

BISHENG平台ARM架构部署手册

1. 引言

1.1 编写目的

本手册旨在为系统管理员、运维工程师及技术支持人员提供一套完整、详细的 **BISHENG平台在ARM架构环境下的部署指南**。随着国产化软硬件生态的发展，越来越多的业务系统需要适配基于ARM架构的服务器与操作系统。为了确保BISHENG平台能够在ARM平台上稳定、高效地运行，本文档从实际部署需求出发，详细说明了软硬件环境准备、依赖组件安装、服务配置与启动等关键步骤，帮助用户快速完成平台的部署与验证工作。

同时，本手册也为后续系统的维护、升级和故障排查提供了参考依据，提升平台在ARM架构下的可维护性与稳定性，助力实现平台全面适配国产化技术路线的目标。

注：本手册只包含BISHENG平台核心服务部署，不包含OCR及其他模型部署！

1.2 适用范围

本手册适用于使用基于ARM架构服务器及操作系统（如鲲鹏CPU，以及Centos、Ubuntu、麒麟Kylin操作系统）进行BISHENG平台部署、运维和管理的相关人员。适用于以下场景：

- BISHENG平台在ARM架构环境中的首次安装与部署；
- 平台版本升级、环境适配及故障排查等升级维护工作。

本手册也可作为开发人员、测试人员及相关技术支持团队参考文档，确保BISHENG平台在ARM架构下的兼容性、稳定性和性能表现满足实际业务需求。

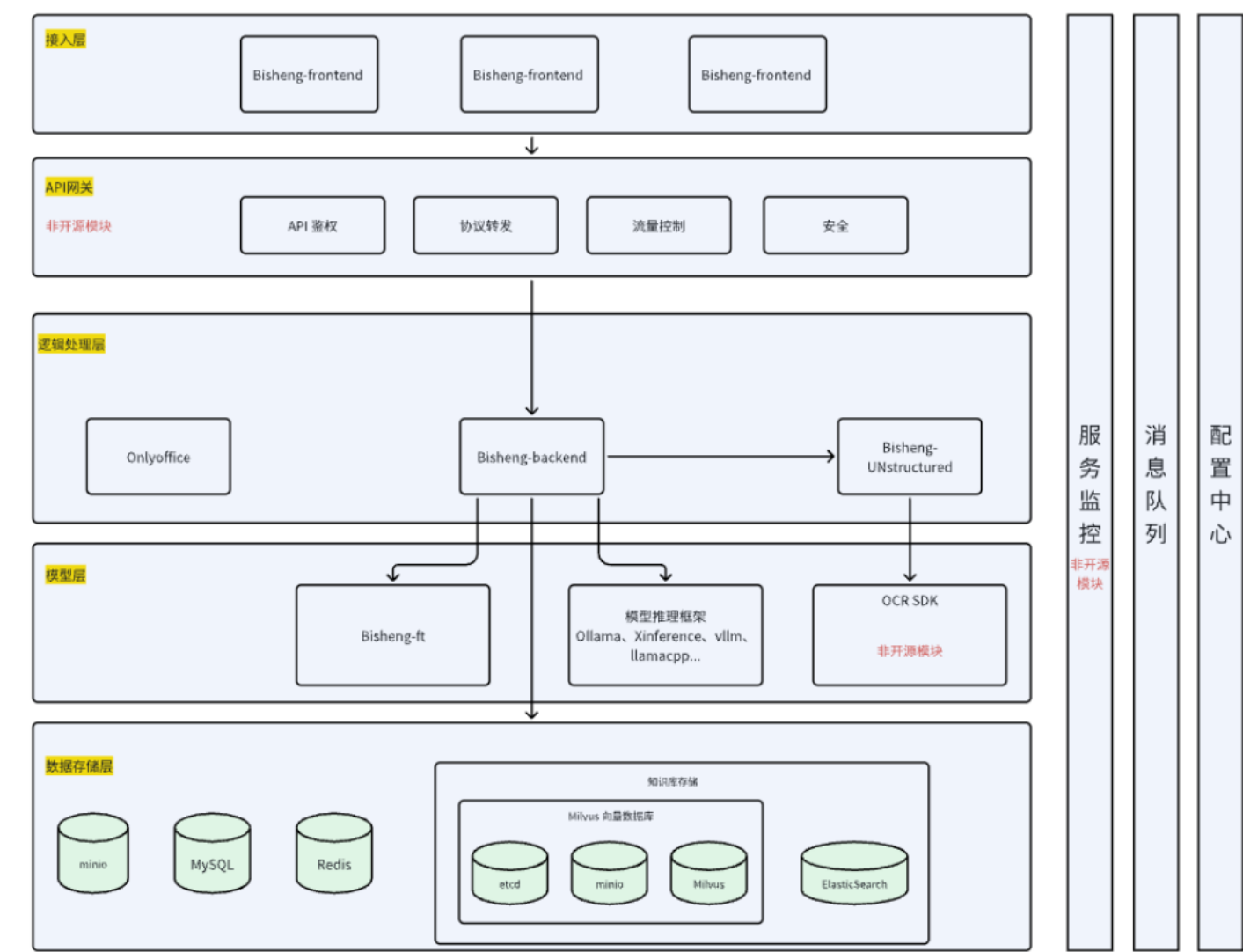
2. 系统概述

2.1 软件简介

应用名称	版本	功能
bisheng-backend	2.0.3-arm	BISHENG后端服务
bisheng-frontend	2.0.3-arm	BISHENG前端服务
mysql	8.0.39-linuxarm64	数据库

redis	7.2.4-linuxarm64	持久化存储
onlyoffice	8.2.0-linuxarm64	word文档生成
elasticsearch	8.12.2-linuxarm64	搜索和分析引擎
etcd	v3.5.13-linuxarm64	存储 Milvus 的配置信息和服务发现与状态监控
minio	RELEASE.2023-04-28T18-11-17Z-linuxarm64	对象存储
milvus	v2.3.21-linuxarm64	向量存储
bisheng-unstructured-arm	release	提供非结构化数据的解析能力

2.2 架构图



2.3 部署环境说明

CPU型号	Kunpeng-920-arm64架构（仅限运行毕昇核心服务）
虚拟CPU核数	32虚拟核
内存	256G
磁盘	SSD 1T*1
操作系统	Kylin Linux Advanced Server V10 (Tercel)
Docker	- Docker 20.10.9+ - Docker Compose version v2.20.3
客户端浏览器	毕昇平台内含有文件溯源展示、word在线编辑功能，需要高级浏览器提供底层能力支持，chrome 建议至少 v92+。建议使用最新版浏览器，以免功能无法正常使用。

2.4 部署方式

本文档部署方式为docker-compose部署，如果是k8s部署请参考：[📖 毕昇组件在K8S上部署的yaml配置参考](#)

3. 部署前准备

3.1 环境要求

- 操作系统版本
 - 只支持ARM架构操作系统：Centos、Ubuntu、麒麟Kylin操作系统
- 硬件配置要求（CPU、内存、磁盘）
 - 虚拟CPU核数： 32核
 - 内存： 256G
 - 磁盘： 1TB
- 网络配置

建议关闭服务器本地防火墙

如果有其他网络安全组的限制，需要按照以下要求开通端口：

应用名称	端口号	是否需要开放端口
backend	7860	否

frontend	3001	是
mysql	3306	否
redis	6379	否
onlyoffice	8701	是
elasticsearch	9200	否
etcd	2379	否
minio	9100、9000	是
milvus	19530	否
bisheng-unstructured	10001	是

- 依赖环境

需要安装docker+docker-compose

- 用户权限要求

建议使用root用户部署，如果不能使用root用户，当前用户需要有sudo权限

3.2 安装包获取

- docker环境安装包

代码块

- 1 通过网盘分享的文件：docker.zip
- 2 链接：<https://pan.baidu.com/s/1IrwkdbbVeKzyEWJkx1pfNg?pwd=6jd2> 提取码：6jd2

- 镜像获取路径

代码块

- 1 # 登陆到毕昇提供的私有镜像仓库
- 2 docker login cr.dataelem.com -u docker -p dataelem
- 3 # 从毕昇私有镜像仓库下载所需的镜像，其中frontend和backend镜像可以相应替换成最新版本，其他镜像较少更新
- 4 docker pull cr.dataelem.com/dataelement/bisheng-frontend:v2.0.3-arm64
- 5 docker pull cr.dataelem.com/dataelement/bisheng-backend:v2.0.3-arm64
- 6 docker pull cr.dataelem.com/bisheng-milvus:v2.3.21-linuxarm64
- 7 docker pull cr.dataelem.com/bisheng-mysql:8.0.39-linuxarm64
- 8 docker pull cr.dataelem.com/bisheng-redis:7.2.4-linuxarm64
- 9 docker pull cr.dataelem.com/bisheng-etcd:v3.5.13-linuxarm64

```
10 docker pull cr.dataelem.com/bisheng-elasticsearch:8.12.2-linuxarm64
11 docker pull cr.dataelem.com/bisheng-minio:RELEASE.2023-04-28T18-11-17Z-
    linuxarm64
12 docker pull cr.dataelem.com/documentserver:7.2.1-arm64
13 docker pull cr.dataelem.com/dataelement/bisheng-unstructured-arm:release
```

- 下载BISHENG代码

代码块

```
1  # 如果系统中有git命令，可以直接下载毕昇代码
2  git clone https://github.com/dataelement/bisheng.git
3  # 进入安装目录
4  cd bisheng/docker
5
6  # 如果系统没有git命令，可以下载毕昇代码zip包
7  wget https://github.com/dataelement/bisheng/archive/refs/heads/main.zip
8  # 解压并进入安装目录
9  unzip main.zip && cd bisheng-main/docker
```

4. 部署步骤

4.1 基础环境配置

代码块

```
1  #永久关闭 SELinux
2  vim /etc/selinux/config
3  #将文件中的SELINUX=enforcing改为SELINUX=disabled
4  #保存更改并退出编辑器
5  #更改需要重启系统才能生效
6  sudo reboot
7
8  #临时关闭 SELinux
9  sudo setenforce 0
10
11 #临时和永久关闭防火墙
12 sudo systemctl stop firewalld
13 sudo systemctl disable firewalld
14
15 #下载docker安装包，并解压
16 unzip docker.zip
17 cd docker
```

```
18 #使用root用户，或者sudo 执行install_docker.sh脚本
19 bash install_docker.sh
20
21 #执行成功后执行以下命令可以返回详细版本信息即为成功
22 [root@ecs-8b0f docker]# docker -v
23 Docker version 20.10.9, build c2ea9bc
24 [root@ecs-8b0f docker]# docker-compose -v
25 Docker Compose version v2.20.3
26
27 #至此运行BISHENG平台需要基础环境安装完成
```

4.2 修改配置文件

- BISHENG平台核心组件配置修改

代码块

```
1 #下载好BISHENG代码或者压缩包并解压进入
2 cd bisheng-main/docker
3 #修改docker-compose.yml配置文件内容
4 #修改内容较多，可以直接复制以下配置
5 services:
6   mysql:
7     container_name: bisheng-mysql
8     image: cr.dataelem.com/bisheng-mysql:8.0.39-linuxarm64
9     ports:
10      - "3306:3306"
11     environment:
12       MYSQL_ROOT_PASSWORD: "1234"
13       MYSQL_DATABASE: bisheng
14       TZ: Asia/Shanghai
15     volumes:
16      - ${DOCKER_VOLUME_DIRECTORY:-.}/mysql/conf/my.cnf:/etc/mysql/my.cnf
17      - ${DOCKER_VOLUME_DIRECTORY:-.}/mysql/data:/var/lib/mysql
18     healthcheck:
19       test: ["CMD-SHELL", "exit | mysql -u root -p$$MYSQL_ROOT_PASSWORD"]
20       start_period: 30s
21       interval: 20s
22       timeout: 10s
23       retries: 4
24     restart: on-failure
25
26   redis:
27     container_name: bisheng-redis
28     image: cr.dataelem.com/bisheng-redis:7.2.4-linuxarm64
29     user: root
```

```
30     ports:
31         - "6379:6379"
32     environment:
33         TZ: Asia/Shanghai
34     volumes:
35         - ${DOCKER_VOLUME_DIRECTORY:-.}/data/redis:/data
36         - ${DOCKER_VOLUME_DIRECTORY:-.}/redis/redis.conf:/etc/redis.conf
37     command: redis-server /etc/redis.conf
38     healthcheck:
39         test: ["CMD-SHELL", 'redis-cli ping|grep -e "PONG\\|NOAUTH"']
40         interval: 10s
41         timeout: 5s
42         retries: 3
43     restart: on-failure
44
45     elasticsearch:
46         container_name: bisheng-elasticsearch
47         image: cr.dataelem.com/bisheng-elasticsearch:8.12.2-linuxarm64
48         environment:
49             - discovery.type=single-node
50             - "ES_JAVA_OPTS=-Xms512m -Xmx512m"
51             - ELASTIC_PASSWORD=es123456!
52             - xpack.security.enabled=false
53             - xpack.security.http.ssl.enabled=false
54             - TZ=Asia/Shanghai
55         volumes:
56             - ${DOCKER_VOLUME_DIRECTORY:-.}/data/es:/usr/share/elasticsearch/data
57         ports:
58             - "9200:9200"
59             - "9300:9300"
60         ulimits:
61             memlock:
62                 soft: -1
63                 hard: -1
64         healthcheck:
65             test: ["CMD-SHELL", "curl -s -f http://localhost:9200/_cluster/health ||
66 exit 1"]
66             interval: 30s
67             timeout: 10s
68             retries: 3
69         restart: on-failure
70
71     backend:
72         container_name: bisheng-backend
73         image: cr.dataelem.com/dataelement/bisheng-backend:v1.1.1-arm
74         ports:
75             - "7860:7860"
```

```

76     environment:
77         TZ: Asia/Shanghai
78         BS_MILVUS_CONNECTION_ARGS:
79             '{"host":"milvus","port":"19530","user":"","password":"","secure":false}'
80         BS_MILVUS_IS_PARTITION: 'true'
81         BS_MILVUS_PARTITION_SUFFIX: '1'
82         BS_ELASTICSEARCH_URL: 'https://elasticsearch:9200'
83         BS_ELASTICSEARCH_SSL_VERIFY: '{"basic_auth": ("elastic", "es123456!"),
84             "verify_certs": False}'
85         BS_MINIO_SCHEMA: 'false'
86         BS_MINIO_CERT_CHECK: 'false'
87         BS_MINIO_ENDPOINT: 'minio:9000'
88         BS_MINIO_SHAREPOINT: 'minio:9000'
89         BS_MINIO_ACCESS_KEY: 'minioadmin'
90         BS_MINIO_SECRET_KEY: 'minioadmin'
91         BISHENG_PRO: 'true'
92     volumes:
93         -
94     ${DOCKER_VOLUME_DIRECTORY:-.}/bisheng/config/config.yaml:/app/bisheng/config.yaml
95     - ${DOCKER_VOLUME_DIRECTORY:-.}/bisheng/entrypoint.sh:/app/entrypoint.sh
96     - ${DOCKER_VOLUME_DIRECTORY:-.}/data/bisheng:/app/data
97     security_opt:
98         - seccomp:unconfined
99     command: sh entrypoint.sh
100    restart: on-failure
101    healthcheck:
102        test: ["CMD", "curl", "-f", "http://localhost:7860/health"]
103        start_period: 30s
104        interval: 90s
105        timeout: 30s
106        retries: 3
107    depends_on:
108        mysql:
109            condition: service_healthy
110        redis:
111            condition: service_healthy
112    frontend:
113        container_name: bisheng-frontend
114        image: cr.dataelem.com/dataelement/bisheng-frontend:v1.1.1-arm
115        ports:
116            - "3001:3001"
117        environment:
118            TZ: Asia/Shanghai
119        volumes:
120            - ${DOCKER_VOLUME_DIRECTORY:-.}/nginx/nginx.conf:/etc/nginx/nginx.conf

```



```
119     - ${DOCKER_VOLUME_DIRECTORY:-.}/nginx/conf.d:/etc/nginx/conf.d
120 restart: on-failure
121 depends_on:
122     - backend
123
124 etcd:
125     container_name: milvus-etcd
126     image: cr.dataelem.com/bisheng-etcd:v3.5.13-linuxarm64
127     environment:
128         - ETCD_AUTO_COMPACTION_MODE=revision
129         - ETCD_AUTO_COMPACTION_RETENTION=1000
130         - ETCD_QUOTA_BACKEND_BYTES=4294967296
131         - ETCD_SNAPSHOT_COUNT=50000
132     volumes:
133         - ${DOCKER_VOLUME_DIRECTORY:-.}/volumes/etcd:/etcd
134     command: etcd -advertise-client-urls=http://127.0.0.1:2379 -listen-client-
135         urls http://0.0.0.0:2379 --data-dir /etcd
136     healthcheck:
137         test: ["CMD", "etcdctl", "endpoint", "health"]
138         interval: 30s
139         timeout: 20s
140         retries: 3
141
142 minio:
143     container_name: milvus-minio
144     image: cr.dataelem.com/bisheng-minio:RELEASE.2023-04-28T18-11-17Z-
145         linuxarm64
146     environment:
147         MINIO_ACCESS_KEY: minioadmin
148         MINIO_SECRET_KEY: minioadmin
149     ports:
150         - "9101:9001"
151         - "9100:9000"
152     volumes:
153         - ${DOCKER_VOLUME_DIRECTORY:-.}/volumes/minio:/minio_data
154     command: minio server /minio_data --console-address ":9001"
155     healthcheck:
156         test: ["CMD", "curl", "-f", "http://localhost:9000/minio/health/live"]
157         interval: 30s
158         timeout: 20s
159         retries: 3
160
161 milvus:
162     container_name: milvus-standalone
163     image: cr.dataelem.com/bisheng-milvus:v2.3.21-linuxarm64
164     command: ["milvus", "run", "standalone"]
165     security_opt:
```

```

164     - seccomp:unconfined
165     environment:
166         ETCD_ENDPOINTS: etcd:2379
167         MINIO_ADDRESS: minio:9000
168     volumes:
169     - ${DOCKER_VOLUME_DIRECTORY:-.}/volumes/milvus:/var/lib/milvus
170     healthcheck:
171         test: ["CMD", "curl", "-f", "http://localhost:9091/healthz"]
172         interval: 30s
173         start_period: 90s
174         timeout: 20s
175         retries: 3
176     ports:
177     - "19530:19530"
178     - "9091:9091"
179     depends_on:
180     - "etcd"
181     - "minio"
182
183     office:
184         container_name: bisheng-office
185         image: cr.dataelem.com/documentserver:7.2.1-arm64
186         ports:
187         - "8701:80"
188         environment:
189             TZ: Asia/Shanghai
190             JWT_ENABLED: false
191         volumes:
192         -
193           ${DOCKER_VOLUME_DIRECTORY:-.}/office/bisheng:/var/www/onlyoffice/documentserver
194           /sdkjs-plugins/bisheng
195         command: bash -c "supervisorctl restart all"
196         restart: on-failure
197
198 #修改完成后:wq保存退出

```

- bisheng-unstructured组件配置修改

代码块

```

1  #下载好BISHENG代码或者压缩包并解压进入
2  cd bisheng-main/docker
3  #修改image配置, 改为arm镜像
4  services:
5      bisheng-unstructured:
6          container_name: bisheng-unstructured

```

```

7      image: cr.dataelem.com/dataelement/bisheng-unstructured-arm:release
8      ports:
9        - "10001:10001"
10     environment:
11       # 填写ocr_sdk或rt服务的根地址
12       # server_address: bisheng-rt:9001
13       # 这里填 ocr_sdk 或 rt
14       # server_type: ocr_sdk
15       TZ: Asia/Shanghai
16     volumes:
17       - ${DOCKER_VOLUME_DIRECTORY:-.}/bisheng-uns/config.yaml:/opt/bisheng-
unstructured/bisheng_unstructured/config/config.yaml
18     healthcheck:
19       test: ["CMD", "curl", "-f", "http://localhost:10001/health"]
20       interval: 30s
21       timeout: 20s
22       retries: 3
23     restart: on-failure
24
25     #修改完成后:wq保存退出

```

4.3 服务启动与验证

- 启动服务命令（必须完成4.2修改配置文件之后再执行以下步骤）

代码块

```

1  #以下启动命令需要在 bisheng-main/docker 目录下执行
2  #BISHENG平台核心组件ls启动
3  docker-compose up -d
4  #bisheng-unstructured组件启动
5  docker-compose -f docker-compose-uns.yml up -d

```

- 查看服务状态

执行 `docker ps -a|grep bisheng` 命令查看服务状态，和下图一致即为启动成功

```

[root@ecs-8b0f docker]# docker ps -a | grep bisheng
ad4de4781168 cr.dataelem.com/dataelement/bisheng-unstructured-arm:release "bash bin/entrypoint..." 4 hours ago Up 4 hours (healthy) 0.0.0.0:10001->10001/tcp, :::10001->10001/tcp
9cd66f44e8fc cr.dataelem.com/dataelement/bisheng-frontend:v1.1.1-arm bisheng-frontend "/docker-entrypoint..." 4 hours ago Up 4 hours 80/tcp, 0.0.0.0:3001->3001/tcp, :::3001->3001/tcp
a838a4e83103 cr.dataelem.com/dataelement/bisheng-backend:v1.1.1-arm bisheng-backend "sh entrypoint.sh" 4 hours ago Up 4 hours (healthy) 0.0.0.0:7860->7860/tcp, :::7860->7860/tcp
1cde05301f69 cr.dataelem.com/bisheng-redis:7.2.4-linuxarm64 bisheng-redis "docker-entrypoint.s..." 4 hours ago Up 4 hours (healthy) 0.0.0.0:6379->6379/tcp, :::6379->6379/tcp
9237a6ca7b18 cr.dataelem.com/bisheng-elasticsearch:8.12.2-linuxarm64 bisheng-elasticsearch "/bin/tini -- /usr/L..." 5 hours ago Up 5 hours (healthy) 0.0.0.0:9200->9200/tcp, :::9200->9200/tcp, 0.0.0.0:9300->9300/tcp, :::9300->9300/tcp
078c59f271b5 cr.dataelem.com/bisheng-milvus:v2.3.21-linuxarm64 "/tini -- milvus run..." 23 hours ago Up 5 hours (healthy) 0.0.0.0:9091->9091/tcp, :::9091->9091/tcp, 0.0.0.0:19530->19530/tcp, :::19530->19530/tcp
71e37b909ade cr.dataelem.com/bisheng-minio:RELEASE.2023-04-28T18-11-17Z-linuxarm64 bisheng-minio "/usr/bin/docker-ent..." 23 hours ago Up 20 hours (healthy) 0.0.0.0:9100->9100/tcp, :::9100->9100/tcp, 0.0.0.0:9101->9101/tcp, :::9101->9101/tcp
176557ea5545 cr.dataelem.com/bisheng-documentserver:8.2.0-linuxarm64 bisheng-office "/app/ds/run-documen..." 23 hours ago Up 20 hours 443/tcp, 0.0.0.0:8701->80/tcp, :::8701->80/tcp
2041676a58c8 cr.dataelem.com/bisheng-mysql:8.0.39-linuxarm64 bisheng-mysql "docker-entrypoint.s..." 23 hours ago Up 5 hours (healthy) 0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp
31b5a1265e78 cr.dataelem.com/bisheng-etcd:v3.5.13-linuxarm64 milvus-etcd "etcd -advertise-cl..." 23 hours ago Up 20 hours (healthy) 2379-2380/tcp
[root@ecs-8b0f docker]#

```

如果服务启动有问题，请先参考“6.常见问题处理”!!!

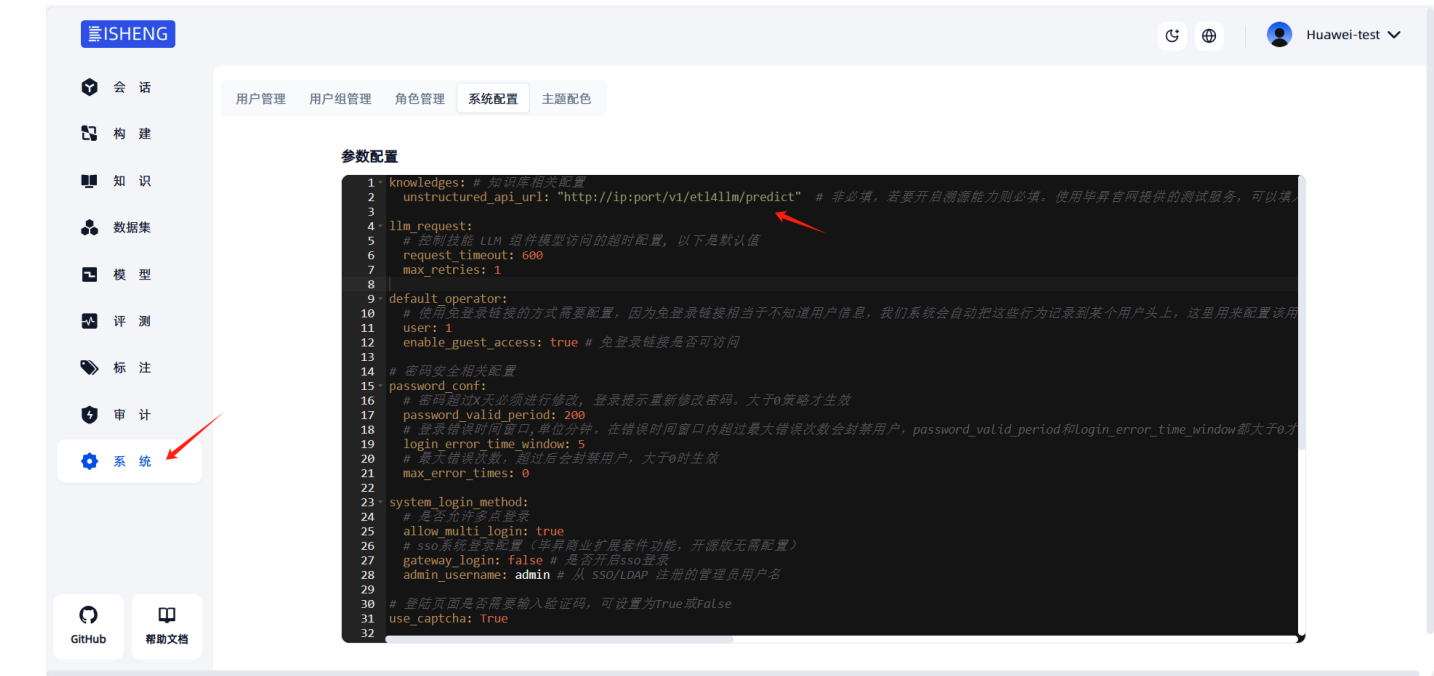
- 访问BISHENG页面

浏览器访问IP:3001端口，进行登录（首次登录需要注册账号，首次注册的账号默认设置为管理员）

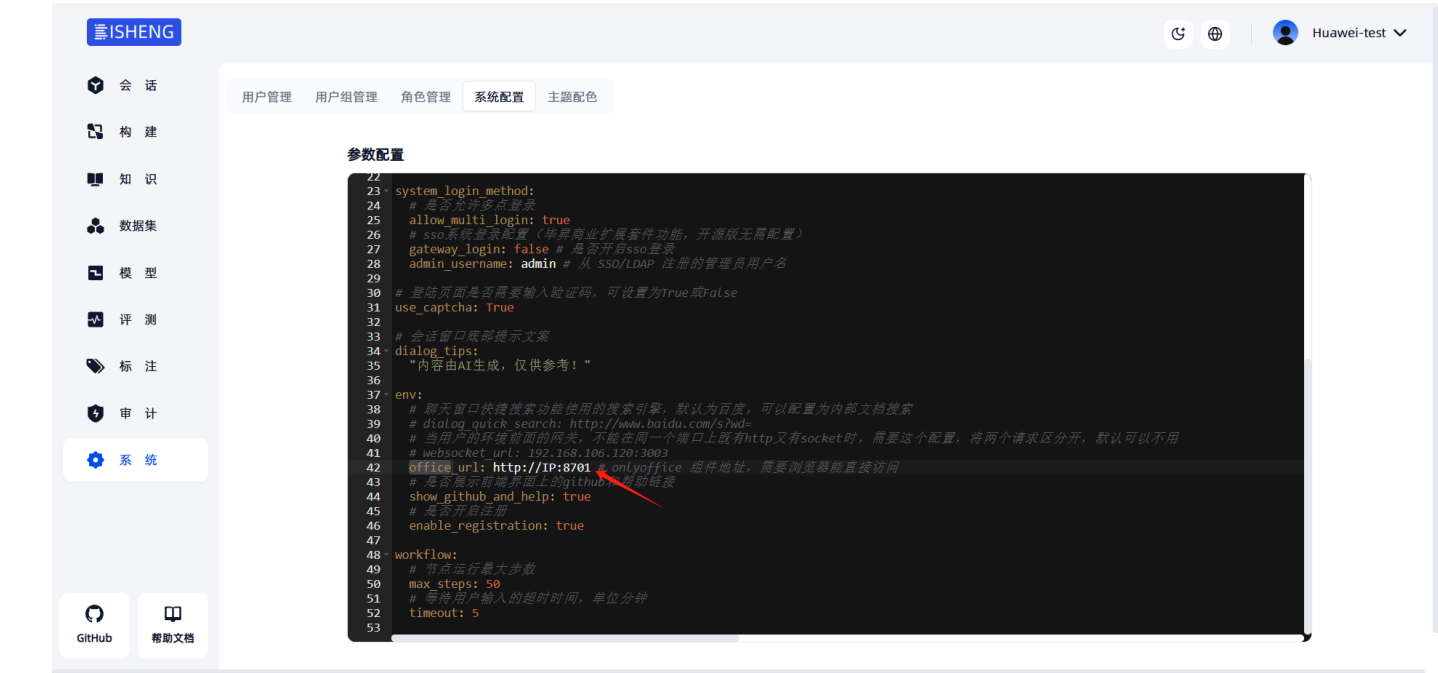


5. 配置说明

5.1 配置unstructured



5.2 配置office



- 工作流、助手、技能配置案例
- 应用案例/场景库

6. 常见问题处理

6.1 毕昇Logo与产品名修改方法

毕昇Logo与产品名修改方法指南

6.2 redis服务启动常见问题：

- redis服务启动失败，查看日志时出现以下报错

```
[root@ecs-b00f docker]# docker logs -f bisheng-redis
1:0 21 May 2025 15:01:27.277 # WARNING Memory overcommit must be enabled! Without it, a background save or replication may fail under low memory condition. Being disabled, it can also cause failures without low memory condition, see https://github.com/jemalloc/jemalloc/issues/1328. To fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
1:0 21 May 2025 15:01:27.278 # WARNING Your kernel has a bug that could lead to data corruption during background save. Please upgrade to the latest stable kernel.
1:0 21 May 2025 15:01:27.278 # Redis will now exit to prevent data corruption. Note that it is possible to suppress this warning by setting the following config: ignore-warnings ARM64-COW-BUG
1:0 21 May 2025 15:01:28.353 # WARNING Memory overcommit must be enabled! Without it, a background save or replication may fail under low memory condition. Being disabled, it can also cause failures without low memory condition, see https://github.com/jemalloc/jemalloc/issues/1328. To fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
1:0 21 May 2025 15:01:28.354 # WARNING Your kernel has a bug that could lead to data corruption during background save. Please upgrade to the latest stable kernel.
1:0 21 May 2025 15:01:28.354 # Redis will now exit to prevent data corruption. Note that it is possible to suppress this warning by setting the following config: ignore-warnings ARM64-COW-BUG
1:0 21 May 2025 15:01:29.413 # WARNING Memory overcommit must be enabled! Without it, a background save or replication may fail under low memory condition. Being disabled, it can also cause failures without low memory condition, see https://github.com/jemalloc/jemalloc/issues/1328. To fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
1:0 21 May 2025 15:01:29.414 # WARNING Your kernel has a bug that could lead to data corruption during background save. Please upgrade to the latest stable kernel.
1:0 21 May 2025 15:01:29.414 # Redis will now exit to prevent data corruption. Note that it is possible to suppress this warning by setting the following config: ignore-warnings ARM64-COW-BUG
1:0 21 May 2025 15:01:30.483 # WARNING Memory overcommit must be enabled! Without it, a background save or replication may fail under low memory condition. Being disabled, it can also cause failures without low memory condition, see https://github.com/jemalloc/jemalloc/issues/1328. To fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
1:0 21 May 2025 15:01:30.484 # WARNING Your kernel has a bug that could lead to data corruption during background save. Please upgrade to the latest stable kernel.
1:0 21 May 2025 15:01:30.484 # Redis will now exit to prevent data corruption. Note that it is possible to suppress this warning by setting the following config: ignore-warnings ARM64-COW-BUG
1:0 21 May 2025 15:01:31.023 # WARNING Memory overcommit must be enabled! Without it, a background save or replication may fail under low memory condition. Being disabled, it can also cause failures without low memory condition, see https://github.com/jemalloc/jemalloc/issues/1328. To fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
1:0 21 May 2025 15:01:31.024 # WARNING Your kernel has a bug that could lead to data corruption during background save. Please upgrade to the latest stable kernel.
1:0 21 May 2025 15:01:31.024 # Redis will now exit to prevent data corruption. Note that it is possible to suppress this warning by setting the following config: ignore-warnings ARM64-COW-BUG
1:0 21 May 2025 15:01:34.074 # WARNING Memory overcommit must be enabled! Without it, a background save or replication may fail under low memory condition. Being disabled, it can also cause failures without low memory condition, see https://github.com/jemalloc/jemalloc/issues/1328. To fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
1:0 21 May 2025 15:01:34.075 # WARNING Your kernel has a bug that could lead to data corruption during background save. Please upgrade to the latest stable kernel.
1:0 21 May 2025 15:01:37.915 # WARNING Memory overcommit must be enabled! Without it, a background save or replication may fail under low memory condition. Being disabled, it can also cause failures without low memory condition, see https://github.com/jemalloc/jemalloc/issues/1328. To fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
1:0 21 May 2025 15:01:37.915 # WARNING Your kernel has a bug that could lead to data corruption during background save. Please upgrade to the latest stable kernel.
1:0 21 May 2025 15:01:45.013 # WARNING Memory overcommit must be enabled! Without it, a background save or replication may fail under low memory condition. Being disabled, it can also cause failures without low memory condition, see https://github.com/jemalloc/jemalloc/issues/1328. To fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
1:0 21 May 2025 15:01:45.014 # WARNING Your kernel has a bug that could lead to data corruption during background save. Please upgrade to the latest stable kernel.
1:0 21 May 2025 15:01:45.014 # Redis will now exit to prevent data corruption. Note that it is possible to suppress this warning by setting the following config: ignore-warnings ARM64-COW-BUG
```

解决方式：

启用内存过提交（依次执行以下命令）：

```
sysctl vm.overcommit_memory=1
```

```
echo "vm.overcommit_memory = 1" | sudo tee -a /etc/sysctl.conf
```


sudo sysctl -p

修改redis.conf文件配置，并添加以下内容：

ignore-warnings ARM64-COW-BUG

修改redis配置文件权限：

chown 999:999 redis/redis.conf

chmod 664 redis/redis.conf

执行完以上操作，重启redis容器！

```
1:C 26 May 2025 15:54:35.518 # Fatal error, can't open config file '/etc/redis.conf': Permission denied
1:C 26 May 2025 15:55:36.102 # Fatal error, can't open config file '/etc/redis.conf': Permission denied
1:C 26 May 2025 15:56:36.613 # Fatal error, can't open config file '/etc/redis.conf': Permission denied
1:C 26 May 2025 15:57:37.204 # Fatal error, can't open config file '/etc/redis.conf': Permission denied
1:C 26 May 2025 15:58:37.762 # Fatal error, can't open config file '/etc/redis.conf': Permission denied
1:C 26 May 2025 15:59:38.414 # Fatal error, can't open config file '/etc/redis.conf': Permission denied
1:C 26 May 2025 16:00:39.002 # Fatal error, can't open config file '/etc/redis.conf': Permission denied
1:C 26 May 2025 16:01:39.603 # Fatal error, can't open config file '/etc/redis.conf': Permission denied
1:C 26 May 2025 16:02:40.183 # Fatal error, can't open config file '/etc/redis.conf': Permission denied
1:C 26 May 2025 16:03:40.782 # Fatal error, can't open config file '/etc/redis.conf': Permission denied
1:C 26 May 2025 16:04:41.362 # Fatal error, can't open config file '/etc/redis.conf': Permission denied
1:C 26 May 2025 16:05:42.003 # Fatal error, can't open config file '/etc/redis.conf': Permission denied
1:C 26 May 2025 16:06:15.922 # Fatal error, can't open config file '/etc/redis.conf': Permission denied
1:C 26 May 2025 16:06:17.023 # Fatal error, can't open config file '/etc/redis.conf': Permission denied
1:C 26 May 2025 16:06:17.972 # Fatal error, can't open config file '/etc/redis.conf': Permission denied
1:C 26 May 2025 16:06:19.034 # Fatal error, can't open config file '/etc/redis.conf': Permission denied
1:C 26 May 2025 16:06:20.807 # Fatal error, can't open config file '/etc/redis.conf': Permission denied
1:C 26 May 2025 16:06:23.072 # Fatal error, can't open config file '/etc/redis.conf': Permission denied
1:C 26 May 2025 16:06:26.861 # Fatal error, can't open config file '/etc/redis.conf': Permission denied
1:C 26 May 2025 16:06:33.853 # Fatal error, can't open config file '/etc/redis.conf': Permission denied
1:C 26 May 2025 16:06:47.223 # Fatal error, can't open config file '/etc/redis.conf': Permission denied
1:C 26 May 2025 16:07:13.372 # Fatal error, can't open config file '/etc/redis.conf': Permission denied
1:C 26 May 2025 16:08:05.102 # Fatal error, can't open config file '/etc/redis.conf': Permission denied
1:C 26 May 2025 16:09:05.705 # Fatal error, can't open config file '/etc/redis.conf': Permission denied
1:C 26 May 2025 16:10:06.292 # Fatal error, can't open config file '/etc/redis.conf': Permission denied
1:C 26 May 2025 16:11:06.843 # Fatal error, can't open config file '/etc/redis.conf': Permission denied
```

6.3 ES服务启动常见问题：

- ES容器自动重启，查看日志出现以下错误：

```
[{"timestamp": "2025-05-21T07:30:14.866Z", "log.level": "INFO", "message": "loaded module [runtime-fields-common]", "ecs.version": "1.2.0", "service.name": "ES_ECS", "event.dataset": "elasticsearch.server", "process.thread.name": "main", "log.logger": "org.elasticsearch.plugins.PluginsService", "elasticsearch.node.name": "d935db2ef468", "elasticsearch.cluster.name": "docker-cluster"}, {"timestamp": "2025-05-21T07:30:14.866Z", "log.level": "INFO", "message": "loaded module [x-pack-async]", "ecs.version": "1.2.0", "service.name": "ES_ECS", "event.dataset": "elasticsearch.server", "process.thread.name": "main", "log.logger": "org.elasticsearch.plugins.PluginsService", "elasticsearch.node.name": "d935db2ef468", "elasticsearch.cluster.name": "docker-cluster"}, {"timestamp": "2025-05-21T07:30:14.866Z", "log.level": "INFO", "message": "loaded module [vector-tile]", "ecs.version": "1.2.0", "service.name": "ES_ECS", "event.dataset": "elasticsearch.server", "process.thread.name": "main", "log.logger": "org.elasticsearch.plugins.PluginsService", "elasticsearch.node.name": "d935db2ef468", "elasticsearch.cluster.name": "docker-cluster"}, {"timestamp": "2025-05-21T07:30:14.867Z", "log.level": "INFO", "message": "loaded module [lang-expression]", "ecs.version": "1.2.0", "service.name": "ES_ECS", "event.dataset": "elasticsearch.server", "process.thread.name": "main", "log.logger": "org.elasticsearch.plugins.PluginsService", "elasticsearch.node.name": "d935db2ef468", "elasticsearch.cluster.name": "docker-cluster"}, {"timestamp": "2025-05-21T07:30:14.867Z", "log.level": "INFO", "message": "loaded module [x-pack-eql]", "ecs.version": "1.2.0", "service.name": "ES_ECS", "event.dataset": "elasticsearch.server", "process.thread.name": "main", "log.logger": "org.elasticsearch.plugins.PluginsService", "elasticsearch.node.name": "d935db2ef468", "elasticsearch.cluster.name": "docker-cluster"}, {"timestamp": "2025-05-21T07:30:15.633Z", "log.level": "ERROR", "message": "fatal exception while booting Elasticsearch", "ecs.version": "1.2.0", "service.name": "ES_ECS", "event.dataset": "elasticsearch.server", "process.thread.name": "main", "log.logger": "org.elasticsearch.bootstrap.Elasticsearch", "elasticsearch.node.name": "d935db2ef468", "elasticsearch.cluster.name": "docker-cluster", "error.type": "java.lang.IllegalStateException", "error.message": "failed to obtain node locks, tried [/usr/share/elasticsearch/data]: maybe these locations are not writable or multiple nodes were started on the same data path?", "error.stack.trace": "java.lang.IllegalStateException: failed to obtain node locks, tried [/usr/share/elasticsearch/data]: maybe these locations are not writable or multiple nodes were started on the same data path?\n\tat org.elasticsearch.server@8.12.2/org.elasticsearch.env.NodeEnvironment.<init>(NodeEnvironment.java:296)\n\tat org.elasticsearch.server@8.12.2/org.elasticsearch.env.NodeEnvironment.<init>(NodeEnvironment.java:248)\n\tat org.elasticsearch.bootstrap@8.12.2/org.elasticsearch.bootstrap.Elasticsearch2.<init>(Elasticsearch2.java:220)\n\tat org.elasticsearch.bootstrap@8.12.2/org.elasticsearch.bootstrap.Elasticsearch.main(Elasticsearch.java:73)\nCaused by: java.io.IOException: failed to obtain lock on /usr/share/elasticsearch/data\n\tat org.elasticsearch.server@8.12.2/org.elasticsearch.env.NodeEnvironmentModelLock.<init>(NodeEnvironmentModelLock.java:241)\n\tat org.elasticsearch.server@8.12.2/org.elasticsearch.env.NodeEnvironmentModelLock.<init>(NodeEnvironmentModelLock.java:288)\n\tat ... 5 more\nCaused by: java.nio.file.NoSuchFileException: /usr/share/elasticsearch/data/node.lock\n\tat java.base/sun.nio.fs.UnixException.translateToIOException(UnixException.java:92)\n\tat java.base/sun.nio.fs.UnixException.rethrowAsIOException(UnixException.java:106)\n\tat java.base/sun.nio.fs.UnixPath.toRealPath(UnixPath.java:834)\n\tat org.apache.lucene.core@9.2/org.apache.lucene.store.NativeFSLockFactory.obtainFSLock(NativeFSLockFactory.java:94)\n\tat org.apache.lucene.core@9.2/org.apache.lucene.store.FSLockFactory.obtainLock(FSLockFactory.java:43)\n\tat org.apache.lucene.core@9.2/org.apache.lucene.store.BaseDirectory.obtainLock(BaseDirectory.java:44)\n\tat org.elasticsearch.server@8.12.2/org.elasticsearch.env.NodeEnvironmentModelLock.<init>(NodeEnvironmentModelLock.java:234)\n\tat ... 8 more\n\tat java.io.AccessDeniedException: /usr/share/elasticsearch/data/node.lock\n\tat java.base/sun.nio.fs.UnixException.translateToIOException(UnixException.java:90)\n\tat java.base/sun.nio.fs.UnixException.rethrowAsIOException(UnixException.java:106)\n\tat java.base/sun.nio.fs.UnixPath.toRealPath(UnixPath.java:834)\n\tat org.apache.lucene.core@9.2/org.apache.lucene.store.NativeFSLockFactory.obtainFSLock(NativeFSLockFactory.java:94)\n\tat ... 11 more\nERROR: Elasticsearch did not exit normally - check the logs at /usr/share/elasticsearch/logs/docker-cluster.log\nERROR: Elasticsearch exited unexpectedly, with exit code 1
```

解决方式：

如果已经关闭了SELinux 还会有以上报错那就是已经挂载的ES存储目录权限有问题，执行以下命令修改目录权限：

chown -R 1000:1000 ./data/es

chmod -R 755 ./data/es

修改完目录权限后重启ES服务：docker-compose up -d

6.4 backend容器启动常见问题：

- backend容器启动报错：

```
ERROR: Traceback (most recent call last):
  File "/usr/local/lib/python3.10/site-packages/sqlalchemy/engine/base.py", line 146, in __init__
    self._dbapi_connection = engine.raw_connection()
  File "/usr/local/lib/python3.10/site-packages/sqlalchemy/engine/base.py", line 3298, in raw_connection
    return self.pool.connect()
  File "/usr/local/lib/python3.10/site-packages/sqlalchemy/pool/base.py", line 449, in connect
    return _ConnectionFairy.checkout(self)
  File "/usr/local/lib/python3.10/site-packages/sqlalchemy/pool/base.py", line 1264, in _checkout
    fairy = _ConnectionRecord.checkout(pool)
  File "/usr/local/lib/python3.10/site-packages/sqlalchemy/pool/base.py", line 713, in checkout
    rec = pool._do_get()
  File "/usr/local/lib/python3.10/site-packages/sqlalchemy/pool/impl.py", line 179, in _do_get
    with util.safe_reraise():
  File "/usr/local/lib/python3.10/site-packages/sqlalchemy/util/langhelpers.py", line 146, in __exit__
    raise exc_value.with_traceback(exc_tb)
  File "/usr/local/lib/python3.10/site-packages/sqlalchemy/pool/impl.py", line 177, in _do_get
    return self._create_connection()
  File "/usr/local/lib/python3.10/site-packages/sqlalchemy/pool/base.py", line 390, in _create_connection
    return _ConnectionRecord(self)
  File "/usr/local/lib/python3.10/site-packages/sqlalchemy/pool/base.py", line 675, in __init__
    self.__connect()
  File "/usr/local/lib/python3.10/site-packages/sqlalchemy/pool/base.py", line 901, in __connect
    with util.safe_reraise():
  File "/usr/local/lib/python3.10/site-packages/sqlalchemy/util/langhelpers.py", line 146, in __exit__
    raise exc_value.with_traceback(exc_tb)
  File "/usr/local/lib/python3.10/site-packages/sqlalchemy/pool/base.py", line 897, in __connect
    self._dbapi_connection = connection = pool._invoke_creator(self)
  File "/usr/local/lib/python3.10/site-packages/sqlalchemy/engine/create.py", line 646, in connect
    return dialect.connect(*cargs, **cparams)
  File "/usr/local/lib/python3.10/site-packages/sqlalchemy/engine/default.py", line 625, in connect
    return self.loaded_dbapi.connect(*cargs, **cparams) # type: ignore[no-any-return] # NOQA: E501
  File "/usr/local/lib/python3.10/site-packages/pymysql/_init_.py", line 94, in Connect
    return Connection(*args, **kwargs)
  File "/usr/local/lib/python3.10/site-packages/pymysql/connections.py", line 327, in __init__
    self.connect()
  File "/usr/local/lib/python3.10/site-packages/pymysql/connections.py", line 588, in connect
    self._request_authentication()
  File "/usr/local/lib/python3.10/site-packages/pymysql/connections.py", line 874, in _request_authentication
    auth_packet = _auth.caching_sha2_password_auth(self, auth_packet)
  File "/usr/local/lib/python3.10/site-packages/pymysql/_auth.py", line 300, in caching_sha2_password_auth
    pkt = conn._read_packet()
  File "/usr/local/lib/python3.10/site-packages/pymysql/connections.py", line 676, in _read_packet
    packet.raise_for_error()
  File "/usr/local/lib/python3.10/site-packages/pymysql/protocol.py", line 223, in raise_for_error
    err.raise_mysql_exception(self._data)
  File "/usr/local/lib/python3.10/site-packages/pymysql/err.py", line 107, in raise_mysql_exception
    raise errorclass(errno, errval)
pymysql.err.OperationalError: (1049, "Unknown database 'bisheng'")
The above exception was the direct cause of the following exception:
```

解决方式：

进入MySQL容器，手动创建对应的database和table

进入bisheng-mysql容器：docker exec -it bisheng-mysql bash

登录mysql：mysql -u root -p （默认密码是1234）



 bisheng.sql

以上sql执行成功后，重启backend容器！

6.5 端口冲突

如果docker-compose启动时遇到端口冲突问题，可以通过修改yml文件中映射的端口来解决

```
milvus:
  container_name: milvus-standalone
  image: cr.dataelem.com/bisheng-milvus:v2.3.21-linuxarm64
  command: ["milvus", "run", "standalone"]
  security_opt:
    - seccomp:unconfined
  environment:
    ETCD_ENDPOINTS: etcd:2379
    MINIO_ADDRESS: minio:9000
  volumes:
    - ${DOCKER_VOLUME_DIRECTORY:-.}/volumes/milvus:/var/lib/milvus
  healthcheck:
    test: ["CMD", "curl", "-f", "http://localhost:9091/healthz"]
    interval: 30s
    start_period: 90s
    timeout: 20s
    retries: 3
  ports:
    - "19530:19530"
    - "9091:9091"
  depends_on:
    - "etcd"
    - "minio"
```


修改以上部分为：xxx:19530 (xxx替换为需要配置的端口号，范围1-65535)

6.6 其他典型错误及解决办法

(正在收集中。。。。。。)

7. 维护指南

7.1 升级流程

- 需要升级BISHENG版本请按照官方文档进行操作：[📖 版本升级注意事项](#)

8. 附录：

8.1 附录1：

代码块

```
1  #BISHENG商业套件Gateway服务ARM镜像下载地址
2  docker pull cr.dataelem.com/bisheng-gateway:release-arm64
3  #部署文档
4  https://dataelem.feishu.cn/wiki/KNWrwK1DtIFli7klsLzczx7enNh?from=from_copylink
```