

第4天 自动化运维利器 Ansible-扩展ansible

一、使用插件

1 回调插件介绍

1.1 修改默认的回调插件

同时只能有一个回调插件作为主要的管理者，用于输出到屏幕。

如果想替换，应该在这个插件中修改 `CALLBACK_TYPE = stdout`,

之后在 `ansible.cfg` 中配置 `stdout` 插件。

```
1 [defaults]
2 stdout_callback = json # 以 JSON 的格式输出结果
```

或使用自定义的回调：

```
1 [defaults]
2 stdout_callback = mycallback
```

默认情况下这仅对 `playbook` 生效，如果想让 `ad-hoc` 方式生效应该在 `ansible.cfg` 文件中做如下设置：


```
21         "action": "debug",
22         "ansible_distribution":
"VARIABLE IS NOT DEFINED!",
23         "changed": false
24     }
25 },
26     "task": {
27         "duration": {
28             "end": "2020-04-
23T02:32:44.163630Z",
29             "start": "2020-04-
23T02:32:44.137440Z"
30         },
31         "id": "0242ac12-0002-b0c7-
074b-000000000000f",
32         "name": "debug"
33     }
34 }
35 ]
36 }
37 ],
38 "stats": {
39     "172.18.0.3": {
40         "changed": 0,
41         "failures": 0,
42         "ignored": 0,
43         "ok": 1,
44         "rescued": 0,
45         "skipped": 0,
46         "unreachable": 0
47     }
48 }
49 }
50
```

1.2 启用其他内置的回调插件

大部分情况下，无论是内置的回调插件还是自定义的回调插件，都需要在 `ansible.cfg` 中添加到白名单中，从而才能启用。

```
1 | callback_whitelist = timer, mail, profile_roles,  
   | custom_callback
```

- `timer` 这个回调插件可以计算整个 playbook 的运行时间
- `mail` 这个回调插件可以实现发送邮件的功能
- `profile_roles` 这个插件是在执行中提添加用时时间
- `custom_callback` 是自定义的插件，稍后会讲

1.3 获取帮助

`ansible-doc -t callback -l` 可以查看当前可用的回调插件列表

`ansible-doc -t callback <callback plugins name>` 可查看具体回调插件的帮助文档

比如：

```
1 | [root@qfedu ~]# ansible-doc -t callback timer  
2 | > TIMER      (/usr/lib/python2.7/site-  
   | packages/ansible/plugins/callback/timer.py)  
3 |
```

```
4         This callback just adds total play duration
to the play stats.
5
6     * This module is maintained by The Ansible
Community
7 REQUIREMENTS:  whitelist in configuration
8
9 CALLBACK_TYPE: aggregate
10
11     METADATA:
12         status:
13             - preview
14         supported_by: community
15
```

2 回调插件类型

回调插件类型在回调插件类中定义：

```
1 class CallbackModule(CallbackBase):
2     CALLBACK_TYPE = 'notification'
```

不同的回调类型对于 playbook 的输出有不一样的效果

- `stdout` 标准输出类型，用在回调的主管理者

- `aggregate` 聚合类型，把此类型插件处理的结果和 `stdout` 类型插件合并一起输出到标准输出。比如：`timer`，`profile_tasks` 等。
- `notification` 通知类型，不参与标准输出，也不影响标准输出插件的正常输出，只是会把执行 `playbook` 的返回值写的指定的媒介中。

比如：`log_plays`，`mail`。假如自定义把执行 `playbook` 的结果输出到数据库中就可以使用此类型。

查看所有默认的查看类型

```
1 [root@qfedu ~]# grep 'CALLBACK_TYPE =.*'
  /usr/lib/python2.7/site-
  packages/ansible/plugins/callback/*.py |cut -d: -f 2
  |sort -u
2     CALLBACK_TYPE = 'aggregate'
3     CALLBACK_TYPE = 'notification'
4     CALLBACK_TYPE = 'stdout'
```

3 把返回结果输出到日志中

内置的回调插件 `log_plays` 会将 `playbook` 的返回信息输出到 `/var/log/ansible/hosts` 目录中。

可以在 `ansible.cfg` 中配置指定的目录，使用 `log_folder`

比如，把日志存到 `/tmp/ansible/hosts/` 目录下

在 `ansible.cfg` 文件的最后添加如下配置

配置日志存放的目录

```
1 [callback_log_plays]
2 log_folder=/tmp/ansible/hosts/
```

配置到白名单

`ansible.cfg`

```
1 callback_whitelist = log_plays
```

Inventory

```
1 [root@qfedu.com ~]# cat hosts
2 [dbservers]
3 172.18.0.3
4
5 [webservers]
6 172.18.0.4
7 172.18.0.5
8
9 [allservers:children]
10 dbservers
11 webservers
```

playbook

remoteDate.yml

```
1 - hosts: all
2   gather_facts: no
3   tasks:
4     - name: test
5       shell: date +"%F %T"
```

执行 playbook

```
1 [root@qfedu.com ~]# ansible-playbook -i hosts
  remoteDate.yml
2
3 PLAY [all]
  *****
  *****
4
5 TASK [test]
  *****
  *****
6 fatal: [172.18.0.5]: UNREACHABLE! => {"changed":
  false, "msg": "Failed to connect to the host via
  ssh: ssh: connect to host 172.18.0.5 port 22:
  Connection refused", "unreachable": true}
7 changed: [172.18.0.3]
8 changed: [172.18.0.4]
9
10 PLAY RECAP
  *****
  *****
11 172.18.0.3                : ok=1    changed=1
    unreachable=0    failed=0    skipped=0    rescued=0
    ignored=0
```



```
12 | 172.18.0.4 : ok=1 changed=1
    unreachable=0 failed=0 skipped=0 rescued=0
    ignored=0
13 | 172.18.0.5 : ok=0 changed=0
    unreachable=1 failed=0 skipped=0 rescued=0
    ignored=0
14
15 | Playbook run took 0 days, 0 hours, 0 minutes, 0
    seconds
16
```

查看输出结果

干锋云计算学院

```

1 [root@qfedu.com ~]# ls /tmp/ansible/hosts/
2 172.18.0.3 172.18.0.4 172.18.0.5
3 [root@qfedu.com ~]# cat /tmp/ansible/hosts/172.18.0.3
4 Apr 24 2020 06:43:57 - OK - {"module_args": {"data":
  "pong"}} => {"changed": false, "ping": "pong",
  "_ansible_no_log": false, "ansible_facts":
  {"discovered_interpreter_python": "/usr/bin/python"}}
5
6 Apr 24 2020 06:45:11 - OK - {"module_args": {"warn":
  true, "executable": null, "_uses_shell": true,
  "strip_empty_ends": true, "_raw_params": "date +\"%F
  %T\"", "removes": null, "argv": null, "creates":
  null, "chdir": null, "stdin_add_newline": true,
  "stdin": null}} => {"stderr_lines": [], "cmd": "date
  +\"%F %T\"", "end": "2020-04-24 06:45:11.110025",
  "_ansible_no_log": false, "stdout": "2020-04-24
  06:45:11", "changed": true, "rc": 0, "start": "2020-
  04-24 06:45:10.878037", "stderr": "", "delta":
  "0:00:00.231988", "stdout_lines": ["2020-04-24
  06:45:11"], "ansible_facts":
  {"discovered_interpreter_python": "/usr/bin/python"}}
7

```

二、开发自定义插件

1 log_plays 插件源码分析

```

1 # (C) 2012, Michael DeHaan,
  <michael.dehaan@gmail.com>
2 # (c) 2017 Ansible Project

```

```
3 # GNU General Public License v3.0+ (see COPYING or
  # https://www.gnu.org/licenses/gpl-3.0.txt)
4
5 from __future__ import (absolute_import, division,
  print_function)
6 __metaclass__ = type
7
8 DOCUMENTATION = '''
9     callback: log_plays
10    type: notification
11    short_description: write playbook output to
  log file
12    version_added: historical
13    description:
14        - This callback writes playbook output to a
  file per host in the `/var/log/ansible/hosts`
  directory
15    requirements:
16        - Whitelist in configuration
17        - A writeable /var/log/ansible/hosts
  directory by the user executing Ansible on the
  controller
18    options:
19        log_folder:
20            version_added: '2.9'
21            default: /var/log/ansible/hosts
22            description: The folder where log files
  will be created.
23        env:
24            - name: ANSIBLE_LOG_FOLDER
25        ini:
26            - section: callback_log_plays
27              key: log_folder
28    '''
29
```

```
30 import os
31 import time
32 import json
33
34 from ansible.utils.path import makedirs_safe
35 from ansible.module_utils._text import to_bytes
36 from
    ansible.module_utils.common._collections_compat
    import MutableMapping
37 from ansible.parsing.ajson import
    AnsibleJSONEncoder
38 from ansible.plugins.callback import CallbackBase
39
40
41 # NOTE: in Ansible 1.2 or later general logging is
    available without
42 # this plugin, just set ANSIBLE_LOG_PATH as an
    environment variable
43 # or log_path in the DEFAULTS section of your
    ansible configuration
44 # file. This callback is an example of per hosts
    logging for those
45 # that want it.
46
47
48 class CallbackModule(CallbackBase):
49     """
50     logs playbook results, per host, in
    /var/log/ansible/hosts
51     """
52     CALLBACK_VERSION = 2.0
53     CALLBACK_TYPE = 'notification'
54     CALLBACK_NAME = 'log_plays'
55     CALLBACK_NEEDS_WHITELIST = True
56
```

```

57     TIME_FORMAT = "%b %d %Y %H:%M:%S"
58     MSG_FORMAT = "%(now)s - %(category)s - %(data)s\n\n"
59
60     def __init__(self):
61
62         super(CallbackModule, self).__init__()
63
64     def set_options(self, task_keys=None,
var_options=None, direct=None):
65         super(CallbackModule,
self).set_options(task_keys=task_keys,
var_options=var_options, direct=direct)
66
67         self.log_folder =
self.get_option("log_folder")
68
69         if not os.path.exists(self.log_folder):
70             makedirs_safe(self.log_folder)
71
72     def log(self, host, category, data):
73         if isinstance(data, MutableMapping):
74             if '_ansible_verbose_override' in
data:
75                 # avoid logging extraneous data
76                 data = 'omitted'
77             else:
78                 data = data.copy()
79                 invocation =
data.pop('invocation', None)
80                 data = json.dumps(data,
cls=AnsibleJSONEncoder)
81                 if invocation is not None:
82                     data = json.dumps(invocation)
+ " => %s " % data

```

```
83
84     path = os.path.join(self.log_folder, host)
85     now = time.strftime(self.TIME_FORMAT,
time.localtime())
86
87     msg = to_bytes(self.MSG_FORMAT %
dict(now=now, category=category, data=data))
88     with open(path, "ab") as fd:
89         fd.write(msg)
90
91     def runner_on_failed(self, host, res,
ignore_errors=False):
92         self.log(host, 'FAILED', res)
93
94     def runner_on_ok(self, host, res):
95         self.log(host, 'OK', res)
96
97     def runner_on_skipped(self, host, item=None):
98         self.log(host, 'SKIPPED', '...')
99
100    def runner_on_unreachable(self, host, res):
101        self.log(host, 'UNREACHABLE', res)
102
103    def runner_on_async_failed(self, host, res,
jid):
104        self.log(host, 'ASYNC_FAILED', res)
105
106    def playbook_on_import_for_host(self, host,
imported_file):
107        self.log(host, 'IMPORTED', imported_file)
108
109    def playbook_on_not_import_for_host(self,
host, missing_file):
110        self.log(host, 'NOTIMPORTED',
missing_file)
```

2 开发插件规则

- 用Python编写
- 引发错误，就是遇到问题后，主动抛出异常
- 返回以unicode编码的字符串，主要是兼容 Jinja2
- 符合Ansible的配置和文档标准，就是可以通过 `ansible.cfg` 进行配置

2.1 使用兼容的 Python 版本编写

由于开发出来的插件将在控制器上执行，因此您必须使用兼容版本的Python（Python 2（2.7版）或 Python 3（3.5版及更高版本）的）进行编写。

2.2 抛出异常错误信息

应该通过引发 `AnsibleError()` 或类似的类并返回描述错误的消息来返回插件执行过程中遇到的错误。

将其他异常包装到错误消息中时，应始终使用Ansible 的函数 `to_native` 来确保跨Python版本的字符串兼容性：

```
1 from ansible.errors import AnsibleError
2 from ansible.module_utils._text import to_native
3
4 try:
5     cause_an_exception()
6 except Exception as e:
7     raise AnsibleError('Something happened, this was
    original exception: %s' % to_native(e))
```

检查不同的[AnsibleError对象](#), 然后查看哪种[对象](#)最适合您的情况。

2.3 妥当处理字符串

您必须将插件返回的所有字符串转换为Python的unicode类型。转换为unicode可确保这些字符串可以通过Jinja2运行。

转换字符串:

```
1 from ansible.module_utils._text import to_text
2 result_string = to_text(result_string)
```

2.4 插件配置和文档标准

Ansible的在线帮助文档是根据每个模块的源代码中的 `DOCUMENTATION` 模块生成的。该 `DOCUMENTATION` 块必须是有效的YAML。

需要为您的插件定义可配置选项，在python文件的部分
`DOCUMENTATION` 中对其进行描述。

自Ansible 2.4版以来，回调和连接插件已经开始以这种方式声明配置要求了。现在大多数插件类型都执行相同的操作。这种方法可确保插件选项的文档始终是正确的和最新的。

`DOCUMENTATION` 块中的所有字段均为小写。除非另有说明，否则所有字段都是必填字段：

```
1 DOCUMENTATION = '''
2     callback: log_plays
3     type: notification
4     short_description: write playbook output to log
file
5     version_added: historical
6     description:
7         - 此插件的详细描述信息。
8         - 使用多条目，不要使用一个较长的语句。
9         - 不应该提及模块名称。
10    requirements:
11        - 必须要求清单
12        - 包括最低版本的限制
13    options:
14        log_folder:
15            version_added: '2.9' 此插件添加到 Ansible 时候
的当时 Ansible 的版本。
16            default: 选项的默认值，如果 required 是 False,
则 default 可以设置
17            description: 此选项的作用的详细说明。应该用完整的
句子写成。
18            env:
19                - name: 环境变量的名字
```

```

20         ini:
21             - section: 在 asible.cfg 中的配置块名称
22               key: log_folder在对应配置块下面的变量名称
23               required: True/False 必需时为 True,如果不设
置, 就认为
24                   不是必须的。
25               type: int/str/list 不是必须的
26         '''

```

要访问插件中的配置设置, 请使用 `self.get_option("log_folder")`。

如果需要显式个配置选项设置值, 请使用 `self.set_options()`

3 开发回调插件 mysql_plays

回调插件会在响应事件时, 向 Ansible添加新行为。

要创建回调插件, 请使用 `CallbacksBase` 类作为父类创建一个新类:

`mysql_plays.py`

```

1  from ansible.plugins.callback import CallbackBase
2
3  class CallbackModule(CallbackBase):
4      pass

```

在 `CallbackModule` 覆盖 `CallbackBase` 中的特定方法。

对于打算与Ansible 2.0及更高版本一起使用的插件，您应该仅覆盖以v2开头的方法。

对于可以重写哪些方法呢，可以参阅 [lib/ansible/plugins/callback](#) 目录下的 `__init__.py` 文件的内容。

我们下面就参考 `log_plays` 插件编写一个可以将 playbook 的执行结果写如到 MySQL 中的插件。

3.1 准备数据库

首先要设计一个库和表用于存储结果

```
1 mysql> create database if not exists ansible default
  charset utf8mb4 collate utf8mb4_general_ci;
2 Query OK, 1 row affected (0.00 sec)
3 mysql> grant all on ansible.* to ansible@'%'
  identified by 'QFedu123!';
4 Query OK, 0 rows affected, 1 warning (0.00 sec)
```

3.2 准备表

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
user	varchar(16)	NO		NULL	
host	varchar(32)	NO		NULL	
category	varchar(11)	NO		NULL	
result	text	YES		NULL	
create_time	datetime	NO		CURRENT_TIMESTAMP	

```

1 mysql> create table playsresult(
2     id int auto_increment primary key,
3     user varchar(16) not null,
4     host varchar(32) not null,
5     category varchar(11) not null,
6     result text,
7     create_time datetime NOT NULL DEFAULT
CURRENT_TIMESTAMP
8 );

```

3.3 编写插件

```

1 #coding:utf-8
2 # (C) 2020, 闫顺军, <sharkyun@aliyun.com>
   <WeChat:y86000153>
3 # (c) 2020 Ansible Custom Plugin Project
4 # GNU General Public License v3.0+ (see COPYING or
   https://www.gnu.org/licenses/gpl-3.0.txt)
5
6 from __future__ import (absolute_import, division,
   print_function)
7 __metaclass__ = type
8
9 DOCUMENTATION = '''
10     callback: mysql_plays
11     type: notification
12     short_description: 将 playbook 的执行结果输出到
MySQL 中。
13     version_added: historical
14     description:
15         - 这个回调插件将会把输出存入 MySQL 服务器中。
16     requirements:

```

```

17 - 需要配置到 ansible.cfg 中 whitelist
18 - 可以被访问的 MySQL 服务器实例
19 - Python 版本对应的 pymysql 或者 mysqlclient 模
块
20 - 创表语句(注意:这里的表名需要根据选项中
mysql_table 的值一致)
21     create table playsresult(
22         id int auto_increment primary key,
23         user varchar(16) not null,
24         host varchar(32) not null,
25         category varchar(11) not null,
26         result text,
27         create_time datetime NOT NULL DEFAULT
CURRENT_TIMESTAMP
28     );
29 options:
30     mysql_host:
31         version_added: '2.9'
32         default: localhost
33         description: MySQL 服务器 IP或者主机名.
34         env:
35             - name: ANSIBLE_MYSQL_HOST
36         ini:
37             - section: callback_mysql_plays
38               key: mysql_host
39     mysql_port:
40         version_added: '2.9'
41         default: 3306
42         description: MySQL 服务器监听端口.
43         env:
44             - name: ANSIBLE_MYSQL_PORT
45         ini:
46             - section: callback_mysql_plays
47               key: mysql_port
48         type: int

```

```
49     mysql_user:
50         version_added: '2.9'
51         default: ansible
52         description: MySQL 服务器登录用户.
53         env:
54             - name: ANSIBLE_MYSQL_USER
55         ini:
56             - section: callback_mysql_plays
57               key: mysql_user
58     mysql_password:
59         version_added: '2.9'
60         default: 'QFedul23!'
61         description: MySQL 服务器登录用户.
62         env:
63             - name: ANSIBLE_MYSQL_PASSWORD
64         ini:
65             - section: callback_mysql_plays
66               key: mysql_password
67     mysql_db:
68         version_added: '2.9'
69         default: ansible
70         description: 存放数据的库名称.
71         env:
72             - name: ANSIBLE_MYSQL_DB
73         ini:
74             - section: callback_mysql_plays
75               key: db
76     mysql_table:
77         version_added: '2.9'
78         default: playsresult
79         description: 存放数据的表名称.
80         env:
81             - name: ANSIBLE_MYSQL_TABLE
82         ini:
83             - section: callback_mysql_plays
```

```
84         key: mysql_table
85     '''
86
87     import json
88     import getpass
89
90     from
91     ansible.module_utils.common._collections_compat
92     import MutableMapping
93     from ansible.parsing.ajson import
94     AnsibleJSONEncoder
95     from ansible.plugins.callback import CallbackBase
96     from ansible.errors import AnsibleError
97     from ansible.module_utils._text import to_native
98
99     try:
100         import pymysql as mysqldb
101         pwd = "password"
102         database = "db"
103     except ImportError:
104         try:
105             import MySQLdb as mysqldb
106             pwd = "passwd"
107             database = "database"
108         except ImportError:
109             raise AnsibleError("找不到 pymysql 或
110             mysqlclient 模块。")
111
112     class CallbackModule(CallbackBase):
113         """
114         把 playbook 的结果保存到 MySQL 数据库中，默认的库.表
115         是 ansible.playsresult
116         """
```

```
114     CALLBACK_VERSION = 2.0
115     CALLBACK_TYPE = 'notification'
116     CALLBACK_NAME = 'mysql_plays'
117     CALLBACK_NEEDS_WHITELIST = True
118
119     TIME_FORMAT = "%b %d %Y %H:%M:%S"
120     MSG_FORMAT = "%(now)s - %(category)s - %(data)s\n\n"
121
122     def __init__(self):
123         super(CallbackModule, self).__init__()
124
125     def set_options(self, task_keys=None,
126 var_options=None, direct=None):
127         """
128         用于设置选项和获取选项，选项包含了自定义的选项
129         """
130         super(CallbackModule,
131 self).set_options(task_keys=task_keys,
132 var_options=var_options, direct=direct)
133
134         self.mysql_host =
135 self.get_option("mysql_host")
136         self.mysql_port =
137 self.get_option("mysql_port")
138         self.mysql_user =
139 self.get_option("mysql_user")
140         self.mysql_password =
141 self.get_option("mysql_password")
142         self.mysql_db =
143 self.get_option("mysql_db")
144         self.mysql_table =
145 self.get_option("mysql_table")
146
147         self.user = getpass.getuser()
```



```

139
140     def _mysql(self):
141         """
142         连接数据库, 返回数据库对象和游标对象
143         """
144         db_conn={ "host": self.mysql_host,
145                   "port": self.mysql_port,
146                   "user": self.mysql_user,
147                   pwd: self.mysql_password,
148                   database: self.mysql_db}
149
150         try:
151             db = MySQLdb.connect(**db_conn)
152         except Exception as e:
153             raise AnsibleError("%s" %
to_native(e))
154
155         cursor= db.cursor()
156
157         return db, cursor
158
159
160     def _execute_sql(self, host, category, data):
161         if isinstance(data, MutableMapping):
162             if '_ansible_verbose_override' in
data:
163                 # avoid save extraneous data
164                 data = 'omitted'
165             else:
166                 data = data.copy()
167                 invocation =
data.pop('invocation', None)
168                 data = json.dumps(data,
cls=AnsibleJSONEncoder)
169                 if invocation is not None:

```

```
170         data = json.dumps(invocation)
171
172         + " => %s " % data
173
174         sql = """
175             insert into {}
176             (host,user,category,result)
177             values(%s,%s,%s,%s)
178             """.format(self.mysql_table)
179
180         db, cursor = self._mysql()
181
182         try:
183             # 执行 sql, 记录事件类型和事件结果
184             cursor.execute(sql, (host, self.user,
185 category, data))
186             db.commit()
187         except Exception as e:
188             raise AnsibleError("%s" %
189 to_native(e))
190         finally:
191             cursor.close()
192             db.close()
193
194         def runner_on_failed(self, host, res,
195 ignore_errors=False):
196             self._execute_sql(host, 'FAILED', res)
197
198         def runner_on_ok(self, host, res):
199             self._execute_sql(host, 'OK', res)
200
201         def runner_on_skipped(self, host, item=None):
202             self._execute_sql(host, 'SKIPPED', '...')
203
204         def runner_on_unreachable(self, host, res):
```

```
199         self._execute_sql(host, 'UNREACHABLE',
    res)
200
201     def runner_on_async_failed(self, host, res,
    jid):
202         self._execute_sql(host, 'ASYNC_FAILED',
    res)
203
204     def playbook_on_import_for_host(self, host,
    imported_file):
205         self._execute_sql(host, 'IMPORTED',
    imported_file)
206
207     def playbook_on_not_import_for_host(self,
    host, missing_file):
208         self._execute_sql(host, 'NOTIMPORTED',
    missing_file)
209
```

请注意，`CALLBACK_VERSION`和`CALLBACK_NAME`定义是Ansible 2.0版及更高版本正确运行的插件所必需的。

3.4 保存插件到有效的目录下

把插件保存为 `mysql_plays.py` 文件，并存到ansible 控制节点的如下目录下: `~/.ansible/plugins/callback/`

```
1 [root@qfedu.com ~]# pwd
2 /root
3 [root@qfedu.com ~]# ls
4 .ansible/plugins/callback/mysql_plays.py
```

或者 `/usr/share/ansible/plugins/callback`

3.5 开启使用插件

在 `ansible.cfg` 中编辑如下配置

```
1 callback_whitelist = mysql_plays
```

如果还使用了其他插件，请用英文的逗号分开。

比如

```
1 callback_whitelist = timer, mysql_plays
```

默认此插件仅对 `playbook` 生效，假如希望在 `ad-hoc` (快捷命令) 中生效，继续打开如下配置，并职位 `True`

```
1 bin_ansible_callbacks = True
```

3.6 关于此插件的使用先决条件等信息

在做好以上步骤后，使用如下方式获取帮助

验证配置的正确性

```
1 [root@qfedu.com ~]# ansible-doc -t callback -l |grep
mysql_plays
2 mysql_plays          将 playbook 的执行结果输出到 MySQL
                        中。
```

查看帮助文档

```
1 [root@qfedu.com ~]# ansible-doc -t callback
mysql_plays
2 > MYSQL_PLAYS
  (/root/.ansible/plugins/callback/mysql_plays.py)
3
4     这个回调插件将会把输出存入 MySQL 服务器中。
5
6     * This module is maintained by The Ansible
Community
7 OPTIONS (= is mandatory):
8
9 - mysql_db
10     存放数据的库名称.
11     [Default: ansible]
12     set_via:
13         env:
14             - name: ANSIBLE_MYSQL_DB
15             ini:
16             - key: db
17             section: callback_mysql_plays
18
19     version_added: 2.9
20
21 - mysql_host
22     MySQL 服务器 IP或者主机名.
23     [Default: localhost]
24     set_via:
```

```
25         env:
26             - name: ANSIBLE_MYSQL_HOST
27         ini:
28             - key: mysql_host
29             section: callback_mysql_plays
30
31         version_added: 2.9
32
33 - mysql_password
34     MySQL 服务器登录用户.
35     [Default: QFedu123!]
36     set_via:
37         env:
38             - name: ANSIBLE_MYSQL_PASSWORD
39         ini:
40             - key: mysql_password
41             section: callback_mysql_plays
42
43         version_added: 2.9
44
45 - mysql_port
46     MySQL 服务器监听端口.
47     [Default: 3306]
48     set_via:
49         env:
50             - name: ANSIBLE_MYSQL_PORT
51         ini:
52             - key: mysql_port
53             section: callback_mysql_plays
54
55         type: int
56         version_added: 2.9
57
58 - mysql_table
59     存放数据的表名称.
```

```

60         [Default: playsresult]
61     set_via:
62         env:
63             - name: ANSIBLE_MYSQL_TABLE
64             ini:
65                 - key: mysql_table
66                 section: callback_mysql_plays
67
68     version_added: 2.9
69
70 - mysql_user
71     MySQL 服务器登录用户.
72     [Default: ansible]
73     set_via:
74         env:
75             - name: ANSIBLE_MYSQL_USER
76             ini:
77                 - key: mysql_user
78                 section: callback_mysql_plays
79
80     version_added: 2.9
81
82
83 REQUIREMENTS: 需要配置到 ansible.cfg 中 Whitelist, 可
84                以被访问的 MySQL 服务器实例, Python 版本对应的 pymysql 或
85                者
86                mysqlclient 模块, 创表语句(注意:这里的表名需要根
87                据选项中 mysql_table 的值一致) create table
88                playsresult( id int auto_increment primary
89                key, user varchar(16) not
90                null, host varchar(32) not null, category
91                varchar(11) not null, result
92                text, create_time datetime NOT NULL DEFAULT
93                CURRENT_TIMESTAMP );

```

```
89         METADATA:
90             status:
91                 - preview
92             supported_by: community
93
94 TYPE: notification
```

3.6 配置插件使用的选项

关于限制条件

此插件已经有默认值，如果想修改需在 `ansible.cfg` 文件的最后添加如下配置

```
1  [callback_mysql_plays]
2  mysql_host = MySQL IP
3  mysql_port = MySQL 监听端口
4  mysql_user = MySQL 用户
5  mysql_password = MySQL 密码
6  mysql_db = MySQL 库名
7  mysql_table = MySQL 表名
```

3.7 执行 playbook

playbook


```
1 - hosts: all
2   gather_facts: no
3   tasks:
4     - name: test
5       shell: date +"%F %T"
```

Inventory

```
1 [root@qfedu.com ~]#[dbservers]
2 172.18.0.3
3
4 [webservers]
5 172.18.0.4
6 172.18.0.5
7
8 [allservers:children]
9 dbservers
10 webservers
```

执行playbook

```

1 [root@qfedu.com ~]# ansible-playbook -i hosts
  remoteDate.yml
2
3 PLAY [all]
  *****
4
5 TASK [test]
  *****
6 fatal: [172.18.0.5]: UNREACHABLE! => {"changed":
  false, "msg": "Failed to connect to the host via
  ssh: ssh: connect to host 172.18.0.5 port 22:
  Connection refused", "unreachable": true}
7 changed: [172.18.0.3]
8 changed: [172.18.0.4]
9
10 PLAY RECAP
  *****
  *****
  *****
11 172.18.0.3                : ok=1    changed=1
    unreachable=0    failed=0    skipped=0    rescued=0
    ignored=0
12 172.18.0.4                : ok=1    changed=1
    unreachable=0    failed=0    skipped=0    rescued=0
    ignored=0
13 172.18.0.5                : ok=0    changed=0
    unreachable=1    failed=0    skipped=0    rescued=0
    ignored=0

```

查询数据库

为了输出效果，已对输出信息做成修改

```

1  mysql> select * from playsresult\G
2  ***** 11. row *****
3      id: 21
4      user: root
5      host: 172.18.0.5
6      category: UNREACHABLE
7      result: {
8          "msg": "Failed to connect to the host via ssh:
ssh: connect to host 172.18.0.5 port 22: Connection
refused",
9          "unreachable": true,
10         "changed": false}
11 create_time: 2020-04-24 01:34:46
12 ***** 12. row *****
13      id: 22
14      user: root
15      host: 172.18.0.3
16      category: OK
17      result: {
18          "module_args": {"warn": true,
19                          "executable": null,
20                          "_uses_shell": true,
21                          "strip_empty_ends": true,
22                          "_raw_params": "date +\"%F
%T\""},
23                          "removes": null,
24                          "argv": null,
25                          "creates": null,
26                          "chdir": null,
27                          "stdin_add_newline": true,
28                          "stdin": null
29                      }
30      } => {"stderr_lines": [],
31            "cmd": "date +\"%F %T\""},

```

```

32         "end": "2020-04-24 01:34:46.762027",
33         "_ansible_no_log": false,
34         "stdout": "2020-04-24 01:34:46",
35         "changed": true,
36         "rc": 0,
37         "start": "2020-04-24 01:34:46.518139",
38         "stderr": "",
39         "delta": "0:00:00.243888",
40         "stdout_lines": ["2020-04-24 01:34:46"],
41         "ansible_facts": {
42             "discovered_interpreter_python":
43             "/usr/bin/python"
44         }
45     create_time: 2020-04-24 01:34:46
46     ***** 13. row *****
47         id: 23
48         user: root
49         host: 172.18.0.4
50     category: OK
51     result: {
52         "module_args": {"warn": true,
53                         "executable": null,
54                         "_uses_shell": true,
55                         "strip_empty_ends": true,
56                         "_raw_params": "date +\"%F
57                         %T\""},
58         "removes": null,
59         "argv": null,
60         "creates": null,
61         "chdir": null,
62         "stdin_add_newline": true,
63         "stdin": null
64     }
65     } => {"stderr_lines": [],

```

```
64         "cmd": "date +\"%F %T\"",
65         "end": "2020-04-24 01:34:46.767316",
66         "_ansible_no_log": false,
67         "stdout": "2020-04-24 01:34:46",
68         "changed": true,
69         "rc": 0,
70         "start": "2020-04-24 01:34:46.528226",
71         "stderr": "",
72         "delta": "0:00:00.239090",
73         "stdout_lines": ["2020-04-24
01:34:46"],
74         "ansible_facts": {
75             "discovered_interpreter_python":
"/usr/bin/python"}
76     }
77     create_time: 2020-04-24 01:34:46
78
```