目录

第一章节：IIOT 平台-2024010-9 迭代

版本号：SIE IIOT V3.3.241129

更新内容：

APP 名称	显示名称	版本号	备注
iiot_data_fabric	数据编织	v3.3.16.241102	新增
iiot_store_monitor	存储监控	v3.3.1.241101	新增
iiot_store_management	存储管理	v3.3.15.241101	更新
iiot_thing	建模管理	v3.3..44.241103	更新

更新注意事项：

前置条件：

- **存储管理**

  1、依次登录 TDengine 集群各节点服务器（步骤 2~4 每个节点都需要执行）。

  2、升级 TDengine 版本到 v3.3.3.0。

  2.1、确保本地已安装并启动 docker 服务，执行以下命令下载 tdengine-3.3.3.0 镜像。

```
docker pull tdengine/tdengine:3.3.3.0
```

  2.2、执行以下命令将 tdengine:3.3.3.0 镜像导出至文件。

```
docker save -o tdengine-3.3.3.0.tar.gz tdengine/tdengine:3.3.3.0
```

  2.3、将 tdengine:3.3.3.0 镜像文件上传到 TDengine 集群节点服务器的/root 目录。

  2.4、将 tdengine:3.3.3.0 镜像文件导入到目标服务器的 docker。

```
docker load < /root/tdengine-3.3.3.0.tar.gz
```

3、编辑/var/tdengine/td-compose.yaml 文件，在 volumes 部分新增红色底色的条目（其它配置维持原样），修改 TDengine 镜像版本。

```
version: "3"

services:
  td1:
    image: harbor.sieiot.com/iidp/tdengine:3.3.3.0
    container_name: tdengine
    restart: always
    environment:
      TAOS_FQDN: "td1"
      TAOS_FIRST_EP: "td1:6030"
      TAOS_SECOND_EP: "td2:6030"
    ports:
      - 6041:6041
      - 6030:6030
    volumes:
      - /var/tdengine/data:/var/lib/taos
      - /var/tdengine/log:/var/log/taos
      - /var/tdengine/conf/taos.cfg:/etc/taos/taos.cfg
      - /var/tdengine/conf/taoskeeper.toml:/etc/taos/taoskeeper.toml
    extra_hosts:
      - "td2:192.168.168.27"
      - "td3:192.168.168.28"
```

4、创建/var/tdengine/conf/taoskeeper.toml 文件，将以下配置信息复制到该文件，红色底色条目根据实际节点信息修改（若集群有 3 个节点，实例 ID 建议依次设置成 1001、1002、1003；host 根据当前节点 /var/tdengine/td-compose.yaml 里 service 名称修改）。

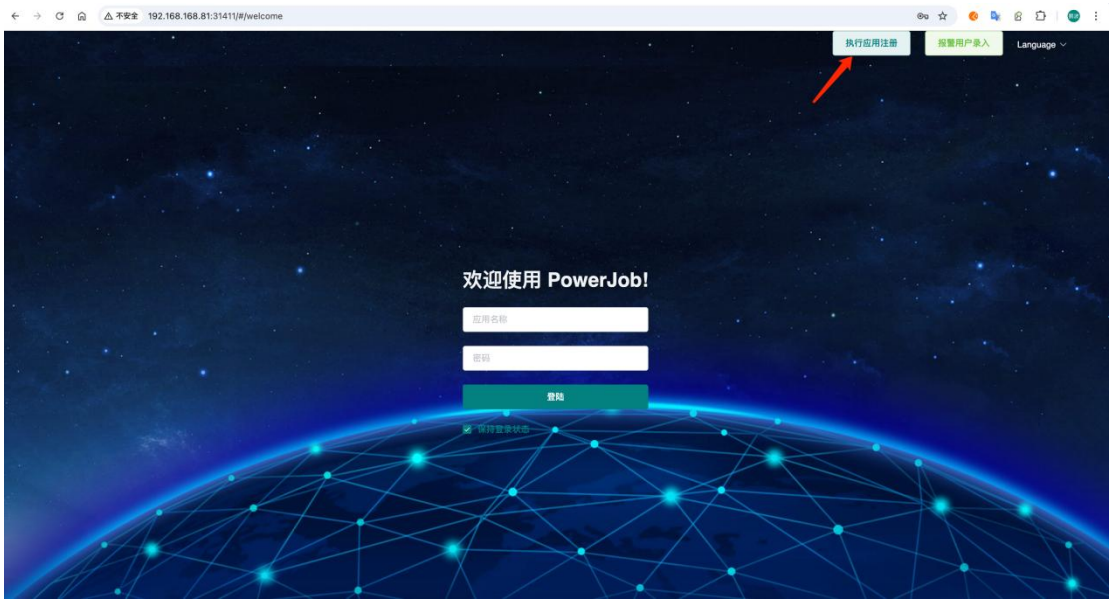
```
instanceId = 1001

# Listen port, default is 6043
port = 6043

# go pool size
gopoolsize = 50000

# interval for metrics
RotationInterval = "15s"

[tdengine]
```

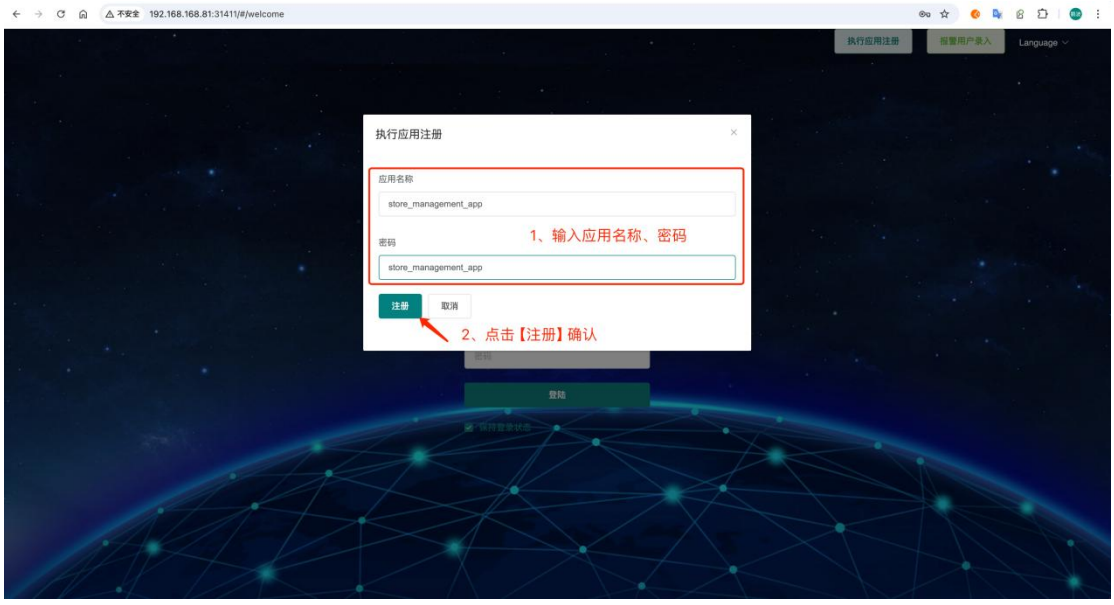
```
host = "td1"
port = 6041
username = "root"
password = "taosdata"
usessl = false

[metrics]
# metrics prefix in metrics names.
prefix = "taos"

# export some tables that are not super table
tables = []

# database for storing metrics data
[metrics.database]
name = "log"
# database options for db storing metrics data
[metrics.database.options]
vgroups = 1
buffer = 64
keep = 90
cachemodel = "both"

[environment]
# Whether running in cgroup.
incgroup = false

[log]
# The directory where log files are stored.
# path = "/var/log/taos"
level = "info"
# Number of log file rotations before deletion.
rotationCount = 30
# The number of days to retain log files.
keepDays = 30
# The maximum size of a log file before rotation.
rotationSize = "1GB"
# If set to true, log files will be compressed.
compress = false
# Minimum disk space to reserve. Log files will not be written if disk space falls
below this limit.
reservedDiskSize = "1GB"
```

5、执行以下命令重启 TDengine 数据库节点。

```
cd /var/tdengine
docker-compose -f td-compose.yaml restart
```

6、检查是否安装 powerjob-server，若已安装，则跳过该步骤；否则参考【SMDC 官网】-【IIOT 部署文档_V3.1.docx】安装 powerjob-server 服务，在浏览器打开 powerjob-server 网页（地址根据实际部署情况确定），注册 store_management_app。





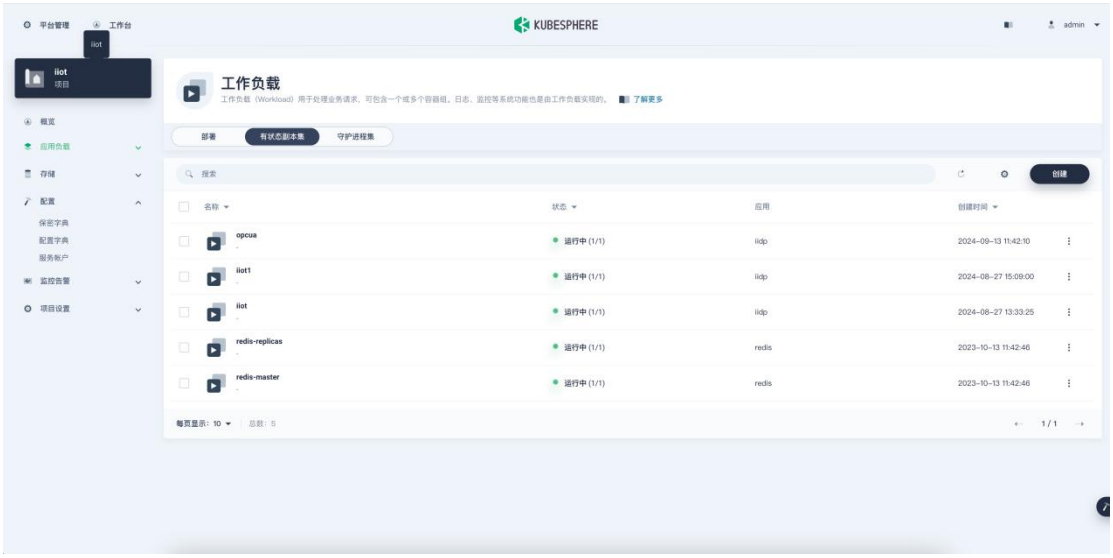
7、配置 powerjob 参数。

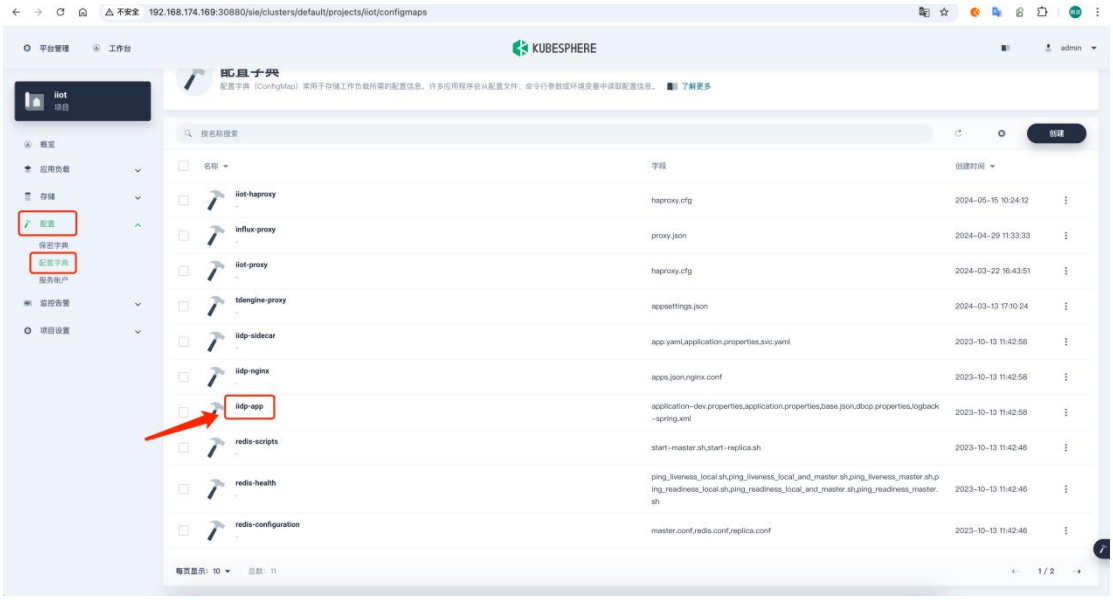
6.1、【单机版】配置

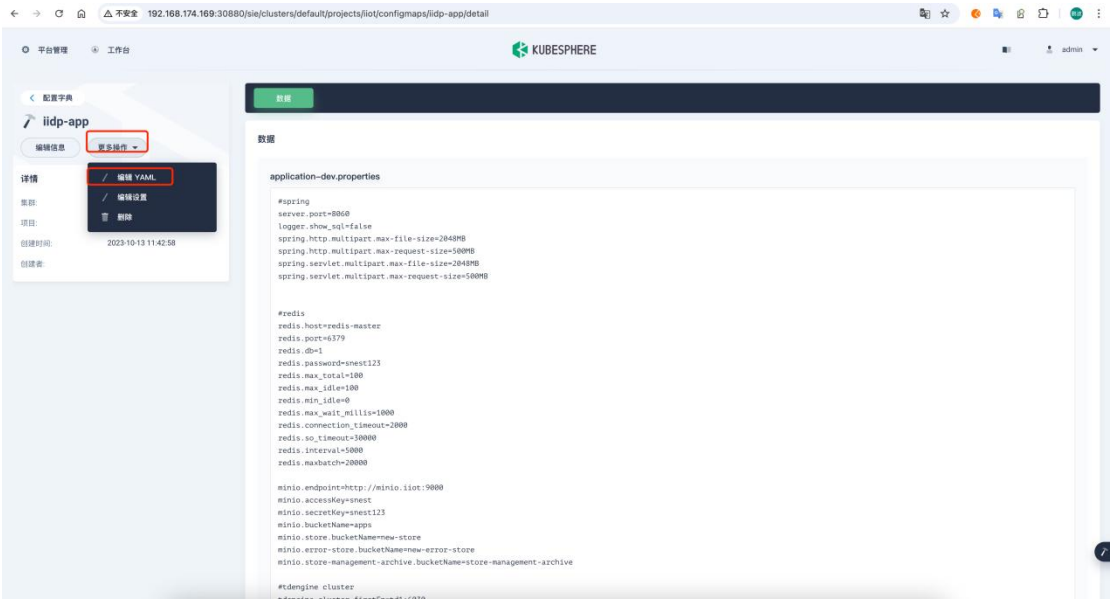
编辑/snest/config/application-dev.propertie 文件配置参数，参数参见 6.3。

6.2、【分布式版】配置

登录 KubeSphere 管理页面，在【配置】-【配置字典】-【iidp-app】-【更多操作】-【编辑 YAML】配置参数，参数参见 6.3。





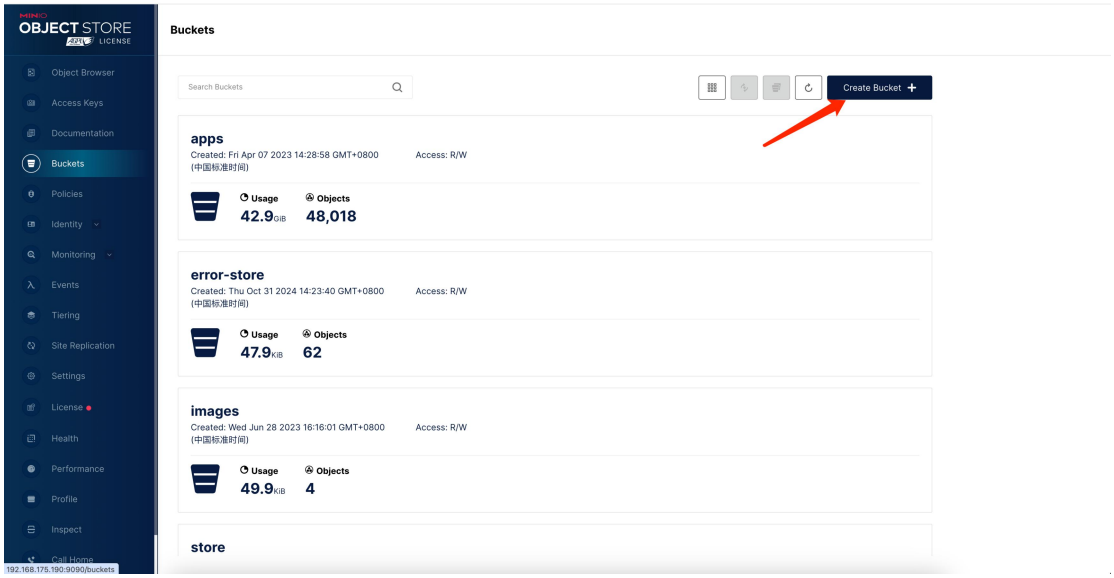


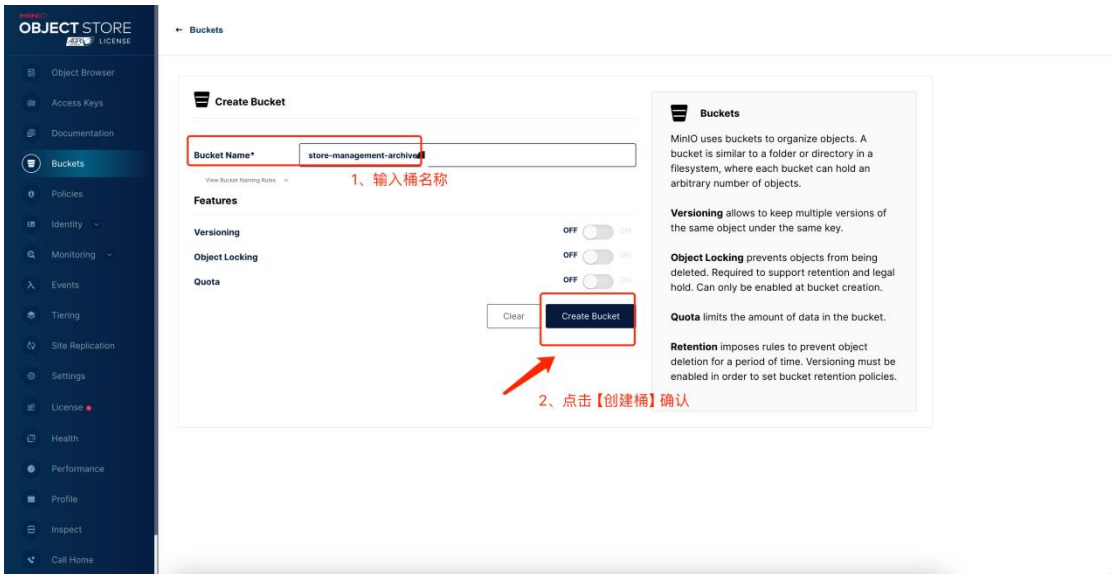
6.3、配置参数详情：

```
# powerjob
powerjob.worker.store-management-app-name=store_management_app
powerjob.worker.store-management-app-password=store_management_app
powerjob.worker.protocol=http
powerjob.worker.server-address=powerjob-server:7700
powerjob.worker.store-strategy=DISK
powerjob.worker.max-result-length=4096
powerjob.worker.max-appended-wf-context-length=4096
powerjob.worker.max-lightweight-task-num=1024
powerjob.worker.max-heavyweight-task-num=64
powerjob.worker.health-report-interval=10
powerjob.worker.enabled=true
```

8、配置 minio 参数。

7.1、登录 minio 管理页面，创建 store-management-app 桶。





7.2、参考步骤 6，配置 minio 参数（找到 minio 参数配置位置）

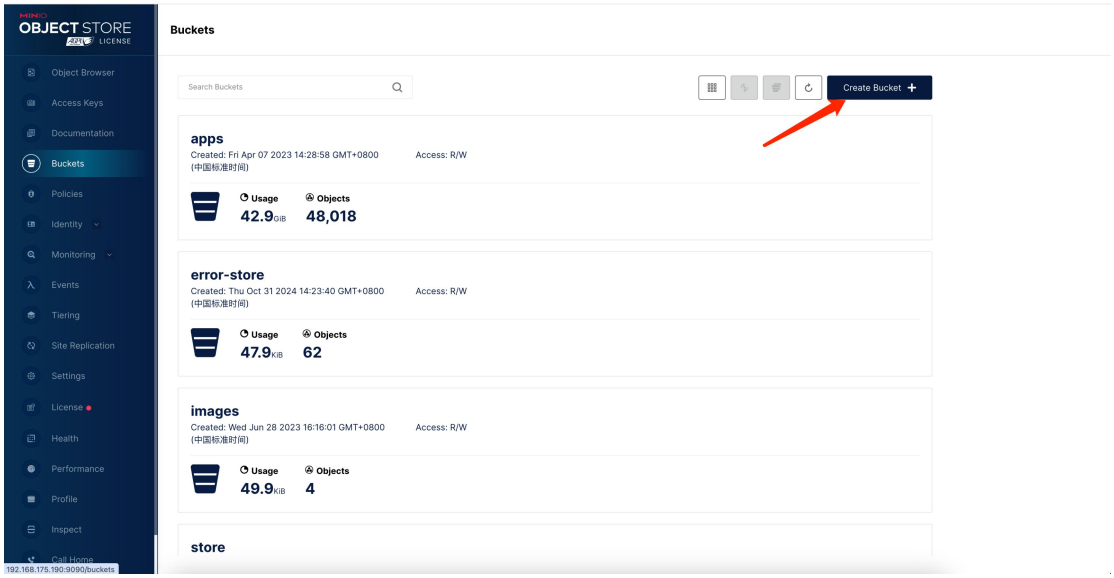
minio.store-management-archive.bucketName=store-management-archive

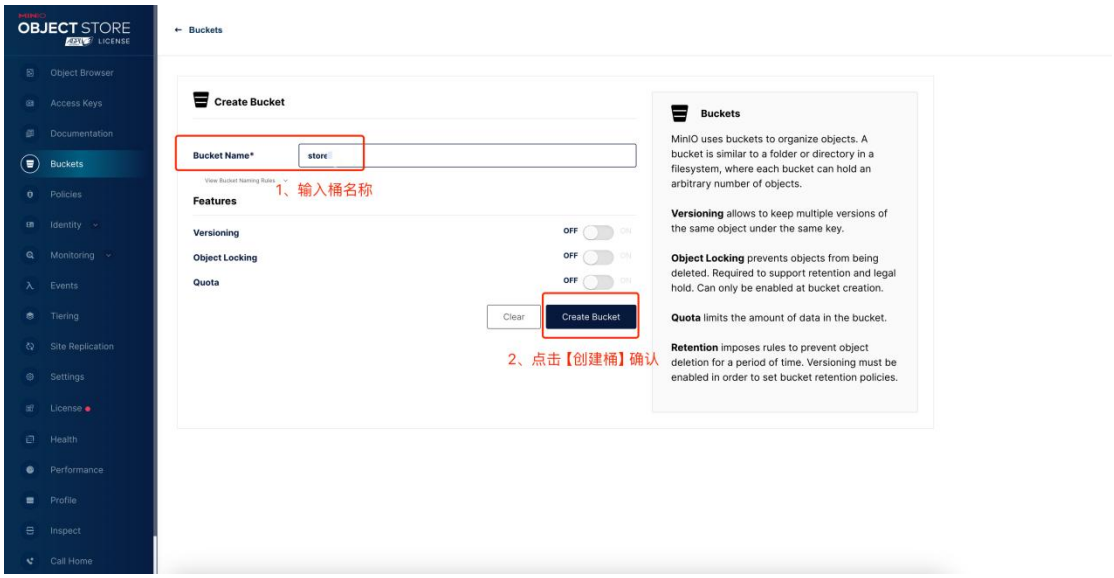
9、同步更新报警管理（iiot_alarm）、TDengine 数据存储（iiot_tdstore）、数据库管理（iiot_db_management）。

- 建模管理

1、配置 minio 参数。

1.1、登录 minio 管理页面，分别创建 store、error-store 桶。





1.2、参考【存储管理】服务步骤 6，配置 minio 参数（找到 minio 参数配置位置）

```
minio.store.bucketName=store
minio.error-store.bucketName=error-store
```

修复脚本：

- **存储管理**（注意：DB_NAME 根据实际部署使用的数据库修改）

```
UPDATE DB_NAME.iiot_system_configuration SET `key` = 'store.alarm_type.bak' WHERE `key` = 'store.alarm_type';
```

```
UPDATE DB_NAME.iiot_system_configuration SET `key` = 'store.alarm_condition.bak'
WHERE `key` = 'store.alarm_condition';
UPDATE DB_NAME.iiot_system_configuration SET `key` = 'store.alarm_threshold.bak'
WHERE `key` = 'store.alarm_threshold';
UPDATE DB_NAME.iiot_system_configuration SET `key` = 'store.alarm_content.bak'
WHERE `key` = 'store.alarm_content';

INSERT INTO DB_NAME.iiot_system_configuration (id, `key`, value, description,
tenant_id, create_user, create_date, update_user, update_date)
VALUES('044vwkyeryo11', 'store.iiot_influxdb.1.alarm_type', 'disk', '时序数据存
储的告警策略配置，如需修改，请移步到系统运维-存储管理页面修改', 'root',
'rbac_user_superuser', '2024-08-25 19:06:26', '044opkssnaef4', '2024-11-12
14:09:15');
INSERT INTO DB_NAME.iiot_system_configuration (id, `key`, value, description,
tenant_id, create_user, create_date, update_user, update_date)
VALUES('044vwkyeryo12', 'store.iiot_influxdb.1.alarm_condition', '>', '时序数据
存储的告警条件配置，如需修改，请移步到系统运维-存储管理页面修改', 'root',
'rbac_user_superuser', '2024-08-25 19:06:26', '044opkssnaef4', '2024-11-12
14:09:15');
INSERT INTO DB_NAME.iiot_system_configuration (id, `key`, value, description,
tenant_id, create_user, create_date, update_user, update_date)
VALUES('044vwkyeryo13','store.iiot_influxdb.1.alarm_threshold','50.0','时序数
据存储的告警阈值（G）配置，如需修改，请移步到系统运维-存储管理页面修改','root',
'rbac_user_superuser', '2024-08-25 19:06:26', '044opkssnaef4', '2024-11-12
14:09:15');
INSERT INTO DB_NAME.iiot_system_configuration (id, `key`, value, description,
tenant_id, create_user, create_date, update_user, update_date)
VALUES('044vwkyeryo14','store.iiot_influxdb.1.alarm_content','节点【#host#】磁
盘占用达到预警值【#store#G】','时序数据存储的告警内容配置，如需修改，请移步到系
统运维-存储管理页面修改', 'root', 'rbac_user_superuser', '2024-08-25 19:06:26',
'044opkssnaef4', '2024-11-12 14:09:15');


INSERT INTO DB_NAME.iiot_system_configuration (id, `key`, value, description,
tenant_id, create_user, create_date, update_user, update_date)
VALUES('044vwkyeryd21', 'store.iiot_tdengine.1.alarm_type', 'disk', '时序数据存
储的告警策略配置，如需修改，请移步到系统运维-存储管理页面修改', 'root',
'rbac_user_superuser', '2024-08-25 19:06:26', '044opkssnaef4', '2024-11-12
14:09:15');
INSERT INTO DB_NAME.iiot_system_configuration (id, `key`, value, description,
tenant_id, create_user, create_date, update_user, update_date)
VALUES('044vwkyeryd22', 'store.iiot_tdengine.1.alarm_condition', '>', '时序数据
存储的告警条件配置，如需修改，请移步到系统运维-存储管理页面修改', 'root',
```

'rbac_user_superuser', '2024-08-25 19:06:26', '044opkssnaef4', '2024-11-12 14:09:15');
INSERT INTO DB_NAME.iiot_system_configuration (id, `key`, value, description, tenant_id, create_user, create_date, update_user, update_date)
VALUES('044vwkyeryd23','store.iiot_tdengine.1.alarm_threshold','50.0','时序数据存储的告警阈值（G）配置，如需修改，请移步到系统运维-存储管理页面修改','root', 'rbac_user_superuser', '2024-08-25 19:06:26', '044opkssnaef4', '2024-11-12 14:09:15');
INSERT INTO DB_NAME.iiot_system_configuration (id, `key`, value, description, tenant_id, create_user, create_date, update_user, update_date)
VALUES('044vwkyeryd24','store.iiot_tdengine.1.alarm_content','节点【#host#】磁盘占用达到预警值【#store#G】','时序数据存储的告警内容配置，如需修改，请移步到系统运维-存储管理页面修改', 'root', 'rbac_user_superuser', '2024-08-25 19:06:26', '044opkssnaef4', '2024-11-12 14:09:15');

INSERT INTO DB_NAME.iiot_system_configuration (id, `key`, value, description, tenant_id, create_user, create_date, update_user, update_date)
VALUES('044vwkyeryd31', 'store.iiot_tdengine.2.alarm_type', 'disk', '时序数据存储的告警策略配置，如需修改，请移步到系统运维-存储管理页面修改', 'root', 'rbac_user_superuser', '2024-08-25 19:06:26', '044opkssnaef4', '2024-09-10 11:55:52');
INSERT INTO DB_NAME.iiot_system_configuration (id, `key`, value, description, tenant_id, create_user, create_date, update_user, update_date)
VALUES('044vwkyeryd32', 'store.iiot_tdengine.2.alarm_condition', '>', '时序数据存储的告警条件配置，如需修改，请移步到系统运维-存储管理页面修改', 'root', 'rbac_user_superuser', '2024-08-25 19:06:26', '044opkssnaef4', '2024-09-10 11:55:52');
INSERT INTO DB_NAME.iiot_system_configuration (id, `key`, value, description, tenant_id, create_user, create_date, update_user, update_date)
VALUES('044vwkyeryd33','store.iiot_tdengine.2.alarm_threshold','50.0','时序数据存储的告警阈值（G）配置，如需修改，请移步到系统运维-存储管理页面修改','root', 'rbac_user_superuser', '2024-08-25 19:06:26', '044opkssnaef4', '2024-09-10 11:55:52');
INSERT INTO DB_NAME.iiot_system_configuration (id, `key`, value, description, tenant_id, create_user, create_date, update_user, update_date)
VALUES('044vwkyeryd34','store.iiot_tdengine.2.alarm_content','节点【#host#】磁盘占用达到预警值【#store#G】','时序数据存储的告警内容配置，如需修改，请移步到系统运维-存储管理页面修改', 'root', 'rbac_user_superuser', '2024-08-25 19:06:26', '044opkssnaef4', '2024-09-10 11:55:52');

INSERT INTO DB_NAME.iiot_system_configuration (id, `key`, value, description, tenant_id, create_user, create_date, update_user, update_date)
VALUES('044vwkyeryd41', 'store.iiot_tdengine.3.alarm_type', 'disk', '时序数据存储的告警策略配置，如需修改，请移步到系统运维-存储管理页面修改', 'root',

'rbac_user_superuser', '2024-08-25 19:06:26', '044opkssnaef4', '2024-09-10 11:55:52');
INSERT INTO DB_NAME.iiot_system_configuration (id, `key`, value, description, tenant_id, create_user, create_date, update_user, update_date)
VALUES('044vwkyeryd42', 'store.iiot_tdengine.3.alarm_condition', '>', '时序数据存储的告警条件配置，如需修改，请移步到系统运维-存储管理页面修改', 'root', 'rbac_user_superuser', '2024-08-25 19:06:26', '044opkssnaef4', '2024-09-10 11:55:52');
INSERT INTO DB_NAME.iiot_system_configuration (id, `key`, value, description, tenant_id, create_user, create_date, update_user, update_date)
VALUES('044vwkyeujd43', 'store.iiot_tdengine.3.alarm_threshold', '50.0', '时序数据存储的告警阈值（G）配置，如需修改，请移步到系统运维-存储管理页面修改', 'root', 'rbac_user_superuser', '2024-08-25 19:06:26', '044opkssnaef4', '2024-09-10 11:55:52');
INSERT INTO DB_NAME.iiot_system_configuration (id, `key`, value, description, tenant_id, create_user, create_date, update_user, update_date)
VALUES('044vwkyeryd44', 'store.iiot_tdengine.3.alarm_content', '节点【#host#】磁盘占用达到预警值【#store#G】', '时序数据存储的告警内容配置，如需修改，请移步到系统运维-存储管理页面修改', 'root', 'rbac_user_superuser', '2024-08-25 19:06:26', '044opkssnaef4', '2024-09-10 11:55:52');

APP 安装指引：

第二章节：IIOT 平台-202409-8 迭代

版本号：SIE IIOT V3.2.241012

更新内容：

APP 名称	显示名称	版本号	备注
iiot_data_query_app	数据查询 APP	v3.2.10.240901	
iiot_db_management	数据库管理	v3.2.10.240901	
iiot_data_archive	iiot 数据归档	v3.2.10.240901	
iiot_data_backup	数据备份 APP	v3.2.10.240901	
iiot_aggregation	数据聚合管理	v3.2.20.240901	从原来 iiot_baseAPP 拆分后新增的

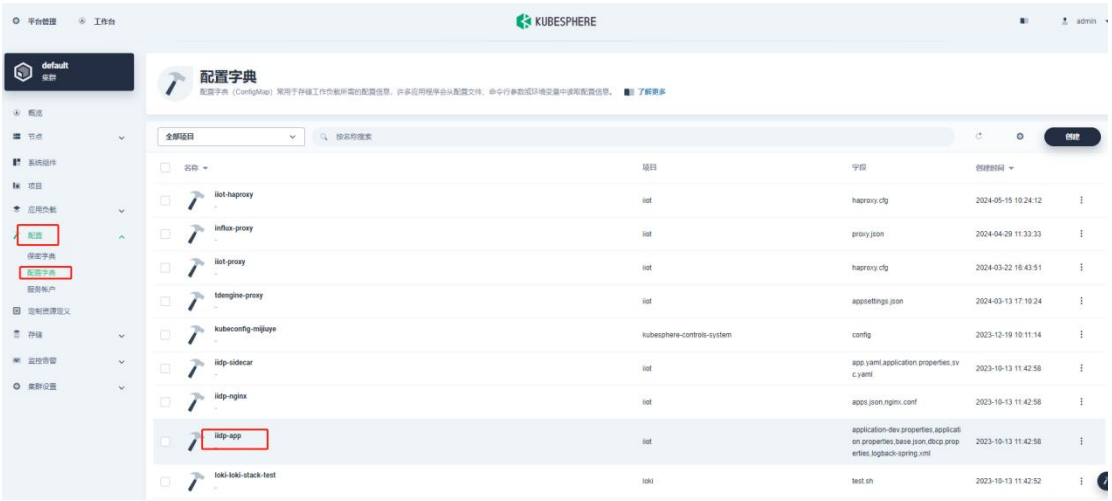
iiot_factory	工厂实体管理	v3.2.20.240901	从原来 iiot_baseAPP 拆分后新增的

1、APP 拆分：把工厂实体和聚合规则单独拆分 APP，依赖建模 APP
2、websocket 支持订阅实体
3、历史工况支持不同类型多测点查询
4、IIOT 部署完成，内置已有默认关系型数据库 MySQL/Oracle、时序数据库 TDengine/InfluxDB，
5、并初始化为数据库管理模块的内置数据。
6、支持添加第三方时序数据库 TDengine
7、平台提供支持 MySQL 和 tdengine 数据表名查询
8、平台提供物实体/工厂实体的历史工况、历史聚合记录、历史报警记录进行数据归档配置
9、平台提供 IIOT 的 TDengine 时序数据库的备份与恢复。

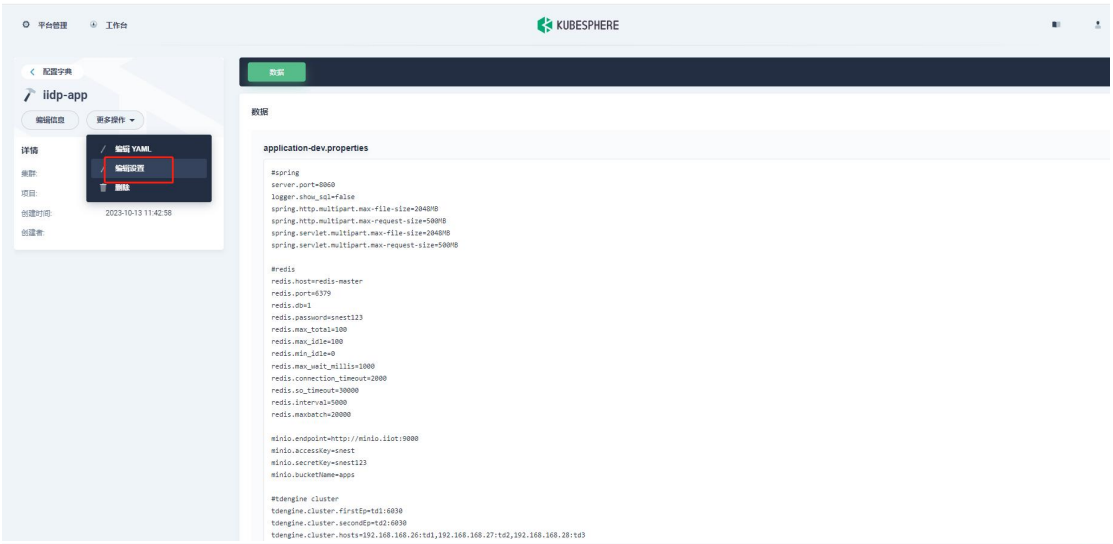
更新注意事项：

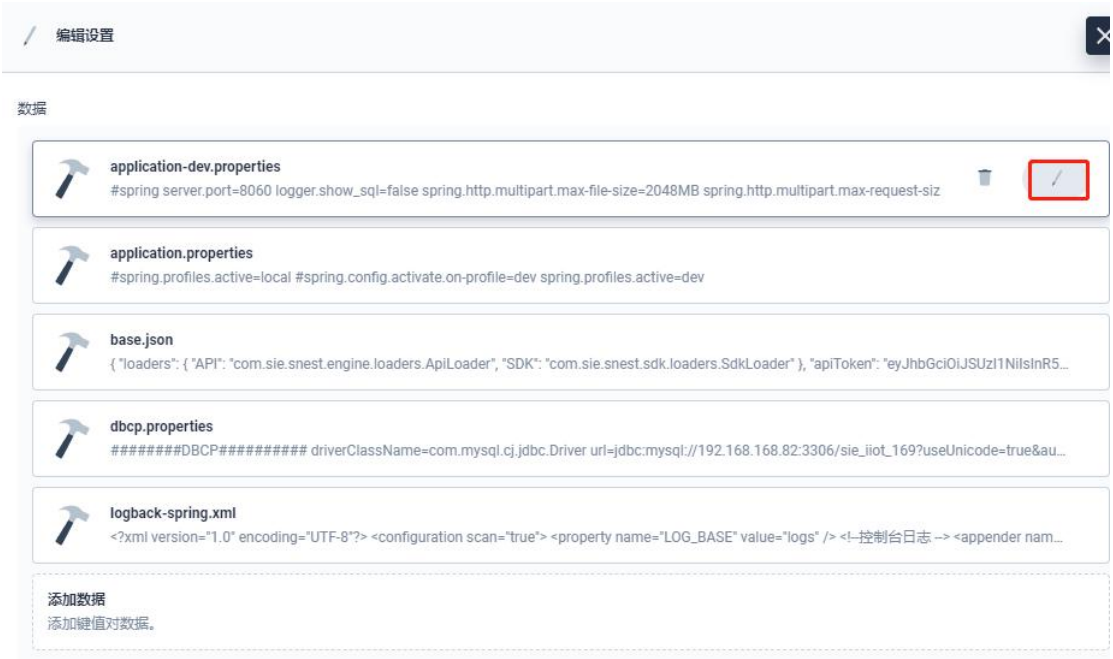
1、检查 IDP 平台 application.dev.properties 配置文件，如果存在 url.whiteList=base.RpcController.download 的配置，如果不存在，需要添加，并重启服务主节点
用于解决边缘端 SMDC 使用平台文件下载失败问题

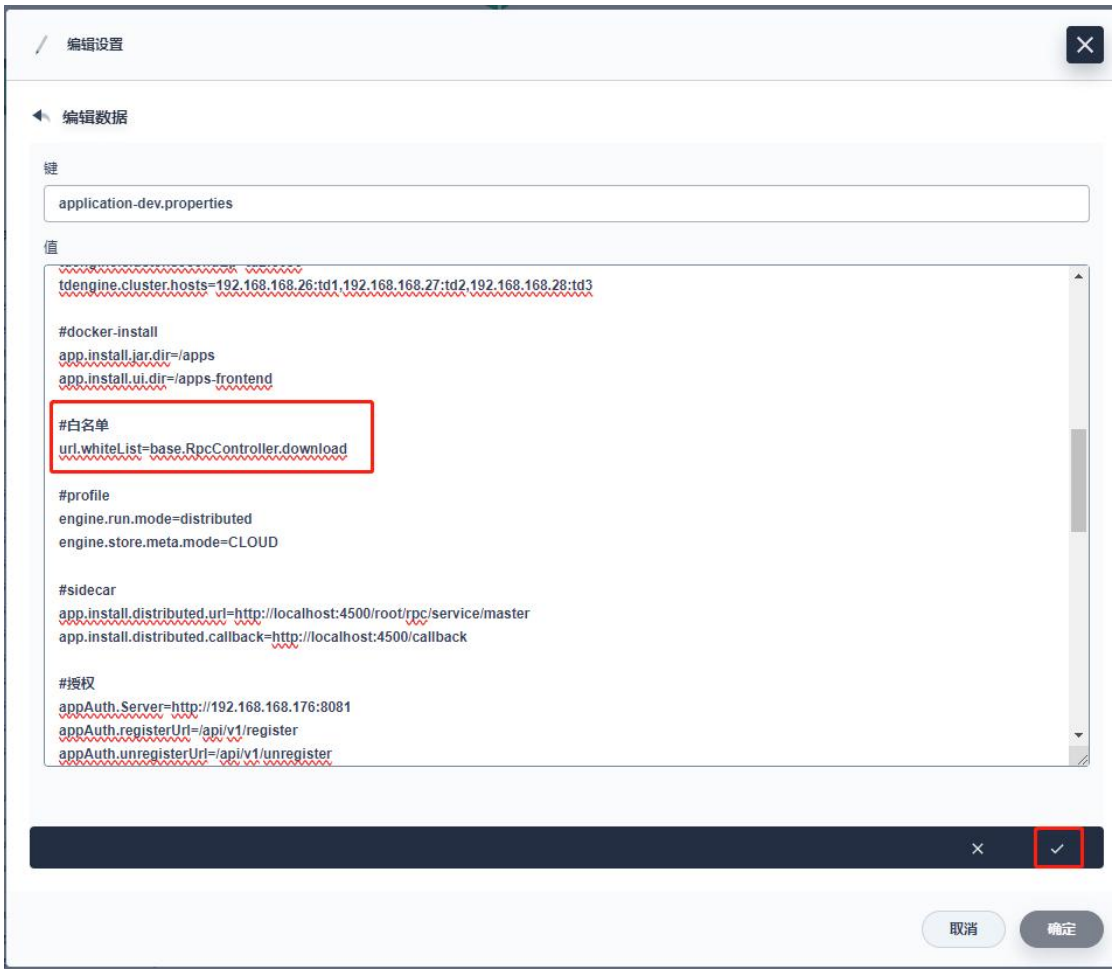
进入 k8s【配置-配置字典】找到所属项目



点击【编辑设置】


平台管理
工作台
KUBESPHERE
配置字典
iidp-app
编辑信息
更多操作
详情
编辑 YAML
编辑设置
删除
2023-10-13 11:42:58
数据
application-dev.properties
#spring
server.port=8060
logger.show_sql=false
spring.http.multipart.max-file-size=2048MB
spring.http.multipart.max-request-size=500MB
spring.servlet.multipart.max-file-size=2048MB
spring.servlet.multipart.max-request-size=500MB
#redis
redis.host=redis-master
redis.port=6379
redis.db=1
redis.password=snest123
redis.max_total=100
redis.max_idle=100
redis.min_idle=0
redis.max_wait_millis=1000
redis.connection_timeout=2000
redis.so_timeout=30000
redis.interval=5000
redis.maxbatch=20000
minio.endpoint=http://minio.iiot:9000
minio.accesskey=snest
minio.secretkey=snest123
minio.bucketName=apps
#tdengine cluster
tdengine.cluster.firstEp=td1:6030
tdengine.cluster.secondEp=td2:6030
tdengine.cluster.hosts=192.168.168.26:td1,192.168.168.27:td2,192.168.168.28:td3


编辑设置
数据
application-dev.properties
#spring server.port=8060 logger.show_sql=false spring.http.multipart.max-file-size=2048MB spring.http.multipart.max-request-siz
application.properties
#spring.profiles.active=local #spring.config.activate.on-profile=dev spring.profiles.active=dev
base.json
{ "loaders": { "API": "com.sie.snest.engine.loaders.ApiLoader", "SDK": "com.sie.snest.sdk.loaders.SdkLoader" }, "apiToken": "eyJhbGciOiJSUzI1NiIsInR5...
dbcp.properties
########DBCP########## driverClassName=com.mysql.cj.jdbc.Driver url=jdbc:mysql://192.168.168.82:3306/sie_iiot_169?useUnicode=true&au...
logback-spring.xml
<?xml version="1.0" encoding="UTF-8"?> <configuration scan="true"> <property name="LOG_BASE" value="logs" /> <!--控制台日志 --> <appender nam...
添加数据
添加键值对数据。



前置条件：

前端版本不能低于：
harbor.devcenter.gushen.sieiot.com/iidp/snest-nginx:v2.7.0-beta.36

修复脚本:

脚本内容如下：
说明：数据库管理新增字段 is_sys_db 和 data_model，需要对原数据进行数据更新
sql 语句如下（注：'DbName' 替换为实际数据库名称）：

```
UPDATE DBName.iiot_db_config SET is_sys_db = '1', data_model = 'iiot_influxdb_data_model', is_time_series_db = '1' WHERE (id = 'iiot_db_config_influxdb');

UPDATE DBName.iiot_db_config SET is_sys_db = '1', data_model = 'iiot_mysql_data_model' WHERE (id = 'iiot_db_config_mysql');
```

```
UPDATE DBName.iiot_db_config SET is_sys_db = '1', data_model = 'iiot_oracle_data_model' WHERE (id = 'iiot_db_config_oracle');

UPDATE DBName.iiot_db_config SET is_sys_db = '1', data_model = 'iiot_tdengine_data_model', is_time_series_db = '1' WHERE (id = 'iiot_db_config_td');
```

APP 安装指引：

归档/备份 APP

第一步：首先确保已安装 iiot_db_management、iiot_store_management、iiot_thing 这 3 个 APP

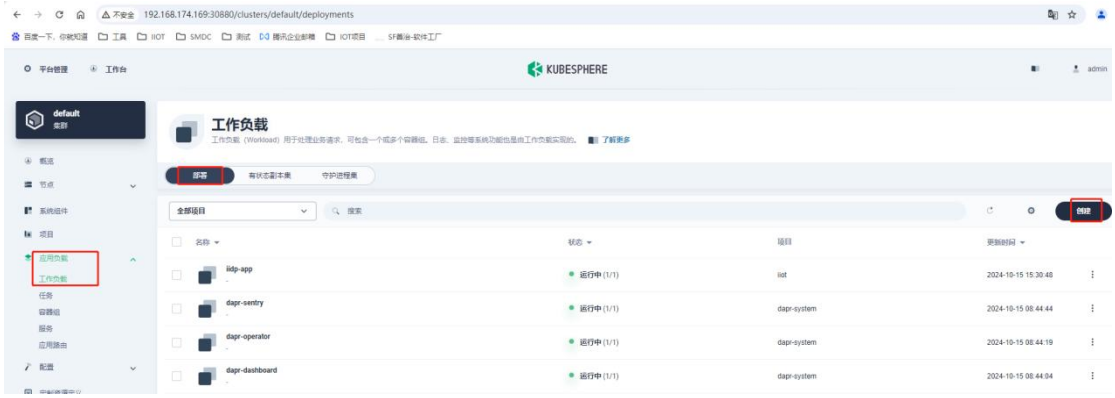
第二步：若安装环境存在网络隔离，需从公司镜像仓库将 powerjob-server 镜像同步到项目镜像仓库（前置条件：确保本地已安装 docker）：

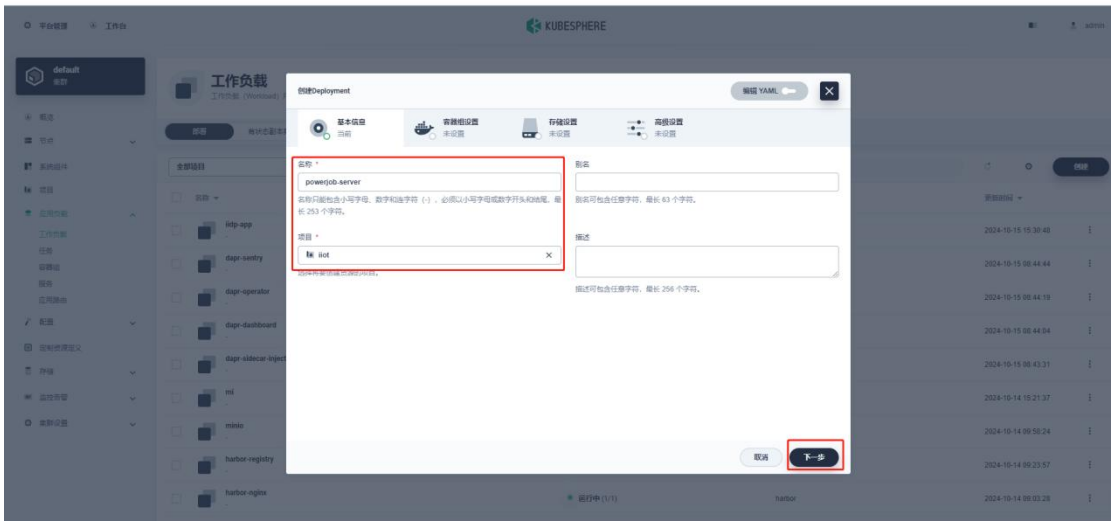
```
# 在终端将 powerjob-server 镜像拉取到本地
docker pull harbor.devcenter.gushen.sieiot.com/iidp-library/powerjob-server:4.3.9
# 在终端将 powerjob-server 镜像导出成文件
docker save -o powerjob-server.tar.gz harbor.devcenter.gushen.sieiot.com/iidp-library/powerjob-server:4.3.9
# 将 powerjob-server.tar.gz 上传至项目部署服务器(上传方式自选，假设目标
```

目录：/root)

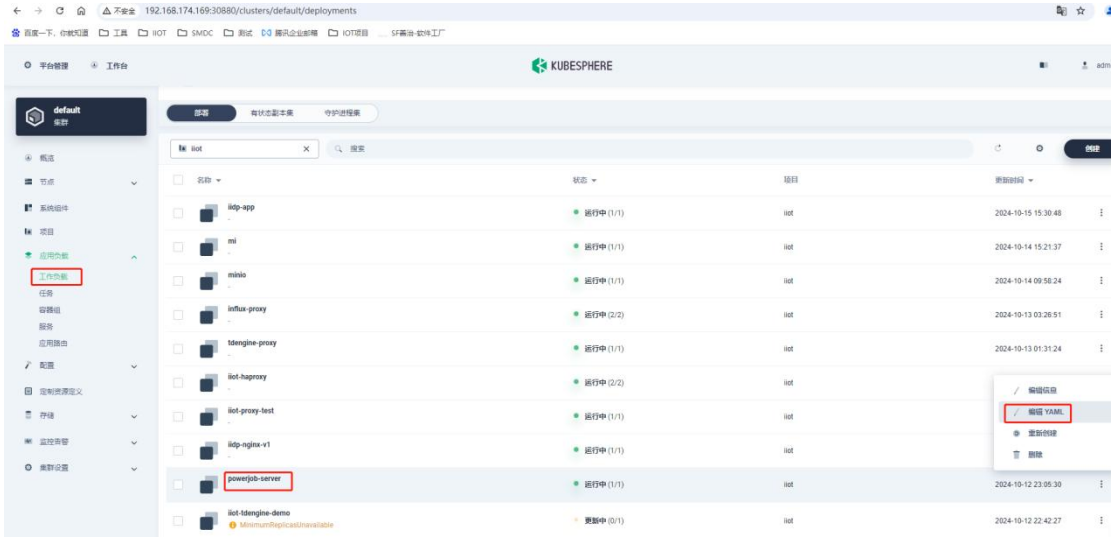
```
# 将 powerjob-server 镜像导入项目部署服务器 docker
docker load < /root/powerjob-server.tar.gz
```

第三步：进入到部署服务器 k8s【应用负载】-【工作负载】选择【部署】点击创建，输入服务名称：powerjob-server，选择所属项目





服务创建成功后，找到 powerjob-server 服务，点击编辑 YAML



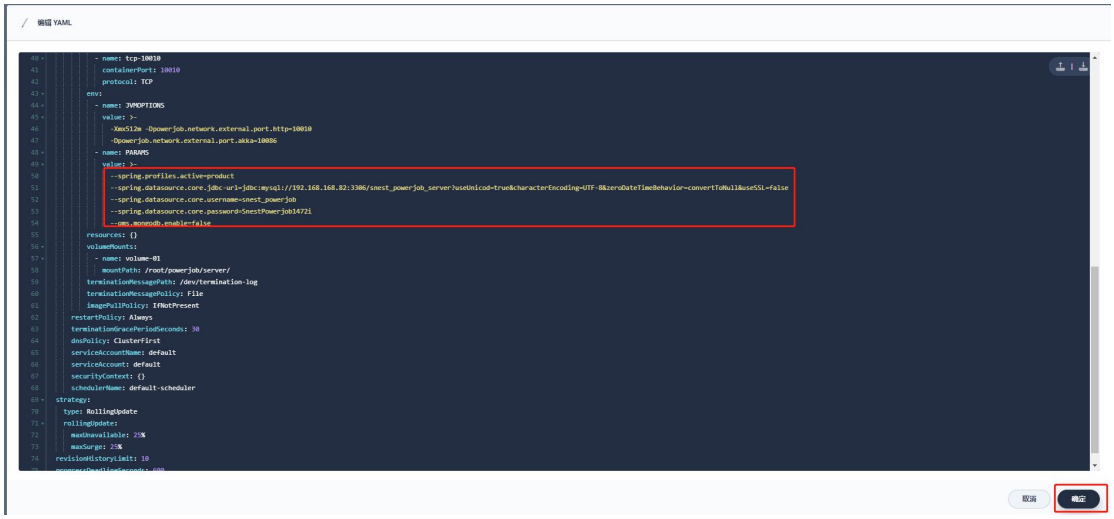
输入配置信息：

请修改红色字体部分

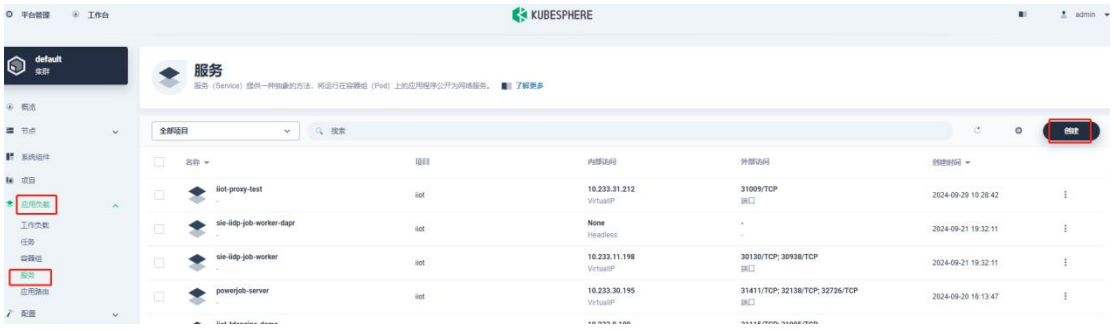
```yaml
kind: Deployment
apiVersion: apps/v1
metadata:
  name: powerjob-server
  namespace: iidp
  labels:
    app: powerjob-server
spec:
  replicas: 1
  selector:
    matchLabels:
      app: powerjob-server
  template:
    metadata:
      labels:
        app: powerjob-server
      annotations:
        logging.kubesphere.io/logsidecar-config: '{}'
    spec:
      volumes:
        - name: volume-01
          persistentVolumeClaim:
            claimName: powerjob-server
      containers:
        - name: container-yxsa3k
          image: >-
            harbor.devcenter.gushen.sieiot.com/iidp-library/powerjob-server:4.3.9
          ports:
            - name: tcp-7700
              containerPort: 7700
              protocol: TCP
            - name: tcp-10086
              containerPort: 10086
              protocol: TCP
            - name: tcp-10010
              containerPort: 10010
              protocol: TCP
          env:
            - name: JVMOPTIONS
```

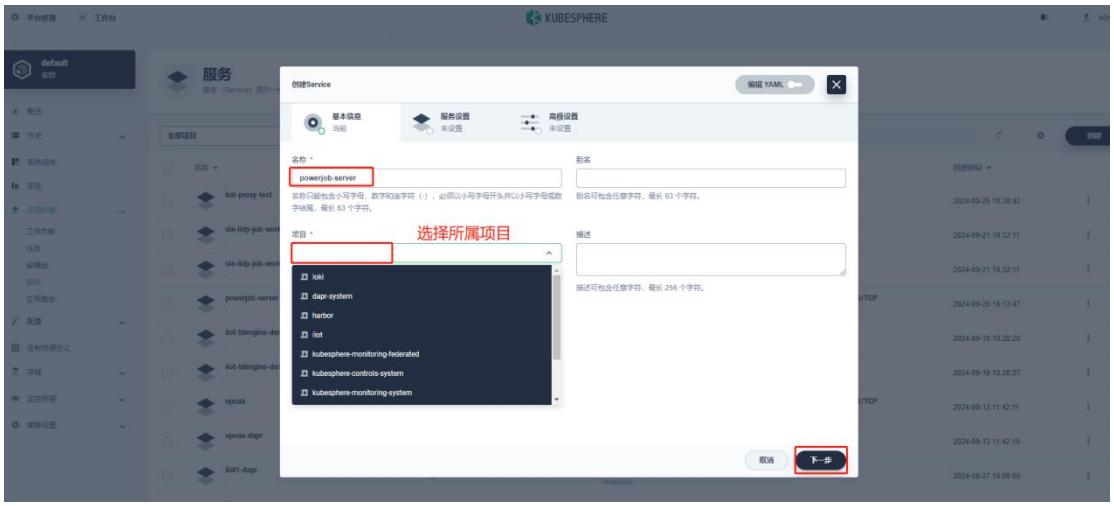
```
              value: >-
                -Xmx512m -Dpowerjob.network.external.port.http=10010
      -Dpowerjob.network.external.port.akka=10086
            - name: PARAMS
              value: >-
                --spring.profiles.active=product
                --spring.datasource.core.jdbc-url=jdbc:mysql://192.168.168.177:3306/snest_dev_2?useUnicode=true&characterEncoding=UTF-8&zeroDateTimeBehavior=convertToNull&useSSL=false
                --spring.datasource.core.username=snest_dev_2
                --spring.datasource.core.password=*******
                --oms.mongodb.enable=false
          resources: {}
          volumeMounts:
            - name: volume-01
              mountPath: /root/powerjob/server/
          terminationMessagePath: /dev/termination-log
          terminationMessagePolicy: File
          imagePullPolicy: IfNotPresent
      restartPolicy: Always
      terminationGracePeriodSeconds: 30
      dnsPolicy: ClusterFirst
      serviceAccountName: default
      serviceAccount: default
      securityContext: {}
      schedulerName: default-scheduler
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 25%
      maxSurge: 25%
  revisionHistoryLimit: 10
  progressDeadlineSeconds: 600
```

将上面信息复制拷贝进入 YAML，然后根据实际情况修改批注内容，点击【确定】

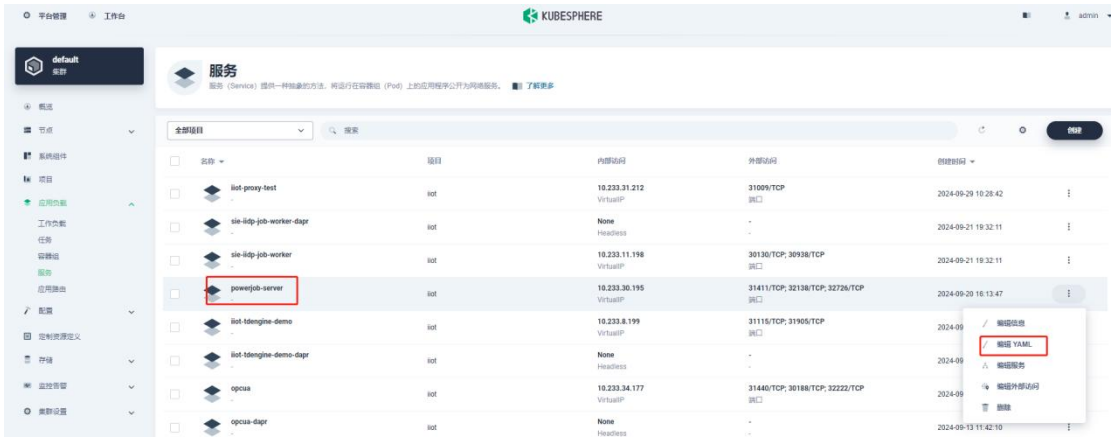


第四步：进入到部署服务器 k8s【应用负载】-【服务】点击创建，输入服务名称：powerjob-server，选择所属项目





1、服务创建成功后，找到 powerjob-server 服务，点击编辑 YAML



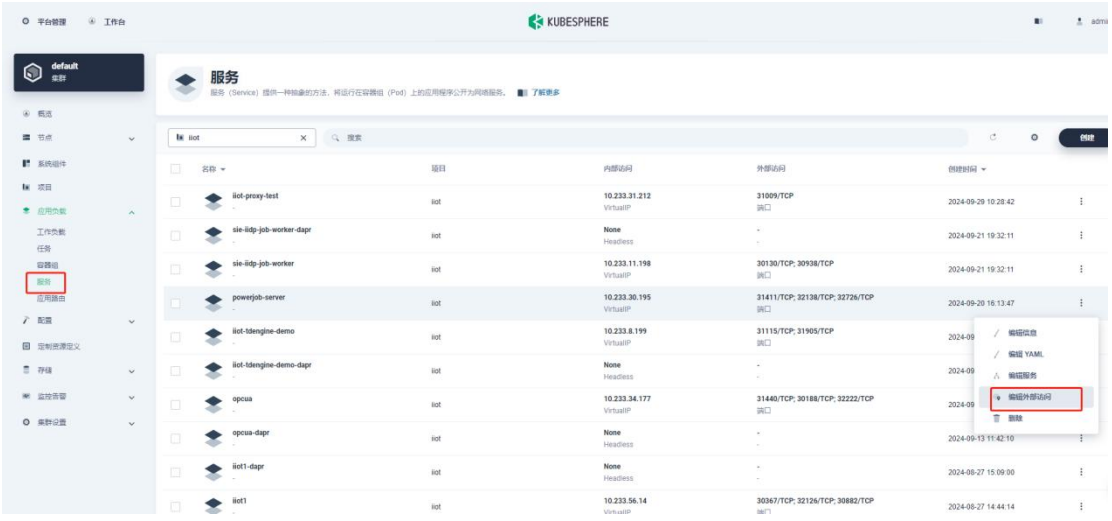
输入配置信息：
请修改红色字体部分

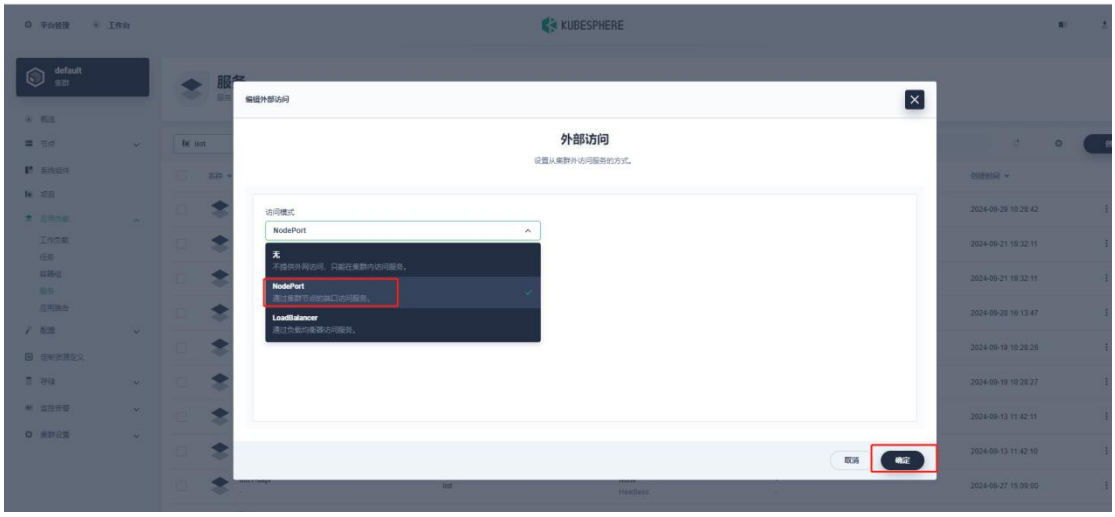
```yaml
kind: Service
apiVersion: v1
metadata:
  name: powerjob-server
  namespace: iidp
  labels:
    app: powerjob-server
  annotations:
spec:
  ports:
    - name: tcp-7700
      protocol: TCP
      port: 7700
      targetPort: 7700
    - name: tcp-10086
      protocol: TCP
```

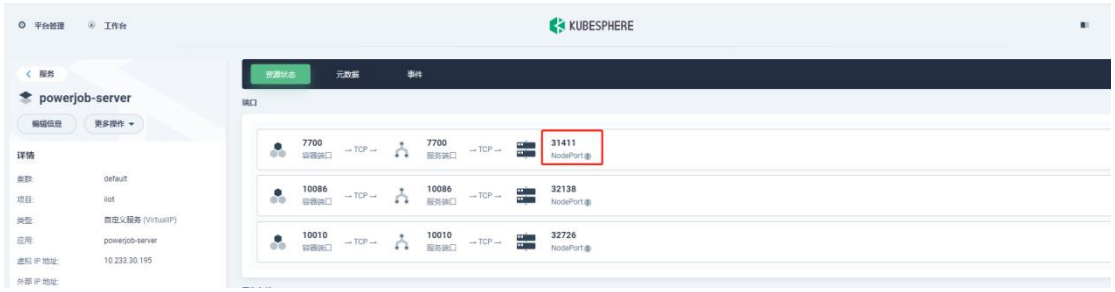
```
      port: 10086
      targetPort: 10086
    - name: tcp-10010
      protocol: TCP
      port: 10010
      targetPort: 10010
  selector:
    app: powerjob-server
  type: NodePort
  sessionAffinity: None
  externalTrafficPolicy: Cluster
  ipFamilies:
    - IPv4
  ipFamilyPolicy: SingleStack
```

将上面信息复制拷贝进入 YAML，然后根据实际情况修改批注内容，点击【确定】

第五步：在 powerjob-server 服务，点击【编辑外部访问】允许外部访问

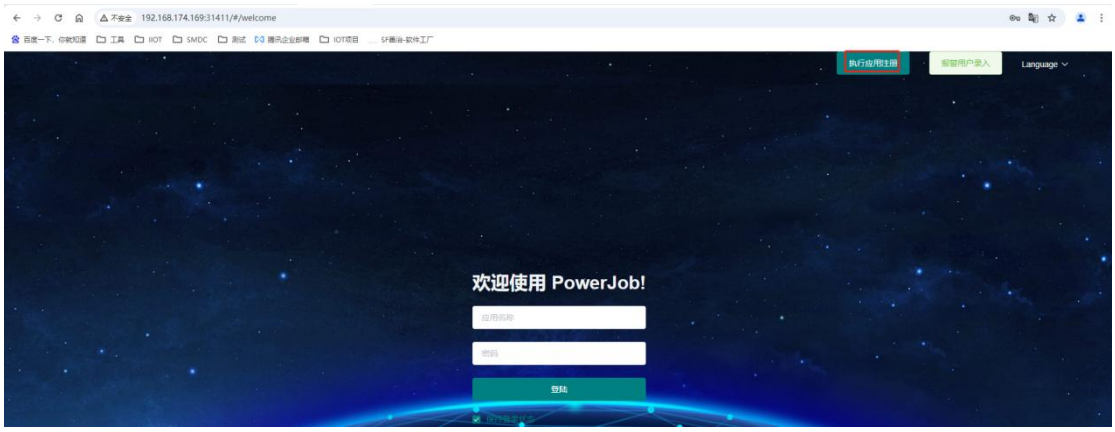






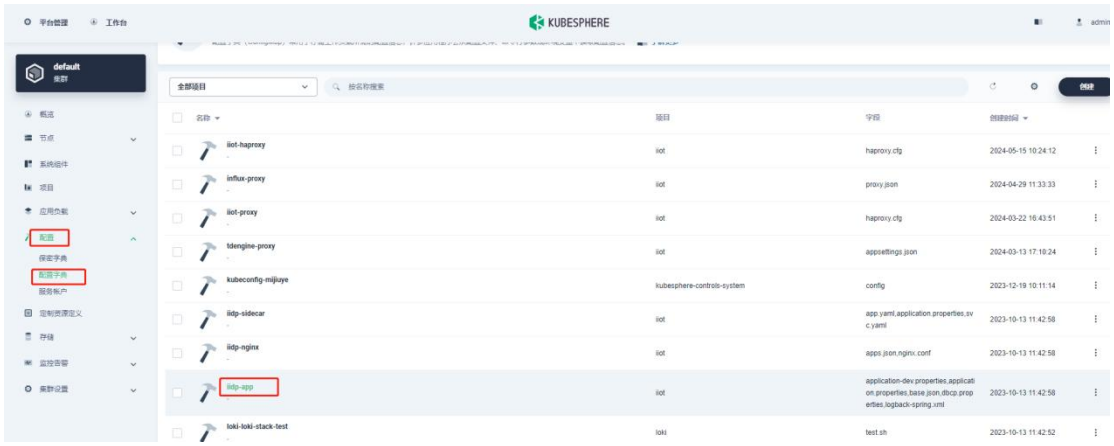
第六步：注册 data_archive_app、data_backup_app 应用

部署完成后，进入 powerjob-server 管理页面（例如：http://192.168.168.81:31411），点击【执行应用注册】



注册【归档应用】和【备份应用】账号和密码，归档应用默认：data_archive_app/data_archive_app，备份应用：data_backup_app/data_backup_app。

第七步：进入 k8s【配置-配置字典】找到所属项目

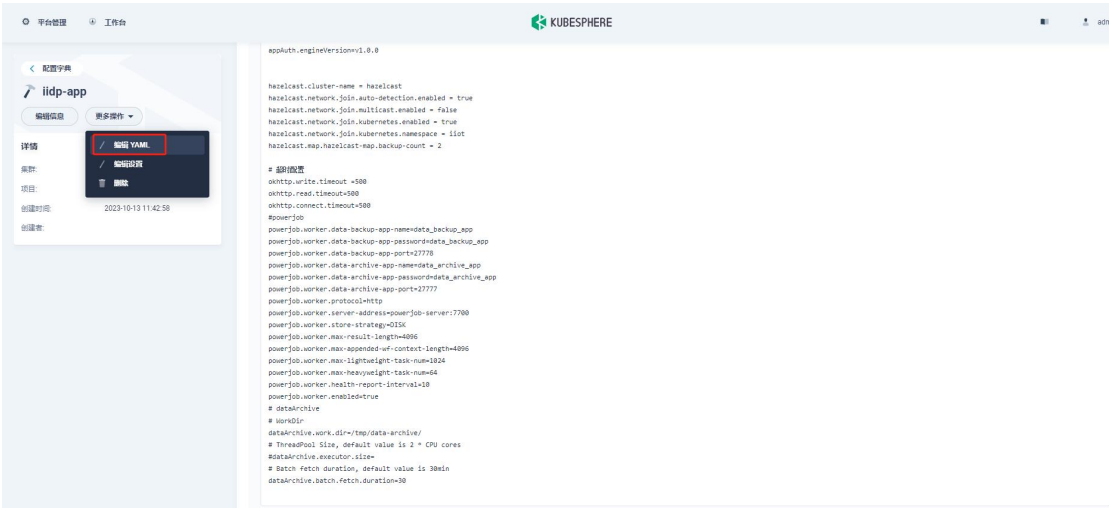


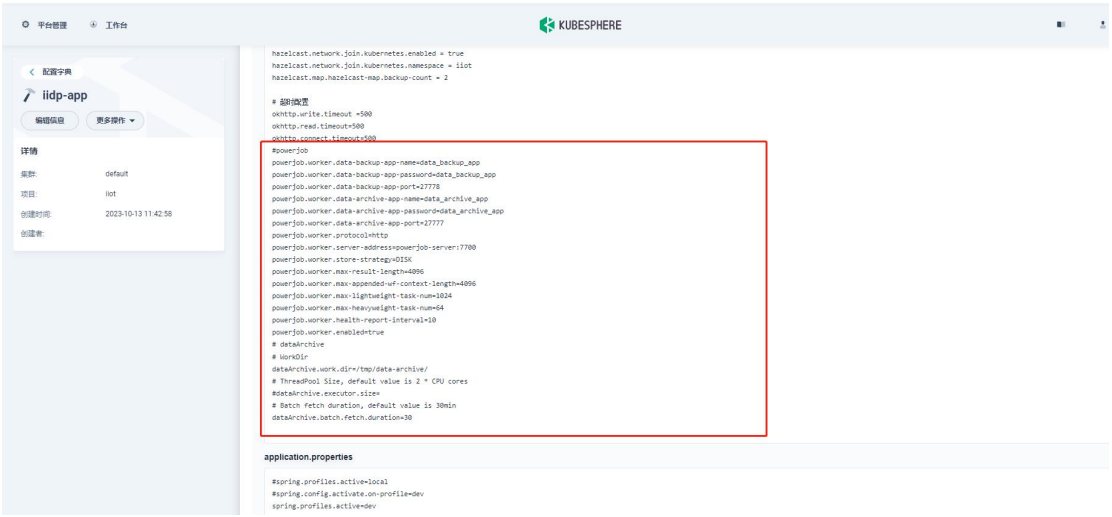
修改 application-dev.properties 配置文件增加 powerjob、dataArchive backup 部分配置信息，点击【编辑 YAML】

配置信息：

```
#powerjob
powerjob.worker.data-backup-app-name=data_backup_app  #备份账号需要修改
powerjob.worker.data-backup-app-password=data_backup_app #备份密码需要修改
powerjob.worker.data-backup-app-port=27778
powerjob.worker.data-archive-app-name=data_archive_app  #归档账号需要修改
powerjob.worker.data-archive-app-password=data_archive_app 归档密码需要修改
powerjob.worker.data-archive-app-port=27777
powerjob.worker.protocol=http
powerjob.worker.server-address=powerjob-server:7700  #单机部署需要根据实际情况填写 IP 地址，分布式部署直接写容器的名称
powerjob.worker.store-strategy=DISK
powerjob.worker.max-result-length=4096
powerjob.worker.max-appended-wf-context-length=4096
powerjob.worker.max-lightweight-task-num=1024
```

```
powerjob.worker.max-heavyweight-task-num=64
powerjob.worker.health-report-interval=10
powerjob.worker.enabled=true
# dataArchive
# WorkDir
dataArchive.work.dir=/tmp/data-archive/
# ThreadPool Size, default value is 2 * CPU cores
#dataArchive.executor.size=
# Batch fetch duration, default value is 30min
dataArchive.batch.fetch.duration=30
```

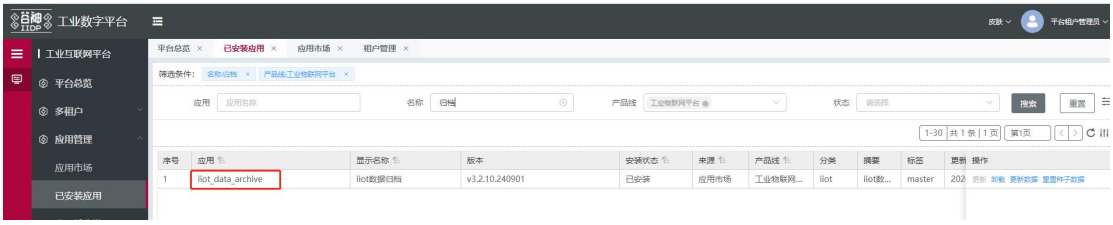




第八步：进入【应用市场】上传 redis、iiot_data_archive 和 iiot_data_backup APP

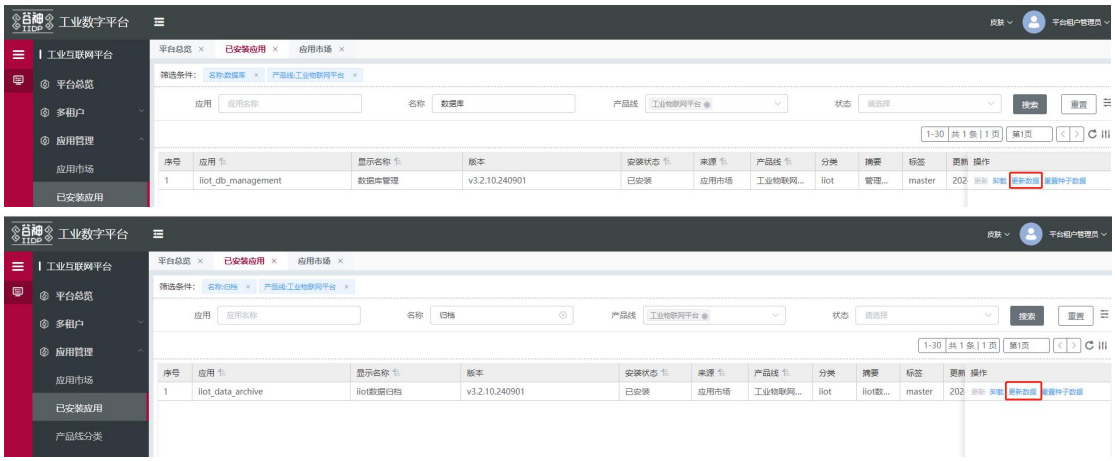
Redis APP 上传成功后，列表找到 APP 进行上架，然后进入到【已安装应用】更新 APP （如果未安装过 app，则需要安装）

iiot_data_archive APP 上传成功后，列表找到 APP 进行上架，然后安装 APP，待服务重启成功后，在【已安装应用】查看 iiot_data_archiveAPP 是否存在

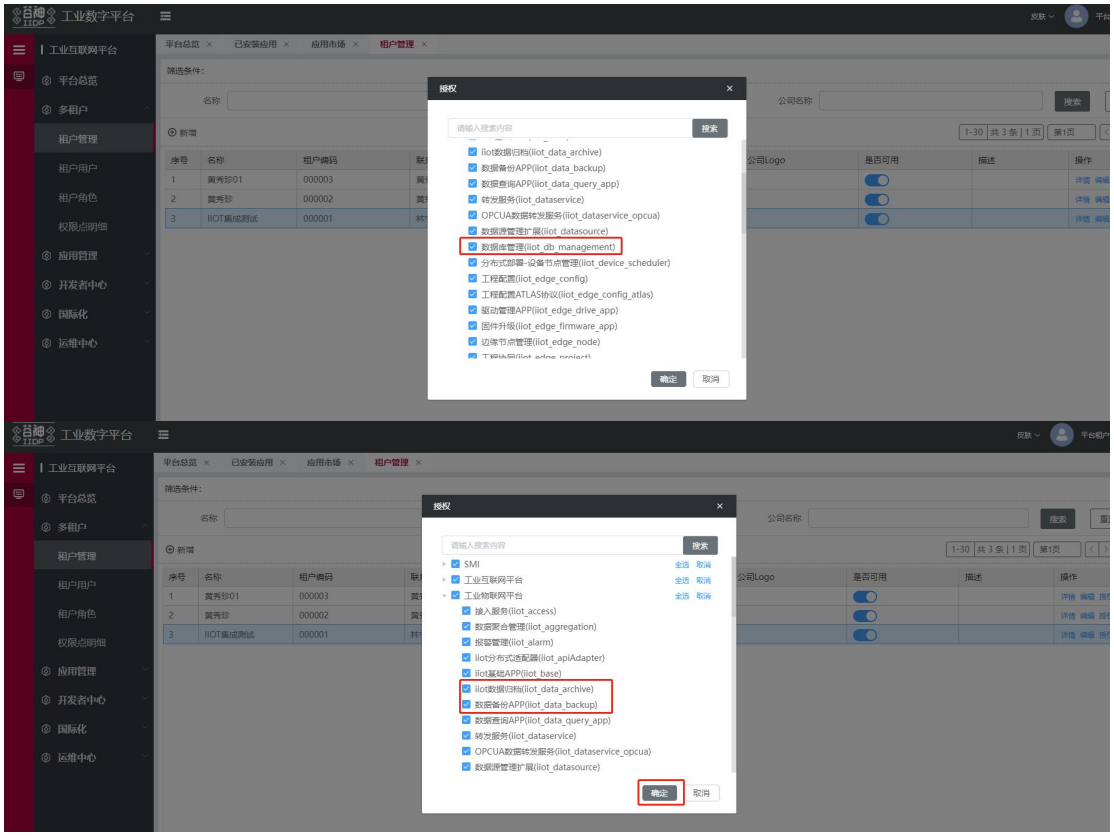


iiot_data_archive APP 上传成功后，列表找到 APP 进行上架，然后安装 APP，待服务重启成功后，在【已安装应用】查看 iiot_data_backup 是否存在

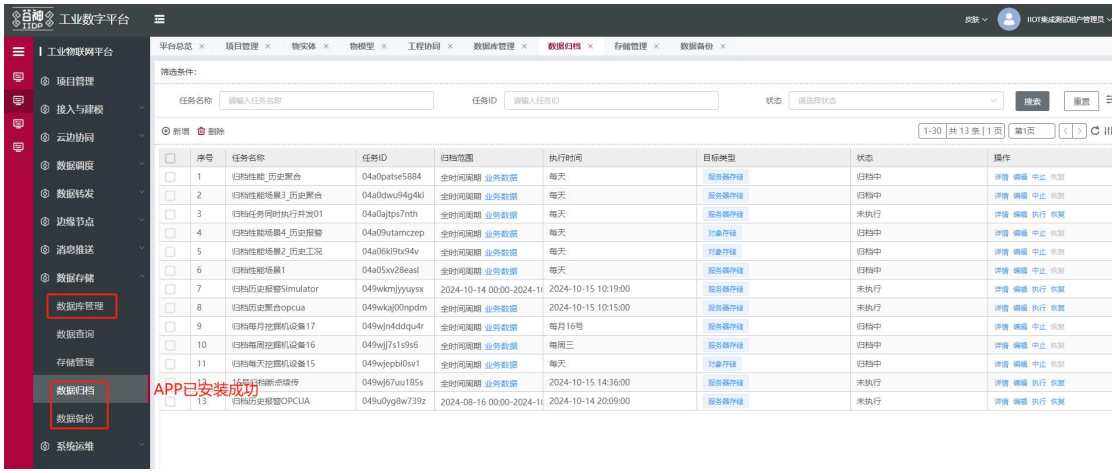
第九步：服务重启成功，进入【已安装应用】列表找到 APP：iiot_db_management 、iiot_data_archive、iiot_data_backup ，点击【更新数据】



第十步：进入【多租户-租户管理】找到对应租户账号，点击【授权】勾选菜单，点击【保存】



保存成功后，进入平台查看



第三章节：IIOT 平台-202408-7 迭代

版本号：SIE IIOT V3.2.0909

更新内容：

APP 名称	显示名称	版本号	备注
iiot_dataservice_opcua	OPCUA 数据转发服务	v3.2.12.240901	
iiot_apiAdapter	iiot 分布式适配器	v3.2.20.240801	
iiot_edge_drive_app	驱动管理 APP	V3.2.10.240801	

1、IIOT 平台的工况数据/报警事件支持 OPC UA 协议转发
2、取消 apiAdapter 对 iiot 应用的影响，修改为 iiot 应用
3、添加驱动管理完善驱动更新发布流程，实现云边驱动协同
4、tdengine-proxy 版本升级、字符串处理、api 压缩
5、存储 http 请求使用压缩

更新注意事项：

前置条件：

修复脚本：

APP 安装指引：

loki 关键日志降频

前置条件：

无

修复脚本：

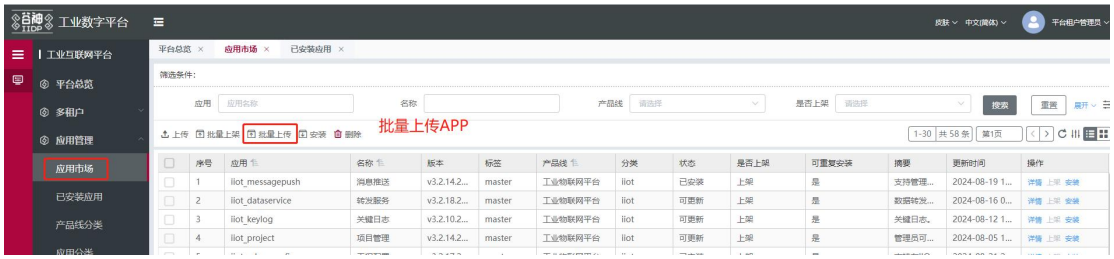
无

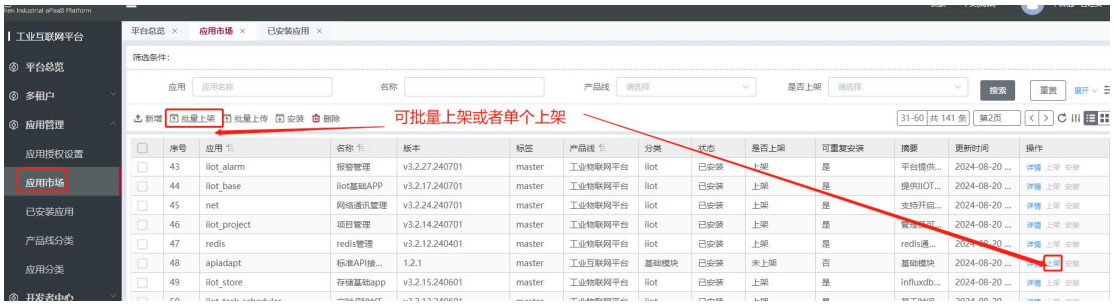
操作步骤：

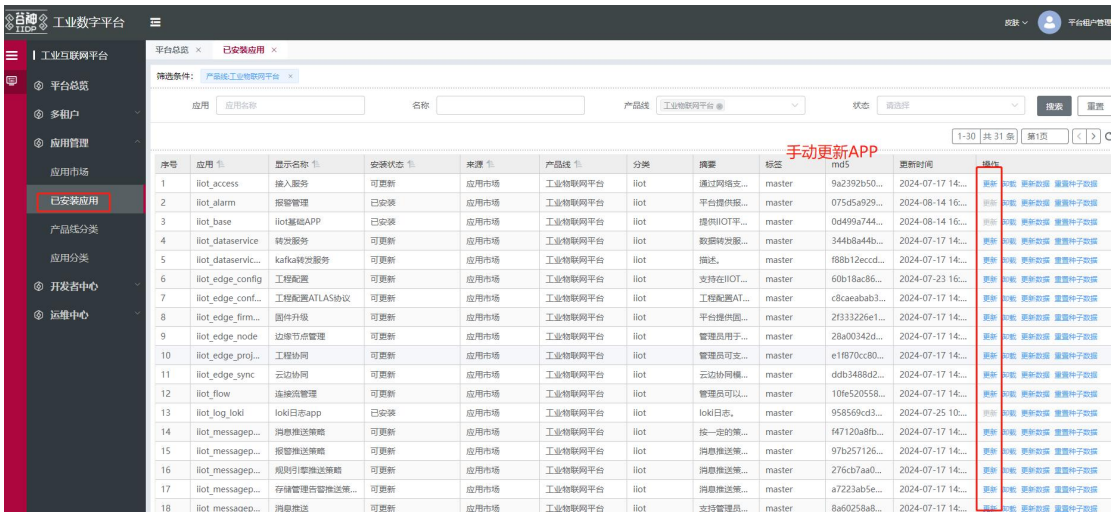
第一步：安装之前，在已安装应用列表把 iiot_log_loki 卸载

第二步：安装之前，在已安装应用列表把 iiot_mysqlstore 卸载(mysqlstore 中间不带下划线)，需要先卸载；如果不存在，则不需要卸载

第三步：在应用市场，上传四个 APP 分别是：**iiot_base、iiot_thing、iiot_access、iiot_keylog**，然后进行上架，（如未安装过 APP，则需安装全部 APP ）如已安装过 APP，则在【已安装应用】更新 APP

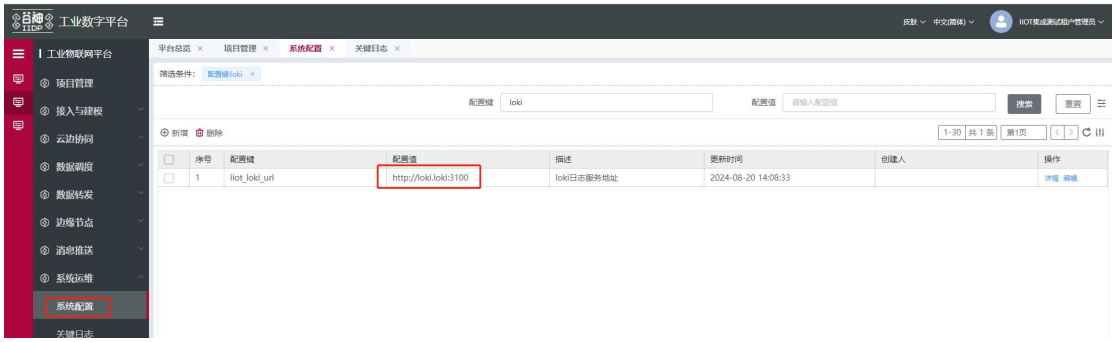






更新检查：

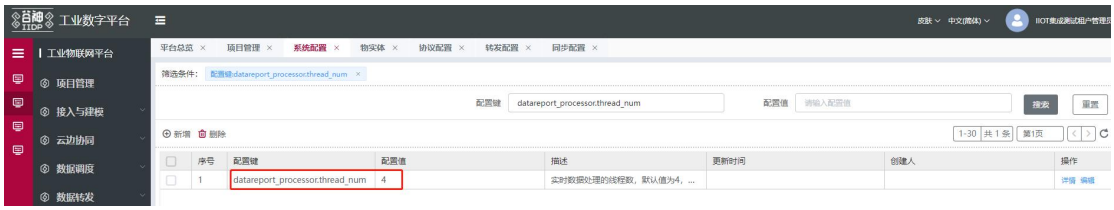
第一步：进入 IOT 的系统配置菜单页面，检测 iiot_loki_url 地址是否正常



第二步：需要修改配置值：http://ip（服务器 ip 地址）+端口（loki 开放端口）；如果是单机部署，则是默认：3100 端口；分布式，则去 k8s 查看
（具体根据实际情况进行调整）

第三步：需要在菜单：系统运维》系统配置 中查询：process.thread_num 的值是否是 1；

datareport_processor.thread_num 的值是否是 4；



第四步：如果不是(项目如果自行调整过，则不用重置），需要在平台的【开发者中心－种子数据管理】搜索“datareport_processor_thread_num”和“process_thread_num“进行重置

IIOT 平台支持多语言

前置条件：

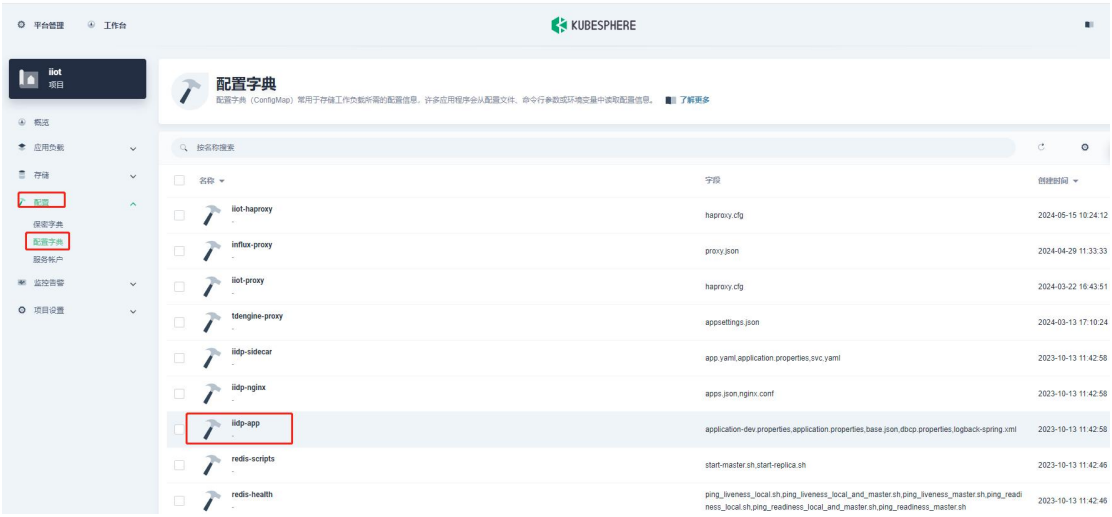
安装 iiot 多语言需要注意：IIDP 平台后端引擎版本要大于 v2.4.1-HOTFIX.009

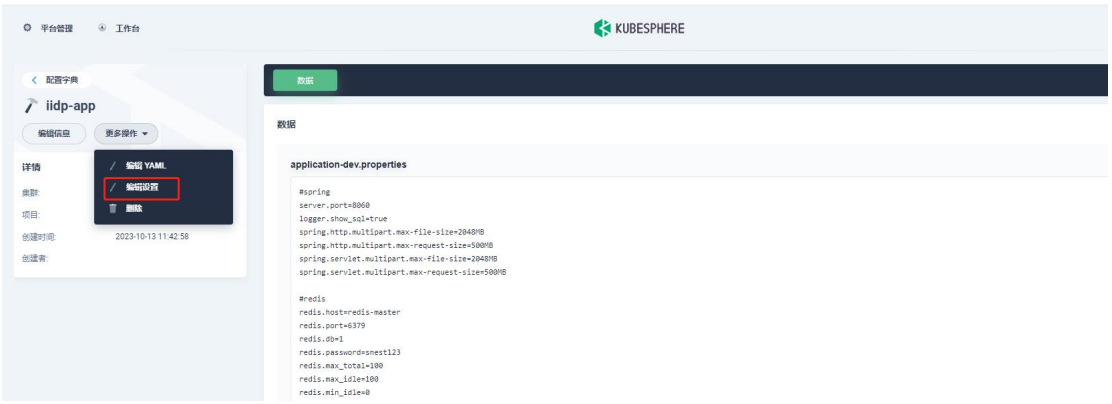
修复脚本

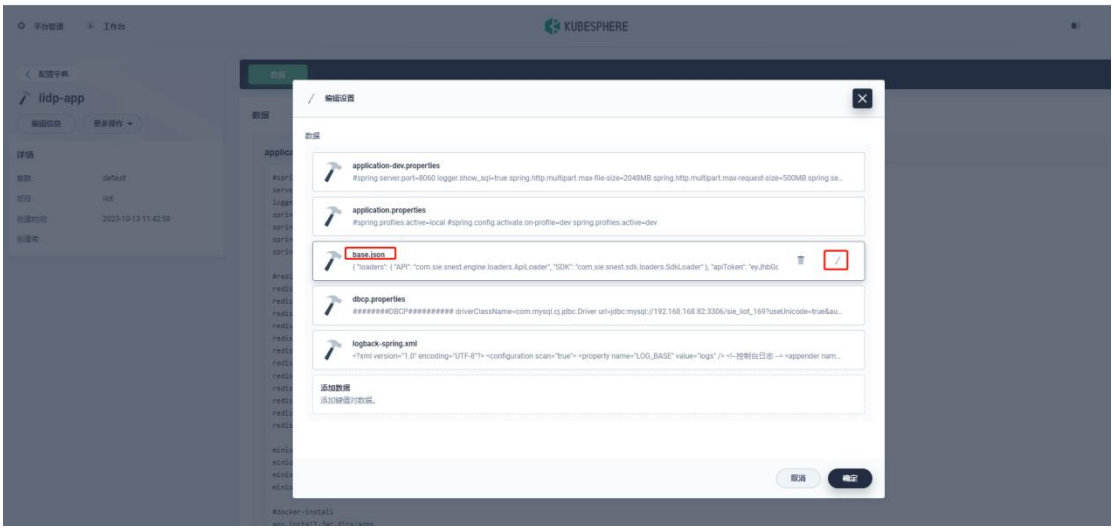
无

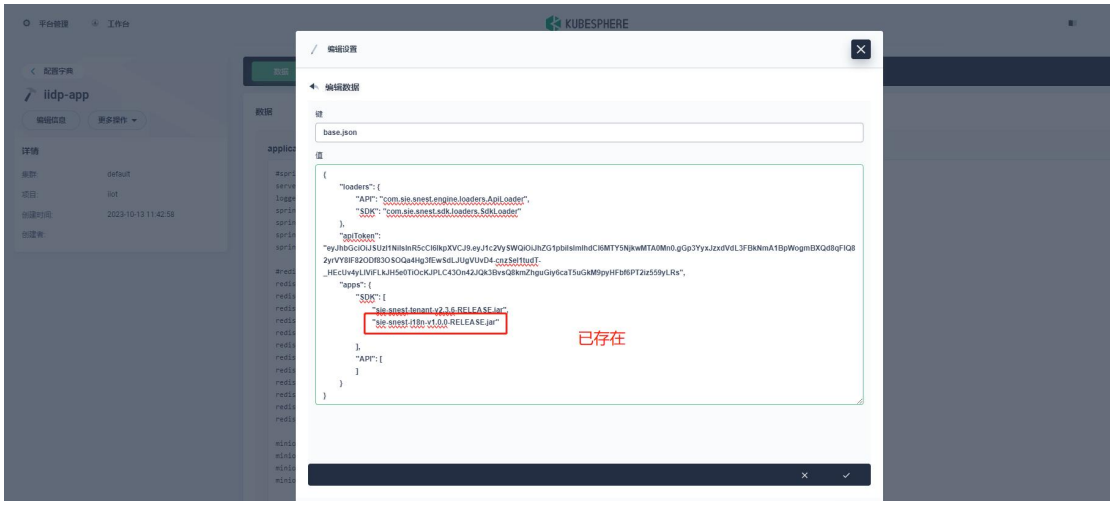
操作步骤：

第一步：首先，进入 k8s,路径：配置-配置字典-iidp-app（找到对应项目配置），点击【更多操作】-编辑设置- “base.json “点击【编辑】查看是否存在“**sie-snest-i18n-v1.0.0-RELEASE.jar**”

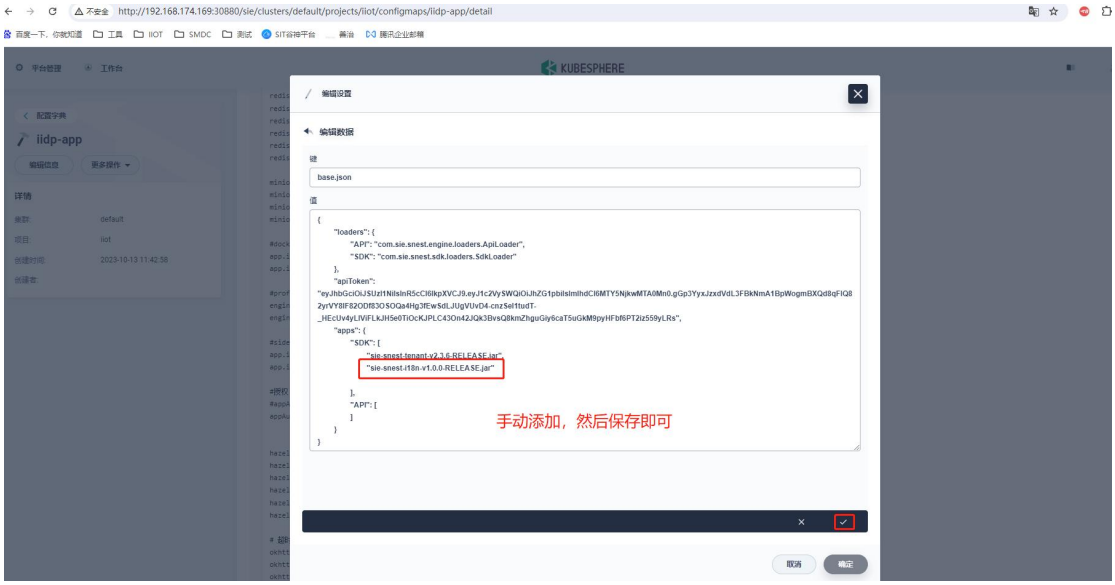

平台管理
工作台
KUBESPHERE
iiot
项目
概览
应用负载
存储
配置
保密字典
配置字典
服务帐户
监控告警
项目设置
配置字典
配置字典（ConfigMap）常用于存储工作负载所需的配置信息。许多应用程序会从配置文件、命令行参数或环境变量中读取配置信息。
了解更多
按名称搜索
名称
字段
创建时间
iiot-haproxy
haproxy.cfg
2024-05-15 10:24:12
influx-proxy
proxy.json
2024-04-29 11:33:33
iiot-proxy
haproxy.cfg
2024-03-22 16:43:51
tdengine-proxy
appsettings.json
2024-03-13 17:10:24
iidp-sidecar
app.yaml,application.properties,svc.yaml
2023-10-13 11:42:58
iidp-nginx
apps.json,nginx.conf
2023-10-13 11:42:58
iidp-app
application-dev.properties,application.properties,base.json,dbcp.properties,logback-spring.xml
2023-10-13 11:42:58
redis-scripts
start-master.sh,start-replica.sh
2023-10-13 11:42:46
redis-health
ping_liveness_local.sh,ping_liveness_local_and_master.sh,ping_liveness_master.sh,ping_readiness_local.sh,ping_readiness_local_and_master.sh,ping_readiness_master.sh
2023-10-13 11:42:46


平台管理
工作台
KUBESPHERE
配置字典
iidp-app
编辑信息
更多操作
编辑 YAML
编辑设置
删除
详情
集群:
项目:
创建时间:
2023-10-13 11:42:58
创建者:
数据
数据
application-dev.properties
#spring
server.port=8060
logger.show_sql=true
spring.http.multipart.max-file-size=2048MB
spring.http.multipart.max-request-size=500MB
spring.servlet.multipart.max-file-size=2048MB
spring.servlet.multipart.max-request-size=500MB
#redis
redis.host=redis-master
redis.port=6379
redis.db=1
redis.password=snest123
redis.max_total=100
redis.max_idle=100
redis.min_idle=0


编辑设置
数据
application-dev.properties
application.properties
base.json
dbcp.properties
logback-spring.xml
添加数据
添加键值对数据。
取消
确定



第二步：如果不存在此配置，则需手动编辑添加“**sie-snest-i18n-v1.0.0-RELEASE.jar**”点击【保存】即可



第三步：在应用市场，上传 **iiot_base** app，然后进行上架，（如未安装过 APP，则需安装 APP ）如已安装过 APP，则在【已安装应用】更新 APP

高亮显示时：点击上架

高亮显示时：点击更新

更新检查：

无

opcua server 转发 app

3.1 分布式+高可用安装指引

前提条件：

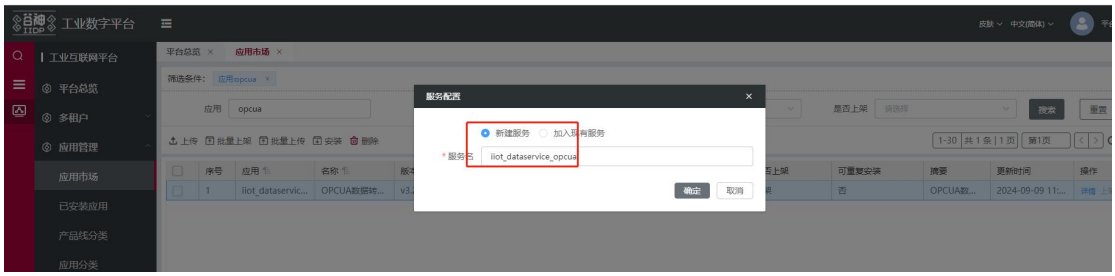
安装 opcUA 服务前，需要先上传最新版本 net　app，版本号　v3.2.27.240804

修复脚本：

无

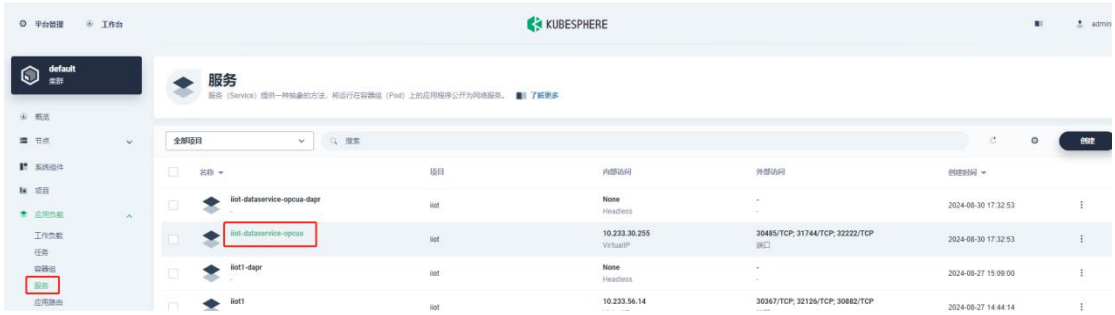
操作步骤：

第一步：上传 APP：iiot_dataservice_opcua，安装方式：点击安装按钮后，选择新建服务：出现默认服务名名称：iiot-dataservice-opcua，后点击确定，等待服务安装完成。

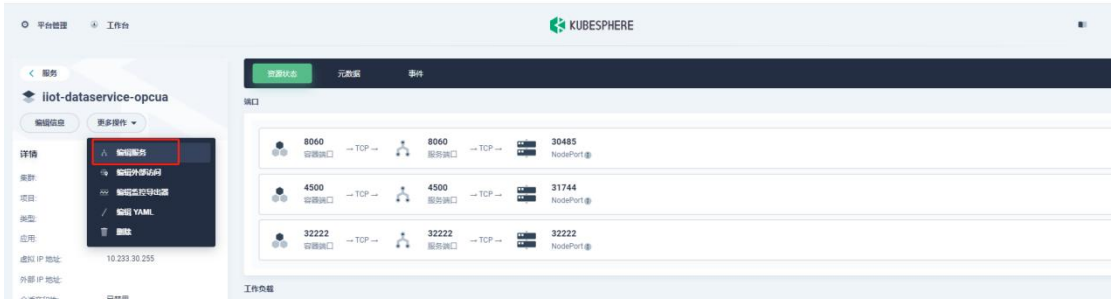


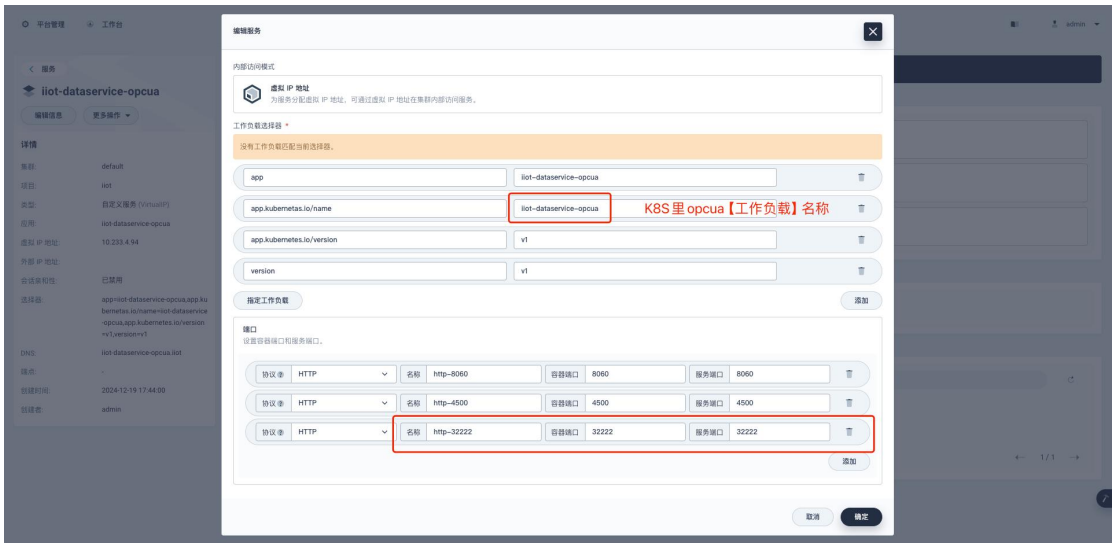
k8s 服务会出现：iiot-dataservice-opcua

第三步：OPCUA 服务端口开放，提前规划 OPCUA 服务的端口号（30000-32222），并在 kubesphere 中的服务列表找到 iiot-dataservice-opcua 服务（如果没有则需创建），并将端口号访问模式改为外部访问（新增 OPCUA 协议配置时，注意填写的端口为 kubesphere 服务配置的端口）

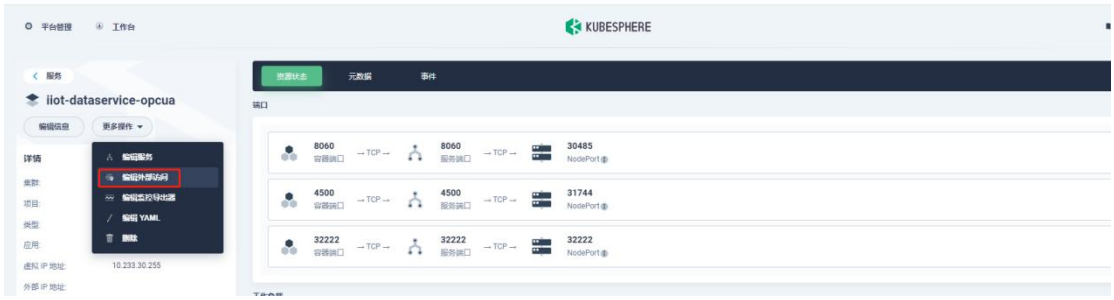


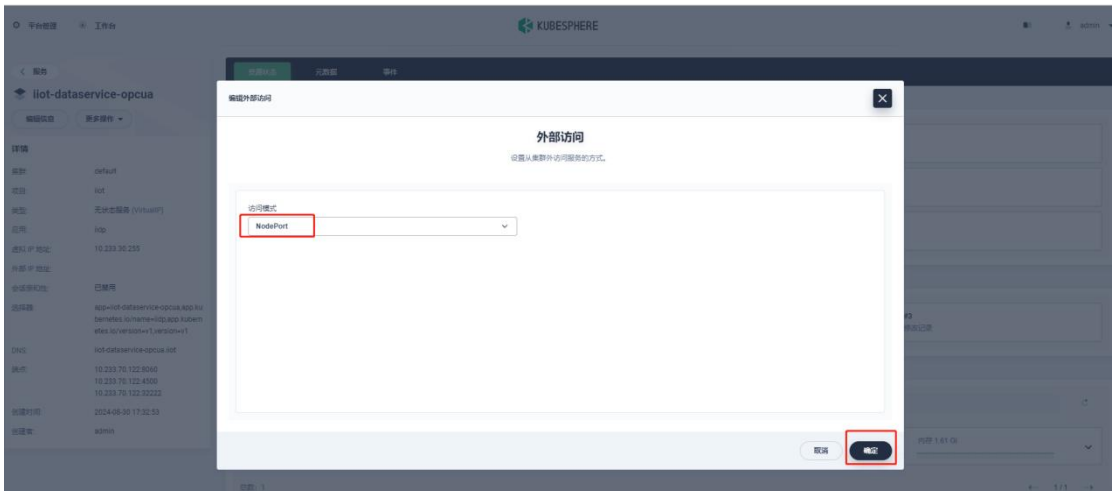
创建服务端口号：

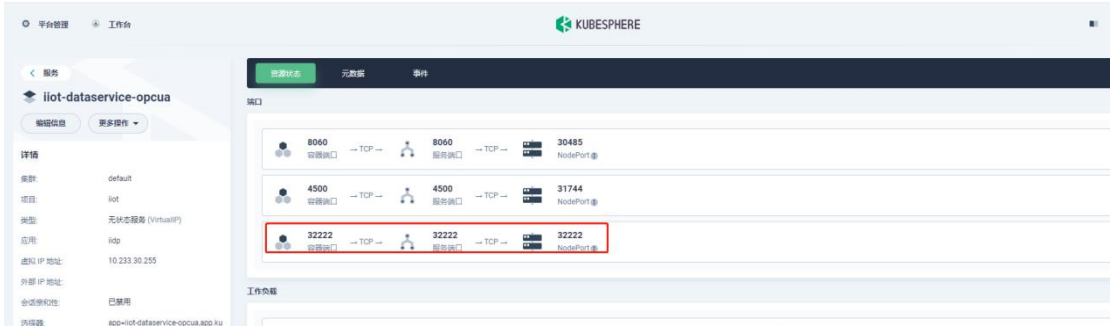



K8S里opcua【工作负载】名称

开放外部访问：


KUBESPHERE
iiot-dataservice-opcua
编辑外部访问
8060
4500
32222
30485
31744
32222


KUBESPHERE
外部访问
NodePort


KUBESPHERE
iiot-dataservice-opcua
8060
4500
32222
30485
31744
32222

更新检查：

无

3.2 单机环境安装指引

前置条件：

安装 opcUA 服务前，需要先上传最新版本 net app，版本号 v3.2.27.240804

修复脚本：

操作步骤

第一步：在【应用市场】上传 APP：iiot_dataservice_opcua，安装方式：点击安装按钮后，等待服务安装完成。

第二步： 待服务重启

更新检查：

无

ApiAdapter APP

前置条件：

修复脚本：

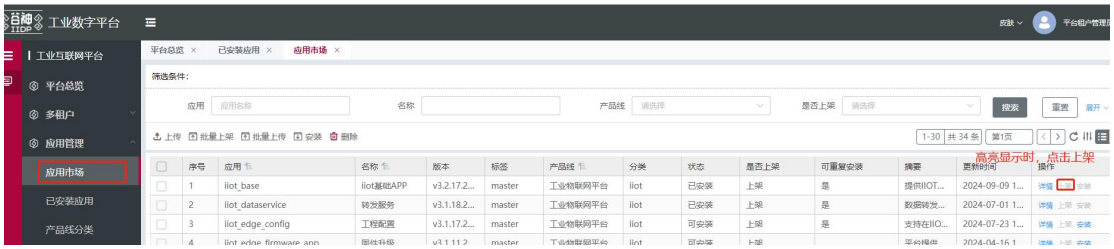
无

操作步骤：

第一步：安装之前，在已安装应用列表把 apiadapt 卸载

第三步：然后再分别上传 APP：iiot_apiAdapter 、iiot_thing、iiot_tdstore、iiot_store；上传成功后，上架 APP，再进入【已安装应用】列表手动更新 app

上传成功后，上架 APP



上架成功后，手动更新 APP

iiot_apiAdapter 未安装过，则需要手动安装

更新检查：

1、iiot 服务重启成功后，验证功能是否正常

influxdb 数据存储异常操作

前置条件：

：

修复脚本：

无

操作步骤：

如果需要使用聚合规则，则需要按照下面操作步骤添加配置文件
第一步：如需用 influxdb 的条件滤波或者聚合功能，需要开启 influxdb 的 flux 功能， 参考《influxdb 部署文档.docx》 在配置文件 influxdb.conf， 如无配置文件 influxdb.conf 需要增加配置文件，增加配置 flux-enabled=true
参考文档进入官网查看：http://smdc.chinasie.com:8097/SMDC/

SMDC

SMDC下载

序号	版本		
1	SIIoT 边缘 V11.0	11.0 SIIoT 边缘	
2	SMDC V10.0	10.0正式版	
3	Api 网关 V1.0	1.0正式版	

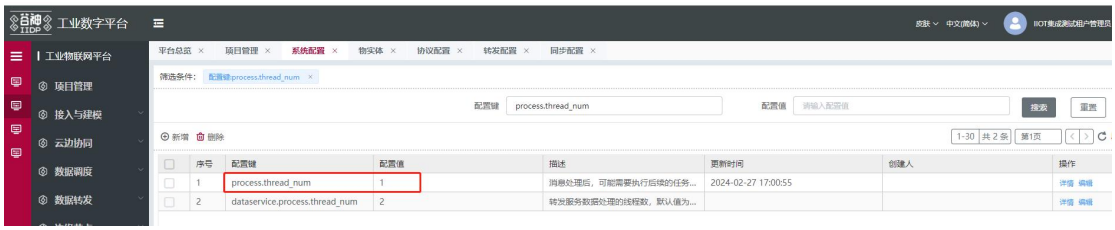
SMDC授权申请　SMDC问题反馈　常用工具下载　SMDC 10.0 使用手册　SIIoT 11.0 使用手册　提工单

IIOT

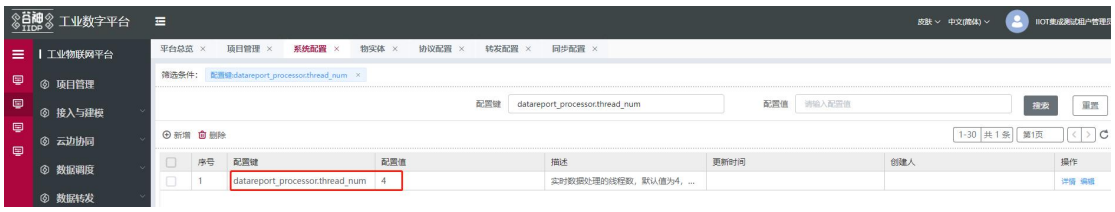
序号	版本迭代	更新内容
1	IIoT平台	迭代更新详细内容

IIOT部署文档　IIOT操作手册　对接文档　工程配置协议扩展文档

第二步：需要在菜单：系统运维》系统配置 中查询：process.thread_num 的值是否是 1；



datareport_processor.thread_num 的值是否是 4；



第三步：如果不是，需要在平台的【开发者中心－种子数据管理】搜索“datareport_processor_thread_num”和“process_thread_num“进行重置

5.4、重启服务，使配置生效

更新检查：

Tdengine-proxy 请求体压缩

6.1 单机部署操作指引

前置条件：

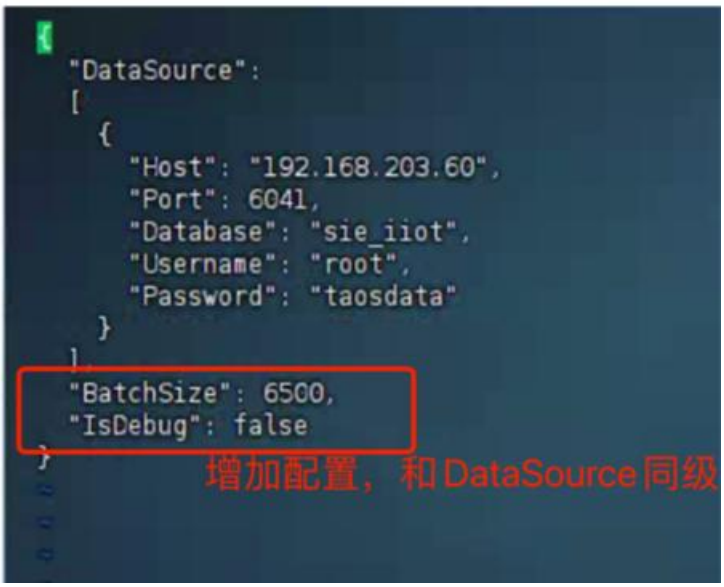
修复脚本：

操作步骤：

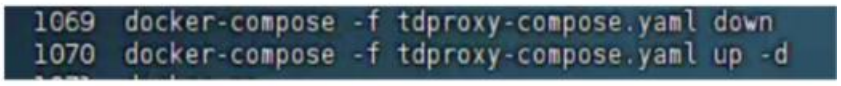
第一步：首先登录服务器，然后拉取 tdengine-proxy 镜像；

第二步：进入目录：cd /var/tdengine-proxy 修改 tdproxy-compose.yaml 镜像名称字段为：

第三步：修改配置文件 appsettings.json；在 DataSource 同级增加配置项："BatchSize": 6500,"IsDebug": false



第四步：重启服务



更新检查：

6.2 分布式+高可用部署操作指引

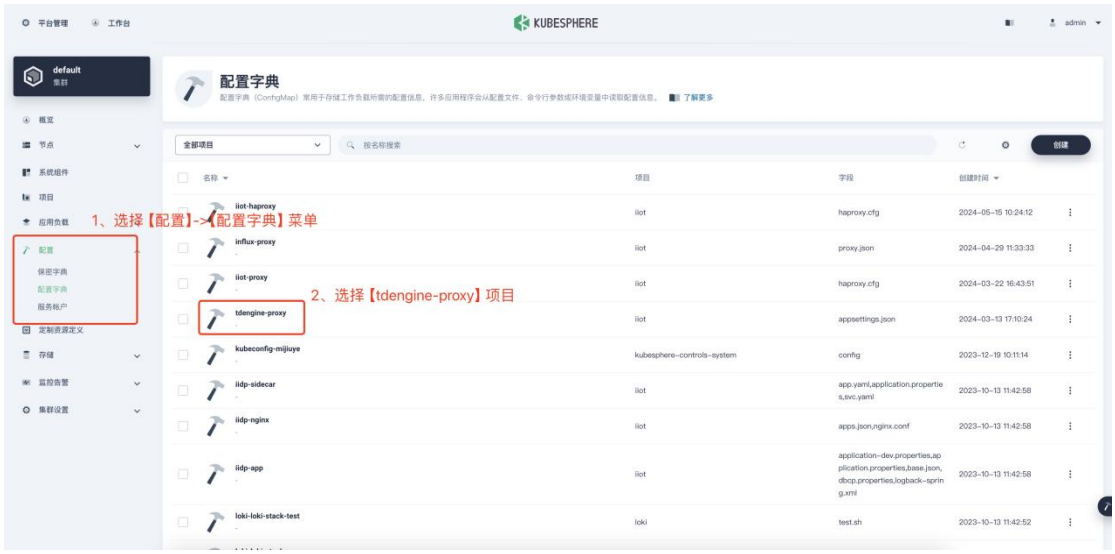
前置条件：

修复脚本：

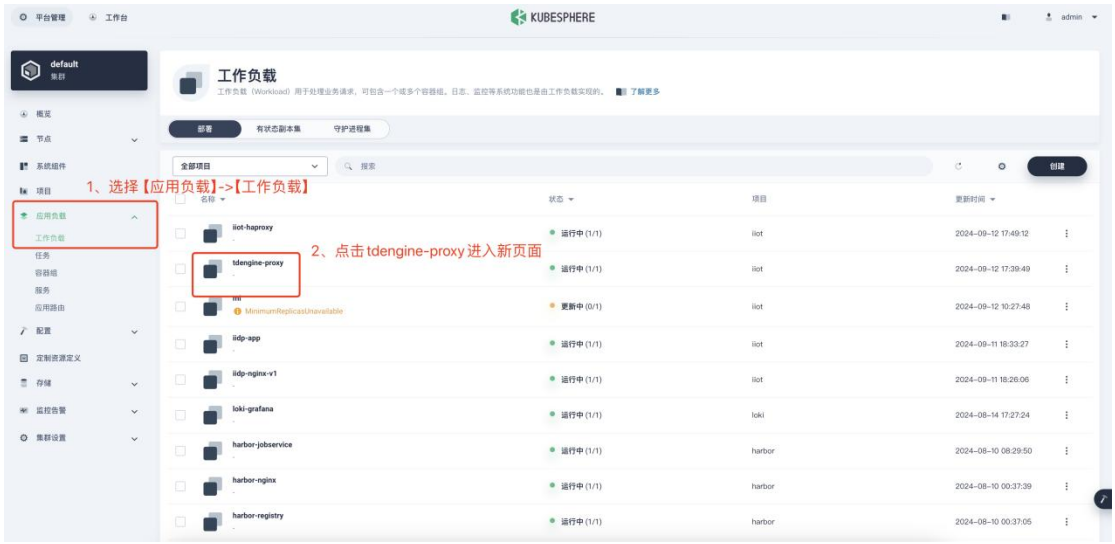
操作步骤：

第一步：登录 K8S web 管理平台，选择【集群管理】进入管理页面。

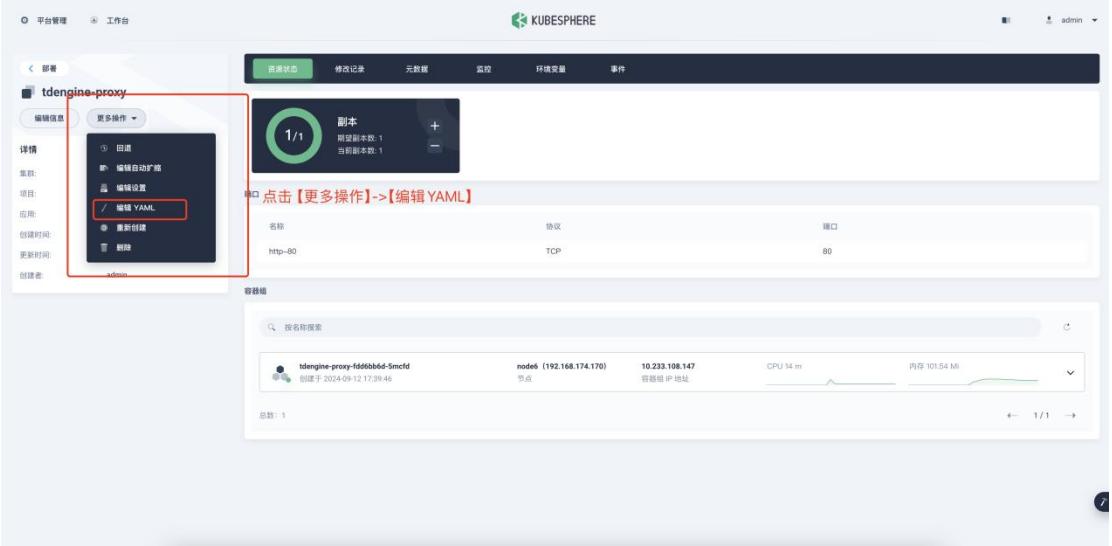
第二步:修改 tdengine-proxy 配置文件,新增 DataSource 同级配置项 BatchSize、IsDebug。点击左侧【配置】→【配置字典】菜单，再点击【tdengine-proxy】项目进入项目配置页面,



第三步：然后点击左侧【应用负载】→【工作负载】菜单，点击右侧 tdengine-proxy 进入服务页面。



第四步：在服务页面点击【更多操作】→【编辑 YAML】选项，根据实际上线版本号修改【image 配置项】里镜像文件的版本号，点击【确定】保存配置修改，服务将自动重启。


KUBESPHERE
tdengine-proxy
编辑信息
更多操作
回退
编辑自动扩缩
编辑设置
编辑 YAML
重新创建
删除
资源状态
修改记录
元数据
监控
环境变量
事件
副本
1/1
点击【更多操作】->【编辑 YAML】
名称
协议
端口
http-80
TCP
80
容器组
tdengine-proxy-fdd6bb6d-5mcfd
node6（192.168.174.170）
10.233.108.147


编辑 YAML
1、修改镜像版本号为目标版本号
2、保存配置修改
取消
确定

更新检查：

无