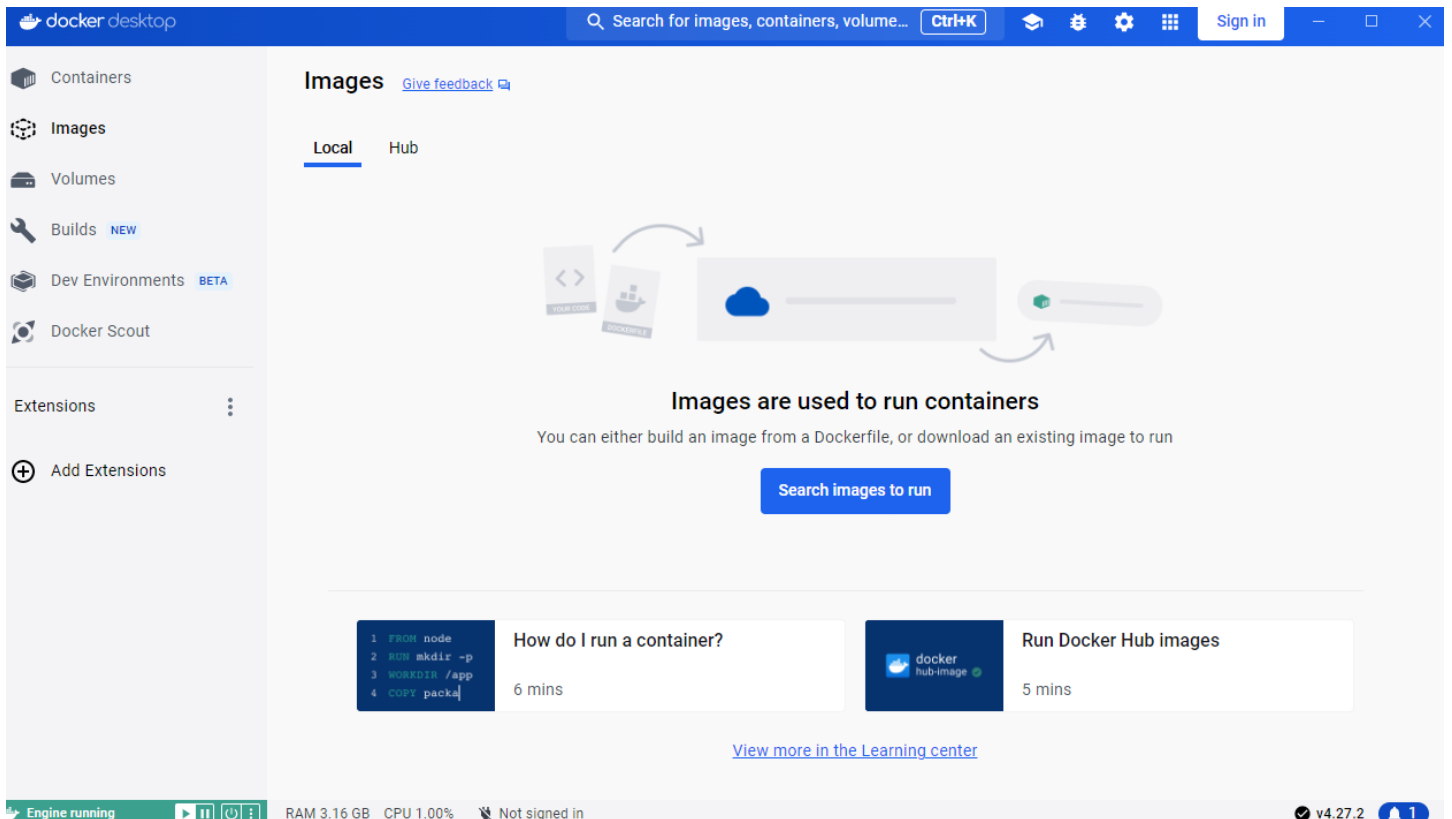
### 21.2.1 安装步骤:

1. 从 Docker 官网: <https://www.docker.com/get-started> 下载 Docker Desktop 安装程序。
2. 运行安装程序并按照屏幕上的说明进行安装。
3. 安装完成后，启动 Docker Desktop 应用程序。
4. 您可以通过以下命令验证 Docker 是否已成功安装:

```
1 docker version
```

### 21.2.2 使用 Docker Desktop 的优势:

- 安装简单，无需配置环境。
- 提供图形界面，方便管理 Docker 镜像、容器和网络等资源。
- 支持 Windows 10 和 Windows 11 操作系统。



### 21.2.3 镜像打包

## 22. AI相关系统部署（可选，仅供参考）

### 22.1 部署FastGPT

```
1 mkdir fastgpt
2 cd fastgpt
3 curl -O
  https://raw.githubusercontent.com/labring/FastGPT/main/files/deploy/fastgpt/doc
  ker-compose.yml
4 curl -O
  https://raw.githubusercontent.com/labring/FastGPT/main/projects/app/data/config
  .json
```

#### 22.1.1 修改docker-compose.yml

如果不需要OneAPI，可以将OneAPI部分以及MySQL部分的配置去掉

修改FastGPT的镜像源为阿里云的

修改大模型接口地址以及大模型KEY，配置为自己的代理商中转地址，以及代理商中转KEY

`OPENAI_BASE_URL`（API 接口的地址，需要加/v1） `CHAT_API_KEY`（API 接口的凭证）。

#### 22.1.2 拉取镜像启动容器

使用docker compose 拉取镜像

```
1 # 启动容器
2 docker-compose up -d
```

重启服务

```
1 docker-compose down
2 docker-compose up -d
```

#### 22.1.3 如何更新FastGPT版本

1. 查看[更新文档](#)，确认要升级的版本，避免跨版本升级。
2. 修改镜像 tag 到指定版本
3. 执行下面命令会自动拉取镜像：



```
1 docker-compose pull
2 docker-compose up -d
```

## 1. 执行初始化脚本（如果有）

## 22.2 one-api 多种大模型LLM兼容OpenAI 接口管理系统

现在GPT大模型或国产大模型有很多，Azure、Anthropic Claude、Google PaLM 2、智谱ChatGLM、百度文心一言、讯飞星火认知、阿里通义千问、360 智脑以及腾讯混元，我们需要有一个中间系统来将多种LLM接口转换为兼容OpenAI的接口形式，对外提供一致的调用方法

### 22.2.1 下载源码

```
git clone https://github.com/songquanpeng/one-api.git
```

### 22.2.2 创建MySQL数据库

```
create database `oneapi`
```

### 22.2.3 设置环境变量

该环境变量是数据库的连接信息

在 Windows 中设置环境变量，需要使用管理员运行cmd命令行

```
setx SQL_DSN "root:123456@tcp(localhost:3306)/oneapi" /m
```

```
D:\software\goworkspace\test\one-api>setx SQL_DSN "root:taoshihan1@tcp(localhost:3306)/oneapi" /m
成功：指定的值已得到保存。

D:\software\goworkspace\test\one-api>echo %SQL_DSN%
root:taoshihan1@tcp(localhost:3306)/oneapi
```

### 22.2.4 在 Linux 中设置环境变量：

使用 ~/.bashrc 或 ~/.bash\_profile：

打开终端。

使用文本编辑器打开 `~/.bashrc` 或 `~/.bash_profile` 文件。

在文件末尾添加如下行：

```
export SQL_DSN="root:123456@tcp(localhost:3306)/oneapi"
```

保存并关闭文件。

运行 `source ~/.bashrc` 或 `source ~/.bash_profile` 使更改生效。

### 22.2.5 编译前端

进入web目录

```
npm install
```

```
npm run build
```

也可以本地windows编译后，直接上传到web/build/目录下面

## 22.2.6 源码运行

```
go run main.go
```

## 22.2.7 访问地址

默认监听3000端口

<http://localhost:3000>

## 22.2.8 编译运行

如果有修改前端资源，必须从新编译程序

```
go build -o oneapi
```

## 22.2.9 使用supervisor启动守护模式

配置文件中需要增加环境变量

```
1 environment=SQL_DSN="oneapi:oneapi@tcp(127.0.0.1:3306)/oneapi"
```

启动命令需要使用绝对路径

```
1 command=/www/wwwroot/one-api/oneapi --port 3001
```

## 22.2.10 增加通义千问渠道

例如增加一个通义千问的渠道

<https://dashscope.console.aliyun.com/apiKey>

### 管理渠道

搜索渠道的 ID, 名称和密钥 ...

当前版本测试是通过按照 OpenAI API 格式使用 gpt-3.5-turbo 模型进行非流式请求实现的, 因此测试报错并不一定代表通道不可用, 该功能后续会修复。另外, OpenAI 渠道已经不再支持通过 key 获取余额, 因此余额显示为 0。对于支持的渠道类型, 请点击余额进行刷新。

ID	名称	分组	类型	状态	响应时间	余额	优先级	操作
1	通义千问	default	阿里通义千问	已启用	未测试	不支持	0	测试 删除 禁用 编辑

添加新的渠道 测试所有已启用通道 更新所有已启用通道余额 删除禁用渠道 刷新

选择类型：阿里通义千问

模型：加上我们想要支持的

重定向：可以把传入的模型名称，重定向到我们设定的通义千问的

密钥：填入阿里通义千问的API-KEY

### 更新渠道信息

类型 \*

阿里通义千问

名称 \*

通义千问

分组 \*

default

插件参数

请输入插件参数, 即 X-DashScope-Plugin 请求头的取值

模型 \*

qwen-turbo qwen-plus gpt-3.5-turbo-16k

填入基础模型

填入所有模型

清除所有模型

输入自定义模型名称

填入

模型重定向

```
{
  "gpt-3.5-turbo-16k": "qwen-turbo"
}
```

密钥 \*

请输入渠道对应的鉴权密钥

代理

此项可选, 用于通过代理站来进行 API 调用, 请输入代理站地址, 格式为: https://domain.com

取消

提交



## 使用通义千问

```
1 import requests
2
3 url = "http://localhost:3000/v1/chat/completions" # 替换为目标URL
4 data = {
5     "model": "qwen-plus",
6     "messages": [
7         {
8             "role": "user",
9             "content": "你是谁"
10        },
11    ]
12 }
13 headers = {
14     "Authorization": "Bearer sk-
15     VdDeP3V7Cjktw0c6964b6eC8A8694519B880DbD93e24357b"
16 }
17 response = requests.post(url, json=data, headers=headers)
18 json_data = response.json()
19 print(json_data["choices"][0]["message"]["content"])
```

## 22.3 Kimi网页接口逆向API

开源地址：<https://gitee.com/llm-red-team/kimi-free-api>

接口格式兼容OpenAI

### 22.3.1 docker部署

```
1 docker run -it -d --init --name kimi-free-api -p 3002:8000 -e TZ=Asia/Shanghai
```

## 22.3.2 接入准备

从 [kimi.moonshot.cn](https://kimi.moonshot.cn) 获取refresh\_token

进入kimi随便发起一个对话，然后F12打开开发者工具，从Application > Local Storage中找到

refresh\_token 的值，这将作为Authorization的Bearer Token值：  
Authorization: Bearer TOKEN

## 22.3.3 多账号接入

目前kimi限制普通账号每3小时内只能进行30轮长文本的问答（短文本不限），你可以通过提供多个账号的refresh\_token并使用 `,` 拼接提供：

```
Authorization: Bearer TOKEN1,TOKEN2,TOKEN3
```

每次请求服务会从中挑选一个。

## 22.4 阿里云语音合成sambert

<https://help.aliyun.com/zh/dashscope/developer-reference/quick-start-13>

Sambert系列模型 1万字1元，每主账号每模型每月3万字免费，

### 创建API-KEY

<https://dashscope.console.aliyun.com/apiKey>

### 调用代码生成音频文件

```
pip install dashscope
```

## 23. 官网网站搭建开发

搭建展示用官方网站，建议使用PHP语言开发的网站系统。

总费用=域名几十块一年 + 服务器一年几百块 + 程序（有免费开源）

国内大的云厂商，都提供域名注册，和服务器购买。如果买的是国内服务器，那么域名解析后，需要进行ICP备案。

如果嫌麻烦，可以买香港或海外服务器，这样可以免备案。

## 24. 浏览器自动化 workflow 插件 automa

微软Edge浏览器和谷歌chrome浏览器同时都支持

将automa.zip的压缩包解压缩，例如解压后的目录是 automa



浏览器【扩展管理】=>开启【开发人员模式】=>【加载解压缩的扩展】，选择上面解压缩的目录



## 25. 微信机器人搭建

WeChatFerry - Hook版微信机器人

<https://github.com/lich0821/WeChatFerry>

下载发行包，安装指定版本微信，将dll文件复制到微信安装目录下

<https://github.com/lich0821/WeChatFerry/releases>

下载应用包

<https://github.com/lich0821/WeChatRobot>

openwechat - web协议版微信机器人

(一般的账号都无法登网页版了，有老号的可以研究)

开源golang库 <https://github.com/eatmoreapple/openwechat>

## 26. 常见问题：

### 域名授权和企业授权版的区别

前端开源，后端编译，域名授权

全部开源，全部授权，仅限对公客户

### 宝塔删除不了.user.ini文件

lsattr .user.ini 查看是否为锁定状态 有个 i标志

chattr -i .user.ini 解除锁定

### 服务器宕机客服系统启动流程

在程序所在目录进入命令行终端

执行

```
1 ./kefu server -d
```

### 容器启动步骤

先查看容器运行情况

```
1 docker ps -a
```

启动容器

```
1 docker start 容器ID
```

如果遇到启动不了容器的情况，可以重启docker

```
1 service docker restart
```

### 发消息不能即时通知推送

如果遇到搭建好客服系统后，发消息不能即时通知推送，那就是webscoket部分没有配置正确  
在反向代理的配置块里

```
1 proxy_set_header Upgrade $http_upgrade;  
2 proxy_set_header Connection "upgrade";
```

## 我重启服务器，服务运行不了？

没事不要乱重启服务器

没事不要乱重启服务器

重启完了，再启动下客服服务