

GBASE

GBase 8s V8.8.5

工具与命令参考手册



GBase 8s工具与命令参考手册，南大通用数据技术股份有限公司**GBase** 版权所有©2024，保留所有权利

版权声明

本文档所涉及的软件著作权及其他知识产权已依法进行了相关注册、登记，由南大通用数据技术股份有限公司合法拥有，受《中华人民共和国著作权法》、《计算机软件保护条例》、《知识产权保护条例》和相关国际版权条约、法律、法规以及其它知识产权法律和条约的保护。未经授权许可，不得非法使用。

免责声明

本文档包含的南大通用数据技术股份有限公司的版权信息由南大通用数据技术股份有限公司合法拥有，受法律的保护，南大通用数据技术股份有限公司对本文档可能涉及到的非南大通用数据技术股份有限公司的信息不承担任何责任。在法律允许的范围内，您可以查阅，并仅能够在《中华人民共和国著作权法》规定的合法范围内复制和打印本文档。任何单位和个人未经南大通用数据技术股份有限公司书面授权许可，不得使用、修改、再发布本文档的任何部分和内容，否则将视为侵权，南大通用数据技术股份有限公司具有依法追究其责任的权利。

本文档中包含的信息如有更新，恕不另行通知。您对本文档的任何问题，可直接向南大通用数据技术股份有限公司告知或查询。

未经本公司明确授予的任何权利均予保留。

通讯方式

南大通用数据技术股份有限公司

天津市高新区华苑产业园区工华道2号天百中心3层(300384)

电话：400-013-9696

邮箱：info@gbase.cn

商标声明

GBASE[®] 是南大通用数据技术股份有限公司向中华人民共和国国家商标局申请注册的注册商标，注册商标专用权由南大通用数据技术股份有限公司合法拥有，受法律保护。未经南大通用数据技术股份有限公司书面许可，任何单位及个人不得以任何方式或理由对该商标的任何部分进行使用、复制、修改、传播、抄录或与其它产品捆绑使用销售。凡侵犯南大通用数据技术股份有限公司商标权的，南大通用数据技术股份有限公司将依法追究其法律责任。

目 录

目 录	II
1 工具一览表	1
2 客户端工具	4
2.1 gsql	4
2.1.1 gsql 概述	4
2.1.1.1 变量	6
2.1.1.2 SQL 代换	9
2.1.1.3 提示符	9
2.1.2 使用指导	13
2.1.3 获取帮助	16
2.1.4 命令参考	19
2.1.5 元命令参考	26
2.1.6 常见问题处理	47
2.1.6.1 连接性能问题	47
2.1.6.2 创建连接故障	48
2.1.6.3 其他故障	52
3 服务端工具	52
3.1 gs_cgroup	53
3.1.1 背景信息	53
3.1.2 语法	53
3.1.3 参数说明	53
3.1.4 示例	60
3.2 gs_check	68
3.2.1 背景信息	68

3.2.2	语法	68
3.2.3	参数说明	69
3.2.4	用户自定义场景	88
3.2.5	用户自定义检查项	89
3.2.6	操作系统参数	92
3.2.7	文件系统参数	98
3.2.8	示例	99
3.2.9	相关命令	101
3.3	gs_checkos	101
3.3.1	背景信息	101
3.3.2	语法	101
3.3.3	参数说明	102
3.3.4	操作系统参数	107
3.3.5	文件系统参数	113
3.3.6	示例	113
3.3.7	相关命令	115
3.4	gs_checkperf	115
3.4.1	背景信息	115
3.4.2	语法	115
3.4.3	参数说明	116
3.4.4	示例	118
3.4.5	相关命令	122
3.5	gs_collector	122
3.5.1	背景信息	122
3.5.2	语法	126
3.5.3	参数说明	127

3.5.4 示例	129
3.6 gs_dump	131
3.6.1 背景信息	131
3.6.2 主要功能	132
3.6.3 语法	134
3.6.4 参数说明	134
3.6.5 说明	144
3.6.6 示例	145
3.6.7 相关命令	147
3.7 gs_dumpall	147
3.7.1 背景信息	147
3.7.2 语法	148
3.7.3 参数说明	148
3.7.4 说明	154
3.7.5 示例	154
3.7.6 相关命令	155
3.8 gs_guc	155
3.8.1 背景信息	155
3.8.2 语法	156
3.8.3 参数参考	160
3.8.4 示例	164
3.9 gs_encrypt	166
3.9.1 背景信息	166
3.9.2 语法	166
3.9.3 参数说明	167
3.9.4 示例	167

3.10 gs_om.....	168
3.10.1 背景信息.....	168
3.10.2 语法.....	169
3.10.3 参数说明.....	170
3.10.4 示例.....	177
3.11 gs_plan_simulator.....	180
3.11.1 背景信息.....	180
3.11.2 使用步骤.....	180
3.11.3 语法.....	181
3.11.4 参数说明.....	182
3.12 gs_restore.....	182
3.12.1 背景信息.....	182
3.12.2 命令格式.....	183
3.12.3 参数说明.....	183
3.12.4 示例.....	190
3.12.5 相关命令.....	192
3.13 gs_ssh.....	192
3.13.1 背景信息.....	192
3.13.2 语法.....	193
3.13.3 参数说明.....	193
3.13.4 示例.....	193
3.14 gs_sdr.....	194
3.14.1 背景信息.....	194
3.14.2 前提条件.....	194
3.14.3 语法.....	194
3.14.4 参数说明.....	194

3.14.5 示例	200
4 系统内部命令	207
4.1 dsscmd	207
4.1.1 概述	207
4.1.2 命令	207
4.1.3 参数说明	211
4.1.4 使用示例	213
4.2 dssserver	213
4.2.1 背景信息	213
4.2.2 语法	214
4.2.3 参数说明	214
4.3 gaussdb	219
4.3.1 背景信息	219
4.3.2 语法	219
4.3.3 描述	219
4.3.4 参数说明	220
4.3.5 环境变量	226
4.3.6 错误处理	227
4.3.7 用法	228
4.3.8 示例	228
4.4 gs_backup	229
4.4.1 背景信息	229
4.4.2 前提条件	229
4.4.3 语法	229
4.4.4 参数说明	229
4.4.5 示例	231

4.5	gs_basebackup	232
4.5.1	背景信息	232
4.5.2	前提条件	232
4.5.3	语法	233
4.5.4	参数说明	234
4.5.5	示例	236
4.6	gs_ctl	236
4.6.1	背景信息	236
4.6.2	参数说明	237
4.7	gs_initdb	247
4.7.1	背景信息	247
4.7.2	系统表	247
4.7.3	使用指导	247
4.7.4	语法	248
4.7.5	操作步骤	248
4.7.6	参数说明	248
4.8	gs_install	256
4.8.1	背景信息	256
4.8.2	前提条件	256
4.8.3	语法	257
4.8.4	参数说明	257
4.8.5	示例	258
4.8.6	相关命令	259
4.9	gs_postuninstall	259
4.9.1	背景信息	259
4.9.2	前提条件	259

4.9.3	语法	260
4.9.4	参数说明	260
4.9.5	示例	261
4.9.6	相关命令	262
4.10	gs_preinstall	262
4.10.1	背景信息	262
4.10.2	注意事项	262
4.10.3	语法	262
4.10.4	参数说明	263
4.10.5	示例	266
4.10.6	相关命令	269
4.11	gs_sshexkey	270
4.11.1	背景信息	270
4.11.2	前提条件	270
4.11.3	语法	271
4.11.4	参数说明	271
4.11.5	示例	272
4.12	gs_tar	273
4.12.1	语法	273
4.12.2	参数说明	273
4.12.3	示例	273
4.13	gs_uninstall	273
4.13.1	背景信息	273
4.13.2	语法	274
4.13.3	参数说明	274
4.13.4	示例	275

4.13.5 相关命令	275
4.14 gs_upgradectl	275
4.14.1 背景信息	275
4.14.2 注意事项	275
4.14.3 语法	277
4.14.4 参数说明	278
4.14.5 示例	278
4.15 gs_expansion	281
4.15.1 背景信息	281
4.15.2 语法	282
4.15.3 参数说明	283
4.15.4 示例	284
4.15.5 相关命令	286
4.16 gs_dropnode	286
4.16.1 背景信息	286
4.16.2 注意事项	286
4.16.3 前提条件	287
4.16.4 语法	287
4.16.5 参数说明	288
4.16.6 示例	288
4.16.7 相关命令	289
4.17 gs_probackup	289
4.17.1 背景信息	289
4.17.2 前提条件	290
4.17.3 限制说明	290
4.17.4 命令说明	291

4.17.5	参数说明	294
4.17.6	备份流程	304
4.17.7	故障处理	305
4.18	gstrace	305
4.18.1	功能介绍	305
4.18.2	语法	306
4.18.3	参数说明	306
4.18.4	示例	308
4.19	kdb5_util	309
4.19.1	功能介绍	309
4.19.2	语法	309
4.19.3	参数说明	309
4.20	kadmin.local	309
4.20.1	功能介绍	309
4.20.2	语法	310
4.20.3	参数说明	310
4.21	kinit	310
4.21.1	功能介绍	310
4.21.2	语法	310
4.21.3	参数说明	310
4.22	klist	310
4.22.1	功能介绍	310
4.22.2	语法	310
4.22.3	参数说明	311
4.23	krb5kdc	311
4.23.1	功能介绍	311

4.23.2 语法	311
4.23.3 参数说明	311
4.24 kdestroy	311
4.24.1 功能介绍	311
4.24.2 语法	311
4.24.3 参数说明	311
4.25 pg_config	312
4.25.1 功能介绍	312
4.25.2 语法	312
4.25.3 参数说明	312
4.26 pg_controldata	314
4.26.1 功能介绍	314
4.26.2 前提条件	314
4.26.3 语法	314
4.26.4 参数说明	315
4.27 pg_recvlogical	315
4.27.1 功能介绍	315
4.27.2 语法	315
4.27.3 参数说明	315
4.27.4 示例	318
4.28 pg_resetxlog	319
4.28.1 功能介绍	319
4.28.2 前提条件	319
4.28.3 语法	319
4.28.4 参数说明	319
4.29 pg_archivecleanup	320

4.29.1 功能介绍	320
4.29.2 语法	320
4.29.3 参数说明	320
4.29.4 示例	321
4.29.5 扩展应用	321
4.30 pssh	321
4.30.1 基本功能	321
4.30.2 语法	321
4.30.3 参数说明	321
4.31 pscp	322
4.31.1 基本功能	322
4.31.2 语法	322
4.31.3 参数说明	323
4.32 transfer.py	324
4.32.1 功能介绍	324
4.32.2 前提条件	324
4.32.3 语法	324
4.32.4 参数说明	325
5 集群管理 (Cluster Manager)	326
5.1 特性介绍	326
5.1.1 cm_agent	326
5.1.2 cm_server	327
5.1.3 自定义资源	328
5.1.4 资源池化	332
5.1.5 CM 支持 DN 仲裁	333
5.1.6 支持集群信息查询和推送	335

5.2 cm_ctl 工具介绍.....	336
5.2.1 命令说明.....	337
5.2.2 命令参考.....	352
5.3 cm_persit 工具介绍.....	354
5.4 安装和卸载工具介绍.....	355
5.5 安全设计.....	356
5.5.1 手动替换证书步骤.....	356
5.5.2 证书使用指南.....	357
5.5.3 ifconfig 提权步骤.....	358
5.6 CM 配置参数介绍.....	358
5.6.1 cm_agent 参数.....	358
5.6.2 log_dir.....	358
5.6.3 log_file_size.....	358
5.6.4 log_min_messages.....	359
5.6.5 incremental_build.....	359
5.6.6 security_mode.....	359
5.6.7 upgrade_from.....	359
5.6.8 alarm_component.....	360
5.6.9 alarm_report_interval.....	360
5.6.10 agent_heartbeat_timeout.....	360
5.6.11 agent_connect_timeout.....	360
5.6.12 agent_connect_retries.....	361
5.6.13 agent_kill_instance_timeout.....	361
5.6.14 agent_report_interval.....	361
5.6.15 alarm_report_max_count.....	361
5.6.16 agent_check_interval.....	361

5.6.17 enable_log_compress	362
5.6.18 process_cpu_affinity	362
5.6.19 enable_xc_maintenance_mode	362
5.6.20 log_threshold_check_interval	362
5.6.21 log_max_size	363
5.6.22 log_max_count	363
5.6.23 log_saved_days	363
5.6.24 agent_phony_dead_check_interval	364
5.6.25 unix_socket_directory	364
5.6.26 dilatation_shard_count_for_disk_capacity_alarm	364
5.6.27 enable_dcf	365
5.6.28 disaster_recovery_type	365
5.6.29 agent_backup_open	365
5.6.30 disk_timeout	365
5.6.31 voting_disk_path	366
5.6.32 agent_rhb_interval	366
5.6.33 enable_ssl	366
5.6.34 ssl_cert_expire_alert_threshold	366
5.6.35 ssl_cert_expire_check_interval	367
5.6.36 enable_fence_dn	367
5.6.37 event_triggers	367
5.7 cm_server 参数	368
5.7.1 log_dir	368
5.7.2 log_file_size	368
5.7.3 log_min_messages	369
5.7.4 thread_count	369

5.7.5 instance_heartbeat_timeout	369
5.7.6 instance_failover_delay_timeout	369
5.7.7 cmserver_ha_connect_timeout	370
5.7.8 cmserver_ha_heartbeat_timeout	370
5.7.9 cmserver_ha_status_interval	370
5.7.10 cmserver_self_vote_timeout	370
5.7.11 phony_dead_effective_time	370
5.7.12 cm_server_arbitrate_delay_base_time_out	371
5.7.13 cm_server_arbitrate_delay_incremental_time_out	371
5.7.14 alarm_component	371
5.7.15 alarm_report_interval	371
5.7.16 alarm_report_max_count	372
5.7.17 instance_keep_heartbeat_timeout	372
5.7.18 az_switchover_threshold	372
5.7.19 az_check_and_arbitrate_interval	372
5.7.20 az_connect_check_interval	373
5.7.21 az_connect_check_delay_time	373
5.7.22 cmserver_demote_delay_on_etcd_fault	373
5.7.23 instance_phony_dead_restart_interval	373
5.7.24 enable_transaction_read_only	373
5.7.25 datastorage_threshold_check_interval	374
5.7.26 datastorage_threshold_value_check	374
5.7.27 max_datastorage_threshold_check	374
5.7.28 enable_az_auto_switchover	374
5.7.29 cm_krb_server_keyfile	375
5.7.30 switch_rto	375

5.7.31	force_promote	375
5.7.32	backup_open	375
5.7.33	enable_dcf	376
5.7.34	ddb_type	376
5.7.35	enable_ssl	376
5.7.36	ssl_cert_expire_alert_threshold	377
5.7.37	ssl_cert_expire_check_interval	377
5.7.38	ddb_log_level	377
5.7.39	ddb_log_backup_file_count	377
5.7.40	ddb_max_log_file_size	378
5.7.41	ddb_log_suppress_enable	378
5.7.42	ddb_election_timeout	378
5.7.43	coordinator_heartbeat_timeout	378
5.7.44	cluster_starting_aribt_delay	378
5.7.45	enable_e2e_rto	379
5.7.46	disk_timeout	379
5.7.47	agent_network_timeout	379
5.7.48	voting_disk_path	379
5.7.49	share_disk_path	379
5.7.50	dn_arbitrate_mode	380
5.7.51	delay_arbitrate_max_cluster_timeout	380
5.7.52	delay_arbitrate_timeout	380
5.7.53	cm_auth_method	380
5.7.54	third_party_gateway_ip	381
5.7.55	cms_enable_failover_on2nodes	381
5.7.56	cms_enable_db_crash_recovery	381

5.7.57 cms_network_isolation_timeout	381
6 可运行脚本功能说明	382
7 gs_collector 工具支持收集的表和视图列表	385
8 FAQ	389
8.1 执行命令报错“Failed to obtain the GPHOME”	389
8.2 gs_ctl 重建备实例过程被中断导致密钥文件不完整恢复方法	390
8.3 使用 gs_om -t status --all 查询状态长时间没有响应	391
8.4 gs_sshexkey 相同用户不同密码报错	391

1 工具一览表

GBase 8s 提供客户端和服务端应用程序（工具），帮助用户更好地维护 GBase 8s，提供的所有工具如下表所示。工具位于安装数据库服务器的 \$GPHOME/script 和 \$GAUSSHOME/bin 路径下。

表 1-1 工具一览表

分类	工具名称	简介
客户端工具	gsql	gsql 是 GBase 8s 提供在命令行下运行的数据库连接工具, 可以通过此工具连接服务器并对其进行操作和维护, 除了具备操作数据库的基本功能, gsql 还提供了若干高级特性, 便于用户使用。
服务端工具	gs_cgroup	gs_cgroup 是 GBase 8s 提供的负载管理工具。负责创建默认控制组、创建用户自定义控制组、删除用户自定义控制组、更新用户自定义控制组的资源配额和资源限额、显示控制组配置文件内容、显示控制组树形结构和删除用户的所有控制组。
	gs_check	gs_check 改进增强, 统一化当前系统中存在的各种检查工具, 例如 gs_check、gs_checkos 等, 帮助用户在 GBase 8s 数据库运行过程中, 全量的检查运行环境, 操作系统环境, 网络环境及数据库执行环境, 也有助于在数据库重大操作之前对各类环境进行全面检查, 有效保证操作执行成功。
	gs_checkos	gs_checkos 用于检查操作系统、控制参数、磁盘配置等内容, 并对系统控制参数、I/O 配置、网络配置和 THP 服务等信息进行配置。
	gs_checkperf	gs_checkperf 工具可定期对 GBase 8s 级别(主机 CPU

分类	工具名称	简介
		占用率、Gauss CPU 占用率、I/O 使用情况等)、节点级别 (CPU 使用情况、内存使用情况、I/O 使用情况)、会话/进程级别 (CPU 使用情况、内存使用情况、I/O 使用情况)、SSD 性能 (写入、读取性能) 进行检查, 让用户了解 GBase 8s 的负载情况, 采取对应的改进措施。
	gs_collector	gs_collector 在 GBase 8s 发生故障时, 收集 OS 信息、日志信息以及配置文件信息, 来定位问题。
	gs_dump	gs_dump 是一款用于导出数据库相关信息的工具, 支持导出完整一致的数据库对象 (数据库、模式、表、视图等) 数据, 同时不影响用户对数据库的正常访问。
	gs_dumpall	gs_dumpall 是一款用于导出数据库相关信息的工具, 支持导出完整一致的 GBase 8s 数据库所有数据, 同时不影响用户对数据库的正常访问。
	gs_guc	gs_guc 用于设置 GBase 8s 配置文件 (“postgresql.conf”、“pg_hba.conf”) 中的参数, 配置文件中参数的默认值是单机的配置模式, 您可以使用 gs_guc 来设置适合的参数值。
	gs_encrypt	gs_encrypt 是一款用于为输入的明文字符串进行加密操作。
	gs_om	GBase 8s 提供了 gs_om 工具, 帮助进行集群维护, 包括启动、停止、查询集群状态、生成静态配置文件、刷新动态配置文件、SSL 证书替换、启停 kerberos 认证、显示帮助信息和显示版本号信息的功能。

分类	工具名称	简介
	gs_plan_simulator	gs_plan_simulator 工具用于收集与执行计划相关的数据并能够在其它环境上进行执行计划的复现,从而定位执行计划类相关问题。
	gs_restore	gs_restore 是 GBase 8s 提供的针对 gs_dump 导出数据的导入工具。通过此工具可由 gs_dump 生成的导出文件进行导入。
	gs_ssh	GBase 8s 提供了 gs_ssh 工具帮助用户在 GBase 8s 各节点上执行相同的命令。
	gs_sdr	GBase 8s 提供了 gs_sdr 工具,在不借助额外存储介质的情况下实现跨 region 的异地容灾。提供流式容灾搭建,灾备升主,计划内主备切换,容灾解除、容灾状态监控功能、显示帮助信息和显示版本号信息等功能。

2 客户端工具

用户可以通过客户端工具 `gsql`，连接到 GBase 8s 数据库并进行各种操作和调试。

2.1 gsql

`gsql` 是 GBase 8s 提供的命令行下运行的数据库连接工具。用户可以通过此工具连接服务器，并对其进行操作和维护。`gsql` 提供操作数据库的基本功能，以及若干高级特性，便于用户使用。

2.1.1 gsql 概述

基本功能

- 连接数据库：详细操作参见《GBase 8s V8.8.5_5.0.0_数据库管理指南》中“使用 `gsql` 连接”章节。

说明

`gsql` 创建连接时，会有 5 分钟超时时间。如果在这个时间内，数据库未正确地接受连接并对身份进行认证，`gsql` 将超时退出。针对此问题，可以参考常见问题处理。

- 执行 SQL 语句：支持交互式地键入并执行 SQL 语句，也可以执行一个文件中指定的 SQL 语句。
- 执行元命令：元命令可以帮助管理员查看数据库对象的信息、查询缓存区信息、格式化 SQL 输出结果以及连接到新的数据库等。详细说明请参见元命令参考。

高级特性

`gsql` 的高级特性如表 2-1 所示。

表 2-1 gsql 高级特性

特性名称	描述
变量	<code>gsql</code> 提供类似于 Linux 的 shell 命令的变量特性，可以使用 <code>gsql</code> 的 <code>\set</code> 原命令设置变量，格式如下： <pre>\set varname value</pre>

	<p>删除变量的格式如下：</p> <pre style="background-color: #f0f0f0; padding: 5px;">\unset varname</pre> <p>说明</p> <ul style="list-style-type: none"> ➤ 变量只是简单的名称/值对，这里的值可以是任意长度。 ➤ 变量名称必须由字母（包括非拉丁字母）、数字和下划线组成，且对大小写敏感。 ➤ 如果使用 <code>\set varname</code> 的格式（不带第二个参数），则只是设置这个变量而没有给变量赋值。 ➤ 可以使用不带参数的 <code>\set</code> 来显示所有变量的值。
SQL 代换	<p>利用 <code>gsql</code> 的变量特性，可以将常用 SQL 语句设置为变量，以简化操作。</p>
自定义提示符	<p><code>gsql</code> 支持使用用户自定义的提示符。通过修改 <code>gsql</code> 预留的三个变量 <code>PROMPT1</code>、<code>PROMPT2</code>、<code>PROMPT3</code> 来改变提示符。这三个变量的值，可以由用户自定义，也可以使用 <code>gsql</code> 预定义值。</p>
命令自动补齐	<p>根据 GBase 8s 语法规则，<code>gsql</code> 支持使用 <code>Tab</code> 键进行命令的自动补齐功能。当编译时指定了 <code>--with-readline</code> 选项，且客户端连接时指定 <code>-r</code> 参数时，此功能处于打开状态。</p> <p>例如，<code>crea</code> 后按 <code>Tab</code> 键，<code>gsql</code> 会将其补齐为 <code>create</code>。</p> <p>说明</p> <ul style="list-style-type: none"> ➤ 支持数据库 SQL 关键字如 <code>select</code>、<code>create</code>、<code>table</code> 等。 ➤ 支持表名、视图名等自定义标识符的补齐。 ➤ 元命令选项 <code>'S'</code>、<code>'+'</code> 不支持自动补齐。 ➤ 对表进行补齐时，只有前缀是 <code>"pg_"</code> 才会补齐系统表。 ➤ 不支持建表时字段类型的补齐。 ➤ <code>select</code> 后不支持任何补齐。 ➤ 不支持常量与宏的自动补齐。 ➤ <code>select * from a,b...</code> 不支持第二个开始表的自动补齐，<code>insert into t1 (col1, col2, ...)</code> 不支持第二个列的自动补齐。 ➤ 不支持 <code>create tablespace</code> 语句 <code>with</code> 以及 <code>with</code> 后参数的自动补齐。

	<ul style="list-style-type: none"> ➢ 创建索引不支持 local、global 的自动补齐，修改索引不支持 rebuild 自动补齐。 ➢ set 语句仅支持自动补全 user 和 superuser 级别的参数。 ➢ 不支持 if exists 的自动补齐。 ➢ 不支持表名.列名的自动补齐，如 alter sequence <name> owned by tableName.colName, owned by。 ➢ 不支持自定义操作符自动补齐。使用复制粘贴这种方式输入命令，如果粘贴的命令里面有 TAB 键有可能会使输入命令的格式错乱，无法正常执行。
<p>客户端操作历史记录</p>	<p>gsq 支持查看客户端操作历史记录。当客户端连接时指定 -r 参数，此功能处于打开状态。可以通过 \set 设置记录历史的条数，例如，\set HISTSIZE 50，将记录历史的条数设置为 50；\set HISTSIZE 0，不记录历史。</p> <p>说明</p> <ul style="list-style-type: none"> ➢ 客户端操作历史记录条数默认设置为 32 条，最多支持记录 500 条。当客户端交互式输入包含中文字符时，只支持 UTF-8 的编码环境。 ➢ 出于安全考虑，将包含 PASSWORD、IDENTIFIED 敏感词的记录识别为敏感信息，不会记录到历史信息中，即不能通过上下翻回显。

2.1.1.1 变量

可以使用 gsql 元命令 \set 设置一个变量。例如把变量 foo 的值设置为 bar：

```
postgres=# \set foo bar
```

要引用变量的值，在变量前面加冒号。例如查看变量 foo 的值：

```
postgres=# \echo :foo
bar
```

这种变量的引用方法适用于正规的 SQL 语句和元命令。

gsq 预定义了一些特殊变量，同时也规划了变量的取值。为了保证和后续版本最大限度地兼容，请避免以其他目的使用这些变量。所有特殊变量见表 2-2。

说明

- 所有特殊变量都由大写字母、数字和下划线组成。
- 查看特殊变量的默认值，请使用元命令 `\echo :varname`（例如 `\echo :DBNAME`）。

表 2-2 特殊变量设置

变量	设置方法	变量说明
DBNAME	<code>\set DBNAME dbname</code>	设置当前连接的数据库的名称。每次连接数据库时都会被重新设置。
ECHO	<code>\set ECHO all queries</code>	<ul style="list-style-type: none"> ● 如果设置为 <code>all</code>，只显示查询信息。等效于使用 <code>gsql</code> 连接数据库时，指定 <code>-a</code> 参数。 ● 如果设置为 <code>queries</code>，显示命令行和查询信息。等效于使用 <code>gsql</code> 连接数据库时，指定 <code>-e</code> 参数。
ECHO_HIDDEN	<code>\set ECHO_HIDDEN on off noexec</code>	<p>当使用元命令查询数据库信息（例如 <code>\dg</code>）时，此变量的取值决定了查询的行为：</p> <ul style="list-style-type: none"> ● 设置为 <code>on</code>，先显示元命令实际调用的查询语句，然后显示查询结果。等效于使用 <code>gsql</code> 连接数据库时，指定 <code>-E</code> 参数。 ● 设置为 <code>off</code>，则只显示查询结果。 ● 设置为 <code>noexec</code>，则只显示查询信息，不执行查询操作。
ENCODING	<code>\set ENCODING encoding_type</code>	设置当前客户端的字符集编码格式。
FETCH_COUNT	<code>\set FETCH_COUNT variable</code>	<ul style="list-style-type: none"> ● 如果将该变量值设置为大于 0 的整数 <code>n</code>，则执行 <code>SELECT</code> 语句时每次从结果集中取 <code>n</code> 行到缓存并显示到屏幕。 ● 如果不设置此变量，或设置的值小于等于 0，则执行 <code>SELECT</code> 语句时一次性把结果都取到缓存。 <p>说明</p> <p>设置合理的 <code>FETCH_COUNT</code> 变量值，将减少内存使用量。一般来说，设为 100 到 1000 之</p>

		间的值比较合理。
HOST	<code>\set HOST <i>hostname</i></code>	设置已连接的数据库主机名称。
IGNOREEOF	<code>\set IGNOREEOF <i>variable</i></code>	<ul style="list-style-type: none"> ● 若设置此变量为整数数值 <i>n</i>，则在 <code>gsql</code> 中输入的前 (<i>n</i>-1) 次 EOF 字符（通常是 Ctrl+D 组合键）都会被忽略，在第 <i>n</i> 次按 EOF 字符才能退出 <code>gsql</code> 程序。 ● 若设置此变量为非数值，则缺省为 10。 ● 若删除此变量，则向交互的 <code>gsql</code> 会话发送一个 EOF 终止应用。
LASTOID	<code>\set LASTOID <i>oid</i></code>	最后影响的 <code>oid</code> 值，即为从一条 <code>INSERT</code> 或 <code>lo_import</code> 命令返回的值。此变量只保证在下一条 SQL 语句的结果显示之前有效。
ON_ERROR_ROLLBACK	<code>\set ON_ERROR_ROLLBACK on interactive off</code>	<ul style="list-style-type: none"> ● 如果设置为 <code>on</code>，当一个事务块里的语句产生错误的时候，这个错误将被忽略而事务继续。 ● 如果设置为 <code>interactive</code>，这样的错误只是在交互的会话里忽略。 ● 如果设置为 <code>off</code>（缺省），事务块里一个语句生成的错误将会回滚整个事务。<code>on_error_rollback-on</code> 模式是通过在一个事务块的每个命令前隐含地发出一个 <code>SAVEPOINT</code> 的方式工作的，在发生错误的时候回滚到该事务块。
ON_ERROR_STOP	<code>\set ON_ERROR_STOP on off</code>	<ul style="list-style-type: none"> ● 如果设置为 <code>on</code>，则命令执行错误时会立即停止，在交互模式下，<code>gsql</code> 会立即返回已执行命令的结果。 ● 如果设置为 <code>off</code>（缺省），则在命令执行错误时将会跳过错误继续执行。
PORT	<code>\set PORT <i>port</i></code>	设置当前连接数据库的端口号。
USER	<code>\set USER <i>username</i></code>	设置当前用于连接的数据库用户。
VERBOSITY	<code>\set VERBOSITY terse default </code>	该变量用于控制错误报告的冗余行。

Y	verbose	<ul style="list-style-type: none"> ● 如果设置为 <code>terse</code>，仅返回严重且主要的错误文本以及文本位置。一般适合于单行错误信息。 ● 如果设置为 <code>default</code>，则返回严重且主要的错误文本及其位置，还包括详细的错误细节、错误提示。可能会跨越多行。 ● 如果设置为 <code>verbose</code>，则返回所有的错误信息。
---	---------	--

2.1.1.2 SQL 代换

像元命令的参数一样，`gsql` 变量的关键特性之一：把 `gsql` 变量替换成正规的 SQL 语句。此外，`gsql` 还提供为变量更换新的别名或其他标识符等功能。使用 SQL 代换方式替换一个变量的值可在变量前加冒号。例如：

```
postgres=# create table public.area (area_id int,area_name varchar(32));
CREATE TABLE
postgres=# insert into area values
(1,'Europe'),(2,'Americas'),(3,'Asia'),(4,'Middle East and Africa');
INSERT 0 4
postgres=# \set foo 'public.area'
postgres=# select * from :foo;
 area_id |      area_name
-----+-----
      1 | Europe
      2 | Americas
      4 | Middle East and Africa
      3 | Asia
(4 rows)
```

执行以上命令，将会查询 `public.area` 表。

须知

变量的值是逐字复制的，甚至可以包含不对称的引号或反斜杠命令。所以必须保证输入的内容有意义。

2.1.1.3 提示符

通过提示符变量设置 `gsql` 的提示符，这些变量是由字符和特殊的转义字符所组成。

表 2-3 提示符变量

变量	描述	示例
PROMPT1	<p>gsq 请求一个新命令时使用的正常提示符。</p> <p>PROMPT1 的默认值为：<code>%%R%#</code></p>	<p>使用变量 PROMPT1 切换提示符：</p> <p>提示符变为[local]：</p> <pre>postgres=# \set PROMPT1 %M [local:/home/gbase/gbase_db/tmp]</pre> <p>提示符变为 name：</p> <pre>postgres=# \set PROMPT1 name name</pre> <p>提示符变为=：</p> <pre>postgres=# \set PROMPT1 %R =</pre>
PROMPT2	<p>在一个命令输入期待更多输入时（例如，查询没有用一个分号结束或者引号不完整）显示的提示符。</p>	<p>使用变量 PROMPT2 显示提示符：</p> <pre>postgres=# \set PROMPT2 TEST postgres=# select * from public.area TEST; area_id area_name -----+----- 1 Europe 2 Americas 4 Middle East and Africa 3 Asia (4 rows)</pre>
PROMPT3	<p>当执行 COPY 命令，并期望在终端输入数据时（例如，COPY FROM STDIN），显示提示符。</p>	<p>使用变量 PROMPT3 显示 COPY 提示符：</p> <pre>postgres=# \set PROMPT3 '>>>>' postgres=# copy public.area from STDIN; Enter data to be copied followed by a newline. End with a backslash and a period on a line by itself. >>>></pre>

提示符变量的值是按实际字符显示的。但是，当设置提示符的命令中出现“%”时，变量的值根据“%”后的字符，替换为已定义的内容，已定义的提示符请参见表 2-4。

表 2-4 已定义的替换

符号	符号说明
%M	主机的全名（包含域名），若连接是通过 Unix 域套接字进行的，则全名为[local]，若 Unix 域套接字不是编译的缺省位置，就是[local:/dir/name]。
%m	主机名删去第一个点后面的部分。若通过 Unix 域套接字连接，则为 [local]。
%>	主机正在侦听的端口号。
%n	数据库会话的用户名。
%/	当前数据库名称。
%~	类似%/，如果数据库是缺省数据库时输出的是波浪线~。
%#	如果会话用户是数据库系统管理员，使用#，否则用>。
%R	<ul style="list-style-type: none"> 对于 PROMPT1 通常是“=”，如果是单行模式则是“^”，如果会话与数据库断开（如果\connect 失败可能发生）则是“!”。 对于 PROMPT2 该序列被“-”、“*”、单引号、双引号或“\$”（取决于 gsql 是否等待更多的输入：查询没有终止、正在一个 /* ... */ 注释里、正在引号或者美元符扩展里）代替。
%x	事务状态： <ul style="list-style-type: none"> ● 如果不在事务块里，则是一个空字符串。 ● 如果在事务块里，则是“*”。 ● 如果在一个失败的事务块里则是“!”。 ● 如果无法判断事务状态时为“?”（比如没有连接）。
%digits	指定字节值的字符将被替换到该位置。
%:name	gsql 变量“name”的值。
%cgbaseand	command 的输出，类似于使用“^”替换。
%[...%]	提示可以包含终端控制字符，这些字符可以改变颜色、背景、提示文本的风格、终端窗口的标题。例如，

	<pre>postgres=# \set PROMPT1 '%[%033[1;33;40m%]n@%/R [%033[0m%]#'</pre> <p>这个句式的结果是在 VT100 兼容的可显示彩色的终端上的一个宽体(1;) 黑底黄字 (33;40) 。</p>
--	---

环境变量

表 2-5 与 gsql 相关的环境变量

名称	描述
COLUMNS	如果\set columns 为 0, 则由此参数控制 wrapped 格式的宽度。这个宽度用于决定在自动扩展的模式下, 是否要把宽输出模式变成竖线的格式。
PAGER	如果查询结果无法在一页显示, 它们就会被重定向到这个命令。可以用\pset 命令关闭分页器。典型的是用命令 more 或 less 来实现逐页查看。缺省值是平台相关的。 说明 ➤ less 的文本显示, 受系统环境变量 LC_CTYPE 影响。
PSQL_EDITOR	\e 和\ef 命令使用环境变量指定的编辑器。变量是按照列出的先后顺序检查的。在 Unix 系统上默认的编辑工具是 vi。
EDITOR	
VISUAL	
PSQL_EDITOR_LINE NUMBER_ARG	当\e 和\ef 带上一行数字参数使用时, 这个变量指定的命令行参数用于向编辑器传递起始行数。像 Emacs 或 vi 这样的编辑器, 这只是个加号。如果选项和行号之间需要空白, 在变量的值后加一个空格。例如: <pre>PSQL_EDITOR_LINENUMBER_ARG = '+' PSQL_EDITOR_LINENUMBER_ARG=' --line '</pre> Unix 系统默认的是+。
PSQLRC	用户的.gsqlrc 文件的交互位置。

SHELL	使用! <code>!</code> 命令跟 <code>shell</code> 执行的命令是一样的效果。
TMPDIR	存储临时文件的目录。缺省是/tmp。

2.1.2 使用指导

前提条件

连接数据库时使用的用户需要具备访问数据库的权限。

背景信息

使用 `gsql` 命令可以连接数据库。在连接远程数据库时，需要在服务器上设置允许远程连接，详细操作请参见《GBase 8s V8.8.5_5.0.0_数据库管理指南》中“远程连接数据库”章节。

操作步骤

(1) 使用 `gsql` 连接到 GBase 8s 服务器。格式为：

```
gsql -d dbname -p port <-U user_name> <-h hostip>
```

- `-d` 参数指定要连接到的数据库名称。首次连接可以指定生成的默认数据库 `postgres`。
- `-p` 参数指定通过端口号信息。
- `-U` 参数指定以数据库用户名。缺省默认为 `gbase`。
- `-h` 参数指定数据库节点所在的服务器 IP。缺省默认为当前服务器 IP。

说明

若未指定数据库名称，则使用初始化时默认生成的数据库名称；若未指定数据库用户名，则默认使用当前操作系统用户作为数据库用户名；当某个值没有前面的参数（`-d`、`-U` 等）时，若连接 的命令中没有指定数据库名（`-d`）则该参数会被解释成数据库名；如果已经指定数据库名（`-d`） 而没有指定数据库用户名（`-U`）时，该参数则会被解释成数据库用户名。

示例如下：

示例 1：使用 `gbase` 用户连接到本机 `postgres` 数据库的 15400 端口。

```
$ gsql -d postgres -p 15400
```

示例 2：使用 `jack` 用户连接到远程主机 `postgres` 数据库的 15400 端口。

```
$ gsql -h 10.180.123.163 -d postgres -U jack -p 15400
```

示例 3: 参数 postgres 和 gbase 不属于任何选项时, 分别被解释成了数据库名和用户名。

```
$ gsql postgres gbase -p 15400
```

等效于

```
$ gsql -d postgres -U gbase -p 15400
```

详细的 gsql 参数请参见[命令参考](#)。

(2) 执行 SQL 语句。

例如, 创建数据库 human_staff:

```
postgres=# CREATE DATABASE human_staff;
CREATE DATABASE
```

通常, 输入的命令在遇到分号的时候结束。如果输入的命令没有错误, 结果就会输出到屏幕上。

(3) 执行 gsql 元命令。

首先创建一个表空间, 例如 EXAMPLE:

```
postgres=# CREATE TABLESPACE EXAMPLE RELATIVE LOCATION
'tablespace1/tablespace_1';
CREATE TABLESPACE
```

然后创建模式, 例如 HR:

```
postgres=# CREATE schema HR;
CREATE SCHEMA
```

以把一个查询分成多行输入为例。注意提示符的变化:

```
postgres=# CREATE TABLE HR.areaS(
postgres(# area_ID NUMBER, area_NAME VARCHAR2(25)
postgres(# )
postgres-# tablespace EXAMPLE;
CREATE TABLE
```

查看表的定义:

```
postgres=# \d HR.areaS
          Table "hr.areaS"
  Column  |          Type          | Modifiers
-----+-----+-----
 area_id  | numeric                |
```



```
area_name | character varying(25) |
Tablespace: "example"
```

向 HR.areaS 表插入数据:

```
postgres=# INSERT INTO HR.areaS (area_ID, area_NAME) VALUES (1, 'Europe'), (2,
'Americas'), (3, 'Asia'), (4, 'Middle East and Africa');
INSERT 0 4
```

切换提示符:

```
postgres=# \set PROMPT1 '%n@m %~%R%#'
gbase@local postgres=#
```

查看表:

```
gbase@local postgres=# SELECT * FROM HR.areaS;
area_id |          area_name
-----+-----
      1 | Europe
      2 | Americas
      3 | Asia
      4 | Middle East and Africa
(4 rows)
```

可以用 \pset 命令以不同的方法显示表:

```
gbase@local postgres=# \pset border 2
Border style is 2
gbase@local postgres=# SELECT * FROM HR.areaS;
+-----+-----+
| area_id |          area_name          |
+-----+-----+
|      1 | Europe                      |
|      2 | Americas                    |
|      3 | Asia                        |
|      4 | Middle East and Africa     |
+-----+-----+
(4 rows)
```

```
gbase@local postgres=# \pset border 0
Border style is 0.
gbase@local postgres=# SELECT * FROM HR.areaS;
area_id          area_name
-----
```

```
1 Europe
2 Americas
3 Asia
4 Middle East and Africa
(4 rows)
```

使用元命令:

```
gbase@local postgres=# \a \t \x
Output format is unaligned.
Showing only tuples.
Expanded display is on.
gbase@local postgres=# SELECT * FROM HR.areaS;
area_id|1
area_name|Europe

area_id|2
area_name|Americas

area_id|3
area_name|Asia

area_id|4
area_name|Middle East and Africa

gbase@local postgres=#
```

2.1.3 获取帮助

操作步骤

- 连接数据库时，可以使用如下命令获取帮助信息。

```
$ gsql --help
```

显示如下帮助信息:

```
Usage:
gsql [OPTION]... [DBNAME [USERNAME]]

General options:
-c, --command=COMMAND    run only single command (SQL or internal) and exit
-d, --dbname=DBNAME      database name to connect to (default: "gbase")
-f, --file=FILENAME      execute commands from file, then exit
```

```

-l, --list                list available databases, then exit
-v, --set=, --variable=NAME=VALUE
                           set gsql variable NAME to VALUE
-V, --version            output version information, then exit
-X, --no-gsqlrc          do not read startup file (~/.gsqlrc)
-l ("one"), --single-transaction
                           execute command file as a single transaction
-?, --help                show this help, then exit

```

Input and output options:

```

-a, --echo-all           echo all input from script
-e, --echo-queries       echo commands sent to server
-E, --echo-hidden        display queries that internal commands generate
-k, --with-key=KEY       the key for decrypting the encrypted file
-L, --log-file=FILENAME  send session log to file
-m, --maintenance       can connect to cluster during 2-pc transaction

```

recovery

```

-n, --no-libedit         disable enhanced command line editing (libedit)
-o, --output=FILENAME    send query results to file (or |pipe)
-q, --quiet              run quietly (no messages, only query output)
-C, --enable-client-encryption
                           enable client encryption feature
-s, --single-step        single-step mode (confirm each query)
-S, --single-line        single-line mode (end of line terminates SQL command)

```

Output format options:

```

-A, --no-align           unaligned table output mode
-F, --field-separator=STRING
                           set field separator (default: "|")
-H, --html               HTML table output mode
-P, --pset=VAR[=ARG]    set printing option VAR to ARG (see \pset command)
-R, --record-separator=STRING
                           set record separator (default: newline)
-r                        if this parameter is set, use libedit
-t, --tuples-only        print rows only
-T, --table-attr=TEXT    set HTML table tag attributes (e.g., width, border)
-x, --expanded           turn on expanded table output
-z, --field-separator-zero
                           set field separator to zero byte
-0, --record-separator-zero
                           set record separator to zero byte

```

```
-2, --pipeline          use pipeline to pass the password, forbidden to use
in terminal
                        must use with -c or -f
```

Connection options:

```
-h, --host=HOSTNAME    database server host or socket directory (default:
"/home/gbase/gbase_db/tmp")
-p, --port=PORT        database server port (default: "15400")
-U, --username=USERNAME database user name (default: "gbase")
-W, --password=PASSWORD the password of specified database user
```

For more information, type "\?" (for internal commands) or "\help" (for SQL commands) from within gsql, or consult the gsql section in the GaussDB Kernel documentation.

- 连接到数据库后，可以使用如下命令获取帮助信息。

```
postgres=# help
```

显示如下帮助信息：

```
You are using gsql, the command-line interface to gaussdb.
Type: \copyright for distribution terms
      \h for help with SQL commands
      \? for help with gsql commands
      \g or terminate with semicolon to execute query
      \q to quit
```

任务示例

- (1) 使用如下命令连接数据库。

```
gsql -d postgres -p 15400
```

连接成功后，系统显示类似如下信息：

```
gsql ((single_node GBase8sV8.8.5 S5.0.0BXX build b0e57b26) compiled at 2024-01-14
16:38:29 commit 0 last mr 443 )
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

postgres=#
```

- (2) 查看 gsql 的帮助信息。具体执行命令请参见表 2-6。

表 2-6 使用 gsql 联机帮助

描述	示例
查看版权信息	<pre>postgres=# \copyright</pre>
查看 GBase 8s 支持的 SQL 语句的帮助	<p>查看 GBase 8s 支持的 SQL 语句的帮助</p> <p>例如, 查看所有支持的 SQL 语句:</p> <pre>postgres=# \h</pre> <p>例如, 查看 CREATE DATABASE 命令的参数可使用下面的命令:</p> <pre>postgres=# \help CREATE DATABASE</pre>
查看 gsql 命令的帮助	<p>例如, 查看 gsql 支持的命令:</p> <pre>postgres=# \?</pre>

2.1.4 命令参考

详细的 gsql 参数请参见表 2-7、表 2-8、表 2-9 和表 2-10。

表 2-7 常用参数

参数	参数说明	取值范围
-c, --command=COMMAN D	声明 gsql 要执行一条字符串命令然后退出。	--
-d, --dbname=DBNAME	<p>指定想要连接的数据库名称。</p> <p>gsql 还允许使用扩展的 DBNAME, 即 'postgres[ql]://[user[:password]@][netloc][:port][,...][/dbname][?param1=value1&...]' 或 '[key=value] [...]' 形式的连接串作为 DBNAME, gsql 将从连接串中解析连接信息, 并优先使用这些信息。</p>	字符串
-f, --file=FILENAME	使用文件作为命令源而不是交互式输入。gsql 将在处理完文件后结束。如果 FILENAME 是	绝对路径或相对路径,

参数	参数说明	取值范围
	<p>- (连字符)，则从标准输入读取。</p> <p>说明</p> <ul style="list-style-type: none"> ➢ 环境场景：一主一备（8U32G）。 ➢ 实测：读取 200M 数据文件时，耗时约为 8 分钟 21 秒。 ➢ 读取 500M 数据文件时，耗时约为 18 分 41 秒。 <p>建议</p> <ul style="list-style-type: none"> ➢ 文件读取的时间随文件数据量逐渐增加，如果文件太大，中间异常时需要重新读取，同时会导致系统 OS 的 IO 过载，建议文件大小控制在 500M 左右。 	且满足操作系统路径命名规则。
-l, --list	列出所有可用的数据库，然后退出。	---
-v, --set, --variable=NAME=VALUE	设置 gsql 变量 NAME 为 VALUE。	---
-X, --no-gsqlrc	<p>不读取启动文件（系统范围的 gsqlrc 或者用户的 ~/.gsqlrc 都不读取）。</p> <p>说明</p> <ul style="list-style-type: none"> ➢ 启动文件默认为 ~/.gsqlrc，或通过 PSQLRC 环境变量指定。 	---
-l ("one"), --single-transaction	<p>当 gsql 使用 -f 选项执行脚本时，会在脚本的开头和结尾分别加上 START TRANSACTION/CgbaseIT 用以把整个脚本当作一个事务执行。这将保证该脚本完全执行成功，或者脚本无效。</p> <p>说明</p> <ul style="list-style-type: none"> ➢ 如果脚本中已经使用了 START 	---

参数	参数说明	取值范围
	TRANSACTION 、 CgbaseIT 、 ROLLBACK, 则该选项无效。	
-?, --help	显示关于 gsql 命令行参数的帮助信息然后退出。	---
-V, --version	显示 gsql 版本信息然后退出。	---

表 2-8 输入和输出参数

参数	参数说明	取值范围
-a, --echo-all	<p>在读取行时向标准输出打印所有内容。</p> <p>注意</p> <ul style="list-style-type: none"> ➤ 使用此参数可能会暴露部分 SQL 语句中的敏感信息，如创建用户语句中的 password 信息等，请谨慎使用。 	---
-e, --echo-queries	<p>把所有发送给服务器的查询同时回显到标准输出。</p> <p>注意</p> <ul style="list-style-type: none"> ➤ 使用此参数可能会暴露部分 SQL 语句中的敏感信息，如创建用户语句中的 password 信息等，请谨慎使用。 	---
-E, --echo-hidden	回显由\d和其他反斜杠命令生成的实际查询。	---
-k, --with-key=KEY	<p>使用 gsql 对导入的加密文件进行解密。</p> <p>须知</p> <ul style="list-style-type: none"> ➤ 对于本身就是 shell 命令中的关键字符如单引号 (') 或双引号 (")，Linux shell 会检测输入的单引号 (') 或双引号 (") 是否匹配。如果不匹配，shell 认为用户没有输入完毕，会一直等待用户输入，从而不会进入到 gsql 程序。 	---

参数	参数说明	取值范围
	不支持解密导入存储过程和函数。	
-L, --log- file=FILE NAME	<p>除了正常的输出源之外，把所有查询输出记录到文件 FILENAME 中。</p> <p>注意</p> <ul style="list-style-type: none"> ➤ 使用此参数可能会暴露部分 SQL 语句中的敏感信息，如创建用户语句中的 password 信息等，请谨慎使用。 <p>此参数只保留查询结果到相应文件中，主要目标是为了查询结果能够更好更准确地被其他调用者（例如自动化运维脚本）解析；而不是保留 gsql 运行过程中的相关日志信息。</p>	绝对路径或相对路径，且满足操作系统路径命名规则。
-m, --maintenance	<p>允许在两阶段事务恢复期间连接 GBase 8s。</p> <p>说明</p> <ul style="list-style-type: none"> ➤ 该选项是一个开发选项，禁止用户使用，仅限专业技术人员使用，功能是：使用该选项时，gsql 可以连接到备机，用于校验主备机数据的一致性。 	--
-n, --no-libedit	关闭命令行编辑。	--
-o, --output= FILENAME	将所有查询输出重定向到文件 FILENAME。	绝对路径或相对路径，且满足操作系统路径命名规则。
-q, --quiet	安静模式，执行时不会打印出额外信息。	缺省时 gsql 将打印许多其他输出信息。

参数	参数说明	取值范围
-s, --single-step	<p>单步模式运行。意味着每个查询在发往服务器之前都要提示用户，用这个选项也可以取消执行。此选项主要用于调试脚本。</p> <p>注意</p> <ul style="list-style-type: none"> ➤ 使用此参数可能会暴露部分 SQL 语句中的敏感信息，如创建用户语句中的 password 信息等，请谨慎使用。 	---
-S, --single-line	单行运行模式，这时每个命令都将由换行符结束，像分号那样。	---
-C, --enable-client-encryption	当使用-C 参数连接本地数据库或者连接远程数据库时，可通过该选项打开密态数据库开关。	---

表 2-9 输出格式参数

参数	参数说明	取值范围
-A, --no-align	切换为非对齐输出模式。	缺省为对齐输出模式
-F, --field-separator=ST RING	设置域分隔符（默认为“ ”）。	---
-H, --html	打开 HTML 格式输出。	---
-P, --pset=VAR[= ARG]	<p>在命令行上以 pset 的风格设置打印选项。</p> <p>说明</p> <ul style="list-style-type: none"> ➤ 这里必须用等号而不是空格分隔名称和值。例如，把输出格式设置为 LaTeX，可以键入 -P format=latex 	---
-R, --record-	设置记录分隔符。	---

参数	参数说明	取值范围
separator=STRING		
-r	开启在客户端操作中可以编辑的模式。	缺省为关闭。
-t, --tuples- only	只打印行。	---
-T, --table-attr=TEXT	允许声明放在 HTML table 标签里的选项。 使用时请搭配参数“-H,--html”，指定为 HTML 格式输出。	---
-x, --expanded	打开扩展表格式模式。	---
-z, --field-separator-zero	设置非对齐输出模式的域分隔符为空。 使用时请搭配参数“-A, --no-align”，指定为非对齐输出模式。	---
-0, --record-separator-zero	设置非对齐输出模式的记录分隔符为空。 使用时请搭配参数“-A, --no-align”，指定为非对齐输出模式。	---
-2, --pipeline	使用管道传输密码，禁止在终端使用，必须和-c 或者-f 参数一起使用。	---
-g,	打印来自文件的所有 SQL。	---

表 2-10 连接参数

参数	参数说明	取值范围
-h, --host=HOSTNAME	指定正在运行服务器的主机名或者 Unix 域套接字的路径。	如果省略主机名，gsqI 将通过 Unix 域套接字与本地主机

参数	参数说明	取值范围
		的服务器相联，或者在没有 Unix 域套接字的机器上，通过 TCP/IP 与 localhost 连接。
-p, --port=PORT	指定数据库服务器的端口号。 可以通过 port 参数修改默认端口号。	0 到 65535
-U, --username=USER NAME	指定连接数据库的用户。 说明 <ul style="list-style-type: none"> 通过该参数指定用户连接数据库时，需要同时提供用户密码用以身份验证。您可以通过交换方式输入密码，或者通过 -W 参数指定密码。 用户名中包含有字符 \$，需要在字符 \$ 前增加转义字符才可成功连接数据库。 	字符串。默认使用与当前操作系统用户同名的用户。
-W, --password=PASS WORD	当使用 -U 参数连接远端数据库时，可通过该选项指定密码。 说明 <ul style="list-style-type: none"> 数据库主节点所在服务器后连接本地数据库主 节点实例时，默认使用 trust 连接，会忽略此参数。 用户密码中包含特殊字符“\”和“'”时，需要增加转义字符才可成功连接数据库。 如果用户未输入该参数，但是数据库连接需要用户密码，这时将出现交互式输入，请用户输入当前连接的密码。该密码最长长度为 999 字节，受限于 GUC 参数 password_max_length 	符合密码复杂度要求。

参数	参数说明	取值范围
	的最大值。	

2.1.5 元命令参考

使用 GBase 8s 数据库命令行交互工具登录数据库后，`gsql` 所提供的元命令。所谓元命令就是在 `gsql` 里输入的任何以不带引号的反斜杠开头的命令。

注意事项

- 一个 `gsql` 元命令的格式是反斜杠后面紧跟一个动词，然后是任意参数。参数命令动词和其他参数以任意个空白字符间隔。
- 要在参数里面包含空白，必须用单引号把它引起来。要在这样的参数里包含单引号，可以在前面加一个反斜杠。任何包含在单引号里的内容都会被进一步进行类似 C 语言的替换：`\n`（新行）、`\t`（制表符）、`\b`（退格）、`\r`（回车）、`\f`（换页）、`\digits`（八进制表示的字符）、`\xdigits`（十六进制表示的字符）。
- ""包围的内容被当做一个命令行传入 `shell`。该命令的输出（删除了结尾的新行）被当做参数值。
- 如果不带引号的参数以冒号（`:`）开头，它会被当做一个 `gsql` 变量，并且该变量的值最终会成为真正的参数值。
- 有些命令以一个 SQL 标识的名称（比如一个表）为参数。这些参数遵循 SQL 语法关于双引号的规则：不带双引号的标识强制转换成小写，而双引号保护字母不进行大小写转换，并且允许在标识符中使用空白。在双引号中，成对的双引号在结果名称中分析成一个双引号。比如，`FOO"BAR"BAZ` 解析成 `fooBARbaz`；而 `"Aweird""name"` 解析成 `Aweird"name`。
- 对参数的分析在遇到另一个不带引号的反斜杠时停止。这里会认为是一个新的元命令的开始。特殊的双反斜杠序列（`\\`）标识参数的结尾并将继续分析后面的 SQL 语句（如果存在）。这样 SQL 和 `gsql` 命令可以自由的在一行里面混合。但是在任何情况下，一条元命令的参数不能延续超过行尾。

元命令

元命令的详细说明请参见表 2-11 至表 2-19。

表 2-11 一般的元命令

参数	参数说明	取值范围
\copyright	显示 GBase 8s 的版本和版权信息。	---
\g [FILE] or [PIPE];	执行查询（并将结果发送到文件或管道）。	---
\h(\help) [NAME]	给出指定 SQL 语句的语法帮助。	如果没有给出 NAME, gsql 将列出可获得帮助的所有命令。如果 NAME 是一个星号 (*), 则显示所有 SQL 语句的语法帮助。
\parallel [on [num]]off	<p>控制并发执行开关。</p> <ul style="list-style-type: none"> ● on: 打开控制并发执行开关, 且最大并发数为 num。 ● off: 关闭控制并发执行开关。 <p>说明:</p> <ul style="list-style-type: none"> ➢ 不支持事务中开启并发执行以及并发中开启事务。 ➢ 不支持\d 这类元命令的并发。 ➢ 并发 select 返回结果混乱问题, 此为客户可接受, core、进程停止响应不可接受。 ➢ 不推荐在并发中使用 set 语句, 否则导致结果与预期不一致。 ➢ 不支持创建临时表。如需使用临时表, 需要在开启 parallel 之前创建好, 并在 parallel 内部使用。parallel 内部不允许创建临时表。 ➢ \parallel 执行时最多会启动 num 个独 	<p>num 的默认值: 1024。</p> <p>须知</p> <ul style="list-style-type: none"> ➢ 服务器能接受的最大连接数受 max_connection 及当前已有连接数限制。 ➢ 设置 num 时请考虑服务器当前可接受的实际连接数合理指定。

参数	参数说明	取值范围
	<p>立的 gsql 进程连接服务器。</p> <ul style="list-style-type: none"> ➤ \parallel 中所有作业的持续时间不能超过 session_timeout，否则可能会导致并发执行过程中断连。 ➤ 在\parallel on 之后一条或多条命令，会等到\parallel off 执行后才会执行，因而，每个\parallel on 需对应一个\parallel off，否则会导致\parallel on 之后的命令无法执行。 	
\q	退出 gsql 程序。在一个脚本文件里，只在脚本终止的时候执行。	---

表 2-12 查询缓存区元命令

参数	参数说明
\e [FILE] [LINE]	使用外部编辑器编辑查询缓冲区（或者文件）。
\ef [FUNCNAME [LINE]]	使用外部编辑器编辑函数定义。如果指定了 LINE（即行号），则光标会指到函数体的指定行。
\p	打印当前查询缓冲区到标准输出。
\r	重置（或清空）查询缓冲区。
\w FILE	将当前查询缓冲区输出到文件。

表 2-13 输入/输出元命令

参数	参数说明
\copy { table [(column_list)] (query) } { from to } { filename stdin stdout pstdin pstdout } [with] [binary] [oids] [delimiter [as]	在任何 gsql 客户端登录数据库成功后，可以执行导入导出数据。这是一个运行 SQL COPY 命令的操作，但不是读取或写入指定文件的服务器，而是读取或写入文件，并在服务器和本地

<pre>'character'] [null [as] 'string'] [csv [header] [quote [as] 'character'] [escape [as] 'character'] [force quote column_list *] [force not null column_list]][parallel integer]</pre>	<p>文件系统之间路由数据。这意味着文件的可访问性和权限是本地用户的权限，而不是服务器的权限，并且不需要数据库初始化用户权限。</p> <p>说明</p> <ul style="list-style-type: none"> ➢ \COPY 只适合小批量，格式良好的数据导入，不会对非法字符进行预处理，也无容错能力。导入数据应优先选择 COPY。 ➢ \COPY 可以指定数据导入时的客户端数量，从而实现数据文件的并行导入，目前并发数范围为[1, 8]。 ➢ \COPY 并行导入目前存在以下约束：临时表的并行导入不支持、在事务内的并行导入不支持、对二进制文件的并行导入不支持、数据导入支持 AES128 加密时不支持。在这些情况下，即使指定了 parallel 参数，仍然会走非并行流程。
<pre>\echo [STRING]</pre>	<p>把字符串写到标准输出。</p>
<pre>\i FILE</pre>	<p>从文件 FILE 中读取内容，并将其当作输入，执行查询。</p>
<pre>\i+ FILE KEY</pre>	<p>执行加密文件中的命令。</p>
<pre>\ir FILE</pre>	<p>和i 类似，只是相对于存放当前脚本的路径。</p>
<pre>\ir+ FILE KEY</pre>	<p>和i+类似，只是相对于存放当前脚本的路径。</p>
<pre>\o [FILE]</pre>	<p>把所有的查询结果发送到文件里。</p>
<pre>\qecho [STRING]</pre>	<p>把字符串写到查询结果输出流里</p>

表 2-14 显示信息元命令

参数	参数说明	取值范围	示例
\d[S+]	列出当前 search_path 中模式	---	列出当前 search_path 中模式

参数	参数说明	取值范围	示例
	下所有的表、视图和序列。当 search_path 中不同模式存在同名对象时，只显示 search_path 中位置靠前模式下的同名对象。		下所有的表、视图和序列。 <pre>postgres=# \d</pre>
<code>\d[S+] NAME</code>	列出指定表、视图和索引的结构。	---	假设存在表 a，列出指定表 a 的结构。 <pre>postgres=# \dtable+ a</pre>
<code>\d+ [PATTERN]</code>	列出所有表、视图和索引。	如果声明了 PATTERN, 只显示名称匹配 PATTERN 的表、视图和索引。	列出所有名称以 f 开头的表、视图和索引。 <pre>postgres=# \d+f*</pre>
<code>\da[S] [PATTERN]</code>	列出所有可用的聚集函数以及它们操作的数据类型和返回值类型。	如果声明了 PATTERN, 只显示名称匹配 PATTERN 的聚集函数。	列出所有名称以 f 开头可用的聚集函数以及它们操作的数据类型和返回值类型。 <pre>postgres=# \da f*</pre>
<code>\db[+] [PATTERN]</code>	列出所有可用的表空间。	如果声明了 PATTERN, 只显示名称匹配 PATTERN 的表空间。	列出所有名称以 p 开头的可用表空间。 <pre>postgres=# \db p*</pre>
<code>\dc[S+]</code>	列出所有字符集之	如果声明了	列出所有字符集

参数	参数说明	取值范围	示例
[PATTERN]	间的可用转换。	PATTERN, 只显示名称匹配 PATTERN 的转换。	之间的可用转换。 <code>postgres=# \dc *</code>
\dC[+] [PATTERN]	列出所有类型转换。	如果声明了 PATTERN, 只显示名称匹配 PATTERN 的转换。	列出所有名称以 c 开头的类型转换。 <code>postgres=# \dC c*</code>
\dd[S] [PATTERN]	显示所有匹配 PATTERN 的描述。	如果没有给出参数, 则显示所有可视对象。“对象”包括: 聚集、函数、操作符、类型、关系 (表、视图、索引、序列、大对象)、规则。	列出所有可视对象。 <code>postgres=# \dd</code>
\ddp [PATTERN]	显示所有默认的使用权限。	如果指定了 PATTERN, 只显示名称匹配 PATTERN 的使用权限。	列出所有默认的使用权限。 <code>postgres=# \ddp</code>
\dD[S+] [PATTERN]	列出所有可用域。	如果声明了 PATTERN, 只显示名称匹配 PATTERN 的域。	列出所有可用域。 <code>postgres=# \dD</code>
\ded[+] [PATTERN]	列出所有的 Data Source 对象。	如果声明了 PATTERN, 只显示名称匹配 PATTERN 的对象。	列出所有的 Data Source 对象。 <code>postgres=# \ded</code>
\det[+] [PATTERN]	列出所有的外部表。	如果声明了 PATTERN, 只显示名称匹配 PATTERN 的表。	列出所有的外部表。 <code>postgres=# \det</code>

参数	参数说明	取值范围	示例
\des[+] [PATTERN]	列出所有的外部服务器。	如果声明了 PATTERN, 只显示名称匹配 PATTERN 的服务器。	列出所有的外部服务器。 <code>postgres=# \des</code>
\deu[+] [PATTERN]	列出用户映射信息。	如果声明了 PATTERN, 只显示名称匹配 PATTERN 的信息。	列出用户映射信息。 <code>postgres=# \deu</code>
\deu[+] [PATTERN]	列出用户映射信息。	如果声明了 PATTERN, 只显示名称匹配 PATTERN 的信息。	列出用户映射信息。 <code>postgres=# \deu</code>
\dew[+] [PATTERN]	列出封装的外部数据。	如果声明了 PATTERN, 只显示名称匹配 PATTERN 的数据。	列出封装的外部数据。 <code>postgres=# \dew</code>
\df[antw][S+] [PATTERN]	列出所有可用函数以及它们的参数和返回的数据类型。 <ul style="list-style-type: none"> ● a 代表聚集函数; ● n 代表普通函数; ● t 代表触发器; ● w 代表窗口函数。 	如果声明了 PATTERN, 只显示名称匹配 PATTERN 的函数。	列出所有可用函数以及它们的参数和返回的数据类型。 <code>postgres=# \df</code>
\dF[+] [PATTERN]	列出所有的文本搜索配置信息。	如果声明了 PATTERN, 只显示名称匹配 PATTERN 的配置	列出所有的文本搜索配置信息。 <code>postgres=# \dF+</code>

参数	参数说明	取值范围	示例
		信息。	
<code>\dFd[+]</code> <code>[PATTERN]</code>	列出所有的文本搜索字典。	如果声明了 PATTERN, 只显示名称匹配 PATTERN 的字典。	列出所有的文本搜索字典。 <pre>postgres=# \dFd</pre>
<code>\dFp[+]</code> <code>[PATTERN]</code>	列出所有的文本搜索分析器。	如果声明了 PATTERN, 只显示名称匹配 PATTERN 的分析器。	列出所有的文本搜索分析器。 <pre>postgres=# \dFp</pre>
<code>\dFt[+]</code> <code>[PATTERN]</code>	列出所有的文本搜索模板。	如果声明了 PATTERN, 只显示名称匹配 PATTERN 的模板。	列出所有的文本搜索模板。 <pre>postgres=# \dFt</pre>
<code>\dg[+]</code> <code>[PATTERN]</code>	列出所有数据库角色。 说明: 因为用户和群组的概念被统一为角色, 所以这个命令等价于 <code>\du</code> 。为了和以前兼容, 所以保留两个命令。	如果指定了 PATTERN, 只显示名称匹配 PATTERN 的角色。	列出名称为 'j_e' 所有数据库角色。 <pre>postgres=# \dg j?e</pre>
<code>\dl</code>	<code>\lo_list</code> 的别名, 显示一个大对象的列表。	---	列出所有的大对象。 <pre>postgres=# \dl</pre>
<code>\dL[S+]</code> <code>[PATTERN]</code>	列出可用的程序语言。	如果指定了 PATTERN, 只列出名称	列出可用的程序语言。

参数	参数说明	取值范围	示例
		匹配 PATTERN 的语言。	<code>postgres=# \dL</code>
<code>\dn[S+] [PATTERN]</code>	列出物化视图。	如果指定了 PATTERN, 只列出名称匹配 PATTERN 的物化视图。	列出物化视图。 <code>postgres=# \dn</code>
<code>\dn[S+] [PATTERN]</code>	列出所有的模式 (名称空间)。	如果声明了 PATTERN, 只列出名称匹配 PATTERN 的模式名。缺省时, 只列出用户创建的模式。	列出所有名称以 d 开头的模式以及相关信 息。 <code>postgres=# \dn+d*</code>
<code>\do[S] [PATTERN]</code>	列出所有可用的操作符以及它们的操作数和返回的数据类型。	如果声明了 PATTERN, 只列出名称匹配 PATTERN 的操作符。缺省时, 只列出用户创建的操作符。	列出所有可用的操作符以及它们的操作数和返回的数据类型。 <code>postgres=# \do</code>
<code>\dO[S+] [PATTERN]</code>	列出排序规则。	如果声明了 PATTERN, 只列出名称匹配 PATTERN 的规则。缺省时, 只列出用户创建的规则。	列出排序规则。 <code>postgres=# \dO</code>
<code>dp [PATTERN]</code>	列出一列可用的表、视图以及相关的权限信息。 <code>\dp</code> 显示结果如下: <ul style="list-style-type: none">● 赋予一个角色的权限 <code>rolename=xxxx/yyy</code>	如果指定了 PATTERN, 只列出名称匹配 PATTERN 的表、视图。	列出一列可用的表、视图以及相关的权限信息。 <code>postgres=# \dp</code>

参数	参数说明	取值范围	示例
	<p>y</p> <ul style="list-style-type: none"> ● 赋予 public 的权限 <p>=xxxx/yyyy</p> <p>xxxx 表示赋予的权限, yyyy 表示授予这个权限的角色。权限的参数说明请参见表 2-15。</p>		
<p>\drds [PATTERN1 [PATTERN2]]</p>	<p>列出所有修改过的配置参数。这些设置可以是针对角色的、针对数据库的或者同时针对两者的。PATTERN1 和 PATTERN2 表示要列出的角色 PATTERN 和数据库 PATTERN。</p>	<p>如果声明了 PATTERN, 只列出名称匹配 PATTERN 的规则。缺省或指定*时, 则会列出所有设置。</p>	<p>列出 postgres 数据库所有修改过的配置参数。</p> <pre>postgres=# \drds * postgres</pre>
<p>\dT[S+] [PATTERN]</p>	<p>列出所有的数据类型。</p>	<p>如果指定了 PATTERN, 只列出名称匹配 PATTERN 的类型。</p>	<p>列出所有的数据类型。</p> <pre>postgres=# \dT</pre>
<p>\du[+] [PATTERN]</p>	<p>列出所有数据库角色。</p> <p>说明:</p> <p>因为用户和群组的概念被统一为角色, 所以这个命令等价</p>	<p>如果指定了 PATTERN, 则只列出名称匹配 PATTERN 的角色。</p>	<p>列出所有数据库角色。</p> <pre>postgres=# \du</pre>

参数	参数说明	取值范围	示例
	于\dg。为了和以前兼容，所以保留两个命令。		
\dE[S+] [PATTERN] \di[S+] [PATTERN] \ds[S+] [PATTERN] \dt[S+] [PATTERN] \dv[S+] [PATTERN]	这一组命令，字母E、i、s、t和v分别代表着外部表、索引、序列、表和视图。可以以任意顺序指定其中一个或者它们的组合来列出这些对象。例如： <code>\dit</code> 列出所有的索引和表。在命令名称后面追加+，则每一个对象的物理尺寸以及相关的描述也会被列出。	如果指定了PATTERN，只列出名称匹配该PATTERN的对象。默认情况下只会显示用户创建的对象。通过PATTERN或者S修饰符可以把系统对象包括在内。	列出所有的索引和视图。 <pre>postgres=# \div</pre>
\dx[+] [PATTERN]	列出安装数据库的扩展信息。	如果指定了PATTERN，则只列出名称匹配PATTERN的扩展信息。	列出安装数据库的扩展信息。 <pre>postgres=# \dx</pre>
\l[+]	列出服务器上所有数据库的名称、所有者、字符集编码以及使用权限。	---	列出服务器上所有数据库的名称、所有者、字符集编码以及使用权限。 <pre>postgres=# \l</pre>
\sf[+] FUNCNAME	显示函数的定义。 说明： 对于带圆括号的函	---	假设存在函数 <code>function_a</code> 和函数名带圆括号的函数 <code>func()name</code> ，列

参数	参数说明	取值范围	示例
	数名, 需要在函数名两端添加双引号, 并且在双引号后面加上参数类型列表。参数类型列表两端添加圆括号。		出函数的定义。 <pre>postgres=# \sf function_a postgres=# \sf "func()name" (argtype1, argtype2)</pre>
\z [PATTERN]	列出数据库中所有表、视图和序列以及它们相关的访问特权。	如果给出任何 pattern, 则被当成一个正则表达式, 只显示匹配的表、视图、序列。	列出数据库中所有表、视图和序列以及它们相关的访问特权。 <pre>postgres=# \z</pre>

 说明

- 表 2-14 中的选项 S 表示显示系统对象, PATTERN 表示显示对象附加的描述信息。用来指定要被显示的对象名称。

表 2-15 权限的参数说明

参数	参数说明
r	SELECT: 允许对指定的表、视图读取数据。
w	UPDATE: 允许对指定表更新字段。
a	INSERT: 允许对指定表插入数据。
d	DELETE: 允许删除指定表中的数据。
D	TRUNCATE: 允许清理指定表中的数据。
x	REFERENCES: 允许创建外键约束。由于当前不支持外键, 所以该参数暂不生效。
t	TRIGGER: 允许在指定表上创建触发器。

参数	参数说明
X	EXECUTE：允许使用指定的函数以及利用这些函数实现的操作符。
U	USAGE： <ul style="list-style-type: none"> ● 对于过程语言，允许用户在创建函数时，指定过程语言。 ● 对于模式，允许访问包含在指定模式中的对象。 ● 对于序列，允许使用 nextval 函数。
C	CREATE： <ul style="list-style-type: none"> ● 对于数据库，允许在该数据库里创建新的模式。 ● 对于模式，允许在该模式中创建新的对象。 ● 对于表空间，允许在其中创建表以及允许创建数据库和模式的时候把该表空间指定为其缺省表空间。
c	CONNECT：允许用户连接到指定的数据库。
T	TEMPORARY：允许创建临时表。
A	ALTER：允许用户修改指定对象的属性。
P	DROP：允许用户删除指定的对象。
m	CgbaseENT：允许用户定义或修改指定对象的注释。
i	INDEX：允许用户在指定表上创建索引。
v	VACUUM：允许用户对指定的表执行 ANALYZE 和 VACUUM 操作。
*	给前面权限的授权选项。

表 2-16 格式化元命令

参数	参数说明
----	------

\a	对齐模式和非对齐模式之间的切换。
\C [STRING]	把正在打印的表的标题设置为一个查询的结果或者取消这样的设置。
\f [STRING]	对于不对齐的查询输出，显示或者设置域分隔符。
\H	<ul style="list-style-type: none"> ● 若当前模式为文本格式，则切换为 HTML 输出格式。 ● 若当前模式为 HTML 格式，则切换回文本格式。
\pset NAME [VALUE]	设置影响查询结果表输出的选项。NAME 的取值见表 2-17。
\t [on off]	切换输出的字段名的信息和行计数脚注。
\T [STRING]	指定在使用 HTML 输出格式时放在 table 标签里的属性。如果参数为空，不设置。
\x [on off auto]	切换扩展行格式。

表 2-17 可调节的打印选项

选项	选项说明	取值范围
border	value 为数值类型。通常这个数字越大，表的边界就越宽线就越多，但是这个取决于特定的格式。	<p>在 HTML 格式下，取值范围为大于 0 的整数。</p> <p>在其他格式下，取值范围：</p> <ul style="list-style-type: none"> ● 0: 无边框 ● 1: 内部分隔线 ● 2: 台架
expanded (或 x)	在正常和扩展格式之间切换。	<p>当打开扩展格式时，查询结果用两列显示，字段名称在左、数据在右。这个模式在数据无法放进通常的“水平”模式的屏幕时很有用。</p> <p>在正常格式下，当查询输出的格式比屏幕宽时，用扩展格式。正常格</p>

选项	选项说明	取值范围
		式只对 aligned 和 wrapped 格式有用。
fieldsep	声明域分隔符来实现非对齐输出。这样就可以创建其他程序希望的制表符或逗号分隔的输出。要设置制表符域分隔符，键入 \pset fieldsep '\t'。缺省域分隔符是 ' ' (竖条符)。	---
fieldsep_zero	声明域分隔符来实现非对齐输出到零字节。	---
footer	用来切换脚注。	---
format	设置输出格式。允许使用唯一缩写（如 u 指代 unaligned）。	取值范围： <ul style="list-style-type: none"> ● unaligned：写一行的所有列在一条直线上中，当前活动字段分隔符分隔。 ● aligned：此格式是标准的，可读性最好的文本输出。 ● wrapped：类似 aligned，但是包装跨行的宽数据值，使其适应目标字段的宽度输出。 ● html：把表输出为可用于文档里的对应标记语言。输出不是完整的文档。 ● latex：把表输出为可用于文档里的对应标记语言。输出不是完整的文档。 ● troff-ms：把表输出为可用于文档里的对应标记语言。输出不是完整的文档。
null	打印一个字符串，用来代替一	缺省情况下不打印。

选项	选项说明	取值范围
	个 null 值。	
numericlocale	切换分隔小数点左边的数值的区域相关的分组符号。	<ul style="list-style-type: none"> ● on: 显示指定的分隔符。 ● off: 不显示分隔符。 忽略此参数，显示默认的分隔符。
pager	控制查询和 gsql 帮助输出的分页器。如果设置了环境变量 PAGER，输出将被指向到指定程序，否则使用系统缺省。	<ul style="list-style-type: none"> ● on: 当输出到终端且不适合屏幕显示时，使用分页器。 ● off: 不使用分页器。 ● always: 当输出到终端无论是否符合屏幕显示时，都使用分页器。
recordsep	声明在非对齐输出格式时的记录分隔符。	---
recordsep_zero	声明在非对齐输出到零字节时的记录分隔符。	---
tableattr (或 T)	声明放在 html 输出格式中 HTML table 标签的属性 (例如: cellpadding 或 bgcolor)。注意: 这里可能不需要声明 border, 因为已经在 \pset border 里用过了。如果没有给出 value, 则不设置表的属性。	---
title	为随后打印的表设置标题。这个可以用于给输出一个描述性标签。如果没有给出 value, 不设置标题。	---
tuples_only (或 t)	在完全显示和只显示实际的表数据之间切换。完全显示将输	---

选项	选项说明	取值范围
	出像列头、标题、各种脚注等信息。在 <code>tuples_only</code> 模式下，只显示实际的表数据。	
<code>feedback</code>	切换是否输出结果行数。	---

表 2-18 连接元命令

参数	参数说明	取值范围
<code>\connect</code> [DBNAME]- USER - HOST - PORT -]	<p>连接到一个新的数据库（默认数据库为 <code>postgres</code>）。当数据库名称长度超过 63 个字节时，默认前 63 个字节有效，连接到前 63 个字节对应的数据库，但是 <code>gsq</code>l 的命令提示符中显示的数据库对象名仍为截断前的名称。</p> <p>说明：</p> <ul style="list-style-type: none"> ➤ 重新建立连接时，如果切换数据库登录用户，将可能会出现交互式输入，要求输入新用户的连接密码。该密码最长长度为 999 字节，受限于 <code>GUC</code> 参数 <code>password_max_length</code> 的最大值。 	---
<code>\encoding</code> [ENCODING]	设置客户端字符编码格式。	不带参数时，显示当前的编码格式。
<code>\conninfo</code>	输出当前连接的数据库的信息。	---

表 2-19 操作系统元命令

参数	参数说明	取值范围
<code>\cd</code> [DIR]	切换当前的工作目录。	绝对路径或相对路径，且满足操作系统路径命名规则。

<code>\setenv NAME [VALUE]</code>	设置环境变量 NAME 为 VALUE，如果没有给出 VALUE 值，则不设置环境变量。	---
<code>\timing [on off]</code>	以毫秒为单位显示每条 SQL 语句的执行时间。	<ul style="list-style-type: none"> ● on 表示打开显示。 ● off 表示关闭显示。
<code>!\ [COMMAND]</code>	返回到一个单独的 Unix shell 或者执行 Unix 命令 COMMAND。	---

表 2-20 变量元命令

参数	参数说明
<code>\prompt [TEXT] NAME</code>	提示用户用文本格式来指定变量名称。
<code>\set [NAME [VALUE]]</code>	<p>设置内部变量 NAME 为 VALUE。如果给出了多于一个值，设置为所有这些值的连接结果。如果没有给出第二个参数，只设变量不设值。</p> <p>有一些常用变量被 gsql 特殊对待，它们是一些选项设置，通常所有特殊对待的变量都是由大写字母组成(可能还有数字和下划线)。表 2-21 是一个所有特殊对待的变量列表。</p>
<code>\unset NAME</code>	不设置（或删除）gsql 变量名。

表 2-21 \set 常用命令

名称	命令说明	取值范围
<code>\set VERBOSITY value</code>	该变量控制错误报告的冗余行。	value 可选值 <ul style="list-style-type: none"> ● default ● verbose

		<ul style="list-style-type: none"> ● terse
<p>\set ON_ERROR_ST OP value</p>	<p>如果设置该变量，脚本处理将马上停止。</p> <p>如果该脚本是从另外一个脚本调用的，那个脚本也会按同样的方式停止。</p> <p>如果最外层的脚本不是从一次交互的 gsql 会话中调用的，而是用 -f 选项调用的，gsql 将返回错误代码 3，以示这个情况与致命错误条件的区别（错误代码为 1）。</p>	<p>value 可选值</p> <ul style="list-style-type: none"> ● on/off ● true/false ● yes/no ● 1/0
<p>\set RETRY [retry_times]</p>	<p>用于控制是否开启语句出错场景下的重试功能，参数 retry_times 用来指定最大重试次数，缺省值为 5，取值范围为 5-10。当重试功能已经开启时，再次执行 \set RETRY 可以关闭该功能。</p> <p>使用配置文件 retry_errcodes.conf 列举需要重试的错误码列表，该文件和 gsql 可执行程序位于同一级目录下。该配置文件为系统配置，非用户定义，不允许用户直接修改。</p> <p>当前支持以下出错场景的重试：</p> <ul style="list-style-type: none"> ● YY001: TCP 通信错误, Connection reset by peer ● YY002: TCP 通信错误, Connection reset by peer ● YY003 : 锁 超 时 , Lock wait timeout.../wait transaction xxx sync time exceed xxx ● YY004 : TCP 通信错误, Connection timed out ● YY005: SET 命令发送失败, ERROR SET query ● YY006 : 内存申请失败, memory is 	<p>retry_times 取值范围为：5-10</p>

	<p>temporarily unavailable</p> <ul style="list-style-type: none"> ● YY007: 通信库错误, Memory allocate error ● YY008: 通信库错误, No data in buffer ● YY009: 通信库错误, Close because release memory ● YY010: 通信库错误, TCP disconnect ● YY011: 通信库错误, SCTP disconnect ● YY012: 通信库错误, Stream closed by remote ● YY013: 通信库错误, Wait poll unknown error ● YY014,YY015,53200,08006,08000,57P01,XX003,XX009 等 <p>同时, 出错时 gsql 会查询数据库节点的连接状态, 当状态异常时会 sleep 1 分钟再进行重试, 能够覆盖大部分主备切换场景下的出错重试。</p> <p>说明:</p> <ul style="list-style-type: none"> ➢ 不支持事务块中的语句错误重试; ➢ 不支持通过 ODBC、JDBC 接口查询的出错重试; ➢ 含有 unlogged 表的 sql 语句, 不支持节点故障后的出错重试; ➢ gsql 客户端本身出现的错误, 不在重跑考虑范围之内; 	
--	--	--

表 2-22 大对象元命令

参数	参数说明
\lo_list	显示一个目前存储在该数据库里的所有数据库大对象和注释。

● **PATTERN**

很多情况下，`\d` 命令指定 `PATTERN` 参数，来指定要被显示的对象名称。最简单使用方法，将 `PATTERN` 参数设置为该对象的准确名称。在 `PATTERN` 中的字符通常会被变成小写形式，与 SQL 名称一样，例如 `\dt FOO` 将会显示名为 `foo` 的表。将 `PATTERN` 放在双引号中可以保持其大小写形式。如果需要在 `PATTERN` 中包括一个真正的双引号字符，则需要把它写成两个相邻的双引号，这同样是符合 SQL 引用标识符的规则。例如，`\dt "FOO""BAR"` 将显示名为 `FOO"BAR` (不是 `foo"bar`) 的表。和普通的 SQL 名称规则不同，不能只在 `PATTERN` 的一部分周围放上双引号，例如 `\dt FOO"FOO"BAR` 将会显示名为 `fooFOObar` 的表。

不使用 `PATTERN` 参数时，`\d` 命令会显示当前 `schema` 搜索路径中可见的全部对象——这等价于用 `*` 作为 `PATTERN`。所谓对象可见是指可以直接用名称引用该对象，而不需要用 `schema` 来进行限定。要查看数据库中所有的对象而不管它们的可见性，可以把 `*.*` 用作 `PATTERN`。

如果放在一个 `PATTERN` 中，`*` 将匹配任意字符序列（包括空序列），而 `?` 会匹配任意的单个字符（这种记号方法就像 Unix shell 的文件名 `PATTERN` 一样）。例如，`\dt int*` 会显示名称以 `int` 开始的表。但是如果被放在双引号内，`*` 和 `?` 就会失去这些特殊含义而变成普通的字符。

包含一个点号 (`.`) 的 `PATTERN` 被解释为一个 `schema` 名称模式后面跟上一个对象名称模式。例如，`\dt foo*.*bar*` 会显示名称以 `foo` 开始的 `schema` 中所有名称包括 `bar` 的表。如果没有出现点号，那么模式将只匹配当前 `schema` 搜索路径中可见的对象。同样，双引号内的点号会失去其特殊含义并且变成普通的字符。

高级用户可以使用字符类等正则表达式记法，如 `[0-9]` 可以匹配任意数字。所有的正则表达式特殊字符都按照 POSIX 正则表达式所说的工作。以下字符除外：

- `.` 会按照上面所说的作为一种分隔符。
- `*` 会被翻译成正则表达式记号 `*`。
- `?` 会被翻译成 `.`。
- `$` 则按字面意思匹配。

根据需要，可以通过书写 `?`、`(R+)`、`(R|)` 和 `R?` 来分别模拟 `PATTERN` 字符 `.`、`R*` 和 `R?`。`$` 不需要作为一个正则表达式字符，因为 `PATTERN` 必须匹配整个名称，而不是像正则表达式的常规用法那样解释（换句话说，`$` 会被自动地追加到 `PATTERN` 上）。如果不希望该 `PATTERN` 的匹配位置被固定，可以在开头或者结尾写上 `*`。注意在双引号内，所有的正则表达式特殊字符会失去其特殊含义并且按照其字面意思进行匹配。另外，在操作符名称 `PATTERN` 中（即

\do 的 PATTERN 参数)，正则表达式特殊字符也按照字面意思进行匹配。

更改 SQL 语句之间分隔符命令，分隔符默认值为 “;”。

DELIMITER 命令为客户端设置一个分隔符。当用户设置分隔符后，gsql 客户端识别到分隔符时，会立即将 SQL 语句发送到服务端执行，但是服务端仍然将 “;” 看做 SQL 语句分隔符，并相应的处理 SQL 语句。

注意事项：

- delimiter 符号目前不是自由设定的，结束符范围有限制，目前接受大小写字母组合或特殊字符组合（~!/@/#/~/^/&/'/?/+/-/*//（除号）/%/</>/=），其中常见的用法是”//”。
- 符号组合中尽量使用无歧义符号组合，特殊符号组合（注释符：“*”、“” - “等）目前不支持用于 delimiter 命名。
- delimiter 长度范围：0~15。
- 设置的结束符的级别是会话级别的，当切换数据库时 delimiter_name 会设置为默认值 ‘;’。
- 用户如果想使用其他字符组合例如”adbc \$\$”，可以使用引号包含，例如 delimiter “adbc \$\$”，但使用时也需要使用引号包含，例如：select 1”adbc \$\$”。
- delimiter 分隔符只有 sql_compatibility = 'B'时支持。

2.1.6 常见问题处理

2.1.6.1 连接性能问题

- 开启了 log_hostname，但是配置了错误的 DNS 导致的连接性能问题。

在连接上数据库，通过 show log_hostname 语句，检查数据库中是否开启了 log_hostname 参数。

如果开启了相关参数，那么数据库内核会通过 DNS 反查客户端所在机器的主机名。这时如果数据库主节点配置了不正确或不可达的 DNS 服务器，那么会导致数据库建立连接过程较慢。此参数的更多信息，详见《GBase 8s V8.8.5 5.0.0_开发者指南》中“GUC 参数说明 > 错误报告和日志 > 记录日志的内容”章节中关于“log_hostname”的描述。

- 数据库内核执行初始化语句较慢导致的性能问题。

此种情况定位较难，可以尝试使用 Linux 的跟踪命令 strace。

```
strace gsql -U user_name -W password -d dbname -h host -p port -r -c '\q'
```

此时便会在屏幕上打印出数据库的连接过程。例如，较长时间停留在下面的操作上：

```
sendto(3, "Q\0\0\0\25SELECT VERSION()\0", 22, MSG_NOSIGNAL, NULL, 0) = 22  
poll([fd=3, events=POLLIN|POLLERR], 1, -1) = 1 ([fd=3, revents=POLLIN])
```

此时便可以确定是数据库执行 `SELECT VERSION()` 语句较慢。

在连接上数据库后，便可以通过执行 `explain performance select version()` 语句来确定初始化语句执行较慢的原因。更多信息，详见《GBase 8s V8.8.5 5.0.0_性能调优指南》中“性能调优 > SQL 调优指南 > SQL 执行计划介绍”章节。

另外还有一种场景不太常见：由于数据库主节点所在机器的磁盘满或故障，此时所查询等受影响，无法进行用户认证，导致连接过程挂起，表现为假死。解决此问题清理数据库主节点的数据盘空间便可。

- TCP 连接创建较慢问题。

此问题可以参考上面的初始化语句较慢排查的做法，通过 `strace` 跟踪。

例如，如果长时间停留在：

```
connect(3, {sa_family=AF_INET, sin_port=htons(61052),  
sin_addr=inet_addr("127.0.0.1")}, 16) = -1 EINPROGRESS (Operation now in  
progress)
```

那么说明客户端与数据库端建立物理连接过慢，此时应当检查网络是否存在不稳定、网络吞吐量太大的问题。

2.1.6.2 创建连接故障

- `gsql: could not connect to server: No route to host`

此问题一般是指定了不可达的地址或者端口导致的。请检查 `-h` 参数与 `-p` 参数是否添加正确。

- `gsql: FATAL: Invalid username/password,login denied.`

此问题一般是输入了错误的用户名和密码导致的，请联系数据库管理员，确认用户名和密码是否正确。

- `gsql: FATAL: Forbid remote connection with trust method!`

数据库由于安全问题，禁止远程登录时使用 `trust` 模式。这时需要修改 `pg_hba.conf` 里的

连接认证信息。具体的设置信息请参见《GBase 8s V8.8.5 5.0.0_开发者指南》中“管理数据库安全 > 客户端接入认证 > 配置文件参考”章节。

说明

- ▶ 不要修改 `pg_hba.conf` 中 GBase 8s 主机的相关设置, 否则可能导致数据库功能故障。
- ▶ 建议业务应用部署在 GBase 8s 之外, 而非 GBase 8s 内部。
- 连接数据库, 添加 “-h 127.0.0.1” 可以连接, 去掉后无法连接问题。

通过执行 SQL 语句 `show unix_socket_directory`, 检查数据库主节点使用的 Unix 套接字目录, 是否与 shell 中的环境变量 `$PGHOST` 一致。

如果检查结果不一致, 那么修改 `PGHOST` 环境变量到 GUC 参数 `unix_socket_directory` 指向的目录便可。

关于 `unix_socket_directory` 的更多信息, 详见《GBase 8s V8.8.5 5.0.0_开发者指南》中“GUC 参数说明 > 连接和认证 > 连接设置”章节中的说明。

- The "libpq.so" loaded mismatch the version of gsql, please check it.

此问题是由于环境中使用的 `libpq.so` 的版本与 `gsql` 的版本不匹配导致的。确认当前加载的 `libpq.so` 的版本, 并通过修改 `LD_LIBRARY_PATH` 环境变量来加载正确的 `libpq.so`。

- gsql: symbol lookup error: xxx/gsql: undefined symbol: libpqVersionString

此问题是由于环境中使用的 `libpq.so` 的版本与 `gsql` 的版本不匹配导致的 (也有可能是环境中存在 PostgreSQL 的 `libpq.so`), 确认当前加载的 `libpq.so` 的版本, 并通过修改 `LD_LIBRARY_PATH` 环境变量来加载正确的 `libpq.so`。

- gsql: connect to server failed: Connection timed out Is the server running on host "xx.xxx.xxx.xxx" and accepting TCP/IP connections on port xxxx?

此问题是由于网络连接故障造成。请检查客户端与数据库服务器间的网络连接。如果发现从客户端无法 PING 到数据库服务器端, 则说明网络连接出现故障。请联系网络管理人员排查解决。

- gsql: FATAL: permission denied for database "postgres" DETAIL: User does not have CONNECT privilege.

此问题是由于用户不具备访问该数据库的权限, 可以使用如下方法解决 (选一即可)。

方法一 使用管理员用户 `gbase` 连接数据库。

```
gsql -d postgres -U gbase -p port
```

方法二 赋予该用户访问数据库的权限。

```
GRANT CONNECT ON DATABASE postgres TO user_name;
```

说明

- 实际上，常见的许多错误操作也可能产生用户无法连接上数据库的现象。如用户连接的数据库不存在，用户名或密码输入错误等。这些错误操作在客户端工具也有相应的提示信息。例如：

```
$ gsql -d postgres -p 15400
gsql: FATAL: database "postgres" does not exist
$ gsql -d postgres -U user1 -W test,123 -p 15400
gsql: FATAL: Invalid username/password, login denied.
```

- `gsql: FATAL: sorry, too many clients already, active/non-active: 197/3.`

此问题是由于系统连接数量超过了最大连接数量。请联系数据库 DBA 进行会话连接数管理，释放无用会话。

关于查看用户会话连接数的方法如表 2-23。

会话状态可以在视图 `PG_STAT_ACTIVITY` 中查看。无用会话可以使用函数 `pg_terminate_backend` 进行释放。

```
postgres=# select datid,pid,state from pg_stat_activity;
 datid |          pid          | state
-----+-----+-----
 16947 | 140473306445568      | idle
 16947 | 140473346356992      | idle
 16947 | 140472335136512      | active
(3 rows)
```

其中 `pid` 的值即为该会话的线程 ID。根据线程 ID 结束会话。

```
postgres=# SELECT PG_TERMINATE_BACKEND(140473306445568);
```

显示类似如下信息，表示结束会话成功。

```
pg_terminate_backend
-----
t
```

(1 row)

表 2-23 查看会话连接数

描述	命令
查看指定用户的会话连接数上限。	<p>执行如下命令查看连接到指定用户的会话连接数上限。其中 -1 表示没有对用户设置连接数的限制。</p> <pre>postgres=# SELECT ROLNAME, ROLCONNLIMIT FROM PG_ROLES WHERE ROLNAME=' gbase' ; rolname rolconnlimit -----+----- gbase -1 (1 row)</pre>
查看指定数据库的会话连接数上限。	<p>执行如下命令查看连接到指定数据库 postgres 的会话连接数上限。其中 -1 表示没有对数据库 postgres 设置连接数的限制。</p> <pre>postgres=# SELECT DATNAME, DATCONNLIMIT FROM PG_DATABASE WHERE DATNAME=' postgres' ; datname datconnlimit -----+----- postgres -1 (1 row)</pre>
查看指定数据库已使用的会话连接数。	<p>执行如下命令查看指定数据库 postgres 上已使用的会话连接数。其中，1 表示数据库 postgres 上已使用的会话连接数。</p> <pre>postgres=# SELECT COUNT(*) FROM PG_STAT_ACTIVITY WHERE DATNAME=' postgres' ; count ----- 12 (1 row)</pre>

- gsql: wait xxx.xxx.xxx.xxx:xxxx timeout expired

gsql 在向数据库发起连接的时候，会有 5 分钟超时机制，如果在这个超时时间内，数据库未能正常的对客户端请求进行校验和身份认证，那么 gsql 会退出当前会话的连接过程，并报出如上错误。

一般来说，此问题是由于连接时使用的 -h 参数及 -p 参数指定的连接主机及端口有误（即错误信息中的 xxx 部分），导致通信故障；极少数情况是网络故障导致。要排除此问题，请

检查数据库的主机名及端口是否正确。

- `mysql: could not receive data from server: Connection reset by peer.`

同时，检查数据库主节点日志中出现类似如下日志“FATAL: cipher file “/data/dbnode/server.key.cipher” has group or world access”，一般是由于数据目录或部分关键文件的权限被误操作篡改导致。请参照其他正常实例下的相关文件权限，修改回来便可。

- `mysql: FATAL: GSS authentication method is not allowed because XXXX user password is not disabled.`

目标数据库主节点的 `pg_hba.conf` 里配置了当前客户端 IP 使用“gss”方式来做认证，该认证算法不支持用作客户端的身份认证，请修改到“sha256”后再试。配置方法见《GBase 8s V8.8.5 5.0.0_开发者指南》中“管理数据库安全 > 客户端接入认证 > 配置文件参考”章节。

说明

- 不要修改 `pg_hba.conf` 中 GBase 8s 主机的相关设置，否则可能导致数据库功能故障。
- 建议业务应用部署在 GBase 8s 之外，而非 GBase 8s 内部。

2.1.6.3 其他故障

- 出现因“总线错误”（Bus error）导致的 core dump 或异常退出

一般情况下出现此种问题，是进程运行过程中加载的共享动态库（在 Linux 为 .so 文件）出现变化；或者进程二进制文件本身出现变化，导致操作系统加载机器的执行码或者加载依赖库的入口发生变化，操作系统出于保护目的将进程杀死，产生 core dump 文件。

解决此问题，重试便可。同时请尽可能避免在升级等运维操作过程中，在 GBase 8s 内部运行业务程序，避免升级时因替换文件产生此问题。

说明

- 此故障的 core dump 文件的可能堆栈是 `dl_main` 及其子调用，它是操作系统用来初始化进程做共享动态库加载的。如果进程已经初始化，但是共享动态库还未加载完成，严格意义上来说，进程并未完全启动。

3 服务端工具

在使用 GBase 8s 过程中，经常需要对 GBase 8s 进行安装、卸载以及健康管理。为了简

单、方便的维护数据库，GBase 8s 提供了一系列的管理工具。

3.1 gs_cgroup

3.1.1 背景信息

集群环境下做批量任务处理时，多任务复杂性会导致不同机器间的负载差距很大。为了充分利用集群资源，负载管理变得尤为重要。gs_cgroup 是 GBase 8s 提供的负载管理工具。负责创建默认控制组、创建用户自定义控制组、删除用户自定义控制组、更新用户自定义组的资源配额和资源限额、显示控制组配置文件内容、显示控制组树形结构和删除用户的所有控制组。

gs_cgroup 工具为使用数据库的操作系统用户创建 Cgroups 配置文件，并且在操作系统中生成用户设定的 Cgroups。同时为用户提供增加、删除 Cgroups、更新 Cgroups 资源配额、设定 Cgroups 的 CPU 或 IO 限额、设定异常处理阈值及相应操作等服务。此工具只负责当前操作系统节点的 Cgroups 操作，使用时需在各个节点上调用相同命令语句进行统一配置。

这里假设读者已经了解了负载管理的相关原理，具体请参考《GBase 8s V8.8.5 5.0.0_开发者指南》中“资源负载管理”章节。

3.1.2 语法

```
gs_cgroup [OPTION]
```

3.1.3 参数说明

- -a [--abort]

对满足设定的异常阈值的作业执行终止动作。

- -b pct

Backend Cgroups 占用 Top Backend 资源的百分比，需同时指定 -B backendname 参数。

取值范围: 为 1 ~ 99。在不进行设置的情况下，默认 CPU 配额设置为 Vacuum 控制组占 20%，DefaultBackend 控制组占 80%。Vacuum 和 DefaultBackend 控制组配额之和设置应小于 100%。

- -B name

Backend Cgroups 名称，仅可指定 -u 参数来更新此 Cgroups 的资源配额。

-b percent 和 -B backendname 参数共同作用来控制数据库后备线程的资源比例。

取值范围：字符串，最长为 64 个字节。

- -c

创建 Cgroups 并指定其标识。

普通用户指定“-c”和“-S classname”可以创建新的 Class Cgroups，如果再指定“-G groupname”则创建属于 Class Cgroups 的 Workload Cgroups，新的 Workload Cgroups 位于 Class Cgroups 的最底层（4 层是最底层）。

- -d

删除 Cgroups 及标识。

普通用户指定“-d”和“-S classname”可以删除已有的 Class Cgroups，如果再指定“-G groupname”则删除属于 Class Cgroups 的 Workload Cgroups，并将附属的线程重置到“DefaultClass:DefaultWD:1”Cgroups 中；若删除的 Workload Cgroups 位于 Class Cgroups 的高层（1 层是最高层），则需调整低层的 Cgroups 的层级，重建新的 Cgroups 附属的线程加载到新的 Cgroups。

- -E data

设定异常阈值，目前阈值包括：blocktime、elapsedtime、allcputime、spillsizes、broadcastsize 以及 qualificationtime 和 cpuskewpercent，指定多个阈值时用“，”分隔。参数值 0 表示取消设置，设置不合法的值时会提示。

表 3-1 异常阈值类型

异常阈值类型	说明	取值范围（0 表示取消设置）	支持的异常操作
blocktime	作业的阻塞时间，单位秒。包括全局并发排队以及局部并发排队的总时间。	0~UINT_MAX	abort

异常阈值类型	说明	取值范围（0表示取消设置）	支持的异常操作
elapsedtime	作业的已被执行时间，单位秒。从开始执行到当前所消耗的时间。	0~UINT_MAX	abort
allcputime	作业在所有数据库节点上执行时所耗费的 CPU 总时间，单位秒。	0~UINT_MAX	abort, penalty
cpuskepercent	作业在数据库节点上执行时的 CPU 时间的倾斜率，依赖于 qualificationtime 的设置。	0~100	abort, penalty
qualificationtime	检查作业执行 cpu 倾斜率的间隔时间，单位秒，需同 cpuskepercent 一起设置。	0~UINT_MAX	none
spillsize	作业在数据库节点上下盘的数据量，单位 MB。	0~UINT_MAX	abort
broadcastsize	作业在数据库节点上算子大表广播数据量，单位 MB。	0~UINT_MAX	abort

- -h [--help]

显示命令帮助信息。

- -H

用户环境中\$GAUSSHOME 信息。

取值范围：字符串，最长为 1023 个字符。

- -f

仅 Gaussdb 控制组可用。设置使用的核数范围，范围必须是 a-b 或 a 的形式。其他控制组可以使用--fixed 进行设置核数范围。

- --fixed

设置控制组使用的核数范围比例占上一层级的百分比或者设置 IO 资源。

设置核数范围比例时--fixed 设置核数范围与-s、-g、-t、-b 参数一起使用。

核数比例范围 0-100，同一层级的核数比例总和小于或者等于 100，0 代表核数与上一层级相同，对于所有的控制组，CPU 限额默认设置为 0。-f 和--fixed 不能同时设置。设置--fixed 之后，-f 设置的范围自动失效。设置的以 quota 值在-p 中进行显示。

设置 IO 资源配额时，与-R、-r、-W、-w 参数一起使用。

- -g pct

指定 Workload Cgroups 的资源占用 Class Cgroups 资源的百分比，需同时指定-G groupname 参数；用于创建（-c 参数）或更新（-u 参数）Workload Cgroups。

取值范围：1 ~ 99。默认 Workload 控制组 CPU 配额设置为 20%。各 Workload 控制组配额之和应小于 99%。

- -G name

指定 Workload Cgroups 的名称，需同时指定-S classname 参数来表示该 group 属于哪个 Class Cgroups；可以连同-c 参数创建新的 Cgroups、-d 参数删除此 Cgroups 及-u 更新此 Cgroups 的资源配额；需要注意，此名称不可是 Timeshare Cgroups 的默认名称，如“Low”、“Medium”、“High”或“Rush”。

如果用户自己创建 Workload 控制组，指定名称时不能指定带 ':' 的名称。不能创建同名控制组。

取值范围：字符串，最长为 28 个字节。

- -N [--group] name

可以将组名简写成 *classname:workloadname*。

- -p

显示 Cgroups 配置文件的信息。

- -P

显示 Cgroups 树形结构信息。

- --penalty

对满足设定的异常阈值的作业执行降级动作, 如果没有设定任何操作, 则该操作将为默认操作。

- -r data

仅用于更新 I/O 资源读数据上限, 用于设置 `blkio.throttle.read_bps_device` 的数值; 为字符串类型, 该字符串由 `major:minor value` 构成, 其中 `major` 为要访问的磁盘的主设备号, `minor` 为要访问的磁盘的次设备号, `value` 为设备每秒读操作次数上限数值, 取值范围为 `0 ~ ULONG_MAX`, 其中取值 `0` 用来初始化此字段为空; 需和 `-u` 参数及 Cgroups 名称一同使用; 如果 Class Cgroups 和 Workload Cgroups 的名称同时指定, 则只应用到 Workload Cgroups。

取值范围: 字符串, 最长为 32 个字符。

- -R data

仅用于更新 IO 资源每秒读操作次数上限, 用于设置 `blkio.throttle.read_iops_device` 的数值; 取值信息同 `-r` 参数; 需和 `-u` 参数及 Cgroups 名称一同使用; 如果 Class Cgroups 和 Workload Cgroups 的名称同时指定, 则只应用到 Workload Cgroups。

取值范围: 字符串, 最长为 32 个字符。

- --recover

仅用于回退 Class 控制组和 Workload 控制组的增删改操作, 且只能回退一步。

- --revert

恢复控制组为默认状态。

- -D mpoint

指定的挂载点, 默认的挂载点 “`/dev/cgroup/subsystem`”。

- -m

挂载 cgroup。

- -M

取消挂载 cgroup。

- -U

数据库用户名称。

- --refresh

刷新控制组状态。

- -s pct

指定 Class Cgroups 的资源占用 Top Class Cgroups 资源的百分比，需同时指定 -S classname 参数；用于创建（-c 参数）或更新（-u 参数）Class Cgroups。

取值范围：1-99。默认 Class 控制组的 CPU 配额设置为 20%，R6C10 版本中，Class 控制组的 CPU 配额设置为 40%，升级过程中，不会对此配额做更新。新创建的 Class 控制组的 CPU 配额以及默认的 DefaultClass 之和应小于 100%。

- -S name

指定 Class group 的名称；可以连同 -c 参数创建新的 Cgroups、-d 参数删除此 Cgroups 及 -u 更新此 Cgroups 的资源配额。创建子 Class 名称不能带 ':'。

取值范围：字符串，最长为 31 个字节。

- -t percent

指定 Top Cgroups (Root、Gaussdb:user、Backend 和 Class Cgroups) 占用资源的百分比，需同时指定 -T name 参数。若指定 -T Root 所在的 Cgroups，其在 Cgroups 配置文件中显示的名称为“Root”，此数值代表的含义是 blkio.weight 值的百分比，最小值为 10%，CPU 资源配额如 cpu.shares 的数值不可修改；若指定 Gaussdb:user Cgroups，则表示占用整个系统 CPU 资源的百分比（可根据 Root Cgroups 的默认 CPU 配额 1024 得出该组的 cpu.shares 数值，此数值默认当前系统仅含有 1 套数据库环境），对于 IO 资源配额为 1000，不再变化；若指定 Class 或 Backend Cgroups，则表示资源占用 Gaussdb Cgroups 资源的百分比。

取值范围：1 ~ 99。默认 Class 控制组配额为 60%，Backend 控制组配额为 40%。修改 Class 控制组配额，同时会自动更新 Backend 控制组配额，使两者之和保持 100%。

- -T name

指定 Top Cgroups 的名称。

取值范围：字符串，最长为 64 个字节。

- -u

更新 Cgroups。

- -V [--version]

显示 gs_cgroup 工具的版本信息。

- -w data

仅用于更新 I/O 资源每秒写数据上限，用于设置 blkio.throttle.write_bps_device 的数值。取值信息同-r 参数，需和-u 参数及 Cgroups 名称一同使用。如果 Class Cgroups 和 Workload Cgroups 的名称同时指定，则只应用到 Workload Cgroups。

取值范围：字符串，最长为 32 个字符。

- -W data

仅用于更新 IO 资源每秒写操作次数上限，用于设置 blkio.throttle.write_iops_device 的数值。取值信息同-r 参数，需和-u 参数及 Cgroups 名称一同使用。如果 Class Cgroups 和 Workload Cgroups 的名称同时指定，则只应用到 Workload Cgroups。

取值范围：字符串，最长为 32 个字符。

说明

- 对于磁盘 major:minor 的编号获取，可以通过下面方式。例如，获取/mpp 目录对应的磁盘编号：

```
$ df
Filesystem      1K-blocks      Used Available Use% Mounted on
/dev/sda1       524173248    41012784 456534008   9% /
devtmpfs        66059264         236 66059028   1% /dev
tmpfs           66059264          88 66059176   1% /dev/shm
/dev/sdb1       2920486864 135987592 2784499272   5% /data
/dev/sdc1       2920486864 24747868 2895738996   1% /data1
/dev/sdd1       2920486864 24736704 2895750160   1% /mpp
/dev/sde1       2920486864 24750068 2895736796   1% /mpp1

$ ls -l /dev/sdd
brw-rw---- 1 root disk 8, 48 Feb 26 11:20 /dev/sdd
```

须知

- 这里一定要查 `sdd` 的磁盘号，不能是 `sdd1` 的磁盘号。否则执行时会报错。如果更新 IO 的限额配置信息超过了可允许的最大配置字符串，则新的更新不存储在配置文件中。如当前设置的字符串长度为 96，更新 IO 的磁盘数量大于 8 个，则有可能超出字符串限制，更新成功，但是不存储在配置文件中。

3.1.4 示例

使用普通用户或数据库管理员执行命令。

- (1) 前置条件：需设置 `GAUSSHOME` 环境变量为数据库安装目录；且 `root` 用户已创建普通用户默认的控制组。例如，创建 `gbase` 用户控制组。

```
[root@gbasehost ~]# gs_cgroup -U gbase -H /home/gbase/gbase_db/app -c
```

- (2) 以普通用户身份，创建控制组及设置对应的资源配额，以便在数据库中运行作业时，指定到此控制组，使用此控制组管理的资源；通常数据库管理员为每个数据库用户创建 Class 组。

- ① 创建 Class 控制组

例如，创建当前用户新的 Class Cgroups 命名为“`class1`”，资源配额为总 Class 的 40%：

```
# gs_cgroup -c -S class1 -s 40
```

- ② 创建 Workload 控制组

创建当前用户新 Cgroups —— “`class1`” 下属的 Workload 控制组，命名为“`grp1`”，资源配额为“`class1`” Cgroups 的 20%：

```
# gs_cgroup -c -S class1 -G grp1 -g 20
```

- (3) 更新已创建控制组的资源配额。

- ① 更新动态资源配额。

例如，更新当前用户下属“`grp1`” Cgroups 资源配额为“`class1`” Cgroups 的 30%。

```
# gs_cgroup -u -S class1 -G grp1 -g 30
```

- ② 更新限制资源配额。

```
# gs_cgroup --fixed -u -S class1 -G grp1 -g 30
```

更新当前用户的 class1 Cgroups 下属 grp1 Cgroups 限制使用 CPU 核数范围占上一级 class1 可用核数的 30%。

(4) 设置异常处理信息。

① 设置组 class1:grp1, 当作业阻塞到 1200 秒或执行 2400 秒时执行终止动作:

```
# gs_cgroup -S class1 -G grp1 -E "blocktime=1200,elapsedtime=2400" -a
```

② 设置组 class1:grp1, 当作业下盘数据量达到 256MB 或大表广播数据量达到 100MB 时执行终止动作:

```
# gs_cgroup -S class1 -G grp1 -E "spillsize=256,broadcastsize=100" -a
```

③ 设置组 class1, 当作业在所有数据库节点上 CPU 总时间到达 100s 时执行降级动作:

```
# gs_cgroup -S class1 -E "allcputime=100" --penalty
```

须知

- 给控制组设置异常处理信息, 需要确保对应的控制组已经创建。指定多个阈值时用“;”分隔, 不指定任何动作时默认为“penalty”操作。

(5) 设置控制组使用的核数范围。

例如, 设置组 class1:grp1 的核数范围占 Class 控制组的 20%。

```
# gs_cgroup -S class1 -G grp1 -g 20 --fixed -u
```

须知

- Class 或 Workload 核数范围必须通过指定--fixed 参数设置。

(6) 回退上一个步骤。

```
# gs_cgroup --recover
```

说明

- --recover 只支持对 Class 控制组和 Workload 控制组的增删改操作进行回退, 且只支持回退一次操作。

(7) 查看已创建的控制组信息

① 查看配置文件中控制组信息。例如：

```
# gs_cgroup -p
```

返回如下控制组配置信息：

```
Top Group information is listed:
GID:  0 Type: Top    Percent (%): 1000 ( 50) Name: Root           Cores:
0-1
GID:  1 Type: Top    Percent (%):  500 ( 50) Name: Gaussdb:gbase     Cores:
0-1
GID:  2 Type: Top    Percent (%):  200 ( 40) Name: Backend           Cores:
0-1
GID:  3 Type: Top    Percent (%):  300 ( 60) Name: Class             Cores:
0-1

Backend Group information is listed:
GID:  4 Type: BAKWD  Name: DefaultBackend   TopGID:  2 Percent (%): 160 (80)
Cores: 0-1
GID:  5 Type: BAKWD  Name: Vacuum             TopGID:  2 Percent (%):  40 (20)
Cores: 0-1

Class Group information is listed:
GID: 20 Type: CLASS  Name: DefaultClass     TopGID:  3 Percent (%): 100 (20)
MaxLevel: 1 RemPCT: 100 Cores: 0-1
GID: 21 Type: CLASS  Name: class1             TopGID:  3 Percent (%): 200 (40)
MaxLevel: 2 RemPCT:  70 Quota (%): 40 Cores: 0-0

Workload Group information is listed:
GID: 86 Type: DEFWD  Name: grp1:2             ClsGID: 21 Percent (%):  60 (30)
WDLevel:  2 Cores: 0-0

Timeshare Group information is listed:
GID: 724 Type: TSWD  Name: Low                Rate: 1
GID: 725 Type: TSWD  Name: Medium             Rate: 2
GID: 726 Type: TSWD  Name: High                Rate: 4
GID: 727 Type: TSWD  Name: Rush                Rate: 8

Group Exception information is listed:
GID:  20 Type: EXCEPTION Class: DefaultClass
PENALTY: QualificationTime=1800 CPUSkewPercent=30

GID:  21 Type: EXCEPTION Class: class1
```


PENALTY: AllCpuTime=100

GID: 86 Type: EXCEPTION Group: class1:grp1:2

ABORT: BlockTime=1200 ElapsedTime=2400 SpillSize=256 BroadcastSize=100

控制组配置信息描述如表 3-1 所示。

表 3-2 控制组配置信息

GID	类型	名称	Percent(%)信息	特定信息
0	Top 控制组	Root	1000 代表总的系统资源为 1000 份。 括号中的 50 代表 IO 资源的 50%。 GBase 8s 不通过控制组对 IO 资源做控制, 因此下面其他控制组信息中仅涉及 CPU 配额情况。	---
1		Gaussdb:gbase	系统中只运行一套数据库程序, 默认配额为 833, 数据库程序和非数据库程序的比值为 (833:167=5:1)	---
2		Backend	Backend 和 Class 括号中的 40 和 60, 代表 Backend 占用 Gaussdb:dbuser 控制组 40%的资源, Class 占用 Gaussdb:dbuser 控制组 60%的资源。	---
3		Class		---
4	Backend 控制组	DefaultBackend	括号中的 80 和 20 代表 DefaultBackend 和	TopGID: 代表 Top 类型控制组

GID	类型	名称	Percent(%)信息	特定信息
5		Vacuum	Vacuum 占用 Backend 控制组 80%和 20%的资源。	中 Backend 组的 GID, 即 2。
20	Class 控制组	DefaultClass	DefaultClass 和 class1 的 20 和 40 代表占 Class 控制组 20%和 40%的资源。因为当前只有两个 Class 组, 所有它们按照 20:40 的比例分配 Class 控制组 499 的系统配额, 则分别为 166 和 332。	TopGID: 代表 DefaultClass 和 class1 所属的上层控制 (Top 控制组中的 Class 组) 的 GID, 即 3。
21		class1		MaxLevel: Class 组当前含有的 Workload 组的最大层次, DefaultClass 没有 Workload Cgroup, 其数值为 1。
86	Workload 控制组	grp1:2 (该名称由 Workload Cgroup Name 和其在 class	根据设置, 其占 class1 的百分比为 30, 则为 $332*30\%=99$ 。	RemPCT: 代表 Class 组分配 Workload 组后剩余的资源百分比。如 class1 中剩余的百分比为 70。
				$332*30\%=99$ 。 ClsGID: 代表 Workload 控制组所属的上层控制组 (class1

GID	类型	名称	Percent(%)信息	特定信息
		中的层级组成,它是class1的第一个Workload组,层级为2,每个Class组最多10层Workload Cgroup。)		控制组)的GID。 WDLevel: 代表当前 Workload Cgroup 在对应的 Class 组所在的层次。
724	Timeshare 控制组	Low	---	Rate : 代表 Timeshare 中的分配比例, Low 最少为 1, Rush 最高为 8。这四个 Timeshare 组的资源配比为 Rush:High:Medium:Low=8:4:2:1。
725		Medium	---	
726		High	---	
727		Rush	---	

② 查看操作系统中树形结构的控制组信息

gs_cgroup -P 显示控制组树形结构信息,其中 shares 代表操作系统中 CPU 资源的动态资源配额“cpu.shares”的数值, cpus 代表操作系统中 CPuset 资源的动态资源限额“cpuset.cpus”的数值,指的是该控制组能够使用的核数范围。

```
# gs_cgroup -P

Top Group information is listed:
GID:  0 Type: Top   Percent(%) : 1000 ( 50) Name: Root           Cores:
0-1
GID:  1 Type: Top   Percent(%) : 500 ( 50) Name: Gaussdb:gbase    Cores:
0-1
```

```

GID: 2 Type: Top Percent(%): 200( 40) Name: Backend Cores:
0-1
GID: 3 Type: Top Percent(%): 300( 60) Name: Class Cores:
0-1

Backend Group information is listed:
GID: 4 Type: BAKWD Name: DefaultBackend TopGID: 2 Percent(%): 160(80)
Cores: 0-1
GID: 5 Type: BAKWD Name: Vacuum TopGID: 2 Percent(%): 40(20)
Cores: 0-1

Class Group information is listed:
GID: 20 Type: CLASS Name: DefaultClass TopGID: 3 Percent(%): 100(20)
MaxLevel: 1 RemPCT: 100 Cores: 0-1
GID: 21 Type: CLASS Name: class1 TopGID: 3 Percent(%): 200(40)
MaxLevel: 2 RemPCT: 70 Quota(%): 40 Cores: 0-0

Workload Group information is listed:
GID: 86 Type: DEFWD Name: grp1:2 ClsGID: 21 Percent(%): 60(30)
WDLevel: 2 Cores: 0-0

Timeshare Group information is listed:
GID: 724 Type: TSWD Name: Low Rate: 1
GID: 725 Type: TSWD Name: Medium Rate: 2
GID: 726 Type: TSWD Name: High Rate: 4
GID: 727 Type: TSWD Name: Rush Rate: 8

Group Exception information is listed:
GID: 20 Type: EXCEPTION Class: DefaultClass
PENALTY: QualificationTime=1800 CPUSkewPercent=30

GID: 21 Type: EXCEPTION Class: class1
PENALTY: AllCpuTime=100

GID: 86 Type: EXCEPTION Group: class1:grp1:2
ABORT: BlockTime=1200 ElapsedTime=2400 SpillSize=256 BroadcastSize=100

[gbase@gbase8s_5_124 gbase_package]$ gs_cgroup -P
Mount Information:
cpu:/sys/fs/cgroup/cpu,cpuacct
cpuacct:/sys/fs/cgroup/cpu,cpuacct

```

```

cpuset:/sys/fs/cgroup/cpuset
memory:/sys/fs/cgroup/memory

Group Tree Information:
- Gausssdb:gbase (shares: 1024, cpus: 0-1, weight: 0)
  - Class (shares: 6144, cpus: 0-1, weight: 0)
    - class1 (shares: 4096, cpus: 0, weight: 0)
      - RemainWD:1 (shares: 1000, cpus: 0, weight: 0)
        - RemainWD:2 (shares: 7000, cpus: 0, weight: 0)
          - Timeshare (shares: 1024, cpus: 0,
weight: 0)
            - Rush (shares: 8192, cpus: 0,
weight: 0)
              - High (shares: 4096, cpus: 0,
weight: 0)
                - Medium (shares: 2048, cpus: 0,
weight: 0)
                  - Low (shares: 1024, cpus: 0,
weight: 0)
                    - grp1:2 (shares: 3000, cpus: 0, weight: 0)
                      - TopWD:1 (shares: 9000, cpus: 0, weight: 0)
                        - DefaultClass (shares: 2048, cpus: 0-1, weight: 0)
                          - RemainWD:1 (shares: 1000, cpus: 0-1, weight: 0)
                            - Timeshare (shares: 1024, cpus: 0-1, weight: 0)
                              - Rush (shares: 8192, cpus: 0-1, weight:
0)
                                - High (shares: 4096, cpus: 0-1, weight:
0)
                                  - Medium (shares: 2048, cpus: 0-1,
weight: 0)
                                    - Low (shares: 1024, cpus: 0-1, weight:
0)
                                      - TopWD:1 (shares: 9000, cpus: 0-1, weight: 0)
                                        - Backend (shares: 4096, cpus: 0-1, weight: 0)
                                          - Vacuum (shares: 2048, cpus: 0-1, weight: 0)
                                            - DefaultBackend (shares: 8192, cpus: 0-1, weight: 0)
- CM:gbase (shares: 8192, cpus: 0-1, weight: 0)

```

(8) 删除 Class 控制组和 Workload 控制组。

① 删除 Workload 控制组。例如，删除当前用户已建的“grp1” Cgroups。

```
# gs_cgroup -d -S class1 -G grp1
```

- ② 删除 Class 控制组。例如，删除当前用户已建的“class1” Cgroups。

```
# gs_cgroup -d -S class1
```

须知

- 如果删除 Class 控制组，则 Workload 控制组也被删除。

3.2 gs_check

3.2.1 背景信息

gs_check 改进增强，统一化当前系统中存在的各种检查工具，例如 gs_check、gs_checkos 等，帮助用户在 GBase 8s 运行过程中，全量的检查 GBase 8s 运行环境、操作系统环境、网络环境及数据库执行环境，也有助于在 GBase 8s 重大操作之前对各类环境进行全面检查，有效保证操作执行成功。

注意事项

- 必须指定 -i 或 -e 参数，-i 会检查指定的单项，-e 会检查对应场景配置中的多项。
- 如果 -i 参数中不包含 root 类检查项或 -e 场景配置列表中没有 root 类检查项，则不需要交互输入 root 权限的用户及其密码。
- 可使用 --skip-root-items 跳过检查项中包含的 root 类检查，以免需要输入 root 权限用户及密码。
- MTU 值不一致时可能导致检查缓慢或进程停止响应，当巡检工具出现提示时请修改各节点 MTU 值一致后再进行巡检。
- 交换机不支持当前设置的 MTU 值时，即使 MTU 值一致也会出现通信问题引起进程停止响应，需要根据交换机调整 MTU 大小。

3.2.2 语法

- 单项检查：

```
gs_check -i ITEM [...] [-U USER] [-L] [-l LOGFILE] [-o OUTPUTDIR]
[--skip-root-items] [--set] [--routing]
```

- 场景检查：

```
gs_check -e SCENE_NAME [-U USER] [-L] [-l LOGFILE] [-o OUTPUTDIR]
[--skip-root-items] [--time-out=SECS] [--set] [--routing] [--skip-items]
```

- 显示帮助信息：

```
gs_check -? | --help
```

- 显示版本号信息：

```
gs_check -V | --version
```

3.2.3 参数说明

- -U

运行 GBase 8s 的用户名称。

取值范围：运行 GBase 8s 的用户名称。

- -L

本地执行。

- -i

指定检查项。格式-i CheckXX 详细的检查项请参见表 3-3。

- -e

场景检查项。默认的场景有 inspect (例行巡检)、upgrade (升级前巡检)、binary_upgrade (就地升级前巡检)、health (健康检查巡检)、install (安装) 等，用户可以根据需求自己编写场景，详见[用户自定义场景](#)。

- -l

指定日志文件路径，指定路径时需添加.log 后缀。

- -o

指定检查结果输出文件夹路径。

- --skip-root-items

跳过需要 root 权限执行的检查项。

- `--skip-items`
跳过指定的检查项。
- `--format`
设置结果报告的格式。
- `--set`
修复支持设置的 Abnormal 项。
- `--cid`
检查 ID，仅被内部 check 进程使用。
- `--time-out`
设置超时时间。单位为秒，默认为 1500s，若用户自定义超时时间不得少于 1500s。
- `--routing`
指定业务 IP 的网段，格式为 IP 地址:子网掩码。
- `--disk-threshold="PERCENT"`
检查磁盘占用时可选指定告警阈值，可指定 1-99 之间的整数，不输入则默认为 90。检查其他项时不需要该参数。
- `-, --help`
显示帮助信息。
- `-V, --version`
显示版本号信息。

表 3-3 GBase 8s 状态检查表

状态	巡检项	检查内容	是否支持 <code>--set</code>

状态	巡检项	检查内容	是否支持 --set
os	CheckCPU (检查 CPU 使用率)	检查主机 CPU 占用率, 如果 idle 大于 30%并且 iowait 小于 30%。则检查项通过, 否则检查项不通过。	否
	CheckFirewall (检查防火墙状态)	检查主机防火墙状态, 如果防火墙关闭则检查项通过, 否则检查项不通过。	是
	CheckTimeZone (检查时区一致性)	检查 GBase 8s 内各节点时区, 如果时区一致则检查通过, 否则检查项不通过。	否
	CheckSysParams (检查系统参数)	检查各节点操作系统参数, 判断是否等于预期值。检查项不满足 warning 域则报 warning, 不满足 NG 域则检查项不通过, 并打印不满足项。	是
	CheckOSVer (检查操作系统版本)	检查 GBase 8s 内各个节点的操作系统版本信息, 如果满足版本兼容列表且 GBase 8s 在同一混搭列表中则检查通过, 否则检查不通过。	否
	CheckNTPD (检查 NTPD 服务)	检查系统 NTPD 服务, 如果服务开启且各节点时间误差在 1 分钟以内则检查项通过, 否则检查项不通过。	否
	CheckTHP (检查 THP 服务)	检查系统 THP 服务, 如果服务开启则检查项通过, 否则检查项不通过。	是

状态	巡检项	检查内容	是否支持 --set
	CheckSshdService(检查 sshd 服务是否已启动)	检查系统是否存在 sshd 服务, 若存在则检查项通过, 否则检查项不通过。	否
	CheckCronService(检查 crontab 服务是否已启动)	检查系统是否存在 crontab 服务, 若存在则检查项通过, 否则检查项不通过。	是
	CheckCrontabLeft(检查 crontab 是否有残留 Gauss 相关信息)	检查 crontab 是否残留 Gauss 相关信息, 若无该信息则检查项通过, 否则检查项不通过。	是
	CheckDirLeft(检查文件目录是否有残留)	检查文件目录 (/home/gbase/data) 是否存在。若 mount 目录包含此目录则忽略; 若不存在则查项通过, 否则检查项不通过。	是
	CheckProcessLeft(检查进程是否有残留)	检查是否残留 gaussdb 和 gbase 进程, 若未残留则检查项通过, 否则检查项不通过。	是
	CheckStack(栈深度检查)	检查栈深度, 若各个节点不一致则报 warning, 若大于等于 3072 则检查项通过, 否则不通过。	是

状态	巡检项	检查内容	是否支持 --set
	CheckgbaseUserExist(检查 gbase 用户是否存在)	检查是否存在 gbase 用户，若不存在 gbase 用户则检查项通过，否则检查项不通过。	是
	CheckPortConflict(检查数据库节点端口是否占用)	检查数据库节点端口是否已被占用，若未占用则检查项通过，否则检查项不通过。	是
	CheckSysPortRange (检查 ip_local_port_range 设置范围)	检查 ip_local_port_range 系统参数范围，若范围在 26000~65535 则检查项通过，否则检查项不通过。	是
	CheckEtcHosts (检查/etc/hosts 中是否有重复地址以及 localhost 配置)	检查/etc/hosts 没有配置 localhost 检查项不通过，存在带有#GBase 8s 注释的映射则检查项不通过，相同 IP 不同 hostname 则检查项不通过，否则通过，若 hostname 相同，但 ip 不同检查项不通过。	否
	CheckCpuCount (检查 CPU 核数)	检查 CPU 核心与可用 CPU 不符检查项不通过，相符但存在不可用信息 Warning。所有节点 CPU 信息不相同检查项不通过。	否
	CheckHyperThread (检查超线程是否)	检查超线程，若打开则检查项通过，否则检查项不通过。	否

状态	巡检项	检查内容	是否支持 --set
	打开)		
	CheckMemInfo (检查内存总大小)	检查各节点总内存大小是否一致, 若检查结果一致, 则检查项通过, 否则报 warning。	否
	CheckSshdConfig (检查 sshd 服务配置是否正确)	检查/etc/ssh/sshd_config 文件, (a>PasswordAuthentication=yes; (b)MaxStartups=1000; (c)UseDNS=no; (d)ClientAliveInterval 大于 10800 或者等于 0 配置如上所示则检查项通过, 若 a、c 配置不正确则报 warning, b、d 配置不正确则检查项不通过。	是
	CheckMaxHandle (检查句柄最大设置)	检查操作系统最大句柄值, 如果该值大于等于 1000000 则检查项通过, 否则检查项不通过。	是
	CheckKernelVer (检查内核版本)	检查各节点系统内核版本信息, 如果版本信息一致则检查项通过, 否则报 Warning。	否
	CheckEncoding (检	检查 GBase 8s 内各个节点的系统编码, 如果编码一致则检查项通过, 否则检查	否

状态	巡检项	检查内容	是否支持 --set
	查编码格式)	项不通过。	
	CheckBootItems (检查启动项)	检查是否有手动添加的启动项, 如果没有则检查通过, 否则检查不通过。	否
	CheckDropCache (检查 DropCache 进程)	检查各节点是否有 dropcache 进程在运行, 若有则检查通过, 否则检查不通过。	否
	CheckFilehandle (检查文件句柄)	此检查项检查以下两项, 两项都通过为通过, 否则为不通过: 检查每个 gaussdb 进程打开的进程数是否超过 80 万, 不超过则检查通过, 否则检查不通过。 检查是否有 slave 进程使用的句柄数超过 master 进程, 如果没有则检查通过, 否则检查不通过。	否
	CheckKeyProAdj (检查关键进程 gbase_adj 的值)	检查所有关键进程, 如果所有关键进程的 gbase_adj 值为 0, 则通过, 否则不通过。	否
	CheckMaxProcMemory (检查 max_process_memory 参数设置是否合	检查数据库节点的 max_process_memory 值, 判断该参数的值是否大于 1G, 若不大于则检查项通过, 否则检查项不通过。	是

状态	巡检项	检查内容	是否支持 --set
	理)		
device	CheckSwapMemory (检查交换内存)	检查交换内存和总内存大小, 若检查结果为 0 则检查项通过, 否则检查项报 Warning 大于总内存时检查项不通过。	是
	CheckLogicalBlock (检查磁盘逻辑块)	检查磁盘逻辑块大小, 若为 512 则检查项通过, 否则检查项不通过。	是
	CheckIOrequestqueue (检查 IO 请求)	检查 IO 值, 如果该值为 32768 则检查项通过, 否则检查项不通过。	是
	CheckMaxAsyIOrequests (检查最大异步 IO 请求)	获取当前异步 IO 请求值, 当前异步 IO 请求值大于当前节点数据库实例数 *1048576 和 104857600 则检查项通过, 否则检查项不通过。	是
	CheckIOConfigure (检查 IO 配置)	检查 IO 配置, 如果是 deadline 则检查项通过, 否则检查项不通过。	是
	CheckBlockdev (检查磁盘预读块)	检查磁盘预读块大小, 如果预读块大小为 16384 则检查项通过, 否则检查项不通过。	是
	CheckDiskFormat (检查磁盘格式参)	检查磁盘 XFS 格式信息, 如果配置为 'rw,noatime,inode64,allocsize=16m' 则检	否

状态	巡检项	检查内容	是否支持 --set
	数)	查项通过, 否则报 warning。	
	CheckInodeUsage (检查磁盘 inodes 使用率)	GBase 8s 路径 (GAUSSHOME/PGHOST/GPHOME/ GAUSSLOG/tmp 及实例目录) 检查以上指定目录使用率, 如果使用率超过 warning 阈值 (默认为 60%) 报 warning 超过 NG 阈值 (默认为 80%) 则检查项不通过, 否则通过。	否
	CheckSpaceUsage (检查磁盘使用率)	GBase 8s 路径 (GAUSSHOME/PGHOST/GPHOME/ GAUSSLOG/tmp 及实例目录) 检查磁盘以上指定目录 (目录列表) 使用率, 如果使用率超过 warning 阈值 (默认为 70%) 报 warning 超过 NG 阈值 (默认为 90%) 则检查项不通过。GBase 8s 路径下检查 GAUSSHOME/PGHOST/GPHOME/GAUSSLOG/tmp/data 路径的剩余空间, 不满足阈值则检查项不通过, 否则通过。	否
	CheckDiskConfig (检查磁盘空间大小一致性)	检查磁盘名大小挂载点是否一致, 若一致则检查项通过, 否则报 warning。	否

状态	巡检项	检查内容	是否支持 --set
	CheckXid (检查 CheckXid 数值)	查询 xid 的数值, 如果大于 10 亿报 Warning, 大于 18 亿则检查项不通过。	否
	CheckSysTabSize (检查每个实例的系统表容量)	如果每一块磁盘的剩余容量大于该磁盘上所有实例的系统表容量总和则检查项通过, 否则检查项不通过。	否
	CheckClusterState (检查 GBase 8s 状态)	检查 fencedUDF 状态, 如果 fencedUDF 状态为 down 则报 warning; 检查 GBase 8s 状态, 如果 GBase 8s 状态为 Normal 则检查项通过, 否则检查项不通过。	否
cluster	CheckDBParams (检查 GBase 8s 参数)	<p>检查数据库主节点检查共享缓冲区大小和 Sem 参数。</p> <p>数据库节点检查共享缓冲区大小和最大连接数。</p> <p>共享缓冲区需要大于 128KB 且大于 shmmax 且大于 shmall*PAGESIZE</p> <p>若存在数据库主节点, 则 Sem 值需大于 (数据库节点最大连接数+150)/16 向上取整。</p> <p>以上项完全满足则检查项通过, 否则检查项不通过。</p>	是
	CheckDebugSwitch (检查日志级别)	在各节点检查各实例的配置文件中 log_min_messages 参数的值, 为空则认	是

状态	巡检项	检查内容	是否支持 --set
		为是 Warning, 判断日志级别非 waring, 则报 warning。	
	CheckUpVer (检查升级版本是否一致)	检查 GBase 8s 各个节点上升级包的版本, 如果一致则检查项通过, 否则检查项不通过。使用时, 需指定升级软件包路径。	否
	CheckDirPermissions (检查目录权限)	检查节点目录 (实例 Xlog 路径、GAUSSHOME、GPHOME、PGHOST、GAUSSLOG) 权限, 如果目录有写入权限且不大于 750 则检查项通过, 否则检查项不通过。	是
	CheckEnvProfile (检查环境变量)	检查节点环境变量 (\$GAUSSHOME、\$LD_LIBRARY_PATH、\$PATH), 检查 CMS/CMA/数据库节点进程的环境变量。如果环境变量存在并配置正确, 进程的环境变量存在则检查项通过, 否则检查项不通过。	否
	CheckGaussVer (检查进程版本)	检查各个节点进程版本是否一致, 如果版本一致则检查项通过, 否则检查项不通过。	否
	CheckPortRange (检查端口范围)	若 ip_local_port_range 的范围在阈值范围内 (默认是 26000 65535), 并且实	否

状态	巡检项	检查内容	是否支持 --set
		例端口不在 ip_local_port_range 范围内则检查项通过，否则检查项不通过。	
	CheckReadOnlyMode (检查只读模式)	检查 GBase 8s 数据库主节点 default_transaction_read_only 值若为 off 则检查通过，否则不通过。	否
	CheckCatchup (检查 Catchup)	检查 gaussdb 进程堆栈是否能搜索到 CatchupMain 函数，若搜索不到则检查项通过，否则检查项不通过。	否
	CheckProcessStatus (检查 gaussdb 进程属主)	检查'gaussdb 进程属主，若不存在 gbase 以外的属主则检查项通过，否则检查项不通过。	否
	CheckSpecialFile (特殊文件检查)	检查 tmp 目录(PGHOST)、OM 目录(GPHOME)、日志目录(GAUSSLOG)、data 目录、程序目录(GAUSSHOME)下文件是否存在特殊字符以及非 gbase 用户的文件，若不存在则检查项通过，否则检查项不通过。	否
	CheckCollector (检查 GBase 8s 的信息收集)	在 output 目录下查看信息收集是否成功，若收集成功则检查项通过，否则检查项不通过。	否
	CheckLargeFile (检查)	检查各个数据库节点目录是否存在超	否

状态	巡检项	检查内容	是否支持 --set
	查数据目录大文件)	过 4G 的文件。任一数据库节点目录及其子目录有超过 4G 的单个文件，则检查不通过，否则检查通过。	
	CheckProStartTime (关键进程启动时间检测)	检查关键进程启动时间是否间隔超过 5 分钟，超过则检查不通过，否则检查通过。	否
	CheckDilateSysTab (检查系统表膨胀)	检查系统表是否膨胀，若膨胀则不通过，否则检查通过。	是
	CheckMpprcFile(检测环境变量分离文件改动)	检查是否存在对环境变量分离文件的改动，若存在则检查不通过，否则检查通过。	否
database	CheckLockNum(检查锁数量)	检查数据库锁数量，查询成功检查项通过。	否
	CheckArchiveParameter(检查归档参数)	检查数据库归档参数，如果未打开或打开且在数据库节点下则检查项通过，打开且不在数据库主节点目录下则检查项不通过。	是
	CheckCurConnCount(检查当前连接数)	检查数据库连接数，如果连接数小于最大连接数的 90%则检查项通过，否则检查项不通过。	否

状态	巡检项	检查内容	是否支持 --set
	CheckCursorNum(检查当前游标数)	检查数据库的游标数，检查成功则检查项通过，否则检查项不通过。	否
	CheckMaxDatanode (检查 cibase_max_datanode 参数值范围小于数据库节点个数)	检查最大数据库节点数，若最大数据库节点数小于 xml 配置的节点数*数据库节点数（默认值为 90*5）报 warning，否则检查项通过。	是
	CheckPgPreparedXacts(检查残留两阶段事务)	检查 pgxc_prepared_xacts 参数，如果不存在二阶段事务则检查项通过，否则检查项不通过。	是
	CheckPgxcgroup(检查 pgxc_group 表中需要重分布的个数)	检查 pgxc_group 表中需要重分布的个数，检查结果为 0 则通过，否则不通过。	否
	CheckLockState(GBase 8s 是否被锁)	检查 GBase 8s 是否被锁，若 GBase 8s 被锁则不通过，否则检查项通过。	否
	CheckIdleSession(检查业务停止)	检查非空闲会话数，如果数量为 0 则检查项通过，否则检查项不通过。	否
	CheckDBConnection(检查数据库连接)	检查能否连接数据库，如果连接成功则检查项通过，否则检查项不通过。	否

状态	巡检项	检查内容	是否支持 --set
	CheckGUCValue(GUC 参数检查)	检查 (max_connections + max_prepared_transactions) * max_locks_per_transaction 的值, 若该值大于等于 1000000 则检查项通过, 否则检查项不通过。	是
	CheckPMKData(检查 PMK 异常数据)	检查数据库 PMK schema 是否包含有异常数据, 如果不存在异常数据则检查项通过, 否则检查项不通过。	是
	CheckSysTable(检查系统表)	检查系统表, 检查成功则检查项通过。	否
	CheckSysTabSize (检查每个实例的系统表容量)	如果每一块磁盘的剩余容量大于该磁盘上所有实例的系统表容量总和则检查项通过, 否则检查项不通过。	否
	CheckTableSpace(检查表空间路径)	表空间路径和 GBase 8s 路径之间不能存在嵌套且表空间路径相互不能存在嵌套, 则检查项通过, 否则检查项不通过。	否
	CheckTableSkew (检查表级别数据倾斜)	若存在表在 GBase 8s 各数据库节点上的数据分布不均衡, 且分布数据最多的数据库节点比最低的数据库节点所分布的数据多 100000 条以上, 则检查不通过, 否则检查通过。	否

状态	巡检项	检查内容	是否支持 --set
	CheckDNSkew (检查数据库节点级别数据分布倾斜)	检查数据库节点级别的表倾斜数据, 若分布数据最高的数据库节点比分布数据最低的数据库节点数据量高于 5%, 则检查不通过, 否则检查通过。	否
	CheckUnAnalyzeTable (检查未做 analyze 的表)	若存在未做 analyze 的表, 并且表中至少包含一条数据, 则检查不通过, 否则检查通过。	是
	CheckCreateView (创建视图检查)	创建视图时, 如果查询语句中含有子查询, 并且子查询结果查询解析和重写之后存在别名重复, 检查不通过, 否则检查通过。	否
	CheckHashIndex (hash index 语法检查)	如果存在 hash index 则检查不通过, 否则检查通过。	否
	CheckNextvalInDefault (检查 Default 表达式中包含 nextval (sequence))	检查 Default 表达式中是否包含 nextval(sequence), 若包含则不通过, 否则通过。	否
	CheckNodeGroupName (Node group 编码格式检查)	存在非 SQL_ASCII 字符的 Node Group 名称则检查不通过, 不存在则检查通过。	是

状态	巡检项	检查内容	是否支持 --set
	CheckPgxcRedistb (检查重分布残留的临时表)	检查数据库中是否存在重分布残留的临时表, 若不存在则检查通过, 否则检查不通过。	否
	CheckReturnType (用户自定义函数返回值类型检查)	检查用户自定义函数是否包含非法返回类型, 若包含则检查不通过, 否则检查通过。	否
	CheckSysadminUser (检查 sysadmin 用户)	检查除 gbase 属主外是否存在数据库管理员用户, 若存在则不通过, 否则检查通过。	否
	CheckTDDate (TD 数据库中 orc 表 date 类型列检查)	检查 TD 模式数据库下的 orc 表中是否包含 date 类型的列, 若包含检查不通过, 否则检查通过。	否
	CheckDropColumn (drop column 检查)	如果存在 drop column 的表, 则检查不通过, 否则检查通过。	否
	CheckDiskFailure (检查磁盘故障)	对 GBase 8s 中的所有数据做全量查询, 若存在查询错误则检查不通过, 否则检查通过。	否
netwo rk	CheckPing(检查网络通畅)	检查 GBase 8s 内所有节点的互通性, 如果各节点所有 IP 均可 ping 通则检查项	否

状态	巡检项	检查内容	是否支持 --set
		通过，否则检查项不通过。	
	CheckRXTX(检查网卡 RXTX 值)	检查节点 backIP 的 RX/TX 值，如果该值为 4096 则检查项通过，否则检查项不通过。	是
	CheckMTU(检查网卡 MTU 值)	检查节点 backIP 对应的网卡 MTU 值 (bond 后的物理网卡要确保一致)，如果该值不是 8192 或 1500 报 warning 若 openGaussMTU 值一致则检查项通过，否则检查项不通过。	是
	CheckNetWorkDrop(检查网络掉包率)	检查各 IP1 分钟内网络掉包率，如果不超过 1%则检查项通过，否则检查项不通过。	否
	CheckBond(检查网卡绑定模式)	检查是否有配置 BONDING_OPTS 或 BONDING_MODULE_OPTS，若没有配置则报 warning。检查各节点 bond 模式是否一致，如果同时满足则检查项通过，否则检查项不通过。	是
	CheckMultiQueue(检查网卡多队列)	检查 cat /proc/interrupts，判断是否开启网卡多队列且绑定不同 CPU，如果满足则检查项通过，否则检查项不通过。	是
	CheckUsedPort(检查)	检查 net.ipv4.ip_local_port_range，范围	否

状态	巡检项	检查内容	是否支持 --set
	查随机端口使用数量)	大于等于 OS 默认值 通过 (32768-61000) ; 检查 TCP 协议随机端口数,小于总随机端口数的 80%通过。	
	CheckNICModel (网卡型号和驱动版本一致性检查)	检查各个节点的网卡型号以及驱动版本是否一致,一致则通过,否则报 warning。	否
	CheckRouting(本地路由表检查)	检查各节点在业务 IP 网段的 IP 个数,超过 1 个则报 warning,否则检查通过。	否
	CheckNetSpeed(检查网卡接收带宽, ping 值, 丢包率)	网络满载时,检查网卡平均接收带宽大于 600MB 通过; 网络满载时,检查网络 ping 值,小于 1 秒通过; 网络满载时,检查网卡丢包率,小于 1% 通过。	否
other	CheckDataDiskUsage(检查数据库节点磁盘空间使用率)	检查磁盘数据库节点目录使用率,如果使用率低于 90%则检查项通过,否则检查项不通过。	否

 说明

CheckNetSpeed 检查项:

- CheckNetSpeed 不支持-L 本地检查模式, -L 模式无法构造网络压力, 检查的结果

不准确。

- 在节点数小于 6 时，speed_test 构造的网络压力可能无法跑满带宽，可能会造成检查结果不准确。

3.2.4 用户自定义场景

- (1) 以操作系统用户 gbase 登录数据库主节点。
- (2) 在 script/gspylib/inspection/config 路径下新建场景配置文件 scene_XXX.xml。
- (3) 将检查项写进场景配置文件中，书写格式为：

```
<?xml version="1.0" encoding="utf-8" ?>
<scene name="XXX" desc="check cluster parameters before XXX.">
<configuration/>
<allowitems>
<item name="CheckXXX"/>
<item name="CheckXXX"/>
</allowitems>
</scene>
```

其中，item name 为检查项名称。

注意

- 用户需自行保证自定义 xml 的正确性

- (4) 在 home/package/script/gspylib/inspection/config 执行如下命令，将此文件分发至执行检查的各个节点

```
scp scene_upgrade.xml SIA1000068994:
home/package/script/gspylib/inspection/config/
```

说明

- home/package/script/gspylib/inspection/config 就是新建的场景配置文件的绝对路径。

- (5) 以 gbase 用户执行以下命令，查看检查结果。

```
gs_check -e XXX
```

3.2.5 用户自定义检查项

(1) 新增巡检项配置，修改 `script/gspylib/inspection/config/items.xml` 文件，格式如下：

```
<checkitem id="10010" name="CheckCPU">
<title>
<zh>检查 CPU 占用率</zh>
<en>Check CPU Idle and I/O wait</en>
</title>
<threshold>
StandardCPUIdle=30;
StandardWIO=30
</threshold>
<suggestion>
<zh>如果 idle 不足 CPU 负载过高，请扩容 CPU 节点，如果 iowait 过高，则磁盘为瓶颈，
更换高性能磁盘</zh>
</suggestion>
<standard>
<zh>检查主机 CPU 占用率，如果 idle 大于 30%并且 iowait 小于 30%，则检查项通过，否
则检查项不通过</zh>
</standard>
<category>os</category>
<permission>user</permission>
<scope>all</scope>
<analysis>default</analysis>
</checkitems>
```

- `id`: 巡检项 id。
- `name`: 巡检项脚本名，和巡检项脚本文件名相同。
- `title`: 巡检项描述名称（支持多语言）。
- `zh`: 中文版检查内容。
- `en`: 英文版检查内容。
- `standard`: 巡检项标准说明（支持多语言）。
- `suggestion`: 巡检项修复建议说明（支持多语言）。
- `threshold`: 巡检项阈值定义，多值之间使用分号隔开，示例 `Key1=Value1;Key2=Value2`。

- **category**: 巡检项分类, 可选参数: os, device, network, cluster, database, other。
- **permission**: 巡检项需要的执行权限, 可选参数: root, user 默认为 user (普通用户)。
- **scope**: 巡检项执行的节点范围, 可选参数: cn-仅在数据库主节点执行, local-仅在当前节点执行, all-在 GBase 8s 所有节点执行, 默认为 all。
- **analysis**: 巡检项执行结果分析方式, default-检查每个节点的结果, 所有节点检查项通过, 则最终检查通过, consistent-gbase 内所有节点一致性检查, 单节点仅返回结果, 各个节点结果一致则判定检查通过, custom-自定义结果分析方式, 默认为 default。

⚠ 注意

- 用户需自行保证自定义 xml 的正确性

- (2) 新建检查脚本, 脚本名称格式遵循 CheckXXXX.py, 必须以 Check 开头, 脚本放置在 script/gspylib/inspection/items 目录下, 该目录下脚本安装巡检项分类组织, 每个分类一个单独的文件夹, 巡检项脚本放置在对应的分类文件夹中。格式如下:

```
class CheckCPU(BaseItem):
    def __init__(self):
        super(CheckCPU, self).__init__(self.__class__.__name__)
        self.idle = None
        self.wio = None
        self.standard = None

    def preCheck(self):
        # check the threshold was set correctly
        if (not self.threshold.has_key('StandardCPUIdle')
            or not self.threshold.has_key('StandardWIO')):
            raise Exception("threshold can not be empty")
        self.idle = self.threshold['StandardCPUIdle']
        self.wio = self.threshold['StandardWIO']

        # format the standard by threshold
        self.standard = self.standard.format(idle=self.idle, iowait=self.wio)

    def doCheck(self):
        cmd = "sar 1 5 2>&1"
        output = SharedFuncs.runShellCmd(cmd)
```

```
self.result.raw = output
# check the result with threshold
d = next(n.split() for n in output.splitlines() if "Average" in n)
iowait = d[-3]
idle = d[-1]
rst = ResultStatus.OK
vals = []
if (iowait > self.wio):
rst = ResultStatus.NG
vals.append("The %s actual value %s is greater than expected value %s" % ("IOWait",
iowait, self.wio))
if (idle < self.idle):
rst = ResultStatus.NG
vals.append("The %s actual value %s is less than expected value %s" % ("Idle",
idle, self.idle))
self.result.rst = rst
if (vals):
self.result.val = "\n".join(vals)
```

所有脚本基于 BaseItem 基类开发，基类定义的通用的检查流程，通用的检查结果分析方法，默认的结果输出格式。可扩展方法：

- doCheck: 该方法包含该检查项具体的检查方法，检查结果格式如下：

result.rst — 检查结果状态，可选参数：

- OK -- 检查项完成，结果通过。
 - NA -- 当前节点不涉及该检查项。
 - NG -- 检查项完成，结果不通过。
 - WARNING -- 检查项完成，结果警告。
 - ERROR -- 检查项发生内部错误，未完成检查。
- preCheck: 检查前条件判定，内置两种实现：
 - cnPreCheck -- 用于检查当前执行节点是否包含数据库主节点实例，
 - localPreCheck -- 用于检查当前执行节点是否指定节点。可通过巡检项配置文件中的 scope 参数进行配置。可重载该方法实现自定义的前置检查

- `postAnalysis`: 检查结果分析方法, 内置两种实现: `default`、`consistent`。可通过巡检项配置文件中的 `analysis` 参数进行配置。可重载该方法实现自定义的结果分析。

⚠ 注意

- 用户自定义的检查项名称不得与已有检查项名称相同, 同时用户需保证自定义检查项脚本的规范性。

(3) 将此脚本分发至所有的执行节点。

(4) 以 `gbase` 用户登录, 执行以下命令, 查看结果。

本地执行:

```
gs_check -i CheckXXX -L
```

非本地执行:

```
gs_check -i CheckXXX
```

3.2.6 操作系统参数

表 3-4 操作系统参数

参数名称	参数说明	推荐取值
<code>net.ipv4.tcp_max_tw_buckets</code>	表示同时保持 <code>TIME_WAIT</code> 状态的 TCP/IP 连接最大数量。如果超过所配置的取值, <code>TIME_WAIT</code> 将立刻被释放并打印警告信息。	10000
<code>net.ipv4.tcp_tw_reuse</code>	允许将 <code>TIME-WAIT</code> 状态的 sockets 重新用于新的 TCP 连接。 ● 0 表示关闭。	1

参数名称	参数说明	推荐取值
	<ul style="list-style-type: none"> ● 1 表示开启。 	
net.ipv4.tcp_tw_recycle	<p>表示开启 TCP 连接中 TIME-WAIT 状态 sockets 的快速回收。</p> <ul style="list-style-type: none"> ● 0 表示关闭。 ● 1 表示开启。 	1
net.ipv4.tcp_keepalive_time	<p>表示当 keepalive 启用的时候, TCP 发送 keepalive 消息的频度。</p>	30
net.ipv4.tcp_keepalive_probes	<p>在认定连接失效之前, 发送 TCP 的 keepalive 探测包数量。这个值乘以 tcp_keepalive_intvl, 决定了一个连接发送了 keepalive 之后可以有多少时间没有回应。</p>	9
net.ipv4.tcp_keepalive_intvl	<p>当探测没有确认时, 重新发送探测的频度。</p>	30
net.ipv4.tcp_retries1	<p>在连接建立过程中 TCP 协议最大重试次数。</p>	5
net.ipv4.tcp_syn_retries	<p>TCP 协议 SYN 报文最大重试次数。</p>	5

参数名称	参数说明	推荐取值
net.ipv4.tcp_synack_retries	TCP 协议 SYN 应答报文最大重试次数。	5
net.ipv4.tcp_retries2	<p>控制内核向已经建立连接的远程主机重新发送数据的次数，低值可以更早的检测到与远程主机失效的连接，因此服务器可以更快的释放该连接。</p> <p>发生 connection reset by peer 时可以尝试调大该值规避问题。</p>	12
vm.overcbaseit_memory	<p>控制在做内存分配的时候，内核的检查方式。</p> <p>0: 表示系统会尽量精确计算当前可用的内存。</p> <p>1: 表示不作检查直接返回成功。</p> <p>2 : 内存总量 ×vm.overcbaseit_ratio/10 + SWAP 的总量，如果申请空间超过此数值则返回失败。</p> <p>内核默认是 2 过于保守，推荐设置为 0，如果系统压力大可以设置为 1。</p>	0

参数名称	参数说明	推荐取值
net.ipv4.tcp_rmem	TCP 协议接收端缓冲区的可用内存大小。分无压力、有压力、和压力大三个区间，单位为页面。	8192 250000 16777216
net.ipv4.tcp_wmem	TCP 协议发送端缓冲区的可用内存大小。分无压力、有压力、和压力大三个区间，单位为页面。	8192 250000 16777216
net.core.wmem_max	socket 发送端缓冲区大小的最大值。	21299200
net.core.rmem_max	socket 接收端缓冲区大小的最大值。	21299200
net.core.wmem_default	socket 发送端缓冲区大小的默认值。	21299200
net.core.rmem_default	socket 接收端缓冲区大小的默认值。	21299200
net.ipv4.ip_local_port_range	物理机可用临时端口范围。	26000-65535
kernel.sem	内核信号量参数设置大小。	250 6400000 1000 25600

参数名称	参数说明	推荐取值
vm.min_free_kbytes	保证物理内存有足够空闲空间，防止突发性换页。	系统总内存的 5%
net.core.somaxconn	定义了系统中每一个端口最大的监听队列的长度，这是个全局的参数。	65535
net.ipv4.tcp_syncookies	<p>当出现 SYN 等待队列溢出时，启用 cookies 来处理，可防范少量 SYN 攻击。</p> <ul style="list-style-type: none"> ● 0 表示关闭 SYN Cookies。 ● 1 表示开启 SYN Cookies。 	1
net.core.netdev_max_backlog	在每个网络接口接收数据包的速率比内核处理这些包的速率快时，允许送到队列的数据包的最大数目。	65535
net.ipv4.tcp_max_syn_backlog	记录的那些尚未收到客户端确认信息的连接请求的最大值。	65535
net.ipv4.tcp_fin_timeout	系统默认的超时时间。	60
kernel.shmall	内核可用的共享内存总	1152921504606846720

参数名称	参数说明	推荐取值
	量。	
kernel.shmmax	内核参数定义单个共享内存段的最大值。	1844674407370955161 5
net.ipv4.tcp_sack	<p>启用有选择的应答，通过有选择地应答乱序接受到的报文来提高性能，让发送者只发送丢失的报文段（对于广域网来说）这个选项应该启用，但是会增加对 CPU 的占用。</p> <ul style="list-style-type: none"> ● 0 表示关闭。 ● 1 表示开启。 	1
net.ipv4.tcp_timestamps	<p>TCP 时间戳（会在 TCP 包头增加 12 节），以一种比重发超时更精确的方式（参考 RFC 1323）来启用对 RTT 的计算，启用可以实现更好的性能。</p> <ul style="list-style-type: none"> ● 0 表示关闭。 ● 1 表示开启。 	1
vm.extfrag_threshold	系统内存不够用时，linux 会为当前系统内存碎片情况打分，如果超过 vm.extfrag_threshold 的值，kswapd 就会触发	500

参数名称	参数说明	推荐取值
	memory compaction。所以这个值设置的接近 1000，说明系统在内存碎片的处理倾向于把旧的页换出，以符合申请的需要，而设置接近 0，表示系统在内存碎片的处理倾向做 memory compaction。	
vm.overcbaseit_ratio	系统使用绝不过量使用内存的算法时，系统整个内存地址空间不得超过 swap+RAM 值的此参数百分比，当 vm.overcbaseit_memory=2 时此参数生效。	90
MTU	节点网卡最大传输单元。OS 默认值为 1500，调整为 8192 可以提升 SCTP 协议数据收发的性能。	8192

3.2.7 文件系统参数

- soft nofile

soft nofile 表示软限制，用户使用的文件句柄数量可以超过该限制，但是如果超过会有告警信息。

推荐取值：1000000

- hard nofile

hard nofile 表示硬限制，是一个严格的限制，用户使用的文件句柄数量一定不能超过该设置。

推荐取值：1000000

- stack size

线程堆栈大小。

推荐值：3072

3.2.8 示例

执行单项检查结果：

```
$ gs_check -i CheckCPU
Parsing the check items config file successfully
Distribute the context file to remote hosts successfully
Start to health check for the cluster. Total Items:1 Nodes:1

Checking... [=====] 1/1
Start to analysis the check result
CheckCPU.....OK
The item run on 1 nodes. success: 1

Analysis the check result successfully
Success. All check items run completed. Total:1 Success:1
For more information please refer to
/home/gbase/gbase_db/om_f5c5a0af/script/gspylib/inspection/output/CheckReport_202212236422332158.tar.gz
```

本地执行结果：

```
$ gs_check -i CheckCPU -L
2022-12-23 17:51:42 [NAM] CheckCPU
2022-12-23 17:51:42 [STD] 检查主机 CPU 占用率，如果 idle 大于 30%，或者 iowait 小于 30%。则检查项通过，否则检查项不通过
2022-12-23 17:51:42 [RST] OK

2022-12-23 17:51:42 [RAW]
Linux 3.10.0-1127.el7.x86_64 (gbase8s_5_124) 12/23/22 _x86_64_
(2 CPU)
```

```

17:51:37
CPU      %user   %nice   %system %iowait  %steal   %idle
17:51:38      all     7.58    0.00    2.53    0.00    0.00
89.90
17:51:39      all     5.56    0.00    1.52    0.00    0.00
92.93
17:51:40      all    10.15    0.00    3.05    0.00    0.00
86.80
17:51:41      all     8.12    0.00    2.54    0.00    0.00
89.34
17:51:42      all     7.07    0.00    2.02    0.00    0.00
90.91
Average:      all     7.69    0.00    2.33    0.00    0.00
89.98
    
```

执行场景检查结果:

```

$ gs_check -e inspect
Skip CheckHdfsForeignTabEncoding because it only applies to V1R5 upgrade V1R6 with
cluster.
Parsing the check items config file successfully
The below items require root privileges to execute:[CheckBlockdev
CheckIOConfigure CheckMTU CheckRXTX CheckMultiQueue CheckFirewall
CheckSshdService CheckSshdConfig CheckCronService CheckMaxProcMemory
CheckBootItems CheckFilehandle CheckNICModel CheckDropCache]
Please enter root privileges user[root]:
Please enter password for user[root]:
Check root password connection successfully
Distribute the context file to remote hosts successfully
Start to health check for the cluster. Total Items:57 Nodes:3
Checking...          [=====] 57/57
Start to analysis the check result
CheckClusterState.....OK
The item run on 3 nodes.  success: 3
CheckDBParams.....OK
.....
CheckMpprcFile.....OK
The item run on 3 nodes.  success: 3

Analysis the check result successfully
Failed. All check items run completed. Total:57  Success:49  Warning:5  NG:3
Error:0
    
```

```
For more information please refer to  
/home/gbase/gbase_db/om_f5c5a0af/script/gspylib/inspection/output/CheckReport  
_inspect_20221223643622205.tar.gz
```

3.2.9 相关命令

`gs_checkos`, `gs_checkperf`

3.3 `gs_checkos`

3.3.1 背景信息

`gs_checkos` 工具用来帮助检查操作系统、控制参数、磁盘配置等内容，并对系统控制参数、I/O 配置、网络配置和 THP 服务等信息进行配置。

前提条件

- 当前的硬件和网络环境正常。
- 各主机间 `root` 互信状态正常。
- 只能使用 `root` 用户执行 `gs_checkos` 命令。

说明

该工具不支持独立调用，出于安全考虑，企业版安装方式下，前置完成后会自动删除。

3.3.2 语法

- 检查操作系统信息

```
gs_checkos -i ITEM [-f HOSTFILE] [-h HOSTNAME] [-X XMLFILE] [--detail] [-o OUTPUT]  
[-l LOGFILE]
```

- 显示帮助信息

```
gs_checkos -? | --help
```

- 显示版本号信息

```
gs_checkos -V | --version
```

3.3.3 参数说明

- -i

列表编号。格式：-i A、-i B1、-i A1 -i A2 或-i A1,A2。

取值范围：A1...A14、B1...B8。

 **说明**

➤ A1...A14 表示只检查操作系统参数，并不设置。

➤ B1...B8 表示将参数系统参数设置为期望值。

➤ A 和 B 不能同时输入。

详细的检查项请参见 3-5。

- -f

主机名称列表文件。

取值范围：主机名称的列表。

 **说明**

➤ -f 和-h 参数不能同时使用。

- -h

指定需要检查的主机名称，可以同时指定多个主机，主机之间使用“;”分割。

取值范围：GBase 8s 的主机名称。如果不指定主机，则检查当前主机。

- -X

配置文件。

- --detail

显示检查结果详情。

- -o

指定 OS 检查报告输出到指定的文件。

不指定则将检查结果输出到屏幕上。

- -l
指定日志文件及存放路径。
默认值： /tmp/gbase_checkos/gbase_checkos-YYYY-MM-DD_hhmmss.log。
- -?, --help
显示帮助信息。
- -V, --version
显示版本号信息。

表 3-5 操作系统检查项

编号	检查项	检查项详细说明
A1	操作系统版本 (OS version status)	操作系统版本检查：保证主机操作系统是 GBase 8s 支持兼容的。同时保证 GBase 8s 各个主机的平台是属于同一混编范围的。
A2	内核版本 (Kernel version status)	内核版本检查。
A3	Unicode 状态 (Unicode status)	字符集设置检查：保证 GBase 8s 各个主机的字符集一致。
A4	时区状态 (Time zone status)	时区时间设置检查：保证 GBase 8s 各个主机的时区状态一致。
A5	交换内存状态 (Swap memory status)	Swap 分区检查：各个主机的 swap 分区大小必须小于等于各主机内存总大小。
A6	系统控制参数 (System	内核参数检查：保证当前内核中的参数值和默认

编号	检查项	检查项详细说明
	control parameters status)	的期望值保持一致。 详细的内核参数请参见表 3-6。
A7	文件系统配置状态 (File system configuration status)	系统文件句柄检查：系统文件句柄的参数值必须大于等于 100 万。 线程堆栈大小检查：线程堆栈大小大于等于 3072KB。 进程最大可用虚拟内存：系统进程最大可用虚拟内存为 unlimited。
A8	磁盘配置状态 (Disk configuration status)	硬盘挂载参数检查：格式为 xfs 的硬盘，其挂载格式必须为：“rw,noatime,inode64,allocsize=16m”。
A9	预读块大小状态 (Pre-read block size status)	预读参数检查：预读参数期望值为 16384。
A10	IO 调度状态 (IO scheduler status)	IO 调度策略检查：IO 调度策略的方式为 deadline。
A11	网卡配置状态 (Network card configuration status)	万兆以下网卡参数检查： 确保网卡 mtu=1500。 万兆及以上网卡参数检查： 当网卡为非绑定模式，同时网卡类型为万兆时，确保 mtu=1500，rx/tx >=4096。 当网卡为绑定模式时，需保证绑定的每个网卡类

编号	检查项	检查项详细说明
		<p>型均为万兆, 然后再确保 mtu=1500, rx/tx>=4096。</p> <p>GBase 8s 周边设备 (例如交换机等) 的网卡也要设置成与 GBase 8s 部署的主机一致的值。</p>
A12	时间一致性 (Time consistency status)	时间一致性检查: 确认 ntp 服务启动, GBase 8s 各个主机的系统时间误差不超过 60s。
A13	防火墙状态 (Firewall service status)	防火墙检查: 确认防火墙的状态为关闭状态。
A14	THP 服务 (THP service status)	THP 服务检查: 确认 THP 服务状态为关闭状态。
B1	设置系统控制参数 (Set system control parameters)	内核参数设置: 当实际值不满足检查条件期望时, 对结果为 Abnormal 项参数进行修改设置; 对 Warning 项不进行设置, 由用户根据实际环境设置。
B2	设置文件系统配置值 (Set file system configuration value)	<p>系统文件句柄设置: 当系统文件句柄的参数值小于 100 万时, 对其进行修改设置。</p> <p>线程堆栈大小设置: 线程堆栈大小小于 3072KB 时, 对其进行设置。</p> <p>进程最大可用虚拟内存设置: 系统进程最大可用虚拟内存不为 unlimited 时对其进行修改设置。</p>
B3	设置预读块大小值 (Set pre-read block size)	硬盘预读参数设置: 当系统的实际值小于 16384 时, 对其进行修改设置。

编号	检查项	检查项详细说明
	value)	
B4	设置 IO 调度值 (Set IO scheduler value)	IO 配置项的设置：当系统的实际值不为 deadline 时，对其进行修改设置。
B5	设置网卡配置值 (Set network card configuration value)	万兆网卡参数设置：对不满足万兆网卡的 RX、TX 参数进行设置，不对 MTU 参数进行设置。
B6	设置 THP 服务 (Set THP service)	THP 服务设置：如果 THP 服务启动，将该服务关闭。
B7	设置系统属性 (Set RemoveIPC value)	系统属性检查：检查设置系统文件 /usr/lib/systemd/system/systemd-logind.service ， /etc/systemd/logind.conf 中配置项 RemoveIPC 属性值是否为 no，当不为 no 时，对其进行修改设置。
B8	设置 sshd 服务应用 pam 模块 (Set Session Process)	远程设备继承系统默认资源：修改/etc/pam.d/sshd 服务文件，添加配置项 session required pam_limits.so，来控制用户使用的资源。

 说明

- A6 选项检查标准来自配置文件 check_list.conf 下 [/etc/sysctl.conf]、[SUGGEST:/etc/sysctl.conf]域：若[/etc/sysctl.conf]下参数值与系统实际参数值不同，A6 检查提示 Abnormal，可以使用 B1 参数进行设置。若[SUGGEST:/etc/sysctl.conf]下参数值与系统实际参数值不同，A6 检查提示 Warning，B1 参数不会进行设置，需根据实际情况进行手动设置。

- A7 选项检查系统文件句柄标准来自配置文件 `check_list.conf` 下 `[/etc/security/limits.conf]`域 `open file` 参数，可以使用 B2 参数进行设置。
- A11 选项检查万兆网卡参数“MTU”、“RX”、“TX”标准来自配置文件 `check_list.conf` 下 `[/sbin/ifconfig]`域，可以使用 B5 参数对 RX、TX 进行设置，MTU 需用户手动设置。
- 使用 `gs_checkos` 设置的内核参数值和文件句柄参数值，需重启新会话生效。

3.3.4 操作系统参数

表 3-6 操作系统参数

参数名称	参数说明	推荐取值
<code>net.ipv4.tcp_max_tw_buckets</code>	表示同时保持 TIME_WAIT 状态的 TCP/IP 连接最大数量。如果超过所配置的取值，TIME_WAIT 将立刻被释放并打印警告信息。	10000
<code>net.ipv4.tcp_tw_reuse</code>	允许将 TIME-WAIT 状态的 sockets 重新用于新的 TCP 连接。 ● 0 表示关闭。 ● 1 表示开启。	1
<code>net.ipv4.tcp_tw_recycle</code>	表示开启 TCP 连接中 TIME-WAIT 状态 sockets 的快速回收。 ● 0 表示关闭。 ● 1 表示开启。	1

参数名称	参数说明	推荐取值
net.ipv4.tcp_keepalive_time	表示当 keepalive 启用的时候，TCP 发送 keepalive 消息的频度。	30
net.ipv4.tcp_keepalive_probes	在认定连接失效之前，发送 TCP 的 keepalive 探测包数量。这个值乘以 tcp_keepalive_intvl 之后决定了一个连接发送了 keepalive 之后可以有多少时间没有回应。	9
net.ipv4.tcp_keepalive_intvl	当探测没有确认时，重新发送探测的频度。	30
net.ipv4.tcp_retries1	在连接建立过程中 TCP 协议最大重试次数。	5
net.ipv4.tcp_syn_retries	TCP 协议 SYN 报文最大重试次数。	5
net.ipv4.tcp_synack_retries	TCP 协议 SYN 应答报文最大重试次数。	5
net.ipv4.tcp_retries2	控制内核向已经建立连接的远程主机重新发送数据的次数，低值可以更早的检测到与远程主机失效的连接，因此服务器可以更	12

参数名称	参数说明	推荐取值
	<p>快释放该连接。</p> <p>发生“connection reset by peer”时可以尝试调大该值规避问题。</p>	
vm.overcbaseit_memory	<p>控制在做内存分配的时候，内核的检查方式。</p> <ul style="list-style-type: none"> ● 0：表示系统会尽量精确计算当前可用的内存。 ● 1：表示不作检查直接返回成功。 ● 2：内存总量 × vm.overcbaseit_ratio / 100 + SWAP 的总量，如果申请空间超过此数值则返回失败。 <p>内核默认是 2 过于保守，推荐设置为 0，如果系统压力大可以设置为 1。</p>	0
net.ipv4.tcp_rmem	<p>TCP 协议接收端缓冲区的可用内存大小。分无压力、有压力、和压力大三个区间，单位为页面。</p>	8192 250000 16777216
net.ipv4.tcp_wmem	<p>TCP 协议发送端缓冲区的可用内存大小。分无压力、有压力、和压力大三个区间，单位为页面。</p>	8192 250000 16777216

参数名称	参数说明	推荐取值
net.core.wmem_max	socket 发送端缓冲区大小的最大值。	21299200
net.core.rmem_max	socket 接收端缓冲区大小的最大值。	21299200
net.core.wmem_default	socket 发送端缓冲区大小的默认值。	21299200
net.core.rmem_default	socket 接收端缓冲区大小的默认值。	21299200
net.ipv4.ip_local_port_range	物理机可用临时端口范围。	26000-65535
kernel.sem	内核信号量参数设置大小。	250 6400000 1000 25600
vm.min_free_kbytes	保证物理内存有足够空闲空间，防止突发性换页。	系统总内存的 5%
net.core.somaxconn	定义了系统中每一个端口最大的监听队列的长度，这是个全局的参数。	65535
net.ipv4.tcp_syncookies	当出现 SYN 等待队列溢出时，启用 cookies 来处理，可防范少量 SYN 攻击。	1

参数名称	参数说明	推荐取值
	<ul style="list-style-type: none"> ● 0 表示关闭 SYN Cookies。 ● 1 表示开启 SYN Cookies。 	
net.core.netdev_max_backlog	在每个网络接口接收数据包的速率比内核处理这些包的速率快时，允许送到队列的数据包的最大数目。	65535
net.ipv4.tcp_max_syn_backlog	记录的那些尚未收到客户端确认信息的连接请求的最大值。	65535
net.ipv4.tcp_fin_timeout	系统默认的超时时间。	60
kernel.shmall	内核可用的共享内存总量。	1152921504606846720
kernel.shmmax	内核参数定义单个共享内存段的最大值。	18446744073709551615
net.ipv4.tcp_sack	启用有选择的应答，通过有选择地应答乱序接受到的报文来提高性能，让发送者只发送丢失的报文段（对于广域网来说）这个选项应该启用，但是会增	1

参数名称	参数说明	推荐取值
	<p>加对 CPU 的占用。</p> <ul style="list-style-type: none"> ● 0 表示关闭。 ● 1 表示开启。 	
net.ipv4.tcp_timestamps	<p>TCP 时间戳（会在 TCP 包头增加 12 节），以一种比重发超时更精确的方式（参考 RFC 1323）来启用对 RTT 的计算，启用可以实现更好的性能。</p> <ul style="list-style-type: none"> ● 0 表示关闭。 ● 1 表示开启。 	1
vm.extfrag_threshold	<p>系统内存不够用时，linux 会为当前系统内存碎片情况打分，如果超过 vm.extfrag_threshold 的值，kswapd 就会触发 memory compaction。所以这个值设置的接近 1000，说明系统在内存碎片的处理倾向于把旧的页换出，以符合申请的需要，而设置接近 0，表示系统在内存碎片的处理倾向做 memory compaction。</p>	500
vm.overcbaseit_ratio	<p>系统使用绝不过量使用内存的算法时，系统整个内</p>	90

参数名称	参数说明	推荐取值
	存地址空间不得超过 swap+RAM 值的此参数百分比，当 vm.overcbaseit_memory=2 时此参数生效。	
MTU	节点网卡最大传输单元。 OS 默认值为 1500, 调整为 8192 可以提升 SCTP 协议数据收发的性能。	8192

3.3.5 文件系统参数

- soft nofile

soft nofile 表示软限制，用户使用的文件句柄数量可以超过该限制，但是如果超过会有告警信息。

推荐取值：1000000

- hard nofile

hard nofile 表示硬限制，是一个严格的限制，用户使用的文件句柄数量一定不能超过该设置。

推荐取值：1000000

- stack size

线程堆栈大小。

推荐值：3072

3.3.6 示例

使用如下命令检查操作系统参数。

```
[root@gbase ~]# gs_checkos -i A -h gbase --detail -o /var/log/checkos
Performing operation system check/set. Output the result to the file
/var/log/checkos.
Operation system check/set is completed.
Total numbers:14. Abnormal numbers:0. Warning numbers:3.
```

查看操作系统参数检查结果。

```
[root@gbase ~]# vim /var/log/checkos
Checking items:
  A1. [ OS version status ] : Normal
      [gbase]
      centos_7.8.2003_64bit
  A2. [ Kernel version status ] : Normal
      The names about all kernel versions are same. The value is
"3.10.0-1127.el7.x86_64".
  A3. [ Unicode status ] : Normal
      The values of all unicode are same. The value is "LANG=en_US.UTF-8".
  A4. [ Time zone status ] : Normal
      The informations about all timezones are same. The value is "+0800".
  A5. [ Swap memory status ] : Warning
      [gbase]
SwapMemory 8455712768 TotalMemory 8201502720
  A6. [ System control parameters status ] : Normal
      [gbase]
      Check_SysCtl_Parameter normal.
  A7. [ File system configuration status ] : Normal
      Both soft nofile and hard nofile are correct.
  A8. [ Disk configuration status ] : Normal
      The value about XFS mount parameters is correct.
  A9. [ Pre-read block size status ] : Normal
      The value about Logical block size is correct.
  A10. [ IO scheduler status ] : Normal
      The value of IO scheduler is correct.
  A11. [ Network card configuration status ] : Warning
      [gbase]
      BondMode Null
  A12. [ Time consistency status ] : Warning
      [gbase]
      The NTPD not detected on machine and local time is "2022-12-26 09:45:45".
  A13. [ Firewall service status ] : Normal
      The firewall service is stopped.
```

A14. [THP service status]

: Normal

The THP service is stopped.

3.3.7 相关命令

[gs_check](#), [gs_checkperf](#)

3.4 gs_checkperf

3.4.1 背景信息

GBase 8s 提供了 `gs_checkperf` 工具来帮助对 GBase 8s 级别、节点级别和会话/进程级别的 CPU 使用情况、内存使用情况、I/O 使用情况、以及 SSD 性能（包含写入和读取性能）进行定期检查，让用户了解 GBase 8s 的负载情况，采取对应的改进措施。

前提条件

- GBase 8s 运行状态正常且不为只读模式。
- 运行在数据库之上的业务运行正常。

注意事项

`gs_checkperf` 工具的监控信息依赖于 `pmk` 模式下的表的数据。如果 `pmk` 模式下的表未执行 `analyze` 操作，则可能导致 `gs_checkperf` 工具执行失败。其报错示例信息如下。

```
LOG: Statistics in some tables or columns (pmk.pmk_snapshot.snapshot_id) are not collected.
```

```
HINT: Do analyze for them in order to generate optimized plan.
```

此时需要登录任一数据库主节点，连接 `postgres` 数据库执行如下 SQL。

```
analyze pmk.pmk_configuration;  
analyze pmk.pmk_meta_data;  
analyze pmk.pmk_snapshot;  
analyze pmk.pmk_snapshot_dbnode_stat;  
analyze pmk.pmk_snapshot_datanode_stat;
```

3.4.2 语法

- 检查 SSD 性能（以 `root` 用户）

```
gs_checkperf -U USER [-o OUTPUT] -i SSD [-l LOGFILE]
```

- 检查 GBase 8s 性能 (以 gbase 用户)

```
gs_checkperf [-U USER] [-o OUTPUT] [-i PMK] [--detail] [-l LOGFILE]
```

- 显示帮助信息

```
gs_checkperf -? | --help
```

- 显示版本号信息

```
gs_checkperf -V | --version
```

3.4.3 参数说明

- -U

指定运行 GBase 8s 的用户名称。以 root 用户身份执行命令时，必须指定该参数。

取值范围：运行 GBase 8s 的用户名称。

- -o

指定性能检查报告输出到指定的文件。不指定则将检查结果输出到屏幕上。

取值范围：指定的文件名称。

- -i

指定检查项编号，-i 参数值不区分大小写。格式：-i PMK、-i SSD。

取值范围：PMK、SSD。

说明

- 只有 GBase 8s 用户才能检查 PMK 选项。
- 只有 root 用户才能检查 SSD 选项。
- 如果不指定该参数，以 GBase 8s 用户身份默认检查 PMK，以 root 用户身份默认检查 SSD。

- --detail

显示 PMK 检查结果详情。

- -l

指定日志文件的存储路径。

默认路径为：/var/log/om/gs_checkperf-YYYY-MM-DD_hhmmss.log

- -?, --help

显示帮助信息。

- -V, --version

显示版本号信息。

表 3-7 性能检查项

分类	性能参数项	描述
数据库级别	CPU 占用率	CPU 占用率。
	共享内存命中率	共享内存的命中率。
	内存中排序比率	内存中完成的排序所占比率。
	I/O 使用情况	文件读写次数和时间。
	磁盘使用情况	文件写次数和平均写时间、最大写时间等。
	事务统计	当前 SQL 执行数和 Session 数。
节点级别	CPU 使用情况	主机使用 CPU 情况，包括 cpu busy time、cpu idle time 等。
	内存使用情况	主机使用内存情况，包括物理内存总量、已使用量等。
	I/O 使用情况	文件读写次数和时间。

分类	性能参数项	描述
会话/进程 级别	CPU 使用情况	会话使用 CPU 情况，包括 cpu busy time、cpu idle time 等。
	内存使用情况	会话使用内存情况，包括物理内存总量、已使用量等。
	I/O 使用情况	会话共享缓冲区命中次数等。
SSD 性能 (只用 root 用户才能 查看)	写入性能	使用 dd 命令 (flag=direct bs=8M count=2560) 向每个 SSD 写入内容，写入每个 SSD 时间应在 10s 左右。
	读取性能	使用 dd 命令 (flag=direct bs=8M count=2560) 从每个 SSD 读取内容，读取每个 SSD 时间应在 7s 左右。

3.4.4 示例

示例一：以简要格式在屏幕上显示性能统计结果。

```
gs_checkperf -i pmk -U gbase
Cluster statistics information:
Host CPU busy time ratio           : 1.43      %
MPPDB CPU time % in busy time     : 1.88      %
Shared Buffer Hit ratio             : 99.96     %
In-memory sort ratio              : 100.00    %
Physical Reads                     : 4
Physical Writes                    : 25
DB size                            : 70        MB
Total Physical writes              : 25
Active SQL count                   : 2
Session count                      : 3
```


示例二：以详细格式在屏幕上显示性能统计结果。

```

gs_checkperf -i pmk -U gbase --detail
Cluster statistics information:
Host CPU usage rate:
  Host total CPU time           : 42386.667 Jiffies
  Host CPU busy time           : 1873.333 Jiffies
  Host CPU iowait time         : 20.000 Jiffies
  Host CPU busy time ratio     : 4.42 %
  Host CPU iowait time ratio   : .05 %
MPPDB CPU usage rate:
  MPPDB CPU time % in busy time : 19.04 %
  MPPDB CPU time % in total time : .84 %
Shared buffer hit rate:
  Shared Buffer Reads           : 0
  Shared Buffer Hits            : 24129
  Shared Buffer Hit ratio       : 100.00 %
  In-memory sort ratio        : 100.00 %
I/O usage:
  Number of files              : 537
  Physical Reads               : 0
  Physical Writes              : 0
  Read Time                    : 0 ms
  Write Time                   : 0 ms
Disk usage:
  DB size                      : 70 MB
  Total Physical writes        : 0
  Average Physical write      : 0
  Maximum Physical write       : 0
Activity statistics:
  Active SQL count             : 2
  Session count                : 3
Node statistics information:
dn_6001_6002:
  MPPDB CPU Time               : 200 Jiffies
  Host CPU Busy Time           : 3490 Jiffies
  Host CPU Total Time          : 42330 Jiffies
  MPPDB CPU Time % in Busy Time : 5.73 %
  MPPDB CPU Time % in Total Time : .47 %
  Physical memory              : 8231776256 Bytes
  DB Memory usage              : 877236224 Bytes
  Shared buffer size           : 33554432 Bytes

```

Shared buffer hit ratio	:	100.00	%
Sorts in memory	:	123	
Sorts in disk	:	0	
In-memory sort ratio	:	100.00	%
Number of files	:	149	
Physical Reads	:	0	
Physical Writes	:	0	
Read Time	:	0	
Write Time	:	0	
dn_6003_6004:			
MPPDB CPU Time	:	170	
Jiffies Host CPU Busy Time	:	1030	Jiffies
Host CPU Total Time	:	42470	Jiffies
MPPDB CPU Time % in Busy Time	:	16.50	%
MPPDB CPU Time % in Total Time	:	.40	%
Physical memory	:	8231776256	Bytes
DB Memory usage	:	881434624	Bytes
Shared buffer size	:	33554432	Bytes
Shared buffer hit ratio	:	100.00	%
Sorts in memory	:	119	
Sorts in disk	:	0	
In-memory sort ratio	:	100.00	%
Number of files	:	165	
Physical Reads	:	0	
Physical Writes	:	0	
Read Time	:	0	
Write Time	:	0	
dn_6005_6006:			
MPPDB CPU Time	:	220	Jiffies
Host CPU Busy Time	:	1100	Jiffies
Host CPU Total Time	:	42360	Jiffies
MPPDB CPU Time % in Busy Time	:	20.00	%
MPPDB CPU Time % in Total Time	:	.52	%
Physical memory	:	8231776256	Bytes
DB Memory usage	:	881430528	Bytes
Shared buffer size	:	33554432	Bytes
Shared buffer hit ratio	:	100.00	%
Sorts in memory	:	122	
Sorts in disk	:	0	
In-memory sort ratio	:	100.00	%
Number of files	:	115	

```

Physical Reads          :    0
Physical Writes        :    0
Read Time              :    0
Write Time             :    0

Session statistics information(Top 10):
Session CPU statistics:
1 dn_6003_6004-postgres-dbazt:
    Session CPU time      :    1211
    Database CPU time     :   37670
    Session CPU time %    :    3.21    %
2 dn_6005_6006-postgres-dbazt:
    Session CPU time      :    1204
    Database CPU time     :   35730
    Session CPU time %    :    3.37    %
3 dn_6001_6002-postgres-dbazt:
    Session CPU time      :    1064
    Database CPU time     :   34120
    Session CPU time %    :    3.12    %

Session Memory statistics:
1 dn_6005_6006-postgres-dbazt:
    Buffer Reads          :    8989
    Shared Buffer Hit ratio :   96.97
    In Memory sorts      :    136
    In Disk sorts        :     0
    In Memory sorts ratio :  100.00
    Total Memory Size    :  21593048
    Used Memory Size     :  18496600
2 dn_6003_6004-postgres-dbazt:
    Buffer Reads          :    9030
    Shared Buffer Hit ratio :   96.94
    In Memory sorts      :    133
    In Disk sorts        :     0
    In Memory sorts ratio :  100.00
    Total Memory Size    :  21576664
    Used Memory Size     :  18495688
3 dn_6001_6002-postgres-dbazt:
    Buffer Reads          :    8668
    Shared Buffer Hit ratio :   96.97
    In Memory sorts      :    138
    In Disk sorts        :     0
    
```

```

In Memory sorts ratio          :    100.00
Total Memory Size              :    21258856
Used Memory Size               :    18159736

Session IO statistics:
1 dn_6003_6004-postgres-dbazt:
  Physical Reads               :         285
  Read Time                    :        5320
2 dn_6005_6006-postgres-dbazt:
  Physical Reads               :         281
  Read Time                    :        5811
3 dn_6001_6002-postgres-dbazt:
  Physical Reads               :         271
  Read Time                    :        4662
    
```

3.4.5 相关命令

gs_check, gs_checkos

3.5 gs_collector

3.5.1 背景信息

当 GBase 8s 发生故障时，使用此工具收集 OS 信息、日志信息以及配置文件等信息，来定位问题。可以使用 -C 参数，指定收集不同的信息内容，具体支持收集的内容信息如表 3-8 所示。

表 3-8 gs_collector 内容收集对照表

类型	内容	描述	是否默认收集
System	HardWareInfo RunTimeInfo	收集现在操作系统的状态信息： ● HardWareInfo： ■ Cpu (cat /proc/cpuinfo) ■ 内存 (cat /proc/meminfo) ■ 磁盘 (df -h)	是

类型	内容	描述	是否默认收集
		<ul style="list-style-type: none"> ● RunTimeInfo : ■ topStatus (ps ux) ■ ioStat (iostat -xm 2 3) ■ 网络状态 (cat /proc/net/dev) ■ 内存使用情况 (free -m) 	
Database	具体列表见表 8-1。	收集系统视图或者系统表的内容。	是，默认收集 pg_locks 、 pg_stat_activity 和 pg_thread_wait_status 三个视图的信息。
Log	DataNode ClusterManager	<ul style="list-style-type: none"> ● 收集 DataNode 的 pg_log 和 gs_profile ● 收集 ClusterManager 的日志信息包括： <ul style="list-style-type: none"> ■ om ■ gs_ctl ■ gs_guc ■ gs_initdb 	是
Config	DataNode	<ul style="list-style-type: none"> ● 收集 DataNode 的配置信息： <ul style="list-style-type: none"> ■ postgresql.conf ■ pg_hba.conf ■ pg_control ■ pg_replslot 	是

类型	内容	描述	是否默认收集
		<ul style="list-style-type: none"> ■ pg_ident.conf 	
Gstack	DataNode	利用 gstack 命令得到 DataNode 的当前堆栈信息。	否
CoreDump	gs_ctl 、 gaussdb 、 GaussMaster、 AioWorker 、 AlarmChecker 、 Archiver 、 Auditor 、 AutoVacLaunc her 、 AutoVacWork er、 AuxMain、 BackendMode 、 BgWriter 、 BootStrap 、 Catchup 、 CBMWriter 、 Checkpointer、 CgbaseAuxStr eam 、 CgbasePoolCl eane 、 CgbaseRcvStr eam 、 CgbaseRcvWo rker 、	通过时间筛选 gaussdb 进程的 core 文件。 注意： 关于 Core 的配置和方式请按照如下要求：在 /proc/sys/kernel/core_pattern 文件中添加 core 文件的格式： core-%e-%p-%t。	否

类型	内容	描述	是否默认收集
	CgbaseSendStream 、 CpMonitor 、 DataRcvWriter 、 DataReceiver、 DataSender 、 ExtremeRTO、 FencedUDFMaster 、 Heartbeater 、 JobExecutor、 JobScheduler、 LWLockMonitor 、 PageWriter 、 ParallelRecover 、 PercentileJob、 Reaper 、 RemoteSrv 、 StartupProcess 、 StatCollector、 Stream 、 SysLogger 、 ThdPoolListener 、 TwoPhaseCleaner 、		

类型	内容	描述	是否默认收集
	WalRcvWriter 、 WalReceiver、 WalSender 、 WalWriter 、 WDRSnapshot 、WlmArbiter、 WlmCollector 、WlmMonitor		
XLog	DataNode	通过时间筛选收集 xlog。	否
Plan	*	收集计划复现信息： <ul style="list-style-type: none"> ● 用户输入*表示收集所有数据库上的计划复现信息。 ● 用户输入具体的数据库名称表示收集指定数据库上的计划复现信息。 	否

前提条件

日志收集工具依赖操作系统工具。例如 gstack 为其中一种，如果未安装该工具，则提示错误后，跳过该收集项。

3.5.2 语法

- 日志收集（非 root 用户）

```
gs_collector --begin-time="BEGINTIME" --end-time="ENDTIME" [-h HOSTNAME | -f HOSTFILE] [--keyword=KEYWORD] [--speed-limit=SPEED] [-o OUTPUT] [-l LOGFILE] [-C CONFIGFILE]
```

- 显示帮助信息


```
gs_collector -? | --help
```

- 显示版本号信息

```
gs_collector -V | --version
```

3.5.3 参数说明

- -h

收集信息的主机名称。

取值范围：主机名称，如果不指定则默认收集所有主机的信息。

- -f

主机名称列表文件。该文件为纯文本格式。

取值范围：主机名称列表。

说明

➤ -f 和 -h 参数不能同时使用。

- -o

将收集日志以压缩包形式输出到指定的文件夹。不指定则将检查结果以压缩包形式输出到配置文件中 `tmpMppdbPath` 选项所指定的目录中。

若配置文件中未配置 `tmpMppdbPath` 选项，则默认将检查结果以压缩包形式输出到“/tmp/用户名_mppdb/”目录中。

- -l

指定的日志文件以及存放路径。

- -C

指定待收集内容的配置文件，利用 `TypeName` 指定需要收集的信息类型，利用 `Content` 指定每一类信息的具体内容，利用 `Count` 指定此类信息收集的次数，利用 `Interval` 指定收集间隔，单位为秒。

`TypeName` 和 `Content` 不允许缺失或者内容为空。

`Interval` 和 `Count` 可以不指定，如果没有指定 `Count`，则默认收集一次。如果没有指定 `Interval` 则表示间隔为 0 秒，`Interval` 和 `Count` 的值不能小于 0。

配置文件格式采用 json 格式，模板如下：

```
{
  "Collect":
  [
    {"TypeName": "name", "Content": "value", "Interval": "seconds",
    "Count": "count"}
  ]
}
```

说明

- TypeName 和对应的 Content 取值范围见表 3-8。
- 对于 Log、CoreDump、Config 和 XLog 四种类型，Interval 和 Count 参数不生效。

如果不指定则使用默认的配置文件的。默认配置文件内容如下：

```
{
  "Collect":
  [
    {"TypeName": "System", "Content": "RunTimeInfo,HardWareInfo",
    "Interval": "0", "Count": "1"},
    {"TypeName": "Log", "Content": "DataNode,ClusterManager",
    "Interval": "0", "Count": "1"},
    {"TypeName": "Database", "Content":
    "pg_locks,pg_stat_activity,pg_thread_wait_status", "Interval": "0",
    "Count": "1"},
    {"TypeName": "Config", "Content": "DataNode", "Interval": "0",
    "Count": "1"}
  ]
}
```

● --keyword=KEYWORD

包含关键字 KEYWORD 的日志文件。

若关键字 KEYWORD 中含有空格，需要使用双引号包围。

说明

- 性能日志为二进制日志，关键字搜集功能不支持该日志的搜集。

● --begin-time

日志的开始时间。输入格式为“yyyymmdd hh:mm”。

- --end-time

日志的结束时间。输入格式为“yyyymmdd hh:mm”。

- --speed-limit

日志收集时的收集速率，输入格式为非负整数，单位为 MB/s。

该参数主要是为了防止日志收集过程中产生过高的磁盘或网络 IO，导致数据库节点故障（如果它们与 \$GAUSSLOG/\$PGHOST 部署在同一个磁盘上）。该值应当不超过 GBase 8s 内上述磁盘 IO 与网络 IO 速率的最小值的 1/3。

- -?, --help

显示帮助信息。

- -V, --version

显示版本号信息。

3.5.4 示例

执行如下命令收集 OS 信息和日志信息。

```
[gbase@gbase ~]$ gs_collector --begin-time="20180131 23:00"
--end-time="20180201 20:00" -h gbase
Successfully parsed the configuration file.
create Dir.
Successfully create dir.
do system check interval 0 : count 1
Collecting OS information.
Successfully collected OS information.
do database check interval 0 : count 1
Collecting catalog statistics.
Successfully collected catalog statistics.
do log check interval 0 : count 1
Collecting Log files.
Successfully collected Log files.
do Config check 0:1
Collecting Config files.
Successfully collected Config files.
Collecting files.
Successfully collected files.
```

```
All results are stored in
/home/gbase/gbase_db/log/collector_20221226_151012.tar.gz.
```

查看收集到的统计信息。

```
[gbase@gbase ~]$ tar -zxvf
/home/gbase/gbase_db/log/collector_20221226_151012.tar.gz
collector_20221226_151012/
collector_20221226_151012/gbase.tar.gz
collector_20221226_151012/Summary.log
collector_20221226_151012/Detail.log
[gbase@gbase ~]$ cd collector_20221226_151012/
[gbase@gbase ~]$ tar -zxvf gbase.tar.gz
gbase/
gbase/systemfiles/
gbase/systemfiles/OS_information_20221226_151013402842.txt
gbase/systemfiles/database_system_info_20221226_151017602152.txt
gbase/catalogfiles/
gbase/catalogfiles/dn_6001_pg_locks_20221226_151021961560.csv
gbase/catalogfiles/dn_6001_pg_stat_activity_20221226_151022221205.csv
gbase/catalogfiles/dn_6001_pg_thread_wait_status_20221226_151022490415.csv
gbase/catalogfiles/gs_clean_20221226_151022746639.txt
gbase/xlogfiles/
gbase/gstackfiles/
gbase/coreDumpfiles/
gbase/planSimulatorfiles/
gbase/logfiles/
gbase/logfiles/log_20221226_151023884151.tar.gz
gbase/configfiles/
gbase/configfiles/config_20221226_151025161378/
gbase/configfiles/config_20221226_151025161378/dn_6001/
gbase/configfiles/config_20221226_151025161378/dn_6001/postgresql.conf
gbase/configfiles/config_20221226_151025161378/dn_6001/pg_hba.conf
gbase/configfiles/config_20221226_151025161378/dn_6001/pg_control
gbase/configfiles/config_20221226_151025161378/dn_6001/gaussdb.state
gbase/configfiles/config_20221226_151025161378/dn_6001/pg_replslot/
gbase/configfiles/config_20221226_151025161378/dn_6001/pg_ident.conf

[gbase@gbase ~]$ cd gbase/logfiles/
[gbase@gbase ~]$ tar -zxvf log_20221226_151023884151.tar.gz
./
./om/
```

```
./om/gs_preinstall-2022-12-26_113654.log
./om/gs_local-2022-12-26_113710.log
./om/gs_install-2022-12-26_113911.log
./om/gs_checkperf-2022-12-26_114311.log
./om/gs_collector-2022-12-26_114240.log
./om/gs_backup-2022-12-26_114101.log
./om/gs_om-2022-12-26_114052.log
./bin/
./bin/gs_initdb/
./bin/gs_initdb/gs_initdb-2022-12-26_114010-current.log
./bin/gs_guc/
./bin/gs_guc/gs_guc-2022-12-26_114033-current.log
./bin/gs_ctl/
./bin/gs_ctl/gs_ctl-2022-12-26_114041-current.log
./pg_log/
./pg_log/dn_6001/
./pg_log/dn_6001/postgresql-2022-12-26_114043.log
./pg_log/dn_6001/postgresql-2022-12-26_114330.log
./pg_log/dn_6001/postgresql-2022-12-26_114316.log
./gs_profile/
./gs_profile/dn_6001_6002/
./gs_profile/dn_6001_6002/postgresql-2022-12-26_114330.prf
./gs_profile/dn_6001_6002/postgresql-2022-12-26_114316.prf
./gs_profile/dn_6001_6002/postgresql-2022-12-26_114043.prf
```

3.6 gs_dump

3.6.1 背景信息

`gs_dump` 是 GBase 8s 用于导出数据库相关信息的工具，用户可以自定义导出一个数据库或其中的对象（模式、表、视图等），回收站对象除外。支持导出的数据库可以是默认数据库 `postgres`，也可以是自定义数据库。

`gs_dump` 工具由操作系统用户 `gbase` 执行。`gs_dump` 工具在进行数据导出时，其他用户可以访问 GBase 8s 数据库（读或写）。

`gs_dump` 工具支持导出完整一致的数据。例如，T1 时刻启动 `gs_dump` 导出 A 数据库，那么导出数据结果将会是 T1 时刻 A 数据库的数据状态，T1 时刻之后对 A 数据库的修改不会被导出。

`gs_dump` 时生成列不会被转储。

gs_dump 支持导出兼容 v1 版本数据库的文本格式文件。

gs_dump 支持将数据库信息导出至纯文本格式的 SQL 脚本文件或其他归档文件中。

- 纯文本格式的 SQL 脚本文件：包含将数据库恢复为其保存时的状态所需的 SQL 语句。通过 gsql 运行该 SQL 脚本文件，可以恢复数据库。即使在其他主机和其他数据库产品上，只要对 SQL 脚本文件稍作修改，也可以用来重建数据库。
- 归档格式文件：包含将数据库恢复为其保存时的状态所需的数据，可以是 tar 格式、目录归档格式或自定义归档格式，详见表 3-9。该导出结果必须与 gs_restore 配合使用来恢复数据库，gs_restore 工具在导入时，系统允许用户选择需要导入的内容，甚至可以在导入之前对等待导入的内容进行排序。

3.6.2 主要功能

gs_dump 可以创建四种不同的导出文件格式，通过[-F 或者--format=]选项指定，具体如下表 3-9 所示。

表 3-9 导出文件格式

格式名称	-F 的参数值	说明	建议	对应导入工具
纯文本格式	p	纯文本脚本文件包含 SQL 语句和命令。命令可以由 gsql 命令行终端程序执行，用于重新创建数据库对象并加载表数据。	小型数据库，一般推荐纯文本格式。	使用 gsql 工具恢复数据库对象前，可以根据需要使用文本编辑器编辑纯文本导出文件。
自定义归档格式	c	一种二进制文件。支持从导出文件中恢复所有或所选数据库对象。	中型或大型数据库，推荐自定义归档格式。	使用 gs_restore 可以选择要从自定义归档导出文件中导入相应的数据库

格式名称	-F 的参数值	说明	建议	对应导入工具
目录归档格式	d	该格式会创建一个目录，该目录包含两类文件，一类是目录文件，另一类是每个表和 blob 对象对应的数据文件。	--	对象。
tar 归档格式	t	tar 归档文件支持从导出文件中恢复所有或所选数据库对象。tar 归档格式不支持压缩且对于单独表大小应小于 8GB。	--	

 说明

- 可以使用 `gs_dump` 程序将文件压缩为目录归档或自定义归档导出文件，减少导出文件的大小。生成目录归档或自定义归档导出文件时，默认进行中等级别的压缩。
`gs_dump` 程序无法压缩已归档导出文件。

注意事项

- 禁止修改导出的文件和内容，否则可能无法恢复成功。
- 为了保证数据一致性和完整性，`gs_dump` 会对需要转储的表设置共享锁。如果表在别的事务中设置了共享锁，`gs_dump` 会等待锁释放后锁定表。如果无法在指定时间内锁定某个表，转储会失败。用户可以通过指定 `--lock-wait-timeout` 选项，自定义等待锁超时时间。
- 不支持加密导出存储过程和函数。

3.6.3 语法

```
gs_dump [OPTION]... [DBNAME]
```

说明

- DBNAME 指定要连接的数据库，前面不需要加短或长选项。例如：不需要-d，直接指定 postgres 数据库。

```
gs_dump -p 15400 postgres -f file.sql
```

或者

```
export PGDATABASE=postgres gs_dump -p 15400 -f file.sql
```

环境变量：PGDATABASE

3.6.4 参数说明

通用参数

- -f, --file=FILENAME

将输出发送至指定文件或目录。如果省略该参数，则使用标准输出。如果输出格式为 (-F c/-F d/-F t) 时，必须指定-f 参数。如果-f 的参数值含有目录，要求当前用户对该目录具有读写权限，并且不能指定已有目录。

- -F, --format=c|d|t|p

选择输出格式。格式如下：

- p|plain：输出一个文本 SQL 脚本文件（默认）。
- c|custom：输出一个自定义格式的归档，并且以目录形式输出，作为 gs_restore 输入信息。该格式是最灵活的输出格式，因为能手动选择，而且能在恢复过程中将归档项重新排序。该格式默认状态下会被压缩。
- d|directory：该格式会创建一个目录，该目录包含两类文件，一类是目录文件，另一类是每个表和 blob 对象对应的数据文件。
- t|tar：输出一个 tar 格式的归档形式，作为 gs_restore 输入信息。tar 格式与目录格式兼容；tar 格式归档形式在提取过程中会生成一个有效的目录格式归档形

式。但是，tar 格式不支持压缩且对于单独表有 8GB 的大小限制。此外，表数据项的相应排序在恢复过程中不能更改。

输出一个 tar 格式的归档形式，也可以作为 gsql 输入信息。

- -v, --verbose

指定 verbose 模式。该选项将导致 gs_dump 向转储文件输出详细的对象注解和启动/停止次数，向标准错误流输出处理信息。

- -V, --version

打印 gs_dump 版本，然后退出。

- -Z, --compress=0-9

指定使用的压缩比级别。

取值范围：0~9

- 0 表示无压缩。
- 1 表示压缩比最小，处理速度最快。
- 9 表示压缩比最大，处理速度最慢。

针对自定义归档格式，该选项指定单个表数据片段的压缩，默认方式是以中等级别进行压缩。纯文本格式或 tar 归档格式目前不支持压缩。

- --lock-wait-timeout=TIMEOUT

请勿在转储刚开始时一直等待以获取共享表锁。如果无法在指定时间内锁定某个表，就选择失败。可以以任何符合 SET statement_timeout 的格式指定超时时间。

- -?, --help

显示 gs_dump 命令行参数帮助，然后退出。

转储参数

- -a, --data-only

只输出数据，不输出模式（数据定义）。转储表数据、大对象和序列值。

- -b, --blobs

该参数为扩展预留接口，不建议使用。

- **-c, --clean**

在将创建数据库对象的指令输出到备份文件之前，先将清理（删除）数据库对象的指令输出到备份文件中。（如果目标数据库中没有任何对象，`gs_restore` 工具可能会输出一些提示性的错误信息）

该选项只对文本格式有意义。针对归档格式，可以调用 `gs_restore` 时指定选项。

- **-C, --create**

备份文件以创建数据库和连接到创建的数据库的命令开始。（如果命令脚本是这种方式执行，可以先指定任意数据库用于执行创建数据库的命令，数据不会恢复到指定的数据库中，而是恢复到创建的数据库中。）

该选项只对文本格式有意义。针对归档格式，可以在调用 `gs_restore` 时指定选项。

- **-E, --encoding=ENCODING**

以指定的字符集编码创建转储。默认情况下，以数据库编码创建转储。（得到相同结果的另一个办法是将环境变量“PGCLIENTENCODING”设置为所需的转储编码。）

- **-n, --schema=SCHEMA**

只转储与模式名称匹配的模式，此选项包括模式本身和所有它包含的对象。如果该选项没有指定，所有在目标数据库中的非系统模式将会被转储。写入多个 `-n` 选项来选择多个模式。此外，根据 `gsql` 的 `\d` 命令所使用的相同规则，模式参数可被理解成一个 `pattern`，所以多个模式也可以通过在该 `pattern` 中写入通配符来选择。使用通配符时，注意给 `pattern` 打引号，防止 `shell` 扩展通配符。

说明

- 当 `-n` 已指定时，`gs_dump` 不会转储已选模式所附着的任何其他数据库对象。因此，无法保证某个指定模式的转储结果能够自行成功地储存到一个空数据库中。
- 当 `-n` 指定时，非模式对象不会被转储。
- 转储支持多个模式的转储。多次输入 `-n schemaname` 转储多个模式。

例如：

```
$ gs_dump -h 192.168.5.124 -p 15400 postgres -f  
/home/gbase/data/backup/bkp_sh12.sql -n sch1 -n sch2
```

在上面这个例子中，sch1 和 sch2 会被转储。

- **-N, --exclude-schema=SCHEMA**

不转储任何与模式 pattern 匹配的模式。pattern 将参照针对-n 的相同规则来理解。可以通过输入多次-N，不转储与任何 pattern 匹配的模式。

当同时输入-n 和-N 时，会转储与至少一个-n 选项匹配、与-N 选项不匹配的模式。如果有-N 没有-n，则不转储常规转储中与-N 匹配的模式。

转储过程支持排除多个模式。

在转储过程中，输入-N exclude schema name 排除多个模式。

例如：

```
$ gs_dump -h 192.168.5.124 -p 15400 postgres -f  
/home/gbase/data/backup/bkp_sh12.sql -N sch1 -N sch2
```

在上面这个例子中，sch1 和 sch2 在转储过程中会被排除。

- **-o, --oids**

转储每个表的对象标识符（OIDs），作为表的一部分数据。该选项用于应用以某种方式参照了 OID 列的情况。如果不是以上这种情况，请勿使用该选项。

- **-O, --no-owner**

不输出设置对象的归属这样的命令，以匹配原始数据库。默认情况下，gs_dump 会发出 ALTER OWNER 或 SET SESSION AUTHORIZATION 语句设置所创建的数据库对象的归属。如果脚本正在运行，该语句不会执行成功，除非是由系统管理员触发（或是拥有脚本中所有对象的同一个用户）。通过指定-O，编写一个任何用户都能存储的脚本，且该脚本会授予该用户拥有所有对象的权限。

该选项只对文本格式有意义。针对归档格式，可以在调用 gs_restore 时指定选项。

- **-s, --schema-only**

只转储对象定义（模式），而非数据。

- **-S, --sysadmin=NAME**

该参数为扩展预留接口，不建议使用。

- **-t, --table=TABLE**

指定转储的表（或视图、或序列、或外表）对象列表，可以使用多个-t 选项来选择多个表，也可以使用通配符指定多个表对象。

当使用通配符指定多个表对象时，注意给 pattern 打引号，防止 shell 扩展通配符。

当使用-t 时，-n 和-N 没有任何效应，这是因为由-t 选择的表的转储不受那些选项的影响。

 说明

- -t 参数选项个数必须小于等于 100。
- 如果-t 参数选项个数大于 100，建议使用参数--include-table-file 来替换。
- 当-t 已指定时，gs_dump 不会转储已选表所附着的任何其他数据库对象。因此，无法保证某个指定表的转储结果能够自行成功地储存到一个空数据库中。
- -t tablename 只转储在默认搜索路径中可见的表。-t '*.tablename'转储数据库下所有模式下的 tablename 表。-t schema.table 转储特定模式中的表。
- -t tablename 不会导出表上的触发器信息。

例如：

```
$ gs_dump -h 192.168.5.124 -p 15400 postgres -f  
/home/gbase/data/backup/bkp_sh12.sql -t schema1.table1 -t schema2.table2
```

在上面这个例子中，schema1.table1 和 schema2.table2 会被转储。

- **--include-table-file=FILENAME**

指定需要 dump 的表文件。

- **-T, --exclude-table=TABLE**

不转储的表（或视图、或序列、或外表）对象列表，可以使用多个-T 选项来选择多个表，也可以使用通配符指定多个表对象。

当同时输入-t 和-T 时，会转储在-t 列表中，而不在-T 列表中的表对象。

例如：

```
$ gs_dump -h 192.168.5.124 -p 15400 postgres -f  
/home/gbase/data/backup/bkp_sh12.sql -T table1 -T table2
```

在上面这个例子中，table1 和 table2 在转储过程中会被排除。

- `--exclude-table-file=FILENAME`

指定不需要 dump 的表文件。

说明

➤ 与 `--include-table-file` 参数格式相同，其内容格式如：`schema1.table1 schema2.table2`

- `-x, --no-privileges|--no-acl`

防止转储访问权限（授权/撤销命令）。

- `-q, --target`

指定导出兼容其他版本数据库的文本文件，目前支持 v1 和 v5 参数。v1 参数用于导出 v5 数据库的数据为兼容 v1 的文本文件。v5 参数用于导出 v5 数据库的数据为 v5 格式的文本文件，减少了导入 v5 时的可能的报错情况。

在使用 v1 参数时，建议和 `--exclude-guc="enable_cluster_resize"`、`--exclude-function`、`--exclude-with` 等选项共用，否则导入到 v1 时可能报错。

- `--exclude-guc`

导出的文本文件中，不包括相关 guc 参数的 set 命令，目前只支持 `enable_cluster_resize`。

- `--exclude-function`

不导出函数和存储过程。

- `--exclude-with`

导出的表定义，末尾不添加 `WITH(orientation=row, compression=on)` 这样的描述。

- `--binary-upgrade`

该参数为扩展预留接口，不建议使用。

- `--binary-upgrade-usermap="USER1=USER2"`

该参数为扩展预留接口，不建议使用。

- `--column-inserts|--attribute-inserts`

以 INSERT 命令带列名 (INSERT INTO 表 (列、...) 值...) 方式导出数据。这会导致恢复缓慢。但是由于该选项会针对每行生成一个独立分开的命令,所以在重新加载某行时出现的错误只会导致那行丢失,而非整个表内容。

- `--disable-dollar-quoting`

该选项将禁止在函数体前使用美元符号\$,并强制使用 SQL 标准字符串语法对其进行引用。

- `--disable-triggers`

该参数为扩展预留接口,不建议使用。

- `--exclude-table-data=TABLE`

指定不转储任何匹配表 `pattern` 的表这方面的数据。依照针对 `-t` 的相同规则理解该 `pattern`。

可多次输入 `--exclude-table-data` 来排除匹配任何 `pattern` 的表。当用户需要特定表的定义但不需要其中的数据时,这个选项很有帮助。

排除数据库中所有表的数据,参见 `--schema-only` 参数。

- `--inserts`

发出 INSERT 命令 (而非 COPY 命令) 转储数据。这会导致恢复缓慢。

但是由于该选项会针对每行生成一个独立分开的命令,所以在重新加载某行时出现的错误只会导致那行丢失,而非整个表内容。注意:如果重排列顺序,可能会导致整个恢复失败。列顺序改变时, `--column-inserts` 选项不受影响,虽然会更慢。

- `--no-publications`

不转储发布。

- `--no-security-labels`

该参数为扩展预留接口,不建议使用。

- `--no-subscriptions`

不转储订阅。

- `--no-tablespaces`

不输出选择表空间的命令。使用该选项，无论默认表空间是哪个，在恢复过程中所有对象都会被创建。

该选项只对文本格式有意义。针对归档格式，可以在调用 `gs_restore` 时指定选项。

- `--no-unlogged-table-data`

该参数为扩展预留接口，不建议使用。

- `--non-lock-table`

该参数为扩展预留接口，不建议使用。

- `--include-alter-table`

转储表删除列。该选项会记录列的删除。

- `--quote-all-identifiers`

强制对所有标识符加引号。为了向后续版本迁移，且其中可能涉及引入额外关键词，在转储相应数据库时该选项会有帮助。

- `--section=SECTION`

指定已转储的名称区段（`pre-data`、`data` 和 `post-data`）。

- `--serializable-deferrable`

转储过程中使用可串行化事务，以确保所使用的快照与之后的数据库状态一致；要实现该操作需要在无异常状况的事务流中等待某个点，因为这样才能保证转储成功，避免引起其他事务出现 `serialization_failure` 要重新再做。

但是该选项对于灾难恢复没有益处。对于在原始数据库进行升级的时候，加载一个数据库的拷贝作为报告或其他只读加载共享的转储是有帮助的。没有这个选项，转储会反映一个与任何事务最终提交的序列化执行不一致的状态。

如果当 `gs_dump` 启动时，读写事务仍处于非活动状态，即便使用该选项也不会对其产生影响。如果读写事务处于活动状态，转储的开始时间可能会延迟一段不确定的时间。

- `--use-set-session-authorization`

输出符合 SQL 标准的 `SET SESSION AUTHORIZATION` 命令而不是 `ALTER OWNER` 命令来确定对象所有权。这样令转储更加符合标准，但是如果转储文件中的对象的历史有些问题，那么可能不能正确恢复。并且，使用 `SET SESSION AUTHORIZATION` 的转储需要数据

库系统管理员的权限才能转储成功，而 ALTER OWNER 需要的权限则低得多。

- `--with-encryption=AES128`

指定转储数据需用 AES128 进行加密。

- `--with-key=KEY`

AES128 密钥长度规则如下：

- 密钥长度为 8~16 字符。
- 至少包含大写字母 (A-Z)、小写字母 (a-z)、数字 (0-9)、非字母数字字符（限定为~!@#\$%^&*()-_+=[]{};:;<.>/?）四类字符中的三类字符。

说明

- 使用 `gs_dump` 工具进行加密导出时，仅支持 `plain` 格式导出。通过 `-F plain` 导出的数据，需要通过 `gsql` 工具进行导入，且如果以加密方式导入，在通过 `gsql` 导入时，需要指定 `--with-key` 参数。

不支持加密导出存储过程和函数。

- `--with-salt=RANDVALUES`

`gs_dumpall` 使用此参数传递随机值。

- `--include-extension`

在转储中包含扩展。

- `--include-depend-objs`

备份结果包含依赖于指定对象的对象信息。该参数需要同 `-t/--include-table-file` 参数关联使用才会生效。

- `--exclude-self`

备份结果不包含指定对象自身的信息。该参数需要同 `-t/--include-table-file` 参数关联使用才会生效。

- `--pipeline`

使用管道传输密码，禁止在终端使用。

- `--dont-overwrite-file`

文本、tar 以及自定义格式情况下会重写现有文件。这对目录格式不适用。

例如：

设想这样一种情景，即当前目录下 backup.sql 已存在。如果在输入命令中输入 -f backup.sql 选项时，当前目录恰好也生成 backup.sql，文件就会被重写。

如果备份文件已存在，且输入 --dont-overwrite-file 选项，则会报告附带“转储文件已经存在信息”的错误。

```
$ gs_dump -p 15400 postgres -f /home/gbase/data/backup.sql -F plain
--dont-overwrite-file
```

 说明

- `-s/--schema-only` 和 `-a/--data-only` 不能同时使用。
- `-c/--clean` 和 `-a/--data-only` 不能同时使用。
- `--inserts/--column-inserts` 和 `-o/--oids` 不能同时使用，因为 INSERT 命令不能设置 OIDS。
- `--role` 和 `--rolepassword` 必须一起使用。
- `--binary-upgrade-usermap` 和 `--binary-upgrade` 必须一起使用。
- `--include-depend-objs/--exclude-self` 需要同 `-t/--include-table-file` 参数关联使用才会生效。
- `--exclude-self` 必须同 `--include-depend-objs` 一起使用。

连接参数：

- `-h, --host=HOSTNAME`

指定主机名称。如果数值以斜杠开头，则被用作到 Unix 域套接字的路径。缺省从 PGHOST 环境变量中获取（如果已设置），否则，尝试一个 Unix 域套接字连接。

该参数只针对 GBase 8s 外，对 GBase 8s 内本机只能用 127.0.0.1。

环境变量：PGHOST

- `-p, --port=PORT`

指定主机端口。在开启线程池情况下，建议使用 `pooler port`，即主机端口+1。

环境变量：`PGPORT`

- `-U, --username=NAME`

指定所连接主机的用户名。

不指定连接主机的用户名时，用户默认系统管理员。

环境变量：`PGUSER`

- `-w, --no-password`

不出现输入密码提示。如果主机要求密码认证并且密码没有通过其它形式给出，则连接尝试将会失败。该选项在批量工作和不存在用户输入密码的脚本中很有帮助。

- `-W, --password=PASSWORD`

指定用户连接的密码。如果主机的认证策略是 `trust`，则不会对系统管理员进行密码验证，即无需输入 `-W` 选项；如果没有 `-W` 选项，并且不是系统管理员，“Dump Restore 工具”会提示用户输入密码。

- `--role=ROLENAME`

指定创建转储使用的角色名。选择该选项，会使 `gs_dump` 连接数据库后，发起一个 `SET ROLE` 角色名命令。当所授权用户（由 `-U` 指定）没有 `gs_dump` 要求的权限时，该选项会起到作用，即切换到具备相应权限的角色。某些安装操作规定不允许直接以超系统管理员身份登录，而使用该选项能够在不违反该规定的情况下完成转储。

- `--rolepassword=ROLEPASSWORD`

指定角色名的密码。

3.6.5 说明

如果 GBase 8s 有任何本地数据要添加到 `template1` 数据库，请谨慎将 `gs_dump` 的输出恢复到一个真正的空数据库中，否则可能会因为被添加对象的定义被复制，出现错误。要创建一个无本地添加的空数据库，需从 `template0` 而非 `template1` 复制，例如：

```
postgres=# CREATE DATABASE foo WITH TEMPLATE template0;
```

`tar` 归档形式的文件大小不得超过 8GB（`tar` 文件格式的固有限制）。`tar` 文档整体大小和

任何其他输出格式没有限制，操作系统可能对此有要求。

由 `gs_dump` 生成的转储文件不包含优化程序用来做执行计划决定的统计数据。因此，最好从某转储文件恢复之后运行 `ANALYZE` 以确保最佳效果。转储文件不包含任何 `ALTER DATABASE...SET` 命令，这些设置由 `gs_dumpall` 转储，还有数据库用户和其他完成安装设置。

3.6.6 示例

使用 `gs_dump` 转储数据库为 SQL 文本文件或其它格式的操作，如下所示。

其中，用户密码为 `Bigdata@123`；`/home/gbase/data/backup/MPPDB_backup.sql` 表示导出的文件；`15400` 表示数据库服务器端口；`postgres` 表示要访问的数据库名。

说明

- 导出操作时，请确保该目录存在并且当前的操作系统用户对其具有读写权限。

示例 1：执行 `gs_dump`，导出 `postgres` 数据库全量信息，导出的 `MPPDB_backup.sql` 文件格式为纯文本格式。

```
[gbase@gbasehot ~]$ gs_dump -U gbase -W Bigdata@123 -f
/home/gbase/data/backup/MPPDB_backup.sql -p 15400 postgres -F p
gs_dump[port='15400'][postgres][2018-06-27 09:49:17]: The total objects number
is 356.
gs_dump[port='15400'][postgres][2018-06-27 09:49:17]: [100.00%] 356 objects
have been dumped.
gs_dump[port='15400'][postgres][2018-06-27 09:49:17]: dump database postgres
successfully
gs_dump[port='15400'][postgres][2018-06-27 09:49:17]: total time: 1274 ms
```

使用 `gsqll` 程序从纯文本导出文件中导入数据。

示例 2：执行 `gs_dump`，导出 `postgres` 数据库全量信息，导出的 `MPPDB_backup.tar` 文件格式为 `tar` 格式。

```
[gbase@gbasehot ~]$ gs_dump -U gbase -W Bigdata@123 -f
/home/gbase/data/backup/MPPDB_backup.tar -p 15400 postgres -F t
gs_dump[port='15400'][postgres][2018-06-27 10:02:24]: The total objects number
is 1369.
gs_dump[port='15400'][postgres][2018-06-27 10:02:53]: [100.00%] 1369 objects
have been dumped.
```

```
gs_dump[port='15400'] [postgres] [2018-06-27 10:02:53]: dump database postgres
successfully
gs_dump[port='15400'] [postgres] [2018-06-27 10:02:53]: total time: 50086 ms
```

示例 3: 执行 `gs_dump`, 导出 `postgres` 数据库全量信息, 导出的 `MPPDB_backup.dmp` 文件格式为自定义归档格式。

```
[gbase@gbasehot ~]$ gs_dump -U gbase -W Bigdata@123 -f
/home/gbase/data/backup/MPPDB_backup.dmp -p 15400 postgres -F c
gs_dump[port='15400'] [postgres] [2018-06-27 10:05:40]: The total objects number
is 1369.
gs_dump[port='15400'] [postgres] [2018-06-27 10:06:03]: [100.00%] 1369 objects
have been dumped.
gs_dump[port='15400'] [postgres] [2018-06-27 10:06:03]: dump database postgres
successfully
gs_dump[port='15400'] [postgres] [2018-06-27 10:06:03]: total time: 36620 ms
```

示例 4: 执行 `gs_dump`, 导出 `postgres` 数据库全量信息, 导出的 `MPPDB_backup` 文件格式为目录格式。

```
[gbase@gbasehot ~]$ gs_dump -U gbase -W Bigdata@123 -f
/home/gbase/data/backup/MPPDB_backup -p 15400 postgres -F d
gs_dump[port='15400'] [postgres] [2018-06-27 10:16:04]: The total objects number
is 1369.
gs_dump[port='15400'] [postgres] [2018-06-27 10:16:23]: [100.00%] 1369 objects
have been dumped.
gs_dump[port='15400'] [postgres] [2018-06-27 10:16:23]: dump database postgres
successfully
gs_dump[port='15400'] [postgres] [2018-06-27 10:16:23]: total time: 33977 ms
```

示例 5: 执行 `gs_dump`, 导出 `postgres` 数据库信息, 但不导出 `/home/MPPDB_temp.sql` 中指定的表信息。导出的 `MPPDB_backup.sql` 文件格式为纯文本格式。

```
[gbase@gbasehot ~]$ gs_dump -U gbase -W Bigdata@123 -p 15400 postgres
--exclude-table-file=/home/gbase/data/MPPDB_temp.sql -f
/home/gbase/data/backup/MPPDB_backup.sql
gs_dump[port='15400'] [postgres] [2018-06-27 10:37:01]: The total objects number
is 1367.
gs_dump[port='15400'] [postgres] [2018-06-27 10:37:22]: [100.00%] 1367 objects
have been dumped.
gs_dump[port='15400'] [postgres] [2018-06-27 10:37:22]: dump database postgres
successfully
gs_dump[port='15400'] [postgres] [2018-06-27 10:37:22]: total time: 37017 ms
```

示例 6: 执行 `gs_dump`, 仅导出依赖于指定表 `testtable` 的视图信息。然后创建新的 `testtable` 表, 再恢复依赖其上的视图。

备份仅依赖于 `testtable` 的视图。

```
[gbase@gbasehot ~]$ gs_dump -s -p 15400 postgres -t PUBLIC.testtable
--include-depend-objs --exclude-self -f
/home/gbase/data/backup/MPPDB_backup.sql -F p
gs_dump[port='15400'][postgres][2018-06-15 14:12:54]: The total objects number
is 331.
gs_dump[port='15400'][postgres][2018-06-15 14:12:54]: [100.00%] 331 objects
have been dumped.
gs_dump[port='15400'][postgres][2018-06-15 14:12:54]: dump database postgres
successfully
gs_dump[port='15400'][postgres][2018-06-15 14:12:54]: total time: 327 ms
```

修改 `testtable` 名称。

```
[gbase@gbasehot ~]$ gsql -p 15400 postgres -r -c "ALTER TABLE PUBLIC.testtable
RENAME TO testtable_bak;"
```

创建新的 `testtable` 表。

```
postgres=# CREATE TABLE PUBLIC.testtable(a int, b int, c int);
```

还原依赖于 `testtable` 的视图。

```
[gbase@gbasehot ~]$ gsql -p 15400 postgres -r -f
/home/gbase/data/backup/MPPDB_backup.sql
```

3.6.7 相关命令

`gs_dumpall`, `gs_restore`

3.7 `gs_dumpall`

3.7.1 背景信息

`gs_dumpall` 是 GBase 8s 用于导出所有数据库相关信息工具, 它可以导出 GBase 8s 数据库的所有数据, 包括默认数据库 `postgres` 的数据、自定义数据库的数据以及 GBase 8s 所有数据库公共的全局对象。

`gs_dumpall` 工具由操作系统用户 `gbase` 执行。

`gs_dumpall` 工具在进行数据导出时, 其他用户可以访问 GBase 8s 数据库 (读或写)。

gs_dumpall 工具支持导出完整一致的数据。例如, T1 时刻启动 gs_dumpall 导出 GBase 8s 数据库, 那么导出数据结果将会是 T1 时刻该 GBase 8s 数据库的数据状态, T1 时刻之后对 GBase 8s 的修改不会被导出。

gs_dumpall 时生成列不会被转储。

gs_dumpall 在导出 GBase 8s 所有数据库时分为两部分:

- gs_dumpall 自身对所有数据库公共的全局对象进行导出, 包括有关数据库用户和组、表空间以及属性 (例如, 适用于数据库整体的访问权限) 信息。
- gs_dumpall 通过调用 gs_dump 来完成 GBase 8s 中各数据库的 SQL 脚本文件导出, 该脚本文件包含将数据库恢复为其保存时的状态所需要的全部 SQL 语句。

以上两部分导出的结果为纯文本格式的 SQL 脚本文件, 使用 gsql 运行该脚本文件可以恢复 GBase 8s 数据库。

注意事项

禁止修改导出的文件和内容, 否则可能无法恢复成功。

为了保证数据一致性和完整性, gs_dumpall 会对需要转储的表设置共享锁。如果某张表在别的事务中设置了共享锁, gs_dumpall 会等待此表的锁释放后锁定此表。如果无法在指定时间内锁定某张表, 转储会失败。用户可以通过指定 --lock-wait-timeout 选项, 自定义等待锁超时时间。

由于 gs_dumpall 读取所有数据库中的表, 因此必须以 GBase 8s 管理员身份进行连接, 才能导出完整文件。在使用 gsql 执行脚本文件导入时, 同样需要管理员权限, 以便添加用户和组以及创建数据库。

3.7.2 语法

```
gs_dumpall [OPTION]...
```

3.7.3 参数说明

通用参数

- -f, --filename=FILENAME

将输出发送至指定文件。如果这里省略, 则使用标准输出。

- **-v, --verbose**

指定 verbose 模式。该选项将导致 `gs_dumpall` 向转储文件输出详细的对象注解和启动/停止次数，向标准错误流输出处理信息。

- **-V, --version**

打印 `gs_dumpall` 版本，然后退出。

- **--lock-wait-timeout=TIMEOUT**

请勿在转储刚开始时一直等待以获取共享表锁。如果无法在指定时间内锁定某个表，就选择失败。可以以任何符合 `SET statement_timeout` 的格式指定超时时间。

- **-, --help**

显示 `gs_dumpall` 命令行参数帮助，然后退出。

转储参数

- **-a, --data-only**

只转储数据，不转储模式（数据定义）。

- **-c, --clean**

在重新创建数据库之前，执行 SQL 语句清理（删除）这些数据库。针对角色和表空间的转储命令已添加。

- **-g, --globals-only**

只转储全局对象（角色和表空间），无数据库。

- **-o, --oids**

转储每个表的对象标识符（OIDs），作为表的一部分数据。该选项用于应用以某种方式参照了 OID 列的情况。如果不是以上这种情况，请勿使用该选项。

- **-O, --no-owner**

不输出设置对象的归属这样的命令，以匹配原始数据库。默认情况下，`gs_dumpall` 会发出 `ALTER OWNER` 或 `SET SESSION AUTHORIZATION` 语句设置所创建的模式元素的所属。如果脚本正在运行，该语句不会执行成功，除非是由系统管理员触发（或是拥有脚本中所有对象的同一个用户）。通过指定 `-O`，编写一个任何用户都能存储的脚本，且该脚本会授予该

用户拥有所有对象的权限。

- **-r, --roles-only**

只转储角色，不转储数据库或表空间。

- **-s, --schema-only**

只转储对象定义（模式），而非数据。

- **-S, --sysadmin=NAME**

在转储过程中使用的系统管理员名称。

- **-t, --tablespaces-only**

只转储表空间，不转储数据库或角色。

- **-x, --no-privileges**

防止转储访问权限（授权/撤销命令）。

- **--column-inserts|--attribute-inserts**

以 INSERT 命令带列名（INSERT INTO 表（列、...）值...）方式导出数据。这会导致恢复缓慢。但是由于该选项会针对每行生成一个独立分开的命令，所以在重新加载某行时出现的错误只会导致那行丢失，而非整个表内容。

- **--disable-dollar-quoting**

该选项将禁止在函数体前使用美元符号\$，并强制使用 SQL 标准字符串语法对其进行引用。

- **--disable-triggers**

该参数为扩展预留接口，不建议使用。

- **--inserts**

发出 INSERT 命令（而非 COPY 命令）转储数据。这会导致恢复缓慢。注意：如果重排列顺序，可能会导致恢复整个失败。--column-inserts 选项更加安全，虽然可能更慢些。

- **--no-publications**

不转储发布。

- `--no-security-labels`

该参数为扩展预留接口，不建议使用。

- `--no-subscriptions`

不转储订阅。

- `--no-tablespaces`

请勿输出创建表空间的命令，也请勿针对对象选择表空间。使用该选项，无论默认表空间是哪个，在恢复过程中所有对象都会被创建。

- `--no-unlogged-table-data`

该参数为扩展预留接口，不建议使用。

- `--include-alter-table`

导出表中已删除的列信息。

- `--quote-all-identifiers`

强制对所有标识符加引号。为了向后续版本迁移，且其中可能涉及引入额外关键词，在转储相应数据库时该选项会有帮助。

- `--dont-overwrite-file`

不重写当前文件。

- `--use-set-session-authorization`

输出符合 SQL 标准的 SET SESSION AUTHORIZATION 命令而不是 ALTER OWNER 命令来确定对象所有权。这样令转储更加符合标准，但是如果转储文件中的对象的历史有些问题，那么可能不能正确恢复。并且，使用 SET SESSION AUTHORIZATION 的转储需要数据库系统管理员的权限才能转储成功，而 ALTER OWNER 需要的权限则低得多。

- `--with-encryption=AES128`

指定转储数据需用 AES128 进行加密。

- `--with-key=KEY`

AES128 密钥长度规则如下：

- 密钥长度为 8~16 字符。
- 至少包含大写字母 (A-Z)、小写字母 (a-z)、数字 (0-9)、非字母数字字符 (限定为~!@#\$\$%^&*()-_+=[]{};:,<>/?) 四类字符中的三类字符。
- --include-extension
如果--include-extension 参数被设置, 将备份所有的 CREATE EXTENSION 语句。
- --include-templatedb
转储过程中包含模板库。
- --binary-upgrade
该参数为扩展预留接口, 不建议使用。
- --binary-upgrade-usermap="USER1=USER2"
该参数为扩展预留接口, 不建议使用。
- --non-lock-table
该参数仅供 OM 工具使用。
- --tablespaces-postfix
该参数为扩展预留接口, 不建议使用。
- --parallel-jobs
指定备份进程并发数, 取值范围为 1~1000。
- --pipeline
使用管道传输密码, 禁止在终端使用。

说明

- -g/--globals-only 和-r/--roles-only 不能同时使用。
- -g/--globals-only 和-t/--tablespaces-only 不能同时使用。
- -r/--roles-only 和-t/--tablespaces-only 不能同时使用。
- -s/--schema-only 和-a/--data-only 不能同时使用。

- `-r/--roles-only` 和 `-a/--data-only` 不能同时使用。
- `-t/--tablespaces-only` 和 `-a/--data-only` 不能同时使用。
- `-g/--globals-only` 和 `-a/--data-only` 不能同时使用。
- `--tablespaces-postfix` 和 `--binary-upgrade` 必须一起使用。
- `--binary-upgrade-usermap` 和 `--binary-upgrade` 必须一起使用。
- `--parallel-jobs` 和 `-f/--file` 必须一起使用。

连接参数

- `-h, --host`

指定主机的名称。如果取值是以斜线开头，它将用作 Unix 域套接字的目录。默认值取自 `PGHOST` 环境变量；如果没有设置，将启动某个 Unix 域套接字建立连接。

该参数只针对 GBase 8s 外，对 GBase 8s 内本机只能用 127.0.0.1。

环境变量：`PGHOST`

- `-l, --database`

指定所连接的转储全局对象的数据库名称，并去寻找还有其他哪些数据库需要被转储。如果没有指定，会使用 `postgres` 数据库，如果 `postgres` 数据库不存在，会使用 `template1`。

- `-p, --port`

指定服务器所侦听的 TCP 端口或本地 Unix 域套接字后缀，以确保连接。默认值设置为 `PGPORT` 环境变量。

在开启线程池情况下，建议使用 `pooler port`，即侦听端口+1。

环境变量：`PGPORT`

- `-U, --username`

所连接的用户名。

环境变量：`PGUSER`

- `-w, --no-password`

不出现输入密码提示。如果服务器要求密码认证并且密码没有通过其它形式给出，则连

接尝试将会失败。该选项在批量工作和不存在用户输入密码的脚本中很有帮助。

- `-W, --password`

指定用户连接的密码。如果主机的认证策略是 `trust`，则不会对系统管理员进行密码验证，即无需输入 `-W` 选项；如果没有 `-W` 选项，并且不是系统管理员，“Dump Restore 工具”会提示用户输入密码。

- `--role`

指定创建转储使用的角色名。选择该选项，会使 `gs_dumpall` 连接数据库后，发起一个 `SET ROLE` 角色名命令。当所授权用户（由 `-U` 指定）没有 `gs_dumpall` 要求的权限时，该选项会起到作用，即切换到具备相应权限的角色。某些安装操作规定不允许直接以系统管理员身份登录，而使用该选项能够在不违反该规定的情况下完成转储。

- `--rolepassword`

指定具体角色用户的角色密码。

3.7.4 说明

由于 `gs_dumpall` 内部调用 `gs_dump`，所以一些诊断信息参见 `gs_dump`。

一旦恢复，最好在每个数据库上运行 `ANALYZE`，优化程序提供有用的统计数据。

`gs_dumpall` 恢复前需要所有必要的表空间目录才能退出；否则，对于处在非默认位置的数据库，数据库创建会失败。

3.7.5 示例

使用 `gs_dumpall` 一次导出 GBase 8s 的所有数据库。

说明

- `gs_dumpall` 仅支持纯文本格式导出。所以只能使用 `gsql` 恢复 `gs_dumpall` 导出的转储内容。

```
$ gs_dumpall -f /home/gbase/data/backup/bkp2.sql -p 15400
gs_dump[port='15400'][dbname='postgres'][2022-12-28 10:00:18]: The total
objects number is 413.
gs_dump[port='15400'][dbname='postgres'][2022-12-28 10:00:18]: [100.00%] 413
objects have been dumped.
```

```
gs_dump[port='15400'][dbname='postgres'][2022-12-28 10:00:18]: dump database
dbname='postgres' successfully
gs_dump[port='15400'][dbname='postgres'][2022-12-28 10:00:18]: total time: 7941
ms
gs_dumpall[port='15400'][2022-12-28 10:00:18]: dumpall operation successful
gs_dumpall[port='15400'][2022-12-28 10:00:18]: total time: 8215 ms
```

3.7.6 相关命令

`gs_dump`, `gs_restore`

3.8 gs_guc

3.8.1 背景信息

应用程序可以通过调用 `gs_guc`, 来设置 GBase 8s 配置文件 `postgresql.conf` 和 `pg_hba.conf` 中的参数。

`gs_guc` 工具由操作系统用户 `gbase` 执行。

- `gs_guc-current.log`

这个文件记录 `gs_guc` 产生的日志。

默认路径: `$GAUSSLOG/bin/gs_guc`。

- `gs_guc-年-月-日_HHMMSS.log`

当 `gs_guc-current.log` 文件大小满 16M 后, 会根据当前时刻生成一个备份文件。

- `server.key.cipher`, `server.key.rand`

使用 `gs_guc encrypt` 命令加密用户密码时, 如果指定的 `-M` 的值为 `server`, 则会生成这两个文件。其中 `server.key.cipher` 存储用户密码的密文, `server.key.rand` 存储的是加密因子。

- `client.key.cipher`, `client.key.rand`

使用 `gs_guc encrypt` 命令加密用户密码时, 如果指定的 `-M` 的值为 `client`, 则会生成这两个文件。其中 `client.key.cipher` 存储用户密码的密文, `client.key.rand` 存储的是加密因子。

说明

➤ `client.key.cipher` 和 `client.key.rand` 是不指定 `-U` 选项时生成的密文文件和加密因子文

件。如果通过-U 指定了用户名，则会生成以用户名开头的文件。以指定-U test 为例，生成的文件名为 test.key.cipher、test.key.rand。

- datasource.key.cipher, datasource.key.rand

使用 gs_guc encrypt 命令加密用户密码时，如果指定的-M 的值为 source，则会生成这两个文件。其中 datasource.key.cipher 存储用户密码的密文，datasource.key.rand 存储的是加密因子。

说明

- datasource.key.cipher 和 datasource.key.rand 是创建 Data Source 对象时调用的密钥文件。gs_guc 生成时即有读权限。使用前需将这两个文件放入各节点目录 \$GAUSSHOME/bin，且确保具有读权限。gs_ssh 工具可以协助您快速将文件放入集群各节点对应目录下。

此外，使用 gs_guc generate 命令也可以生成这两个文件，可以选择以下两种方式中的任一种，并根据提示输入密码。

方式 1

```
gs_guc encrypt -M source -D ./
```

方式 2

```
gs_guc generate -o datasource -D ./
```

3.8.2 语法

- 检查配置文件中参数。

```
gs_guc check [-N NODE-NAME] [-I INSTANCE-NAME | -D DATADIR] -c "parameter"
```

- 修改配置文件中参数。

```
gs_guc set [-N NODE-NAME] [-I INSTANCE-NAME | -D DATADIR] -c "parameter = value"
```

- 将已设置的参数值修改为默认值。

```
gs_guc set|reload [-N NODE-NAME] [-I INSTANCE-NAME | -D DATADIR] -c "parameter"
```

- 修改配置文件中参数，同时发送信号量到 postgresql.conf。

```
gs_guc reload [-N NODE-NAME] [-I INSTANCE-NAME | -D DATADIR] -c parameter=value
```

- 修改客户端认证策略。若选择 reload 会同时发送信号量到 pg_hba.conf，即无需重启即可生效。

```
gs_guc set|reload [-N NODE-NAME] {-I INSTANCE-NAME | -D DATADIR} -h "HOSTTYPE
DATABASE USERNAME IPADDR IPMASK AUTHMEHOD authentication-options"
gs_guc set|reload [-N NODE-NAME] {-I INSTANCE-NAME | -D DATADIR} -h "HOSTTYPE
DATABASE USERNAME IPADDR-WITH-IPMASK AUTHMEHOD authentication-options"
gs_guc set|reload [-N NODE-NAME] {-I INSTANCE-NAME | -D DATADIR} -h "HOSTTYPE
DATABASE USERNAME HOSTNAME AUTHMEHOD authentication-options"
```

说明

AUTHMEHOD 后的 authentication-options 为可选参数，AUTHMEHOD 支持以下选项：

- trust：不验密，禁止远程主机使用 trust 方式访问数据库。
- reject：拒绝访问。
- md5：md5 认证，默认不支持（MD5 加密算法安全性低，存在安全风险，不建议使用）。
- sha256：sha256 认证（推荐使用）。
- cert：客户端证书认证。
- gss：kerberos 认证，仅用于内部节点间认证。
- sm3：sm3 认证（国密 SM3）。
- pg_hba.conf 中的认证策略越靠前优先级越高，使用 gs_guc 工具配置时会按一定规则排序将新策略插入到原有认证策略中。配置字段比较顺序为：IPADDR/HOSTNAME > HOSTTYPE > DATABASE > USERNAME，即优先比较 IPADDR 或 HOSTNAME，如果无法区分优先级则继续比较 HOSTTYPE，以此类推。对于每个配置字段，通常越严格的配置参数优先级越高、排序越靠前，越宽松的配置参数优先级越低、排序越靠后，具体如下：
 - IPADDR：当配置为全 0 时表示不限制 IP，会放在指定具体某个 IP 地址的策略后面。
 - DATABASE：当配置为 all 时表示不限制数据库，会放在指定具体某个数据库的策略

略后面；当数据库配置为 replication 时会放在其他策略后面。

- USERNAME: 当配置为 all 时表示不限制用户，会放在指定具体某个用户的策略后面。
- HOSTTYPE: local > hostssl > hostnossl > host。
- 注释已经设置的客户端认证策略。若选择 reload 会同时发送信号量到 pg_hba.conf，即无需重启即可生效。

```
gs_guc set|reload [-N NODE-NAME] {-I INSTANCE-NAME | -D DATADIR} -h"HOSTTYPE
DATABASE USERNAME IPADDR-WITH-IPMASK AUTHMEHOD"
```

- 显示帮助信息。

```
gs_guc -? | --help
```

- 显示版本号信息。

```
gs_guc -V | --version
```

- gs_guc encrypt 生成加密密码和加密因子文件。

```
gs_guc encrypt [-M keymode] -K password [-U username] -D DATADIR
```

说明

- -K 是用户指定的密码，gs_guc 会对该密码进行长度 ($8 \leq \text{len} \leq 16$) 和密码复杂度要求，如果不满足，将会报错。此密码用于保证生成密码文件的安全性和唯一性，用户无需保存或记忆。
- -M 是加密类型，当前仅支持 server、client 和 source。默认值为 server。
- gs_guc generate 生成其他前缀的加密密码和加密因子文件。

```
gs_guc generate [-o prefix] [-S cipherkey] -D DATADIR
```

说明

- -o 是输出 cipher 和 rand 文件前缀名称，默认输出文件名前缀为 obsserver。其内容仅支持数字、字母和下划线。
- -S 是用户指定的密码，密码需要满足长度要求 ($8 \leq \text{len} \leq 16$) 和复杂度要求，如不满足将会报错。当其值为 default 时，会随机生成一段字符串作为密码，该密码

长度为 13。如果不带-S 参数则会提示交互式输入密码。为了系统安全，推荐使用交互式输入密码方式。

- 使用 `gs_guc encrypt` 或 `generate` 命令生成加密密码和加密因子文件时只是参数不同，本质上是一样的。生成过程中会使用随机数作为加密密钥材料和盐值，因此是每次生成的文件都是不同的。每次生成的加密密码和加密因子文件需要成对使用，不能更换或交替使用，加密和解密时需要使用相同的加密密码和加密因子文件。

说明

- `gs_guc` 工具不支持参数值中包含 '#' 的设置。可以使用 `vi` 工具通过手工修改配置文件来设置。
- 如果已经在环境变量中设置 `PGDATA`，则可以省略-D 参数。否则提示设置参数失败。
- 环境变量 `PGDATA` 设置方法：先将参数 `PGDATA` 设置为 `/home/gbase/data`，然后在执行“`export PGDATA`”使设置生效。
- `authpolicy` 包含一串认证参数：`HOSTTYPE DATABASE USERNAME IPADDR IPMASK` 或者 `HOSTTYPE DATABASE USERNAME IPADDR-WITH-IPMASK` 或者 `HOSTTYPE DATABASE USERNAME HOSTNAME`。
- 如果设置 GUC 参数时使用 `-c “parameter”`，则会将已设置的 GUC 参数值设置成该参数的内核默认值（注意 `log_directory` 和 `audit_directory` 不会被设置为内核参数默认值，而是设为 `$GAUSSLOG/pg_audit/instance_name`）。因 GUC 参数间存在依赖关系，因此请慎用该功能。
- 设置 `-c` 参数时，参数都可以省略双引号。
- 如果 `value` 中含有特殊字符（如 `$`），请转义后使用。
- 如果同一个配置参数在配置文件里面出现多行，且有两行或多于两行同时生效（即没有用“#”注释掉），那么只有最后一个配置参数会被设置，而前面的都会被忽略。
- 通过 `reload` 模式设置或修改 GBase 8s 节点配置文件（`postgresql.conf`）的参数，生

效存在短暂延迟，有可能导致配置后 GBase 8s 各实例参数极短时间不一致。

3.8.3 参数参考

- set

只修改配置文件中的参数。

- check

只检查配置文件中的参数。

- reload

表示修改配置文件中的参数，同时发送信号量给数据库进程，使其重新加载配置文件。

- encrypt/generate

生成加密密码文件。

- -N

需要设置的主机名称。

取值范围：已有主机名称。

当参数取值为 ALL 时，表示设置 GBase 8s 中所有的主机。

 说明

需依赖静态文件，仅在 om 安装的情况下有效，编译安装由于没有静态文件，所以无效。

- -I INSTANCE-NAME

需要设置的实例名称。

取值范围：已有实例名称。

当参数取值为 ALL 时，表示设置主机中所有的实例。

 说明

需依赖静态文件，仅在 om 安装的情况下有效，编译安装由于没有静态文件，所以无效。

- -D

需要执行命令的 GBase 8s 实例路径。使用 encrypt 命令时，此参数表示指定的密码文件

生成的路径。

 说明

- 不能与-I 参数一块使用

- -c parameter=value

要设定的 GBase 8s 配置参数的名称和参数值。

 说明

- 如果参数是一个字符串变量，则使用-c parameter="value"或者使用-c "parameter = 'value'"。
- 如果需要配置的 value 内容中包含双引号，则需要在双引号前加上转义符。例如 value 为 a"b"c，则命令为-c "parameter = 'a\"b\"c'"
- 当使用 gs_guc set/reload 为"log_directory"恢复默认值时，其默认值会被置为具体的 data 目录。
- 当使用 gs_guc reload 进行参数设定，并指定-N 参数时，当指定的节点为主节点时，主备节点的值都会被修改；当指定节点为备节点时，只会修改备节点的值，不会修改主节点的值。
- 当使用 gs_guc reload 进行参数设定，未指定-N 参数时，当在主节点上执行时，主备节点的值都会被修改；当在备节点上执行时，只会修改备节点的值，不会修改主节点的值。

取值范围：postgresql.conf 中的所有参数。

- --lcname=LCNAME

要设定的逻辑数据库名称。

 说明

- gs_guc 只允许作用于逻辑数据库的 DN 实例。
- 逻辑数据库允许操作的参数同完整数据库不同。具体差异可参见 \$GAUSSHOME/bin/cluster_guc.conf。

取值范围：已经创建的逻辑数据库名称。

- `--ignore-node=NODES`

需要忽略的主机名称。

 说明

- 该参数必须与 `set/reload` 一起使用。
- 该参数不支持与 `-D` 一起使用。
- 在与 `reload` 一起使用时，如果 `--ignore-node` 没有指定主节点，则集群中所有节点的参数依然会全部同步修改。

- `-c parameter`

当进行 `check` 操作时，表示需要检查的参数名称；当进行 `set/reload` 操作时，参数值不允许为空，为空时不会恢复为数据库参数的默认值。

- `-h host-auth-policy`

指定需要在“`pg_hba.conf`”增加的客户端认证策略。

取值范围：

- `HOSTTYPE DATABASE USERNAME IPADDR IPMASK`

[authmethod-options]

- `HOSTTYPE DATABASE USERNAME IPADDR-WITH-IPMASK`

[authmethod-options]

- `HOSTTYPE DATABASE USERNAME HOSTNAME [authmethod-options]`

`HOSTTYPE` 是必选参数，取值：

- `local`

- `host`

- `hostssl`

- `hostnossl`

其中 local 是一个 Unix 域套接字, host 是普通或 SSL 加密的 TCP/IP 套接字, hostssl 是 SSL 加密的 TCP/IP 套接字, hostnssl 是一个纯 TCP/IP 套接字。

authmethod-options 是可选输入, 取值:

- trust
- reject
- md5
- sha256
- sm3
- cert
- gss

详细的参数说明请参见 pg_hba.conf 配置文件中的描述。

说明

- 指定-h 的情况下, 默认会尝试修改所有节点的 pg_hba 文件, 但是如果是编译安装, 由于没有静态文件, 可以成功修改本节点的 pg_hba 文件, 但是会提示静态文件不存在。

- -?, --help

显示帮助信息。

- -V, --version

显示版本信息。

- -M, --keymode=MODE

设定该密码在数据库运行过程中, 用于服务端(server)、客户端(client)还是创建 Data Source 对象时调用。

说明

- 在使用 ssl 功能时, 会涉及到服务端证书和私钥文件的加密密码和客户端证书和私钥文件的加密密码, 在加密存储加密密码时, 需要指定模式, 否则默认是生成服务

端的密码文件。

取值范围：

- server (缺省)
- client
- source

● -K PASSWORD

指定需要加密的密码。

取值范围：字符串，符合密码复杂度的要求。

● -U, --keyuser=USER

指定要加密的用户，隶属于 OS 用户。

说明

- GBase 8s 可以为每个用户配置不同的 ssl 证书和私钥文件，指定该选项，可以生成按用户名区分的密码文件。

● -S CIPHERKEY

指定需要加密的字符串。

取值范围：字符串。

● -o PREFIX

指定需要输出的 OBS 类型的 cipher 和 rand 文件前缀名称字符串，默认输出文件名前缀为 obsserver。

取值范围：仅支持数字、字母和下划线。

3.8.4 示例

示例 1：修改数据库允许的最大连接数为 800。修改后需要重启数据库才能生效。

```
$ gs_guc set-D /home/gbase/data/dn1/dn1_1 -c "max_connections = 800"  
The gs_guc run with the following arguments: [gs_guc -D  
/home/gbase/data/dn1/dn1_1 -c max_connections = 800 set].
```

```
expected instance path: [/home/gbase/data/dn1/dn1_1/postgresql.conf]
gs_guc reload: max_connections=800:
[/home/gbase/data/dn1/dn1_1/postgresql.conf]
server signaled

Total instances: 1.
Success to perform gs_guc!
```

示例 2: 将已设置的数据库允许的最大连接数值修改为默认值。修改后需要重启数据库才能生效。

```
$ gs_guc set -D /home/gbase/data/dn1/dn1_1 -c "max_connections "
The gs_guc run with the following arguments: [gs_guc -D
/home/gbase/data/dn1/dn1_1 -c max_connections set].
expected instance path: [/home/gbase/data/dn1/dn1_1/postgresql.conf]
gs_guc reload: #max_connections: [/home/gbase/data/dn1/dn1_1/postgresql.conf]
server signaled

Total instances: 1.
Success to perform gs_guc!
```

示例 3: 设置客户端认证策略。

```
$ gs_guc set -N all -I all -h "host replication testuser 10.252.95.191/32 sha256"
The gs_guc run with the following arguments: [gs_guc -N all -I all -h host
replication testuser 10.252.95.191/32 sha256 set ].
Begin to perform the total nodes: 1.
Popen count is 1, Popen success count is 1, Popen failure count is 0.
Begin to perform gs_guc for coordinators.
Cgbaseand count is 1, Cgbaseand success count is 1, Cgbaseand failure count is
0.

Total instances: 1.
ALL: Success to perform gs_guc!
```

示例 4: 注释清理已经设置的客户端认证策略。

```
$ gs_guc set -N all -I all -h "host replication testuser 10.252.95.191/32"
The gs_guc run with the following arguments: [gs_guc -N all -I all -h host
replication testuser 10.252.95.191/32 set ].
Begin to perform the total nodes: 1.
Popen count is 1, Popen success count is 1, Popen failure count is 0.
Begin to perform gs_guc for coordinators.
```

```
Cgbaseand count is 1, Cgbaseand success count is 1, Cgbaseand failure count is 0.
```

```
Total instances: 1.
```

```
ALL: Success to perform gs_guc!
```

示例 5: 加密服务器端存储用户密码的明文, 避免密码泄露。

```
$ gs_guc encrypt -K GBase@123 -D /home/gbase/data
```

```
The gs_guc run with the following arguments: [gs_guc -K -D /home/gbase/data encrypt ].
```

```
gs_guc encrypt -K *
```

示例 6: 检查 GBase 8s 中各个实例的参数配置情况。

```
$ gs_guc check -D /home/gbase/data/dn1/dn1_1 -c "max_connections"
```

```
The gs_guc run with the following arguments: [gs_guc -D /home/gbase/data/dn1/dn1_1 -c max_connections check ].
```

```
expected guc information: gbase: max_connections=NULL:
```

```
[/home/gbase/data/dn1/dn1_1/postgresql.conf]
```

```
gs_guc check: gbase: max_connections=800:
```

```
[/home/gbase/data/dn1/dn1_1/postgresql.conf]
```

```
Total GUC values: 1.
```

```
The value of parameter max_connections is same on all instances.
```

```
max_connections=800
```

示例 7: 通过用户输入的明文密码, 生自定义的 test.key.cipher 和 test.key.rand 文件。

```
$ gs_guc generate -S GBase@123 -D /home/gbase/data/ -o test
```

```
The gs_guc run with the following arguments: [gs_guc -S -D /home/gbase/data/ -o test generate ].
```

```
gs_guc generate -S *
```

3.9 gs_encrypt

3.9.1 背景信息

GBase 8s 提供了 gs_encrypt 工具为输入的明文字符串进行加密操作。

3.9.2 语法

```
gs_encrypt [OPTION]... PLAINTEXT
```



说明

- PLAINTEXT 前不需要加短选项或长选项。
- PLAINTEXT 是需要加密的明文字符串。

3.9.3 参数说明

- `-?, --help`

显示帮助信息。

- `-V, --version`

显示版本号信息。

- `-k _PASSWORD_, --key=PASSWORD`

用户输入的口令，用于派生出加密操作时使用的密钥。口令长度有效范围为 8~16 个字符，并且至少包含大写字母、小写字母、数字和特殊字符中的三种字符。

- `-v _VectorValue_, --vector=VectorValue`

用户输入的盐值，计算口令单向哈希时用于对口令进行加扰。字符串长度必须等于 16。

- `-f FilePrefix, --file-prefix=FilePrefix`

加密操作时使用的 `cipher` 和 `rand` 文件前缀字符串。`cipher` 和 `rand` 文件必须存放在 `$GAUSSHOME/bin` 目录下。

- `-B _Value_, --key-base64=Value`

加密时使用的明文密钥，以 base64 编码。

- `-D _Value_, --vector-base64=Value`

加密时使用的随机数，以 base64 编码。base64 编码前的字符串长度必须为 16。

说明

- 出于安全考虑，建议用户在使用 `gs_encrypt` 工具前设置 `HISTCONTROL=ignorespace`，强制不记录敏感历史命令，然后在执行 `gs_encrypt` 命令行前面加空格。

3.9.4 示例

示例 1：通过输入 `key` 和 `vector` 明文值来对明文字符串进行加密操作。由于每次加密时

使用的 IV 值是随机生成的，因此使用同样的 key 和 vector 以及明文字符串加密后对应的密文串不相同。单次支持加密的明文字符串最大长度为 335 字节。

```
$ gs_encrypt -k password@123 -v 1234567890123456 test_encrypt
G536u0mqnZb7bohzy0j3dKIBFx1qvoo88EIUzZHpeLQxMjMONTY30DkwMTIzNDU2
```

示例 2: 已经存在 cipher 和 rand 文件，通过文件前缀来对明文字符串进行加密操作。

生成 cipher 和 rand 文件。

```
$ gs_guc generate -S GBase@123 -D $GAUSSHOME/bin -o test
The gs_guc run with the following arguments: [gs_guc -S -D
/home/gbase/gbase_db/app/bin -o test generate ].
gs_guc generate -S *
```

执行加密操作。

```
$ gs_encrypt -f test test_encrypt
IhxvguK1Q4R5FtX0XbrGP0790HfQbHpNC2oITPg0+P81aROMS2GgTo2fQTpj/Wvg
```

示例 3: 通过输入 base64 加密后的 key 和 vector 密文来对明文字符串进行加密操作。

输入 base64 加密后的 key。

```
python
>>> import base64
>>> print base64.b64encode("XXXXXXXXXX");
R2F1c3NAMTIz
```

输入 base64 加密后 vector 密文。

```
>>> print base64.b64encode("1234567890abcdef");
MTIzNDU2Nzg5MGFiY2RlZg==
```

执行加密操作。

```
$ gs_encrypt -B R2F1c3NAMTIz -D MTIzNDU2Nzg5MGFiY2RlZg== test_encrypt
V7mpnsfIHTjVYtYgW1YHI0c6aYON4J5L7BtU4CqDoeIkxMjMONTY30DkwYWJjZGVm
```

3.10 gs_om

3.10.1 背景信息

GBase 8s 提供了 gs_om 工具，帮助用户对 GBase 8s 实例进行维护操作，包括启动实例、停止实例、查询实例状态、查询静态配置、生成静态配置文件、查询实例状态详细信息、生成动态配置文件、SSL 证书替换、显示帮助信息和显示版本号信息等功能。

前提条件

需以操作系统用户 `gbase` 执行 `gs_om` 命令。

3.10.2 语法

- 启动实例

```
gs_om -t start [-h HOSTNAME] [-D dataDir] [--time-out=SECS] [--security-mode=MODE]
[--cluster-number=None] [-l LOGFILE]
```

- 停止实例

```
gs_om -t stop [-h HOSTNAME] [-D dataDir] [--time-out=SECS] [-m MODE] [-l LOGFILE]
```

- 重启实例

```
gs_om -t restart [-h HOSTNAME] [-D dataDir] [--time-out=SECS]
[--security-mode=MODE] [-l LOGFILE] [-m MODE]
```

- 查询实例状态

```
gs_om -t status [-h HOSTNAME] [-o OUTPUT] [--detail] [--all] [-l LOGFILE]
```

- 查询 GBase 8s 状态详细信息

```
gs_om -t query [-o OUTPUT]
```

- SSL 证书替换

```
gs_om -t cert --cert-file=CERTFILE [-l LOGFILE]
gs_om -t cert --rollback
```

- 开启、关闭数据库内 kerberos 认证

```
gs_om -t kerberos -m [install|uninstall] -U USER [-l LOGFILE]
[--krb-client|--krb-server]
```

- 显示帮助信息

```
gs_om -? | --help
```

- 显示版本号信息

```
gs_om -V | --version
```

3.10.3 参数说明

gs_om 参数可以分为如下几类：

通用参数

- -t

gs_om 命令的类型。

取值范围：start、stop、status、generateconf、cert、view、query、refreshconf、kerberos。

- -l

指定日志文件及存放路径。

默认值：\$GAUSSLOG/om/gs_om-YYYY-MM-DD_hhmmss.log（virtualip 的默认值：
/tmp/gs_virtualip/gs_om-YYYY-MM-DD_hhmmss.log）

- -?, --help

显示帮助信息。

- -V, --version

显示版本号信息。

启动 GBase 8s 参数：

- -h

指定需要启动的服务器名称。一次只能启动一个服务器。

取值范围：服务器名称。

不指定服务器名称时，表示启动 GBase 8s。

- -D

指定数据库节点路径。

取值范围：数据库节点路径。

不指定数据库节点路径，表示使用静态文件中的数据库节点路径。



说明

- 针对部署了 CM 工具的集群，如果在对单个实例进行启停时候指定了 -D 参数，只会启动或者停止数据库进程，不会影响 CM 相关进程。
- 假设当前实例的数据库和 CM 进程都停止，在启动该实例时候如果指定了 -D 参数，那么只会尝试拉起数据库进程，此时由于 CM 进程不存在会导致数据库进程也拉起失败。可以去掉 -D 参数，启动整个实例。

- --time-out=SECS

指定超时时间，如果超时，om 脚本自动退出。单位：s。

取值范围：正整数，建议值 300。

默认值：300。

- --security-mode

指定是否以安全模式启动数据库。

取值范围：

- on 以安全模式启动。
- off 不以安全模式启动，默认不开启安全模式。

- --cluster-number

指定启动时内核运行版本号。

取值范围：内核版本号。

停止 GBase 8s 参数：

- -h

指定需要停止实例所在的服务器名称。一次只能停止一个服务器。

取值范围：实例所在的服务器名称。

不指定服务器名称时，表示停止 GBase 8s。

- -m, --mode=MODE

停止模式设置。

取值范围：支持两种停止模式。

- fast 方式：保证有主备关系的实例数据是一致的。
- immediate 方式：不保证有主备关系的实例数据是一致的。

默认值：fast 方式。

- --time-out=SECS

指定超时时间，如果超时，om 脚本自动退出。单位：s。

取值范围：正整数，建议值 300。

默认值：300。

查询状态参数

GBase 8s 状态显示结果的参数说明请参见表 3-10。

- -h

指定需要待查询的服务器名称。

取值范围：服务器名称。

不指定服务器时，表示查询 GBase 8s。

- --az

指定需要查询的 AZ 名称，比如--az=AZ1。

取值范围：AZ 名称，不指定 AZ 名称时，表示查询所有 AZ。

- -o

输出到指定的 output 文件中。

如果不指定，直接显示在屏幕上。

- --detail

显示详细信息。如果不指定，只提示该服务器是否正常。

说明

- 本参数通过在每个数据库节点执行 gs_ctl query 命令进行查询并汇总结果，来获取实例的详细信息。

- --all

显示 GBase 8s 所有节点信息。

 说明

- 本参数通过在每个数据库节点执行 gsql 查询系统表并汇总结果，来获取集群所有节点的信息。为保证显示信息的正确性，在使用本参数前请确认 gsql 可以查询。

生成配置文件参数

- -X

选择 GBase 8s 配置文件路径。

取值范围：clusterconfig.xml 的路径。

- --old-values=old

需要修改的静态文件中的旧值。

取值范围：port、ip、hostname，可以输入多个值，用逗号隔开。

- --new-values=new

需要修改的静态文件中的新值。

取值范围：port、ip、hostname，可以输入多个值，用逗号隔开，与 old-values 一一对应。

- --distribute

将静态配置文件发布到 GBase 8s 实例的安装目录。

开启、关闭数据库内 kerberos 认证

- -U

指定数据库部署用户。

取值范围：数据库部署用户。

- -m

指定所要进行的操作。

取值范围：

- install: 开启数据库内 kerberos 认证。
- uninstall: 关闭数据库内 kerberos 认证。
- --krb-server 安装 kerberos 服务端认证。
- --krb-client 安装 kerberos 客户端认证。

注: 需要先安装--krb-server, 卸载时同步卸载不需要该参数。

SSL 证书替换

- --cert-file

指定本地 SSL 证书 zip 压缩包所在路径。

- --rollback

证书回退到上一次成功备份的状态。

- -L

制定该参数时, 仅对当前节点进行。

表 3-10 状态说明

字段	字段含义	字段值
cluster_state	显示实例状态。	<ul style="list-style-type: none"> ● Normal: 表示 GBase 8s 可用, 且数据有冗余备份。所有进程都在运行, 主备关系正常。 ● Unavailable: 表示 GBase 8s 不可用。 ● Degraded: 表示 GBase 8s 可用, 但数据没有冗余备份。
redistributing	数据重分布状态。	<ul style="list-style-type: none"> ● Yes: 表示 GBase 8s 处于数据重分布状态。 ● No: 表示 GBase 8s 未处于数据重分布状态。
balanced	平衡状态。显示是否有	<ul style="list-style-type: none"> ● Yes: 表示 GBase 8s 处于负载均衡状态。 ● No: 表示 GBase 8s 未处于负载均衡状态。

字段	字段含义	字段值
	GBase 8s 实例发生过主备切换而导致主机负载不均衡。	
node	主机名称。	表示该实例所在的主机名称。多 AZ 时会显示 AZ 编号。
node_ip	主机 IP。	表示该实例所在的主机 IP。
instance	实例 ID。	表示该实例的 ID。
state	实例状态。	<ul style="list-style-type: none"> ● P: 节点的初始角色是 Primary, 数据库安装后就不再变动, 从系统静态文件读取。 ● S: 节点的初始角色是 Standby, 数据库安装后就不再变动, 从系统静态文件读取。 ● C: 节点的初始角色是 Cascade Standby, 数据库安装后就不再变动, 从系统静态文件读取。 ● Primary: 表示实例为主实例。 ● Standby: 表示实例为备实例。 ● Cascade Standby: 表示实例为级联备实例。 ● Pending: 表示该实例在仲裁阶段。 ● Unknown: 表示实例状态未知。 ● Down: 表示实例处于宕机状态。

表 3-10 特性 ID 说明

特性名称	特性 ID	GBase 8s 产品

特性名称	特性 ID	GBase 8s 产品
多值列	0	基础版
JSON	1	不支持 License 控制
XML	2	不支持
数据存储格式 ORC	3	基础版
单机支持一主多备，备机可读	5	策略不放开
多维碰撞分析 GPU 加速	7	高级特性
全文索引	8	基础版
Extension_CONNECTOR	9	基础版
EXPRESS_CLUSTER	12	高级版
跨 DC 协同	13	高级版
图	14	不支持
时序	15	不支持
PostGis 对接	16	基础版

特性名称	特性 ID	GBase 8s 产品
GBase 8s 内高可用—主多备	17	策略不放开
行级权限控制	18	高级版
透明加密	19	高级版
私有表	20	高级版

3.10.4 示例

- 启动实例。

```
$ gs_om -t start
Starting cluster.
=====
[SUCCESS] gbase8s_5_124:
[2022-12-29 14:51:54.190][28320][][gs_ctl]: gs_ctl started,datadir is
/home/gbase/data/dn1/dn1_1
[2022-12-29 14:51:54.210][28320][][gs_ctl]: another server might be running;
Please use the restart command
=====
Successfully started.
```

- 停止实例。

```
$ gs_om -t stop
Stopping cluster.
=====
Successfully stopped cluster.
=====
End stop cluster.
```

- 查看集群详细状态信息。

```
$ gs_om -t status --detail
[ Cluster State ]
```

```

cluster_state      : Normal
redistributing     : No
current_az         : AZ_ALL

[ Datanode State(group id: 2) ]

   node      node_ip      port      instance
state
-----
1 gbase8s_5_124 192.168.5.124 115400    6001 /home/gbase/data/dn1/dn1_1
P Primary Normal

```

- 在 GBase 8s 上执行如下命令，生成配置文件。

```

gs_om -t generateconf -X /opt/software/gbase/clusterconfig.xml --distribute
Generating static configuration files for all nodes.
Creating temp directory to store static configuration files.
Successfully created the temp directory.
Generating static configuration files.
Successfully generated static configuration files.
Static configuration files for all nodes are saved in
/opt/gbase/Bigdata/gaussdb/wisquery/script/static_config_files.
Distributing static configuration files to all nodes.
Successfully distributed static configuration files.

```

- 在 GBase 8s 上执行如下命令，修改配置文件，动态配置文件需要执行 `gs_om -t refreshconf` 手动刷新，实际端口或者 ip 生效需要手动修改 `guc` 参数。

```

gs_om -t generateconf --old-values=26000,192.168.1.1
--new-values=36000,192.168.1.2 --distribute
Generating static configuration files for all nodes.
Creating temp directory to store static configuration files.
Successfully created the temp directory.
Generating static configuration files.
Successfully generated static configuration files.
Static configuration files for all nodes are saved in
/opt/huawei/Bigdata/gaussdb/wisquery/script/static_config_files.
Distributing static configuration files to all nodes.
Successfully distributed static configuration files.

```

- 然后打开生成的配置文件目录，会看到新生成的 3 个文件。

```
cd /opt/gbase/Bigdata/gaussdb/wisquery/script/static_config_files
ll
total 456
-rwxr-xr-x 1 gbase gbase 155648 2016-07-13 15:51 cluster_static_config_plat1
-rwxr-xr-x 1 gbase gbase 155648 2016-07-13 15:51 cluster_static_config_plat2
-rwxr-xr-x 1 gbase gbase 155648 2016-07-13 15:51 cluster_static_config_plat3
```

- SSL 证书回退。

```
gs_om -t cert --rollback
[plat1] SSL cert files rollback successfully.
[plat2] SSL cert files rollback successfully.
[plat3] SSL cert files rollback successfully.
```

- 新 License 注册。

```
gs_om -t license -m register -f MTgsMTkK
Preparing for the program initialization.
Lock the OPS operation of OM components.
Check and make sure the consistency of the license file.
Backup the license file on all of the cluster hosts.
Encrypt the product feature information and generate the license file.
Send message to the 数据库节点 instances to reload the license file.
Remove the backup license file on all of the cluster hosts.
License register successfully.
```

- 新 License 反注册。

```
gs_om -t license -m unregister -f MTgsMTkK
Preparing for the program initialization.
Lock the OPS operation of OM components.
Check and make sure the consistency of the license file.
Backup the license file on all of the cluster hosts.
Encrypt the product feature information and generate the license file.
Send message to the 数据库节点 instances to reload the license file.
Remove the backup license file on all of the cluster hosts.
License unregister successfully.
```

3.11 gs_plan_simulator

3.11.1 背景信息

某些用户的生产环境不允许维护人员进行任何问题定位和分析的操作，因此对于此类场景，唯一的手段就是将生产环境的各种数据收集后，依据这些数据对问题进行分析和定位。对于和执行计划相关的问题，gs_plan_simulator 可以收集与执行计划相关的数据并能够在其它环境上进行执行计划的复现，从而定位执行计划类相关问题。

前提条件

- 用户数据库正常运行。
- 用户可以连接到数据库并正常执行功能。
- 用于复现的 OS 环境和生产环境应该一致。
- 用于复现的环境安装的 GBase 8s 安装包应和生产环境的安装包一致。

注意事项

收集数据功能目前已经整合到工具 gs_collector 中，需要使用 gs_collector 收集相关数据，之后再使用 gs_plan_simulator 复现执行计划。

使用 gs_collector 收集数据时默认不会收集与执行计划相关数据，因为收集该类数据可能会需要很长时间。收集执行计划相关数据需要修改配置文件，详情见使用步骤。

在复现环境中需要配置 GAUSSHOME 和 LD_LIBRARY_PATH。

工具使用时需要注意集群状态，扩容状态等影响执行计划相关数据的情况下，勿使用此工具。

3.11.2 使用步骤

(1) 使用 gs_collector 收集数据。收集执行计划相关数据时，需要先写配置文件，例如 plan.json，

建议配置文件内容如下：

```
{
  "Collect":
  [
    {"TypeName": "Plan", "Content": "*", "Interval": "0", "Count": "1"}
  ]
}
```

```
}
```

这里 Content 内容为“*”，用于收集所有数据库的相关数据，也可以指定数据库进行收集如“test1,test2”。

收集命令示例：

```
$ gs_collector --begin-time="20191219 15:00" --end-time="20191219 15:15"  
-C ../plan.json -l plan.log -o ./
```

更多使用方法见 [gs_collector](#) 一节。

(2) 拷贝数据文件夹到复现环境。

数据收集成功后，放在指定的压缩包内，解压压缩包会有各个节点的压缩包。解压任意一个压缩包，目标数据在 planSimulatorfiles 文件夹下，进入 planSimulatorfiles 文件夹可以看到以数据库命名的文件夹，拷贝 planSimulatorfiles 文件夹到复现环境。

(3) 复现环境的配置。

- ① 使用普通用户创建用于复现执行计划的目录。例如 plan 文件夹。
- ② 在该目录中解压 GBase 8s 安装包，再解压安装包解压出来的 bin 文件。
- ③ 配置 GAUSSHOME 为 plan 文件夹的路径，配置 LD_LIBRARY_PATH 为 plan/lib 文件夹的路径。

(4) 使用 gs_plan_simulator 复现执行计划。

在复现环境中运行如下命令，复现命令示例：

```
$ $GAUSSHOME/bin/gs_plan_simulator.sh -m restore -p 15400 -A planSimulatorfiles/
```

该命令会在复现环境中创建一个数据库，端口号为 15400，同时会有信息提示如何连接数据库，进入数据库就可以执行 explain+SQL 语句进行问题定位。

3.11.3 语法

- 复现数据库环境。

```
$GAUSSHOME/bin/gs_plan_simulator.sh -m OPTION -p PORT -A DATADIR
```

- 关闭数据库。

```
$GAUSSHOME/bin/gs_plan_simulator.sh -m off
```

- 打开数据库。

```
$GAUSSHOME/bin/gs_plan_simulator.sh -m start
```

- 显示帮助信息。

```
$GAUSSHOME/bin/gs_plan_simulator.sh --help
```

3.11.4 参数说明

- -m

工具的功能信息，表示工具要执行的操作。

取值范围：

- restore：利用收集的数据库信息来复现数据库环境，同时创建数据库。
- off：关闭已经存在的由该工具复现数据库。
- start：启动由该工具复现的数据库。

- -p

创建数据库的 CN 端口。

- -A

从用户环境拷贝的数据文件夹。

3.12 gs_restore

3.12.1 背景信息

gs_restore 是 GBase 8s 提供的针对 gs_dump 导出数据的导入工具。通过此工具可将由 gs_dump 生成的导出文件进行导入。

gs_restore 工具由操作系统用户 gbase 执行。

主要功能包含：

- 导入到数据库

如果连接参数中指定了数据库，则数据将被导入到指定的数据库中。其中，并行导入必须指定连接的密码。导入时生成列会自动更新，并像普通列一样保存。

- 导入到脚本文件

如果未指定导入数据库，则创建包含重建数据库所必须的 SQL 语句脚本并写入到文件或者标准输出。等效于直接使用 `gs_dump` 导出为纯文本格式。

3.12.2 命令格式

```
gs_restore [OPTION]... FILE
```

说明

- FILE 没有短选项或长选项。用来指定归档文件所处的位置。
- 作为前提条件，需输入 `dbname` 或 `-l` 选项。不允许用户同时输入 `dbname` 和 `-l` 选项。
- `gs_restore` 默认是以追加的方式进行数据导入。为避免多次导入造成数据异常，在进行导入时，建议使用“`-c`”参数，在重新创建数据库对象前，清理（删除）已存在于将要还原的数据库中的数据库对象。
- 日志打印无开关，若需隐藏日志，请将日志重定向到日志文件。若恢复表数据时，数据量很大，会分批恢复，因此会多次出现“表数据已完成导入”的日志。

3.12.3 参数说明

通用参数

- `-d, --dbname=NAME`

连接数据库 `dbname` 并直接导入到该数据库中。

- `-f, --file=FILENAME`

指定生成脚本的输出文件，或使用 `-l` 时列表的输出文件。

默认是标准输出。

说明

- `-f` 不能同 `-d` 一起使用。

- `-F, --format=c|d|t`

指定归档格式。由于 `gs_restore` 会自动决定格式，因此不需要指定格式。

取值范围：

- `c/custom`：该归档形式为 `gs_dump` 的自定义格式。
 - `d/directory`：该归档形式是一个目录归档形式。
 - `t/tar`：该归档形式是一个 `tar` 归档形式。
- `-l, --list`

列出归档形式内容。这一操作的输出可用作 `-L` 选项的输入。注意如果像 `-n` 或 `-t` 的过滤选项与 `-l` 使用，过滤选项将会限制列举的项目（即归档形式内容）。

- `-v, --verbose`

指定 `verbose` 模式。

- `-V, --version`

打印 `gs_restore` 版本，然后退出。

- `-, --help`

显示 `gs_restore` 命令行参数帮助，然后退出。

导入参数

- `-a, --data-only`

只导入数据，不导入模式（数据定义）。`gs_restore` 的导入是以追加方式进行的。

- `-c, --clean`

在重新创建数据库对象前，清理（删除）已存在于将要还原的数据库中的数据库对象。

- `-C, --create`

导入到数据库之前会先使用 `CREATE DATABASE` 创建数据库（指定该选项后，`-d` 指定的数据库仅用以执行 `CREATE DATABASE` 命令，所有数据将被导入到创建的数据库中）。

- `-e, --exit-on-error`

当发送 `SQL` 语句到数据库时如果出现错误，请退出。默认状态下会继续，且在导入后会显示一系列错误信息。

- `-I, --index=NAME`

只导入已列举的 `index` 的定义。允许导入多个 `index`。如果多次输入 `-I index` 导入多个 `index`。

例如：

```
gs_restore -h 192.168.5.124 -p 15400 -d postgres -I Index1 -I Index2  
/home/gbase/data/backup/MPPDB_backup.tar
```

在上面这个例子中，`Index1` 和 `Index2` 会被导入。

- `-j, --jobs=NUM`

运行 `gs_restore` 最耗时的部分（如加载数据、创建 `index` 或创建约束）使用并发任务。该选项能大幅缩短导入时间，即将一个大型数据库导入到某一多处理器的服务器上。

每个任务可能是一个进程或一个线程，这由操作系统决定；每个任务与服务器进行单独连接。

该选项的最优值取决于服务器的硬件设置、客户端以及网络。还包括这些因素，如 CPU 核数量、硬盘设置。建议是从增加服务器上的 CPU 核数量入手，更大的值（服务器上 CPU 核数量）在很多情况下也能导致数据文件更快的被导入。当然，过高的值会由于超负荷反而导致性能降低。

该选项只支持自定义归档格式。输入文件必须是常规文件（不能是像 `pipe` 的文件）。如果是通过脚本文件，而非直接连接数据库服务器，该选项可忽略。而且，多任务不能与 `--single-transaction` 选项一起使用。

- `-L, --use-list=FILENAME`

只导入列举在 `list-file` 中的那些归档形式元素，导入顺序以它们在文件中的顺序为准。注意如果像 `-n` 或 `-t` 的过滤选项与 `-L` 使用，它们将会进一步限制导入的项目。

一般情况下，`list-file` 是通过编辑前面提到的某个 `-l` 参数的输出创建的。文件行的位置可更改或直接删除，也可使用分号（`;`）在行的开始注出。

- `-n, --schema=NAME`

只导入已列举的模式中的对象。

该选项可与 `-t` 选项一起用以导入某个指定的表。

多次输入 `-n _schemaname_` 可以导入多个模式。

例如：

```
gs_restore -h 192.168.5.124 -p 15400 -d postgres -n sch1 -n sch2  
/home/gbase/data/backup/MPPDB_backup.tar
```

在上面这个例子中，sch1 和 sch2 会被导入。

- **-O, --no-owner**

不输出设置对象的归属这样的命令，以匹配原始数据库。默认情况下，gs_restore 会发出 ALTER OWNER 或 SET SESSION AUTHORIZATION 语句设置所创建的模式元素的所属。除非是由系统管理员(或是拥有脚本中所有对象的同一个用户)进行数据库首次连接的操作，否则语句会失败。使用-O 选项，任何用户名都可用于首次连接，且该用户拥有所有已创建的对象。

- **-P, --function=NAME(args)**

只导入已列举的函数。请按照函数所在转储文件中的目录，准确拼写函数名称和参数。

当-P 单独使用时，表示导入文件中所有'function-name(args)'函数；当-P 同-n 一起使用时，表示导入指定模式下的'function-name(args)'函数；多次输入-P，而仅指定一次-n，表示所有导入的函数默认都是位于-n 模式下的。

可以多次输入-n schema-name -P 'function-name(args)'同时导入多个指定模式下的函数。

例如：

```
gs_restore -h 192.168.5.124 -p 15400 -d postgres -n test1 -P 'Func1(integer)' -n  
test2 -P 'Func2(integer)' /home/gbase/data/backup/MPPDB_backup.tar
```

在上面这个例子中，test1 模式下的函数 Func1(i integer)和 test2 模式下的函数 Func2(j integer)会被一起导入。

- **-s, --schema-only**

只导入模式（数据定义），不导入数据（表内容）。当前的序列值也不会导入。

- **-S, --sysadmin=NAME**

该参数为扩展预留接口，不建议使用。

- **-t, --table=NAME**

只导入已列举的表定义、数据或定义和数据。该选项与-n 选项同时使用时，用来指定某个模式下的表对象。-n 参数不输入时，默认为 PUBLIC 模式。多次输入-n -t 可以导入指定模式下的多个表。

例如，在 192.168.5.124 节点进行导入操作。

导入 PUBLIC 模式下的 table1

```
gs_restore -h 192.168.5.124 -p 15400 -d postgres -t table1  
/home/gbase/data/backup/MPPDB_backup.tar
```

导入 test1 模式下的 test1 和 test2 模式下 test2

```
gs_restore -h 192.168.5.124 -p 15400 -d postgres -n test1 -t test1 -n test2 -t  
test2 /home/gbase/data/backup/MPPDB_backup.tar
```

导入 PUBLIC 模式下的 table1 和 test1 模式下 table1

```
gs_restore -h 192.168.5.124 -p 15400 -d postgres -n PUBLIC -t table1 -n test1 -t  
table1 /home/gbase/data/backup/MPPDB_backup.tar
```

说明

- -t 参数不支持 schema_name.table_name 的输入格式。
- -T, --trigger=NAME
该参数为扩展预留接口。
- -x, --no-privileges/--no-acl
防止导入访问权限（GRANT/REVOKE 命令）。
- -l, --single-transaction
执行导入作为一个单独事务（即把命令包围在 BEGIN/CgbaseIT 中）。
该选项确保要么所有命令成功完成，要么没有改变应用。该选项意为--exit-on-error。
- --disable-triggers
该参数为扩展预留接口，不建议使用。
- --no-data-for-failed-tables
默认状态下，即使创建表的命令失败（如表已经存在），表数据仍会被导入。使用该选项，像这种表的数据会被跳过。如果目标数据库已包含想要的表内容，这种行为会有帮助。
该选项只有在直接导入到某数据库中时有效，不针对生成 SQL 脚本文件输出。
- --no-publications

不导入发布。

- `--no-security-labels`

该参数为扩展预留接口，不建议使用。

- `--no-subscriptions`

不导入订阅。

- `--no-tablespaces`

不输出选择表空间的命令。使用该选项，无论默认表空间是哪个，在导入过程中所有对象都会被创建。

- `--section=SECTION`

导入已列举的区段（如 `pre-data`、`data` 或 `post-data`）。

- `--use-set-session-authorization`

该选项用来进行文本格式的备份。

输出 `SET SESSION AUTHORIZATION` 命令，而非 `ALTER OWNER` 命令，用以决定对象归属。该选项使转储更加兼容标准，但通过参考转储中对象的记录，导入过程可能会有问题。使用 `SET SESSION AUTHORIZATION` 的转储要求必须是系统管理员，同时在导入前还需参考“`SET SESSION AUTHORIZATION`”，手工对导出文件的密码进行修改验证，只有这样才能进行正确的导入操作，相比之下，`ALTER OWNER` 对权限要求较低。

- `--pipeline`

使用管道传输密码，禁止在终端使用。

须知

- 如果安装过程中有任何本地数据要添加到 `template1` 数据库，请谨慎将 `gs_restore` 的输出载入到一个真正的空数据库中；否则可能会因为被添加对象的定义被复制，而出现错误。

- `CREATE DATABASE foo WITH TEMPLATE template0;`

`gs_restore` 不能选择性地导入大对象；例如只能导入那些指定表的对象。如果某个归档形式包含大对象，那所有大对象都会被导入或一个都不会被导入，如果它们通过 `-L`、`-t` 或其

他选项被排除。

说明

- `-d/--dbname` 和 `-f/--file` 不能同时使用；
- `-s/--schema-only` 和 `-a/--data-only` 不能同时使用；
- `-c/--clean` 和 `-a/--data-only` 不能同时使用；
- 使用 `--single-transaction` 时，`-j/--jobs` 必须为单任务；
- `--role` 和 `--rolepassword` 必须一起使用。

连接参数

- `-h, --host=HOSTNAME`

指定的主机名称。如果取值是以斜线开头，他将用作 Unix 域套接字的目录。默认值取自 `PGHOST` 环境变量；如果没有设置，将启动某个 Unix 域套接字建立连接。

该参数只针对 GBase 8s 外，对 GBase 8s 内本机只能用 127.0.0.1。

- `-p, --port=PORT`

指定服务器所侦听的 TCP 端口或本地 Unix 域套接字后缀，以确保连接。默认值设置为 `PGPORT` 环境变量。

在开启线程池情况下，建议使用 `pooler port`，即侦听端口+1。

- `-U, --username=NAME`

所连接的用户名。

- `-w, --no-password`

不出现输入密码提示。如果服务器要求密码认证并且密码没有通过其它形式给出，则连接尝试将会失败。该选项在批量工作和不存在用户输入密码的脚本中很有帮助。

- `-W, --password=PASSWORD`

指定用户连接的密码。如果主机的认证策略是 `trust`，则不会对系统管理员进行密码验证，即无需输入 `-W` 参数；如果没有 `-W` 参数，并且不是系统管理员，“`gs_restore`”会提示用户输入密码。

- `--role=ROLENAME`

指定导入操作使用的角色名。选择该参数, 会使 `gs_restore` 连接数据库后, 发起一个 `SET ROLE` 角色名命令。当所授权用户 (由 `-U` 指定) 没有 `gs_restore` 要求的权限时, 该参数会起到作用, 即切换到具备相应权限的角色。某些安装操作规定不允许直接以初始用户身份登录, 而使用该参数能够在不违反该规定的情况下完成导入。

- `--rolepassword=ROLEPASSWORD`

指定具体角色用户的角色密码。

3. 12. 4 示例

特例: 执行 `gsq1` 程序, 使用如下选项导入由 `gs_dump/gs_dumpall` 生成导出文件夹 (纯文本格式) 的 `MPPDB_backup.sql` 文件到 `postgres` 数据库。

```
$ gsq1 -d postgres -p 15400 -W Bigdata@123 -f /home/gbase/data/MPPDB_backup.sql
SET
SET
SET
SET
SET
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
CREATE INDEX
CREATE INDEX
CREATE INDEX
SET
CREATE INDEX
REVOKE
REVOKE
GRANT
GRANT
total time: 30476 ms
```

`gs_restore` 用来导入由 `gs_dump` 生成的导出文件。

示例 1: 执行 `gs_restore`, 将导出的 `MPPDB_backup.dmp` 文件 (自定义归档格式) 导入到 `postgres` 数据库。


```
$ gs_restore -W Bigdata@123 /home/gbase/data/backup/MPPDB_backup.dmp -p 15400
-d postgres
gs_restore: restore operation successful
gs_restore: total time: 13053 ms
```

示例 2: 执行 `gs_restore`, 将导出的 `MPPDB_backup.tar` 文件 (tar 格式) 导入到 postgres 数据库。

```
$ gs_restore /home/gbase/data/backup/MPPDB_backup.tar -p 15400 -d postgres
gs_restore[2017-07-21 19:16:26]: restore operation successful
gs_restore[2017-07-21 19:16:26]: total time: 21203 ms
```

示例 3: 执行 `gs_restore`, 将导出的 `MPPDB_backup` 文件 (目录格式) 导入到 postgres 数据库。

```
$ gs_restore /home/gbase/data/backup/MPPDB_backup -p 15400 -d postgres
gs_restore[2017-07-21 19:16:26]: restore operation successful
gs_restore[2017-07-21 19:16:26]: total time: 21003 ms
```

示例 4: 执行 `gs_restore`, 使用自定义归档格式的 `MPPDB_backup.dmp` 文件来进行如下导入操作。导入 `PUBLIC` 模式下所有对象的定义和数据。在导入时会先删除已经存在的对象, 如果原对象存在跨模式的依赖则需手工强制干预。

```
$ gs_restore /home/gbase/data/backup/MPPDB_backup.dmp -p 15400 -d postgres -e
-c -n PUBLIC
gs_restore: [archiver (db)] Error while PROCESSING TOC:
gs_restore: [archiver (db)] Error from TOC entry 313; 1259 337399 TABLE table1
gaussdba
gs_restore: [archiver (db)] could not execute query: ERROR: cannot drop table
table1 because other objects depend on it
DETAIL: view t1.v1 depends on table table1
HINT: Use DROP ... CASCADE to drop the dependent objects too.
Cgbaseand was: DROP TABLE public.table1;
```

手工删除依赖, 导入完成后再重新创建。

```
$ gs_restore /home/gbase/data/backup/MPPDB_backup.dmp -p 15400 -d postgres -e
-c -n PUBLIC
gs_restore[2017-07-21 19:16:26]: restore operation successful
gs_restore[2017-07-21 19:16:26]: total time: 2203 ms
```

示例 5: 执行 `gs_restore`, 使用自定义归档格式的 `MPPDB_backup.dmp` 文件来进行如下导入操作。只导入 `PUBLIC` 模式下表 `table1` 的定义。

```
$ gs_restore /home/gbase/data/backup/MPPDB_backup.dmp -p 15400 -d postgres -e  
-c -s -n PUBLIC -t table1  
gs_restore[2017-07-21 19:16:26]: restore operation successful  
gs_restore[2017-07-21 19:16:26]: total time: 21000 ms
```

示例 6: 执行 `gs_restore`, 使用自定义归档格式的 `MPPDB_backup.dmp` 文件来进行如下导入操作。只导入 `PUBLIC` 模式下表 `table1` 的数据。

```
$ gs_restore /home/gbase/data/backup/MPPDB_backup.dmp -p 15400 -d postgres -e  
-a -n PUBLIC -t table1  
gs_restore[2017-07-21 19:16:26]: restore operation successful  
gs_restore[2017-07-21 19:16:26]: total time: 20203 ms
```

3.12.5 相关命令

[gs_dump](#), [gs_dumpall](#)

3.13 gs_ssh

3.13.1 背景信息

GBase 8s 提供了 `gs_ssh` 工具, 帮助用户在各节点上执行相同的命令。

注意事项

- `gs_ssh` 只可以执行当前数据库用户有权限执行的命令。
- `gs_ssh` 所执行的命令不会对当前执行的会话产生影响, 比如类似 `cd` 或 `source` 的命令, 只会在执行的进程环境中产生影响, 而不会影响到当前执行的会话环境。

前提条件

- 各个主机间互信正常。
- GBase 8s 已经正确安装部署。
- 调用命令可用 `which` 查询到且在当前用户下有执行权限。
- 需以操作系统用户 `gbase` 执行 `gs_ssh` 命令。
- 集群 GTM 节点不使用同主机, CN、DCS 节点同样。

3.13.2 语法

- 同步执行命令

```
gs_ssh -c cmd
```

- 显示帮助信息

```
gs_ssh -? | --help
```

- 显示版本号信息

```
gs_ssh -V | --version
```

3.13.3 参数说明

- -c

指定需要在 GBase 8s 各主机上执行的 linux shell 命令名。

- -?, --help

显示帮助信息。

- -V, --version

显示版本号信息。

3.13.4 示例

在 GBase 8s 各主机上执行相同命令。以执行 hostname 命令为例。

```
$ gs_ssh -c "hostname"
Successfully execute command on all nodes.
Output:
[SUCCESS] plat1:
plat1
[SUCCESS] plat2:
plat2
[SUCCESS] plat3:
plat3
[SUCCESS] plat4:
plat4
```

3.14 gs_sdr

3.14.1 背景信息

GBase 8s 提供了 `gs_sdr` 工具，在不借助额外存储介质的情况下实现跨 region 的异地容灾。提供流式容灾搭建，容灾升主，计划内主备切换，容灾解除、容灾状态监控功能、显示帮助信息和显示版本号信息等功能。

3.14.2 前提条件

需以操作系统用户 `gbase` 执行 `gs_sdr` 命令。

3.14.3 语法

- 容灾搭建

```
gs_sdr -t start -m [primary|disaster_standby] [-U DR_USERNAME] [-W DR_PASSWORD] [-X XMLFILE] [--json JSONFILE] [--time-out=SECS] [-l LOGFILE]
```

- 容灾升主

```
gs_sdr -t failover [-l LOGFILE]
```

- 计划内主备切换

```
gs_sdr -t switchover -m [primary|disaster_standby] [--time-out=SECS] [-l LOGFILE]
```

- 容灾解除

```
gs_sdr -t stop [-X XMLFILE] [--json JSONFILE] [-l LOGFILE]
```

- 容灾状态监控

```
gs_sdr -t query [-l LOGFILE]
```

3.14.4 参数说明

`gs_sdr` 参数可以分为如下几类：

- 通用参数：

- `-t`

`gs_sdr` 命令的类型。

取值范围：start、failover、switchover、stop、query。

- -l

指定日志文件及存放路径。

默认值：\$GAUSSLOG/om/gs_sdr-YYYY-MM-DD_hhmmss.log

- -?, -help

显示帮助信息。

- -V, -version

显示版本号信息。

- 搭建容灾参数：

- -m

期望该集群在容灾关系中成为的角色

取值范围：primary(主集群)或 disaster_standby(灾备集群)

- -U

具有流复制权限的容灾用户名称

- -W

容灾用户密码

说明：

- 搭建容灾关系前，主集群需创建容灾用户，用于容灾鉴权，主备集群必须使用相同的容灾用户名和密码，一次容灾搭建后，该用户密码不可修改。若需修改容灾用户名与密码，需要解除容灾，使用新的容灾用户重新进行搭建。容灾用户密码中不可包含以下字符 “|;&\$<>`"{}()~*?! \n 空白”。

- -U、-W 参数如果搭建命令行未带，则在搭建过程中支持交互式输入。

- -X

集群安装时的 xml，xml 中也可以配置容灾信息用于容灾搭建，即在安装 xml 的基础上扩展三个字段（“localStreamIpmap1”、“remoteStreamIpmap1”、“remotedataPortBase”）

新增字段的配置方式如下，加粗字体内容为示例。每行信息均有注释进行说明。

```
<!-- 每台服务器上的节点部署信息 -->
```

```

<DEVICELIST>
<DEVICE sn="pekpomdev00038">
<!-- 当前主机上需要部署的主 DN 个数 -->
<PARAM name="dataNum" value="1"/>
<!-- 主 DN 的基础端口号 -->
<PARAM name="dataPortBase" value="26000"/>
<!-- 本集群 dn 分片各节点用于 SSH 可信通道的 IP 与流复制的 IP 映射关系 -->
<PARAM name="localStreamIpmap1"
value="(10.244.44.216, 172.31.12.58), (10.244.45.120, 172.31.0.91)"/>
<!-- 对端集群 dn 分片各节点用于 SSH 可信通道的 IP 与流复制的 IP 映射关系 -->
<PARAM name="remoteStreamIpmap1"
value="(10.244.45.144, 172.31.2.200), (10.244.45.40, 172.31.0.38), (10.244.46.138,
172.31.11.145), (10.244.48.60, 172.31.9.37), (10.244.47.240, 172.31.11.125)"/>
<!-- 对端集群的主 dn 端口号 -->
<PARAM name="remotedataPortBase" value="26000"/>
</DEVICE>

```

■ -json

带有容灾信息的 json 文件。

json 文件的配置方式如下，加粗字体内容为示例。

```

{"remoteClusterConf": {"port": 26000, "shards": [{"ip": "10.244.45.144",
"dataIp": "172.31.2.200"}, {"ip": "10.244.45.40", "dataIp": "172.31.0.38"},
{"ip": "10.244.46.138", "dataIp": "172.31.11.145"}, {"ip": "10.244.48.60",
"dataIp": "172.31.9.37"}, {"ip": "10.244.47.240", "dataIp": "172.31.11.125"}]},
"localClusterConf": {"port": 26000, "shards": [{"ip": "10.244.44.216", "dataIp":
"172.31.12.58"}, {"ip": "10.244.45.120", "dataIp": "172.31.0.91"}]}}

```

参数说明：

remoteClusterConf: 对端集群的 dn 分片信息。其中 **port** 为对端集群主 dn 的端口，{"ip": "10.244.45.144", "dataIp": "172.31.2.200"} 为对端集群 dn 分片上各节点用于 SSH 可信通道的 IP 与流复制的 IP 映射关系。

localClusterConf: 本集群的 dn 分片信息。其中 **port** 为本集群主 dn 的端口，{"ip": "10.244.44.216", "dataIp": "172.31.12.58"} 为本集群 dn 分片上各节点用于 SSH 可信通道的 IP 与流复制的 IP 映射关系。

说明：

- **-X** 与 **-json** 参数支持二选一方式进行配置容灾信息，如果命令行中两个参数全部下发，则以 **json** 为准。

■ -time-out=SECS

指定超时时间，主集群会等待备集群连接的超时时间，超时则判定失败，om 脚本自动退出。单位：s。

取值范围：正整数，建议值 1200。

默认值：1200

● 容灾升主参数：

无

● 容灾解除参数：

■ -X

集群安装时的 xml，需要额外配置容灾信息，即扩展三个字段（“localStreamIpmap1”、“remoteStreamIpmap1”、“remotedataPortBase”）

■ -json

带有本端及对端容灾信息的 json 文件。

说明：

-X、-json 的配置方式请参考本节容灾搭建参数配置。

● 容灾查询参数：

无

容灾状态查询结果说明如下：

项目	含义	值	说明	备注
hadr_cluster_stat	流式容灾中数据库实例状态	normal	标识该数据库实例未参与流式容灾	-
		full_backup	主数据库实例数据全量复制中	流式容灾中仅主数据库实例有此状态

项目	含义	值	说明	备注
		archive	主数据库实例日志流式复制中	流式容灾中仅主数据库实例有此状态
		backup_fail	主数据库实例数据全量复制失败	流式容灾中仅主数据库实例有此状态
		archive_fail	主数据库实例日志流式复制失败	流式容灾中仅主数据库实例有此状态
		switchover	计划内主备倒换中	流式容灾中主备数据库实例皆有此状态
		restore	灾备数据库实例数据全量恢复中	流式容灾中仅灾备数据库实例有此状态
		restore_fail	灾备数据库实例全备恢复失败	流式容灾中仅灾备数据库实例有此状态
		recovery	灾备数据库实例日志流式复制中	流式容灾中仅灾备数据库实例有此状态
		recovery_fail	灾备数据库实例	流式容灾中仅灾

项目	含义	值	说明	备注
			日志流式复制失败	备数据库实例有此状态
		promote	灾备数据库实例升主中	流式容灾中仅灾备数据库实例有此状态
		promote_fail	灾备数据库实例升主失败	流式容灾中仅灾备数据库实例有此状态
hadr_switchover_stat	流式容灾主备数据库实例计划内倒换进度展示	百分比	倒换进度展示	-
hadr_failover_stat	流式容灾灾备数据库实例升主进度展示	百分比	倒换进度展示	-
RTO	此时发生灾难，数据恢复所需要的时长	空	数据库实例停机，网络异常等导致流式容灾中断	流式容灾中仅主数据库实例可查
		非空	数据恢复所需要的时长，单位 s	
RPO	此时发生灾难，数据库实	空	数据库实例停机，网络异常等导致	流式容灾中仅主

项目	含义	值	说明	备注
	例丢失的数据 时长		流式容灾中断	数据库实例可查
		非空	数据库实例可能 丢失的数据时长, 单位 s	

3.14.5 示例

- 主集群搭建容灾关系。

```
gs_sdr -t start -m primary -X /opt/install_streaming_primary_cluster.xml
--time-out=1200 -U 'hadr_user' -W 'gbase@123'
```

```
-----
Streaming disaster recovery start 2b9bc268d8a111ecb679fa163e2f2d28
-----
```

```
Start create streaming disaster relationship ...
Got step:[-1] for action:[start].
Start first step of streaming start.
Start cgbaseon config step of streaming start.
Start generate hadr key files.
Streaming key files already exist.
Finished generate and distribute hadr key files.
Start encrypt hadr user info.
Successfully encrypt hadr user info.
Start save hadr user info into database.
Successfully save hadr user info into database.
Start update pg_hba config.
Successfully update pg_hba config.
Start second step of streaming start.
Successfully check cluster status is: Normal
Successfully check instance status.
Successfully check cm_ctl is available.
Successfully check cluster is not under upgrade opts.
Start checking disaster recovery user.
```

```

Successfully check disaster recovery user.
Start prepare secure files.
Start copy hadr user key files.
Successfully copy secure files.
Start fourth step of streaming start.
Starting reload wal_keep_segments value: 16384.
Successfully reload wal_keep_segments value: 16384.
Start fifth step of streaming start.
Successfully set [/gbase/CMServer/backup_open][0].
Start sixth step of streaming start.
Start seventh step of streaming start.
Start eighth step of streaming start.
Waiting main standby connection..
Main standby already connected.
Successfully check cluster status is: Normal
Start ninth step of streaming start.
Starting reload wal_keep_segments value: {'6001': '128'}.
Successfully reload wal_keep_segments value: {'6001': '128'}.
Successfully removed step file.
Successfully do streaming disaster recovery start.

```

- 备集群搭建容灾关系。

```

gs_sdr -t start -m disaster_standby -X
/opt/install_streaming_standby_cluster.xml --time-out=1200 -U 'hadr_user' -W
'gbase@123'
-----
Streaming disaster recovery start e34ec1e4d8a111ecb617fa163e77e94a
-----
Start create streaming disaster relationship ...
Got step:[-1] for action:[start].
Start first step of streaming start.
Start cgbaseon config step of streaming start.
Start update pg_hba config.
Successfully update pg_hba config.
Start second step of streaming start.
Successfully check cluster status is: Normal
Successfully check instance status.
Successfully check cm_ctl is available.
Successfully check cluster is not under upgrade opts.

```

```
Start build key files from remote cluster.
Start copy hadr user key files.
Successfully build and distribute key files to all nodes.
Start fourth step of streaming start.
Start fifth step of streaming start.
Successfully set [/gbase/CMServer/backup_open][2].
Stopping the cluster by node.
Successfully stopped the cluster by node for streaming cluster.
Start sixth step of streaming start.
Start seventh step of streaming start.
Start eighth step of streaming start.
Starting the cluster.
Successfully started primary instance. Please wait for standby instances.
Waiting cluster normal...
Successfully started standby instances.
Successfully check cluster status is: Normal
Start ninth step of streaming start.
Successfully removed step file.
Successfully do streaming disaster recovery start.
```

- 计划内主集群降备。

```
gs_sdr -t switchover -m disaster_standby
-----
Streaming disaster recovery switchover 6897d15ed8a411ec82acfa163e2f2d28
-----
Start streaming disaster switchover ...
Streaming disaster cluster switchover...
Successfully check cluster status is: Normal
Parse cluster conf from file.
Successfully parse cluster conf from file.
Successfully check cluster is not under upgrade opts.
Got step:[-1] for action:[switchover].
Stopping the cluster.
Successfully stopped the cluster.
Starting the cluster.
Successfully started primary instance. Please wait for standby instances.
Waiting cluster normal...
Successfully started standby instances.
Start checking truncation, please wait...
```

```
Stopping the cluster.
Successfully stopped the cluster.
Starting the cluster.
Successfully started primary instance. Please wait for standby instances.
Waiting cluster normal...
Successfully started standby instances.
.
The cluster status is Normal.
Successfully removed step file.
Successfully do streaming disaster recovery switchover.
```

- 计划内备集群升主。

```
gs_sdr -t switchover -m primary
-----
Streaming disaster recovery switchover 20542bbcd8a511ecbbdbfa163e77e94a
-----
Start streaming disaster switchover ...
Streaming disaster cluster switchover...
Waiting for cluster and instances normal...
Successfully check cluster status is: Normal
Parse cluster conf from file.
Successfully parse cluster conf from file.
Successfully check cluster is not under upgrade opts.
Waiting for switchover barrier...
Got step:[-1] for action:[switchover].
Stopping the cluster by node.
Successfully stopped the cluster by node for streaming cluster.
Starting the cluster.
Successfully started primary instance. Please wait for standby instances.
Waiting cluster normal...
Successfully started standby instances.
Successfully check cluster status is: Normal
Successfully removed step file.
Successfully do streaming disaster recovery switchover.
```

- 灾备集群容灾升主。

```
gs_sdr -t failover
```

```
Streaming disaster recovery failover 65535214d8a611ecb804fa163e2f2d28
```

```
-----  
Start streaming disaster failover ...  
Got step:[-1] for action:[failover].  
Successfully check cluster status is: Normal  
Successfully check cluster is not under upgrade opts.  
Parse cluster conf from file.  
Successfully parse cluster conf from file.  
Got step:[-1] for action:[failover].  
Starting drop all node replication slots  
Finished drop all node replication slots  
Stopping the cluster by node.  
Successfully stopped the cluster by node for streaming cluster.  
Start remove replconninfo for instance:6001  
Start remove replconninfo for instance:6002  
Start remove replconninfo for instance:6003  
Start remove replconninfo for instance:6005  
Start remove replconninfo for instance:6004  
Successfully removed replconninfo for instance:6001  
Successfully removed replconninfo for instance:6004  
Successfully removed replconninfo for instance:6003  
Successfully removed replconninfo for instance:6002  
Successfully removed replconninfo for instance:6005  
Start remove pg_hba config.  
Finished remove pg_hba config.  
Starting the cluster.  
Successfully started primary instance. Please wait for standby instances.  
Waiting cluster normal...  
Successfully started standby instances.  
Successfully check cluster status is: Normal  
Try to clean hadr user info.  
Successfully clean hadr user info from database.  
Successfully removed step file.  
Successfully do streaming disaster recovery failover.
```

- 主集群容灾解除。

```
gs_sdr -t stop -X /opt/install_streaming_standby_cluster.xml
```

```
-----  
Streaming disaster recovery stop dae8539ed8a611ecade9fa163e77e94a
```

```
-----  
Start remove streaming disaster relationship ...  
Got step:[-1] for action:[stop].  
Start first step of streaming stop.  
Start second step of streaming start.  
Successfully check cluster status is: Normal  
Check cluster type succeed.  
Successfully check cluster is not under upgrade opts.  
Start third step of streaming stop.  
Start remove replconninfo for instance:6001  
Start remove replconninfo for instance:6002  
Successfully removed replconninfo for instance:6001  
Successfully removed replconninfo for instance:6002  
Start remove cluster file.  
Finished remove cluster file.  
Start fourth step of streaming stop.  
Start remove pg_hba config.  
Finished remove pg_hba config.  
Start fifth step of streaming start.  
Starting drop all node replication slots  
Finished drop all node replication slots  
Start sixth step of streaming stop.  
Successfully check cluster status is: Normal  
Try to clean hadr user info.  
Successfully clean hadr user info from database.  
Successfully removed step file.  
Successfully do streaming disaster recovery stop.
```

- 查询容灾状态。

```
gs_sdr -t query  
-----  
Streaming disaster recovery query 1201b062d8a411eca83efal63e2f2d28  
-----  
Start streaming disaster query ...  
Successfully check cluster is not under upgrade opts.  
Start check archive.  
Start check recovery.  
Start check RPO & RTO.
```

Successfully execute streaming disaster recovery query, result:

```
{'hadr_cluster_stat': 'archive', 'hadr_failover_stat': '',  
'hadr_switchover_stat': '', 'RPO': '0', 'RTO': '0'}
```


4 系统内部命令

本节所涉及的工具为 GBase 8s 进程间运行使用的工具，或者说模块间调用的工具。这些工具只用于满足内部调用场景，未对其他场景下的使用正确性做过验证。因此，不建议用户直接使用这些工具执行业务，以防对系统运行带来影响。

4.1 dsscmd

4.1.1 概述

GBase 8s 部署资源池化模式且开启 `ss_enable_dss` 功能情况下，经常需要对基于磁阵实现的集群文件系统进行各种查询，操作和管理。为了对 `dss` 创建的文件系统进行有效的管理，资源池化提供了 `dsscmd` 管理工具。

4.1.2 命令

- 显示帮助信息

```
dsscmd -h  
dsscmd --help
```

- 显示 `dss` 的版本号

```
dsscmd -v  
dsscmd --version
```

- 创建卷组

```
dsscmd cv <-g vg_name> <-v vol_name> [-s au_size] [-D DSS_HOME]
```

此处的 `vg_name` 为卷组名，命名长度不能超过 64，仅支持数字，大小写字母，和部分特殊字符 `'_'`, `'.'`, `'-'`。其他字符不支持。

- 显示卷组和磁盘使用信息

```
dsscmd lsvg [-m measure_type] [-t show_type] [-U UDS:socket_domain]
```

- 卷组中添加卷

```
dsscmd adv <-g vg_name> <-v vol_name> [-U UDS:socket_domain]
```

此处的 `vg_name` 为卷组名，不需要以 `‘+’` 开头。

- 新建目录

```
dsscmd mkdir <-p path> <-d dir_name> [-U UDS:socket_domain]
```

- 创建文件

```
dsscmd touch <-p path> [-U UDS:socket_domain]
```

- 显示路径下文件信息

```
dsscmd ls <-p path> [-m measure_type] [-U UDS:socket_domain]
```

- 将源文件复制到目标文件

```
dsscmd cp <-s src_file> <-d dest_file> [-U UDS:socket_domain]
```

此处的 `src_file` 和 `dest_file` 为带路径的文件名，使用限制参考“参数说明”部分的 `path`。

- 删除卷组文件

```
dsscmd rm <-p path> [-U UDS:socket_domain]
```

- 删除卷组中的卷

```
dsscmd rmv <-g vg_name> <-v vol_name> [-U UDS:socket_domain]
```

此处的 `vg_name` 为卷组名，不需要以 ‘+’ 开头。

- 删除目录及其内容

```
dsscmd rmdir <-p path> [-r] [-U UDS:socket_domain path]
```

此处 `-r` 代表级联删除。

- 查询 LUN/注册信息

```
dsscmd inq <-t inq_type>
```

说明：`inq_type` 表示查询信息的类型。取值如下：

`lun`: 查询 LUN 信息。

`reg`: 查询 `reservations` 信息。

- 查询该节点是否注册

```
dsscmd inq_reg <-i inst_id> [-D DSS_HOME]
```

- 显示客户端信息

```
dsscmd lscli
```

- 从集群中踢出非当前节点，第一个参数是要移除的节点 `id`

```
dsscmd kickh <-i inst_id> [-D DSS_HOME]
```

- 将节点添加到集群，不走服务端

```
dsscmd reghl [-D DSS_HOME]
```

- 将节点从集群中移除，不走服务端

```
dsscmd unreghl [-t type] [-D DSS_HOME]
```

说明:

type: 值为 int 类型，0 表示不使用 vg 锁，非 0 表示使用 vg 锁，默认值为 1。

扫描并打开指定路径下指定用户和属组的盘符，不走服务端

```
dsscmd scandisk <-t type> <-p path> <-u user_name> <-g group_name>
```

说明:

type: 盘符类型，当前只支持 block 块设备。

[path][user_name][group_name]: 该字段中不能含有命令注入安全隐患的非法字符：'|', ';', '&', '\$', '<', '>', '"', '\', '^', '^"', '{', '}', '(', ')', '[', ']', '~', '*', '?', ',', '!', '\n'。

user_name: 操作系统的用户名。

group_name: 操作系统的属组。

- 设置 dss 节点的 auid

```
dsscmd auid <-a auid>
```

此处的 auid 是一个 64 位的值。

- 读取 dss 文件内容

```
dsscmd examine <-p path> <-o offset> <-f format> [-s read_size] [-D DSS_HOME] [-U UDS:socket_domain]
```

此处 format 为读取文件内容的格式，取值范围为：c char、h unsigned short、u unsigned int、l unsigned long、s string、x hex。

- 读取磁阵文件内容

```
dsscmd dev <-p path> <-o offset> <-f format> [-D DSS_HOME]
```

此处 format 为读取文件内容的格式，取值范围为：c char、h unsigned short、u unsigned int、l unsigned long、s string、x hex。

- 显示磁盘信息

```
dsscmd showdisk <-g vg_name> <-s struct_name> [-D DSS_HOME]
```

```
dsscmd showdisk <-g vg_name> <-b block_id> <-n node_id> [-D DSS_HOME]
```

说明:

- `struct_name`: 指定输出信息的文件类型。取值范围: `core_ctrl`、`vg_header`、`volume_ctrl`、`root_ft_block`。
- 如果指定参数 `-b block_id`, 则需指定 `-n node_id`。
- `block_id` 是一个 64 位的值, 前 10 位是 `volume_id`, 34 位是 `au_id`, 17 位是 `block_id`, 最后 3 位是预留。

- 重命名文件名

```
dsscmd rename <-o old_name> <-n new_name> [-U UDS:socket_domain]
```

此处的 `old_name` 和 `new_name` 为带路径的文件名, 使用限制参考“参数说明”部分的 `path`。

显示带有可选参数的文件/目录的磁盘使用情况

```
dsscmd du <-p path> [-f format] [-U UDS:socket_domain]
```

说明: `format`: 支持 3 种参数, 参数之间不需要任何分隔符。

B|K|M|G|T B: Byte, K: KB, M: MB, G: GB, T: TB。

`s|a|s`: 目录, `a`: 计算所有文件, 而不仅仅是目录。

`S` 对于目录不包括子目录的大小。

- 按文件名查找文件所在路径

```
dsscmd find <-p path> <-n name> [-U UDS:socket_domain]
```

- 创建软连接

```
dsscmd ln <-s src_path> <-t target_path> [-U UDS:socket_domain]
```

此处的 `src_path` 和 `target_path` 是目录或者带路径的文件名, 使用限制参考“参数说明”部分的 `path`。

- 显示文件链接信息

```
dsscmd readlink <-p path> [-U UDS:socket_domain]
```

- 取消软连接

```
dsscmd unlink <-p path> [-U UDS:socket_domain]
```

- 加密

```
dsscmd encrypt
```

- 设置配置项

```
dsscmd setcfg <-n name> <-v value> [-s scope] [-U UDS:socket_domain]
```

- 获取配置项信息

```
dsscmd getcfg <-n name> [-U UDS:socket_domain]
```

- 通知 dss 服务端进行优雅退出，异步关闭 dssserver 进程

```
dsscmd stopdss [-U UDS:socket_domain]
```

- 清理 dss 内部的锁资源

```
dsscmd clean_vglock [-D DSS_HOME]
```

4.1.3 参数说明

- show_type

d 显示详细信息，t 显示表中的信息。

- au_size:

au 的大小，单位 KB，范围为 2MB-64MB。

- vg_name

卷组名。

- measure_type

显示文件大小的单位。例如，B:Byte, K:KB, M:MB, G:GB, T:TB。

默认为 Byte。

- path

表示目录或带路径的文件，需要带 '+' 且不支持使用相对路径，命名长度不能超过 1K，仅支持数字，大小写字母，和部分特殊字符 '_'、'.'、'-'、'\', 其中 '\' 是分隔符。其他字符不支持。文件名本身长度不能超过 63 位，仅支持数字，大小写字母，和部分特殊字符 '_'、'.'、'-'，其他字符不支持。

说明：dsscmd 命令使用 UDS:socket_domain 的时候，需要与服务端通信，由于客户端和服务端通信的报文最长为 1024，所以实际 path 支持的长度可能会少于 1K。

- dir_name

目录名，命名长度不能超过 64，仅支持数字，大小写字母，和部分特殊字符 '_'、'.'、'-'、'\ '。其他字符不支持。

- vol_name

卷名。文件命名长度不能超过 1K（包括目录），仅支持数字，大小写字母，和部分特殊字符 '_'、'.'、'-'、'\ '，其中'\ '是分隔符。其他字符不支持。

- -D

\$DSS_HOME 所在路径。

- -U UDS:socket_domain

dss 实例进程使用的 socket 文件路径。

- inst_id |kicked_inst_id

集群中 dss 实例 id。

- addr

volume 上锁的起始地址。

- read_size

读取长度。

- name:

配置项名称。

- value

配置项的值。

- scope

配置项生效的范围。

取值范围：memory、pfile 或 both。

memory：表示在内存中进行修改，立即生效。

pfile：表示在 pfile 中进行修改，修改后需要重启数据库才能生效。

both：表示既在内存也在 pfile 中进行修改。

4.1.4 使用示例

创建卷组，卷组名为 `data` 和 `log`，`au` 大小分别为 2048 和 65536，`-D` 指定 `dss_home`。

```
dsscmd cv -g data -v /dev/tpcc_data -s 2048 -D /home/ss_test/dss_home
dsscmd cv -g log -v /dev/tpcc_log -s 65536 -D /home/ss_test/dss_home
```

创建文件夹 `pg_xlog0`，`UDS` 指定 `dss` 进程的 `socket` 文件。

```
dsscmd mkdir -p +log -d pg_xlog0 -U
UDS:/home/ss_test/dss_home/.dss_unix_d_socket
```

文件复制，将 `data` 中的文件复制到 `log` 中。

```
dsscmd cp -s +data/pg_xlog0/00000001000000000000000001 -d
+log/pg_xlog0/00000001000000000000000001 -U
UDS:/home/ss_test/dss_home/.dss_unix_d_socket
```

文件删除，删除 `data` 中文件。

```
dsscmd rm -p +data/pg_xlog0/00000001000000000000000001 -U
UDS:/home/ss_test/dss_home/.dss_unix_d_socket
```

删除 `data` 中文件夹 `pg_xlog0/archive_status`（文件夹为空，若文件夹内有文件需要级联删除，需要加 `-r`）。

```
dsscmd rmdir -p +data/pg_xlog0/archive_status -U
UDS:/home/ss_test/dss_home/.dss_unix_d_socket
```

创建软链接。

```
dsscmd ln -s +log/pg_xlog0 -t +data/pg_xlog0 -U
UDS:/home/ss_test/dss_home/.dss_unix_d_socket
```

查看 `data` 中 `pg_xlog0` 目录下的文件信息。

```
dsscmd ls -p +data/pg_xlog0 -U UDS:/home/ss_test/dss_home/.dss_unix_d_socket
```

4.2 dssserver

4.2.1 背景信息

`dssserver` 是独立进程，直接管理磁阵裸设备，并对外提供类似分布式文件系统的能力。通过共享内存和客户端 `API` 动态库，为数据库提供创建文件、删除文件、扩展和收缩文件、读写文件的能力。也是一个可执行的命令，能够通过其启动一个 `dssserver` 进程。

4.2.2 语法

启动 dssserver 进程

```
dssserver -D $DSS_HOME
```

4.2.3 参数说明

dssserver 作为一个独立的进程，在启动之前需要提供配置文件 dss_inst.ini，相关的配置参数整理如下：

- LSNR_PATH

参数说明：dssserver 进程需要的 socket 文件路径。

取值范围：字符串。

默认值：\$DSS_HOME

- LOG_HOME

参数说明：日志路径。

取值范围：字符串类型。

默认值：""

- _LOG_BACKUP_FILE_COUNT

参数说明：日志文件最多保留的个数。

取值范围：整型，0~1024。

默认值：10

- _LOG_MAX_FILE_SIZE

参数说明：单个日志文件最大大小。

取值范围：整型，[1M,4G]。

默认值：10M

- INST_ID

参数说明：实例号，每个主备下的 dssserver 进程各不相同。

取值范围：整型，0~63。

默认值：0

- STORAGE_MODE

参数说明：DSS 对应的存储设备类型，磁阵的话配置为 RAID 或者 CLUSTER_RAID。

取值范围：字符串。

默认值：DISK

- _LOG_LEVEL

参数说明：日志级别。

取值范围：整型，0~4087。

默认值：7

- MAX_SESSION_NUMS

参数说明：最大的会话连接数量。

取值范围：整型，[16,16320]。

默认值：8192

- DISK_LOCK_INTERVAL

参数说明：磁盘锁加锁超时时间。

取值范围：整型，1~600000，单位：毫秒。

默认值：100

- DLOCK_RETRY_COUNT

参数说明：磁盘加锁重试次数。

取值范围：整型，1~500000。

默认值：50

- _AUDIT_BACKUP_FILE_COUNT

参数说明：审计备份文件个数。

取值范围：整型，0~1024。

默认值：10

- _AUDIT_MAX_FILE_SIZE

参数说明：审计备份文件最大大小。

取值范围：整型，1M~4G。

默认值：10M

- `_LOG_FILE_PERMISSIONS`

参数说明：日志文件权限。

取值范围：整型，600~777。

默认值：600

- `_LOG_PATH_PERMISSIONS`

参数说明：日志路径权限。

取值范围：整型，700~777。

默认值：700

- `_SHM_KEY`

参数说明：共享内存 KEY。

取值范围：整型，1~64。

默认值：1

- `DSS_NODES_LIST`

参数说明：dssserver 节点之间 mes 通信 url。

取值范围：字符串。格式为 `inst_Id: ip: port, inst_Id: ip: port.....`。

默认值：0:127.0.0.1:1611

- `INTERCONNECT_TYPE`

参数说明：通信协议类型。

取值范围：字符串，TCP/RDMA。

默认值：TCP

- `INTERCONNECT_CHANNEL_NUM`

参数说明：内部通信 mes 链接通道个数。

取值范围：整型，1~32。

默认值：2

- **WORK_THREAD_COUNT**

参数说明：内部通信 mes 工作线程数量。

取值范围：整型，2~64。

默认值：2

- **RECV_MSG_POOL_SIZE**

参数说明：消息接收内存池大小。

取值范围：整型，1M~1G。

默认值：16M

- **MES_ELAPSED_SWITCH**

参数说明：内部通信 mes 统计计数时延开关。

取值范围：布尔型，FALSE/TRUE。TRUE 表示当前开启 mes 通信计数统计时延，FALSE 表示不开启。

默认值：FALSE

- **DISK_LOCK_FILE_PATH**

参数说明：STORAGE_MODE 为 DISK 时，所用锁的文件保存路径。

取值范围：字符串。

默认值：\$DSS_HOME

- **SSL_CA**

参数说明：CA 服务器的根证书。

取值范围：字符串。

默认值：""

- **SSL_KEY**

参数说明：指定服务器私钥文件，用以对公钥加密的数据进行解密。

取值范围：字符串。

默认值：""

- **SSL_CRL**

参数说明：证书吊销列表。

取值范围：字符串。

默认值：""

- **SSL_CERT**

参数说明：指定服务器证书文件，包含服务器端的公钥。

取值范围：字符串。

默认值：""

- **SSL_CIPHER**

参数说明：SSL 通讯使用的加密算法。

取值范围：字符串。

默认值：""

- **SSL_PWD_CIPHERTEXT**

参数说明：私钥密码。

取值范围：字符串。

默认值：""

- **SSL_CERT_NOTIFY_TIME**

参数说明：SSL 服务器证书到期前提醒的天数。

取值范围：整型，7~180。

默认值：30

- **POOL_NAMES**

参数说明：存储池配置。

取值范围：字符串。

默认值：""

- **IMAGE_NAMES**

参数说明：image 配置。

取值范围：字符串。

默认值：空

- CEPH_CONFIG

参数说明：ceph 集群配置文件。

取值范围：字符串。

默认值：“/etc/ceph/ceph.conf”

- VOLUME_TYPES

参数说明：卷类型。

取值范围：字符串。

默认值：“”

- DSS_CM_SO_NAME

参数说明：cm 的二进制文件名。

取值范围：字符串。

默认值：“”

4.3 gaussdb

4.3.1 背景信息

gaussdb 是 GBase 8s 数据库的主进程，也是一个可执行的命令，能够通过其启动一个数据库进程。

4.3.2 语法

```
gaussdb [OPTION]...
```

4.3.3 描述

客户端应用程序为了访问数据库，将连接（通过网络或本地）到一个正在运行的 gaussdb 进程。然后该进程实例会启动一个独立的线程来处理这个连接。

须知

- 通过 `gaussdb` 启动数据库时，需要再开一个窗口来连接到数据库，也可以使用 `&` 符号使程序在后台执行。

一个 `gaussdb` 进程总是管理来自同一个数据库的数据。一个系统上可以同时运行多个 `gaussdb` 进程，只要使用不同的数据目录和不同的端口号。`gaussdb` 启动时需要知道数据目录的位置，该位置必须通过 `-D` 指定。通常，`-D` 直接指向由 `gs_initdb` 创建的数据库目录。

缺省时 `gaussdb` 在前台启动并将日志信息输出到标准错误。但在实际应用中，`gaussdb` 应当作为后台进程启动，而且多数是在系统启动时自动启动。

`gaussdb` 还能以单用户模式运行，这种用法主要用于 `gs_initdb` 的初始化过程中，有时候也被用于调试灾难性恢复。不过，单用户模式运行的服务器并不适合于调试，因为没有实际的进程间通讯和锁动作发生。当从 `shell` 上以单用户模式调用时，用户可以输入查询，然后结果会在屏幕上以一种更适合开发者阅读（不适合普通用户）的格式显示出来。在单用户模式下，将把会话用户 ID 设为 1 并赋予系统管理员权限。该用户不必实际存在，因此单用户模式运行的服务器可以用于对某些意外损坏的系统表进行手工恢复。

4.3.4 参数说明

通用参数

- `-B NBUFFERS`

设置服务器进程使用的共享内存缓冲区的数量。

- `-b BINARY UPGRADES`

标志二进制升级。

- `-c NAME=VALUE`

给一个正在运行的参数赋值。`-c` 可以出现多次从而设置多个参数。

- `-C NAME`

打印一个命名的运行时的参数的值然后退出。可以用在正在运行的服务器上，从 `postgresql.conf` 返回值。

取值范围：取值为 `postgresql.conf` 中的所有参数。

- `-d 1-5`

设置调试级别，1-5 记录对应 debug 级别信息。级别越高，写到服务器日志的调试输出越多。

取值范围：1-5。

说明

➤ 如果取值小于等于 0，则只会记录 notice 级别的信息。

- -D DATADIR

声明数据目录或者配置文件的文件系统路径。

- -e

把缺省日期风格设置为 European，也就是说用 DMY 规则解释日期输入，并且在一些日期输出格式里日在月份前面打印。

- -F

关闭 fsync 调用以提高性能，但是要冒系统崩溃时数据毁坏的风险。声明这个选项等效关闭了 fsync 参数。

- -h HOSTNAME

指定 gaussdb 进程侦听来自前端应用 TCP/IP 连接的主机名或 IP 地址。

取值范围：前端存在的主机名或 IP 地址。

- -i

该选项允许远程客户通过 TCP/IP（网际域套接字）与服务器通讯。

- -k DIRECTORY

指定 gaussdb 侦听来自前端应用连接的 Unix 域套接字的目录。

取值范围：缺省通常是/tmp，但是可以在编译的时候修改。

- -l

该选项允许远程客户通过 SSL（安全套接层）与服务器通讯。

- -N MAX-CONNECT

设置服务器接受的客户端连接的最大数。缺省时由 gs_initdb 自动选择。声明这个选项

等价于声明 `max_connections` 配置参数。

取值范围：正整数。

- **-M SERVERMODE**

在启动时指定数据库的启动模式。

取值范围：SERVERMODE 可以取下面四个值：

- `primary`：本端以主机模式启动。
- `standby`：本端以备机模式启动。
- `cascade_standby`：本端以级联备机模式启动。
- `pending`：本端处于等待状态，等待提升为主机或者备机。

- **-o OPTIONS**

向每个服务器进程传递 "OPTIONS"。保留语法，已没有实际功能。

- **-p PORT**

指定 `gaussdb` 侦听客户端连接的 TCP/IP 端口或本地 Unix domain socket 文件的扩展。默认端口号为 15400。

取值范围：正整数，在操作系统支持的端口范围内。

- **-R**

指示以 `xlogriver` 运行。仅在 `-M standby` 情况下使用。

- **-s**

在每条命令结束时打印时间信息和其他统计信息。

- **-S WORK-MEM**

声明内部排序和散列在求助于临时磁盘文件之前可以使用的内存大小。

取值范围：单位为 KB。

- **-u NUM**

设置升级前内核版本的编号。

- `-V, --version`

打印 `gaussdb` 的版本信息然后退出。

- `--NAME=VALUE`

给一个正在运行的参数赋值。

- `--describe-config`

描述配置参数然后退出。

- `--securitymode`

设置允许数据库系统在安全模式下运行。

- `--single_node`

拉起单机数据库。

- `-, --help`

显示关于 `gaussdb` 命令行参数的帮助信息，然后退出。

服务器模式

- `primary`

数据库系统作为主服务器启动，向备服务器发送 `xlog`。

- `standby`

数据库系统作为备用服务器启动，从主服务器接收 `xlog`。

- `pending`

数据库系统作为挂起服务器启动，等待升级为主服务器或降级为备用服务器。

- `fenced`

数据库系统启动一个隔离的主进程，以安全模式执行 UDF (与 `Gaussdb` 进程分开运行)。

开发者选项

开发者选项便于开发人员调试使用，有时也用于帮助恢复严重损坏的数据库。在应用程序使用数据库提供服务时，请不要使用这些参数进行调试。

- -f s|i|n|m|h

禁止使用某种扫描和连接方法。

取值范围：

- s: 关闭顺序
- i: 索引扫描
- n: 关闭嵌套循环
- m: 融合 (merge) 连接
- h: Hash 连接

- -n

主要用于调试导致服务器进程异常崩溃的问题。一般策略是通知所有其他服务器进程终止并重新初始化共享内存和信号灯。该选项指定 `gaussdb` 异常退出后，不重新初始化共享内存。

- -O

允许修改系统表结构。

须知

- 此选项可能导致系统表损坏，甚至数据库无法启动。

- -P

读系统表时忽略系统索引，但在修改表时仍然更新索引。

- -t pa|pl|ex

显示与每个主要系统模块相关的查询计时。

- -T

主要用于调试导致服务器进程异常崩溃的问题。该选项指定 `gaussdb` 通过发送 `SIGSTOP` 信号停止其他所有服务器进程，但是并不让它们退出。这样就允许系统程序员手动从所有服务器进程搜集内核转储。

- -W NUM

指定新服务器进程开始需要等待的时间。

取值范围：单位为秒。

- --localxid

使用本地事务 id，而不是全局事务 ID。

取值范围：已存在的本地事务 ID。

须知

➤ 此选项仅用于 gs_initdb。使用此选项可能会导致数据库不一致。

单用户模式选项

- --single

启动单用户模式。必须是命令行第一个参数。

- DBNAME

设置数据库名称。必须是命令行最后一个参数。

取值范围：字符串。默认为用户名。

- -d 0-5

重新指定调试级别。

取值范围：0-5

- -E

回显所有命令。

➤ -j

禁止使用新行作为语句分隔符。

- -r FILENAME

将所有服务器标准输出和标准错误保存到文件 filename 中。

自启动模式选项

- --boot

启动自启动模式。必须是命令行中的第一个选项。该参数是对数据库中的参数变量及相关配置初始化，常用在数据库安装的流程中，直接使用该参数，无明显行为感知。

- **DBNAME**

设置数据库名称。必须是命令行最后一个参数。

取值范围：字符串。默认为用户名。

- **-r FILENAME**

将所有服务器标准输出和标准错误保存到文件 `filename` 中。

- **-x NUM**

指定一个新的服务器线程的类型。在初始化数据库时，会用到自启动模式，通过设置这个参数，启动不同线程来执行一些逻辑，正常情况下，不会用到，因为自启动模式很少会用到。

节点选项

- **--coordinator**

作为 Coordinator 协调器启动。

- **--datanode**

作为 Datanode 数据节点启动。

- **--restoremode**

开始还原要添加的新节点上的现有架构。

- **--single_node**

作为单节点启动。

除以上参数外，出现在 `postgres.conf` 中的所有参数，均可以用 `--guc_name=guc_value` 的方式在 `gaussdb` 启动时传给 `gaussdb`。例如，`--enable_fast_query_shipping=off`。

4.3.5 环境变量

PGLINETENCODING：客户端使用的缺省字符编码。客户端可以独立地覆盖它。这个值也可以在配置文件里设置。

PGDATESTYLE: 运行时参数 DateStyle 的缺省值。现在反对使用该环境变量。

TZ: 服务器的时区。

4.3.6 错误处理

一个提到了 `semget` 或 `shmget` 的错误信息可能意味着需要重新配置内核，提供足够的共享内存和信号灯。可以通过降低 `shared_buffers` 值以减少 GBase 8s 的共享内存的消耗，或者降低 `max_connections` 值减少 GBase 8s 的信号灯的消耗。

如果发现类似“另外一个服务器正在运行”的错误信息，可以根据系统使用不同的命令：

```
ps ax | grep gaussdb
```

或

```
ps -ef | grep gaussdb
```

如果确信没有冲突的服务器正在运行，可以删除消息里提到的锁文件然后再次运行。

无法绑定端口的错误信息可能表明该端口已经被其他非 `gaussdb` 进程使用。如果终止 `gaussdb` 后又马上用同一端口号运行它，也可能得到错误信息。这时，必须多等几秒，等操作系统关闭了该端口再试。最后，如果使用了一个操作系统认为是保留的端口，也可能导致这个错误信息。例如：Unix 版本认为低于 1024 的端口号是“可信任的”，因而只有 Unix 系统管理员可以使用它们。

须知

- 如果有可能，不要使用 `SIGKILL` 杀死主进程。这会阻止 `gaussdb` 在退出前释放它持有的系统资源（例如共享内存和信号灯）。这样可能会影响到将来启动新的进程。
- 可以使用 `SIGTERM`、`SIGINT`、`SIGQUIT` 信号正常结束服务器进程。第一个信号将等待所有的客户端退出后才退出。第二个将强制断开所有客户端，而第三个将不停止立刻退出，导致在重启时的恢复运行。
- 信号 `SIGHUP` 将会重新加载服务器配置文件。它也可能给单个服务器进程发送 `SIGHUP` 信号，但是这通常是不明显的。
- 要取消一个正在执行的查询，向正在运行的进程发送 `SIGINT` 信号。
- 主服务器进程向子进程发送 `SIGTERM` 信号让它们正常退出；发送 `SIGQUIT` 信号

立即退出且不做清理工作。用户有三种信号不能用。同时，发送 SIGKILL 信号也是不明智的：主进程将把这个信号当作崩溃信号，然后会强制其他兄弟进程作为标准的崩溃回复过程退出。

4.3.7 用法

启动一个单用户模式的服务器：

```
gaussdb --single -D /usr/local/pgsql/data other-options my_database
```

用-D 给服务器提供正确的数据库目录的路径。同时还要声名已存在的特定数据库名称。

须知

- 通常，独立运行的服务器把换行符当做命令输入完成字符；要想把一行分成多行写，必需在除最后一个换行符以外的每个换行符前面敲一个反斜杠。
- 如果使用了-j 命令行选项，新行将不被当作命令结束符。此时服务器将从标准输入一直读取到 EOF 标志为止，然后把所有读到的内容当作一个完整的命令字符串看待，并且反斜杠与换行符也被当作普通字符来看待。
- 输入 EOF (Control+D) 即可退出会话。如果已经使用了-j 则必须连续使用两个 EOF 才行。
- 单用户模式运行的服务器不会提供复杂的行编辑功能（比如没有命令历史）。单用户模式也不会做任何后台处理，像自动检查点。

4.3.8 示例

用缺省值在后台启动 gaussdb：

```
$ nohup gaussdb >logfile 2>&1 </dev/null &
```

在指定端口启动 gaussdb，如 1234：

```
$ gaussdb --single_node -p -D /home/gbase/data/
```

使用 gsql 与服务器连接，用-p 指定端口号：

```
$ gsql -d postgres -p 1234
```

4.4 gs_backup

4.4.1 背景信息

GBase 8s 部署成功后，在数据库运行的过程中，会遇到各种问题及异常状态。GBase 8s 提供了 gs_backup 工具帮助 GBase 8s 备份、恢复重要数据、显示帮助信息和版本号信息。

4.4.2 前提条件

- 可以正常连接 GBase 8s 数据库。
- 在进行还原时，需要保证各节点备份目录中存在备份文件。
- 需以操作系统用户 gbase 执行 gs_backup 命令。

4.4.3 语法

- 备份数据库主机

```
gs_backup -t backup --backup-dir=BACKUPDIR [-h HOSTNAME] [--parameter] [--binary]
[--all] [-l LOGFILE]
```

- 恢复数据库主机

```
gs_backup -t restore --backup-dir=BACKUPDIR [-h HOSTNAME] [--parameter]
[--binary] [--all] [-l LOGFILE] [--force]
```

- 显示帮助信息

```
gs_backup -? | --help
```

- 显示版本号信息

```
gs_backup -V | --version
```

4.4.4 参数说明

gs_backup 参数可以分为如下几类：

备份数据库主机参数：

- -h

指定存储备份文件的主机名称。

取值范围：主机名称。如果不指定主机名称，则备份当前集群。

- `--backup-dir=BACKUPDIR`

备份文件保存路径。

- `--parameter`

备份参数文件，不指定`--parameter`、`--binary`、`--all`参数时默认只备份参数文件。

- `--binary`

备份 `app` 目录下的二进制文件。

- `--all`

备份 `app` 目录下的二进制文件、`pg_hba.conf` 和 `postgresql.conf` 文件。

- `-l`

指定日志文件及存放路径。

默认值：`$GAUSSLOG/om/gs_backup-YYYY-MM-DD_hhmmss.log`

恢复数据库主机参数：

- `-h`

指定需要恢复主机的名称。

取值范围：主机名称。如果不指定主机，则恢复对应的备份节点。

- `--backup-dir=BACKUPDIR`

恢复文件提取路径。

- `--parameter`

恢复参数文件，不指定`--parameter`、`--binary`、`--all`参数时默认只恢复参数文件。

- `--binary`

恢复二进制文件。

- `--all`

恢复二进制和参数文件。

- -l

指定日志文件及存放路径。

默认值：\$GAUSSLOG/om/gs_backup-YYYY-MM-DD_hhmmss.log

- --force

节点的静态文件丢失之后强行 restore，仅限--all 或者--binary 参数使用时才生效。

其他参数：

- -?, --help

显示帮助信息。

- -V, --version

显示版本号信息。

- -t

指定操作类型。

取值范围：backup 或者 restore。

4.4.5 示例

- 使用 gs_backup 脚本备份数据库主机。

```
$ gs_backup -t backup --backup-dir=/opt/software/gbase/backup_dir --parameter
Parsing configuration files.
Successfully parsed the configuration file.
Performing remote backup.
Remote backup succeeded.
Successfully backed up cluster files.
```

- 使用 gs_backup 脚本恢复数据库主机。

```
$ gs_backup -t restore --backup-dir=/opt/software/gbase/backup_dir --parameter
Parsing configuration files.
Successfully parsed the configuration file.
Performing remote restoration.
Successfully restored cluster files.
```

4.5 gs_basebackup

4.5.1 背景信息

GBase 8s 部署成功后，在数据库运行的过程中，会遇到各种问题及异常状态。GBase 8s 提供了 `gs_basebackup` 工具做基础的物理备份。`gs_basebackup` 的实现目标是对服务器数据库文件的二进制进行拷贝，其实现原理使用了复制协议。远程执行 `gs_basebackup` 时，需要使用系统管理员账户。`gs_basebackup` 当前支持热备份模式和压缩格式备份模式。

说明

- `gs_basebackup` 仅支持主机和备机的全量备份，不支持增量。
- `gs_basebackup` 当前支持热备份模式和压缩格式备份模式。
- 若打开增量检测点功能且打开双写，`gs_basebackup` 也会备份双写文件。
- `gs_basebackup` 在备份包含绝对路径的表空间时，如果在同一台机器上进行备份，可以通过 `tablespace-mapping` 重定向表空间路径或使用归档模式进行备份。
- 若 `pg_xlog` 目录为软链接，备份时将不会建立软链接，会直接将数据备份到目的路径的 `pg_xlog` 目录下。
- 备份过程中收回用户备份权限，可能导致备份失败或者备份数据不可用。
- 如果因为网络临时故障等原因导致 Server 端无法应答，`gs_basebackup` 将在最长等待 120 秒后退出。

4.5.2 前提条件

- 可以正常连接 GBase 8s 数据库。
- 备份过程中用户权限没有被回收。
- `pg_hba.conf` 中需要配置允许复制链接，且该连接必须由一个系统管理员建立。
- 如果 `xlog` 传输模式为 `stream` 模式，需要配置 `max_wal_senders` 的数量，至少有一个可用。

- 如果 xlog 传输模式为 fetch 模式，有必要把 wal_keep_segments 参数设置得足够高，这样在备份末尾之前日志不会被移除。
- 在进行还原时，需要保证各节点备份目录中存在备份文件，若备份文件丢失，则需要从其他节点进行拷贝。

4.5.3 语法

- 显示帮助信息

```
gs_basebackup -? | --help
```

- 显示版本号信息

```
gs_basebackup -V | --version
```

从备份文件恢复数据

当数据库发生故障时，需要从备份文件进行恢复。因为 gs_basebackup 是对数据库按二进制进行备份，因此恢复时可以直接拷贝替换原有的文件，或者直接在备份的库上启动数据库。

说明

- 若当前数据库实例正在运行，直接从备份文件启动数据库可能会存在端口冲突，这时需要修改配置文件的 port 参数，或者在启动数据库时指定一下端口。
- 若当前备份文件为主备数据库，可能需要修改一下主备之间的复制连接。即配置文件中的 postgresql.conf 中的 replconninfo1、replconninfo2 等。
- 若配置文件 postgresql.conf 的参数 data_directory 打开且有配置，当使用备份目录启动数据库时候，data_directory 和备份目录不同会导致启动失败。可以修改 data_directory 的值为新的数据目录，或者注释掉该参数。

若要在原库的地方恢复数据库，参考步骤如下：

- (1) 停止数据库服务器，具体操作请参见《GBase 8s V8.8.5 5.0.0_管理员指南》。
- (2) 将原数据库和所有表空间复制到另外一个位置，以备后面需要。
- (3) 清理原库中的所有或部分文件。
- (4) 使用数据库系统用户权限从备份中还原需要的数据库文件。

- (5) 若数据库中存在链接文件，需要修改使其链接到正确的文件。
- (6) 重启数据库服务器，并检查数据库内容，确保数据库已经恢复到所需的状态。

说明

- 暂不支持备份文件增量恢复。

恢复后需要检查数据库中的链接文件是否链接到正确的文件。

4.5.4 参数说明

gs_basebackup 参数可以分为如下几类：

- -D directory

备份文件输出的目录，必选项。

常用参数

- -c, --checkpoint=fast|spread

设置检查点模式为 fast 或者 spread（默认）。

- -l, --label=LABEL

为备份设置标签。

- -P, --progress

启用进展报告。

- -v, --verbose

启用冗长模式。

- -V, --version

打印版本后退出。

- -?, --help

显示 gs_basebackup 命令行参数。

- -T, --tablespace-mapping=olddir=newdir

在备份期间将目录 olddir 中的表空间重定位到 newdir 中。为使之有效，olddir 必须正好

匹配表空间所在的路径（但如果备份中没有包含 `olddir` 中的表空间也不是错误）。`olddir` 和 `newdir` 必须是绝对路径。如果一个路径凑巧包含了一个 `=` 符号，可用反斜线对它转义。对于多个表空间可以多次使用这个选项。

- `-F, --format=plain|tar`

设置输出格式为 `plain`（默认）或者 `tar`。没有设置该参数的情况下，默认 `--format=plain`。`plain` 格式把输出写成平面文件，使用和当前数据目录和表空间相同的布局。当集簇没有额外表空间时，整个数据库将被放在目标目录中。如果集簇包含额外的表空间，主数据目录将被放置在目标目录中，但是所有其他表空间将被放在它们位于服务器上的相同的绝对路径中。`tar` 模式将输出写成目标目录中的 `tar` 文件。主数据目录将被写入到一个名为 `base.tar` 的文件中，并且其他表空间将被以其 `OID` 命名。生成的 `tar` 包，需要用 `gs_tar` 命令解压。

- `-X, --xlog-method=fetch|stream`

设置 `xlog` 传输方式。没有设置该参数的情况下，默认 `--xlog-method=stream`。在备份中包括所需的预写式日志文件（WAL 文件）。这包括所有在备份期间产生的预写式日志。`fetch` 方式在备份末尾收集预写式日志文件。因此，有必要把 `wal_keep_segments` 参数设置得足够高，这样在备份末尾之前日志不会被移除。如果在要传输日志时它已经被轮转，备份将失败并且是不可用的。`stream` 方式在备份被创建时流传送预写式日志。这将开启一个到服务器的第二连接并且在运行备份时并行开始流传输预写式日志。因此，它将使用最多两个由 `max_wal_senders` 参数配置的连接。只要客户端能保持接收预写式日志，使用这种模式不需要在主机上保存额外的预写式日志。

`-x, --xlog` 使用这个选项等效于和方法 `fetch` 一起使用 `-X`。

- `-Z --compress=level`

启用对 `tar` 文件输出的 `gzip` 压缩，并且制定压缩级别（0 到 9，0 是不压缩，9 是最佳压缩）。只有使用 `tar` 格式时压缩才可用，并且会在所有 `tar` 文件名后面自动加上后缀 `.gz`。

- `-z`

启用对 `tar` 文件输出的 `gzip` 压缩，使用默认的压缩级别。只有使用 `tar` 格式时压缩才可用，并且会在所有 `tar` 文件名后面自动加上后缀 `.gz`。

- `-t, --rw-timeout`

设置备份期间 `checkpoint` 的时间限制，默认限制时间为 120s。当数据库全量 `checkpoint` 耗时较长时，可以适当增大 `rw-timeout` 限制时间。

连接参数

- `-h, --host=HOSTNAME`

指定正在运行服务器的主机名或者 Unix 域套接字的路径。

- `-p, --port=PORT`

指定数据库服务器的端口号。

可以通过 `port` 参数修改默认端口号。

- `-U, --username=USERNAME`

指定连接数据库的用户。

- `-s, --status-interval=INTERVAL`

发送到服务器的状态包的时间（以秒为单位）。

- `-w, --no-password`

不出现输入密码提示。

- `-W, --password`

当使用 `-U` 参数连接本地数据库或者连接远端数据库时，可通过指定该选项出现输入密码提示。

4.5.5 示例

```
$ gs_basebackup -D /home/test/trunk/install/data/backup -p 15400
INFO: The starting position of the xlog copy of the full build is: 0/1B800000.
The slot minimum LSN is: 0/1B800000.
```

4.6 gs_ctl

4.6.1 背景信息

`gs_ctl` 是 GBase 8s 提供的数据库服务控制工具，可以用来启停数据库服务和查询数据库状态。主要供 GBase 8s 管理模块调用。

`gs_ctl` 工具由操作系统用户 `gbase` 执行。

- 启动、停止、重启 GBase 8s 节点。

- 在不停止数据库的情况下，重新加载配置文件（postgresql.conf, pg_hba.conf）。
- 主备切换、主备状态查询、重建和重建状态查询。

4.6.2 参数说明

gs_ctl 参数可分为 option 参数、公共参数、start 和 restart 模式的参数、stop 和 restart 模式的参数、switchover 模式的参数、build 参数、query 参数、member 参数、changerole 参数、setrunmode 参数。

表 4-1 option 参数

参数	参数说明
init	创建数据库。
start	启动数据库。
restart	重启数据库。
build	在主机上重建备实例或级联备实例。
stop	停止数据库。
reload	重载配置文件（postgresql.conf, pg_hba.conf）。
status	显示数据库运行状态。
finishredo	主机故障时，备机停止回放，强制升主。
failover	在主机异常时，将备机切换为主机。 切换成功后，需要执行 gs_om -t refreshconf 命令记录当前主备机信息，确保 gs_om -t refreshconf 命令执行成功，否则再次重启会影响数据库状态。dcf 模式下不支持该操作。

参数	参数说明
switchover	<p>在主备机正常时，出于维护的需要，将备机切换为主机，可保证切换过程中数据不丢失。</p> <p>必须在要切换为主机的备机上，执行 switchover 命令才会生效。在主机上执行 switchover 命令，仅作查询使用。</p> <p>切换成功后，需要执行 gs_om -t refreshconf 命令记录当前主备机信息，确保 gs_om -t refreshconf 命令执行成功，否则再次重启会影响数据库状态。</p> <p>switchover 命令下发后，命令如果超时返回，后台进程的执行状态可能处于不可确定状态。</p>
query	查询主备机之间的状态信息。
notify	启动后再指定主备机。
kill	给指定的进程发送信号量。
querybuild	查看数据库的重建进度。
stack	获取 gaussdb 的调用栈。

表 4-2 公共参数

参数	参数说明	取值范围
-D, --pgdata=DATADIR	指定数据目录的位置。若指定目录中包含配置文件 postgresql.conf，且配置文件内 data_directory 值与 -D 指定目录不同，将优先按	DATADIR 的取值必须为有效的数据目录。

参数	参数说明	取值范围
	data_directory 目录执行。	
-s, --silent	对 reload、restart、stop 命令仅打印部分提示信息, 对其他命令不生效。打印信息如 : server promoting、server shutting down。不打印信息如 : waiting for server to shut down、server stopped 等提示信息。	-
-t, --timeout=SECS	等待数据库启动、关闭或者主备切换完成的最大秒数。如果等待超时, 命令会结束退出, 并通知不再等待。此后, 可以通过 gs_om -t status --detail 或其他查看数据库状态的命令, 来确认数据库是否启动、关闭或主备切换成功。	取值范围: 整型 (秒)。 默认值: 等待数据库启动、停止或者主备切换完成: 60 秒。
-V, --version	打印 gs_ctl 的版本信息然后退出。	-
-w	需要等待操作完成。	默认值: 数据库启动、停止或重新启动时需要等待操作完成。
-W	不用等待操作完成。	-
-M	-M 后面需要跟	SERVERMODE 的取值范围:

参数	参数说明	取值范围
	SERVERMODE 参数, 表示在启动时指定数据库的启动模式。	<p>primary: 本端以主机模式启动。</p> <p>standby: 本端以备机模式启动。</p> <p>cascade_standby: 本端以级联备机模式启动。</p> <p>pending: 本端处于等待状态, 等待提升为主机或者备机。</p>
-T	-T 后面跟 term, 升主命令时会用此 term 作为主机 term, build 时会连接大于等于此 term 的主机进行 build	term 的取值范围, 无符号整型。
-d	打印更多调试信息	无参数
-q	生成完成后不自动启动, 需要调用者启动。	无参数
-P PASSWORD	<p>与-U 参数配合使用, 指定连接本地数据库的用户密码。</p> <p>说明:</p> <p>当省略-U 参数, 且认证方法为 trust 时, -P 参数无效。</p>	-
-U USERNAME	指定连接数据库的用户。此参数只能与 notify、query、failover、switchover 和	<p>取值范围: GBase 8s 中存在的用户。</p> <p>默认值: 省略此参数则使用与当</p>

参数	参数说明	取值范围
	querybuild 参数配合使用。 对于域用户，使用 DOMAIN\username 格式。	前操作系统用户同名的用户。
-.?, -h, --help	显示关于 gs_ctl 命令行参数的帮助信息。	-

表 4-3 start 或 restart 共有参数

参数	参数说明	取值范围
-c, --core-file	允许 gaussdb 进程产生内核文件。	-
-l, --log=FILENAME	把服务器日志输出附加在 FILENAME 文件上。	FILENAME 的取值为启动数据库服务的用户有权限的文件。 例如，data 目录下的某文件。
-o OPTIONS	声明要直接传递给由 GBase 8s 执行的 gaussdb 的命令行选项。 参数通常都用单或者双引号包围以保证它们作为一个整体传递。	gaussdb 支持的参数。
-p PATH-TO-POSTGRES	声明 gaussdb 可执行文件的位置。	缺省位于 gs_ctl 自身所在目录，一般不需要此选项。

表 4-4 stop 和 restart 共有参数

参数	参数说明	取值范围
-m, --mode=MODE	声明关闭模式。	mode 的取值： fast：不等待客户端中断连接，所有活跃事务都被回滚并且客户端都被强制断开，然后服务器将被关闭。 immediate：强行关闭，在下次重新启动的时候将导致故障恢复。 默认值：fast

表 4-5 switchover 参数

参数	参数说明	取值范围
-m SWITCHOVER-MODE	声明双机主备实例正常时切换的切换模式。	mode 的取值： fast：不等待客户端中断连接，所有活跃事务都被回滚并且客户端都被强制断开，然后服务器将被切换。 smart：本版本暂不支持该模式。 默认值：fast

表 4-6 build 参数

参数	参数说明	取值范围
-b MODE	指定重建备机的模式。	mode 的取值： <ul style="list-style-type: none"> ● full：通过全量镜像的方式重新同步 主机的数据目录。 ● incremental：通过解析 Xlog 日志获取主

参数	参数说明	取值范围
		<p>备机差异的数据进行增量修复备机。</p> <p>说明：</p> <p>增量重建适用于主备双主等因日志造成的一致场景。</p> <p>增量重建不适用于一主一备并且没有开启最大高可用的场景，此种场景下需要使用全量重建或者开启最大高可用后再进行增量重建。</p> <p>备机数据文件损坏、数据目录丢失 等故障通过增量重建的方式无法修复，此时可通过全量重建的方式重新修复备机。</p> <ul style="list-style-type: none"> ● auto（不指定）：先增量，根据失败后是否可以再增量选择继续增量或者全量，三次增量失败后进行全量。 ● standby_full：通过指定的备机 ip 和 port 全量重建故障备机。使用该参数时需要同时使用-C 蚕食指定镜像的 ip 和 port。 <p>默认值：auto</p> <p>说明：</p> <p>重建级联备机需要加上-M cascade_standby 参数。</p>
<p>-r, --recvtimeout=INT ERVAL</p>	<p>指定重建过程中备机日志接收等待主机响应的超时时间。</p>	<p>取值范围：整型（秒）。</p> <p>默认值：120 秒</p> <p>提示：超时时间需根据业务繁忙程度设置，业务繁忙等待主机的响应时间需加长，建议超时时间=checkpoint 的执行时间+1min。</p>

参数	参数说明	取值范围
-q	<p>重建结束后, 是否自动重启。</p> <p>指定参数场合, 不自动重启。</p> <p>未指定场合, 自动重启, 重启默认最大等待时间是 60 秒。如果在此时间内数据库未能启动成功, 就会报不再等待, 命令退出。此后可以通过 <code>gs_om -t status --detail</code> 或其他查看数据库状态的命令, 来确认数据库是否启动成功。</p>	无参数

表 4-7 query 参数

参数	参数说明	取值范围
-L	查询 lsn 并展示最大长度。	无参数

表 4-8 member 参数

参数	参数说明	取值范围
--operation	GBase 8s 数据库实例, DCF	add: 向数据库实例的 DCF 节点

参数	参数说明	取值范围
	模式下，增删节点的操作。	配置中增加一个节点。 remove：从数据库实例 DCF 节点配置中删除一个节点。
-u, --nodeid	新增节点的 DCF NODE ID。	无符号整型
-i, --ip	新增节点与数据库实例 DCF 模块通信的 IP。	字符串。
-e, --port	新增节点与数据库实例 DCF 模块通信的 port。	整型。
[-D, --pgdata=]DATADIR	指定数据目录的位置。若指定目录中包含配置文件 postgresql.conf，且配置文件内 data_directory 值与 -D 指定目录不同，将优先按 data_directory 目录执行。	DATADIR 的取值：用户自定义。

表 4-9 changerole 参数

参数	参数说明	取值范围
-R, --role	GBase 8s 数据库实例，DCF 模式下节点角色（需要在角色为 primary 的节点上使用）。	passive：passive 角色。 follower：follower 角色。
-u, --nodeid	新增节点的 DCF NODE ID。	无符号整型。

参数	参数说明	取值范围
[-D, --pgdata=]DATADIR	指定数据目录的位置。若指定目录中包含配置文件 postgresql.conf, 且配置文件内 data_directory 值与 -D 指定目录不同, 将优先按 data_directory 目录执行。	DATADIR 的取值: 用户自定义。

表 4-10 setrunmode 参数

参数	参数说明	取值范围
-v, --votenum	投票个数。	无符号整型。
-x, --xmode	运行模式 (在少数派的 DN 节点上使用)。	minority: 少数派强起。 normal: 加回。
[-D, --pgdata=]DATADIR	指定数据目录的位置。若指定目录中包含配置文件 postgresql.conf, 且配置文件内 data_directory 值与 -D 指定目录不同, 将优先按 data_directory 目录执行。	DATADIR 的取值: 用户自定义。

表 4-11 stack 参数

参数	参数说明	取值范围
-l	用于指定需要获取调用栈的线程的 lwtid。	正整数。

4.7 gs_initdb

4.7.1 背景信息

gs_initdb 初始化数据库时，会创建数据库目录、生成系统表、创建默认数据库和模板数据库。

4.7.2 系统表

初始化数据库时会生成大量的系统表和视图，其中绝大部分都对任何数据库用户开放查看权限。

说明

- pg_user_status、pg_auth_history 系统表权限只对初始化数据库用户和 sysadmin 用户开放。

生成的数据库

- template1：是一个模板数据库，当以后再创建一个新的数据库时，template1 数据库里的所有内容都会拷贝到新数据库中。通过 gs_initdb 的参数可以决定 template1 数据库的设置。
- template0：是 GBase 8s 提供的最初级的备份数据库，当需要时可用 template0 作为模板生成“干净”的数据库。
- postgres：是一个提供给用户、工具和第三方应用的缺省数据库。

4.7.3 使用指导

在安装时，推荐使用-D 参数调用 gs_initdb 初始化数据库。如果由于故障恢复等原因，需要重新初始化一个数据库，可以通过执行 gs_initdb 来完成。

- 尽管 gs_initdb 会尝试创建相应的数据目录，但可能没有权限执行此操作，因为要创建目录的父目录通常被 root 所拥有。如果要创建数据目录，首先用 root 用户创建一个空数据目录，然后用 chown 把该目录的所有权交给数据库用户。
- gs_initdb 决定 template1 数据库的设置，而该设置将会成为其他数据库的默认设置。

- `gs_initdb` 初始化数据库的缺省区域和字符集编码。字符集编码、字符编码排序 (LC_COLLATE) 和字符集类 (LC_CTYPE, 如大写、小写数字等) 可以在创建数据库时独立设置。

4.7.4 语法

```
gs_initdb [OPTION]... [DATADIR]
```

4.7.5 操作步骤

- (1) 以操作系统用户 `gbase` 登录数据库主节点。
- (2) 规划数据库目录。

切换到 `root` 用户。

```
su - root
```

根据提示输入用户名和密码。

- (3) 进入 `/opt/gbase` 目录, 并创建“`data1`”目录。

```
cd /opt/gbase
mkdir data1
```

将 `data1` 目录的所有权分配给数据库用户 `gbase`。

```
chown gbase:gbase data1
```

退出 `root` 用户。

```
exit
```

- (4) 执行 `gs_initdb` 命令初始化数据库。

```
gs_initdb -D /opt/gbase/data1 -w "GBase@123" --nodename='data1'
```

4.7.6 参数说明

`gs_initdb` 支持的常用参数和不常用参数请分别参见以下两表。

表 4-12 常用参数说明

参数	参数说明	取值范围
----	------	------

参数	参数说明	取值范围
-A, --auth=METHOD	<p>指定本地用户连接数据库时的认证方法，即 pg_hba.conf 配置文件中 host 和 local 所在行的认证方法。</p> <p>除非用户对本地用户都是信任的，否则不要使用默认值 trust。</p> <p>须知： 若取值为 md5，则需手动修改参数文件 postgresql.conf.sample 中的密码存储类型 password_encryption_type 参数的值，修改为 0，且放开注释使之生效。gs_initdb 工具需同时配合 -W 的使用。</p>	<p>METHOD 的取值：</p> <ul style="list-style-type: none"> ● trust ● reject ● md5（不安全的算法，为了兼容老版本而存在） ● sha256 ● sm3 <p>默认值：trust</p>
--auth-host=METHOD	<p>指定本地用户通过 TCP/IP 连接数据库时的认证方法，即： pg_hba.conf 配置文件中 host 所在行的认证方法。</p> <p>指定此参数则会覆盖 -A 参数的值。</p>	<p>METHOD 的取值：</p> <ul style="list-style-type: none"> ● trust ● reject ● md5（不安全的算法，为了兼容老版本而存在） ● sha256 ● sm3 <p>默认值：trust</p>
--auth-local=METHOD	<p>指定本地用户通过 Unix 域套接字连接数据库时的认证方法，</p>	<p>METHOD 的取值：</p> <ul style="list-style-type: none"> ● trust

参数	参数说明	取值范围
	<p>即 pg_hba.conf 配置文件中 local 所在行的认证方法。</p> <p>指定此参数则会覆盖 -A 参数的值。</p>	<ul style="list-style-type: none"> ● reject ● md5 (不安全的算法, 为了兼容老版本而存在) ● sha256 ● sm3 ● peer (仅用于 local 模式) <p>默认值: trust</p>
-c, --enable-dcf	设置安装的节点为 DCF 模式	-
[-D, --pgdata=]DATADIR	指定数据目录的位置。	DATADIR 的取值: 用户自定义。不能包括 “ ”, “;”, “&”, “\$”, “<”, “>”, “””, “\”, “!” 这几个字符。
--nodename=NODENAME	初始化的节点名称。	<p>节点的命名需要遵守如下规范:</p> <ul style="list-style-type: none"> ● 节点名称必须为小写字母 (a-z)、下划线 (_)、特殊符号 #、数字 (0-9)。 ● 节点名称必须以小写字母 (a-z) 或下划线 (_) 开头。 ● 节点名称不能为空, 且最大的长度为 64 个字符。

参数	参数说明	取值范围
<p>-E, --encoding=ENCODING</p>	<p>为新数据库设置编码格式。</p>	<p>如果使用此参数, 需要加上--locale 选项指定支持此编码格式的区域。如果不加--locale 选项, 则采用系统默认的区域, 如果系统默认区域的编码格式和用此参数指定的编码格式不匹配则会导致数据库初始化失败。</p> <p>如果不指定此参数, 则使用系统默认区域的编码格式。系统默认区域和编码格式可以使用 locale 命令查看, 如下:</p> <pre>locale grep LC_CTYPE LC_CTYPE="en_US.UTF-8"</pre> <p>其中 UTF-8 表示系统默认区域的编码格式。</p> <p>不能包括“ ”, “;”, “&”, “\$”, “<”, “>”, “””, “\”, “!”这几个字符。</p>
<p>--locale=LOCALE</p>	<p>为新数据库设置缺省的区域。可以用 locale -a 查看可用的区域, 如 zh_CN.gbk 等。如果不希望指定特定的区域, 则可以用 C。</p>	<p>不能包括“ ”, “;”, “&”, “\$”, “<”, “>”, “””, “\”, “!”这几个字符。</p> <p>例如用户要将数据库编码格式初始化为 GBK,</p>

参数	参数说明	取值范围
	<p>须知:</p> <p>如果用户设置了数据库的编码格式, 则用户选择区域的编码格式必须与用户设置的编码格式一致, 否则数据库初始化会失败。</p>	<p>可以采用如下步骤:</p> <p>用 <code>locale -a grep gbk</code> 命令查看系统支持 <code>gbk</code> 编码的区域, 如下:</p> <pre>locale -a grep gbk zh_CN.gbk zh_SG.gbk</pre> <p>初始化数据库时加入 <code>--locale=zh_CN.gbk</code> 选项。</p>
<code>--dbcompatibility=DB COMPATIBILITY</code>	指定兼容的数据库的类型。	取值范围: A、B、C、PG。分别表示兼容 O、MY、TD 和 POSTGRES。
<code>--lc-collate=LOCALE</code> <code>--lc-ctype=LOCALE</code> <code>--lc-messages=LOCALE</code> <code>--lc-monetary=LOCALE</code> <code>--lc-numeric=LOCALE</code> <code>--lc-time=LOCALE</code>	为新数据库设置指定范畴的区域。	<p>各参数的取值必须是操作系统支持的值。不能包括不能包括“ ”, “;”, “&”, “\$”, “<”, “>”, “'”, “\”, “!”这几个字符。</p> <p>说明:</p> <p>如果用户在数据库安装的时候没有指定 <code>--lc-collate</code> 参数, 则 <code>--lc-collate</code> 参数的默认值为 C。</p>
<code>--no-locale</code>	和 <code>--locale=C</code> 等价。	-

参数	参数说明	取值范围
--pwfile=FILE	gs_initdb 时从文件 FILE 中读取数据库中系统管理员的密码。该文件的第一行将被当作密码使用。	FILE 可以是“相对路径+文件”的形式，也可以是“绝对路径+文件”的形式。相对路径是相对当前路径的。不能包括“ ”，“;”，“&”，“\$”，“<”，“>”，“””，“\”，“!”这几个字符。
-T, --text-search-config=CFG	设置缺省的文本搜索方式。此配置项的值不会做正确性校验，配置成功后，有日志记录提醒当前配置项的取值。	text-search-config 的取值： <ul style="list-style-type: none"> ● english 全文搜索 ● simple 普通文本搜索 默认值：simple
-U, --username=NAME	选择数据库系统管理员的用户名。	取值范围：正常的数据库用户。不能包括“ ”，“;”，“&”，“\$”，“<”，“>”，“””，“\”，“!”这几个字符。 默认值：运行 gs_initdb 的操作系统用户。
-W, --pwprompt	gs_initdb 时强制交互式输入数据库管理员的密码。	-
-w, --ppasswd=PASSWD	gs_initdb 时通过命令行指定的管理员用户的密码，而不是交互式输入。	设置的密码要符合复杂度要求： <ul style="list-style-type: none"> ● 最少包含 8 个字符； ● 不能和用户名和当前

参数	参数说明	取值范围
		<p>密码(ALTER)相同, 或和当前密码反序;</p> <ul style="list-style-type: none"> 至少包含大写字母(A-Z)、小写字母(a-z)、数字、非字母数字字符(限定为 ~!@#\$\$%^&*()-_+=\ [{}];;<.>/?) 四类字符中的三类字符。
<p>-C, --enpwdfiledir=DIR</p>	<p>gs_initdb 时指定的经 AES128 加密算法加密过的密码文件所在目录。gs_initdb 会将该目录下密码文件进行解密, 同时把解密后的密码做密码复杂度校验, 校验通过的话会将此密码作为用户的密码。</p> <p>说明: 加密密码文件需使用 gs_guc 工具生成:</p> <pre>gs_guc encrypt -K GBase@123 -D Dir</pre> <p>如果用户指定多个 -w 和 -C 参数, gs_initdb 会将用户输入的最后一个 -w 或 -C 参数作为用户的需求, 即输入密码的明文或经过 AES128 加密后的密码。</p>	<p>不能包括“ ”, “;”, “&”, “\$”, “<”, “>”, “'”, “\”, “!”这几个字符。</p>
<p>-X,</p>	<p>声明事务日志存储的目录。</p>	<p>只支持绝对路径。不能包括“ ”, “;”, “&”, “\$”, “<”,</p>

参数	参数说明	取值范围
--xlogdir=XLOGDIR	所设置的目录，必须满足运行 GBase 8s 的用户有读写权限。	“>”，“”，“\”，“!”这几个字符。
-S, --security	安全方式初始化数据库。	以-S 方式初始化的数据库后，创建的数据库用户权限受到限制，默认不再具有 public schema 的使用权限。

表 4-13 不常用参数说明

参数	参数说明	取值范围
-d, --debug	从初始化后端打印调试输出信息。初始化后端是 gs_initdb 用于创建系统表的程序。	-
-L DIRECT ORY	gs_initdb 初始化数据库所需要的输入文件的路径。通常是不必要的。如果需要明确声明的话，程序会提示输入。因该参数意为创建指定配置信息的数据库，建议将 share/postgresql 下的所有涉及启动相关的子目录及文件全部复制过来，避免其他因素的影响。	初始化数据库所需的输入文件路径。不能包括“ ”，“;”，“&”，“\$”，“<”，“>”，“”，“\”，“!”这几个字符。
-n, --noclean	不指定该参数时，当 gs_initdb 发现一些错误妨碍它完成创建数据库的工作时，它将在检测到不能结束工作之	-

参数	参数说明	取值范围
	前将其创建的所有文件删除。这个选项禁止任何清理动作，因而对调试很有用。	
-s, --show	显示内部设置。	-
-H, --host-ip	初始化 GBase 8s 节点 node_host。	不能包括“ ”, “;”, “&”, “\$”, “<”, “>”, “””, “\”, “!”这几个字符。
-V, --version	打印 gs_initdb 版本信息，然后退出。	-
?, --help	显示关于 gs_initdb 命令行参数的帮助信息，然后退出。	-

4.8 gs_install

4.8.1 背景信息

数据库的部署是一个复杂的过程。GBase 8s 提供了 gs_install 工具来帮助完成 GBase 8s 的安装和部署。

GBase 8s 安装部署，要求用户指定配置文件，配置文件中会指定程序安装路径、实例数据目录、主备关系、实例数、各实例的业务 IP 端口等信息。

4.8.2 前提条件

- 已成功执行前置脚本 gs_preinstall。
- 用户需确保各个节点上的 locale 保持一致。
- 需要使用前置时设置的 GBase 8s 用户进行安装操作。

4.8.3 语法

- 安装 GBase 8s

```
gs_install -X XMLFILE [--gsinit-parameter="PARAMETER" [...]]  
[--dn-guc="PARAMETER" [...]] [--alarm-component=ALARMCOMPONENT]  
[--time-out=SECS] [-l LOGFILE]
```

说明

- 安装时若不指定字符集，默认字符集为 SQL_ASCII，为简化和统一区域 locale 默认设置为 C，若想指定其他字符集和区域，请在安装时使用参数 --gsinit-parameter="--locale=LOCALE"来指定，LOCALE 为新数据库设置缺省的区域。

- 显示帮助信息

```
gs_install -? | --help
```

- 显示版本号信息

```
gs_install -V | --version
```

4.8.4 参数说明

- -X

GBase 8s 配置文件。

取值范围：xml 配置文件。

- -l

指定安装日志文件及日志文件存放的路径。

当不明确指定-l，但在 XML 文件中配置了 gaussdbLogPath 时，默认值为“gaussdbLogPath 的值、用户名和 om/gs_install-YYYY-MM-DD_hhmmss.log”的组合。

- --gsinit-parameter=PARAM

数据实例参数指定。

取值范围请参见 gs_initdb 参数说明，其中对-A、-D、-U、-C、-X 参数的设置不生效。

- `--dn-guc=PARAM`

配置参数。

取值范围请参见 `gs_guc` 的参数说明。

- `--time-out=SECS`

启动超时等待时间。单位：s。

若`--time-out`取值过小，有可能会因为需要启动的实例数过多而引起超时。若在超时时间内启动不成功，则会报错启动超时，但是 GBase 8s 会在后台继续启动。可以等待一段时间后，查询 GBase 8s 状态，检查 GBase 8s 是否启动成功。

取值范围：正整数，若无特殊需求，不需要配置该参数，系统会自动调整超时时间。

- `--alarm-component=alarm_component`

指定告警上报组件的绝对路径。

- `-, --help`

显示帮助信息。

- `-V, --version`

显示版本号信息。

4.8.5 示例

使用 `gs_install` 脚本进行 GBase 8s 安装。

```
gs_install -X /opt/software/gbase/clusterconfig.xml
Parsing the configuration file.
Check preinstall on every node.
Successfully checked preinstall on every node.
Creating the backup directory.
Successfully created the backup directory.
begin deploy..
Installing the cluster.
begin prepare Install Cluster..
Checking the installation environment on all nodes.
begin install Cluster..
Installing applications on all nodes.
Successfully installed APP.
```

```
begin init Instance..
encrypt cipher and rand files for database.
Please enter password for database:
Please repeat for database:
begin to create CA cert files
The sslcert will be generated in /opt/gbase/cluster/app/share/sslcert/om
Cluster installation is completed.
Configuring.
Deleting instances from all nodes.
Successfully deleted instances from all nodes.
Checking node configuration on all nodes.
Initializing instances on all nodes.
Updating instance configuration on all nodes.
Check consistence of memCheck and coresCheck on DN nodes.
Successful check consistence of memCheck and coresCheck on all nodes.
Configuring pg_hba on all nodes.
Configuration is completed.
Successfully started cluster.
Successfully installed application.
```

4.8.6 相关命令

[gs_postuninstall](#), [gs_preinstall](#), [gs_uninstall](#)

4.9 gs_postuninstall

4.9.1 背景信息

GBase 8s 提供了 `gs_postuninstall` 工具来帮助清理准备 GBase 8s 环境阶段所做配置，使得卸载后的环境得到清理。

4.9.2 前提条件

- GBase 8s 卸载执行成功。
- root 用户互信可用。
- 只能使用 root 用户执行 `gs_postuninstall` 命令。

4.9.3 语法

- GBase 8s 清理用户、用户组以及虚拟 IP 清理

```
gs_postuninstall -U USER -X XMLFILE [-L] [--delete-user] [--delete-group] [-l LOGFILE]
```

- 显示帮助信息

```
gs_postuninstall -? | --help
```

- 显示版本号信息

```
gs_postuninstall -V | --version
```

4.9.4 参数说明

- -U

运行 GBase 8s 的操作系统用户名。

取值范围：字符串，要符合标识符的命名规范。

- -X

GBase 8s 配置文件路径。

取值范围：xml 文件的存储路径。

- -L

只清理本主机的环境。

如果 GBase 8s 内某主机做单机环境清理后，GBase 8s 不能再做全量环境清理。

- --delete-user

删除-U 参数指定的操作系统用户。

如果在 Redhat 环境下，且用户名与用户组名相同，选择此项必须指定--delete-group 参数。

- --delete-group

删除操作系统用户所在的用户组（选择此选项必须指定--delete-user 参数）。

- -l

指定日志文件名及路径。在内部会自动给日志名添加一个时间戳。

当既不明确指定 `-l`，又不在 XML 文件中配置 `gaussdbLogPath` 时，默认值为：“`/var/log/gaussdb/om/gs_local-YYYY-MMDD_hhmmss.log`”。

说明

➤ 由于在执行 `gs_postuninstall` 后，系统会自动删除 GBase 8s 相关目录（包含 `$GAUSSLOG` 目录）。因此建议用户通过该参数指定日志文件到非 GBase 8s 相关路径。

- `-?, --help`

显示帮助信息。

- `-V, --version`

显示版本号信息。

4.9.5 示例

清理主机的环境

```
[root@gbasehost ~]$ gs_postuninstall -U gbase -X
/opt/software/gbase/clusterconfig.xml --delete-user
Parsing the configuration file.
Successfully parsed the configuration file.
Check log file path.
Successfully checked log file path.
Checking unpreinstallation.
Successfully checked unpreinstallation.
Deleting the instance's directory.
Successfully deleted the instance's directory.
Deleting the installation directory.
Successfully deleted the installation directory.
Deleting the temporary directory.
Successfully deleted the temporary directory.
Deleting remote OS user.
Successfully deleted remote OS user.
Deleting software packages and environmental variables of other nodes.
Successfully deleted software packages and environmental variables of other
nodes.
```

```
Deleting logs of other nodes.
Successfully deleted logs of other nodes.
Deleting software packages and environmental variables of the local node.
Successfully deleted software packages and environmental variables of the local
nodes.
Deleting local OS user.
Successfully deleted local OS user.
Deleting local node's logs.
Successfully deleted local node's logs.
Successfully cleaned environment.
```

4.9.6 相关命令

[gs_preinstall](#), [gs_uninstall](#)

4.10 gs_preinstall

4.10.1 背景信息

GBase 8s 提供了 `gs_preinstall` 工具来帮助完成 GBase 8s 的环境配置，以保证 GBase 8s 安装的正常进行。

4.10.2 注意事项

- 用户需要检查上层目录权限，保证安装用户对安装包和配置文件目录读写执行的权限。
- xml 文件中各主机的名称与 IP 映射配置正确。
- 只能使用 root 用户执行 `gs_preinstall` 命令。
- 执行 `gs_preinstall` 会清理 `/etc/hosts` 中的 GBase 8s 映射信息，可能存在已部署数据库连接丢失风险，可以在安装结束后手动添加其它数据库映射至 `/etc/hosts` 文件中。

4.10.3 语法

- 准备 GBase 8s 环境

```
gs_preinstall -U USER -G GROUP -X XMLFILE [-L] [--skip-os-set]
[--env-var="ENVVAR" [...]] [--sep-env-file=MPPRCFILE] [--skip-hostname-set] [-l
LOGFILE] [--non-interactive]
```


- 显示帮助信息

```
gs_preinstall -? | --help
```

- 显示版本号信息

```
gs_preinstall -V | --version
```

4.10.4 参数说明

- -U

运行 GBase 8s 的操作系统用户名。

取值范围：字符串，要符合标识符的命名规范。

说明

- 配置-U 参数值时，字符串中不能包含“|”、“;”、“&”、“\$”、“<”、“>”、“”、“\|”、“”、“\”、“{”、“}”、“(”、“)”、“[”、“]”、“~”、“*”、“?”特殊字符。

- -G

运行 GBase 8s 的操作系统用户的群组名。

取值范围：字符串，要符合标识符的命名规范。

- -X

GBase 8s 配置文件路径。

取值范围：xml 文件的存储路径。

- -L

指定该参数则仅给当前节点准备好 GBase 8s 安装环境。适用于不允许使用 root 互信的场景和设置安全加固参数 PermitRootLogin 为 no 的场景。需要满足如下前提条件：

- 安装包和 GBase 8s 配置文件已提前上传到所有主机；
- GBase 8s 安装用户已提前创建好并建立好互信关系；
- 用户需要自己保证各台主机上所执行命令的一致性；
- 手动将所有节点的主机名和 ip 映射关系写入各个主机的/etc/hosts, 并在每

个映射关系后边加入注释内容：`#Gauss OM IP Hosts Mapping`。

- `--skip-os-set`

是否设置操作系统参数。默认设置系统参数。如果指定该参数则不设置。

- `--env-var="ENVVAR"`

配置普通用户环境变量。可以指定多个。

取值范围：字符串。

 说明

- 本参数不能指定为 GBase 8s 默认创建的环境变量。否则，GBase 8s 的环境变量将被覆盖。GBase 8s 默认创建的环境变量见 4-3。

- `--sep-env-file=MPPRCFILE`

保存环境变量的文件。如果指定该参数，表示将使用环境变量分离的版本。

 说明

- 不能指定环境变量文件为 GBase 8s 相关目录或者和这些目录同名，不能和 GBase 8s 用户的目录相同。

- `--skip-hostname-set`

是否将 xml 配置文件中主机名与 IP 的映射关系写入“/etc/hosts”文件中。默认写入，如果指定该参数则不写入。

- `-l`

指定日志文件名及路径。在内部会自动给日志名添加一个时间戳。

默认值：`/var/log/gbase/用户名/om/gs_preinstall.log`

当不明确指定 `-l`，但在 XML 文件中配置了 `gaussdbLogPath` 时，默认值为 `gaussdbLogPath` 的值、用户名和 `om/gs_preinstall-YYYY-MM-DD_hhmmss.log` 的组合。

- `--non-interactive`

指定前置执行模式。

当不指定该参数时，则为安全交互模式，在此模式下用户需要人机交互输入密码。

当指定该参数时，为非交互模式，不需要进行人机交互。

- `-, --help`

显示帮助信息。

- `-V, --version`

显示版本号信息。

表 4-14 GBase 8s 默认创建的环境变量

环境变量名称	说明
MPPDB_ENV_SEPARATE_PATH	GBase 8s 环境变量分离文件路径
GPHOME	GBase 8s 工具目录
PATH	GBase 8s 工具脚本目录
LD_LIBRARY_PATH	GBase 8s 引用第三方动态库路径
PYTHONPATH	python 软件路径
GAUSS_WARNING_TYPE	告警类型
GAUSSHOME	GBase 8s 安装路径
GAUSS_VERSION	GBase 8s 版本号
PGHOST	GBase 8s 用户的临时目录路径
GS_CLUSTER_NAME	GBase 8s 名称

环境变量名称	说明
GAUSSLOG	GBase 8s 日志路径
GAUSS_ENV	GBase 8s 环境变量标识

手动设置服务端 SYSLOG 配置

设置“/etc/syslog-ng/syslog-ng.conf”文件，在文件中添加如下内容：

```
template t_gaussdb {template("$DATE $SOURCEIP
$MSGONLY\n");template_escape(no);};
source s_gaussdb{ udp(); };
filter f_gaussdb { level(err, crit) and match('GaussDB'); };
destination d_gaussdb { file("/var/log/syslog_MPPDB", template(t_gaussdb)); };
log { source(s_gaussdb); filter(f_gaussdb); destination(d_gaussdb); };
```

设置“/etc/sysconfig/syslog”文件，在文件中添加如下内容：

```
SYSLOGD_OPTIONS="-r -m 0"
KLOGD_OPTIONS="-x"
```

说明

- 该配置需要在 GBase 8s 每台机器上都要修改。

4. 10. 5 示例

使用非交互方式进行 GBase 8s 环境准备：

```
[root@gbasehost ~]$ gs_preinstall -U gbase -G gbase -X
/opt/software/gbase/clusterconfig.xml --non-interactive
Parsing the configuration file.
Successfully parsed the configuration file.
Installing the tools on the local node.
Successfully installed the tools on the local node.
Distributing package.
Successfully distributed package.
Installing the tools in the cluster.
Successfully installed the tools in the cluster.
Checking hostname mapping.
Successfully checked hostname mapping.
```

```
Checking OS version.
Successfully checked OS version.
Creating cluster's path.
Successfully created cluster's path.
Set and check OS parameter.
Setting OS parameters.
Successfully set OS parameters.
Warning: Installation environment contains some warning messages.
Please get more details by "/package_r8/sudo/gbase_checkos -i A -h
host179174,host179175,host179176".
Set and check OS parameter completed.
Preparing CRON service.
Successfully prepared CRON service.
Preparing SSH service.
Successfully prepared SSH service.
Setting user environmental variables.
Successfully set user environmental variables.
Configuring alarms on the cluster nodes.
Successfully configured alarms on the cluster nodes.
Setting the dynamic link library.
Successfully set the dynamic link library.
Setting finish flag.
Successfully set finish flag.
Preinstallation succeeded.
```

使用交互模式进行 GBase 8s 环境准备：

```
[root@gbasehost ~]$ gbase_preinstall -U gbase -G gbase -X
/opt/software/gbase/clusterconfig.xml
Parsing the configuration file.
Successfully parsed the configuration file.
Installing the tools on the local node.
Successfully installed the tools on the local node.
Are you sure you want to create trust for root (yes/no)? yes
Please enter password for root.
Password:
Creating SSH trust for the root permission user.
Please enter password for current user[root].
Password:
Checking network information.
All nodes in the network are Normal.
Successfully checked network information.
```

```
Creating SSH trust.
Creating the local key file.
Successfully created the local key files.
Appending local ID to authorized_keys.
Successfully appended local ID to authorized_keys.
Updating the known_hosts file.
Successfully updated the known_hosts file.
Appending authorized_key on the remote node.
Successfully appended authorized_key on all remote node.
Checking cgbaseon authentication file content.
Successfully checked cgbaseon authentication content.
Distributing SSH trust file to all node.
Successfully distributed SSH trust file to all node.
Verifying SSH trust on all hosts.
Successfully verified SSH trust on all hosts.
Successfully created SSH trust.
Successfully created SSH trust for the root permission user.
Distributing package.
Successfully distributed package.
Are you sure you want to create the user[gbase] and create trust for it (yes/no)?
yes
Please enter password for cluster user.
Password:
Please enter password for cluster user again.
Password:
Creating [gbase] user on all nodes.
Successfully created [gbase] user on all nodes.
Installing the tools in the cluster.
Successfully installed the tools in the cluster.
Checking hostname mapping.
Successfully checked hostname mapping.
Creating SSH trust for [gbase] user.
Please enter password for current user[gbase].
Password:
Checking network information.
All nodes in the network are Normal.
Successfully checked network information.
Creating SSH trust.
Creating the local key file.
Successfully created the local key files.
Appending local ID to authorized_keys.
```

```
Successfully appended local ID to authorized_keys.  
Updating the known_hosts file.  
Successfully updated the known_hosts file.  
Appending authorized_key on the remote node.  
Successfully appended authorized_key on all remote node.  
Checking cgbaseon authentication file content.  
Successfully checked cgbaseon authentication content.  
Distributing SSH trust file to all node.  
Successfully distributed SSH trust file to all node.  
Verifying SSH trust on all hosts.  
Successfully verified SSH trust on all hosts.  
Successfully created SSH trust.  
Successfully created SSH trust for [gbase] user.  
Checking OS version.  
Successfully checked OS version.  
Creating cluster's path.  
Successfully created cluster's path.  
Set and check OS parameter.  
Setting OS parameters.  
Successfully set OS parameters.  
Warning: Installation environment contains some warning messages.  
Please get more details by "/package_r8/sudo/g_s_checkos -i A -h  
host179174,host179175,host179176".  
Set and check OS parameter completed.  
Preparing CRON service.  
Successfully prepared CRON service.  
Preparing SSH service.  
Successfully prepared SSH service.  
Setting user environmental variables.  
Successfully set user environmental variables.  
Configuring alarms on the cluster nodes.  
Successfully configured alarms on the cluster nodes.  
Setting the dynamic link library.  
Successfully set the dynamic link library.  
Setting finish flag.  
Successfully set finish flag.  
Preinstallation succeeded.
```

4. 10. 6相关命令

[gs_install](#), [gs_postuninstall](#)

4.11 gs_sshexkey

4.11.1 背景信息

GBase 8s 在安装过程中，需要在 GBase 8s 中的节点间进行执行命令、传送文件等操作。因此，安装前需要确保互信是连通的。GBase 8s 提供了 `gs_sshexkey` 工具来帮助用户建立互信。

须知

- `root` 用户互信可能会存在安全隐患，因此建议用户在使用完成后，立即删除各主机上 `root` 用户的互信。

使用 `gs_sshexkey` 工具建立互信成功之后，会在用户目录下面生成 `gaussdb_tmp` 目录，该目录用于存放 `ssh-agent` 进程文件 `gauss_socket_tmp` 和加密私钥的密码短语文件（随机生成 32 位密码短语，通过 `encrypt` 工具加密），同时用户下会拉起两个 `ssh-agent` 进程，此进程用于挂载私钥自动输入对应的密码短语。进程拉起完毕之后，需要 `source` 用户的环境变量，才能使互信功能生效。

4.11.2 前提条件

- 确保 `ssh` 服务打开。
- 确保 `ssh` 端口不会被防火墙关闭。
- 确保 `xml` 文件中各主机名称和 IP 配置正确。
- 确保所有机器节点间网络畅通。
- 如果为普通用户建立互信，需要提前在各主机创建相同用户并设置密码。
- 执行命令前，需要检查是否所有集群节点都支持 `expect` 命令，如果不支持请自行下载 `expect` 工具，建议使用 `expect version 5.45`。
- 如果各主机安装并启动了 SELinux 服务，需要确保 `/root` 和 `/home` 目录安全上下文为默认值（`root` 目录：`system_u:object_r:home_root_t:s0`，`home` 目录：`system_u:object_r:admin_home_t:s0`）或者关闭掉 SELinux 服务。
- 检查系统 SELinux 状态的方法：执行命令 `getenforce`，如果返回结果是 `Enforcing`，说

明 SELinux 安装并启用。

检查目录安全上下文的命令：

```
ls -ldZ /root | awk '{print $4}'  
ls -ldZ /home | awk '{print $4}'
```

恢复目录安全上下文命令：

```
restorecon -r -vv /home/  
restorecon -r -vv /root/
```

4.11.3 语法

- 建立互信

```
gs_sshexkey -f HOSTFILE [...] [--skip-hostname-set] [-l LOGFILE]
```

- 显示帮助信息

```
gs_sshexkey -? | --help
```

- 显示版本号信息

```
gs_sshexkey -V | --version
```

4.11.4 参数说明

- -f

主机列表，列出所有需要建立互信主机的 IP。



说明

- 确保 hostfile 文件中只配置主机 IP（每行只写一个 IP），不包含其它信息。

- -l

指定日志文件的保存路径。

取值范围：任意存在的可访问的绝对路径。

- --skip-hostname-set

是否将“-f”参数文件中 IP 与其 hostname 的映射关系写入“/etc/hosts”文件中。默认写入，如果指定该参数则不写入。普通用户无权限编辑/etc/hosts 文件，故有无指定该参数，普通用户均不生成映射关系。

- `--no-deduplicate`

是否在配置互信修改/etc/hosts 文件的时候, 去除/etc/hosts 文件中已经配置过的主机信息, 默认是去重复。

- `-, --help`

显示帮助信息。

- `-V, --version`

显示版本号信息。

4.11.5 示例

如下是为 root 用户建立互信的示例。仅支持交互式输入密码。

用户密码相同情况下, 交互式模式使用以下命令建立互信。

```
gs_sshexkey -f /opt/software/hostfile
Please enter password for current user[root].
Password:
Checking network information.
All nodes in the network are Normal.
Successfully checked network information.
Creating SSH trust.
Creating the local key file.
Appending local ID to authorized_keys.
Successfully appended local ID to authorized_keys.
Updating the known_hosts file.
Successfully updated the known_hosts file.
Appending authorized_key on the remote node.
Successfully appended authorized_key on all remote node.
Checking cgbaseon authentication file content.
Successfully checked cgbaseon authentication content.
Distributing SSH trust file to all node.
Successfully distributed SSH trust file to all node.
Verifying SSH trust on all hosts.
Successfully verified SSH trust on all hosts.
Successfully created SSH trust.
```

4.12 gs_tar

通过 `gs_basebackup` 的压缩格式备份，主数据目录将被写入到一个名为 `base.tar` 的文件中，并且其他表空间将被以其 `OID` 命名。生成的数据文件，需要通过 `gs_tar` 命令作解压。

说明

- `gs_tar` 命令当前仅支持解压通过 `gs_basebackup` 生成的归档文件。

如果 `gs_basebackup` 制定了压缩等级，会生成以 `gz` 结尾的文件。此时需要使用 `gzip` 命令先解压缩生成 `tar` 包，之后才可以使用 `gs_tar` 命令解压生成的 `tar` 文件。

4.12.1 语法

- 显示帮助信息

```
gs_tar -? | --help
```

- 显示版本号信息

```
gs_tar -V | --version
```

4.12.2 参数说明

- `-F, --filename=FILENAME`

解压文件，必选项。

- `-D, --destination=DIRECTORY`

解压文件输出的目录，必选项。

4.12.3 示例

```
$ gs_tar -D /home/gbase/test/dn1 -F  
/home/test/trunk/install/data/backup/base.tar
```

4.13 gs_uninstall

4.13.1 背景信息

GBase 8s 提供了 `gs_uninstall` 工具来帮助完成 GBase 8s 的卸载。

4.13.2 语法

- 卸载 GBase 8s

```
gs_uninstall [--delete-data] [-L] [-l LOGFILE]
```

- 显示帮助信息

```
gs_uninstall -? | --help
```

- 显示版本号信息

```
gs_uninstall -V | --version
```

4.13.3 参数说明

- --delete-data

删除数据文件。

- -L

只卸载本地主机。如果 GBase 8s 内某主机做单点卸载后，GBase 8s 不能再做全量卸载。

- -l

指定日志文件名及可访问的绝对路径。在内部会自动给日志名添加一个时间戳。

当既不明确指定 -l，又不在 XML 文件中配置 gaussdbLogPath 时，默认值为“\$GAUSSLOG/om/gs_uninstall-YYYY-MM-DD_hhmmss.log”。

当不明确指定 -l，但在 XML 文件中配置了 gaussdbLogPath 时，默认值为“gaussdbLogPath/用户名/om/gs_uninstall-YYYY-MM-DD_hhmmss.log”。

说明

- 由于在执行 gs_uninstall 时，系统会自动删除 GBase 8s 相关目录。因此建议用户通过该参数指定日志文件到非 GBase 8s 相关路径。

- -?, --help

显示帮助信息。

- -V, --version

显示版本号信息。

4.13.4 示例

使用数据库用户执行 `gs_uninstall` 脚本进行卸载 GBase 8s。

```
$ gs_uninstall --delete-data
Checking uninstallation.
Successfully checked uninstallation.
Stopping the cluster.
Successfully stopped the cluster.
Successfully deleted instances.
Uninstalling application.
Successfully uninstalled application.
Uninstallation succeeded.
```

4.13.5 相关命令

`gs_install`, `gs_postuninstall`

4.14 gs_upgradectl

4.14.1 背景信息

用户根据 GBase 8s 提供的新特性和数据库现状，确定是否对现有系统进行升级。

当前支持的升级模式为就地升级和灰度升级。升级方式的策略又分为大版本升级和小版本升级。

用户挑选升级方式后，系统会自动判断并选择合适的升级策略。

- 就地升级：升级期间需停止业务进行，一次性升级所有节点。
- 灰度升级：灰度升级支持全业务操作，也是一次性升级所有节点（GBase 8s1.1.0 版本之后的版本支持该功能）。

4.14.2 注意事项

- 升级操作不能和扩容、缩容同时执行。
- 不支持虚拟 IP。

- 升级过程中，不允许对 `wal_level`、`max_connections`、`max_prepared_transactions`、`max_locks_per_transaction` 这四个 GUC 参数的值进行修改。如果修改，会导致回滚后实例启动异常。
- 建议在数据库系统空闲情况下进行升级，尽量避开业务繁忙的时间段（可按照经验判断，如节假日等）。
- 升级前尽可能保证数据库正常。可以通过 `gs_om -t status` 查询，查询结果的 `cluster_state` 为 `Normal` 代表数据库正常。
- 升级前保证数据库互信正常，可以在任意节点上，通过 `ssh hostname` 命令，连接另外一个节点进行验证。如果各机器间互连不用输入密码，说明互信正常（通常数据库状态正常时，互信一般都是正常的）。
- 升级前后，数据库的部署方式（配置文件）不能发生变化。升级前会对部署方式进行校验，如果改变，会报错。
- 升级前要保证操作系统处于健康状态，通过 `gs_checkos` 工具可以完成操作系统状态检查。
- 就地升级需要停止业务，灰度升级支持全业务操作。
- 数据库运行正常且主数据库节点的数据完全同步到备数据库节点。
- 升级过程中不允许打开 `kerberos` 开关。
- 请不要修改安装包中解压出来的 `version.cfg` 文件。
- 如果升级过程中出现异常导致升级失败，需用户手动回滚，并且必须回滚成功后才能进行下一次升级。
- 如果升级回滚成功后，再次升级成功，未提交阶段设置的 GUC 参数将失效。
- 执行升级的过程中请不要手动设置 GUC 参数。
- 灰度升级中，升级的时候都会产生不超过 10s 的业务中断。
- 升级过程中，必须保持内核版本与 `om` 版本一致才可执行 `om` 操作。这里的一致是指，

内核代码和 om 代码都来自同一个软件包。如果执行了升级包的前置脚本却没有升级，或者升级回滚后没有执行基线包的前置脚本，就会造成内核代码和 om 代码的不一致。

- 升级过程中如果系统表新增了字段，升级后通过 \d 命令将查看不到这些新增的字段。此时通过 select 命令可以查到这些新增的字段。
- 升级需要 guc 参数 enable_stream_replication=on，该参数为 off 时不允许升级。
- 灰度升级中，业务并发要小于 200 并发读加 200 并发写的情况。
- 建议数据库节点磁盘使用率低于 80%时再执行升级操作。
- 执行 gs_upgradectl -t auto-upgrade 之后，没有提交之前，不能执行快照生成，即升级过程中不能执行快照生成。

4.14.3 语法

- 显示帮助信息

```
gs_upgradectl -? | --help
```

- 显示版本号信息

```
gs_upgradectl -V | --version
```

- 选择升级策略

```
gs_upgradectl -t chose-strategy [-l LOGFILE]
```

- 自动升级 GBase 8s

```
gs_upgradectl -t auto-upgrade -X XMLFILE [-l LOGFILE] [--grey]
```

- 自动回滚升级

```
gs_upgradectl -t auto-rollback -X XMLFILE [-l LOGFILE] [--force]
```

- 升级提交

```
gs_upgradectl -t cgbaseit-upgrade -X XMLFILE [-l LOGFILE]
```



说明

- 一旦提交操作完成，则不能再执行回滚操作。

4.14.4 参数说明

- -t
gs_upgradectl 命令的类型。
取值范围：chosed-strategy、auto-upgrade、auto-rollback 和 cgbaseit-upgrade。
- -l
用于记录升级或回滚过程中的日志信息。
取值范围：任意存在的可访问的绝对路径。
默认值：/var/log/gaussdb/用户名/om/gs_upgradectl-YYYY-MM-DD_hhmmss.log
- -?, --help
显示帮助信息。
- -V, --version
显示版本号信息。
- -X
指定 GBase 8s 配置文件。
取值范围：xml 文件的存储路径。
- --grey
使用灰度升级方式来进行升级操作。
- --force
当 GBase 8s 状态不正常，无法支持正常回滚时，用此参数进行强制回滚操作。

4.14.5 示例

示例一：升级前使用新包进行前置操作。

```
$ gs_preinstall -U gbase -G gbase -X /data/xml/3node_3c3d_1m2s_etc.d.xml
Parsing the configuration file.
Successfully parsed the configuration file.
Installing the tools on the local node.
```


Successfully installed the tools on the local node.
Setting pssh path
Successfully set core path.
Distributing package.
Begin to distribute package to tool path.
Successfully distribute package to tool path.
Begin to distribute package to package path.
Successfully distribute package to package path.
Successfully distributed package.
Installing the tools in the cluster.
Successfully installed the tools in the cluster.
Checking hostname mapping.
Successfully checked hostname mapping.
Checking OS version.
Successfully checked OS version.
Creating cluster's path.
Successfully created cluster's path.
Set and check OS parameter.
Setting OS parameters.
Successfully set OS parameters.
Set and check OS parameter completed.
Preparing CRON service.
Successfully prepared CRON service.
Preparing SSH service.
Successfully prepared SSH service.
Setting user environmental variables.
Successfully set user environmental variables.
Configuring alarms on the cluster nodes.
Successfully configured alarms on the cluster nodes.
Setting the dynamic link library.
Successfully set the dynamic link library.
Setting pssh path
Successfully set pssh path.
Setting Cgroup.
Successfully set Cgroup.
Set ARM Optimization.
Successfully set ARM Optimization.
Preinstalling secbox.
Successfully installed secbox.
Setting finish flag.
Successfully set finish flag.

Preinstallation succeeded.

示例二：使用 `gs_upgradectl` 脚本执行就地升级。

```
$ gs_upgradectl -t upgrade -X /data/xml/3node_3c3d_1m2s_etcd.xml
Static configuration matched with old static configuration files.
Performing inplace rollback.
Rollback succeeded.
Checking upgrade environment.
Successfully checked upgrade environment.
Successfully checked upgrade environment.
Successfully started cluster.
Start to do health check.
Successfully checked cluster status.
Backing up current application and configurations.
Successfully backed up current application and configurations.
Stopping the cluster.
Successfully stopped cluster.
Backing up cluster configuration.
Successfully backup hotpatch config file.
Successfully backed up cluster configuration.
Installing new binary.
Restoring cluster configuration.
Successfully restored cluster configuration.
Modifying the socket path.
Successfully modified socket path.
Switch symbolic link to new binary directory.
Successfully switch symbolic link to new binary directory.
Successfully started cluster.
Start to do health check.
Successfully checked cluster status.
Upgrade main process has been finished, user can do some check now.
Once the check done, please execute following command to cgbaseit upgrade:

gs_upgradectl -t cgbaseit-upgrade -X /data/xml/3node_3c3d_1m2s_etcd.xml
```

示例三：使用 `gs_upgradectl` 脚本执行自动回滚，撤销已经成功/失败的升级操作（升级回滚）。

```
$ gs_upgradectl -t auto-rollback -X /data/xml/3node_3c3d_1m2s_etcd.xml
Static configuration matched with old static configuration files.
Performing inplace rollback.
Checking static configuration files.
```

```
Successfully checked static configuration files.  
Restoring cluster configuration.  
Successfully rollback hotpatch config file.  
Successfully restored cluster configuration.  
Switch symbolic link to old binary directory.  
Successfully switch symbolic link to old binary directory.  
Stopping the cluster.  
Successfully stopped cluster.  
Restoring application and configurations.  
Successfully restored application and configuration.  
Restoring cluster configuration.  
Successfully rollback hotpatch config file.  
Successfully restored cluster configuration.  
Successfully started cluster.  
Successfully cleaned new install path.  
Rollback succeeded.
```

示例四：使用 `gs_upgradectl` 脚本执行升级后提交（升级提交）。

```
$ gs_upgradectl -t cgbaseit -X /data/xml/3node_3c3d_1m2s_etcd.xml  
Old cluster app path is /data/gauss/app_e67b8bcd  
Successfully Cleaned old install path.  
Cgbaseit binary upgrade succeeded.
```

4.15 gs_expansion

4.15.1 背景信息

GBase 8s 提供了 `gs_expansion` 工具对数据库的备机进行扩容。支持从单机或者一主多备最多扩容到单高可用组内一主八备。

注意事项

- 扩容后不会自动更新 `synchronous_standby_names` 参数。如果需要为该参数增加扩容的机器，请在扩容完成后手动更新。
- 扩容级联备之前要确保原集群中有处于同一 AZ（Available Zone）且状态正常的备机，或扩容级联备的同时也扩容了处于同 AZ 的备机。
- 对数据库集群进行扩容前，需要关注主机及新扩容节点 CPU、IO、网络等情况，不建议在硬件压力较大时执行扩容，否则可能导致扩容耗时较长甚至扩容失败。
- 当原集群数据量较大时，在进行扩容操作前应当在主机上先执行 `checkpoint`，否则可能

导致扩容耗时较长甚至扩容失败。

- 在单节点扩容时，如果节点 `hot_standby` 被关闭过，则需要每个备节点上也按照如下流程修改：
- 备节点需要先修改 `wal_level` 大于等于 `hot_standby` 后重启
- 然后再修改 `hot_standby` 为 `on` 重启，之后主机扩容才能成功。否则扩容时备机会启动失败

前提条件

- 数据库主机上存在 GBase 8s 安装包，在解压目录 `script` 子目录下执行 `gs_expansion` 命令进行扩容。
- 在新增的扩容备机上创建好与主机上相同的用户和用户组。
- 已存在的数据库节点和新增的扩容节点之间需要建立好 `root` 用户互信以及数据库管理用户（如 `gbase`）的互信。
- 只能使用 `root` 用户执行 `gs_expansion` 命令。
- 不允许同时在主节点上执行 `gs_dropnode` 命令删除其他备机。
- 执行扩容命令前需要通过 `source` 命令导入主机数据库的环境变量。如果当前数据库是分离环境变量方式安装，则 `source` 导入分离的环境变量。如果未进行分离，则需要 `source` 导入子用户的 `.bashrc` 配置文件。一般该文件路径为：`/home/[user]/.bashrc`。
- 扩容备机的操作系统与主机保持一致。
- 操作过程中不允许同时在其他备节点上执行主备倒换或者故障倒换的操作。
- 不允许同时执行 2 次相同的 `gs_expansion` 命令。
- 扩容备节点的操作只能在主节点上执行。

4.15.2 语法

- 备机扩容

```
gs_expansion -U USER -G GROUP -X XMLFILE -h hostlist [-L]
```

- 显示帮助信息

```
gs_expansion -? | --help
```

- 显示版本号信息

```
gs_expansion -V | --version
```

4.15.3 参数说明

- -U

运行 GBase 8s 的操作系统用户名。

新增扩容的备机用户名必须与已安装数据库的主机保持一致，且要提前创建好。

- -G

运行 GBase 8s 的操作系统用户组。

新增扩容的备机用户组必须与已安装数据库的主机保持一致。

- -X

xml 配置文件路径。

取值范围：xml 文件的存储路径。xml 文件里面需要包含已安装的数据库以及新增扩容数据库所有节点配置信息。

- -h

指定扩容备机的 IP 地址。

取值与 xml 配置文件里面的 backup 保持一致。如果有多个节点，节点之间以逗号分隔。

- -L

如果要扩容的节点已经安装了单机版的数据库，扩容时候添加 --L 参数可以跳过在新扩容备机上安装数据库的步骤，直接建立主备关系。

需要注意：

主备机器安装的数据库需要使用相同的用户和用户组，分离环境变量路径也需要保持一致。

主备机器安装时候 xml 配置里面的 gaussdbAppPath、gaussdbLogPath、gaussdbToolPath、corePath 地址需要保持一致。

扩容备机上的数据必须使用 om 方式安装，使用编译方式启动的数据库不支持与主机扩容。

新增扩容节点的数据库版本需要与主库保持一致。

- --time-out=SECS

指定在扩容时候传输软件包的超时时间，对于网络性能较差的情况下，可以适当增加超时时间，避免发送软件包超时。

默认值：300s。

单位：s。

- `-?, --help`

显示帮助信息。

- `-V, --version`

显示版本号信息。

4.15.4 示例

使用 `gs_expansion` 扩容步骤。

```
$ gs_expansion -U gbase -G gbase -X /home/gbase/cluster_config.xml -h
192.168.5.124
Start to preinstall database on new nodes.
Start to send soft to each standby nodes.
End to send soft to each standby nodes.
Start to preinstall database step.
Preinstall 192.168.5.124 success
End to preinstall database step.
End to preinstall database on new nodes.

Start to install database on new nodes.
Installing database on node 192.168.5.124:
Please enter the password of user [gbase] on node [192.168.5.124]:
Parsing the configuration file.
Check preinstall on every node.
Successfully checked preinstall on every node.
Creating the backup directory.
Successfully created the backup directory.
begin deploy..
Installing the cluster.
begin prepare Install Cluster..
Checking the installation environment on all nodes.
begin install Cluster..
Installing applications on all nodes.
Successfully installed APP.
```

```
begin init Instance..
encrypt cipher and rand files for database.
Please enter password for database:
Please repeat for database:
begin to create CA cert files
The sslcert will be generated in /opt/gbase/cluster/app/share/sslcert/om
Cluster installation is completed.
Configuring.
Deleting instances from all nodes.
Successfully deleted instances from all nodes.
Checking node configuration on all nodes.
Initializing instances on all nodes.
Updating instance configuration on all nodes.
Check consistence of memCheck and coresCheck on database nodes.
Configuring pg_hba on all nodes.
Configuration is completed.
Successfully started cluster.
Successfully installed application.
end deploy..
192.168.5.124 install success.
Finish to install database on all nodes.
Database on standby nodes installed finished.

Checking gaussdb and gs_om version.
End to check gaussdb and gs_om version.

Start to establish the relationship.
Start to build standby 192.168.5.124.
Build standby 192.168.5.124 success.
Start to generate and send cluster static file.
End to generate and send cluster static file.

Expansion results:
192.168.5.124: Success
Expansion Finish.
```

对数据库集群进行扩容前，需要关注主机及新扩容节点 cpu、io、网络等情况，不建议在硬件压力较大时执行扩容，否则可能导致扩容耗时较长甚至扩容失败。

当原集群数据量较大时，在进行扩容操作前应当在主机上先执行 **checkpoint**，否则可能导致扩容耗时较长甚至扩容失败。

4.15.5 相关命令

`gs_preinstall`, `gs_install`

4.16 `gs_dropnode`

4.16.1 背景信息

GBase 8s 提供了 `gs_dropnode` 工具从一主多备的数据库中移除不需要的备机，最多可以删除到只剩下单机。

4.16.2 注意事项

从主备数据库实例中移除当前仍可连通的备机时，会自动停止目标备机上正在运行的数据库服务，并删除备机上的 GRPC 证书（证书位置：`$GAUSSHOME/share/sslcert/grpc/`），但是不会删除备机上的应用。

如果删除后数据库实例中只剩下一个主机时，会提示建议重启当前主机，此时建议用户根据当前业务运行环境重启主机。

如果目标备机在执行操作前处于不可连通的状态，需要用户在目标备机恢复后手动停止或删除目标备机的数据库服务，并删除备机上的 GRPC 证书（证书位置：`$GAUSSHOME/share/sslcert/grpc/`）。

仅支持使用 `om` 方式安装的主备数据库实例中移除备机，不支持使用编译方式安装组建的主备数据库。

当移除的备机处于同步复制模式时，如果执行删除命令的同时主机上存在事务操作，事务提交时会出现短暂卡顿，删除完成后事务处理可继续。

当目标备机被移除后，如果暂时不确定是否需要目标备机，可以选择如下方法拒绝从目标备机的远程 `ssh` 连接，避免在目标备机上的误操作。

方式一：在当前主机上使用 `root` 用户修改 `/etc/ssh/sshd_config` 文件，添加如下记录（如果已存在 `DenyUsers` 记录，请在后面追加）

```
DenyUsers gbase@10.11.12.13
```

修改后需要重启 `ssh` 服务使其生效，修改后限制从目标备机不能使用 `gbase` 用户远程到该主机。

方式二：在当前主机上将目标备机加入到 `/etc/hosts.deny` 文件中（例如：`sshd:10.11.12.13:deny`），拒绝从目标备机的远程 ssh 连接（对所有用户生效），此方法需要系统 `sshd` 服务绑定到 `libwrap` 库。

当目标备机被移除后，如果不再需要目标备机，请在目标备机上使用 `gs_uninstall -delete-data -L` 命令单点卸载，请注意务必添加 `-L` 选项。

当目标备机被移除后，如果需要以单机方式使用目标备机且保留原数据，请在目标备机上先执行 `gs_guc set -D /gaussdb/data/dbnode -c "replconninfoX"` 其中 `/gaussdb/data/dbnode` 表示数据目录，`replconninfoX` 表示主备集群中的除本节点外的其他节点，比如一主一备则需要配置 `replconninfo1`，一主两备需要配置 `replconninfo1` 和 `replconninfo2`，以此类推；如果无需保留原数据，请先执行 `gs_uninstall -delete-data -L` 命令卸载后重新安装。

当目标备机被移除后，如果需要以备机方式使用目标备机，请参考 `gs_expansion` 命令重新将目标备机添加到集群中。

使用流式容灾功能时，不支持此工具。

4.16.3 前提条件

- 删除备节点的操作只能在主节点上执行。
- 操作过程中不允许同时在其他备节点上执行主备倒换或者故障倒换的操作。
- 不允许同时在主节点上执行 `gs_expansion` 命令进行扩容。
- 不允许同时执行 2 次相同的 `gs_dropnode` 命令。
- 执行删除操作前，需要确保主节点和备节点之间建立好 `gbase` 用户（数据库管理用户）的互信。
- 需要使用数据库管理用户（比如 `gbase`）执行该命令。
- 执行命令前需要通过 `source` 命令导入主机数据库的环境变量。如果当前数据库是分离环境变量方式安装，则 `source` 导入分离的环境变量。如果未进行分离，则需要 `source` 导入子用户的 `.bashrc` 配置文件。一般该文件路径为：`/home/[user]/.bashrc`。

4.16.4 语法

- 移除备机

```
gs_dropnode -U USER -G GROUP -h hostlist
```

- 显示帮助信息

```
gs_dropnode -? | --help
```

- 显示版本号信息

```
gs_dropnode -V | --version
```

4.16.5 参数说明

- -U

运行 GBase 8s 的操作系统用户名。

- -G

运行 GBase 8s 的操作系统用户组。

- -h

指定需要移除的备机的 IP 地址。如果有多个节点，节点之间以逗号分隔。

- -?, -help

显示帮助信息。

- -V, -version

显示版本号信息。

4.16.6 示例

使用 gs_dropnode 移除步骤。

```
$ gs_dropnode -U gbase -G gbase -h 10.11.12.13
The target node to be dropped is (['StandbyNode1'])
Do you want to continue to drop the target node (yes/no)? yes
[gs_dropnode]Start to drop nodes of the cluster.
[gs_dropnode]Start to stop the target node StandbyNode3.
[gs_dropnode]End of stop the target node StandbyNode3.
[gs_dropnode]Start to backup parameter config file on PrimaryNode.
[gs_dropnode]End to backup parameter config file on PrimaryNode.
[gs_dropnode]The backup file of PrimaryNode is
/tmp/gs_dropnode_backup20201119205019/parameter_PrimaryNode.tar
[gs_dropnode]Start to parse parameter config file on PrimaryNode.
[gs_dropnode]End to parse parameter config file on PrimaryNode.
[gs_dropnode]Start to parse backup parameter config file on PrimaryNode.
```

```
[gs_dropnode]End to parse backup parameter config file on PrimaryNode.
[gs_dropnode]Start to set postgresql config file on PrimaryNode.
[gs_dropnode]End of set postgresql config file on PrimaryNode.
[gs_dropnode]Start to backup parameter config file on StandbyNode2.
[gs_dropnode]End to backup parameter config file on StandbyNode2.
[gs_dropnode]The backup file of StandbyNode2 is
/tmp/gs_dropnode_backup20201119205019/parameter_StandbyNode2.tar
[gs_dropnode]Start to parse parameter config file on StandbyNode2.
[gs_dropnode]End to parse parameter config file on StandbyNode2.
[gs_dropnode]Start to parse backup parameter config file on StandbyNode2.
[gs_dropnode]End to parse backup parameter config file on StandbyNode2.
[gs_dropnode]Start to set postgresql config file on StandbyNode2.
[gs_dropnode]End of set postgresql config file on StandbyNode2.
[gs_dropnode]Start to get repl slot on PrimaryNode.
[gs_dropnode]Start to set repl slot on PrimaryNode.
[gs_dropnode]End of set repl slot on PrimaryNode.
[gs_dropnode]Start to set pg_hba config file on PrimaryNode.
[gs_dropnode]End of set pg_hba config file on PrimaryNode.
[gs_dropnode]Start to modify the cluster static conf.
[gs_dropnode]End of modify the cluster static conf.
[gs_dropnode]Success to drop the target nodes.
```

4.16.7 相关命令

`gs_expansion`, `gs_ctl`

4.17 `gs_probackup`

4.17.1 背景信息

`gs_probackup` 是一个用于管理 GBase 8s 数据库备份和恢复的工具。它对 GBase 8s 实例进行定期备份，以便在数据库出现故障时能够恢复服务器。

- 可用于备份单机数据库，也可对主机或者主节点数据库备机进行备份，为物理备份。
- 可备份外部目录的内容，如脚本文件、配置文件、日志文件、`dump` 文件等。
- 支持增量备份、定期备份和远程备份。
- 可设置备份的留存策略。

4.17.2 前提条件

- 可以正常连接 GBase 8s 数据库。
- 若要使用 PTRACK 增量备份,需在 postgresql.conf 中手动添加参数“enable_cbm_tracking = on”。
- 为了防止 xlog 在传输结束前被清理,请适当调高 postgresql.conf 文件中 wal_keep_segments 的值。

4.17.3 限制说明

- 备份必须由运行数据库服务器的用户执行。
- 备份和恢复的数据库服务器的主版本号必须相同。
- 如果要通过 ssh 在远程模式下备份数据库,需要在本地和远程主机安装相同主版本的数据库,并通过 ssh-copy-id remote_user@remote_host 命令设置本地主机备份用户和远程主机数据库用户的无密码 ssh 连接。
- 远程模式下只能执行 add-instance、backup、restore 子命令。
- 使用 restore 子命令前,应先停止 gaussdb 进程。
- 当存在用户自定义表空间时,备份的时候要加上 --external-dirs 参数,否则,该表空间不会被备份。
- 当备份的规模比较大时,为了防止备份过程中 timeout 发生,请适当调整 postgresql.conf 文件的参数 session_timeout、wal_sender_timeout。并且在备份的命令行参数中适当调整参数--rw-timeout 的值。
- 恢复时,使用 -T 选项把备份中的外部目录重定向到新目录时,请同时指定参数 --external-mapping。
- 当使用远程备份时,请确保远程机器和备份机器的时钟同步,以防止使用 --recovery-target-time 恢复的场合,启动 gaussdb 时有可能会失败。

- 当远程备份有效时(remote-proto=ssh), 请确保-h 和--remote-host 指定的是同一台机器。
当远程备份无效时, 如果指定了-h 选项, 请确保-h 指定的是本机地址或本机主机名。
- 当前暂不支持备份逻辑复制槽。

4.17.4 命令说明

- 打印 gs_probackup 版本。

```
gs_probackup -V|--version  
gs_probackup version
```

- 显示 gs_probackup 命令的摘要信息。如果指定了 gs_probackup 的子命令, 则显示可用于此子命令的参数的详细信息。

```
gs_probackup -?|--help  
gs_probackup help [command]
```

- 初始化备份路径_backup-path_中的备份目录, 该目录将存储已备份的内容。如果备份路径_backup-path_已存在, 则_backup-path_必须为空目录。

```
gs_probackup init -B backup-path [--help]
```

- 在备份路径_backup-path_内初始化一个新的备份实例, 并生成 pg_probackup.conf 配置文件, 该文件保存了指定数据目录_pgdata-path_的 gs_probackup 设置。

```
gs_probackup add-instance -B backup-path -D pgdata-path --instance=instance_name  
[-E external-directories-paths]  
[remote_options]  
[--help]
```

- 在备份路径_backup-path_内删除指定实例相关的备份内容。

```
gs_probackup del-instance -B backup-path --instance=instance_name  
[--help]
```

- 将指定的连接、压缩、日志等相关设置添加到 pg_probackup.conf 配置文件中, 或修改已设置的值。不推荐手动编辑 pg_probackup.conf 配置文件。

```
gs_probackup set-config -B backup-path --instance=instance_name  
[-D pgdata-path] [-E external-directories-paths] [--archive-timeout=timeout]  
[--retention-redundancy=retention-redundancy]  
[--retention-window=retention-window] [--wal-depth=wal-depth]
```

```
[--compress-algorithm=compress-algorithm] [--compress-level=compress-level]
[-d dbname] [-h hostname] [-p port] [-U username]
[logging_options] [remote_options]
[--help]
```

- 将备份相关设置添加到 `backup.control` 配置文件中，或修改已设置的值。

```
gs_probackup set-backup -B backup-path --instance=instance_name -i backup-id
[--note=text] [pinning_options]
[--help]
```

- 显示位于备份目录中的 `pg_probackup.conf` 配置文件的内容。可以通过指定 `--format=json` 选项，以 `json` 格式显示。默认情况下，显示为纯文本格式。

```
gs_probackup show-config -B backup-path --instance=instance_name
[--format=plain|json]
[--help]
```

- 显示备份目录的内容。如果指定了 `instance_name` 和 `backup_id`，则显示该备份的详细信息。可以通过指定 `--format=json` 选项，以 `json` 格式显示。默认情况下，备份目录的内容显示为纯文本格式。

```
gs_probackup show -B backup-path
[--instance=instance_name [-i backup-id]] [--archive] [--format=plain|json]
[--help]
```

- 创建指定实例的备份。

```
gs_probackup backup -B backup-path --instance=instance_name -b backup-mode
[-D pgdata-path] [-C] [-S slot-name] [--temp-slot] [--backup-pg-log] [-j
threads_num] [--progress]
[--no-validate] [--skip-block-validation] [-E external-directories-paths]
[--no-sync] [--note=text]
[--archive-timeout=timeout] [-t rvertimeout]
[logging_options] [retention_options] [compression_options]
[connection_options] [remote_options] [pinning_options]
[--help]
```

- 从备份目录 `_backup-path_` 中的备份副本恢复指定实例。如果指定了恢复目标选项，`gs_probackup` 将查找最近的备份并将其还原到指定的恢复目标。否则，使用最近一次备份。

```
gs_probackup restore -B backup-path --instance=instance_name
```

```
[ -D pgdata-path ] [ -i backup_id ] [ -j threads_num ] [ --progress ] [ --force ]
[ --no-sync ] [ --no-validate ] [ --skip-block-validation ]
[ --external-mapping=OLDDIR=NEWDIR ] [ -T OLDDIR=NEWDIR ] [ --skip-external-dirs ]
[ -I incremental_mode ]
[recovery_options] [remote_options] [logging_options]
[ --help ]
```

- 将指定的增量备份与其父完全备份之间的所有增量备份合并到父完全备份。父完全备份将接收所有合并的数据，而已合并的增量备份将作为冗余被删除。

```
gs_probackup merge -B backup-path --instance=instance_name -i backup_id
[ -j threads_num ] [ --progress ] [ logging_options ]
[ --help ]
```

- 删除指定备份，或删除不满足当前保留策略的备份。

```
gs_probackup delete -B backup-path --instance=instance_name
[ -i backup-id | --delete-expired | --merge-expired | --status=backup_status ]
[ --delete-wal ] [ -j threads_num ] [ --progress ]
[ --retention-redundancy=retention-redundancy ]
[ --retention-window=retention-window ]
[ --wal-depth=wal-depth ] [ --dry-run ]
[ logging_options ]
[ --help ]
```

- 验证恢复数据库所需的所有文件是否存在且未损坏。如果未指定 `_instance_name_`，`gs_probackup` 将验证备份目录中的所有可用备份。如果指定 `_instance_name_` 而不指定任何附加选项，`gs_probackup` 将验证此备份实例的所有可用备份。如果指定了 `_instance_name_` 并且指定 `_backup-id_` 或恢复目标相关选项，`gs_probackup` 将检查是否可以使用这些选项恢复数据库。

```
gs_probackup validate -B backup-path
[ --instance=instance_name ] [ -i backup-id ]
[ -j threads_num ] [ --progress ] [ --skip-block-validation ]
[ --recovery-target-time=time | --recovery-target-xid=xid |
--recovery-target-lsn=lsn | --recovery-target-name=target-name ]
[ --recovery-target-inclusive=boolean ]
[ logging_options ]
[ --help ]
```

4.17.5 参数说明

通用参数

- `command`

`gs_probackup` 除 `version` 和 `help` 以外的子命令：`init`、`add-instance`、`del-instance`、`set-config`、`set-backup`、`show-config`、`show`、`backup`、`restore`、`merge`、`delete`、`validate`。

- `-?`, `--help`

显示 `gs_probackup` 命令行参数的帮助信息，然后退出。

子命令中只能使用 `--help`，不能使用 `-?`。

- `-V`, `--version`

打印 `gs_probackup` 版本，然后退出。

- `-B backup-path`, `--backup-path=backup-path`

备份的路径。

系统环境变量：`$BACKUP_PATH`

- `-D pgdata-path`, `--pgdata=pgdata-path`

数据目录的路径。

系统环境变量：`$PGDATA`

- `--instance=instance_name`

实例名。

- `-i backup-id`, `--backup-id=backup-id`

备份的唯一标识。

- `--format=format`

指定显示备份信息的格式，支持 `plain` 和 `json` 格式。

默认值：`plain`

- `--status=backup_status`

删除指定状态的所有备份，包含以下状态：

OK：备份已完成且有效。

DONE：备份已完成但未经过验证。

RUNNING：备份正在进行中。

MERGING：备份正在合并中。

DELETING：备份正在删除中。

CORRUPT：部分备份文件已损坏。

ERROR：由于意外错误，备份失败。

ORPHAN：由于其父备份之一已损坏或丢失，备份无效。

- `-j threads_num, --threads=threads_num`

设置备份、还原、合并进程的并行线程数。

- `--archive`

显示 WAL 归档信息。

- `--progress`

显示进度。

- `--note=text`

给备份添加 note。

备份相关参数

- `-b backup-mode, --backup-mode=backup-mode`

指定备份模式，支持 FULL 和 PTRACK。

FULL：创建全量备份，全量备份包含所有数据文件。

PTRACK：创建 PTRACK 增量备份。

- `-C, --smooth-checkpoint`

将检查点在一段时间内展开。默认情况下，`gs_probackup` 会尝试尽快完成检查点。

- `-S slot-name, --slot=slot-name`

指定 WAL 流处理的复制 slot。

- `--temp-slot`

在备份的实例中为 WAL 流处理创建一个临时物理复制 slot，它确保在备份过程中，所有所需的 WAL 段仍然是可用的。

默认的 slot 名为 `pg_probackup_slot`，可通过选项 `--slot/-S` 更改。

- `--backup-pg-log`

将日志目录包含到备份中。此目录通常包含日志消息。默认情况下包含日志目录，但不包含日志文件。如果修改了默认的日志路径，需要备份日志文件时可使用 `-E` 参数进行备份，使用方法见下文。

- `-E external-directories-paths, --external-dirs=external-directories-paths`

将指定的目录包含到备份中。此选项对于备份位于数据目录外部的脚本、sql 转储和配置文件很有用。如果要备份多个外部目录，请在 Unix 上用冒号分隔它们的路径。

例如：`-E /tmp/dir1:/tmp/dir2`

- `--skip-block-validation`

关闭块级校验，加快备份速度。

- `--no-validate`

在完成备份后跳过自动验证。

- `--no-sync`

不将备份文件同步到磁盘。

- `--archive-timeout=timeout`

以秒为单位设置流式处理的超时时间。

默认值：300

- `-t rvertimeout`

以秒为单位的连接的超时时间。

默认值：120

恢复相关参数

- `-I, --incremental-mode=none|checksum|lsn`

若 PGDATA 中可用的有效页没有修改，则重新使用它们。

默认值：none

- `--external-mapping=OLDDIR=NEWDIR`

在恢复时，将包含在备份中的外部目录从 `_OLDDIR_` 重新定位到 `_NEWDIR_` 目录。`_OLDDIR_` 和 `_NEWDIR_` 都必须是绝对路径。如果路径中包含“=”，则使用反斜杠转义。此选项可为多个目录多次指定。

- `-T OLDDIR=NEWDIR, --tablespace-mapping=OLDDIR=NEWDIR`

在恢复时，将表空间从 `_OLDDIR_` 重新定位到 `_NEWDIR_` 目录。`_OLDDIR_` 和 `_NEWDIR_` 必须都是绝对路径。如果路径中包含“=”，则使用反斜杠转义。多个表空间可以多次指定此选项。此选项必须和 `--external-mapping` 一起使用。

- `--skip-external-dirs`

跳过备份中包含的使用 `--external-dirs` 选项指定的外部目录。这些目录的内容将不会被恢复。

- `--skip-block-validation`

跳过块级校验，以加快验证速度。在恢复之前的自动验证期间，将仅做文件级别的校验。

- `--no-validate`

跳过备份验证。

- `--force`

允许忽略备份的无效状态。如果出于某种原因需要从损坏的或无效的备份中恢复数据，可以使用此标志。请谨慎使用。

恢复目标相关参数(recovery_options)

 说明

- 当前不支持配置连续的 WAL 归档的 PITR，因而使用这些参数会有一定限制，具体如下描述。如果需要使用持续归档的 WAL 日志进行 PITR 恢复，请按照下面描

述的步骤：

1. 将物理备份的文件替换目标数据库目录。
2. 删除数据库目录下 `pg_xlog/` 中的所有文件。
3. 将归档的 WAL 日志文件复制到 `pg_xlog` 文件中（此步骤可以省略，通过配置 `recovery.conf` 恢复命令文件中的 `restore_command` 项替代）。
4. 在数据库目录下创建恢复命令文件 `recovery.conf`，指定数据库恢复的程度。
5. 启动数据库。
6. 连接数据库，查看是否恢复到希望预期的状态。若已经恢复到预期状态，通过 `pg_xlog_replay_resume()` 指令使主节点对外提供服务。

- `--recovery-target-lsn=lsn`

指定要恢复到的 lsn，当前只能指定备份的 stop lsn。

- `--recovery-target-name=target-name`

指定要将数据恢复到的已命名的保存点，保存点可以通过查看备份中 `recovery-name` 字段得到。

- `--recovery-target-time=time`

指定要恢复到的时间，当前只能指定备份中的 `recovery-time`。

- `--recovery-target-xid=xid`

指定要恢复到的事务 ID，当前只能指定备份中的 `recovery-xid`。

- `--recovery-target-inclusive=boolean`

当该参数指定为 `true` 时，恢复目标将包括指定的内容。

当该参数指定为 `false` 时，恢复目标将不包括指定的内容。

该参数必须和 `--recovery-target-name`、`--recovery-target-time`、`--recovery-target-lsn` 或 `--recovery-target-xid` 一起使用。

留存相关参数(`retention_options`)

 说明

➤ 可以和 backup 和 delete 命令一起使用这些参数。

- --retention-redundancy=retention-redundancy

指定在数据目录中留存的完整备份数。必须为正整数。0 表示禁用此设置。

默认值：0

- --retention-window=retention-window

指定留存的天数。必须为正整数。0 表示禁用此设置。

默认值：0

- --wal-depth=wal-depth

每个时间轴上必须留存的执行 PITR 能力的最新有效备份数。必须为正整数。0 表示禁用此设置。

默认值：0

- --delete-wal

从任何现有的备份中删除不需要的 WAL 文件。

- --delete-expired

删除不符合 pg_probackup.conf 配置文件中定义的留存策略的备份。

- --merge-expired

将满足留存策略要求的最旧的增量备份与其已过期的父备份合并。

- --dry-run

显示所有可用备份的当前状态，不删除或合并过期备份。

固定备份相关参数(pinning_options)

 说明

➤ 如果要将某些备份从已建立的留存策略中排除，可以和 backup 和 set-backup 命令一起使用这些参数。

- --ttl=interval

指定从恢复时间开始计算，备份要固定的时间量。必须为正整数。0 表示取消备份固定。

支持的单位：ms、s、min、h、d（默认为 s）。

例如：--ttl=30d。

- --expire-time=time

指定备份固定失效的时间戳。必须是 ISO-8601 标准的时间戳。

例如：--expire-time='2020-01-01 00:00:00+03'

日志相关参数(logging_options)

日志级别：verbose、log、info、warning、error 和 off。

- --log-level-console=log-level-console

设置要发送到控制台的日志级别。每个级别都包含其后的所有级别。级别越高，发送的消息越少。指定 off 级别表示禁用控制台日志记录。

默认值：info

- --log-level-file=log-level-file

设置要发送到日志文件的日志级别。每个级别都包含其后的所有级别。级别越高，发送的消息越少。指定 off 级别表示禁用日志文件记录。

默认值：off

- --log-filename=log-filename

指定要创建的日志文件的文件名。文件名可以使用 strftime 模式，因此可以使用 %-escapes 指定随时间变化的文件名。

例如，如果指定了“pg_probackup-%u.log”模式，则 pg_probackup 为每周的每一天生成单独的日志文件，其中 %u 替换为相应的十进制数字，即 pg_probackup-1.log 表示星期一；pg_probackup-2.log 表示星期二，以此类推。

如果指定了 --log-level-file 参数启用日志文件记录，则该参数有效。

默认值：“pg_probackup.log”

- --error-log-filename=error-log-filename

指定仅用于 error 日志的日志文件名。指定方式与 --log-filename 参数相同。

此参数用于故障排除和监视。

- `--log-directory=log-directory`

指定创建日志文件的目录。必须是绝对路径。此目录会在写入第一条日志时创建。

默认值: `$BACKUP_PATH/log`

- `--log-rotation-size=log-rotation-size`

指定单个日志文件的最大大小。如果达到此值,则启动 `gs_probackup` 命令后,日志文件将循环,但 `help` 和 `version` 命令除外。0 表示禁用基于文件大小的循环。

支持的单位: KB、MB、GB、TB (默认为 KB)。

默认值: 0

- `--log-rotation-age=log-rotation-age`

单个日志文件的最大生命周期。如果达到此值,则启动 `gs_probackup` 命令后,日志文件将循环,但 `help` 和 `version` 命令除外。`$BACKUP_PATH/log/log_rotation` 目录下保存最后一次创建日志文件的时间。0 表示禁用基于时间的循环。

支持的单位: ms、s、min、h、d (默认为 min)。

默认值: 0

连接相关参数(connection_options)



说明

➤ 可以和 `backup` 命令一起使用这些参数。

- `-d dbname, --pgdatabase=dbname`

指定要连接的数据库名称。该连接仅用于管理备份进程,因此您可以连接到任何现有的数据库。如果命令行、`PGDATABASE` 环境变量或 `pg_probackup.conf` 配置文件中没有指定此参数,则 `gs_probackup` 会尝试从 `PGUSER` 环境变量中获取该值。如果未设置 `PGUSER` 变量,则从当前用户名获取。

系统环境变量: `$PGDATABASE`

- `-h hostname, --pghost=hostname`

指定运行服务器的系统的主机名。如果该值以斜杠开头,则被用作到 Unix 域套接字的

路径。

系统环境变量：\$PGHOST

默认值：local socket

- -p port, --pgport=_p_ort

指定服务器正在侦听连接的 TCP 端口或本地 Unix 域套接字文件扩展名。

系统环境变量：\$PGPORT

默认值：15400

- -U username, --pguser=username

指定所连接主机的用户名。

系统环境变量：\$PGUSER

- -w, --no-password

不出现输入密码提示。如果主机要求密码认证并且密码没有通过其它形式给出，则连接尝试将会失败。该选项在批量工作和不存在用户输入密码的脚本中很有帮助。

- -W password, --password=password

指定用户连接的密码。如果主机的认证策略是 trust，则不会对系统管理员进行密码验证，即无需输入-W 选项；如果没有-W 选项，并且不是系统管理员，则会提示用户输入密码。

压缩相关参数(compression_options)



说明

- 可以和 backup 命令一起使用这些参数。

- --compress-algorithm=compress-algorithm

指定用于压缩数据文件的算法。

取值包括 zlib、pglz 和 none。如果设置为 zlib 或 pglz，此选项将启用压缩。默认情况下，压缩功能处于关闭状态。

默认值：none

- --compress-level=compress-level

指定压缩级别。取值范围：0~9

0 表示无压缩。

1 表示压缩比最小，处理速度最快。

9 表示压缩比最大，处理速度最慢。

可与--compress-algorithm 选项一起使用。

默认值：1

- --compress

以--compress-algorithm=zlib 和--compress-level=1 进行压缩。

远程模式相关参数(remote_options)

 说明

➤ 通过 SSH 远程运行 gs_probbackup 操作的相关参数。可以和 add-instance、set-config、backup、restore 命令一起使用这些参数。

- --remote-protocol=protocol

指定用于远程操作的协议。目前只支持 SSH 协议。取值包括：

ssh：通过 SSH 启用远程备份模式。这是默认值。

none：显式禁用远程模式。

如果指定了--remote-host 参数，可以省略此参数。

- --remote-host=destination

指定要连接的远程主机的 IP 地址或主机名。

- --remote-port=port

指定要连接的远程主机的端口号。

默认值：22

- --remote-user=username

指定 SSH 连接的远程主机用户。如果省略此参数，则使用当前发起 SSH 连接的用户。

默认值：当前用户

- `--remote-path=path`

指定 `gs_probackup` 在远程系统的安装目录。

默认值：当前路径

- `--remote-libpath=libpath`

指定 `gs_probackup` 在远程系统安装的 `lib` 库目录。

- `--ssh-options=ssh_options`

指定 SSH 命令行参数的字符串。

例如：`--ssh-options='-c cipher_spec -F configfile'`



说明：

如果因为网络临时故障等原因导致 `server` 端无应答, `gs_probackup` 将在等待 `archive-timeout` (默认 300 秒) 后退出。

如果备机 `lsn` 与主机有差别时, 数据库会不停地刷以下 `log` 信息, 此时应重新 `build` 备机。

```
LOG: walsender thread shut down
LOG: walsender thread started
LOG: received wal replication command: IDENTIFY_VERSION
LOG: received wal replication command: IDENTIFY_MODE
LOG: received wal replication command: IDENTIFY_SYSTEM
LOG: received wal replication command: IDENTIFY_CONSISTENCE 0/D0002D8
LOG: remote request lsn/crc: [xxxxxx] local max lsn/crc: [xxxxxx]
```

4.17.6 备份流程

- (1) 初始化备份目录。在指定的目录下创建 `backups/`和 `wal/`子目录, 分别用于存放备份文件和 WAL 文件。

```
gs_probackup init -B backup_dir
```

- (2) 添加一个新的备份实例。 `gs_probackup` 可以在同一个备份目录下存放多个数据库实例的备份。

```
gs_probackup add-instance -B backup_dir -D data_dir --instance instance_name
```

- (3) 创建指定实例的备份。在进行增量备份之前, 必须至少创建一次全量备份。

```
gs_probackup backup -B backup_dir --instance instance_name -b backup_mode
```

(4) 从指定实例的备份中恢复数据。

```
gs_probackup restore -B backup_dir --instance instance_name -D pgdata-path -i backup_id
```

4.17.7 故障处理

问题描述：ERROR: query failed: ERROR: canceling statement due to conflict with recover (错误：查询失败：由于与恢复操作冲突，正在取消语句命令)

原因：在备机上执行的操作正在访问存储行，主机上更改或者删除了对应的行，并将 xlog 在备机上重放，迫使备机上操作取消。

解决方法：

(1) 适当增加如下配置参数的值

```
max_standby_archive_delay
```

```
max_standby_streaming_delay
```

(2) 增加如下配置

```
hot_standby_feedback = on
```

4.18 gstrace

4.18.1 功能介绍

gstrace 是 GBase 8s 提供的用来跟踪内核代码执行路径、记录内核数据结构、分析代码性能的工具。Trace 的有限点位和数据在版本中被固化，无法动态添加和删除。

警告

- 对内核 dump 指定内存变量的数据用于诊断分析，不存在直接指定任意寄存器或者内存地址的行为。读取的内存地址均是在开发阶段硬编码，没有任意地读取或者修改寄存器或内存的操作。
- Trace 点可能涉及敏感数据，收集 trace 信息前需要同用户协商，授权和许可后方可收集。

- GBase 8s 不会在敏感信息上打点，不会 TRACE 和用户相关的数据。
- Trace 仅用于诊断目的，开启 trace 将对性能产生一定影响，影响的大小视负载的高低，trace 的模块而不同。
- Trace 工具的权限为 0700，仅限于数据库用户读、写和执行。

说明

- 如果进程异常终止，/dev/shm/ 目录下将会有 gstrace_trace_cfg_*残留，可以手动清除。

4.18.2 语法

```
gstrace [start|stop|config|dump|detailcodepath|analyze] [-p <port>][-s <BUFFER_SIZE>] [-f <DUMP_FILE>] [-o <OUTPUT_FILE>] [-t <STEP_SIZE>]
```

4.18.3 参数说明

表 4-15 gs_trace 参数说明

参数	说明
start	开始记录 trace。
stop	停止 trace，释放存储信息的共享内存 trace buffer。注意：trace buffer 中已捕获 trace 信息会丢失。
config	显示 trace 的配置信息。
dump	将共享内存中的 trace 信息写入指定文件。若没有启动 trace 将报错。
detail	将 dump 出来的二进制文件解析成文本文件，显示 trace 点的线程、时间信息。

参数	说明
codepath	提取 dump 文件中的函数调用信息，按照调用栈的方式显示。
analyze	统计各个函数的执行次数、总耗时、平均耗时、最长耗时、最短耗时。
-p PORT	指定启动 trace 功能的实例侦听的端口号。
-f DUMP_FILE	指定 dump 导出的 trace 文件。
-o OUTPUT_FILE	指定写入的文件。
-t STEP_SIZE	指定分片分析的时间跨度（秒），可选。将生成单独的 {OUTPUT_FILE}.step 文件。
-m MASK	<p>指定哪些模块、函数需要追踪，如果参数为空，则追踪添加 trace 接口的所有模块的所有函数。</p> <p>格式：[comp...][ALL].[func...][ALL]</p> <p>描述：</p> <ul style="list-style-type: none"> ● comp 是模块列表，使用‘;’分隔，也可以使用序号。 ● func 是 function 列表，使用‘;’分隔。例如：sql_execute_ddl, sql_execute_dcl。也可以使用序号指代模块中第几个 function，例如：2。 ● 设置成 ALL 表示所有模块或所有 function 被 trace。 <p>示例：</p> <p>executer,gbase.ALL : 所有定义定义在 executer 和 GBase 8s 模块下的 function 被 trace。</p> <p>executer,gbase.sql_execute_ddl : 定义在 executer 下的 Function</p>

参数	说明
	<p>sql_execute_ddl 被 trace. GBase 8s 模块被忽略, 因为下面没有该 function 。</p> <p>executer,gbase.1 : 定义在模块 executer 和 GBase 8s 下的第一个 function 被 trace。</p> <p>ALL.1,2 : 定义在所有模块下的第一, 二个 function 被 trace。</p>
-s BUFFER_SIZE	<p>指定用于 trace 功能的共享内存大小, 默认为 1G。如果指定的 BUFFER_SIZE 小于最小值 2048, 系统将自动调整为最小值。如果指定的 BUFFER_SIZE 不是 2^N (二的 N 次方), 则向下对齐 2^N; 例如: 指定 BUFFER_SIZE=3072, 由于 2¹¹<3072<2¹², 系统将调整为 2048。</p>

4. 18. 4示例

- 启动 trace

```
gstrace start -p 8000
```

- 停止 trace

```
gstrace stop -p 8000
```

- 查看 trace 配置

```
gstrace config -p 8000
```

- dump trace

```
gstrace dump -p 8000 -o /data/8000.dump
```

- 解析 dump detail 信息

```
gstrace detail -f /data/8000.dump -o /data/8000.detail
```

- 解析 dump codepath

```
gstrace codepath -f /data/8000.dump -o /data/8000.codepath
```

- 分析全局性能

```
gstrace analyze -f /data/8000.dump -o /data/8000.perf
```

- 分析分片性能

```
gstrace analyze -f /data/8000.dump -o /data/8000.perf -t 1
```

4.19 kdb5_util

4.19.1 功能介绍

kdb5_util 是为 GBase 8s 认证服务 Kerberos 提供数据库管理操作的工具，可以新建、销毁、导入和导出数据库。

kdb5_util 允许管理员对数据库执行维护过程。数据库可以创建、销毁、转储到 ASCII 文件或从 ASCII 文件加载。kdb5_util 可以创建 Kerberos 主密钥存储文件或执行主密钥的实时滚动。

在运行 kdb5_util 时，它尝试获取主密钥并打开数据库。但无论是否成功打开数据库，执行都会继续，因为数据库可能还不存在，或者存储文件可能已损坏。

4.19.2 语法

```
kdb5_util [-r realm] [-d dbname] [-k mkeytype] [-kv mkeyVNO] [-M mkeyname] [-m] [-sf stashfilename] [-P password] [-x db_args] command [command_options]
```

4.19.3 参数说明

kdb5_util 工具为开源第三方提供，具体参数说明请参考 Kerberos 官方文档：https://web.mit.edu/kerberos/krb5-1.17/doc/admin/admin_commands/kdb5_util.html

4.20 kadmin.local

4.20.1 功能介绍

kadmin.local 为 GBase 8s 认证服务 Kerberos 提供命令行接口，直接接入 kdc 服务的数据库，进行增加、删除、修改 Kerberos 用户等操作。

4.20.2 语法

```
kadmin.local [-r realm] [-p principal] [-q query] [-d dbname] [-e enc:salt ...]  
[-m] [-x db_args] [command args...]
```

4.20.3 参数说明

kadmin.local 工具为开源第三方提供，具体参数说明请参考 Kerberos 官方文档：
https://web.mit.edu/kerberos/krb5-1.17/doc/admin/admin_commands/kadmin_local.html?highlight=kadmin%20local

4.21 kinit

4.21.1 功能介绍

kinit 为 GBase 8s 认证服务 Kerberos 提供特定用户获取和缓存 TGT 能力。

4.21.2 语法

```
kinit [-V] [-l lifetime] [-s start_time] [-r renewable_life] [-p | -P] [-f | -F]  
[-a] [-A] [-C] [-E] [-v] [-R] [-k [-t keytab_file]] [-c cache_name] [-n] [-S  
service_name] [-I input_ccache] [-T armor_ccache] [-X attribute[=value]]  
[principal]
```

4.21.3 参数说明

kinit 工具为开源第三方提供，具体参数说明请参考 Kerberos 官方文档：
https://web.mit.edu/kerberos/krb5-1.17/doc/user/user_commands/kinit.html?highlight=kinit

4.22 klist

4.22.1 功能介绍

klist 为 GBase 8s 认证服务 Kerberos 提供列出凭证缓存中的用户和票据能力。

4.22.2 语法

```
klist [-e] [[-c] [-l] [-A] [-f] [-s] [-a [-n]]] [-C] [-k [-t] [-K]] [-V]  
[cache_name|keytab_name]
```


4.22.3 参数说明

klist 工具为开源第三方提供，具体参数说明请参考 Kerberos 官方文档：
https://web.mit.edu/kerberos/krb5-1.17/doc/user/user_commands/klist.html?highlight=klist

4.23 krb5kdc

4.23.1 功能介绍

krb5kdc 为 GBase 8s 认证服务 Kerberos 提供列认证和密钥管理服务。

4.23.2 语法

```
krb5kdc [-x db_args] [-d dbname] [-k keytype] [-M mkeyname] [-p portnum] [-m] [-r realm] [-n] [-w numworkers] [-P pid_file] [-T time_offset]
```

4.23.3 参数说明

krb5kdc 工具为开源第三方提供，具体参数说明请参考 Kerberos 官方文档：
https://web.mit.edu/kerberos/krb5-1.17/doc/admin/admin_commands/krb5kdc.html?highlight=krb5kdc

4.24 kdestroy

4.24.1 功能介绍

kdestroy 为 GBase 8s 认证服务 Kerberos 提供销毁用户的授权票据操作。

4.24.2 语法

```
kdestroy [-A] [-q] [-c cache_name]
```

4.24.3 参数说明

kdestroy 工具为开源第三方提供，具体参数说明请参考 Kerberos 官方文档：
https://web.mit.edu/kerberos/krb5-1.17/doc/user/user_commands/kdestroy.html?highlight=kdestroy

4.25 pg_config

4.25.1 功能介绍

通过 pg_config 工具打印的当前所安装数据库的配置参数信息，可用于找到与数据库相关的软件包所对应的头文件和库。

4.25.2 语法

```
pg_config [option...]
```

4.25.3 参数说明

pg_config 工具支持下面一个或多个选项，如果给出多于一个选项，那么信息以该顺序打印，每行一条。如果没有给出选项，则打印所有可获得的信息，连着标签一起打印。

- --bindir

打印用户可执行文件的路径。比如，可以用这个找 psql 程序。通常这也是 pg_config 程序存在的路径。

- --docdir

打印文档文件的位置。

- --htmldir

输出 HTML 文档文件的位置。

- --includedir

打印客户端接口 C 头文件的路径。

- --pkgincludedir

打印其它 C 头文件的位置。

- --includedir-server

打印那些做服务器编程时使用的 C 头文件的位置。

- --libdir

打印目标代码库的路径。

- `--pkglibdir`

打印动态加载的模块的位置，或者服务器在哪里寻找他们。其它体系相关的数据文件可能也放在这个目录里。

- `--localedir`

打印区域支持文件的位置。如果在编译数据库的时候没有配置区域支持，则其值为空字符串。

- `--mandir`

打印手册页的位置。

- `--sharedir`

打印体系无关的支持文件的位置。

- `--sysconfdir`

打印系统范围的配置文件位置。

- `--pgxs`

打印扩展 `makefiles` 文件的位置。

- `--configure`

打印配置和编译数据库时给 `configure` 脚本传递的选项，用于复制相同配置，或者找出二进制包是用哪些选项编译的。不过请注意二进制包通常包含供应商提供的补丁。

- `--cc`

打印用于编译数据库的 `CC` 变量的数值。这个选项显示使用的 C 编译器。

- `--cppflags`

打印用于编译数据库的 `CPPFLAGS` 变量的数值。这个选项显示在预处理的时候所需要的 C 编译器开关（通常是 `-I` 开关）。

- `--cflags`

打印编译数据库时使用的 `CFLAGS` 变量的数值。这个选项显示 C 编译器开关。

- `--cflags_sl`

打印编译数据库时使用的 CFLAGS_SL 变量的数值。 这个选项显示了编译共享库的额外的 C 编译器开关。

- --ldflags

打印编译数据库时使用的 LDFLAGS 变量的数值。 它显示了链接器的开关。

- --ldflags_ex

输出用于创建数据库的 LDFLAGS_EX 变量的值。 这显示用于构建可执行文件的链接器开关。

- --ldflags_sl

打印编译数据库时使用的 LDFLAGS_SL 变量的数值。 它显示了用于编译共享库的链接器的开关。

- --libs

打印编译数据库时使用的 LIBS 变量的数值。 其内容通常包含连接到数据库的外部库的-l 开关。

- --version

打印数据库的版本。

- -?, --help

显示关于 pg_config 命令行参数的帮助文件，并且退出。

4.26 pg_controldata

4.26.1 功能介绍

pg_controldata 工具用来显示数据库目录下 control 文件的信息。

4.26.2 前提条件

确保数据目录正确，以及相应数据目录下的 pg_control 文件存在。

4.26.3 语法

- 查看 control 文件信息

```
pg_controldata DATADIR
```

- 显示版本号信息

```
pg_controldata -V | --version
```

- 显示帮助信息

```
pg_controldata -? | --help
```

4.26.4 参数说明

- DATADIR

查看 control 文件信息时所用的参数，DATADIR 为实例对应的数据库目录。

- -V, --version

显示版本信息。

- -?, --help

打印帮助信息。

4.27 pg_recvlogical

4.27.1 功能介绍

pg_recvlogical 工具通过连接指定的节点，创建或删除逻辑复制槽，以及持续、实时的从该节点获取逻辑解码中间结果，输出到文件或标准输出。

4.27.2 语法

```
pg_recvlogical [OPTION]...
```

其中，逻辑复制槽行为参数为必选参数，必须指定一个。其余参数为可选参数，如不指定，将会读取默认值。

4.27.3 参数说明

逻辑复制槽行为参数

必选参数

- --create

创建一个新的逻辑复制槽。内部调用了 `pg_create_logical_replication_slot` 函数，调用该函数的用户需要具有 `SYSADMIN` 权限或具有 `REPLICATION` 权限或继承了内置角色 `gs_role_replication` 的权限，且此函数目前只支持在主机调用。参考《开发者指南》`pg_create_logical_replication_slot` 函数的说明。

- `--start`

启动该逻辑复制槽的流复制。

- `--drop`

删除一个逻辑复制槽。

连接参数

- `-d, --dbname=DBNAME`

连接的目标数据库名。

- `-h, --host=HOSTNAME`

连接的目标主机名或 `socket`。

- `-p, --port=PORT`

连接的目标数据库节点的 HA 端口号。

- `-U, --username=NAME`

连接目标库所使用的用户名。

- `-w, --no-password`

不使用密码进行连接。

- `-W, --password`

使用指定密码进行连接。

复制参数

- `-F --fsync-interval=INTERVAL`

同步到目标输出文件的间隔时间。

单位为秒，默认值 10。

- `-o, --option=NAME[=VALUE]`

指定向逻辑复制槽的输出插件添加参数。

参数的取值为：`include-xids`、`skip-empty-xacts`、`include-timestamp`、`only-local`、`force-binary`、`white-table-list`、`standby-connection`、`parallel-decode-num`、`decode-style`、`sending-batch`。

详细请参见《GBase 8s V8.8.5 5.0.0_开发者指南》中“SQL 参考 > 函数和操作符 > 系统管理函数 > 逻辑复制函数 > `pg_logical_slot_peek_changes` > options”以及“应用程序开发教程 > 基于 JDBC 开发 > 示例：逻辑示例”章节。

- `-P, --plugin=PLUGIN`

指定使用的逻辑复制槽输出插件。

默认值为 `mppdb_decoding`。

- `-s, --status-interval=INTERVAL`

发送的心跳包间隔时间。

单位为秒，默认值 10。

- `-S, --slot=SLOT`

指定逻辑复制槽的名称。

- `-I, --startpos=PTR`

当使用已有逻辑复制槽时，指定初始复制的 LSN 位置。

其它参数

- `-f, --file=FILE`

设置输出到指定文件。

直接使用“-”表示输出到 `stdout`。

- `-n, --no-loop`

指定此参数，连接失败后不重试。

- `-v, --verbose`

输出详细信息。

- -V, --version

输出版本信息，随后立即退出。

- -?, --help

输出帮助信息，随后立即退出。

- -r, --raw

在并行解码场景下，设置此参数表示不会将二进制格式或是批量发送的结果转化为文本格式，该参数仅供测试人员测量并行解码性能时使用。设置该参数后，二进制格式或批量发送的解码结果将为不可读的格式。

4.27.4 示例

创建名为 test_slot 的逻辑复制槽。

```
pg_recvlogical -d postgres -S test_slot -p 26000 --create
```

开启流式解码，结果输出到 stdout。

```
pg_recvlogical -d postgres -S test_slot -p 26000 --start -v -f -
pg_recvlogical: starting log streaming at 0/0 (slot test_slot)
pg_recvlogical: initiated streaming
pg_recvlogical: confirming write up to 0/0, flush to 0/0 (slot test_slot)
pg_recvlogical: confirming write up to 0/2A342E8, flush to 0/2A342E8 (slot
test_slot)
pg_recvlogical: confirming write up to 0/2A34320, flush to 0/2A34320 (slot
test_slot)
pg_recvlogical: confirming write up to 0/2A34320, flush to 0/2A34320 (slot
test_slot)
BEGIN 16039
table public.t: INSERT: a[integer]:4 b[integer]:4
CgbaseIT 16039
pg_recvlogical: confirming write up to 0/2A34450, flush to 0/2A34450 (slot
test_slot)
pg_recvlogical: confirming write up to 0/2A34450, flush to 0/2A34450 (slot
test_slot)
```

删除逻辑复制槽。

```
pg_recvlogical -d postgres -S test_slot -p 26000 --drop
```


4.28 pg_resetxlog

4.28.1 功能介绍

pg_resetxlog 是一个重新设置数据库事务文件的工具。

注意

- 通过 pg_resetxlog 重新设置之前，需要先停止数据库。

4.28.2 前提条件

保证数据库目录存在且路径正确。

4.28.3 语法

```
pg_resetxlog [OPTION]... DATADIR
```

4.28.4 参数说明

OPTION 取值如下所示：

- DATADIR

需要修改的数据库目录，确保路径正确。

- -e XIDPOCH

设置下一个事务 id。

- -f

强制更新。

- -l xlogfile

为新的事务日志指定最小的 WAL 起始位置。

- -m XID

设置下一个 multitransaction ID。

- -n

不更新，仅显示配置文件的值。

- -o OID

设置下一个 OID。

- -O OFFSET

设置下一个 multitransaction 的偏移量。

- -V, --version

显示版本信息。

- -x XID

设置下一个事务 ID。

- -?, --help

打印帮助信息。

4.29 pg_archivecleanup

4.29.1 功能介绍

pg_archivecleanup 是一个用于清理旧的归档日志的工具。

4.29.2 语法

```
pg_archivecleanup [OPTION]... ARCHIVELOCATION OLDESTKEPTWALFILE
```

4.29.3 参数说明

- -d

在 stderr 上打印调试日志。

- -n

打印将要删除的文件名。

- -V, --version

打印 pg_archivecleanup 的版本并退出。

- -x extension

当该程序用作单独的工具时，提供一个扩展选项，只选择对应扩展名格式的文件。

- -?,--help

显示关于 pg_archivecleanup 命令行参数的帮助信息。

4.29.4 示例

删除比 000000010000000000000010 更早的日志

```
pg_archivecleanup [OPTION]... ARCHIVELOCATION OLDESTKEPTWALFILE
```

4.29.5 扩展应用

备机恢复时，在 recovery.conf 里配置 archive_cleanup_command 参数。

```
archive_cleanup_command = 'pg_archivecleanup /mnt/server/archiverdir %r'
```

恢复完成后从归档目录中清除不再需要的文件。

4.30 pssh

4.30.1 基本功能

用于执行数据库节点间的访问，发送命令给远程节点。

4.30.2 语法

```
python3 pssh [OPTIONS]
```

4.30.3 参数说明

- --help

显示帮助信息。

- -H HOSTNAME

主机列表。

- -h HOSTFILE

写入主机列表的文件全路径。

- -t TIMEOUT
连接超时时间，单位秒，默认 300s。
- -p PARALLEL
最大并行个数，默认 32。
- -o OUTDIR
执行结果输出全文件路径。
- -e ERRDIR
错误信息输出全文件路径。
- -P
线上远端主机执行的命令。
- -s
显示执行结果。
- -x EXTRA
额外的命令行参数。
- -i
显示所有节点的输出和错误。
- -O OPT
补充的线下模式 ssh 的参数。

4.31 pscp

4.31.1 基本功能

用于执行数据库节点间的文件传输。

4.31.2 语法

```
python3 pscp [OPTIONS]
```

4. 31. 3参数说明

- **--help**
显示帮助信息。
- **-H HOSTNAME**
主机列表。
- **-h HOSTFILE**
写入主机列表的文件全路径。
- **-t TIMEOUT**
连接超时时间，单位秒，默认 300s。
- **-p PARALLEL**
最大并行个数，默认 32。
- **-o OUTDIR**
执行结果输出全文件路径。
- **-e ERRDIR**
错误信息输出全文件路径。
- **-r**
递归复制目录。
- **-v**
打开诊断消息。
- **-s**
显示执行结果。
- **-x EXTRA**
额外的命令行参数。

- -i
显示所有节点的输出和错误。
- -O OPT
补充的线下模式 scp 的参数。

4.32 transfer.py

4.32.1 功能介绍

在非单节点模式下，可以通过 transfer.py 工具将全文检索所使用的词典文件和 PostGIS 插件所需的文件发送到其他节点。

4.32.2 前提条件

执行该工具前，需确保在 \$GAUSSHOME 目录下存在以下四个文件，否则会报错，错误码为 52200。

- script/gspylib/cgbaseon/DbClusterInfo.py。
- script/gspylib/cgbaseon/Cgbaseon.py。
- script/gspylib/cgbaseon/GaussLog.py。
- script/gspylib/threads/SshTool.py。

\$GPHOME 环境变量配置正确。

4.32.3 语法

- 将 sourcefile 发送到所有节点的 destinationpath 路径。

```
transfer.py 1 sourcefile destinationpath
```

- 将 sourcefile 发送到 pgxc_node_name 的 standby 节点的相同路径下。

```
transfer.py 2 sourcefile pgxc_node_name
```

- 显示帮助信息。

```
transfer.py -? | --help
```

4.32.4 参数说明

OPTION 取值如下所示：

- (1) 将当前文件发送到所有节点的目标文件路径下。
- (2) 将当前文件发送到目标节点的 standby 节点的相同路径下。

- **sourcefile**

被发送的源文件路径。

- **destinationpath**

目标文件路径。

- **pgxc_node_name**

目标节点名。

5 集群管理 (Cluster Manager)

5.1 特性介绍

5.1.1 cm_agent

cm_agent 是部署在数据库每个主机上，用来启停和监控各个数据库实例进程的数据库管理组件。

主要功能有：

- 数据库实例启动和停止时负责拉起和停止本主机上部署的实例进程。
- 监控本主机上运行的实例状态并将状态上报发送给 CM Server。
- 执行 CM Server 仲裁下发的命令。

命令说明：

- 公共选项：

- -V, -version

打印 cm_agent 版本信息，然后退出。

- -?, -h, -help

显示关于 cm_agent 命令行参数的帮助信息，然后退出。

日志信息记录的位置选项：

0

记录在设定的日志文件中。

1

记录在 syslog 文件中。

2

记录在设定的日志文件中。

3

空文件，即不记录日志信息。

- 启动模式选项：

- normal

正常模式启动。

- abnormal

非正常模式启动。

5.1.2 cm_server

- cm_server 是用来进行数据库实例管理和实例仲裁的组件。主要功能有：
- 接收各个节点上 cm_agent 发送的数据库各实例状态。
- 提供数据库实例整体状态的查询功能。
- 监控实例的状态变化并进行仲裁命令的下发。

命令说明：

- 公共选项：

- -V, -version

打印 cm_server 版本信息，然后退出。

- -?, -h, -help

显示关于 cm_server 命令行参数的帮助信息，然后退出。

日志信息记录的位置选项：

0

记录在设定的日志文件中。

1

记录在 syslog 文件中。

2

记录在设定的日志文件中。

3

空文件，即不记录日志信息。

5.1.3 自定义资源

当前 CM 支持对无状态资源（即各资源实例角色平等，不区分主备，或资源自身就能够自行进行主备仲裁）进行监控，主要功能包括：

资源配置

资源配置文件 `cm_resource.json`，文件包含所有自定义资源的相关属性，可以通过 `cm_ctl res` 命令对配置文件修改。不支持动态生效，修改配置文件后需要重启 `cm` 才能生效。

客户端

CM 提供客户端动态库给资源进行集成，提供集群状态查询、状态变更通知、集群锁能力。

自动启停资源

资源需要提供脚本，脚本包含启停、检测等能力，脚本路径需要配置在资源配置文件中。

手动启停资源

可以通过 `cm_ctl start/stop -n -I` 命令实现资源实例的启停操作，详细参见 工具介绍。

自定义资源状态

自定义资源有四种状态：`online`，`offline`，`deleted`，`unknown`。可以通过 `cm_ctl` 查询。

配置方法

安装好自定义资源后，若要使用自定义资源监控功能，需要配置两个文件：

(1) 资源脚本。主要用于指定资源的启停、状态检查等指令，一个样例如下：

```
#!/bin/bash
#set -ex #取消该行注释可帮助调试脚本
#资源名称
resName=sharding
#资源 binpath
shardingPath=/home/test/home/apache-shardingsphere-5.1.1-shardingsphere-proxy-bin/bin
#用于过滤资源实例的命令关键词
cmdKey=org.apache.shardingsphere.proxy.Bootstrap
#用于保存首次检测到资源僵死时间的文件
phony_dead_time_file=.sharding_phony_dead_time
#最长僵死时间，单位为 s
PHONY_MAX_TIME=20
```

```
function exec_start
{
    #资源启动命令
    sh ${shardingPath}/start.sh; exit $?
}

function exec_stop
{
    #资源停止命令
    sh ${shardingPath}/stop.sh; exit $?
}

function exec_check
{
    #查询资源实例 pid
    pid=`ps x | grep "$cmdKey" | grep -v grep | awk '{print $1}'`
    if [ "${pid}" == "" ]; then
        echo "$resName is not running."
        exit 1
    fi
    #查询资源实例进程状态
    state=`cat /proc/$pid/status | grep "State" | awk '{print $2}'`
    if [ "$state" == "T" ]; then
        #僵死检查和处理
        if [ ! -f $phony_dead_time_file ]; then
            touch ./${phony_dead_time_file}
            echo "export firstphonytime=" > ./${phony_dead_time_file}
        fi
        source ./${phony_dead_time_file};
        curtime=$(date +%s);
        if [ "$firstphonytime" == "" ]; then
            #首次检测到资源僵死，将首次检测到僵死的时间写入僵死时间存储文件
            #firstphonytime 为用于保存当前资源实例僵死时间的变量名称，
            #若当前节点存在多个自定义资源实例，该名称需要指定为不同的名称
            echo "export firstphonytime=$curtime" > ./${phony_dead_time_file};
            exit 0;
        fi
        dead_time=$(( $curtime - $firstphonytime ));
        #若僵死时间大于等于用户设定的最大僵死时间，则立即杀死资源实例，否则不做处理
        #正常退出
    fi
}
```

```
if [ $dead_time -ge $PHONY_MAX_TIME ]; then
    echo "$resName is detected in a state of phony dead(T) and will be forcibly
killed!"
    kill -9 $pid
    rm ./${phony_dead_time_file} -f
    sh ${shardingPath}/start.sh; exit $?
else
    exit 0
fi
elif [ "$state" == "S" ]; then
    #未处于僵死状态清理环境后正常退出
    rm ./${phony_dead_time_file} -f
    exit 0
fi
}
```

#以下为固定接口无需更改，必须实现

```
if [ $1 == '-start' ]; then
    exec_start $2
elif [ $1 == '-stop' ]; then
    exec_stop $2
elif [ $1 == '-check' ]; then
    exec_check $2
elif [ $1 == '-clean' ]; then
    exec_stop $2
elif [ $1 == '-reg' ]; then
    exit 0
elif [ $1 == '-unreg' ]; then
    exit 0
elif [ $1 == '-isreg' ]; then
    exit 11
else
    echo "Please confirm the input parameters."
    exit 1
fi
```

以上样例可以作为模板使用，用户主要需要修改的地方包括：资源名称、资源 binPath、用于过滤资源实例的命令关键词、用于保存首次检测到资源僵死时间的文件（可选）、最长僵死时间、记录首次僵死时间的变量名（如果同一节点存在多个不同的自定义资源实例）

- (2) 自定义资源配置文件 cm_resource.json 该文件位置为 cmdir/cm_agent/cm_resource.json, 配置该文件后需要重启集群

```
{
  "resources": [
    {
      "name": "sharding",
      "resource_type": "APP",
      "instances": [
        {
          "node_id": 1,
          "res_instance_id": 1
        },
        {
          "node_id": 2,
          "res_instance_id": 2
        }
      ],
      "script": "/usr2/gbase/install/cm/cm_agent/sharding.sh",
      "check_interval": 1,
      "time_out": 5,
      "restart_delay": 3,
      "restart_period": 5,
      "restart_times": 10
    },
    {
      "name": "test",
      "resource_type": "APP",
      "instances": [
        {
          "node_id": 1,
          "res_instance_id": 1
        },
        {
          "node_id": 2,
          "res_instance_id": 2
        }
      ],
      "script": "/usr2/gbase/install/cm/cm_agent/test.sh",
      "check_interval": 1,
      "time_out": 5,
      "restart_delay": 0,
      "restart_period": 0,
      "restart_times": 1000
    }
  ]
}
```

```
}  
]  
}
```

配置说明：

- resources: 自定义资源对象列表，名称固定不能更改。
- name: 自定义资源对象名称，字符串类型，最大长度为 32（包含末尾'\0'）。
- resource_type: 资源类型，取值[“APP”，“DN”]，APP 表示为自定义资源，DN 表示为数据库资源
- instances: 自定义资源所在节点列表。
- node_id: 资源实例所在节点的 node_id。
- res_instance_id: 资源实例 id，大于等于 0，同一种资源的不同实例 id 不同。
- script: 资源脚本位置。
- check_interval: 上报资源状态时间间隔，大于等于 0，单位 s。
- time_out: 脚本执行的超时时间，大于等于 0，单位 s。
- restart_delay: 故障之后重启延迟时间，单位为 s，取值范围[0,1800]。
- restart_period: 当前时间-最近重启时间若大于 restart_period，则再次重启资源重启次数加 1。
- restart_times: 周期内最多重启次数，超过则不再重启，并将资源标记为不可用，取值范围[0,9999]，0 表示无限重启。

注意：

资源配置文件需要在所有节点上都有，且保持一致。

用户需要保证资源脚本能够正确运行。

5.1.4 资源池化

磁盘心跳

各节点 cm_agent 定时向投票盘写入心跳，cm_server 获取投票盘中的磁盘心跳，作为主备共享模式下的仲裁依据。

网络心跳

节点间的连通性网络心跳检测的主逻辑在 `cma` 实现。为了避免频繁建连，心跳采用长连接的方式，即 `cma` 之间通过 TCP 长连接，进行心跳交互，由每个 `cma` 节点定时广播，并周期性将列表信息上报给 `cms`，此功能依赖各个节点的时钟完全同步。

仲裁最大集群

`cms` 根据网络心跳数据、磁盘心跳数据、共享盘状态等进行仲裁，选主子集群。

子集群满足条件：

- (1) 实例正常；
- (2) 相互网络通信正常；
- (3) 磁盘心跳正常；

不满足条件的剔除集群。

子集群仲裁规则：

- (1) 拥有节点多的子集群胜出；
- (2) 如果节点数一样多，则节点号小的胜出。

约束：

- 集群节点时钟同步。
- CM 使用的共享盘在安装之前请确保至少 150M 为空，否则可能会有历史数据影响。
- CM 使用的投票盘在安装之前请确保已清空，否则可能会有历史数据影响。
- 由于心跳是集群所有节点间进行的，在大集群的情况下可能会对网络产生一定影响，因此，最多只会启动 64 个节点的心跳检测（DMS 最多只支持 64 节点，满足使用要求）。
- 由于 CM 集群规模最多只支持 8 个备机，因此当前部署最多支持 9 副本的集群。
- Voting disk 和 share disk 确保包括至少 1G 空间，由 CM 独占，其他应用不能使用。
- 在资源池化架构仲裁模式下，才启用磁盘心跳。

5.1.5CM 支持 DN 仲裁

VIP 是虚拟 IP，随主 DN 状态进行动态绑定和切换，即如果 DN 主发生了切换，原主 DN 上的虚拟 IP 可能动态绑定到新主 DN 上，用户可以只通过该 IP 与数据库连接，不用感知数据库在哪个节点上。

仲裁流程：

新主：

- (1) 在新主上用别名(网卡名:DN 端口号)方式绑定 VIP。
- (2) 把 VIP 添加到数据库配置项 “listen_addresses” 中。

原主：

- (1) 在原主上解绑 VIP。
- (2) 把 VIP 从数据库配置项 “listen_addresses” 中删除
- (3) 功能开启：在数据库实例节点上每一个 cm_agent 数据目录下生成 cm_resource.json 配置文件，文件中包含 VIP 的 json 字段，可通过 cm_ctl 工具命令生成 VIP 配置文件，参考命令，可以通过执行 res 命令：cm_ctl show 方法查看 VIP 信息。

说明：在选主成功，业务恢复后，5s 内 VIP 可以生效。

约束：

- 不支持沙箱环境。
- 不支持灾备集群。
- VIP(网卡名:DN 端口号)不允许被其他非虚拟 IP 占用
- 每个节点上 cm_resource.json 配置文件必须一致。
- 只支持重启方式加载。
- 允许配置多个 VIP，但是每个实例最多支持 6 个。
- 只支持 IPV4。
- 需要可用 VIP，即需要可对外提供服务的 VIP（跟现有网卡处于同一网段中）。
- 手动生成 VIP 配置文件步骤，在每个节点上都要生成该 VIP 配置文件，且每个节点要求一致，需要重启集群才能生效（不建议手动操作）。

```
cm_ctl res --add --res_name="资源名"
"--res_attr="resources_type=VIP,float_ip=IP1"

cm_ctl res --edit --res_name="资源名" --add_inst="instance_id=6001"
--inst_attr="base_ip=IP2"

cm_ctl res --edit --res_name="资源名" --add_inst="instance_id=6002"
--inst_attr="base_ip=IP2"
```



```
cm_ctl res --edit --res_name="资源名" --add_inst="instance_id=6003"  
--inst_attr="base_ip=IP2"
```

在数据库 pg_hba.conf 以 sha256 方式添加 float_ip

警告：由于需要动态绑定和解绑 VIP，需要对网卡进行操作，故而需要对 ifconfig 进行提权，允许集群用户使用 sudo ifconfig 方式 绑定和解绑 VIP，此操作可能带来窃听、篡改、仿冒、拒绝服务等安全风险。

模式：

- CM 支持的 DN 仲裁模式主要分为：

- Quorum 模式：基于多数派模式仲裁，选出同步备

简介：CM 基于 Quorum 模式进行仲裁，当 DN 分片处于无主场景时，CM 在多数派 DN redo 完成后，选择 term 和 lsn 最大的节点(同步备)发送 failover 升主。

约束：最小满足一主两备集群

- DCF 模式：

自动选主模式：基于 paxos 协议：

简介：dcf 模式自动选主，在此场景下，CM 不再进行对 DN 选主，只负责数据采集，假死检测等。

约束：switchover 只能使用 cm_ctl switchover -n NODEID -D DATADIR

CM 配置：enable_dcf=ON、dn_arbitrate_mode=paxos

DN 配置：enable_dcf=ON

总体约束：

最小满足一主两备集群

默认安装：DCF 自动选主模式

- 资源池化模式：

简介：在此场景下，CM 不再进行对 DN 选主，只负责数据采集，假死检测等。

CM 配置：dn_arbitrate_mode=share_disk

5.1.6 支持集群信息查询和推送

功能介绍：

通过运行组件 CMRestAPI, CM 能够支持:

通过 http/https 服务远程查询到集群的状态, 便于管理人员、运维平台等监控集群状态。

在数据库集群发生切主事件时, 通过 http/https 服务及时地将集群最新的主备信息推送到应用端注册的接收地址, 便于应用端及时的感知到集群的主备变化, 从而能够快速连接到新的主机和备机。

参数说明:

-e 数据库环境变量文件, 必须指定。

-w 访问来源 ip 白名单, 如果不需要设置白名单则可以不用指定。 启动命令:

5.2 cm_ctl 工具介绍

cm_ctl 是 GBase 8s 提供的用来控制数据库实例服务的工具。该工具主要供 OM 调用, 及数据库实例服务自恢复时使用。cm_ctl 的主要功能有:

- 启动数据库实例服务、AZ 的所有实例、单个主机上的所有实例或单独启动某个实例进程。
- 停止数据库实例服务、AZ 的所有实例、单个主机上的所有实例或单独停止某个节点实例进程。
- 重启逻辑数据库实例服务。
- 查询数据库实例状态或者单个主机的状态。
- 切换主备实例或重置实例状态。
- 重建备机。
- 查看数据库实例配置文件。
- 设置日志级别, 一主多备数据库实例部署下 cm_server 的仲裁模式、AZ 之间的切换模式。
- 获取日志级别, 一主多备数据库实例部署下 cm_server 的仲裁模式、AZ 之间的切换模式。
- 检测实例进程状态。

与 cm_ctl 工具相关的文件:

- cluster_manual_start

该文件是数据库实例启停标志文件。文件位于 `$_GAUSSHOME/bin_` 下。其中，`GAUSSHOME` 为环境变量。启动数据库实例时，`cm_ctl` 会删除该文件；停止数据库实例时，`cm_ctl` 会生成该文件，并向文件写入停止模式。

- `instance_manual_start_X` (X 是实例编号)

该文件是单个实例启停标志文件。文件位于 `$_GAUSSHOME/bin_` 下。其中，`GAUSSHOME` 为环境变量。启动实例时，`cm_ctl` 会删除该文件；停止实例时，`cm_ctl` 会生成该文件，并向文件写入停止模式。

`cm_ctl` 的相关约束：

在集群模式下，使用 `cm_ctl` 集群工具来切换数据库角色，而不是 `gs_ctl` 数据库工具。

5.2.1 命令说明

`cm_ctl` 参数可分为 option 参数、公共参数、start 模式的参数、switchover 模式的参数、build 模式的参数、check 模式的参数、stop 模式的参数、query 模式的参数、view 模式的参数、set 模式的参数、get 模式的参数、setrunmode 模式的参数、changerole 模式的参数、changemember 功能的参数、reload 模式的参数、list 模式的参数、encrypt 模式的参数、ddb 模式的参数、switch 模式的参数、res 模式的参数。

使用方法：

```
cm_ctl start [-z AVAILABILITY_ZONE [--cm_arbitration_mode=ARBITRATION_MODE]] |
[-n NODEID [-D DATADIR]] [-t SECS]
cm_ctl switchover [-z AVAILABILITY_ZONE] | [-n NODEID -D DATADIR [-f]] | [-a] |
[-A] [-t SECS]
cm_ctl finishredo
cm_ctl build [-c] [-n NODEID] [-D DATADIR [-t SECS] [-f] [-b full] [-j NUM]]
cm_ctl check -B BINNAME -T DATAPATH
cm_ctl stop [[-z AVAILABILITY_ZONE] | [-n NODEID [-D DATADIR]]] [-t SECS] [-m
SHUTDOWN-MODE]
cm_ctl query [-z ALL] [-l FILENAME] [-v [-C [-s] [-S] [-d] [-i] [-F] [-x] [-p]]
| [-r]] [-t SECS] [--minorityAz=AZ_NAME]
cm_ctl view [-v | -N | -n NODEID] [-l FILENAME]
cm_ctl set [--log_level=LOG_LEVEL] [--cm_arbitration_mode=ARBITRATION_MODE]
[--cm_switchover_az_mode=SWITCHOVER_AZ_MODE]
[--cmsPromoteMode=CMS_PROMOTE_MODE -I INSTANCEID]
cm_ctl set --param --agent | --server [-n [NODEID]] -k [PARAMETER]="[value]"
cm_ctl get [--log_level] [--cm_arbitration_mode] [--cm_switchover_az_mode]
```

```

cm_ctl setrunmode -n NODEID -D DATADIR [[--xmode=normal] | [--xmode=minority
--votenum=NUM]]
cm_ctl changerole [--role=PASSIVE | --role=FOLLOWER] -n NODEID -D DATADIR [-t
SECS]
cm_ctl changemember [--role=PASSIVE | --role=FOLLOWER] [--group=xx]
[--priority=xx] -n NODEID -D DATADIR [-t SECS]
cm_ctl reload --param [--agent | --server]
cm_ctl list --param --agent | --server
cm_ctl encrypt [-M MODE] -D DATADIR
cm_ctl ddb DCC_CMD
cm_ctl switch [--ddb_type=DDB]] [--cgbaseit] [--rollback]

```

表 5-1 option 参数

参数	参数说明
start	一主多备数据库部署模式下启动数据库实例服务、单个主机上的所有实例或单独启动某个节点实例进程，或者直接启动整个 AZ。
switchover	一主多备数据库部署模式下切换数据库主备实例。
finishredo	所有备机停止回放，每个分片中选择一个强制升主。 注意： 该参数属于高风险操作，请谨慎执行。
build	重建备实例。
check	检测实例进程运行状态，用户无需关注，不建议使用。
stop	一主多备数据库部署模式下停止数据库实例服务、单个主机上的所有实例或单独停止某个节点实例进程。
query	一主多备数据库部署模式下查询数据库实例状态或者单个主机的状态。

参数	参数说明
view	查看数据库实例配置文件。
set	设置日志级别，一主多备数据库部署模式下 cm_server 的仲裁模式、AZ 之间的切换模式、cm_server 升主模式。
set --param	设置 cm 参数，默认 set 所有节点上的参数，也可以通过 -n 参数指定 set 某个节点，具体参数可以参考 CM 配置参数介绍
get	获取日志级别，一主多备数据库部署模式下 cm_server 的仲裁模式、AZ 之间的切换模式。
setrunmode	DCF 部署方式下，设置 DCF 投票数，主要用于 DCF 强启。
changerole	DCF 模式下，将角色为 primary 的修改为 passive 或者 follower。
changemember	DCF 模式下，改变指定 DCF 节点属性，包括节点角色、节点所在的逻辑组、节点的选举优先级等。
reload	在线加载数据库实例静态配置文件，用户无需关注，不建议使用。
reload --param	加载可以动态生效的 cm 参数，部分参数不支持 reload，只能重启 cm 才能生效。
list	列出 cm_agent 或 cm_server 所有的参数。
encrypt	对输入的密码进行加密操作，密码支持 8~15 位，且必须包含三种字符（数字，字母，符号）。
ddb	DCC 或者 share disk 模式下，执行对 DCC 组件或者 share disk 的配置数

参数	参数说明
	据存取命令。CM 依赖 DCC 组件或者 share disk 对配置数据分布式存取，实现集群配置管理高可用能力。
switch	<p>执行 ddb 模式的切换。</p> <p>说明：</p> <p>GBase 8s 当前只支持切换到 DCC 模式。</p>
res	在节点上进行资源配置的新增、修改、查询、删除。
show	<p>展示集群 DN 节点网络连通性和每个节点的共享盘连通性</p> <p>说明：</p> <ol style="list-style-type: none"> 1. 节点连通性使用的是 DN 的 HAIP。 2. 共享盘连通性的前提是配置了 cma 的 voting_disk_path 参数，否则默认显示不连通。

表 5-2 公共参数

参数	参数说明
-D DATADIR	指定实例数据目录。仅用于对数据库节点进行操作的命令，如 start、stop、switchover、build、setrunmode、changerole、changemember、encrypt。
-l FILENAME	查询结果输出到指定文件。仅用于查询类的命令，如 query、view。
-n NODEID	指定节点。
-z	指定 AZ 名称。

参数	参数说明
AVAILABILITY_ZONE	
-t SECS	指定超时时间。
-V, --version	打印 cm_ctl 版本信息，然后退出。
-, -h, --help	显示关于 cm_ctl 命令行参数的帮助信息，然后退出。

说明：

此处列出的公共参数并不一定适用于所有命令，而是多个命令支持，为避免冗余信息，所以统一在此说明，详细的使用方法见以上使用方法，也可以使用 `cm_ctl -help` 进行查询。

表 5-3 switchover 参数

参数	参数说明
-a	将集群的主机重置到初始配置的节点。
-A	将主机切换到一个合适的备机节点上。
-f	指定进行-f 类型 switchover。使用方式： <code>cm_ctl switchover -n NODEID -D DATADIR -f。</code>
-z	将主机切换到-z 参数指定的 AZ。 注意： switchover 为维护操作：确保数据库实例状态正常，所有业务结束，并使用 <code>pgxc_get_senders_catchup_time()</code> 视图查询无主备追赶后，再进行 switchover 操作。

表 5-4 build 参数

参数	参数说明
-f	强制重建备机。
-b full	指定进行全量 build。不指定情况下，对于一主多备数据库实例部署模式进行 auto build。auto build 指：先调用增量 build，失败之后调用全量 build。
-c	重建 cm_server（将主节点的 DCC 数据目录拷贝到指定节点，只适用于一主一备模式）。

表 5-5 check 参数

参数	参数说明
-B BINNAME	指定进程名，其进程名包括“cm_agent”，“gaussdb”和“cm_server”。
-T DATAPATH	指定实例数据目录。

表 5-6 stop 参数

参数	参数说明
-m SHUTDOWN-MODE	指定停止模式，停止模式有以下几种： smart (s)：等待用户业务结束后，停止所有数据库实例。 fast (f)：不等待用户业务结束，指定数据库实例退出。 immediate (i)：不等待用户业务结束，指定数据库实例强制退出。
-I	指定需要停止的资源实例号，可以通过 cm_ctl query -Cv 查看资源实例和实例号的映射关系

表 5-7 query 参数

参数	参数说明
-s	<p>显示导致各个节点主备实例数量不均衡的实例。</p> <p>说明：</p> <p>-s 参数需要和-v 以及-C 参数一起使用才能按主备关系成对显示导致各个节点主备实例数量不均衡的实例，使用-s 参数时，必须指定-C、-v 参数。</p>
-C	<p>按主备关系成对显示数据库实例状态。</p> <p>说明：</p> <p>-C 参数需要和-v 参数一起使用才能按主备关系成对显示数据库实例详细状态信息，使用-C 时，必须指定-v 参数。</p>
-v	<p>显示详细数据库实例状态。</p> <p>说明：</p> <p>数据库实例有如下几种状态：</p> <p>Normal：表示数据库实例可用，且数据有冗余备份。所有进程都在运行，主备关系正常。</p> <p>Degraded：表示数据库实例可用，但数据没有冗余备份。</p> <p>Unavailable：表示数据库实例不可用。</p>
-d	<p>显示实例数据目录。</p> <p>说明：</p> <p>-d 参数需要和-v 以及-C 参数一起使用。</p>
-i	<p>显示物理节点 ip。</p> <p>说明：</p>

参数	参数说明
	-i 参数需要和-v 以及-C 参数一起使用。
-F	<p>显示各个节点 Fenced UDF 状态。</p> <p>说明：</p> <p>-F 参数需要和-v 以及-C 参数一起使用才能显示各个节点 Fenced UDF 状态，使用-F 参数时，必须指定-C、-v 参数。</p>
-z ALL	<p>显示数据库实例所在 AZ 名称。</p> <p>说明：</p> <p>-z 参数需要和-v 以及-C 参数一起使用，并且-z 后面需要跟参数 ALL。</p>
-r	<p>显示备机 redo 状态。</p> <p>说明：</p> <p>使用-r 时，必须指定-v 参数。</p>
-g	显示备份和恢复群集信息。
-x	<p>显示所有异常的数据库实例。</p> <p>说明：</p> <p>-x 参数需要和-v 以及-C 参数一起使用。</p>
-S	<p>显示数据库实例启动时的状态检查结果。</p> <p>说明：</p> <p>-S 参数需要和-v 以及-C 参数一起使用才能按显示数据库实例的状态检查结果。有以下三种状态：</p> <p>Normal：表示数据库实例可用，且数据有冗余备份。所有进程都在运行，主备关系正常。</p>

参数	参数说明
	<p>Degraded: 表示数据库实例可用, 但数据没有冗余备份。</p> <p>Unavailable: 表示数据库实例不可用。</p>
--minorityAz	<p>只查询指定 AZ 的 cms。</p> <p>说明:</p> <p>此参数会忽略非指定 AZ 的 cms 节点, 可以在少数派场景提高查询速度。</p>
-p	<p>显示数据库实例所有节点端口。</p> <p>说明:</p> <p>-p 参数需要和-v 以及-C 参数一起使用。</p>

表 5-8 set 参数

参数	参数说明
--log_level=LOG_LEVEL	<p>设置主 cm_server 日志级别。共分为 DEBUG5、DEBUG1、WARNING、LOG、ERROR 和 FATAL 六个级别, 日志打印信息级别越来越高。设置日志级别越高, 输出日志信息就越少。</p>
--cm_arbitration_mode=ARBITRATION_MODE	<p>一主多备功能, 设置 cm_server 的仲裁模式。共有 MAJORITY、MINORITY 两种模式, MAJORITY 为多数派模式, MINORITY 为少数派模式。GBase 8s 不支持少数派, 此参数可以设置成 MINORITY, 但不会生效。</p>
--cm_switchover_az_mode=SWITCHOVER_AZ_MODE	<p>一主多备功能, 设置 AZ 之间的自动切换开关。共有 NON_AUTO、AUTO 两种模式, NON_AUTO 为非自动切换模式, AUTO 为自动切换模式。AUTO 模</p>

参数	参数说明
	式由主 cm_server 自动控制 AZ1 和 AZ2 之间的节点实例切换。
--cmsPromoteMode=CMS_PROMOTE_MODE -I INSTANCEID	设置 cm_server 的升主方式，共有 AUTO、PRIMARY_F 两种模式，AUTO 为默认自选主方式，PRIMARY_F 会强制使-I 指定的节点升主，无论当前是否有主，因此，有可能会导导致 cms 存在多主情况。

表 5-9 set cm 参数

参数	参数说明
--param	表明需要设置 cm 参数，不带此参数则不能执行设置 cm 参数。
--agent --server	此参数为必带参数，表明当前需要设置 cm_server 还是 cm_agent 的参数。
-k parameter="value"	指定设置的参数和参数的值，只能设置已经存在的参数，不支持增加或删除参数。

表 5-10 get 参数

参数	参数说明
--log_level=LOG_LEVEL	获取主 cm_server 日志级别。共分为 DEBUG5、DEBUG1、WARNING、LOG、ERROR 和 FATAL 六个级别，日志打印信息级别越来越高。设置日志级别越高，输出日志信息就减少。

参数	参数说明
<pre>--cm_arbitration_mode=ARBITRATION_MODE</pre>	<p>一主多备功能，获取 cm_server 的仲裁模式。共有 MAJORITY、MINORITY 两种模式，MAJORITY 为多数派模式，MINORITY 为少数派模式。少数派模式适用于一主多备数据库部署并且只有 AZ3 存活时，此时 cm_server 可以进行正常的仲裁业务，非此模式下将仲裁模式设置成少数派成功后，cm 会自动将仲裁模式改为多数派，以保证集群正常运转；多数派模式适用于一主多备数据库部署并且各个组件（cm_server，节点）存活数量大于一半的场景。数据库实例正常情况下默认为多数派模式。</p> <p>注意： GBase 8s 不支持少数派</p>
<pre>--cm_switchover_az_mode=SWITCHOVER_AZ_MODE</pre>	<p>一主多备功能，获取 AZ 之间的自动切换开关。共有 NON_AUTO、AUTO 两种模式，NON_AUTO 为非自动切换模式，AUTO 为自动切换模式。AUTO 模式由主 cm_server 自动控制 AZ1 和 AZ2 之间的节点实例切换。</p>

表 5-11 view 参数

参数	参数说明
-v	<p>显示数据库实例所有节点的静态配置详细信息。</p> <p>说明： 相对于 cm_ctl view 命令，-v 显示内容增加了 cmserver、节点组件的编号显示，如下：cmserverInstanceID, datanodeInstanceID。</p>
-N	<p>只显示本节点的静态配置信息，即执行 cm_ctl view 命令的节点的信息，N 表示 Native。</p>

表 5-12 setrunmode 参数

参数	参数说明	取值范围
--xmode	指定 DCF 的运行模式。	normal: 正常模式。 minority: 少数派模式, 需要--votenum 指定投票数。
--votenum	指定 DCF 少数派运行方式的投票数。	正整数, 不高于 DCF 总副本数。

表 5-13 changerole 参数

参数	参数说明	取值范围
--role	DCF 模式下, 将角色为 primary 的修改为 passive 或者 follower。	passive: passive 角色。 follower: follow 角色。

表 5-14 changemember 参数

参数	参数说明	取值范围
--role	DCF 模式下, 将角色为 primary 的修改为 passive 或者 follower。	passive: passive 角色。 follower: follow 角色。
--group	DCF 模式下, 修改 group 的值。	0~2147483647
--priority	DCF 模式下, 修改	0~2147483647

参数	参数说明	取值范围
	priority 的值。	

表 5-15 start 参数

参数	参数说明
<code>--cm_arbitration_mode=ARBITRATION_MODE</code>	<p>一主多备功能, 获取 cm_server 的仲裁模式。共有 MAJORITY、MINORITY 两种模式, MAJORITY 为多数派模式, MINORITY 为少数派模式。少数派模式适用于一主多备数据库部署并且只有 AZ3 存活时, 此时 cm_server 可以进行正常的仲裁业务, 非此模式下将仲裁模式设置成少数派成功后, cm 会自动将仲裁模式改为多数派, 以保证集群正常运转; 多数派模式适用于一主多备数据库部署并且各个组件 (cm_server, 节点) 存活数量大于一半的场景。数据库实例正常情况下默认为多数派模式。</p> <p>注意:</p> <p>GBase 8s 不支持少数派</p>
<code>-I</code>	指定需要启动的资源实例号, 可以通过 <code>cm_ctl query -Cv</code> 查看资源实例和实例号的映射关系

表 5-16 reload 参数

参数	参数说明
<code>--param</code>	表明需要加载 cm 参数, 不带此参数则不能执行加载 cm 参数。
<code>--agent --server</code>	表明当前需要动态加载 cm_server 还是 cm_agent 的参数。

表 5-17 list 参数

参数	参数说明
--param	此参数为必带参数，表明需要列出 cm 参数信息。
--agent --server	此参数为必带参数，表明当前需要查看 cm_server 还是 cm_agent 的参数。

表 5-18 encrypt 参数

参数	参数说明
-M	指定加密类型，支持 server、client。默认类型为 server。
-D	指定生成的加密密码文件路径。

表 5-19 switch 参数

参数	参数说明
--ddb_type=[DDB]	选择需要切换 DDB 模式（GBase 8s 只支持 DCC 模式）。
--cgbaseit	执行切换会使数据库实例无法选主，需要执行 cgbaseit 恢复数据库实例。
--rollback	回滚操作，执行切换失败需要执行回滚操作。

表 5-20 ddb 参数

参数	参数说明
--put [key] [value]	往 DCC 或者 share disk 中插入键值对，如果键值对已存在则会修改键 key 所对应的值 value。
--get [key]	查询 DCC 或者 share disk 中 key 对应的 value。
--delete [key]	删除 DCC 或者 share disk 中指定的键值对。
--prefix	get 或者 delete 后添加 prefix 参数,可以实现模糊匹配查询和删除。
--cluster_info	DCC 部署模式下获取数据库实例信息，share disk 部署模式下不支持该命令。
--leader_info	DCC 部署模式下获取主节点信息，share disk 部署模式下不支持该命令。
--help, -h	显示 DDB 命令帮助信息。
--version, -v	DCC 部署模式下显示 DCC 版本信息，share disk 部署模式下不支持该命令。

表 5-21 res 参数

参数	参数说明
--add --res_name=[name] --res_attr=[res_info]	新增资源配置，name 为新增资源的名称，res_info 为新增资源的信息。

参数	参数说明
<code>--del --res_name=[name]</code>	删除资源配置，name 为删除资源的名称。
<code>--edit --res_name=[name] --res_attr=[res_info]</code>	修改资源配置，name 为被修改资源的名称，res_info 为新修改的资源信息。
<code>--edit --res_name=[name] --add_inst=[inst_info]</code>	新增资源实例，name 为新增实例的资源名称，inst_info 为新增实例的信息。
<code>--edit --res_name=[name] --del_inst=[inst_info]</code>	删除资源实例，name 为删除实例的资源名称，inst_info 为删除实例的信息。
<code>--check</code>	检查资源配置。不合理处会显示 Warning，配置不可用才会报错。

5.2.2 命令参考

启动实例：

```
cm_ctl start [-z AVAILABILITY_ZONE [--cm_arbitration_mode=ARBITRATION_MODE]] |
[-n NODEID [-D DATADIR]] [-t SECS]
```

数据库主备倒换：

```
cm_ctl switchover [-z AVAILABILITY_ZONE] | [-n NODEID -D DATADIR [-f]] | [-a] |
[-A] [-t SECS]
```

所有备机停止回放，每个分片中选择一个强制升主：

```
cm_ctl finishredo
```

重建备节点：

```
cm_ctl build -n NODEID -D DATADIR [-t SECS] [-f] [-b full]
```

检测实例进程运行状态：

```
cm_ctl check -B BINNAME -T DATAPATH
```

停止实例：

```
cm_ctl stop [[-z AVAILABILITY_ZONE] | [-n NODEID [-D DATADIR [-R]]]] [-t SECS]
[-m SHUTDOWN-MODE]
```

查询集群状态：

```
cm_ctl query [-z ALL] [-l FILENAME] [-v [-C [-s] [-S] [-d] [-i] [-F] [-x] [-p]]
| [-r]] [-t SECS] [--minorityAz=AZ_NAME]
```

查看集群配置文件：

```
cm_ctl view [-v | -N | -n NODEID] [-l FILENAME]
```

设置参数：

```
cm_ctl set [--log_level=LOG_LEVEL] [--cm_arbitration_mode=ARBITRATION_MODE]
[--cm_switchover_az_mode=SWITCHOVER_AZ_MODE]
```

设置 CM 参数：

```
cm_ctl set --param --agent | --server [-n NODEID] -k "PARAMETER='value'"
```

获取参数：

```
cm_ctl get [--log_level] [--cm_arbitration_mode] [--cm_switchover_az_mode]
```

设置 DCF 投票数：

```
cm_ctl setrunmode -n NODEID -D DATADIR [[--xmode=normal] | [--xmode=minority
--votenum=NUM]]
```

改变 dcf 角色信息：

```
cm_ctl changerole [--role=PASSIVE | --role=FOLLOWER] -n NODEID -D DATADIR [-t
SECS]
```

改变 dcf 节点属性：

```
cm_ctl changemember [--role=PASSIVE | --role=FOLLOWER] [--group=xx]
[--priority=xx] -n NODEID -D DATADIR [-t SECS]
```

动态加载 CM 参数：

```
cm_ctl reload --param [--agent | --server]
```

列出所有 CM 参数：

```
cm_ctl list --param [--agent | --server]
```

加密：

```
cm_ctl encrypt [-M MODE] -D DATADIR
```

执行 DDB 命令行：

```
cm_ctl ddb DDB_CMD
设置：cm_ctl ddb --put [key] [value]
删除：cm_ctl ddb --delete [key]
```

查看 DDB 命令帮助信息：cm_ctl ddb --help

执行 switch ddb 命令：

```
cm_ctl switch [--ddb_type=[DDB]] [--cgbaset] [--rollback]
```

执行 res 命令：

```
新增资源：cm_ctl res --add --res_name=[name] --res_attr=[res_info]
删除资源：cm_ctl res --del --res_name=[name]
修改资源：cm_ctl res --edit --res_name=[name] --res_attr=[res_info]
新增资源实例：cm_ctl res --edit --res_name=[name] --add_inst=[inst_info]
删除资源实例：cm_ctl res --edit --res_name=[name] --del_inst=[inst_info]
检查资源：cm_ctl res --check
```

5.3 cm_persit 工具介绍

功能介绍

cm_persist 工具用来在资源池化设备上实现抢占磁盘锁功能。该工具是系统内部工具，由 cm_server 实例自动调用，不建议用户使用。

前提条件

确保存储设备支持 SCSI-3 及 CAW 协议。

确保 cm_persist 工具拥有 CAP_SYS_RAWIO 权限，拥有该权限后才可以调用 ioctl 系统函数操作资源池化设备。

语法

执行 cm_persist 命令

```
cm_persist [DEVICEPATH] [INSTANCE_ID] [OFFSET]
```

显示帮助信息

- cm_persist -? | --help

参数说明

- DEVICEPATH

资源池化设备的路径。

- INSTANCE_ID

cm_server 实例的 instance id, 可通过 cm_ctl query -Cv 命令查询获取。

- OFFSET

在指定的磁盘偏移地址上抢占磁盘锁。

- -?, -help

打印帮助信息。

5.4 安装和卸载工具介绍

使用 cm_install 工具可以在未部署 CM 的 GBase 8s 数据库集群上部署 CM, cm_uninstall 可以在已部署 CM 的 GBase 8s 集群上卸载掉 CM, 且不会影响 DN 集群。

注意事项

执行 cm_install 之后默认会启动集群。

需要以个人用户执行。

本工具允许在集群停止状态下进行安装或卸载 CM 工具, 但是需要注意的是, 集群停止之前的主应该为 xml 配置的初始主, 否则安装或卸载 CM 工具后再次启动后集群的主机会与集群停止前的主机不一致。

如果是在集群停止状态下卸载 CM 工具, 卸载后会删除动态文件, 但是由于集群处于停止状态, 所以无法生成, 如果需要的话可以在 gs_om 启动集群后使用 gs_om -t refreshconf 命令重新生成即可。

使用该工具前需要进入到该工具所在的 cm_tool 目录下。

前提条件

GBase 8s 集群是通过 OM 工具安装的, 需要有 OM 工具。

集群各节点之间存在互信, 通常通过 OM 工具安装的集群就能够保证这一点。

GBase 8s 集群的版本号与待安装的 CM 的版本号相同。

部署 CM 前集群状态需要为正常或停止, 并且主机的 term 必须为非 0 值且为集群中最大。

使用方法

安装

```
cm_install -? | --help  
cm_install -X XMLFILE [-e envFile] --cmpkg=cmpkgPath
```

卸载

```
cm_uninstall -? | --help  
cm_uninstall -X XMLFILE [-e envFile] [--deleteData] [--deleteBinary]
```

参数说明:

- -X
XML 文件的路径。
- -e
环境变量文件的路径，默认值为"~/bashrc"。
- -cmpkg
CM 包的路径。
- -deleteData
删除 CM 数据目录，默认不删除。
- -deleteBinary
删除 CM 相关二进制，包括 om_monitor、cm_agent、cm_server、cm_ctl，默认不删除。
- -?, -help
显示帮助信息。

5.5 安全设计

5.5.1 手动替换证书步骤

创建自认证证书

- 自认证证书生成请参考（证书生成）。
对私钥密码的保护，使用 cm encrypt 工具，请参考 cm_ctl 工具介绍。

```
cm_ctl encrypt [-M MODE] -D DATADIR
```

生成服务器端密钥因子 server.key.cipher、server.key.rand。

```
cm_ctl encrypt -M server -D DATADIR
```

```
please enter the password: (使用密码需要与服务器私钥的保护密码相同)
```

生成客户端密钥因子 client.key.cipher、client.key.rand。

```
cm_ctl encrypt -M client -D DATADIR
```

```
please enter the password: (使用密码需要与客户端私钥的保护密码相同)
```

5.5.2 证书使用指南

- (1) 使用证书需配置 cm_server 参数为 on (默认开启), cm_agent 有同名参数, 配置相同, 这里以 cm_server 为例。

```
cm_ctl set --param --server -k enable_ssl="on"
```

- (2) 证书文件需要存在于所有节点的 \$GAUSSHOME/share/sslcrt/cm 中, 证书替换后重启集群生效。所需证书文件: server.crt、server.key、client.crt、client.key、cacert.pem、server.key.cipher、server.key.rand、client.key.cipher、client.key.rand 根证书、密钥、证书以及密钥密码加密文件的权限, 需保证权限为 400。如果权限不满足要求, 则无法使用 ssl。

```
chmod 400 cacert.pem
```

```
chmod 400 server.crt
```

```
chmod 400 server.key
```

```
chmod 400 server.key.cipher
```

```
chmod 400 server.key.rand
```

```
chmod 400 client.crt
```

```
chmod 400 client.key
```

```
chmod 400 client.key.cipher
```

```
chmod 400 client.key.rand
```

证书有效期的检测周期为 1 天, 可通过 ssl_cert_expire_check_interval 设置。证书有效期剩余 90 天时会开始产生告警, 可通过 ssl_cert_expire_alert_threshold 设置。(配置参数详见 ssl_cert_expire_alert_threshold)。

- (3) 客户端和服务端的证书吊销列表分别为 client.crl、server.crl。

5.5.3 ifconfig 提权步骤

(1) 执行以下命令获取 ifconfig 绝对路径。

```
which ifconfig
```

(2) 执行以下命令修改权限文件/etc/sudoers。

```
Cmnd_Alias COMMAND_FLAG = ifconfig 绝对路径
```

```
集群用户名 ALL=(root) NOPASSWD: COMMAND_FLAG
```

(3) 可以缩小权限范围，例如：

```
Cmnd_Alias COMMAND_FLAG = /usr/sbin/ifconfig * netmask * up, /usr/sbin/ifconfig * down
```

注意：/usr/sbin/ifconfig 需要按实际修改为 ifconfig 的绝对路径

```
集群用户名 ALL=(root) NOPASSWD: COMMAND_FLAG
```

5.6 CM 配置参数介绍

cm_agent 相关参数可通过 cm_agent 数据目录下的 cm_agent.conf 文件查看，cm_server 相关参数可通过 cm_server 数据目录下的 cm_server.conf 文件查看。

5.6.1 cm_agent 参数

5.6.2 log_dir

参数说明：log_dir 决定存放 cm_agent 日志文件的目录。可以是绝对路径，或者是相对路径（相对于 \$GAUSSLOG 的路径）。通过 cm_ctl 设置绝对路径时需要将路径用"把路径包含起来，例如：cm_ctl set -param -agent -k log_dir="/log/dir"。

取值范围：字符串，最大长度为 1024。修改后需要重启 cm_agent 才能生效。参数修改请参考表 5-9 进行设置。

默认值：“log”，表示在 \$GAUSSLOG 下对应的 cm 目录下生成 cm_agent 日志。

5.6.3 log_file_size

参数说明：控制日志文件的大小。当 cm_agent-xx-current.log 日志文件达到指定大小时，则重新创建一个日志文件记录日志信息。

取值范围：整型，[0, 2047]，实际生效范围[1, 2047]，单位：MB。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：16MB。

5.6.4 log_min_messages

参数说明：控制写到 cm_agent 日志文件中的消息级别。每个级别都包含排在它后面的所有级别中的信息。级别越低，服务器运行日志中记录的消息就越少。

取值范围：枚举类型，有效值有 debug5、debug1、warning、error、log、fatal（不区分大小写）。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：warning

5.6.5 incremental_build

参数说明：控制重建备节点模式是否为增量。打开这个开关，则增量重建备节点；否则，全量重建备节点。

取值范围：布尔型。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

on、yes、true、1：表示增量重建备节点。

off、no、false、0：表示全量重建备节点。

默认值：on

5.6.6 security_mode

参数说明：控制是否以安全模式启动节点。打开这个开关，则以安全模式启动节点；否则，以非安全模式启动节点。

取值范围：布尔型。修改后可以动态生效。参数修改请参考表 5-9 进行设置。

on、yes、true、1：表示以安全模式启动节点。

off、no、false、0：表示以非安全模式启动节点。

默认值：off

5.6.7 upgrade_from

参数说明：就地升级过程中使用，用于标示升级前数据库的内部版本号，此参数禁止手

动修改。

取值范围：非负整型，[0, 4294967295]。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：0

5.6.8 alarm_component

参数说明：在使用第一种告警方式时，设置用于处理告警内容的告警组件的位置。通过 cm_ctl 设置绝对路径时需要将路径用"把路径包含起来，例如：`cm_ctl set -param -agent -k alarm_component="/alarm/dir"`。

取值范围：字符串，最大长度为 1024。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：/opt/huawei/snas/bin/snas_cm_cmd

5.6.9 alarm_report_interval

参数说明：指定告警上报的时间间隔。

取值范围：非负整型，[0, 2147483647]，单位：秒。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：1

5.6.10 agent_heartbeat_timeout

参数说明：cm_server 心跳超时时间。

取值范围：整型，[2, 2147483647]，单位：秒。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：5

5.6.11 agent_connect_timeout

参数说明：cm_agent 连接 cm_server 超时时间。

取值范围：整型，[0, 2147483647]，单位：秒。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：1

5.6.12agent_connect_retries

参数说明：cm_agent 连接 cm_server 尝试次数。

取值范围：整型，[0, 2147483647]。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：15

5.6.13agent_kill_instance_timeout

参数说明：当 cm_agent 在无法连接 cm_server 主节点后，发起一次杀死本节点上所有实例的操作之前，所需等待的时间间隔。

取值范围：整型，[0, 2147483647]。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：0，不发起杀死本节点上所有实例的操作。

5.6.14agent_report_interval

参数说明：cm_agent 上报实例状态的时间间隔。

取值范围：整型，[0, 2147483647]。单位：秒。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：1

5.6.15alarm_report_max_count

参数说明：指定告警上报的最大次数。

取值范围：非负整型，[1, 2592000]。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：1

5.6.16agent_check_interval

参数说明：cm_agent 查询实例状态的时间间隔。

取值范围：整型，[0, 2147483647]，单位：秒。修改后可以 reload 生效，参数修改请参考

考表 5-9 进行设置。

默认值：2

5.6.17 enable_log_compress

参数说明：控制压缩日志功能。

取值范围：布尔型。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

on、yes、true、1：表示允许压缩日志。

off、no、false、0：表示不允许压缩日志。

默认值：on

5.6.18 process_cpu_affinity

参数说明：控制是否以绑核优化模式启动主节点进程。配置该参数为 0，则不进行绑核优化；否则，进行绑核优化，且物理 CPU 片数为 2n 个。仅支持 ARM。

取值范围：整型，[0, 2]。修改后需要重启数据库、cm_agent 才能生效，参数修改请参考表 5-9 进行设置。

默认值：0

5.6.19 enable_xc_maintenance_mode

参数说明：在数据库为只读模式下，控制是否可以修改 pgxc_node 系统表。

取值范围：布尔型。修改后需要重启 cm_agent 才能生效。参数修改请参考表 5-9 进行设置。

on、yes、true、1：表示开启可以修改 pgxc_node 系统表功能。

off、no、false、0：表示关闭可以修改 pgxc_node 系统表功能。

默认值：on

5.6.20 log_threshold_check_interval

参数说明：cm 日志压缩和清除的时间间隔，每 1800 秒压缩和清理一次。

生效范围：\$GAUSSLOG/cm/cm_ctl；\$GAUSSLOG/cm/cm_server；\$GAUSSLOG/cm/om_monitor 目录下的文件以及 \$GAUSSLOG/cm/cm_agent 目录下以

cm_agent-、system_call-、system_alarm-为前缀的日志文件。

取值范围：整型，[0, 2147483647]，单位：秒。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：1800

5.6.21log_max_size

参数说明：控制 cm 日志最大存储值，当 CM 日志总大小超过 (log_max_size*95/100) MB 时，根据压缩日志生成时间，依次删除历史压缩日志，直到日志总大小小于 (log_max_size*95/100) MB。

生效范围：\$GAUSSLOG/cm/cm_ctl；\$GAUSSLOG/cm/cm_server；\$GAUSSLOG/cm/om_monitor 目录下的文件以及 \$GAUSSLOG/cm/cm_agent 目录下以 cm_agent-、system_call-、system_alarm-为前缀的日志文件。

取值范围：整型，[0, 2147483647]，单位：MB。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：10240

5.6.22log_max_count

参数说明：cm 可存储的最多日志数量，当 cm 日志总个数超过该值，根据压缩日志文件名时间，删除超过保留天数 log_saved_days 的压缩日志。

生效范围：\$GAUSSLOG/cm/cm_ctl；\$GAUSSLOG/cm/cm_server；\$GAUSSLOG/cm/om_monitor 目录下的文件以及 \$GAUSSLOG/cm/cm_agent 目录下以 cm_agent-、system_call-、system_alarm-为前缀的日志文件。

取值范围：整型，[0, 10000]，单位：个。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：10000

5.6.23log_saved_days

参数说明：cm 压缩日志保存的天数，cm 压缩日志超过该值并且 cm 日志总个数超过 log_max_count，删除压缩日志。

生效范围：\$GAUSSLOG/cm/cm_ctl；\$GAUSSLOG/cm/cm_server；

\$GAUSSLOG/cm/om_monitor 目录下的文件以及 \$GAUSSLOG/cm/cm_agent 目录下以 cm_agent-、system_call-、system_alarm-为前缀的日志文件。

取值范围：整型，[0, 1000]，单位天。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：90



须知：日志压缩能力受限于内存限制，最多只能检测到 30000 个日志文件。总日志量超过 30000 个文件时，则无法保证日志能被正常压缩及删除。可以通过调整 log_saved_days 和 log_threshold_check_interval 快速清理已压缩日志文件。

5.6.24 agent_phony_dead_check_interval

参数说明：cm_agent 检测进程是否僵死的时间间隔。

取值范围：整型，[0, 2147483647]，单位：秒。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：10

5.6.25 unix_socket_directory

参数说明：unix 套接字的目录位置。通过 cm_ctl 设置绝对路径时需要将路径用"把路径包含起来，例如：cm_ctl set -param -agent -k unix_socket_directory="/unix/dir"。

取值范围：字符串，最大长度为 1024。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值："

5.6.26 dilatation_shard_count_for_disk_capacity_alarm

参数说明：扩容场景下，设置新增的扩容分片数，用于上报磁盘容量告警时的阈值计算。

取值范围：整型，[0, 2147483647]，单位：个。该参数设置为 0，表示关闭磁盘扩容告警上报；该参数设置为大于 0，表示开启磁盘扩容告警上报，且告警上报的阈值根据此参数设置的分片数量进行计算。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：1

5.6.27 enable_dcf

参数说明：DCF 模式开关。

取值范围：布尔型。修改后需要重启 cm_agent 才能生效。参数修改请参考表 5-9 进行设置。

on、yes、true、1：表示启用 dcf。

off、no、false、0：表示不启用 dcf。

默认值：off

5.6.28 disaster_recovery_type

参数说明：主备数据库灾备关系的类型。

取值范围：整型，[0, 2]。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

0 表示未搭建灾备关系。

1 表示搭建了 obs 灾备关系。

2 表示搭建了流式灾备关系

默认值：0

5.6.29 agent_backup_open

参数说明：灾备模式设置，开启后 CM 按照灾备模式运行。

取值范围：整型，[0, 2]。修改后需要重启 cm_agent 才能生效。参数修改请参考表 5-9 进行设置。

0 表示未搭建灾备关系。

1 表示搭建了 obs 灾备关系（之后不再支持）。

2 表示搭建了流式灾备关系。

默认值：0

5.6.30 disk_timeout

参数说明：磁盘心跳超时时间。

取值范围：整型，[0, 2147483647]，单位：秒。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：200

5.6.31 voting_disk_path

参数说明：投票盘路径。

取值范围：字符串，最大长度为 1024。修改后需要重启 cm_agent 才能生效。参数修改请参考表 5-9 进行设置。

默认值：无，需要安装时进行配置。

5.6.32 agent_rhb_interval

参数说明：cma 节点间网络连通性检测周期。

取值范围：整型，[0, 2147483647]，单位：毫秒。修改后需要重启 cm_agent 才能生效。参数修改请参考表 5-9 进行设置。

默认值：1000

5.6.33 enable_ssl

参数说明：ssl 证书开关。

取值范围：布尔型。打开后使用 ssl 证书加密通信。修改后需要重启 cm_agent 才能生效。参数修改请参考表 5-9 进行设置。

on、yes、true、1：表示启用 ssl。

off、no、false、0：表示不启用 ssl。

默认值：on



须知：出于安全性考虑，建议不要关闭该配置。关闭后 cm 将不使用加密通信，所有信息明文传播，可能带来窃听、篡改、冒充等安全风险。

5.6.34 ssl_cert_expire_alert_threshold

参数说明：ssl 证书过期告警时间。

取值范围：整型，[7, 180]，单位：天。证书过期时间少于该时间时，上报证书即将过

期告警。修改后需要重启 `cm_agent` 才能生效，参数修改请参考表 5-9 进行设置。

默认值：90

5.6.35ssl_cert_expire_check_interval

参数说明：ssl 证书过期检测周期。

取值范围：整型，[0, 2147483647]，单位：秒。修改后需要重启 `cm_agent` 才能生效，参数修改请参考表 5-9 进行设置。

默认值：86400

5.6.36enable_fence_dn

参数说明：`cm_agent` 连不上任何除了自身节点的 `cms`，并且自身节点 `cms` 不是 Primary 时，设置是否重启 `datanode` 进程。

取值范围：布尔型。修改后可以 `reload` 生效，参数修改请参考表 5-9 进行设置。

on、yes、true、1：表示重启 `datanode` 进程。

off、no、false、0：表示不重启 `datanode` 进程。

默认值：

5.6.37event_triggers

参数说明：该参数用于定义事件触发器。

取值范围：以字符串表示的 json 类型。

配置形式为：'{"trigger_type_1": "trigger_value_1", ..., "trigger_type_n": "trigger_value_n"}'
其中：

`trigger_type` 为事件触发器类型，当前支持的事件触发器类型为：`on_start`、`on_stop`、`on_failover`、`on_switchover` `trigger_value` 为发生对应事件时待执行的用户自定义触发器脚本

修改后重载 `cm_agent` 参数生效，参数修改请参考表 5-9 进行设置。

自定义脚本的输出会重定向至 `cm_agent` 日志目录下的 `system-callxxx.log` 中。

默认值："

约束条件：

trigger_value 即自定义脚本，必须为真实存在的 shell 脚本，且为绝对路径，并且对当前用户至少有读取和执行权限。

使用 cm_ctl set 命令配置该参数时，参数值必须符合 json 格式，并且将 json 类型表示为字符串类型，中间不能包含换行和空格。

参数值最大长度为 1024。

配置样例：`{ "on_start" :"/dir/on_start.sh", "on_stop" :"/dir/on_stop.sh", "on_failover" :"/dir/on_failover.sh", "on_switchover" :"/dir/on_switchover.sh"}`



注意：由于 CM 内部对各事件的执行均是异步执行，即将事件置于后台执行，所以 CM 在调用用户自定义的触发器脚本时，有可能事件还尚未执行完成，所以用户自定义触发器脚本中如果是需要等待事件完成后才执行动作的话，则需要在脚本中添加对应的状态检查，以确保事件完成。

db_service_vip

参数说明：数据库集群对外提供服务的 VIP。

取值范围：字符串。修改后可以 reload 生效。参数修改请参考表 5-9 进行设置。

默认值：无，安装时可选配置。

5.7 cm_server 参数

5.7.1 log_dir

参数说明：log_dir 决定存放 cm_server 日志文件的目录。它可以是绝对路径，或者是相对路径（相对于 \$GAUSSLOG 的路径）。通过 cm_ctl 设置绝对路径时需要将路径用"把路径包含起来，例如：`cm_ctl set -param -server -k log_dir="/log/dir"`。

取值范围：字符串，最大长度为 1024。修改后需要重启 cm_server 才能生效。参数修改请参考表 5-9 进行设置。

默认值：“log”，表示在 \$GAUSSLOG 下对应的 cm 目录下生成 cm_server 日志。

5.7.2 log_file_size

参数说明：控制日志文件的大小。当'cm_server-xx-current.log'日志文件达到指定大小时，则重新创建一个日志文件记录日志信息。

取值范围：整型，[0, 2047]，实际生效范围[1, 2047]，单位：MB。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：16MB

5.7.3 log_min_messages

参数说明：控制写到 cm_server 日志文件中的消息级别。每个级别都包含排在它后面的所有级别中的信息。级别越低，服务器运行日志中记录的消息就越少。

取值范围：枚举类型，有效值有 debug5、debug1、log、warning、error、fatal（不区分大小写）。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：warning

5.7.4 thread_count

参数说明：cm_server 线程池的线程数。

取值范围：整型，[2, 1000]。修改后需要重启 cm_server 才能生效。参数修改请参考表 5-9 进行设置。

默认值：1000

5.7.5 instance_heartbeat_timeout

参数说明：实例心跳超时时间。

取值范围：整型，[1, 2147483647]，单位：秒。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：6

5.7.6 instance_failover_delay_timeout

参数说明：cm_server 检测到主机宕机，failover 备机的延迟时间。

取值范围：整型，[0, 2147483647]，单位：秒。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：0

5.7.7 cmserver_ha_connect_timeout

参数说明：cm_server 主备连接超时时间。

取值范围：整型，[0, 2147483647]，单位：秒。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：2

5.7.8 cmserver_ha_heartbeat_timeout

参数说明：cm_server 主备心跳超时时间。

取值范围：整型，[1, 2147483647]，单位：秒。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：6

5.7.9 cmserver_ha_status_interval

参数说明：cm_server 主备同步状态信息间隔时间。

取值范围：整型，[1, 2147483647]，单位：秒。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：1

5.7.10 cmserver_self_vote_timeout

参数说明：cm_server 之间相互投票的超时时间。旧版本遗留参数，实际不生效。

取值范围：整型，[0, 2147483647]，单位：秒。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：6

5.7.11 phony_dead_effective_time

参数说明：用于数据库节点僵死检测，当检测到的僵死次数大于该参数值，认为进程僵死，将进程重启。

取值范围：整型，[1, 2147483647]，单位：次数。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：5

5.7.12cm_server_arbitrate_delay_base_time_out

参数说明：cm_server 仲裁延迟基础时长。cm_server 主断连后，仲裁启动计时开始，经过仲裁延迟时长后，将选出新的 cm_server 主。其中仲裁延迟时长由仲裁延迟基础时长、节点 index (server ID 序号) 和增量时长共同决定。公式为：仲裁延迟时长=仲裁延迟基础时长+节点 index*仲裁延迟增量时长参数。

取值范围：整型，[0, 2147483647]，单位：秒。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：10

5.7.13cm_server_arbitrate_delay_incremental_time_out

参数说明：cm_server 仲裁延迟增量时长。cm_server 主断连后，仲裁启动计时开始，经过仲裁延迟时长后，将选出新的 cm_server 主。其中仲裁延迟时长由仲裁延迟基础时长、节点 index (server ID 序号) 和增量时长共同决定。公式为：仲裁延迟时长=仲裁延迟基础时长+节点 index*仲裁延迟增量时长参数。

取值范围：整型，[0, 2147483647]，单位：秒。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：3

5.7.14alarm_component

参数说明：在使用第一种告警方式时，设置用于处理告警内容的告警组件的位置。参数修改请参考表 5-9 进行设置。通过 cm_ctl 设置绝对路径时需要将路径用"把路径包含起来，例如：cm_ctl set -param -server -k alarm_component="/alarm/dir"。

取值范围：字符串，最大长度为 1024。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：/opt/huawei/snas/bin/snas_cm_cmd

5.7.15alarm_report_interval

参数说明：指定告警上报的时间间隔。

取值范围：非负整型，[0, 2147483647]，单位：秒。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：3

5.7.16alarm_report_max_count

参数说明：指定告警上报的最大次数。

取值范围：非负整型，[1, 2592000]。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：1

5.7.17instance_keep_heartbeat_timeout

参数说明：cm_agent 会定期检测实例状态并上报给 cm_server，若实例状态长时间无法成功检测，累积次数超出该数值，则 cm_server 将下发命令给 agent 重启该实例。

取值范围：整型，[0, 2147483647]，单位：秒。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：40

5.7.18az_switchover_threshold

参数说明：若一个 AZ 内节点分片的故障率(故障的节点分片数 / 总节点分片数 * 100%)超过该数值，则会触发 AZ 自动切换。

取值范围：整型，[1, 100]。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：100

5.7.19az_check_and_arbitrate_interval

参数说明：当某个 AZ 状态不正常时，会触发 AZ 自动切换，该参数是检测 AZ 状态的时间间隔。

取值范围：整型，[1, 2147483647]，单位：秒。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：2

5.7.20az_connect_check_interval

参数说明：定时检测 AZ 间的网络连接，该参数表示连续两次检测之间的间隔时间。

取值范围：整型，[1, 2147483647]，单位：秒。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：60

5.7.21az_connect_check_delay_time

参数说明：每次检测 AZ 间的网络连接时有多次重试，该参数表示两次重试之间的延迟时间。

取值范围：整型，[1, 2147483647]，单位：秒。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：150

5.7.22cmserver_demote_delay_on_etcd_fault

参数说明：因为 etcd 不健康而导致 cm_server 从主降为备的时间间隔。

取值范围：整型，[1, 2147483647]，单位：秒。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：8

5.7.23instance_phony_dead_restart_interval

参数说明：当数据库实例僵死时，会被 cm_agent 重启，相同的实例连续因僵死被杀时，其间隔时间不能小于该参数数值，否则 cm_agent 不会下发命令。

取值范围：整型，[0, 2147483647]，单位：秒。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：21600

5.7.24enable_transaction_read_only

参数说明：控制数据库是否为只读模式开关。

取值范围：布尔型，有效值有 on, off, true, false, yes, no, 1, 0。修改后可以 reload

生效，参数修改请参考表 5-9 进行设置。

默认值：on

5.7.25 `datastorage_threshold_check_interval`

参数说明：检测磁盘占用的时间间隔。间隔时间由用户指定，表示检测一次磁盘的间隔时间。

取值范围：整型，[1, 2592000]，单位：秒。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：10

5.7.26 `datastorage_threshold_value_check`

参数说明：设置数据库只读模式的磁盘占用阈值，当某个节点的数据目录所在磁盘占用阈值超过这个阈值，如果该节点为备机，自动将该节点设置为只读，如果该节点为主机，则会自动将主机切换到一个合适的主机上。如果所有节点的阈值均达到阈值，则会将集群设置为只读。

取值范围：整型，[1, 99]。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：85

5.7.27 `max_datastorage_threshold_check`

参数说明：设置磁盘使用率的最大检测间隔时间。当用户手动修改只读模式参数后，会自动在指定间隔时间后开启磁盘检测操作。

取值范围：整型，[1, 2592000]，单位：秒。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：43200

5.7.28 `enable_az_auto_switchover`

参数说明：AZ 自动切换开关，若打开，则表示允许 `cm_server` 自动切换 AZ。否则当发生节点故障等情况时，即使当前 AZ 已经不再可用，也不会自动切换到其他 AZ 上，除非手动执行切换命令。

取值范围：非负整型，0 或 1，0：开关关闭，1：开关打开。修改后可以 reload 生效，

参数修改请参考表 5-9 进行设置。

默认值：1

5.7.29cm_krb_server_keyfile

参数说明：kerberos 服务端 key 文件所在位置，需要配置为绝对路径。该文件通常为 `/${GAUSSHOME}/kerberos` 路径下，以 `keytab` 格式结尾，文件名与集群运行所在用户名相同。与上述 `cm_auth_method` 参数是配对的，当 `cm_auth_method` 参数修改为 `gss` 时，该参数也必须配置为正确路径，否则将影响集群状态。通过 `cm_ctl` 设置绝对路径时需要将路径用"把路径包含起来，例如：`cm_ctl set -param -server -k cm_krb_server_keyfile=""/krb/dir"`。

取值范围：字符串类型，修改后可以 `reload` 生效，参数修改请参考表 5-9 进行设置。

默认值：`/${GAUSSHOME}/kerberos/{UserName}.keytab`，默认值无法生效，仅作为提示。

5.7.30switch_rto

参数说明：`cm_server` 强起逻辑等待时延。在 `force_promote` 被置为 1 时，当集群的某一片处于无主状态开始计时，等待该延迟时间后开始执行强起逻辑。

取值范围：整型，`[60, 2147483647]`，单位：秒。修改后可以 `reload` 生效，参数修改请参考表 5-9 进行设置。

默认值：600

5.7.31force_promote

参数说明：`cm_server` 是否打开强起逻辑（指集群状态为 `Unknown` 的时候以丢失部分数据为代价保证集群基本功能可用）的开关。0 代表功能关闭，1 代表功能开启。

取值范围：整型，`[0, 1]`。修改后可以 `reload` 生效，参数修改请参考表 5-9 进行设置。

默认值：0

5.7.32backup_open

参数说明：灾备集群设置，开启后 CM 按照灾备集群模式运行。

取值范围：整型，`[0, 1]`。修改后需要重启 `cm_server` 才能生效。非灾备集群不能开启该参数。参数修改请参考表 5-9 进行设置。

0 表示关闭。

1 表示开启

默认值：0

5.7.33enable_dcf

参数说明：DCF 模式开关。

取值范围：布尔型。修改后需要重启 `cm_server` 才能生效。参数修改请参考表 5-9 进行设置。

on、yes、true、1：表示启用 dcf。

off、no、false、0：表示不启用 dcf。

默认值：off

5.7.34ddb_type

参数说明：ETCD, DCC、share disk 模式切换开关。

取值范围：整型。0：ETCD；1：DCC；2：share disk。修改后需要重启 `cm_server` 才能生效。参数修改请参考表 5-9 进行设置。

默认值：1



说明：（GBase 8s 只支持 DCC 或者 share disk 模式）。

5.7.35enable_ssl

参数说明：ssl 证书开关。

取值范围：布尔型。打开后使用 ssl 证书加密通信。修改后需要重启 `cm_server` 才能生效。参数修改请参考表 5-9 进行设置。

on、yes、true、1：表示启用 ssl。

off、no、false、0：表示不启用 ssl。

默认值：on



须知：出于安全性考虑，建议不要关闭该配置。关闭后 `cm` 将不使用加密通信，所有信息明文传播，可能带来窃听、篡改、冒充等安全风险。

5.7.36ssl_cert_expire_alert_threshold

参数说明：ssl 证书过期告警时间。

取值范围：整型，[7, 180]，单位：天。证书过期时间少于该时间时，上报证书即将过期告警。修改后需要重启 cm_server 才能生效，参数修改请参考表 5-9 进行设置。

默认值：90

5.7.37ssl_cert_expire_check_interval

参数说明：ssl 证书过期检测周期。

取值范围：整型，[0, 2147483647]，单位：秒。修改后需要重启 cm_server 才能生效，参数修改请参考表 5-9 进行设置。

默认值：86400

5.7.38ddb_log_level

参数说明：设置 ddb 日志级别。

关闭日志：“NONE”，NONE 表示关闭日志打印，不能与以下日志级别混合使用。

开 启 日 志 ；
“RUN_ERR|RUN_WAR|RUN_INF|DEBUG_ERR|DEBUG_WAR|DEBUG_INF|TRACE|PROFILE|OPER” 日志级别可以从上述字符串中选取字符串并使用竖线组合使用，不能配置空串。

取 值 范 围 ； 字 符 串 ，
RUN_ERR|RUN_WAR|RUN_INF|DEBUG_ERR|DEBUG_WAR|DEBUG_INF|TRACE|PROFILE|OPER。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：RUN_ERR|RUN_WAR|DEBUG_ERR|OPER|RUN_INF|PROFILE

5.7.39ddb_log_backup_file_count

参数说明：最大保存日志文件个数。

取值范围：整型，[1, 100]。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：10

5.7.40ddb_max_log_file_size

参数说明：单条日志最大字节数。

取值范围：字符串，长度最大为 1024，[1M, 1000M]。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：10M

5.7.41ddb_log_suppress_enable

参数说明：是否开启日志抑制功能。

取值范围：整型，0：关闭； 1：开启。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：1

5.7.42ddb_election_timeout

参数说明：DCC 选举超时时间。

取值范围：整型，[1, 600]，单位：秒。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：3

5.7.43coordinator_heartbeat_timeout

参数说明：节点故障自动剔除心跳超时时间。设置后立即生效，不需要重启 cm_server。该参数设置为 0，则节点故障后不会自动剔除。

取值范围：整型，单位为秒。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：25

5.7.44cluster_starting_aribt_delay

参数说明：cm_server 在集群启动阶段，等待节点静态主升主的时间。

取值范围：整型，[1, 2592000]，单位：秒。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：180

5.7.45enable_e2e_rto

参数说明：端到端 RTO 开关，开启后僵死检测周期及网络检测超时时间将缩短，CM 可以达到端到端 RTO 指标（单实例故障 RTO≤10s，叠加故障 RTO≤30s）。

取值范围：整型，[0, 1]。1 表示开启，0 表示关闭。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：0

5.7.46disk_timeout

参数说明：磁盘心跳超时时间。

取值范围：整型，[0, 2147483647]，单位：秒。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：200

5.7.47agent_network_timeout

参数说明：节点间网络超时时间。

取值范围：整型，[0, 2147483647]，单位：秒。修改后可以 reload 生效，参数修改请参考表 5-9 进行设置。

默认值：6

5.7.48voting_disk_path

参数说明：投票盘路径。

取值范围：字符串，最大长度为 1024。修改后需要重启 cm_server 才能生效。

默认值：无，需要安装时进行配置。

5.7.49share_disk_path

参数说明：共享盘路径。

取值范围：字符串，最大长度为 1024。修改后需要重启 cm_server 才能生效。

默认值：无，需要安装时进行配置。

5.7.50dn_arbitrate_mode

参数说明：dn 仲裁模式。

取值范围：字符串。修改后可以 reload 生效。参数修改请参考表 5-9 进行设置，share_disk 模式下，不建议用户修改仲裁模式。

quorum

paxos

share_disk

默认值：quorum

5.7.51delay_arbitrate_max_cluster_timeout

参数说明：启动过程中，延迟仲裁最大集群时间。

取值范围：整型，[0, 1000]，单位：秒。0：表示不进行仲裁。修改后可以 reload 生效。参数修改请参考表 5-9 进行设置。

默认值：300

5.7.52delay_arbitrate_timeout

参数说明：设置等待跟主 DN 同 AZ 节点 redo 回放，优先选择同 AZ 升主的时间。

取值范围：整型，[0, 2147483647]，单位：秒。参数修改请参考表 5-9 进行设置。

默认值：0

5.7.53cm_auth_method

参数说明：CM 模块端口认证方式，trust 表示未配置端口认证，gss 表示采用 kerberos 端口认证。必须注意的是：只有当 kerberos 服务端和客户端成功安装后才能修改为 gss，否则 CM 模块无法正常通信，将影响数据库状态。

取值范围：枚举类型，有效值有 trust, gss。修改后需要重启 cm_server 才能生效。参数修改请参考表 5-9 进行设置。

默认值：trust

5.7.54third_party_gateway_ip

参数说明：CM 两节点部署模式必备参数。当前 AZ 中第三方网关 IP 地址或任何其他独立于当前集群的可用 IP 地址，需要确保其与集群中节点间的网络相通。

取值范围：字符串。修改后可以 reload 生效。参数修改请参考表 5-9 进行设置。

默认值：无，需要安装时进行配置。

5.7.55cms_enable_failover_on2nodes

参数说明：CM 两节点部署模式必备参数。是否允许 CM 集群自身故障自动切换，默认禁止 CM 集群自身自动故障切换。

取值范围：布尔型。修改后可以 reload 生效。参数修改请参考表 5-9 进行设置。

on、yes、true、1：表示允许 CM 集群自身自动故障切换。

off、no、false、0：表示禁止 CM 集群自身自动故障切换。

默认值：false

5.7.56cms_enable_db_crash_recovery

参数说明：CM 两节点部署模式必备参数。是否允许数据库集群脑裂自动故障恢复。为了确保数据库集群的数据一致性，默认不支持其自动故障恢复。

取值范围：布尔型。修改后可以 reload 生效。参数修改请参考表 5-9 进行设置。

on、yes、true、1：表示允许 DN 集群脑裂自动故障恢复。

off、no、false、0：表示禁止 DN 集群脑裂自动故障恢复。

默认值：false

5.7.57cms_network_isolation_timeout

参数说明：CM 两节点部署模式必备参数。cm_server 检索 db 集群信息同步异常的最大连续检测次数，超过最大检测次数则认为发生网络分区故障。

取值范围：整型，单位：次。修改后需要重启 cm_server 才能生效。

默认值：20

6 可运行脚本功能说明

以下是 GBase 8s 常用可执行脚本，仅供 GBase 8s 接口使用。

表 6-1 脚本功能说明

脚本名称	功能说明
UpgradeUtility.py	升级工具。
KerberosUtility.py	安装/卸载 kerberos 认证方式。
Uninstall.py	卸载数据库数据库实例。
LocalCheck.py	本地节点检查。
Install.py	安装数据库节点实例。
CheckUninstall.py	检查节点是否已卸载。
PreInstallUtility.py	前置工具。
CleanInstance.py	删除数据库实例。
CleanOsUser.py	删除节点 Osuser。
Resetreplconninfo.py	是用于重置本地 Replconninfo。
LocalPerformanceCheck.py	检查 SSD 信息。
CheckUpgrade.py	升级之前检查节点环境变量。

脚本名称	功能说明
InitInstance.py	初始化数据库。
Backup.py	备份二进制文件和参数文件。
ConfigInstance.py	配置数据库实例。
CheckConfig.py	检查节点配置信息。
ConfigHba.py	配置 pghba.cong 文件等。
ExecuteSql.py	执行数据库 sql 命令。
LocalCheckOS.py	检查本地 OS 信息。
UnPreinstallUtility.py	清除前置过程中的配置。
CheckPreinstall.py	检查节点是否已进行前置。
CheckInstall.py	检查节点安装信息。
GaussStat.py	统计数据库中的节点信息。
CheckPythonVersion.py	检查 python 版本。
expect.sh	自动输入交互密码。
CheckSshAgent.py	负责保护 ssh 互信中的 ssh-agent 进程，互信进程保护工具。

脚本名称	功能说明
LocalCollect.py	收集本地的文件信息及参数信息。
killall	killall 是系统不存在 killall 命令时供 om 使用的自研工具，其替代部分系统 killall 功能。
transfer.py	用于将 C 函数 lib 文件传输到所有节点或备用节点。
Restore.py	恢复二进制文件和参数文件。
StartInstance.py	用于启动数据库。
StopInstance.py	用于停止数据库。
py_pstree.py	用于杀死进程。
install.sh	单节点安装脚本。
one_master_one_slave.sh	一主一备一键安装脚本。
sshkey_encrypt_tool.sh	负责用 ssh-keygen 命令生成秘钥文件（经过密码短语加密）的脚本，互信工具专用。
dms_contrl.sh	资源池化提供给 CM 使用，用于启停 gaussdb（不建议外界单独调用）。
dss_contrl.sh	资源池化提供给 CM 使用，用于启停 dssserver 进程（不建议外界单独调用）。
ssh-agent.sh	负责拉起 ssh-agent 进程的 shell 脚本，互信模块专用。

7 gs_collector 工具支持收集的系统和视图列表

表 7-1 gs_collector 工具支持收集的系统和视图列表

Class	View Name
OS	gs_os_run_info
	pg_os_threads
Session /Thread	gs_session_stat
	gs_session_time
	gs_session_cpu_statistics
	gs_session_memory_statistics
	gs_session_memory_context
	gs_session_memory
	gs_session_memory_detail
	pg_stat_activity
WLM	gs_thread_memory_context
	gs_cluster_resource_info
	pg_total_user_resource_info

Class	View Name
	pg_total_user_resource_info_oid
	gs_wlm_rebuild_user_resource_pool
	gs_wlm_resource_pool
	gs_wlm_workload_records
	pg_wlm_statistics
	pg_app_workloadgroup_mapping
Query-AP Operator	gs_wlm_operator_info
	gs_wlm_operator_history
	gs_wlm_operator_statistics
Query-AP	gs_wlm_session_info_all
	gs_wlm_session_history
	gs_wlm_session_info
	gs_wlm_user_session_info
	gs_wlm_session_statistics
	pg_session_iostat

Class	View Name
	pg_session_wlmstat
Cache I/O	pg_statio_all_indexes
	pg_statio_all_sequences
	pg_statio_all_tables
	pg_statio_sys_indexes
	pg_statio_sys_sequences
	pg_statio_sys_tables
	pg_statio_user_indexes
	pg_statio_user_sequences
	pg_statio_user_tables
	gs_stat_db_cu
	gs_stat_session_cu
Memory	pg_total_memory_detail
	pg_shared_memory_detail
	gs_total_memory_detail

Class	View Name
File	gs_file_stat
	gs_redo_stat
	pg_stat_bad_block
Object	pg_stat_user_tables
	pg_stat_user_indexes
	pg_stat_sys_tables
	pg_stat_sys_indexes
	pg_stat_all_tables
	pg_stat_all_indexes
	pg_stat_database
	pg_stat_database_conflicts
	pg_stat_xact_all_tables
	pg_stat_xact_sys_tables
	pg_stat_xact_user_tables
	pg_stat_xact_user_functions

Class	View Name
Lock	pg_locks
Utility	pg_stat_replication
	pg_replication_slots
	pg_stat_bgwriter
Transaction	pg_running_xacts
	pg_prepared_xacts
Waits	pg_thread_wait_status

8 FAQ

8.1 执行命令报错 “Failed to obtain the GPHOME”

问题现象

如果在 root 用户下执行工具命令时，出现如下信息。

```
Failed to obtain the GPHOME.
```

原因分析

可能是环境变量中 GPHOME 变量没有配置正确。需要检查环境变量 GPHOME 是否包含 XML 中 gaussdbToolPath 路径。

操作步骤

使用如下命令检查 \$GPHOME 路径。

```
echo $GPHOME
```

如果不是安装时默认的路径，请在配置文件中修改 GPHOME 的路径。

```
vim /etc/profile
```

8.2 gs_ctl 重建备实例过程被中断导致秘钥文件不完整恢复方法

问题现象

用户在重建备实例的过程中被中断，再次重建备实例失败，出现以下报错信息。

```
CRC checksum does not match value stored in file, maybe the cipher file is corrupt
non obs cipher file or random parameter file is invalid.
read cipher file or random parameter file failed.
2020-06-18 20:58:12.080 5eeb64e3.1 [unknown] 140697304617088 [unknown] 0
dn_6001_6002 F0000 0 [BACKEND] FATAL: could not load server certificate file
"server.crt": no start line
[2020-06-18 20:58:12.086][24066][dn_6001_6002][gs_ctl]: waitpid 24446 failed,
exitstatus is 256, ret is 2
```

原因分析

重建中断时证书文件不完整，再次重建会由于证书文件不完整而失败。

操作步骤

在数据目录下查看证书文件大小。

```
ll
```

查看秘钥文件大小

```
-rw----- 1 gbase gbase      0 Jun 18 20:58 server.crt
-rw----- 1 gbase gbase      0 Jun 18 20:58 server.key
-rw----- 1 gbase gbase      0 Jun 18 20:58 server.key.cipher
-rw----- 1 gbase gbase      0 Jun 18 20:58 server.key.rand
```

若证书文件大小为 0，删除证书文件。

```
rm -rf server.crt server.key server.key.cipher server.key.rand
```

重建备实例。

```
gs_ctl build -D data_dir
```

说明：若备机数据库已停止，需要重新生成证书文件或者复制证书文件（\$GAUSSHOME/share 下的证书文件）到数据目录，启动备机并重建备实例。生成证书文件的相关操作请参见《GBase 8s V8.8.5_5.0.0_数据库管理指南》。

8.3 使用 `gs_om -t status --all` 查询状态长时间没有响应

问题现象

使用 `gs_om -t status --all` 命令后长时间无响应。

原因分析

可能是 `gaussdb` 进程 hang 住，查询动作会调用 `gsql` 或者 `gs_ctl` 工具查询数据库状态，进程 hang 住后不会给响应，直到超时后退出。

操作步骤

查看 `gsql` 能否访问数据库，出现下面提示说明 `gaussdb` 进程 hang 住，数据库异常。

```
gsql -d postgres -p 29776
gsql: wait (null):29776 timeout expired, errno: Success
```

查看 `postgresql-*.log` 是否有错误提示，根据提示解决问题。

```
cd $GAUSSLOG/pg_log/dn_6001;grep "ERROR|FATAL" postgresql-*.log
```

数据库已经 hang 住，`gs_om` 命令不起作用，可以直接到每个节点上查找进程 `pid` 后 `kill`。

```
ps -ef|grep $GAUSSHOME/bin/gaussdb|grep -v grep
kill -9 $pid
```

所有节点进程 `kill` 完毕后在某一节点执行启动命令。测试环境下可以直接重启数据库，生产商用环境请联系技术支持工程师。

```
gs_om -t start
```

8.4 `gs_sshexkey` 相同用户不同密码报错

问题现象

`openEuler` 环境下，`gs_sshexkey` 支持相同用户不同密码做互信，但是输入了正确的密码还是报鉴权失败。

原因分析

打开系统日志 `/var/log/secure`，查看到有 `pam_faillock(sshd:auth): Consecutive login failures for user` 日志，说明当前用户密码试错次数太多被暂时锁定。

操作步骤

在 `/etc/pam.d` 目录下修改相关配置文件 `system-auth`、`password-auth`、`password-auth-cron`，

将文件中 deny=3 的地方适当放大，待互信全部建立完成后再恢复。

GBASE[®]

南大通用数据技术股份有限公司
General Data Technology Co., Ltd.



微信二维码



■ ■ 技术支持热线：400-013-9696