

dtbus 部署文档

修订记录

版本	修订人	修订时间	备注
V0.1	金皓天	2017-10-16	创建文档（dtbus 安装配置）
V0.2	杨立平	2017-10-16	Dtbus 文档整理

dtbus 部署文档	1
一、 文档说明	3
1、 项目背景和需求	3
2、 文档目标	3
3、 参考文档	3
二、 部署准备	3
1、 系统架构及部署	3
1.1 mysql 数据库数据采集架构	3
1.2 应用 log 日志采集架构	4
1.3 部署组件	4
2、 java 环境确认	4
3、 接入 mysql 主库日志模式	4
4、 接入 mysql 从库日志模式	5
5、 接入 mysql 用户配置	5
6、 接入 log 用户配置	6
三、 canal server 部署	6
1、 canal server 下载解压	6
2、 canal server 配置监控主机	6
3、 canal server 启动	7
4、 canal server 确认	7
四、 dtbus server 部署	7
1、 dtbus mysql 部署	7
2、 dtbus server 部署源和目录	8
3、 dtbus server jenkins 部署	8
4、 dtbus server 部署接口确认	8
五、 dtbus ui 部署	9
1、 nginx 确认	9

2、	dtbus-ui 编译打包部署	9
3、	nginx 配置	9
4、	nginx 启动停止	10
5、	dtbus-ui 确认	10
六、	dtbus flume 部署	11
1、	flume 部署	11
2、	flume 部署测试	11
七、	dtbus 平台使用	11
1、	mysql 接入鉴权	11
2、	mysql 接入表同步	12
3、	mysql 接入表同步监测	12
4、	日志接入鉴权.....	13
5、	日志接入同步监控.....	13

一、文档说明

1、项目背景和需求

该项目是个人根据多年数据经验发起的数据实时采集处理系统，主要针对数据的采集，处理，以及数据推送的场景，根据各个开源系统整合而开发的管理系统。

该系统主要针对实时目标是实时数据处理。离线数据也可以走该系统。

2、文档目标

该文档针主要针对开发，运维人员，满足系统部署要求。也满足测试等人员对系统了解使用。

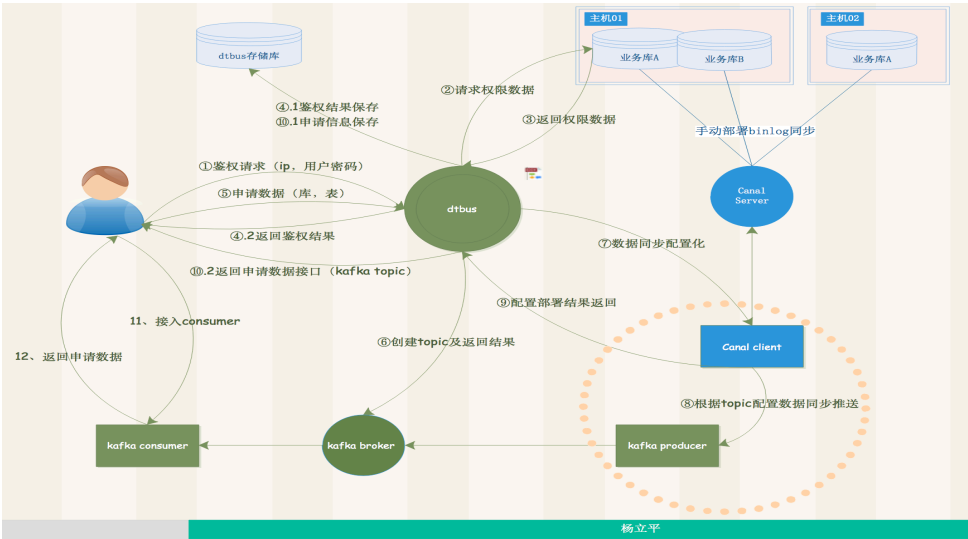
3、参考文档

二、部署准备

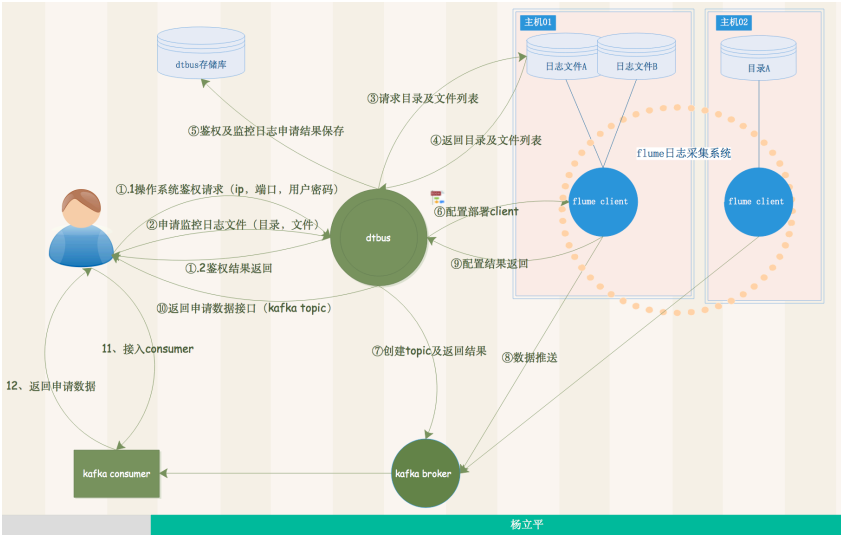
1、系统架构及部署

系统数据采集当前集中在 2 个地方：数据库 mysql 和日志文本文件

1.1 mysql 数据库数据采集架构



1.2 应用 log 日志采集架构



1.3 部署组件

mysql 数据数据收集架构如下，需要部署的组件如下：

组件名称	组件说明	部署路径	备注
canal server	模拟 mysql slave	/opt/dtbus/canal	
dtbus server	实时数据平台管理系统	/opt/dtbus/dtbus	
mysql	dtbus 存储库		
dtbus-ui	Dtbus 前端管理系统	/opt/dtbus/dtbus-ui	
flume 安装包	安装到 dtbus 指定到目录下	/opt/dtbus/flume	该 flume 是经过自己处理的安装包

2、 java 环境确认

确认服务器上 java 已经安装，并且在 1.7 以上版本

```
# java -version
```

```
java version "1.8.0_131"
```

```
Java(TM) SE Runtime Environment (build 1.8.0_131-b11)
```

```
Java HotSpot(TM) 64-Bit Server VM (build 25.131-b11, mixed mode)
```

3、 接入 mysql 主库日志模式

如果 mysql 只有主库，需要将主库配置文件(/etc/my.cnf)开启 binlog 日志，在 mysql 下添加如下配置：

```
# vi /etc/my.cnf
```

```
[mysql]
```

```
log-bin=mysql-bin #添加这一行就 ok
```

```
binlog-format=ROW #选择 row 模式
```

```
server_id=1 #配置 mysql replaction 需要定义，不能和 canal 的 slaveId 重复
```

修改配置文件后重新启动数据库即可，然后先确认数据库日志模式，如下

```
mysql> show global variables like '%binlog_format%';
```

```
+-----+-----+
| Variable_name | Value |
+-----+-----+
| binlog_format | ROW   |
+-----+-----+
1 row in set (0.00 sec)
```

4、 接入 mysql 从库日志模式

部署数据库实时采集时，需要首先确认实时采集到数据库到 binlog 模式

如果 mysql 已经存在主从库架构，这个需要将主库配置文件(/etc/my.cnf)只需要修改日志模式，在 mysql 下添加如下配置：

```
# vi /etc/my.cnf
```

```
[mysql]
```

```
log-bin=mysql-bin #添加这一行就 ok
```

```
binlog-format=ROW #选择 row 模式
```

```
server_id=1 #配置 mysql replaction 需要定义，不能和 canal 的 slaveId 重复
```

修改配置文件后重新启动数据库即可，然后先确认数据库日志模式，如下

```
mysql> show global variables like '%binlog_format%';
```

```
+-----+-----+
| Variable_name | Value |
+-----+-----+
| binlog_format | ROW   |
+-----+-----+
1 row in set (0.00 sec)
```

5、 接入 mysql 用户配置

不管接入是主库还是主从库，canal 的原理是模拟自己为 mysql slave，所以都需要在 mysql 主库给 mysql slave 建立账户并且赋予相关权限如下：

```
mysql> CREATE USER 'canal' IDENTIFIED BY 'canal';
```

```
mysql> grant Replication client on *.* to 'canal'@localhost;
```

```
mysql> FLUSH PRIVILEGES;
```

用此账号密码登陆 mysql 确认出现类似如下内容则为权限设置成功：

```
mysql> show master status;
```

```

+-----+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+-----+-----+-----+-----+-----+
| mysql-bin.000007 | 133301909 |              |                  |                   |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

```

6、 接入 log 用户配置

由于接入实时日志采集需要在客户机安装 flume-ng agent 因此需要先确认客户机器上是否已经配置 java 环境,如果没有则进行安装,

```
# java -version
```

```
java version "1.8.0_131"
```

```
Java(TM) SE Runtime Environment (build 1.8.0_131-b11)
```

```
Java HotSpot(TM) 64-Bit Server VM (build 25.131-b11, mixed mode)
```

注意：如果使用拉取日志的用户不是 root，则需要手动启动 flume.

三、 canal server 部署

1、 canal server 下载解压

canal 是阿里巴巴的开源数据同步组件，这里需要从以下网址下载 canal server：
<https://github.com/alibaba/canal/releases>

这里使用的版本是 [v1.0.24](#)，下载后解压到目录/opt/dtbus/canal 下

```
# mkdir -p /opt/dtbus/canal
```

```
# tar -zxvf canal.deployer-1.0.24.tar.gz -C /opt/dtbus/canal
```

```
# cd /opt/dtbus/canal/
```

```
# ls
```

```
bin  conf  lib  logs
```

2、 canal server 配置监控主机

根据自带的配置文件目录复制一份为需要监控到主机配置文件目录，如下：

```
# cd conf/
```

```
# cp -r example 172.18.2.7_3306
```

注意：其中目录命名 172.18.2.7_3306 为所需要监控的数据库 ip 和 port,以_相连接

```
# vi 172.18.2.7_3306/instance.properties
```

```
#####
## mysql serverId
canal.instance.mysql.slaveId = 10007

# position info
canal.instance.master.address = 172.18.2.7:3306
canal.instance.master.journal.name =
canal.instance.master.position =
canal.instance.master.timestamp =
```

注意：这里是 slaveid 不能重复，命名采用 10000+ip 尾数区分。

3、 canal server 启动

启动 canal server

```
# cd /opt/dtbus/canal/
# sh bin/startup.sh
```

4、 canal server 确认

确认进程已经启动：

```
# netstat -ntlp | grep 11111
tcp        0      0 0.0.0.0:11111          0.0.0.0:*              LISTEN
9358/java
```

查看日志确认已经成功启动：

```
# cat logs/172.18.2.7_3306/172.18.2.7_3306.log
。。。。。。省略 N 行
2017-10-16      20:57:53.527      [canal-instance-scan-0]      INFO
c.a.otter.canal.instance.core.AbstractCanalInstance - start successful....
2017-10-16 20:57:53.584 [destination = 172.18.2.7_3306 , address = /172.18.2.7:3306 ,
EventParser] WARN  c.a.otter.canal.parse.inbound.mysql.MysqlEventParser - prepare to find
start position just show master status
```

出现 successful 则为启动成功

四、 dtbus server 部署

1、 dtbus mysql 部署

首先需要执行 dtbus 的存储库和表的创建语句，在 dtbus 存储库里执行如下语句：

```
#mysql -u root -p
mysql> source dtbus.sql
```

2、 dtbus server 部署源和目录

从 github 上下载最新的 dtbus 开发包: <http://git.jc/SarbanesD/dtbus.git>

创建 dtbus 安装目录:

```
# mkdir -p /opt/dtbus/server
```

3、 dtbus server jenkins 部署

执行 jenkins 自动打包发布, 并且运行。

```
# nohop java -jar dtbus-server-1.0.jar &
```

部署需要确认日志正常如下:

```
# tail -f nohop.out
```

4、 dtbus server 部署接口确认

查看端口是否启动

```
# netstat -ntlp | grep 8080
```

```
tcp6          0      0 :::8080          :::*              LISTEN
12770/java
```

在日志目录下可以查看运行日志, 确保正常如下:

```
# pwd
```

```
/opt/dtbus/server/logs/server
```

```
5402 22:42:18.765 [main] INFO    o.s.m.s.b.SimpleBrokerMessageHandler - Started.
5414 22:42:18.777 [main] INFO    o.a.coyote.http11.Http11NioProtocol - Initializing ProtocolHandler ["http-nio-8080"]
5435 22:42:18.798 [main] INFO    o.a.coyote.http11.Http11NioProtocol - Starting ProtocolHandler ["http-nio-8080"]
5436 22:42:18.799 [main] INFO    o.a.tomcat.util.net.NioSelectorPool - Using a shared selector for servlet write/read
5466 22:42:18.829 [main] INFO    o.s.b.c.e.t.TomcatEmbeddedServletContainer - Tomcat started on port(s): 8080 (http)
5473 22:42:18.836 [main] INFO    com.hawker.Application - Started Application in 5.091 seconds (JVM running for 5.714)
5474 22:42:18.837 [main] INFO    com.hawker.Application - SpringBoot server stated on port: 8080
5496    22:42:18.859    [org.springframework.kafka.KafkaListenerEndpointContainer#0-2-kafka-consumer-1]    INFO
o.a.k.c.c.i.AbstractCo
ordinator - Discovered coordinator dx02-bigdata-prod:9092 (id: 2147483584 rack: null) for group helloworld.
```


五、dtbus ui 部署

1、nginx 确认

确认机器上已经安装了 nginx,如果没有则需要安装

```
# nginx -v
```

```
nginx version: nginx/1.12.1
```

2、dtbus-ui 编译打包部署

从 gitlab 上下载最新的后台管理系统代码: <http://git.ic/SarbanesD/dtbus-ui.git>
用 jenkins 打包部署, 命令为: npm run build

编译打包部署目录:

```
# mkdir -p /opt/dtbus/ui
```

```
# tar -zxvf build.gz
```

```
# ls
```

```
build.gz  build
```

3、nginx 配置

增加 dtbus 的 nginx 配置文件如下:

```
# cd /etc/nginx/conf.d/
```

```
# vi dtbus.conf
```

```
server {  
    listen      3000;  
    server_name 172.18.2.4;  
  
    location / {  
        root    /root/dtbus/static;  
        index   index.html index.htm;  
        try_files $uri $uri/ index.html;  
    }  
  
    location /api {  
        proxy_pass http://127.0.0.1:8080/dtapi/v1;  
    }  
  
    location /ws {
```

```
        proxy_pass    http://127.0.0.1:8080/ws;
    }

    #error_page  404                /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }
}
```

4、 nginx 启动停止

配置后，启动 nginx

```
# nginx
```

如果已经启动则重新加载配置文件：

```
# ps -ef | grep nginx
```

```
root      30690      1   0 09:49 ?                00:00:00 nginx: master process /usr/sbin/nginx -c
/etc/nginx/nginx.conf
```

```
# nginx -s reload
```

5、 dtbus-ui 确认

确认 nginx 端口启动

```
# netstat -ntlp | grep 3000
```

```
tcp        0      0 0.0.0.0:3000          0.0.0.0:*             LISTEN
30690/nginx: master
```

查看 web 页面是否正常访问：<http://172.18.2.4:3000/>

不安全 | 172.18.2.4:3000/login

* 用户名

请输入用户名

* 密码

请输入密码

登录

重置

六、dtbus flume 部署

1、 flume 部署

首先创建部署目录

```
# mkdir -p /opt/dtbus/flume
```

上传 flume 的 tar 包 apache-flume-1.7.0-bin.tgz 到安装目录即可。

2、 flume 部署测试

由于 flume 是部署到客户端机器，因此在 dtbus server 端只是将包上传，如果需要部署到其它客户端，则会自动化完成部署。

七、dtbus 平台使用

1、 mysql 接入鉴权

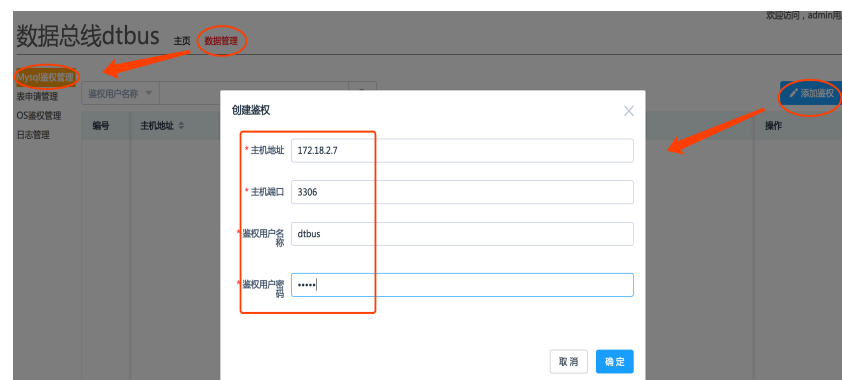
访问 <http://192.168.43.21:3000> 初始账号密码为 admin/jcgroup123



* 用户名

* 密码

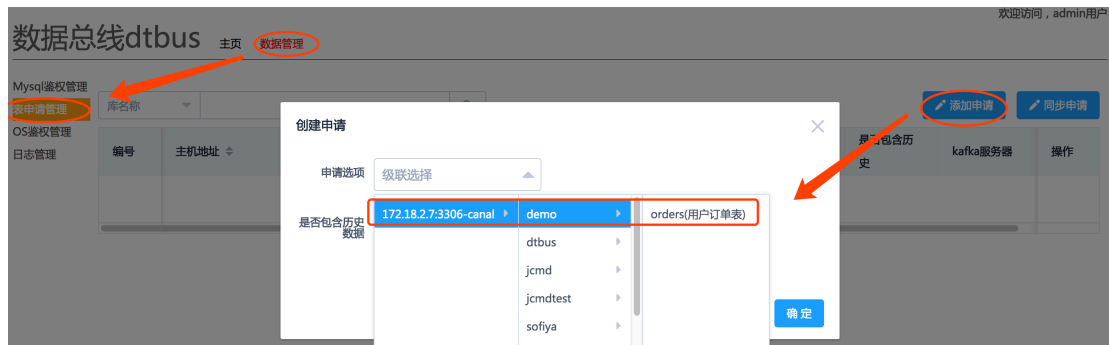
添加鉴权，输入要监控的数据库，但是再次之前要监控的数据库需要先赋予系统 binglog 监控的权限，详细咨询管理员。





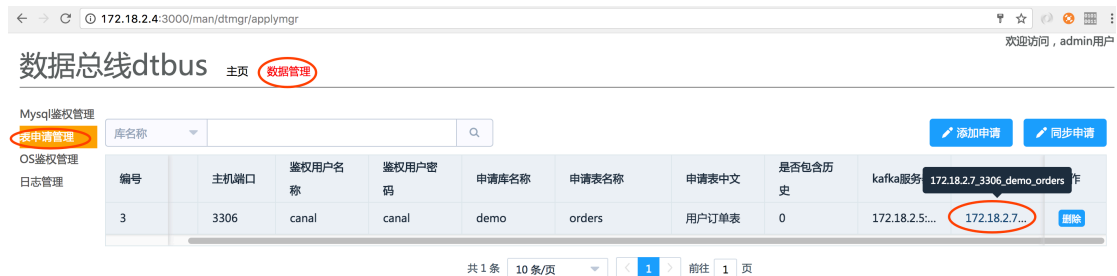
2、mysql 接入表同步

选择希望添加监听的表



3、mysql 接入表同步监测

打开申请同步当表当监控超连接，打开监控新窗口：

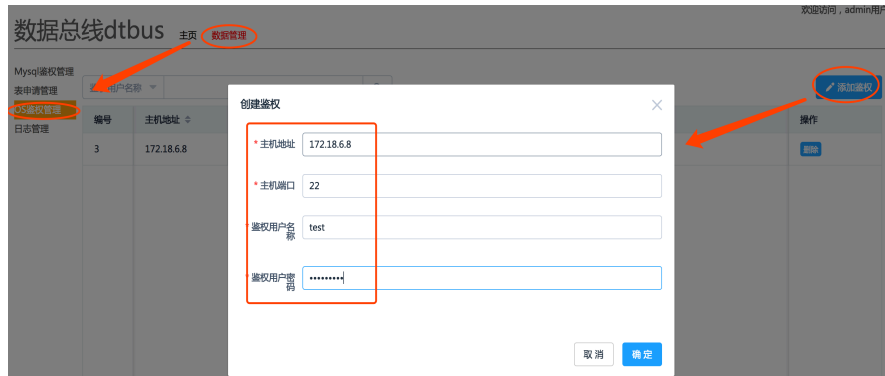


当同步当表订单表发生变化，监控窗口会显示数据变化：



4、 日志接入鉴权

根据服务器账号密码进行鉴权,注意, 整理如果采用非 root 账户, 则需要一些手工处理, 建议用 root 账户, 后期会逐步完善非 root 当日志接入。



5、 日志接入同步监控

选择需要监控主机的日志文件

