**配电网云主站系统**

**指标分析功能部署**

**说明文档**

**修订记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本号 | 修订内容 | 修订人 | 审签人 | 发布日期 | 更新说明 |
| V1.0 | 建立文档 | 郁楠 |  | 2020.11.04 |  |
|  | 跟新稳定 | 郁楠 |  | 2021.07.07 | 阿波罗新增redis配置补充说明哪些是必须配置项 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

[1. 部署总览： 4](#_Toc533534523)

[1.1 表 4](#_Toc533534524)

[1.2 涉及到后台C++程序服务 4](#_Toc533534525)

[1.3 jar包 4](#_Toc533534526)

[1.4 配置文件 5](#_Toc533534527)

[2. 四区功能部署 6](#_Toc533534528)

[2.1 相关建表和表结构 6](#_Toc533534529)

[2.2 后台服务部署 7](#_Toc533534530)

[2.3 首页面主菜单配置 7](#_Toc533534531)

[2.4 相关配置文件 9](#_Toc533534532)

## 部署总览：

### 表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **相关表名** | **一商** | **一实** | **四商** | **四实** | **备注** |
| areaindexinfo\_new |  | 否 | √ | 否 | 新区域统计指标信息表（有新增域） |
| yk\_his\_content |  | 否 | √ | 否 | 遥控使用率明细表 |
| yx\_his\_content |  | 否 | √ | 否 | 遥信正确率明细表 |
| term\_his\_stat |  | 否 | √ | 否 | 终端在线明细表 |
| yk\_success\_content |  | 否 | √ | 否 | 遥控成功率明细表 |
| dms\_dev\_term\_rel |  | 否 | √ | 否 | 一次二次设备表（新增表） |
| Fa\_fg\_content |  | 否 | √ | 否 | 通用fa覆盖率明细表 |
| Combined\_gz\_content |  | 否 | √ | 否 | 开关站改造率明细表 |







### 涉及到后台C++程序服务

无

### 前端页面

|  |  |  |
| --- | --- | --- |
| **前端页面文件** | **存放路径** | **备注** |
| zbfx | nginx代理前台页面的路径 | 前端包根据3.1章节进行部署 |

### 镜像包或jar包

|  |  |  |
| --- | --- | --- |
| **镜像包** | **存放路径** | **备注** |
| decloud-zdhzb.tar | 上传到harbor的websvc仓库中 | 后端镜像根据3.2章节进行部署 |

### 配置文件

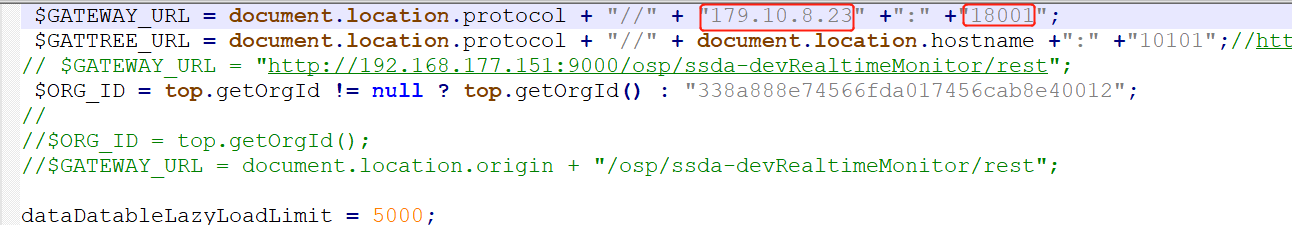
1. 前台配置
2. zbfx/zbsy/js/config.js

ip和port配置为k8s容器提供的ip和port



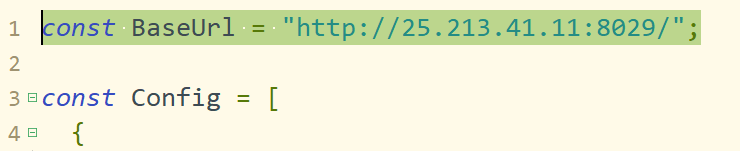
1. zbfx/myconfig/config.js

配置k8s容器提供的ip和port



1. Zbfx/zbsy/config.js

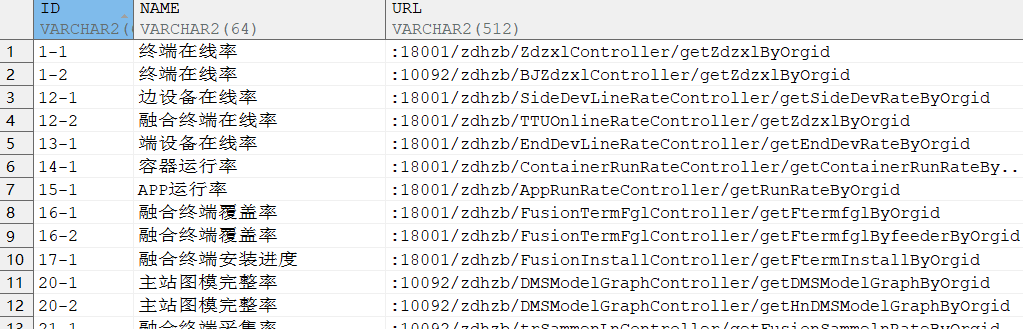
配置k8s容器提供的ip和port



1. 数据库中配置数据
2. zb\_lager

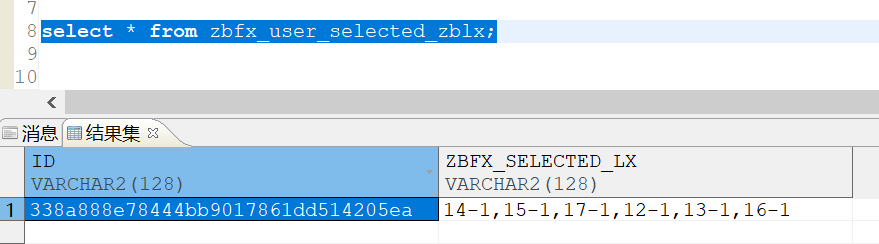
zb\_lager表中URL中修改端口号为k8s容器提供的端口号

例如 :18001/zdhzb/EndDevLineRateController/getEndDevRateByOrgid

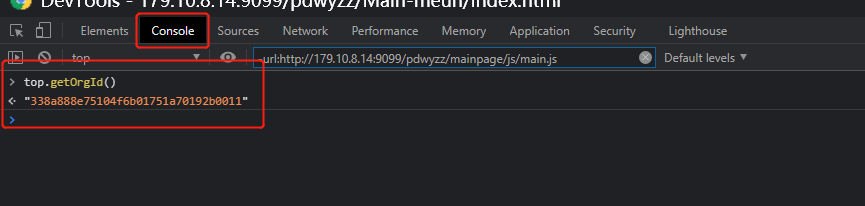


1. zbfx\_user\_selected\_zblx

zbfx\_user\_selected\_zblx表中id配置为orgid；zbfx\_selected\_lx配置指标类型

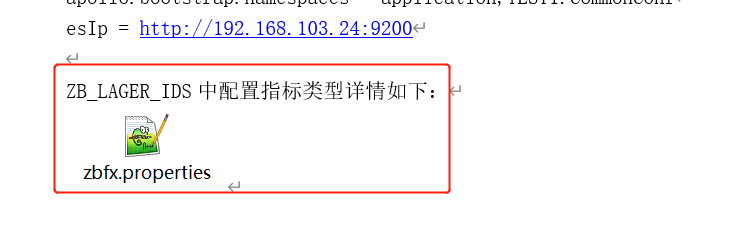


orgid可以在浏览器F12的console中执行top.getOrgId()进行查看，例如



指标类型可参考3.2步骤中指标类型详情进行配置

例如：14-1,15-1,17-1,12-1,13-1,16-1



### 菜单配置

|  |  |
| --- | --- |
| 指标分析 | nginx代理后的地址  例如：http://179.10.8.14:9099/zbfx/zbsy/index.html |

## 服务依赖

### 2.1业务服务调用

终端管理（边设备在线率功能，依赖终端管理模块term\_his\_stat\_pdwyzz的数据）

### 2.2平台服务调用

* 微服务注册中心
* 商用库微服务(DBSERVER)
* Redis

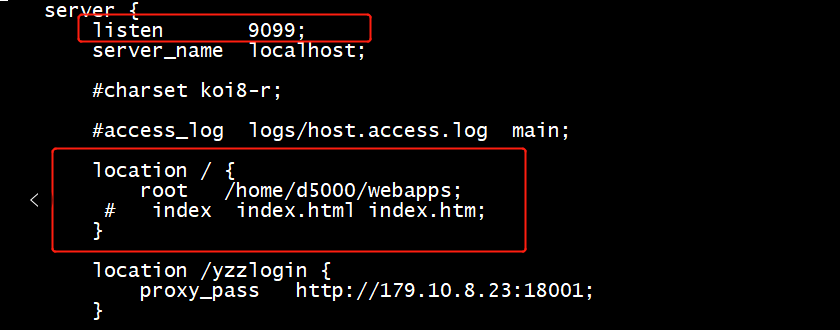
## 启动服务

### 3.1 前端：

(1)通过nginx代理访问

使用nginx代理时，请确认nginx.conf中配置了前台路径。

例如：zbfx的前台文件放在/home/d5000/webapps/下，则nginx.conf的server中配置了该地址，即<http://nginx>的ip:9099/zbfx/zbsy/index.html时能够访问指标分析页面，1.6步骤中的前台菜单配置需要配成该url

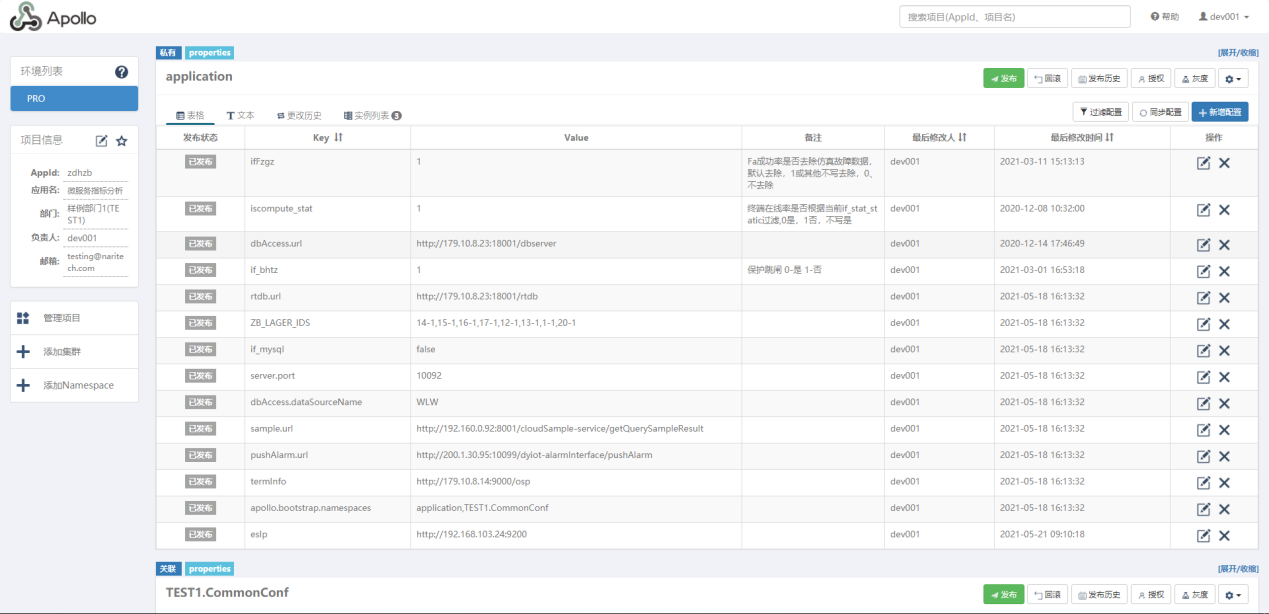


### 3.2 后台

容器内部署：

1. 进行apollo配置

登录apollo，创建项目，应用id为zdhzb



添加私有配置，提交并发布

请结合现场实际情况修改配置文件

参考样例：

ifFzgz = 1

iscompute\_stat = 1

#（必须配置）

dbAccess.url = http://179.10.8.23:18001/dbserver

if\_bhtz = 1

rtdb.url = http://179.10.8.23:18001/rtdb

ZB\_LAGER\_IDS = 14-1,15-1,16-1,17-1,12-1,13-1

if\_mysql = false

server.port = 10092

#（必须配置）

dbAccess.dataSourceName = WLW

sample.url = http://192.160.0.92:8001/cloudSample-service/getQuerySampleResult

pushAlarm.url = http://200.1.30.95:10099/dyiot-alarmlnterface/pushAlarm

termInfo = http://179.10.8.14:9000/osp

apollo.bootstrap.namespaces = application,TEST1.CommonConf

esIp = <http://192.168.103.24:9200>

#直接使用redis配置（以下必须配置）

spring.redis.cluster.nodes=179.10.8.7:17500,179.10.8.7:17501

spring.redis.password=root.2019

spring.redis.timeout=30000

spring.redis.cluster.max-redirects=3

spring.redis.lettuce.max-active=1000

spring.redis.lettuce.max-idle=10

spring.redis.lettuce.max-wait=-1

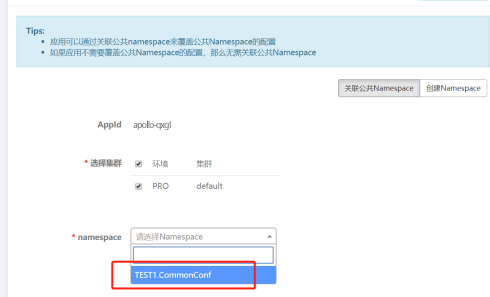
spring.redis.lettuce.min-idle=5

#配变和组织的关系，1：配变直接找组织 。2配变-线路-组织,默认2

trToorg=2

ZB\_LAGER\_IDS中配置指标类型详情如下：







Apollo关联公共配置

（2）获取镜像tar包，在安装了docker的节点执行

docker load -i tar包，将镜像包导入docker内

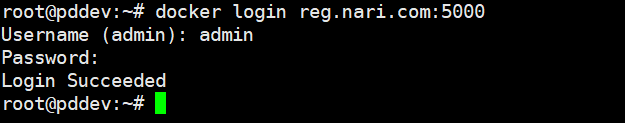
例如：



（3）登录harbor reg.nari.com:5000(需要确保服务器上/etc/hosts文件中已添加了harbor的地址)

docker login reg.nari.com:5000

例如：

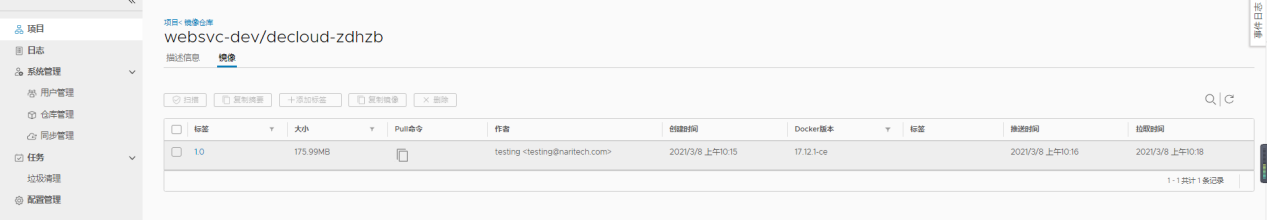


（4）分别执行

docker tag reg.nari.com:5000/websvc/decloud-zdhzb:1.0.0 reg.nari.com:5000/现场实际的镜像仓库名/decloud-zdhzb:1.0.0

docker push reg.nari.com:5000/现场实际的镜像仓库名/decloud-zdhzb:1.0.0

Harbor上可见镜像



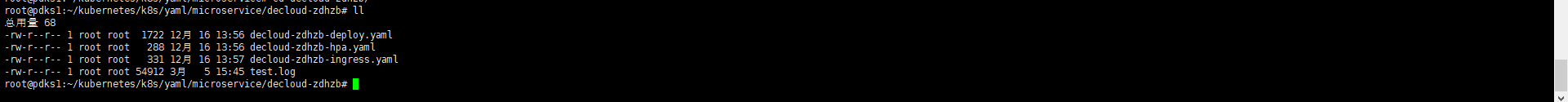
（5）在k8s的master节点上创建yaml文件

进入k8s的master节点/root/kubernetes/k8s/yaml/microservice路径

创建路径 mkdir -p decloud-zdhzb文件夹

上传yaml文件

下图：





（6）启动容器

在k8s的master节点上进入上一步骤创建的路径

执行kubectl get pod查看容器内是否已经部署了应用，

如果已经部署了可以执行kubectl delete -f xxx-deploy.yaml进行删除

分别执行 kubectl create -f decloud-zdhzb-deploy.yaml和kubectl create -f decloud-zdhzb-ingress.yaml

