# GaussDB 轻量化部署形态 24.1.30

# 安装指南

**文档版本** 01

发布日期 2024-04-30





#### 版权所有 © 华为云计算技术有限公司 2024。 保留一切权利。

非经本公司书面许可,任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部,并不得以任何形式传播。

#### 商标声明



HUAWE和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标,由各自的所有人拥有。

#### 注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束,本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定,华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因,本文档内容会不定期进行更新。除非另有约定,本文档仅作为使用指导,本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

# 华为云计算技术有限公司

地址: 贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编: 550029

网址: <a href="https://www.huaweicloud.com/">https://www.huaweicloud.com/</a>

# 前言

# 概述

GaussDB轻量化部署形态包含的组件有GaussDB内核、数据复制软件(Data Replication Software,简称DRS)以及云数据库GaussDB管理平台(TPOPS),简称:TPOPS。

# 读者对象

本手册适用于云数据库GaussDB管理平台(TPOPS)的安装,需要了解以下知识:

- 计算机原理。
- Linux操作系统知识。
- Windows操作系统知识。
- 网络通信知识。

# 符号约定

在本文中可能出现下列标志,它们所代表的含义如下。

符号	说明
▲ 危险	表示如不避免则将会导致死亡或严重伤害的具有高等级风险的危害。
<u></u> 警告	表示如不避免则可能导致死亡或严重伤害的具有中等级风险的危害。
<u></u> 注意	表示如不避免则可能导致轻微或中度伤害的具有低等级风险的危害。
须知	用于传递设备或环境安全警示信息。如不避免则可能会导致设备 损坏、数据丢失、设备性能降低或其它不可预知的结果。 "须知"不涉及人身伤害。
□ 说明	对正文中重点信息的补充说明。 "说明"不是安全警示信息,不涉及人身、设备及环境伤害信 息。

# 目录

前言	ii
1 什么是 GaussDB 轻量化部署形态	1
2 部署方案	2
2.1 管理面部署方案	2
2.1.1 单机房场景	2
2.1.2 同城双机房部署	3
2.1.3 云数据库 GaussDB 管理平台(TPOPS)最佳配置要求	4
2.2 GaussDB 分布式部署方案	4
2.2.1 GaussDB 分布式支持的部署形态	4
2.2.2 同城高可用的部署场景	5
2.2.2.1 同城 2AZ 9 节点 4C4D4 副本+仲裁 AZ	5
2.2.2.2 同城 3AZ 3 节点 3C3D3 副本	6
2.2.2.3 单 AZ 3 节点 3C3D3 副本	7
2.2.2.4 同城 2AZ 5 节点 4C4D4 副本+仲裁 AZ	7
2.2.3 同城高可用+异地容灾的部署场景	8
2.2.3.1 同城 1AZ3 节点+异地 1AZ1 节点	8
2.2.3.2 同城 3AZ3 节点+异地 1AZ1 节点	9
2.2.3.3 同城 1AZ3 节点+异地 1AZ3 节点	10
2.2.3.4 同城 3AZ3 节点+异地 1AZ3 节点	11
2.2.3.5 同城 3AZ5 节点+异地 1AZ4 节点	12
2.2.3.6 同城 3AZ9 节点+异地 1AZ4 节点	13
2.3 GaussDB 主备(集中式)部署方案	14
2.3.1 GaussDB 主备版本支持的部署形态	14
2.3.2 同城高可用的部署场景	15
2.3.2.1 同城 2AZ 5 节点 4 副本+仲裁 AZ	15
2.3.2.2 同城 3AZ 3 节点 3 副本	15
2.3.2.3 单 AZ 3 节点 3 副本	16
2.3.2.4 同城 2AZ 2 节点 1 主 1 备 1 日志副本	17
2.3.3 同城高可用+异地容灾的部署场景	
2.3.3.1 同城 1AZ3 节点+异地 1AZ1 节点	18
2.3.3.2 同城 3AZ3 节点+异地 1AZ1 节点	
2.3.3.3 同城 1AZ3 节点+异地 1AZ3 节点	19

2.3.3.4 同城 3AZ3 节点+异地 1AZ3 节点	20
2.3.3.5 同城 3AZ5 节点+异地 1AZ1 节点	
2.3.3.6 同城 3AZ5 节点+异地 1AZ3 节点	
2.3.3.7 同城 3AZ5 节点(1 主 4 备)+异地 1AZ1 节点	
3 部署流程	
4 安装云数据库 GaussDB 管理平台(TPOPS)	
4.1 安装简介	
4.1.1 服务简介	
4.1.2 部署架构	
4.1.3 系统要求	
4.1.3.1 公共要求	
4.1.3.2 TPOPS 独立部署场景硬件要求	
4.1.3.3 TPOPS 与 DRS 合部场景硬件要求	
4.1.4 账号信息	
4.1.5 安装流程	
4.2 安装前准备	
4.2.1 准备工具、软件包	
4.2.2 设置时钟源	
4.2.2.1 配置说明	
4.2.2.2 前提条件	
4.2.2.3 使用 Chrony 配置时间同步	
4.2.2.4 使用 NTP 配置时间同步	
4.2.3 配置 yum 源	
4.2.4 环境配置	
4.2.5 上传软件包	
4.2.6 修改配置参数	
4.3 安装过程	
4.3.1 前置检查	
4.3.2 安装云数据库 GaussDB 管理平台(TPOPS)	
4.4 安装后检查	
5 安装实例	
5.1 系统要求	
5.2 修改操作系统配置	
5.2.1 使用须知	
5.2.2 配置操作系统防火墙	
5.2.3 关闭 swap 交换内存	
5.2.4 设置字符集参数	
5.2.5 设置时钟源	
5.2.6 设置网卡 MTU 值	
5.2.7 安装 OpenSSH	
5.2.8 检查 Expect	59

5.2.9 配置 sshd_config	59
5.3 准备初始化环境	60
5.3.1 准备磁盘	60
5.3.2 准备磁盘目录	61
5.3.3 网络配置	62
5.3.4 检查 Python 依赖包	63
5.3.5 设置管理程序日志目录(可选)	63
5.3.6 准备浮动 IP(可选)	64
5.4 添加机房	64
5.5 添加主机	65
5.5.1 使用须知	66
5.5.2 添加单个主机	66
5.5.3 批量导入主机	70
5.6 配置 NAS 服务器	73
5.7 安装 GaussDB 实例	74
5.7.1 在本地磁盘安装实例	74
5.7.2 在 Dorado 存储池中安装实例	81
5.8 创建 DBMind 实例	83
6 安装数据复制软件(可选)	86
7 卸载云数据库 GaussDB 管理平台(TPOPS)	87
8 常见问题	91
8.1 云数据库 GaussDB 管理平台(TPOPS)元数据库 GaussDB 安装报错解决方法	
8.1.1 重新安装云数据库 GaussDB 管理平台 ( TPOPS ) 元数据库 GaussDB 的方法	
8.1.2 omagent 安装失败	92
	93
8.3 如何关闭防火墙	94
8.4 如何重新分发安装包	94
8.5 SFTP 安装失败	
8.6 主机管理	95
8.6.1 添加主机失败	96
8.6.2 删除主机失败	98
8.6.3 主机数据盘扩容	98
8.6.4 检查实例安装使用的 AZ 名称	100
8.7 麒麟系统 audit 服务内存泄露占用大量内存	101
A 附录	103
A.1 安装配置文件参考	103
A.2 前置检查报错处理	104
A.3 开源软件列表	106
A.4 安装 JRE	129
A.5 安装 Python3	129

又农田田	<del></del>
A.6 安装主机的 Python3	130
A.7 安装 Expect	131
A.8 安装 NET-TOOLS	131
A.9 磁盘挂载步骤	132
A.10 依赖的 Python 库版本	133
A.11 免密登录管理面节点 root 账户设置	134
A.12 管理面节点之间互信设置	137
A.13 本地 SSD 盘扩容方法	139
A.14 docker-service 目录被删除的卸载步骤	141
A.15 联系技术支持	142

# ◆ 什么是 GaussDB 轻量化部署形态

GaussDB 轻量化部署形态是基于华为云Stack标准全栈方案的轻量化。包含的组件有 GaussDB数据库内核、数据复制软件(Data Replication Software,简称DRS)以及云 数据库GaussDB管理平台(TPOPS),简称:TPOPS。

- GaussDB是华为公司自主研发的新一代企业级分布式关系型数据库,基于Shared-Nothing架构,同时支持X86和ARM计算架构,提供高吞吐强一致性事务处理能力、金融级高可用能力、分布式高扩展能力、大数据高性能查询能力,应用于金融、电信、政府等行业关键核心系统。
- DRS是一种易用、稳定、高效、用于数据库实时迁移和数据库实时同步的云服务,可帮助用户将本地数据轻松迁移至GaussDB。
- 云数据库GaussDB管理平台(TPOPS)是基于HCS DBS(Database Service)服务孵化的一款可即开即用、稳定可靠、便捷管理的数据库运维管理平台。云数据库GaussDB管理平台(TPOPS)不依赖华为云Stack云底座,与华为云Stack标准云部署形成场景互补,可获得与华为云Stack云上一致的用户体验。

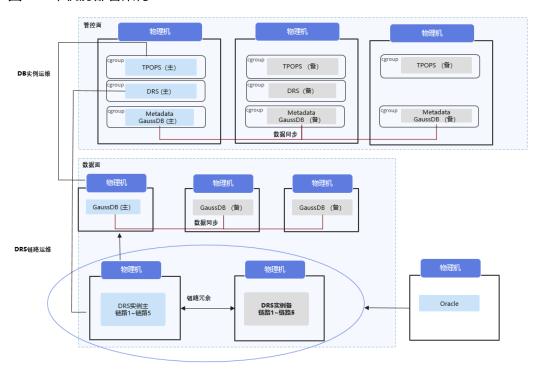
# 2.1 管理面部署方案

# 2.1.1 单机房场景

- 商用场景下部署云数据库GaussDB管理平台(TPOPS)需要3台服务器。
- 在标准典型配置规格下,管理数据库实例规模支持最大500台物理机数据库节点。
- 云数据库GaussDB管理平台(TPOPS)和DRS管理面支持各自单独部署,也支持合并部署。
- 数据库实例支持物理机组网方式。

GaussDB轻量化部署形态的管理面部署架构,如图2-1所示。



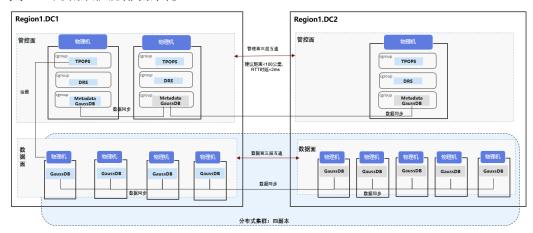


# 2.1.2 同城双机房部署

- 管理节点三台起步(主机房两台,备机房一台)。
- 管理节点两节点故障场景下,GaussDB轻量化管理平台(TPOPS)将无法使用。
- 主机房单机故障,自动倒换。
- 备机房故障,不影响业务使用。
- 云数据库GaussDB管理平台(TPOPS)和DRS管理面支持各自单独部署,也支持合并部署。
- GaussDB轻量化管理平台(TPOPS)中存在SFTP服务两节点部署,SFTP在主节点 故障时,安装包管理上传功能无法使用,查询和下载功能不受影响。

GaussDB轻量化部署形态的管理面部署架构,如图2-2所示。

#### 图 2-2 同城双机房部署架构



#### 须知

如果数据库实例部署的是同城双机房高可用场景,则需要把管理面多数节点部署在实例少数派机房。这样部署后,如果在同城高可用场景下,数据库实例的多数派所在机房故障后,可以通过管理面下发实例少数派AZ强启。

## 2.1.3 云数据库 GaussDB 管理平台(TPOPS)最佳配置要求

当前云数据库GaussDB管理平台与DRS的配置要求分为各自独立部署和合并部署场景,请安装前分别准备根据不同场景参考所需配置要求。

- 各自独立部署场景:云数据库GaussDB管理平台请参见**管理平台独立部署场景**。 DRS部署要求请参见"安装指南 > 安装前准备 > 操作系统配置"章节。
- 合并部署场景:云数据库GaussDB管理平台请参见管理平台与DRS合部场景。

#### 支持的显示规格

云数据库GaussDB管理平台(TPOPS)支持1920x1080、1920x1200及以上分辨率规格(100%缩放)。

# 2.2 GaussDB 分布式部署方案

### 2.2.1 GaussDB 分布式支持的部署形态

部署形态	节点	分片	AZ	说明
同城高可用	9	4	2+1 (仲裁 AZ)	每个AZ 4个节点对称部署(4+4+1), 加上1个仲裁AZ。
	3	3	3	每个AZ下部署一个分片的主DN副本和另外2个分片的备DN副本。
	3	3	1	3节点部署在1个AZ。
	5	4	2+1 (仲裁 AZ)	每个AZ 2个节点对称部署(2+2+1), 加上1个仲裁AZ。
同城高可用 +异地容灾	4	4	2	同城1AZ3节点(3C3D ),异地1AZ1节 点。
	4	4	4	同城3AZ 3节点(3C3D),异地1AZ 1节 点。
	6	6	2	同城1AZ 3节点(3C3D),异地1AZ 3节 点(3C3D)。
	6	6	4	同城3AZ 3节点(3C3D),异地1AZ 3节 点(3C3D)。

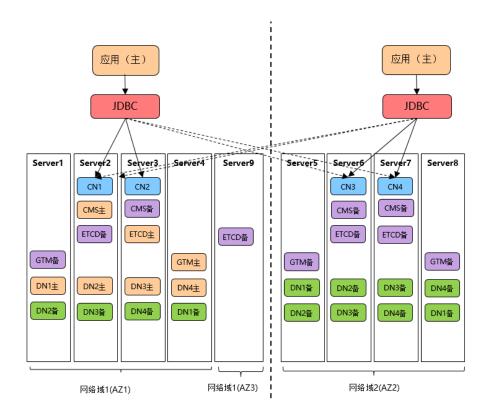
部署形态	节点	分片	AZ	说明
	9	8	3+1 (仲裁 AZ)	同城3AZ 5节点(4C4D),异地1AZ 4节 点(2C4D)。
	13	8	3+1 (仲裁 AZ)	同城3AZ 9节点(4C4D),异地1AZ 4节 点(2C4D)。

# 2.2.2 同城高可用的部署场景

#### 2.2.2.1 同城 2AZ 9 节点 4C4D4 副本+仲裁 AZ

完整的同城双活部署方案,由两个业务AZ和一个仲裁AZ组成。两个业务AZ之间对等部署,任何一个机房都接入业务。仲裁AZ负责辅助仲裁,不能接入业务。可抵抗任意数据库实例单点故障和AZ级故障,及可抵抗机房间网络故障。支持1主3备(4副本)+1仲裁节点的部署方案;

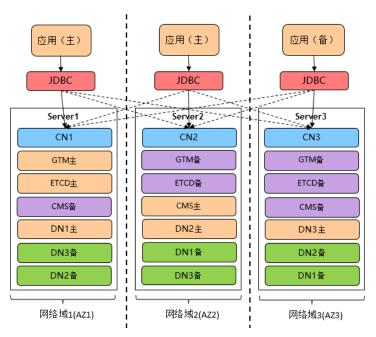
- AZ1、AZ2都有完整数据,AZ3作为仲裁的节点。
- AZ1、AZ2可以同时接入业务,实现双AZ双活。
- AZ3作为仲裁AZ,在1个AZ故障状态下,保证ETCD的存活节点超过多数,从而保障数据库集群仍可正常仲裁。
- DN主备Quorum复制,跨AZ存在同步备,数据不会丢失。
- DN备节点故障,不中断业务的进行;DN主节点故障,自动进行主备切换。
- 可以提供AZ故障的高可用。AZ1、AZ2其中任何一个故障时,故障AZ中所有主服 务会自动切换到另一个AZ,切换完成后业务可以继续运行。
- AZ1、AZ2任意一个AZ+仲裁AZ故障,需要手动启动。



#### 2.2.2.2 同城 3AZ 3 节点 3C3D3 副本

支持同城3AZ高可用场景,3个AZ之间完全对等部署,均可以接入业务。任何机房故障 RPO=0。可抵抗机房之间网络断连问题。

- 1. AZ1、AZ2、AZ3都可以同时接入业务。
- 2. DN主备Quorum复制,跨AZ存在同步备份,数据不会丢失。
- 3. DN备节点故障,不中断业务的进行;DN主节点故障,自动进行主备切换。
- 4. 可以提供AZ故障的高可用。AZ1、AZ2、AZ3其中任何一个故障,故障AZ中所有 主服务会自动切换到另一个AZ,切换完成后业务可以继续运行。

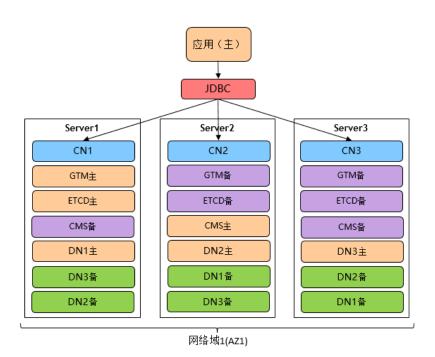


#### 2.2.2.3 单 AZ 3 节点 3C3D3 副本

单AZ 3副本的部署形态,提供了抵御节点级故障的能力,没有AZ级容灾的能力。

适用于不要求机房级别容灾,可抵御数据库组件和物理服务器故障。

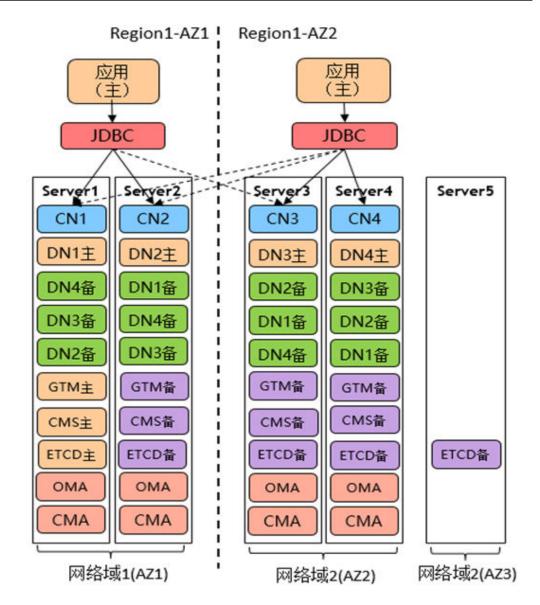
- DN主备间Quorum复制,至少同步到一台备机,保证RPO=0。
- DN备节点故障,不中断业务的进行;DN主节点故障,自动进行主备切换。
- 数据有三份,任何一个节点故障,系统仍然有双份数据确保继续运行。
- 同一数据分片的DN主备实例之间不可部署在同一台服务器上。



### 2.2.2.4 同城 2AZ 5 节点 4C4D4 副本+仲裁 AZ

适用于同城有两个双活数据中心,两个数据中心同时承载业务; 政企、金融等行业业务。适用客户的高性能、高可靠、资源物理隔离、对成本不敏感的业务系统。

- 5节点起步,扩容步长为4;
- 同城是双活数据中心的4副本部署方案,有两个业务AZ和一个仲裁AZ组成,两个数据中心都有主角色。两个业务AZ之间对等部署,任何一个数据中心都可以接入业务。仲裁AZ负责辅助仲裁,不能接入业务。
- 同城可实现节点级、AZ级、数据中心级等故障高可用。AZ1、AZ2、AZ3其中任何一个故障时,故障AZ中所有主服务会自动切换到另一个AZ,AZ1或AZ2故障时在主服务切换完成后业务可以继续运行,AZ3故障时不中断业务。AZ1、AZ2中任意一个AZ+仲裁AZ故障,需要手动强起。
- DN备节点故障,不中断业务的进行; DN主节点故障,自动进行主备切换。
- 当发生节点级、AZ级、数据中心级故障时,可保证数据一致性,RPO=0, RTO<60秒(自动切换)。



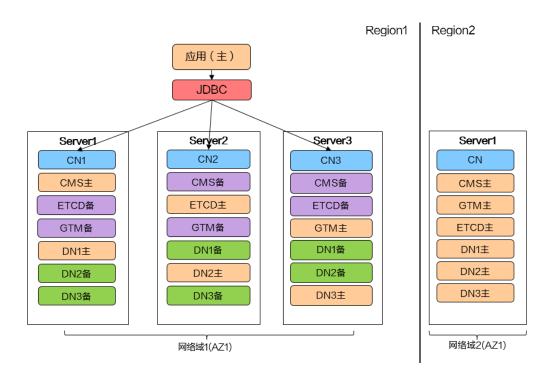
# 2.2.3 同城高可用+异地容灾的部署场景

#### 2.2.3.1 同城 1AZ3 节点+异地 1AZ1 节点

同城一个数据中心和异地一个数据中心的容灾部署形态,同城采用3副本部署,异地采用1副本部署,提供了同城抵御组件级故障的能力,跨城的Region级容灾的能力。

3个副本的可靠性为4个9,AZ的可靠性一般也是4个9,因此对于单AZ场景,即使超过 3个副本数,受到AZ的可靠性限制,整个系统的可靠性也不会有提升。

- 同城和异地都有完整的数据,同城和异地都独立部署一套完整的数据库集群。
- DN主备间Quorum复制,至少同步到一台备机,保证RPO=0。
- DN备节点故障,不中断业务的进行;DN主节点故障,自动进行主备切换。
- 同城数据有三份,任何一个节点故障,系统仍然有双份数据确保继续运行。任何 一个备份都可以升主。
- 跨Region容灾需要手工切换。

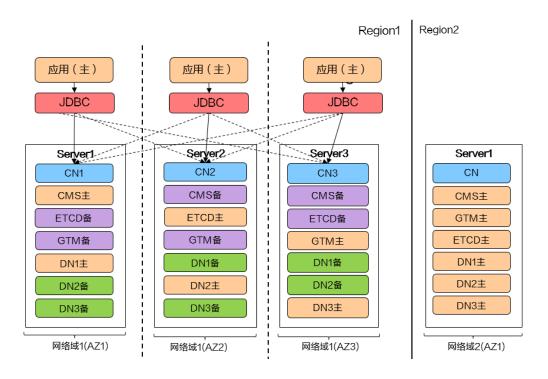


#### 2.2.3.2 同城 3AZ3 节点+异地 1AZ1 节点

同城一个数据中心和异地一个数据中心的容灾部署形态,同城采用3副本部署,异地采用1副本部署,提供了同城抵御组件级故障的能力,跨城的Region级容灾的能力。

3个副本的可靠性为4个9,AZ的可靠性一般也是4个9,因此对于单AZ场景,即使超过 3个副本数,受到AZ的可靠性限制,整个系统的可靠性也不会有提升。

- 同城和异地都有完整的数据,同城和异地都独立部署一套完整的数据库集群。
- DN主备间Quorum复制,至少同步到一台备机,保证RPO=0。
- DN备节点故障,不中断业务的进行; DN主节点故障,自动进行主备切换。
- 同城数据有三份,任何一个节点故障,系统仍然有双份数据确保继续运行。任何 一个备份都可以升主。
- 跨Region容灾需要手工切换。

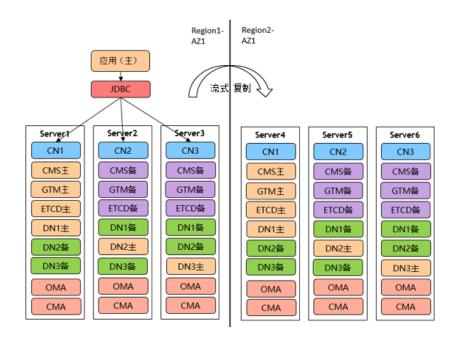


#### 2.2.3.3 同城 1AZ3 节点+异地 1AZ3 节点

同城一个数据中心和异地一个数据中心的容灾部署形态,同城采用3副本部署,异地也采用3副本部署,提供了同城抵御组件级故障的能力,跨城的Region级容灾的能力。

3个副本的可靠性为4个9,AZ的可靠性一般也是4个9,因此对于单AZ场景,即使超过 3个副本数,受到AZ的可靠性限制,整个系统的可靠性也不会有提升。

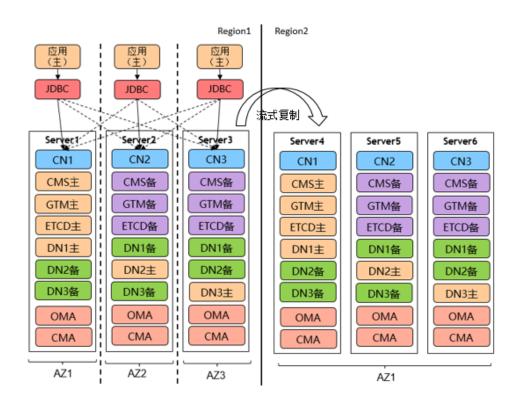
- 同城和异地都有完整的数据,同城和异地都独立部署一套完整的数据库集群。
- DN主备间Quorum复制,至少同步到一台备机,保证RPO=0。
- DN备节点故障,不中断业务的进行;DN主节点故障,自动进行主备切换。
- 数据有三份,任何一个节点故障,系统仍然有双份数据确保继续运行。任何一个 备份都可以升主。
- 跨Region容灾需要手工切换。



#### 2.2.3.4 同城 3AZ3 节点+异地 1AZ3 节点

同城三个数据中心和异地一个数据中心的容灾部署形态,同城采用3副本部署,异地也采用3副本部署,提供了同城抵御组件级故障和跨AZ级故障的能力,跨城的Region级容灾的能力。

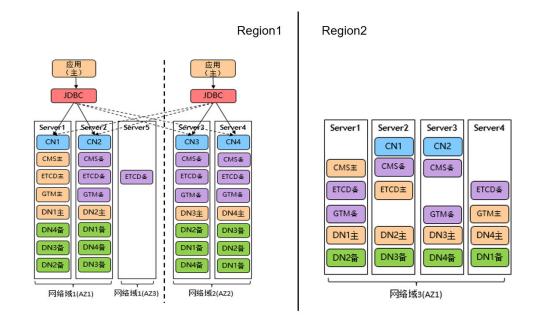
- 同城和异地都有完整的数据,同城和异地都独立部署一套完整的数据库集群。
- DN主备间Quorum复制,至少同步到一台备机,保证RPO=0。
- DN备节点故障,不中断业务的进行;DN主节点故障,自动进行主备切换。
- 数据有三份,任何一个节点故障,系统仍然有双份数据确保继续运行。任何一个 备份都可以升主。
- 同城可以提供AZ级的故障的高可用。AZ1、AZ2、AZ3其中任何一个故障,故障 AZ中所有主服务会自动切换到另一个AZ,切换完成后业务可以继续运行。
- 跨Region容灾需要手工切换。



#### 2.2.3.5 同城 3AZ5 节点+异地 1AZ4 节点

同城三个数据中心和异地一个数据中心的容灾部署形态,同城采用4副本部署,异地采用2副本部署,提供了同城抵御实例级故障和跨AZ级故障的能力,跨城的Region级容灾的能力。

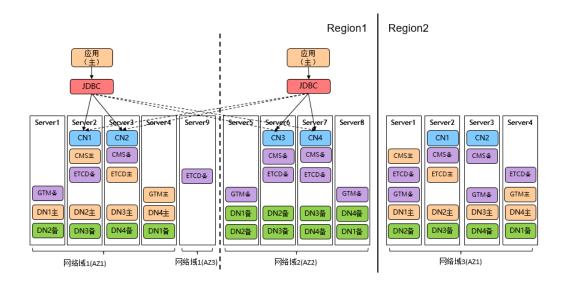
- 同城和异地都有完整的数据,同城和异地都独立部署一套完整的数据库集群。
- DN主备之间采用流复制进行数据同步,至少同步到一台备机,保证RPO=0。
- DN备节点故障,不中断业务的进行; DN主节点故障,自动进行主备切换。
- 同城中数据有四份,任何一个节点故障,系统仍然有三份数据确保继续运行。任何一个备份都可以升主。
- 同城可以提供机房级的故障的高可用。AZ1、AZ2、AZ3其中任何一个故障,故障 AZ中所有主服务会自动切换到另一个AZ,切换完成后业务可以继续运行。
- 跨Region容灾需要手工切换。



#### 2.2.3.6 同城 3AZ9 节点+异地 1AZ4 节点

同城两个数据中心和异地一个数据中心的容灾部署形态,同城采用4副本部署,异地采用相同分片数2副本部署。同城双活部署方案,由两个业务AZ和一个仲裁AZ组成。两个业务AZ之间对等部署,任何一个机房都接入业务;仲裁AZ负责辅助仲裁,不能接入业务;可抗任意单点故障;任何机房故障RPO=0;可抗机房之间网络断连;支持2AZ1主3备(4副本)+1仲裁AZ的部署方案。异地容灾提供跨Region级容灾的能力。

- 同城和异地都有完整的数据,同城和异地都独立部署一套完整的数据库集群。
- 同城中,AZ1、AZ2都有完整数据,AZ3作为第三方仲裁的节点;AZ1、AZ2可以同时接入业务,实现双AZ双活;AZ3作为仲裁AZ,在1个AZ故障状态下,保证ETCD的存活节点超过多数,从而保证数据的一致性。
- DN主备之间采用流复制进行数据同步,至少同步到两台备机,保证RPO=0。
- DN备节点故障,不中断业务的进行; DN主节点故障,自动进行主备切换。
- 同城中数据有四份,任何一个节点故障,系统仍然有三份数据确保继续运行。任何一个备份都可以升主。
- 同城可以提供机房级的故障的高可用。AZ1、AZ2、AZ3其中任何一个故障,故障 AZ中所有主服务会自动切换到另一个AZ,切换完成后业务可以继续运行;AZ1、 AZ2任意一个AZ+仲裁AZ故障,需要手动强起。
- 跨Region容灾需要手工切换。



# 2.3 GaussDB 主备 (集中式)部署方案

# 2.3.1 GaussDB 主备版本支持的部署形态

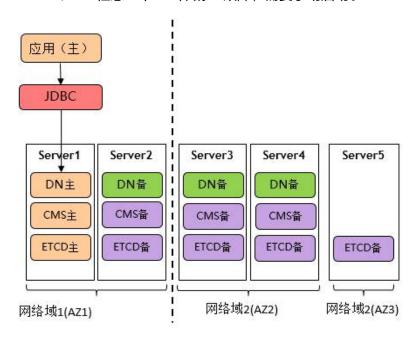
部署形态	节 点	副本	AZ	说明
同城高 可用	5	4	2+1(仲裁 AZ)	每个AZ 2个节点对称部署 (2+2+1)。
	3	3	1	3节点部署在1个AZ。
	3	3	3	每个AZ 1个节点。
	2	2+1(日 志副本)	2	1个节点部署1DN副本和1日志副本,另外 个1节点部署1DN副本。
同城高	4	4	2	同城1AZ,3个节点;异地1AZ,1节点。
可用   +异地	4	4	4	同城3AZ,3个节点;异地1AZ,1节点。
容灾	6	5	4	同城3AZ,5个节点;异地1AZ,1节点。
	6	6 (1主4 备)	4	同城3AZ,5个节点;异地1AZ,1节点。
	6	6	2	同城1AZ,3个节点;异地1AZ,3节点。。
	6	6	4	同城3AZ,每个AZ有1个节点;异地1AZ, 3节点。
	8	7	3+1(仲裁 AZ)	同城3AZ,5节点,异地1AZ,3节点。

## 2.3.2 同城高可用的部署场景

#### 2.3.2.1 同城 2AZ 5 节点 4 副本+仲裁 AZ

由两个业务AZ和一个仲裁AZ组成;可抗任意单点故障,任何AZ机房故障RPO=0。可抗AZ机房之间网络断连。支持2AZ 1主3备(4副本)+1仲裁AZ的部署方案。

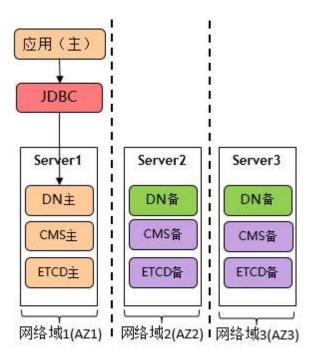
- AZ1、AZ2都有完整数据,AZ3作为第三方仲裁的节点。
- AZ3作为仲裁AZ,在1个AZ故障状态下,保证ETCD的存活节点超过多数,从而保障数据库集群仍可正常仲裁。
- DN主备复制支持Quorum和Paxos两种协议,跨AZ存在同步备,数据不会丢失。
- DN备节点故障,不中断业务的进行; DN主节点故障,自动进行主备切换。
- 可以提供机房级的故障的高可用。AZ1、AZ2其中任何一个故障,故障AZ中所有 主服务会自动切换到另一个AZ,切换完成后业务可以继续运行。
- AZ1、AZ2任意一个AZ+仲裁AZ故障,需要手动启动。



#### 2.3.2.2 同城 3AZ 3 节点 3 副本

支持同城3AZ高可用场景,3个AZ之间完全对等部署,均可以接入业务。任何机房故障RPO=0。可抗机房之间网络断连。

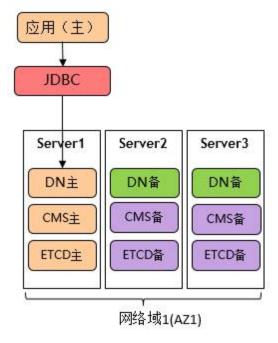
- 1. DN主备复制支持Quorum和Paxos两种协议,跨AZ存在同步备份,数据不会丢失。
- 2. DN备节点故障,不中断业务的进行;DN主节点故障,自动进行主备切换。
- 3. 可以提供机房级的故障的高可用。AZ1、AZ2、AZ3其中任何一个故障,故障AZ中所有主服务会自动切换到另一个AZ,切换完成后业务可以继续运行。
- 4. AZ1、AZ2、AZ3任意两个AZ故障,需要手动强起。



#### 2.3.2.3 单 AZ 3 节点 3 副本

单AZ3副本的部署形态,提供了抵御实例级故障的能力,没有AZ级容灾的能力。适用于不要求机房级别容灾,但是需要抵御个别服务器故障的应用场景。

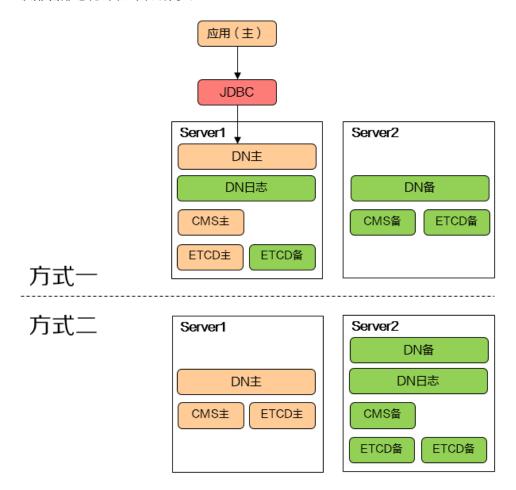
- DN主备间复制支持Quorum和Paxos两种协议,至少同步到一台备机,保证RPO=0。
- DN备节点故障,不中断业务的进行;DN主节点故障,自动进行主备切换。
- 数据有三份,任何一个节点故障,系统仍然有双份数据确保继续运行。



#### 2.3.2.4 同城 2AZ 2 节点 1 主 1 备 1 日志副本

- 在两台物理机上安装:
  - 一主一备一日志、两个CMS、三个ETCD。
- 有两份完整的副本,一个日志副本:
  - 日志副本主要用于节省存储资源和计算资源。
  - 整个数据库高可用统一采用多数派协议,实现软件级的高可用(DN、CMS、ETCD)。
  - 当DN少数派所在机器故障后,不影响服务。
  - 当DN多数派所在机器故障后,高可用优先场景集群自动降副本强启恢复业务 (可能导致数据丢失),高可靠场景服务停止,需要人工介入强启。
  - 不提供备机读。
  - 两节点形态相较于三节点形态可靠性下降,存在单节点故障后数据库变只读的风险。
- 整体服务的SLA
  - a. 强一致下因需要人工介入,不承诺整体服务的SLA。
  - b. 可以实现软件的高可用(DN、CMS、ETCD),即软件SLA为99.95%。

该部署形态方案如下如所示。

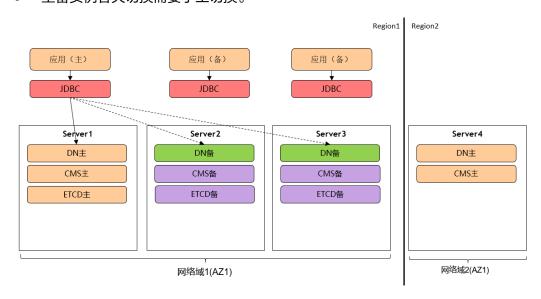


# 2.3.3 同城高可用+异地容灾的部署场景

#### 2.3.3.1 同城 1AZ3 节点+异地 1AZ1 节点

同城一个数据中心和异地一个数据中心的容灾部署形态,同城采用1主2备部署,异地采用单节点部署,提供了同城抵御组件级和节点级故障的能力,跨城的Region级容灾的能力。同城1AZ,异地1AZ。

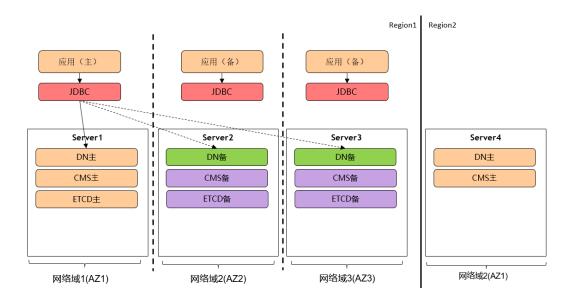
- 同城和异地都独立部署一套完整的数据库集群。
- 同城DN主备间Quorum复制,至少同步到一台备机,保证RPO=0。
- DN备节点故障,不中断业务; DN主节点故障,自动进行主备切换。
- 任何一个节点故障,系统仍然有双份数据确保继续运行。
- 主备实例容灾切换需要手工切换。



#### 2.3.3.2 同城 3AZ3 节点+异地 1AZ1 节点

同城一个数据中心和异地一个数据中心的容灾部署形态,同城采用1主2备部署,异地采用单节点部署,提供了同城抵御组件级和节点级故障的能力,跨城的Region级容灾的能力。同城3AZ,异地1AZ。

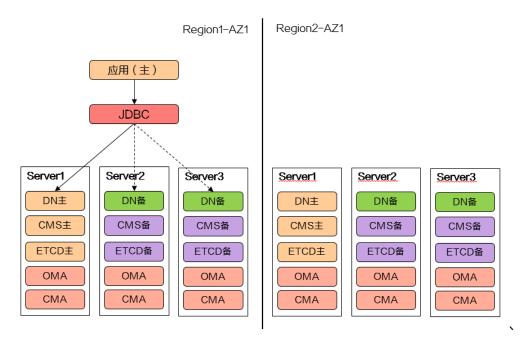
- 同城和异地都独立部署一套完整的数据库集群。
- 同城DN主备间Quorum复制,至少同步到一台备机,保证RPO=0。
- DN备节点故障,不中断业务;DN主节点故障,自动进行主备切换。
- 任何一个节点故障,系统仍然有双份数据确保继续运行。
- 主备实例容灾切换需要手工切换。



#### 2.3.3.3 同城 1AZ3 节点+异地 1AZ3 节点

同城一个数据中心和异地一个数据中心的容灾部署形态,同城采用1主2备部署,异地也采用1主2备部署,提供了同城抵御组件级和节点级故障的能力,跨城的Region级容灾的能力。

- 同城和异地都独立部署一套完整的数据库集群。
- 同城DN主备间Quorum复制,至少同步到一台备机,保证RPO=0。
- DN备节点故障,不中断业务;DN主节点故障,自动进行主备切换。
- 任何一个节点故障,系统仍然有双份数据确保继续运行。
- 主备实例容灾切换需要手工切换。

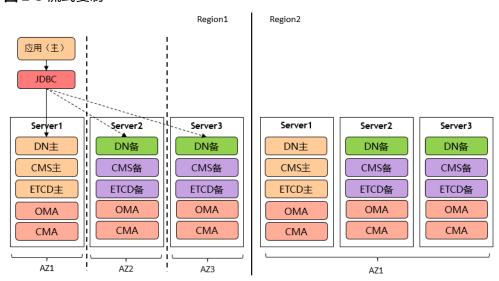


#### 2.3.3.4 同城 3AZ3 节点+异地 1AZ3 节点

同城三个数据中心和异地一个数据中心的容灾部署形态,同城采用1主2备部署,异地也采用1主2备部署,提供了同城抵御实例级故障和跨AZ级故障的能力,跨城的Region级容灾的能力。

- 同城和异地都独立部署一套完整的数据库集群。
- 同城DN主备间Quorum复制,至少同步到一台备机,保证RPO=0。
- DN备节点故障,不中断业务的进行;DN主节点故障,自动进行主备切换。
- 数据有三份,任何一个节点故障,系统仍然有双份数据确保继续运行。
- 同城可以提供机房级的故障的高可用。AZ1、AZ2、AZ3其中任何一个故障,故障 AZ中所有主服务会自动切换到另一个AZ,切换完成后业务可以继续运行。
- 主备实例容灾切换需要手丁切换。

#### 图 2-3 流式复制

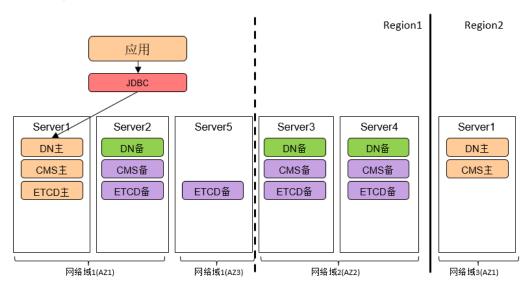


#### 2.3.3.5 同城 3AZ5 节点+异地 1AZ1 节点

同城两个数据中心和异地一个数据中心的容灾部署形态,同城采用4副本部署,异地采用单节点部署。同城双活部署方案,由两个业务AZ和一个仲裁AZ组成。两个业务AZ之间对等部署,任何一个机房都接入业务;仲裁AZ负责辅助仲裁,不能接入业务;可抗任意单点故障;任何机房故障RPO=0;可抗机房之间网络断连;支持2AZ1主3备(4副本)+1仲裁AZ的部署方案。异地容灾提供跨Region级容灾的能力。

- 同城和异地都有完整的数据,同城和异地都独立部署一套完整的数据库集群。
- 同城中,AZ1、AZ2都有完整数据,AZ3作为第三方仲裁的节点;AZ1、AZ2可以同时接入业务,实现双AZ双活;AZ3作为仲裁AZ,在1个AZ故障状态下,保证ETCD的存活节点超过多数,从而保证数据的一致性。
- DN主备之间采用流复制进行数据同步,至少同步到两台备机,保证RPO=0。
- DN备节点故障,不中断业务的进行;DN主节点故障,自动进行主备切换。
- 同城中数据有四份,任何一个节点故障,系统仍然有三份数据确保继续运行。任何一个备份都可以升主。

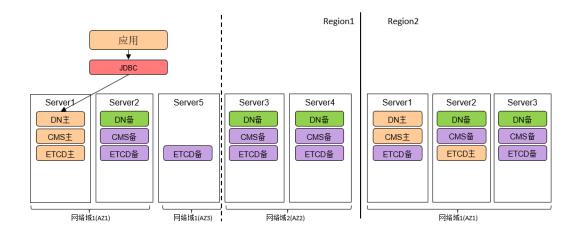
- 同城可以提供机房级的故障的高可用。AZ1、AZ2、AZ3其中任何一个故障,故障 AZ中所有主服务会自动切换到另一个AZ,切换完成后业务可以继续运行;AZ1、 AZ2任意一个AZ+仲裁AZ故障,需要手动强起。
- 跨Region容灾需要手工切换。



#### 2.3.3.6 同城 3AZ5 节点+异地 1AZ3 节点

同城两个数据中心和异地一个数据中心的容灾部署形态,同城采用4副本部署,异地采用1主2备部署。同城双活部署方案,由两个业务AZ和一个仲裁AZ组成。两个业务AZ之间对等部署,任何一个机房都接入业务;仲裁AZ负责辅助仲裁,不能接入业务;可抗任意单点故障;任何机房故障RPO=0;可抗机房之间网络断连;支持2AZ1主3备(4副本)+1仲裁AZ的部署方案。异地容灾提供跨Region级容灾的能力。

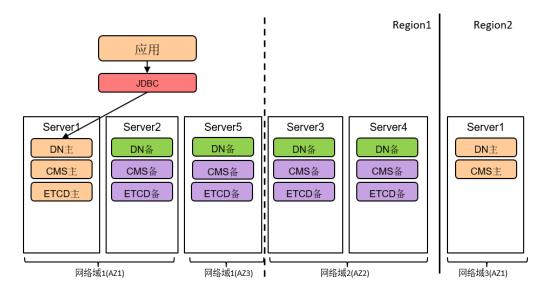
- 同城和异地都有完整的数据,同城和异地都独立部署一套完整的数据库集群。
- 同城中,AZ1、AZ2都有完整数据,AZ3作为第三方仲裁的节点;AZ1、AZ2可以同时接入业务,实现双AZ双活;AZ3作为仲裁AZ,在1个AZ故障状态下,保证ETCD的存活节点超过多数,从而保证数据的一致性。
- DN主备之间采用流复制进行数据同步,至少同步到两台备机,保证RPO=0。
- DN备节点故障,不中断业务的进行; DN主节点故障,自动进行主备切换。
- 同城中数据有四份,任何一个节点故障,系统仍然有三份数据确保继续运行。任何一个备份都可以升主。
- 同城可以提供机房级的故障的高可用。AZ1、AZ2、AZ3其中任何一个故障,故障 AZ中所有主服务会自动切换到另一个AZ,切换完成后业务可以继续运行;AZ1、 AZ2任意一个AZ+仲裁AZ故障,需要手动强起。
- 跨Region容灾需要手工切换。



#### 2.3.3.7 同城 3AZ5 节点 (1 主 4 备)+异地 1AZ1 节点

同城两个数据中心和异地一个数据中心的容灾部署形态,同城采用5副本部署,异地采用单节点部署。同城双活部署方案,由三个业务AZ组成。可抗任意单点故障;任何机房故障RPO=0;可抗机房之间网络断连;

- 同城和异地都有完整的数据,同城和异地都独立部署一套完整的数据库集群。
- 同城中,AZ1、AZ2都有完整数据;AZ1、AZ2、AZ3可以同时接入业务。
- DN主备之间采用流复制进行数据同步,至少同步到两台备机,保证RPO=0。
- DN备节点故障,不中断业务的进行;DN主节点故障,自动进行主备切换。
- 同城中数据有5份,任何一个节点故障,系统仍然有4份数据确保继续运行。任何 一个备份都可以升主。
- 同城可以提供机房级的故障的高可用。AZ1、AZ2、AZ3其中任何一个故障,故障 AZ中所有主服务会自动切换到另一个AZ,切换完成后业务可以继续运行;多数AZ 故障,需要手动强起。
- 跨Region容灾需要手工切换。



# **3** 部署流程

GaussDB 轻量化部署形态部署流程如下图所示。

图 3-1 部署流程

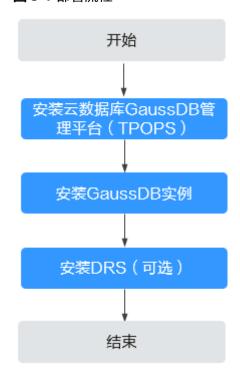


表 3-1 GaussDB 轻量化部署形态部署流程

编号	流程	参考资料
1	确定GaussDB部署方案	GaussDB 分布式部署方案或GaussDB 主备(集中式)部署方案
2	安装云数据库GaussDB管理平台(TPOPS)	安装云数据库GaussDB管理平台 (TPOPS)
3	安装GaussDB实例	安装实例

编号	流程	参考资料
4	安装DRS(可选)	安装数据复制软件(可选)

# 4 安装云数据库 GaussDB 管理平台 (TPOPS)

# 4.1 安装简介

#### 4.1.1 服务简介

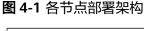
云数据库GaussDB管理平台(TPOPS),简称: TPOPS,是基于HCS DBS(Database Service)服务孵化的一款可即开即用、稳定可靠、便捷管理的数据库运维管理平台。云数据库GaussDB管理平台(TPOPS)不依赖华为云Stack云底座,与华为云Stack标准云部署形成场景互补,可获得与华为云Stack云上一致的用户体验。

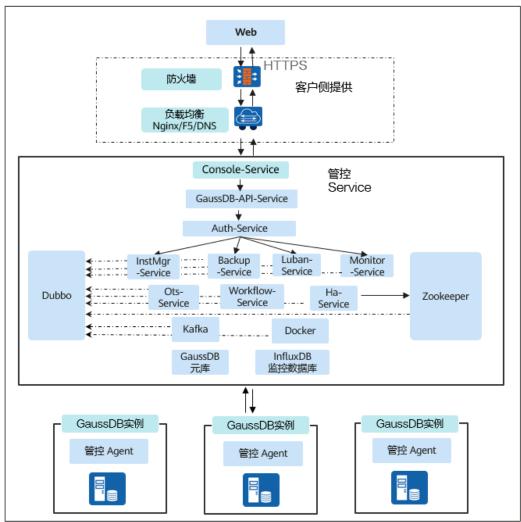
云数据库GaussDB管理平台(TPOPS)当前仅支持HA模式部署,即在三个节点上安装 云数据库GaussDB管理平台(TPOPS)。

HA模式具有高可靠模式;支持跨机房、跨地域部署;可有效避免微服务单点故障、单机故障、单机房故障等异常情况,保证业务稳定运行。

# 4.1.2 部署架构

云数据库GaussDB管理平台(TPOPS)基于B/S架构开发,由Web、管控Service、管控Agent三部分组成,软件结构如<mark>图4-1</mark>所示。





- Web:作为用户接入子系统,用于将用户在Web下发的操作指令通过管控Service 下发到数据库实例,同时也可以将管控的Console-Service服务分析处理后的数据 通过Web交互模块传递给Web用户界面向客户展示。
- 管控Service:管控Service是管控的分析控制子系统,包含Web交互模块、实例业务模块、组件业务模块、数据业务模块等微服务模块。向上通过Web交互模块对Web展示实例的监控数据,向下通过组件业务模块对管控Agent下发操作指令。管控Agent收集的信息存储在管控Service的监控数据库和元数据库中,由管控Service的实例业务模块和数据业务模块进行分析处理。
- 微服务模块有:

表 4-1 微服务说明

组件名	说明
DBS-docker-service	安装工具包,无状态服务。
DBS-platform-data	云数据库GaussDB管理平台(TPOPS)元数据库基础 SQL。

组件名	说明
DBS-GaussDB- feature-data	云数据库GaussDB管理平台(TPOPS)元数据库基础 SQL。
DBS-GaussDB-data	云数据库GaussDB管理平台(TPOPS)元数据库基础 SQL。
Docker	用于构建、部署和运行微服务应用程序的容器化平台。
InfluxDB	云数据库GaussDB管理平台(TPOPS)的时序数据库, 用于存放监控、告警数据。
SFTP	文件服务器。
GaussDB	云数据库GaussDB管理平台(TPOPS)的元数据库。
DBS-zookeeper	注册中心。
DBS-kafka	消息中间件。
DBS-monitor- service	监控指标(告警、指标、TOP SQL)。
DBS-rds-ha-admin	实例监控服务(更新实例状态、节点角色)。
DBS-resource- manager	资源管理。
DBS-workflow	任务流、实例创建、实例删除。
DBS-common- service	公共服务,主要提供给实例参数组使用。
DBS-auth	用户鉴权。
DBS-ots	操作审计。
DBS-luban	监控运维管理。
DBS-GaussDB- open-api	云数据库GaussDB管理平台(TPOPS)open-api微服务的API入口。
DBS-GaussDB- instancemanager	实例管理服务。
DBS-GaussDB- backupmanager	备份管理服务。
DBS-gaussdb- console	前端控制台。

管控Agent:用于收集实例、主机、组件等运行数据,上报给管控Service进行分析处理。同时也根据管控Service下发的指令在实例上进行相应的操作。实例各节点上均部署一个管控Agent,所有管控Agent共同组成云数据库GaussDB管理平台(TPOPS)管控的实例监控子系统。

# 4.1.3 系统要求

#### 4.1.3.1 公共要求

#### 操作系统要求

表 4-2 支持的操作系统及相应安装包

操作系统	支持的版本	架构
银河麒麟 V10	SP1: 4.19.90-23.8.v2101.ky10.a arch64 SP2: 4.19.90-24.4.v2101.ky10.a arch64	ARM
	SP1: 4.19.90-23.8.v2101.ky10.x 86_64 SP2: 4.19.90-24.4.v2101.ky10.x 86_64	X86
统信	4.19.90-2201.4.0.0135.up 1.uel20.x86_64	X86
	4.19.90-2201.4.0.0135.up 1.uel20.aarch64	ARM

#### 软件要求

表 4-3 云数据库 GaussDB 管理平台 (TPOPS) 软件要求

软件	规格	
浏览器	版本要求:	
	• Google Chrome 120,119,118	
	• Firefox 121,120	
	● Microsoft Edge: 随Windows 10更新	
JRE	云数据库GaussDB管理平台(TPOPS)支持Open JDK中的 JRE:	
	Open JDK: 1.8.0_272	
	若有版本漏洞,请及时修复。推荐使用麒麟镜像自带的Open JDK。	

软件	规格
Python	Python 3.7.4或3.7.9(操作系统自带Python版本即可,支持的操作系统一般自带python版本为3.7.4或者3.7.9)。可执行如下指令,检查Python版本。 python3version
	如果查询到的版本不是Python3.7.4或Python3.7.9,需要参考 <b>安装Python3</b> 安装Python3.7.4或Python3.7.9版本。

#### 纳管节点规格

表 4-4 实例纳管规格

服务器规格	最多可纳管节点数	
8U64G	60个节点	
>=16U128G	500个节点	

#### 4.1.3.2 TPOPS 独立部署场景硬件要求

下表列出了云数据库GaussDB管理平台(TPOPS)与DRS各自单独部署时,云数据库GaussDB管理平台(TPOPS)的硬件要求。

表 4-5 节点硬件要求

硬件	配置	说明
СРИ	8核处理器及以上	云数据库GaussDB管理平台(TPOPS)所 在服务器的CPU规格要求。
可用内存	64GB及以上	云数据库GaussDB管理平台(TPOPS)所 在服务器的可用内存。
磁盘	1150GB及以上	云数据库GaussDB管理平台(TPOPS)所在服务器的可用磁盘空间。 要求使用SSD(SATA SSD或SAS SSD均可)。 说明 所需磁盘空间,与数据需保存的时间有关。

表 4-6 磁盘空间划分要求

硬件	配置	文件系统 格式	说明
安装工具包磁盘 空间 默认解压路径: / data 总计: 50G	50G	建议Ext4	云数据库GaussDB管理平台安装工具包及各软件包上传安装所需磁盘空间。 默认路径:/data
主程序磁盘空间 默认挂盘路 径:/opt/cloud 总计:200GB	200GB	建议Ext4	云数据库GaussDB管理平台 (TPOPS)主程序安装所需磁盘空间。 默认路径:/opt/cloud 说明 该路径是公共路径,这里仅给出主程序 安装需要的磁盘空间,不含其它如日 志、数据库等。
公共日志磁盘空间 默认挂盘路 径:/opt/cloud/logs 总计:100GB	100GB(各 微服务共 用)	建议Ext4	云数据库GaussDB管理平台 (TPOPS)各服务产生日志所需磁盘 空间。 默认路径:/opt/cloud/logs/
GaussDB数据库 磁盘空间 默认挂盘路 径:/opt/ gaussdb 总计: 200GB	200GB	建议Ext4	云数据库GaussDB管理平台 (TPOPS)主数据库安装部署、以及 存储所需磁盘空间。 默认路径:/opt/gaussdb
SFTP 默认挂盘路 径: /opt/ sftphome 总计: 100GB	100GB	建议Ext4	云数据库GaussDB管理平台 (TPOPS)SFTP安装部署、以及存 储所需磁盘空间。 默认路径:/opt/sftphome
备份磁盘空间 默认挂盘路 径:/opt/backup 总计:200GB	200GB	建议Ext4	云数据库GaussDB管理平台 (TPOPS)备份存储所需磁盘空间。 默认路径:/opt/backup
Docker磁盘空间 默认挂盘路 径:/opt/docker	200GB	建议Ext4	云数据库GaussDB管理平台 (TPOPS)容器运行服务平台 Docker数据存储所需磁盘空间。 默认路径:/opt/docker

硬件	配置	文件系统 格式	说明
时序数据库 InfluxDB磁盘空 间 默认路径:/opt/ influxdb	100GB	建议Ext4	云数据库GaussDB管理平台 (TPOPS)存储监控告警等数据存储 所需磁盘空间。 默认路径:/opt/influxdb

### □ 说明

- /opt/cloud, /opt/cloud/logs, /opt/gaussdb, /opt/sftphome, /opt/backup, /opt/docker, /opt/influxdb目录路径为安装云数据库GaussDB管理平台(TPOPS)专用路径,不与其它文件共享磁盘空间,卸载清理云数据库GaussDB管理平台(TPOPS)时会清理这些目录路径下的所有文件。
- 建议将每个部分单独一块磁盘进行挂载,在分盘上对磁盘空间进行隔离。
- 可通过**df**-h命令查看文件系统磁盘使用情况。若不满足磁盘划分要求,建议参考**磁盘挂载步** 骤进行磁盘挂载。
- 表4-6中磁盘路径仅支持字母、数字和下划线。
- 最多支持纳管实例数,请参见**纳管节点规格**。
- /data目录建议独立挂盘,至少有50GB的剩余容量,用于存放安装工具包和待安装软件包。

## 4.1.3.3 TPOPS 与 DRS 合部场景硬件要求

下表列出了云数据库GaussDB管理平台(TPOPS)与DRS合并部署时,云数据库GaussDB管理平台(TPOPS)和DRS的硬件要求。

表 4-7 节点硬件要求

硬件	配置	说明
CPU	96核处理器	云数据库GaussDB管理平台(TPOPS)与 DRS合部所在服务器的CPU规格要求。
可用内存	576GB及以上	云数据库GaussDB管理平台(TPOPS)与 DRS合部所在服务器的可用内存。
磁盘	1150G + DRS- Service所需磁盘空间 + DRS-Gaussdb元数 据库所需磁盘空间	云数据库GaussDB管理平台(TPOPS)与DRS合部所在服务器的可用磁盘空间。DRS-Service所需磁盘空间以及DRS-Gaussdb元数据库所需磁盘大小详情请参考"LLD模板 > 磁盘挂载点信息"页签。要求使用SSD(SATA SSD或SAS SSD均可)。 说明所需磁盘空间,与数据需保存的时间有关。

表 4-8 磁盘空间划分要求

硬件	配置	文件系统 格式	说明
安装工具包磁盘 空间 默认解压路径: / data 总计: 50G	50G	建议Ext4	云数据库GaussDB管理平台安装工具包及各软件包上传安装所需磁盘空间。 默认路径:/data
主程序磁盘空间 默认挂盘路 径:/opt/cloud 总计:200GB	200GB	建议Ext4	云数据库GaussDB管理平台 (TPOPS)主程序安装所需磁盘空间。 默认路径:/opt/cloud 说明 该路径是公共路径,这里仅给出主程序 安装需要的磁盘空间,不含其它如日 志、数据库等。
公共日志磁盘空间 默认挂盘路 径:/opt/cloud/logs 总计:100GB	100GB ( 各 微服务共 用 )	建议Ext4	云数据库GaussDB管理平台 (TPOPS)各服务产生日志所需磁盘 空间。 默认路径:/opt/cloud/logs/
GaussDB数据库 磁盘空间 默认挂盘路 径:/opt/ gaussdb 总计: 200GB	200GB	建议Ext4	云数据库GaussDB管理平台 (TPOPS)主数据库安装部署、以及 存储所需磁盘空间。 默认路径:/opt/gaussdb
SFTP 默认挂盘路 径: /opt/ sftphome 总计: 100GB	100GB	建议Ext4	云数据库GaussDB管理平台 (TPOPS)SFTP安装部署、以及存 储所需磁盘空间。 默认路径:/opt/sftphome
备份磁盘空间 默认挂盘路 径:/opt/backup 总计:200GB	200GB	建议Ext4	云数据库GaussDB管理平台 (TPOPS)备份存储所需磁盘空间。 默认路径:/opt/backup
Docker磁盘空间 默认挂盘路 径:/opt/docker	200GB	建议Ext4	云数据库GaussDB管理平台 (TPOPS)容器运行服务平台 Docker数据存储所需磁盘空间。 默认路径:/opt/docker

硬件	配置	文件系统 格式	说明
时序数据库 InfluxDB磁盘空 间 默认路径:/opt/ influxdb	100GB	建议Ext4	云数据库GaussDB管理平台 (TPOPS)存储监控告警等数据存储 所需磁盘空间。 默认路径:/opt/influxdb
DRS-Service数据 盘 默认路 径:/opt/drs	数据盘大小 要求请参见 "LLD模板 > 磁盘挂载 点信息"页 签。	建议Ext4	DRS-Service安装部署及存储日志和业务数据所需的磁盘空间。 默认路径:/opt/drs
DRS-Service本地 备份集盘 默认路径:/opt/ drs-backup	数据盘大小要求请参见 "LLD模板 > 磁盘挂载 点信息页 签。	建议Ext4	DRS-Service备份磁盘空间,存储 drs-Service备份数据 默认路径:/opt/drs-backup
DRS-Service数据 库磁盘空间 默认路径: / data/cluster	数据盘大小要求请参见 "LLD模板 >磁盘挂载点信息"页 签。	建议Ext4	DRS-Service主数据库安装部署、以及存储所需磁盘空间。 默认路径:/data/cluster

#### 山 说明

- 建议将每个部分单独一块磁盘进行挂载,在分盘上对磁盘空间进行隔离。
- 可通过**df**-h命令查看文件系统磁盘使用情况。若不满足磁盘划分要求,建议参考**磁盘挂载步** 骤进行磁盘挂载。
- 表4-8中磁盘路径仅支持字母、数字和下划线。
- 最多支持纳管实例数,请参见**纳管节点规格**。
- /data目录建议独立挂盘,至少有50GB的剩余容量,用于存放安装工具包和待安装软件包。

# 4.1.4 账号信息

本章介绍了云数据库GaussDB管理平台(TPOPS)的账号信息。

安装时需要生成使用<mark>表4-9</mark>中所述的service、dbadmin、sftpservice用户,故安装前需确保环境中不存在上述用户。

### 表 4-9 账号列表

账号信息	解释说明
root	操作系统权限用户。登录待安装云数据库GaussDB管理平台 (TPOPS)管理节点时需要输入密码。root密码仅支持数字、 特殊字符+@以及字母。
service	启动微服务进程的用户,可在安装配置文件user_edit_file.conf 中指定用户service_group_id、service_user_id,默认值为 1010。若已被占用,可修改为其他值。 配置文件默认路径/data/docker-service/config/ user_edit_file.conf。
dbadmin	元数据库使用的操作系统用户。
sftpservice	SFTP服务用户。
influxdb	时序数据库InfluxDB使用的操作系统用户。

数据复制软件相关账号信息,请参阅《 HCS Data Replicate Software 安装指南》相关内容。

## 4.1.5 安装流程

下图展示了云数据库GaussDB管理平台(TPOPS)的安装流程。

图 4-2 安装流程

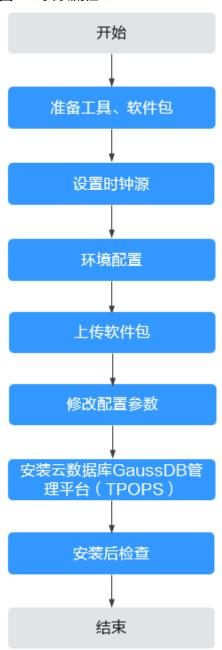


表 4-10 云数据库 GaussDB 管理平台 (TPOPS)管理安装流程

编号	流程	描述
1	获取并校验软件包	详情请参见 <b>准备工具、软件包</b> 。
2	配置安装环境	详情请参见 <b>环境配置</b> 。
3	上传软件包	详情请参见 <b>上传软件包</b> 。
4	修改配置参数	请参见 <b>修改配置参数</b> 。

编号	流程	描述
5	安装云数据库GaussDB管理 平台(TPOPS)	详情请参见 <b>安装云数据库GaussDB管理平台</b> (TPOPS)。
6	验证安装结果	详情请参见 <b>安装后检查</b> 。

# 4.2 安装前准备

# 4.2.1 准备工具、软件包

## 准备工具

本文安装环境以Kylin V10(SP2)ARM服务器为例。

准备三台华为TaiShan服务器,操作系统银河麒麟V10。同时在本地Windows系统上安装<mark>表中软件</mark>。

具体软件,请到对应的官方网站获取。

表 4-11 在 Windows 系统本机需要准备的软件

工具名称	说明	获取方式
SSH远程登录工 具(例如 PuTTY)	跨平台远程访问工具。用于在软件安装过程中 在Windows系统中访问各节点,例如登录节点 执行安装前配置、安装前检查、安装等命令。	您可以访问工 具对应的官方 网站下载软
WinSCP	跨平台文件传输工具。用于在Windows系统和 Linux系统间传输文件,例如上传软件包、上传 配置文件等操作。	件。
解压缩软件,例 如7-zip	用于解压*.tar.gz文件。	
浏览器,例如:	用于登录各类操作界面。	
• Chrome	版本要求:	
Firefox	• Google Chrome 120,119,118	
Microsoft	• Firefox 121, 120	
Edge	● Microsoft Edge: 随Windows 10更新	

## 准备软件包

根据当前操作系统类型,通过root用户下载对应的云数据库GaussDB管理平台(TPOPS)软件包和校验文件。

\*表示具体版本号,使用时请以软件包中实际版本号为准。

不同操作系统镜像的开源软件列表,请参见**开源软件列表**。安装前请对比开源软件列表,避免冲突。

表 4-12 版本软件包

软件包名称	子软件包名称	说明	获取方式
DBS-GaussDB- Server_*tar.gz	DBS-GaussDB- backupmanager_* _all.tar.gz	备份管理服务软件包	• 企业用 户: <b>点此</b> <b>前往</b>
	DBS-GaussDB- feature- data_*_all.tar.gz	云数据库GaussDB管理 平台(TPOPS)元数据 库基础SQL包	<ul><li>返营商用</li><li>户:点此</li><li>前往</li></ul>
	DBS-GaussDB- data_*_all.tar.gz	云数据库GaussDB管理 平台(TPOPS)元数据 库基础SQL包	
	DBS-GaussDB- instancemanager_ *_all.tar.gz	实例管理服务软件包	
	DBS-GaussDB- open- api_*_all.tar.gz	云数据库GaussDB管理 平台(TPOPS)open- api微服务的API入口	
	DBS-GaussDB- agent_*_all.tar.gz	云数据库GaussDB Agent包	
DBS-docker- service_*_all.tar.gz	-	安装工具包,无状态服 务	
DBS- GaussDBConsole- Server_*tar.gz	DBS-gaussdb- console_*_all.tar.gz	前端控制台软件包	
DBS-Platform- Server_*tar.gz	DBS- luban_*_all.tar.gz	监控运维管理软件包	
	DBS-monitor- service_*_all.tar.gz	监控指标软件包	
	DBS- ots_*_all.tar.gz	操作审计软件包	
	DBS-platform- data_*_all.tar.gz	云数据库GaussDB管理 平台(TPOPS)元数据 库基础SQL包	
	DBS-rds-ha- admin_*_all.tar.gz	实例监控服务软件包	
	DBS-resource- manager_*_all.tar. gz	资源管理软件包	

软件包名称	子软件包名称	说明	获取方式
	DBS- workflow_*_all.tar. gz	任务流公共服务软件包	
	DBS- auth_*_all.tar.gz	用户鉴权软件包	
	DBS-common- service_*_all.tar.gz	公共服务软件包	
GaussDB_OS_PATC H_*zip	-	主机上线os_patch包	
DBS-GaussDB- Kernel_*tar.gz	DBS-GaussDB-om- agent_*tar.gz	云数据库GaussDB om-agent包	
DBS-GaussDB- Kylin- Kernel_*tar.gz	-	麒麟操作系统云数据库 GaussDB内核包	
DBS-GaussDB- Uniontech- Kernel_*tar.gz	-	统信操作系统云数据库 GaussDB内核包	
DBS-DBMind- Manual_*tar.gz	-	云数据库GaussDB DBMind内核包	
DBS- tools_*_all.tar.gz	-	安装工具包,含证书文 件	
DBS- MetaDB_Kylin_Cent ralized_505.1.0.B02 6.tar.gz	-	麒麟操作系统元数据库 内核包	
DBS- MetaDB_UnionTech _Centralized_505.1. 0.B026.tar.gz	-	统信操作系统元数据库 内核包	

# 4.2.2 设置时钟源

### 4.2.2.1 配置说明

安装云数据库GaussDB管理平台(TPOPS)需要保证各节点间的时钟源同步,现提供如下两种可供参考的时钟源配置指导。请用户依据实际情况进行配置。

 推荐使用Chrony(Chrony Time Daemon)来自动同步各主机上的系统时间,使 云数据库GaussDB管理平台(TPOPS)各服务器之间的时间保持正确和一致。可 参考使用Chrony配置时间同步进行配置。请注意,统信系统仅可使用Chrony配置 时钟源。 ● 若需使用NTP(Network Time Protocol)来自动同步各主机上的系统时间,使云数据库GaussDB管理平台(TPOPS)各服务器之间的时间保持正确和一致。可参考使用NTP配置时间同步进行配置。

## 4.2.2.2 前提条件

- 配置时钟源前,确保您拥有所有机器的root用户权限。
- 配置时钟源的服务端主机与客户端主机的网络需要互通,并且Chrony或者NTP服务端口不被防火墙拦截。
- 配置时间同步前,确保所有机器的时区均为UTC时区。可执行以下命令,检查当前时区。

#### timedatectl

- 回显中"TIME Zone"一行显示时区为"UTC",表示当前时区正确。
- 若显现不为UTC时区,请执行以下命令,修改时区为"UTC"。

#### timedatectl set-timezone UTC

如果回显提示"Failed to set time zone: Failed to activate service 'org.freedesktop.timedate1': timed out",可执行如下步骤修改时区。

i. 使用vi命令编辑/etc/sysconfig/clock文件。

## vi /etc/sysconfig/clock

ii. 添加 "ZONE"信息。

ZONE="Etc/UTC"

iii. 执行如下命令,将系统时区设置为UTC时间。

In -sf /usr/share/zoneinfo/UTC /etc/localtime

## 4.2.2.3 使用 Chrony 配置时间同步

### 操作场景

为了保证时间误差在30秒以内,推荐使用Chrony(Chrony Time Daemon)来自动同步各主机上的系统时间,使云数据库GaussDB管理平台(TPOPS)各服务器之间的时间保持正确和一致。可参考以下操作步骤进行配置。

## 操作步骤

步骤1 以root用户登录到待配置时间同步的所有服务器节点。

步骤2 键入 "chrony" 并连按两次"Tab"键观察,检查是否安装了chrony。

- 若显示chronyc和chronyd,则表示已经安装了chrony。继续执行后续步骤。
- 若未显示则表示当前未安装chrony,执行以下命令进行安装。

#### yum install chrony -y

步骤3 执行以下命令,修改服务端配置。

#### 山 说明

该步骤修改的是服务端配置,请勿修改客户端。

1. 使用vi命令编辑/etc/chrony.conf。

vi /etc/chrony.conf

2. 参照如下图示,添加"allow all"信息。

# Allow NTP client access from local network. #allow 192. allow all

3. 参照如下图示,删除"#",取消"local stratum 10"所在行的注释。

# Serve time even if not synchronized to a time source. local stratum 10

- 4. 按 "Esc"键后执行:wq!命令,保存并退出。
- 5. 执行以下命令,重启服务端chrony服务使配置生效。

systemctl restart chronyd

步骤4 执行以下命令,修改客户端配置。

#### □ 说明

该步骤修改的是客户端配置,请勿修改服务端。

1. 使用vi命令编辑客户端的/etc/chrony.conf文件。

#### vi /etc/chrony.conf

2. 参照如下图示,添加"#"注释掉配置文件最前面原有的pool行,并新增"server时间同步服务器域名/IP地址 iburst"。

```
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
#pool pool.ntp.org iburst
server 192. iburst
```

- 3. 按"Esc"键后执行:wq!命令,保存并退出。
- 4. 执行以下命令,重启客户端chrony服务使配置生效。

#### systemctl restart chronyd

步骤5 执行以下命令,配置后检查。

在客户端与服务端分别使用如下命令查询,如服务端与客户端时间保持一致说明时钟源配置成功。

#### date

回显如下:

Fri Dec 15 07:39:58 UTC 2023

----结束

## 4.2.2.4 使用 NTP 配置时间同步

### 操作场景

为了保证时间误差在30秒以内,推荐使用NTP(Network Time Protocol)来自动同步各主机上的系统时间,使云数据库GaussDB管理平台(TPOPS)各服务器之间的时间保持正确和一致。可参考以下操作步骤进行配置。

### 操作步骤

#### 山 说明

通常使用系统自带的NTP服务可以同步时间。如果当前机器环境有稳定可靠的NTP服务器,则选它作为所有服务器的NTP源。

如果没有,则选固定的一台服务器,把它作为NTP源。

步骤1 在每台机器上通过yum包管理器安装NTP。

yum install ntp ntpdate -y

步骤2 配置ntp.conf文件,配置前做好备份。

#### 【NTP服务器的搭建】

在NTP服务端/etc/ntp.conf文件的最后,添加如下内容,其中:

restrict 192.168.0.0 mask 255.255.0.0中: 192.168.0.0需要替换为所配置NTP服务端IP 地址对应的网段。

[root@\* /]# vi /etc/ntp.conf restrict default ignore restrict 127.0.0.1 restrict 192.168.0.0 mask 255.255.0.0

driftfile /var/lib/ntp/drift pidfile /var/run/ntpd.pid #logfile /var/log/ntp.log

# local clock server 127.127.1.0 fudge 127.127.1.0 stratum 10

#### 【NTP客户端的搭建】

在NTP客户端/etc/ntp.conf文件的最后,添加如下内容,其中:

- restrict 192.168.0.0 mask 255.255.0.0中: 192.168.0.0需要替换为所配置NTP客户端IP地址对应的网段。
- server 192.168.\*.\* iburst minpoll 4 maxpoll 6中: 192.168.\*.\*需要替换为上述所配置的NTP服务端的IP地址。
- fudge 192.168.\*.\* stratum 10中: 192.168.\*.\*需要替换为上述所配置的NTP服务端的IP地址。

[root@\* /]# vi /etc/ntp.conf restrict default [*ignore*] restrict 127.0.0.1

restrict 192.168.0.0 mask 255.255.0.0

driftfile /var/lib/ntp/drift pidfile /var/run/ntpd.pid #logfile /var/log/ntp.log

# local clock

server 192.168.\*.\* iburst minpoll 4 maxpoll 6

fudge 192.168.\*.\* stratum 10

#### 参数解释:

restrict: 用于指定IP地址的相关时间同步命令权限。

restrict [客户端IP] mask [netmask\_IP] [parameter]

其中parameter包括ignore、nomodify、noquery、notrap和notrust等。

- ignore: 默认指拒绝所有类型的NTP同步。
- nomodify:客户端不能更改服务端的时间参数,但是客户端可以通过服务端进行网络校时。
- notrust: 客户端除非通过认证,否则该客户端来源将被视为不信任子网。
- noquery:不提供客户端的时间查询:用户端不能使用ntpq,ntpc等命令来查询NTP服务器。
- notrap:不提供trap远端登录:拒绝为匹配的主机提供模式6控制消息陷阱服务。陷阱服务是ntpdq控制消息协议的子系统,用于远程事件日志记录程序。
- server: 用于指定上层NTP源服务器。

### server [IP or hostname] [prefer]

如果没有上层NTP源服务器,可以设置为**127.127.1.0**,即本机作为NTP源服务器。

#### 步骤3 重启NTP同步服务。

systemctl restart ntpd

systemctl status ntpd

#### 步骤4 检查NTP同步状态。

#### ntpq-p

#### 回显如下:

#### 其中:

- remote:表示使用的NTP服务器。
  - \*表示目前选择的NTP服务器。
  - LOCAL表示本机。
  - x表示已不再使用。
  - 表示已不再使用。
  - +良好的优先考虑的。
  - #良好的但未使用的。
- refid: 远程NTP服务器使用的更高一级NTP服务器。INIT表示还在获取。
- st: 远程NTP服务器的Stratum(级别)。
- when:最后一次同步到现在的时间(默认为秒,h表示小时,d表示天 )。
- poll: 同步的频率,单位: 秒。
- delay:从本机到远程NTP服务器的往返时间(单位毫秒)。
- offset: 本机与远程NTP服务器的时间偏移量(单位毫秒)。
- jitter:本机与远程NTP服务器的时间偏移量平均偏差(单位毫秒)。

#### 步骤5 最终检查方法。

在客户端与服务端分别使用如下命令查询,如服务端与客户端时间保持一致说明时钟源配置成功。

#### date

回显如下:

Thu Sep 14 02:03:11 UTC 2023

步骤6 (可选)手动同步方法。

如果时间总是有很大误差,可以使用以下命令手动订正时间。这个与自动同步服务是 冲突的,需要先停掉NTP服务。

systemctl stop ntpd

ntpdate -u 192.168.\*.\*

回显如下:

18 Apr 14:54:20 ntpdate[108001]: adjust time server 192.168.\*.\* offset -0.000180 sec

如果这个方法有效,则配置到系统的crontab下。

crontab -e

回显如下:

\* \* \* \* \* root /sbin/ntpdate -u 192.168.\*.\* 2>&1 1>>/tmp/ntpupdate.log

----结束

# 4.2.3 配置 yum 源

步骤1 以root用户登录云数据库GaussDB管理平台(TPOPS)所有节点进行yum源配置。

步骤2 上传与支持的操作系统版本一致的操作系统ISO文件到服务器任意目录,例如/mnt。

步骤3 执行以下命令,进入/mnt。

cd /mnt

步骤4 执行以下命令,将ISO文件挂载至/mnt路径下。

mount -o loop <iso文件名 > /mnt

步骤5 清空原不可用yum源。

rm -rf /etc/yum.repos.d/\*

步骤6 执行以下命令,进入local.repo文件。

vi /etc/yum.repos.d/local.repo

步骤7 新增以下信息,创建本地yum源配置。

[local] name=local baseurl=file:///mnt gpgcheck=0 enabled=1

#### 山 说明

其中baseurl中的/mnt为ISO文件的挂载路径。

步骤8 按 "Esc"键后执行以下命令,保存并退出。

:wq!

步骤9 清除yum缓存。

yum clean all

步骤10 缓存本地yum源。

yum makecache

----结束

## 4.2.4 环境配置

## 操作场景

详细介绍如何检查以及准备安装云数据库GaussDB管理平台(TPOPS)所依赖的一些组件以及系统配置。

#### 山 说明

- 本文均以root用户登录云数据库GaussDB管理平台(TPOPS)所有节点进行检查,支持密码 登录或密钥登录节点,使用密钥登录节点的设置,可以参考免密登录管理面root账户设置。
- 本文中提到的用户名、用户目录和软件包等信息,仅作示例说明。
- 监控、告警等服务,依赖服务器时间,请正确配置时间同步并确保时间服务器正常。

### 检查项

1. 执行以下命令,检查防火墙是否关闭。关闭防火墙。

#### systemctl status firewalld

- 若防火墙状态显示为inactive (dead),表示防火墙已关闭。
- 若防火墙状态显示为active (running),表示防火墙未关闭。请参照**如何关闭 防火墙**,关闭防火墙。
- 2. 执行以下命令,设置操作系统的pam规则。
  - a. 执行以下命令,打开/etc/pam.d/system-auth。

#### vi /etc/pam.d/system-auth

b. 参照如下图示,找到图示中红框圈中的行,并按照图示(黄框)在行末添加"minlen=8"。

```
#WAPM-1.0

# User changes will be destroyed the next time authconfig is run.
auth required pam_env.so
auth required pam_env.so
auth sufficient pam_unix.so nullok try_first_pass
auth sufficient pam_faillock.so authfail audit deny=3 even_deny_root unlock_time=60
auth sufficient pam_faillock.so authfail audit deny=0 even_deny_root unlock_time=0
auth sufficient pam_faillock.so authfail audit deny=3 even_deny_root unlock_time=0
auth required pam_faillock.so authfail audit deny=3 even_deny_root unlock_time=60
auth required pam_faillock.so authsucc audit deny=3 even_deny_root unlock_time=60
auth sufficient pam_succeed_if.so uid >= 1000 quiet_success
auth required pam_prinit.so
auth required pam_prinit.so
auth required pam_prinit.so sufficient pam_unix.so sha512 shadow nullok try_first_pass use_authtok
sufficient pam_unix.so sha512 shadow nullok try_first_pass use_authtok
sufficient pam_unix.so suse_authtok use_first_pass
auth required pam_unix.so sha512 shadow nullok try_first_pass use_authtok
session optional pam_keyinit.so revoke
pam_lunix.so
pam_deny.so
session optional pam_systemd.so
session optional pam_systemd.so
```

c. 执行以下命令,保存并退出。

:wq!

3. (可选)排查audit服务是否有内存泄露风险。具体排查方法,请参考**麒麟系统** audit服务内存泄露占用大量内存。

## 4.2.5 上传软件包

步骤1 以root用户登录待安装的云数据库GaussDB管理平台(TPOPS)节点。

步骤2 执行以下命令,创建目录。

需为所有待安装节点创建/data目录(以/data目录为例,实际可创建任意目录),然后在任一节点执行后续操作。

mkdir -p /data

步骤3 上传云数据库GaussDB管理平台(TPOPS)安装包至待安装节点的/data/目录。

例如:将准备好的DBS-docker-service\_\*\_all.tar.gz上传到目录/data/。

步骤4 执行以下命令,解压安装脚本包到安装目录(/data)下。

cd /data

tar -xzvf DBS-docker-service\*\_all.tar.gz -C /data

步骤5 上传以下安装包到指定目录/data/docker-service/pkgs下。

- 元数据库内核包(共1个):
  - 麒麟操作系统对应的包名为:DBS-MetaDB\_Kylin\_Centralized\_505.1.RC1.SPC0100.B006.tar.gz
  - 统信操作系统对应的包名为:
    DBS-MetaDB UnionTech Centralized 505.1.RC1.SPC0100.B006.tar.gz
- 微服务包(共3个):

DBS-GaussDB-Server \* .tar.gz

DBS-Platform-Server\_\*\_.tar.gz

DBS-GaussDBConsole-Server\_\*\_.tar.gz

GaussDB实例安装包(共6个):

GaussDB\_OS\_PATCH\_\*zip

DBS-DBMind-Manual\_\*.tar.gz

DBS-tools\_\*\_all.tar.gz

DBS-GaussDB-Kernel\_\*.tar.gz

DBS-GaussDB-Kylin-Kernel\_\*.tar.gz

DBS-GaussDB-Uniontech-Kernel\_\*.tar.qz

#### 山 说明

DBS-GaussDB-Kylin-Kernel\_\*.tar.gz为麒麟操作系统对应的安装包,DBS-GaussDB-Uniontech-Kernel\_\*.tar.gz为统信系统对应的安装包。

上传DBS-GaussDB-Kylin-Kernel\_\*.tar.gz或DBS-GaussDB-Uniontech-Kernel\_\*.tar.gz安装包时,可根据数据面的实际需要,选择对应的包进行上传。

#### ----结束

#### 须知

- 需要保证包的唯一,同一个微服务或组件的包只能上传一个。
- 如果**步骤5**安装包有更换,需要参考**如何重新分发安装包**重新分发安装包。

## 4.2.6 修改配置参数

### 操作场景

介绍如何配置安装所需的配置文件,配置参数详细说明见步骤4。

## 操作步骤

#### □ 说明

- 本节所涉及的IP地址均需在ifconfig查询的范围内。
- 所配置的目录长度不超过64位。

步骤1 以root用户登录待安装云数据库GaussDB管理平台(TPOPS)的节点。

该节点为上传软件包所在节点。

步骤2 执行以下命令,进入到配置文件所在目录。

cd /data/docker-service/config

步骤3 执行以下命令,进入配置文件。

vi user\_edit\_file.conf

步骤4 参照如下conf示例修改配置文件。

### **注意**

- 下述配置文件中的node1\_ip需要配置为执行节点IP地址,执行节点IP地址为下发命令的节点的IP地址。node2\_ip和node3\_ip也需要配置成其他的实际节点IP地址。node\*\_ip2,要求可以和GaussDB实例SSH通以及Ping通。具体配置参数描述,可参见安装配置文件参考。
- 如果配置参数有变化,需要参考如何重新分发安装包重新分发安装包。

```
[user_edit]
ssh_port = 22  # 节点之间登录的SSH端口号
gauss_path = /opt/gaussdb  # 元数据库安装目录(不能在/home目录下)
node1_ip = 192.168.0.1  # 节点1本机IP地址(执行节点IP地址)
node2_ip = 192.168.0.2  # 节点2本机IP地址
node3_ip = 192.168.0.3  # 节点3本机IP地址
influxdb_install_ip1 = 192.168.0.1  # InfluxDB安装节点1
influxdb_install_ip2 = 192.168.0.2  # InfluxDB安装节点2
sftp_install_ip1 = 192.168.0.1  # SFTP安装节点2
sftp_install_ip2 = 192.168.0.2  # SFTP安装节点2
main_path = /opt/cloud  # 微服务运行目录,目录可自定义,仅支持自定义第一级目录,二级目录cloud不允许修改
node1_ip2 = 100.95.0.1  # 节点1可以和GaussDB实例地址通信的IP地址(执行节点可以和GaussDB实例通信的IP地址,要求可以和GaussDB实例SSH通以及Ping通)
node2_ip2 = 100.95.0.2  # 节点2可以和GaussDB实例地址通信的IP地址(要求可以和GaussDB实例SSH通以及Ping通)
```

```
node3_ip2 = 100.95.0.3 # 节点3可以和GaussDB实例通信的IP地址 (要求可以和GaussDB实例SSH通以及Ping
涌)
log_path = /opt/cloud/logs # 日志目录, 目录可自定义,仅支持自定义第一级目录,二级目录cloud及之后目
录不允许修改
sftp_path = /opt/sftphome # SFTP数据目录, 目录可自定义,仅支持自定义第一级目录,二级目录sftphome
不允许修改
influx_path = /opt/influxdb # InfluxDB数据目录,需要保证influxDB_install_ip1以及influxDB_install_ip2节点
InfluxDB的父目录都有InfluxDB用户的可执行权限(如遍历父目录,对目录执行chmod a+x /directory_1/
directory_2),目录可自定义,仅支持自定义第一级目录,二级目录influxdb目录不允许修改
docker_path = /opt/docker # Docker数据目录,目录可自定义,仅支持自定义第一级目录,二级目录docker
不允许修改,如果用户环境已安装Docker,则以用户配置为准,该项配置不生效。
backup_path = /opt/backup # 备份数据目录,目录可自定义,仅支持自定义第一级目录,二级目录backup不
service_group_id = 1010 # Service用户ID(需未被使用)
service_user_id = 1010 # Service用户组ID(需未被使用)
```

uninstall\_all = no # 卸载时需将参数值配置成yes,默认为no

use\_cgroup = no # 是否使用cgroup限制资源(需要和DRS合部时,配置为yes,其余场景配置为no)

**步骤5** 执行以下命令,保存并退出。

#### :wq!

#### ----结束

#### □ 说明

云数据库GaussDB管理平台(TPOPS)各个服务器之间需要配置NTP服务器,保证云数据库 GaussDB管理平台(TPOPS)各个服务器之间的时间一致性。

# 4.3 安装过程

## 4.3.1 前置检查

步骤1 以root用户进入上传软件包节点appctl.sh文件所在目录。

#### cd /data/docker-service

步骤2 执行以下命令,进行安装前检查。未配置节点互信的情况下,请根据提示输入root密 码。

#### □ 说明

如果是SUSE操作系统,在执行前置检查的命令前,需要执行以下命令替换前置检查脚本。

\cp -af /data/docker-service/action/mainAction/precheck.sh.suse /data/docker-service/ action/mainAction/precheck.sh

#### sh appctl.sh precheck

回显如下,说明前置检查通过。

```
*** CHECK BASE SETTINGS ***
check python
                        | OK
                      OK
check jdk
                       | OK
check net tools
check expect
                       | OK
                      OK: Installed secc.
check secc
check libcgroup
                        | OK: Installed libcgroup.
check dos2unix
                          OK: Installed dos2unix.
check custom dir
                         | OK
check install mode
                         OK: HA mode.
check IP format
                        | OK
check net condition
                         | OK
```

```
start check node authentication
node not support auto authentication, will input root password
请输入root用户密码:
check node authentication success.
Now scp precheck files...
scp precheck files done
Now doing precheck for 192.168.0.175
Now doing precheck for 192.168.0.140
Now doing precheck for 192.168.0.243
[WARNING]-[check_drs]===>[192.168.0.140: The switch of use_cgroup is no.You are not allow to install drs
together on this machine.]
[WARNING]-[check_dir_total]===>[192.168.0.140: all directory are not belong to same top directory, the
check of directory mem size may be incorrect.]
[WARNING]-[check_dir_mount]===>[192.168.0.140: /opt/cloud was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.140: /opt/cloud/logs was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.140: /opt/docker was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.140: /opt/gaussdb was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.140: /opt/influxdb was not mounted.]
[WARNING]-[check_users]===>[192.168.0.140: [service sftpservice] will be used, if the users already exists,
we will delete the user before creating it.]
[WARNING]-[check_drs]===>[192.168.0.175: The switch of use_cgroup is no. You are not allow to install drs
together on this machine.]
[WARNING]-[check_dir_total]===>[192.168.0.175: all directory are not belong to same top directory, the
check of directory mem size may be incorrect.]
[WARNING]-[check_dir_mount]===>[192.168.0.175: /opt/cloud was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.175: /opt/cloud/logs was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.175: /opt/docker was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.175: /opt/gaussdb was not mounted.] [WARNING]-[check_dir_mount]===>[192.168.0.175: /opt/influxdb was not mounted.]
[WARNING]-[check_users]===>[192.168.0.175: [service sftpservice] will be used, if the users already exists,
we will delete the user before creating it.]
[WARNING]-[check_drs]===>[192.168.0.243: The switch of use_cgroup is no.You are not allow to install drs
together on this machine.
[WARNING]-[check_dir_total]===>[192.168.0.243: all directory are not belong to same top directory, the
check of directory mem size may be incorrect.]
[WARNING]-[check_dir_mount]===>[192.168.0.243: /opt/cloud was not mounted.]
[WARNING]-[check dir mount]===>[192.168.0.243: /opt/cloud/logs was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.243: /opt/docker was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.243: /opt/gaussdb was not mounted.]
[WARNING]-[check_users]===>[192.168.0.243: [service sftpservice] will be used, if the users already exists,
we will delete the user before creating it.]
Precheck all completed.
```

#### 山 说明

前置检查回显中,[WARNING]级别消息为提示消息,不会阻塞安装。[ERROR]级别消息为错误 消息,需要解决所有ERROR消息后,再进行安装。

ERROR信息处理可参考前置检查报错处理。

#### ----结束

## 4.3.2 安装云数据库 GaussDB 管理平台(TPOPS)

### 操作场景

介绍安装云数据库GaussDB管理平台(TPOPS)的详细步骤,以及安装回显的展示和 安装耗时的说明。

### 操作步骤

#### 山 说明

- 需要保证云数据库GaussDB管理平台(TPOPS)各个服务器的root密码保持一致,如需要配置免密安装,可以参考管理面节点之间互信设置。
- 完整安装过程预计耗时50分钟左右,其中元数据库GaussDB的安装约20分钟。
- 安装过程中禁止手动退出。

步骤1 以root用户进入上传软件包节点appctl.sh文件所在目录。

#### cd /data/docker-service

步骤2 执行以下命令,进行安装。未配置节点互信的情况下,请根据提示输入root密码。

#### sh appctl.sh install

回显如下,说明安装完成。

```
*** CHECK BASE SETTINGS ***
check python
                       | OK
check jdk
                      OK
check net tools
                       | OK
check expect
                       | OK
                      | OK: Installed secc.
check secc
                       | OK: Installed libcgroup.
check libcgroup
check dos2unix
                        OK: Installed dos2unix.
check custom dir
                       | OK
check install mode
                        OK: HA mode.
check IP format
                        OK
check net condition
                         | OK
start check node authentication
node not support auto authentication, will input root password
请输入root用户密码:
start check host: 192.168.0.1 root password
check host: 192.168.0.1 root password success
start check host: 192.168.0.2 root password
check host: 192.168.0.2 root password success
start check host: 192.168.0.3 root password
check host: 192.168.0.3 root password success
*** UNTAR SERVICE ***
untar service
                       I OK
*** BUILD MICRO SERVICE IMAGES ***
check docker status
                       | OK
check service origin packages | OK
                          | OK
load docker base image
build & save zookeeper
                           I OK
build & save kafka
                        | OK
build & save common-service | OK
build & save monitor-service | OK
build & save rds-ha-admin | OK
build & save resource-manager | OK
build & save workflow
                        | OK
build & save auth
                        I OK
build & save gaussdb-console | OK
                        | OK
build & save luban
build & save ots
                        | OK
build & save GaussDB-open-api | OK
build & save GaussDB-instancemanager | OK
build & save GaussDB-backupmanager | OK
*** DISTRIBUTE PACKAGES ***
/data/docker-service left space | OK
check packages
                        OK
distribute gaussdb packages | OK
distribute service packages | OK
distribute data packages | OK
```

```
distribute sftp packages | OK
distribute remote ip
                          | OK
*** PREPARE ***
192.168.0.1
                        OK
192.168.0.2
                        OK
192.168.0.3
                      | OK
Now doing precheck for 192.168.0.1
Now doing precheck for 192.168.0.2
Now doing precheck for 192.168.0.3
[WARNING]-[check_drs]===>[192.168.0.1: The switch of use_cgroup is no.You are not allow to install drs
together on this machine.]
[WARNING]-[check_dir_total]===>[192.168.0.1: all directory are not belong to same top directory, the
check of directory mem size may be incorrect.]
[WARNING]-[check_dir_mount]===>[192.168.0.1: /opt/cloud was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.1: /opt/cloud/logs was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.1: /opt/docker was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.1: /opt/gaussdb was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.1: /opt/influxdb was not mounted.]
[WARNING]-[check_users]===>[192.168.0.1: [service sftpservice] will be used, if the users already exists, we
will delete the user before creating it.]
[WARNING]-[check_drs]===>[192.168.0.2: The switch of use_cgroup is no. You are not allow to install drs
together on this machine.]
[WARNING]-[check_dir_total]===>[192.168.0.2: all directory are not belong to same top directory, the
check of directory mem size may be incorrect.]
[WARNING]-[check_dir_mount]===>[192.168.0.2: /opt/cloud was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.2: /opt/cloud/logs was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.2: /opt/docker was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.2: /opt/gaussdb was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.2: /opt/influxdb was not mounted.]
[WARNING]-[check_users]===>[192.168.0.2: [service sftpservice] will be used, if the users already exists, we
will delete the user before creating it.]
[WARNING]-[check_drs]===>[192.168.0.3: The switch of use_cgroup is no. You are not allow to install drs
together on this machine.
[WARNING]-[check_dir_total]===>[192.168.0.3: all directory are not belong to same top directory, the
check of directory mem size may be incorrect.]
[WARNING]-[check_dir_mount]===>[192.168.0.3: /opt/cloud was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.3: /opt/cloud/logs was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.3: /opt/docker was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.3: /opt/gaussdb was not mounted.]
[WARNING]-[check_users]===>[192.168.0.3: [service sftpservice] will be used, if the users already exists, we
will delete the user before creating it.]
Precheck all completed.
Start to init manifest...
init manifest successful for 192.168.0.1.
init manifest successful for 192.168.0.2.
init manifest successful for 192.168.0.3.
====== 192.168.0.1: patch ======
                  | complete
 patch
 base_env
                    complete
====== 192.168.0.2: patch ======
                 | complete
 patch
                    | complete
 base_env
====== 192.168.0.3: patch =======
                  | complete
 patch
 base_env
                    complete
====== 192.168.0.1: base_enviornment =======
 docker
                   complete
 InfluxDB
                   complete
                 complete
 sftn
 gaussdb
                    complete
===== 192.168.0.2: base enviornment ======
                  | complete
 docker
 InfluxDB
                   complete
                 | complete
 sftp
 gaussdb
                    | complete
===== 192.168.0.3: base_enviornment ======
                   | complete
 docker
 InfluxDB
                   complete
                 | complete
 sftp
```

```
gaussdb | complete
====== 192.168.0.1: CommonbaseData ======
common-base | complete
====== 192.168.0.2: CommonbaseData =======
common-base | complete
====== 192.168.0.3: CommonbaseData =======
common-base | complete
====== 192.168.0.1: PlatformData =======
platform-data | complete
====== 192.168.0.2: PlatformData ======
platform-data | complete
======= 192.168.0.3: PlatformData =======
platform-data | complete
====== 192.168.0.1: Zookeeper =======
zookeeper | complete
 GaussDB-feature-data | complete
GaussDB-data | complete
====== 192.168.0.2: Zookeeper ======
zookeeper | complete
 GaussDB-feature-data | complete
GaussDB-data | complete
====== 192.168.0.3: Zookeeper ======
zookeeper
            | complete
 GaussDB-feature-data | complete
GaussDB-data | complete
====== 192.168.0.1: Kafka =====
kafka
           complete
======= 192.168.0.2: Kafka ===
kafka
           complete
====== 192.168.0.3: Kafka ======
               complete
====== 192.168.0.1: docker_service ======
common-service | complete
monitor-service
                  complete
rds-ha-admin | complete resource-manager | complete
          | complete
workflow
====== 192.168.0.2: docker_service ======
common-service | complete
monitor-service | complete
resource-manager | complete workflow
                   | complete
====== 192.168.0.3: docker_service ======
common-service | complete
                  | complete
monitor-service
rds-ha-admin | complete resource-manager | complete
workflow | complete
====== 192.168.0.1: gaussdb_service =======
          | complete
auth
 gaussdb-console
                  complete
luban
               complete
             complete
ots
 GaussDB-open-api | complete
 GaussDB-instancemanager | complete
 GaussDB-backupmanager | complete
====== 192.168.0.2: gaussdb_service ======
auth
               | complete
 gaussdb-console | complete
luban | complete
ots
              | complete
GaussDB-open-api | complete
 GaussDB-instancemanager | complete
GaussDB-backupmanager | complete
====== 192.168.0.3: gaussdb_service =======
          | complete
gaussdb-console | complete
```

```
luban | complete
ots | complete
GaussDB-open-api | complete
GaussDB-instancemanager | complete
GaussDB-backupmanager | complete
Installation progress [72/72] ==> 100.00%
Upload sftp packages successful for 192.168.0.1
请登录「云数据库GaussDB管理平台-任务中心」查看安装包上传任务状态
Welcome to TPOPS:

https://{EIP}:8443/gaussdb/#/login
EIP: Elastic IP address of any TPOPS node
```

#### ----结束

## <u>注意</u>

升级、卸载都依赖docker-service目录,请谨慎删除。

若docker-service目录被删除后需要卸载云数据库GaussDB管理平台(TPOPS),请参 照**docker-service目录被删除的卸载步骤**,不需要参照**卸载云数据库GaussDB管理平** 台(TPOPS)。

# 4.4 安装后检查

步骤1 参考如下URL使用浏览器登录云数据库GaussDB管理平台(TPOPS)。

https://{ EIP}:8443/gaussdb/#/login

#### 其中:

- {*EIP*}为安装云数据库GaussDB管理平台(TPOPS)的任一节点的EIP(弹性公网IP)地址。
- 使用默认管理员账号密码登录。默认账号为admin,密码请参见《云数据库 GaussDB管理平台(TPOPS)账户一览表 01》中对应的默认密码。

步骤2 页面显示正常,即为云数据库GaussDB管理平台(TPOPS)安装成功。





步骤3 单击"任务中心",查看云数据库GaussDB安装包上传任务状态。

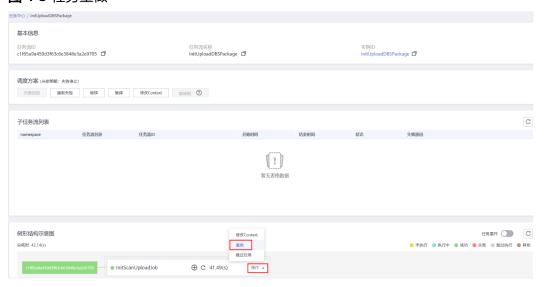
步骤4 出现"完成"字样,则代表安装包上传成功。

### 图 4-4 安装包上传成功



否则,单击"任务中心 > 任务详情 > 操作 > 重做",重做该任务。如再次失败,请**联 系技术支持**人员,解决云数据库GaussDB安装包上传问题。

#### 图 4-5 任务重做



#### ----结束

**5** 安装实例

# 5.1 系统要求

## 操作系统要求

表 5-1 支持的操作系统及相应安装包

操作系统	支持的版本	架构
银河麒麟 V10	SP1: 4.19.90-23.8.v2101.ky10.aarch64	ARM
	SP2: 4.19.90-24.4.v2101.ky10.aarch64	
	SP1: 4.19.90-23.8.v2101.ky10.x86_64	X86
	SP2: 4.19.90-24.4.v2101.ky10.x86_64	
统信	4.19.90-2201.4.0.0135.up1.uel20.x86_64	X86
	4.19.90-2201.4.0.0135.up1.uel20.aarch6 4	ARM

## <u> 注意</u>

只支持英文操作系统。

## 软件要求

表 5-2 数据库 GaussDB 数据面节点软件要求

软件	规格
Python	Python 3.7.9。 可执行如下指令,检查Python版本。 python3version 如果查询到的版本不是Python3.7.9,需要参考 <mark>安装主机的</mark> Python3,安装Python3.7.9版本。

### 硬件要求

类型	推荐配置
X86	2*6248R,24 Core@3.0GHz, 24*32GB 内存, 2*960GB SATA SSD, 24*960G SATA SSD, SR450C-M 2G(Avago3508), 2*2*10GE/25GE
ARM	2*Kunpeng 920,64 Core@2.6GHz,32*32GB内存,2*960GB SATA SSD,24*960G SATA SSD,1*Avago3508, 2*2*25GE

## <u>注意</u>

请使用硬件规格大于或等于8U64G的主机。

GaussDB实例仅支持8U64G、16U128G、32U256G、64U512G、96U768G、128U1024G、196U1568G(仅DBmind)七档规格。安装实例时,将根据所选择主机中的最小规格,往下匹配合适的实例规格。若主机的规格小于最小实例规格,将无法安装实例。

# 5.2 修改操作系统配置

## 5.2.1 使用须知

- 配置操作系统防火墙~5.2.10-设置umask章节的操作均需以root用户在所有实例 节点进行,操作完成后请及时登出root用户,避免误操作。
- 配置操作系统防火墙和关闭swap交换内存两个配置均需要重启机器,可以两个操作配置完成后一起重启。
- 数据库实例安装完成后,在SATA/SAS SSD下磁盘调度方式建议使用deadline调度方式,NVMe建议使用none调度方式。

## 5.2.2 配置操作系统防火墙

在防火墙关闭的状态下进行安装,关闭防火墙操作步骤如下。

步骤1 执行以下命令,检查防火墙是否关闭。

#### systemctl status firewalld

```
[root@host-192-168-0-36 data] # systemctl status firewalld
• firewalld.service - firewalld - dynamic firewall daemon
  Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
  Active: [inactive (dead)] since Fri 2023-11-24 15:09:25 CST; 24h ago
  Docs: man:firewalld(1)
Main PID: 970 (code=exited, status=0/SUCCESS)
```

#### 如图所示, 查看红框位置所示状态:

- 若防火墙状态显示为active(running),则表示防火墙未关闭,请执行<mark>步骤2</mark>。
- 若防火墙状态显示为inactive(dead),则无需再关闭防火墙。

步骤2 执行以下命令,关闭防火墙并禁止开机启动。

systemctl stop firewalld.service

systemctl disable firewalld.service

步骤3 修改/etc/selinux/config文件中的 "SELINUX" 值为 "permissive"。

#### □ 说明

一般情况下,开启selinux会提高系统的安全性,但是可能导致程序无法运行。为保证安装顺利,建议用户设置值为permissive。

1. 使用vi打开config文件。

vi /etc/selinux/config

2. 修改"SELINUX"的值"permissive"。

**SELINUX**=permissive

3. 按 "Esc"键后执行:wq!保存并退出修改。

步骤4 执行以下命令,重新启动操作系统。

reboot

步骤5 在其他主机上重复步骤1到步骤4。

## <u> 注意</u>

数据库安装后会开启iptables,并将数据库的服务、协议、IP地址以及端口添加到数据库各主机的防火墙白名单中,可使用iptables -vnL查看。

请慎重使用iptables -F,这可能造成某些功能不可用,如恢复功能等。

----结束

## 5.2.3 关闭 swap 交换内存

步骤1 以root用户登录各主机。

步骤2 (临时关闭)执行以下命令,关闭交换内存。

swapoff -a

步骤3 (永久关闭)注释/etc/fstab下swap启动项,并重启机器。

1. 执行以下命令,进入/etc/fstab。

vi /etc/fstab

2. 找到swap启动项,在对应命令行前键入"#"进行注释。

麒麟系统如下:

# /dev/mapper/klas-swap none swap defaults 0 0

统信系统如下:

# /dev/mapper/uos-swap none swap defaults 0 0

3. 按"Esc"键后执行以下命令,保存并退出修改。

:wa!

4. 执行以下命令,重启机器。

reboot

----结束

## 5.2.4 设置字符集参数

将各主机的字符集设置为相同的字符集。

步骤1 使用vi打开/etc/profile文件。

vi /etc/profile

步骤2 添加以下字段。

export LANG=en\_US.UTF-8

步骤3 按 "Esc"键后执行:wq!保存并退出修改。

步骤4 设置/etc/profile生效。

source /etc/profile

步骤5 使用vi打开/etc/sysconfig/i18n文件。

vi /etc/sysconfig/i18n

步骤6 修改参数 "LANG" 的值为en US.UTF-8。

**步骤7** 执行以下命令,确保安装用户的字符集生效。

source /etc/sysconfig/i18n

#### 山 说明

若机器没有/etc/sysconfig/i18n文件,可进入/etc/locale.conf文件。

vi /etc/locale.conf

修改参数 "LANG"的值为en\_US.UTF-8后执行source /etc/locale.conf即可。

----结束

## 5.2.5 设置时钟源

安装实例需要保证各主机的时钟源同步,因此主机上线时会强制校验时间,请设置时钟源,确保时间同步在1秒之内。

支持使用Chrony(Chrony Time Daemon)和NTP(Network Time Protocol)来设置时钟源。如何设置时钟源,可参见**设置时钟源**。

#### 须知

- 1. 如果不使用Chrony或者NTP来设置时钟源,在手动确认了主机时间同步之后,跳过添加主机任务流中的"NebulaHostDetectionTask"任务即可。
- 2. 如果使用手动同步方式配置NTP,主机标准化检查无法检测,跳过添加主机任务流中的"NebulaHostDetectionTask"任务即可。

## 5.2.6 设置网卡 MTU 值

#### □ 说明

推荐架构类型为X86的主机网卡MTU值为1500,架构类型为ARM的主机网卡MTU值为8192(可 选 )。

**步骤1** 执行命令**ifconfig**,查看IP地址绑定的网卡,如eth0。

步骤2 执行以下命令,查看与IP地址绑定的网卡后显示的MTU的值。

ifconfig eth0 | grep mtu

回显如下。

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

步骤3 如上回显MTU值为1500,检查是否满足推荐值。

- 是: 无需修改。
- 否,请参照后续步骤,修改网卡MTU的值。

步骤4 执行以下命令, 打开文件ifcfq-\*。

vi /etc/sysconfig/network-scripts/ifcfg-\*

\*表示步骤1中查询到的IP地址绑定的网卡,如eth0。

步骤5 按"i"进入编辑模式,添加如下语句,设置网卡MTU值,以设置MTU为1500为例。

MTU=1500

步骤6 按 "Esc"键后输入:wq!保存并关闭文件。

步骤7 执行以下命令,重启网络。

systemctl restart NetworkManager

----结束

## 5.2.7 安装 OpenSSH

### 操作场景

主机上线前需要允许SSH跳转进行**scp**命令拷贝安装包,否则可能会导致主机上线失败。

#### 前提条件

银河麒麟环境需要提前配置yum源。yum源配置参考配置yum源。

### 操作步骤

步骤1 通过如下命令,检查是否安装了OpenSSH工具。

ssh -V

回显如下:

OpenSSH\_8.2p1, OpenSSL 1.1.1f 31 Mar 2020

步骤2 若没有出现OpenSSH版本,或者提示命令不存在,需执行以下命令进行安装。

yum -y install openssh-server

步骤3 安装完成后,执行以下命令启动OpenSSH服务。

systemctl start sshd

----结束

## 5.2.8 检查 Expect

### 操作场景

主机上线前需要允许expect命令处理SSH交互过程,否则可能会导致主机上线失败。

### 前提条件

银河麒麟环境需要提前配置yum源。yum源配置参考配置yum源。

### 操作步骤

步骤1 通过如下命令检查主机是否安装了Expect。

expect -v

回显如下:

expect version 5.45.4

步骤2 若没有出现Expect版本,或者提示命令不存在,执行以下命令,进行安装。

yum install expect -y

步骤3 检查软件是否安装成功。

expect -v

----结束

## 5.2.9 配置 sshd\_config

### 操作场景

主机上线前需要在sshd\_config文件中将GSSAPIAuthentication参数设置为no,否则可能会导致主机上线失败。

### 操作步骤

步骤1 使用vi打开/etc/ssh/sshd\_config文件。

vi /etc/ssh/sshd\_config

步骤2 设置GSSAPIAuthentication参数为no。

**GSSAPIAuthentication no** 

步骤3 按 "Esc"键后执行:wq!保存并退出修改。

步骤4 重启SSH服务。

systemctl restart sshd.service

----结束

## 5.2.10 设置 umask

步骤1 执行如下指令,查看umask回显值。

umask

- 若回显小于等于0022,表示umask设置正确。
- 若回显大于0022,请执行后续步骤,设置umask。

步骤2 执行如下命令,打开bashrc文件。

vi /etc/bashrc

步骤3 在bashrc文件的最下方增加一行,使umask的值等于0022。

umask 0022

步骤4 按 "Esc"键后执行:wq!保存并退出修改。

步骤5 执行如下命令,使umask修改生效。

source /etc/bashrc

步骤6 如果主机上已经安装Agent,执行如下命令,重启Agent。

touch /home/Ruby/need\_shut\_down.touch

否则,执行步骤7。

步骤7 再次执行umask命令,回显等于0022表示设置umask成功。

----结束

# 5.3 准备初始化环境

## 5.3.1 准备磁盘

GaussDB支持使用SSD盘作为数据库的主存储设备,支持SAS/SATA和NVME协议的SSD盘(以下分别简称SSD盘和NVMe盘)。

建议将数据盘配置为RAID10(NVMe盘不支持组RAID,有关RAID的配置方法请参考硬件厂家的手册或互联网上的方法进行配置,其中Disk Cache Policy一项需要设置为Disabled,否则机器异常掉电后有数据丢失的风险)。

要求数据盘总容量300GB以上(如果组RAID10,需要确保组RAID10后的总容量大于300GB),否则实例无法创建成功。

请保持所有磁盘(含系统盘、数据盘)类型一致,避免混用。

#### 须知

- 安装实例时,管理系统会将节点上除系统盘以外的磁盘都作为数据盘。
- 请提前清理数据盘,并保持为裸磁盘(如回显中的sdb、nvme0n1),请勿提前组数据盘。
- 如果数据盘里有NVMe盘,则优先使用NVMe盘;如果该节点既有SSD盘又有NVMe盘,则只会使用NVMe盘。
- 不支持NVMe系统盘。

如下所示磁盘为磁盘混用情况。

#### 须知

不支持系统盘多盘架构。例如: sdb上也规划了分区挂载uos-root。

```
[root@localhost ~]# lsblk
NAME
                MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda
                8:0 0 894.3G 0 disk
                8:1 0 600M 0 part /boot/efi
 -sda1
                8:2 0 1G 0 part /boot
8:3 0 892.7G 0 part
  -sda2
  -sda3
  -uos-root
                 253:0 0 1.8T 0 lvm /
                 253:1 0 4G 0 lvm
  -uos-swap
                8:16 0 894.3G 0 disk
sdb
                 8:17 0 894.3G 0 part
 -sdb1
                 253:0 0 1.8T 0 lvm /
  -uos-root
```

## 5.3.2 准备磁盘目录

步骤1 准备系统盘和数据盘。

建议系统盘由两块盘组成RAID1,主要为操作系统使用。

安装实例时程序自动识别除系统盘外所有磁盘作为数据盘使用,并自动挂载在/var/chroot子目录下,一般情况下无须手动执行挂载,所以在使用前请确保这些数据盘干净可用,执行以下命令查看并清理。

## **注意**

该操作会清空磁盘数据,在输入"y"前请先确认已备份或可删除数据,再进行操作。

#### pvcreate /dev/vdb

#### 回显如下:

~# pvcreate --test /dev/vdb

TEST MODE: Metadata will NOT be updated and volumes will not be (de)activated. Physical volume "/dev/vdb" successfully created.

以上回显则表明该磁盘已清空可用。

步骤2 添加主机前请确保无/var/chroot目录或/var/chroot目录下为空,注意,如果主机上有实例或者已经安装Agent,则不需要清空沙箱目录。

执行以下命令检查沙箱目录是否为空。

#### ls -Al /var/chroot

● 回显如下表示/var/chroot下为空,满足要求。

~# ll /var/chroot/ total 0

.....

或

~# ll /var/chroot/

ls: cannot access '/var/chroot/': No such file or directory

• 若不为空,执行以下命令进行清空。

#### rm -rf /var/chroot/{\*,.\*}

注意,如果出现"rm: cannot remove '\*\*\*' : Operation not permitted"回显,表示文件无法进行删除操作,可能是因为存在磁盘挂载,请将磁盘解挂后再重试清空沙箱目录。

清空后再次执行检查命令,确保沙箱目录不存在或为空。

#### ----结束

#### □ 说明

管理程序会自行组盘,并将逻辑卷挂载在/var/chroot路径下的数据目录、日志目录以及备份目 录等位置,目录文件系统格式为Ext4。

## 5.3.3 网络配置

#### 网络平面划分如下:

- 管理平面:云数据库GaussDB管理平台(TPOPS)与数据库节点的通信、数据库实例管理。
- 业务平面:业务访问数据库实例通信(分布式CN、主备版DN)。
- 数据平面: DN主备间通信、CN与DN间通信、备份恢复、DBMind纳管、主备实例间流式容灾通信。

轻量化部署形态支持管理、数据业务不同平面与共平面(即支持1~3IP地址),可基于业务安全、网络性能、机房条件等来选择单平面(单IP地址)、两平面(双IP地址)或三平面(三IP地址),一般基于业务安全、网络性能保障推荐使用三平面网络划分隔离。

## **注意**

- 1. 管理、业务、数据网络建议通过网络交换机划分网络平面进行物理隔离。
- 2. 对于管理IP地址,Agent进程启动时会查询默认网关并创建套接字连接默认网关,最后获取本地套接字的IP地址作为Agent的监听地址。不正确的网络配置会导致Agent无法提供服务。可通过**ip route show default**查看当前机器的默认网关,如果缺少相应路由,可通过**route add default** gw <*ip*>来添加默认路由。
- 3. 实例网络如果配置bond,请保证bond模式一致,不一致的bond配置可能导致实例工作异常。

# 5.3.4 检查 Python 依赖包

Agent依赖版本见<mark>依赖的Python库版本</mark>,如系统已经安装不在依赖范围内的Python依赖包,则在执行**添加主机**时会被自动覆盖。

系统当前的Python包版本可使用以下命令查看。

pip freeze --all

#### □ 说明

如果执行结果返回"-bash: pip: command not found",请先安装PIP工具。有关PIP工具的安装方法,请联系系统提供商支持或者参考**PIP官网**进行安装。

# 5.3.5 设置管理程序日志目录(可选)

### 使用建议

数据库管理程序日志默认打印在/home/Ruby/log下,并软链到/var/log下,建议为/var/log设置分区。

推荐单独为/var/log设置分区,如下:

```
~# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda 0:8 0 50G 0 disk
|_sda1 8:1 0 1G 0 part /boot
|_sda2 8:2 0 49G 0 part
|_klas-root 253:0 0 20G 0 lvm /
|_klas-log 253:1 0 1G 0 1vm /home
|_klas-log 253:3 0 10G 0 1vm /var/log
```

/var/log设置分区须在添加主机前完成。

如需将数据库管理程序日志设置到其他目录,可在<mark>添加主机</mark>完成后设置,以设置到/gauss/agent\_log目录为例:

## 操作步骤

步骤1 以root用户登录各主机。

步骤2 执行以下命令,修改/home/Ruby/log软链接到/gauss/agent\_log。

mkdir -m 700 -p /gauss/agent\_log

cd /home/Ruby

cp -fr log/\* /gauss/agent\_log

rm -fr log

ln -s /gauss/agent\_log /home/Ruby/log

chown -R Ruby:Ruby /gauss/agent\_log

步骤3 执行以下命令,重启进程生效。

ps -ef | grep 'python'| grep -v grep | awk '{print \$2}' | xargs kill -9

步骤4 执行以下命令,查看是否设置成功。

ls -l /home/Ruby/log

lrwxrwxrwx 1 root root 16 Jan 21 09:25 /home/Ruby/log -> /gauss/agent\_log

#### ls -ld /gauss/agent\_log

drwx----- 6 Ruby Ruby 4096 Jan 21 09:29 /gauss/agent\_log

回显如上则表示已修改成功。

----结束

# 5.3.6 准备浮动 IP (可选)

云数据库GaussDB管理平台(TPOPS)在创建主备版非单节点实例时提供了浮动IP功能,使用该功能需要准备一个可用的浮动IP,实例安装后可以通过浮动IP访问DN节点数据。如针对主备切换场景:主备版三节点设置了相同的浮动IP,主备切换后,仍然可以通过配置的浮动IP连接数据节点。

#### □ 说明

- 浮动IP功能需选择503.1.0.SPC1200、503.1.0.SPC1300版本或者503.2.0及以后版本的内核安装包。
- 配置浮动IP时需保证配置的IP有效、唯一;若IP无效,实例安装后无法通过浮动IP进行访问, 浮动IP地址必须和virtualIp使用同一网段。
- 实例安装后,浮动IP不支持修改。
- 在待安装实例节点ping浮动IP地址,ping不通表明该浮动IP地址未使用。

## 5.4 添加机房

### 操作场景

云数据库GaussDB管理平台(TPOPS)提供了数据中心管理的能力,使得用户可以添加机房。

### 前提条件

当前登录用户需要具备"添加机房"权限。

## 操作步骤

步骤1 登录云数据库GaussDB管理平台(TPOPS)。

步骤2 单击"平台管理 > 数据中心管理",进入"数据中心管理 > 机房管理"页面。

步骤3 单击页面右上方"添加机房"按钮。

图 5-1 添加机房

添加机房			
★ 机房别名			
机房描述			
		0/1	00
确定	取消		

表 5-3 添加机房参数解释

参数名称	是否 必填	解释
机房别名	是	数据中心管理列表中显示的机房别名。机房别名唯一且不超过 20个字符,可以包含英文、数字、中划线或者下划线,不能 包含其他特殊字符。
机房描述	否	描述信息不得超过100个字符。

步骤4 单击"确定",添加机房。

也可单击"取消",取消添加机房。

----结束

# 5.5 添加主机

## 5.5.1 使用须知

## 操作场景

云数据库GaussDB管理平台(TPOPS)提供了主机上线管理的能力,使得用户可以在创建实例、恢复新实例、扩分片等场景下,选择并使用相应主机创建数据库。

主机上线指的是这台主机经过操作系统安装、网络初始化、磁盘初始化后,已具备发放实例的条件,可将该主机纳管为可发放实例的主机。

## 前提条件

- 如果主机上已有实例,准备纳管实例到云数据库GaussDB管理平台(TPOPS), 请确保主机上线时实例正常。
- 当前登录用户需要具备"添加主机"权限。
- 需要保证各主机间的时钟源同步,如何设置时钟源,如何操作请参见设置时钟源。
- 闪存存储为白名单功能,特性白名单名称为: gaussdbv5\_feature\_supportDorado。如何开启白名单,请参见**如何开启/关闭白名单**。
- 主机上线支持root密码及SSH私钥两种方式,必须选择一种上线,推荐使用SSH私 钥方式。
- 如果使用SSH私钥作为主机上线的连接凭证,需要将公钥内容复制到~/.ssh/authorized\_keys中。如何操作,可参见《云数据库GaussDB管理平台(TPOPS)用户指南(for GaussDB轻量化部署形态)》中"平台管理>数据中心管理>密钥管理>下载公钥"章节。

## 约束限制

- 主机允许root用户登录。
- 支持批量导入主机,一次批量最多可导入50台主机。
- 主机允许所有云数据库GaussDB管理平台(TPOPS)节点SSH跳转,并进行scp命令拷贝安装包。具体操作可参见安装OpenSSH。
- 主机允许**expect**命令处理SSH交互过程。具体操作可参见<mark>检查Expect</mark>。
- 主机sshd\_config文件中将GSSAPIAuthentication参数设置为no。具体操作可参见 配置sshd\_config。
- 如果主机上没有实例,则需要主机沙箱目录为空,即/var/chroot目录不存在或为空。
- 主机umask回显小于等于0022。
- 同一主机不允许被多个云数据库GaussDB管理平台(TPOPS)重复添加。
- 如果主机上有实例,则添加主机时所选机房的名称需要和实例安装时使用的AZ名称保持一致。检查实例安装使用的AZ名称可参见8.6.4-检查实例安装使用的AZ名称。

## 5.5.2 添加单个主机

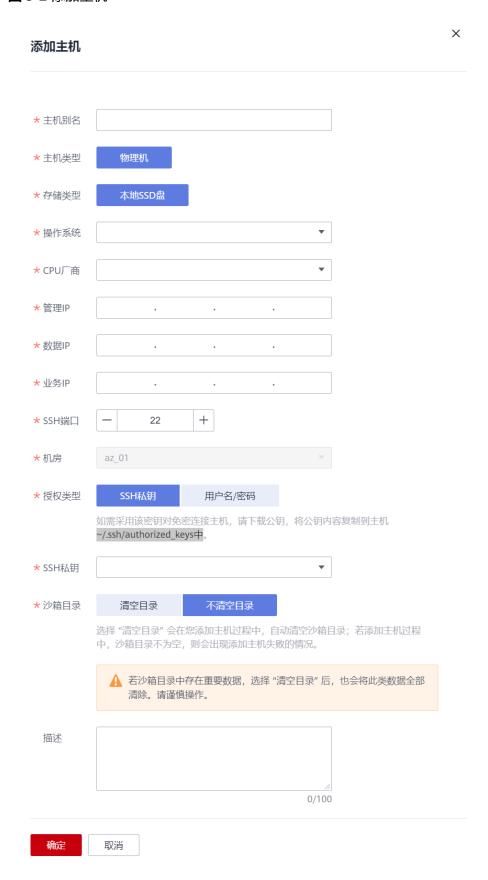
步骤1 登录云数据库GaussDB管理平台(TPOPS)。

步骤2 单击"平台管理 > 数据中心管理",进入"数据中心管理"页面。

步骤3 单击需要查询主机所在的"机房别名/ID"。

步骤4 单击"添加主机"。

## 图 5-2 添加主机



## 表 5-4 添加主机参数解释

参数名称	解释
主机别名	主机管理列表中显示的主机别名。主机别名不超过64个字 节。
主机类型	选择物理机。
存储类型	选择闪存存储或者本地SSD盘。
操作系统	选择麒麟或者统信。
CPU厂商	选择鲲鹏、英特尔或者海光。
管理IP	填写主机管理IP地址。 推荐使用三平面(三IP地址),即三张网卡承载三个平面的网络,若网卡上承载了不同平面网络,可能会导致不同平面间网络的物理隔离被破坏。 如果是单平面(单IP地址),存在网络未隔离的风险,可以使用iptables进行网络隔离。
数据IP	填写主机数据IP地址。 如果是单平面(单IP地址)或者两平面(双IP地址),则主 机数据IP地址即为主机管理IP地址。
业务IP	填写主机业务IP地址。 如果是单平面(单IP地址),则主机业务IP地址即为主机管 理IP地址。 如果是两平面(双IP地址),则主机业务IP地址为单独一个 网络平面,主机数据IP和主机管理IP共用一个网络平面。
SSH端口	端口范围为1-65535,默认为22。限制范围: 8002、 12017。
机房	默认选择当前机房,不可修改。
授权类型	支持SSH私钥和用户名/密码两种方式,推荐使用SSH私钥方式。
描述	描述不超过100个字符。
是否清空沙箱目录	<ul> <li>清空目录:会在您添加主机过程中,自动清空沙箱目录。若沙箱目录中存在重要数据,选择"清空目录"后,也会将此类数据全部清除。请谨慎操作。</li> <li>不清空目录:在添加主机过程中,不清空沙箱目录。若添加主机过程中,沙箱目录不为空,则会出现添加主机失败的情况。</li> </ul>
	的情况。

**步骤5** 单击"确定",添加主机。等待3~5分钟,主机添加成功。 也可单击"取消",取消添加主机。

## □ 说明

如果主机标准化检查未通过导致添加主机失败,则可通过主机标准化检查结果查看未通过的检查 项,并在重新配置主机后,重试任务流,再次对主机进行标准化检查。

主机标准化检查项及配置方法请参见"云数据库GaussDB管理平台(TPOPS)用户指南 > 附录 > 主机管理标准化检查项章节及主机管理标准化配置方法章节"。

## ----结束

## 5.5.3 批量导入主机

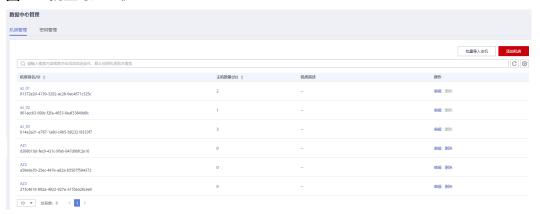
## 操作步骤

步骤1 登录云数据库GaussDB管理平台(TPOPS)。

步骤2 单击"平台管理 > 数据中心管理",进入"数据中心管理 > 机房管理"页面。

步骤3 单击"批量导入主机"。

图 5-3 批量导入主机



步骤4 单击"下载文件模板"。

## 图 5-4 下载文件模板



**步骤5** 完成导入主机文件模板下载并填写。模板中有**如下**字段。

表 5-5 文件模板参数解释

参数名称	参数意义	解释
name	主机别名	主机管理列表中显示的主机别名。不超过64个字节。
type	主机类型	选择物理机BMS。
os_type	操作系统	选择麒麟kylin或者统信uniontech。
cpu_spec	CPU类型	选择鲲鹏kunpeng、英特尔intel或者海光hygon。
storage_typ e	存储类型	选择闪存存储dorado或者本地SSD盘localssd。 闪存存储为白名单功能,请确认 gaussdbv5_feature_supportDorado白名单是否开 启。
manage_ip	管理IP	填写主机管理IP地址。 推荐使用三平面(三IP地址),即三张网卡承载三个平面的网络,若网卡上承载了不同平面网络,可能会导致不同平面间网络的物理隔离被破坏。 如果是单平面(单IP地址),存在网络未隔离的风险,可以使用iptables进行网络隔离。
data_ip	数据IP	填写主机数据IP地址。 如果是单平面(单IP地址)或者两平面(双IP地 址),则主机数据IP地址即为主机管理IP地址。

参数名称	参数意义	解释
virtual_ip	业务IP	填写主机业务IP地址。
		如果是单平面(单IP地址),则主机业务IP地址即为 主机管理IP地址。
		如果是两平面(双IP地址),则主机业务IP地址为单独一个网络平面,主机数据IP和主机管理IP共用一个网络平面。
ssh_port	SSH端口	端口范围为1-65535,限制范围:8002、12017。
data_center _id	机房ID	主机所属机房ID。
os_root_pw d	用户密码	主机root用户密码,如果使用root密码连接主机,则 需填写该字段,ssh_key_id字段为空。
ssh_key_id	SSH私钥	SSH私钥ID,如果使用SSH私钥连接主机,则需填写 该字段,os_root_pwd字段为空。推荐使用SSH私钥 方式。
description	描述	描述不超过100个字符。
is_force_rm _chroot_dir	是否清空沙 箱目录	选择"true"表示清空沙箱目录,选择"false"表示不清空。不填则默认选择"false"。
is_use_ssh	是否使用SSH	选择"true",表示使用SSH私钥或者用户名/密码添加主机。

步骤6 单击"导入&检查",上传填写的主机导入模板文件。

- 如果通过文件格式检查,则可单击"全部导入"按钮,批量导入主机。
- 如果未通过文件格式检查,则可查看格式错误,请修改文件后重新导入。

## 图 5-5 单击导入&检查



----结束

## 5.6 配置 NAS 服务器

## <u> 注意</u>

- 以下配置方式仅供参考,具体配置方式以NAS服务器厂商提供的为准。
- 配置NAS服务器用户映射为: all\_squash。
- 修改挂载目录所属用户和用户组为: nobody:nobody。
- 确保挂载目录最低权限为:700。
- 需确保实例节点和NAS服务器间网络可通。请确保未开启iptables服务或配置 iptables服务对实例节点网段开放NFS服务端口,默认端口号111。
- NAS文件系统共享目录创建必须以左斜杠开头,左斜杠之间字符必须为字母、数字。

步骤1 以root用户登录NAS服务器。

**步骤2** 执行以下命令,创建NAS文件系统共享目录(目录必须以左斜杠开头,左斜杠之间字符必须为字母、数字,目录不得超过255个字符)。

### mkdir -p /data/nas

**步骤3** 打开NFS配置文件/etc/exports,追加以下内容配置NAS服务器。其中,*{nasDir}*即为共享目录,*{ip}*为可以访问该共享目录的主机,括号中的内容为权限参数。

/{nasDir} {ip}(insecure,rw,sync,no\_subtree\_check,all\_squash)

配置参考示例如下:

/data/nas \*(insecure,rw,sync,no\_subtree\_check,all\_squash)

步骤4 执行以下命令,重启NFS使配置生效。

service nfs-server restart

## □说明

若提示找不到nfs-server服务,则说明当前服务器未安装NFS。可执行如下命令,安装NFS。yum源配置参考配置yum源。

yum install nfs-utils -y

步骤5 执行以下命令,修改挂载目录所属用户和用户组为: nobody:nobody。

chown nobody:nobody -R /data/nas

步骤6 执行以下命令,使修改生效。

exportfs -rv

**步骤7** 执行以下命令,确保挂载目录最低权限为700。

chmod 700 /data/nas

步骤8 执行以下命令,设置NAS服务自启动。

systemctl enable nfs-server

----结束

## 5.7 安装 GaussDB 实例

## 5.7.1 在本地磁盘安装实例

## 操作场景

使用云数据库GaussDB管理平台(TPOPS)安装实例,在本地磁盘缺省目录下安装元数据库已纳管版本的实例,实例支持8.0及以上版本。

## 前提条件

- 首次安装实例前,需要先申请试用许可容量。具体操作可参见《云数据库 GaussDB管理平台(TPOPS)用户指南 (for GaussDB轻量化部署形态)》中"平台管理 > License管理"相关章节。
- 如果数据节点原先使用非沙箱环境,请先手动清理节点。具体操作可参考《云数据库GaussDB管理平台(TPOPS)用户指南 (for GaussDB轻量化部署形态)》中"附录 > 转换安装模式前手动清理节点方法"相关章节。
- 如果数据节点在其他平台安装过数据库,请在安装前检查。具体操作可参考《云数据库GaussDB管理平台(TPOPS)用户指南 (for GaussDB轻量化部署形态)》中"常见问题 > 安装实例 > > 安装前置检查"相关章节。
- 如果要安装带浮动IP的主备版非单节点实例,需提前准备与业务IP同网段的浮动 IP,保证浮动IP唯一且有效。

## 约束限制

- 分布式五节点不支持128U1024G。
- GaussDB实例仅支持8U64G、16U128G、32U256G、64U512G、96U768G、128U1024G、196U1568G(仅DBmind)七档规格,请选择硬件规格大于或等于8U64G的主机。

## 操作步骤

步骤1 登录云数据库GaussDB管理平台(TPOPS)。

步骤2 单击"安装实例",进入"安装实例"界面。

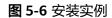




表 5-6 安装实例参数说明

参数名称	参数解释
实例名称	实例名称长度在4个到64个字符之间,必须以字母开头,可以包含字母、数字、中划线或下划线,不能包含其他特殊字符。
数据库引 擎版本	GaussDB实例支持8.0及以上版本。
操作系统	根据注册的主机情况进行展示。当前支持麒麟和统信两种系统。
架构类型	根据注册的主机情况进行展示。当前支持X86、ARM。
CPU厂商	根据注册的主机情况进行展示。当前支持鲲鹏、英特尔和海光。
主机类型	根据注册的主机情况进行展示。当前支持物理机类型。
实例类型	<ul><li>分布式版:分布式形态能够支撑较大的数据量,且提供了横向扩展的能力,可以通过扩容的方式提高实例的数据容量和并发能力。</li><li>主备版:适用于数据量较小且不会大幅度增长,对数据的可靠性、业务的可用性有一定诉求的场景。</li></ul>

参数名称	参数解释
部署形态	● 分布式版(默认): - 单节点:包含三个分片。 - 3节点:采用一主两备三节点的部署模式,包含三个分片。 - 4节点:采用一主一备四节点的混合部署模式,包含四个分片。 - 5节点:采用一主三备五节点的混合部署模式,包含四个分片和一个仲裁节点。 - 9节点(默认):采用一主三备九节点的混合部署模式。包含四个分片和一个仲裁节点。 ● 主备版: - 单节点:包含一个分片。 - 2节点(默认):采用一主一备一日志两节点的部署模式,其中日志节点和数据节点合并部署,包含一个分片。仅支持3.301及以上的GaussDB数据库引擎版本。 - 3节点:采用一主两备三节点的部署模式,包含一个分片。 - 5节点(1主3备1仲裁):采用一主三备五节点的部署模式,包含一个分片。 - 5节点(1主4备):采用一主四备五节点的部署模式,包含一个分片。 "单节点"需要开启白名单才能选择,特性白名单名称为:gaussdb_feature_supportSingleMode。如何操作请参见如何开启/关闭白名单。 8.102版本开始,主备版"2节点"、"3节点"、"5节点(1主3备1仲裁)"、"5节点(1主4备)"可使用去ETCD形态,开启特性白名单:gaussdb_feature_supportDccCluster即可启用。如何操作请参见如何开启/关闭白名单。
副本一致 性协议	该参数在"实例类型"配置为"主备版"时为必选参数。仅支持3.200 及以上的GaussDB数据库引擎版本。  • Quorum: Quorum模式主备同步复制机制。客户端发起事务后,必须要等待对应的WAL日志复制到多个副本后,主库才会响应给客户端,少数节点的宕机不影响全局可用性,保证数据的一致性。  • Paxos: Paxos模式解决了Quorum模式下日志复制过程中日志分叉的问题,提高日志复制吞吐量,增加DN自仲裁能力。 主备版"2节点"形态仅支持Paxos。 启用去ETCD形态后,主备版"2节点"、"3节点"、"5节点(1主3备1仲裁)"、"5节点(1主4备)"仅支持Paxos。

参数名称	参数解释
事务一致	该参数在"实例类型"配置为"分布式版"时为必选参数。
性	强一致性:应用更新数据时,用户都能查询到全部已经成功提交的数据,对性能有影响。
	最终一致性:应用更新数据时,用户查询到的数据可能不相同,有可能是更新后的值,也有可能是更新前的值,但经过一段时间后,查询到的数据是更新后的值,该种类型通常具有较高的性能。注意,不支持分布式事务强一致性的读操作;不支持insert into、select * from等依赖于查询结果的事务一致性;不支持拆分成多语句的写操作;不支持涉及多个节点执行的写操作;不支持搭建容灾。
切换策略	该参数在"实例类型"配置为"主备版"且"部署形态"配置为"2节点"时开放。
	<ul><li>数据高可靠:对数据一致性要求高的系统推荐选择数据高可靠,在 故障切换的时候优先保障数据一致性。该策略下,日志节点将部署 在备节点上。</li></ul>
	<ul><li>业务高可用:对业务在线时间要求高的系统推荐使用业务高可用, 在故障切换的时候优先保证数据库可用性。该策略下,日志节点将 部署在主节点上。</li></ul>
	默认选择为数据高可靠,实例安装后如需修改切换策略,可参见《 GaussDB轻量化部署形态 使用指南》中"云数据库GaussDB管理平台(TPOPS) 用户指南 (for GaussDB轻量化部署形态) > 具体实例管理 > 基本信息 > 修改切换策略"章节。
	<b>说明</b>   在业务高可用场景下需要谨慎修改如下数据库参数:
	recovery_time_target: 不当修改该参数会导致实例频繁进行强制切换,请 <del>联系技术支持</del> 指导修改。
	● audit_system_object:不当修改该参数会导致丢失DDL审计日志,请 <b>联系技术支持</b> 指导修改。
存储类型	根据注册的主机情况进行展示。当前支持本地SSD磁盘和闪存存储, 这里需要选择本地SSD磁盘。
是否支持 浮动IP	选择待安装实例是否支持浮动IP功能。该参数在"实例类型"配置为 "主备版"且"部署形态"不为"单节点"时展示。
	● 不支持: 不支持浮动IP功能。
	• 支持: 需要输入一个浮动IP,实例安装后可以通过浮动IP访问DN节点数据。如针对主备切换场景:主备版三节点设置了相同的浮动IP,通过浮动IP连接了数据节点,主备切换后,仍然可以通过配置的浮动IP连接数据节点。
	<b>说明</b>   ● 浮动IP功能需选择503.1.0.SPC1200、503.1.0.SPC1300版本或者503.2.0及以
	后版本的内核安装包。
	● 配置浮动IP时需保证配置的IP有效、唯一;若IP无效,实例安装后无法通过 浮动IP进行访问,浮动IP地址必须和virtualIp使用同一网段。
	● 实例安装后,浮动IP不支持修改。

参数名称	参数解释
参数模板	数据库参数模板就像是数据库引擎配置值的容器,参数模板中的参数可应用于一个或多个相同类型的数据库实例。实例创建成功后,参数模板可进行修改。 实例创建完成之后可以根据业务需要进行调整。
可用区	<ul> <li>● 分布式版:</li> <li>- 单节点仅支持部署在一个可用区。</li> <li>- 3节点支持部署在一个可用区。</li> <li>- 4节点仅支持部署在一个可用区。</li> <li>- 5节点和9节点仅支持部署在三个可用区。</li> <li>● 主备版:</li> <li>- 单节点仅支持部署在一个可用区。</li> <li>- 2节点仅支持部署在一个可用区。</li> <li>- 2节点仅支持部署在两个可用区。</li> <li>- 3节点支持部署在一个或者三个可用区。</li> <li>- 5节点(1主3备1仲裁)仅支持部署在三个可用区。</li> <li>- 5节点(1主4备)支持部署在一个或三个可用区。</li> <li>● 在选择完实例部署的可用区之后,9节点和5节点(1主3备1仲裁)形态支持自主选择主、备、仲裁可用区,并且主、备、仲裁可用区不能相同。3节点(主备版)仅支持选择主AZ(Available Zone,可用分区)。</li> <li>● 可用区指在同一区域下,电力、网络隔离的物理区域,可用区之间内网互通,不同可用区之间物理隔离。</li> </ul>
管理地址	用户已完成添加的主机,且只展示当前未被纳管的主机。
数据库端口	数据库对外开放的端口,默认为8000,可选范围为: 1024~39989。 限制端口为2378~2380、2400、4999~5001、5100、5500、 5999~6001、6009~6010、6500、8015、8097、8098、8181、 9090、9100、9180、9187、9200、12016、12017、20049、 20050、21731、21732、32122~32126、39001。

## 步骤3 填写NAS配置信息。

## 图 5-7 NAS 设备信息

存储设备 ras\_server\_single (默认设备) ▼ C 默认路径: /data/nas 请输入自定义路径 ②

## 表 5-7 NAS 设备信息参数说明

参数名称	参数解释
存储设备	可以选择对应的存储设备进行挂载,自定义路径为备份文件的存储路径。
	若当前平台未配置NAS设备,需要先在存储设备管理中完成存储设备 配置。

## 步骤4 填写数据库配置信息。

## 图 5-8 数据库信息



表 5-8 数据库信息参数说明

参数名称	参数解释
管理员账户名	数据库的登录名称默认为root。
管理员密码	请您输入高强度密码并定期修改,以提高安全性,防止出现密 码被暴力破解等安全风险。
	新密码符合如下复杂程度要求:
	● 密码不能少于8个字符且不超过32个字符。
	● 密码必须至少由如下四类字符中的三类组合而成:
	- 至少包含一个大写字母(A-Z )。
	- 至少包含一个小写字母(a-z )。
	- 至少包含一个数字字符(0-9)。
	- 至少包含1个特殊字符(~!@#%^*=+?,)。
	请妥善管理您的密码,因为系统将无法获取您的密码信息。
确认密码	必须和管理员密码相同。

步骤5 单击"立即申请",进入信息确认页,确认实例信息。

## 图 5-9 实例信息

产品规格	
实例名称	gauss-b958
数据库引擎	GaussDB
数据库版本	3,000
操作系统	麒麟
架构类型	X86
CPU厂商	英特尔
主机类型	物理机
实例类型	主备版
部署形态	2节点
存储类型	本地SSD盘
管理IP	Control of the Control
数据库端口	8000
副本一致性协议	Paxos
切换策略	数据高可靠
存储设备名称	default (默认设备)
存储设备默认路径	/data/nas/
NAS设备挂载	自动挂载
挂载路径	/home/nfs/6442d426-f56d-448e-92b8-7f2da5dc1d2c

- 如果需要重新选择实例规格,单击"上一步",回到上个页面修改实例信息。
- 如果规格确认无误,单击"提交",完成安装实例的申请。安装过程约10分钟, 请耐心等待。

步骤6 实例创建成功后,用户可以在"实例管理"页面对其进行查看和管理。

- 可依据产品类型、实例名称、关键字等信息进行查询。
- 单击实例名称后面的
   一,可修改已创建的实例名称。
- 单击实例名称下面的 <sup>□</sup> ,可复制已创建的实例ID。

## ----结束

## □ 说明

- 创建GaussDB实例过程中,状态显示为"创建中"。
- 在实例列表的右上角,单击刷新列表,可查看到创建完成的实例状态显示为"正常"。
- 数据库端口默认为8000,仅支持创建时设置,后期不可修改。
- 若已配置NAS服务器,实例创建完成后将自动开启"自动备份"功能,并会自动进行一次全量备份,用于记录实例的初始状态。

## 5.7.2 在 Dorado 存储池中安装实例

## 操作场景

使用云数据库GaussDB管理平台(TPOPS),将GaussDB实例安装到Dorado存储池上。

支持创建Dorado单实例GaussDB主备版一主两备实例。

该功能为白名单功能,使用前请开启白名单。如何开启请参见开启白名单。

## 前提条件

- 首次安装实例前,需要先申请试用许可容量。
- 当前版本仅支持沙箱环境安装Dorado单实例GaussDB主备版一主两备实例。
- 实例所有节点都完成了Dorado的lun划分,该操作由存储侧负责。
- 实例所有节点完成磁盘挂载,每个节点包括系统盘、数据盘、日志盘、本地备份集盘、ETCD盘需要进行初始化挂盘。如何进行磁盘挂载,请参见《云数据库GaussDB管理平台(TPOPS)用户指南(for GaussDB轻量化部署形态)》中"常见问题>安装实例>磁盘挂载要求"相关章节。
- 主机管理需成功录入实例所有节点,并标识存储类型为Dorado存储。
- 使用前请开启白名单,特性白名单名称为: gaussdbv5\_feature\_supportDorado。
   如何开启请参见开启白名单。
- 如果数据节点在其他平台安装过数据库,请在安装前检查。具体操作可参考《云数据库GaussDB管理平台(TPOPS)用户指南 (for GaussDB轻量化部署形态)》中"常见问题 > 安装实例 > 安装前置检查"相关章节。

## 操作步骤

- 步骤1 登录云数据库GaussDB管理平台(TPOPS)。
- 步骤2 单击"实例管理>安装实例",进入"安装实例"界面。
- 步骤3 安装Dorado实例的特有参数详见表5-9,其余参数与在本地磁盘安装实例一致。

## 图 5-10 安装实例

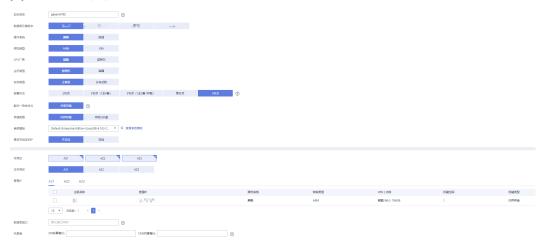


表 5-9 安装实例特殊参数说明

参数名称	参数选择
实例类型	主备版。
部署形态	3节点。
存储类型	根据注册的主机情况进行展示,这里需要选择闪存存储。
副本一致性协议	共享存储:共享存储模式基于云原生计算存储分离架构设计,支持 灵活弹性部署能力,可按需规划数据库容量,支持存储容量横向扩 展。利用Dorado全闪存存储卸载日志复制,实现同城RPO=0的双 集群容灾方案。
管理IP	选择"存储类型"为"闪存存储"的主机。
共享卷	由存储根据组网模式提供"DN共享卷id"和"CMS共享卷id"。  • 组网模式为NOF模式,共享卷id使用NGUID。  • 组网模式为FC模式,共享卷id使用WWN。 联系技术支持,获取"DN共享卷id"和"CMS共享卷id"。 说明  • DN共享卷一般取名为xlog共享卷。 • CMS共享卷一般取名为om共享卷。

步骤4 单击"立即申请",进入信息确认页,确认实例信息。

图 5-11 实例信息



步骤5 单击"提交",等待实例创建任务下发成功。

----结束

## 5.8 创建 DBMind 实例

## 操作场景

使用云数据库GaussDB管理平台(TPOPS)创建DBMind实例。

## 约束限制

当前仅支持创建银河麒麟 V10 ARM架构的DBMind实例。

## 操作步骤

步骤1 登录云数据库GaussDB管理平台(TPOPS)。

步骤2 单击"DBMind管理",进入"DBMind管理"页面。

步骤3 单击"创建DBMind实例",填写参数。

图 5-12 创建 DBMind 实例



表 5-10 基本信息参数说明

参数名称	参数解释
实例名称	实例名称长度在4个到64个字符之间,必须以字母开头,可以包含字母、数字、中划线或下划线,不能包含其他特殊字符。
版本	DBMind支持8.0及以上版本。
操作系统	根据DBMind版本绑定的操作系统进行展示,目前仅支持麒麟ARM。
资源类型	根据注册的主机情况进行展示。当前支持物理机类型。
架构类型	根据DBMind版本绑定的架构类型进行展示。
CPU厂商	当前区域下可用主机的所有CPU厂商。
管理地址	用户已完成注册的主机,且只展示当前可用的主机。

步骤4 单击"立即申请",进入信息确认页,确认DBMind实例信息。

图 5-13 DBMind 实例信息



- 如果需要重新选择DBMind实例规格,单击"上一步",回到上个页面修改 DBMind实例信息。
- 如果规格确认无误,单击"提交",完成创建DBMind实例的申请。

步骤5 DBMind实例创建成功后,用户可以在"DBMind管理"页面对其进行查看和管理。

## 图 5-14 DBMind 创建成功



- 可依据名称/ID、关键字信息进行查询。
- 单击DBMind实例名称下面的 □ ,可复制已创建的DBMind实例ID 。
- 单击具体DBMind实例的纳管数量,可显示当前该DBMind实例已纳管的"实例名称/ID"、"实例类型"、"运行状态"等。

单击已纳管实例具体"实例名称/ID",显示该已纳管实例的基本信息。

## 图 5-15 已纳管实例



## ----结束

# **6** 安装数据复制软件(可选)

如果您需要使用数据复制软件(Data Replication Software,简称DRS)进行数据在线迁移和数据库实时同步,请**下载**并安装所需版本的数据复制软件。

# **7** 卸载云数据库 GaussDB 管理平台 (TPOPS)

## 前提条件

如果卸载后需要重装云数据库GaussDB管理平台(TPOPS),请先将所有主机下线。

## 约束限制

若docker-service目录被删除后需要卸载云数据库GaussDB管理平台(TPOPS),请参照docker-service目录被删除的卸载步骤进行卸载,不需要参照本章节内容。

## 操作步骤

**步骤1** 以root用户登录云数据库GaussDB管理平台(TPOPS)任一节点。如需要配置免密卸载,可以参考**管理面节点之间互信设置**进行配置。

## □ 说明

- 操作步骤以/data/docker-service目录为例进行说明,操作时请替换为docker-service所在实际目录。
- 执行节点和远程节点(即除执行节点以外的节点)的目录结构及配置信息需保持一致,否则可能会导致卸载失败。

步骤2 执行以下命令, 打开参数配置文件。

vi /data/docker-service/config/user\_edit\_file.conf

步骤3 修改参数uninstall\_all为yes。

"uninstall all":"yes"

步骤4 按 "Esc"键后输入:wq!保存并关闭文件。

步骤5 执行以下命令,卸载云数据库GaussDB管理平台(TPOPS)。

cd /data/docker-service

sh appctl.sh uninstall\_all

回显如下,说明卸载成功。

[root@dbsnoname1 docker-service]# sh appctl.sh uninstall\_all start check node authentication

```
node not support auto authentication, will input root password
请输入root用户密码:
start check host: 192.168.0.1 root password
check host: 192.168.0.1 root password success
start check host: 192.168.0.2 root password
check host: 192.168.0.2 root password success
start check host: 192.168.0.3 root password
check host: 192.168.0.3 root password success
Start to init manifest...
init manifest successful for 192.168.0.1.
init manifest successful for 192.168.0.2.
init manifest successful for 192.168.0.3.
====== 192.168.0.1: gaussdb_service =======
               | complete
 gaussdb-console
                   | complete
 luban
                complete
 ots
                | complete
 GaussDB-open-api
                     complete
 GaussDB-instancemanager | complete
 GaussDB-backupmanager | complete
====== 192.168.0.2: gaussdb_service ======
                | complete
 gaussdb-console
                   complete
 luban
                | complete
 ots
                complete
 GaussDB-open-api | complete
 GaussDB-instancemanager | complete
 GaussDB-backupmanager | complete
====== 192.168.0.3: gaussdb_service =======
                complete
 auth
 gaussdb-console
                    | complete
                complete
 luban
               | complete
 GaussDB-open-api | complete
 GaussDB-instancemanager | complete
 GaussDB-backupmanager | complete
====== 192.168.0.1: docker_service ======
 common-service
                     | complete
 monitor-service
                    | complete
                    complete
 rds-ha-admin
                    | complete
 resource-manager
 workflow
                  | complete
====== 192.168.0.2: docker_service ======
 common-service
                    complete
 monitor-service
                    | complete
 rds-ha-admin
                    | complete
 resource-manager
                     | complete
 workflow
                  | complete
====== 192.168.0.3: docker_service ======
 common-service
                     | complete
 monitor-service
                    complete
                    complete
 rds-ha-admin
 resource-manager
                     | complete
 workflow
                  | complete
====== 192.168.0.1: Kafka =======
 kafka
                | complete
====== 192.168.0.2: Kafka ======
                | complete
====== 192.168.0.3: Kafka ======
                | complete
====== 192.168.0.1: Zookeeper =======
 zookeeper
               complete
 GaussDB-feature-data | complete
 GaussDB-data | complete
====== 192.168.0.2: Zookeeper =======
 zookeeper
              complete
 GaussDB-feature-data | complete
                  complete
 GaussDB-data
====== 192.168.0.3: Zookeeper ======
```

```
zookeeper | complete
 GaussDB-feature-data | complete
 GaussDB-data
                | complete
====== 192.168.0.1: PlatformData =======
platform-data | complete
======= 192.168.0.2: PlatformData =======
 platform-data | complete
 ====== 192.168.0.3: PlatformData =======
platform-data | complete
====== 192.168.0.1: CommonbaseData =======
 common-base | complete
====== 192.168.0.2: CommonbaseData ======
 common-base | complete
====== 192.168.0.3: CommonbaseData ======
 common-base
                   | complete
===== 192.168.0.1: base_enviornment ======
          | complete
 docker
 InfluxDB
                complete
sftp
gaussdb
              | complete
                complete
====== 192.168.0.2: base_enviornment =======
         | complete
 docker
 InfluxDB
                complete
gaussdb complete
                complete
====== 192.168.0.3: base_enviornment =======
 docker | complete
InfluxDB | complete
gaussdb | complete
                complete
                complete
====== 192.168.0.1: patch ======
patcn | complete
base_env '
                | complete
====== 192.168.0.2: patch ======
 patch | complete
 base_env
                 complete
====== 192.168.0.3: patch ======
 patch
           complete
 base_env
                 | complete
Uninstallation progress [72/72] ==> 100.00%
```

步骤6 执行以下命令,清理残存用户、目录及SFTP数据。

### 须知

清理命令会删除SFTP服务器中的所有数据,若SFTP数据仍要使用,请提前备份。 备份方式:将/opt/sftphome/目录下的所有内容复制出来即可。例如:**cp -r /opt/ sftphome /data/sftphome\_bak** 

## sh appctl.sh cleanup\_all

回显如下,说明清理成功。

```
[root@dbsnoname1 docker-service]# sh appctl.sh cleanup_all start check node authentication node not support auto authentication, will input root password 请输入root用户密码: start check host: 192.168.0.1 root password check host: 192.168.0.1 root password success start check host: 192.168.0.2 root password check host: 192.168.0.2 root password check host: 192.168.0.3 root password success start check host: 192.168.0.3 root password check host: 192.168.0.3 root password success Start to init manifest...
```

init manifest successful for 192.168.0.1. init manifest successful for 192.168.0.2. init manifest successful for 192.168.0.3. Cleanup 192.168.0.1 Success! Cleanup 192.168.0.2 Success! Cleanup 192.168.0.3 Success!

**步骤7** 以root用户登录云数据库GaussDB管理平台(TPOPS)所有节点,执行如下命令,删除docker-service目录。

rm -rf /data/docker-service

-----结束

## 8 常见问题

## 8.1 云数据库 GaussDB 管理平台(TPOPS)元数据库 GaussDB 安装报错解决方法

## 8.1.1 重新安装云数据库 GaussDB 管理平台(TPOPS)元数据库 GaussDB 的方法

## 问题现象

安装时出现如下报错。

```
docker | complete
InfluxDB | complete
sftp | complete
gaussdb | error!
```

## 问题原因

GaussDB安装报错不支持继续安装。

## 解决方法

步骤1 通过/tmp/install\_cluster.log查看GaussDB安装日志,确定具体的报错信息。

步骤2 联系技术支持人员,解决GaussDB问题。

步骤3 以root用户登录云数据库GaussDB管理平台(TPOPS)执行节点。

步骤4 执行以下命令,进入配置文件。

vi /data/docker-service/config/user\_edit\_file.conf

步骤5 修改"uninstall\_all"值为"yes",打开卸载的开关。

步骤6 按 "Esc"键后执行:wq!保存退出。

步骤7 执行以下命令,卸载云数据库GaussDB管理平台(TPOPS)。

cd /data/docker-service

sh appctl.sh uninstall\_all

其中/data/docker-service为执行安装目录。

步骤8 执行全量清理命令,清理安装残留。

sh appctl.sh cleanup\_all

步骤9 重新安装云数据库GaussDB管理平台(TPOPS)。具体操作可参见安装云数据库 GaussDB管理平台(TPOPS)。

----结束

## 8.1.2 omagent 安装失败

## 报错信息

报错日志文件所在位置: /tmp/install\_cluster.log

报错信息: check omagent failed in loaclhost: ...

## 报错原因

omagent安装失败。

## 操作步骤

步骤1 以root用户登录到云数据库GaussDB管理平台(TPOPS)执行节点。

步骤2 执行以下命令,获取Python3安装路径。

which python3

回显如下:

/usr/bin/python3

步骤3 执行以下命令,查看Python3的依赖。

## ldd /usr/bin/python3

回显如下:安装云数据库GaussDB管理平台(TPOPS)的所有节点的Python3须包含 加粗的依赖。

linux-vdso.so.1 (0x00007ffc7bb2a000)

libpython3.7m.so.1.0 => /lib64/libpython3.7m.so.1.0 (0x00007fb1ad8a4000)

libcrypt.so.1 => /lib64/libcrypt.so.1 (0x00007fb1ad869000) libpthread.so.0 => /lib64/libpthread.so.0 (0x00007fb1ad848000) libdl.so.2 => /lib64/libdl.so.2 (0x00007fb1ad843000)

libutil.so.1 => /lib64/libutil.so.1 (0x00007fb1ad83e000) libm.so.6 => /lib64/libm.so.6 (0x00007fb1ad6b9000)

libc.so.6 => /lib64/libc.so.6 (0x00007fb1ad4f8000)

/lib64/ld-linux-x86-64.so.2 (0x00007fb1adc40000)

- 若存在上述依赖,请执行后续步骤。
- 若不存在上述依赖,请使用yum源重新安装Python3。

步骤4 执行以下命令,查看磁盘空间。

df -h

回显如下: 加粗的代表根目录剩余可用的容量占比。

- 若磁盘空间不超过90%,请**联系技术支持**。
- 若磁盘空间超过90%,需要清理根目录空间。

步骤5 执行以下命令,查看是否有不为755权限的Python包。

ls -l /usr/local/lib/python3.7/site-packages

- 如果有:请执行以下命令,修改权限为755。
   chmod -R 755 /usr/local/lib/python3.7/site-packages
- 如果没有,请联系技术支持。

----结束

## 8.2 如何开启/关闭白名单

## 操作场景

本章节介绍如何开启/关闭特性白名单。

## 操作步骤

步骤1 以root用户登录云数据库GaussDB管理平台(TPOPS)主节点。

步骤2 执行以下命令,连接core库。

gsql -p 8635 -U core -W { password} -q core -h 127.0.0.1

其中{password}为core库连接密码。密码可参见《云数据库GaussDB管理平台(TPOPS)账户一览表 01》。

步骤3 执行以下命令,开启/关闭特性白名单。

UPDATE CORE.DBS\_FEATURE SET STATUS = '{open/closed}'

WHERE ID = (SELECT ID FROM CORE.DBS\_FEATURE WHERE NAME = '{featureName}' AND SITE = 'pcs-lite');

其中:

- {open/closed}: open表示开启、closed表示关闭。
- {featureName}为特性白名单名称。

步骤4 执行以下命令,确认特性白名单是否开启/关闭。

## SELECT STATUS FROM CORE.DBS\_FEATURE WHERE NAME = '{ featureName}' AND SITE = 'pcs-lite';

其中,回显为open表示开启、closed表示关闭。

步骤5 执行\q命令,退出core库。

----结束

## 8.3 如何关闭防火墙

步骤1 以root用户登录云数据库GaussDB管理平台(TPOPS)各个安装节点。

步骤2 在每个节点上均执行以下命令,关闭防火墙并禁止开机启动。

systemctl stop firewalld.service

systemctl disable firewalld.service

步骤3 使用vi打开config文件。

vi /etc/selinux/config

步骤4 修改"SELINUX"的值"permissive",按"Esc"键后执行:wq!保存并退出修改。

**SELINUX**=permissive

□说明

一般情况下,开启selinux会提高系统的安全性,但是可能导致程序无法运行。为保证安装顺利,建议用户设置值为permissive。

步骤5 执行以下命令,重新启动操作系统。

reboot

----结束

## 8.4 如何重新分发安装包

## 操作场景

本章介绍云数据库GaussDB管理平台(TPOPS)安装失败后,如何修改配置或者更换包之后重新安装。

## 操作步骤

步骤1 以root用户登录云数据库GaussDB管理平台(TPOPS)执行节点。

步骤2 执行以下命令,删除分发安装包标识文件。

rm -rf /data/docker-service/config/FIRST\_DISTRIBUTE\_ON

步骤3 执行以下命令,进入到安装脚本路径。

cd /data/docker-service

**步骤4** 执行以下命令,下发安装。

sh appctl.sh install

----结束

## 8.5 SFTP 安装失败

步骤1 根据回显确定是哪个节点的SFTP安装失败。

步骤2 以root用户登录上述SFTP安装失败的节点。

步骤3 执行以下命令,进入到日志目录查看报错日志。

cd /opt/cloud/logs/deploy

vi install.error.log

在日志中搜索SFTP相关内容,如果有报错内容为"Failed to set password for user",则说明安装环境中的pam规则不支持预置密码,需要执行后续步骤修改pam规则。

## □说明

/opt/cloud/logs目录为配置文件/data/docker-service/config/user\_edit\_file.conf中配置的 log\_path,请依据实际情况填写。

步骤4 执行以下步骤,修改system-auth文件。

vi /etc/pam.d/system-auth

找到pam\_deepin\_pw\_check.so,将pam\_deepin\_pw\_check.so修改为pam\_pwquality.so。

步骤5 按 "Esc"键后执行以下命令,保存并退出。

:wq!

**步骤6** 执行以下命令,进入安装目录。

cd /data/docker-service

步骤7 执行以下命令,再次下发安装。

sh appctl.sh install

----结束

□ 说明

若再次安装仍然报上述错误,请联系华为技术支持。

## 8.6 主机管理

## 8.6.1 添加主机失败

## 添加主机失败,提示"主机网络连接失败"

步骤1 检查主机管理IP是否填写正确。

- 如果主机管理IP填写正确,请查看后续步骤。
- 如果主机管理IP填写不正确,请执行如下命令删除主机后再重新添加主机。
  - 以root用户登录云数据库GaussDB管理平台(TPOPS)主节点。
  - 执行以下命令,连接core库。

gsql -p 8635 -U core -W {*password*} -q core -h 127.0.0.1 其中{password}为core库连接密码。

- 执行如下命令,删除主机。

UPDATE CORE.DBS\_HOST\_STATIC\_INFO SET STATUS='deleted' WHERE HOST\_ID={hostId};

UPDATE CORE.DBS\_HOST\_SPECIFICATION\_INFO SET STATUS='deleted'
WHERE ID={hostId};

其中{hostId}为主机ID。

执行如下命令,检查主机是否删除。

SELECT STATUS FROM CORE.DBS\_HOST\_STATIC\_INFO WHERE HOST\_ID={hostId};

SELECT STATUS FROM CORE.DBS\_HOST\_SPECIFICATION\_INFO WHERE ID ={hostId};

其中,回显为deleted表示主机已经删除。

- 执行\**q**命令,退出core库。

## 步骤2 检查主机是否损坏。

- 如果主机正常,请查看其他步骤。
- 如果主机损坏,请根据步骤1中的命令删除主机,待主机修复之后再重新添加主机。

步骤3 检查主机的网络配置和网络连接是否正确。

步骤4 检查主机防火墙是否允许ping请求通过。

步骤5 检查主机DNS设置和网卡设置是否正确。

----结束

## 添加主机失败,提示"主机 SSH 连接失败,检查密码或 sshd 配置"

步骤1 检查主机root用户密码和SSH端口号是否填写正确。

- 如果删除主机成功,表明添加主机时root用户密码或者SSH端口号填写错误,重新添加主机即可。
- 如果删除主机失败,请查看后续步骤。

步骤2 检查云数据库GaussDB管理平台(TPOPS)上的主机网络路由是否配置正确。

步骤3 检查主机SSH服务是否启动。

步骤4 检查主机SSHD配置是否正确。常见的SSHD配置如下所示。

PermitRootLogin yes

PasswordAuthentication yes

**GSSAPIAuthentication no** 

----结束

## 添加主机失败,提示"该主机已在其他平台上线"

步骤1 登录云数据库GaussDB管理平台(TPOPS)。

步骤2 单击"平台管理 > 数据中心管理",进入"数据中心管理"页面。

步骤3 检查主机是否在其他云数据库GaussDB管理平台(TPOPS)上已添加。

如果主机在其他云数据库GaussDB管理平台(TPOPS)上添加,请先在其他云数据库GaussDB管理平台(TPOPS)上删除主机后再添加主机。

----结束

## 添加主机失败,提示"下载 OS PATCH 安装包失败"

步骤1 登录云数据库GaussDB管理平台(TPOPS)。

步骤2 单击"平台管理 > 安装包管理",进入"安装包管理"页面。

步骤3 检查是否上传OS PATCH安装包。请在"任务中心"中查看该任务NebulaInitAgentJob的Context详情,以确认需要上传OS PATCH安装包的名称。

- 如果已经上传OS PATCH安装包,请查看后续步骤。
- 如果未上传OS PATCH安装包,请上传OS PATCH安装包,或者上传OS PATCH安装包到主机/dbs/osPatch目录下。

步骤4 检查SFTP服务器的服务是否正常。

步骤5 检查SFTP服务器的用户名和密码是否正确。

----结束

## 添加主机失败,提示"主机安装 agent 失败"

检查主机/dbs/osPatch/os\_patch\_all/路径下的build.log日志,根据日志信息进行修改。

## 添加主机失败,提示"网卡映射失败"

检查主机网卡配置是否正确。

## 添加主机失败,提示"主机标准化检查未通过"

查询主机标准化检查结果,若主机标准化检查未通过,则在主机重新配置后,重试任 务流,再次对主机进行标准化检查。

 如果EXPECT或者SFTP检查项未通过,请在主机重新配置后,重试 NebulaHostDetectionPrepareTask。 ● 如果其他检查项未通过,请在主机重新配置后,重试 NebulaHostDetectionTask。

## 添加主机失败,提示"请检查安装实例的 azName 是否和机房名称相同"

步骤1 查看任务失败原因,确定安装实例的AZ名称。

**步骤2** 查看是否存在和AZ名称同名的机房。如果该机房已存在,则继续执行**步骤3**。如果没有,则通过添加机房,新建一个同名机房。

步骤3 通过修改主机静态信息,修改主机所在机房。

步骤4 重试任务流。

----结束

## 8.6.2 删除主机失败

## 删除主机失败,提示"主机 SSH 连接失败,检查密码或 SSHD 配置"

步骤1 再次删除主机,检查主机root用户密码和SSH端口号是否填写错误。

- 如果删除主机成功,表明添加主机时root用户密码或者SSH端口号填写错误,重新添加主机即可。
- 如果删除主机失败,请查看其他解决方法。

步骤2 检查云数据库GaussDB管理平台(TPOPS)上的主机网络路由是否配置正确。

步骤3 检查主机SSH服务是否启动。

步骤4 检查主机SSHD配置是否正确。常见的SSHD配置如下所示。

PermitRootLogin yes

PasswordAuthentication ves

**GSSAPIAuthentication no** 

----结束

## 8.6.3 主机数据盘扩容

## 操作场景

如果在添加主机之后,对数据盘进行了扩容操作,请执行本章内容,更新主机数据盘大小。

本地SSD盘扩容可参考本地SSD盘扩容方法。

## 约束限制

- 主机状态为"初始化完成"、"待纳管"或者"使用中"。
- 新增的数据盘需要和主机上原有数据盘的磁盘IO类型一致。
- 如果主机上有沙箱类型的实例,则需将数据盘扩容到/var/chroot/var/lib/engine/ data\*路径下。如果主机上有非沙箱类型的实例,则需将数据盘扩容到当前数据目 录。

可执行如下命令,判断主机上有何种类型的实例:

cat /dbs/om-agent/agent\_\*/common/public\_cloud.conf | grep dataDir 回显值dataDir如果以/var/chroot为开头则表示为沙箱类型,否则表示非沙箱类型。

● 实例下的所有节点需要同时进行数据盘扩容,并且扩容的大小要保持一致,如果 不一致可能会影响实例及相关功能的正常使用。

## 操作步骤

步骤1 登录云数据库GaussDB管理平台(TPOPS)。

步骤2 查询主机列表,查看需要更新数据盘大小的主机,记录主机ID为{hostId},并记录主机 状态和存储类型。如果主机状态为使用中,则记录节点ID为{nodeId}。

步骤3 以root用户登录云数据库GaussDB管理平台(TPOPS)主节点。

步骤4 执行以下命令,连接core库。

gsql -p 8635 -U core -W {password} -q core -h 127.0.0.1

其中{password}为core库连接密码。

步骤5 执行如下命令,更新主机数据盘大小。

UPDATE CORE.DBS\_HOST\_STATIC\_INFO SET DATA\_DISK={dataDisk} WHERE HOST ID={hostId};

其中{dataDisk}为主机数据盘大小,单位GB。以新增数据盘vdc进行扩容为例,扩容前数据盘大小为300GB,扩容后数据盘大小为400GB。

```
[root@host-192-168-1-106 ~]# lsblk
NAME
                         MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
vda
                          252:0
                                        40G 0 disk
                                 0 0
⊢vda1
−vda2
                         252:1
                                        1G 0 part /boot
                         252:2
                                        39G 0 part
                                        35G 0 lvm /
  —klas-root
                         253:0
  ∟klas-swap
                         253:1 0
                                        4G 0 lvm [SWAP]
                         252:16 0 300G 0 disk 扩容前为300GB
                         253:2 0 200G 0 lvm /var/chroot/var/lib/engine/datal
253:3 0 20G 0 lvm /var/chroot/var/lib/log/backup
253:4 0 64G 0 lvm /var/chroot/usr/local/etcd
—gaussdbvg-mydatal
  -gaussdbvg-backupdata 253:3
                                        64G 0 lvm /var/chroot/usr/local/etcd
 -gaussdbvg-etcddata
                          252:32 0 100G 0 disk 扩容后为300+100=400GB
```

步骤6 如果主机状态为"使用中","存储类型"为"本地SSD盘",计算并更新节点磁盘大小。

● 执行如下命令,计算节点磁盘大小,记录为{data}。

SELECT ROUND(({dataDisk}\*0.95-SUM(v1.SIZE\_IN\_BYTES)/1000000000)/
(SELECT COUNT(\*) FROM DBS\_VOLUME v2 JOIN DBS\_PARENTSHIP p2 ON v2.ID=p2.ENTITY\_ID WHERE p2.PARENT\_ENTITY\_ID = {nodeld} AND p2.ENTITY\_TYPE\_TAG = 'vo' AND v2.PURPOSE = 'DATA')/40,
0)\*40\*1000000000 AS RESULT FROM DBS\_PARENTSHIP p1 LEFT JOIN DBS\_VOLUME v1 ON p1.ENTITY\_ID = v1.ID AND v1.PURPOSE in ('ETCD', 'BACKUP') WHERE p1.PARENT\_ENTITY\_ID = {nodeld} AND p1.ENTITY\_TYPE\_TAG = 'vo';

执行如下命令,更新节点磁盘大小。

UPDATE DBS\_VOLUME SET UPDATED\_AT=NOW(), SIZE\_IN\_BYTES={data} WHERE ID in (SELECT ID FROM DBS\_VOLUME WHERE ID IN (SELECT

## ENTITY\_ID FROM DBS\_PARENTSHIP WHERE PARENT\_ENTITY\_ID = {nodeld} AND ENTITY\_TYPE\_TAG = 'vo') AND PURPOSE = 'DATA');

**步骤7** 如果主机状态为"使用中","存储类型"为"闪存存储",执行如下命令,更新节点磁盘大小。

UPDATE DBS\_VOLUME SET UPDATED\_AT=NOW(), SIZE\_IN\_BYTES=
ROUND({dataDisk}\*0.95\*1024\*1024, 0) WHERE ID in (SELECT ID FROM
DBS\_VOLUME WHERE ID IN (SELECT ENTITY\_ID FROM DBS\_PARENTSHIP
WHERE PARENT\_ENTITY\_ID = {nodeld} AND ENTITY\_TYPE\_TAG = 'vo') AND
PURPOSE = 'DATA');

步骤8 执行\q命令,退出core库。

步骤9 登录云数据库GaussDB管理平台(TPOPS)。

**步骤10** 查询主机列表,查看主机的数据盘大小是否更新。如果没有更新,请检查**步骤5**执行是 否成功。

**步骤11** 如果主机状态为使用中,查看主机上实例的存储空间大小是否更新。如果没有更新, 请检查步骤6或者步骤7执行是否成功。

----结束

## 8.6.4 检查实例安装使用的 AZ 名称

## 操作场景

本章节介绍有实例的主机怎么检查实例安装使用的AZ名称。

## 操作步骤

步骤1 以root用户登录主机。

步骤2 执行以下命令,切换数据库用户并更新环境变量,以Ruby用户为例。

su - Ruby

source ~/gauss\_env\_file

步骤3 执行以下命令,检查实例安装使用的AZ名称。

## cm\_ctl query -CvzALL

```
[Ruby@host-192-168-1-90 ~]$ cm_ctl query -CvzALL
node
test 1 192.168.1.184 1
test 2 192.168.1.75 2
test 3 192.168.1.90 3
[ ETCD State ]
node instance state
test 1 192.168.1.184 7001 StateFollow
test 2 192.168.1.75 7002 StateFollow
test 3 192.168.1.90 7003 StateLeader
                                   StateEnllower
[ Cluster State ]
cluster_state : Normal redistributing : No balanced : Yes
balanced : Yes
current_az : AZ_ALL
[ Datanode State ]
                        instance state
                                                      I node
                                                                                  instance state
                                                                                                               node
test 1 192.168.1.184 6001 P Primary Normal | test 2 192.168.1.75 6002 S Standby Normal | test 3 192.168.1.90 6003
                                                                                                                                                     S Standby Normal
```

回显如上图所示,实例安装使用的AZ名称为"test"。

----结束

## 8.7 麒麟系统 audit 服务内存泄露占用大量内存

## 现象描述

本章节介绍系统特定audit版本存在内存泄漏风险,会导致audit服务占用大量内存,用户可按需进行升级audit操作。

## 操作步骤

步骤1 以root用户登录管理面机器。

步骤2 执行如下命令,查看audit服务版本。

rpm -qa audit

[root@dbsnoname2 test]# rpm -qa audit
audit-3.0-5.se.06.ky10.aarch64

- 若为audit-3.0-5.se.06版本,则需要进行升级。
- 若非audit-3.0-5.se.06版本,忽略步骤3~步骤5。

步骤3 参考修复指南,获取audit-3.0-5.se.08.ky10版本包(x86系统下载网站为: x86系统下载地址,arm系统下载网站为: arm系统下载地址),需要下载如下几个rpm包。

python3-audit-3.0-5.se.08.ky10.\*.rpm

audit-libs-3.0-5.se.08.ky10.\*.rpm

audit-3.0-5.se.08.ky10.\*.rpm

**步骤4** 将<mark>步骤3</mark>下载的rpm包传到任一临时目录,之后进入该目录,并执行如下命令,升级组件。

rpm -Uvh \*.rpm

步骤5 执行以下命令, 重启服务。

systemctl daemon-reload

systemctl start auditd.service

**步骤6** 执行以下命令,查看服务是否正常。

systemctl status auditd.service

```
[root@localhost BUILD]# systemctl status auditd.service

auditd.service - Security Auditing Service

Loaded: loaded (/usr/lib/systemd/system/auditd.service; enabled; vendor prespactive; active (running) since Thu 2022-01-06 13:41:14 CST; 15h ago

Docs: man:auditd(8)

https://github.com/linux-audit/audit-documentation

Process: 930 ExecStart=/sbin/auditd (code=exited, status=0/SUCCESS)

Process: 943 ExecStartPost=/sbin/augenrules --load (code=exited, status=0/SUCPMain PID: 937 (auditd)

Tasks: 5

Memory: 4.9M

CGroup: /system.slice/auditd.service

-947 /sbin/auditd

-940 /usr/sbin/sedispatch
-941 /bin/audisp-secaudit
```

显示"active (running)"表示服务正常。

步骤7 执行如下命令,查看audit服务版本。

rpm -qa audit

```
[root@dbsnoname2 ~]# rpm -qa audit
audit-3.0-5.se.08.ky10.aarch64
```

显示audit-3.0-5.se.08.ky10.\*则表示升级成功。

----结束



## **注意**

以下动作需以root用户进行操作,操作完成后请及时登出root用户,避免误操作。

# A.1 安装配置文件参考

本节描述云数据库GaussDB管理平台(TPOPS)安装配置文件中可用的配置参数。

步骤1 以root用户执行以下命令,打开软件包所在目录。

cd /data/docker-service/config

步骤2 打开user\_edit\_file.conf文件进行参数设置。

vi user\_edit\_file.conf

下表提供了参数详细信息。

表 A-1 参数

配置项	描述
ssh_port	节点之间SSH通信的端口号,默认为22。
influxDB_install_ip1	InfluxDB安装节点1。
influxDB_install_ip2	InfluxDB安装节点2。
sftp_install_ip1	SFTP安装节点1。
sftp_install_ip2	SFTP安装节点2。
node1_ip	云数据库GaussDB管理平台(TPOPS)元数据库 IP地址1。
node2_ip	云数据库GaussDB管理平台(TPOPS)元数据库 IP地址2。

配置项	描述
node3_ip	云数据库GaussDB管理平台(TPOPS)元数据库 IP地址3。
main_path	微服务运行目录。
log_path	日志目录。
sftp_path	SFTP安装目录。
gauss_path	元数据库安装目录。
influx_path	InfluxDB安装目录。
docker_path	Docker安装目录。
backup_path	备份目录。
node1_ip2	云数据库GaussDB管理平台(TPOPS)元数据库 IP地址1对应的可以和GaussDB实例通信的IP地 址,要求可以和GaussDB实例SSH通以及Ping 通。
node2_ip2	云数据库GaussDB管理平台(TPOPS)元数据库IP地址2对应的可以和GaussDB实例通信的IP地址,要求可以和GaussDB实例SSH通以及Ping通。
node3_ip2	云数据库GaussDB管理平台(TPOPS)元数据库 IP地址3对应的可以和GaussDB实例通信的IP地 址,要求可以和GaussDB实例SSH通以及Ping 通。
service_group_id	Service用户的组ID。
service_user_id	Service用户的用户ID。
uninstall_all	控制卸载的时候是否全量卸载的开关。
use_cgroup	控制与DRS合部时使用cgroup进行资源隔离的开 关。

### 山 说明

- 配置文件中不允许出现中文路径。
- 安装过程中不允许修改服务的IP地址和端口号。

## ----结束

# A.2 前置检查报错处理

### □ 说明

对于前置检查中的报错信息,可参考以下对应步骤进行配置。

1. 在云数据库GaussDB管理平台(TPOPS)所有节点上,通过如下命令检查是否安装了JRE。

### java -version

#### 回显如下:

openjdk version "1.8.\*"

OpenJDK Runtime Environment Bisheng (build 1.8.\*)

OpenJDK 64-Bit Server VM Bisheng (build 25.\*, mixed mode)

若没有出现如上回显,或提示命令不存在,请参考安装JRE进行安装及配置JRE。

2. 在云数据库GaussDB管理平台(TPOPS)所有节点上,通过如下命令检查是否安装了Expect。

### expect -v

#### 回显如下:

expect version 5.45.4

若没有出现expect版本,或者提示命令不存在,请参考安装Expect进行安装。

3. 在云数据库GaussDB管理平台(TPOPS)所有节点上,通过如下命令检查是否安装了openssl工具。

### openssl version

OpenSSL 1.1.1f 31 Mar 2020

若没有出现openssl版本,或者提示命令不存在,需执行以下命令进行安装。配置yum源方法参考配置yum源。

### yum -y install openssl

4. 在管理面节点上,通过如下命令检查是否安装了dos2unix工具。

### dos2unix -V

若回显如下,则说明已经安装。

dos2unix 7.4.1 (2019-09-24)

With Unicode UTF-16 support.

With native language support.
With support to preserve the user and group ownership of files.

LOCALEDIR: /usr/share/locale

http://waterlan.home.xs4all.nl/dos2unix.html

若没有出现dos2unix版本,或提示提示命令不存在,请需执行以下命令进行安装。

#### yum -y install dos2unix

5. 安装用户需使用**locale**命令检查操作系统字符集,必须为en\_US.UTF-8。若不是, 执行以下命令进行修改或添加。

### vi /etc/sysconfig/i18n

修改LANG为en\_US.UTF-8, 执行source /etc/sysconfig/i18n即可。

#### □ 说明

部分机器没有 /etc/sysconfig/i18n,需执行以下命令进行修改

### vi /etc/locale.conf

修改LANG为en\_US.UTF-8, 执行source /etc/locale.conf即可。

6. 以root用户在所有云数据库GaussDB管理平台(TPOPS)管理节点,执行以下命令检查是否安装了net-tools。

#### ifconfig

回显为网卡信息即为已安装。若提示命令不存在,请参考安装NET-TOOLS。

7. 以root用户在所有云数据库GaussDB管理平台(TPOPS)管理节点,执行以下命令检查是否安装了libcgroup。

### rpm -qa | grep libcgroup

- 若回显如下,显示为libcgroup的版本和架构,则说明已安装。 libcgroup-0.41-23.ky10.aarch64
- 若无回显,说明没有安装libcgroup,则需要执行以下命令安装libcgroup。

### yum -y install libcgroup

8. 使用root用户登录待安装的云数据库GaussDB管理平台(TPOPS)所有节点,可执行以下命令,查看Python版本。

### python3 --version

回显如下,为python3的版本号:

Python 3.7.9

安装云数据库GaussDB管理平台(TPOPS)的所有节点必须安装Python 3.7以上版本。若版本非Python 3.7以上版本,需参考<mark>安装Python3</mark>,重新安装。

#### □ 说明

云数据库GaussDB管理平台(TPOPS)要求系统镜像自带Python3。本次安装环境为: Python3.7.9。

## A.3 开源软件列表

### 山 说明

开源软件所在的文件路径均为GaussDB\_OS\_PATCH\_\*\*\*\*\*\*.zip/os\_patch\_all目录。

## 银河麒麟 V10 操作系统 X86 架构 SP1 版本

表 A-2 kylin\_v10\_x86\_sp1

软件名称	文件路径	文件名
libbasicobjects.so.0	os_patch_all/os/ kylin_v10_x86_sp1/rpm	ding- libs-0.6.1-42.ky10.x86_64.rp m
libbasicobjects.so.0. 1.0	os_patch_all/os/ kylin_v10_x86_sp1/rpm	ding- libs-0.6.1-42.ky10.x86_64.rp m
libcollection.so.4	os_patch_all/os/ kylin_v10_x86_sp1/rpm	ding- libs-0.6.1-42.ky10.x86_64.rp m
libcollection.so.4.1.1	os_patch_all/os/ kylin_v10_x86_sp1/rpm	ding- libs-0.6.1-42.ky10.x86_64.rp m
libdhash.so.1	os_patch_all/os/ kylin_v10_x86_sp1/rpm	ding- libs-0.6.1-42.ky10.x86_64.rp m

软件名称	文件路径	文件名
libdhash.so.1.1.0	os_patch_all/os/ kylin_v10_x86_sp1/rpm	ding- libs-0.6.1-42.ky10.x86_64.rp m
libini_config.so.5	os_patch_all/os/ kylin_v10_x86_sp1/rpm	ding- libs-0.6.1-42.ky10.x86_64.rp m
libini_config.so.5.2.1	os_patch_all/os/ kylin_v10_x86_sp1/rpm	ding- libs-0.6.1-42.ky10.x86_64.rp m
libpath_utils.so.1	os_patch_all/os/ kylin_v10_x86_sp1/rpm	ding- libs-0.6.1-42.ky10.x86_64.rp m
libpath_utils.so.1.0.1	os_patch_all/os/ kylin_v10_x86_sp1/rpm	ding- libs-0.6.1-42.ky10.x86_64.rp m
libref_array.so.1	os_patch_all/os/ kylin_v10_x86_sp1/rpm	ding- libs-0.6.1-42.ky10.x86_64.rp m
libref_array.so.1.2.1	os_patch_all/os/ kylin_v10_x86_sp1/rpm	ding- libs-0.6.1-42.ky10.x86_64.rp m
libcom_err.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	e2fsprogs- devel-1.45.6-0.p01.ky10.x86 _64.rpm
libe2p.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	e2fsprogs- devel-1.45.6-0.p01.ky10.x86 _64.rpm
libext2fs.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	e2fsprogs- devel-1.45.6-0.p01.ky10.x86 _64.rpm
libss.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	e2fsprogs- devel-1.45.6-0.p01.ky10.x86 _64.rpm
libexpect.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	expect-5.45.4-3.ky10.x86_64. rpm
libexpect5.45.4.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	expect-5.45.4-3.ky10.x86_64. rpm
proxymech.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	gssproxy-0.8.0-11.ky10.x86_ 64.rpm

软件名称	文件路径	文件名
libkeyutils.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	keyutils-libs- devel-1.5.10-11.ky10.x86_64 .rpm
libkadm5clnt_mit.so .11	os_patch_all/os/ kylin_v10_x86_sp1/rpm	krb5-1.17-9.ky10.x86_64.rp m
libkadm5clnt_mit.so .11.0	os_patch_all/os/ kylin_v10_x86_sp1/rpm	krb5-1.17-9.ky10.x86_64.rp m
libkadm5srv_mit.so. 11	os_patch_all/os/ kylin_v10_x86_sp1/rpm	krb5-1.17-9.ky10.x86_64.rp m
libkadm5srv_mit.so. 11.0	os_patch_all/os/ kylin_v10_x86_sp1/rpm	krb5-1.17-9.ky10.x86_64.rp m
libgssapi_krb5.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	krb5- devel-1.17-9.ky10.x86_64.rp m
libgssrpc.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	krb5- devel-1.17-9.ky10.x86_64.rp m
libk5crypto.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	krb5- devel-1.17-9.ky10.x86_64.rp m
libkadm5clnt.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	krb5- devel-1.17-9.ky10.x86_64.rp m
libkadm5clnt_mit.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	krb5- devel-1.17-9.ky10.x86_64.rp m
libkadm5srv.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	krb5- devel-1.17-9.ky10.x86_64.rp m
libkadm5srv_mit.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	krb5- devel-1.17-9.ky10.x86_64.rp m
libkdb5.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	krb5- devel-1.17-9.ky10.x86_64.rp m
libkrad.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	krb5- devel-1.17-9.ky10.x86_64.rp m
libkrb5.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	krb5- devel-1.17-9.ky10.x86_64.rp m

软件名称	文件路径	文件名
libkrb5support.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	krb5- devel-1.17-9.ky10.x86_64.rp m
libcgroup.so.1	os_patch_all/os/ kylin_v10_x86_sp1/rpm	libcgroup-0.41-23.ky10.x86_ 64.rpm
libcgroup.so.1.0.41	os_patch_all/os/ kylin_v10_x86_sp1/rpm	libcgroup-0.41-23.ky10.x86_ 64.rpm
pam_cgroup.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	libcgroup-0.41-23.ky10.x86_ 64.rpm
libffi.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	libffi- devel-3.3-7.ky10.x86_64.rpm
libselinux.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	libselinux-devel-2.9- se.05.ky10.x86_64.rpm
libsepol.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	libsepol- devel-2.9-1.ky10.x86_64.rpm
libstdc++.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	libstdc++- devel-7.3.0-20190804.h30.k y10.x86_64.rpm
libverto-glib.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	libverto- devel-0.3.1-2.ky10.x86_64.rp m
libverto-libev.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	libverto- devel-0.3.1-2.ky10.x86_64.rp m
libverto-libevent.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	libverto- devel-0.3.1-2.ky10.x86_64.rp m
libverto.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	libverto- devel-0.3.1-2.ky10.x86_64.rp m
libnfsidmap.so.1	os_patch_all/os/ kylin_v10_x86_sp1/rpm	nfs- utils-2.4.2-2.ky10.x86_64.rp m
libnfsidmap.so.1.0.0	os_patch_all/os/ kylin_v10_x86_sp1/rpm	nfs- utils-2.4.2-2.ky10.x86_64.rp m
nsswitch.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	nfs- utils-2.4.2-2.ky10.x86_64.rp m

软件名称	文件路径	文件名
static.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	nfs- utils-2.4.2-2.ky10.x86_64.rp m
umich_ldap.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	nfs- utils-2.4.2-2.ky10.x86_64.rp m
libcrypto.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	openssl- devel-1.1.1f-2.ky10.x86_64.r pm
libssl.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	openssl- devel-1.1.1f-2.ky10.x86_64.r pm
libpcre2-16.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	pcre2- devel-10.33-2.ky10.x86_64.r pm
libpcre2-32.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	pcre2- devel-10.33-2.ky10.x86_64.r pm
libpcre2-8.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	pcre2- devel-10.33-2.ky10.x86_64.r pm
libpcre2-posix.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	pcre2- devel-10.33-2.ky10.x86_64.r pm
libpython3.7m.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	python3- devel-3.7.9-6.ky10.x86_64.rp m
_ctypes_test.cpytho n-37m-x86_64- linux-gnu.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	python3- devel-3.7.9-6.ky10.x86_64.rp m
_testbuffer.cpython- 37m-x86_64-linux- gnu.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	python3- devel-3.7.9-6.ky10.x86_64.rp m
_testcapi.cpython-3 7m-x86_64-linux- gnu.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	python3- devel-3.7.9-6.ky10.x86_64.rp m
_testimportmultiple .cpython-37m- x86_64-linux-gnu.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	python3- devel-3.7.9-6.ky10.x86_64.rp m

软件名称	文件路径	文件名
_tkinter.cpython-37 m-x86_64-linux- gnu.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	python3- devel-3.7.9-6.ky10.x86_64.rp m
_xxtestfuzz.cpython -37m-x86_64-linux- gnu.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	python3- devel-3.7.9-6.ky10.x86_64.rp m
libz.so	os_patch_all/os/ kylin_v10_x86_sp1/rpm	zlib- devel-1.2.11-17.1.ky10.x86_ 64.rpm

## 银河麒麟 V10 操作系统 X86 架构 SP2 版本

表 A-3 kylin\_v10\_x86\_sp2

软件名称	文件路径	文件名
libbasicobjects.so.0	os_patch_all/os/ kylin_v10_x86_sp2/rpm	ding- libs-0.6.1-42.ky10.x86_64.rp m
libbasicobjects.so.0.1	os_patch_all/os/ kylin_v10_x86_sp2/rpm	ding- libs-0.6.1-42.ky10.x86_64.rp m
libcollection.so.4	os_patch_all/os/ kylin_v10_x86_sp2/rpm	ding- libs-0.6.1-42.ky10.x86_64.rp m
libcollection.so.4.1.1	os_patch_all/os/ kylin_v10_x86_sp2/rpm	ding- libs-0.6.1-42.ky10.x86_64.rp m
libdhash.so.1	os_patch_all/os/ kylin_v10_x86_sp2/rpm	ding- libs-0.6.1-42.ky10.x86_64.rp m
libdhash.so.1.1.0	os_patch_all/os/ kylin_v10_x86_sp2/rpm	ding- libs-0.6.1-42.ky10.x86_64.rp m
libini_config.so.5	os_patch_all/os/ kylin_v10_x86_sp2/rpm	ding- libs-0.6.1-42.ky10.x86_64.rp m
libini_config.so.5.2.1	os_patch_all/os/ kylin_v10_x86_sp2/rpm	ding- libs-0.6.1-42.ky10.x86_64.rp m

软件名称	文件路径	文件名
libpath_utils.so.1	os_patch_all/os/ kylin_v10_x86_sp2/rpm	ding- libs-0.6.1-42.ky10.x86_64.rp m
libpath_utils.so.1.0.1	os_patch_all/os/ kylin_v10_x86_sp2/rpm	ding- libs-0.6.1-42.ky10.x86_64.rp m
libref_array.so.1	os_patch_all/os/ kylin_v10_x86_sp2/rpm	ding- libs-0.6.1-42.ky10.x86_64.rp m
libref_array.so.1.2.1	os_patch_all/os/ kylin_v10_x86_sp2/rpm	ding- libs-0.6.1-42.ky10.x86_64.rp m
libcom_err.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	e2fsprogs- devel-1.45.6-0.p01.ky10.x86_ 64.rpm
libe2p.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	e2fsprogs- devel-1.45.6-0.p01.ky10.x86_ 64.rpm
libext2fs.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	e2fsprogs- devel-1.45.6-0.p01.ky10.x86_ 64.rpm
libss.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	e2fsprogs- devel-1.45.6-0.p01.ky10.x86_ 64.rpm
libexpect.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	expect-5.45.4-3.ky10.x86_64. rpm
libexpect5.45.4.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	expect-5.45.4-3.ky10.x86_64. rpm
proxymech.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	gssproxy-0.8.3-1.ky10.x86_64 .rpm
libkeyutils.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	keyutils-libs- devel-1.6.3-1.ky10.x86_64.rp m
libkadm5clnt_mit.so. 12	os_patch_all/os/ kylin_v10_x86_sp2/rpm	krb5-1.18.2-1.ky10.x86_64.rp m
libkadm5clnt_mit.so. 12.0	os_patch_all/os/ kylin_v10_x86_sp2/rpm	krb5-1.18.2-1.ky10.x86_64.rp m
libkadm5srv_mit.so. 12	os_patch_all/os/ kylin_v10_x86_sp2/rpm	krb5-1.18.2-1.ky10.x86_64.rp m

软件名称	文件路径	文件名
libkadm5srv_mit.so. 12.0	os_patch_all/os/ kylin_v10_x86_sp2/rpm	krb5-1.18.2-1.ky10.x86_64.rp m
libgssapi_krb5.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	krb5- devel-1.18.2-1.ky10.x86_64.r pm
libgssrpc.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	krb5- devel-1.18.2-1.ky10.x86_64.r pm
libk5crypto.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	krb5- devel-1.18.2-1.ky10.x86_64.r pm
libkadm5clnt.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	krb5- devel-1.18.2-1.ky10.x86_64.r pm
libkadm5clnt_mit.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	krb5- devel-1.18.2-1.ky10.x86_64.r pm
libkadm5srv.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	krb5- devel-1.18.2-1.ky10.x86_64.r pm
libkadm5srv_mit.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	krb5- devel-1.18.2-1.ky10.x86_64.r pm
libkdb5.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	krb5- devel-1.18.2-1.ky10.x86_64.r pm
libkrad.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	krb5- devel-1.18.2-1.ky10.x86_64.r pm
libkrb5.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	krb5- devel-1.18.2-1.ky10.x86_64.r pm
libkrb5support.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	krb5- devel-1.18.2-1.ky10.x86_64.r pm
libcgroup.so.1	os_patch_all/os/ kylin_v10_x86_sp2/rpm	libcgroup-0.42.2-1.ky10.x86_ 64.rpm
libcgroup.so.1.0.42	os_patch_all/os/ kylin_v10_x86_sp2/rpm	libcgroup-0.42.2-1.ky10.x86_ 64.rpm
pam_cgroup.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	libcgroup-0.42.2-1.ky10.x86_ 64.rpm

软件名称	文件路径	文件名
libstdc++.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	libstdc++- devel-7.3.0-20190804.35.p02 .ky10.x86_64.rpm
libverto-glib.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	libverto- devel-0.3.1-2.ky10.x86_64.rp m
libverto-libev.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	libverto- devel-0.3.1-2.ky10.x86_64.rp m
libverto-libevent.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	libverto- devel-0.3.1-2.ky10.x86_64.rp m
libverto.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	libverto- devel-0.3.1-2.ky10.x86_64.rp m
libnfsidmap.so.1	os_patch_all/os/ kylin_v10_x86_sp2/rpm	nfs- utils-2.5.1-3.ky10.x86_64.rp m
libnfsidmap.so.1.0.0	os_patch_all/os/ kylin_v10_x86_sp2/rpm	nfs- utils-2.5.1-3.ky10.x86_64.rp m
nsswitch.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	nfs- utils-2.5.1-3.ky10.x86_64.rp m
regex.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	nfs- utils-2.5.1-3.ky10.x86_64.rp m
static.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	nfs- utils-2.5.1-3.ky10.x86_64.rp m
umich_ldap.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	nfs- utils-2.5.1-3.ky10.x86_64.rp m
libcrypto.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	openssl- devel-1.1.1f-4.p01.ky10.x86_ 64.rpm
libssl.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	openssl- devel-1.1.1f-4.p01.ky10.x86_ 64.rpm

软件名称	文件路径	文件名
_ctypes_test.cpython -37m-x86_64-linux- gnu.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	python3- devel-3.7.9-6.ky10.x86_64.rp m
_testbuffer.cpython- 37m-x86_64-linux- gnu.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	python3- devel-3.7.9-6.ky10.x86_64.rp m
_testcapi.cpython-3 7m-x86_64-linux- gnu.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	python3- devel-3.7.9-6.ky10.x86_64.rp m
_testimportmultiple. cpython-37m- x86_64-linux-gnu.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	python3- devel-3.7.9-6.ky10.x86_64.rp m
_tkinter.cpython-37 m-x86_64-linux- gnu.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	python3- devel-3.7.9-6.ky10.x86_64.rp m
_xxtestfuzz.cpython- 37m-x86_64-linux- gnu.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	python3- devel-3.7.9-6.ky10.x86_64.rp m
libpython3.7m.so	os_patch_all/os/ kylin_v10_x86_sp2/rpm	python3- devel-3.7.9-6.ky10.x86_64.rp m

# 银河麒麟 V10 操作系统 ARM 架构 SP1 版本

表 A-4 kylin\_v10\_arm\_sp1

软件名称	文件路径	文件名
libbasicobjects.so. 0	os_patch_all/os/ kylin_v10_arm_sp1/rpm	ding- libs-0.6.1-42.ky10.aarch64 .rpm
libbasicobjects.so. 0.1.0	os_patch_all/os/ kylin_v10_arm_sp1/rpm	ding- libs-0.6.1-42.ky10.aarch64 .rpm
libcollection.so.4	os_patch_all/os/ kylin_v10_arm_sp1/rpm	ding- libs-0.6.1-42.ky10.aarch64 .rpm
libcollection.so.4.	os_patch_all/os/ kylin_v10_arm_sp1/rpm	ding- libs-0.6.1-42.ky10.aarch64 .rpm

软件名称	文件路径	文件名
libdhash.so.1	os_patch_all/os/ kylin_v10_arm_sp1/rpm	ding- libs-0.6.1-42.ky10.aarch64 .rpm
libdhash.so.1.1.0	os_patch_all/os/ kylin_v10_arm_sp1/rpm	ding- libs-0.6.1-42.ky10.aarch64 .rpm
libini_config.so.5	os_patch_all/os/ kylin_v10_arm_sp1/rpm	ding- libs-0.6.1-42.ky10.aarch64 .rpm
libini_config.so.5. 2.1	os_patch_all/os/ kylin_v10_arm_sp1/rpm	ding- libs-0.6.1-42.ky10.aarch64 .rpm
libpath_utils.so.1	os_patch_all/os/ kylin_v10_arm_sp1/rpm	ding- libs-0.6.1-42.ky10.aarch64 .rpm
libpath_utils.so.1. 0.1	os_patch_all/os/ kylin_v10_arm_sp1/rpm	ding- libs-0.6.1-42.ky10.aarch64 .rpm
libref_array.so.1	os_patch_all/os/ kylin_v10_arm_sp1/rpm	ding- libs-0.6.1-42.ky10.aarch64 .rpm
libref_array.so.1.2.	os_patch_all/os/ kylin_v10_arm_sp1/rpm	ding- libs-0.6.1-42.ky10.aarch64 .rpm
libcom_err.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	e2fsprogs- devel-1.45.6-0.p01.ky10.a arch64.rpm
libe2p.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	e2fsprogs- devel-1.45.6-0.p01.ky10.a arch64.rpm
libext2fs.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	e2fsprogs- devel-1.45.6-0.p01.ky10.a arch64.rpm
libss.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	e2fsprogs- devel-1.45.6-0.p01.ky10.a arch64.rpm
libexpect.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	expect-5.45.4-3.ky10.aarc h64.rpm
libexpect5.45.4.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	expect-5.45.4-3.ky10.aarc h64.rpm

软件名称	文件路径	文件名
proxymech.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	gssproxy-0.8.0-11.ky10.aa rch64.rpm
libkeyutils.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	keyutils-libs- devel-1.5.10-11.ky10.aarc h64.rpm
libkadm5clnt_mit. so.11	os_patch_all/os/ kylin_v10_arm_sp1/rpm	krb5-1.17-9.ky10.aarch64. rpm
libkadm5clnt_mit. so.11.0	os_patch_all/os/ kylin_v10_arm_sp1/rpm	krb5-1.17-9.ky10.aarch64. rpm
libkadm5srv_mit. so.11	os_patch_all/os/ kylin_v10_arm_sp1/rpm	krb5-1.17-9.ky10.aarch64. rpm
libkadm5srv_mit. so.11.0	os_patch_all/os/ kylin_v10_arm_sp1/rpm	krb5-1.17-9.ky10.aarch64. rpm
libgssapi_krb5.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	krb5- devel-1.17-9.ky10.aarch64 .rpm
libgssrpc.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	krb5- devel-1.17-9.ky10.aarch64 .rpm
libk5crypto.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	krb5- devel-1.17-9.ky10.aarch64 .rpm
libkadm5clnt.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	krb5- devel-1.17-9.ky10.aarch64 .rpm
libkadm5clnt_mit. so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	krb5- devel-1.17-9.ky10.aarch64 .rpm
libkadm5srv.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	krb5- devel-1.17-9.ky10.aarch64 .rpm
libkadm5srv_mit. so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	krb5- devel-1.17-9.ky10.aarch64 .rpm
libkdb5.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	krb5- devel-1.17-9.ky10.aarch64 .rpm
libkrad.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	krb5- devel-1.17-9.ky10.aarch64 .rpm

软件名称	文件路径	文件名
libkrb5.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	krb5- devel-1.17-9.ky10.aarch64 .rpm
libkrb5support.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	krb5- devel-1.17-9.ky10.aarch64 .rpm
libcgroup.so.1	os_patch_all/os/ kylin_v10_arm_sp1/rpm	libcgroup-0.41-23.ky10.aa rch64.rpm
libcgroup.so.1.0.4	os_patch_all/os/ kylin_v10_arm_sp1/rpm	libcgroup-0.41-23.ky10.aa rch64.rpm
pam_cgroup.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	libcgroup-0.41-23.ky10.aa rch64.rpm
libffi.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	libffi- devel-3.3-7.ky10.aarch64.r pm
libselinux.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	libselinux-devel-2.9- se.05.ky10.aarch64.rpm
libsepol.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	libsepol- devel-2.9-1.ky10.aarch64.r pm
libstdc++.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	libstdc++- devel-7.3.0-20190804.h30 .ky10.aarch64.rpm
libverto-glib.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	libverto- devel-0.3.1-2.ky10.aarch6 4.rpm
libverto-libev.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	libverto- devel-0.3.1-2.ky10.aarch6 4.rpm
libverto- libevent.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	libverto- devel-0.3.1-2.ky10.aarch6 4.rpm
libverto.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	libverto- devel-0.3.1-2.ky10.aarch6 4.rpm
libnfsidmap.so.1	os_patch_all/os/ kylin_v10_arm_sp1/rpm	nfs- utils-2.4.2-2.ky10.aarch64. rpm
libnfsidmap.so.1.0 .0	os_patch_all/os/ kylin_v10_arm_sp1/rpm	nfs- utils-2.4.2-2.ky10.aarch64. rpm

软件名称	文件路径	文件名
nsswitch.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	nfs- utils-2.4.2-2.ky10.aarch64. rpm
static.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	nfs- utils-2.4.2-2.ky10.aarch64. rpm
umich_ldap.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	nfs- utils-2.4.2-2.ky10.aarch64. rpm
libcrypto.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	openssl- devel-1.1.1f-2.ky10.aarch6 4.rpm
libssl.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	openssl- devel-1.1.1f-2.ky10.aarch6 4.rpm
libpcre2-16.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	pcre2- devel-10.33-2.ky10.aarch6 4.rpm
libpcre2-32.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	pcre2- devel-10.33-2.ky10.aarch6 4.rpm
libpcre2-8.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	pcre2- devel-10.33-2.ky10.aarch6 4.rpm
libpcre2-posix.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	pcre2- devel-10.33-2.ky10.aarch6 4.rpm
libpython3.7m.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	python3- devel-3.7.9-6.ky10.aarch6 4.rpm
_ctypes_test.cpyt hon-37m- aarch64-linux- gnu.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	python3- devel-3.7.9-6.ky10.aarch6 4.rpm
_testbuffer.cpytho n-37m-aarch64- linux-gnu.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	python3- devel-3.7.9-6.ky10.aarch6 4.rpm
_testcapi.cpython -37m-aarch64- linux-gnu.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	python3- devel-3.7.9-6.ky10.aarch6 4.rpm

软件名称	文件路径	文件名
_testimportmulti ple.cpython-37m- aarch64-linux- gnu.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	python3- devel-3.7.9-6.ky10.aarch6 4.rpm
_tkinter.cpython- 37m-aarch64- linux-gnu.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	python3- devel-3.7.9-6.ky10.aarch6 4.rpm
_xxtestfuzz.cpyth on-37m-aarch64- linux-gnu.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	python3- devel-3.7.9-6.ky10.aarch6 4.rpm
libz.so	os_patch_all/os/ kylin_v10_arm_sp1/rpm	zlib- devel-1.2.11-17.1.ky10.aar ch64.rpm

# 银河麒麟 V10 操作系统 ARM 架构 SP2 版本

表 A-5 kylin\_v10\_arm\_sp2

软件名称	文件路径	文件名
libbasicobjects.so.0	os_patch_all/os/ kylin_v10_arm_sp2/rpm	ding- libs-0.6.1-42.ky10.aarc h64.rpm
libbasicobjects.so.0.1.	os_patch_all/os/ kylin_v10_arm_sp2/rpm	ding- libs-0.6.1-42.ky10.aarc h64.rpm
libcollection.so.4	os_patch_all/os/ kylin_v10_arm_sp2/rpm	ding- libs-0.6.1-42.ky10.aarc h64.rpm
libcollection.so.4.1.1	os_patch_all/os/ kylin_v10_arm_sp2/rpm	ding- libs-0.6.1-42.ky10.aarc h64.rpm
libdhash.so.1	os_patch_all/os/ kylin_v10_arm_sp2/rpm	ding- libs-0.6.1-42.ky10.aarc h64.rpm
libdhash.so.1.1.0	os_patch_all/os/ kylin_v10_arm_sp2/rpm	ding- libs-0.6.1-42.ky10.aarc h64.rpm
libini_config.so.5	os_patch_all/os/ kylin_v10_arm_sp2/rpm	ding- libs-0.6.1-42.ky10.aarc h64.rpm

软件名称	文件路径	文件名
libini_config.so.5.2.1	os_patch_all/os/ kylin_v10_arm_sp2/rpm	ding- libs-0.6.1-42.ky10.aarc h64.rpm
libpath_utils.so.1	os_patch_all/os/ kylin_v10_arm_sp2/rpm	ding- libs-0.6.1-42.ky10.aarc h64.rpm
libpath_utils.so.1.0.1	os_patch_all/os/ kylin_v10_arm_sp2/rpm	ding- libs-0.6.1-42.ky10.aarc h64.rpm
libref_array.so.1	os_patch_all/os/ kylin_v10_arm_sp2/rpm	ding- libs-0.6.1-42.ky10.aarc h64.rpm
libref_array.so.1.2.1	os_patch_all/os/ kylin_v10_arm_sp2/rpm	ding- libs-0.6.1-42.ky10.aarc h64.rpm
libcom_err.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	e2fsprogs- devel-1.45.6-1.ky10.aa rch64.rpm
libe2p.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	e2fsprogs- devel-1.45.6-1.ky10.aa rch64.rpm
libext2fs.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	e2fsprogs- devel-1.45.6-1.ky10.aa rch64.rpm
libss.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	e2fsprogs- devel-1.45.6-1.ky10.aa rch64.rpm
libexpect.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	expect-5.45.4-3.ky10.a arch64.rpm
libexpect5.45.4.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	expect-5.45.4-3.ky10.a arch64.rpm
proxymech.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	gssproxy-0.8.3-1.ky10. aarch64.rpm
libkeyutils.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	keyutils-libs- devel-1.6.3-1.ky10.aar ch64.rpm
libkadm5clnt_mit.so. 12	os_patch_all/os/ kylin_v10_arm_sp2/rpm	krb5-1.18.2-1.ky10.aar ch64.rpm
libkadm5clnt_mit.so. 12.0	os_patch_all/os/ kylin_v10_arm_sp2/rpm	krb5-1.18.2-1.ky10.aar ch64.rpm

软件名称	文件路径	文件名
libkadm5srv_mit.so.1	os_patch_all/os/ kylin_v10_arm_sp2/rpm	krb5-1.18.2-1.ky10.aar ch64.rpm
libkadm5srv_mit.so.1 2.0	os_patch_all/os/ kylin_v10_arm_sp2/rpm	krb5-1.18.2-1.ky10.aar ch64.rpm
libgssapi_krb5.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	krb5- devel-1.18.2-1.ky10.aa rch64.rpm
libgssrpc.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	krb5- devel-1.18.2-1.ky10.aa rch64.rpm
libk5crypto.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	krb5- devel-1.18.2-1.ky10.aa rch64.rpm
libkadm5clnt.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	krb5- devel-1.18.2-1.ky10.aa rch64.rpm
libkadm5clnt_mit.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	krb5- devel-1.18.2-1.ky10.aa rch64.rpm
libkadm5srv.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	krb5- devel-1.18.2-1.ky10.aa rch64.rpm
libkadm5srv_mit.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	krb5- devel-1.18.2-1.ky10.aa rch64.rpm
libkdb5.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	krb5- devel-1.18.2-1.ky10.aa rch64.rpm
libkrad.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	krb5- devel-1.18.2-1.ky10.aa rch64.rpm
libkrb5.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	krb5- devel-1.18.2-1.ky10.aa rch64.rpm
libkrb5support.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	krb5- devel-1.18.2-1.ky10.aa rch64.rpm
libcgroup.so.1	os_patch_all/os/ kylin_v10_arm_sp2/rpm	libcgroup-0.42.2-1.ky1 0.aarch64.rpm
libcgroup.so.1.0.42	os_patch_all/os/ kylin_v10_arm_sp2/rpm	libcgroup-0.42.2-1.ky1 0.aarch64.rpm

软件名称	文件路径	文件名
pam_cgroup.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	libcgroup-0.42.2-1.ky1 0.aarch64.rpm
libstdc++.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	libstdc++- devel-7.3.0-20190804. 35.p02.ky10.aarch64.r pm
libverto-glib.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	libverto- devel-0.3.1-2.ky10.aar ch64.rpm
libverto-libev.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	libverto- devel-0.3.1-2.ky10.aar ch64.rpm
libverto-libevent.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	libverto- devel-0.3.1-2.ky10.aar ch64.rpm
libverto.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	libverto- devel-0.3.1-2.ky10.aar ch64.rpm
libnfsidmap.so.1	os_patch_all/os/ kylin_v10_arm_sp2/rpm	nfs- utils-2.5.1-3.ky10.aarc h64.rpm
libnfsidmap.so.1.0.0	os_patch_all/os/ kylin_v10_arm_sp2/rpm	nfs- utils-2.5.1-3.ky10.aarc h64.rpm
nsswitch.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	nfs- utils-2.5.1-3.ky10.aarc h64.rpm
regex.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	nfs- utils-2.5.1-3.ky10.aarc h64.rpm
static.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	nfs- utils-2.5.1-3.ky10.aarc h64.rpm
umich_ldap.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	nfs- utils-2.5.1-3.ky10.aarc h64.rpm
libcrypto.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	openssl- devel-1.1.1f-4.p01.ky1 0.aarch64.rpm

软件名称	文件路径	文件名
libssl.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	openssl- devel-1.1.1f-4.p01.ky1 0.aarch64.rpm
libpython3.7m.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	python3- devel-3.7.9-6.ky10.aar ch64.rpm
_ctypes_test.cpython -37m-aarch64-linux- gnu.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	python3- devel-3.7.9-6.ky10.aar ch64.rpm
_testbuffer.cpython- 37m-aarch64-linux- gnu.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	python3- devel-3.7.9-6.ky10.aar ch64.rpm
_testcapi.cpython-37 m-aarch64-linux- gnu.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	python3- devel-3.7.9-6.ky10.aar ch64.rpm
_testimportmultiple. cpython-37m- aarch64-linux- gnu.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	python3- devel-3.7.9-6.ky10.aar ch64.rpm
_tkinter.cpython-37 m-aarch64-linux- gnu.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	python3- devel-3.7.9-6.ky10.aar ch64.rpm
_xxtestfuzz.cpython- 37m-aarch64-linux- gnu.so	os_patch_all/os/ kylin_v10_arm_sp2/rpm	python3- devel-3.7.9-6.ky10.aar ch64.rpm

# 统信操作系统 X86 架构

表 A-6 uniontech\_x86

软件名称	文件路径	文件名
libbasicobjects.so.0	os_patch_all/os/ uniontech_x86/rpm	ding- libs-0.6.1-42.uel20.x86_64 .rpm
libbasicobjects.so.0.1.0	os_patch_all/os/ uniontech_x86/rpm	ding- libs-0.6.1-42.uel20.x86_64 .rpm
libcollection.so.4	os_patch_all/os/ uniontech_x86/rpm	ding- libs-0.6.1-42.uel20.x86_64 .rpm

软件名称	文件路径	文件名
libcollection.so.4.1.1	os_patch_all/os/ uniontech_x86/rpm	ding- libs-0.6.1-42.uel20.x86_64 .rpm
libdhash.so.1	os_patch_all/os/ uniontech_x86/rpm	ding- libs-0.6.1-42.uel20.x86_64 .rpm
libdhash.so.1.1.0	os_patch_all/os/ uniontech_x86/rpm	ding- libs-0.6.1-42.uel20.x86_64 .rpm
libini_config.so.5	os_patch_all/os/ uniontech_x86/rpm	ding- libs-0.6.1-42.uel20.x86_64 .rpm
libini_config.so.5.2.1	os_patch_all/os/ uniontech_x86/rpm	ding- libs-0.6.1-42.uel20.x86_64 .rpm
libpath_utils.so.1	os_patch_all/os/ uniontech_x86/rpm	ding- libs-0.6.1-42.uel20.x86_64 .rpm
libpath_utils.so.1.0.1	os_patch_all/os/ uniontech_x86/rpm	ding- libs-0.6.1-42.uel20.x86_64 .rpm
libref_array.so.1	os_patch_all/os/ uniontech_x86/rpm	ding- libs-0.6.1-42.uel20.x86_64 .rpm
libref_array.so.1.2.1	os_patch_all/os/ uniontech_x86/rpm	ding- libs-0.6.1-42.uel20.x86_64 .rpm
libcom_err.so	os_patch_all/os/ uniontech_x86/rpm	e2fsprogs- devel-1.45.6-7.uel20.x86_ 64.rpm
libe2p.so	os_patch_all/os/ uniontech_x86/rpm	e2fsprogs- devel-1.45.6-7.uel20.x86_ 64.rpm
libext2fs.so	os_patch_all/os/ uniontech_x86/rpm	e2fsprogs- devel-1.45.6-7.uel20.x86_ 64.rpm
libss.so	os_patch_all/os/ uniontech_x86/rpm	e2fsprogs- devel-1.45.6-7.uel20.x86_ 64.rpm
libexpect.so	os_patch_all/os/ uniontech_x86/rpm	expect-5.45.4-5.uel20.x86 _64.rpm

软件名称	文件路径	文件名
libexpect5.45.4.so	os_patch_all/os/ uniontech_x86/rpm	expect-5.45.4-5.uel20.x86 _64.rpm
proxymech.so	os_patch_all/os/ uniontech_x86/rpm	gssproxy-0.8.3-1.uel20.x8 6_64.rpm
libkeyutils.so	os_patch_all/os/ uniontech_x86/rpm	keyutils-libs- devel-1.6.3-1.uel20.x86_6 4.rpm
libkadm5clnt_mit.so.1	os_patch_all/os/ uniontech_x86/rpm	krb5-1.18.2-5.uel20.x86_6 4.rpm
libkadm5clnt_mit.so.1 2.0	os_patch_all/os/ uniontech_x86/rpm	krb5-1.18.2-5.uel20.x86_6 4.rpm
libkadm5srv_mit.so.12	os_patch_all/os/ uniontech_x86/rpm	krb5-1.18.2-5.uel20.x86_6 4.rpm
libkadm5srv_mit.so.12.	os_patch_all/os/ uniontech_x86/rpm	krb5-1.18.2-5.uel20.x86_6 4.rpm
libgssapi_krb5.so	os_patch_all/os/ uniontech_x86/rpm	krb5- devel-1.18.2-5.uel20.x86_ 64.rpm
libgssrpc.so	os_patch_all/os/ uniontech_x86/rpm	krb5- devel-1.18.2-5.uel20.x86_ 64.rpm
libk5crypto.so	os_patch_all/os/ uniontech_x86/rpm	krb5- devel-1.18.2-5.uel20.x86_ 64.rpm
libkadm5clnt.so	os_patch_all/os/ uniontech_x86/rpm	krb5- devel-1.18.2-5.uel20.x86_ 64.rpm
libkadm5clnt_mit.so	os_patch_all/os/ uniontech_x86/rpm	krb5- devel-1.18.2-5.uel20.x86_ 64.rpm
libkadm5srv.so	os_patch_all/os/ uniontech_x86/rpm	krb5- devel-1.18.2-5.uel20.x86_ 64.rpm
libkadm5srv_mit.so	os_patch_all/os/ uniontech_x86/rpm	krb5- devel-1.18.2-5.uel20.x86_ 64.rpm
libkdb5.so	os_patch_all/os/ uniontech_x86/rpm	krb5- devel-1.18.2-5.uel20.x86_ 64.rpm

软件名称	文件路径	文件名
libkrad.so	os_patch_all/os/ uniontech_x86/rpm	krb5- devel-1.18.2-5.uel20.x86_ 64.rpm
libkrb5.so	os_patch_all/os/ uniontech_x86/rpm	krb5- devel-1.18.2-5.uel20.x86_ 64.rpm
libkrb5support.so	os_patch_all/os/ uniontech_x86/rpm	krb5- devel-1.18.2-5.uel20.x86_ 64.rpm
libcgroup.so.1	os_patch_all/os/ uniontech_x86/rpm	libcgroup-0.42.2-1.uel20.x 86_64.rpm
libcgroup.so.1.0.42	os_patch_all/os/ uniontech_x86/rpm	libcgroup-0.42.2-1.uel20.x 86_64.rpm
pam_cgroup.so	os_patch_all/os/ uniontech_x86/rpm	libcgroup-0.42.2-1.uel20.x 86_64.rpm
libffi.so	os_patch_all/os/ uniontech_x86/rpm	libffi- devel-3.3-8.uel20.x86_64.r pm
libstdc++.so	os_patch_all/os/ uniontech_x86/rpm	libstdc++- devel-7.3.0-20211123.43. uel20.x86_64.rpm
libverto-glib.so	os_patch_all/os/ uniontech_x86/rpm	libverto- devel-0.3.1-2.uel20.x86_6 4.rpm
libverto-libev.so	os_patch_all/os/ uniontech_x86/rpm	libverto- devel-0.3.1-2.uel20.x86_6 4.rpm
libverto-libevent.so	os_patch_all/os/ uniontech_x86/rpm	libverto- devel-0.3.1-2.uel20.x86_6 4.rpm
libverto.so	os_patch_all/os/ uniontech_x86/rpm	libverto- devel-0.3.1-2.uel20.x86_6 4.rpm
libnfsidmap.so.1	os_patch_all/os/ uniontech_x86/rpm	nfs- utils-2.5.1-4.uel20.x86_64. rpm
libnfsidmap.so.1.0.0	os_patch_all/os/ uniontech_x86/rpm	nfs- utils-2.5.1-4.uel20.x86_64. rpm

软件名称	文件路径	文件名
nsswitch.so	os_patch_all/os/ uniontech_x86/rpm	nfs- utils-2.5.1-4.uel20.x86_64. rpm
regex.so	os_patch_all/os/ uniontech_x86/rpm	nfs- utils-2.5.1-4.uel20.x86_64. rpm
static.so	os_patch_all/os/ uniontech_x86/rpm	nfs- utils-2.5.1-4.uel20.x86_64. rpm
umich_ldap.so	os_patch_all/os/ uniontech_x86/rpm	nfs- utils-2.5.1-4.uel20.x86_64. rpm
libcrypto.so	os_patch_all/os/ uniontech_x86/rpm	openssl- devel-1.1.1f-13.uel20.x86_ 64.rpm
libssl.so	os_patch_all/os/ uniontech_x86/rpm	openssl- devel-1.1.1f-13.uel20.x86_ 64.rpm
libpython2.7.so	os_patch_all/os/ uniontech_x86/rpm	python2- devel-2.7.18-1.uel20.x86_ 64.rpm
_ctypes_test.so	os_patch_all/os/ uniontech_x86/rpm	python2- devel-2.7.18-1.uel20.x86_ 64.rpm
libpython3.7m.so	os_patch_all/os/ uniontech_x86/rpm	python3- devel-3.7.9-18.uel20.x86_ 64.rpm
_ctypes_test.cpython-3 7m-x86_64-linux- gnu.so	os_patch_all/os/ uniontech_x86/rpm	python3- devel-3.7.9-18.uel20.x86_ 64.rpm
_testbuffer.cpython-37 m-x86_64-linux- gnu.so	os_patch_all/os/ uniontech_x86/rpm	python3- devel-3.7.9-18.uel20.x86_ 64.rpm
_testcapi.cpython-37m -x86_64-linux-gnu.so	os_patch_all/os/ uniontech_x86/rpm	python3- devel-3.7.9-18.uel20.x86_ 64.rpm
_testimportmultiple.cp ython-37m-x86_64- linux-gnu.so	os_patch_all/os/ uniontech_x86/rpm	python3- devel-3.7.9-18.uel20.x86_ 64.rpm

软件名称	文件路径	文件名
_tkinter.cpython-37m- x86_64-linux-gnu.so	os_patch_all/os/ uniontech_x86/rpm	python3- devel-3.7.9-18.uel20.x86_ 64.rpm
_xxtestfuzz.cpython-3 7m-x86_64-linux- gnu.so	os_patch_all/os/ uniontech_x86/rpm	python3- devel-3.7.9-18.uel20.x86_ 64.rpm

## 统信操作系统 ARM 架构

不携带开源软件

## A.4 安装 JRE

此处以麒麟JRE为例。yum配置方式详见配置yum源。

步骤1 以root用户执行以下命令,进行OpenJdk的安装。

yum install java-1.8.\*-openjdk -y

步骤2 执行以下命令,检查Java是否正确配置。

java -version

如果回显如下,正确输出版本号,则表示安装完成。

openjdk version "1.8.0\_\*" OpenJDK Runtime Environment Bisheng (build 1.8.\*) OpenJDK 64-Bit Server VM Bisheng (build 25.\*, mixed mode)

----结束

# A.5 安装 Python3

### 前提条件

环境需要提前配置yum源。yum源配置参考配置yum源。

## 操作步骤

步骤1 以root用户登录实例所在节点。

步骤2 执行如下命令,安装Python3。

yum -y install python3

----结束

# A.6 安装主机的 Python3

### 前提条件

环境需要提前配置yum源。yum源配置参考配置yum源。

请先安装Python3后再添加主机。如果主机标准化检查出Python3版本不通过,则需要 先删除主机,安装Python3后再添加主机。

## 操作步骤

**步骤1** 执行如下命令,安装依赖环境。

yum -y install make rng-tools perl libffi-devel sqlite-devel openssl-devel python3-devel gcc-c++ libcgroup libcgroup-tools

systemctl restart rngd

systemctl restart cgconfig

**步骤2** 安装需要版本的Python3,可使用yum源或者Python3安装包进行安装,以安装Python3.7.9为例。

- 使用yum源安装:请确定yum源中提供的Python3安装包版本是否为3.7.9,确定 后可执行yum -y install python3.7命令。
- 使用Python3安装包安装:以Python3.7.9.tgz安装包为例,执行如下命令。
  - a. 执行**mkdir -p /usr/local/python3**命令,创建python3文件夹,以 /usr/local/python3文件夹为例。
  - b. 执行tar -zxvf Python-3.7.9.tgz命令,解压Python-3.7.9.tgz安装包。
  - c. 执行cd Python-3.7.9命令,进入解压后的目录。
  - d. 执行如下命令,进行编译安装。

./configure --prefix=/usr/local/python3 --enable-optimizations -- enable-shared

make -sj && make install

步骤3 执行如下命令,修改软链接,以Python3的安装路径为/usr/local/python3为例。

ln -sf /usr/local/python3/lib/libpython3.7m.so.1.0 /usr/lib64/ libpython3.7m.so.1.0

ln -sf /usr/lib64/libpython3.7m.so.1.0 /usr/lib64/libpython3.7m.so

In -sf /usr/local/python3/lib/libpython3.so /usr/lib64/libpython3.so

ln -s /usr/lib64/libffi.so.6 /usr/lib64/libffi.so.7

rm -f /usr/bin/pip /usr/bin/pip3 /usr/bin/python /usr/bin/python3

ln -s /usr/local/python3/bin/pip /usr/local/bin/pip

ln -s /usr/local/python3/bin/pip3 /usr/local/bin/pip3

ln -s /usr/local/python3/bin/pip3.7 /usr/local/bin/pip3.7

ln -s /usr/local/python3/bin/python3 /usr/bin/python3

In -s /usr/bin/python3 /usr/bin/python

chmod -R 755 /usr/local/python3

chmod 755 -R /usr/local/lib64/python3.7

chmod 755 -R /usr/local/lib/python3.7

### 须知

如果出现"ln: failed to create symbolic link '\*\*\*': File exists"报错,跳过即可。

步骤4 执行如下命令,检查Python3是否安装成功。

python --version

如果回显显示Python 3.7.9,则表示安装成功。

----结束

# A.7 安装 Expect

### 前提条件

银河麒麟环境需要提前配置yum源。yum源配置参考配置yum源。

## 操作步骤

步骤1 执行如下指令,安装Expect软件。

yum install expect -y

步骤2 检查软件是否安装成功。

expect -v

----结束

# A.8 安装 NET-TOOLS

### 前提条件

银河麒麟环境需要提前配置yum源,配置方法可参见<mark>配置yum源</mark>。

### 操作步骤

步骤1 以root用户登录节点。

步骤2 执行如下指令,安装net-tools软件。

yum install net-tools -y

步骤3 检查软件是否安装成功。

### ifconfig

----结束

## A.9 磁盘挂载步骤

## 操作场景

本节描述安装云数据库GaussDB管理平台(TPOPS)过程中进行磁盘挂载的方法。安装节点均需要进行磁盘挂载。

## 前提条件

需提前规划7块独立的磁盘。各磁盘的大小可参考表4-6或表4-8。

本章以挂载/opt/cloud为例,其他目录参考以下操作步骤进行磁盘挂载。

## 操作步骤

步骤1 执行以下命令,查看磁盘信息。

### lsblk

### 回显如下:

```
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

sda 8:0 0 557.9G 0 disk

sda1 8:1 0 600M 0 part /boot/efi

sda2 8:2 0 1G 0 part /boot

sda3 8:3 0 556.3G 0 part

euleros-root 253:0 0 70G 0 lvm /

euleros-swap 253:1 0 4G 0 lvm

euleros-home 253:12 0 482.3G 0 lvm /home

sdb 8:16 0 7T 0 disk
```

以/dev/sdb为例,按照以下步骤进行磁盘挂载。

步骤2 执行如下命令,进行格式化。

mkfs.ext4 /dev/sdb

步骤3 执行以下命令查询所需挂载磁盘分区的UUID。

blkid /dev/sdb

回显如下:

/dev/sdb: UUID="\*" BLOCK\_SIZE="4096" TYPE="Y"

**步骤4** 执行以下命令,进入fstab文件并配置挂载参数(\*为刚查出来的UUID)。以/opt/cloud目录为例。

vi /etc/fstab

UUID=\* /opt/cloud Y defaults 1 2

#### □ 说明

其中第一个参数为刚查出来的UUID;第二个参数为需要被挂载的路径;第三个参数为blkid命令查出来的type值;defaults为挂载选项;最后两个分别为备份和检测选项。

步骤5 按"Esc"键后执行以下命令,保存并返回。

:wq!

步骤6 执行以下命令,使配置生效。

mount -a

----结束

# A.10 依赖的 Python 库版本

setuptools-57.0.0

greenlet-1.1.0

setuptools-scm-3.3.3

typing-extensions-3.10.0.0

zipp-3.4.1

importlib-metadata-4.6.0

SQLAlchemy-1.4.20

PyYAML-5.4.1

pyrsistent-0.18.0

attrs-21.2.0

jsonschema-3.2.0

kafka-python-2.0.2

dnspython-1.16.0

eventlet-0.31.0

protobuf-3.17.3

esdk-obs-python-3.20.11

portalocker-2.3.0

concurrent-log-handler-0.9.19

psutil-5.8.0

pexpect-4.8.0

pycparser-2.20

cffi-1.14.5

bcrypt-3.2.0

six-1.16.0

ptyprocess-0.7.0

PyNaCl-1.3.0

cryptography-3.3.1

paramiko-2.7.2

future-0.18.2

pkgconfig-1.5.1

lz4-3.1.3

# A.11 免密登录管理面节点 root 账户设置

## 操作场景

本章节指导如何使用密钥登录管理面root账户。

## 前提条件

客户可以密码登录管理面节点root账户,并同意生成私钥并下发。

## 操作步骤

步骤1 以root用户登录管理面任一节点。若为多节点,需在每个节点执行以下修改。

步骤2 执行以下命令, 打开下述文件。

vi /etc/ssh/sshd\_config

步骤3 查找如下两行配置。

PubkeyAuthentication yes

AuthorizedKeysFile .ssh/authorized\_keys

如没有或不一致,则添加或修改该配置,按"Esc"键后执行:wq!保存并退出。执行以下命令,重启SSHD服务。

systemctl restart sshd

● 若存在并与上述配置一致,继续执行后续步骤。

步骤4 执行以下命令,进入指定目录。

cd /root/.ssh

**步骤5** 执行如下命令,不输入任何值,按回车键,生成密钥对。

ssh-keygen

**步骤6** 在当前目录下执行下述命令,生成授权公钥文件。

touch authorized\_keys

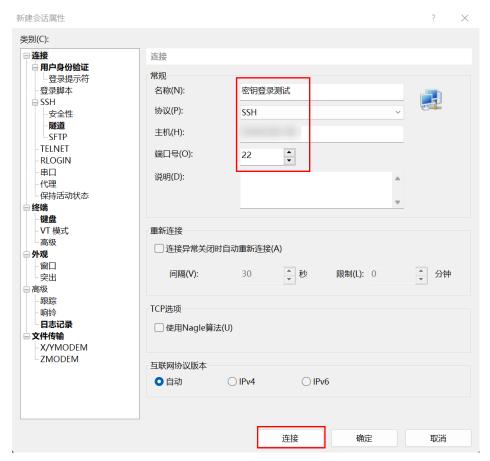
chmod 600 authorized\_keys

cat id\_rsa.pub >> authorized\_keys

步骤7 下载当前目录下生成的SSH私钥文件id\_rsa至本地。

步骤8 以Xshell为例,按下述流程配置密钥登录管理面节点。

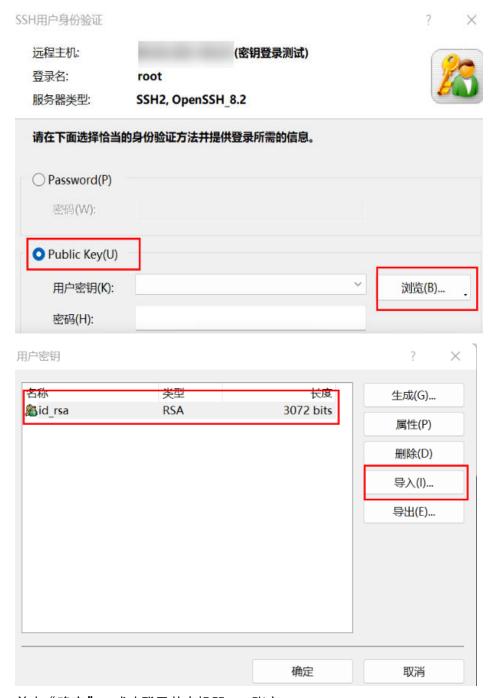
1. 新建会话,输入名称、主机、端口号,单击"连接"。



2. 输入登录用户名root, 单击"确定"。



3. 选择Public Key,单击"浏览 > 导入",导入下载至本地的id\_rsa私钥文件。



4. 单击"确定",成功登录节点机器root账户。

----结束

# A.12 管理面节点之间互信设置

## 操作场景

本章节指导如何配置管理面节点间root账户互信。

## 前提条件

客户可以登录管理面节点root账户,并同意配置节点间互信。

## 操作步骤

步骤1 以root用户登录管理面任一节点。若为多节点,需在每个节点执行以下修改。

步骤2 执行以下命令, 打开下述文件。

vi /etc/ssh/sshd\_config

步骤3 查找如下两行配置。

PubkeyAuthentication yes AuthorizedKeysFile .ssh/authorized\_keys

如没有或不一致,则添加或修改该配置,按"Esc"键后执行:wq!保存并退出。执行以下命令,重启SSHD服务。

systemctl restart sshd

若存在并与上述配置一致,继续执行后续步骤。

**步骤4** 执行以下命令,检查/root/.ssh目录下是否已经生成id\_rsa,id\_rsa.pub这两个SSH私钥和公钥文件。

cd /root/.ssh

ls id rsa

ls id\_rsa.pub

● 若未生成,则执行下述命令,不输入任何值,一直回车,生成id\_rsa和id\_rsa.pub。

#### ssh-keygen

• 若已生成,继续执行后续步骤。

步骤5 在目录下执行如下命令,生成授权公钥文件。

touch authorized\_keys

chmod 600 authorized\_keys

**步骤6** 执行如下命令,将当前节点公钥添加至当前节点授权公钥文件。

1. 执行以下命令,呈现公钥信息。

cat id\_rsa.pub

- 2. 复制公钥信息。
- 3. 执行以下命令,将复制的公钥信息写入authorized\_keys文件并保存。 vi authorized\_keys

**步骤7** 登录管理面另外两个节点,分别执行下述操作。

cd /root/.ssh

touch authorized\_keys

chmod 600 authorized keys

vi authorized\_keys

将步骤6复制的公钥信息写入节点的authorized\_keys文件并保存。

**步骤8** 重复执行**步骤1~步骤7**,将每个节点的公钥都添加至三个节点的授权公钥文件。

----结束

## A.13 本地 SSD 盘扩容方法

## 操作场景

本章介绍有实例的主机新增一块数据盘进行扩容的方法。

### 操作步骤

步骤1 以root用户登录主机。

步骤2 执行lsblk命令,确保扩容的本地SSD盘已被主机识别到。

以主机上有沙箱类型的实例、新增数据盘vdc进行扩容为例,回显如下所示,数据盘扩容路径为/var/chroot/var/lib/engine/data1。

```
[root@host-192-168-1-106 ~]# lsblk
NAME
                      MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
vda
                      252:0
                              0
                                   40G 0 disk
-vda1
                      252:1
                               0
                                   1G 0 part /boot
 -vda2
                      252:2
                               0
                                   39G 0 part
  ⊢klas-root
                      253:0
                                   35G 0 lvm
                               Θ
  ∟klas-swap
                                    4G 0 lvm [SWAP]
                      253:1
                               0
vdb
                      252:16
                               0 300G 0 disk
—gaussdbvg-mydatal
                               0
                                 200G 0 lvm /var/chroot/var/lib/engine/datal
                      253:2
 -gaussdbvg-backupdata 253:3
                               0
                                       0 lvm /var/chroot/var/lib/log/backup
                                   20G
                                   64G 0 lvm
                                              /var/chroot/usr/local/etcd
 -gaussdbvg-etcddata
                      253:4
                               0
                      252:32 0 100G 0 disk
```

步骤3 执行如下命令,新建并打开脚本extend.sh。

vi extend.sh

步骤4 复制如下代码至extend.sh脚本中。

#! /bin/bash new disks=\$1

```
target=$2
extend size=$3
# new_disk make sure the disk is valid. It can be a list(when split by ',').
# target eg:/var/chroot/var/lib/engine/data1
# show lsblk info, check the disk is managed by lvm or lvm2.
read vg_name disk_type mount_point <<< $(echo $(lsblk -l|grep $target|awk {'print($ 1, $6,$7)'}))
echo "lsblk info: $vg_name, $disk_type, $mount_point"
if [[ $disk_type == lvm* ]]
then
echo "Type: $disk_type, ..."
echo "It's not a lvm filesystem"
 exit 1
fi
# split the new_disks
disk_list=(`echo $new_disks |tr ',' ' '`)
# ready to create a pv one by one.
for new_disk in ${disk_list[@]}
do
  # cmd : pvcreate /dev/sdc
  pvcreate $new_disk
done
# pv without vg_group
echo "-----
echo "$(pvs)"
echo "-----"
# vg extend volume one by one
for new_disk in ${disk_list[@]}
  # cmd : vgextend gaussdbvg /dev/sdc
  vgextend gaussdbvg $new_disk
done
# figure out the target mount_point's FileSystem.
read file_system <<< $(df -h |grep $target |awk {'print($1)'})
echo "The $target 's mapper is: $file_system .'
# cmd: 100% vg free; lvextend -l +100%FREE /dev/mapper/gaussdbvg-mydata1
# lvextend -l +100%FREE $file_system , should be sum(new_disks)
for new_disk in ${disk_list[@]}
do
  # cmd : vgextend gaussdbvg /dev/sdc
  read size <<< $(echo $(fdisk -l|grep -w $new_disk |awk {'print($3)'}))
  lvextend -L +$extend_size""G $file_system
resize2fs $file_system
```

步骤5 按 "Esc"键后执行如下命令,保存并退出extend.sh脚本。

:wq!

步骤6 执行如下命令,执行extend.sh脚本,以扩容盘vdc为例,扩容100GB。

chmod +x extend.sh; ./extend.sh /dev/vdc /var/chroot/var/lib/engine/data1 100;

该脚本支持多盘扩容,以扩容盘vdc、vdd为例,命令如下所示。

chmod +x extend.sh; ./extend.sh /dev/vdc,/dev/vdd /var/chroot/var/lib/engine/data1 100;

注意,如果有多个数据盘扩容路径,请将扩容大小均分。

### 步骤7 执行lsblk命令,检查扩容结果。

```
[root@host-192-168-1-106 ~]# lsblk
NAME
                     MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
vda
                     252:0
                            0
                                 40G 0 disk
⊢vda1
                     252:1
                                 1G 0 part /boot
                             Θ
∟vda2
                                 39G 0 part
                     252:2
                     253:0 0 35G 0 lvm /
  —klas-root
   -klas-swap
                            0
                                 4G 0 lvm [SWAP]
                     253:1
                     252:16 0
                                300G 0 disk
                            0 300G 0 lvm /var/chroot/var/lib/engine/datal
—gaussdbvg-mydatal
                     253:2
 -gaussdbvg-backupdata 253:3 0
                                     0 lvm /var/chroot/var/lib/log/backup
                                 20G
 -gaussdbvg-etcddata 253:4 0
                                 64G 0 lvm /var/chroot/usr/local/etcd
vdc
                     252:32 0 100G 0 disk
└gaussdbvg-mydatal
                     253:2 0 300G 0 lvm /var/chroot/var/lib/engine/datal
```

----结束

# A.14 docker-service 目录被删除的卸载步骤

### 操作场景

若docker-service目录被删除,需参照如下步骤进行云数据库GaussDB管理平台(TPOPS)的卸载。

### 操作步骤

步骤1 以root用户登录到云数据库GaussDB管理平台(TPOPS)的任一节点。

步骤2 将要卸载的版本对应的docker-service上传到/data目录下。

步骤3 执行以下命令,解压软件包。

cd /data

tar -xvf DBS-docker-service\_\*\_all.tar.gz -C /data

步骤4 执行以下命令,修改配置文件。

vi /data/docker-service/config/user edit file.conf

与安装时配置一致即可。安装时配置的修改,可参考修改配置参数。

### □ 说明

- SFTP的安装节点可以通过ps -ef | grep sftpd | grep -v grep命令分别在三个节点查询,有回显的代表是SFTP的安装节点。
- InfluxDB的安装节点可以通过**ps** -**ef** | **grep** influxd | **grep** -**v grep**命令分别在三个节点查询,有回显的代表是InfluxDB的安装节点。

**步骤5** 执行以下命令,渲染参数。

sh /data/docker-service/action/optionAction/render\_args\_new.sh install

步骤6 执行以下命令,将执行卸载节点的docker-service拷贝到另外两台节点。

scp -r /data/docker-service root@{ip}:/data

其中{ip}为另外两台管理面节点的IP地址。

**步骤7** 以root用户登录到另外两台节点,并分别执行以下命令修改配置文件(render\_args.yml)。

sed -i -e '/local\_ip:/d' -e '/local\_100:/d' -e '/static\_route\_list:/d' -e '/openapi\_float\_url\_for\_open\_gauss:/d' /data/docker-service/package/cloud/common/render/render\_args.yml

echo "local\_ip: node2\_ip" >> render\_args.yml

echo "local\_100: node2\_ip2" >> render\_args.yml

echo "static\_route\_list: 'node2\_ip,255.255.255.0'" >> render\_args.yml

echo "openapi\_float\_url\_for\_open\_gauss: 'node2\_ip:8002'" >> render\_args.yml

假设node1\_ip为执行节点,此处以node2\_ip和node2\_ip2举例,其中node2\_ip和node2\_ip2为远程节点的IP地址,对应于user\_edit\_file.conf中配置的node2\_ip和node2\_ip2。node3\_ip同理。

修改之后的render\_args.yml图示如下:

隐藏部分即为对应机器的IP地址。

**步骤8** 以root用户登录到执行卸载命令的节点(解压docker-service目录的节点),并执行以下命令,卸载轻量化管理面。

cd /data/docker-service

sh appctl.sh uninstall\_all

sh appctl.sh cleanup\_all

----结束

## A.15 联系技术支持

请通过下列方式反馈您对产品文档、联机帮助、版本说明书的意见和建议。

- 通过support网站中的热线电话反馈。
- 通过support网站中"产品支持"页面提供的求助方式反馈。