# 智能监控-资源监控

## monitor/monitor/resources/index

### 获取接受人数

SELECT

g.id,

g.grp\_name AS grpName,

g.create\_user AS createUser,

g.create\_at AS createAt,

g.come\_from AS comeFrom,

(

SELECT

MAX(alarm\_level)

FROM

grp\_tpl,

tpl

WHERE

g.id = grp\_tpl.grp\_id

AND grp\_tpl.tpl\_id = tpl.id) AS max\_alarmLevel,

(

SELECT

COUNT(grp\_tpl.tpl\_id)

FROM

grp,

grp\_tpl

WHERE

grp.id = grp\_tpl.grp\_id

AND grp.id=g.id) AS countTemplates,

(

SELECT

COUNT(grp\_host.host\_id)

FROM

grp ,

grp\_host

WHERE

grp.id = grp\_host.grp\_id

AND grp.id=g.id) AS countHosts,

(

SELECT

COUNT(DISTINCT rtu.user\_id)

FROM

grp ,

grp\_tpl gt,

tpl t,

action a,

team te,

rel\_team\_user rtu

WHERE

grp.id = gt.grp\_id

AND gt.tpl\_id = t.id

AND t.action\_id = a.id

AND a.uic = te.`name`

AND te.id = rtu.tid

AND grp.id = g.id) AS countUsers

FROM

grp AS g

WHERE

g.grp\_type BETWEEN 1 AND 5

GROUP BY

g.id

ORDER BY

g.grp\_type

## monitor/monitor/resources/edit?id=225

### 告警接收人

/ob-monitor-web/src/main/java/com/harmazing/openbridge/monitor/controller/MonitorResourcesController.java

根据类别获取Team

Team team = monitorResourcesService.findTeamByType(group.getGrpType());

E:\workspace\yh\OpenBridge-monitor\ob-monitor-web\src\main\java\com\harmazing\openbridge\monitor\dao\ResourcesMapper.java

@Select("SELECT t.id, t.name, t.resume, t.creator, t.created, t.creator\_user FROM team t WHERE type = #{type} ")

@Results({

@Result(id = **true**, column = "id", property = "id"),

@Result(property = "member", column = "id", one = @One(select = "com.harmazing.openbridge.alarm.dao.TeamMapper.getMember")) })

Team findTeamByType(String type);

SELECT t.id, t.name, t.resume, t.creator, t.created, t.creator\_user FROM team t WHERE type = 1;

SELECT su.\* FROM rel\_team\_user re,sys\_user su WHERE re.user\_id = su.user\_id and re.tid =890;

# Ob-monitor-web

## 需求讨论

### 2016-8-27

策略详情条件显示1%

节点监控查看提供自定义时间范围

节点信息放到节点组同一页显示（告警和告警邮件等全部放到外面）

节点信息放到首页

资源监控和应用监控页面显示统一

告警策略项

告警输出到ELK日志

## 192.168.31.210虚拟机启动服务



## 查看ngnix映射

/etc/nginx/conf.d/openbridge.f3322.net.conf

server {

listen 88;

server\_name localhost;

location /svn1/ {

proxy\_pass http://192.168.31.214:88;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header Host $http\_host;

}

location /svn2/ {

proxy\_pass http://192.168.31.214:88;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header Host $http\_host;

}

location /monitor {

proxy\_pass http://192.168.31.220:8180;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header Host $http\_host;

}

location /home {

proxy\_pass http://192.168.31.220:8980;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header Host $http\_host;

}

location /store {

proxy\_pass http://192.168.31.220:8280;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header Host $http\_host;

}

location /app {

proxy\_pass http://192.168.31.219:8090;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header Host $http\_host;

}

location /api {

proxy\_pass http://192.168.31.219:8080;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header Host $http\_host;

}

location /os {

proxy\_pass http://192.168.31.220:8880;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header Host $http\_host;

}

location /jenkins/ {

proxy\_pass http://192.168.31.210:8080;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header Host $http\_host;

}

}

## 设置匿名访问(免密码)

SpringWebMvcConfig

security.setAnonymousPaths(**"/portal/\*\*;/docs/\*\*;/auth/\*\*;/metainfo/\*\*;/common/\*\*;/assets/\*\*;"** +  
 **"/login.jsp\*;/index.jsp\*;/logout.jsp\*;/login\*;/mlogin\*;/html/\*\*;/teams/users\*"**);

security.setAnonymousPaths(**"/portal/\*\*;/docs/\*\*;/auth/\*\*;/metainfo/\*\*;/common/\*\*;/assets/\*\*;"** +  
 **"/login.jsp\*;/index.jsp\*;/logout.jsp\*;/login\*;/mlogin\*;/html/\*\*;/team/users\*;/api/action/\*\*;/email"**);

/etc/nginx/conf.d

## api/v1/deploys/dockers

http://127.0.0.1:8080/monitor/api/v1/deploys/dockers?pageNo=1&pageSize=10&tenantId=716mo7m10myxdnfbgfpr3f3dh8npzk

## 策略设置

应用告警添加策略组“保存”操作：

1.保存的时候，传两个参数 add：tplId ，delete：tplId （分别表示添加的tplId和删除的tplId)

2.后台根据add:tplId 执行addExpression ;根据delete：tplId执行deleteExpression

1.对比数据库存储的tplId和页面最终提交的tplId，

对于新增策略，由于后台没有存储expression，则后台对应addExpression操作；

对于删除策略，由于后台已存储expression，则后台对应deleteExpression操作；

策略设置；

基本思路

1.在页面上也在数据库中的metric项直接修改

2.不在页面上而在数据库中的直接删除

3.在页面上而不在数据库中的添加操作

算法实现：

1.增加策略：

直接在tpl表中执行新增操作，并把策略项数据保存到tplData

2.修改策略：

2.1维护tpl 里面tplData，即维护策略项expression模板信息

2.2新增策略项

根据tplId查询expression表里面expression项

如果数据库里面不包含该项metric就添加

根据metric从页面构造的expressoin Map获取想要新增的expression,

如果tpl无对应的策略项则不执行任何动作，

如果tpl有对应的策略项则循环distinctMetricsFromDb，批量新增expression

2.3修改策略项

在页面上也在数据库中的metric项则根据tplId和metric去修改

2.4删除策略项

不在页面上而在数据库中的metric则根据tplId和metric去删除

## @清理默认节点策略

delete from falcon\_portal.strategy where strategy.tpl\_id in (select tpl.id from falcon\_portal.tpl where tpl.type in (1,2,3,4,5));

delete from falcon\_portal.grp\_tpl where grp\_tpl.tpl\_id in (select tpl.id from falcon\_portal.tpl where tpl.type in (1,2,3,4,5));

delete from falcon\_portal.tpl where tpl.type in (1,2,3,4,5);

delete from falcon\_portal.grp where grp.grp\_type in (1,2,3,4,5);

delete from falcon\_portal.action where uic in (select team.name from falcon\_portal.team where team.type in (1,2,3,4,5));

delete from falcon\_portal.team where team.type in (1,2,3,4,5);

delete from falcon\_portal.strategy where strategy.tpl\_id in (select tpl.id from falcon\_portal.tpl where tpl.type=5);

delete from falcon\_portal.grp\_tpl where grp\_tpl.tpl\_id in (select tpl.id from falcon\_portal.tpl where tpl.type=5);

delete from falcon\_portal.tpl where tpl.type=5;

delete from falcon\_portal.grp where grp.grp\_type=5;

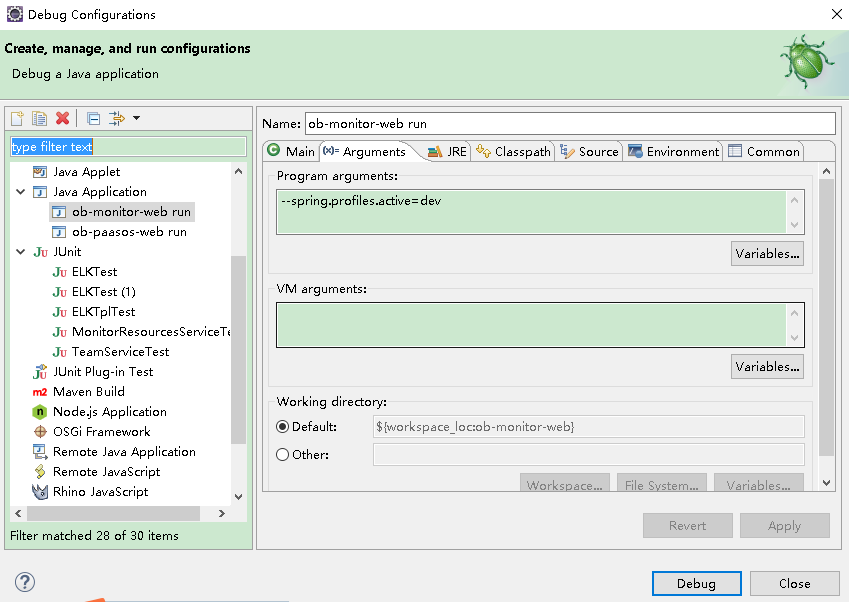
delete from falcon\_portal.strategy where strategy.tpl\_id in (select tpl.id from falcon\_portal.tpl where tpl.type in (1,2,3,4,5));

delete from falcon\_portal.grp\_tpl where grp\_tpl.tpl\_id in (select tpl.id from falcon\_portal.tpl where tpl.type in (1,2,3,4,5));

delete from falcon\_portal.tpl where tpl.type in (1,2,3,4,5);

delete from falcon\_portal.grp where grp.grp\_type in (1,2,3,4,5);

# spring boot profile



# 动态更新配置

ConfigUtil.loadPropertiesOnStartup();

# Falon-alarm提供查询功能

使用PAAS Monitor提供的RESTful服务/team/users\*;/api/action/\*\*

E:\workspace\yh\OpenBridge-monitor\ob-monitor-web\src\main\java\com\harmazing\openbridge\alarm\controller\AlarmFalconController.java

## AlarmFalconController

### /team/users\*

@RequestMapping(value = "/api/action/{id}", method = RequestMethod.***GET***)

@ResponseBody

**public** Map<String,Object> getById(@PathVariable **long** id) {

Map<String,Object> map = **new** HashMap<>();

map.put("msg","");

map.put("data",actionService.findFalconById(id));

**return** map;

}

### /api/action/\*\*

@RequestMapping(value="/team/users",method=RequestMethod.***GET***)

@ResponseBody

**public** Map<String,Object> getAllUsesrByTeamName(@RequestParam String name){

Map<String,Object> map = **new** HashMap<>();

map.put("msg","");

map.put("users",teamService.getAllUserByTeamName(name));

**return** map;

}

## ActionServiceImpl

E:\workspace\yh\OpenBridge-monitor\ob-monitor-web\src\main\java\

com\harmazing\openbridge\alarm\service\impl\ActionServiceImpl.java

### findFlaconById

仅提供uic也就是team.name

@Override

@Transactional(readOnly = **true**)

**public** ActionFalconDTO findFalconById(**long** id) {

**return** actionMapper.findFlaconById(id);

}

@[Select](eclipse-javadoc:%E2%98%82=ob-monitor-web/C:%5C/Users%5C/dell%5C/.m2%5C/repository%5C/org%5C/mybatis%5C/mybatis%5C/3.3.0%5C/mybatis-3.3.0.jar%3Corg.apache.ibatis.annotations(Select.class%E2%98%83Select)([value](eclipse-javadoc:%E2%98%82=ob-monitor-web/C:%5C/Users%5C/dell%5C/.m2%5C/repository%5C/org%5C/mybatis%5C/mybatis%5C/3.3.0%5C/mybatis-3.3.0.jar%3Corg.apache.ibatis.annotations(Select.class%E2%98%83Select~value)={"SELECT id, uic, url, callback, before\_callback\_sms, before\_callback\_mail, after\_callback\_sms, after\_callback\_mail FROM action WHERE id = #{id} "})

## TeamServiceImpl

E:\workspace\yh\OpenBridge-monitor\ob-monitor-web\src\main\java\

com\harmazing\openbridge\alarm\service\impl\TeamServiceImpl.java

### getAllUserByTeamName

/\*\*

\* 根据teamName返回open-falcon所需的User

\*/

**public** List<User> getAllUserByTeamName(String teamName){

**return** teamMapper.getAllUserByTeamName(teamName);

}

Team仅提供teamName，查询到后，判断rel\_team\_user as rtu有没有关联的用户

信息主要由sys\_user as u提供

如果想更改关联用户直接修改rel\_team\_user as rtu中的user\_id即可



@[Select](eclipse-javadoc:%E2%98%82=ob-monitor-web/C:%5C/Users%5C/dell%5C/.m2%5C/repository%5C/org%5C/mybatis%5C/mybatis%5C/3.3.0%5C/mybatis-3.3.0.jar%3Corg.apache.ibatis.annotations(Select.class%E2%98%83Select)([value](eclipse-javadoc:%E2%98%82=ob-monitor-web/C:%5C/Users%5C/dell%5C/.m2%5C/repository%5C/org%5C/mybatis%5C/mybatis%5C/3.3.0%5C/mybatis-3.3.0.jar%3Corg.apache.ibatis.annotations(Select.class%E2%98%83Select~value)={"select u.user\_id as id,u.login\_name as name,u.user\_name as cnname,u.email,u.mobile as phone from team as t,rel\_team\_user as rtu, sys\_user as u where t.id=rtu.tid and rtu.user\_id=u.user\_id and t.name = #{teamName}"})