

正式环境

```
#harbor地址
http://10.166.157.189/harbor
#应用发布地址
http://10.166.157.192:30880/
#nacos地址
http://10.166.157.192:32203/nacos
#网关地址
10.166.157.192:31315
#websocket地址
10.166.157.191:8083/mqtt
#mqtt Tcp地址
10.166.157.191:8099
#控制台页面
http://10.166.157.192:30000/
#10.166.157.189
admin 密码 95FjdwjicpdvrgrkrHc
root 密码 CPs6i7cqjEthTZ4I8wU
```

数据库分配

IP地址	数据库	用户名	密码
10.166.157.184:3306	jq_traffic_vms	trafficvms	trafficVms123!@#
10.166.157.185:3306	jq_vehicle_equipment	vehicleequip	vehicle212Equipment!@#
10.166.157.185:3306	jq_traffic_monitor	trafficmonitor	trafficMonitor123!@#
10.166.157.184:3306	jq_its_skywalking	skywalking	skyWalking123!@#
10.166.157.184:3306	jq_smart_service	smartService	smartService123!@#
10.166.157.185:3306	jq_emergency_server	emergency	Emergency123!@#
10.166.157.185:3306	jq_flow_service	flow	Flow123!@#
10.166.157.184:3306	jq_its_tomic	tomic	Tomic123!@#

nacos配置

nacos用户密码及配置空间所有用户密码同演示服务器

2

public | platform | trafficVms | trafficMonitor | smartService | emergency | flowService | tomic | gisService | assets | demo | wade | syncData | masterdata | vehicleService | iotGateway

配置管理 | platform platform

查询结果: 共查询到 24 条满足要求的配置。

Data ID:

添加通配符""进行模糊查询

Group:

添加通配符""进行模糊查询

查询

高级查询

导出查询结果

新版导出查询结果

导入配置

<input type="checkbox"/>	Data Id ↕	Group ↕	归属应用: ↕	操作
<input type="checkbox"/>	datasource.yaml	QZ_JEEMIS_CLOUD_GROUP	平台	详情 示例代码 编辑 删除 更多
<input type="checkbox"/>	gateway.json	QZ_JEEMIS_CLOUD_GROUP	网关	详情 示例代码 编辑 删除 更多
<input type="checkbox"/>	common.yaml	QZ_JEEMIS_CLOUD_GROUP	公共配置 (redis, 白名单)	详情 示例代码 编辑 删除 更多
<input type="checkbox"/>	swagger-routes.json	QZ_JEEMIS_CLOUD_GROUP	swagger api 配置	详情 示例代码 编辑 删除 更多
<input type="checkbox"/>	stream-rocketmq-common.yaml	QZ_JEEMIS_CLOUD_GROUP	消息队列通用配置	详情 示例代码 编辑 删除 更多
<input type="checkbox"/>	auth-server-oauth-config.yaml	QZ_JEEMIS_CLOUD_GROUP	auth配置	详情 示例代码 编辑 删除 更多
<input type="checkbox"/>	datasource-log.yaml	QZ_JEEMIS_CLOUD_GROUP	日志管理	详情 示例代码 编辑 删除 更多
<input type="checkbox"/>	minio-common.yaml	QZ_JEEMIS_CLOUD_GROUP	对象存储 minio	详情 示例代码 编辑 删除 更多
<input type="checkbox"/>	workflow-datasource.yaml	QZ_JEEMIS_CLOUD_GROUP	工作流	详情 示例代码 编辑 删除 更多
<input type="checkbox"/>	datacenter-datasource.yaml	QZ_JEEMIS_CLOUD_GROUP	数据中心	详情 示例代码 编辑 删除 更多

harbor配置

登录harbor

<http://10.166.157.189/harbor>

各个系统仓库

<div><div>+ 新建项目</div><div>其他操作 ▼</div></div>			
<input type="checkbox"/>	项目名称	访问级别	角色
<input type="checkbox"/>	emergency-server	私有	项目管理员
<input type="checkbox"/>	flow-service	私有	项目管理员
<input type="checkbox"/>	its-tomic	私有	项目管理员
<input type="checkbox"/>	library	公开	项目管理员
<input type="checkbox"/>	optical	私有	项目管理员
<input type="checkbox"/>	platform	私有	项目管理员
<input type="checkbox"/>	plug	私有	项目管理员
<input type="checkbox"/>	pms	私有	项目管理员
<input type="checkbox"/>	project	私有	项目管理员
<input type="checkbox"/>	roadnet	私有	项目管理员
<input type="checkbox"/>	sky-walking	私有	项目管理员
<input type="checkbox"/>	smart-service	私有	项目管理员
<input type="checkbox"/>	traffic-monitor	私有	项目管理员
<input type="checkbox"/>	traffic-vms	私有	项目管理员
<input type="checkbox"/>	vehicle_equipment	私有	项目管理员

用户，各用户密码同数据库密码登录harbor后可进行修改

<input type="checkbox"/>	用户名	管理员 ⓘ	邮件
<input type="checkbox"/>	emergency	否	emergency@sdhs.com
<input type="checkbox"/>	flow	否	flow@sdhs.com
<input type="checkbox"/>	menglq	否	menglq@sdhs.com
<input type="checkbox"/>	pms	否	pms@sdhs.com
<input type="checkbox"/>	skywalking	否	skywalking@sdhs.com
<input type="checkbox"/>	smartService	否	smartService@sdhs.com
<input type="checkbox"/>	tomic	否	tomic@sdhs.com
<input type="checkbox"/>	trafficMonitor	否	tm@sdhsg.com
<input type="checkbox"/>	trafficvms	否	trafficvms@sdhs.com
<input type="checkbox"/>	vehicle	否	3558453274@qq.com
<input type="checkbox"/>	vehicleequipment	否	vq@sdhs.com
<input type="checkbox"/>	zhangmy	否	zhangmy@sdhs.com
<input type="checkbox"/>	zhaosy	否	zhaosy@sdhs.com

```
docker login -u admin -p Sdjhys@12jq6a? http://10.166.157.189
```

```
[root@iZp1e00p3u8ggzr1zgju2Z harbor]# docker login -u admin -p Sdhs@123456a? http://10.166.139.46:9600
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

docker打包镜像并上传

#打包镜像

```
docker tag nginx:1.23 10.166.157.189/plugin/nginx:1.23
```

#上传镜像

```
docker push 10.166.157.189/plugin/nginx:1.23
```

```
[root@iZple0lc6ctakzi6r4wh8aZ docker]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
docker.io/tomcat     latest             c0e850d7b9bb      4 days ago        667 MB
docker.io/mysql      5.7               87eca374c0ed      7 days ago        447 MB
docker.io/portainer/portainer latest           580c0e4e98b0      5 weeks ago       79.1 MB
[root@iZple0lc6ctakzi6r4wh8aZ docker]# docker tag mysql 10.166.139.120/docker.io/mysql
Error response from daemon: no such id: mysql
[root@iZple0lc6ctakzi6r4wh8aZ docker]# docker tag mysql 10.166.139.120/docker.io/mysql:5.7
Error response from daemon: no such id: mysql
[root@iZple0lc6ctakzi6r4wh8aZ docker]# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
[root@iZple0lc6ctakzi6r4wh8aZ docker]# docker run -d -p 3306:3306 -e MYSQL_ROOT_PASSWORD=123456 --name hello-mysql docker.io/mysql:5.7
e950e5d33985b44103808a016b1b87263c02cd0be82bf02cb3d0f05c33c46f06
[root@iZple0lc6ctakzi6r4wh8aZ docker]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
e950e5d33985      docker.io/mysql:5.7  "docker-entrypoint..." 6 seconds ago       Up 4 seconds        0.0.0.0:3306->3306/tcp, 33060/tcp  hello-mysql
[root@iZple0lc6ctakzi6r4wh8aZ docker]# docker tag docker.io/mysql:5.7 10.166.139.120/zhongtai/hello-mysql:v1
[root@iZple0lc6ctakzi6r4wh8aZ docker]# docker push 10.166.139.120/zhongtai/hello-mysql:v1
The push refers to a repository [10.166.139.120/zhongtai/hello-mysql]
065db6810608: Pushed
c72710eaafe9: Pushed
f06c93a4b5ba: Pushed
711202fcd50: Pushed
1ce52ff7c16f: Pushed
2cb169012988: Pushed
2b1742830571: Pushed
7afbfb38fd1da: Pushed
f68961560ec1: Pushed
76233144372b: Pushed
7e178b9c0c8c: Pushed
v1: digest: sha256:9f768489d306402ea11243f1b96aeaa4696adb9ed7c1bb0318724759b9c9bd1a6 size: 2621
```

```
[root@iZple00p3u8ggzr1zguju2Z harbor]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
10.166.139.46:9600/soft/tomcat  openjdk            a993860b1d06      3 days ago        208MB
tomcat              jre8-openjdk-slim-buster a993860b1d06      3 days ago        208MB
cchabhar/harbor-exporter      v2.5.1            02548be65612      5 days ago        86.0MB
```

docker 拉取镜像

```
docker pull 10.166.157.189/plugin/nginx:1.23
```

后端打包

```
FROM openjdk:8-jdk
LABEL maintainer=chenqp

#docker run -e PARAMS="--server.port 9090"
ENV PARAMS="--server.port=8080 --spring.cloud.nacos.username=platform --
spring.cloud.nacos.password=Sdhs@123456a --spring.cloud.nacos.discovery.server-
addr=10.166.157.192:32203 --spring.cloud.nacos.config.file-extension=yaml"
RUN /bin/cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime && echo 'Asia/Shanghai'
>/etc/timezone

COPY target/*.jar /app.jar
EXPOSE 8080

#
ENTRYPOINT ["/bin/sh", "-c", "java -Xms2048m -Xmx2048m -XX:PermSize=1024m -
XX:MaxPermSize=2048m -Dfile.encoding=utf8 -Djava.security.egd=file:/dev/./urandom -jar
app.jar ${PARAMS}"]
```

打包命令

```
docker build -t vehicle-equipment:v2022110402 -f Dockerfile .

docker tag vehicle-equipment:v2022110402 10.166.157.189/vehicle_equipment/vehicle-
equipment:v2022110402

docker push 10.166.157.189/vehicle_equipment/vehicle-equipment:v2022110402
```

前端打包

修改.env.prod文件 VUE_APP_PUBLIC_PATH内容改成分配给子系统的目录名字

```
# 指定构建模式
NODE_ENV=production

VUE_APP_BUILD_MODE=nomock

#所在目录（设置为/为根目录）
VUE_APP_PUBLIC_PATH=/分配的子目录
```

```
npm run build:prod
```

分配信息

系统名	Package.json	二级后缀
情报板业务系统	sdhsg_its_app 1.6.12	vms
交通运行监测与智能管控系统	sdhsg_its_app 1.6.12	monitor
应急指挥调度	sdhsg_its_app 1.6.12	emergency
伴随式服务	sdhsg_its_app 1.6.12	flow
交通运行监测与主动管控	sdhsg_its_app 1.6.12	tomic
机电设备智慧管理系统	sdhsg_its_app 1.6.12	assets
涉路工程		wade
Gis		

目录结构

```
[root@189dataware platform_ui]# ls
conf  Dockerfile  html
[root@189dataware platform_ui]#
```

conf下nginx.conf

这个地方需要注意如果引入模块，在顶部

```
load_module modules/nginx_http_cache_purge_module.so;  
load_module modules/nginx_http_concat_module.so;
```

```
user root;  
worker_processes 1;  
  
events {  
    worker_connections 1024;  
}  
  
http {  
    include mime.types;  
    default_type application/octet-stream;  
    sendfile on;  
    keepalive_timeout 65;  
  
    server {  
        listen 80;  
        server_name -;  
        location / {  
            root html;  
            index index.html index.htm;  
        }  
        location ^~ /platform {  
            alias /home/platform/ui;  
            index index.html index.htm;  
        }  
  
        error_page 500 502 503 504 /50x.html;  
        location = /50x.html {  
            root html;  
        }  
    }  
}
```

Dockerfile文件内容

```
# 基础镜像  
FROM nginx  
  
# author  
MAINTAINER platform  
  
# 挂载目录  
VOLUME /home/platform/ui  
  
# 创建目录  
RUN mkdir -p /home/platform/ui
```

```
# 指定路径
WORKDIR /home/platform/ui
# 复制conf文件到路径
COPY ./conf/nginx.conf /etc/nginx/nginx.conf
# 复制html文件到路径
COPY ./html /home/platform/ui
```

打包命令

```
docker build -t jq-test-ui:v2022102103 -f Dockerfile .
docker tag jq-test-ui:v2022102103 10.166.157.189/project/jq-test-ui:v2022102103
docker push 10.166.157.189/project/jq-test-ui:v2022102103
```

日志配置

在common.yaml中进行配置

```
jeemis:
  logging:
    logstash: # Forward logs to logstash over a socket, used by LoggingConfiguration
      enabled: true
      host: 10.166.157.190
      port: 9092
      queue-size: 512
```

```
100   graylog:
101     enabled: false
102     host: 172.31.11.199
103     port: 12201
104     origin-host: qz_opm
105   logstash: # Forward logs to logstash over a socket, used by LoggingConfiguration
106     enabled: true
107     host: 10.166.157.190
108     port: 9092
109     queue-size: 512
110 interface:
```

本地日志文件

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <include resource="org/springframework/boot/logging/logback/base.xml"/>
  <property name="log.root.level" value="DEBUG"/>
  <property name="log.base" value="/data/logs"/>
  <property name="log.moduleName" value="jq-alioss"/>
  <property name="log.max.size" value="500MB"/>
  <property name="log.pattern"
    value="%date{ISO8601} %-1level [%thread] %logger{56}.%method:%L -
    %msg%n"/>
```



```

<property name="log.charset" value="UTF-8" />

<logger name="org.springframework.web" level="INFO" />
<logger name="org.springframework.sample" level="TRACE" />
<logger name="org.springframework.cloud.gateway" level="TRACE" />

<!-- 开发、测试环境 -->
<springProfile name="dev,test">
    <logger name="org.springframework.web" level="DEBUG" />
    <logger name="org.springframework.sample" level="DEBUG" />
    <logger name="com.qzsoft" level="DEBUG" />
    <logger name="org.springframework.cloud.gateway" level="DEBUG" />
</springProfile>

<!-- 生产环境 -->
<springProfile name="prod">
    <logger name="org.springframework.web" level="ERROR" />
    <logger name="org.springframework.sample" level="ERROR" />
    <logger name="com.qzsoft" level="ERROR" />
</springProfile>

<!-- 日志文件输出 -->
<appender name="infoFile" class="ch.qos.logback.core.rolling.RollingFileAppender">
    <!-- 设置日志不超过${log.max.size}时的保存路径, 注意如果 是web项目会保存到Tomcat的bin目
录下 -->
    <file>${log.base}/info/${log.moduleName}_info.log</file>
    <!-- 日志输出的文件的格式 -->
    <encoder class="ch.qos.logback.classic.encoder.PatternLayoutEncoder">
        <pattern>%date{yyyy-MM-dd HH:mm:ss.SSS} %-5level
[%thread]%logger{56}.%method:%L -%msg%n</pattern>
        <charset>${log.charset}</charset>
    </encoder>
    <!-- 滚动记录文件, 先将日志记录到指定文件, 当符合某个条件时, 将日志记录到其他文件。-->
    <rollingPolicy
class="ch.qos.logback.core.rolling.SizeAndTimeBasedRollingPolicy">
        <!--按天回滚-->
        <fileNamePattern>${log.base}/info/archive/${log.moduleName}_info_%d{yyyy-
MM-dd}.%i.log.zip</fileNamePattern>
        <!--日志最大存储天数-->
        <maxHistory>60</maxHistory>
        <!-- 当天的日志大小 超过${log.max.size}时,压缩日志并保存 -->
        <maxFileSize>${log.max.size}</maxFileSize>
    </rollingPolicy>
    <!--过滤器, 只记录INFO级别的日志-->
    <filter class="ch.qos.logback.classic.filter.LevelFilter">
        <level>INFO</level>
        <onMatch>ACCEPT</onMatch>
        <onMismatch>DENY</onMismatch>
    </filter>
</appender>

```

```

        </filter>
    </appender>

    <!-- 日志文件输出 -->
    <appender name="warnFile" class="ch.qos.logback.core.rolling.RollingFileAppender">
        <!-- 设置日志不超过${log.max.size}时的保存路径，注意如果 是web项目会保存到Tomcat的bin目
录下 -->
        <file>${log.base}/warn/${log.moduleName}_warn.log</file>
        <!-- 日志输出的文件的格式 -->
        <encoder class="ch.qos.logback.classic.encoder.PatternLayoutEncoder">
            <pattern>%date{yyyy-MM-dd HH:mm:ss.SSS} %-5level
[%thread]%logger{56}.%method:%L -%msg%n</pattern>
        </encoder>
        <!-- 滚动记录文件，先将日志记录到指定文件，当符合某个条件时，将日志记录到其他文件。-->
        <rollingPolicy
class="ch.qos.logback.core.rolling.SizeAndTimeBasedRollingPolicy">
            <fileNamePattern>${log.base}/warn/archive/${log.moduleName}_warn_%d{yyyy-
MM-dd}_%i.log.zip
            </fileNamePattern>
            <!--日志最大存储天数-->
            <maxHistory>60</maxHistory>
            <!-- 当天的日志大小 超过${log.max.size}时,压缩日志并保存 -->
            <maxFileSize>${log.max.size}</maxFileSize>
        </rollingPolicy>
        <filter class="ch.qos.logback.classic.filter.LevelFilter">
            <level>WARN</level>
            <onMatch>ACCEPT</onMatch>
            <onMismatch>DENY</onMismatch>
        </filter>
    </appender>

    <!-- 日志文件输出 -->
    <appender name="errorFile" class="ch.qos.logback.core.rolling.RollingFileAppender">
        <!-- 设置日志不超过${log.max.size}时的保存路径，注意如果 是web项目会保存到Tomcat的bin目
录下 -->
        <file>${log.base}/error/${log.moduleName}_error.log</file>
        <!-- 日志输出的文件的格式 -->
        <encoder class="ch.qos.logback.classic.encoder.PatternLayoutEncoder">
            <pattern>%date{yyyy-MM-dd HH:mm:ss.SSS} %-5level
[%thread]%logger{56}.%method:%L -%msg%n</pattern>
        </encoder>
        <!-- 滚动记录文件，先将日志记录到指定文件，当符合某个条件时，将日志记录到其他文件。-->
        <rollingPolicy
class="ch.qos.logback.core.rolling.SizeAndTimeBasedRollingPolicy">
            <fileNamePattern>${log.base}/error/archive/${log.moduleName}_error_%d{yyyy-
MM-dd}_%i.log.zip</fileNamePattern>
            <!--日志最大存储天数-->
            <maxHistory>60</maxHistory>
            <!-- 当天的日志大小 超过${log.max.size}时,压缩日志并保存 -->











```

```
        <maxFileSize>${log.max.size}</maxFileSize>
    </rollingPolicy>
    <filter class="ch.qos.logback.classic.filter.LevelFilter">
        <level>ERROR</level>
        <onMatch>ACCEPT</onMatch>
        <onMismatch>DENY</onMismatch>
    </filter>
</appender>



</configuration>
```

kubesphere配置

用户数据初始化密码 sDh1s@6，第一次登录必须更改密码

	tomic tomic@sdhs.com	<div><div></div>活跃</div>	platform-regular	尚未登录
	flow flow@sdhs.com	<div><div></div>活跃</div>	platform-regular	尚未登录
	emergency emergency@sdhs.com	<div><div></div>活跃</div>	platform-regular	尚未登录
	skywalking skywalking@sdhs.com	<div><div></div>活跃</div>	platform-regular	尚未登录
	smart-service smart@sdhs.com	<div><div></div>活跃</div>	platform-regular	尚未登录
	trafficmonitor trafficmonitor@sdhs.com	<div><div></div>活跃</div>	platform-regular	尚未登录
	vehicleequip vehicleequip@sdhs.com	<div><div></div>活跃</div>	platform-regular	尚未登录
	trafficvms trafficvms@sdhs.com	<div><div></div>活跃</div>	platform-regular	尚未登录
	vehicle 3558453274@qq.com	<div><div></div>活跃</div>	platform-regular	2022-11-04 13:36:11
	zhaosy zhaosy@sdhs.com	<div><div></div>活跃</div>	platform-regular	2022-11-04 11:28:42

按名称搜索

<input type="checkbox"/> 名称	状态	平台角色	最近登录
<input type="checkbox"/>  zhangmy zhangmy@sdhs.com	<div><div></div>活跃</div>	platform-regular	2022-10-17 14:36:09
<input type="checkbox"/>  mengxb mengxb@sdhs.com	<div><div></div>活跃</div>	platform-regular	2022-11-10 13:52:50



企业空间

企业空间是一个组织您的项目和 DevOps 项目、管理资源访问权限以及在团队内部共享资源等的逻辑单元，可以作

☐

名称 ▾

<input type="checkbox"/>		jq-masterdata(主数据)	-
<input type="checkbox"/>		jq-service-area(服务区管理)	-
<input type="checkbox"/>		jq-roadnet-workspace(路网管理空间)	-
<input type="checkbox"/>		jq-pms(济青物联平台)	-
<input type="checkbox"/>		jq-platfrom(济青控制台)	-
<input type="checkbox"/>		system-workspace	system-workspace is a built-in workspace automaticall...

挂载磁盘

更新策略

容器

存储卷

容器组调度规则

存储卷设置

存储卷




host-time


类型: HostPath

/etc/localtime

主机路径



container-alioss



/etc/localtime (只读)



volume-p87es4

存储类型: nfs-storage

jq-platform-logs


存储卷

100Gi

容量

ReadWriteOnce

访问模式



container-alioss



/data (读写)

挂载存储卷

为容器挂载现有存储卷、临时存储卷或 HostPath 存储卷。

挂载配置字典或密码字典

为容器挂载配置字典或密码字典。