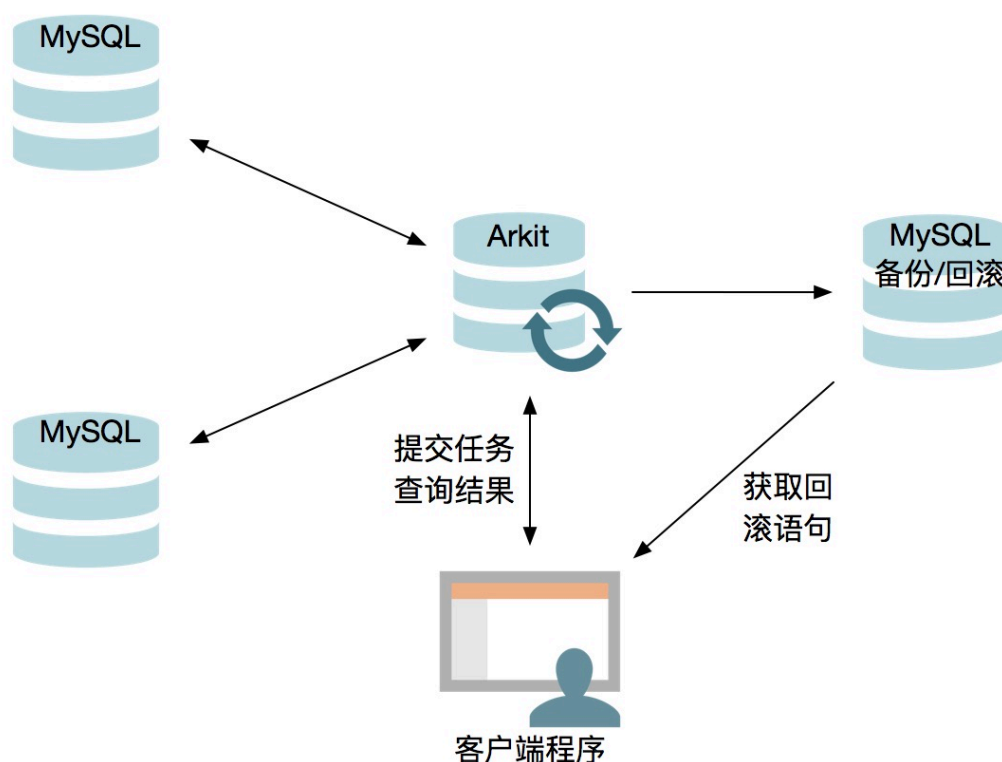


# 1. Arkcontrol-SQL 管控功能介绍

Arkcontrol SQL 管控功能是基于极数云舟自主研发的 Arkit(MySQL 自动审核和执行系统)，通过 Web 平台进行流程定制化，完成日常 SQL 工单记录，自动化审核、执行、回滚等操作。其 SQL 管控的 SQL 审核 100%兼容 MySQL 语法，同时它也可以自动执行审核成功的语句。在执行 SQL 过程中，SQL 管控可以生成 DML 的回滚语句(MySQL 的 `binlog_format` 必须为 `ROW` 模式)，用来实现数据库的闪回功能。对于审核和执行的结果，SQL 管控以结果集的方式展现出来，方便后续审计。

使用 SQL 管控实现操作数据库的自动化运维，可以极大提升 DBA 的工作效率，标准化 SQL 审核，降低线上执行 SQL 出错的概率，并有机会实现数据库操作的回滚，并对操作数据库的行为提供统计和审计功能。



## 2. SQL 管控-相关资源包安装

Arkit 是一款 MySQL 插件，所以安装起来非常简单，只需要 arkit.so，然后将其放置在 MySQL 的参数 plugin\_dir 所指的目录中，然后通过 install plugin 的方式进行安装。

具体安装步骤如下：

1. 准备相关信息：假设 Arkcontrol 后台数据库配置如下：

```
端口为：3306  
账号为：root  
密码：arkcontrol_root  
plugin_dir: /opt/ark/arkdb/3306/private/lib/plugin/
```

2. 将相关资源包放置的相关位置：登录到 Arkcontrol 的后台配置数据库，通过 show variables like '%plugin\_dir%' 的参数配置。从 Arkcontrol 的资源中，利用 cp 命令，将 arkit.so 和 arkit.key 拷贝到 plugin\_dir 位置。其中 arkit 的资源包在 arkcontrol 的 package 目录下 arkit.tar.gz 中。
3. 安装插件：登录到 arkcontrol 的数据库，执行以下命令：

```
install plugin arkit soname 'arkit.so';  
install plugin arkit_order_queue soname 'arkit.so';  
install plugin arkit_osc_status soname 'arkit.so';  
install plugin arkit_processlist soname 'arkit.so';
```

安装完成后，可以通过 show variables like '%arkit%'; 命令查看是否安装成功，如果安装成功，可以查看 arkit 执行的配置参数。

4. 初始化配置参数：arkit 运行需要依赖相关的配置参数，在 Arkcontrol 的配置库中的 MySQL 的配置文件(例如：arkdb/3306/etc/my.cnf)中新增关于 arkit 的配置参数。具体参数如下：

```
# ARKIT PLUGIN #  
arkit_license_file = /opt/ark/arkdb/3306/private/lib/plugin/arkit.key  
arkit_remote_backup_host = 127.0.0.1  
arkit_remote_backup_port = 3306
```

```
arkit_remote_backup_user = root
arkit_remote_backup_password = arkcontrol_root
arkit_backup_dbname_suffix_cut = .mysql.rds.aliyuncs.com
arkit_metadata_host = 127.0.0.1
arkit_metadata_port = 3306
arkit_metadata_user = root
arkit_metadata_password = arkcontrol_root
```

5. 重启 arkcontrol 的数据库：使得 Arkit 的新配置生效。

## 3. SQL 管控-相关设置

### 3.1 业务线设置

业务线划分，主要用于满足后期的功能搜索过滤、报表功能。使得 SQL 工单能够更加清晰的分类。

在平台中，可以通过点击【SQL 管控】->【业务线设置】->【业务线添加】，完成对业务线的添加，与其他操作。

SQL 管控 > 业务线设置 > 添加

基本信息

业务线名称:

业务线名称

业务线描述:

业务线描述

提交

### 3.2 数据源设置

数据源是线上使用的数据库的 IP、PORT、USER、PASSWORD 总称，用于满足后续的 SQL 审核、SQL 执行时的工单提交时依赖的线上数据库资源，提前配置，避免每次提交工单时候的手工输入。当然平台中，支持手动输入数据库信息。

在平台中，可以通过点击【SQL 管控】->【数据源设置】->【数据源添加】，完成对数据源的添加，与其他操作。

SQL 管控 > 数据源设置 > 数据源添加

数据源信息

数据源名称:

数据库地址:

数据库端口:

数据库账号:

数据库密码:

备注:

预检查

### 3.3 流程设置

流程是用于工单审核与执行过程，都有哪些类型的人员参与审核与执行，流程可以灵活配置，可以根据每一个业务线配置符合自己要的流程。

注意：

1. 流程中的最多只有三个阶段：1. 提交工单、2. 审核、3. 执行。
2. 每个流程请以【执行】阶段为流程中最后一个阶段，不然失去 SQL 执行的意义。
3. 【执行】阶段，推荐让 DBA 人员类型来完成执行操作，确保数据库安全。

在平台中，可以通过点击【SQL 管控】->【流程设置】->【流程添加】，完成对流程的添加，与其他操作。

基本信息

流程名称:

请输入流程名称，最大长度不超过10个字符

流程描述:

流程描述

流程信息

阶段名称

阶段名称 提交工单

处理人员: ☐ 测试  
☐ 数据库管理员  
☐ 开发

保存

## 3.4 参数设置

参数设置中包含三个部分：执行选项、审核参数、OSC 参数。

其中：

1. 执行选项：用于控制 SQL 执行过程的，相关控制机制，例如：是否备份、是否忽略警告、执行的间隔时间等。为系统全局的默认值，修改后，会影响以后所有功能的默认值。
2. 审核参数：控制 Arkit 审核与执行过程中的相关规则，例如：`arkit_check_identifier`、`arkit_enable_autoincrement_unsigned` 等。
3. OSC 参数：SQL 管控默认的改表使用 `percona-online-schema-change` 的工具，协助该表。该参数用于控制 OSC 相关参数。为系统全局的默认值，修改后，会影响以后所有功能的默认值。

在平台中，可以通过点击【SQL 管控】->【参数设置】，完成对相关参数的修改。

## 参数列表

| 参数名称           | 参数值 | 参数可修改值   | 参数说明                 | 操作者         | 更新时间                | 操作                 |
|----------------|-----|----------|----------------------|-------------|---------------------|--------------------|
| sleep_interval | 2   | 0-1000   | SQL语句执行间隔时间,单位是ms    | ark_control | 2018-11-23 06:38:50 | <a href="#">修改</a> |
| force_exec     | OFF | [ON OFF] | 执行过程中遇到错误是否继续执行      | ark_control | 2018-11-02 04:18:24 | <a href="#">修改</a> |
| ignore_warn    | OFF | [ON OFF] | 执行过程中是否忽略警告信息        | ark_control | 2018-11-23 06:53:55 | <a href="#">修改</a> |
| backup         | ON  | [ON OFF] | 是否打开备份,备份SQL执行后的回滚语句 | ark_control | 2018-11-23 06:55:05 | <a href="#">修改</a> |

## 4. SQL 管控-SQL 语法检查

SQL 语法检查主要检查 SQL 语句是否存在语法问题，可以帮助开发，在开发 SQL 时，测试 SQL 语句使用。它不会连接线上数据库，只会根据 Arkit 自身所在的数据库进行。有效的提升了开发的效率。

在平台中可以通过点击【SQL 管控】->【SQL 语法检查】，完成对相关 SQL 语句进行语法检查。

## 基本信息

SQL语句:

输入待检查的SQL语句，多条SQL请以';'分隔。

填写待检查的SQL语句，  
多条SQL请以';'分隔。多  
行请换行

语法检查

← 点击【语法检查】完成对输入的SQL语句进行检查。

点击【语法检查】后，在页面下方会自动将检查结果展示出来。开发可以根据检查结果来修改 SQL 语句，直到 SQL 语句无语法错误为止。提升开发效率。

基本信息

SQL语句:

```
use arkcontrol_db;
select * from arkcontrol_table;
```

语法检查

检查结果

展示检查结果

| ID | 错误代码    | 错误信息 | SQL语句                          |
|----|---------|------|--------------------------------|
| 1  | Success | None | use arkcontrol_db              |
| 2  | Success | None | select * from arkcontrol_table |

共有2条, 每页显示: 500条

1

## 5. SQL 管控-SQL 审核

SQL 审核主要检查 SQL 语句检查 SQL 语句是否存在语法问题 ,并且会根据数据源的配置连接到相关数据中，检查语义是否存在问题。在审核的过程中，还会获取每一条 SQL 语句预估的影响行数，判断用户判断 SQL 是否存在风险。

在平台中可以通过点击【SQL 管控】->【SQL 审核】->【审核提交】，完成对相关 SQL 语句进行审核操作。

数据源类型:

从配置中选择

手动填写

选择数据源信息，此处根据【数据源配置】的配置信息中获取

数据源配置:

sql\_check\_demo

\*所选数据源信息: IP:10.0.0.196 port:3306

SQL语句:

use arkcontrol\_db;  
create table arkcontrol\_table(  
id int not null auto\_increment,  
host\_name varchar(50) not null default "",  
primary key(id)  
) engine=innodb,charset=utf8mb4;

输入待审核的SQL语句

填写待检查的SQL语句。  
多条SQL请以';'分隔。多  
行请换行

审核

点击【审核】，完成对SQL语句审核动作

审核信息

展示审核结果

错误代码:

包含全部

Success

Error

Warning

Error&Warning

| ID | 错误代码    | 错误信息 | SQL语句   | 影响行数 |
|----|---------|------|---|------|
| 1  | Success | None | use arkcontrol_db   | 0    |
| 2  | Success | None | create table arkcontrol_table( id int not null auto_increment, host_name varchar(50) not null default "", primary key(id) ) engine=innodb,charset=utf8mb4 | 0    |

## 6. SQL 管控-SQL 执行

SQL 执行是 SQL 管控中最核心功能之一。在此处用户可以提交工单，然后嵌入审核流程，相关审核人员参与审核，再配合执行，做到真正的在线上数据库进行执行。并且在执行时，可以通过控制执行选项、执行的相关参数，控制整个执行过程，并且可以生产回滚语句，保证线上数据库安全。

关于工单执行的相关说明：

1. 工单状态：工单主要包括【审核状态】、【执行状态】，其中【审核状态】包含：未审核、审核中、审核通过、审核不通过。【执行状态】包含：未执行、执行中、执行成功、执行成功(含警告)、执行成功(含错误)、执行失败、生成重做数据。审核状态用于判断工单审核是否结束，如果审核都通过后，工单会进行一种称为：待执行的状态。用户点击执行后，工单开始执行，相应的执行状态也会随之发生变化。
2. 审核流程：在申请工单的时候，选择配合该工单的审核路程，并指定参与流程的相关人员，完成工单的人员分配。
3. 工单审核操作：工单提交后，审核人员根据工单的情况进行审核，可以选择【通过】与【不通过】两种状态，并且配合审核说明。
4. 工单撤销：当工单的审核状态处于【未审核】、【审核中】时，用户可



以撤销工单，使得工单不进行后续操作，直接结束工单。

5. 工单修改：当工单的审核状态处于【未审核】、【审核中】和【不通过】时，用户可以修改自己提交的工单。再次进入审核流程。
6. 工单拆分：在 SQL 语句执行前，平台会自动将 SQL 语句拆分，使得相关的 DML 与 DDL 分离。因为 DDL 执行需要接入 OSC 的工具。
7. 工单出错处理：工单在执行的过程，避免不了出错，比如主键冲突、比如网络中断等，用户可以根据出错信息进行相应的操作。主要包括：  
【生成未执行重做数据】和【生成未执行重做数据(含)】。
8. 工单执行进度：对于 DDL 类型的工单，点击【自动执行】后，工单开始进行执行，在执行中，用户可以查看工单的执行进度，便于展示工单的实时状态。
9. OSC 终止：对于 DDL 类型的工单，点击【查看执行进度】后，查看具体的某一条 DDL 语句执行状态，如果发现问题，可以通过点击【OSC 终止】按钮，完成 OSC 的终止，避免产生对数据库的影响。
10. 工单备份：在工单执行前，可以选择是否【备份】，来完成 SQL 执行后，将影响的数据生成回滚的 SQL 语句存储下来。前提：MySQL 的 binlog\_format 必须为 ROW 模式。
11. 执行结果：工单执行后，都有相关的执行状态与执行结果。用户可以轻松的查看每一个工单的执行结果信息。

在平台中可以通过点击【SQL 管控】->【SQL 执行】->【工单申请】，完成对相关 SQL 语句的工单操作。

SQL 管控 > SQL 执行 > 工单申请

#### 基本信息

|  |  |
|--|--|
| 数据源类型:                                       | <input checked="" type="button" value="从配置中选择"/> <input type="button" value="手动填写"/> |
| 数据源配置:                                       | <input type="text" value="请选择"/>   |
| 提交方式:  | <input checked="" type="button" value="文本"/> <input type="button" value="附件"/>       |
| SQL 语句:                                      | <div></div>  |
| <input checked="" type="button" value="审核"/> |  |