

RestCloud平台运维手册V2.0

文档编号	V2.0
文档密级	商密 (本文档只授权给谷云公司服务的企业内部传阅，未经谷云许可不得对外传阅)

RestCloud平台 运维手册



谷云科技(广州)有限责任公司

修订历史记录

版本	日期	AMD	修订者	说明
1.0	2020-05-18	A	RestCloud	发布文档
2.0	2021-03-18	M	RestCloud	添加数据库自动备份、Tomcat启停

A-添加，M-修改，D-删除)

目录

- 1. 概述
 - 1.1. 平台系统结构
 - 1.2. 配置数据库备份
 - 1.3. 配置数据库主要结构
 - 1.4. 日志数据库备份
 - 1.5. 日志数据库结构
 - 1.6. Token有效期及加密设置
 - 1.7. 更换SN
 - 1.8. Tomcat启停
- 2. 运维环境介绍
 - 2.1. 服务器配置
 - 2.2. 开发运行环境
 - 2.2.1. 数据库版本
 - 2.2.2. 相关软件包
 - 2.2.3. 下载中心地址
 - 2.2.4. Java Doc地址
 - 2.3. 数据库支持
 - 2.3.1. MONGODB应用场景
 - 2.3.2. 选用MONGODB缘由
- 3. 基本维护
 - 3.1. 业务软件安装
 - 3.2. 配置文件参数配置
 - 3.3. 常见问题解决
 - 3.3.1. 界面无法打开
 - 3.3.2. 接口列表已失效
 - 3.3.3. 数据源无法链接
 - 3.3.4. Mongodb宕机
 - 3.4. 升级安装

- 4. MongoDB维护
- 4.1. 状态监控
- 4.1.1. mongostat监控
- 4.1.2. mongotop监控
- 4.2. 数据备份与恢复
- 4.2.1. 数据备份mongodump
- 4.2.2. 自动备份
- 4.2.2.1创建MongoDB备份目录
- 4.2.2.2新建MongoDB数据库备份脚本 (/data/mongodb_bak/MongoDB_bak.sh)
- 4.2.2.3修改文件属性，使其可执行
- 4.2.2.4修改/etc/crontab，添加计划任务
- 4.2.2.5crontab使用说明
- 4.2.3. 数据恢复mongorestore
- 5. 日志监控
- 5.1. 异常日志

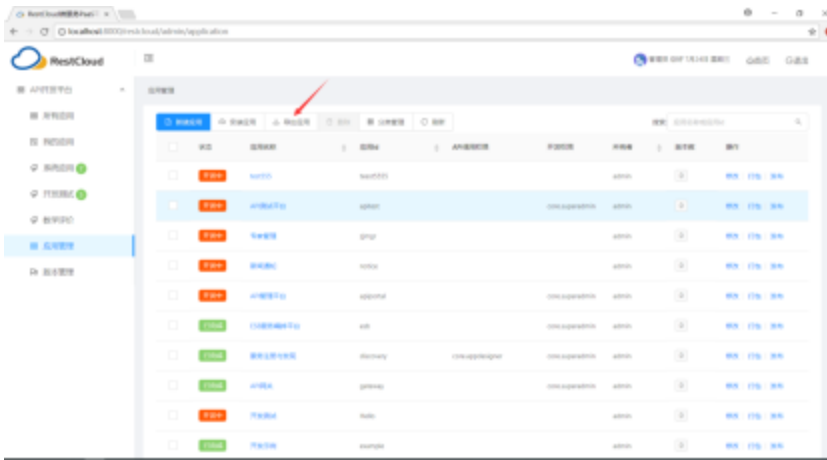
1. 概述

1.1. 平台系统结构

- 1.本平台主要工程文件存储在tomcat/webapps/ROOT目录下，ROOT目录不可以更名。
- 2.ROOT目录下主要有个性化配置的文件为：WEB-INF/classes/application.properties
- 3.所以每次升级时只需要保留application.properties文件即可，其他文件都可以复盖(没有把附件上传到此工程的情况下)
- 4.平台中所有配置数据是存储在mongodb中的

1.2. 配置数据库备份

本平台中所有配置数据存储在MongoDB中，所以需要及时备份mongodb数据库以免丢失数据，如果mongodb数据库丢失或损坏将造成开发的API和配置注册的API全部丢失，备份方式可以使用mongodb的原生备份方法，也可以使用平台界面提供的数据导出功能进行备份。



点击导出应用导出全部应用到tomcat的目录下，然后把导出的json文件进行备份即可，如果mongodb中的数据库丢失只需要重新启动tomcat即可恢复所有运行环境的配置数据以及开发和注册的API。

如果没有购买API开发平台的情况下，无法使用平台提供的备份功能，只能使用原生的mongodb备份方式进行数据库和备份。

在mongodb的bin目录下输入：

```
/data/mongodb/bin/mongodump -u admin -p pass --authenticationDatabase "admin" -d dbName -o /data/mongodb/mongodb_backup/mongodb_backup_temp
```

/data/mongodb/bin/mongodump: mongodb安装目录下mongodump脚本

-u: 用户名

-p: 密码

--authenticationDatabase: 认证数据库

-d: 要备份的数据库名

-o: 备份数据库存放目录

即可完成数据库的备份

1.3. 配置数据库主要结构

平台配置数据库主要表结构如下：

表名	数据说明	备注
P_ActionUrlConfig	平台所有开发和注册的API均存储在本表中, tomcat启动时会自动扫描所有java类的注解并更新此表中的API，如果是通过SQL或脚本发布的则不存在此过程	APi 比较多时需要索引
P_ApiConfigList	API管理门户中的所有API存储表，API发布时会从P_ActionUrlConfig表中拷贝一份到此表中	APi 比较多时需要索引
...

P_ApplicationConfig	所有开发平台中创建的应用列表	
P_ApiProtalApplication	API管理门户中创建的应用列表	
P_BeanConfig	平台所有开发的Java Bean配置数据存储表，tomcat启动时会自动扫描所有java类的注解并更新此表	Bean比较多时需要索引
P_CodeRepository	平台的代码仓库，所有可编辑代码的地方每次保存时会把历史版本保存到本表中	
P_CoreSNConfig	本平台序列号存储表，如果表中没有数据则系统会启动失败,如果application.properties中配置了sn项则会自动创建本表和数据	
P_DataSourceConfig	所有数据源的配置信息均在此表中	
P_DiscoveryServerConfig	如果有注册中心，则所有服务实例的数据存在此表中	
P_DistributedTransaction	分布式事务的记录表，每次请求外部API时系统拦截器会自动记录API的请求数据到此表中，在失败时会使用此表中的数据进行重试。	
P_ESBProcessModelConfig	ESB编排平台中所有流程的配置数据表	
P_ESBProcessNodeConfig	ESB编排平台中所有节点的配置数据表	
P_ESBProcessNodeInstance	ESB编排平台中所有节点的运行历史数据表	
P_GatewayRouterConfig	网关的所有路由配置数据存储表	
P_ModelConfig	所有数据模型的存储表	
P_OrgPersons	用户数据	
P_OrgRoles	角色数据	
P_OrgRolesMembers	角色成员映射表	
P_PermissionConfig	权限列表	
P_PermissionMapResource	权限与API的映射表	
P_PermissionsMapOrg	权限与角色的映射表	
P_PlatformConfig	平台配置的相关表	
P_RdbSqlConfig	所有SQL脚本和JS脚本存储在此表中	
P_RecycleDocs	平台的回收站数据	

P_ViewTemplateConfig	API开发平台的视图模板表	
P_WarningMessageEntry	平台内置的预警消息表	

1.4. 日志数据库备份

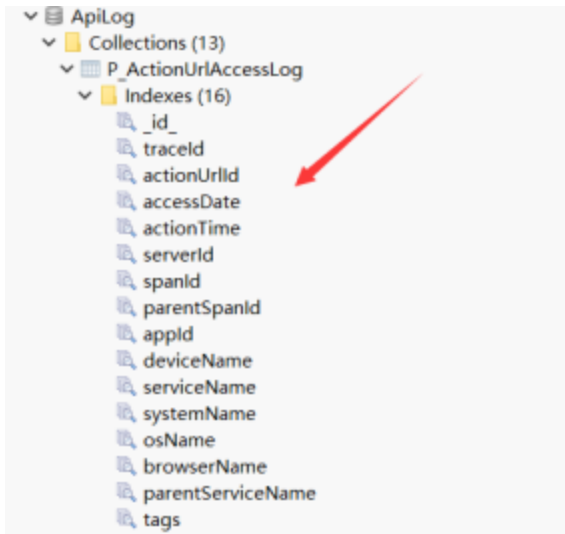
平台的日志数据和配置数据是分开存储的，日志数据在application.properties文件中配置的名称
spring.data.mongodb.log.database=ApiLog

表示日志数据存储(ApiLog库中，如果日志库比较大需要按月或按年进行存储，可以按如下规则进行配置：

spring.data.mongodb.log.database=ApiLog{yyyy-MM}

表示每月一个日志文件数据库，数据库名为ApiLog2019-07，ApiLog2019-08

如果日志数据库文件比较大的情况下，会影响API监控平台的展示速度，所以需要对日志库进行索引，索引的字段如下：



索引字段

如果日志文件选 存储到文件中再转存到mongodb下则可以在\classes\static\log\api中看到产生的日志文件。

名称	修改日期	类型	大小
00_ACCESS.log	2019/7/24 0:23	文本文档	10 KB
09_ACCESS.log	2019/7/24 9:43	文本文档	16 KB
10_ACCESS.log	2019/7/24 10:59	文本文档	117 KB
11_ACCESS.log	2019/7/24 11:34	文本文档	211 KB
14_ACCESS.log	2019/7/24 14:46	文本文档	4 KB
15_ACCESS.log	2019/7/24 15:47	文本文档	208 KB
18_ACCESS.log	2019/7/24 18:11	文本文档	21 KB
19_ACCESS.log	2019/7/24 19:23	文本文档	17 KB

日志文件为每小时一个，系统通过定时任务10秒一次读取此log文件并写入到mongodb中进行日志分析。

1.5. 日志数据库结构

表名	数据说明	备注
P_ActionUrlAccessLog	所有日志数据会存储到此表中，默认的情况下系统会直接把日志数据插入到此表中，如果不想直接插入可以配置平台先写log文件，再用定时任务回写mongodb中的本表，在application.properties文件中的 restcloud.apilog.writer.type=file 即可，如果不配表示直写mongodb	
P_TestServiceLogEntry	如果测试平台的数据没有特别配置则测试平台的数据也会存储在此表中,此表是所有用户的测试记录，如果需要把测试平台的数据独立成一个库可以在属性文件中配置选项： spring.data.mongodb.testter.database=testdb 即可把测试平台的数据存储到此库中	
P_TestPlanConfig	所有测试任务	

日志表数据字典如下：

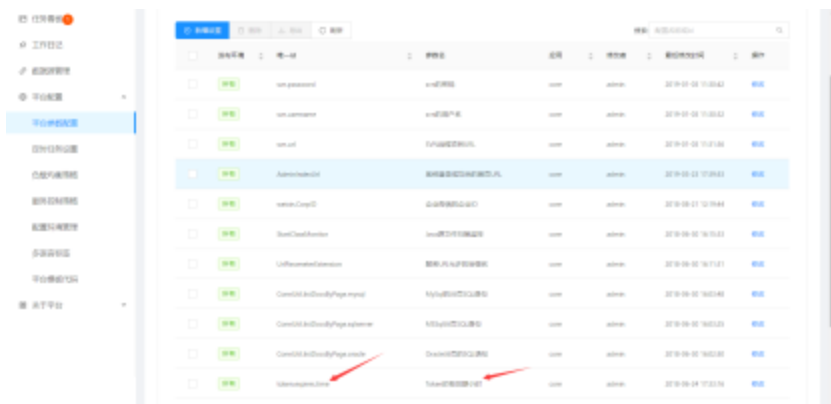
```
private String traceId;//本次服务请求的全局统一id
private String parentSpanId;//本次请求的上级spanId
private String parentServiceName;//本次请求的上级服务名
private String spanId;//本次请求的spanId
private String requestUrl;//请求的url地址
private String actionMapUrl;//服务的配置地址
private String actionName; //服务配置的名称
private String actionUrlId;//服务的配置的_id
private String systemId;//访问本次服务的系统id
private String methodType;//url发生method类型GET|PUT|DELETE|PUT
private String accessDate;//访问日期,按日期统计时使用
private String viewId;//最后响应的视图id
private String success;//1表示调用成功，0表示响应出错或异常
private long runTotalTime; //运行总时间毫秒
private String ip;//请求ip
private String userId;//请求用户
```



```
private String userName; //用户名
private String serverId; //当前服务器的唯一id
private String serviceName; //服务实例名
private String serverIP; //运行此服务的服务器ip
private String port; //服务所在端口
private String deptCode; //用户所在部门id
private String systemName; //请求的业务系统名称
private String actionTime; //记录时间
private String rolesCode; //当前用户所具有的角色
private String browserName; //请求浏览器名称
private String browserVersion; //请求浏览器版本
private String exceptionLog; //本次请求产生的异常信息
private String osName; //请求端操作系统的名称
private String deviceName; //设备类型

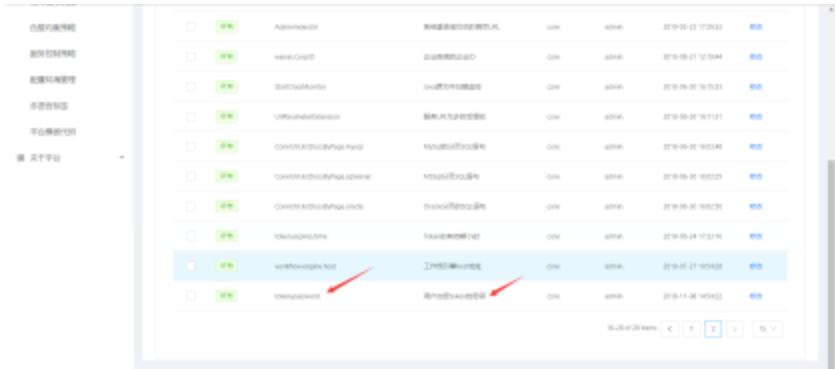
private String backendUrl; //请求后端的url地址
private String referer; //请求页面的url
private String inParams; //客户端传入的post或者get数据
private String inRequestBody; //客户端传入的requestBody数据
private String inHeaderStr; //客户端传入的http header头数据
private String responseHeaderStr; //后端服务器响应的http头数据
private String responseBody; //后端服务响应的内容
private int responseCode; //后端服务响应的状态码
private List<String> tags; //自定义日记的业务标签
```

1.6. Token有效期及加密设置



Token的有效期可以`token.expires.time`中设定为几个小时，最小单位为1个小时，超时后token将失效。

Tokne的加密密码需要进行修改，否则有可能多个项目地的token可以共享使用，存在安全性问题，token的加密密码设置如下：



在平台配置中的token.password中进行设置，可以是任意8位的字符串。

系统之间通信的密码设定，在多个集群服务器之间底层是通过http的api接口进行通信的，api之间通信也有一个token但是这个token与用户的token不是同一个密码，这个密码必须要在application.properties中进行设定。

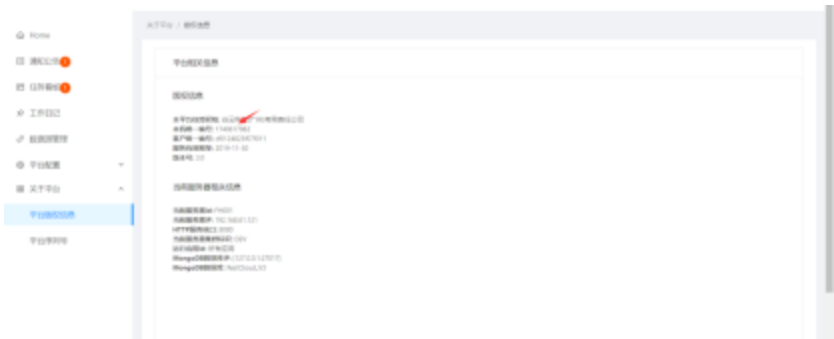
restcloud.token.password.sys=dispass

建议所有同一个环境安装的系统的通信密码设置成一样，不同环境的密码修改成不一样，否则测试环境的机器可以与生产环境的机器进行通信。

1.7. 更换SN

平台测试环境一般提供是有期限的SN，有期限的SN的好处时不与机器进行绑定，但是到期后需要进行定时更换一般用于测试和开发人员使用。

正式生产环境需要把机器码发给我们获取无期限的SN序列号，机器码的获取方式如下：



把本机的唯一编码发送给我们，如果有多台机器都需要发送。

获取新的SN后在application.properties文件中的最后一行的restcloud.sn=进行更换并重启tomcat即可(4.2以上版本可以直接在页面修改保存SN)。

注意：SN过期后系统在启动时会进行提示，并且tomcat启动会不成功。

1.8. Tomcat启停

Tomcat启动命令

```
sh tomcat/bin/startup.sh
```

Tomcat关闭命令（shutdown.sh脚本不能完全停止平台，需要使用杀死进程的办法）

```
ps -ef | grep tomcat
```

```
kill -9 $pid
```

2. 运维环境介绍

2.1. 服务器配置

第1台

用途	MongoDB数据库,数据库版本3.X以上					
操作系统	CPU	内存	磁盘	网络	端口号	备注
Windows2012 以上 或Linux CentOS7	2.4GH 8核	16G	80 G	内网	27 017	Jdk1.8 以 上 版本,MongoDB3.X以上版本

第2台

用途	API网关					
操作系统	CPU	内存	磁盘	网络	端口号	备注
Windows2012 以上 或Linux CentOS7	2.4GH 8核	16 G	80 G	内网	80	Jdk1.8以上版本,MongoDB3.X以上版本，验证环境暂不做集群部署

具体配置和部署以实际情交为准

2.2. 开发运行环境

2.2.1. 数据库版本

名称,版本,客户端连接工具

名称	版本	客户端连接工具
MongoDB	3.X	Robomongo

2.2.2. 相关软件包

本系统包含以下安装软件:

1. 数据库包:

Linux服务器:mongodb-linux-x86_64-3.6.4.tgz

Windows服务器:mongodb-windows-64.msi

2. 数据库客户端工具包(用于连接MongoDB数据库)

mongoClient-0.9.rar

3. API网关应用包

restcloudv3.6.30.rar安装包(安装包会不定期升级)

4. JDK包

Linux版本:jdk-8u201-linux-x64.tar.gz

Windows版本:jdk-8u201-windows-x64.exe

5. Tomcat应用服务器包

Linux版本:apache-tomcat-8.5.39.tar.gz

Windows版本:apache-tomcat-8.5.37-windows-x64.zip

2.2.3. 下载中心地址

<http://47.92.110.139:8080/download>

可以在此下载中心获取最新的版本进行升级.

2.2.4. Java Doc地址

<http://120.79.90.89/javadoc/index.html>

可以查看平台中所有常用类的方法和属性

2.3. 数据库支持

选用MONGODB做为核心数据存储数据库

2.3.1. MONGODB应用场景

网站数据：适合实时的插入，更新与查询，并具备网站实时数据存储所需的复制及高度伸缩性。

缓存：由于性能很高，也适合作为信息基础设施的缓存层。在系统重启之后，搭建的持久化缓存可以避免下层的数据源过载。

大尺寸、低价值的数据：使用传统的关系数据库存储一些数据时可能会比较贵，在此之前，很多程序员往往会选择传统的文件进行存储。

高伸缩性的场景：非常适合由数十或者数百台服务器组成的数据库。

用于对象及JSON数据的存储：MongoDB的BSON数据格式非常适合文档格式化的存储及查询。

2.3.2. 选用MONGODB缘由

选用MongoDB的数据是以BSON的数据格式，高度伸缩方便扩展，并且数据水平扩展非常简单，支持海量数据存储，性能强悍。

3. 基本维护

3.1. 业务软件安装

详情见安装手册

3.2. 配置文件参数配置

#当前应用所处环境

spring.profiles.active=DEV

#http端号，与tomcat保持一至

server.port=8080

#是否开启feign调用api接口功能

feign.okhttp.enabled=true

#tomcat的一些配置信息

server.tomcat.uri-encoding=UTF-8

spring.http.encoding.charset=UTF-8

spring.http.encoding.enabled=true

spring.http.encoding.force=true

spring.messages.encoding=UTF-8

#在maven工程中是否开启spring-boot的热加载功能，建议关闭

spring.devtools.restart.enabled=false

spring.devtools.restart.exclude=test/**,static/**,public/**

#链接mongodb的数据库配置信息

spring.data.mongodb.authentication-database=admin

#mongodb中的数据库名可以任意指定

spring.data.mongodb.database=RestCloud_V40

#日志库的数据库名

spring.data.mongodb.log.database=ApiLog

spring.data.mongodb.org.database=

#如果是集群配置多个ip用逗号加端口

spring.data.mongodb.host=127.0.0.1:27017

spring.data.mongodb.repositories.enabled=true

#数据库账号

spring.data.mongodb.username=admin

#数据库密码

spring.data.mongodb.password=pass

#eureka注册中心地址

#Discovery client option for eureka

#eureka.client.service-url.defaultZone=http://47.107.60.192:8082/eureka

#dubbo提供者服务名称

#dubbo.application.name=restcloud-dubbo-provider

#dubbo服务的ZKClient地址

#dubbo.registry.address=zookeeper://120.77.47.73:2181

```

#restcloud config start*****
#cross and header config
restcloud.Access-Control-Allow-Origin=*
#content-type,x-requested-with,Authorization,Accept,
identitytoken,systemid,systempwd,spandid,traceid,servicename,currentroleid
restcloud.Access-Control-Allow-Headers=content-type,x-requested-with,Authorization,Accept,
identitytoken,systemid,systempwd,spanid,traceid,servicename,currentroleid
restcloud.Access-Control-Max-Age=360000
restcloud.Access-Control-Allow-Credentials=false
#是否支持异步servlet功能
restcloud.async.supported=false
#异步servlet的最大线程数
restcloud.async.thread.maxnum=300
#异步servlet的超时时间
restcloud.async.thread.timeout=300000

#本系统的登录的地址如果有多个用逗号分隔
restcloud.LoginUrl=/rest/core/auth/login
#是否开启appkey认证方式,默认为true(不修改)
restcloud.token.appkey.login=true
#appkey认证时的变量ID
restcloud.token.appkey.keyid=appkey
#appkey认证时的用户ID
token.appkey.default.userid=defaultUser

#系统根目录名称
#can be set to other values
restcloud.api.root.path=/restcloud
#网关根目录名称
#The root path of the gateway can be set to "/"
restcloud.gateway.root.path=/gateway

#系统启动时是否停止数据初始化true表示停止， false表示开启
restcloud.StopInitInstallData=true
#是否停止所有定时任务
restcloud.StopSchedulerTask=false

```

```
#关闭所有PrintUtil打印的消息
restcloud.StopPrintUtilInfo=false
#maven工程中自动扫描注解的包路径
restcloud.AutoScanPackageName=cn.restcloud
#是否开启自动扫描功能
restcloud.AutoScanAndRegJavaBean=true
restcloud.AutoScanAndRegService=true
restcloud.StartHotLoadClass=true
#组织架构所在数据库的类型
restcloud.OrgDbType=mongo
restcloud.ViewTemplateType=FreeMarkerView
#集群系统之间通信的加密密码
restcloud.token.password.sys=dispass
#配置本系统可以显示的系统模块
restcloud.show.module=

#工程打包成jar时,配置LOG库的地址
#If you run in jar package, you must specify a path and copy res dir to this dir sample:
d:/restcloudlog
restcloud.ResourceLocationPath=
restcloud.ResourceLocationPath.log=
#配置日志记录
#config the api log save type file or mongo
#日志记录方式,默认mongo (写入mongodb数据库),file(写入日志文件)
restcloud.apilog.writer.type=mongo
#日志缓存多少秒后写入库或文件
restcloud.apilog.writer.cache.maxtime=0
#日志写入时启动debug功能
restcloud.apilog.writer.debug=true

#Discovery client option
#指定注册中心的服务器地址
restcloud.discovery.server.host=
#当前用户的权重
restcloud.discovery.server.weight=10
restcloud.discovery.server.extprops=
```



```
#当前服务器的名称
restcloud.discovery.current.servicename=FH001
#当前服务器的ip, 不指定表示自动计算
restcloud.discovery.current.server.ip=

#configcenter client option
#是否开启配置中心功能
restcloud.discovery.config.enable=false
#配置中心的服务器地址
restcloud.discovery.config.server.host=
#使用配置中心的应用id
restcloud.discovery.config.appid=RC001
#配置环境
restcloud.discovery.config.environment=DEV
restcloud.discovery.config.update-local-data=false
#开启debug功能
restcloud.discovery.config.debug=true

#Current Server Config option
restcloud.CoreServerAppld=core,base,discovery,gateway,esb
restcloud.CurrentServerRunAppld=*
#多个集群服务器需要使用同样的集群标识
restcloud.CurrentServerClusterFlag=DEV
#当前服务器的唯一id
restcloud.CurrentServerId=FH001
#当前服务器提供服务的根路径
restcloud.CurrentServerHost=http://{ip}:{CurrentServerPort}/{root}
restcloud.CurrentServerType=

#配置网关的信息,无需修改
#Gateway Server url
restcloud.gateway.forward.maxtime=
restcloud.gateway.service.name=gateway
#序列号
restcloud.sn=
```

```
#restcloud config end*****
```

3.3. 常见问题解决

3.3.1. 界面无法打开

1. 请确保tomcat服务器中应用安装目录正确(tomcat应用服务器下webapps目录下只有ROOT一个文件夹,其余全部删除)
2. 若还是无法打开,重启tomcat服务器
3. 访问路径地址http://{ip}:{port}/restcloud/admin

3.3.2. 接口列表已失效

1. 重启mongodb服务器,若还有错误提示可能是SN过期或者mongodb中没有初始化数据.
2. 重启tomcat服务器前请先启动mongodb数据库

3.3.3. 数据源无法链接

如果链接的数据库中间断过,有可能会造成数据源重试链接多次失败后造成后继无法再进行链接,这时需要对tomcat进行重启才能链接成功。

正常情况下如无法链接多半是数据库地址或者网络不通造成的,链接不成功的情况下应第一时间查看控制台日志见4.1中查看的地方。

3.3.4. Mongodb宕机

当出现mongodb不可用时系统中所有对外提供的API可以正常运行不受影响,因为平台中所有的数据都已经缓存到本地map对象中,但如果是直写日志到mongodb的情况下日志数据写入失败会造成api运行变慢,如果是先写log再写mongodb的情况下不影响api的正常运行,但是tomcat会有一些异常信息报出来,不影响基本运行,此时管理平台暂时不可用。Mongodb恢复后系统可以立即恢复所有功能。

3.4. 升级安装

1. 新工程替换旧工程,须删除(最好时移动并进行备份以便回滚时使用)原来的工程文件目录所有文件(其中application.properties配置文件可以复制出来保留).确保tomcat服务器webapps文件夹目录下无ROOT文件夹,
2. 新工程(war)文件直接复制到tomcat服务器webapps文件夹下,启动tomcat服务器,查看tomcat是否正常启动成功,如果成功,先停止tomcat服务器.将webapps文件夹下新工程的文件夹名改成ROOT.并删

除war文件.再次启动tomcat服务器,同时旧的application.properties替换下载安装包中的application.properties文件

3. 新工程(rar),(zip)等压缩文件,须在本机解压成文件夹形式,再上传至tomcat服务器webapps文件夹下ROOT文件夹下(ROOT文件夹没有就手工创建一个).启动tomcat服务器即可.

4. 部分升级,因为整个war包比较大,也可以部分进行升级,平台比较大的原因是因为lib目录下的jar包较多,但一般情况下jar包不会变化,所以无需重新覆盖,可以只覆盖webapps\WEB-INF\classes即可。

4. MongoDB维护

4.1. 状态监控

单机命令

```
db.status();
```

集群命令

```
db.serverStatus();
```

说明:通过此命令,可以查询集群的成员的集合数量、索引数量等相关数据。

4.1.1. mongostat监控

mongostat是mongodb自带的状态检测工具,在命令行下使用。它会间隔固定时间获取mongodb的当前运行状态,并输出。如果你发现数据库突然变慢或者有其他问题的话,你第一手的操作就考虑采用mongostat来查看mongo的状态。

启动你的Mongod服务,进入到你安装的MongoDB目录下的bin目录,然后输入mongostat命令,例:

```
./mongostat
```

或(设有认证时)

```
./mongostat -u admin -p pass --authenticationDatabase admin
```

4.1.2. mongotop监控

mongotop也是mongodb下的一个内置工具, mongotop提供了一个方法,用来跟踪一个MongoDB的实例,查看哪些大量的时间花费在读取和写入数据。 mongotop提供每个集合的水平的统计数据。默认情况下, mongotop返回值的每一秒。

启动你的Mongod服务,进入到你安装的MongoDB目录下的bin目录,然后输入mongotop命令,例:

```
./mongotop
```

或（设有认证时）

```
./mongotop-u admin -p pass --authenticationDatabase admin
```

4.2. 数据备份与恢复

4.2.1. 数据备份mongodump

在Mongodb中我们使用mongodump命令来备份MongoDB数据。该命令可以导出所有数据到指定目录中。

mongodump命令可以通过参数指定导出的数据量级转存的服务器


例，

```
./mongodump -u admin -p pass 表示备份MONGODB所有数据库
```

4.2.2. 自动备份

4.2.2.1创建MongoDB备份目录


Plain Text

 复制代码

```
1 [root@localhost ~]# mkdir -p /backup/mongodb_bak/mongodb_bak_now
2 [root@localhost ~]# mkdir -p /backup/mongodb_bak/mongodb_bak_list
```

4.2.2.2新建MongoDB数据库备份脚本（/data/mongodb_bak/MongoDB_bak.sh）

Plain Text

 复制代码

```
1 [root@localhost ~]#vi /data/mongo/MongoDB_bak.sh
```

```
1  #!/bin/bash
2  #backup MongoDB
3  #mongodump命令路径
4  DUMP=/data/mongo/bin/mongodump
5  #临时备份目录
6  OUT_DIR=/backup/mongodb_bak/mongodb_bak_now
7  #备份存放路径
8  TAR_DIR=/backup/mongodb_bak/mongodb_bak_list
9  #获取当前系统时间
10 DATE=`date +%Y_%m_%d`
11 #数据库账号
12 DB_USER=root
13 #数据库密码
14 DB_PASS=root@123
15 #DAYS=15代表删除15天前的备份，即只保留近15天的备份
16 DAYS=15
17 #最终保存的数据库备份文件
18 TAR_BAK="mongodb_bak_${DATE}.tar.gz"
19 cd $OUT_DIR
20 rm -rf $OUT_DIR/*
21 mkdir -p $OUT_DIR/$DATE
22 #备份全部数据库
23 $DUMP -h 10.171.53.75:27017 -u $DB_USER -p $DB_PASS --authenticationDatabase
    "admin" -o $OUT_DIR/$DATE
24 #压缩为.tar.gz格式
25 tar -zcvf $TAR_DIR/$TAR_BAK $OUT_DIR/$DATE
26 #删除15天前的备份文件
27 find $TAR_DIR/ -mtime +$DAYS -delete
```

4.2.2.3修改文件属性，使其可执行

```
1  [root@localhost ~]#chmod +x /data/mongo/MongoDB_bak.sh
```

4.2.2.4修改/etc/crontab，添加计划任务

```
1  [root@localhost ~]# vi /etc/crontab
```

常用指定：

#每星期六晚上20:30开始执行MongoDB数据库备份脚本

```
30 20 * * 6 root /data/mongo/MongoDB_bak.sh
```

添加一下内容：表示每天早上23.59备份一次数据库

```
59 23 * * * root /data/mongo/MongoDB_bak.sh
```

4.2.2.5crontab使用说明

1 crontab命令

cron服务提供crontab命令来设定cron服务的，以下是这个命令的一些参数与说明：

crontab -u //设定某个用户的cron服务，一般root用户在执行这个命令的时候需要此参数

crontab -l //列出某个用户cron服务的详细内容

crontab -r //删除某个用户的cron服务

crontab -e //编辑某个用户的cron服务

比如说root查看自己的cron设置:crontab -u root -l

再例如，root想删除fred的cron设置:crontab -u fred -r

在编辑cron服务时，编辑的内容有一些格式和约定，输入:crontab -u root -e

进入vi编辑模式，编辑的内容一定要符合下面的格式:*/1 * * * * ls >> /tmp/ls.txt

任务调度的crond常驻命令

crond 是linux用来定期执行程序命令。当安装完成操作系统之后，默认便会启动此

任务调度命令。crond命令每分钟会定期检查是否有要执行的工作，如果有要执行的工作便会自动执行该工作。

14 6. crontab命令选项:

-u指定一个用户

-l列出某个用户的任务计划

-r删除某个用户的任务

-e编辑某个用户的任务

20 7. cron文件语法:

分	小时	日	月	星期	命令	
0-59	0-23	1-31	1-12	0-6	command	(取值范围,0表示周日一般一行对应一个任务)

记住几个特殊符号的含义:

“*”代表取值范围内的数字,

“/”代表“每”,

“-”代表从某个数字到某个数字,

“,”分开几个离散的数字

31 8. 任务调度设置文件的写法

可用crontab -e命令来编辑,编辑的是/var/spool/cron下对应用户的cron文件,也可以直接修改/etc/crontab文件

具体格式如下:

Minute	Hour	Day	Month	Dayofweek	command	
分钟	小时	天	月	天每星期	命令	

每个字段代表的含义如下:

Minute 每个小时的第几分钟执行该任务

Hour 每天的第几个小时执行该任务

Day 每月的第几天执行该任务

Month 每年的第几个月执行该任务

DayOfWeek 每周的第几天执行该任务

Command 指定要执行的程序

在这些字段里,除了“Command”是每次都必须指定的字段以外,其它字段皆为可选

字段,可视需要决定。对于不指定的字段,要用“*”来填补其位置。

```

46      举例如下：
47      5**** ls              指定每小时的第5分钟执行一次ls命令
48      30      5      *      *      *      ls              指定每天的 5:30 执行ls
命令
49      30      7      8      *      *      ls              指定每月8号的7: 30分执行
ls命令
50      30      5      8      6      *      ls              指定每年的6月8日5: 30执行
ls命令
51      30      6      *      *      0      ls              指定每星期日的6:30执行
ls命令[注：0表示星期天，1表示星期1，
52
53      以此类推，也可以用英文来表示，sun表示星期天，mon表示星期一等。]
54
55      30      3      10,20      *      *      ls              每月10号及20号的3: 30执行ls命令
[注：“，”用来连接多个不连续的时段]
56
57      25      8-11 *      *      *      ls              每天8-11点的第25分钟执行ls命令
[注：“-”用来连接连续的时段]
58
59      */15      *      *      *      *      ls              每15分钟执行一次ls命令 [即
每个小时的第0 15 30 45 60分钟执行ls命令 ]
60
61      30      6      */10      *      *      ls              每个月中，每隔10天6:30执行一次
ls命令[即每月的1、11、21、31日是6: 30执行一次ls 命令。 ]
62
63      每天7: 50以root 身份执行/etc/cron.daily目录中的所有可执行文件
64
65      50      7      *      *      *      root      run-parts
/etc/cron.daily      [ 注：run-parts参数表示，执行后面目录中的所有可执行文件。 ]
66
67  9. 新增调度任务
68      新增调度任务可用两种方法：
69      1)、在命令行输入：crontab -e 然后添加相应的任务，wq存盘退出。
70      2)、直接编辑/etc/crontab 文件，即vi /etc/crontab，添加相应的任务。
71
72  10. 查看调度任务
73      crontab -l //列出当前的所有调度任务
74      crontab -l -u jp //列出用户jp的所有调度任务
75
76  11. 删除任务调度工作
77      crontab -r //删除所有任务调度工作
78
79  12. 任务调度执行结果的转向
80      例1：每天5: 30执行ls命令，并把结果输出到/jp/test文件中
81      30 5 * * * ls >/jp/test 2>&1
82      注：2>&1 表示执行结果及错误信息。
83      编辑/etc/crontab 文件配置cron
84
85      cron服务每分钟不仅要读一次/var/spool/cron内的所有文件，还需要读一
次 /etc/crontab,因此我们配置这个文件也能运用cron服务做一些事情。用crontab配置是针对某个

```


用户的，而编辑/etc/crontab是针对系统的任务。此文件的文件格式是：

```
86
87     SHELL=/bin/bash
88
89     PATH=/sbin:/bin:/usr/sbin:/usr/bin
90
91     MAILTO=root //如果出现错误，或者有数据输出，数据作为邮件发给这个帐号
92
93     HOME=/ //使用者运行的路径,这里是根目录
94     # run-parts
95
96     01 * * * * root run-parts /etc/cron.hourly //每小时执行
97
98         /etc/cron.hourly内的脚本
99
100     02 4 * * * root run-parts /etc/cron.daily //每天执
行/etc/cron.daily内的脚本
101
102     22 4 * * 0 root run-parts /etc/cron.weekly //每星期执
行 /etc/cron.weekly内的脚本
103
104     42 4 1 * * root run-parts /etc/cron.monthly //每月去执
行/etc/cron.monthly内的脚本
105     大家注意“run-parts”这个参数了，如果去掉这个参数的话，后面就可以写要运行的某个脚本名，而
不是文件夹名了
106
107     例如：
108
109     1) 在命令行输入：crontab -e 然后添加相应的任务，wq存盘退出。
110
111     2)直接编辑/etc/crontab 文件，即vi /etc/crontab，添加相应的任务
112     11 2 21 10 * rm -rf /mnt/fb
```

4.2.3. 数据恢复mongorestore

mongodb使用 mongorestore 命令来恢复备份的数据，例：

```
./mongorestore -u admin -p pass --authenticationDatabase admin -d dbName --dir
mongodb_backup
```

注释：-d：数据库名，可以不是原来数据库名

--dir：要恢复的数据库文件存放目录

5. 日志监控

5.1. 异常日志

日记内容只能使用PrintUtil和LogUtil工具进行输出如果在Java代码中直接使用System.out.println或者其他第三方的日记类输出则不能在此界面中查看到。

1. 在API监控中心->平台日志->控制台日志, 刷新日志并查看。
2. 在tomcat应用服务器logs文件夹里查看

当API或程序运行异常时请先把此监控平台中的日志发送给我们, 如果此平台中没有则有可能需要提供tomcat原生的控制台日志给我们。

