

常见问题说明

蓝鲸社区版，是蓝鲸智云提供的面向社区用户的基于 PaaS 的运维技术解决方案套件。它永久免费，支持公有云环境、私有环境的独立搭建部署。

本 FAQ 文档主要介绍蓝鲸社区版的在日常安装部署、维护、更新升级中遇到的常见问题及故障排查等运维相关的内容。

关于蓝鲸各大平台、SaaS 应用的相关使用说明，请参考蓝鲸社区版产品白皮书。

部署常见问题

部署 CMDB 常见问题

prot 31001 start failed, please check

```
initdata for cmdb()
[10.0.5.92]20190930-160712 172 exec initdata_cmdb on 10.0.5.103
[10.0.5.103]20190930-160713 104 start to make migration for cmdb ...
[10.0.5.103]20190930-160713 112 on-migrate ... /data/bkce/cmdb/server/on_migrate
[10.0.5.103]20190930-160713 22 waiting for ports ready ...
[10.0.5.103]20190930-160724 153 port 31001 start failed, please check
[10.0.5.103]20190930-160724 153 migrate failed for cmdb(server)
[10.0.5.92]20190930-160724 179 Abort
```

- 检查 cmdb 服务状态，参照下图

```
./bkcec status cmdb
```

```
installl]# ./bkcec status cmdb
cmdb-nginx: RUNNING
server      cmdb_adminserver      RUNNING      pid 26842, uptime 15 days, 17:43:03
server      cmdb_apiserver         RUNNING      pid 26839, uptime 15 days, 17:43:03
server      cmdb_auditcontroller   RUNNING      pid 26835, uptime 15 days, 17:43:03
server      cmdb_datacollection    RUNNING      pid 26841, uptime 15 days, 17:43:03
server      cmdb_eventserver       RUNNING      pid 26840, uptime 15 days, 17:43:03
server      cmdb_hostcontroller    RUNNING      pid 26829, uptime 15 days, 17:43:03
server      cmdb_hostserver        RUNNING      pid 26830, uptime 15 days, 17:43:03
server      cmdb_objectcontroller  RUNNING      pid 26832, uptime 15 days, 17:43:03
server      cmdb_proccontroller    RUNNING      pid 26846, uptime 15 days, 17:43:03
server      cmdb_procserver        RUNNING      pid 26834, uptime 15 days, 17:43:03
server      cmdb_toposerver        RUNNING      pid 26831, uptime 15 days, 17:43:03
server      cmdb_webserver         RUNNING      pid 26833, uptime 15 days, 17:43:03
```

- 检查防火墙端口是否有开(8300, 8301, 8302)
- 查看日志，登录所在机器的路径：`/data/bkce/logs/cmdb/`
- `cmdb_adminserver` 服务状态 failed

- 检查依赖服务是否正常 `redis mongodb nginx gse zk`
- 若服务状态都是 RUNNING 则查看能否解析

```
dig zk.service.consul
```

```
[root@rbtnode1 /data/install]$ dig zk.service.consul

; <<>> DiG 9.11.4-P2-RedHat-9.11.4-9.P2.el7 <<>> zk.service.consul
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59031
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;zk.service.consul.      IN      A

;; ANSWER SECTION:
zk.service.consul.      0      IN      A      10.0.5.1
zk.service.consul.      0      IN      A      10.0.5.1
zk.service.consul.      0      IN      A      10.0.5.1

;; Query time: 88 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Fri Dec 20 15:28:40 CST 2019
;; MSG SIZE rcvd: 83
```

解析异常处理方法:

- 检查三台服务器 `/etc/resolv.conf` 配置, 首行是否有配置 `nameserver 127.0.0.1`, 如无, 请添加
- 检查 consul 配置中是否有 `ls /data/bkce/etc/consul.d/zk-config.json`, 如无, 则重装 consul 服务

```
./bkcec stop consul #(或在consul服务所在的三台主机, ps -ef |grep consul | awk '{print $2}' |xargs kill -9)
./bkcec install consul 1 #(1代表覆盖安装)
./bkcec start consul
```

- 若安装 consul 报错

检查 `/data/src/service/consul/` 是否有这两个文件夹 `bin` 、 `conf`

- 备份一下 src 下的 `.pip/pip.conf` 文件, 然后重新解压一下 `bkce_src` 安装包, 继续检查是否有文件, 如果还没有
- 解压时直接用 `tar xf` 包名, 不要加 `-C`, 若还是没有文件则去官网下载新包重新解压
- 对比包的 md5 是否和官网一致

cmdb-nginx 服务状态 failed

检查 CMDB 模块所在机器上是否能 YUM 安装 Nginx `yum info nginx`

安装 epel YUM 源, 重装 CMDB

```
./bkcec stop cmdb
./bkcec install cmdb 1
./bkcec start cmdb
./bkcec initdata cmdb
```

三台机器的 YUM 源都更新一致, 确保 YUM 源能安装 Nginx

其他进程状态 EXIT, 请前往 CMDB 所在服务器

`/data/bkce/logs/cmdb/` 目录下查看相应的日志

部署 App_mgr 常见问题

```
[root@rbtndel install]# ./bk_install
usage: bk_install PLATFORM
PLATFORM:
./bk_install paas      安装 paas 及依赖环境
./bk_install cmdb      安装 cmdb 及依赖环境
./bk_install job       安装 job 及依赖环境
./bk_install app_mgr   安装 appo, appt 及依赖
./bk_install bkdata    安装 bkdata 及依赖环境
./bk_install gse_agent 在蓝鲸服务器上安装 gse_agent, 并将主机注册到cmdb
./bk_install fta       安装 故障自愈后台
./bk_install saas-[o|t] [APP_CODE[==x.x.x]] 从后台部署 saas 应用
[root@rbtndel install]# ./bk_install app_mgr
[172.23.33.34]20190827-154022 28 install rabbitmq has been performed without FATAL stat... skip
[172.23.33.34]20190827-154022 28 initdata rabbitmq has been performed without FATAL stat... skip
[172.23.33.34]20190827-154022 28 activate rabbitmq has been performed without FATAL stat... skip
[172.23.33.34]20190827-154022 28 install appt has been performed without FATAL stat... skip
[172.23.33.34]20190827-154022 28 initdata appt has been performed without FATAL stat... skip
[172.23.33.34]20190827-154022 28 start appt has been performed without FATAL stat... skip
[172.23.33.34]20190827-154023 72 activate appt failed. api:/v1/agent/init/?agent_ip=172.23.33.34, response: [{"msg": "ac
tive 172.23.33.34 fail, the paas_agent return: {'u'-'sid or token is not valid, please check', 'u'-'error': 20100}"]
```

该报错是激活 paas_agent 失败, 需要查看的是 appo 服务还是 appt 服务

```
./bkcec status appo
./bkcec status appt
```

若是异常重启进程, 启动失败需要查看日志详情 `/data/bkce/logs/paas_agent/`

进程正常启动后再激活

```
./bkcec activate appo
./bkcec activate appt
```

部署 BKDATA 常见问题

`./bk_install bkdata` 报错

```
failed with error code 1 in /tmp/pip-build-8g97ci/MySQL-python/
install python package for bdata(monitor) failed pip optin: --no-index --find-links=/da
```

```
ta/src/bkdata/support-files/pkgs
```

MYSQL-PYTHON 安装失败

```
Collecting wcwidth (from prompt-toolkit<2.0.0,>=1.0.4->ipython==5.2.2->-r requirements.txt (line 22))
Collecting amqp<2.0,>=1.4.9 (from kombu<3.1,>=3.0.24->celery==3.1.17->-r requirements.txt (line 34))
Collecting anyjson<0.3.3 (from kombu<3.1,>=3.0.24->celery==3.1.17->-r requirements.txt (line 34))
Installing collected packages: MySQL-python, SQLAlchemy, inflect, sqlalchemy, Werkzeug, MarkupSafe, Jinja2, itsdangerous, Flask, greenlet, gevent, coverage, readline, gunicorn, pygal, simplegeneric, scandir, pathlib2, pickleshare, backports.shutil-get-terminal-size, pyprocess, pexpect, enum34, ipython-genutils, decorator, traitlets, pygments, wcwidth, prompt-toolkit, ipython, urllib3, elasticsearch, elasticsearch-dsl, simplejson, affinity, python-snappy, avro, Mako, python-editor, alembic, Django, ujson, amqp, anyjson, kombu, billiard, celery, python-consul, netifaces, psutil
Running setup.py install for MySQL-python: started
Running setup.py install for MySQL-python: finished with status 'error'
Complete output from command /data/bkce/ens/monitor/bin/python -u -c 'import setuptools, tokenize; __file__ = '/tmp/pip-build-8g97c1/MySQL-python/setup.py'; f=getattr(tokenize, 'open', open)(__file__); code=f.read().replace('\r\n', '\n'); f.close(); exec(compile(code, __file__, 'exec'))' install --record /tmp/pip-Pd0ZJU-record/install-record.txt --single-version-externally-managed --compile --install-headers /data/bkce/ens/monitor/include/site/python2.7/MySQL-python:
running install
running build
running build_py
creating build
creating build/lib.linux-x86_64-2.7
copying _mysql_exceptions.py -> build/lib.linux-x86_64-2.7
creating build/lib.linux-x86_64-2.7/MySQLdb
copying MySQLdb/_init_.py -> build/lib.linux-x86_64-2.7/MySQLdb
copying MySQLdb/converters.py -> build/lib.linux-x86_64-2.7/MySQLdb
copying MySQLdb/connections.py -> build/lib.linux-x86_64-2.7/MySQLdb
copying MySQLdb/cursors.py -> build/lib.linux-x86_64-2.7/MySQLdb
copying MySQLdb/release.py -> build/lib.linux-x86_64-2.7/MySQLdb
copying MySQLdb/times.py -> build/lib.linux-x86_64-2.7/MySQLdb
creating build/lib.linux-x86_64-2.7/MySQLdb/constants
copying MySQLdb/constants/_init_.py -> build/lib.linux-x86_64-2.7/MySQLdb/constants
copying MySQLdb/constants/CR.py -> build/lib.linux-x86_64-2.7/MySQLdb/constants
copying MySQLdb/constants/FIELD_TYPE.py -> build/lib.linux-x86_64-2.7/MySQLdb/constants
copying MySQLdb/constants/ER.py -> build/lib.linux-x86_64-2.7/MySQLdb/constants
copying MySQLdb/constants/FLAG.py -> build/lib.linux-x86_64-2.7/MySQLdb/constants
copying MySQLdb/constants/REFRESH.py -> build/lib.linux-x86_64-2.7/MySQLdb/constants
copying MySQLdb/constants/CLIENT.py -> build/lib.linux-x86_64-2.7/MySQLdb/constants
running build_ext
building 'mysql' extension
creating build/temp.linux-x86_64-2.7
gcc -pthread -fno-strict-aliasing -g -O2 -DDEBUG -g -fwrapv -O3 -Wall -Wstrict-prototypes -fPIC -Dversion_info=(1,2,5,'final',1) -D__version__=1.2.5 -I/usr/include/mysql -I/opt/py27/include/python2.7 -c _mysql.c -o build/temp.linux-x86_64-2.7/_mysql.o
gcc -pthread -shared build/temp.linux-x86_64-2.7/_mysql.o -L/usr/lib -lmysqlclient -lpthread -lm -lrt -ldl -o build/lib.linux-x86_64-2.7/_mysql.so
/usr/bin/ld: cannot find -lmysqlclient
collect2: error: ld returned 1 exit status
error: command 'gcc' failed with exit status 1

-----
Command "/data/bkce/ens/monitor/bin/python -u -c 'import setuptools, tokenize; __file__ = '/tmp/pip-build-8g97c1/MySQL-python/setup.py'; f=getattr(tokenize, 'open', open)(__file__); code=f.read().replace('\r\n', '\n'); f.close(); exec(compile(code, __file__, 'exec'))' install --record /tmp/pip-Pd0ZJU-record/install-record.txt --single-version-externally-managed --compile --install-headers /data/bkce/ens/monitor/include/site/python2.7/MySQL-python" failed with error code 1 in /tmp/pip-build-8g97c1/MySQL-python/
[IL 20190507-073657.153] install python package for bkdata(monitor) failed. pip optin: --no-index --find-links=/data/src/bkdata/support-files/pkgs
[IL 20190507-073657.37] Abort
```

- 解决方案:

1. 确保 mysql-devel 已经安装

```
yum install mysql-devel
```

```
[root@rbtnode1 install]# yum install mysql-devel
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
Package 1:mariadb-devel-5.5.64-1.el7.x86_64 already installed and latest version
Nothing to do
```

```
rpm -qa | grep mariadb-devel
```

```
[root@rbtnode1 install]# rpm -qa | grep mariadb-devel
mariadb-devel-5.5.64-1.el7.x86_64
```

2. 建立软连接

```
dnwxr-xr-x 3 root root 4096 May 27 16:10 influxdb
dnwxr-xr-x 6 root root 4096 Mar 7 14:39 lbd
dnwxr-xr-x 2 root root 4096 Mar 7 14:39 kdump
dnwxr-xr-x 3 root root 4096 Oct 3 2018 kernel
lnwxrwxrwx 1 root root 36 May 27 16:30 libmysqlclient_r.so -> /usr/lib64/mysql/libmysqlclient_r.so
lnwxrwxrwx 1 root root 34 May 27 16:31 libmysqlclient.so -> /usr/lib64/mysql/libmysqlclient.so
lnwxrwxrwx 1 root root 37 May 27 16:31 libmysqlclient.so.18 -> /usr/lib64/mysql/libmysqlclient.so.18
lnwxrwxrwx 1 root root 41 May 27 16:31 libmysqlclient.so.18.0.0 -> /usr/lib64/mysql/libmysqlclient.so.18.0.0
dnwxr-xr-x 2 root root 4096 May 27 15:53 locate
dnwxr-xr-x 2 root root 4096 Mar 7 14:39 modprobe.d
dnwxr-xr-x 3 root root 4096 Mar 7 14:39 modules
dnwxr-xr-x 2 root root 4096 Oct 31 2018 modules-load.d
dnwxr-xr-x 4 root root 4096 Mar 7 14:39 NetworkManager
dnwxr-xr-x 3 root root 4096 May 27 16:03 ocf
dnwxr-xr-x 2 root root 4096 Mar 7 14:39 polkit-1
dnwxr-xr-x 3 root root 4096 Apr 9 22:31 python2.7
dnwxr-xr-x 4 root root 4096 May 27 16:03 rabbitmq
dnwxr-xr-x 4 root root 4096 Mar 7 14:39 rpm
lnwxrwxrwx 1 root root 30 Mar 7 14:39 sendmail -> /etc/alternatives/mta-sendmail
lnwxrwxrwx 1 root root 34 Mar 7 14:39 sendmail.cf -> /etc/mail/sendmail.cf
```

```
ln -s /usr/lib64/mysql/libmysqlclient_r.so /usr/lib/libmysqlclient_r.so
ln -s /usr/lib64/mysql/libmysqlclient.so /usr/lib/libmysqlclient.so
ln -s /usr/lib64/mysql/libmysqlclient.so.18 /usr/lib/libmysqlclient.so.18
ln -s /usr/lib64/mysql/libmysqlclient.so.18.0.0 /usr/lib/libmysqlclient.so.18.0.0
```

重新执行命令部署 BKDATA 即可恢复

安装 PYTHON-SNAPPY 包失败

原因是缺少 snappy-c.h 导致 pip 安装 python-snappy 包失败

```
Installing collected packages: python-snappy, zero, Mako, python-editor, Alembic, Django, ujson, amqp, anyjson, kombu, billiard, celery, python-consul, netifaces, poutils
Running setup.py install for python-snappy: started
Running setup.py install for python-snappy: finished with status 'error'
Complete output from command /data/bkce/.envs/monitor/lib/python2.7/bin/python -m pip install --record /tmp/pip-KPvPT-record/install-record.txt --single-version-externally-managed --compile --install-header /data/bkce/.envs/monitor/include/site/python2.7/python-snappy-c.h:
running install
running build
running build_py
creating build/lib.linux-x86_64-2.7
copying snappy.py -> build/lib.linux-x86_64-2.7
running build_ext
building 'snappy' extension
creating build/temp.linux-x86_64-2.7
gcc -pthread -fno-strict-aliasing -g -O2 -DNDEBUG -g -fwrapup -O3 -Wall -Wstrict-prototypes -fPIC -I/opt/python27/include/python2.7 -c snappymodule.c -o build/temp.linux-x86_64-2.7/snappymodule.o
snappymodule.c:31:12: fatal error: snappy-c.h: No such file or directory
snappymodule.c:31:12: fatal error: snappy-c.h: No such file or directory
compilation terminated.
error: command 'gcc' failed with exit status 1

Complete output from command /data/bkce/.envs/monitor/lib/python2.7/bin/python -m pip install --record /tmp/pip-KPvPT-record/install-record.txt --single-version-externally-managed --compile --install-header /data/bkce/.envs/monitor/include/site/python2.7/python-snappy-c.h:
running install
running build
running build_py
creating build/lib.linux-x86_64-2.7
copying snappy.py -> build/lib.linux-x86_64-2.7
running build_ext
building 'snappy' extension
creating build/temp.linux-x86_64-2.7
gcc -pthread -fno-strict-aliasing -g -O2 -DNDEBUG -g -fwrapup -O3 -Wall -Wstrict-prototypes -fPIC -I/opt/python27/include/python2.7 -c snappymodule.c -o build/temp.linux-x86_64-2.7/snappymodule.o
snappymodule.c:31:12: fatal error: snappy-c.h: No such file or directory
snappymodule.c:31:12: fatal error: snappy-c.h: No such file or directory
compilation terminated.
error: command 'gcc' failed with exit status 1

Complete output from command /data/bkce/.envs/monitor/lib/python2.7/bin/python -m pip install --record /tmp/pip-KPvPT-record/install-record.txt --single-version-externally-managed --compile --install-header /data/bkce/.envs/monitor/include/site/python2.7/python-snappy-c.h:
running install
running build
running build_py
creating build/lib.linux-x86_64-2.7
copying snappy.py -> build/lib.linux-x86_64-2.7
running build_ext
building 'snappy' extension
creating build/temp.linux-x86_64-2.7
gcc -pthread -fno-strict-aliasing -g -O2 -DNDEBUG -g -fwrapup -O3 -Wall -Wstrict-prototypes -fPIC -I/opt/python27/include/python2.7 -c snappymodule.c -o build/temp.linux-x86_64-2.7/snappymodule.o
snappymodule.c:31:12: fatal error: snappy-c.h: No such file or directory
snappymodule.c:31:12: fatal error: snappy-c.h: No such file or directory
compilation terminated.
error: command 'gcc' failed with exit status 1
```

- 解决方案:

通过安装 snappy-devel 解决, `yum install -y snappy-devel`

启动报 "DATAAPI.SERVICE.CONSUL START FAILED ERROR: INIT_SNAPSHOT_CONFIG"

启动 bkdata 报错: `dataapi.service.consul start failed ERROR: init_snapshot_config (databus.tests.DatabusHealthTestCase)`

```
[root@vvvv-22-bk:/data/install]# ./bkceo start bkdata
[192.168.2.246]20190731-171020 72 starting bkdata(ALL) on host: 192.168.2.246
[192.168.2.246]20190731-171044 452 dataapi.service.consul start failed.
[192.168.2.246]20190731-171044 79 going to init snapshot data. this may take a while.
/data/bkce/.envs/dataapi/lib/python2.7/site-packages/django/db/backends/mysql/base.py:124: Warning: Field 'concurrency' doesn't have a default value
return self.cursor.execute(query, args)

/data/bkce/.envs/dataapi/lib/python2.7/site-packages/django/db/backends/mysql/base.py:124: Warning: Field 'trt_type' doesn't have a default value
return self.cursor.execute(query, args)

/data/bkce/.envs/dataapi/lib/python2.7/site-packages/django/db/backends/mysql/base.py:124: Warning: Field 'time_zone' doesn't have a default value
return self.cursor.execute(query, args)

http://databus.service.consul:10052/connectors
5
ERROR: init_snapshot_config (databus.tests.DatabusHealthTestCase)

Traceback (most recent call last):
  File "/data/bkce/bkdata/dataapi/databus/tests.py", line 154, in init_snapshot_config
    self._submit_snapshot_etl(biz_id, rt_id)
  File "/data/bkce/bkdata/dataapi/databus/tests.py", line 79, in _submit_snapshot_etl
    ret = requests.post(url, json.dumps(params), headers=headers)
  File "/data/bkce/.envs/dataapi/lib/python2.7/site-packages/requests/api.py", line 110, in post
    return request('post', url, data=data, json=json, **kwargs)
  File "/data/bkce/.envs/dataapi/lib/python2.7/site-packages/requests/api.py", line 56, in request
    return session.request(method=method, url=url, **kwargs)
  File "/data/bkce/.envs/dataapi/lib/python2.7/site-packages/requests/sessions.py", line 471, in request
    resp = self.send(prep, **kwargs)
  File "/data/bkce/.envs/dataapi/lib/python2.7/site-packages/requests/sessions.py", line 581, in send
    r = adapter.send(request, **kwargs)
  File "/data/bkce/.envs/dataapi/lib/python2.7/site-packages/requests/adapters.py", line 481, in send
    raise ConnectionError(e, request=request)
ConnectionError: HTTPConnectionPool(host='databus.service.consul', port=10052): Max retries exceeded with url: /connectors (Caused by NewConnectionError(<requests.packages.urllib3.connection.HTTPConnection object at 0x716f7b8f4890>: Failed to establish a new connection: [Errno -2] Name or service not known'))

Ran 1 test in 0.102s
FAILED (errors=1)
```

- 解决方案:

登陆到 `bkdata` 机器(社区版 5.1 登陆到 `databus` 所在机器)查看 `consul` 配置是否生成 `databus.json` 配置。

```
ls /data/bkce/etc/consul.d/bkdata.json
# 若无则重装 consul
./bkcec stop consul
./bkcec install consul 1
./bkcec start consul
./bkcec status consul

# 登陆到 databus 所在机器查看是否生成 bkdata.json(社区版 5.1 为 bkdata-databus.json, bkdata-
dataapi.jsonbkdata-monitor.json)
ls /data/bkce/etc/consul.d/bkdata.json

# 启动 bkdata
./bkcec start bkdata
```

部署 SaaS 常见问题

安装 saas-o 报错 `KeyError: "name='bk_csrf_token', domain=None, path=None"`

```
root@rbtmodel install]# ./bkcec install saas-o
Install saas-o(!!!)
Deploy official saas bk_fta_solutions
2019-04-18 11:02:35 35 INFO request login token
2019-04-18 11:02:35 44 INFO emulate login to http://paas.bk.local:80/login/, form data: {'username': 'admin', 'csrfmiddlewaretoken': 'VdcLTw010gx0nziGL0M6u0x09Ehtdpkx', 'password': 'P9S2Kt(0dH)'}
2019-04-18 11:02:35 48 INFO bklogin_csrf_token: VdcLTw010gx0nziGL0M6u0x09Ehtdpkx
Traceback (most recent call last):
  File "deck/saas.py", line 212, in <module>
    appmgr.upload_pkg(args.pkg_path, upload_token_url, upload_url)
  File "deck/saas.py", line 60, in upload_pkg
    upload_csrf_token = self.get_csrf_token(token_url, 'bk_csrf_token')
  File "deck/saas.py", line 26, in get_csrf_token
    return resp.cookies[token_name]
  File "/opt/python2.7/site-packages/requests/cookies.py", line 329, in __getitem__
    return self._find_no_duplicates(name)
  File "/opt/python2.7/site-packages/requests/cookies.py", line 400, in _find_no_duplicates
    raise KeyError('name=%r, domain=%r, path=%r' % (name, domain, path))
KeyError: "name='bk_csrf_token', domain=None, path=None"
10.1.90.123]20190418-110235 130 Deploy saas bk_fta_solutions failed.
10.1.90.123]20190418-110235 35 Abort
You have new mail in /var/spool/mail/root
root@rbtmodel install]# cd
```

- 解决方案

确认是否是在 PaaS 页面个人信息重置了密码后,但是 `globals.env` 文件没同步更新。请在 `globals.env` 文件中更新重置后的密码后确认是否恢复正常。同步配置信息

NameError: global name 'login' is not defined

```
./bkcec sync common
```

```
10.1.90.133]20191210-181552 128 SaaS application bk_log_search has been deployed successfully
Deploy official saas bk_monitor
2019-12-10 18:15:54 43 INFO request login token
2019-12-10 18:15:54 33 ERROR <RequestsCookieJar[<Cookie bklogin_csrf_token=mXpVNGHEa4q68RlwTlXaaXdtcf9Dzg5 for paas.bk.com/>]>
2019-12-10 18:15:54 52 INFO emulate login to http://paas.bk.com:80/login/, form data: {'username': 'admin', 'csrfmiddlewaretoken': 'mXpVNGHEa4q68RlwTlXaaXdtcf9Dzg5', 'password': 'P9S2Kt(0dH)'}
2019-12-10 18:15:55 56 INFO bklogin_csrf_token: 0Znvf6r8mm9pFddhgQ59d6nbfwMQI57y
2019-12-10 18:15:55 33 ERROR <RequestsCookieJar[<Cookie bk_csrf_token=Ujqo58Cjf3HoWQx1UenNcsK7N07mNz1a for paas.bk.com/>]>
2019-12-10 18:15:55 78 INFO get upload token:Ujqo58Cjf3HoWQx1UenNcsK7N07mNz1a from http://paas.bk.com:80/saas/upload/0/
2019-12-10 18:15:55 99 INFO uploading file /data/src/official_saas/bk_monitor_V1.4.02.tar.gz, url:http://paas.bk.com:80/saas/upload/0/, data: {'csrfmiddlewaretoken': 'Ujqo58Cjf3HoWQx1UenNcsK7N07mNz1a'} ...
Traceback (most recent call last):
  File "deck/saas.py", line 220, in <module>
    appmgr.upload_pkg(args.pkg_path, upload_token_url, upload_url)
  File "deck/saas.py", line 94, in upload_pkg
    login_error("upload failed:{}".format(resp.content))
  File "deck/saas.py", line 94, in login_error
    login_error("upload failed:{}".format(resp.content))
NameError: global name 'login' is not defined
10.1.90.133]20191210-181706 130 Deploy saas bk_monitor failed.
10.1.90.133]20191210-181707 37 Abort
您在 /var/spool/mail/root 中有新邮件
```

- 解决方案

```
./bkcec stop paas
./bkcec upgrade paas
```

```
./bkcec start paas
再次执行部署 saas
./bkcec install saas-o
```

PaaS 登陆无响应

校验登陆接口

- 登陆 `open_paas` 服务器，进入日志目录(`cd /data/bkce/logs/open_paas`)，并运行命令 `tail -f *login*`
- 点击浏览器出错的平台链接，观察日志是否滚动，请求到，且有 `is_login` 字样
- 通过查看日志中的 `is_login` 接口调用日志对应的域名来确认调用登录校验接口的方式，是直接调用 login 还是通过 ESB 调用。
- 在服务器上用 curl 模拟调用 `is_login` 方法看是否 OK `curl -v http://xxxx/is_login?bk_cookie=xxxxxxxxx`
- 查看 logs/open_paas/login 相关日志特别是 login_uwsgi 的日志，看是否有请求到

常见原因：

1. 其他机器到 paas 请求时有 http 代理被劫持。
2. 浏览器 cookie 缓存等问题(换 chrome 无痕，或者其他浏览器)

请求到了 PaaS 但校验失败

应用日志里提示"Login validity is illegal"

这种常出现在蓝鲸 server 机器之间的时间不同步。表现为有时候可以正常登陆，有时候不能。原因是，A 机器生成的 cookie，到 B 机器校验时间时，因为时间间隔大于默认的 1 分钟，会判断为过期，导致失败

解决方法：

- 调整蓝鲸 server 服务器时间，保持同步。
- 若客观原因导致没法同步时间，可以通过 `open_paas/login/conf/default.py` 中的 `BK_TOKEN_OFFSET_ERROR_TIME` 配置修改默认容忍的时间间隔。
- 因没有正确修改 `/data/install/globals.env` 下的域名配置。 `bash vim`
`/data/install/globals.env` # 域名信息 `export BK_DOMAIN="bk.com"` # 蓝鲸根域名(不含主机名) `export PAAS_FQDN="paas.$BK_DOMAIN"` # PAAS 完整域名 `export CMDB_FQDN="cmdb.$BK_DOMAIN"` # CMDB 完整域名 `export JOB_FQDN="job.$BK_DOMAIN"` # JOB 完整域名 `export DOCS_FQDN="docs.$BK_DOMAIN"` # 私有文档准备 `export APPO_FQDN="o.$BK_DOMAIN"` # 正式环境完整域名 `export APPT_FQDN="t.$BK_DOMAIN"` #

测试环境完整域名 如上 PAAS CMDB JOB 完整域名格式“.”这个连接符一定是在变量

`$BK_DOMAIN` 前面, `$BK_DOMAIN` 变量不可修改为实际值, 保留变量格式。

PaaS 启动提示 FATAL Exited too quickly

表象: 此问题多为正常状态情况下, `supervisord.sock` 被清理, 用 `stop paas` 提示可以停掉, `status paas` 时也显示 `EXIT`, 实际的进程还是异常的

```
[10.X.X.X] open_paas    appengine          FATAL    Exited too quickly (
process log may have details)
[10.X.X.X] open_paas    esb              FATAL    Exited too quickly (
process log may have details)
[10.X.X.X] open_paas    login            FATAL    Exited too quickly (
process log may have details)
[10.X.X.X] open_paas    paas             FATAL    Exited too quickly (
process log may have details)
[root@rbtnode1 /data/install]# ./bkcec stop paas
[10.X.X.X]20180730-094141 117    stopping open_paas(ALL) on host: 10.X.X.X
Shut down
[root@rbtnode1 /data/install]# ./bkcec status paas
-----
[10.X.X.X] open_paas: EXIT

[root@paas-1 /root]# ps auxf|grep open_paas
root      28606  0.0  0.6 534188 54916 ?        Sl   Jul27   0:42 /data/bkce/.envs/paas/
bin/uwsgi --ini /data/bkce/etc/uwsgi-open_paas-paas.ini
root      28676  0.0  0.4 534188 37572 ?        S    Jul27   0:00 \_ /data/bkce/.envs/p
aas/bin/uwsgi --ini /data/bkce/etc/uwsgi-open_paas-paas.ini
```

思路方法: 解决办法, 杀掉已经不正常的进程(此情况 `rabbitmq` 在异常时, 也可以杀掉 `epmd` 及 `beam`)

```
# 可选
[root@paas-1 /root]# /data/bkce/.envs/open_paas/bin/python /data/bkce/.envs/open_paas/b
in/supervisorctl -c /data/bkce/etc/supervisor-open_paas.conf shutdown
# 可尝试删掉 supervisor.sock 文件, 再重新创建, 但需注意赋权, 设置为 srwx-----
[root@paas-1 /data/bkce/logs/open_paas]# chown o+s supervisor.sock
# 可选
[root@paas-1 /root]# /data/bkce/.envs/open_paas/bin/python /data/bkce/.envs/open_paas/b
in/supervisord -c /data/bkce/etc/supervisor-open_paas.conf
[root@paas-1 /root]# for x in `ps auxf|grep open_paas|awk '{print $2}'`;do kill -9 $x;d
one
-bash: kill: (8230) - 没有那个进程
[root@paas-1 /root]# ps auxf|grep open_paas
root      8269  0.0  0.0 115748   720 pts/0    S+   09:47   0:00 \_ grep --col
or=auto open_paas
```

再重新启动

```
[root@rbtnode1 /data/install]# ./bkcec start paas
[10.X.X.X]20180730-094728 78    starting open_paas(ALL) on host: 10.178.138.39
[root@rbtnode1 /data/install]# ./bkcec status paas
-----
```



```
[10.X.X.X] open_paas    appengine    RUNNING    pid 9803, uptime 0:0
0:06
[10.X.X.X] open_paas    esb          RUNNING    pid 9802, uptime 0:0
0:06
[10.X.X.X] open_paas    login        RUNNING    pid 9801, uptime 0:0
0:06
[10.X.X.X] open_paas    paas         RUNNING    pid 9800, uptime 0:0
0:06
```

PaaS 重置访问密码

```
# admin 密码修改错误后，无法登陆。如何后台重置密码？
source /data/install/utils.fc

ssh $PAAS_IP
workon login
export BK_ENV='production'
python manage.py shell
# 看到 Python 终端后输入，密码'xxxxxx'改为自己要重置的内容
from bkaccount.models import BkUser
password = 'xxxxx'
all_user = BkUser.objects.filter(username='admin')
for user in all_user:
    user.set_password(password)
    user.save()
# 然后退出终端
unset BK_ENV
```

PaaS 登陆提示 502 Bad GateWay

访问 **集成平台** (open_paas)容易出现 502，原因比较多

- 确认 open_paas 4 个 web 服务启动正常，且可以访问(最可能的原因)，注意如果端口被占用，web 访问将启动失败；
- 确认防火墙策略，即 nginx 所在服务器能够请求到 **open_paas** 所在机器的对应服务 **curl**
http://{open_paas_ip}:8000
- 确认 nginx 中 paas.conf-server{listen 80;}，如果此时 502，可能是 nginx 部署机器多块网卡，而 listen 使用的网卡和访问域名配置的网卡不同导致的，变更 server{listen{网卡}:80}

appt 访问报 502 bad gateway

- **http://{PAAS_DOMAIN}/t/bk_framework/** 访问报 502 错误，确认实际访问路径

```
# 在 paas.conf 查到对应访问请求路径
location ~ ^/t/ {
    proxy_pass http://PAAS_AGENT_TEST;

# 真实访问服务器及端口
upstream PAAS_AGENT_TEST {
    server 10.x.x.x:8010 max_fails=1 fail_timeout=30s;
}
```

- 在 paas 服务器上进行测试 `curl http://10.x.x.x:8010`

```
[root@nginx-1 /data/bkee/etc/nginx]# curl http://10.x.x.x:8010
curl: (7) Failed connect to 10.x.x.x:8010; 拒绝连接
```

- 在 appt 上进行确认

```
# 8010 未监听起来
[root@rbtnode2 /data/bkee/etc/nginx]# netstat -lnpt|grep 8010
[root@rbtnode2 /data/bkee/etc/nginx]#

# 确认 nginx 的路径
[root@rbtnode2 /data/bkee/etc/nginx]# ps -ef|grep nginx
root      17847 26444  0 12:12 pts/1    00:00:00 grep --color=auto nginx
root      20934      1  0 Jul18 ?        00:00:00 nginx: master process nginx
[root@rbtnode2 /data/bkee/etc/nginx]# ll /proc/20934/|grep exe

# 重新 reloadnginx, 8010起来
[root@rbtnode2 /data/bkee/etc/nginx]# /usr/sbin/nginx -s reload
[root@rbtnode2 /data/bkee/etc/nginx]# netstat -lnpt|grep 8010
tcp        0      0 0.0.0.0:8010          0.0.0.0:*             LISTEN      20934/n
ginx: master
```

- 重新测试, `http://{PAAS_DOMAIN}/t/bk_framework/` 访问 OK

```
[root@nginx-1 ~]# curl http://10.X.X.X:8010
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Test Page for the Nginx HTTP Server on Fedora</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <style type="text/css">
      /**/</pre></div><div data-bbox="175 667 700 695" data-label="Section-Header"><h2>SaaS 部署的超时时间可以自行配置</h2></div><div data-bbox="175 700 822 736" data-label="Text"><p>蓝鲸智云 PaaS 平台部署应用时, 存在一个超时时间, 当应用部署超过超时时间后, 系统主动将部署任务设置为失败.</p></div><div data-bbox="175 757 347 772" data-label="Text"><p>当前存在两个地方的配置:</p></div><div data-bbox="189 792 820 857" data-label="List-Group"><ul><li>PaaSAgent 默认超时时间: <code>300s</code>, 当部署作业执行超过 300s 时, 设置任务状态为失败</li><li>PaaS 部署默认超时时间: <code>360s</code>, 注意一定要大于 PaaSAgent 设置的超时时间; 某些情况下网络超时或 PaaSAgent 失联, 确保部署任务不会卡在 <code>正在部署</code> 状态</li></ul></div><div data-bbox="175 876 317 892" data-label="Text"><p>如果要变更超时时间:</p></div>
```

PaaS Agent 变更

- 登录 PaaSAgent 服务器(APPO/APPT)
- 编辑 `paas_agent_config.yaml`, 修改 `EXECUTE_TIME_LIMIT` 字段值

```
/data/bkce/paas_agent/paas_agent/etc/paas_agent_config.yaml
```

- 重启 PaaSAgent

```
supervisorctl -c /data/bkce/etc/supervisor-paas_agent.conf restart all
```

PaaS 变更

- 登录 PaaS 服务器
- 编辑 `/data/bkce/open_paas/paas/conf/default.py` 中的 `EVENT_STATE_EXPIRE_SECONDS` 值
- 重启 PaaS

如何确认 PaaS 部署正常

首先 `ping {PAAS_DOMAIN}` 确定通域名通

其次 确认所有 healthz 接口正常:

- `http://{PAAS_DOMAIN}/healthz/` 开发者中心
- `http://{PAAS_DOMAIN}/login/healthz/` 登录服务
- `http://{PAAS_DOMAIN}/api/healthz/` esb 接口
- `http://{PAAS_DOMAIN}/console/healthz/` 桌面
- `http://{PAAS_DOMAIN}/v1/healthz/` appengine

healthz 接口会检查服务本身及其所有外部依赖, 若服务不可用或依赖有问题则非 200 并提示错误

如果有问题, 根据相应错误处理

在开发者中心无法查看日志

- 确认 es 服务正常, 且 PaaS 中配置的 es 地址正确
- 确认在 appo/appt 服务器上均部署了 `paas_plugins/log_agent`
- 确认部署了 `paas_plugins/log_parser`

- 确认 `log_agent` 及 `log_parser` 的 redis 配置一致
- 确认所有机器的时区及时间一致(重要)

其他问题

(1)蓝鲸 DevOps 技能有培训教程吗？

答：蓝鲸为社区用户准备了在线视频教程，可以免费观看。（链接：<https://cloud.tencent.com/developer/edu/major-100008>）

(2)学习蓝鲸 DevOps 技能过程中遇到问题怎么办？

答：学习过程中，任何疑问可咨询蓝鲸 QQ [在线客服](#)，

除此之外，蓝鲸为社区用户准备了交流 QQ 群：[蓝鲸社区版交流 1 群](#)495299374，

群内的社区用户和蓝鲸助手会热心为大家提供咨询服务。

(3)如何将蓝鲸体系的用户体系对接到企业内已有的 OA 系统？

答：蓝鲸提供了登录 API 接口，可以对接普通的账户 Vs 密码形式的 OA 账号，可参考 [企业内部登录对接蓝鲸智云统一登录指南](#) 完成对接，更多对接的功能在持续开发中。

配置平台常见问题

CMDB 无快照数据

“

此文档描述 4.0 至 5.0 的社区版的问题排查

Windows 主机暂不支持快照数据

”

表象：配置管理系统的实时状态显示 `当前主机没有安装 Agent 或者 Agent 已经离线`

原因

1. gse 及 gse_agent 状态不正常
2. bkdata 的快照数据任务不存在
3. kafka 内节点或 topic 数据异常

思路方法

- 检查模块状态
- 确定 cmdb, gse, gse_agent, zk, kafka 模块的状态是否正常，可以使用 `./bkcec status XXX模块` 来确认
- 检查 cmdb 日志

```
检查 /data/bkce/log/cmdb/cmdb_datacollection.INFO文件
出现 ccapi.go93] fail to get configure, will get again, 表示不正常
返回 hostsnap.go:xxx] master check : iam still master, 表示正常
检查 cmdb_datacollection.ERROR 文件，确认是否出现 err:fail to connect zookeeper. err: looku
p zk.service.consul等
如果日志出现“subscribing channel 2_snapshot”后没有 subChan Close，那么表明收数据协程正常工作
如果上述条都正常，但没有“handle xx num mesg, routines xx”，说明通道里没数据，请到 redis 里 sub
scribe ${biz}_snapshot 确认通道是否没数据，参考如下检查 redis 数据方法
```

- gse agent 采集端排查

```
# 检查进程是否存在，basereport 进程存在且唯一
Linux: ps -ef | grep basereport
Windows: tasklist | findstr basereport

# 若进程不存在，手动启动，检查启动失败的原因
Linux: cd /usr/local/gse/plugins/bin && ./basereport -c ../etc/basereport.conf
Windows(cygwin): cd /cygdrive/c/gse/plugins/bin/ && ./basereport -c ../etc/basereport.c
onf
Windows(无cygwin) : cd C:/gse/plugins/bin/ && start.bat basereport

# 检查数据上报连接，有正常 ESTABLISHED 的连接则 ok
# 若存在 proxy，登陆 proxy 机器：检测 58625 端口同上
Linux netstat -antp | grep 58625 | grep ESTABLISHED
Windows netstat -ano | grep 58625
```

- gse 服务端排查

```
# 登陆 GSE 后台服务器，检测 gse_data 是否连上 9092 端口：
Linux: lsof -nP -c dataWorker | grep :9092
Windows: netstat -ano | grep 9092

# 看有没有 gse_data 的 pid 开头命名的日志。若有，tail 查看日志内容
datapid=$(pgrep -x dataWorker)
ls -l /data/bkce/public/gse/data/${datapid}*
```

- 检查 kafka

- 登陆任意 KAFKA 机器：查看 KAFKA 最新数据，等待 1 分钟查看是否有数据。如果有数
据，在最后一行命令后加上 `| grep $ip` \$ip 用无快照数据的 ip 替换，再次查看是否有数
据

```
# 登录到 kafka 所在的机器上
$ source /data/install/utils.fc
```

```

/data/bkce/service/zk/bin/zkCli.sh -server zk.service.consul:2181 get /gse/config/etc/d
ataserver/data/1001
# 确认存在 topic
zkaddr=`cat /data/bkce/service/kafka/config/server.properties | grep common_kafka | cut
-d '=' -f 2`
topic=`bash /data/bkce/service/kafka/bin/kafka-topics.sh --list --zookeeper $zkaddr|gre
p ^snap`

# 查看 topic 中的最新数据
bash /data/bkce/service/kafka/bin/kafka-console-consumer.sh --zookeeper $zkaddr --topic
$topic
# 每隔一分钟会上报数据，有数据上报则表示正常

```

• 检查 bkdata

```

# 快照数据对应 bkdata, databus 的 redis 任务，需确认 databus 状态下的 redis 任务是否存在
$ bash /data/bkce/bkdata/dataapi/bin/check_databus_status.sh
# 检查是否存在如下 redis 任务
=====REDIS=====
  % Total      % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100    25    100    25    0    0    142    0 --:--:-- --:--:-- --:--:--   142
redis_1001_2_snapshot
{"name":"redis_1001_2_snapshot","connector":{"state":"RUNNING","worker_id":"x.x.x.x:100
53"},"tasks":[{"state":"RUNNING","id":0,"worker_id":"x.x.x.x:10053"}]}
=====

# 若上述任务不存在，可能在初始化 bkdata 数据时异常，可以采用如下方法重新创建，先确认 init_bkdata_s
napshot 是否存在
$ ll /data/bkce/.init_bkdata_snapshot
$ rm -f /data/bkce/.init_bkdata_snapshot
$ deactivate
$ source /data/install/utils.fc
$ init_bkdata_snapshot
# 再根据上面的重新确认是否有 redis 任务

```

• 检查 redis 通道

```

# 此步主要检查 redis 内是否有快照数据，在 redis 服务器上
source /data/install/utils.fc
$ redis-cli -h $REDIS_IP -p $REDIS_PORT -a $REDIS_PASS
10.X.X.X:6379> AUTH "REDIS密码"
OK
10.X.X.X:6379> SUBSCRIBE 2_snapshot
Reading messages... (press Ctrl-C to quit)
1) "subscribe"
2) "2_snapshot"
3) (integer) 1
1) "message"
2) "2_snapshot"
3) "{\"localTime\": \"2018-08-15 11:18:00\", \"data\": \"{\\\"beat\\\":{\\\"address\\\"
:

```

CMDB 无主机信息

表象：CMDB 内主机自动发现信息为空

原因

1. gse 安装异常或 gse 数据初始化不对
2. gse_agent 安装异常

思路办法

- 1: 参考 GSE 数据初始化失败解决方法，需要更新 GSE 版本中的初始化程序文件 on_migrate 和 parse_bizid，路径 `/data/bkce/gse/server/bin/`
- 2: 重装 gse_agent

作业平台常见问题

JOB 启动失败

job 启动失败常见原因：

- 若有日志，根据日志定位
 - 证书是否匹配
 - RabbitMQ 连接是否失败
 - Redis 连接是否失败
 - Mysql 连接是否失败
 - 确定 IP/密码/用户名是否存在及正确
- consul 是否路由正确
- 端口问题 `/data/bkce/etc/job.conf` 配置项内的端口是否有冲突
- 环境问题导致没有日志，打开 `/data/bkce/job/job/bin/jo.sh` 的 `NOHUPLOG=job_jvm_console.log` 配置，再重启确认 JOB 的日志目录下的日志文件 `job_jvm_console.log` 进行确认
- 确认 License 合法及可以连接

JOB 作业一直等待执行

在执行作业时，步骤状态一直为等待执行，解决方法如下：

- RabbitMQ 连接异常，作业启动信息收不到，请检查 RabbitMQ，并重启 Job 进程尝试恢复

- 检查项: ijobs.amqp.addresses, ijobs.amqp.username, ijobs.amqp.password。其中 ijobs.amqp.username, ijobs.amqp.password 与 app.code 和 app.secret 配置项相同, 并且 RabbitMQ 会在创建帐号后并授权 vhost 名为 bk_job 的使用权限

JOB 一直跳转登陆页面

此问题一般为 PaaS 登陆接口通信失败, 解决方法如下:

- 检查 `/data/bkce/etc/job.conf` 中 bk.paas.host 配置项, 确认地址是否可以连通, 并确认 PaaS 是否正常
- 检查 `/data/bkce/etc/job.conf` 中 bk.paas.host 配置项 app.code 和 app.secret 配置是否正确, 否则访问 PaaS 接口会鉴权试下而无法登录

JOB 无执行日志

在排除用户的脚本本身就不输出日志的正常情况, JOB 出现无日志的情况有很多种,

第一种是 JOB 本身问题, 与 GSE 无关, 当 error.log 出现 Table has no partition for value xxxx 这种错, 就是 JOB 本身的问题, 原因:

JOB 对日志数据库表进行表分区, 并且要求 MYSQL 中启动事件调度功能, JOB 默认会在版本升级时自动启用这个功能, 在 JOB 的版本 1.2.49 之前, 蓝鲸出厂默认的 MySQL 配置中没有开启这个功能, 所以在 MySQL 被重启后, 这个事件功能被关闭, 随着时间流转, 分区不够用了, 会出现这个错误。

解决办法可根据情况选择:

1.不想升级蓝鲸版本

- 在 MySQL 配置文件 my.cnf 中的[mysqld]部分添加以下内容 event_scheduler=ON 并重启 MySQL
- 请联系蓝鲸人员提供一个 JOB 的临时启用 event 的 SQL

2.升级蓝鲸到最新版本。

JOB 耗时时间长并无执行日志: Execution result log always

此问题出现在 job.log 文件内, 并一直打印 `redis reply is not string` 这种情况, 原因如下:

- 此情况属于 GSE 的任务管道或者 Agent 的原因导致的无日志, 这种情况需要排查 GSE 服务是否正常, 以及执行机器 Agent 进程是否正常, 有可能在执行中被杀掉, 导致长时间无上报执行日志给服务器

- 最终结果为 JOB 的作业执行日志出现 `Execution result log always`，这个是在脚本作业的超时时间范围内如果连接 10 分钟内一直没有收到任何日志信息，JOB 会触发保护，强制终止作业并返回上述信息给用户

JOB 脚本耗时长并失败：Script log timed out

出现此问题可能两种原因：

- 用户业务脚本问题：执行过于耗时的脚本，并且超过了设置的脚本超时时间，一般默认是 1000 秒，在此时间基础上多 1-10% 以内是正常的。若因为此原因，解决方法：修改耗时脚本或者修改脚本超时时间
- GSE Agent 的问题：作业最终超时时间是在用户设置的超时时间再 +20%，若用户设置了 1000 秒超时，但脚本最终在 1200 秒以上出现超时，则表示这个任务是因为 Agent 原因导致长时间无响应，最终 JOB 容错了 20% 的时间(1000+200 秒)而强行触发终止。解决方法：重启 GSE Agent 再重试

JOB 连接 GSE 失败

这种报错，说明 JOB 连接 gse_task 异常 可能的原因如下：

- gse_task 进程异常，48669 端口(task 提供给 JOB 通信的端口)未监听
- JOB 配置文件 `etc/job.conf` 里配置的 gse.taskserver.ip 的值无法连通
- 证书问题，会爆出 ssl 字样的错误信息

JOB 无法发现 Agent

这类报错说明 JOB 连接 task 是正常，但是 agent 状态异常

- agent 安装问题，失败，进程未正常启动
- agent 到 gse_task 的 48533 端口未建立 tcp 连接
- agent 证书和 gse_task 的不匹配，会爆 ssl 字样错误信息
- 云区域 ID 不匹配

分发文件，执行历史报 "redis no result found"

- 问题现象

节点名称快速分发文件-2019814102926642节点类型文件传输服务器帐户root总时间(s)0 1.112

节点状态执行失败开始时间2019-08-14 10:29:49结束时间2019-08-14 10:29:51

任务日志错误(1)

输入搜索内容搜索日志导出执行日志

IP过滤:完整日志

子网名称IP耗时(s)

10.1.2.820.000

第 1 页 / 共 1 页 每页显示 5 条 共 1 条

【2019-08-14 10:29:51.0】redis no result found

- 解决方案
 - 1.gse 或者 redis 进程状态异常。
gse 或者 redis 异常查看相应日志文件。/data/bkce/logs/gse/、/data/bkce/logs/redis。
 - 2.登陆至 gse server 模块所在机器上，ps -ef |grep gse_agent 确认该机器上成对出现 gse_agent 进程。如无，执行 /usr/local/gse/agent/bin/gsectl start 手动拉起 gse_agent。
 - 3.gse 和 redis 需要同在一台机器上部署。
 - 4.蓝鲸 server 端 agent 需确保正常 ps -ef |grep gse_agent 进程成对出现(gse所在机器)
bash root 19467 1 0 Nov04 ? 00:00:00 ./gse_agent -f /usr/local/gse/agent/etc/agent.conf root 19469 19467 0 Nov04 ? 00:08:05 ./gse_agent -f /usr/local/gse/agent/etc/agent.conf

JOB 平台错误代码

错误码	源	目标	描述
1210001	JOB	GSE	GSE TaskServer 不可用
1210101	JOB	GSE	当前证书服务不可用，请检查 license_server!
1250001	JOB	Redis	Redis 服务不可：IP 不对或者配置错误
1250002	JOB	Redis	Redis 服务内存满或者其他问题：内存不足
1259001	JOB	NFS	NFS 存储不可用
1252001	JOB	MYSQL	数据库异常
1255001	JOB	MQ	Rabbit MQ 不可用或者连接不上
1211001	JOB	CMDB	CMDB 服务状态不可达：地址配置错误
1211002	JOB	CMDB	CMDB 接口返回数据结构异常。一般是被网关防火墙重定向返回非 JSON 协议内容
1211121	JOB	CMDB	蓝鲸业务下的 Git 模块没有 IP(包管理)
1213001	JOB	PAAS	PAAS 服务不可达 - 地址配置错误或者地址无法正确解析
1213002	JOB	PAAS	PaaS 接口返回数据结构异常。一般是被网关防火墙重定向返回非 JSON 协议内容

管控平台常见问题

GSE initdata 失败

表象：在 `./bkcec initdata gse` 时，会出现如下 2 种报错

1. `parse cc response error`
2. `parse dataid error`

```
# 提示解析 CC 响应失败，或者解析 dataid 失败
$ parse dataid failed, [{"server_id": -1, "data_set": "snapshot", "partition": 1, "cluster_index": 0, "biz_id": 2, "msg_system": 1}, {"server_id": -1, "data_set": "snapshot", "partition": 0, "cluster_index": 1, "biz_id": 2, "msg_system": 4}]
migrate failed for gse (server)
```

出现版本：4.1.X

原因

1. GSE 对应初始化数据的程序及脚本不正确；
2. cmdb 未正常安装，或 cmdb-nginx 启动不正常/未启动；

解决办法

原因 1

- 需要更新 GSE 版本中的初始化程序文件 `on_migrate` 和 `parse_bizid`，路径 `/data/bkce/gse/server/bin`

```
bash [root@rbtnd1 /data/install]# md5sum
/data/bkce/gse/server/bin/on_migrate /data/bkce/gse/server/bin/parse_bizid
addc6eeec6e72e73cc330cc7fa69e9b4 /data/bkce/gse/server/bin/on_migrate
7ba79e36b731ef9678a3b8bfb41dc2ef /data/bkce/gse/server/bin/parse_bizid *用户
```

操作方法：

- 把这 2 个文件放到中控机 `/data/src/gse/server/bin` 下面；
- 在中控机，进行如下操作

```
bash source /data/install/utlis.fc cd /data/install ./bkcec sync all ./bkcec
stop gse ./bkcec install gse 1 ./bkcec initdata gse
```

原因 2

- 确定 cmdb 上 nginx 是否安装，若没安装，根据用户的设置源，让用户安装 nginx，同时需确认 `/etc/nginx/nginx.conf` 配置文件是否包含 `/data/bkce/etc/nginx/*.conf`

- 需确认 cmdb 的状态，包括 cmdb-nginx，确定 `./bkcec status cmdb` 结果里面 `cmdb-nginx` 的状态不是 EXIT
- 测试 `curl -v http://cmdb.service.consul`，是否有 `502 Bad Gateway` 错误返回
- 检查失败时，生成在 `/tmp/gse.tmp` 的文件，不能出现有 `502 bad gateway` 的错误提示，若出现，注意检查 cmdb 的 8029 端口是否 OK

GSE 启动失败

内网 IP 自动获取不对时

LAN_IP 表示 GSE 服务器真实可用的内网 IP(ip addr 输出查看)，需要根据实际 IP 替换 修改以下配置文件项，新增相关配置，注意 json 格式，逗号问题。

- data.conf

```
"datasvrip": "LAN_IP",
```

- task.conf

```
"tasksvrip": "LAN_IP",
"tasksvrthriftip": "LAN_IP",
"tasksvrtrunkip": "LAN_IP",
```

- dba.conf

```
"servers": [{"ip": "LAN_IP", "port": 58817}],
```

- btsvr.conf

```
"filesvrthriftip": "LAN_IP",
"btServerOuterIP": [{"ip": "LAN_IP", "port": 59173}],
"btServerInnerIP": [{"ip": "LAN_IP", "port": 59173}],
Copy
```

- api.conf

```
"cacheApiAddr": [{"ip": "LAN_IP", "port": 59313}],
Copy
```

- agent.conf

```
"agentip": "LAN_IP",
```

GSE agent 状态异常

节点管理 SaaS 或 Job 显示 agent 状态异常:

1. 检查对应 ip 机器上的 gse_agent 进程是否正常 `ps -ef |grep gse_agent`
2. 检查 gse_agent 的 48533 连接是否正常
3. 检查 gse_agent 与 gse server 的证书是否匹配
4. 检查该 ip 在 CC 上的业务及云区域 id 是否正确
5. 检查 gse_api 日志, 查看启动时是否有 `UPDATE_REDIS_FAILED` 信息, 若有则重启 gse_api

直连的 agent

- 查看 agent 机器上的 gse_agent 进程是否成对出现
- 查看是否和 gse_task 的48533端口建立链接: `netstat -antp | grep :48533`

```
[root@nginx-1 ~]# netstat -antp |grep :48533
tcp        0      0 10.0.1.2:35544      10.0.1.226:48533    ESTABLISHED 26714/.
/gse_agent
```

- 登陆到第一步显示链接的 gse_task 的 IP(10.X.X.X), 继续查看链接: `netstat -antp | grep :48533 | grep 10.0.1.2` 确认 gse_task 上看到的 ip 和 agent 的 ip 一致。若不一致, 可能 agent->gse_task 时发生了 NAT 转换

Proxy 下的 agent

- 查看 agent 是否和 proxy(gse_agent)的 48533 建立链接: `netstat -antp | grep :48533`
- 和直连 agent 的排查同理, 到 proxy 上查看建立链接的 ip 是否一致

GSE 相关组件自动获取 IP 失败

下文 `__LAN_IP__` 均表示 GSE 服务器真实可用的内网 IP

gse 相关组件默认会自动尝试获取内网网卡 ip 去监听, 但是有些网卡复杂的情况下 会监听到不正确的网卡时, 可以尝试修改配置文件里的 IP 来解决

如果配置文件不包含以下配置段, 可以自己新增相关配置, 要注意配置文件都是 json 格式, 留意逗号问题

- GSE 后台服务

GSE 后台服务的配置文件路径默认在 `/data/bkce/etc/gse/` 下

譬如 gse_task 进程对应的配置是 task.conf，修改后，使用 `cd /data/bkce/gse/server/bin/ && ./gsectl restart task` 来重启。其他进程以此类推。

- data.conf

```
"datasvrnip":"LAN_IP",
```

- task.conf

```
"tasksvrnip":"LAN_IP",  
"tasksvrthirftip":"LAN_IP",  
"tasksvrtrunkip":"LAN_IP",
```

- dba.conf

```
"servers":[{"ip":"LAN_IP","port":58817}],
```

- btsvr.conf

```
"filesvrthriftp":"LAN_IP",  
"btServerOuterIP":[{"ip":"LAN_IP","port":59173}],  
"btServerInnerIP":[{"ip":"LAN_IP","port":59173}],
```

- api.conf

```
"cacheApiAddr":[{"ip":"LAN_IP","port":59313}],
```

- Gse Proxy 模块

proxy 的配置文件在 `/usr/local/gse/proxy/etc/` 下，修改配置后使用 `cd /usr/local/gse/proxy/bin/ && ./gsectl restart` 来重启。

- agent.conf

```
"agentip":"LAN_IP",  
"proxylistenip":"LAN_IP",
```

日志采集问题排查

查看源日志是否有更新

确保采集的数据源有日志持续输出

注意：文件内容不可直接清空，文件轮转可采用删除或者移动 MV。

检测日志采集器进程

```
ps -ef | grep unifytlogc  
Copy
```

若进程不存在，进入采集器目录，手动尝试启动采集器查看是否有错误信息

```
cd /usr/local/gse/plugins/bin/  
./unifytlogc -c ../etc/unifytlogc.conf  
Copy
```

检查连接

agent 机器：有正常 ESTABLISHED 的连接则 ok

- Linux `netstat -antp | grep 58625 | grep ESTABLISHED`
- Windows `netstat -ano | grep 58625`

若存在 proxy，登陆 proxy 机器：检测 58625 端口同上。

- Linux `netstat -tnp | grep 58625`

登陆 GSE 后台服务器，检测 gse_data 是否连上 9092 端口：

- Linux: `lsof -nP -c dataWorker | grep :9092`
- Windows: `netstat -ano | grep 9092`

检测日志采集配置

```
cat /usr/local/gse/plugins/etc/unifytlogc.conf  
Copy
```

找到对应任务的 dataid，(在 tlogcfg->files->dataid)

```
{  
  "tlogcfg": [  
    {  
      "files": [  
        {  
          "dataid": 123,  
          "condition": [  
            ]  
          }  
        ],  
      "dataid": 123,  
      "isNotPack": 0,  
      "beJson": 1,  
      "ignore_file_end_with": "...",  
      "private": {  
        "_plat_id_": 1  
      }  
    }  
  ]  
}
```

```

    },
    "dataset": "datatest",
    "clear_file_cycle": 157680000,
    "file": "/tmp/datatest/ddd.log",
    "field_sep": "|",
    "log_type": "logbyline"
  }
],
"ipc_socket": "/var/run/ipc.state.report",
"log_path": "/var/log/gse",
"data_path": "/var/lib/gse",
"pidfile_path": "/var/run/gse/unifytlogc.pid",
"log_level": "ERROR"
}
Copy

```

在 ZK_IP 上查看 dataid 对应的 topic

\$dataid 替换为上一步查询出来的 dataid

```

/data/bkce/service/zk/bin/zkCli.sh -server zk.service.consul:2181 (ip通常为本机内网IP)
get /gse/config/etc/dataserver/data/$dataid
Copy

```

取出 data_set 和 biz_id 两个字段，合并在一起 例如：

```

{"server_id": -1, "data_set": "datatest", "partition": 1, "cluster_index": 0, "biz_id":
4, "msg_system": 1}
Copy

```

topic 为 datatest4

检测 kafka 数据

在 KAFKA 机器上

```

source /data/install/utils.fc
cd /data/bkce/service/kafka
topic=<上面查询的结果>
sh bin/kafka-console-consumer.sh --bootstrap-server $LAN_IP:9092 --topic $topic
Copy

```

等待有没有新的数据产生

若 kafka 没有数据，查看 gse_data 日志

登陆 GSE Server 的机器，看有没有 gse_data 的 pid 开头命名的日志。若有，tail 查看日志内容

```

datapid=$(pgrep -x dataWorker)
ls -l /data/bkce/public/gse/data/${datapid}*
Copy

```


GSE 端口列表

- 直连 agent 和 GSE 之间的互通策略

源地址	目标地址	协议	端口	用途
agent	zk	TCP	2181	获取配置
agent	gse_task	TCP	48533	任务服务端口
agent	gse_data	TCP	58625	数据上报端口
agent	gse_btsvr	TCP	59173	bt 传输
agent	gse_btsvr	TCP,UDP	10020	bt 传输
agent	gse_btsvr	UDP	10030	bt 传输
gse_btsvr	agent	TCP,UDP	60020-60030	bt 传输
gse_btsvr	gse_btsvr	TCP	58930	bt 传输
gse_btsvr	gse_btsvr	TCP,UDP	10020	bt 传输
gse_btsvr	gse_btsvr	UDP	10030	bt 传输
agent	agent	TCP,UDP	60020-60030	bt 传输
agent			监听随机端口	bt 传输，可不开通
gse_btsvr			监听随机端口	bt 传输，可不开通

- proxy 和 GSE 之间的互通策略

源地址	目标地址	协议	端口	用途
proxy(gse_agent)	gse_task	TCP	48533	任务服务端口
proxy(gse_transit)	gse_data	TCP	58625	数据上报端口
proxy(gse_btsvr)	gse_btsvr	TCP	58930	bt 传输
proxy(gse_btsvr)	gse_btsvr	TCP,UDP	10020	bt 传输
proxy(gse_btsvr)	gse_btsvr	UDP	10030	bt 传输
gse_btsvr	proxy(gse_btsvr)	TCP	58930	bt 传输
gse_btsvr	proxy(gse_btsvr)	TCP,UDP	10020	bt 传输
gse_btsvr	proxy(gse_btsvr)	UDP	10030	bt 传输
proxy(gse_btsvr)	proxy(gse_btsvr)	TCP	58930	bt 传输(同一子网)
proxy(gse_btsvr)	proxy(gse_btsvr)	TCP,UDP	10020	bt 传输(同一子网)
proxy(gse_btsvr)	proxy(gse_btsvr)	UDP	10030	bt 传输(同一子网)
proxy(gse_opts)	gse_ops	TCP	58725	ping告警数据上报端口
proxy(gse_agent)			监听随机端口	bt 传输，可不开通
proxy(gse_btsvr)			监听随机端口	bt 传输，可不开通

- proxy 下 agent 和 proxy 之间的互通策略

源地址	目标地址	协议	端口	用途
agent	proxy(gse_agent)	TCP	48533	任务服务端口
agent	proxy(gse_transit)	TCP	58625	数据上报端口
agent	proxy(gse_btsvr)	TCP	59173	bt 传输
agent	proxy(gse_btsvr)	TCP,UDP	10020	bt 传输
agent	proxy(gse_btsvr)	UDP	10030	bt 传输
proxy(gse_btsvr)	agent	TCP,UDP	60020-60030	bt 传输
agent	agent	TCP,UDP	60020-60030	bt 传输(同一子网)
agent			监听随机端口	bt 传输, 可不开通

数据平台常见问题

initdata bkdata 失败

表象：在 `./bkcec initdata bkdata` 时，会出现类似如下报错

```
ERROR:init_snapshot_config (databus.tests.DatabusHealthTestCase)
ConnectionError:HTTPConnectionPool(host='databus.service.consul', port=10052):Max retri
ed exceeded with url: /connectors (Caused by NewConnectionError('<requests.packages.url
lib3.connection.HTTPConnection object at 0x7f90939a7110>: Failed to establish a new con
nection: [Errno -2] Name or service not known',))
```

思路方法

“

升级用户：确认在升级前，若需要初始化 bkdata 数据，先删除 bkdata 服务
器 `/data/bkce/.dataapi_snaphost` 和 `/data/bkce/.init_bkdata_snapshot` 文件

注意：如下操作，要求安装路径为 `/data/bkce`，源路径为 `/data/install`

”

1. 确认 databus 日志是否有 Exception 的错误，示例如下

```
bash grep -nE "Exception" /data/bkce/logs/bkdata/databus_etl.log
/data/bkce/logs/bkdata/databus_tsdb.log
199:org.apache.kafka.common.errors.wakeupException 12:Exception in thread "main"
org.apache.kafka.common.config.configException: Invalid value for configuration
rest.port: Not a number of type INT
```

1. 若有 1 的错误存在，确认端口配置，10054 端口是否返回，若为空需要添加上

```
bash grep -nE port /data/bkce/bkdata/databus/conf/tsdb.cluster.properties
16:cluster.rest.port=10054
```

1. 确认 `/data/bkce/bkdata/dataapi/bin/check_databus_status.sh` 状态，不能出现有 `Failed connect to databus.service.consul:10054; connection refused` 或者 `JSON object could be decoded` 错误输出
2. 确认 kafka，若社区版为 3 台部署的，必须返回[1, 2, 3]才正常，示例如下
若 `brokers ids` 不为[1, 2, 3]，可能存在 `/data/bkce/public/kafka/.lock` 文件，有的话，删除此文件，再重新使用 `./bkcec stop kafka` 和 `./bkcec start kafka` 重启 kafka，重启完再次确认状态
``bash [root@rbtnode1 /data/install]# /data/bkce/service/zk/bin/zkCli.sh -server zk.service.consul:2181 ls /common_kafka/brokers/ids Connecting to zk.service.consul:2181 log4j:WARN No appenders could be found for logger (org.apache.zookeeper.ZooKeeper). log4j:WARN Please initialize the log4j system properly. log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info. WATCHER:: WatchedEvent state:SyncConnected type:None path:null [1, 2, 3] ``
3. 若上述 1, 2, 3, 4 均 OK，采用如下方法重新进行初始化数据
``bash

在 bkdata 机器：

```
rm -rf /data/bkce/.dataapi_snaphost /data/bkce/.init_bkdata_snapshot
```

在中控机停掉 bkdata

```
./bkcec stop bkdata
```

在中控机重新初始化 bkdata，此处正常会等很久，出现很多 **add xxx connector** 的输出，若出现的话，等它全部正常结束

```
./bkcec initdata bkdata [10.X.X.X]20180821-095319 120 migrate bkdata(dataapi) done
[10.X.X.X]20180821-095320 78 starting bkdata(ALL) on host: 10.X.X.X [10.X.X.X]20180821-
095334 85 going to init snapshot data. this may take a while. ....
http://databus.service.consul:10052/connectors init data of snapshot and components add etl
connector of 2_uptimecheck_heartbeat add etl connector of 2_uptimecheck_http add etl
connector of 2_redis_cpu add etl connector of 2_redis_client ..... add tsdb connector of
2_mysql_performance add tsdb connector of 2_mysql_rep
```

等待上述完成，再启动 bkdata

```
./bkcec start bkdata ``
```

如何确认 `initdata bkdata` 的结果是正常

1. 确认在 `initdata bkdata`，最后有如下的正常输出

```
``bash update reserved dataids DONE [10.X.X.X]20180821-095319 120 migrate bkdata(dataapi)
done [10.X.X.X]20180821-095320 78 starting bkdata(ALL) on host: 10.X.X.X [10.X.X.X]20180821-
095334 85 going to init snapshot data. this may take a while. ....
http://databus.service.consul:10052/connectors init data of snapshot and components add etl
connector of 2_uptimecheck_heartbeat add etl connector of 2_uptimecheck_http add etl connector of
2_redis_cpu add etl connector of 2_redis_client ..... add tsdb connector of 2_mysql_performance
add tsdb connector of 2_mysql_rep add tsdb connector of 2_apache_net add tsdb connector of
2_apache_performance .
```

Ran 1 test in 156.952s

OK ``

1. 确认在 bkdata 服务器上，`check_databus_status.sh`，不能出现有 `Failed connect to databus.service.consul:10054; connection refused` 或者 `JSON object could be decoded` 错误输出。正常的输出示例如下

```
``bash [root@rbtnode1 /data/install]# /data/bkce/bkdata/dataapi/bin/check_databus_status.sh
=====TSDB===== % Total % Received % Xferd Average Speed Time Time
Time Current Dload Upload Total Spent Left Speed 100 773 100 773 0 0 14599 0 --:--:-- --:--:-- --:--:--
14865 tsdb_2_system_cpu_summary {"name":"tsdb_2_system_cpu_summary","connector":
{"state":"RUNNING","worker_id":"10.X.X.X:10054"},"tasks":
[{"state":"RUNNING","id":0,"worker_id":"10.X.X.X:10054"}]}

=====MYSQL===== % Total % Received % Xferd Average Speed Time Time
Time Current Dload Upload Total Spent Left Speed 100 27 100 27 0 0 1118 0 --:--:-- --:--:-- --:--:--
1125 jdbc_2_ja_gse_proc_port {"name":"jdbc_2_ja_gse_proc_port","connector":
{"state":"RUNNING","worker_id":"10.X.X.X:10051"},"tasks":
[{"state":"RUNNING","id":0,"worker_id":"10.X.X.X:10051"}]}

=====ETL===== % Total % Received % Xferd Average Speed Time Time
Time Current Dload Upload Total Spent Left Speed 100 1040 100 1040 0 0 166k 0 --:--:-- --:--:-- --:--:--
- 169k etl_1001_2_system_cpu_summary
{"name":"etl_1001_2_system_cpu_summary","connector":
```

```
{"state":"RUNNING","worker_id":"10.X.X.X:10052"},"tasks":  
[{"state":"RUNNING","id":0,"worker_id":"10.X.X.X:10052"}]}
```

bkdata 常见问题排查

- zookeeper 配置错误，数据无法发送
- agent 制定 IP 有误，和 CMDB 无法管理
- crontab 未设置，未启动定时同步 IP 到业务映射，导致监控无数据
- consul 服务异常，导致 kafka 及其他 consul 模块无法解析
- tsdb-proxy 未启动
- RabbitMQ 密码错误，celery 启动失败
- kafka broker 启动失败，或者节点缺失
- cmdb 未启动，初始化 dataapi 失败
- agent 机器时间需要和蓝鲸 server 机器时间同步

平台其他常见问题

变更域名

- 修改 globale.env 中的域名配置信息。
- 修改 每台机器上的 /etc/hosts 匹配上新的域名
- 修改完成后按如下命令顺序执行：

```
# V4.0 版本及以前的需要执行，V4.1 以后的无需执行  
./bkcec clean cron  
  
./bkcec sync common  
echo fta bkdata job cmdb paas nginx | xargs -n 1 ./bkcec stop  
echo fta bkdata job cmdb paas nginx | xargs -n 1 ./bkcec render  
echo nginx paas cmdb job bkdata fta | xargs -n 1 ./bkcec start  
  
# V4.0 版本及以前的需要执行，V4.1 以后的无需执行  
./bkcec install cron
```

如果有安装 SaaS，到 [开发者中心-Smart应用-已上线](#) 的 SaaS 操作栏里的部署按钮，重新一键部署 SaaS

装蓝鲸后主机名被自动修改了可以改回去吗

主机名可以改回去，但是要注意 /etc/hosts 里自动添加的 rbtnode1 的映射不能去掉，rabbitmq 组件依赖它解析 NODENAME

root 邮箱每分钟都收到 mail 告警

内容包含类似 “xxxx is running” 的信息

这是因为 crontab 里配置了进程监控，但是没有重定向 STDOUT 和 STDERR，当前版本可以自己手动添加：

比如将

```
* * * * * export INSTALL_PATH=/data/bkce; /data/bkce/bin/process_watch mysql
```

改为

```
* * * * * export INSTALL_PATH=/data/bkce; /data/bkce/bin/process_watch mysql &>/dev/null
```

SAAS 部署常见问题

SaaS 部署提示超时

可修改 paas_agent 的 paas_agent_config.yaml，将 EXECUTE_TIME_LIMIT 调大，重启 paas_agent 即可

```
$ /data/bkce/etc/paas_agent_config.yaml
$ /data/bkce/paas_agent/paas_agent/etc/paas_agent_config.yaml
EXECUTE_TIME_LIMIT: 300
```

节点管理部署问题

```
-----create virtualenv for bk_nodeman-----
-----create virtualenv success-----
-----Extract app_code for bk_nodeman-----
-----Extract app_code success-----
-----yum install-----
error: rpmdB: BDB0113 Thread/process 3864/139771538343936 failed: BDB1507 Thread died i
n Berkeley DB library
error: db5 error(-30973) from dbenv->failchk: BDB0087 DB_RUNRECOVERY: Fatal error, run
database recovery
error: cannot open Packages index using db5 - (-30973)
error: cannot open Packages database in /var/lib/rpm
CRITICAL:yum.main:
```

```
Error: rpmdb open failed
-----yum gcc python-devel openssl-devel libffi libffi-devel fail-----
```

出现这种问题，原因为 rpm 数据库损坏

可以尝试重启机器，再使用 `yum list all`，若能够列出软件包则 OK。还有问题参考下面的解决方法

```
$ yum list all
已加载插件: fastestmirror
Loading mirror speeds from cached hostfile
.....
```

解决方法(重新构建 rpm 数据库)

```
$ cd /var/lib/rpm
$ ls
Basenames __db.001 __db.003 Group Name Packages Requirename Sigmd5
Conflictname __db.002 Dirnames Installtid Obsoletename Providename Sha1header Triggerna
me
rm -rf __db.*
$ rpm --rebuilddb
$ yum clean all
```

节点管理常见问题

部署的时候报 Unknown column

```
(1054, u"Unknown column 'miya_ip.desc' in 'field list'")
Traceback (most recent call last):
```

- 原因: 因为升级数据初始化是根据 3.0 以上版本的 DB 结构做数据迁移的，低于 3.0 的 agent 版本原有的机器表中不存在对应的字段，所以会报错
- 解决办法：
 1. 先升级 agent 安装到最新版本
 2. 重建数据库(drop database bk_nodeman; create database bk_nodeman)
 3. 重新部署节点管理

检测 Agent 状态和版本异常

表象：在 step 6 检测 Agent 状态和版本过程一直报如下错误

```
[2018-08-23 11:11:18]: check agent status (21/60)(bk_agent_alive: 1->alive, 0->dead):0
[2018-08-23 11:11:19]: check agent status (22/60)(bk_agent_alive: 1->alive, 0->dead):0
[2018-08-23 11:11:21]: check agent status (23/60)(bk_agent_alive: 1->alive, 0->dead):0
[2018-08-23 11:15:20]: CELERY_TASK_TIMEOUT: install script running timeout(600s), install failed, please connect us
```

解决方法:

此种问题, 常见于云区域, 请确认 GSE 和 Agent 的策略 OK

安装 Proxy 报错

```
root@localhost:~# proxy agents
```

>>step 3 执行安装脚本

```
[2018-08-22 19:19:46]: prepare to call real install script ["export LC_ALL=C; /tmp/agents/agent_setup_pro.sh -m proxy -i 2 -w 180.167.73.103 -l 10.7.49.225 -g 180.167.73.103:9999"]
[2018-08-22 19:19:47]: export LC_ALL=C; /tmp/agents/agent_setup_pro.sh
-m proxy -i 2 -w 180.167.73.103 -l 10.7.49.225 -g 180.167.73.103:9999
/tmp/agents/agent_setup_pro.sh: line 1: syntax error near unexpected token `newline'
/tmp/agents/agent_setup_pro.sh: line 1: `
[root@localhost_BKproxy agents]$
[2018-08-22 19:19:47]: start install script success, ret code is 【SUCCESS】
[2018-08-22 19:21:27]: SOCKET_TIMEOUT: socket timeout, install failed , please connect us.
```

解决方法:

1. 修改一下 nginx 的配置, 把外网 ip 绑定上
2. 登录到 nginx 的机器

```
vi /data/bkce/etc/nginx/miniweb.conf
server_name x.x.x.x(内网ip) x.x.x.x(外网ip);
```

1. 退出到中控机,重启 nginx:

```
./bkcec stop nginx
./bkcec start nginx
```

Windows 没有 cygwin 的无法连接

参考 [Windows 开 139, 445 端口](#)

原来使用 agent 安装, 在部署节点管理后, 主机信息没有同步到节点管理

原因: 节点管理低于 1.0.52 的版本还没有数据迁移功能。数据迁移动作会在第一次安装节点管理

SaaS 时从 agent-setup SaaS 迁移过来。升级蓝鲸企业版时，由于节点管理的数据库已经存在，故没有进行迁移动作

解决办法：

- 先升级 agent 安装到最新版本(1.0.52)
- 重建数据库 `drop database bk_nodeman; create database bk_nodeman`
- 重新部署节点管理最新版本

节点管理中部署 agent 失败, 详情日志中报错: 组件调用异常: unknown component's error: the response is None(code=UNKOWN ERROR)

>>step 4 安装脚本执行完毕

>>step 5 注册主机到CMDB

[2018-06-29 11:46:41]: REGISTER_FAILED: register to cmdb failed: 组件调用异常:unknown component's error: the response is None(code=UNKNOWN ERROR)

可能原因: bk_nodeman 没有添加到 paas 白名单

解决办法: 登陆中控机执行

```
source utils.fc
_add_app_token bk_nodeman $(app_token bk_nodeman)
```

注册主机到 CMDB 失败: 0 行[['import_from']] 数据校验参数不通过(社区版4.1及其之前版本)



原因: CMDB 中主机属性配置信息错误

解决办法:

- 打开 CMDB, 进入 [后台配置] -> [模型管理]
- 在视图区域点击[主机], 点击 [模型配置] 标签页

在字段列表中找到 字段名 为 录入方式(import_from)的行, 并点击展开如下

枚举

录入方式(import_from)

字段配置

中文名 *

录入方式

英文名 *

import_from

单位

请输入单位

提示语

请输入提示语

选项

类型

枚举

是否可编辑

☐

☐

1

excel

☐

2

agent

☐

3

api

保存

取消

编辑上图中 api 类型的枚举值为 3 并保存. 然后回到节点管理重新安装

注册主机到 CMDB 失败: 0 行"xxxxx"未赋值

>>step 4 安装脚本执行完毕

>>step 5 注册主机到CMDB

[2018-06-26 14:28:52]: registered to cmdb failed: 0行'asdfasd' 未赋值

[2018-06-26 14:28:52]: REGISTER_FAILED: register to cmdb failed: Partially failed

原因: 在 CMDB 导入主机的字段中增加了一个自定义的必填字段

解决办法: 在 CMDB 的主机模型配置中, 将该字段去除必填属性

检查 Agent 状态报错: 组件调用异常 exec redis_pipe_line_cmd failed

>>step 4 安装脚本执行完毕

>>step 5 注册主机到CMDB

[2018-06-25 16:01:38]: registered to cmdb success: success

>>step 6 检测Agent状态和版本

[2018-06-25 16:01:39]: query agent status failed: 组件调用异常:exec redis_pipe_line_cmd failed(code=1000184)

原因: gse_api 查询 redis 状态失败

解决办法: 升级 gse 版本 或 重启 gse_api, gse_dba

检查 agent 状态超时

[2018-06-27 16:02:06]: check agent status(60/60)(bk_agent_alive: 1->alive, 0->dead): 0
[2018-06-27 16:02:09]: query agent status timeout , give up, install failed.

原因: 通过 esb 查询 agent 状态, 太长时间获取不到 agent 状态信息. 可能原因如下

- agent 到 gse server 之间网络不通, 检查端口策略
- proxy 到 server 之间网络不通, 检查端口策略
- 注册到 cmdb 的 ip 与从 gse 侧访问 proxy 或 agent 所使用的 ip 不一致
- gse_dba 异常, 尝试重启
- agent 配置文件 agent.conf 中的 agentip 字段的值与注册到 CMDB 中的值不一致

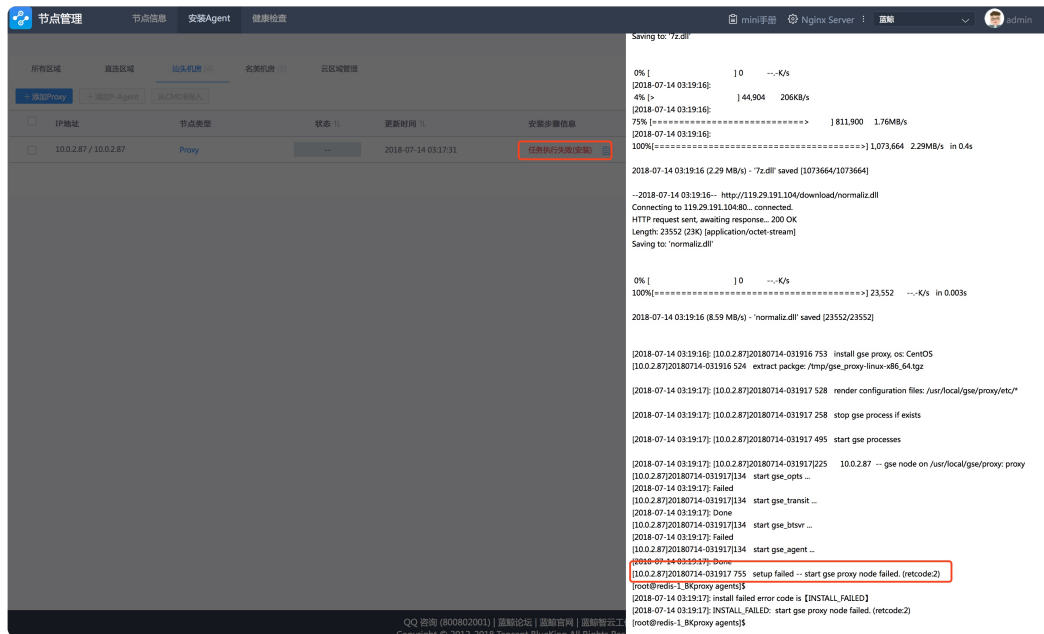
安装 windows 在第一步卡住



原因: 检测 Windows 系统版本需要通过 SMB 协议向待安装主机发送命令时可能阻塞

解决办法: 升级节点管理到最新版本或参考 [Windows 开 139, 445 端口](#)

启动进程失败



The screenshot shows the Nginx Agent management interface. On the left, a table lists nodes with columns for IP address, node type, status, update time, and installation progress. The node at IP 10.0.2.87 is of type 'Proxy' and its status is '失败' (Failed). A red box highlights the '查看详细安装日志' (View detailed installation log) link. On the right, the installation log is displayed, showing the process of downloading and installing the GSE service. A red box highlights the error message: 'SSH_WRONG_PASSWORD: SSH authentication failed. root@10.0.2.76:22 - exception: Authentication failed.'

解决办法: 联系蓝鲸的同学

密码错误: Authentication failed

```
>>step 1 登录目标主机
[2018-07-13 22:55:17]: start ssh login.
[2018-07-13 22:55:19]: ssh login failed , error code 【SSH_WRONG_PASSWORD】 : SSH authentication failed.
root@10.0.2.76:22 - exception: Authentication failed.
[2018-07-13 22:55:19]: SSH_WRONG_PASSWORD: SSH authentication failed. root@10.0.2.76:22 - exception:
Authentication failed.
```

解决办法: 检查密码并修正后重试 Note

“

windows 无 Cygwin 的环境中, 密码中不能包含@符号

”

检查 Agent 状态结果提示: failed to connect GSE service

```
[2018-07-14 03:27:33]: check agent status(33/60)(bk_agent_alive: 1->alive, 0->dead): 0
[2018-07-14 03:27:36]: check agent status(34/60)(bk_agent_alive: 1->alive, 0->dead): 0
[2018-07-14 03:27:39]: query agent status failed: 组件调用异常:Fail to connect GSE service. Please check if GSE service
is normal.(code=1306000)
```

若为 proxy

- 检查 proxy 到 gse 外网之间是否能正常通信(结合网络策略)
- 检查 gse 服务是否正常

若为 P-Agent

- 检查 P-Agent 到 proxy 之间网络是否通畅
- 检查所有 Proxy 是否正常

GSE/appo 到 Proxy 之间有 NAT, Proxy 到 GSE 可直连, Proxy 到 P-Agent 之间直连, 要怎么填写 Proxy 的内外网 IP

- 外网 IP 填写从 appo 登陆 Proxy 时所使用的 IP
- 内网 IP 填写 P-Agent 能连上的 IP.

已知问题列表

- 若曾使用过 **Agent安装** 这个 app 的早期版本,安装节点管理后,数据会自动迁移过来,但 Windows 机器的操作系统类型为 Linux,这是因为 Agent 安装 的历史版本问题,需要进行手动修改
- 从 A 业务迁移主机到 B 业务时,虽然提示成功,但在当前区域下 Agent 状态异常,因为在 CMDB 中,主机还在旧的业务下,查询主机状态时,指定的是当前业务,因此异常

标准运维常见问题

标准运维原子能否支持用户接入企业内 IT 系统

支持, 接入方式请参考 [附录 2: 原子开发](#)。

标准运维点击开始执行任务后报错

`taskflow[id=1] get status error: node(nodeee37e20...c7fb131) does not exist, may have not by executed`, 并且在任务列表中查看任务状态是“未知”, 可能是什么原因?

标准运维执行引擎依赖于蓝鲸的 RabbitMQ 服务和 App 启动的 celery 进程, 请登录服务器确认服务已启动并正常运行, 可以查看 App 的 celery.log 日志文件帮助定位问题原因。

标准运维能执行任务, 但是原子节点报错

Traceback...TypeError:int() argument must be a string or a number,not 'NoneType',
可能是什么原因？

标准运维任务流程的执行状态和原子输入、输出等信息缓存依赖 Redis 服务，所以首次部署请务必按照“[标准运维部署安装](#)”，配置 Redis 环境变量后重新部署。

蓝鲸监控常见问题



蓝鲸监控问题，多见用户 cmdb 及快照正常，而监控无基础数据上报，或者点击数据上报报错提示，或者提示正常但实际仍无数据



蓝鲸监控无数据上报

此问题常见表象

1. 启动入库任务失败【模块：data】接口结果返回超时
2. 进程端口下发失败: 0:x.x.x.x 进程协议错误，支持 TCP、UDP, 参数传递协议为, 请检查输入参数和 CMDB 配置
3. 基础性能接入失败：缺少[bk_biz_id]参数，解析 RT 信息失败：13
4. 模块：data 接口返回结果错误：参数错误 无 Linux 机器（机器多 IP，CMDB IP 匹配，）
5. 数据入库失败【模块：data】接口返回结果错误：All Servers failed to process request:
[('kafka.service.consul',9092,0)]
6. 模块：data 接口返回结果错误：调用接口失败 创建 tsdb 库失败（Dataapi 没升级好，把 saas 升级。）
7. 【模块：data】接口返回结果错误：组件请求第三方系统 DATA 接口 data_component 出错：状态码：404，错误消息：第三方系统不存在此接口（Dataapi 没升级好，确认下版本
8. 请求 agent 状态数据失败（job 接口，）
9. 查询失败，原因：【模块：data】接口返回结果错误：SQL 语法错误，请检查语法：06（进程资源无数据 bug，没入库）
10. _exporter_url_参数不能为空（自定义组件导入）
11. 系统信息可上报，组件信息不上报
12. 启动入库任务失败【模块：data】接口返回结果错误：添加总线任务失败，添加etl任务失败
6_ja_gse_proc_port（）

13. config_schema: 必须提供_exporter_url_参数的配置
14. 基础性能接入失败: 2: x.x.x.x 下发配置失败
15. 【模块: data】接口返回结果错误: 调用接口失败 gse_push_file: 该业务{0}下没有 IP

可能原因

1. nginx未更新安装, `install nginx 1` 后解决
2. gse_agent在节点管理, 重新安装
3. bk_monitor 在 supervisor 缺少 celery 和 beat 配置, 重新安装部署相对应的最新版本
4. bk_monitor 的服务使用错误使用 root 启动, 停掉进程, 删掉原启动对应 root 的文件, 用 apps 用户启动
5. influxdb表结构为空, 所有 `system_*` 的库, `show measurements` 为空
6. `/data/bkce/bkdata/dataapi/bin/check_databus_status.sh` 检查结果, `=====ETL=====` 发现未更新的, 结果类似 `etl_1004_2_ja_gse_net` 老的结构,
7. databus的 `tsdb.cluster.properties` 配置
中 `cluster.rest.port=__CONNECTOR_TSDB_PORT__` 未替换, 为空
8. `./kafka-console-consumer.sh --bootstrap-server kafka.service.consul:9092 --topic snapshot3|grep -P '"ip":".*?"' -o` 出现 `"ip": "__EXTERNAL_IP__"`
9. CMDB里面配置进程的时候需要配置网络协议

排查方法

检查进程是否正常

若为 4.1.X 的版本, bk_monitor 有对应 celery 进行异步任务处理, 需在 APPO 模块对应机器上, 确认 bk_monitor 的进程是否包含 `uwsgi`, `celery beat`, `celery worker` 3部分, 示例如下

```
[root@rbtndel1 /data/install]# ps -ef | grep bk_monitor
apps      1930      1  0 Aug09 ?        00:03:37 /data/bkce/paas_agent/apps/Envs/bk_monitor/bin/python /data/bkce/paas_agent/apps/Envs/bk_monitor/bin/supervisord -c /data/bkce/paas_agent/apps/projects/bk_monitor/conf/supervisord.conf
apps      1958    1930  0 Aug09 ?        00:00:24 /data/bkce/paas_agent/apps/Envs/bk_monitor/bin/uwsgi --ini /data/bkce/paas_agent/apps/projects/bk_monitor/conf/bk_monitor.ini
apps      1959    1930  0 Aug09 ?        00:10:50 /data/bkce/paas_agent/apps/Envs/bk_monitor/bin/python /data/bkce/paas_agent/apps/projects/bk_monitor/code/bk_monitor/manage.py
celery beat
apps      1960    1930  0 Aug09 ?        00:18:10 /data/bkce/paas_agent/apps/Envs/bk_monitor/bin/python /data/bkce/paas_agent/apps/projects/bk_monitor/code/bk_monitor/manage.py
celery worker -n bk_monitor -l INFO --autoscale=8,1
root      5315 11297  0 18:02 pts/5    00:00:00 grep --color=auto bk_monitor
apps      7480    1958  0 16:17 ?        00:00:01 /data/bkce/paas_agent/apps/Envs/bk_monitor/bin/uwsgi --ini /data/bkce/paas_agent/apps/projects/bk_monitor/conf/bk_monitor.ini
apps      7481    1958  0 16:17 ?        00:00:01 /data/bkce/paas_agent/apps/Envs/bk_monitor/bin/uwsgi --ini /data/bkce/paas_agent/apps/projects/bk_monitor/conf/bk_monitor.ini
apps     10815    1958  0 16:17 ?        00:00:00 /data/bkce/paas_agent/apps/Envs/bk_moni
```



```
tor/bin/uwsgi --ini /data/bkce/paas_agent/apps/projects/bk_monitor/conf/bk_monitor.ini
apps      10816  1958   0 16:17 ?          00:00:00 /data/bkce/paas_agent/apps/Envs/bk_moni
tor/bin/uwsgi --ini /data/bkce/paas_agent/apps/projects/bk_monitor/conf/bk_monitor.ini
apps      22477  1960   0 18:00 ?          00:00:00 /data/bkce/paas_agent/apps/Envs/bk_moni
tor/bin/python /data/bkce/paas_agent/apps/projects/bk_monitor/code/bk_monitor/manage.py
celery worker -n bk_monitor -l INFO --autoscale=8,1
apps      31585  1960   0 18:01 ?          00:00:00 /data/bkce/paas_agent/apps/Envs/bk_moni
tor/bin/python /data/bkce/paas_agent/apps/projects/bk_monitor/code/bk_monitor/manage.py
celery worker -n bk_monitor -l INFO --autoscale=8,1
apps      31587  1960   0 18:01 ?          00:00:00 /data/bkce/paas_agent/apps/Envs/bk_moni
tor/bin/python /data/bkce/paas_agent/apps/projects/bk_monitor/code/bk_monitor/manage.py
celery worker -n bk_monitor -l INFO --autoscale=8,1
apps      31588  1960   0 18:01 ?          00:00:00 /data/bkce/paas_agent/apps/Envs/bk_moni
tor/bin/python /data/bkce/paas_agent/apps/projects/bk_monitor/code/bk_monitor/manage.py
celery worker -n bk_monitor -l INFO --autoscale=8,1
```

检查BKDATA日志

确认 databus 日志 `/data/bkce/logs/bkdata/databus_etl.log` 以

及 `/data/bkce/logs/bkdata/databus_tsdb.log` 是否有 Exception 或者 Error 的错误, 示例如下

```
bash grep -nE "Exception|Error" /data/bkce/logs/bkdata/databus_etl.log
/data/bkce/logs/bkdata/databus_tsdb.log
199:org.apache.kafka.common.errors.wakeupException 12:Exception in thread "main"
org.apache.kafka.common.config.configException: Invalid value for configuration
rest.port: Not a number of type INT
```

检查 BKDATA DATABUS 任务

确认在bkdata服务器上, `check_databus_status.sh`, 不能出现有 `Failed connect to databus.service.consul:10054; connection refused` 或者 `JSON object could be decoded` 错误输出。正常的输出示例如下 (若此处有错误, 参考 `initdata bkdata` 失败的处理方法)

异常举例

```
etl_1003_2_redis_rep1
{"name":"etl_1003_2_redis_rep1","connector":{"state":"RUNNING","worker_id":"10.88.22.210:10052"},"tasks":[]}
etl_1001_2_system_io
{"name":"etl_1001_2_system_io","connector":{"state":"RUNNING","worker_id":"10.88.22.210:10052"},"tasks":[{"state":"FAILED","trace":"java.lang.NullPointerException\n\tat com.tencent.bkdata.co
nnect.common.sink.BKSinkTask.stop(BKSinkTask.java:109)\n\tat com.tencent.bkdata.connect.transform.TransformSinkTask.stop(TransformSinkTask.java:238)\n\tat org.apache.kafka.connect.runtime.Wor
kerSinkTask.close(WorkerSinkTask.java:131)\n\tat org.apache.kafka.connect.runtime.WorkerTask.doClose(WorkerTask.java:118)\n\tat org.apache.kafka.connect.runtime.WorkerTask.doRun(WorkerTask.jav
a:145)\n\tat org.apache.kafka.connect.runtime.WorkerTask.run(WorkerTask.java:182)\n\tat java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)\n\tat java.util.concurrent.Futu
reTask.run(FutureTask.java:266)\n\tat java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)\n\tat java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecu
tor.java:617)\n\tat java.lang.Thread.run(Thread.java:748)\n","id":0,"worker_id":"10.88.22.210:10052"}]}
etl_1003_2_redis_mem
{"name":"etl_1003_2_redis_mem","connector":{"state":"RUNNING","worker_id":"10.88.22.210:10052"},"tasks":[]}
etl_1010_2_uptimecheck_udp
{"error_code":404,"message":"No status found for connector etl_1010_2_uptimecheck_udp"}
etl_1003_2_redis_stat
{"name":"etl_1003_2_redis_stat","connector":{"state":"RUNNING","worker_id":"10.88.22.210:10052"},"tasks":[{"state":"FAILED","trace":"java.lang.NullPointerException\n\tat com.tencent.bkdata.co
nnect.common.sink.BKSinkTask.stop(BKSinkTask.java:109)\n\tat com.tencent.bkdata.connect.transform.TransformSinkTask.stop(TransformSinkTask.java:238)\n\tat org.apache.kafka.connect.runtime.Wor
kerSinkTask.close(WorkerSinkTask.java:131)\n\tat org.apache.kafka.connect.runtime.WorkerTask.doClose(WorkerTask.java:118)\n\tat org.apache.kafka.connect.runtime.WorkerTask.doRun(WorkerTask.ja
va:145)\n\tat org.apache.kafka.connect.runtime.WorkerTask.run(WorkerTask.java:182)\n\tat java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)\n\tat java.util.concurrent.Futu
reTask.run(FutureTask.java:266)\n\tat java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)\n\tat java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecu
tor.java:617)\n\tat java.lang.Thread.run(Thread.java:748)\n","id":0,"worker_id":"10.88.22.210:10052"}]}
etl_1001_2_system_net
{"name":"etl_1001_2_system_net","connector":{"state":"RUNNING","worker_id":"10.88.22.210:10052"},"tasks":[]}
etl_1003_2_redis_cpu
{"name":"etl_1003_2_redis_cpu","connector":{"state":"RUNNING","worker_id":"10.88.22.210:10052"},"tasks":[{"state":"FAILED","trace":"java.lang.NullPointerException\n\tat com.tencent.bkdata.co
nnect.common.sink.BKSinkTask.stop(BKSinkTask.java:109)\n\tat com.tencent.bkdata.connect.transform.TransformSinkTask.stop(TransformSinkTask.java:238)\n\tat org.apache.kafka.connect.runtime.Wor
kerSinkTask.close(WorkerSinkTask.java:131)\n\tat org.apache.kafka.connect.runtime.WorkerTask.doClose(WorkerTask.java:118)\n\tat org.apache.kafka.connect.runtime.WorkerTask.doRun(WorkerTask.jav
a:145)\n\tat org.apache.kafka.connect.runtime.WorkerTask.run(WorkerTask.java:182)\n\tat java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)\n\tat java.util.concurrent.Futu
reTask.run(FutureTask.java:266)\n\tat java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)\n\tat java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecu
tor.java:617)\n\tat java.lang.Thread.run(Thread.java:748)\n","id":0,"worker_id":"10.88.22.210:10052"}]}
etl_1002_2_mysql_innodb
```

```
[root@rbtndoe1 /data/install]# /data/bkce/bkdata/dataapi/bin/check_databus_status.sh
=====TSDB=====
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                       Dload  Upload    Total   Spent    Left   Speed
100  773  100  773    0     0  14599      0 --:--:-- --:--:-- --:--:-- 14865
tsdb_2_system_cpu_summary
```



```
{ "name": "tsdb_2_system_cpu_summary", "connector": { "state": "RUNNING", "worker_id": "10.X.X.X:10054" }, "tasks": [ { "state": "RUNNING", "id": 0, "worker_id": "10.X.X.X:10054" } ] }
```

```
=====MYSQL=====
```

% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current
			Dload Upload	Total	Spent	Left	Speed
100	27	100	27	0	0	1118	0

```
jdbc_2_ja_gse_proc_port
{ "name": "jdbc_2_ja_gse_proc_port", "connector": { "state": "RUNNING", "worker_id": "10.X.X.X:10051" }, "tasks": [ { "state": "RUNNING", "id": 0, "worker_id": "10.X.X.X:10051" } ] }
```

```
=====ETL=====
```

% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current
			Dload Upload	Total	Spent	Left	Speed
100	1040	100	1040	0	0	166k	0

```
etl_1001_2_system_cpu_summary
{ "name": "etl_1001_2_system_cpu_summary", "connector": { "state": "RUNNING", "worker_id": "10.X.X.X:10052" }, "tasks": [ { "state": "RUNNING", "id": 0, "worker_id": "10.X.X.X:10052" } ] }
```

检查 KAFKA

1. kafka 节点确认

若社区版为 3 台部署的，必须返回[1, 2, 3]才正常，示例如下 若 brokers ids 不为[1, 2, 3]，存在 `/data/bkce/public/kafka/.lock` 文件，有的话，删除此文件，再重新使用 `./bkcec stop kafka` 和 `./bkcec start kafka` 重启 kafka，重启完再次确认状态

```
[zk: zk.service.consul:2181(CONNECTED) 4] ls /common_kafka/brokers/ids
[1, 3]
```

```
[root@kafka-2 ~]# ll /data/bkce/public/kafka/.
./      ../      .lock
```

```
[root@kafka-2 ~]# /data/bkce/service/zk/bin/zkCli.sh -server zk.service.consul:2181/common_kafka ls /brokers/ids
Connecting to zk.service.consul:2181/common_kafka
log4j:WARN No appenders could be found for logger (org.apache.zookeeper.ZooKeeper).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.

WATCHER::

WatchedEvent state:SyncConnected type:None path:null
[1, 2, 3]
```

```
[root@rbtnode1 /data/install]# /data/bkce/service/zk/bin/zkCli.sh -server zk.service.consul:2181 ls /common_kafka/brokers/ids
Connecting to zk.service.consul:2181
log4j:WARN No appenders could be found for logger (org.apache.zookeeper.ZooKeeper).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
WATCHER::

WatchedEvent state:SyncConnected type:None path:null
[1, 2, 3]
```

1. kafka 僵死

此种情况，确认上面第一点，若节点缺失，可以用 `./bkcec stop kafka` 停掉 kafka，再到各台机器上使用 `jps -l` 确认是否还存在 `kafka.kafka` 没有关闭掉的，这种就是僵死的，可以杀掉僵死进程 `jps -l | grep kafka | xargs kill -9`，再重新在中控机上启动 kafka

1. kafka 数据确认

正常可以显示数据，异常会卡住

```
[root@mongodb-1 kafka]# bash /data/bkce/service/kafka/bin/kafka-console-consumer.sh --bootstrap-server kafka.service.consul:9092 --topic connect-configs.etl --from-beginning | head
{"properties":{"group.id":"bk_data_etl","topics":"snapshot2","producer.records.in.msg":"100","tasks.max":"1","producer.topic.badmsg":"test_badmsg","producer.max.block.ms":"60000","connector.class":"com.tencent.bkdata.databases.transform.TransformSinkConnector","db.dataid":"1001","name":"etl_1001_2_snapshot","producer.acks":"1","rt.id":"2_snapshot","producer.bootstrap.servers":"","producer.request.timeout.ms":"60000","producer.retries":"5","producer.max.in.flight.requests.per.connection":"5"}}
{"properties":{"connector.class":"com.tencent.bkdata.databases.transform.TransformSinkConnector","producer.topic.badmsg":"test_badmsg","producer.request.timeout.ms":"60000","producer.bootstrap.servers":"","topics":"snapshot2","tasks.max":"1","producer.max.block.ms":"60000","group.id":"bk_data_etl","producer.retries":"5","db.dataid":"1001","producer.acks":"1","producer.records.in.msg":"100","task.class":"com.tencent.bkdata.connect.transform.TransformSinkTask","name":"etl_1001_2_snapshot","producer.max.in.flight.requests.per.connection":"5","rt.id":"2_snapshot"}}
{"tasks":1}
```

正常可以显示数据，异常会卡住

```
[root@localhost install]# bash /data/bkce/service/kafka/bin/kafka-topics.sh --zookeeper zk.service.consul:2181/common_kafka --describe --topic connect-configs.tsdb
Topic:connect-configs.tsdb      PartitionCount:1      ReplicationFactor:1      Configs:retention.ms=315360000000
      Topic: connect-configs.tsdb      Partition: 0      Leader: 3      Replicas: 3
Isr: 3
```

```
[root@localhost kafka]# grep configs.tsdb server.log* -H
server.log.2018-08-14-11:[2018-08-14 11:44:13,539] INFO [ReplicaFetcherManager on broker 3] Removed fetcher for partitions connect-configs.tsdb-0 (kafka.server.ReplicaFetcherManager)
server.log.2018-08-14-11:[2018-08-14 11:44:13,548] INFO Completed load of log connect-configs.tsdb-0 with 1 log segments and log end offset 0 in 1 ms (kafka.log.Log)
server.log.2018-08-14-11:[2018-08-14 11:44:13,582] INFO Created log for partition [connect-configs.tsdb,0] in /data/bkce/public/kafka with properties {compression.type -> producer, message.format.version -> 0.10.2-IV0, file.delete.delay.ms -> 60000, max.message.bytes -> 1000012, min.compaction.lag.ms -> 0, message.timestamp.type -> CreateTime, min.insync.replicas -> 1, segment.jitter.ms -> 0, preallocate -> false, min.cleanable.dirty.ratio -> 0.5, index.interval.bytes -> 4096, unclean.leader.election.enable -> true, retention.bytes -> -1, delete.retention.ms -> 86400000, cleanup.policy -> [delete], flush.ms -> 9223372036854775807, segment.ms -> 604800000, segment.bytes -> 1073741824, retention.ms -> 315360000000, message.timestamp.difference.max.ms -> 9223372036854775807, segment.index.bytes -> 10485760, flush.messages -> 9223372036854775807}. (kafka.log.LogManager)
server.log.2018-08-14-11:[2018-08-14 11:44:13,584] INFO Partition [connect-configs.tsdb,0] on broker 3: No checkpointed highwatermark is found for partition connect-configs.tsdb-0 (kafka.cluster.Partition)
```

```
$ [root@rbtnode1 install]# /data/bkce/service/kafka/bin/kafka-console-consumer.sh --bootstrap-server $LAN_IP:9092 --topic snapshot2
{"beat":{"address":["192.168.26.132","192.168.26.150"],"fe80::be16:6e1a:e42f:f329"},"hostname":"paas-1","name":"paas-1","version":"10.1.0"},"bizid":0,"cloudid":0,"data":{"city":"","country":"","cpu":{"cpuinfo":[{"cacheSize":20480,"coreId":0,"cores":1,"cpu":0,"family":6,"flags":["fpu","vme","de","pse","tsc","msr","pae","mce","cx8","apic","sep","mtrr","pge","mca","cmov","pat","pse36","clflush","dts","mmx","fxsr","sse","sse2","ss","ht","syscall","nx","rdtscp","lm","constant_tsc","arch_perfmon","pebs","bts","nopl","xt
```

```
opology", "tsc_reliable", "nonstop_tsc", "aperfmpperf", "eagerfpu", "pni", "pclmulqdq", "vmx", "
ssse3", "fma", "cx16", "

/data/bkce/service/kafka/bin/kafka-console-consumer.sh --bootstrap-server kafka.service
.consul:9092 --topic snapshot2|grep -P '"ip":'.*?' -o
```

检查INFLUXDB

确认influxdb内的数据库和结构，正常返回如下

```
$ influx -host $INFLUXDB_HOST -port $INFLUXDB_PORT -execute 'show databases'
name: databases
name
----
_internal
system_2

$ influx -host $INFLUXDB_HOST -port $INFLUXDB_PORT -database system_2 -execute 'show me
asurements'
name: measurements
name
----
system_cpu_detail_2
system_cpu_summary_2
system_disk_2
system_env_2
system_inode_2
system_io_2
system_load_2
system_mem_2
system_net_2
system_netstat_2
system_proc_2
system_swap_2
```

检查CRON任务

```
# 确认是否存在update_cc_cache crontab任务
crontab -l | grep update_cc_cache
*/10 * * * * /data/bkce/bkdata/dataapi/bin/update_cc_cache.sh

# 手动运行update_cc_cache.sh
/data/bkce/bkdata/dataapi/bin/update_cc_cache.sh

# 检查crontab是否在运行，若没有运行，可采用service crond start或者/etc/init.d/cron start来启动
ps -ef | grep crond | grep -v grep
root      27979      1  0 7月26 ?        00:01:23 /usr/sbin/crond -n

# 检查crontab是否正常运行，确认crontab的日志
tail /var/log/cron
```

如果 CMDB 有快照数据，监控没有，或者部分没有时，检查下 bkdata 所在机器的 crontab 是否包含 update_cc_cache.sh 的定时任务：

如果没有，应该安装时漏执行，或者执行过 `./bkcec clean cron` 后忘记加回来。可在中控机执行 `./bkcec install cron` 重新安装上 cron 任务检查 bkdata 的 cron 任务

部分有监控数据，部分没有

若蓝鲸平台的监控 OK，而新增加的 Agent 监控没有，点击数据上报，提示 10-20 分钟会有新数据，实际一直没有数据，可能为 crontab 的 update_cc_cache 未正常运行，导致 cache 未更新，新的机器未添加进来

检查 cron 任务

```
# 确认是否存在update_cc_cache crontab任务
crontab -l | grep update_cc_cache
*/10 * * * * /data/bkce/bkdata/dataapi/bin/update_cc_cache.sh

# 检查crontab是否在运行，若没有运行，可采用service crond start或者/etc/init.d/cron start来启动
ps -ef | grep crond | grep -v grep
root      27979      1  0 7月26 ?        00:01:23 /usr/sbin/crond -n

# 检查crontab是否正常运行，确认crontab的日志
tail /var/log/cron
```

如果 CMDB 有快照数据，监控没有，或者部分没有时，检查下 bkdata 所在机器的 crontab 是否包含 update_cc_cache.sh 的定时任务：

如果没有，应该安装时漏执行，或者执行过 `./bkcec clean cron` 后忘记加回来。可在中控机执行 `./bkcec install cron` 重新安装上 cron 任务检查 bkdata 的 cron 任务

手动运行update_cc_cache.sh

```
$ /data/bkce/bkdata/dataapi/bin/update_cc_cache.sh
```

- 确认无数据上报机器的时间和部署蓝鲸server机器的时间是否同步。
- 确认无数据的机器上是否有basereport进程采集器
 - 若无可尝试手动拉起 `/usr/local/gse/plugins/bin/start.sh basereport`
 - 若采集器进程存在则查看采集器日志 `/var/log/gse/basereport` (日志为 error 级别，有进程无日志说明采集器进程正常)。

组件没有监控数据

- 到 cmdb 里面，主机管理->进程管理，搜索组件名称，类似搜索 nginx，然后把 common_nginx 改名为 nginx
- 到蓝鲸监控里面，勾选这个 nginx 对应的主机，批量采集上报

缺少操作系统类型

在 cc 上，确认所选主机操作系统类型是否为空

后台：Exited too quickly (process log may have detail)

问题表象：

```
[root@rbtnode1]# supervisorctl -c /data/bkce/etc/supervisor-bkdata-monitor.conf status
common:scheduler
common:scheduler      FATAL      Exited too quickly (process log may have details)
```

思路方法：- 确认有问题的进程名

```
supervisorctl -c /data/bkee/etc/supervisor-bkdata-monitor.conf status all
```

- 找到有问题的进程名。例如上图中显示为：common:scheduler，其中common是group，scheduler是进程名 找到该进程的启动命令

```
vim /data/bkee/etc/supervisor-bkdata-monitor.conf

# 在[program:scheduler]需要找到一下配置

environment=C_FORCE_ROOT=true,DJANGO_CONF_MODULE=conf.worker.production.enterprise,DJANGO_SETTINGS_MODULE=settings,LOGGER_WITHOUT_CONSOLE=1
command=/data/bkee/.envs/monitor/bin/celery -A kernel.scheduler.celery_app worker -l info
```

- 手动执行以下命令

```
workon monitor
# export environment的配置，逗号改成空格。
export C_FORCE_ROOT=true DJANGO_CONF_MODULE=conf.worker.production.enterprise DJANGO_SETTINGS_MODULE=settings LOGGER_WITHOUT_CONSOLE=1
# 直接运行command的配置
/data/bkee/.envs/monitor/bin/celery -A kernel.scheduler.celery_app worker -l info
```

查看报错信息以确认具体错误日志

主机百分比指标超过 100

出现此问题，优先去对应机器上看下是否有 2 个 basereport 进程在运行

节点名称快速执行脚本-2018619114010335

节点类型执行脚本

服务器帐户root

总时间(s)1.074

节点状态执行成功

开始时间2018-06-19 11:40:33 +0800

结束时间2018-06-19 11:40:35 +0800

执行成功(1)

输入搜索内容

Q 搜索日志 导出执行日志

IP过滤:

云区域名称	IP	返回码	耗时(s)
default area	192.168.132.15	0	0.194

第 1 页 / 共 1 页 每页显示 6 条 共 1 条

[2018-06-19 11:40:34][PID:26332] job_start

root 16903 0.0 0.0 106172 332 ? S May29 0:00 bash -login -c cd /usr/local/gse/plugins/bin && ./basereport-f ./etc/basereport.conf &

root 16904 0.0 0.0 14772 982 ? S May29 16:49 ./basereport-f ./etc/basereport.conf

root 19142 0.0 0.0 106176 356 ? S May29 0:00 bash -login -c cd /usr/local/gse/plugins/bin && ./basereport-f ./etc/basereport.conf &

root 19143 0.0 0.0 14772 984 ? S May29 16:13 ./basereport-f ./etc/basereport.conf

root 19569 0.0 0.0 106172 332 ? S May29 0:00 bash -login -c cd /usr/local/gse/plugins/bin && ./basereport-f ./etc/basereport.conf &

root 19570 0.0 0.0 14772 942 ? S May29 16:48 ./basereport-f ./etc/basereport.conf

root 26343 0.0 0.0 103384 2136 ? S 11:40 0:00 grep basereport

exporter 采集下发时，模块 data 接口返回结果错误，但没有详细的错误信息

3 以上步骤完成的前提下，[到此步点击 立即采集](#) 完成配置下发和验证测试即可保存

❌ 检查组件版本失败：【模块：data】接口返回结果错误：

主机IP	当前状态
------	------

可能是 bk_bkdata_api.collector_exporter 表没有创建，可查看 dataapi 的 sys.log 查看更准确错误信息

```
source /data/install/utils.fc && mysql -h $MYSQL_IP0 -u $MYSQL_USER -p"$MYSQL_PASS" bk_bkdata_api
```

自定义字符型未收到告警问题排查

在蓝鲸监控配置好监控项

蓝鲸配置

当前用户: 管理员

admin

主机监控 自定义监控 告警配置

监控名称搜索...

监控名称	监控类型	策略组(启用/未启用)	阈值	操作
接收字节流量	网络	1/0	100.0%	+
磁盘写满	事件	1/0	100.0%	+
Corefile产生	事件	1/0	100.0%	+
磁盘使用率	磁盘	1/0	100.0%	+
发送字节流量	网络	1/0	100.0%	+
Agent心跳丢失	事件	1/0	100.0%	+
磁盘只读	事件	1/0	100.0%	+
PING不可达告警	事件	1/0	100.0%	+
自定义字符型	字符型	1/0	100.0%	+

策略	告警级别	告警条件	处理动作	处理动作通知方式	通知角色	状态	操作
-	普通	集群: 全部; 模块: 全部	不处理, 仅通知		选择角色	已启用	

应用内存使用率 内存 1/0 100.0% +

显示条数 1-10 共 43

按配置时的页面提示，去到主机上执行命令

指标分类*

CPU

网络

内存

磁盘

进程

系统

事件

指标名称*

自定义字符串

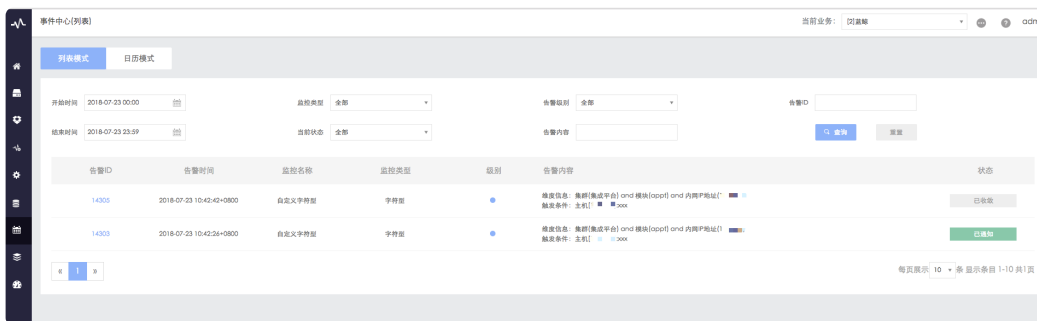
上报ID: 2001

通过直输Agent目录中的gsecmdline命令上报自定义字符串告警
用法: /usr/local/gse/plugins/bin/gsecmdline -d 2001 -l "This service is offline."

在机器上执行命令 `/usr/local/gse/plugins/bin/gsecmdline -d 2001 -l "This service is offline."`

```
root@rbtnode1 ~]#  
root@rbtnode1 ~]# /usr/local/gse/plugins/bin/gsecmdline -d 2001 -l "xxx"  
root@rbtnode1 ~]#  
root@rbtnode1 ~]#
```

正常情况下，在事件中心可以看到有告警产生，如下图所示



排查方法

1. 先检查下机器上的 agent 是否正常，数据是否有正确上报
2. 第一步正常的情况下，执行下面的步骤

- 去到bkdata机器tailf一下日志

```
bash workon monitor # （社区版5.1用 workon bkdata-monitor） tail -f  
../../logs/bkdata/kernel.log | grep "gse_custom_out_str_"
```

- 去到业务下任意一台机器，触发一条自定义字符串告警

```
bash /usr/local/gse/plugins/bin/gsecmdline -d 2001 -l "xxx"
```

- 回到bkdata机器，看下日志，正常会输出如下的日志

```
2018-07-23 10:42:35 INFO 23646 poll_alarm gse_custom_out_str.py[299] gse_custom_out_str.49: []  
2018-07-23 10:42:35 INFO 23646 poll_alarm gse_custom_out_str.py[299] gse_custom_out_str.2: [{"u_uptime": "2018-07-23 02:42:26", "u_time": "2018-07-23 10:42:26", "u_bizid": "0", "u_server": "u  
gse_custom_out_str.1: [{"u_bizid": "0", "u_time": "2018-07-23 10:42:26", "u_bizid": "0", "u_server": "u  
2018-07-23 10:42:35 INFO 23646 poll_alarm gse_custom_out_str.py[299] gse_custom_out_str.678: []  
2018-07-23 10:42:35 INFO 23646 poll_alarm gse_custom_out_str.py[299] gse_custom_out_str.657: []
```

3. 如果第二步正常，可以看到日志输出

托管进程启动失败：fail to restart process

问题表象：在配置组件采集参数，完成配置下发和验证测试的过程中，出现如下错误提示

1. 测试不通过，[datadog]托管程序启动失败：fail to restart process
2. 存在执行失败的主机，需要将失败主机剔除方可进行下一步操作

思路方法： - 进入下发机器 - 进入 datadog 安装目录，尝试启动 datadog server，查看是否有报错

```
bash $ cd /usr/local/gse/external_plugins/datadog $ source ./env.sh $  
${datadog_python_path} ./datadog/datadog_httpserver.py ${datadog_conf_path}
```

- 常见问题
- env.sh文件不存在，在页面点击“重试”即可
- python 版本过低，安装 python2.7 版本，并指定正确的 python 程序路径
- socket.error: [Errno 98] Address already in use `bash ps -ef | grep datadog`
- 如果存在 `/datadog/datadog/jar/jmxfetch-0.19.0-jar-with-dependencies.jar` 的进程，则kill掉即可

故障自愈常见问题

故障自愈的通知渠道有哪些？

故障自愈默认有 4 种通知渠道：电话、微信、短信、邮件

你在接入自愈的过程中，可以设置通知类型

通知方式	开始时	<input checked="" type="checkbox"/> 微信	<input checked="" type="checkbox"/> 邮件	<input type="checkbox"/> 短信	<input type="checkbox"/> 电话
	成功时	<input checked="" type="checkbox"/> 微信	<input checked="" type="checkbox"/> 邮件	<input type="checkbox"/> 短信	<input type="checkbox"/> 电话
	失败时	<input checked="" type="checkbox"/> 微信	<input checked="" type="checkbox"/> 邮件	<input type="checkbox"/> 短信	<input type="checkbox"/> 电话

腾讯云版本的故障自愈默认已经打通 4 种通知渠道

独立部署版本，需要企业对接蓝鲸的 ESB

故障自愈依赖哪些周边系统？

故障自愈依赖蓝鲸的 `配置平台`、`作业平台`

你需要提前在 `配置平台` 上创建一个业务，并且把你的服务器录入到配置平台中

接入自愈注意事项

接入自愈需要注意告警类型必须要和实际告警相匹配

日志检索常见问题

通用问题

- 日志检索下发失败
- 查询不到日志日志检索下发失败
- 查询不到日志
- 日志占满磁盘空间

检查用户 kafka 的机器是不是磁盘满了 `df -lh`
如果是的话，检查是否是 kafka 的数据日志满了 `du -sh /data/bkce/public/kafka`
如果是的话，看下用户的 `/data/bkce/service/kafka/config/server.properties` 里面是否有 `log.rete`
`ntion.bytes` 配置，如果没有的话加上 `log.retention.bytes=21474836480`
停掉 kafka
启动 kafka，去磁盘满的机器看是否磁盘空间释放了(这里可能要等 Kafka 启动后一段时间才启动，刚才操作大约10分钟)

环境问题

如果用户的蓝鲸后台机器上也部署了 zabbix agent 时，在使用日志检索时，可能会遇到如下截图的错误：



这个问题一般是 bkdata 模块的 databus_es 进程监听的 10050 端口和该机器上 zabbix agent 的端口冲突。

解决方法如下：

1. 修改中控机的 /data/install/ports.env 中下面两行配置的 10050 端口为 10049，避开冲突 `bash`
`export DATABUS_ES_PORT=10050 export CONNECTOR_ES_PORT=10050`
2. `./bkcec sync common`
3. `./bkcec render bkdata`
4. `./bkcec stop bkdata databus`
5. `./bkcec start bkdata databus`
6. `./bkcec stop bkdata dataapi`
7. `./bkcec start bkdata dataapi`

内存问题

/data/bkce/logs/bkdata/databus_es.log 日志报错 memory is low 提示内存不足，但是分配了足够的内存

```
2019-03-06 16:17:51.540|127.0.0.1|com.tencent.bkdata.connect.sink.es.EsClientUtils|144|WARN|0|pool-8-thread-1|Memory is low!! used: 155395160
    free: 113040296
2019-03-06 16:17:52.534|127.0.0.1|com.tencent.bkdata.connect.sink.es.EsClientUtils|144|WARN|0|pool-8-thread-1|Memory is low!! used: 165563248
    free: 102872208
2019-03-06 16:17:53.534|127.0.0.1|com.tencent.bkdata.connect.sink.es.EsClientUtils|144|WARN|0|pool-8-thread-1|Memory is low!! used: 165660784
    free: 102774672
2019-03-06 16:17:54.534|127.0.0.1|com.tencent.bkdata.connect.sink.es.EsClientUtils|144|WARN|0|pool-8-thread-1|Memory is low!! used: 165963624
    free: 102471832
2019-03-06 16:17:55.534|127.0.0.1|com.tencent.bkdata.connect.sink.es.EsClientUtils|144|WARN|0|pool-8-thread-1|Memory is low!! used: 169855168
    free: 98830288
2019-03-06 16:17:56.533|127.0.0.1|com.tencent.bkdata.connect.sink.es.EsClientUtils|144|WARN|0|pool-8-thread-1|Memory is low!! used: 170173736
    free: 98261720
2019-03-06 16:17:57.534|127.0.0.1|com.tencent.bkdata.connect.sink.es.EsClientUtils|144|WARN|0|pool-8-thread-1|Memory is low!! used: 170462888
    free: 97972568
2019-03-06 16:17:58.534|127.0.0.1|com.tencent.bkdata.connect.sink.es.EsClientUtils|144|WARN|0|pool-8-thread-1|Memory is low!! used: 170569560
    free: 97865896
2019-03-06 16:17:59.534|127.0.0.1|com.tencent.bkdata.connect.sink.es.EsClientUtils|144|WARN|0|pool-8-thread-1|Memory is low!! used: 170674616
    free: 97760840
```

```
vim /data/bkce/bkdata/databus/conf/es.cluster.properties
# 启动 jvm 是最大分配内存
deploy.cluster.memory.max=2G
```

```
部署相关配置项 (deploy.sh, start.sh脚本使用)
# 用于预先生成config/status/offset topic使用
deploy.bkdata.cluster.replication=1
deploy.bkdata.cluster.zookeeper=zk.service.consul:2181/common kafka
# cluster所属类型, 取值范围为[access, clean, shipper], 用于构建dns、log文件名称等
deploy.cluster.type=shipper
# 启动jvm是最大分配内存
deploy.cluster.memory.max=2G
```

重启 databus_es

```
supervisorctl -c /data/bkce/etc/supervisor-bkdata-databus.conf restart databus_es
```

网络管理常见问题

如何开通设备的 snmp 和 syslog 服务

网络设备需要开通和设置好 snmp 和 syslog 服务，才能纳入监控管理。有 3 种情况：

- 请求网络管理员用手工方式开通和设置 Cisco，华为，H3C 等网络设备的 snmp 和 syslog 服务
- Linux 和 Windows 的 snmp 设置可参考系统的“开通 snmp 和 syslog”功能页面中的说明
- 使用“开通 snmp 和 syslog”功能页面，采用 ssh 方式一台台开通设备的 snmp 和 syslog 服务

如何自动发现设备及其模块

首先设备要开通 snmp 服务。然后在“自动发现设备”页面中，输入起始 IP 地址和终止 IP 地址，点击“确认”，如下图所示。发现过程会提示本次新发现了多少设备，重新发现了多少设备。如果是重新发现的设备，可能会同时发现出设备名称的改变，也会做出提示，并生成一条事件信息。

发现完毕后，可以转到“设备信息”页面中查询新发现设备的详细信息

自动发现设备

部门/区域名称: 数据中心

开始(或种子)IP*: 192.168.2.101

结束IP地址: 192.168.2.105

重要性: 重要

Communities: public

SNMP版本: v2cv1

☒ 是否受控设备 ☒ 允许重新发现 ☒ 采集路由表 ☒ 仅发现SNMP设备 ☒ 采集Vlan表 ☒ 采集CDP表 ☒ 采集STP表

☐ LOOPBACK地址做为管理IP地址 ☐ 最小IP地址做为管理IP地址 ☐ IP地址做为设备名称 ☐ IP地址做为设备名称的前缀

确认

发现结果

取消

初始化数据...

开始Ping网段: 192.168.2.101---> 192.168.2.105 以产生 ARP/MAC 信息

Finished to generate mac by ping for 192.168.2.101---> 192.168.2.105

Ping IP地址: 192.168.2.101, 耗时(ms): 1

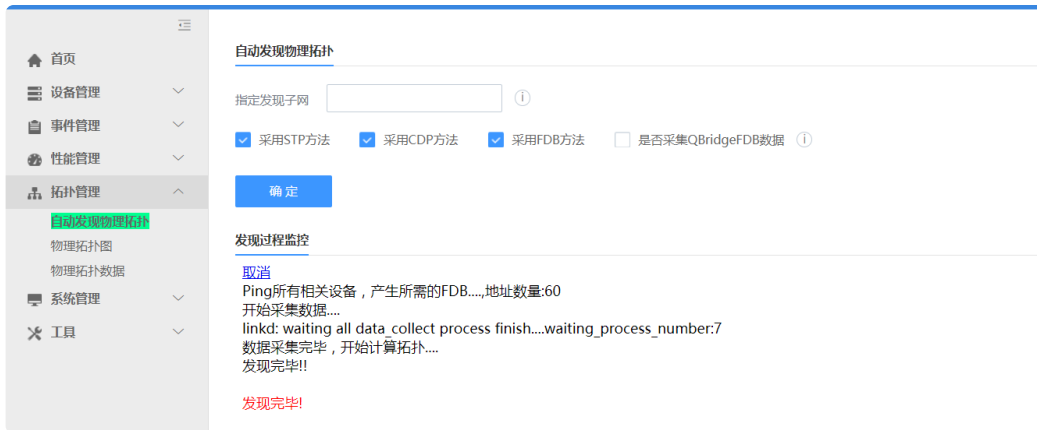
开始发现节点: 192.168.2.101 跳数(nowHop): -1

Ping IP地址: 192.168.2.102, 耗时(ms): -1

Ping IP地址: 192.168.2.103, 耗时(ms): -1

如何自动发现物理拓扑和调整物理拓扑

系统采用了 CDP,STP 和 FDB 三种发现方式，以尽可能完整准确的发现异构网络环境的真实的物理拓扑结构。选择“自动发现物理拓扑”菜单项，出现下面的页面：



然后不要输入任何信息，再直接点击“确认”，将显示发现过程，发现信息会提示本次发现的(新的或修改的)物理连接信息。发现完毕后，可转入“物理拓扑图”，查看物理拓扑。

没有进行拓扑图的发现前，物理拓扑图中显示的是散列的各个设备。

物理拓扑图初始是没有进行布局调整的界面，所以会显得比较零乱，此时需要进行手工调整布局，然后保存下来即可。

如何请求和使用其他应用功能

社区版提供了网络管理基本和实用的功能，能满足日常基本上的网络监控管理需要。而其他工作则需要升级到企业版或云版，它们提供了更为全面的功能，如：

- 统计报表
- 拓扑连接的手工可视化调整
- 客户化的 KPI 定义
- 客户化的阈值定义
- 流量管理
- 链路管理
- 专线管理
- 网络服务(ISM)管理
- 逻辑拓扑的自动发现
- 多租户支持

Consul

Consul 解析逻辑

在部署和使用时，如果遇到类似这样的日志信息："Name or service not known" 或者 "host=xxx.service.consul port=xxxx max retries....."

意味着内部域名无法解析的问题。内部域名，指的是蓝鲸集群模块之间使用 consul 模块注册的以".service.consul"结尾 的域名。它由每台机器上运行的 consul 进程监听的 53 端口提供解析服务。

Consul 配置说明

- 主配置

```
[root@rbtnode1 /data/install]# cat /data/bkee/etc/consul.conf
{
  "rejoin_after_leave": true,
  "skip_leave_on_interrupt": true,
  "recursors": [],
  "bind_addr": "10.X.X.X",
  "node_id": "8fb274be-245f-4301-926f-76e1c1abf316",
  "retry_join": [
    "10.X.X.X",
    "10.X.X.X",
    "10.X.X.X"
  ],
  "log_level": "info",
  "server": true,
  "datacenter": "dc",
  "data_dir": "/data/bkee/public/consul",
  "leave_on_terminate": false,
  "node_name": "gse-1",
  "bootstrap_expect": 3,
  "pid_file": "/data/bkee/logs/consul.pid",
  "encrypt": "uUrZvLe8gff5jNKRwH1Q0w==",
  "ports": {
    "dns": 53
  }
}
```

- 服务配置

```
[root@rbtnode1 /data/install]# cat /data/bkee/etc/consul.d/license.json
{
  "service": {
    "id": "license-1",
    "checks": [
      {
        "service_id": "license-1",
        "interval": "10s",
        "script": "/data/bkee/bin/health_check/check_proc_exists -m license"
      }
    ],
    "name": "license",
    "enableTagOverride": false,
    "address": "10.X.X.X"
  }
}
```

Consul 内部域名无法解析

表象：

在部署和使用时，如果遇到类似这样的日志信息："Name or service not known" 或者 "host=xxx.service.consul port=xxxx max retries....."

意味着内部域名无法解析的问题。内部域名，指的是蓝鲸集群模块之间使用 consul 模块注册的以".service.consul"结尾 的域名。它由每台机器上运行的 consul 进程监听的 53 端口提供解析服务

思路方法：

当无法解析时，第一步，在报错的机器上使用 dig 看看 consul 能否解析：

```
$ dig xxx.service.consul @127.0.0.1
```

@127.0.0.1 表示使用 127.0.0.1:53 这个作为 dns 服务器，也就是使用 consul 提供的 dns 服务

正常情况下，可以看到类似下图的记录。如果命令换成 **dig 域名** 没出现正确的记录，说明 **/etc/resolv.conf** 里没有配置上 127.0.0.1 的 nameserver，确认 **/etc/resolv.conf** 里第一行是 **nameserver 127.0.0.1**

```
;; ANSWER SECTION:
zk.service.consul.    0      IN      A       10.x.x.x
zk.service.consul.    0      IN      A       10.x.x.x
zk.service.consul.    0      IN      A       10.x.x.x

;; Query time: 1 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
```

如果出现以下信息，说明 consul 没有正常启动。那么 使用 supervisor 启动 consul 进程

```
;; <<>> DiG 9.9.4-RedHat-9.9.4-29.el7_2.2 <<>> zk.service.consul @127.0.0.1
;; global options: +cmd
;; connection timed out; no servers could be reached
```

如果出现以下信息 "IN A" 后面没有 ip 地址，说明 consul 启动了，但是无法解析域名

```
;; QUESTION SECTION:
;zk.service.consul.      IN      A

;; AUTHORITY SECTION:
consul.                  0      IN      SOA     ns.consul. postmaster.consul. 1530849644 3600 600
                        86400 0
```

此时按照以下步骤:

- 运行 `consul monitor` 看看日志, 主要确认 consul 集群状态是否正常。观察是否有"no cluster leader"的输出。
- 针对具体的域名, 譬如 zk.service.consul, 那么登陆到 zk 所在机器, 查看 `/data/bkcec/etc/consul.d/zk.json` 文件 运行里面的 check 脚本, 看返回的输出。

对于出现"no cluster leader"的输出时, 说明 consul 之间没有成功组成集群, 选举出 leader:

- 检查 consul server 节点是否都 running
- 在任意一台 consul 上输入 `consul join <另外一个 consul 节点>`
- 查看节点状态: `consul operator raft list-peers`

MySQL 常见问题

MySQL 密码更新流程

“

如下指引, 若无特殊说明, 全部在中控机 `/data/install` 目录进行

”

停止 MySQL

```
./bkcec stop mysql

# 确认 mysql 真正停掉
./bkcec status mysql

# 在 mysql 机器确认
ps -ef | grep mysql | grep -v grep
```

修改 MySQL 密码

修改 globals.env 里的 mysql_PASS 值, 密码不要包含 `[] / : @ ?` 等特殊字符

```
export MYSQL_PASS='新密码'
```

同步 install 目录

```
./bkcec sync common
```

关闭相关服务

```
# 关闭平台服务
echo bkdata gse job paas gse kafka cmdb | xargs -n 1 ./bkcec stop
echo bkdata gse job paas gse kafka cmdb | xargs -n 1 ./bkcec status
```

社区版 5.1+bcs 环境中与 MySQL 相关的模块为 MySQL, PaaS, Job, bkdata, SaaS, iam

```
# 关闭平台服务
echo iam bkdata gse job paas gse kafka cmdb | xargs -n 1 ./bkcec stop
echo iam bkdata gse job paas gse kafka cmdb | xargs -n 1 ./bkcec status
```

关闭 SaaS 应用

在 appo 服务器上执行

```
# 若为单机部署, 请使用如下指令
ls /data/bkce/paas_agent/apps/projects | awk '{print $1}' | sed 's/.$/ /' | xargs -n 1 ./bkcec stop saas-o

# 若为多台部署, 请在中控机 /data/install 目录下使用如下指令
rcmd root@$APPO_IP "ls /data/bkce/paas_agent/apps/projects" | xargs -n 1 ./bkcec stop saas-o

# 不论单机还是多台部署, 建议在 appo 的服务器上确认应用的进程真正停掉, 若存在未停掉进程, 可以采用强杀方法
ps -ef | grep bk_
for x in `ls /data/bkce/paas_agent/apps/projects | awk '{print $1}' | sed 's/.$/ /'`; do
  ps -ef | grep $x | grep -v grep | awk '{print $2}' | xargs -n 1 kill -9; done
```

重新生成配置

和 MySQL 相关的模块为 MySQL, PaaS, Job, bkdata, SaaS

```
echo mysql paas job bkdata | xargs -n 1 ./bkcec render
```

社区版 5.1+bcs 环境中与 MySQL 相关的模块为 MySQL, PaaS, Job, bkdata, SaaS, iam

```
echo mysql paas job bkdata iam | xargs -n 1 ./bkcec render
```

bkdata 特别注意, 有 3 个地方

```
配置1:
/data/bkce/bkdata/databus/conf/redis.cluster.properties 确认 connector.redis.auth=redis密码

配置2:
/data/bkce/bkdata/databus/conf/jdbc.cluster.properties 确认 connector.connection.password=mysql密码

配置3:
/data/bkce/bkdata/databus/conf/etl.cluster.properties 确认 cc.cache.passwd=mysql密码
```


更改 saas-o 应用的密码

```
# 在 appo 服务器上，先测试一下，确认打印出来的是新密码，注意有*的话，注意转义
find /data/bkce/paas_agent/apps/projects/bk_*/conf -name "*.conf" | grep "bk" | xargs g
rep "老密码" -l | xargs sed "s/老密码/新密码/g"

# 测试没问题，加-i，修改文件
find /data/bkce/paas_agent/apps/projects/bk_*/conf -name "*.conf" | grep "bk" | xargs g
rep "老密码" -l | xargs sed -i "s/老密码/新密码/g"
```

启动 MySQL

```
./bkcec start mysql
```

重新初始化 MySQL

```
./bkcec initdata mysql
```

重新初始化 PaaS

```
./bkcec initdata paas
```

重新启动平台

```
echo paas gse cmdb kafka job bkdata | xargs -n 1 ./bkcec start
```

社区版 5.1+bcs 环境中与 MySQL 相关的模块为 MySQL，PaaS，Job，bkdata，SaaS，iam

```
echo paas gse cmdb kafka job bkdata iam | xargs -n 1 ./bkcec start
```

重新初始化 APPO

```
# 在中控机
./bkcec stop appo
./bkcec initdata appo
./bkcec start appo
./bkcec activate appo
```

重新启动应用

```
# 若为单机部署，请使用如下指令
ls /data/bkce/paas_agent/apps/projects | awk '{print $1}' | sed 's/.$//' | xargs -n 1 .
/bkcec start saas-o

# 若为多台部署，请在中控机 /data/install 目录下使用如下指令
rncmd root@$APPO_IP "ls /data/bkce/paas_agent/apps/projects" | xargs -n 1 ./bkcec start
saas-o
```

检查

确保 CMDB，JOB，蓝鲸监控等模块功能全部 OK

```
# 1.确认 bkdata 任务
$ /data/bkce/bkdata/dataapi/bin/check_databus_status.sh
=====TSDB=====
  % Total      % Received % Xferd  Average Speed   Time    Time       Time  Current
   Dload  Upload   Total     Spent    Left     Speed
100   773   100   773    0     0  14599      0 --:--:-- --:--:-- --:--:-- 14865
tsdb_2_system_cpu_summary
{"name":"tsdb_2_system_cpu_summary","connector":{"state":"RUNNING","worker_id":"10.X.X.X:10054"},"tasks":[{"state":"RUNNING","id":0,"worker_id":"10.X.X.X:10054"}]}

=====MYSQL=====
  % Total      % Received % Xferd  Average Speed   Time    Time       Time  Current
   Dload  Upload   Total     Spent    Left     Speed
100    27   100    27    0     0   1118      0 --:--:-- --:--:-- --:--:-- 1125
jdbc_2_ja_gse_proc_port
{"name":"jdbc_2_ja_gse_proc_port","connector":{"state":"RUNNING","worker_id":"10.X.X.X:10051"},"tasks":[{"state":"RUNNING","id":0,"worker_id":"10.X.X.X:10051"}]}

=====ETL=====
  % Total      % Received % Xferd  Average Speed   Time    Time       Time  Current
   Dload  Upload   Total     Spent    Left     Speed
100  1040   100  1040    0     0   166k      0 --:--:-- --:--:-- --:--:-- 169k
etl_1001_2_system_cpu_summary
{"name":"etl_1001_2_system_cpu_summary","connector":{"state":"RUNNING","worker_id":"10.X.X.X:10052"},"tasks":[{"state":"RUNNING","id":0,"worker_id":"10.X.X.X:10052"}]}

# 2.确认 kafka 节点数量
$ /data/bkce/service/zk/bin/zkCli.sh -server zk.service.consul:2181 ls /common_kafka/brokers/ids
.....
WatchedEvent state:SyncConnected type:None path:null
[1, 2, 3]

# 此步主要检查 redis 内是否有快照数据，在 redis 服务器上
source /data/install/utils.fc
$ redis-cli -h $REDIS_IP -p $REDIS_PORT -a $REDIS_PASS
10.X.X.X:6379> AUTH "REDIS密码"
OK
10.X.X.X:6379> SUBSCRIBE 2_snapshot
Reading messages... (press Ctrl-C to quit)
1) "subscribe"
2) "2_snapshot"
3) (integer) 1
1) "message"
2) "2_snapshot"
3) "{\"localTime\": \"2018-08-15 11:18:00\", \"data\": \"{\\\"beat\\\":{\\\"address\\\"":
```

MySQL 清理 binlog 日志方法



MySQL 中的 binlog 日志记录了数据库中数据的变动，便于对数据的基于时间点和基于位置的恢复，但是 binlog 也会日渐增大，占用很大的磁盘空间，因此，要对 binlog 使用正确安全的方法清理掉一部分没用的日志

注意：如下提供的方法仅供用户参考，具体操作请务必按照自己的实际情况设置

手动清理 binlog

若社区版设置了多台 mysql，需查看主库和从库正在使用的 binlog 是哪个文件

```
MySQL [(none)]> show master status\G
***** 1. row *****
      File: mysql-bin.000006
      Position: 97013298
      Binlog_Do_DB:
      Binlog_Ignore_DB:
1 row in set (0.00 sec)

MySQL [(none)]> show slave status\G
Empty set (0.00 sec)
```

在删除 binlog 日志之前，首先对 binlog 日志备份，以防万一

清理方法一：删除指定日期以前的日志索引中 binlog 日志文件

```
purge master logs before '2018-08-01 17:20:00';
```

清理方法二：删除指定日志文件的日志索引中 binlog 日志文件

```
purge master logs to 'mysql-bin.000006';
```

注意

- 时间和文件名一定不可以写错，尤其是时间中的年和文件名中的序号，以防不小心将正在使用的 binlog 删除！！
- 切勿删除正在使用的 binlog！！
- 使用该语法，会将对应的文件和 mysql-bin.index 中的对应路径删除！！

自动清理 binlog

使用如下方法查询当前 binlog 的过期时间，若为 0 表示不过期

```
mysql> show variables like 'expire_logs_days';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| expire_logs_days | 0 |
+-----+-----+
```

使用如下方法设置 binlog 过期时间，设置 30 表示 30 天后自动清理之前的过期日志

```
mysql> set global expire_logs_days = 30;
```

Redis 常见问题

Redis 密码修改

“

如下指引，若无特殊说明，全部在中控机 `/data/install` 目录进行

”

停止 Redis

```
./bkcec stop redis
```

修改 Redis 密码

修改 `globals.env` 里的 `REDIS_PASS` 值，密码不要包含 `[] / : @ ?` 等特殊字符

同步 install 目录

```
./bkcec sync common
```

重新生成配置

```
echo bkdata fta gse job cmdb paas redis | xargs -n 1 ./bkcec render
```

关闭相关服务

```
echo bkdata fta gse job cmdb paas | xargs -n 1 ./bkcec stop
```

更新 ZooKeeper 内 Redis 密码

此步有 2 种方式，推荐方式 1

- 方式 1：通过命令修改 zk 内 redis 密码

```
# 修改方法，注意把引号内新密码调整为 redis 的新密码，密码不要包含 [ ] / : @ ? 等特殊字符
$ /data/bkce/service/zk/bin/zkCli.sh -server zk.service.consul:2181
Connecting to zk.service.consul:2181
Welcome to ZooKeeper!
JLine support is enabled
[zk: zk.service.consul:2181(CONNECTED) 0]
```

```
WATCHER::
```

```
WatchedEvent state:SyncConnected type:None path:null
```

```
# 此处获取到老密码
```

```
[zk: zk.service.consul:2181(CONNECTED) 0] get /gse/config/etc/dataserver/storage/all/0_1
```

```
([{"host":"redis.service.consul","port":6379,"type":4,"passwd":"老密码位置"}])
```

```
cZxid = 0x100000096
```

```
ctime = Mon Aug 20 11:32:05 CST 2018
```

```
mZxid = 0x200015eec
```

```
mtime = Thu Aug 23 19:38:04 CST 2018
```

```
pZxid = 0x100000096
```

```
cversion = 0
```

```
dataVersion = 5
```

```
aclVersion = 0
```

```
ephemeralOwner = 0x0
```

```
dataLength = 79
```

```
numChildren = 0
```

```
# 此处使用 set 命令，设置新密码，严格按照上面获取的串，仅修改密码位置
```

```
[zk: zk.service.consul:2181(CONNECTED) 1] set /gse/config/etc/dataserver/storage/all/0_1 [{"host":"redis.service.consul","port":6379,"type":4,"passwd":"新密码"}]
```

```
# 可以再查询是否为新的密码，为新的密码表示 OK
```

```
[zk: zk.service.consul:2181(CONNECTED) 1] get /gse/config/etc/dataserver/storage/all/0_1
```

- 方式2：可以通过重新安装 gse 来实现

```
./bkcec stop gse
./bkcec install gse 1
./bkcec start gse
```

修改 bkdata databus 的配置

在 bkdata 服务器上修改 `/data/bkce/bkdata/databus/conf/redis.cluster.properties` 配置，新增 `connector.redis.auth=新密码` 配置

注意新密码不要用任何符号引起来，类似单引号，双引号

```
connector.redis.auth=新密码
```

启动服务验证

```
echo redis paas cmdb gse job bkdata fta | xargs -n 1 ./bkcec start
```

确保 JOB，CMDB，蓝鲸监控等模块功能全部 OK

ZooKeeper 常见问题

ZooKeeper 无法启动

表象

使用 `./bkcec start zk` 提示 started, 但很快 EXIT 退出

原因

1. java 是否安装正常
2. 2181 端口被占用
3. 上述 1、2 均正常, 仍无法启动的异常

解决方法

- 原因 1: 检查 java 版本及 java 环境是否正常

```
# 使用 java 或者 java -version 命令来验证
$ java
$ java -version
```

- 原因 2: 检查端口是否被占用

```
# 检查端口
$ netstat -apn | grep 2181
tcp        0      0 :::2181                :::*                    LISTEN
1403/java
$ kill -9 1403
$ netstat -apn | grep 2181
$
# 重新启动
```

- 原因 3: 脏数据导致启动 zk 失败

通过查看 zk 日志如下报错则是 zk 脏数据问题

`/data/bkce/logs/zk/zookeeper.log`

```
[root@rbtnode1 /data/install]$ cat /data/bkce/logs/zk/zookeeper.log |grep version-2
2019-11-14 10:01:16,503 [myid:1] - INFO [PurgeTask:PurgeTxnLog@147] - Removing file: Nov 13, 2019 6:16:55 AM /data/bkce/public/zk/datalog/version-2/log.1003bbb20
2019-11-14 10:01:16,522 [myid:1] - INFO [PurgeTask:PurgeTxnLog@147] - Removing file: Nov 13, 2019 6:16:55 AM /data/bkce/public/zk/data/version-2/snapshot.1003cbc12
2019-11-14 10:01:16,612 [myid:1] - INFO [main:FileSnap@171] - invalid snapshot /data/bkce/public/zk/data/version-2/snapshot.1003cbc12
java.io.FileNotFoundException: /data/bkce/public/zk/data/version-2/snapshot.1003cbc12 (No such file or directory)
2019-11-14 10:01:16,613 [myid:1] - INFO [main:FileSnap@83] - Reading snapshot /data/bkce/public/zk/data/version-2/snapshot.10043a527
2019-11-14 10:01:18,416 [myid:1] - INFO [QuorumPeer[myid=1]/10.0.5.92:2181:ZooKeeperServer@173] - Created server with tickTime 2000 minSessionTimeout 4000 maxSessionTimeout
```

```
40000 datadir /data/bkce/public/zk/data/log/version-2 snapdir /data/bkce/public/zk/data/
version-
```

```
# 进入到zk data目录, 找到 zoo.conf 中配置的 dataDir 和 dataLogDir 路径。然后删除两个文件夹下的
version -2文件夹
$ cd /data/bkce/public/zk/data

# 把有 zookeeper_server.pid 以及 version-X 开头的文件和文件夹删掉
$ rm -rf version-1 zookeeper_server.pid

# 重新启动 zk, 即可解决
$ ./bkcec start zk
```

上述 3 条确认后, 仍无法启动的, 可能 zk.sh 版本不对, 请和蓝鲸运营人员联系, 联系蓝鲸助手 QQ: 800802001

RabbitMQ 常见问题

RabbitMQ 启动失败

4.1 社区版本 RabbitMQ 启动失败问题处理

表象: 在部署蓝鲸 JOB 过程中需要进行 RabbitMQ 的安装, 数据初始化, 激活步骤, 此问题多发生在此过程

思路方法: 如果是在添加用户和 vhost 时报错, 那么说明启动 rabbitmq-server 没有成功, 通过以下方式确认

```
# 查看进程是否存在
$ ps -ef | grep beam

# 查看监听端口是否存在(5672, 15672, 25672 三个端口必须都在)
$ netstat -tnlpu | grep 5672
```

若没有启动, 通过 `systemctl start rabbitmq-server` 启动。若系统没有 systemctl 命令, 通过 `service rabbitmq start` 启动

首先排查 `/data/bkce/etc/rabbitmq` 目录, 对 RabbitMQ 用户是否有读权限, `/data/bkce/public/rabbitmq` 目录对 RabbitMQ 用户是否有写权限

自己处理好目录的权限问题后, 再尝试重启 `rabbitmq-server`

rabbitmq activate 失败

表象: 此问题发生在 `./bk_install app_mgr`, 会发生如下报错

```
$ [X.X.X.X] register and activate rabbitmq failed. request env: .
$ [X.X.X.X] api response: {"msg": "HTTPConnectionPool(host='X.X.X.X', port=15672): Max r
etries exceeded with url: /api/overview (caused by NewConnectionError('<requests.packag
es.urllib3.connection.HTTPConnection object at 0x7fc5175c4e10>: Failed to establish a ne
w connection: [Errno 111] Connection refused',))"}

```

思路方法：

1. 确认 umask，若不是 022，修改 `/etc/profile`，然后 `source /etc/profile`，再卸载 RabbitMQ，重新安装
2. 确认在安装过程，或在 `rabbitmq activate` 前主机域名是否做过调整修改

rabbitmq initdata 失败

表象：在部署蓝鲸 JOB 过程中需要进行 RabbitMQ 的安装，数据初始化，激活步骤，此问题多发生在此过程

```
[ root@rbtnode1 install)# ./bkcec initdata rabbitmq
Warning: Permanently added '10.x.x.x' (RSA) to the list of known hosts.
bash: line 5: systemctl: command not found

Creating user "admin"
Error: unable to connect to node rabbit@rbtnode1 : nodedown
DIAGNOSTICS
=====

nodes in question: [ rabbit@rbtnode1 ]
hosts, their running nodes and ports :
- rbtnode1: Hrabbitmqctl127684,36040))

current node details :
- node name: rabbitmqctl127684@rbtnode1
- home dir: /var/lib/rabbitmq
- cookie hash: UgOyBrCIoJjXfjMxhu7+Dg::
add rabbitmq user admin failed .
[10.x.x.x] 20180828.130149.337      add rabbitmq user admin failed

```

思路方法：通过以下方式来解决

“

注意：若系统没有 `systemctl` 命令，注意修改下 `/data/install/Utils.fc` 文件，查找
到 `init_rabbitmq_cluster ()` 函数，把 `systemctl start rabbitmq-server` 修改
为 `service rabbitmq-server start`

”

```
# rabbitmq 现在是运行状态？是的话。
./bkcec stop rabbitmq
rm -rf /root/.erlang.cookie /var/lib/rabbitmq/* /data/bkce/public/rabbitmq/*
# 确认 rabbitmq 进程真正停掉，若存在未停掉的，使用如下强制停掉

```



```
ps -ef | grep rabbitmq | awk '{print $2}' | xargs -n 1 kill -9

# 若系统没有 systemctl 命令, 注意修改下/data/install/utils.fc文件, 查找到init_rabbitmq_cluster
r ()函数, 把 systemctl start rabbitmq-server修改为service rabbitmq-server start
1773 init_rabbitmq_cluster () {
1774     ckv=$(uuid -v4)
1775     cookie=/var/lib/rabbitmq/.erlang.cookie
1776     rcmd root@$RABBITMQ_IP "
1777         echo -n $ckv >$cookie;
1778         echo -n $ckv >/root/${cookie##*/};
1779         chown rabbitmq:rabbitmq $cookie;
1780         chmod 400 $cookie /root/${cookie##*/};
1781         systemctl start rabbitmq-server"

# 再手动重新进行数据初始化
./bkcec initdata rabbitmq

# 初始化成功后, 在 /data/install/.bk_install.step 文件里面, 把下面的加进去, 防止安装时再报错
initdata rabbitmq
```

rabbitmq 15672 不存在

如果是在激活 RabbitMQ 时报错**15672** 端口拒绝链接, 那说明 `rabbitmq-server` 没有成功加载 `rabbitmq_management` 插件

原因可能有 2 种:

1. umask 不正确, 导致无法访问对应目录
2. 主机名发生变更, 导致节点发现异常

Kafka 常见问题

Kafka 常用操作

- Kafka 查询 topic

```
$ /data/bkce/service/kafka/bin/kafka-topics.sh --zookeeper zk.service.consul:2181/common_kafka --describe | grep Topic
```

- 查看 topic 状态

```
$ /data/bkce/service/kafka/bin/kafka-topics.sh --zookeeper zk.service.consul:2181/common_kafka --describe --topic connect-configs.tsdb
```

- 查看 topic 能否读

```
$ /data/bkce/service/kafka/bin/kafka-console-consumer.sh --bootstrap-server kafka.service.consul:2181/common_kafka --topic connect-configs.tsdb
```

```
ce.consul:9092 --topic connect-configs.tsdb --from-beginning | head
```

- 确认实时的 topic 能否读

```
$ /data/bkce/service/kafka/bin/kafka-console-consumer.sh --bootstrap-server kafka.service.consul:9092 --topic connect-configs.etl --from-beginning | head
```

Kafka broker 节点缺失

若社区版为 3 台部署的，必须返回[1, 2, 3]才正常，示例如下

若 brokers ids 不为[1, 2, 3]，可能存在 `/data/bkce/public/kafka/.lock` 文件，有的话，删除此文件，再重新使用 `./bkcec stop kafka` 和 `./bkcec start kafka` 重启 kafka，重启完再次确认状态

```
[root@rbtnode1 /data/install]# /data/bkce/service/zk/bin/zkCli.sh -server zk.service.consul:2181 ls /common_kafka/brokers/ids
Connecting to zk.service.consul:2181
log4j:WARN No appenders could be found for logger (org.apache.zookeeper.ZooKeeper).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
WATCHER::

WatchedEvent state:SyncConnected type:None path:null
[1, 2, 3]
```

kafka 数据或日志清理

“

Kafka 将数据持久化到了硬盘上，允许配置一定的策略对数据清理，清理的策略有两个，删除和压缩

严格注意：下面清理策略，请根据实际业务，服务器状况，及需求来定制

”

有如下 2 种方式进行设置

- 方式一：通过调整配置文件

```
# 配置文件位置
/data/bkce/service/kafka/config/server.properties

# 可以增加 log.cleanup.policy 这个数据清理方式设置，此行为删除动作
log.cleanup.policy=delete

# 下面有 2 种方式，保留时间或大小，请自行根据实际情况调整此处设置，1G 为 1073741824。具体保留大小根据实际情况设置
# 注意：下面为直接删除，删除后的消息不可恢复
log.retention.hours=168(超过指定时间168小时后，删除旧的消息)
```

```
log.retention.bytes=10737418240(超过指定大小10G后，删除旧的消息)
```

设置完毕，重启服务来生效

- 方式二：Kakfa 设置 Topic 过期时间

```
# 设置过期时间，只能用毫秒(retention.ms)，或者 bytes(retention.bytes)
$ /data/bkce/service/kafka/bin/kafka-topics.sh --zookeeper zk.service.consul:2181/commo
n_kafka --topic snapshot2 --alter --config retention.ms=17280000
$ WARNING: Altering topic configuration from this script has been deprecated and may be
removed in future releases.
    Going forward, please use kafka-configs.sh for this functionality
$ updated config for topic "snapshot2"
```

Kafka gse_data 报错

在 gse 的模块 gse_data 的日志中，会出现有如下报错，这种是 Kafka 消息机制的正常行为，只要确定快照数据 OK，就可确认 `gse_data->kafka->bkdata_>cmdb` 的链路正常

```
52 [2018-08-23 16:47:05.109] <11297--805308672>[ERROR][kafka_producer:18]KAFKA-3-E
RROR: rdkafka#producer-15 10.X.X.X:9092/1: Receive failed: Disconnected
53 [2018-08-23 16:47:05.614] <11297--318793984>[ERROR][kafka_producer:18]KAFKA-3-E
RROR: rdkafka#producer-4 kafka.service.consul:9092/bootstrap: Receive failed: Disconnec
ted
54 [2018-08-23 16:52:05.198] <11297--176183552>[ERROR][kafka_producer:18]KAFKA-3-E
RROR: rdkafka#producer-12 10.X.X.X:9092/1: Receive failed: Disconnected
55 [2018-08-23 16:52:05.936] <11297--998275328>[ERROR][kafka_producer:18]KAFKA-3-E
RROR: rdkafka#producer-14 10.178.181.35:9092/3: Receive failed: Disconnected
56 [2018-08-23 16:57:05.115] <11297--956311808>[ERROR][kafka_producer:18]KAFKA-3-E
RROR: rdkafka#producer-16 10.178.181.35:9092/3: Receive failed: Disconnected
57 [2018-08-23 16:57:05.115] <11297--1166031104>[ERROR][kafka_producer:18]KAFKA-3-
FAIL: rdkafka#producer-12 kafka.service.consul:9092/bootstrap: Receive failed: Disconne
cted
```

InfluxDB 常见问题

InfluxDB 查询

Influxdb 为蓝鲸监控数据存储载体，在发生蓝鲸监控没有数据时，有个 check 点，确认 InfluxDB 是否正常

检查 InfluxDB 的数据库

```
$ influx -host $INFLUXDB_HOST -port $INFLUXDB_PORT -execute 'show databases'
name: databases
name
----
_internal
system_2
```

检查 InfluxDB 的结构

```
$ influx -host $INFLUXDB_HOST -port $INFLUXDB_PORT -database system_2 -execute 'show measurements'
name: measurements
name
----
system_cpu_detail_2
system_cpu_summary_2
system_disk_2
system_env_2
system_inode_2
system_io_2
system_load_2
system_mem_2
system_net_2
system_netstat_2
system_proc_2
system_swap_2
```

检查 InfluxDB 的数据

```
> select * from system_cpu_detail_2 limit 10;
name: system_cpu_detail_2
time                company_id device_name hostname idle          iowait
ip                  plat_id stolen system          usage          user
----
--
1535439967000000000 0          cpu0          rbtnode1 0.6058552226105512 0.06945605603853
58  10.x.x.x 0          0          0.04019670092112785 48.09914587171289 0.2838099103946990
5
1535439967000000000 0          cpu7          rbtnode1 0.7863101634785973 0.00837019347933
527 10.x.x.x 0          0          0.030681614533304973 24.205951186893405 0.1741338821686602
1
1535439967000000000 0          cpu2          rbtnode1 0.6678563495185631 0.00910008103576
8667 10.x.x.x 0          0          0.040028319084854255 35.79896476874237 0.2824698411674115
1535439967000000000 0          cpu6          rbtnode1 0.7829082548721787 0.01285437762695
7599 10.x.x.x 0          0          0.03067846131995839 24.644470470134607 0.1730757661552659
1535439967000000000 0          cpu5          rbtnode1 0.7646966906635414 0.02864247136806
9893 10.x.x.x 0          0          0.030657712793992715 30.637870416875536 0.1755306285693026
1535439967000000000 0          cpu1          rbtnode1 0.6646903677764837 0.01120261179190
2923 10.x.x.x 0          0          0.03985501626475787 35.816906114265606 0.2837339392108164
1535439967000000000 0          cpu4          rbtnode1 0.7346243011891722 0.05592099142758
973  10.x.x.x 0          0          0.0312518993637984 38.19095477386919 0.1776992052751767
1535439967000000000 0          cpu3          rbtnode1 0.6710939376132079 0.00795270572568
7499 10.x.x.x 0          0          0.0398088272799649 34.2078877005351 0.2806170793291083
4
1535440027000000000 0          cpu4          rbtnode1 0.7345565004723897 0.05588967150219
944  10.x.x.x 0          0          0.03126793860725039 28.370927318295703 0.1777816774563495
1535440027000000000 0          cpu3          rbtnode1 0.6710074762738196 0.00793005575830
115  10.x.x.x 0          0          0.03980792783012481 35.225375626044226 0.2807265704162614
```

开发常见问题

开发环境与配置类

pip install 安装某个包时，提示包未找到

可能是仓库源配置问题。建议使用腾讯云提供的 PYPI 源：

`mirrors.cloud.tencent.com/pypi/simple/`

用户与鉴权相关

如何模拟登录蓝鲸框架，需要哪些值？

只要确保 cookie 中的 bk_ticket 和 bk_uid 一样，就可以复用同样的登录态

SaaS 开发框架相关

如何判断开发框架当前加载的是那个配置文件？能否自己指定？

1. 三种配置文件分别对应：本地开发时，加载 dev；预发布环境下，加载 stag；正式环境下，加载 prod。
2. 不能指定，因为框架写死通过运行环境加载。

如何确认当前框架所在的运行环境？

可以使用 `RUN_MODE` 变量，它被定义在 `dev.py` / `stag.py` / `prod.py` 文件的开头位置。

```
from django.conf import settings
settings.RUN_MODE
```

蓝鲸框架在哪里添加全局配置？

可以在 `conf` 目录下的 `default.py` 文件里添加自定义全局配置。

SaaS 开发相关

应用内如何获取请求客户端 IP 地址？

请求在到达应用前，会经过多层负载均衡服务。如果应用直接拿请求 IP 作为客户端 IP，通常是有问题的，因为这个 IP 是负载均衡器 IP，而非真实客户端 IP。想要获取真实 IP，需要解析请求头信息里的 `X-Forwarded-For` 字段。

`X-Forwarded-For` 头信息里包含的是以逗号连接的 IP 地址列表，如 `8.8.8.8,4,4,4,4`。第一个 `8.8.8.8` 就是客户端 IP。

下面是一段在 Django 框架内获取客户端 IP 的代码样例：

```
def get_client_ip(request):
```

```
"""Get real client IP address from request
"""
# Try "x-forwarded-for" header
x_forwarded_for = request.META.get('HTTP_X_FORWARDED_FOR')
if x_forwarded_for:
    return x_forwarded_for.split(',')[0].strip()
return request.META.get('REMOTE_ADDR')
```

django 如何对通过接口请求的数据进行gzip压缩？

1. 添加 `django.middleware.gzip.GZipMiddleware` 中间件。
2. 在视图文件中添加 `from django.views.decorators.gzip import gzip_page`。
3. 给视图函数添加上 `@gzip_page` 装饰器，返回的数据将会被压缩。

本地访问前端，{{ SITE_URL }} Django 模版语法未生效

查检本地环境 dev.py 内是否有 `SITE_URL='xxx'` 的定义，如没有，请在本地环境 dev.py 里添加 `SITE_URL='xxx'` 变量即可。