

# QianBase 安装部署指南 1.6.6 x86 版本

#### 版权

© Copyright 2015-2020 贵州易鲸捷信息技术有限公司

#### 公告

本文档包含的信息如有更改,恕不另行通知。

保留所有权利。除非版权法允许,否则在未经易鲸捷预先书面许可的情况下, 严禁改编或翻译本手册的内容。易鲸捷对于本文中所包含的技术或编辑错误、 遗漏概不负责。

易鲸捷产品和服务附带的正式担保声明中规定的担保是该产品和服务享有的唯一担保。本文中的任何信息均不构成额外的保修条款。

#### 声明

Microsoft® 和 Windows® 是美国微软公司的注册商标。Java® 和 MySQL® 是 Oracle 及其子公司的注册商标。Bosun 是 Stack Exchange 的商标。Apache®、 Hadoop®、HBase®、Hive®、openTSDB®、Sqoop® 和 Trafodion® 是 Apache 软 件基金会的商标。Esgyn, EsgynDB 和 QianBase 是易鲸捷的商标。

# 目录

目录	ii
前言	viii
本文简介	viii
目标读者	viii
修订历史	viii
1. 前提条件	1
2. 准备 PC	1
3. 硬件和操作系统配置建议	2
3.1 硬件物理部署建议	3
3.2 磁盘设置建议	3
3.3 网卡及网络配置建议	4
3.4 物理内存配置建议	6
3.5 SWAP 配置建议	6
3.6 CPU 配置建议	7
3.7 文件系统配置建议	7
3.8 操作系统参数设置	7
3.9 网络及内核参数设置	7
4. 验证集群环境	8
4.1 集群要求	9
4.2 检查磁盘空间	9
5. 安装 CDH Hadoop	
5.1 必要的 Hadoop 服务和设置	12
5.2 准备工作	13

5.2.1 上传依赖包	
5.2.2 配置/etc/hosts	
5.3 PRE_INSTALL	17
5.4 CDH_INSTALL	
5.4.1 配置 cdh_config.ini	
5.4.2 运行 cdh_install.py	
5.5 页面部署 CDH	
5.5.1 登录 Cloudera 网页	
5.5.2 选择部署节点	
5.5.3 选择存储库	
5.5.4 角色分配	
5.5.5 数据库设置	
5.5.6 安装成功	
5.6 Hadoop 配置 HA	
5.6.1 HDFS 配置高可用配置	
5.6.2 YARN HA 配置	
5.6.3 HBASE HA 配置	
5.6.4 HIVE HA 配置	
5.7 Hadoop 服务优化	
5.7.1 HDFS heap 大小参数调优	
5.7.2 ZooKeeper 参数调优	
5.7.3 HBase 参数调优	
6. 安装与配置 OM	40
6.1 OM HA 的安装	
6.1.1 准备安装包	40
6.1.2 安装 OM 基础包	40
6.1.3 安装 OM HA	

	6.1.4 验证 OM HA	59
	6.1.5 修改数据库配置文件	60
	6.1.6 重启数据库 OM CLIENT 服务	60
	6.1.7 重启数据库 MDS 服务	60
	6.1.8 重新配置数据库的 LDAP 用户认证	61
	6.1.9 重新验证 OM 监控界面	62
	6.2 与数据库集成的组件	62
	6.2.1 安装	62
	6.2.2 配置	63
	6.2.3 启动 Esgyn DBManager	69
	6.2.4 停止 Esgyn DBManager	69
	6.2.5 日志设置	70
	6.2.6 查看 Esgyn DBManager 的状态	70
	6.2.7 查看 Esgyn DBManager 的版本	70
	6.2.8 高可用性配置	72
7	. 准备安装 QianBase	74
	7.1 获取 sudo 访问权限和无密码 SSH(命令行安装)	75
	7.2 配置 LDAP Identity Store	75
	7.3 用户 ID 和密码	76
	7.4 所需软件	76
	7.5 收集信息	77
8	. 安装 QianBase	81
	8.1 命令行安装程序	82
	8.2 管理	84
	8.3 验证	85
9	. 升级 QianBase	88

	9.1 环境检查	89
	9.2 备份配置文件	90
	9.3 在线备份数据	93
	9.3.1 备份简介	93
	9.3.2 在线备份前准备	93
	9.3.3 在线备份 QianBase	95
	9.3.4 检查在线备份数据	96
	9.3.5 导出在线备份的数据	97
	9.4 备份 metadata	98
	9.4.1 手动备份 snapshot	98
	9.5 升级 QianBase	99
	9.5.1 停用 QianBase	99
	9.5.2 解压 Installer	99
	9.5.3 升级 QianBase	100
	9.5.4 检查 QianBase	100
	9.6 版本回退	102
	9.6.1 收集日志	102
	9.6.2 回退准备	102
	9.6.3 回退安装	102
	9.6.4 恢复 metadata 数据	103
	9.6.5 在线恢复数据	104
	9.7 已知问题及解决办法	105
	9.7.1 关于 installer	105
	9.7.2 关于备份和恢复	106
1	l0. 卸载 QianBase	110
	10.1 停止 QianBase	111

10.2 卸载 QianBase	
11. 卸载 CDH 及 Hadoop 服务	
12. 故障排除	117
13. 启用安全功能	118
13.1 配置 LDAP	
13.2 安装与配置 OpenLDAP 服务器	
13.2.1 安装 OpenLDAP	
13.2.2 配置 OpenLDAP 服务器	
13.2.3 配置 OpenLDAP HA	126
13.2.4 使用 KeepAlived 提供服务器故障切换	
13.2.5 如何开启 LDAP 日志功能	136
13.2.6 开启 OpenLDAP 的密码策略	
13.2.7 开启 OpenLDAP 审查日志	
13.2.8 开启 OpenLDAP access log	
13.2.9 如何删除 overlay	161
13.2.10 如何删除数据库	
13.2.11 如何卸载 OpenLDAP	
13.2.12 Idapconfigcheck 工具	
13.2.13 ldapcheck 工具	164
13.2.14 故障排除	
13.3 生成服务器证书	170
13.3.1 自签名证书	170
13.3.2 生成 CSR	171
13.3.3 CA 签名证书	171
13.4 管理用户	
14. 提高安全性	

14.1 提高 Linux 安全性	174
14.2 提高 Hadoop 安全性	174
14.3 提高 Jetty Server 安全性	175
14.4 更新密码	175
14.5 提高端口安全性	175
附录1. 验证配置文件	176
附录 2. Inspector 工具	
附录 3. QianBase 和 Hbase 参数优化	184
附录 4. QianBase 在线增加节点	
附录 5. QianBase 离线删除节点	192
附录 6. 安装后配置 DCS Master 的 HA	201
附录7.内外网映射指南	204
附录 8. Hadoop 官方安装方法	209
1. CDH 安装准备依赖配置	
2. Hadoop 集群角色规划	214
3. Hadoop 集群安装配置	217
附录 9. Preload 功能使用说明	225
附录10. 端口列表	228

# 本文简介

本文介绍如何在Hadoop集群上安装和配置QianBase核心产品(基于Trafodion)和所需组件。

安装 QianBase 之前,请安装好操作系统或 Hadoop 发行版以及 LDAP。更多关于 这些组件的安装和配置信息,请参阅相关供应商文档。

# 目标读者

本指南的目标读者为 QianBase 和 Hadoop 系统管理员。

大版本	小版本	日期	摘要
1.6	1.6.6	2021年03月	更新版本号
1.6	1.6.6_rc2	2021年02月	附录10中添加 Hadoop 端口列表
1.6	1.6.6_rc1	2020年12月	更改 OM 安装和配置内容,形成 x86 版 本。 修改 CPU 和内存参数。 修改安装,卸载和升级章节顺序。 增加升级数据库后清除 TLOG 注意事项
1.6	1.6.3 rc9	2020年10月	修改 <u>11.2备份配置文件</u> 中第2点ms.env文 件部分。
1.6	1.6.2	2020年7月	删除高可用指南章节,详情请参阅 《QianBase 高可用性指南》一文。

# 修订历史

1.6	1.6.1	2020年6月	修改 <u>高可用指南</u> ,安装步骤做了调整,加 入了 addon 自动化安装,删除了手动安装 部分,并更新了检查部分
1.5	1.5.4	2020年3月	新增高可用指南
1.5	1.5.0	2019年12月	丰富了10.2 卸载 QianBase 的内容
1.4	1.4.0	2019年11月	新增附录10: 端口列表
1.1	1.1.2	2019年9月	丰富了 13.2 安装与配置 OpenLDAP 服务 器章节的内容
1.1	1.1.1	2019年7月	增加了OM-Installer 的安装(第七章)。 删除了原 5.7.4Yarn 参数调优。 删除了附录 4 中 balance_switch 的停用和 启用的设置。 附录中添加 Preload 功能使用说明。
1.1	1.1.0	2019年6月	

# 1. 前提条件

安装 QianBase 需要许可证秘钥。安装前,请确保已获取许可证秘钥。

# 2. 准备 PC

如果您使用 Windows 客户端远程安装 QianBase, 在安装之前,建议您在 Windows 上安装好以下软件,或功能类似的软件:

- PuTTY和PuTTYgen(如需下载,请参阅 http://www.chiark.greenend.org.uk/~sgtatham/PuTTY/download.html)
- VNC 客户端 (如需下载,请参阅 <u>http://www.realvnc.com/</u>)
- Firefox 或 Chrome 浏览器
- SFTP 客户端 (传输 PC 和服务器之间的文件): WinSCP 或 FileZilla

# 3. 硬件和操作系统配置建议

本章讲述以下内容:

- 3.1 硬件物理部署建议
- 3.2 磁盘设置建议
- 3.3 网卡及网络配置建议
- 3.4 物理内存配置建议
- 3.5 SWAP 配置建议
- 3.6 CPU 配置建议
- 3.7 文件系统配置建议
- 3.8 操作系统参数设置
- 3.9 网络及内核参数设置

# 3.1 硬件物理部署建议

QianBase 对硬件的要求分两个方面,数据节点和控制节点。对于数据节点,每 个节点的基本硬件配置建议如下:

节点	配置建议
CPU	英特尔 XEON 或者 AMD 64-bit 处理器
	每节点的 CPU 核数≥16
Memory	128GB 以上的内存空间,并且根据业务量按照每多一个 mxosrvr 进程增
	加 0.5GB
	mxosrvr 进程数的计算方式为:
	当前最大连接数/节点数
Network	10GigE, 1GigE, 或 2x10GigE 网络绑定
Storage	SATA 或者 SAS 或者 SSD, 通常在 JBOD 配置中配置 12-24 个 1TB 磁盘,
	系统盘推荐配置为 RAID1,或者 RAID1+0
对于控制节点	点,每个节点的基本硬件配置建议如下:

节点	配置建议		
CPU	英特尔 XEON 或者 AMD 64-bit 处理器		
	16≤ 每节点的 CPU 核数 ≤64。		
Memory	整个 hadoop 生态圈组件推荐使用 256GB 以上的内存空间,并且根据		
	务量按照每多一个 mxosrvr 进程增加 128MB。		
Network	10GigE, 1GigE, 或者 2x10GigE 网络绑定,并搭配合适的交换机。		
Storage	SATA 或者 SAS 或者 SSD, 通常 配置 6-12 块 1TB 磁盘, 系统盘推荐配		
	置为 RAID1.		

### 3.2 磁盘设置建议

对于安装 QianBase 的服务器,建议将服务器的前两块磁盘配置为 RAID1,用于 安装操作系统,以及存放软件配置文件。数据盘不需要做 RAID,数据盘的个数 可根据系统总数据量而定,一般数据节点12~24块,管理节点6~12块。

### 3.3 网卡及网络配置建议

对于安装 QianBase 的服务器,建议采用万兆网络,并采用2张万兆网卡做成链 路聚合 (bond),通过 bonding 可以将多个以太网口的网络连接聚合起来,一方 面可以提供更大的网络带宽,另一方面还可以提供更好的可靠性和端口冗余保 障。

在 RHEL7.x 中提供了常用的两种双网卡绑定模式。

activebackup - 主备模式,一个网卡处于活动状态,另一个处于备份状态,所有 流量都在主链路上处理,当活动网卡 down 掉时,启用备份网卡。

roundrobin - 轮询模式,所有链路处于负载均衡状态,这种模式的特点增加了带宽,同时支持容错能力。用户可以根据需求进行配置。

建议将网卡的 MTU 值设置为 9000。

1. 主备模式的配置方式

1) 查看当前网卡设备

nmcli dev

运行结果如下图所示:

[root@localhost network-scripts]#			nmcli dev	
DEVICE	TYPE	STATE	CONNECTION	
virbr0	bridge	connected	virbr0	
eno16777736	ethernet	connected	team0-port2	
eno33554984	ethernet	connected	team0-port1	
virbr0-nic	ethernet	connected	virbr0-nic	
team0	team	connected	teamO	
10	loopback	unmanaged		
[root@localhost network-scripts]#				

2) 查看当前网卡连接状态

nmcli con show

运行结果如下图所示:

```
[root@localhost network-scripts]# nmcli con show

NAME UUID TYPE DEVICE

team0 559fc3fb-9077-465e-b167-fd6cf3dba960 team team0

virbr0-nic da61dc25-f066-49e4-85fe-587a6845ad1 802-3-ethernet virbr0-nic

virbr0 f0c766ee-d11b-4848-9090-896f5d4b8af6 bridge virbr0

team0-port2 a34fd625-592a-44f7-b092-a7fde9af4c83 802-3-ethernet eno16777736

team0-port1 f0b2e93c-4bfd-411e-85ea-5282fe8d9f76 802-3-ethernet eno33554984

[root@localhost network-scripts]#
```

3) 将 en 开头需要进行聚合的两张网卡从网络连接中删除

nmcli con del eno33554984 eno50332208

红色部分需要根据机器具体的网卡名称进行更改

4) 创建网卡聚合 team0, 并配置为主备模式

nmcli con add type team con-name team0 ifname team0
config '{"runner":{"name":"activebackup"}}'

5) 将 team0 地址方式修改为手动

nmcli con modify team0 ipv4.method manual

6) 配置 team0 的 IP 地址

```
nmcli con modify team0 ipv4.address
192.168.118.122/24 ipv4.gateway 192.168.118.1
红色部分需要根据具体的 IP 地址进行更改。
```

7) 将需要聚合的两张网卡加入 team0

```
nmcli con add type team-slave con-name team0-port1
ifname eno33554984 master team0
nmcli con add type team-slave con-name team0-port2
ifname eno50332208 master team0
```

红色部分需要根据机器具体的网卡名称进行更改。

8) 重启网络服务

systemctl restart network.service

9) 查看聚合后的网卡 team0 状态

teamdctl team0 st

2. 轮询模式配置方式:

只需在创建的 team0 的时候参数修改为如下即可:

nmcli con add type team con-name team0 ifname team0
config '{"runner":{"name": "roundrobin"}}'

# 3.4 物理内存配置建议

对于安装 QianBase 的服务器建议的内存配置为 64G 以上,一般建议 128G 或 256G,并根据业务量按照每个 mxosrvr 增加 0.5G 的空间递增。并在内核参数中 进行如下设置:

vm.zone\_reclaim\_mode (内存管理模式,将其设置为0)

1) vi /etc/sysctl.d/99-sysctl.conf

2) vm.zone reclaim mode =0 //如果没有这个参数,添加,有的话修

改成0

3) sysctl -p //使内核参数修改生效

### 3.5 SWAP 配置建议

关于 SWAP 配置, RedHat 官方的建议如下:

物理内存	建议的交换空间大小	如果开启休眠功能建议的交换空间大小
≤ 2GB	2倍内存大小	3倍内存大小
> 2GB - 8GB	与内存大小相同	2倍内存大小
> 8GB - 64GB	至少4GB	1.5 倍内存大小
> 64GB	至少4GB	不建议开启休眠功能

对于安装 QianBase 的服务器,服务器的内存通常大于 64G,我们给出的 SWAP

配置建议为4GB<=服务器的物理内存<=128G。

### 3.6 CPU 配置建议

对于安装QianBase的服务器,我们建议将CPU的运行模式修改为Performance, 在REHL7.x系统中可以通过如下命令查看,和修改:

cat /sys/devices/system/cpu/cpu0/cpufreq/scaling\_governor conservative

cpupower frequency-set -g performance cat /sys/devices/system/cpu/cpu0/cpufreq/scaling\_governor performance

### 3.7 文件系统配置建议

对于安装QianBase的服务器,建议除了SWAP分区之外,为boot分区划分500M的空间,建议文件系统使用EXT4的文件系统。

QianBase 默认安装在/opt 目录下,建议/opt 目录下至少有 20G 可用空间。

### 3.8 操作系统参数设置

对于安装 QianBase 的服务器,在安装操作系统的时候建议将"Server withGUI"的软件包安装上,以方便进入图形化界面。

# 3.9 网络及内核参数设置

对于安装 QianBase 的服务器,我们建议在系统内核的参数中添加如下:

```
vi /etc/sysctl.d/99-sysctl.conf
vm.swappiness=1
vm.zone_reclaim_mode =0
kernel.msgmnb=65536
kernel.msgmax=65536
sysctl -p
```

# 4. 验证集群环境

本章讲述以下内容:

4.1 集群要求

4.2 检查磁盘空间

安装 QianBase 之前,请验证集群环境。

### 4.1 集群要求

硬件平台	x86-64
操作系统	Red Hat 7.4 (64 位)
Hadoop 发行版	Cloudera CDH 5.16.2
用户 ID	如果使用命令行安装方式,用户需具备无密码 sudo 访问
	权限。
	更多信息,请参阅获取 sudo 访问权限和无密码 SSH。
集群大小	集群由n (n>=2) 个节点组成。
	目前暂无上限,最小为2个节点,建议至少安装4个节
	点。
磁盘空间	最小 20 GB,生产环境>=500GB。
	更多信息,请参阅检查磁盘空间。
内存	最小配置 = 1GB * 集群中每个节点连接服务器
	(MXOSRVR 进程)的数量,生产环境>=128GB。

### 4.2 检查磁盘空间

安装 Hadoop 发行版之前,请确保至少有 20 GB 可用空间。

Cloudera CDH 的默认安装路径为/var/lib/cloudera-scm-server-db。 如需检查/var 可用空间,请启动 PuTTY 会话或在待安装 Cloudera 集群的节点 上启动 VNC 终端。执行以下命令需要 root 或 sudo 访问权限。

确认/var 至少有 20 GB 可用空间。

\$ cd "	'/var"
\$ df -	-hP

如果/var 没有足够空间,请为 Cloudera 数据库提供一个连接至其它驱动的软

连接,该驱动应有足够空间。

\$ cd <new drive=""> (eg. cd /DATA)</new>
\$ mkdir cloudera-scm-server-db
\$ chmod 777 cloudera-scm-server-db
\$ cd /var/lib
\$ <pre>ln -s <new drive="">/cloudera-scm-server-db</new></pre>

如果/var 是集群根文件系统的子目录, Cloudera 数据库应具有足够的可用空

间。

如果已安装 Cloudera 发行版且 log 目录为红色,这表示 Cloudera 安装在 /var/lib 且/var 文件系统较小。此时,您可以使用一个非正式脚本 (clouderaMoveDB.sh)移动目录,该脚本位于 installer/tools,它在

安装程序(tar.gz 文件)被解压时创建。直接执行命令 clouderaMoveDB.sh 将显示帮助信息。

# 5. 安装 CDH Hadoop

本章讲述的是用于安装 CDH 的工具的使用步骤,包括准备工作和执行过程记录

- 等。关于 Hadoop 官方安装详情请参考附录 8.
- 5.1 必要的 Hadoop 服务和设置
- 5.2 准备工作
- 5.3 PRE\_INSTALL
- $5.4 \ \text{CDH\_INSTALL}$
- 5.5 页面部署 CDH
- 5.6 Hadoop 配置 HA
- 5.7 Hadoop 服务优化

发行版	版本	HBase版本	安装
Cloudera发行版	CDH 5.4 ~ 5.13	1.2	更多关于安装Cloudera的信
包括Apache Hadoop			息,请参阅Cloudera官网
(CDH)			
Hortonworks数据平台	HDP 2.4 ~ 2.6.3	1.1	更多关于安装 Hortonworks 的
(HDP)			信息,请参阅 Hortonworks 官
			网

QianBase 与 Cloudera 和 Hortonworks 发行版兼容。

# 5.1 必要的 Hadoop 服务和设置

安装 Hadoop 发行版之前,请检查以下必要的服务和设置,确保安装时选择这些

服务和设置:

- HDFS
- Yarn/MapReduce
- ZooKeeper
- HBase
- Hive
- Embedded Databases

# ▲注意

QianBase 命令行安装程序需要在 QianBase 集群中的某一节点上运行。目前暂不

支持在集群外节点上安装。所有 QianBase 节点必须安装 HBase。

# ▲注意

QianBase 安装之前请检查集群中各节点时间是否同步,使用命令 ntpstat,如显示结果如以下所示表示同步成功,若时钟不同步,请先配置 ntp 服务后再进行

数据库安装!

[root@gy12 ~] # ntpstat

synchronized to NTP server (10.10.12.11) at stratum 10

time correct to within 49 ms

polling server every 1024 s

### 5.2 准备工作

安装 CDH 前需要做下列的准备工作,才能开始相关的预安装工作(执行 PRE\_INSTALL)。

#### 5.2.1 上传依赖包

拿到环境后需要上传相关的依赖包,将这些包上传到第一个节点即可(执行预安 装脚本的节点),具体包含如下几个:

1. 包含预安装脚本的Python Installer

```
[root@host-10-10-23-62 ~]# cd pre_install/
[root@host-10-10-23-62 pre_install]# ll
total 148
-rw-r--r-. 1 root root 147740 Apr 19 14:55
esgyn_corebank_pyinstaller-1.0.0-RH7.tar.gz
```

[root@host-10-10-23-62 pre\_install]# tar zxf
esgyn corebank pyinstaller-1.0.0-RH7.tar.gz

```
[root@host-10-10-23-62 pre_install]# 11
total 152
-rw-r--r-. 1 root root 147740 Apr 19 14:55
esgyn corebank pyinstaller-1.0.0-RH7.tar.gz
```

drwxr-xr-x. 5 500 500 4096 Apr 19 14:53 python-installer [root@host-10-10-23-62 pre install]# cd python-installer/ [root@host-10-10-23-62 python-installer]# 11 total 184 -rwxr-xr-x. 1 500 500 13117 Apr 1 16:08 add nodes.py -rwxr-xr-x. 1 500 500 6138 Apr 18 17:42 cdh install.py drwxr-xr-x. 2 500 500 4096 Apr 19 14:52 configs -rwxr-xr-x. 1 500 500 29248 Apr 11 19:19 db install.py -rwxr-xr-x. 1 500 500 5578 Apr 1 16:08 db uninstall.py -rwxr-xr-x. 1 500 500 7946 Apr 1 16:08 delete nodes.py -rwxr-xr-x. 1 500 500 9700 Apr 1 16:08 inspector.py -rwxr-xr-x. 1 500 500 9167 Apr 19 13:04 pre install.py -rw-r--r-. 1 500 500 54204 Apr 1 15:52 prettytable.py -rw-----. 1 500 500 36 Apr 19 14:53 PyInstallerVer -rw-r--r-. 1 500 500 1324 Apr 1 15:52 README.md drwxr-xr-x. 3 500 500 4096 Apr 19 14:53 scripts -rwxr-xr-x. 1 500 500 8243 Apr 1 16:08 secure setup.py drwxr-xr-x. 2 500 500 4096 Apr 10 13:54 templates

#### 2. RedHat的所有依赖包镜像

```
[root@host-10-10-23-62 ~]# ll iso/
total 7587844
-rw-r--r-. 1 root root 7769948160 Apr 19 11:54 rhel-server-
7.4-x86 64-dvd.iso
```

#### 3. CM的所有rpm包

[root@host-10-10-23-62 ~]# 11 cdh/cm\_rpms/ total 690312 -rw-r--r--. 1 root root 9805556 Apr 19 11:45 clouderamanager-agent-5.13.3-1.cm5133.p0.6.el7.x86\_64.rpm -rw-r--r--. 1 root root 697039048 Apr 19 11:45 clouderamanager-daemons-5.13.3-1.cm5133.p0.6.el7.x86\_64.rpm -rw-r--r--. 1 root root 8696 Apr 19 11:38 clouderamanager-server-5.13.3-1.cm5133.p0.6.el7.x86\_64.rpm -rw-r--r-. 1 root root 10608 Apr 19 11:38 clouderamanager-server-db-2-5.13.3-1.cm5133.p0.6.el7.x86\_64.rpm drwxrwxr-x. 2 root root 4096 Apr 19 11:45 repodata

#### 4. CDH的parcels文件

[root@host-10-10-23-62 ~]# ll cdh/cdh\_parcels/ total 1889784 -rw-r--r-. 1 root root 1935128068 Apr 19 11:51 CDH-5.13.3-1.cdh5.13.3.p0.2-el7.parcel -rw-r--r-. 1 root root 41 Apr 19 11:47 CDH-5.13.3-1.cdh5.13.3.p0.2-el7.parcel.sha

#### 5. 安装CM所需的依赖包

[root@host-10-10-23-62 ~]# ll cdh/dependencies/7/ total 380544 -rw-r-xr--. 1 root root 105728 Apr 19 11:45 apr-1.4.8-3.el7\_4.1.x86\_64.rpm

```
-rwxr-xr-x. 1 root root 105572 Apr 19 11:45 apr-1.4.8-

3.el7.x86_64.rpm

-rwxr-xr-x. 1 root root 94132 Apr 19 11:47 apr-util-1.5.2-

6.el7.x86_64.rpm

-rw-r-xr--. 1 root root 51976 Apr 19 11:47 at-3.1.13-

22.el7_4.2.x86_64.rpm

drwxrwxr-x. 2 root root 4096 Apr 19 11:47 repodata

...
```

#### 6. MySQL的tar文件

[root@ host-10-10-23-62 ~]# 11 cdh/mysql/ total 318920 -rw-r--r-. 1 root root 325604229 Apr 19 11:38 mysql-5.6.31.tar.gz -rw-r--r-. 1 root root 960372 Apr 19 11:38 mysqlconnector-java-5.1.34.jar

#### 7. MySQL约jdbc driver

[root@host-10-10-23-62 ~]# 11 cdh/mysql/ total 318920 -rw-r--r-. 1 root root 325604229 Apr 19 11:38 mysql-5.6.31.tar.gz -rw-r--r-. 1 root root 960372 Apr 19 11:38 mysqlconnector-java-5.1.34.jar

### 5.2.2 配置/etc/hosts

需要在首节点配置/etc/hosts文件,必须确保该文件的配置正确,其余节点的配

置均以该文件内容为模板进行配置。

```
[root@host-10-10-23-62 ~]# cat /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4
localhost4.localdomain4
::1 localhost localhost.localdomain localhost6
localhost6.localdomain6
10.10.23.62 esggy-qa-n012.esgyncn.local esggy-qa-n012
10.10.23.63 esggy-qa-n013.esgyncn.local esggy-qa-n014
10.10.23.65 esggy-qa-n015.esgyncn.local esggy-qa-n015
```

到此所有准备工作完毕。

# 5.3 PRE\_INSTALL

执行 pre\_install.py 脚本, 它会自动做如下相关的配置:

- 1. 配置主机名 hostname
- 2. 配置无密码 ssh 访问
- 3. 配置 linux 系统限制参数 ulimits
- 4. 配置内存页面交换 swappiness
- 5. 制作 Linux 的 http 安装源
- 6. 关闭防火墙 firewall
- 7. 关闭 selinux
- 8. 配置 ntp 服务

#### 9. 禁用透明大页内存

具体执行如下所示(以下黄底粗体字为需要手动执行和输入的,灰色斜体字为注释

项):

[root@host-10-10-23-62 ~] # cd pre install/python-installer/ [root@host-10-10-23-62 python-installer]# ./pre install.py \*\*\*\* Trafodion Pre-Installation ToolKit Enter user name to set up passwordless SSH for [root]: --此处填写要配置无密码访问的用户,建议使用默认用户 root Enter remote host SSH Password: linux00 -- 此处填写上面用户的密码,须确保所有节点的用户密码一样 Confirm Enter remote host SSH Password: linux00 --再次输入密码,确认无误 Enter list of Cloudera Nodes separated by comma, support simple numeric Regular Expression, i.e. "n[01-12],n[21-25]","n0[1-5].com": 10.10.23.[62-65] --填写需要配置的主机 IP 或 hostname, 逗号隔开, 也可用正则表达式 Enter full path to iso file: /root/iso/rhel-server-7.4-x86 64dvd.iso --填写 centos 的依赖包镜像全路径 Enter the gateway of the LAN (used for NTP setting) [10.10.23.1]: --填写该集群的网关,默认会自动识别,也可手动填写 Enter the subnet mask of the LAN (used for NTP setting) [255.255.255.128]: --填写该集群的子网掩码, 默认会自动识别, 也可手动填写 \*\*\*\* Final Configs \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* +-----| config type | value +-----+ | iso file | /root/iso/rhel-server-7.4-x86 64-dvd.iso | | node list 10.10.23.62,10.10.23.63,10.10.23.64,10.10.23.65 | ntp gateway | 10.10.23.1 | ntp\_mask | 255.255.255.128

```
| ssh username | root
+-----+
Confirm result (Y/N) [N]: y
                          --确认配置无误后输入 y 开始 pre_install
** Generating config file [/root/pre install/python-
installer/preinstall config] to save configs ...
****
 Pre-Installation Start
** Log file location:
[/var/log/trafodion/preinstall 20190419 152004.log]
TASK: Passwordless SSH Set Up
Setting up passwordless SSH across nodes
[10.10.23.62,10.10.23.63,10.10.23.64,10.10.23.65] for user
[root]
***[INFO]: Setting up ssh on host [10.10.23.62]
***[INFO]: Setting up ssh on host [10.10.23.63]
***[INFO]: Setting up ssh on host [10.10.23.64]
***[INFO]: Setting up ssh on host [10.10.23.65]
Host [localhost]: Script
[gen sshkey.py] .....
[ OK ]
TASK: Linux Repo Setup
****
Setting up linux repo on nodes
[10.10.23.62,10.10.23.63,10.10.23.64,10.10.23.65]
***[INFO]: Setting up linux repo on host [10.10.23.62]
***[INFO]: Setting up linux repo on host [10.10.23.63]
```

```
***[INFO]: Setting up linux repo on host [10.10.23.64]
***[INFO]: Setting up linux repo on host [10.10.23.65]
Host [localhost]: Script
[gen_repo.py] .....
[ OK ]
TASK: Services Setup
******
Host [10.10.23.62]: Script
[services_set.py] .....
[ OK ]
Host [10.10.23.63]: Script
[services set.py] .....
[ OK ]
Host [10.10.23.65]: Script
[services_set.py] .....
[ OK ]
Host [10.10.23.64]: Script
[services set.py] .....
[ OK ]
Time Cost: 0 hour(s) 0 minute(s) 59 second(s)
*****
 Pre-Installation Complete
****
TASK: Environment Discover
Time Cost: 0 hour(s) 0 minute(s) 3 second(s)
*****
 Pre-Install results
****
Hosts: esggy-qa-n012,esggy-qa-n013,esggy-qa-n014,esggy-qa-n015
```

OverView | Stat | Expected | +----+ FQDN Linux distro \_ Firewall status 1 \_ 0 Selinux status 1 \_ Ntp service status \_ 1 Ntp service status | o | \_ | Transparent huge pages status | o | \_ 

pre\_install.py脚本运行结束。

### 5.4 CDH\_INSTALL

执行cdh install.py脚本, 它会自动做如下相关的配置:

1. 安装 JDK 1.8

- 2. 安装 MySQL,并做主主备份
- 3. 安装 Cloudera
- 4. 配置 CDH 资源库
- 5. 配置 MySQL jdbc 驱动
- 6. 为 Cloudera 管理器配置外部数据库

7. 启动 Cloudera

具体执行如下所示(以下黄底粗体字需要手动执行和输入,灰色斜体字为注释

项):

### 5.4.1 配置 cdh\_config.ini

先要填写 cdh\_config.ini 配置文件,以便于 cdh\_install.py 脚本从中读取需要的相关信息:

```
[root@host-10-10-23-62 python-installer]# cd configs/
[root@host-10-10-23-62 configs]# 11
total 40
-rw-r--r-. 1 500 500 753 Apr 19 12:51 cdh config.ini
-rw-r--r-. 1 500 500 2252 Apr 12 20:22 central config.json
-rw-r--r-. 1 500 500 4858 Apr 1 15:52 db config default.ini
-rw-r--r-. 1 500 500 2510 Apr 3 15:33 default ports.ini
-rw-r--r-. 1 500 500 2896 Apr 1 15:52 mod cfgs.json
-rw-r--r-. 1 500 500 9508 Apr 11 19:19 prompt.json
-rw-r--r-. 1 500 500 3222 Apr 12 03:10 script.json
[root@host-10-10-23-62 configs]# vi cdh config.ini
[mysql]
#NOTE: if mysql dir is not defined, use yum to install by
default
#mysql path = /path/to/mysql binary installation package
#mysql hosts = esgyn1.localhost,esgyn2.localhost
#mysql jdbc path = /path/to/mysql jdbc file
mysql path = /root/cdh/mysql/mysql-5.6.31.tar.gz
                                          -- 配置 mysql tar 包的全路径
mysql hosts = esggy-qa-n012.esgyncn.local,esggy-qa-
n013.esgyncn.local
                                     --配置需要安装 mysql 的两个节点
```

mysql\_jdbc\_path = /root/cdh/mysql/mysql-connector-java-5.1.34.jar

-- 配置 mysql jdbc driver 的全路径

[mysqld]

#Mysql HA setting character-set-server=utf8 server-id = log-bin = bin.log log-slave-updates = log-bin-index = log-bin.index relay-log = relay.log relay-log-info-file = relay-log.info relay-log-index = relay-log.index auto\_increment\_increment = 2 binlog\_format = mixed auto increment offset =

[dirs]

# NOTE: if repo\_url is defined, repo\_dir takes no effect

#repo url = http://cmlocalrepo/

#repo dir = /path/to/cmlocalrepo/,/path/to/dependences/repo/

#parcel dir = /path/to/parcel/

repo dir = /root/cdh/cm rpms,/root/cdh/dependencies/7

--配置 cloudera rpm 和其依赖包的目录,两个目录,用逗号隔开

parcel\_dir = /root/cdh/cdh\_parcels

-- 配置 CDH parcels 文件的目录

[hosts]

#hosts = n0[1-2].example.com

hosts = esggy-qa-n01[2-5].esgyncn.local

--配置待安装的节点主机名,逗号隔开,或用简单正则表达式

### 5.4.2 运行 cdh\_install.py

运行 cdh\_install.py 脚本做相关安装及配置:

[root@host-10-10-23-62 python-installer]# ./cdh\_install.py

Trafodion CDH-Installation ToolKit

Enter the password for mysql root user: traf123

--配置 mysql 的用户密码

Confirm Enter the password for mysql root user:**traf123** --再次输入密码确认

Enter the password for mysql backup user: **replpassword**--配置 mysql 备份用户的密码

Confirm Enter the password for mysql backup user: **replpassword** --再次输入密码确认

Copying mysql binary installation package to nodes [esggy-qan012.esgyncn.local,esggy-qa-n013.esgyncn.local]

\*\*\*[INFO]: Copying mysql binary installation package on host
[esggy-qa-n012.esgyncn.local]

\*\*\*[INFO]: Copying mysql binary installation package on host
[esggy-qa-n013.esgyncn.local]

\*\*\*[INFO]: Copying parcel repo to master nodes

\*\*\*[INFO]: Starting temporary python http server

24

```
***[INFO]: Starting temporary python http server
TASK: Deploy Mysql
Host [esqqy-qa-n014.esqyncn.local]: Script
[deploy mysql.py] ..... [ OK ]
Host [esggy-qa-n015.esgyncn.local]: Script
[deploy mysql.py] ..... [ OK ]
Host [esggy-qa-n012.esgyncn.local]: Script
[deploy mysql.py] ..... [ OK ]
Host [esqqy-qa-n013.esqyncn.local]: Script
[deploy mysql.py] ..... [ OK ]
TASK: Deploy Cloudera
Host [esggy-qa-n012.esgyncn.local]: Script
[deploy cdh.py] ..... [ OK ]
Host [esggy-qa-n014.esgyncn.local]: Script
[deploy cdh.py] ..... [ OK ]
Host [esggy-qa-n015.esgyncn.local]: Script
[deploy cdh.py] ..... [ OK ]
```

Host [esggy-qa-n013.esgyncn.local]: Script

[deploy\_cdh.py] ..... [ OK ]

Time Cost: 0 hour(s) 3 minute(s) 26 second(s)

\*\*\*[OK]: Cloudera RPMs installed successfully! Check cloudera manager status (timeout: 200 secs) ..... \*\*\*[OK]: cloudera manager successfully!

CDH-Installation Complete

cdh install.py 脚本运行结束。

# 5.5 页面部署 CDH

前面的两个步骤都成功完成后就可以到Cloudera网页上部署CDH了。

### 5.5.1 登录 Cloudera 网页

打开浏览器,进入 <u>http://10.10.23.62:7180</u> 网页,这是安装 Cloudera Manager Server 的节点的 IP 地址,默认登陆用户密码是 admin/admin,用户可用 此账号信息登录后自行修改密码:

admin		
□ 保留我的信息		
	登录	
欢迎使用 Cloudera Manager 最终用户许可条款和条件		
---	---	
Cloudera Standard License	*	
Version 2016-05-26		
END USER LICENSE TERMS AND CONDITIONS		
THESE TERMS AND CONDITIONS (THESE "TERMS") APPLY TO YOUR USE OF THE PRODUCTS (AS DEFINED BELOW) PROVIDED BY CLOUDERA, INC. ("CLOUDERA").		
PLEASE READ THESE TERMS CAREFULLY.		
IF YOU ("YOU" OR "CUSTOMER") PLAN TO USE ANY OF THE PRODUCTS ON BEHALF OF A COMPANY OR OTHER ENTITY, YOU REPRESENT THAT YOU ARE THE EMPLOYEE OR AGENT OF SUCH COMPANY (OR OTHER ENTITY) AND YOU HAVE THE AUTHORITY TO ACCEPT ALL OF THE TERMS AND CONDITIONS SET FORTH IN AN ACCEPTED REQUEST (AS DEFINED BELOW) AND THESE TERMS (COLLECTIVELY, THE "AGREEMENT") ON BEHALF OF SUCH COMPANY (OR OTHER ENTITY).		
BY USING ANY OF THE PRODUCTS, YOU ACKNOWLEDGE AND AGREE THAT: (A) YOU HAVE READ ALL OF THE TERMS AND CONDITIONS OF THIS AGREEMENT; (B) YOU UNDERSTAND ALL OF THE TERMS AND CONDITIONS OF THIS AGREEMENT; (C) YOU AGREE TO BE LEGALLY BOUND BY ALL OF THE TERMS AND CONDITIONS SET FORTH IN THIS AGREEMENT		
IF YOU DO NOT AGREE WITH ANY OF THE TERMS OR CONDITIONS OF THESE TERMS, YOU MAY NOT USE ANY PORTION OF THE PRODUCTS.	-	
☑		

#### 欢迎使用 Cloudera Manager

#### 您想要部署哪个版本?

升级到 Cloudera Enterprise 将提供可以帮助您在关键任务环境下管理和监控 Hadoop 群集的重要功能。

	Cloudera Express	Cloudera Enterprise Cloudera Enterprise 试用版	Cloudera Enterprise	
许可证	免费	60 天	年度订阅	
在试用期之后,该产品将继续作为 Cloudera Express 运行。您的群集和数据将会6 持不受影响。		在试用期之后,该产品将继续作为	上载许可证	
	选择许可证文件 上载			
		177/1°26/8949a	Cloudera Enterprise 在三个版本中可用: • Basic Edition • Flex Edition	
			Cloudera Enterprise	

# 5.5.2 选择部署节点

为 CDH 群集安装指定主机。					
新主机 当前管理的主机(4) 这些主机不属于任何群集。请选择组成器	样集的主机。				
2 名称	≑ IP	≑ 机架	⇔ CDH 版本		↓ 上一检测信号 ↓
任何名称	任何IP	任何机架	全部 *	全部	▼ 全部 ▼
esggy-qa-n012.esgyncn.local	10.10.23.62	/default	无	◎ 未知运行状况	10.12s ago
esggy-qa-n013.esgyncn.local	10.10.23.63	/default	无	◎ 未知运行状况	11.97s ago
esggy-qa-n014.esgyncn.local	10.10.23.64	/default	无	◎ 未知运行状况	8.67s ago
esggy-qa-n015.esgyncn.local	10.10.23.65	/default	无	❷ 未知运行状况	10.69s ago

# 5.5.3 选择存储库

群集安装	
选择存储库	
Cloudera 建议使用 parcel 3 当有软件更新可用时,将需	朱代蕾软件包进行安装,因为 parcel 可以使服务二进制文件的部署和升级自动化,让 Cloudera Manager 轻松地管理群集上的软件。如果选择不使用 parcel, 要您手动升级群集中所有主机上的包,并会阻止您使用 Cloudera Manager 的踪动升级功能
选择方法	◎ 使用数据包 🛛
	<ul> <li>● 使用 Parcel (建议) ● 更多远项</li> <li>代理设置</li> </ul>
选择 CDH 的版本	@ (μμ+5.13.3-1.cdh5.13.3.p0.2
	CDH-4.7.1-1.cdh4.7.1.p0.47
	对于此 Cloudera Manager 版本 (5.13.3) 太新的 CDH 版本不会显示。

群集安装				
正在安装选定 Parcel				
选定的 Parcel 正在下载并安装在群集的所有主机	几上。			
<b>∨ CDH</b> 5.13.3-1.cdh5.13.3.p0.2	已下载: 100%	已分配: 4/4 (95.7 MiB/s)	已解压: 4/4	已激活: 4/4

# 5.5.4 角色分配

群集设置							
自定义角色分配							
您可在此处自定义新群集的角色分配,但如果分配不正确(例如,分配到某个主机上的角色大多)会影响服务性能。除非您有特殊需求,如已为特定角色预先选择特定主机,否则 Cloudera 不 建议改变分配情况。							
还可以按主机查看角色分配。 按主机查看							
H HBase							
M Master × 1 新建	HBRE HBase REST Server	HBTS HBase Thrift Server	RS RegionServer × 3 新建				
esggy-qa-n012.esgyncn.local 选择主机		选择主机	与 DataNode 相同 ▼				
B HDFS							
NN NameNode × 1新建	SNN SecondaryNameNode × 1新建	B Balancer × 1新建	HFS HttpFS				
esggy-qa-n012.esgyncn.local	esggy-qa-n013.esgyncn.local 🗸	esggy-qa-n012.esgyncn.local	选择主机				
NFSC NFS Gateway	DN DataNode × 3 新建						
选择主机	esggy-qa-n[013-015].esgyncn.local 🗸						
寝 Hive							
G Gateway × 4新建	HMS Hive Metastore Server × 1新建	WHC WebHCat Server	HS2 HiveServer2 × 1 新建				
esggy-qa-n[012-015].esgyncn.local	esggy-qa-n012.esgyncn.local	选择主机	esggy-qa-n012.esgyncn.local				

# 5.5.5 数据库设置

君	样集设置					
WKX.	y据库设置					
	配置和测试数据库连接。首先根据Installation Gu	iide ご的Installing and Config	<b>juring an External Database</b> 小节仓	刘建数据库。		
	Hive				<b>√</b> :	Successful
	数据库主机名称:*	数据库类	型: 数据库名	;称:*    用户名:*	蜜码:	
	and an anti-an-	14.001	• hive	hivo		
	esggy-qa-no rz.esgynch.iocai	MySQL		Inve	traf123	
	esygyrdanio i z.esygnon.iocai	MySQL		IIIVe	traf123	2 显示密码

ŧ	样集设置		
Ē	軍核更改		
	HDFS 根目录	Cluster 1 > HBase ( 服务范围 )	0
	hbase.rootdir	/hbase	
	启用复制 hbase.replication	Cluster 1 > HBase(服务范围)	0
	启用编制索引	□ Cluster 1 > HBase(服务范围)	0
	HDFS 块大小 dfs.block.size, dfs.blocksize	Cluster 1 > HDFS(服务范围) 128 兆字节 *	0

# 5.5.6 安装成功

群集设置		
首次运行 命令		
状态 🔮 已完成 🏻 🏥 4月 19, 4:34:44 下午 🛛 Ø 3.6m		
Finished First Run of the following services successfully: ZooKeeper, HDFS, HBase, YARN (MR2 I	included), Hive, Cloudera Manage	ement Service.
✓ 已完成7个步骤(共7个)。		
● Show All Steps		
➤ Ensuring that the expected software releases are installed on hosts. 已成功完成 1 个步骤。	4月 19, 4:34:44 下午	84ms
➤ 正在部署客户稿配置 Cluster1C Successfully deployed all client configurations.	4月 19, 4:34:44 下午	16.32s
▶ ② 启动 Cloudera Management Service, ZooKeeper 已成功完成 2 个步骤。	4月 19, 4:35:01 下午	27.27s
▶ ● 启动 HDFS 已成功完成 1 个步骤。	4月 19, 4:35:28 下午	53.59s
▶ ○ 启动 HBase 已成功完成 1 个步骤。	4月 19, 4:36:22 下午	27.87s
▶ ● 启动 YARN (MR2 Included) 已成功完成 1 个步骤。	4月 19, 4:36:49 下午	30.31s
▶ ● 启动 Hive 已成功完成 1 个步骤。	4月 19, 4:37:20 下午	59.55s

cloudera MANAGER N## -	主机 • 诊断	乐▼ 审核 图表▼ 管理▼			● G 投索 支持 × admi
主页					
状态 所有运行状况问题 配置 ▶ 5 - 所有	最新命令				试用 Cloudera Enterprise 60 天 添加群集
Cluster 1 (CDH 5.13.3, Percel)		国农		HDFS ID	30万1年「196」2月6日 6月6日 12月6日 1天 76 308 タマ
• H HDase	•			권 300b/s	E 39.1K/s
• B HDFS - >1				g 100b/s	8 19.5K/s
V W HWe	•			04:15 04:30	04:16 04:30
CooKeeper	•			■答 DataNode 333b/s ■答 DataNodes 5.2b/s	■各网络胺口 41.4K/s ■各网络胺口 42.1K/s
		群衆磁盘 IO	群集 CPU		
Cloudera Management Service		2 1.9M/s	50%		
🛇 🖸 Cloudera M 🥕 4		p///vs			
		■ 普磁盘中的 1.5M/s ■ 普磁盘中的 814K	/s = Cluster 1,整个主机中的主机 CPU 使用率 21.1%		

CDH 就部署成功。

# 5.6 Hadoop 配置 HA

5.6.1 HDFS 配置高可用配置

1) HDFS 服务---》操作--》启用 High Availability

cloudera MANA	GER	群集 ▼	主机 ▼
HDFS (Cluster 1)	操作▼		
状态实例配置命令	启动 停止		
HDFS 汇总	<b>重启</b> 滚动重	启	
配置的容量	添加角	色实例	_
	重命名	5	
运行状况测试	进入维	护模式	
♥ 显示 7 个运行良好状况	重新平	<sup>Z</sup> 衡	
状态摘要	部署書 升级 H 完成溪	F户端配置 IDFS 元数据 該动升级	II Jay
DataNode	下载者	沪端配置	
NameNode	启用⊦	ligh Availab	ility
SecondaryNameNode	滚动编	辑	
主机	设置H	IDFS 静态数	7据加密

2) 制定 HA 名字的名称

启用 HDFS 的 Hig	h Availability	
入门 此向导引导您添加备用 NameN	ode,重启此 HDFS 服务和任何依赖服务,然后重	新部署客户端配置。
Nameservice 名称	nameservice1	
	启用 High Availability 将创建新的 nameservice。	请接受默认名称 nar

3) 指定 NameNode 节点的主机名称, 2个。选择 JournalNodeJN 的主机名,奇数

个,一般3个

启用 HDFS 的 Hig	h Availability	
万配用巴		
NameNode 主机	solway-nm1.cn (当前)	
	solway-nm2.cn	]
JournalNode 主机	solway-hbase1.cn; solway-n	
	我们建议在主机上运行 JournalNod 以上的奇数 JournalNode。	e 时使用与 NameNode 相同的硬件规格。
	以上的奇数 JournalNode。	

4) 制定多个磁盘的多个目录

服务 HDFS		
NameNode 数据目录* dfs.namenode.name.dir	solway-nm1	<b>/hadoop/dfs/nn</b> 继承自:NameNode Default Group
	solway-nm2	<b>/hadoop/dfs/nn</b> 继承自:NameNode Default Group
JournalNode 编辑目录* dfs.journalnode.edits.dir	solway-hbase1	/hadoop/dfs/jn
solway-nm1		重苦为空默认值 ↔ /hadoop/dfs/jn
		<u>重置为空默认值</u> 为
	solway-nm2	/hadoop/dfs/jn 軍業社会課題/信告
		里已为全私队阻节

**启用 HDFS 的 High Availability** 基意您! 己成功应用 High Availability。 完成本的导版必须手动执行下列步骤: ・ 対于每个 Hive 服务 Hive,停止 Hive 服务,将 Hive Metastore 数据库备份到永久性存储中,运行服务命令"更新 Hive Metastore NameNodes",然后重应 Hive 服务。

启动后,一个NN 是活动,一个NN 是备份状态。

## 5.6.2 YARN HA 配置

1) YARN 服务--》操作--》启用 High Availability

YARN (MR2)	Included) ( Cluster 1 )	操作▼		
状态实例配置命令。	如用程序 资源池 图表库 审核	启动 <b>停止</b>		
运行状况测试		<b>重启</b> 滚动重启		
●显示3个运行良好状况		添加角色实例		
YARN Container Usage Ag This health test is disable YARN.	gregation d because container usage metric i	重命名		
状态摘要		进入维护模式		
JobHistory Server	●1 个运行状况良好	部署客户端配置 创建作业历史记录目录		
NodeManager	●3 个运行状况良好	创建 NodeManager 远程应用程序日志目录		
ResourceManager	●1 个运行状况良好	导入 MapReduce 配置 格式化 StateStore		
主机	●4 个运行状况良好	Create CM Container Usage Metrics Dir		
运行状况历史记录		下载客户端配置		
▶ ● 晚上10点00	NodeManager 运行状况 良好	启用 High Availability		

2) 指定 RM 的 2 个主机名

启用 YARN (MR2 Included) 的 High Availability						
入门 此向导引导您添加备用 ResourceManager, 重启此 YARN (MR2 Included) 服务和任何依赖服务, 然后重新部署客户端配置。						
ResourceManager 主	solway-nm1.cn (当前)					
0.6	solway-nm2.cn					

## 5.6.3 HBASE HA 配置

1) Hbase 服务--》实例--》添加角色实例

● HBase ( Cluster 1 ) 操作・							
状态实例的配置命令图表	状态实例 配置命令 图表库表统计信息 审核 HBase Web UI @ 快速链接 ▼						
筛选器		搜索					
✔ 状态		已选定	的操作• 添加角色实例	角色组			
● 存在隐患的运行状况	2	÷.	角色类型	* 状态 🕴	主机		
● 运行状况良好	2		Master (活动)	已启动	esgzb-del-n004.esgyn.com		
> 授权状态			RegionServer	已启动	esgzb-del-n005.esgyn.com		
> 维护模式			RegionServer	已启动	esgzb-del-n007.esgyn.com		
> 机架			RegionServer	已启动	esgzb-del-n006.esgyn.com		

2) 点击 Master 下的选择主机

添加角色实例到 HBase					
自定义角色分配 <sup>您可以在此处为新角色指定角色分配。</sup>					
还可以按主机查看角色分配。 按主机查看					
Master × 1	HBTS HBase Thrift Server				
选择主机	选择主机				
R8 RegionServer × 3					
选择主机・					

3) 选择多个主机

Г

为新	的或现有角色选择主机。主机列	表E	轻过筛选,以	HU P	余无效候选主	机;这些主	机包括:运行机	状况不信
FIBUL	CENTRANT . NOSLOT, NOSLOT-1	~ <u>[</u>	IF ABALEWING	0			152.65	
	主机名称	0	IP 地址	÷	机架 👌	内核	物理内存	现有角
C	esgzb-del-n004.esgyn.com		10.10.14.13		/default	4	31.5 GiB	м
C	esgzb-del-n005.esgyn.com		10.10.14.14		/default	4	31.5 GiB	RS
	esgzb-del-n006.esgyn.com		10.10.14.15		/default	4	31.5 GiB	R\$
	esgzb-del-n007.esgyn.com		10.10.14.16		/default	4	31.5 GiB	R\$

添加角色实例

添加角色实例到 HBase						
自定义角色分配 您可以在此处为新角色指定角色分配。						
还可以按主机查看角色分配。  按主机查看						
Master × (1+1新建)	HBIS HBase Thrift Server					
esgzb-del-n005.esgyn.com -	选择主机					
Rs RegionServer × 3						
选择主机・						

继续下一步

● HBase ( Cluster 1 ) 操作▼						
状态 实例 配置 命令 图表库表统计信息 审核 HBase Web UI e 快速链接 -						
筛选器	搜索					
✔ 状态	已选定的操作 - 添加角色实例	角色组				
●已停止 1	□ ♦ 角色类型 ◆	状态	主机			
<ul> <li>存在隐患的运行状况 3</li> <li>运行状况良好 1</li> </ul>	Master	已停止	esgzb-del-n005.esgyn.com			
	🥅 🌒 Master (活动)	已启动	esgzb-del-n004.esgyn.com			
> 授权状态	🔲 😑 RegionServer	已启动	esgzb-del-n005.esgyn.com			
/ 班// 保巧 √ √ 小 如	🔲 兽 RegionServer	已启动	esgzb-del-n007.esgyn.com			
>角色组	🔲 🔴 RegionServer	已启动	esgzb-del-n006.esgyn.com			

4) 启动新添加的 Master 服务即可, 一个 Master 状态是活动, 一个 Master 状态时备份。

搜索				
已选定的操作▼	添加角色实例	角色组		
■   角色类型	t +	状态	÷	主机
🔲 🥚 Master (†	昏份)	已启动		esgzb-del-n005.esgyn.com
🔲 🥚 Master ()	舌动)	已启动		esgzb-del-n004.esgyn.com
🔲 🥚 RegionS	erver	已启动		esgzb-del-n005.esgyn.com
🔲 🔴 RegionS	erver	已启动		esgzb-del-n007.esgyn.com
🔲 🔴 RegionS	erver	已启动		esgzb-del-n006.esgyn.com

## 5.6.4 HIVE HA 配置

1) 添加 HIVE MetaStore Server, 保证 2 个提供 HA.

E	选定	的操作▼		
	÷.	*角色类型	↑ 状态	♦ 主机
	•	Gateway	不适用	solway-hbase2.cn
	0	Gateway	不适用	solway-hbase3.cn
	•	Gateway	不适用	solway-hbase1.cn
	•	Hive Metastore Server	已启动	solway-nm2.cn
	•	Hive Metastore Server	已启动	solway-nm1.cn
	•	HiveServer2	已启动	solway-hbase2.cn
	•	HiveServer2	已启动	solway-hbase3.cn
	•	HiveServer2	已启动	solway-hbase1.cn

# 5.7 Hadoop 服务优化

## 5.7.1 HDFS heap 大小参数调优

HDFS 属性	推荐设置	备注
DataNode Java Heap Size	4 GiB	在大型配置中应用此设置。
NameNode Java Heap Size	4 GiB	在大型配置中应用此设置。
Secondary NameNode Java	4 GiB	在大型配置中应用此设置。
Heap Size		
dfs.datanode.handler.count	64	
dfs.namenode.handler.count	100	

## 5.7.2 ZooKeeper 参数调优

Zookeeper 配置属性名	配置值	缺省值
maxClientCnxns	0	60
maxSessionTimeout	60000	40000

ZooKeeper Server 的 Java 堆	1G	1G
栈大小 (字节)		

5.7.3 HBase 参数调优

HBase 配置属性	推荐设置	备注
hbase.rpc.timeout	10 分钟	此设置取决于表的大小。
		默认值是 60 seconds。 对
		于大表需要调高这个值。
		此值需保持跟参
		数 hbase.client.scanner.timeo
		ut.period 一样。我们发现
		把这个值调为 600 秒会减
		少超时相关的错误,譬
		∲ OutOfOrderNextExcepti
		on 错误。
hbase.regionserver.le	60 分钟	跟上面的配置类似, 默认
ase.period,		值是 60 秒。根据用户表大
hbase.client.scanner.		小的不同,在做 count(*)和
timeout.period		update statistics 命令时我们
		有遇到过超时相关的错
		误。底层根源是 HBase 的
		coprocessor 执行时间问
		题。
hbase.snapshot.master	10 分钟	HBase 默认设置是 60000 毫
.timeout.millis		秒。如果您使用 Trafodion
hbase.snapshot.master		Bulk Loader 或其他语句时
.timeoutMillis		遇到与 LIBasa 协照相关的
hbase.snapshot.region		些刊与 FIDASE 依照相大的
.timeout		超时问题,可以设置这两

		个 HBase 的属性为 10 分钟
		(600,000 毫秒)的值。
hbase.hregion.max.fil	107374182400 bytes	HBase 默认设置是
esize		10737418240 (10 GB)。我们
		把这个值增加到
		107374182400 (100 GB), 这
		会减少每个 HBase 表的
		HStoreFiles 数目,另外似
		乎也会减少因为 region 分
		裂对当前活动事务的干
		扰。
hbase.hstore.blocking	10	CDH 5.4 默认值是 10。安
StoreFiles		装 QianBase 之后 Installer 会
		改成 200.这也是 HBase 早
		期版本的默认值。
hbase.regionserver.ha	参见备注	此设置需要跟 mxosrvr 的并
ndler.count		发会话数目设置匹配。默
		认值是 30。
HBase	2G	缺省 1G
Master 的 Java 堆栈大小		
(字节)		
HBase	31G	缺省 4G
RegionServer 的 Java		
堆栈大小 (字节)		
客户端 Java 堆大小 (字	256M	缺省 256M
节)		
HBase Memstore 刷新大	512M	缺省 128M

小		
hbase.hregion.memstor		
e.flush.size		
hbase.hregion.majorco	0	缺省7天
mpaction		
hbase.regionserver.ha	200	缺省 10
ndler.count		

# 6. 安装与配置 OM

本章介绍了如何安装与配置 Operational Management。

6HA 的安装

6.2 与数据库集成的组件

## 6.1 OM HA 的安装

6.1.1 准备安装包

### 6.1.2 安装 OM 基础包

#### 6.1.2.1 解压 OM 基础包

```
[root@hxpernode01 data]# tar -xzvf OM_KubeOffline-0.1.3-
20191105.tar.gz
```

#### 6.1.2.2 安装 OM 基础包

(注: --hosts 后面的机器名必须小写,并且必须填写主机的短域名)

[root@hxpernode01 d	ata]# cd OM_KubeOffli	ne-0.1.3		
[root@hxpernode01	OM_KubeOffline	e-0.1.3]#	./install.py	hosts
hxpernode01,hxperno	de02,hxpernode11			
日志打印如下表示安	装成功,			
PLAY				RECAP
*****	******	*****	*****	*****
******	******	*****	*****	*****
****				
hxpernode01	: ok=249 changed=75	unreachable=0	failed=0	

hxpernode02 : ok=249 changed=80 unreachable=0 failed=0
hxpernode11 : ok=313 changed=101 unreachable=0 failed=0
localhost : ok=1 changed=0 unreachable=0 failed=0
Friday 22 November 2019 13:18:54 +0800 (0:00:00.099) 0:05:43.467 ******
kubernetes-apps/rotate_tokens : Rotate Tokens   Test if default certificate is expired
30.38s
container-engine/docker : Ensure docker packages are installed
22.01s
kubernetes/master : kubeadm   Initialize first master
20.29s
kubernetes/master : kubeadm   Init other uninitialized masters
19.02s
download : container download   Load containers from registry pull is required or told to
always pull (all nodes) 14.52s
etcd : reload etcd
10.60s
etcd : wait for etcd up
7.26s
kubernetes/preinstall : Copy files to remote locations
5.81s
etcd : Gen_certs   Write etcd master certs
5 538
download · file download   Copy it
3.81s
container engine/docker · file_download   Conv_it
arthur facts from all instances
kubernetes/master : kubeadm   write out kubeadm certs
3.36s
kubernetes-apps/ansible : Kubernetes Apps   Lay Down CoreDNS Template
2.99s
kubernetes-apps/ansible : Kubernetes Apps   Start Resources

kubernetes-apps/network_plugin/calico : Start Calico resources
2.43s
policy_controller/calico : Start of Calico kube controllers
2.21s
kubernetes-apps/rotate_tokens : Rotate Tokens   Delete expired tokens
2.11s
etcd : Gen_certs   Gather etcd master certs
2.03s
network_plugin/calico : Calico   Create calico manifests
2.01s
Congratulation

问题一:大小写机器名不支持

```
fatal: [HXnode13]: FAILED! => {
    "assertion": "inventory_hostname is match(\"[a-z0-9]([-
a-z0-9]*[a-z0-9])?(\\.[a-z0-9]([-a-z0-9]*[a-z0-9])?)*$\")",
    "changed": false,
    "evaluated_to": false,
    "msg": "Hostname must consist of lower case alphanumeric
characters, '.' or '-', and must start and end with an
alphanumeric character"
}
...ignoring
```

#### 解决办法:

```
[root@HXnode01 OM_KubeOffline-0.1.3]# grep 'Stop if bad
hostname' * -R
extra_playbooks/roles/kubernetes/preinstall/tasks/0020-
verify-settings.yml:- name: Stop if bad hostname
roles/kubernetes/preinstall/tasks/0020-verify-settings.yml:-
name: Stop if bad hostname
%改 roles/kubernetes/preinstall/tasks/0020-verify-
settings.yml
- name: Stop if bad hostname
assert:
that: inventory_hostname is match("[a-z0-9]([-a-z0-9]*[a-
z0-9])?(\.[a-z0-9]([-a-z0-9]*[a-z0-9])?)*$")
```

```
msg: "Hostname must consist of lower case alphanumeric
characters, '.' or '-', and must start and end with an
alphanumeric character"
  #ignore_errors: "{{ ignore_assert_errors }}"
  ignore errors: true
```

#### 问题二: libcgroup 包版本问题

#### 解决办法:

```
root@HXnode01 OM KubeOffline-0.1.3]# grep 'Ensure docker
packages are installed' * -R
extra playbooks/roles/container-
engine/docker/tasks/main.yml:- name: Ensure docker packages
are installed
roles/container-engine/docker/tasks/main.yml:- name: Ensure
docker packages are installed
(1) 修改 roles/container-engine/docker/tasks/main.yml
#no log: true
no log: false
(2) 重新执行打印详细日志
TASK [container-engine/docker : Ensure docker packages are
installed]
****
Friday 08 November 2019 11:52:22 +0800 (0:00:03.698)
0:00:07.176 ******
```

```
fatal: [localhost]: FAILED! => {"changed": true, "cmd": "yum
install -y -q audit-2.8.4-4.el7.x86 64.rpm audit-libs-2.8.4-
4.el7.x86 64.rpm libselinux-2.5-14.1.el7.x86 64.rpm
libselinux-utils-2.5-14.1.el7.x86_64.rpm libsepol-2.5-
10.el7.x86 64.rpm libselinux-python-2.5-14.1.el7.x86 64.rpm
libsemanage-2.5-14.el7.x86 64.rpm policycoreutils-2.5-
29.el7_6.1.x86_64.rpm selinux-policy-3.13.1-
229.el7 6.15.noarch.rpm selinux-policy-targeted-3.13.1-
229.el7 6.15.noarch.rpm libsemanage-python-2.5-
14.el7.x86 64.rpm audit-libs-python-2.8.4-4.el7.x86 64.rpm
setools-libs-3.3.8-4.el7.x86 64.rpm python-IPy-0.75-
6.el7.noarch.rpm libcgroup-0.41-20.el7.x86 64.rpm
checkpolicy-2.5-8.el7.x86 64.rpm policycoreutils-python-2.5-
29.el7 6.1.x86 64.rpm container-selinux-2.107-
1.el7 6.noarch.rpm audit-2.8.4-4.el7.x86 64.rpm
containerd.io-1.2.6-3.3.el7.x86 64.rpm docker-ce-cli-18.09.5-
3.el7.x86 64.rpm docker-ce-18.09.5-3.el7.x86 64.rpm",
"delta": "0:00:01.405806", "end": "2019-11-08
11:52:24.303285", "msg": "non-zero return code", "rc": 1,
"start": "2019-11-08 11:52:22.897479", "stderr": "Error:
Package: libcgroup-tools-0.41-13.el7.x86 64 (@iso) \n
Requires: libcgroup(x86-64) = 0.41-13.el7 \n
Removing: libcgroup-0.41-13.el7.x86 64 (@iso)\n
libcgroup(x86-64) = 0.41-13.el7 n
                                            Updated By:
libcgroup-0.41-20.el7.x86 64 (/libcgroup-0.41-
                               libcgroup(x86-64) = 0.41-
20.el7.x86 64)\n
20.el7", "stderr lines": ["Error: Package: libcgroup-tools-
0.41-13.el7.x86 64 (@iso)", "
                                       Requires:
libcgroup(x86-64) = 0.41-13.el7", "
                                              Removing:
libcgroup-0.41-13.el7.x86 64 (@iso)", "
libcgroup(x86-64) = 0.41-13.el7", "
                                              Updated By:
libcgroup-0.41-20.el7.x86 64 (/libcgroup-0.41-
20.el7.x86 64)", "
                                libcgroup(x86-64) = 0.41-
20.el7"], "stdout": " You could try using --skip-broken to
work around the problem \n** Found 3 pre-existing rpmdb
problem(s), 'yum check' output follows:\nlibcom err-devel-
1.42.9-10.el7.x86_64 has missing requires of libcom_err(x86-
```

```
64) = ('0', '1.42.9', '10.el7')\nlibselinux-devel-2.5-
11.el7.x86 64 has missing requires of libselinux(x86-64) =
('0', '2.5', '11.el7')\nlibsepol-devel-2.5-6.el7.x86 64 has
missing requires of libsepol(x86-64) = ('0', '2.5',
'6.el7')", "stdout lines": [" You could try using --skip-
broken to work around the problem", "** Found 3 pre-existing
rpmdb problem(s), 'yum check' output follows:", "libcom_err-
devel-1.42.9-10.el7.x86 64 has missing requires of
libcom err(x86-64) = ('0', '1.42.9', '10.el7')", "libselinux-
devel-2.5-11.el7.x86 64 has missing requires of
libselinux(x86-64) = ('0', '2.5', '11.el7')", "libsepol-
devel-2.5-6.el7.x86 64 has missing requires of libsepol(x86-
64) = ('0', '2.5', '6.el7')"] \}
... ignoring
(3)移除以下目录的 rpm
[root@HXnode01 OM_KubeOffline-0.1.3]# cd kube-
packages/docker/
[root@HXnode01 docker]# mv libcgroup-0.41-20.el7.x86 64.rpm
/root/
(4)修改 roles/container-engine/docker/tasks/main.yml, 删除以下行
libcgroup-0.41-20.el7.x86 64.rpm
```

#### 问题 3: 节点缺少 libcgroup

```
fatal: [hxpernode11]: FAILED! => {"changed": true, "cmd":
"yum install -y -q audit-2.8.4-4.el7.x86_64.rpm audit-libs-
2.8.4-4.el7.x86_64.rpm libselinux-2.5-14.1.el7.x86_64.rpm
libselinux-utils-2.5-14.1.el7.x86_64.rpm libsepol-2.5-
10.el7.x86_64.rpm libselinux-python-2.5-14.1.el7.x86_64.rpm
libsemanage-2.5-14.el7.x86_64.rpm policycoreutils-2.5-
29.el7_6.1.x86_64.rpm selinux-policy-3.13.1-
229.el7_6.15.noarch.rpm selinux-policy-targeted-3.13.1-
229.el7_6.15.noarch.rpm libsemanage-python-2.5-
14.el7.x86_64.rpm audit-libs-python-2.8.4-4.el7.x86_64.rpm
setools-libs-3.3.8-4.el7.x86_64.rpm python-IPy-0.75-
6.el7.noarch.rpm checkpolicy-2.5-8.el7.x86_64.rpm container-
```

```
selinux-2.107-1.el7 6.noarch.rpm audit-2.8.4-4.el7.x86 64.rpm
containerd.io-1.2.6-3.3.el7.x86 64.rpm docker-ce-cli-18.09.5-
3.el7.x86 64.rpm docker-ce-18.09.5-3.el7.x86 64.rpm",
"delta": "0:00:00.674916", "end": "2019-11-22
12:58:53.110585", "msg": "non-zero return code", "rc": 1,
"start": "2019-11-22 12:58:52.435669", "stderr": "Error:
Package: policycoreutils-python-2.5-29.el7_6.1.x86_64
(/policycoreutils-python-2.5-29.el7 6.1.x86 64) \n
Requires: libcgroup\nError: Package: 3:docker-ce-18.09.5-
3.el7.x86 64 (/docker-ce-18.09.5-3.el7.x86_64)\n
Requires: libcgroup", "stderr lines": ["Error: Package:
policycoreutils-python-2.5-29.el7 6.1.x86 64
(/policycoreutils-python-2.5-29.el7 6.1.x86 64)", "
Requires: libcgroup", "Error: Package: 3:docker-ce-18.09.5-
3.el7.x86 64 (/docker-ce-18.09.5-3.el7.x86 64)", "
Requires: libcgroup"], "stdout": " You could try using --
skip-broken to work around the problem\n** Found 3 pre-
existing rpmdb problem(s), 'yum check' output
follows:\nlibcom err-devel-1.42.9-10.el7.x86 64 has missing
requires of libcom err(x86-64) = ('0', '1.42.9',
'10.el7')\nlibselinux-devel-2.5-11.el7.x86 64 has missing
requires of libselinux(x86-64) = ('0', '2.5',
'11.el7')\nlibsepol-devel-2.5-6.el7.x86_64 has missing
requires of libsepol(x86-64) = ('0', '2.5', '6.el7')",
"stdout lines": [" You could try using --skip-broken to work
around the problem", "** Found 3 pre-existing rpmdb
problem(s), 'yum check' output follows:", "libcom err-devel-
1.42.9-10.el7.x86 64 has missing requires of libcom err(x86-
64) = ('0', '1.42.9', '10.el7')", "libselinux-devel-2.5-
11.el7.x86 64 has missing requires of libselinux(x86-64) =
('0', '2.5', '11.el7')", "libsepol-devel-2.5-6.el7.x86 64 has
missing requires of libsepol(x86-64) = ('0', '2.5',
'6.el7')"]}
... ignoring
```

解决办法:

此节点缺少 libcgroup, 安装 libcgroup [root@hxpernode11 ~]# rpm -qa | grep libcgroup [root@hxpernode11 ~]# yum install -y libcgroup

## 6.1.2.3 验证 OM 基础包安装成功

[root@hxpernod	de01 OM_Ki	ubeOffline	e-0.1.3];	<pre># kubectl get nodes</pre>
NAME	STATUS	ROLES	AGE	VERSION
hxpernode01	Ready	master	3m29s	v1.14.3
hxpernode02	Ready	master	3m29s	v1.14.3
hxpernode11	Ready	master	3m58s	v1.14.3

## 6.1.2.4 检查 docker images

[root@hxpernode01 0	M_KubeOffline-0.1.3]	# <mark>docker images</mark>
REPOSITORY		
TAG	IMAGE ID	CREATED
SIZE		
reg.esgyn.com:5000/	prom/alertmanager	
v0.19.0	30594e96cbe8	2 months ago
53.2MB		
reg.esgyn.com:5000/	grafana	
6.2.5-plugins	f3d4e9c498fe	3 months ago
250MB		
reg.esgyn.com:5000/	redis	
5.0.5-alpine	ed7d2ff5a623	3 months ago
29.3MB		
reg.esgyn.com:5000/jimmidyson/configmap-reload		
v0.3.0	7ec24a279487	3 months ago
9.7MB		
reg.esgyn.com:5000/	logstash	
7.2.0	4470777ac65e	5 months ago
836MB		
reg.esgyn.com:5000/	elasticsearch/elasti	csearch
7.2.0	0efa6a3de177	5 months ago
861MB		
reg.esgyn.com:5000/	google-containers/ku	be-proxy

```
v1.14.3
                   004666307c5b
                                      5 months ago
82.1MB
reg.esgyn.com:5000/google-containers/kube-apiserver
v1.14.3
                  9946£563237c
                                       5 months ago
210MB
reg.esgyn.com:5000/google-containers/kube-controller-manager
v1.14.3
                  ac2ce44462bc 5 months ago
158MB
reg.esgyn.com:5000/google-containers/kube-scheduler
v1.14.3
                  953364a3ae7a
                                       5 months ago
81.6MB
reg.esgyn.com:5000/aledbf/kube-keepalived-vip
0.35
                  dfebe29fdd75
                                      5 months ago
157MB
reg.esgyn.com:5000/prom/prometheus
v2.10.0
                  5d62a6125e7e
                                      6 months ago
123MB
reg.esgyn.com:5000/coredns/coredns
1.5.0
                  7987f0908caf
                                      7 months ago
42.5MB
reg.esgyn.com:5000/registry
2
                   f32a97de94e1
                                     8 months ago
25.8MB
reg.esgyn.com:5000/google-containers/cluster-proportional-
autoscaler-amd64 1.4.0
                                                         9
                                     1243932ce9f7
           45.9MB
months ago
reg.esgyn.com:5000/google-containers/k8s-dns-node-cache
1.15.1
                   abd5f1988d71
                                      10 months ago
77.8MB
reg.esgyn.com:5000/coreos/etcd
v3.2.26
                  e87e7804f2f5
                                     10 months ago
37.6MB
reg.esgyn.com:5000/calico/node
v3.4.0
                  a89b45f36d5e
                                     11 months ago
75.9MB
reg.esgyn.com:5000/calico/cni
v3.4.0
                   d531d047a4e2
                                      11 months ago
```

75.4MB				
reg.esgyn.com:5000/	calico/kube-controll	ers		
v3.4.0	6f1a824b2c81	11	months	ago
56.5MB				
reg.esgyn.com:5000/	google-containers/pa	use		
3.1	da86e6ba6ca1	23	months	ago
742kB				
reg.esgyn.com:5000/	google_containers/pa	use	-amd64	
3.1	da86e6ba6ca1	23	months	ago
742kB				

## 6.1.3 安装 OM HA

#### 6.1.3.1 解压 OM HA Installer

```
[root@hxpernode01 data]# tar -xzv OM_HAInstaller-8fe58d5-
20191108.tar.gz
```

### 6.1.3.2 编辑配置文件

修改 values.yaml 以适合您的集群环境,以下部分是必填项:

• nodes

设置3个Kubernetes节点的主机名,也即kubectl get nodes查到的主机名,例如:

nc	odes:
-	node1

- node2

- node3

• virtualIp

为HA设置虚拟IP(与当前主机同网段的浮动IP),例如:

virtualIp: 10.10.12.222

clusters/nodesIP

为所有节点设置节点 IP (DBMgr 页面所监控集群的数据库节点 IP),例如:

clusters:

```
- name: cluster1
```

# nodes IP list
nodesIP:
- 10.10.12.13
- 10.10.12.15
- 10.10.12.24

OM HA 基础结构可以支持多个集群,您可以在此处添加更多集群,如下所示:

```
clusters:
- name: cluster1
nodesIP:
- x.x.x.x
dbmgrRemoteUr1: http://x.x.x.x:4205
- name: cluster2
nodesIP:
- xx.xx.xx.xx
dbmgrRemoteUr1: https://xx.xx.xx.xx:4206
```

• clusters/dbmgrRemoteUrl

设置远程 DBMgr URL, 例如:

dbmgrRemoteUrl: http://10.10.12.13:4205

如果在远程 DBMgr 上启用了 https, 请设置 https URL, 例如:

dbmgrRemoteUrl: https://10.10.12.13:4206

• clusters/jdbcUrl

设置中间件连接用的主机 IP, 即 DcsMaster 节点的 IP, 例如:

jdbcUrl: 10.10.12.13

cluster/hadoopNode

配置当前集群的 Zookeeper、HDFS namenode、HBase master 节点的 IP,例如:

```
zkNode: 10.10.12.13,10.10.12.15,10.10.12.24
nameNode: 10.10.12.13
hbaseMasterNode: 10.10.12.13
```

您还可以在 values.yaml 中更改其他设置,相关详细信息,请参阅 yaml 文件。

实际使用中具体内容参考如下:

```
[root@esgzb-qa-n109 v2.0.4-0]# cd OM HAInstaller-v2.0.4-0-
q37454764
[root@esgzb-ga-n109 OM HAInstaller-v2.0.4-0-g37454764]# vi
values.yaml
### Default values for OM-HA
# *** DO NOT CHANGE *** unless you know what you are doing
enableOM: true
# to enable HA, you must have at least 3 nodes
# change to false while running OM on a single k8s node
enableHA: true
persistence:
  # Set to 'true' means all data will be saved into local
path on the hosts
  # Set to 'false' means all data will be lost if OM is
uninstalled
  enabled: true
  # define kubernetes nodes hostname here
  # set 3 k8s nodes for OM-HA or only set one node to run on
single node mode
  # set the nodes hostname here, one line for one hostname
  # e.g.
  # - nodel
  # - node2
  # - node3
 hosts:
  - esgzb-qa-n109
  - esgzb-qa-n110
  - esgzb-qa-n111
 volumes:
    # local directory for keeping OM components persistent
```

```
data
    localPath: /omdata
# keepalived virtual IP, set it to a not-used IP
# within the same IP range as the nodes above
# e.g. 192.168.0.252
virtualIp: 10.10.14.203
# set different value if there are more than one
# OM HA cluster in the same IP range
# e.g. 101, 102
vrid: 103
localRegistryUri: reg.esgyn.com:5000
omImageTag: v2.0.4-0-g37454764
tlsSecret:
 certificate: certs/nginx.crt
 key: certs/nginx.key
# OM dependency images
images:
 alertmanager: alertmanager:v0.19.0
 prometheus: prometheus:v2.10.0
 configmapreload: configmap-reload:v0.3.0
 elasticsearch: elasticsearch:7.2.0
 grafana: grafana:6.2.5-plugins
 ldap: openldap:2.4.44
 logstash: logstash:7.2.0
  redis: redis:5.0.5-alpine
 singleredis: redis:5.0.7-debian-9-r12
 keepalived: kube-keepalived-vip:0.35
 mariadb: mariadb:10.3.22
# multi-cluster support
clusters:
```

```
- name: cluster1
    # set *** EsgynDB nodes IP *** here, one line for one IP
    # e.g.
    # - 192.168.0.1
    # - 192.168.0.2
    # - 192.168.0.3
    nodesIP:
    - 10.10.14.110
    - 10.10.14.111
    - 10.10.14.112
     # e.g. http://192.168.0.1:4205
    dbmgrRemoteUrl: https://10.10.14.110:4206
    # used by middleware, e.g. 192.168.0.1
    jdbcUrl: 10.10.14.110
    # separated by comma if more than one, should be node IP
    # e.g. zkNode: 192.168.0.1, 192.168.0.2, 192.168.0.3
    zkNode: 10.10.14.109,10.10.14.110,10.10.14.111
    nameNode: 10.10.14.109
    hbaseMasterNode: 10.10.14.109
    # cluster id and instance id should be the same as remote
EsgynDB
    # cluster id and instance id must be integer
    clusterId: 1
    instanceId: 1
middleware:
  enabled: true
  port: 22400
  # If multi-cluster is enabled, the node port of the first
cluster is 32400
  \# the node port of the second cluster is 32401, and so on
  nodePort: 32400
```

```
elasticsearch:
  # this port should be equal to the one configured in mds on
esgyn node
 nodePort: 30002
  # set to a larger value if your cluster is powerful
  # e.g. 1, 4, 8
  jvmHeapSizeGB: 1
logstash:
  # this port should be equal to the one configured in
filebeat on esgyn node
 nodePort: 30003
  # set to a larger value if your cluster is powerful
  # e.g. 1, 4, 8
  jvmHeapSizeGB: 1
dbm:
 enableHttps: true
 port: 4203
 httpPort: 80
  # this is dbmgr web http port
 httpNodePort: 30005
 httpsPort: 443
  # this is dbmgr web https port
 httpsNodePort: 30006
  timeZone: Asia/Shanghai
ldap:
  #ldapHost:
 ldapNodePort: 30389
  ldapBaseDn: ou=Users,dc=esgyn,dc=local
 ldapRootDn: cn=Manager,dc=esgyn,dc=local
 ldapRootPwd: ldap123
  # If TLS is used, this path for CA
  #ldapTlsCa:
  # If the server requires a client cert, the path to that
TLS cert. If this is passed, ldapTlsKey must also be passed
```

```
#ldapTlsCert:
  # If the server requires a client key, the path to that TLS
key. If this is passed, ldapTlsCert must also be passed
  #ldapTlsKey:
  # If specified, expect this name for TLS handshakes rather
than using the hostname parsed from ldapUri
  #ldapTlsServerName:
  # If given, do not do any verification of the server's
cert. Insecure and allows for MITM
  ldapTlsSkipVerify: false
 olcSuffix: dc=esgyn,dc=local
 organization: esgyn
 organizationalUnit: Users
  DBRoot:
    uid: trafodion
    sn: db root
   cn: DB ROOT
    givenName: db root
   displayName: DB ROOT
   mail: db_root@esgyn.local
    userpassword: traf123
  DBAdmin:
    uid: admin
    sn: db admin
    cn: DB ADMIN
    givenName: db admin
    displayName: DB ADMIN
    mail: db admin@esgyn.local
    userpassword: admin123
prometheus:
  # these ports should be equal to the ones configured in
esgyn/node exporter on esgyn node
 nodeExporterPort: 23301
 esgynExporterPort: 23300
 mdsPort: 8989
```

```
mgmtexporter:
  port: 23302
  zkPort: 2181
  hadoopPort: 50070
  hbasePort: 60010
# Password for MySQL root user(Default: use empty password)
mariadb:
  mysqlRootPassword: ""
```

### 6.1.3.3 安装 OM HA

安装过程大概需要3分钟。

```
[root@hxpernode01 OM HAInstaller-8fe58d5]# pwd
/data/OM HAInstaller-8fe58d5
[root@hxpernode01 OM HAInstaller-8fe58d5]# ./install.py
Generating a 2048 bit RSA private key
.+++
.....+++
writing new private key to '/tmp/nginx.key'
secret/nginxsecret created
Loading images...
Load image: dbm:8fe58d5
Load image: es-adapter:8fe58d5
Load image: dbm-web:8fe58d5
Push image: reg.esgyn.com:5000/dbm:8fe58d5
The push refers to repository [reg.esgyn.com:5000/dbm]
91af1ba8ea07: Preparing
862953ce6689: Preparing
8942b2346f53: Preparing
30ea022dfc86: Preparing
4ddb96edc20d: Preparing
03901b4a2ea8: Preparing
```

```
03901b4a2ea8: Waiting
30ea022dfc86: Pushed
03901b4a2ea8: Mounted from redis
91af1ba8ea07: Pushed
862953ce6689: Pushed
4ddb96edc20d: Pushed
8942b2346f53: Pushed
8fe58d5: digest:
sha256:f2a8c138a03b986c486e0a3620aab1e8505690709c857f96d6a90e
5b11fdd71e size: 1572
Push image: reg.esgyn.com:5000/es-adapter:8fe58d5
The push refers to repository [reg.esgyn.com:5000/es-adapter]
296f7ee7345b: Preparing
03901b4a2ea8: Preparing
03901b4a2ea8: Mounted from dbm
296f7ee7345b: Pushed
8fe58d5: digest:
sha256:61cc1aa7a5e8ea33b2f5a71aea06742b585d2f91a019668342bd41
f75d9af04f size: 739
Push image: reg.esgyn.com:5000/dbm-web:8fe58d5
The push refers to repository [reg.esgyn.com:5000/dbm-web]
ef9f22ff3a06: Preparing
4d7837279dad: Preparing
ca7ae868699b: Preparing
03de76c18fcf: Preparing
03901b4a2ea8: Preparing
03901b4a2ea8: Mounted from es-adapter
ca7ae868699b: Pushed
4d7837279dad: Pushed
03de76c18fcf: Pushed
ef9f22ff3a06: Pushed
8fe58d5: digest:
sha256:486c1db40602d4527b604284dafc77a44675e4f8eb8f347d4c2a9e
58bf479b44 size: 1362
```

Release "esgyn-om" does not exist. Installing it now. OM HA componnets had been deployed successfully Please check the status of OM by running below command: kubectl get pods Please wait until all pods are in running state

### 6.1.3.4 检查安装是否成功

[root@hxpernode01 OM_HAInstaller-8fe58d5]# <a href="https://kubectl.get.pods">kubectl.get.pods</a>				
NAME	READY	STATUS	RESTARTS	
AGE				
alertmanager-76f6bcb6b4-s2tzj	2/2	Running	0	
2m25s				
dbm-6d49fc5c88-gkf89	1/1	Running	2	
2m25s				
dbm-6d49fc5c88-gkzzn	1/1	Running	2	
2m25s				
dbm-web-7698fc864-hrzzp	1/1	Running	0	
2m25s				
dbm-web-7698fc864-sdq2x	1/1	Running	0	
2m25s				
elasticsearch-0	1/1	Running	0	
2m25s				
elasticsearch-1	1/1	Running	0	
2m25s				
elasticsearch-2	1/1	Running	0	
2m25s				
es-adapter-1-6d9665cd48-mcvn6	1/1	Running	0	
2m25s				
grafana-1-54b9f844d-4j7jc	1/1	Running	0	
2m25s				
kube-keepalived-vip-2jmzb	1/1	Running	3	
2m25s				
kube-keepalived-vip-b4kqb	1/1	Running	0	
2m25s				
kube-keepalived-vip-vkxn7	1/1	Running	0	
2m25s				

logstash-0	1/1	Running	0
2m25s			
logstash-1	1/1	Running	0
113s			
logstash-2	1/1	Running	0
79s			
prometheus-1-78b8974589-fft4x	2/2	Running	0
2m25s			
redis-ha-server-0	2/2	Running	0
2m25s			
redis-ha-server-1	2/2	Running	0
2m7s			
redis-ha-server-2	2/2	Running	0
115s			

# 6.1.4 验证 OM HA

浏览器输入网址验证: 10.19.41.204:30005 (这个 IP 是 values.yaml 中配置的

virtualIp)

こ、易鯨捷		Ū.		🕲 ل 📀 🗘 🖉 A.db_root 🕶	
6	数据库管理器	篇页 / <b>实别</b>			自动副新: 毎30 秒 🗸 😡 新
<ul> <li>点 実例</li> <li>○ 监括</li> <li>主び</li> <li>系統</li> <li>密数</li> <li>音響</li> <li>追線</li> <li>勧</li> </ul>	2 A 示板 (以表板 ) 線の発板 線の発	1-Cluster1 License expire date: 2020-12-31 License tope:: Demo Product version:: QianBase 1.6.2 Database version: EsgynD8 Advanced 1.6.2	xt2.novalocal	stő. novalocal NIT. novalocal NIT. novalocal stő. novalocal	2-Cluster2 License expire date: 2020-12-31 License rodes: 10 License type: Demo Product version: QianBase 1.6.2 Database version: EsgynD8 Advanced 1.6.2
<ul> <li>「「」</li> <li>「」</li> <l< td=""><td>中央数 ^ 作会戦 和臣国 ^ 時間 中心理 所 ^ 平 計 らの折</td><td></td><td></td><td></td><td></td></l<></ul>	中央数 ^ 作会戦 和臣国 ^ 時間 中心理 所 ^ 平 計 らの折				

### 6.1.5 修改数据库配置文件

安装 OM HA 后,易鲸捷数据库需要修改两个配置文件。具体修改文件为

mdsconf.yaml 及 filebeat.yaml。

[trafodion@hxpernode03 ~]\$ cd /etc/trafodion/conf/dcs/				
[trafodion@hxpernode03 dcs]\$ vi mdsconf.yaml				
修改esurl: <u>http://10.19.41.49:9200</u> 为esurl:				
http://10.19.41.204:30002				
<pre>[trafodion@hxpernode03 dcs]\$ pdcp \$(trafconf -wname)</pre>				
mdsconf.yaml \$PWD				
[trafodion@hxpernode03 dcs]\$ cd				
/etc/trafodion/conf/mgblty/filebeat/				
[trafodion@hxpernode03 filebeat]\$ vi filebeat.yml				
修改 hosts: ["10.19.41.49:5044"]为 hosts:				
["10.19.41.204:30003"]				
<pre>[trafodion@hxpernode03 filebeat]\$ pdcp \$(trafconf -wname)</pre>				
filebeat.yml \$PWD				

也可重新安装数据库,在安装过程中选择启用 OM HA,并指定浮动 IP。

## 6.1.6 重启数据库 OM CLIENT 服务

在每个数据库节点,重启 om-client 服务

```
[trafodion@hxpernode03 filebeat]$ om-client restart
2019-11-22_14:16:48: OptMgmt client components are running
2019-11-22_14:16:48: Stopping OptMgmt client components ...
2019-11-22_14:16:48: Stopped OptMgmt client components
2019-11-22_14:16:49: OptMgmt client components are stopped
2019-11-22_14:16:49: Starting OptMgmt client components ...
2019-11-22_14:16:50: Started OptMgmt client components
```

### 6.1.7 重启数据库 MDS 服务

此步骤是为了使得 MDS 重新生效,若不重启 MDS,则 mds 日志中会报错

" [Info]2019/11/22 13:53:39 Error getting response: dial tcp 10.19.41.49:9200:

connect: connection refused"

[trafodion@hxpernode03 trafodion]\$ mds-client restart

### 6.1.8 重新配置数据库的 LDAP 用户认证

此配置是为了更改数据库的 ldap server,改用 OM ldap server,使得 dbmgr 和数据库用同一个 ldap,保证一致性,避免用户认证失败。使用 python installer 中的 auto\_config.py –ldap 来重新配置 ldap (该脚本需要在安装了数据库的节点以 root

用户执行):

[root@esgzb-qa-n110 python-installer]# ./auto config.py -ldap Trafodion Automatic Configuration Tool \*\*\*\*\* Modify LDAP security (Y/N) [N]: y Enter LDAP user name to be assigned DB root privileges (DB ROOT) [trafodion]: Enter LDAP user name to be assigned DB Admin privileges [admin]: Enter LDAP user password to be assigned DB Admin privileges: Confirm Enter LDAP user password to be assigned DB Admin privileges: Enter list of LDAP Hostnames (comma separated if more than one host): 10.10.14.203 --这个 IP 是 OM 中的 virtual Ip Enter LDAP Port number (Example: 389 for no encryption or TLS, 636 for SSL) [389]: 30389 --这个 Port 是 OM 中配置的默认端口号, 若未改过就填 30389 Enter all LDAP unique identifiers (semi-colon separated if more than one identifier) [uid=,ou=Users,dc=esgyn,dc=local]: Enter LDAP Encryption Level (0: Encryption not used, 1: SSL, 2: TLS) [0]:

Does the LDAP server require search user name/password (Y/N) (optional) [N]: Does the LDAP server support group searches (Y/N) (optional) [N]: ...

也可重新安装数据库,在安装过程中选择使用 OM HA 的 LDAP。

## 6.1.9 重新验证 OM 监控界面

こ 易鯨捷	· · · · · · · · · · · · · · · · · · ·		🕲 🗘 🧿 🙆 Adb_root 🔻	
し 数据库管理器	篇页 / <b>实</b> 制			自动刷新: 每 30 秒 🗸 🛤 新
▲ 実例 ○ 监控 ▲ 主化表板 系统化表板 数通库化表板 告答 违建级另 数	1-Cluster1 License expire date: 2020-12-31 License nodes: 10 License type: Demo Product version: QianBase 1.6.2 Database version: EigynDB Advanced 1.6.2	st2.novalocal at3.novalocal	silő.novalocal silő.novalocal silő.novalocal	2-Cluster2 License expire date: 2020-12-31 License nodes: 10 License type: Demo Product version: QianBase 1.6.2 Database version: EsgynDB Advanced 1.6.2
<ul> <li>③ 工作负载 ^</li> <li></li></ul>				
<ul> <li>所戸設置</li> <li>前建本</li> <li>合 诊断 ^</li> <li>SQL申计</li> <li>日志</li> <li>Cone分析</li> </ul>				

现在左侧菜单栏可以看到所有的菜单选项:

## 6.2 与数据库集成的组件

## 6.2.1 安装

Esgyn DBManager 与数据库集成的组件以及 omclient (filebeat、node\_exporter、

esgyn\_exporter)随易鲸捷数据库核心组件一并安装与配置,无需额外安装。

Esgyn DBManager 安装配置文件位于\$TRAF\_CONF/dbmgr 文件夹中。

由于这些相关组件和配置文件涉及到 OM HA 安装后的服务,所以建议先安装

OM HA, 然后再来安装易鲸捷数据库, 也就无需再手动更改相关配置。
### 6.2.2 配置

Esgyn DBManager 从配置文件中读取其运行时配置,该配置文件是根据安装易

鲸捷数据库时 python installer 按所选的选项产生的,一般不需要用户手动修改。

Esgyn DBManager 与数据库集成的组件的配置文件路径如下:

\$TRAF CONF/dbmgr/config.xml 配置文件,以下为配置文件的说明。

#### 6.2.2.1 配置属性 (Configuration Property)

• jdbcUrl

易鲸捷数据库连接 DCS Master 的 JDBC 地址。

<entry

key="jdbcUrl">jdbc:t4jdbc://my.esgyndb.server:23400/:</entry
>

如果配置了多个 DCS Master,则会看到以逗号分隔的 DCS Master。JDBC 驱动程序会尝试连接到第一台服务器,如果失败,它会尝试连接下一个服务器。

>jdbc:t4jdbc://my.esgyndb.server1:23400,my.esgyndb.server2:2
3400/:</entry>

adminUserID

用于 JDBC 连接的数据库用户 ID,运行内部系统查询。

<entry key="adminUserID">DB ADMIN</entry>

adminPassword

用于 JDBC 连接的数据库用户 ID 的密码,运行内部系统查询。

安装程序对密码进行了加密。

<entry key="adminPassword">OBF:1sov1sov</entry>

connectionTimeout

JDBC 连接(连接 Esgyn DBManager 和易鲸捷数据库)的超时时间。

单位为秒, 默认值为 60 秒。

<entry key="connectionTimeout">60</entry>

• MinPoolSize

易鲸捷数据库的内部查询使用连接池。

该参数指定连接池的初始值和最小值。

默认值为2。

<entry key="minPoolSize">2</entry>

• MaxPoolSize

易鲸捷数据库的内部查询使用连接池。

该参数指定连接池的最大值。

默认值为8。

<entry key="maxPoolSize">8</entry>

• sessionTimeoutMinutes

会话超时时间。超过该值后,系统将提示您重新登陆 Esgyn DBManager。

单位为分钟, 默认值为120分钟。

<entry key="sessionTimeoutMinutes">120</entry>

• timeZoneName

易鲸捷数据库服务器的时区名称。

必须使用规范的时区格式,例如,America/New\_York。

<entry key="timeZoneName">America/New\_York</entry>

### • trafodionRestServerUri

易鲸捷数据库 REST 服务器的 HTTP 地址。

<entry

key="trafodionRestServerUri">http://my.esgyndb.server:4200</</pre>

entry>

• qwbMaxRows

Query Workbench 中允许获取的最大行数。

<entry key="qwbMaxRows">100000</entry>

httpPort

Esgyn DBManager 嵌入式 jetty 服务器的 HTTP 端口。

默认值为4205。

<entry key="httpPort">4205</entry>

httpsPort

Esgyn DBManager 嵌入式 jetty 服务器的 HTTPS 端口。

默认值为4206。

<entry key="httpsPort">4206</entry>

keyStoreFile

HTTPS 连接的 SSL 密钥库文件。

<entry

key="keyStoreFile">/opt/trafodion/sqcert/server.keystore</en
try>

securePassword

易鲸捷数据库安装程序生成自签名证书,并保存在 SSL 密钥库 (密码加密)。

<entry

key="securePassword">OBF:liupligf1x8altvj1x8klidrlirx</entry
>

requestHeaderSize

嵌入式 jetty 服务器的 HTTP 请求头的大小。

单位为 byte。

<entry key="requestHeaderSize">98304</entry>

# 6.2.2.2 配置文件示例 (Sample Configuration file)

```
以下示例为$TRAF_CONF/dbmgr/config.xml。
```

xml version="1.0" encoding="UTF-8"?
properties SYSTEM</td
"http://java.sun.com/dtd/properties.dtd">
<properties></properties>
The JDBC url for the Trafodion/EsgynDB instance that</td
you are
connecting to>
<entry< td=""></entry<>
<pre>key="jdbcUrl"&gt;jdbc:t4jdbc://my.esgyndb.server:23400/:</pre>
>
The JDBC driver class name for the Trafodion/EsgynDB</td
JDBC driver>
<entry< td=""></entry<>
<pre>key="jdbcDriverClass"&gt;org.trafodion.jdbc.t4.T4Driver</pre>
Minimum number of connections for jdbc connection pool</td
>
<entry key="minPoolSize">2</entry>
Max number of connections for jdbc connection pool
<entry key="maxPoolSize">8</entry>
Connection timeout in seconds for jdbc connections
<pre><entry key="connectionTimeout">60</entry></pre>
maxIdleTime in seconds for jdbc connection pool
<pre><entry key="maxIdleTime">1200</entry></pre>
Max number of statements for jdbc statement caching

```
<entry key="maxStatementsCache">100</entry>
<!-- The Admin User ID used by DBManager for its internal
queries-->
<entry key="adminUserID">DB ADMIN</entry>
<!-- The Admin User's password-->
<entry key="adminPassword">OBF:1sov1sov</entry>
<!-- Session Timeout in minutes. Your DB Manager client
browser session
to the DB Manager server will be timed out after this
interval and
you will be asked to login again -->
<entry key="sessionTimeoutMinutes">120</entry>
<!-- The TimeZone name of the EsgynDB server. Enter in
canonical time zone
format like Etc/UTC or America/New York -->
<entry key="timeZoneName">America/New York</entry>
<!-- The Trafodion REST Server URI -->
<entry
key="trafodionRestServerUri">http://my.esgyndb.server:4200
entry>
<!-- The openTSDB HTTP URI-->
<entry
key="openTSDBUri">http://my.esgyndb.server:5242</entry>
<!-- The Bosun Alerts HTTP URI-->
<entry key="alertsUri">http://my.esgyndb.server:8070</entry>
<!--HTTP read timeout in seconds for external REST calls
issued by
DB Manager Server -->
```

```
<entry key="httpReadTimeOutSeconds">120</entry>
<!--Maximum number of rows allowed to be fetched in Query
Workbench-->
<entry key="qwbMaxRows">100000</entry>
<!-- The following properties are only required and used by
the embedded
jetty server -->
<!-- The HTTP Port for the DB Manager embedded jetty server
-->
<entry key="httpPort">4205</entry>
<!-- The HTTPS Port for the DB Manager embedded jetty server
-->
<entry key="httpsPort">4206</entry>
<!-- The SSL keystore file for the EsgynDB Manager embedded
jetty server -->
<entry
key="keyStoreFile">/opt/trafodion/sqcert/server.keystore</en
try>
<!-- The SSL keystore password for the DB Manager embedded
jetty server -->
<entry
key="securePassword">OBF:liupligf1x8altvj1x8klidrlirx</entry</pre>
>
<!-- The HTTP request header size for DB Manager embedded
jetty server -->
<entry key="requestHeaderSize">98304</entry>
</properties>
```

### 6.2.3 启动 Esgyn DBManager

#### 6.2.3.1 使用 Python 安装程序安装 (When Installed Using Python Installer)

对于与数据库集成的 Esgyn DBManager 组件, 使用 sqstart 或 trafstart

脚本启动易鲸捷数据库时, Esgyn DBManager 这部分组件将自动启动。

#### 6.2.3.2 手动启动 Esgyn DBManager (Manual Start of DB Manager)

对于与数据库集成的 Esgyn DBManager 组件,使用以下脚本启动:

\$ cd \$TRAF HOME/dbmgr-2.7.0/bin

./dbmgr.sh start

如果 Esgyn DBManager 启动成功,您将看到以下提示:

2017-11-06\_17:17:58: Esgyn DBManager is up and running with pid (17122)

如果 Esgyn DBManager 启动失败,您将看到以下提示。此时,请查看dbmgr.log。

2017-11-06\_17:23:36: Failed to start QianBase Manager. Please check the dbmgr log.

#### 6.2.4 停止 Esgyn DBManager

### 6.2.4.1 使用 Python 安装程序安装

对于与数据库集成的 Esgyn DBManager 组件,使用 sqstop 或 trafstop 脚本 停止易鲸捷数据库时,这部分 Esgyn DBManager 组件将自动停止。您也可以使 用 dbmgr.sh 脚本停止 Esgyn DBManager。

### 6.2.4.2 手动停止 DB Manager

对于与数据库集成的 Esgyn DBManager 组件,使用如下脚本停止。

\$ cd \$TRAF HOME/dbmgr-2.7.0/bin

./dbmgr.sh stop

### 6.2.5 日志设置

Esgyn DBManager 使用 logback 记录日志。

如需更改日志回滚设置,或设置默认日志级别,编辑

\$TRAF\_CONF/dbmgr/logback.xml 文件。

Esgyn DBManager运行时日志存储在\$TRAF LOG/dbmgr 目录中。

### 6.2.6 查看 Esgyn DBManager 的状态

对于与数据库集成的 Esgyn DBManager 组件,使用以下脚本启动查看状态:

\$ cd \$TRAF\_HOME/dbmgr-2.7.0/bin

./dbmgr.sh status

### 6.2.7 查看 Esgyn DBManager 的版本

如需查看 Esgyn DBManager 的版本,选择以下任一方法:

• [方法一]

对于与数据库集成的 Esgyn DBManager 组件,使用以下脚本启动查看版本:

- \$ cd \$TRAF HOME/dbmgr-2.7.0/bin
- ./dbmgr.sh version
- [方法二]

Ē	<b>易鲸捷</b> 数据库管理器	
R	实例	
ø	监控	~
ŝ	工作负载	~
	数据管理	~
Ð	诊断	~
ß	运维	~
ŝ	设置	
0	帮助 🔓	^
	×Ŧ 💫	

在 Esgyn DBManager 主页上,点击帮助 > 关于。

如下图所示:

该页面显示您当前易鲸捷数据库的版本和许可证信息。



### 6.2.8 高可用性配置

与数据库集成的 Esgyn DBManager 组件具有持久性和容错性。

• 持久性

如果进程异常退出,则它会重新启动。

容错性

如果某一节点出现故障,则进程将在其它节点上启动。

#### 6.2.8.1 DB Manager 高可用性 (DB Manager HA)

#### 6.2.8.1.1 使用 Python 安装程序安装 (When Installed Using Python Installer)

- 如果通过 python 安装程序安装易鲸捷数据库,则需要为 HA 选择多个 DCS Master, DB Manager 会在所有这些 DCS Master 节点上运行,我们会利用 DCS 虚拟 IP (keepalived 或者浮点 IP) 来访问 DB Manager。如果主节点出现 故障,则 Esgyn DBManager 将在备节点上启动,此时,备节点变为主节点。
- 节点监控进程\$ NMON 会持续监控 DB Manager,如果 DB Manager 出现故障, 它将自动重新启动,更多细节内容请参阅 Persistence using NMON。

#### 6.2.8.2 使用 NMON 的持久性 (Persistence using NMON)

NMON(节点监控器)为 DB Manager(仅在使用 python 安装程序安装时) 提供容错性和持久性。

NMON 在每个节点上运行,由易鲸捷数据库 monitor 进程启动和管理,因此 NMON 具有持久性。

NMON 从**\$TRAF\_HOME**/sql/scripts/node\_monitor.cmd 文件中读取 命令列表,每分钟读取一次。

node\_monitor.cmd 文件包含检查和启动 DB Manager 的命令,该检查只运行在 python 安装模式。

\$DBMGR INSTALL DIR/bin/dbmgr.sh watch

通过监控(watch)选项,dbmgr.sh 首先检查 DB Manager 是否正在运行,如果正在运行,则脚本退出,否则启动 DB Manager。

# 7. 准备安装 QianBase

本章讲述以下内容:

- 7.1 获取 sudo 访问权限和无密码 SSH (命令行安装)
- 7.2 配置 LDAP Identity Store
- 7.3 用户 ID 和密码
- 7.4 所需软件
- 7.5 收集信息

## 7.1 获取 sudo 访问权限和无密码 SSH (命令行安装)

QianBase 安装要求用户 ID 具备以下属性:

- sudo 访问权限
- ssh 无密码访问集群上所有节点

# ┛注意

如需获得此类权限,请向集群管理者申请权限。

#### 如需获取 ssh 无密码访问权限,请按照以下步骤设置您的用户 ID:

1. 在当前安装的节点上执行以下命令。

\$ echo -e 'y\n'   ssh-keygen -t rsa -N "" -f \$HOME/.ssh/id_rsa
<pre>\$ cat \$HOME/.ssh/id_rsa.pub &gt;&gt; \$HOME/.ssh/authorized_keys</pre>
<pre>\$ echo localhost \$(cat /etc/ssh/ssh_host_rsa_key.pub) &gt;&gt;</pre>
\$HOME/.ssh/known_hosts
<pre>\$ echo "NoHostAuthenticationForLocalhost=yes" &gt;&gt;</pre>
\$HOME/.ssh/config
\$ chmod 600 \$HOME/.ssh/config

- 将公钥文件\$HOME/.ssh/id\_rsa.pub 的内容复制至每个节点的 \$HOME/.ssh/authorized keys 文件。
- 3. 将私钥文件\$HOME/.ssh/id\_rsa 复制至其它节点的\$HOME/.ssh 目录, 将该文件的所有者更改为您,将文件属性更改为 700 (chmod 700)。

# 7.2 配置 LDAP Identity Store

如需启用 QianBase 验证 (Authentication) 功能,您需使用 LDAP Identity Store 进行验证。QianBase 安装程序将提示您设置指向一个 (或多个) LDAP 服务器 的 验证 配置 文件,这将启用 QianBase 系统中的安全功能,即认证 (Authentication) 和授权 (Authorization) 功能。

如需手动设置验证配置文件并启用安全功能,请参阅13.

# 7.3 用户 ID 和密码

以下为安装 QianBase 时使用的用户 ID 和密码。

# ●注意

您将使用两个用户 ID:

- 具备 sudo 权限的用户。
- 用户 trafodion。如果用户 trafodion 不存在,安装程序将自动创建用 户 trafodion。

登录	用户ID	密码	说明
Cloudera Manager	admin	admin	• 安装 Cloudera 后,系统会提示您登
网页登录	(默认)	(默认)	录 Cloudera Manager 页面。
			• 请使用默认的用户 ID 和密码
			(admin, admin)。
			• 如果您已安装 Cloudera,请使用您
			过去指定的用户和密码。
具备sudo权限的	<sudo-< td=""><td><password></password></td><td>• 安装时,您可能需要 sudo 或 sudo</td></sudo-<>	<password></password>	• 安装时,您可能需要 sudo 或 sudo
用户	username>		userid权限。
			• 该用户具备 sudo 权限,能 ssh 无密
			码访问集群中所有节点。
QianBase 登录	trafodion	traf123	• 安装 QianBase 时,该用户 ID 由
		(默认)	EsgynDB 安装程序 <b>自动创建</b> 。
			• 请勿手动创建该用户。

# 7.4 所需软件

- 需要 JDK1.8。
- 需要在集群上安装 Linux 软件依赖包,这些依赖包通常不是核心 Linux 发行 版的一部分。安装程序会通过互联网自动获取这些软件包,但如果集群不能 访问互联网,则您需手动下载软件包。

操作系统	软	件包
CentOS Linux 6.5 ~ 6.9, 7.2 ~ 7.4	pdsh	apr
Red Hat Linux $6.5 \sim 6.9$ , $7.2 \sim 7.4$	log4cxx	apr-util
	sqlite	protobuf
	expect	lzo
	perl-DBD-SQLite	lzop
	xerces-c	unzip
	perl-Params-Validate	gcc-c++
	perl-Time-HiRes	unixODBC
	gzip	unixODBC-devel
	gnuplot	libiodbc s
	lsof	libiodbc-devel
	keepalived	openIdap-clients
	libcgroup	snappy

# 7.5 收集信息

安装时,安装程序将提示您输入某些信息。开始安装之前,请确保您已了解以下信息:

信息	默认	备注
安装程序解压文件的路径	无	您需指定该路径。
许可秘钥	无	易鲸捷公司提供。
每个节点并行客户端会话	8	指定每个节点并行会话的数量。
的数量		每个会话最多需要 1GB 内存,您

		能在安装后更改并行客户端会话
		的数量1。
在现有 QianBase 上升级	无	如果在新集群上安装,安装程序
安装或全新安装		将执行额外操作。
trafodion 用户ID和	用户 ID:	建议您不要更改用户ID。
密码	trafodion	• python 命令行安装程序会提示
		您输入密码。
	<b></b>	
	****	
集群中的节点列表	无	可以使用 sudo 或 root 用户无密码
		访问所有节点。
trafodion 用户 ID 的	/home	如果trafodion 用户 ID 的根目
根目录路径前缀		录为/opt/home/trafodion,
		则前缀为/opt/home。
JDK 的路径	无	JDK 的绝对路径。
		1 示例
		/usr/java/jdk1.8.0_112-
		cloudera
QianBase 安装程序压缩文	无	指定 QianBase 安装程序的完整路
件的路径		径。
Hadoop 发行版 URL	无	在表单中指定:
		<ip-address>:<port> 或</port></ip-address>

<sup>1</sup>更多信息,请参阅《EsgynDB DCS 安装指南》。

		<node name="">:<port></port></node>
		<b>全</b> 示例 vm-1.yourcompany.local: 7180
Hadoop 发行版详情		
• 管理员界面用户 ID、	• 依赖于发	
密码	行版	
• 集群名称	• Cluster1	
• HDFS 用户 ID	• hdfs	
• HBase 用户 ID、组	• hbase,	
• HBase 服务名称	<ul><li>hbase</li><li>hbase</li></ul>	
QianBase 安装目录	无	指定目录的绝对路径。
		允许您维护软件的多个版本。
DCS HA (高可用)	未启用	您将需要浮动 IP 地址和接口。
		如果存在多个节点,建议您指定
		DCS Master 节点列表。
		如果启用了 DCS HA,则客户端
		使用浮动 IP 地址;如果未启用,
		则客户端使用多个IP功能。

安全	未启用	如需启用安全功能,确保已配置
		LDAP, 且LDAP 配置文件名称可
		用。

# 8. 安装 QianBase

本章讲述以下内容:

8.1 命令行安装程序

8.2 管理

8.3 验证

# 8.1 命令行安装程序

QianBase 命令行安装工具为 db install.py, 它是一个独立的安装包。

# ●注意

QianBase 必须安装在已安装了 HBase Regionserver 的节点上(即已安装 Hadoop 发行版。更多信息,请参阅 5. 安装 CDH Hadoop)。

### 如需使用命令行安装程序安装 QianBase,执行以下步骤:

- 以用户<sudo-username>身份,在集群的任一节点(通常是集群的第一 个节点)上创建 esgyndb downloads 目录。
- 2. 下载以下文件,将它们放在 esgyndb downloads 目录:
  - 服务器包

数据库软件包如下:

- o esgyn\_corebank\_1.0.0\_server-2.7.0-RH7x86 64.tar.gz
- 相应的安装软件包如下:
  - esgyn corebank pyinstaller-1.0.0-RH7.tar.gz

\$ mkdir \$HOME/esgyndb\_downloads \$ mv <your-download-path>/ esgyn\_corebank\_1.0.0\_server-2.7.0-RH7-x86\_64.tar.gz \$HOME/esgyndb\_downloads \$ mv <your-download-path>/ esgyn\_corebank\_pyinstaller-1.0.0-RH7.tar.gz \$HOME/esgyndb\_downloads \$ cd \$HOME/esgyndb\_downloads

3. 解压下载的安装文件。

\$ tar -xzf esgyn\_corebank\_pyinstaller-1.0.0-RH7.tar.gz

4. 打开安装目录。

\$ cd python-installer

5. 运行安装脚本,您能使用以下任一安装模式:

模式	说明	操作
引导	推荐新手使用。	执行./db_install.py
设置	安装时,系统将提示所需信息。	
专家	推荐以下用户使用:	在安装配置文件中输入所需参数
设置	• 资深用户	后,调用安装程序
	• 使用无人值守安装(Unattended Setup)方式的用户	./db_install.py
		<pre>-config_file <installer-< pre=""></installer-<></pre>
	配置文件是一个预编辑好的文本文件,它包含安装所需信	config-file>
	息。安装程序如果使用该配置文件,在安装时不会提示用户	
	输入信息。	
	配置文件的模板在 python-	
	installer/configs/db_config_default.ini 中。	
	将该文件复制至您的文件夹,输入安装所需参数。	

▲注意

- 如果出现配置错误,安装程序将中止。请在重新运行之前更正错误。
- 如果安装后不启动 Trafodion (即,在Start Trafodion after install (Y/N)时输入N),则您需要在db\_install.py 完成后手动启 动和初始化 Trafodion。更多信息,请参阅以下提示。
- 更多有关手动启用 QianBase 安全功能的详细信息,请参阅 13. 启用安全功能。

₩ 提示

安装过程将:

(1) 提示输入许可证秘钥。

Add a new license file or license string (Y/N) [N]: Y Enter full path to license file or the license string [NONE]:

- (2) 在 CentOS/Redhat Linux 系统安装必要的 RPM 包。
- (3) 创建 trafodion 用户 ID。
- (4) 为 trafodion 用户 ID 设置无密码 ssh。
- (5) 将 QianBase 发行版文件复制至集群所有节点。
- (6) 生成启动文件。
- (7) 启动 QianBase、数据库连接服务<sup>2</sup>和 QianBaseManager 服务。
- 6. 安装完成后,您将看到以下消息:

7. QianBase 现已启动并正在运行。请以trafodion 用户身份登录 QianBase。

如果您选择安装后不启动 trafodion,则需启动并初始化 trafodion,执行以下

步骤。

```
cds
sqstart
[trafodion@esgvm-test ~]$ trafci
Welcome to EsgynDB Core Banking Command Interface
Copyright (c) 2015-2019 Esgyn Corporation
>>initialize trafodion;
```

### 8.2 管理

管理 QianBase 需要以 trafodion 用户身份登录。

<sup>&</sup>lt;sup>2</sup> P Database Connectivity Services, DCS.

以下为管理子系统的脚本。

组件	启动	停止	状态
QianBase core	sqstart	sqstop	sqcheck
RMS Server	rmsstart	rmsstop	rmscheck
REST Server	reststart	reststop	_
Manageability	mgblty_start	mgblty_stop	mgblty_check
DCS (Database	dcsstart	dcsstop	dcscheck
Connectivity Services)			

```
₽
示例
```

启动 QianBase。

```
cd $MY_SQROOT/sql/scripts
```

sqstart

sqcheck

# 8.3 验证

执行基础完备性检查。

1. 使用 trafci 创建一张表,并写入数据。

[trafodion@edb001 ~]\$ trafci

Welcome to EsgynDB Core Banking Command Interface

Copyright (c) 2015-2019 Esgyn Corporation

>>CREATE TABLE test1 (f1 int, f2 int);

--- SQL operation complete.

>>INSERT INTO test1 VALUES(1,1);

--- 1 row(s) inserted.

>>INSERT INTO test1 VALUES(2,2);
--- 1 row(s) inserted.

```
>>SELECT * FROM test1;
F1
           F2
_____
           _____
        1
                    1
        2
                    2
--- 2 row(s) selected.
>>GET TABLES;
Tables in Schema TRAFODION.SEABASE
_____
TEST1
--- SQL operation complete.
>>exit;
```

2. 使用 mgblty check 工具验证所有管理组件 (QianBase Manager、TSD、

TCollector 和 Bosun)正在运行。

```
[trafodion@edb001 ~]$ mgblty_check
Status of OpenTSD...(Expect 1 per node)
Total count of TSD process: 12
Status of Tcollector...(Expect 1 per node)
Total count of tcollector process: 12
Status of DBMgr...(Expect 1 per DB Manager node)
Total count of dbmgr process: 1
Status of Bosun...(Expect 1 per DB Manager node)
Total count of bosun process: 1
```

3. 在浏览器中输入 http://localhost:4205, 验证是否能启动并登陆 QianBase

Manager<sup>3</sup>。

 如需将客户端应用程序连接至 QianBase,您可以下载并安装 QianBase JDBC 和/或 ODBC 驱动程序(安装程序名称为 esgynDB\_clients-2.7.0-RH7-x86 64.tar.gz<sup>4</sup>)。

<sup>&</sup>lt;sup>3</sup>更多信息,请参阅《QianBaseManager用户指南》。

<sup>&</sup>lt;sup>4</sup>更多信息,请参阅《QianBase 客户端安装指南》,该指南说明如何安装 JDBC 和 ODBC 驱动程

序、如何连接到 QianBase 以及如何运行示例程序测试连接。

# 9. 升级 QianBase

本章以 QianBaseR1.1 升级到 R1.2 为例介绍了如何升级 QianBase 并保证数据的

- 完整性,内容如下,
- 9.1 环境检查
- 9.2 备份配置文件
- 9.3 在线备份数据
- 9.4 备份 metadata
- 9.5 升级 QianBase
- 9.6 版本回退
- 9.7 已知问题及解决办法
- 从企业数据安全角度,强烈建议采用以下步骤:
- 升级前:环境检查 (见章节9.1环境检查)、

备份数据和文件(见章节9.2&9.3&9.4)

- 升级 QianBase (见章节 9.5)
- 升级后:在线备份数据库(见章节9.3)

### 9.1 环境检查

升级前检查当前环境是否符合升级条件。

1) 停止 QianBase:

```
[root@node01 ~]# su - trafodion
[trafodion@node01 ~]$ cstat
[trafodion@node01 ~]$ ckillall
```

.....

2) 在 Cloudera 或 ambari 管理页面上先停止再重启 hbase 服务;

3) 启动 QianBase, 检查是否可以启动成功:

[root@node01 ~]# su - trafodion
[trafodion@node01 ~]\$ sqstart

4) Hbase check, 确保 hbase 运行正常:

```
[trafodion@node01 ~]$ hbcheck
Stderr being written to the file:
/var/log/trafodion/hbcheck.log
ZooKeeper Quorum: esggy-n066.esgyncn.local,esggy-
n068.esgyncn.local,esggy-n067.esgyncn.local, ZooKeeper Port :
2181
HBase is available!
HBase version: 1.2.0-cdh5.13.0
HMaster: esggy-qa-n016.esgyncn.local,60000,1546927184811
Number of RegionServers available:3
RegionServer #1: esggy-qa-
n017.esgyncn.local,60020,1546927175148
RegionServer #2: esggy-qa-
n016.esgyncn.local,60020,1546927182923
RegionServer #3: esggy-qa-
n018.esgyncn.local,60020,1546927174578
Number of Dead RegionServers:0
Number of regions: 142
Number of regions in transition: 0
Average load: 47.3333333333333336
```

## 9.2 备份配置文件

备份以下自定义配置文件:

1) trafodion用户的.bashrc文件: 位于trafodion用户的家目录下, ~/.bashrc

[trafodion@node01 ~]\$ mkdir -p ~/backup\_conf [trafodion@node01 ~]\$ cp ~/.bashrc ~/backup conf

2) ms.env文件: 位于/var/lib/trafodion/ms.env

[trafodion@node01 ~]\$ cp /var/lib/trafodion/ms.env ~/backup\_conf

检查该文件下的 STFS\_HDD\_LOCATION 参数,查看当前集群的 scratch file 配置。

请注意,从1.6.3 rc9 开始, ms.env 发生了较大的变化,详情见如下:

从低版本升级到 QB1.6.3 版本时需要特别注意上个版本 ms.env 文件中的内容,因为在 QB1.6.3 中改动了 ms.env 的新逻辑,增加了新文件 ms.env.custom, ms.env 每次升级都会被重置, ms.env.custom 每次都会被保留。

升级时,需要将 ms.env 中非默认的条目拷贝出来(包含手动修改的和其它脚本 修改的参数),待升级后写到新文件\$TRAF\_CONF/ms.env.custom 中,只需要在 第一次升级时拷贝,之后的升级无需再拷贝。

升级后,如果需要手动修改 ms.env 的条目,将它们全都写到新文件 \$TRAF\_CONF/ms.env.custom 中,不要再写到 ms.env 中,使得在以后的每次升 级都能被保留下来。升级后如果发现某个功能不可用,可以先检查下 ms.env 中 的配置是否正确,否则手动将其写到 ms.env.custom;

原文件 ms.env 所在的目录是\$TRAF\_VAR, 新文件 ms.env.custom 所在的目录是 \$TRAF\_CONF;

3) dcs conf: 位于\$DCS INSTALL DIR/conf/\*

[trafodion@node01	~]\$	ср	-r	\$DCS_INSTALL_DIR/conf
~/backup conf				

记录当前集群的 floating IP 和 external interface, 以便升级的时候使用。

4) trafodion conf: 位于/etc/trafodion/trafodion config

[trafodion@node01	~]\$	ср	<pre>/etc/trafodion/trafodion_config</pre>
~/backup_conf			

5) ldap	conf:	位	于
/etc/trafodio	n/conf/.traf_authe	ntication_con	fig
[trafodion@node0	1~]\$		ср
/etc/trafodion/	conf/.traf_authenti	cation_config	~/backup_conf

检查并记录当前环境的 ldap 配置,以备升级过程中使用,目的是为了保证升级前后 ldap 配置的一致性:

[trafodion@node01~]\$ cat .traf authentication config

(备份 ldap 的 LdapHostname、UniqueIdentifier、LdapPort、LDAPSearchDN、 LDAPSearchPwd)

```
[trafodion@node01~]$ trafci
SQL>showddl user db_root;
SQL>showddl user db_admin;
(记录系统中注册的 external name)
```

6) sqenvcom.sh: 位于\$TRAF HOME/sqenvcom.sh

[trafodion@node01 ~]\$ cp \$TRAF HOME/sqenvcom.sh ~/backup conf

7) 记录 hadoop server 的部分参数值 (在 Cloudera manager 或 ambari 网页上查找), 如下表所示, 加粗文本是参数值:

<hbase config=""></hbase>
"hbase-site": {
"hbase.snapshot.master.timeoutMillis": "600000",
"hbase.table.sanity.checks": "false",
"hbase.hregion.impl":
"org.apache.hadoop.hbase.regionserver.transactional.Transactio
nalRegion",
"hbase.regionserver.region.split.policy":
"org.apache.hadoop.hbase.regionserver.ConstantSizeRegionSplitP
olicy",
"hbase.regionserver.lease.period": "3600000",

```
"hbase.snapshot.enabled": "true",
    "hbase.snapshot.region.timeout": "600000",
    "hbase.hregion.memstore.flush.size": "536870912",
    "hbase.hregion.memstore.block.multiplier": "7",
    "hbase.hstore.blockingStoreFiles": "200",
    "hbase.rootdir.perms": "750"
    },
<HDFS config>
    "hdfs-site": { "dfs.namenode.acls.enabled": "true" },
    <zookeeper config>
    "zoo.cfg": { "maxClientCnxns": "0" }
```

8) 记录Cloudera网页上hbase configuration中Master Advanced Configuration Snippet、RegionServer Advanced Configuration Snippet和hbase.coprocessor.region.classes的 配置值,避免升级之后被删除。升级后如果自定义值被删除,则手动将需要 的值配置回去,然后重启hbase即可,示例如下:

✓ STATUS	Clear	Master Advanced	Master Default Group 🔹		
C Error	0	Valve) for hbase-site.xml	Name	hbase.snapshot.master.timeoutMillis	Î
A Warning	0		Value	600000	
Edited	0		Description	Description	1
Has Overrides	0			Final	
			Name	hbase.rootdir.perms	<b>n</b>
			Value	750	
			Description	Description	
				Final	
			+		
HBase Coprocesso	r Region	RegionServer Default Group	*		
Classes	-	org.apache.hadoop.hbase.coprocessor.transactional.TrxRegionObserver			
hbase.coprocessor.regio	n.classes				
		org.apache.hadoop.hbase.coprocessor.transactional.TrxRegionEndpoint			
		org.apache.hadoop.hbase.coprocessor.AggregateImplementation			

RegionServer Advanced Configuration Snippet (Safety	RegionServer Default Group			
Valve) for hbase-site.xml	Name	hbase.hregion.impl	Ê	
	Value	org.apache.hadoop.hbase.regionserver.transactional.TransactionalR		
	Description	Description		
		Final		
	Name	hbase.regionserver.region.split.policy	Ê	
	Value	org.apache.hadoop.hbase.regionserver.ConstantSizeRegionSplitPol		
	Description	Description		
		Final		
	Name	hbase.snapshot.enabled	Ê	
	Value	true		
	Description	Description		
		Final		
	Name	hbase.snapshot.region.timeout	Ê	
	Value	600000		
	Description	Description		
		Final		
	+			

# 9.3 在线备份数据

### 9.3.1 备份简介

QianBase R1.1中支持在线备份,进行全量备份会备份当前数据库中的所有对象;为防止升级QianBase失败导致数据丢失,故先在R1.1中做一个全量备份(online backup),以防止升级失败后可利用该备份恢复到R1.1。

### 9.3.2 在线备份前准备

1) 检查hbase snapshot功能是否开启,若没开启,将其打开:

登录cloudera manager (例如: http://<Cloudera Manager IP>:7180), 在hbase configuration中启用snapshot,勾选hbase.snapshot.enabled并重启hbase即 可:

启用快照	HBase(服务范围)	•
hbase.snapshot.enabled		

2) 检查metadata数据是否一致,否则提示用户清除不一致数据:

trafodion@node01 ~]\$ trafci SQL><mark>cleanup metadata,check,return details;</mark>

DDL OUTPUT

\_\_\_\_\_

Metadata Cleanup: started, check only

Start: Cleanup Orphan Objects Entries
Entry #1(OBJECT): TRAFODION.ABC.B1<sup>5</sup>
End: Cleanup Orphan Objects Entries (1 entry found)

Start: Cleanup Orphan Hbase Entries
Entry #1(OBJECT): TRAFODION.ABC.A1<sup>6</sup>
End: Cleanup Orphan Hbase Entries (1 entry found)

Start: Cleanup Inconsistent Objects Entries
Entry #1(UID): 8046956827968092354
End: Cleanup Inconsistent Objects Entries (1 entry found)

Start: Cleanup Inconsistent Views Entries
End: Cleanup Inconsistent Views Entries (0 entries found)

Start: Cleanup Inconsistent Hive Entries End: Cleanup Inconsistent Hive Entries (0 entries found)

Metadata Cleanup: done

--- SQL operation complete.

若存在不一致数据,用户可自主决定是否将其清除7,清除语句如下:

trafodion@node01 ~]\$ trafci

 $<sup>^{5}</sup>$ 本例中A1表为不一致数据,已经从QianBase 中删除,却还存在于HBase中。

<sup>&</sup>lt;sup>6</sup>本例中 B1 表为不一致数据,已经从 HBase 中删除,却还存在于 Qianbase 中。

<sup>7</sup>建议不清除,不影响升级,知道有这些不一致数据即可。

> cleanup metadata, return details; DDL OUTPUT \_\_\_\_\_ Metadata Cleanup: started Start: Cleanup Orphan Objects Entries Entry #1(OBJECT): TRAFODION.ABC.B1 End: Cleanup Orphan Objects Entries (1 entry cleaned up) Start: Cleanup Orphan Hbase Entries Entry #1(OBJECT): TRAFODION.ABC.A1 End: Cleanup Orphan Hbase Entries (1 entry cleaned up) Start: Cleanup Inconsistent Objects Entries Entry #1(OBJECT): TRAFODION.ABC.B1 Entry #2(UID): 8046956827968092354 End: Cleanup Inconsistent Objects Entries (2 entries cleaned up) Start: Cleanup Inconsistent Views Entries End: Cleanup Inconsistent Views Entries (0 entries cleaned up) Start: Cleanup Inconsistent Hive Entries End: Cleanup Inconsistent Hive Entries (0 entries cleaned up) Metadata Cleanup: done --- SQL operation complete.

3) 获取并记录当前数据库里面所有对象信息

[trafodion@node01 ~]\$ hbase shell >list

### 9.3.3 在线备份 QianBase

QianBase正常启动下,做在线全量备份(Online Full Backup),以防后续需要, 在线备份示例如下高亮部分:

[trafodion@node01 ~]\$ trafci

SQL>backup trafodion, tag 'bkfulldb';

--- SQL operation complete.

### 9.3.4 检查在线备份数据

#### 1) 监控备份进度8

在备份过程中,可以通过查看trafodion.sql.java.log监控备份进度,如下 示例中,粗体字部分写明了何时开始备份哪张表(ENTER),何时对该表备份结 束(EXIT),整个备份结束后也有结束标志(finalizeBackup),如下:

```
[trafodion@node01 logs]$ tail -f trafodion.sql.java.log
.....
2019-01-08 14:52:07,481 INFO pit.BackupRestoreClient: ENTER
doSnapshot Backup Tag: bkfulldb Thread ID 100 TableName:
TRAFODION.BR_TEST.CUSTOMER SnapshotPath:
3c001796c4c8cd1d5fabea8eb9b7c4e1 SnapshotName:
TRAFODION.BR_TEST.CUSTOMER_SNAPSHOT_
bkfulldb_1589293009763769
2019-01-08 14:52:07,892 INFO pit.BackupRestoreClient: EXIT
doSnapshot thread 100 TableName TRAFODION.BR_TEST.CUSTOMER
......
2019-01-08 14:52:30,911 INFO pit.BackupRestoreClient:
finalizeBackup backupTag: bkfulldb
```

2) 在线全量备份结束之后,执行下面语句来确保可获取到备份列表:

<sup>8</sup> 针对全量备份,与数据库数据量无关,与表个数有关,备份 1000 张表,大概需要 3 分 10 秒 <sup>9</sup> 实际查询得到的 backup tag 名会加上时间戳后缀,若要得到实际输入名称,可使用 backup trafodion, tag 'bkfulldb',override;

### 9.3.5 导出在线备份的数据

在线备份结束后,可将在线备份的数据文件导出存放在本地

1) 先将其导出到hdfs中:

```
[trafodion@node01 ~]$trafci
SQL>export backup to location
'hdfs://node01:8020/user/trafodion/trafodion_backups', tag '
bkfulldb_00212413689918115206 ';
--- SQL operation complete.
```

2) 检查导出的备份文件是否存在:

```
[trafodion@node01 ~]$ hdfs dfs -ls
/user/trafodion/trafodion backups/
drwxr-xr-x - trafodion trafodion
                                          0 2019-01-08
15:58
/user/trafodion/trafodion backups/TRAFODION.SCH0501.T1INCR1
drwxr-xr-x - trafodion trafodion
                                          0 2019-01-08
15:57
/user/trafodion/trafodion_backups/TRAF_inctest1_TRAFODION.NAS
CH1.T1INCR2
drwxr-xr-x - trafodion trafodion
                                         0 2019-01-08
16:08
/user/trafodion/trafodion backups/TRAF inctest2 TRAFODION.NAS
CH2.T1INCR3
drwxr-xr-x - trafodion trafodion
                                          0 2019-01-08
16:12
/user/trafodion/trafodion backups/bkfulldb 002124136899181152
06
```

3) 从hdfs中导出到linux上

```
[trafodion@node01 ~]$ mkdir onlinebackup
[trafodion@node01 ~]$ hdfs dfs -get
/user/trafodion/trafodion backups/* ~/onlinebackup/
```

# 9.4 备份 metadata

### 9.4.1 手动备份 snapshot

 因为在线备份不会备份如下元数据表,所以需要在hbase shell里备份所有的 QianBase元数据表以用于当元数据表损坏时的后续恢复,其中包括: TRAFODION.\_MD\_.\* & TRAFODION.\_PRIVMGR\_MD\_.\* & TRAFODION.\_REPOS\_.\* & TRAFODION.\_DTM\_.\* & TRAFODION.
 \_LIBMGR\_.\*,使用hbase中的snapshot指令来备份,语法为(snapshot'表 名','快照名')快照名可自定义,此处仅举了其中一张表的快照示例,实际过 程中要对所有的metadata表做快照,示例如下:

```
[trafodion@node01 ~]$ hbase shell
>list<sup>10</sup>
.....
TRAF_RSRVD_1:TRAFODION._MD_.TABLES
.....
> snapshot 'TRAF_RSRVD_1:TRAFODION._MD_.TABLES' ,
'SNAP_TRAF_RSRVD_1_TRAFODION._MD_.TABLES'
>list_snapshots<sup>11</sup>
SNAPSHOT TABLE + CREATION
TIME
.....
SNAP_TRAF_RSRVD_1_TRAFODION._MD_.TABLES
TRAF_RSRVD_1:TRAFODION._MD_.TABLES (Tue Jan 08 03:50:32 -0500
2019)
.....
```

复制 metadata 的备份文件。该文件保存在 hdfs 里的/hbase/.hbasesnapshot 下,可将其复制到 trafodion 的自定义目录下,以 trafodion 用户执行:

<sup>11</sup> 查看备份的 metadata 快照

<sup>&</sup>lt;sup>10</sup> 查出 R1.1 QianBase 所有的元数据表(当有多租户时,还 有'TRAF RSRVD 1:TRAFODION. TENANT MD.\*')
```
[trafodion@node01 ~]$ hdfs dfs -mkdir
/trafodion_backups/backup_metadata
[trafodion@node01 ~]$ hdfs dfs -cp /hbase/.hbase-snapshot/*
/trafodion_backups/backup_metadata/
```

## 9.5 升级 QianBase

## 9.5.1 停用 QianBase

升级前确保 QianBase 处于停用状态,若还有遗留进程可以用 ckillall 将其停止:

```
[root@node01 ~]# su - trafodion
[trafodion@node01 ~]$ cstat
[trafodion@node01 ~]$ ckillall
.....
```

## 9.5.2 解压 Installer

1) 解压已下载的R1.2 QianBase installer文件

[root@node01 r2.5]# tar -zxf esgyn\_corebank\_pyinstaller-1.2.0-RH7.tar.gz

2) 进入安装文件目录:

[root@node01 r2.5]# cd python-installer/

```
[root@node01 python-installer]# ll
total 164
-rwxr-xr-x 1 1002 1004 13201 May 18 12:46 add nodes.py
-rwxr-xr-x 1 1002 1004 8305 May 18 12:46 auto config.py
-rwxr-xr-x 1 1002 1004 6231 May 18 12:46 cdh install.py
drwxr-xr-x 2 1002 1004 4096 May 18 12:46 configs
-rwxr-xr-x 1 1002 1004 29649 May 18 12:46 db install.py
-rwxr-xr-x 1 1002 1004 5592 May 19 22:14 db uninstall.py
-rwxr-xr-x 1 1002 1004 7946 May 18 12:46 delete nodes.py
-rwxr-xr-x 1 1002 1004 9700 May 18 12:46 inspector.py
-rw-r--r-- 1 1002 1004 14403 May 19 22:14 LICENSE
-rw-r--r-- 1 1002 1004
                        291 May 19 22:14 NOTICE
-rwxr-xr-x 1 1002 1004 8420 May 18 12:46 pre install.py
-rw-r--r-- 1 1002 1004 54204 Oct 7 2017 prettytable.py
-rw-r--r-- 1 root root 52781 May 20 15:16 prettytable.pyc
                         36 May 19 22:14 PyInstallerVer
-rw----- 1 1002 1004
-rw-r--r-- 1 1002 1004 3672 May 18 12:46 README.md
drwxr-xr-x 3 1002 1004 4096 May 20 15:16 scripts
-rwxr-xr-x 1 1002 1004 9440 May 18 12:46 secure_setup.py
```

drwxr-xr-x 2 1002 1004 4096 May 18 12:46 templates

### 9.5.3 升级 QianBase<sup>12</sup>

执行安装脚本、填写相应的集群信息、开始升级QianBase、具体操作可参

EsgynDB 安装指南 2.5 (中文版)。

升级时,强烈建议选择安装时启动数据库,如下例中高亮部分所示:

```
[root@node01 python-installer]# ./db_install.py
Enter trafodion scratch file folder location(should be a large
disk),
if more than one folder, use comma separated [$TRAF_VAR]:
Start instance after installation (Y/N) [Y]:
Enable LDAP security (Y/N) [N]:
.....
```

### 9.5.4 检查 QianBase

升级成功后,对新版本的 QianBase 做一个完整性检查。

1) trafcheck / dcscheck / mgblty check / trafci

2) 登陆 DB Manager 页面(http://active-dcs-master-IP:4205), 检查各组件工作是否

正常, active-dcs-master 可由 dcscheck 获取;

```
3) 检查升级后 QianBase 版本:
```

```
[trafodion@node01 ~]$ trafci
SQL>get version of software;
Software Version: 2.7.0. Expected Version: 2.7.0.
--- SQL operation complete.
```

4) smoke test, 下面是一些简单的检查示例:

检查原表:

```
[trafodion@node01 ~]$ trafci
SQL>get schemas;
```

SQL>get tables;

```
<sup>12</sup>如果升级 QianBaseR1.2 时遇到问题,可参考 9.7.1 关于 installer,如果有相同问题,对应解决即可。
```

SQL>select count(\*) from <xx\_table>;

• 检查是否可以创建新表:

```
[trafodion@node01 ~]$ trafci
SQL>create schema upgrade test;
--- SQL operation complete.
SQL>set schema upgrade test;
--- SQL operation complete.
SQL>create table a1 (a int, b char(99));
--- SQL operation complete.
SQL>insert into table a1 values
(1, 'crt66878ygui'), (2, 'ahdfg79r483wghef'), (3, 'augkhf1934orh43'
), (4, 'qliuh2389'), (5, '8976489'), (6, 'adkjhfb');
--- 6 row(s) inserted.
SQL>select * from a1;
А
           В
_____
     1 crt66878ygui
     2 ahdfg79r483wghef
     3 augkhf1934orh43
     4 qliuh2389
     5 8976489
     6 adkjhfb
--- 6 row(s) selected.
SQL>delete from a1;
--- 6 row(s) deleted.
SQL>drop table a1;
--- SQL operation complete.
SQL>drop schema upgrade_test;
--- SQL operation complete
```

5) 权限检查:升级后 QianBase 用户对数据库对象的访问权限与升级前相同

6)再次全量备份,升级成功后,建议同章节4,再次对数据库进行一次全量备份,并导出到本地

# ▲注意

升级完数据库后,需要清除数据库所有数据节点的TLOG。

## 9.6 版本回退

如果 QianBase 升级失败, 需回退到 R1.1 版本, 参考以下步骤:

## 9.6.1 收集日志

如果升级 QianBase 失败, 需收集 python\_installer/logs/\*、\$TRAF\_LOG/\*、/var/log/下 hdfs & hbase & zookeeper 的日志, 以便分析失败原因。

## 9.6.2 回退准备

1) 停止QianBase

[root@node01 ~]# su - trafodion
[trafodion@node01 ~]\$ ckillall
[trafodion@node01 ~]\$ cstat

2) 将9.2备份配置文件中备份的trafodion用户下的.bashrc配置文件复制过来覆盖

当前的.bashrc,以便能读到R1.1正确的环境变量,每个节点都要复制:

[trafodion@node01 ~]\$ cp ~/backup\_conf/.bashrc ~/

## 9.6.3 回退安装

回退安装时,与正常安装一样

1) 解压 R1.1 的 installer:

```
[root@node01 CBanl1.1]#tar -xzf esgyn_corebank_pyinstaller-
1.1.0-RH7.tar.gz
```

2) 进入installer 目录:

[root@node01 python-installer]# cd python-installer/

3) 执行安装脚本db\_install.py,按提示信息填写相应的集群信息。

## 9.6.4 恢复 metadata 数据

当回退R1.1,如有元数据损坏,需要恢复metadata数据时,不需要停止数据库, 在hbase shell中恢复即可:

 利用9.4<u>备份metadata</u>中备份的快照进行恢复,恢复前,需在hbase中disable所 有的R1.1 QianBase元数据表,示例如下

```
[trafodion@node01 ~]$ hbase shell
>disable_all 'TRAFODION._MD_.*'
Y
>disable_all 'TRAFODION._PRIVMGR_MD_.*'
Y
>disable_all 'TRAFODION._REPOS_.*'
Y
```

2)利用快照恢复metadata表,恢复时,需恢复9.4<u>备份metadata</u>中备份的所有 metadata快照,此处仅举了其中两张表的快照恢复示例,实际过程中要恢复 所有的metadata快照,示例如下:

```
[trafodion@node01 ~]$ hbase shell
.....
>restore_snapshot 'SNAP_TRAF_RSRVD_1_TRAFODION._MD_.TABLES'
0 row(s) in 0.9230 seconds
.....
>restore_snapshot 'SNAP_TRAFODION._REPOS_.METRIC_QUERY_TABLE'
0 row(s) in 0.9060 seconds
```

3) 在hbase中Enable所有的metadata表

```
[trafodion@node01 ~]$ hbase shell >enable_all 'TRAFODION._MD_.*'
```

Y

```
>enable_all 'TRAFODION._PRIVMGR_MD_.*'
Y
>enable_all 'TRAFODION._REPOS_.*'
Y
>enable_all 'TRAFODION._DTM_.*'
Y
```

4) 通过Trafci连接QianBase,检查恢复后的表

## 9.6.5 在线恢复数据

回退安装成功后,利用9.3.3在线备份QianBase中备份的数据进行恢复,步骤如

下:

```
1) 恢复时可用 get all backup snapshots;查看备份的数据
```

[trafodion@node01 ~]\$ trafci SQL>get all backup snapshots; BackupTag BackupTime BackupStatus BackupOperation

\_\_\_\_\_\_\_\_\_\_

bkfulldb\_00212413689918115206<sup>13</sup> 2019-01-08:14:52:30 VALID REGULAR >>restore trafodion,tag 'bkfulldb 00212413689918115206'

--- SQL operation complete.

2) 如果数据库中备份丢失或不能使用时也可以从本地导入

```
[trafodion@node01 ~]$ hdfs dfs -put ~/onlinebackup/*
/user/trafodion/trafodion backups/
```

[trafodion@node01 ~]\$ hdfs dfs -chmod -R 757 /user/trafodion/trafodion\_backups >>import backup to location ' hdfs://node01:8020/user/trafodion/trafodion\_backups', tag ' bkfulldb\_00212413689918115206 ';

>>restore trafodion, tag 'bkfulldb 00212413689918115206';

3)恢复成功之后启动 QianBase,参照 <u>9.5.4 检查 QianBase</u>检查检查数据;然后 根据 <u>9.6.1 收集日志</u>分析升级失败的原因,并针对具体问题进行解决,然后 再次升级

### 9.7 已知问题及解决办法

### 9.7.1 关于 installer

1) CDH 版本过高的问题

R1.2 版本的 python installer 支持最高的 CDH 版本为 5.9, 当 CDH 版本过高时,

安装会报以下错误:

<sup>&</sup>lt;sup>13</sup> 实际查询得到的 backup tag 名会加上时间戳后缀,若要得到实际输入名称,可使用 backup trafodion, tag 'bkfulldb', override;

\*\*\*[ERROR]: Sorry, currently Trafodion doesn't support CDH5.13.0
version

操作步骤:

修改/python-installer/configs/central\_config.json 文件,在"cdh\_ver"添加所需版本 号,如: "5.13",再重新安装即可

"linux_distro": [	"Ubuntu16", "SuSE11", "centos6", "centos7",
"redhat6", "redhat	:7"],
"hadoop_distro":	["mapr", "cloudera", "hortonworks", "apache"],
"java_ver":	["1.8"],
"cdh_ver":	["5.4", "5.5", "5.6", "5.7", "5.8", "5.9" <mark>, "5.13"</mark> ],
"hdp_ver":	["2.3", "2.4", "2.5", "2.6"],
"hbase_ver":	["1.0", "1.1", "1.2"],
"hdfs_ver":	["2.6", "2.7"],
"hive_ver":	["1.0", "1.1", "1.2"],

## 9.7.2 关于备份和恢复

1) R1.1&R1.2的备份兼容

R1.1 版本备份的数据能在 R1.2 中恢复,升级失败回退 R1.1, R1.1 版本备份也能

在 R1.2 中使用。

2) 如果R1.1的Export/Import不可用, 报如下错误时

如果 R1.1 的 Export/Import 不可用, 报如下错误时:

>>export backup to location '/user/trafodion/backups', tag
'bkmysc1';

\*\*\* ERROR[5050] EXPORT BACKUP command could not be completed. Reason: Error returned from exportOrImportBackup method. See next error for details.

\*\*\* ERROR[8448] Unable to access Hbase interface. Call to ExpHbaseInterface::exportOrImportBackup returned error HBASE\_EXPORT\_IMPORT\_BACKUP\_ERROR(726). Cause: java.lang.Exception: ExportBackup FAILED with error:1 Error Detail: OpenJDK 64-Bit Server VM warning: Using incremental CMS is deprecated and will likely be removed in a future release 18/01/29 22:34:16 INFO snapshot.ExportSnapshot: Copy Snapshot Manifest 18/01/29 22:34:17 WARN mapreduce.TableMapReduceUtil: The hbaseprefix-tree module jar containing PrefixTreeCodec is not present. Continuing without it. 18/01/29 22:34:17 INFO hdfs.DFSClient: Created HDFS DELEGATION TOKEN token 17 for trafodion on 10.10.23.69:8020 18/01/29 22:34:17 INFO security.TokenCache: Got dt for hdfs://esggy-qa-n019.esgyncn.local:8020; Kind: HDFS DELEGATION TOKEN, Service: 10.10.23.69:8020, Ident: (HDFS DELEGATION TOKEN token 17 for trafodion) 22:34:17 INFO client.RMProxy: Connecting 18/01/29 to ResourceManager at esggy-qa-n019.esgyncn.local/10.10.23.69:8032 18/01/29 22:34:18 INFO snapshot.ExportSnapshot: Loading Snapshot 'lebee4f0e4a90e11dcc3c6ec6717a67a' hfile list 18/01/29 22:34:18 Configuration.deprecation: INFO hadoop.native.lib deprecated. Instead, is use io.native.lib.available 18/01/29 22:34:19 INFO mapreduce.JobSubmitter: number of splits:0 18/01/29 22:34:19 INFO mapreduce.JobSubmitter: Submitting tokens for job: job 1516860540526 0007 18/01/29 22:34:19 mapreduce.JobSubmitter: INFO Kind: HDFS DELEGATION TOKEN, Service: 10.10.23.69:8020, Ident: (HDFS DELEGATION TOKEN token 17 for trafodion) impl.YarnClientImpl: Submitted 18/01/29 22:34:19 INFO application application 1516860540526 0007 18/01/29 22:34:20 INFO mapreduce.Job: The url to track the job: http://esggy-qan019.esgyncn.local:8088/proxy/application 1516860540526 0007/ mapreduce.Job: 18/01/29 22:34:20 INFO Running job: job 1516860540526 0007 18/01/29 22:34:23 INFO mapreduce.Job: Job job 1516860540526 0007 running in uber mode : false 18/01/29 22:34:23 INFO mapreduce.Job: map 0% reduce 0% 18/01/29 22:34:23 INFO mapreduce.Job: Job job 1516860540526 0007 state FAILED due failed with to: Application application 1516860540526 0007 failed 2 times due to ΑM Container for appattempt 1516860540526 0007 000002 exited with exitCode: -1000 For more detailed output, check application tracking page:http://esqqy-qan019.esgyncn.local:8088/proxy/application 1516860540526 0007/T hen, click on links to logs of each attempt. application 1516860540526 0007 Diagnostics: Application initialization failed (exitCode=255) with output: main : command provided 0 main : run as user is trafodion main : requested yarn user is trafodion Requested user trafodion is not whitelisted and has id 501, which is below the minimum allowed 1000

Failing this attempt. Failing the application. 18/01/29 22:34:23 INFO mapreduce.Job: Counters: 0 18/01/29 22:34:23 ERROR snapshot.ExportSnapshot: Snapshot export failed org.apache.hadoop.hbase.snapshot.ExportSnapshotException: Copy Files Map-Reduce Job failed at org.apache.hadoop.hbase.snapshot.ExportSnapshot.runCopyJob(Exp ortSnapshot.java:828) at org.apache.hadoop.hbase.snapshot.ExportSnapshot.run(ExportSnap shot.java:1023) at org.apache.hadoop.util.ToolRunner.run(ToolRunner.java:70) at org.apache.hadoop.hbase.snapshot.ExportSnapshot.innerMain(Expo rtSnapshot.java:1097) at org.apache.hadoop.hbase.snapshot.ExportSnapshot.main(ExportSna pshot.java:1101) org.apache.hadoop.hbase.pit.BackupRestoreClient.exportBackup(B ackupRestoreClient.java:1050) org.apache.hadoop.hbase.pit.BackupRestoreClient.exportOrImport Backup(BackupRestoreClient.java:1298).

### 解决办法:

把 trafodion 用户加入 yarn allowed.system.users 的白名单, 或降低 min.user.id 的值 (要低于 trafodion user ID), 如下图所示:

cloudera MANAGE	R Clus	ters ▼ Hosts ▼ Diagnostics ▼ A	udits Charts - Administration -	🚆 🖧 Search Support - admin -
🔵 YARN (MR2 li	nclude	ed)(Cluster 1) Actions •		Today, 10:56 PM C ST
Status Instances Configurat	ion Com	mands Applications Resource Poo	Is Charts Library Audits Web UI - Quick Links -	
				Switch to the classic layout Role Groups
Fliters		allowed.system.users		
♥ SCOPE				Show All Descriptions
YARN (MR2 Included) (Service-	Wide) 0	Allowed System Users	NodeManager Default Groupand 2 others C	•
Gateway	0	allowed.system.users Edit Individual Values	nobody	+ -
JobHistory Server	0		impala	+ -
NodeManager	1		Impaid	
ResourceManager	0		hive	+ -
✓ CATEGORY			llama	+ -
Advanced	0		hbase	+ -
Compression	0		Annels attend	
Logs	0		traiodion	<b>+</b> -
Main	0			
			3 Edited Values	Reason for change Save Changes

### 9. 升级 QianBase

cloudera MANAGER	sters • Hosts • Diagnostics • Audits Charts • Administration •	🚆 🖧 Search Support 🕶 admin 🗣
YARN (MR2 Inclu	led) ( Cluster 1 ) Actions -	Today, 10:58 PM C ST
Status Instances Configuration C	mmands Applications Resource Pools Charls Library Audits Web UI - Quick Links -	
		Switch to the classic layout Role Groups
Filters	min.user.id	
♥ SCOPE		Show All Descriptions
YARN (MR2 Included) (Service-Wide) 0 Gateway 0 JobHistory Server 0 NordeManager 1	Minimum User ID NodeManager Default Groupand 2 others C min.user.id 400 Edit Individual Values	0
ResourceManager D	Display 25 v Per Page	
✓ CATEGORY		
Advanced 0 Compression 0		
Main 0		
	3 Edited Values Reason for	change Save Changes

# 10. 卸载 QianBase

本章讲述以下内容:

10.1 停止 QianBase

10.2 卸载 QianBase

卸载 QianBase 前,确定数据已保存。

使用 Trafodion 配置用户 ID 卸载 QianBase。

在集群的第一个节点上执行命令,请勿在非 QianBase 集群的节点上执行命令。

# ┛注意

如需更新 QianBase, 无需卸载现有 QianBase。

## 10.1 停止 QianBase

执行以下命令:

```
$ su trafodion
$ cd $TRAF_HOME/sql/scripts
$ sqstop
$ exit
```

## 10.2 卸载 QianBase

QianBase 卸载工具为 db\_uninstall.py, 该工具包含于安装软件包中。

该工具仅卸载 QianBase 的实例,并不会删除库中的对象 (表、视图、索引等)。

### 卸载步骤如下:

1. 下载安装包,将其放在 QianBase 集群中的任一节点:

QianBase\_pyinstaller-1.5.1-RH7.tar.gz

2. 解压安装包

tar zxf QianBase pyinstaller-1.5.1-RH7.tar.gz

3. 执行 db uninstall.py 卸载 QianBase

(下表中黄底加粗高亮部分为需要手动输入并执行,灰色斜体部分为注释)

Enter Trafodion node list to uninstall (separated by comma): 10.10.23.76,10.10.23.77,10.10.23.78 -- 输入 QianBase 节点 IP 或主机名,以逗号隔开,也可使用正则表达式,如 10.10.23.[76-78] Enter Trafodion user name: trafodion -- 输入 QianBase 实例在 linux 系统下的用户名, 默认为 trafodion Uninstall Trafodion on [10.10.23.76 10.10.23.77 10.10.23.78], it will kill all trafodion processes and remove all files in trafodion user, do you really want to continue (Y/N) [N]: Y -- 输入 Y/y 确认卸载 QianBase \*\*\*[INFO]: Remove Trafodion on node [10.10.23.76] ... \*\*\*[INFO]: Remove Trafodion on node [10.10.23.77] ... \*\*\*[INFO]: Remove Trafodion on node [10.10.23.78] ... \*\*\*\*\* Trafodion Uninstall Completed \*\*\*\*\*

# 11. 卸载 CDH 及 Hadoop 服务

1. 在Cloudera Manager控制台停止所有服务,包括集群名【Cluster 1】 和

Cloudera Management Service 服务



2. 移除parcel包

假如是通过 packages 来安装的,那么请跳过这个步骤。本文是针对parcel包方式进行安装,在CM界面右上角点击Parcel包的图标:

cloudera <sup>®</sup> MANAGER	Clusters 🔻	Hosts -	Diagnostics 🔻	Audits Charts	s 👻 Administr	ation 💌	🚆 🖧 Sear	rch Support 🕶 admin
Parcels							Parcel Usage Configuration	Check for New Parcels
Location	Clust	er 1					在CM界面右上角点击【Parcel】包的图标,再停用 CDH 5	
Cluster 1	Par	cel Name	Ve	ersion		Status		Actions
Available Remotely	AC	CUMULO	1.3	7.2-5.5.0.ACCUMULO	5.5.0.p0.8	Available Remotely		Download
Filters			1.	4.4-1.cdh4.5.0.p0.65		Unavailable		
V ERROR STATUS		Error for parce	I ACCUMULO-1.4.4-1	l.cdh4.5.0.p0.65-el7 : I	Parcel not available	for OS Distribution RHEL7.	\	
Error	2 CDI	4 5	5.	13.0-1.cdh5.13.0.p0.2	29	Distributed, Activat	ted 👋	Deactivate
V PARCEL NAME	KAI	KA	4.0	0.0-1.4.0.0.p0.1		Available Remotely		Download
CDH 5	1 KUI	DU	1.	4.0-1.cdh5.12.2.p0.8		Available Remotely		Download
KAFKA	1 1 SP/	RK	0.9	9.0-1.cdh4.6.0.p0.98		Unavailable		

3. 通过CM删除集群Cluster 1

O Cluster	1 (CDH 5.13.0, Parcels)	Charts		30m 1h 2h 6h 12h 1d 7d 30d ♂~
Ø I≣ Hosts	Add Service	Unable to issue query: the Host Monitor is not runn	ing	
O H HBase	Start			
O 🔤 HDFS	Stop			
O 😵 Hive	Restart			
O III YARN	Rolling Restart			
O 🟦 ZooKe	Deploy Client Configuration			
_	Deploy Kerberos Client Configuration			
	Upgrade Cluster			
Cloudera M	Refresh Cluster		Cluster Network IO	
O Cloude	Refresh Dynamic Resource Pools			
	Inspect Hosts in Cluster			
	Enable Kerberos		QUERY ERROR	
	Delete Kerberos Credentials			
	Set up HDFS Data At Rest Encryption			
	View Client Configuration URLs	Cluster Disk IO	Cluster CPU	
	Rename Cluster			
	Delete	QUERY ERROR	QUERY ERROR	eed has k
	Enter Maintenance Mode			ů.
	View Maintenance Mode Status			

4. 删除Cloudera Management Service

<	CC	http://140.246.192.2	45:7180/cmf/home			器 ∲ ∨ 如何修改	cdh 日志目录 🍳 🛛	Ļ,
$\mid$	📩 收藏 ∨ 📋	EsgynDi 🚞 Delivery 🚞 Life 🛅	Hadoop 🛅 DataBas	🛅 编程 🛅 应用开发 🛅 Linux 🛅 官方网站 🔮	🖢 华为 UI 🚺 shell实 🛛 🖏 51CT	0. 🔇 架构师的 🛛 🛞 百度云摆	y 有道翻译 C Linux系	*
-								
	Cloudera	Management Service		图表		30分钟 1小时 2小时 6小时	12小时 1天 7d 30d d	90° -
	O Cloud	lera M 🎤 4	•	• 无法发出查询:Host Monitor 未运行				
		Cloudera Management Servic	ce 项操作					
		启动		群集 CPU	群集磁盘 10			
		得止 <b>雷</b> 启		本为供归		本为供知		
		±/8		互间错误		互叫钜庆		
		实例						
		配置						
		添加角色实例		群集网络 10				
		重命名		查询错误				
		删除						
		视图维护模式状态						

5. 停掉并卸载,所有节点的cloudera-manager-agent

每个节点,使用 root 用户,执行以下操作。

- 1) 停掉 cloudera-manager-agent 服务 service cloudera-scm-agent stop
- 2) 卸载 cloudera-manager-agent 服务的相关 RPM 包

```
[root@sjtl-n001 ~]# rpm -qa|grep -i cloudera-manager-agent
cloudera-manager-agent-5.13.0-1.cm5130.p0.55.el7.x86_64
[root@sjtl-n001 ~]# rpm -e cloudera-manager-agent-5.13.0-
1.cm5130.p0.55.el7.x86_64
[root@sjtl-n001 ~]# rpm -qa|grep cloudera-manager-daemons
cloudera-manager-daemons-5.13.0-1.cm5130.p0.55.el7.x86_64
```

```
[root@sjtl-n001 ~]# rpm -e cloudera-manager-daemons-5.13.0-
1.cm5130.p0.55.el7.x86_64
```

3) 删除 agent 配置目录

```
[root@sjtl-n001 ~]# rm -rf /var/lib/cloudera-scm-agent
You have new mail in /var/spool/mail/root
[root@sjtl-n001 ~]# rm -rf /etc/cloudera-scm-agent
[root@sjtl-n001 ~]#
```

# ┛注意

查看是否有遗留进程,如果有 kill 掉

ps -ef|grep cmf|grep supervisord;ps -ef|grep cmf|grep flood

```
4) 卸载 agent 文件系统
```

```
[root@sjtl-n001 ~]# umount /run/cloudera-scm-agent/process
[root@sjtl-n001 ~]#
```

6. 通过CM删除节点

在 Cloudera Manager 控制台中, 删除所有主机

🗋 网页无法访问	CDH5.X完全卸载步骤 - wulant OB載CDH5.14.2	- qq_3098232 🕐 All Hosts - Cloudera Manage 🗙 😤 如何修改cdh 日志目录_百度进 + 5 省 - 日 🗙
$<$ $>$ $\bigcirc$ $\bigcirc$ $\bigcirc$ http://1	40.246.192.245:7180/cmf/hardware/hosts	28 🗇 🗸 如何修改cdh 日志目录 🔍 🔲 🕽 🗸 🚍
D 📩 收藏 🗸 🚞 EsgynD 📄 Deliver	y 🛅 Life 🛅 Hadoop 🛅 DataBas 🛅 编程 🚞 应	用开发 📋 Linux 🗎 官方网站 🌺 华为 U 🗾 shell家 👒 51CTO. 🔕 架构原始 : 🛞 百度云振 🙂 有道翻译 🧧 Linux系 💿
cloudera MANAGER	Clusters - Hosts - Diagnostics -	Audits Charts • Administration • 🏥 🖧 Search Support • admin •
All Hosts	$\setminus$	Configuration Add New Hosts to Cluster Re-run Upgrade Wizard Inspect All Hosts
Filters	Search	
✓ STATUS	Actions for Selected (1) -	Columns: 10 Selected -
O Unknown Health	1 Assign Rack	♦ Roles ♦ Commission State ♦ Last Heartbeat ♦ Load Average ♦ Disk Usage ♦ Physical Memory ♦ Swap !
> CLUSTERS	Regenerate Keytab 5.0	7 Commissioned 9.93s ago 0.09 0.07 0.36 62.4 GiB / 527.3 GiB 3.8 GiB / 31.3 GiB 1.8 GiE
> CORES	Hasta Dagamerization	
> COMMISSION STATE	Hosts Decommission	>
> LAST HEARTBEAT	Hosts Recommission	点击 【Hosts】,依次执行【Stop Roles Hosts】、【Remove From Cluster】
> LOAD (1 MINUTE)	Apply Host Template	
> LOAD (5 MINUTES)	Start Roles on Hosts	
> LOAD (15 MINUTES)	Stop Roles on Hosts	
> MAINTENANCE MODE		
> RACK	Enter Maintenance Mode	ž
	Exit Maintenance Mode	Feedback
	Remove From Cluster	
	Remove From Cloudera Manager	

7. 卸载clouder-manager-server

在部署 clouder-manager-server 服务的节点上,使用 root 用户,执行以下操作。

1) 停掉 cloudera-manager-server 服务

<pre>[root@sjtl-n001 ~]# service cloudera-scm-server stop</pre>	停止
clouder-manager-server 服务	
Stopping cloudera-scm-server (via systemctl):	
[ OK ]	
You have new mail in /var/spool/mail/root	

2) 卸载 cloudera-manager-server 服务的相关 RPM 包

[root@sjtl-n001 ~]# rpm -qa|grep cloudera-manager-server cloudera-manager-server-5.13.0-1.cm5130.p0.55.el7.x86\_64 [root@sjtl-n001 ~]# rpm -e cloudera-manager-server-5.13.0-1.cm5130.p0.55.el7.x86\_64 warning: /etc/cloudera-scm-server/db.properties saved as /etc/cloudera-scm-server/db.properties.rpmsave [root@sjtl-n001 ~]#

- 删除 server 及管理服务配置目录: rm -rf /var/lib/cloudera-scm-server
   rm -rf /etc/cloudera-scm-server
- 4) 删除 Cloudera Management Service 配置目录:
   rm -rf /var/lib/cloudera-host-monitor
   rm -rf /var/lib/cloudera-scm-eventserver
   rm -rf /var/lib/cloudera-scm-headlamp
  - rm -rf /var/lib/cloudera-service-monitor
- 8. 所有节点删除CDH软件目录
  - rm -rf /opt/cloudera/parcels/\*
- 9. 所有节点删除ZK数据目录

rm -rf /var/lib/zookeeper

10. 所有节点删除HDFS数据目录

```
rm -rf /dfs/nn
rm -rf /dfs/snn
```

- rm -rf /dfs/dn
- 11. 删除CM 自带的数据库, 通常使用MySQL或PostgreSQL

以下举例, MySQL卸载步骤

```
Centos7 上卸载 Mariadb 数据库
查询所安装的 MariaDB 组件:
[root@localhost logs]# rpm -qa | grep Maria*
MariaDB-server-5.5.49-1.el7.centos.x86_64
MariaDB-common-5.5.49-1.el7.centos.x86_64
MariaDB-client-5.5.49-1.el7.centos.x86_64
卸载数据库: [root@localhost logs]# yum -y remove mari*
删除数据库文件:[root@localhost logs]# rm -rf
/var/lib/mysql/*
```

# 12. 故障排除

- 如果启动环境失败或运行 trafci 出现问题,使用 trafcheck 命令检查 所有 QianBase 进程是否正常工作。
- 2. 如果进程未正常运行,执行以下操作:
  - 使用 sqstop 命令关闭 QianBase。如果无法关闭部分 QianBase 进程,
     使用 ckillall 命令。
  - (2) 使用 sqstart 命令重启 QianBase。
- 3. 如果问题仍然存在,检查日志 \$TRAF\_LOG: EsgynDB logs。

## 13. 启用安全功能

本章讲述以下内容:

13.1 配置 LDAP

13.2 安装与配置 OpenLDAP 服务器

## 13.1 配置 LDAP

QianBase 自身不管理用户名和密码,它通过使用 OpenLDAP 和 AD/LDAP 协议 (即 AD/LDAP 服务器)的目录服务器支持验证功能。如需配置 AD/LDAP 服务 器,请在安装过程中回答安装程序的问题。安装 AD/LDAP 将启用数据库权限功 能。

启用验证和权限功能后,QianBase 允许在数据库中注册用户,并向用户和角色 授予对象的权限。QianBase 还支持组件级(或系统级)权限,例如,您能向用 户和角色授予 MANAGE USERS 权限<sup>14</sup>。

## ● 注意

如果 QianBase 未启用 AD/LDAP,则客户端需输入用户名和密码才能连接至 QianBase,但 trafodion 忽略该用户名和密码,会话将以数据库根用户 (DB\_ROOT)身份运行(无限制)。如需限制用户,即限制某些用户访问数 据库或限制访问对象或操作,您需要启用安全功能,即启用验证和权限功能。

LDAP 启动安全性需要通过执行 python-installer/secure\_setup.py 命令实现,在此 过程中用户需要根据命令行提示输入相关配置启用,具体如下:

<sup>&</sup>lt;sup>14</sup> 更多关于向用户和角色授予权限的信息,请参阅《QianBase SQL 参考手册》的 GRANT Statement 章节。

Time Cost: 0 hour(s) 0 minute(s) 7 second(s)
Enter one option you need to set (kerberos, ldap) [kerberos]:
ldap

--选择1dap 配置

Enable LDAP security (Y/N) [N]: Y

--启用ldap

Enter LDAP user name to be assigned DB root privileges (DB\_ROOT) [trafodion]:

--提示用户输入 LDAP 中已存在的用户名,用于指定给 EsgynDB 的 DB\_\_ROOT 用户

Enter LDAP user name to be assigned DB Admin privileges [admin]: qaadmin

--提示用户输入 LDAP 中已存在的用户名,用于指定给 EsgynDB 的 DB\_ADMIN 用

Enter LDAP user password to be assigned DB Admin privileges: --提示用户输入 DB ADMIN 对应 LDAP 用户名的密码

Confirm Enter LDAP user password to be assigned DB Admin privileges:

-- 再次输入密码确认

Enter list of LDAP Hostnames (comma separated if more than one host): 10.10.23.48

--提示用户输入 LDAP 主机名/IP

Enter LDAP Port number (Example: 389 for no encryption or TLS, 636 for SSL) [389]:

--提示用户输入 LDAP 端口号

Enter all LDAP unique identifiers (semi-colon separated if more

than one identifier): uid=,ou=users,dc=esgyncn,dc=local

--提示用户输入 LDAP unique identifiers

Enter LDAP Encryption Level (0: Encryption not used, 1: SSL, 2: TLS) [0]:

--提示用户选择 LDAP 加密级别

Does the LDAP server require search user name/password (Y/N) (optional) [N]:

--提示用户是否需要搜索 LDAP 用户名和密码,建议选 N, 不搜索 Does the LDAP server support group searches (Y/N) (optional) [N]:

--提示用户是否启用 LDAP 组查询, 当为多租户环境时才需要

## 13.2 安装与配置 OpenLDAP 服务器

QianBase 安装程序设置并传播 AD/LDAP 配置文件

.traf\_authentication config, 该文件位于

\$TRAF HOME/sql/scripts,它是由一系列属性/值对组成的文本文件。

示例文件位于\$TRAF HOME/sql/scripts/traf authentication config。

### 13.2.1 安装 OpenLDAP

1.安装 openLDAP 组件

有两种方式安装 LDAP,一个是源码编译安装,另一种就是直接使用 yum 仓库安装

yum install openldap openldap-servers openldap-clients
compat-openldap

#### 2.检查安装是否成功:

rpm -qa|grep openldap

## 13.2.2 配置 OpenLDAP 服务器

如下步骤,在2个节点均要执行; 1. 拷贝配置文件 cp /usr/share/openldap-servers/DB CONFIG.example /var/lib/ldap/DB CONFIG 新版本的 OpenLDAP 中已经没有/usr/share/openIdap-servers/slapd.conf.obsolete 这 个文件了,所以现在配置都在/etc/openIdap/slapd.d 的文件中修改。但是其中的 文件都是自动生成的,所以修改需要使用 ldapmodify 指令。 2. 修改权限 chown -R ldap:ldap /var/lib/ldap/ chown -R ldap:ldap /etc/openldap/ 3. 启动服务 systemctl enable slapd && systemctl start slapd 4. 查看 ldap 启动是否成功, 默认端口号为 389 netstat -antup|grep slapd 5. 生成 rootpw 加密密码串: slappasswd -s <passwd>,此命令会生成{SSHA}开头的密码串 6. 编写如下 ldif 格式文件,并将管理员密码导入 ldap 配置文件 ldapadd -Y EXTERNAL -H ldapi:/// -f chrootpw.ldif #chrootpw.ldif dn: olcDatabase={0}config,cn=config changetype: modify

add: olcRootPW

olcRootPW: {SSHA}BXTlwAUMwVSgqHr0WlIEud3iu9ddjvQE

7. 导入基本 schema 文件

之前是在/etc/openIdap/slapd.conf 文件中,使用 include 来载入需要使用的

schema, 现在可以使用 ldapadd 命令来加载

cd /etc/openldap/schema

依次执行以下的命令,或者写入脚本批量执行,但是需要按照下面的顺序,因

为 schem 之间还有相互继承的依赖关系

```
ldapadd -Y EXTERNAL -H ldapi:/// -D "cn=config" -f
cosine.ldif
ldapadd -Y EXTERNAL -H ldapi:/// -D "cn=config" -f nis.ldif
ldapadd -Y EXTERNAL -H ldapi:/// -D "cn=config" -f
collective.ldif
ldapadd -Y EXTERNAL -H ldapi:/// -D "cn=config" -f corba.ldif
ldapadd -Y EXTERNAL -H ldapi:/// -D "cn=config" -f core.ldif
ldapadd -Y EXTERNAL -H ldapi:/// -D "cn=config" -f
duaconf.ldif
ldapadd -Y EXTERNAL -H ldapi:/// -D "cn=config" -f
dyngroup.ldif
ldapadd -Y EXTERNAL -H ldapi:/// -D "cn=config" -f
inetorgperson.ldif
ldapadd -Y EXTERNAL -H ldapi:/// -D "cn=config" -f java.ldif
ldapadd -Y EXTERNAL -H ldapi:/// -D "cn=config" -f misc.ldif
ldapadd -Y EXTERNAL -H ldapi:/// -D "cn=config" -f
openldap.ldif
ldapadd -Y EXTERNAL -H ldapi:/// -D "cn=config" -f pmi.ldif
ldapadd -Y EXTERNAL -H ldapi:/// -D "cn=config" -f
ppolicy.ldif
```

8. 配置 LDAP 根域

将如下配置文件, 通过如下命令导入 LDAP ldapmodify -Y EXTERNAL -H ldapi:/// -f chdomain.ldif chdomain 文件内容:

#chdomain.ldif
#用实际域名代替"dc=esgyn,dc=local"语句块
dn: olcDatabase={1}monitor,cn=config
changetype: modify
replace: olcAccess
olcAccess: {0}to \* by
dn.base="gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=a
uth"
read by dn.base="cn=admin,dc=esgyn,dc=local" read by \* none

13. 启用安全功能

```
dn: olcDatabase={2}hdb, cn=config
changetype: modify
replace: olcSuffix
olcSuffix: dc=esgyn,dc=local
dn: olcDatabase={2}hdb, cn=config
changetype: modify
replace: olcRootDN
olcRootDN: cn=admin,dc=esgyn,dc=local
dn: olcDatabase={2}hdb, cn=config
changetype: modify
add: olcRootPW
olcRootPW: {SSHA}BXTlwAUMwVSgqHr0WlIEud3iu9ddjvQE
dn: olcDatabase={2}hdb, cn=config
changetype: modify
add: olcAccess
olcAccess: {0}to attrs=userPassword,shadowLastChange by
 dn="cn=admin,dc=esgyn,dc=local" write by anonymous auth by
self write by * none
olcAccess: {1}to dn.base="" by * read
olcAccess: {2}to * by dn="cn=admin,dc=esgyn,dc=local" write
by * read
```

```
9.使用如下命令,在根域基础上创建组织,并在其下创建一个 admin 的组织角
色(该组织角色内的用户具有管理整个 LDAP 的权限)和 People 和 Group 两个
组织单元
ldapadd -x -D cn=admin,dc=esgyn,dc=local -W -f
```

basedomain.ldif

```
#basedomain.ldif
dn: dc=esgyn,dc=local
objectClass: top
objectClass: dcObject
```

```
objectClass: organization
o: esgyn
dc: esgyn
dn: cn=admin,dc=esgyn,dc=local
objectClass: organizationalRole
cn: admin
dn: ou=People,dc=esgyn,dc=local
objectClass: organizationalUnit
ou: People
dn: ou=Group,dc=esgyn,dc=local
objectClass: organizationalRole
cn: Group
```

10.添加用户

在 cn=admin, dc=esgyn, dc=local, ou=Group 组下添加用户;

```
ldapadd -x -D cn=admin,dc=esgyn,dc=local -W -f user.ldif user.ldif 文件的内容如下 (仅供参考):
```

```
#user.ldif
# db_root
dn: uid=db_root,ou=Group,dc=esgyn,dc=local
ou: Users
uid: db_root
sn: db_root
cn: DB_ROOT
givenName: db_root
displayName: DB_ROOT
mail: db_root@esgyn.local
objectClass: person
objectClass: inetOrgPerson
userpassword: DB_ROOT_PASSWORD #明文密码
```

# db\_admin

13. 启用安全功能

```
dn: uid=db_admin,ou=Group,dc=esgyn,dc=local
ou: Users
uid: db_admin
sn: db_admin
cn: DB_ADMIN
givenName: db_admin
displayName: DB_ADMIN
mail: db_admin@esgyn.local
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
userpassword: DB_ADMIN_PASSWORD #明文密码
```

如果没有加载 schema, 会遇到下面的错误:

```
adding new entry
"uid=trafodion,ou=Users,dc=esgyn,dc=local" ldap_add:
Invalid syntax (21) additional info: objectClass: value
#2 invalid per syntax
```

11.防火墙放行 LDAP 操作

如果开启了 firewalld, 默认是拦截 ldap 操作的。需要设置放行端口。

查询现在使用 zone:

```
[root@testa ~]# firewall-cmd --get-active-zones
public
interfaces: eth0
```

将修改 public, 让其放行 LDAP 的操作.

389 是明文传输端口、636 是 ssl 密文传输端口

```
firewall-cmd --permanent --zone=public --add-port=389/tcp
firewall-cmd --permanent --zone=public --add-port=636/tcp
```

如需删除设定可以

```
firewall-cmd --permanent --zone=public --remove-port=636/tcp
firewall-cmd --permanent --zone=public --remove-port=389/tcp
```

### 使设置生效:

firewall-cmd --reload

### 13.2.3 配置 OpenLDAP HA

如下步骤,在两个节点均要执行;

#### 13.2.3.1 前置条件

要把2台LDAP server 配置成双主模式,互相复制,需要满足以下几个条件:

1. OpenLDAP 的两台服务之间需要保持时间同步 (ntpd)

2. 软件包版本保持一致

3. 节点之间域名可以相互解析

4. schema 文件保持一致

5. 需要提供完全一样的配置及目录树信息(配置信息中只有 server ID 和

provider 的信息不同)

### 13.2.3.2 添加同步模块

#### 1. 添加 syncprov module

ldapadd -Y EXTERNAL -H ldapi:/// -f mod syncprov.ldif

```
#mod_syncprov.ldif
dn: cn=module,cn=config
objectClass: olcModuleList
cn: module
olcModulePath: /usr/lib64/openldap
olcModuleLoad: syncprov.la
```

ldapadd -Y EXTERNAL -H ldapi:/// -f syncprov.ldif

```
# syncprov.ldif
dn: olcOverlay=syncprov,olcDatabase={2}hdb,cn=config
objectClass: olcOverlayConfig
objectClass: olcSyncProvConfig
olcOverlay: syncprov
olcSpCheckpoint: 100 10
```

olcSpSessionLog: 100

olcSpCheckpoint: 每更新多少 ops 或每间隔分钟多久同步一次数据。

Ops:并非指硬盘操作数目,而是ldap操作数。

olcSpSessionLog: 100 开启 session log 将记录 olcSpCheckpoint 期间内所有对数据

的操作,最大记录操作数100,当达到同步间隔时,如果 olcSpCheckpoint 里没

有记录,将直接跳过这次同步。

在 replicate type 为 refreshOnly 时使用 session log 可最小化更新的数据量。

※olcSpCheckpoint 数值不宜设置过大,过大将导致同步迟迟不进行。

### 13.2.3.3 配置 HA

在主节点1和主节点2均执行以下步骤,但ldif文件内olcServerID和 provider 参数需要修改,其他参数保持不变;

ldapmodify -Y EXTERNAL -H ldapi:/// -f master.ldif

```
#master.ldif
# create new
dn: cn=config
changetype: modify
replace: olcServerID
# specify uniq ID number on each server
olcServerID: 0 URI
                    #该参数为唯一值,主节点1为0,主节点2为1
#必须保留一个与本地 hostname 一致的 olcServerID
dn: olcDatabase={2}hdb, cn=config
changetype: modify
add: olcSyncRepl
olcSyncRepl: rid=001
 provider=ldap://10.10.14.47:389/
                                 #主节点1上配置主节点2服务
器地址, 主节点2上配置主节点1服务器地址;
 bindmethod=simple
 binddn="cn=admin,dc=esgyn,dc=local"
 credentials=ldap123
                                 #明文密码
 searchbase="dc=esgyn,dc=local"
 scope=sub
 schemachecking=on
 type=refreshAndPersist
 retry="30 5 300 3"
 interval=00:00:05:00
```

```
add: olcMirrorMode
olcMirrorMode: TRUE
dn: olcOverlay=syncprov,olcDatabase={2}hdb,cn=config
changetype: add
objectClass: olcOverlayConfig
objectClass: olcSyncProvConfig
olcOverlay: syncprov
```

※必须保留一个与本机的 hostname 一致的 olcServerID,或者使用

ldap://localhost 作为 olcServerID。如果没有 olcServerID 与 slapd 启动

参数 -h

一致将无法启动并显示错误日志

read\_config: no serverID / URL match found

例如:

olcServerID: 1 ldap:///localhost

※olcSyncRepl不能出现本机信息,这会造成死循环

※修改或添加 olcSyncRepl 时将会检查 searchbase 对于当前库是否可见,如果不可

见将报错并提示:

Base DN is not within the database naming context

※如果没有添加 olcOverlay 就修改 olcSyncRepl 将报错并提示:

additional info: <olcMirrorMode> database is not a shadow

※olcMirrorMode必须在 olcSyncRepl 被添加完后添加。

※如果 olcSyncRepl 里的 filter 格式有误, 在添加 olcSyncRepl 时 openIdap 是不会

有任何报错的,但添加 olc Mirror Mode 时,报错

additional info: <olcMirrorMode> database is not a shadow

#### olcSyncRepl 格式说明:

provider=#服务器 IP:端口/服务器 URI

searchbase=#同步的 base

type=refreshOnly|refreshAndPersist

#同步方式,如果数据更新频度低,可使用 refreshOnly

interval=dd:hh:mm:ss#同步间隔

retry=#retry 次数

filter=#过滤 filter,如果不写将是(objectclass=\*)

scope=sub|one|base|subord #subord 是 sub 但包括自己

attrs=#使用逗号分隔的属性名,必须是有效的 LDAP 属性,默认值是\*,+

exattrs=#排除属性,格式同 attrs,默认为空

attrsonly#特殊的 flag,仅同步缺失的属性而不是比较属性值是否相同

sizelimit=#整个条目最大属性项数,默认是 unlimited

timelimit=#检索最大持续时间(秒),超时视为失败

schemachecking=on|off

#跳过 schemacheck,用于双方有不同的 schema 但同步时检索条件能确保圈定的数

据 schema 一致,使用这个 flag 将忽视 schema 检查,慎用。

network-timeout=<seconds>

timeout=<seconds>

#两个 timeout 不同于 timelimit,为连接 LDAP 服务器和 LDAP 函数操作时的失败,

不触发 retry 逻辑

bindmethod=simple|sasl #一般用 simple 即可

binddn=<dn> credentials=<passwd>

#这两个参数是对端 LDAP 服务器同步账户的凭据

starttls=yes|critical #当使用 389 端口承接 TLS 操作时,需要手动启动 starttls 以表示 TLS 开始,

但默认如果 TLS 失败将降级为明文操作,如果设置了 critical 将在失败时直接报

错。不建议使用 critical

#tls 开头的参数只有在 tls 操作时使用

tls cert=<file>tls key=<file>tls cacert=<file>tls cacertdir=<path>

#如果设置了 tls\_cacert 则不需要 tls\_cacertdir,需同步的服务器多时建议使用 tls cacertdir

#这几个路径均是"本地服务器路径非目标服务器路径"

tls reqcert=never|allow|try|demand

#建议使用 allow 或 never 防止不合规的自签名证书认证失败。

syncdata=default|accesslog|changelog

#默认是 default 将执行全属性同步,如果使用了 accesslog 插件时,将可以开启增量

同步, accesslog 需配置合适的记录方式。

### 13.2.3.4 配置 ldap 客户端绑定两台 ldap 主服务器

配置 ldap 客户端绑定两台 ldap 主服务器 authconfig --ldapserver=10.10.14.47,10.10.14.48 --update

### 13.2.3.5 测试

1.在任意一个节点添加 test 用户,并将其分配到 develop 组下;

ldapadd -x -D "cn=admin,dc=esgyn,dc=local" -W -f

test.ldif

dn: uid=test,ou=People,dc=esgyn,dc=local
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: test
cn: test
sn: test
userPassword: {SSHA}Bb53GHy2YUcLVYhF0DxgJx35x+qBT4Xd
uidNumber: 1100
gidNumber: 1100
homeDirectory: /home/test
dn: cn=develop,ou=Group,dc=esgyn,dc=local
objectClass: posixGroup
cn: develop
gidNumber: 1100
memberUid: develop

2. 分别在两个节点查询该用户,两个节点都能查询到;

ldapsearch -x -b "dc=esgyn,dc=local" -H ldap://127.0.0.1|grep uid=test

到此 openLDAP HA 就配置完成了;

### 13.2.4 使用 KeepAlived 提供服务器故障切换

OpenLDAP 服务器相互建立的同步关系后,可确保数据的一致性,用户无论对哪台服务器进行操作最终结果是一样的。配置虚拟IP 以供用户访问,由 KeepAlived 提

供定时监控,发生故障时切换服务器。

KeepAlived 监控的 OpenIdap 主机分为 Master 主机与 Backup 主机两种。同时只 有一台主机提供对外操作,其它主机(Backup)后备等待。因设置了同步关系,对 Master 主机数据的修改操作会同步至 Backup 主机。一旦 Master 主机发生故障, 其余 Backup 主机将取代其成为 Master 主机继续提供服务,如果 Backup 主机不 只一台,会进行选举取出一台成为 Master,其余继续后备。Master 主机恢复工作后, 根据 OpenIdap 的同步关系从其它 Backup 主机处同步数据,然后恢复 Master 的 身份。

为公平竞争,可不单独设置 Master 主机,而全部为 Backup 主机。各个主机设置 不同的权值,启动后相互将根据权值竞选 Master。由各主机运行中设定健康检 查脚本定期检查监控的 OpenIdap 服务工作是否正常,遇到不正则降低自身权值。 当 Master 的权值低于 Backup 时,让出 Master 权利,反之则获取 Master 权利。

以下文为例,将多台LDAP服务器组成HA环境,Qianbase整合LDAP认证时, 只需要使用 192.168.138.200 这个 IP 即可,LDAP 主机之前故障切换将由 Keepalived 软件自动提供。

### Keepalived 配置要点:

- 1. 所有主机均使用约定好的虚拟 IP 对外提供服务。
- 2. 同时只能有一台主机能占用虚拟 IP 进行服务。
- 3. 主机间通过权值高低竞争成为 Master 机,这台主机将拥有虚拟 IP 的控制权。
- 4. 定期进行健康检查,不满足的主机降低权值从而失去 Master 机的权利。
- 5. 当发生 Keepalived 主机切换时需及时通告网关 mac 地址变换。

#### 13.2.4.1 Keepalived 配置

Keepalived.conf

```
global defs {
   router id ldapha
   script user root #为了安全期间,可单独设置用户用于执行各种检查脚
本
}
vrrp script check slapd status {
   script "/etc/keepalived/check slapd status.sh"
   timeout 2 #以防脚本卡住, 2 秒无回应视为错误
   interval 3 #执行健康检查脚本间隔
   weight -50 #当脚本返回非0时,降低vrrp instance 的权重。当权重低
于其它 BACKUP 时,让出 Master 权利。
}
vrrp instance VI 1 {
   state BACKUP #不设置 MASTER,由 BACKUP 根据权值竞选 MASTER
   interface eth0 #eth0 是用于 keepalived 主机间通信的网口,可与
vip 不同。
   virtual router id 247
   priority 100 #默认充当 MASTER 的主机将初始的权值设置高,将立即成为
MASTER
   advert int 1
   authentication {
       auth type PASS
       auth pass 568423
   }
   virtual ipaddress {
       192.168.138.200/24 dev eth0 label eth0:vip
       #对外使用的虚拟 IP
   }
 #当一台主机成为 MASTER 后,需立即告知网关 VIP 对应的 MAC 地址变更,使得访
问端可以继续访问
```

```
notify_master "/usr/bin/logger -it keepalived -p
local0.info 'DS<192.168.138.100> becoming MASTER' &&
/home/refresh_arp_gateway.sh 192.168.138.200 eth0"
notify_backup "/usr/bin/logger -it keepalived -p
local0.info 'DS<192.168.138.100> becoming BACKUP'"
track_script {
    #使用脚本监控此机的健康
    check_slapd_status
  }
}
```

这个配置每台主机只需修改 priority 以区分即可,充当 master 的主机比其它主机高出 50 即可。

Log也可根据不同主机来区分书写。

Keepalived 提供4处可定义脚本的位置:

notify master: 当本实例状态变更为 MASTER 时执行

notify\_backup: 当本实例状态变更为 BACKUP 时执行

notify stop: 当keepalived 服务器停止时执行

track script: 定义执行脚本

#### 13.2.4.2 健康检查脚本

下文为健康检查脚本的例子,参考使用,实际操作时需根据现场环境组织业务。 注意:如果逻辑通过,自定义的脚本需要返回0,否则返回1。当返回1时,将 根据配置项 weight 修改当前实例的权值。

```
check slapd status.sh
```

```
#!/bin/bash
slapd_Pid=`pidof slapd`
ret=1
if [ "${slapd_Pid}" == "" ]
then
    ret=1
```

```
else
    kill -0 ${slapd Pid}
    if [ $? != 0 ]
    then
        ret=1
    else
        nc -z -v localhost 389 &>/dev/null
        if [ $? != 0 ]
        then
            ret=1
        else
            ldapsearch -b dc=TestDB,dc=local -D
cn=admin,dc=TestDB,dc=local -w 'abc123$' -H ldapi:/// -s one
-l 3 -A '(cn=admin)' 'cn' &>> /dev/null
            if [ $? != 0 ]
            then
                ret=1
            else
                ret=0
            fi
        fi
    fi
fi
if [ $ret == 1 ]
then
    systemctl restart slapd
fi
exit $ret
```

逻辑检查如下:

- 1. 判定 slapd 进程是否存在(PID 存在否判断)
- 2. 向该 PID 发出空信号,反馈进程是否僵死
- 3. 尝试连接本地 389 端口,查看是否有回应 ※nc 命令需要单独安装,
- 4. 简单的 search 操作,判断 OpenLDAP 服务器是否在工作。

※如果使用 openIdap 自带的命令(例如 Idapsearch)对已死锁的 openIdap 服务器进
行操作,命令将一直僵死。由 Keepalived 执行检查脚本时提供了超时参数,即使命令死锁也能退出,这种情况下视为命令执行失败。

当以上4点有不满足时,尝试重启 openIdap 服务,并脚本返回1。

#### 13.2.4.3 刷新网关 ARP 脚本

VIP 是由多台 Keepalived 共同使有的,每次只有一台主机可以占用 ip。当 Master 所有权发生变更时, VIP 的 mac 地址也随之改变。为告知访问端此变化,需尽 快向网关汇报 mac 的变化,则在实例变更 Master 时,执行自定义脚本。

refresh\_arp\_gateway.sh

```
#! /bin/sh
vip=$1
eth=$2
gw=`/sbin/route -n |grep -E 'UG_?' | grep ${eth} | awk
'{print $2}'`
/sbin/arping -I ${eth} -c 5 -s ${vip} ${gw} &>/dev/null
```

调用此脚本需要提供2个参数:VIP的IP、通告所用网口。脚本将根据通告网口找到对应的网关,然后发出ARP报文更新VIP对应的mac。

#### 13.2.4.4 开启 keepalived log

修改/etc/sysconfig/keepalived,原文如下

KEEPALIVED OPTIONS="-D"

在-D 后加入-d-S0

KEEPALIVED OPTIONS="-D -d -S0"

-SO代表着向 rsyslog 写入 log 所对应的日志设备,不要与其它软件配置的重复, 否则会写入其它软件的 log 文件内。

-d 将在 keepalived 启动时将配置文件信息 dump 至日志,可以不配置。

默认 log 将输出至/var/log/messages 处,为独立收集 Log,修改/etc/rsyslog.conf 加

入:

local0.\* /var/log/keepalived.log

重启 rsyslog:

systemctl restart rsyslog

以后,keepalived 的 log 将会输出到/var/log/keepalived.log

13.2.4.5 书写自定义 log

使用 loger 命令可以将自定义的 log 写至 rsyslog, 配合-p 将自定义的 log 写到指定的位置。这里将自定义 log 输出到 keepalived 的 log 里方便查询。

/usr/bin/logger -it keepalived -p local0.info
'DS<192.168.138.100> becoming MASTER'

实际打出的 log 如下:

```
Jul 17 11:55:52 localhost keepalived[6321]:
```

DS<192.168.138.100> becoming MASTER

-i 打出进程 pid

-t 标头

-p 输出位置,格式为[local][序号].[等级],可在/etc/rsyslog.conf 里将指定位置的 log 重定向至单独文件。如果没有指定,则输出至/var/log/message 里。

#### 13.2.5 如何开启 LDAP 日志功能

修改日志配置文件,在/etc/rsyslog.conf 文件件中添加下列语句:

local4.\* /var/log/ldap.log

然后重启服务 service rsyslog restart

ldapmodify -Y EXTERNAL -H ldapi:/// -f

/etc/openldap/loglevel.ldif -W

/etc/openIdap/loglevel.ldif 文件内容如下:

```
dn: cn=config
changetype: modify
add: olcLoglevel
olcLogLevel: 256
```

# ●注意

谨慎设置 loglevel, 过多的 log 会影响速度

日志等级的说明见下	图	:
-----------	---	---

等级	关键字	描述	
-1	Any	Enable all debugging	
0		No debugging	
1	(0x1 trace)	Trace function calls	
2	(0x2 packets)	Debug packet handling	
4	(0x4 args)	heavy trace debugging	
8	(0x8 conns)	connection management	
16	(0x10 BER)	print out packets sent and received	
32	(0x20 filter)	search filter processing	
64	(0x40 config)	configuration processing	
128	(0x40 ACL)	Access control list processing	
256	(0x100 stats)	stats log connections/operations/results	
512	(0x200 stats2)	stats log entries sent	
1024	(0x400 shell)	print communication with shell backends	
2048	(0x800 parse)	print entry parsing debugging	
16384	(0x4000 sync)	Syncrepl consumer processing	
32768	(0x8000 none)	Only messages that get logged whatever log level	
		is set	

# 13.2.5.1 临时打开 debug log

如果遭遇未知问题导致 OpenLDAP 无法启动但又没开启 debug log 时,可以手动 启动 slapd 进程以获取 debug log。

```
/usr/sbin/slapd -u ldap -h "ldap:/// ldapi:///" -d -1 
&>1.log&
```

openLDAP 将向 1.log 文件输出 debug log。

手动停止:

pkill -15 slapd

## 13.2.6 开启 OpenLDAP 的密码策略

#### 13.2.6.1 授予普通用户修改自身密码的权利

openLDAP 默认是不允许用户修改自身属性的,需要通过 ACL 控制。

```
ldapmodify -H ldapi:/// -Y EXTERNAL <<eof
dn: olcDatabase={2}hdb,cn=config
changetype: modify
add: olcAccess
olcAccess: to *
by self manage
by * auth
-
add: olcAccess
olcAccess: to dn.subtree="dc=esgyn,dc=local"
by dn.children="ou=People,dc=esgyn,dc=local" read
```

# eof

# 要点:

- 1. 需要给予普通用户根节点的访问权限,否则前端客户端会出错
- 2. 除了自身节点外其它节点该用户无法访问的
- 3. 给予了用户自身属性的修改权

```
完成后用户可以使用 ldappasswd 命令(推荐使用)修改自己的密码。
```

```
ldappasswd -H ldapi:/// -D
cn=user2,ou=People,dc=esgyn,dc=local -W
cn=user2,ou=People,dc=esgyn,dc=local -S
如果需要使用前端工具进行密码变更(以ldapadmin 为例),请将 base 设定为自己,
因只给予普通用户访问自己的权利。
```

13. 启用安全功能

Connection properties	×				
Connection name: test					
General Options Attributes					
Connection:					
Host: 192.168.137.155 Port: 389 Version: 3 ~					
Base: cn=user2,ou=People,dc=esgyn,dc=local Fetch DNs					
Simple authentication     SSL     TLS					
◯ GSS-API SASL					
Account					
Username: cn=user2,ou=People,dc=esgyn,dc=local					
Password: ••••					
Anonymous connection					
Test connection OK Cancel					

#### 13.2.6.2 使用 ppolicy 模块对用户密码进行控制

OpenLDAP 的密码策略分为 subtree 型(全局默认密码)和条目型。

※使用 rootDN 对条目修改密码将不受密码策略限制。

※OpenLDAP 记录用户密码时是可以指定不同的加密方式的,但是加密后的密码是无法进行密码强度测试(例如长度、复杂度)。所以为了集中控制用户密码的质量,请在设置密码时使用明文密码。例如:

{CLEARTEXT}abc

abc

是明文密码

{SSHA}2XFLWVu5hqEAoeuEvty2UkJUEz/TWjk4

是密文密码

由 ldappasswd 变更的密码默认是明文,由 ldapadmin 或 phpLDAPAdmin 变更密码时,需要手动选中明文。

#### Ldapadmin

Set Password		×
New password:		<u>0</u> K
Confirm password:		<u>C</u> ancel
Encryption method:	Plain text $\sim$	

#### phpLDAPAdmin

密码			alias
	•	clear	~
	Check password		
	(赋值)		

## Ldapmodify

ldapmodify -H ldapi:/// -D cn=user3,ou=People,dc=esgyn,dc=local -W <<eof dn: cn=user3,ou=People,dc=esgyn,dc=local changetype: modify replace: userPassword userPassword: abc123\$

eof

ldapmodify 更新时什么都不要写就是 cleartext。

# 13.2.6.2.1 建立节点用于存放密码策略模板

```
ldapadd -H ldap:/// -D cn=admin,dc=esgyn,dc=local -W <<eof
dn: ou=Pwpolicy,dc=esgyn,dc=local
objectClass: organizationalUnit
ou: Pwpolicy
eof
```

将密码策略模板集中存放便于管理。

#### 加载模块:

ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/openldap/ppolicy.ldif
-W
dn: cn=module{0},cn=config
changetype: modify

add: olcModuleLoad

olcModuleLoad: ppolicy.la

#### 配置 overlay:

```
ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/openldap/
overlay_ppolicy.ldif -W
```

```
dn: olcOverlay={0}ppolicy,olcDatabase={2}hdb,cn=config
objectclass: olcPPolicyConfig
olcOverlay: {0}ppolicy
olcPPolicyDefault: cn=default,ou=Pwpolicy,dc=esgyn,dc=local
olcPPolicyUseLockout: FALSE
```

olcPPolicyDefault 设置的节点为 subtree 型密码策略,将为整颗 DIT 树下所有 person 类型条目的默认密码策略。

如果条目自身设置了密码策略则不使用默认密码策略。

如果即没有指定该属性,条目自身也没有设置密码策略时不进行密码检查。

※ppolicy 模块不会对这个 dn 进行检查,如果 DN 不存在将导致"没有默认密码 策略",在 log 也不会有任何体现,请添加时注意。

olcPPolicyUseLockout 默认 bind 一个已被锁定的账户时会返回 InvalidCredentials 错误,无法分清是由于账户锁定还是密码错误导致登陆失败。

开启后返回的错误将是AccountLocked,明确表示是由于账户锁定导致登陆失败。 默认为关闭,因账户锁定与密码错误相区分易造成安全隐患(猜测密码)。如需区 分请开启。

#### 13.2.6.2.2 加入全局默认密码密码策略:

实现了 pwdPolicy 类的条目可作为密码规范的模板以供条目使用, overlay

ppolicy 的属性 olcPPolicyDefault 指定的条目将影响整个 DIT 树下所有 person 型 条目(例如 person、inetOrgPerson 等), 作为它们的默认密码策略。这个密码策略称为 subtree 型密码策略。

ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/openldap/ subtree ppolicy.ldif -W

dn: cn=defaultpwpolicy,dc=TestDB,dc=local
objectClass: pwdPolicy
objectClass: person
sn: defaultpwpolicy
cn: defaultpwpolicy
pwdAttribute: userPassword
pwdMinLength: 5

※pwdPolicy 为辅助类,需要配合构造类才能工作,这里使用 person 类,密码策略 需要的仅为比如控制密码过期这类属性,存储属性所用条目本身是什么不用关 心,用 person 类还是 organizationalUnit 类都无关,条目的 objectclass 有 pwdPolicy则可以使用。

※pwdAttribute 只能填写为 userPassword

#### 13.2.6.2.3 为条目添加密码策略

条目可指定自己的密码策略或者使用设置好的模板,通过设置 pwdPolicySubentry 实现。

在自身设置策略:

dn: cn=user1,ou=People,dc=esgyn,dc=local

sn: user1

cn: user1

userPassword: abc123\$

objectClass: person

#### objectClass: pwdPolicy

pwdAttribute: userPassword

pwdMinLength: 5

#### pwdPolicySubentry: cn=user2,ou=People,dc=esgyn,dc=local

objectClass 里添加 pwdPolicy,使自己成为密码策略模板,然后配置

pwdPolicySubentry 指向自己。

pwdAttribute 必须添加。

pwdMinLength 为想要添加的检查项目。

使用已有的密码策略模板 dn: cn=user1,ou=People,dc=esgyn,dc=local sn: user1 cn: user1 userPassword: abc123\$ objectClass: person **pwdPolicySubentry: cn=test\_ppolicy,ou=Pwpolicy,dc=esgyn,dc=local** ※获取密码策略模板的逻辑是由 slapd 内部实现的,则本条 目(cn=user1,ou=People,dc=esgyn,dc=local)无需有对模板

# 13.2.6.2.4 已存在的条目变更密码策略方案

(cn=test ppolicy,ou=Pwpolicy,dc=esgyn,dc=local)有访问权。

```
ldapmodify -H ldapi:/// -D cn=admin,dc=esgyn,dc=local -W
<<eof
dn: cn=user2,ou=People,dc=esgyn,dc=local
changetype: modify
add: objectClass
objectClass: pwdPolicy
-
add: pwdAttribute
pwdAttribute: userPassword
-
add: pwdMinLength
pwdMinLength: 3
```

add: pwdPolicySubentry
pwdPolicySubentry: cn=user2,ou=People,dc=esgyn,dc=local

eof

# 或者

```
ldapmodify -H ldapi:/// -D cn=admin,dc=esgyn,dc=local -W <<
eof
dn: cn=user2,ou=People,dc=esgyn,dc=local
changetype: modify
add: pwdPolicySubentry
pwdPolicySubentry:
cn=test_pwdPolicy,ou=Pwpolicy,dc=esgyn,dc=local
```

eof

※pwdPolicySubentry 可能在某些 LDAP 客户端上无法正常添加,建议使用 ldapmodify 命令以避免。

※出现 pwdPolicySubentry 后,即使指向的 DN 不存在或是并非 pwdPolicy 子类, openLDAP 不会有任何报错信息,但该条目将不再受到默认的 subtree 型密码策 略的限制,导致其工作不正常。当发现条目工作不正常时,请确认或删除该属性。 典型的情况是添加了一个不存在的 DN,则该用户会继续表现出触发密码策略的 现象(例如,出现 pwdFailureTime 等应用属性),但最终无法被锁定 (pwdAccountLockedTime 属性不出现,或人为添加 pwdAccountLockedTime 后条 目依然可以使用)。

#### 13.2.6.2.5 可设置的密码策略种类有:

pwdAllowUserChange:允许用户修改其密码,默认是 TRUE。当设置为 FALSE 时,只能由管理员变更密码。 pwdExpireWarning: 密码过期前发出警告,单位是秒,默认值是 0 表示不警 告。

计算方式为当前时间-上一次密码变更实际 < pwdExpireWarning 时发出警告 pwdFailureCountInterval: 多久时间后重置密码失败次数,单位是秒。默认是 0 表示当认证成功后立刻重置密码失败次数。如果大于 0,则即使没有 bind 成功 也会在规定时间(自上次 bind 动作后)后重置密码失败次数。

pwdGraceAuthNLimit: **密码过期**后额外允许登陆的次数,单位是次。默认值为 0表示过期即锁定,当大于0时使用完允许次数后密码依然被锁定。

pwdInHistory:开启密码历史记录,用于保证不能和之前设置的密码相同。存 至 history 记录的密码是密文,如果此参数被设置为 0 时,仅能确保密码不能与旧 密码一样。

pwdLockout: 定义用户错误密码输入次数超过 pwdMaxFailure 定义后, 是否锁 定条目, TRUE 锁定(默认).

pwdLockoutDuration: 密码连续输入错误次数后,帐号锁定时间,单位是秒。 需要配合 pwdLockout 使用,默认为 0 表示直接锁定没有间隔。

pwdMaxAge: 密码有效期,到期后用户自动锁定, 2592000 是 30 天。设置为 0 表示密码永不过期。

pwdMaxAge 并非是通过添加额外属性标记过期的,所以当一个用户由于密码 到期而被锁定后,可以通过修改 pwdMaxAge 从而解锁账户(或者直接由管理 员改密码)。

pwdMaxFailure: 密码最大失效次数,超过后帐号被锁定,单位是次。需要配合 pwdLockout 使用,默认为 0 表示可无限次猜测密码。

pwdMinAge: 密码最小有效期, 默认为 0,没有最小有效期间。如果定义了, 用 户在离上次更改密码+定义的时间之内不能更改密码。

pwdMinLength:用户修改密码时最短的密码长度,需要配合 pwdCheckQuality 使用。

※标准 LDAP 草案中规定了密码复杂度检查可以使用

#### pwdMinLength,pwdMaxLength

对输入的密码长度进行判断,但 OpenLDAP 未实现 pwdMaxLength。

pwdMustChange:用户在帐户锁定后由管理员重置帐户后是否必须更改密码, 并且只有在 pwdLockout 为 TRUE 时才启用。如果值为 FLASE(默认值),管理 员帮用户解锁条目后,用户不必更改密码,如果为 TRUE,就必须更改密码。如 果使用 pwdReset 来解锁条目,其值将覆盖此属性。

pwdSafeModify: 该属性控制用户在密码修改操作期间是否必须发送当前密码。如果属性值为FALSE(缺省值),则用户不必发送其当前密码。如果属性值为TRUE,那么修改密码值时用户必须发送当前密码。

#### 13.2.6.2.6 特殊属性

ppolicy 会在条目上追加运用属性用于记录信息,这些属性不可由用户修改或为只读属性。

pwdAccountLockedTime 账户被锁定的时间,该属性出现后用户将被锁定。

pwdChangedTime 记录上一次密码变更的时间。

pwdFailureTime 记录上一次输错密码的时间。

pwdGraceUseTime 配合 pwdGraceAuthNLimit 使用(必须大于 0),记录上一次使用额外登陆次数时的时间。由 pwdMaxRecordedFailure 控制数量。

pwdHistory 记录之前已使用的密码。

pwdReset 用于"解锁"账户。当设置为 true 时,允许用户再登陆后立即修改自己密码,设置为 false 时,用户解锁可继续使用。

当管理员对条目的密码进行了变更时 pwdAccountLockedTime 会自动消失,用户可正常登陆。

※如果 pwdLockout 设定为 FALSE 或设定了错误的 pwdPolicySubentry 时,即使出现了 pwdAccountLockedTime 用户也不会被锁定。

※设置了pwdGraceAuthNLimit,密码达到生命周期后再次bind时不会有任何警告, 但属性 pwdGraceUseTime 出现并记录,此记录会在密码被重置时删除。当达到 pwdGraceAuthNLimit 上限后,用户被锁定(pwdAccountLockedTime 出现)。

在 log 里体现为

# 206033 5ep 17 17:30:53 testa slapd[5369]: ppolicy\_bind: Entry cn=user2,ou=People,dc=TestDB,dc=local has an expired password: 1 grace logins

※设置 pwdExpireWarning 后,客户端不会有任何提示,只能在 openLDAP 的日志 里体现:

ppolicy bind: Setting warning for password expiry for cn=user2,ou=People,dc=TestDB,dc=local = 3140 seconds

#### 13.2.6.3 同步密码策略

在多主环境下,同步双方均配置好 ppolicy 模块后(此过程需手动进行),当一台 主机修改了密码策略或触发密码策略(比如输错密码)后这份变化会被同步至其 它主机之上,则条目在其中一台主机被锁定,其它主机上也被锁定。但对 ppolicy 模块(overlay)的修改是不能同步的,需要手动进行。

配置 syncprov 时只需要将 searchbase 指定到可以覆盖密码策略节点的层级、attrs 包含了 ppolicy 所需的属性(应用属性和运用属性)即可。下例为一个同步策略 olcSyncrepl: {0}rid=001 provider=ldap://192.168.137.156 bindmethod=simple binddn="cn=admin,dc=esgyn,dc=local" credentials="abc123\$" searchbase="dc=esgyn,dc=local" type=refreshAndPersist retry="5 5 300 5"

timeout=1

schemachecking=off

加粗部分为关键点,其余属性依实际环境不同而不同。没有 attrs 时,默认属性为"

\*,+"表示所有的属性都作为同步的属性。

实际环境:



密码策略的模板在节点 ou=Pwpolicy,dc=TestDB,dc=local 下,当一方添加了新策略(节点),将会自动同步给其它主机。对 ou=People 节点下条目的密码策略的触发(例如输错密码)也会同步至其它主机。

例如:条目 cn=user3 由于多次密码错误至锁定,出现属性 pwdAccountLockedTime 和 pwdFailureTime,同样也会被同步至其它主机,确保该条目在各个主机上均 被锁定。

≪   Z ⊡ ∧ /> y   ₩ V		
dc=TestDB,dc=local [192.168.137.155]	Attribute	Value
> - 🙆 ou=Group	sn	user3
v - 🔟 ou=People	cn	user3
>	objectClass	person
> m=test1	telephoneNumber	123
> ··· cn=trafodion	userPassword	123
> in cn=user 1	sn	user3
> an=user2	a	user3
> 🗀 on=user3	objectClass	person
> 🛅 ou=Pwpolicy	structuralObjectClass	person
	entryUUID	ebe43538-3283-1039-9929-837695c37a31
	creatorsName	cn=admin,dc=TestDB,dc=local
	createTimestamp	20190704084612Z
	telephoneNumber	123
	userPassword	123
	pwdFailureTime	20190918093324.951816Z
	pwdFailureTime	20190918093325.520574Z
	pwdFailureTime	20190918093326.158944Z
	pwdFailureTime	20190918093327.118147Z
	pwdFailureTime	20190918095711.966756Z
	pwdPolicySubentry	cn=test_pwdPolicy,ou=Pwpolicy,dc=TestDB,dc=local
	pwdAccountLockedT	20190918095711Z
	entryCSN	20190918095711.9667672#000000#002#000000
	modifiersName	cn=admin,dc=TestDB,dc=local
	modifyTimestamp	20190918095711Z
	entryDN	cn=user3,ou=People,dc=TestDB,dc=local
	subschemaSubentry	cn=Subschema
	hasSubordinates	FALSE

※因 syncprov 同步是由间隔的(由用户配置),过长的间隔可能导致数据同步不一致从而使得用户锁定状态无法及时被更新,为避免此问题,请降低同步间隔。

#### 13.2.6.4 额外的密码检查

openIdap 自带的 ppolicy 只能进行简单的密码复杂度检查,如果需要更复杂的检查方法则需要用户提供检查接口。

配置 pwdPolicy 类的 pwdCheckQuality 属性启用密码复杂度检查:

未指定、0服务器端不对密码进行品质检查

当指定为 1、2 时,先使用 ppolicy 自带的检 查规则(例如密码长度)后调用用户提供的接口:

1进行密码强度检查,如果用户指定的检查函数不存在,默许为通过。

2进行密码强度检查,如果用户指定的检查函数不存在,直接视为失败。

通过配置 pwdCheckModule 提供用户自定义的检查接口,值为含有用户检查接口

动态库的全路径。

pqchecker 模块,提供大小写、符号检查 下载地址: https://github.com/rammnco/pqchecker 1. 需要预先编译好 openIdap 源码 ※不需要特定版本,建议按照生产环境上版本在

ftp://ftp.openldap.org/pub/OpenLDAP/openldap-release/

处下载对应版本即可

※不能用 openIdap-devel 代替

编译 openLDAP

※因只需要编译出 openLDAP 的源代码, Berkeley DB 库不需要特别对应版本。

yum install libdb-devel libtool-ltdl-devel

./configure CPPFLAGS="-D\_GNU\_SOURCE"

make depend && make

2. 需要 JDK

※不需要特定版本,建议按照生产环境使用 yum 安装对应的 devel 库

yum install java-1.8.0-openjdk-devel

3. 编译 pqchecker

将LDAPSRC 制定为已编译好的 openLDAP 代码目录

将 JAVAHOME 指向本地的 jdk 目录, 一般在/usr/lib/jvm/目录下 ./configure LDAPSRC=/home/openIdap-2.4.44 JAVAHOME=/usr/lib/jvm/java-1.8.0openjdk-1.8.0.181-7.b13.el7.x86\_64 PARAMDIR=/etc/openIdap/pqchecker libdir=/usr/lib64/openIdap

make && make install chown ldap:ldap -R /etc/openldap/pqchecker

libdir 建议/usr/lib64/openldap 这样可不用指定全路径 root@testa openldap]# ll 总用量80

lrwxrwxrwx. 1 ldap ldap 189月 16 16:37 libpqchecker.so -> pqchecker.so.1.2.2 lrwxrwxrwx. 1 ldap ldap 189月 16 16:37 pqchecker.so -> pqchecker.so.1.2.2 -rwxr-xr-x. 1 ldap ldap 788729月 16 16:37 pqchecker.so.1.2.2

4. 手动更换

可以实现编译好该模块后逐一为生产环境替换,而不需要每台主机单独编译//拷贝 pqchecker.so.1.2.2 至目标主机

cp -a /home/pqchecker.so.1.2.2 /usr/lib64/openldap

In -s /usr/lib64/openIdap/pqchecker.so.1.2.2 /usr/lib64/openIdap/pqchecker.so

5. 设置规则

PARAMDIR 为检查规则文件存放的路径,模块将读取其下 pqparams.dat 文件,文件 内格式

UULLDDSS@)..

UU 两位数表示至少出现的大写字符数

LL 两位数表示至少出现的小写字符数

DD 两位数表示至少出现的数字字符数

SS 两位数表示至少出现的特殊字符数

从第9位开始(例如@)..)表示禁止使用的字符

前8位不足的用0占位,数字为0表示禁止使用,最大数目99

6. 配置 pqchecker

1) 加载模块

ldapmodify -D cn=admin,dc=esgyn,dc=local -W -H ldapi:/// <<eof dn: cn=default,ou=Pwpolicy,dc=esgyn,dc=local changetype: modify replace: pwdCheckQuality pwdCheckQuality: 2 add: objectclass objectclass: pwdPolicyChecker add: pwdcheckmodule pwdcheckmodule: pqchecker.so

eof

这个配置可配置在密码策略模板或者条目自己的密码策略上 ※一个密码策略只能对应一个用户检查接口(pwdcheckmodule 属性是单数值) ※此模块同 ppolicy, 无法对已加密的密码进行检查。

由于错误信息是需要客户端主动获取的,根据客户端的不一样,提示信息可能 不一样。但共同点为返回码是0x13。

当密码不符合规范时提示

phpLDAPadmin 提示



#### phpLDAPadmin 提示

٠		不能完成ldap_m	odify 操作.
	$\odot$	LDAP said:	Constraint violation
<b>4</b>		Error number:	0x13 (LDAP_CONSTRAINT_VIOLATION)
t 退出		描述:	Some constraint would be violated by performing the action. This can happen when you try to add a second value to a single-valued attribute, for example.

ldappasswd 提示



#### Log 中体现为

18]: Checking password quality for cn=user2,ou=People,dc=TestDB,dc=local.

# 13.2.6.5 锁定用户

ldapmodify -H ldapi:/// -D cn=admin,dc=esgyn,dc=local -W <<eof dn: cn=user2,ou=People,dc=esgyn,dc=local changetype: modify add: pwdAccountLockedTime pwdAccountLockedTime: 20190703072325Z

eof

时间随意,满足时间戳格式即可

#### 13.2.6.6 解锁被锁定的用户

需要管理员执行

1-删除 pwdAccountLockedTime

2-为 pwdReset 选择合适的值, pwdReset 被设定为 false 时需要确定条目不是由于

密码过期而被锁定的,否则下次登陆依然会被锁定。pwdReset 用户下次登陆必须修改密码。

ldapmodify -H ldapi:/// -D cn=admin,dc=esgyn,dc=local -W << eof

dn: cn=user2,ou=People,dc=esgyn,dc=local

changetype: modify

delete: pwdAccountLockedTime

-

add: pwdReset

pwdReset: TRUE

eof

# 13.2.7 开启 OpenLDAP 审查日志

审查日志将记录用户对 OpenLDAP 进行的修改操作以供审查。日常使用时可利用审查日志进行排故和数据恢复。

## 加载模块:

```
ldapadd -Y EXTERNAL -H ldapi:/// -f
/etc/openldap/auditlog.ldif -W
```

```
dn: cn=module{0}, cn=config
changetype: modify
add: olcModuleLoad
```

olcModuleLoad: auditlog.la

# 配置 Overlay:

```
ldapadd -Y EXTERNAL -H ldapi:/// -f
```

```
/etc/openldap/overlay_auditlog.ldif -W
```

```
dn: olcOverlay=auditlog,olcDatabase={2}hdb,cn=config
```

objectClass: olcAuditLogConfig

olcOverlay: auditlog

```
olcAuditlogFile: /tmp/auditlog/auditlog.ldif
```

olcAuditlogFile 将作为审查日志输出的文件,输出格式为标准 LDIF。

※注意输出文件及其目录的权限, OpenIdap 使用 Idap:Idap 用户组及用户。

例子:

```
# modify 1562643882 dc=TestDB,dc=local
cn=admin,dc=TestDB,dc=local IP=192.168.137.1:45436 conn=1004
dn: cn=user3,ou=People,dc=TestDB,dc=local
changetype: modify
replace: userPassword
userPassword::
e1NTSEF9NG5zeFc0QnpvNFlaZkdQNmExbzd2T1JXSWNQRjZNck4=
-
replace: pwdChangedTime
pwdChangedTime: 20190709034442Z
-
replace: entryCSN
```

```
entryCSN: 20190709034442.144302Z#000000#001#000000
-
replace: modifiersName
modifiersName: cn=admin,dc=TestDB,dc=local
-
replace: modifyTimestamp
modifyTimestamp: 20190709034442Z
-
# end modify 1562643882
```

cn=admin,dc=TestDB,dc=local 对 dn: cn=user3,ou=People,dc=TestDB,dc=local

更新了密码。

OpenIdap 关联的也更新了其它可选属性。

# 13.2.8 开启 OpenLDAP access log

此模块记录用户对 OpenIdap 服务器的访问、修改等操作。相比审查日志,其记录的信息是记录在数据库内能进行查询,且内容更详细,在同步时可用于增量更新。

※谨慎使用针对 session 和 all 的记录,将会严重劣化 openLDAP 性能。

## 加载模块

```
ldapadd -Y EXTERNAL -H ldapi:/// -f
/etc/openldap/accesslog.ldif -W
```

```
dn: cn=module{0},cn=config
changetype: modify
add: olcModuleLoad
olcModuleLoad: accesslog.la
```

# 建立记录日志的数据库

先建立数据库存放目录:

mkdir -p /var/lib/ldap/accesslog
chown ldap:ldap -R /var/lib/ldap/accesslog

导入数据库结构:

ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/openldap/accesslogDB.ldif -W dn: olcDatabase=hdb,cn=config objectClass: olcBdbConfig olcDatabase: hdb olcDbDirectory: /var/lib/ldap/accesslog olcSuffix: cn=accesslog olcAccess: to \* by dn.base="cn=admin,dc=TestDB,dc=local" read olcDbIndex: entryCSN,objectClass,reqEnd,reqResult,reqStart eq

※olcSuffix 必须以 cn=开头

※为了让指定用户可以获取其中数据,需要为其设置 ACI,这里以 cn=admin,dc=TestDB,dc=local 为例开启只读权限

※需要建立必要的索引

配置 Overlay:

ldapadd -Y EXTERNAL -H ldapi:/// -f
/etc/openldap/overlay accesslog.ldif -W

```
dn: olcOverlay=accesslog,olcDatabase={2}hdb,cn=config
objectClass: olcAccessLogConfig
olcOverlay: accesslog
olcAccessLogDB: cn=accesslog
olcAccessLogOps: abandon bind unbind
olcAccessLogSuccess: TRUE
olcAccessLogPurge: 07+00:00 01+00:00
```

olcAccessLogDB: 用于记录日志的数据库

※会检查这个节点是否存在或是否可见,如果不存在将打 log

additional info: <olcAccessLogDB> no matching backend found for suffix

```
※olcAccessLogDB 指向的数据库不能使用被记录的数据库,这会造成 slapd 死 锁。
```

olcAccessLogSuccess: TRUE 只记录成功操作 FLASE 所有操作均记录 olcAccessLogOps: 记录的操作类型

```
writes - add, delete, modify, modrdn
reads - compare, search
session - abandon, bind, unbind
all - all operations
```

可指定大分类操作,例如 reads;也可指定特定操作 bind, unbind。多个操作间使用

空格分隔。

olcAccessLogOps: abandon bind unbind

olcAccessLogPurge: 表示记录日志多久会失效删除和多久执行一次检查, 格式

为

```
olcAccessLogPurge: age interval
```

[ddd+]hh:mm[:ss] [ddd+]hh:mm[:ss]

指定操作发生后,这次操作信息会记录在 cn=accesslog 里,使用标准的 ldap 查询

即可获取。下文为一次操作记录的条目

```
dn: reqStart=20190709083457.000000Z,cn=accesslog
objectClass: auditBind
reqStart: 20190709083457.000000Z
reqEnd: 20190709083457.000001Z
reqType: bind
reqSession: 1006
reqAuthzID:
reqDN: cn=admin,dc=TestDB,dc=local
reqResult: 0
reqVersion: 3
reqMethod: SIMPLE
```

可使用标准 LDAP 检索获取信息.

```
base: cn=accesslog
scope: one
filter:
(&(reqType=bind)(reqDN=cn=admin*)(reqStart>=20180709083457Z))
例如:
ldapsearch -D cn=admin,dc=TestDB,dc=local -w xxx -H ldapi:///
```

```
-b cn=accesslog -s one
"(&(reqType=bind)(reqDN=cn=admin*)(reqStart>=20180709083457Z))
"
```

#### 13.2.8.1 将 access log 用于记录

下例为一次 modify 操作的记录:

```
dn: regStart=20190718095715.000000Z,dc=TestDB,dc=local
objectClass: auditModify
reqStart: 20190718095715.00000Z
reqEnd: 20190718095715.000001Z
reqType: modify
reqSession: 1000
reqAuthzID: cn=admin,dc=TestDB,dc=local
reqDN: cn=user1,ou=People,dc=TestDB,dc=local
reqResult: 0
reqMod: description:= 12345
reqMod: entryCSN:= 20190718095715.135772Z#000000#000#000000
reqMod: modifiersName:= cn=admin,dc=TestDB,dc=local
reqMod: modifyTimestamp:= 20190718095715Z
reqOld: description: 1234
regold: entryCSN: 20190718094027.008537Z#000000#000#000000
reqOld: modifiersName: cn=admin,dc=TestDB,dc=local
reqOld: modifyTimestamp: 20190718094027Z
reqEntryUUID: 77609966-31b3-1039-886b-5ddd54d7a2ca
```

reqType:此次操作的类型

reqDN: 被操作的 DN

reqAuthzID: 操作的 DN

reqMod reqOld:操作前发生变化的值

由于 overlay 操作是在数据已写入数据库后才执行的, reqStart 不一定为值修改的 时间。顾在进行 writes 类型操作时, 精确的时间由 modifyTimestamp 属性提供。 而其它非修改的操作因没有可用标识的时间戳, 只设定 reqStart 约等于其操作执 行的时间。

13.2.8.2 将 access log 用于同步

使用 refreshOnly 方式进行同步时,每次进行同步前需要重新初始化,计算出距离上次同步发生时数据库内发生的变化。即使期间数据库未发生任何变化也会重新计算。虽然没有发生变化,生成端与消费端还会进行数据交换以定位变化, 当数据量过大时会浪费可观的资源。

syncprov 默认是进行全属性替换,即只要有任一属性发生变化,将把这个条目所有属性直接推送。当一个条目数据量大时,细微的变化也会导致整个条目被传输。

使用 access log 作为同步依据后, refreshOnly 模式依照 access log 里的变动进行推送,免去计算条目的时间,推送的属性也仅是发生变化的属性。这有助于降低网络带宽和负载。

refreshOnly和 refreshAndPersist都能使用 access log。

由于使用 access log 会将同步行为变得复杂化,如果在用于数据量小、单个条目小(20K 作为分界)的情况下,可以不用这个模式。

#### 重新配置 syncprov

#### 1. 建立数据库存放目录

mkdir -p /var/lib/ldap/deltalog
chown ldap:ldap -R /var/lib/ldap/deltalog

2. 导入数据库结构

ldapadd -Y EXTERNAL -H ldapi:/// -f
/etc/openldap/deltalogDB.ldif -W

```
dn: olcDatabase=hdb,cn=config
objectClass: olcBdbConfig
olcDatabase: hdb
olcDbDirectory: /var/lib/ldap/deltalog
olcSuffix: cn=deltalog
olcDbIndex: entryCSN,objectClass,reqEnd,reqResult,reqStart eq
```

※olcSuffix 和 olcRootDN(可选)必须以 cn=开头

※因这个数据库不是为查阅而使用的,为确保安全,不设置ACI。

※需要建立必要的索引

配置 Overlay

ldapadd -Y EXTERNAL -H ldapi:/// -f
/etc/openldap/overlay\_deltalog.ldif -W

dn: olcOverlay=accesslog,olcDatabase={2}hdb,cn=config
objectClass: olcAccessLogConfig
olcOverlay: accesslog
olcAccessLogDB: cn=deltalog
olcAccessLogOps: writes
olcAccessLogSuccess: TRUE
olcAccessLogPurge: 07+00:00 01+00:00

※Overlay 是可以出现多个并存的,相互组成链式,串行操作,所以同步的

accesslog 与上文用于记录的 accesslog 是可以同时存在的。

olcAccessLogDB: 用于记录日志的数据库

※会检查这个节点是否存在,如果不存在将打 log

additional info: <olcAccessLogDB> no matching backend found for suffix

※olcAccessLogDB 指向的数据库不能使用被记录的数据库,这会造成 slapd 死锁。

olcAccessLogSuccess: TRUE 只记录成功操作 FLASE 所有操作均记录

olcAccessLogOps: writes 只需记录对数据修改的操作

olcAccessLogPurge: 表示记录日志多久会失效删除和多久执行一次检查,格式

为

```
olcAccessLogPurge: age interval
[ddd+]hh:mm[:ss] [ddd+]hh:mm[:ss]
```

修改 syncprov

```
假定 syncprov 已顺利配置好,现在修改 syncprov 使其支持 accesslog。
```

```
ldapadd -Y EXTERNAL -H ldapi:/// -f
```

```
/etc/openldap/overlay_syncprov.ldif -W
```

13. 启用安全功能

```
dn: olcDatabase={2}hdb,cn=config
changetype: modify
replace: olcSyncrepl
olcSyncRepl: {0}rid=000
...
#其它配置照旧
logbase="cn=deltalog"
#可以使用 filter 进行过滤
logfilter="(&(objectClass=auditWriteObject)(reqResult=0))"
syncdata=accesslog
olcSyncRepl: {0}rid=001
...
#如果还有其它方向的 olcSyncRepl 则继续如上方式补充
```

※logfilter="(&(objectClass=auditWriteObject)(reqResult=0))"的用途是从已存储的 accesslog 中筛出 write 操作并且操作为成功的条目。这个动作是可选的,需根据 业务自行组织。

# 13.2.9 如何删除 overlay

停止 openIdap 服务:

systemctl stop slapd

以ppolicy 为例,进入:

/etc/openldap/slapd.d/cn=config/olcDatabase={2}hdb

目录下, 删除:

olcOverlay={0}ppolicy.ldif

即可。

注意:如果这个目录下只有要删除的模块一个 overlay 则直接删除。如果还有其 它 overlay,倘若待删除的 ppolicy.ldif 前缀"{}"里的数字是最大的(也就是最后 一个 overlay),则直接删除即可;否则删除后需要确保其它的 overlay 还能保持 连续不断,则重新将 ppolicy 之后的 overlay 的{N} N 重命名为合适的数字。

# 13.2.10 如何删除数据库

停止 openIdap 服务:

以 cn= accesslog 为例, 进入/etc/openIdap/slapd.d/cn=config

找到类似 olcDatabase={2}hdb.ldif 的文件, vi 打开确认内容为

olcSuffix: cn= accesslog

olcDbDirectory: /var/lib/ldap/accesslog

则为待删除数据库,删除 olcDatabase={2}hdb.ldif 和 olcDatabase={2}hdb 目录(如 果有的话),再删除/var/lib/ldap/accesslog 即可。

# 13.2.11 如何卸载 OpenLDAP

```
service slapd stop
yum remove openIdap-servers openIdap-clients compat-openIdap
rm -rf /etc/openIdap/slapd.d/*
rm -rf /var/lib/ldap/*
```

如果将/etc/openIdap 全部删除,在下次重新安装 openIdap 的时候可能存在问题。

# 13.2.12 Idapconfigcheck 工具

ldapconfigcheck 检查验证配置文件中的语法错误。

- 如果已加载 EsgynDB 环境 (sqenv.sh),则 ldapconfigcheck 会自动检查 文件\$MY SQROOT/sql/scripts/.traf authentication config。
- 如果未加载 EsgynDB 环境,您可以指定待检查的文件, EsgynDB 无需运行 ldapconfigcheck。

13.2.12.1 语法

如果未指定配置文件名,则ldapconfigcheck会查找使用环境变量的文件。

以下是环境变量和查找顺序:

- TRAFAUTH\_CONFIGFILE 指定完全限定名称。
- 2. TRAFAUTH\_CONFIGDIR

文件名.traf\_authentication\_config/追加至指定目录。

3. TRAF\_HOME

/sql/scripts/.traf\_authentication\_config追加至

 $TRAF\_HOME_{\,\circ}$ 

# ₽ 1110 <sub>示例</sub>

ldapconfigcheck - file myconfigfile

File myconfigfile is valid.

如果发现错误,ldapconfigcheck 会显示错误及行号。

# 13.2.12.2 错误

使用 ldapconfigcheck 时,可能会返回以下值,但仅报告遇到的第一个错误。

代码	文本
0	filename 文件有效。
1	未找到 filename 文件。
2	文件: filename line-number 行中的属性名称无效。
3	文件: <i>filename</i> line-number 行缺少值。
4	文件: <i>filename</i> line-number 行中的值超出范围。
5	文件: <i>filename</i> 打开 traf_authentication_config 文件失败。

6	文件: filename
	读取 traf_authentication_config 文件失败。
7	未提供文件。请指定文件参数或验证环境变量。
8	配置文件中至少已使用一次 TLS,但未提供
	TLS_CACERTFilename。
9	配置文件中至少有一组缺失主机名。
	每个 LDAP 连接配置必须提供至少一个主机名。
10	配置文件中至少有一组缺失唯一标识符。
	每个 LDAP 连接配置必须提供至少一个唯一标识符。
11	必须指定至少一个 LDAP 连接配置。
12	解析.traf_authentication_config 文件时出现内部错误。

# 13.2.13 ldapcheck 工具

ldapcheck 测试 AD/ LDAP 连接。

使用该命令时必须加载 EsgynDB 环境 (sqenv.sh), 但无需运行 EsgynDB 实例。如果仅测试连接,您能指定任何用户名或 group 名,此时,

traf\_authentication\_config 中的属性将查找您指定的用户名或 group 名。

13.2.13.1 语法

#### --verbose

如需查看更详细的错误信息,请使用-verbose选项。ldapcheck将日志事件记录在文件夹\$TRAF LOG,日志名称格式为

dbsecurity <host> <pid>.log.

如果提供密码,则ldapcheck将尝试验证指定的用户名和密码。以下示例显示了密码,但建议您将密码留空(--password=),ldapcheck将提示输入密码(不回显)。

#### ₽ 示例

ldapcheck --username=user1 --password=user1passwd Authentication request: externalName user1, configName 'local' (configNumber 0), result 0 (Authentication successful)

Member of group: group1

# 13.2.14 故障排除

问题:长时间使用后, OpenIdap 数据库无法使用, log 中出现类似

XXX Too many open files

的警告或错误。

回答: OpenIdap 默认对已超时未关闭的连接不做释放,长期开机会导致 FD 使用

到上限, 可配置

ldapmodify -Y EXTERNAL -H ldapi:/// -f /etc/openldap/dbtimeout.ldif -W

dn: cn=config
changetype: modify
add: olcIdleTimeout
olcIdleTimeout: 100

olcIdleTimeout:超时后的空闲连接多久关闭,单位为秒,0则为不关闭。 ※因 syncprov 模块(提供服务期间数据同步)的 refreshAndPersist(数据推送)需要维 持长连接,并长期处于空闲。这个"空闲"连接尽量不要断开。请将 olcIdleTimeout 设置大于 olcSyncRepl 属性中 interval 的值(见 13.2.3.3 章)。

问题: 如何解除对 slapd 进程的 ulimit 限制?

回答:修改 slapd 的 systemd 启动脚本, 增加 ulimit 所需参数

修改/usr/lib/systemd/system/slapd.service 加入对应参数

LimitCPU=	ulimit -t	Seconds
LimitFSIZE=	ulimit -f	Bytes
LimitDATA=	ulimit -d	Bytes
LimitSTACK=	ulimit -s	Bytes
LimitCORE=	ulimit -c	Bytes
LimitRSS=	ulimit -m	Bytes
LimitNOFILE=	ulimit -n	Number of File
Descriptors		
LimitAS=	ulimit -v	Bytes
LimitNPROC=	ulimit -u	Number of Processes
LimitMEMLOCK=	ulimit -l	Bytes
LimitLOCKS=	ulimit -x	Number of Locks
LimitSIGPENDING=	ulimit -i	Number of Queued
Signals		
LimitMSGQUEUE=	ulimit -q	Bytes
LimitNICE=	ulimit -e	Nice Level
LimitRTPRIO=	ulimit -r	Realtime Priority
LimitRTTIME=	No equivalent	

如果是要等价为 ulimit 的 unlimited,则配置为 infinity

问题:执行 sladindex 后无法启动数据库

回答:重新给/etc/openIdap 和 数据库存储目录权限为 Idap:Idap

问题:偶然遇见使用 yum 同时安装 openIdap-server 和 openIdap-client 后 openLDAP 服务无法正常启动。

回答:检查/etc/openIdap 及其下属目录和数据库存储目录权限是否为 Idap:Idap。

问题: cn=module,cn=config 节点的 olcModulePath 无法进行修改

回答:只能通过关闭 openLDAP 服务后手动修改

slap.d/cn=config/cn={0}module.ldif 来修改。

13.2.14.1 数据库备份

#### 1. 直接备份文件(冷备份)

需要备份的文件系统有:

1) 实例的动态配置文件

/etc/openIdap/slapd.d 目录

实例的 schema 文件(可选)

/etc/openldap/schema

2) 数据库本身

一般在/var/lib/ldap下,如不同时具体目录根据动态配置文件的olcDbDirectory属性决定。

备份前关闭 openIdap 服务,还原时注意将文件夹\文件的所有权改为 Idap:Idap

#### 2. 使用 slapcat 命令导出 DIT(热备份)

slapcat -a filter -b 数据库前缀 [-HURI] -l 输出文件 [-v]

-b:指定待备份数据的前缀,例如 cn=config

如果不指定-b则默认是导出第一个可用实例的属性,一般为这个 openIdap 服务器的一个租户信息

查询现在数据库中已存在的前缀:

slapcat -b cn=config |grep -E "^dn: olcDatabase="

-c 途中出错不中断

-1 默认输出为标准输出,-1 将输出至文件,内容为标准 ldif

示例:

slapcat -H ldapi:/// -b ou=People,dc=TestDB,dc=local -a

"(!(cn=test\*))" -1 /home/backup.ldif

将已导出的 ldif 重新导入数据

使用 slapcat 导出的数据库会包括一些由 openIdap 生成的只读属性,例如时间戳,导致无法导入。使用如下步骤删除这些只读属性:

```
建立过滤规则
vi filterInFile.regex
/^creatorsName: /d
/^modifiersName: /d
/^modifyTimestamp: /d
/^modifyTimestamp: /d
/^structuralObjectClass: /d
/^entryUUID: /d
/^entryUUID: /d
/^entryCSN: /d
执行命令:
cat /home/backup.ldif | sed -f filterInFile.regex >
/home/backup_new.ldif
导入 ldif:
ldapadd -H ldapi:/// -x -D "cn=admin,dc=TestDB,dc=local" -w
xxx -f /home/backup_new.ldif -c
```

#### 13.2.14.2 数据库恢复

#### 配置恢复:

由于 olc 配置出现故障导致 OpenLDAP 无法启动,则删除 slap.d 目录,从备份或 者是其它 HA 环境拷贝 slap.d 目录覆盖,在启动前请检查该 slap.d 配置里是否有 与 IP/hostname 相关的设置,例如 olcSyncrepl、 olcServerID 等,请使用文本编辑 器手动修改其值。

#### 数据恢复:

如果组成了 LDAP HA 环境,安装以下步骤操作。

1. 确保 HA 组里有一台完好的主机,将作为数据恢复的源头。

2. 检查步骤1的主机的 olcSyncRepl 属性,由

searchbase

scope

filter

attrs

等属性圈定的数据检索范围是否能覆盖待恢复主机整目录树。

例如作为恢复源头的 LDAP 配置信息:

主机根节点是 dc=TestDB,dc=local,这个属性来自

slapd.d/cn=config/olcDatabase={x}hdb.ldif 内 olcSuffix。

同步策略

olcSyncrepl: {0}rid=001 provider=ldap://192.168.137.155 bindmethod=simple bi nddn="cn=admin,dc=TestDB,dc=local" credentials="abc123\$" **searchbase="dc=Tes tDB,dc=local"** type=refreshAndPersist retry="5 5 300 5" timeout=1 schemachec king=off

表示将同步dc=TestDB,dc=local节点下所有的子节点,因这个节点就是根节点,则同步范围相当于整个数据库的条目都将同步。

3. 受损的 LDAP 主机停止服务 systemctl stop slapd

4. 备份并删除数据,目录在

slap.d/cn=config/olcDatabase={x}hdb.ldif 的 olcDbDirectory 可查看到,将该目录 下数据完全删除。

- 5. 启动 openLDAP 服务 systemctl start slapd
- 如果步骤2的检索范围不能覆盖整个目录树,需要预先导入基本的目录树结构,例如

```
dn: dc=TestDB,dc=local
objectClass: top
objectClass: dcObject
objectClass: organization
o: TestDB local
dc: TestDB
```

dn: ou=People,dc=TestDB,dc=local

objectClass: organizationalUnit ou: People dn: ou=Group,dc=TestDB,dc=local objectClass: organizationalUnit ou: Group

dn: ou=Pwpolicy,dc=TestDB,dc=local
objectClass: organizationalUnit
ou: Pwpolicy

7. 启动 OpenLDAP 服务后等待些许,该 OpenLDAP 服务器将会向其它 master 主机索取数据。

如果没有 HA 环境则需要定期使用 slapcat 导出数据,故障时删除数据库目录下 所有文件,启动 OpenLDAP 服务器后使用 ldapmodify 导回数据。

# 13.3 生成服务器证书

QianBase 使用证书加密/解密密码(验证用户),并为网页应用程序提供 HTTPS 支持。默认情况下,自签名证书使用 OpenSSL 生成,保存在集群每个节点的 \$HOME/sqcert 中。另外,您还能使用 CA 签名证书。

# 13.3.1 自签名证书

自签名证书是一个身份认证证书,该证书由一个实体签发,该实体的身份由其 自身进行认证。安装和更新时,QianBase 安装程序将生成自签名证书,并保存 在集群每个节点的\$HOME/sqcert 中。为避免频繁更新证书,现证书有效期设 为10年,保存以下文件:

• server.crt

证书。

server.key

私钥。
server.keystore

Java KeyStore, 用于保存实例在 SSL 加密时的安全证书。

## ●注意

Java KeyStore 保存授权证书或公钥证书,通常被Java 应用程序用于加密、验证和服务 HTTPS。keystore 密码保护 Java KeyStore 中的实体。keystore 实体由别名 (alias) 识别,它由 key 和证书组成,从而形成了信任链。

如果集群使用了证书且您收到证书过期的通知,则您需要手动生成自签名证书 (运行脚本 sqcertgen 和 sqcertget gen\_keystore),再重启连接和管 理服务(运行命令 dcsstop、mgblty\_stop、dcsstart 和 mgblty\_start)。

如需验证证书,运行命令 certcheck。

#### 13.3.2 生成 CSR

- 1. 使用服务器秘钥生成 CSR<sup>15</sup>,运行命令 sqcertgen gen\_csr。
- 2. 发送 CSR 至 CA<sup>16</sup>。

CA 签名后,您将得到一个签名证书,您可以在集群中部署该签名证书。

#### 13.3.3 CA 签名证书

CA 是发放数字证书的实体。数字证书证明证书主体拥有公钥,这允许依赖方 (Relying Parties) 依赖签名或使用对应的私钥进行认证。

CA 是受信任的第三方,它被证书所有者和依赖证书者信任。证书的格式遵循 X.509标准。

<sup>&</sup>lt;sup>15</sup>即 Certificate Signing Request, 证书注册请求。

<sup>&</sup>lt;sup>16</sup>即 Certificate Authorit, 证书授权中心。

安装 QianBase 后,您能部署 CA 签名证书 (运行命令 distcacert cosigned <pem file name>),再重启连接和管理服务 (运行命令 dcsstop、 mgblty stop、dcsstart 和 mgblty start)。

公共 (server.crt) 和私人 (server.key) 文件都应存放在

\$HOME/sqcert 中。

## 13.4 管理用户

AD/LDAP 对任何连接至 QianBase 的用户实施强制验证。QianBase 支持数据库 级、Schema 级、对象级 (表、视图和其它等)和操作级权限。启用权限功能后, 您能授予权限。如果启用了 AD/LDAP, 权限功能也将自动启用。

如需查看验证功能和权限功能的状态,在 sqlci 中执行命令 env。

示例
 11 04

Ĺ

>>env;	
Current Environment	;
AUTHENTICATION AUTHORIZATION CURRENT DIRECTORY	enabled enabled /opt/trafodion/esgynDB-2.5.x

启用权限功能后,QianBase 将会创建预定义数据库用户 DB\_ROOT 和 DB\_ADMIN,这些用户与您指定的 AD/LDAP 用户名(在安装 QianBase 时设 置)相关。请以用户 DB\_ROOT 或 DB\_ADMIN 身份登录 QianBase,创建所 需的 Schema、用户17、角色和权限18。

<sup>&</sup>lt;sup>17</sup> 更多关于**如何注册用户**的信息,请参阅《QianBase SQL 参考手册》的 **REGISTER USER Statement** 章节。

<sup>&</sup>lt;sup>18</sup>更多关于**如何向对象和角色授予权限**的信息,请参阅《QianBase SQL 参考手册》的 GRANT Statement 章节。

# 14. 提高安全性

本章讲述以下内容:

- 14.1 提高 Linux 安全性
- 14.2 提高 Hadoop 安全性
- 14.3 提高 Jetty Server 安全性
- 14.4 更新密码
- 14.5 提高端口安全性

## 14.1 提高 Linux 安全性

操作系统	版本	网址
RedHat		• Red Hat Enterprise Linux 7 安全指南
Enterprise	7 <b>.</b> x	https://access.redhat.com/documentation/en-
Linux		us/red_hat_enterprise_linux/7/html/security_guide/index
RedHat		• Red Hat Enterprise Linux 6 安全指南
Enterprise	6.x	https://access.redhat.com/documentation/en-
Linux		us/red_hat_enterprise_linux/6/html/security_guide/index
		• CentOS 安全指南
CentOS	通用	https://wiki.centos.org/HowTos/OS_Protection#head-
		f80d332aeea03f57d34d7a5c09493a7d69cce177

更多关于如何提高 Linux 安装过程的安全性,请参阅以下信息。

## 14.2 提高 Hadoop 安全性

更多关于如何提高 Hadoop 安装过程的安全性,请参阅以下信息。

Hadoop 发行版	网址			
Hortonworks' HDP 2.x	• Hadoop 安全指南			
	https://docs.hortonworks.com/HDPDocuments/HDP2/HDP-			
	2.5.x/bk_Security_Guide/content/ch_hdp-security-guide-			
	overview.html			
	• CDH 5 安全指南			
Cloudera CDH 5.x	https://www.cloudera.com/documentation/cdh/5-0-x/CDH5-			
	Security-Guide/CDH5-Security-Guide.html			

## 14.3 提高 Jetty Server 安全性

QianBase 服务器网页的大量组件使用 Jetty web 服务器模块, Jetty web 服务器使用 HTTPS 和强 SSL 密码。

更多关于如何提高 Jetty 的安全性,请参阅以下信息。

- 配置 Jetty Connector http://www.eclipse.org/jetty/documentation/current/configuring-connectors.html
- 配置 Jetty 安全性 <u>http://www.eclipse.org/jetty/documentation/current/configuring-security.html</u>

## 14.4 更新密码

将默认密码替换成安全性更高的密码。更多信息,请参阅 7.3 用户 ID 和密码。

## 14.5 提高端口安全性

以下端口需要对外部应用程序开放:

应用程序	端口号范围	说明
DCS Master	23400 ~ 23400+n n=MXOSRVR 的数量	开放端口号的范围取决于配置的 MXOSRVR 的数量
DB Manager	4206	

## 附录1. 验证配置文件

默认情况下, QianBase 验证配置文件保存在

 $\texttt{STRAF}_\texttt{HOME}/\texttt{sql}/\texttt{scripts}/\texttt{.traf}_\texttt{authentication}_\texttt{config} \ \psi_\circ$ 

下表为.traf\_authentication\_config 支持的属性及其说明。

属性名称	用途	示例值	注意
LDAPHostName	本地 LDAP 服务器的主机	ldap.master.com	如果提供两个及以上的
	名		LDAPHostName 值,
			EsgynDB 将尝试与每个
			LDAP 服务器连接。
			如果全部验证失败,系统
			将返回验证错误。
			更多信息,请参阅
			RetryCount 和
			RetryDelay.
LDAPPort	本地 LDAP 服务器的端口	345	必须为数值。与
	号		LDAPSSL 相关。
			OpenLDAP 标准端口
			号:
			<ul> <li>非安全: 389</li> </ul>
			• SSL: 636
			• TLS: 389

(续前表)

属性名称	用途	示例值	注意
LDAPSearchDN	如需搜索用户,请指定搜索	cn=aaabbb,	如果本地服务器允许匿名搜
	用户的区别名 (Distinguished Name)。	dc=net	索,则无需指定该属性或指 定该属性为空值。目前系统
			支持匿名搜索功能。
LDAPSearchPWD	LDAPSearchDN 的密码。 更多信息,请参阅 LDAPSearchDN。	welcome	/
LDAPSSL	指定本地 LDAP 服务器接口 不加密或使用 TLS 或 SSL。 0 表示未加密, 1 表示 SSL, 2 表示 TLS。	0	/
UniqueIdentifier	包含用户唯一标识符的目录 属性。	uid=,ou=Users ,dc=demo, dc=net	考虑到给定LDAP服务器支 持DN的多种形式,使用不 同值多次指定 UniqueIdentifier参数。查找 过程中,每个 UniqueIdentifier按照在配置 文件中列出的顺序被使用。

(续前表)

属性名称	用途	示例值	注意
LDAPNetworkTimeout	如果连接请求没有响应,	20	该参数的值必须为正数或
	指定下一个		-1, -1 表示无限超时。
	LDAPHostName 的超时		
	时间 (单位:秒)。		
	该参数与 ldap_conf(5)中		
	約 NETWORK_TIMEOUT		
	类似。		
	默认值为 30 秒。		
LDAPTimelimit	指定在 LDAP 服务器上执	15	在搜索过程中,服务器端
	行用户名搜索的等待时		可能使用比
	间。		LDAPTimelimit 更短的超
	该参数的值必须为正数。		时时间。
	该参数与 ldap_conf(5)中		
	的 TIMELIMIT 类似。		
	默认值为 30 秒。		
LDAPTimeout	指定在调用同步 LDAP	15	该参数的值必须为正数或
	API 后, 未收到回应到终		-1, -1 表示无限超时。
	止调用之间的超时时间		
	(单位:秒)。		
	该参数 ldap_conf(5)中的		
	TIMEOUT 类似。默认值		
	为30秒。		

(续前表)

属性名称	用途	示例值	注意
	, i - C		

RetryCount	建立成功 LDAP 连接的尝	10	重试失败操作时,
	试次数。		EsgynDB 将尝试连接每个
	默认值为 5。如果重试 5		配置 LDAP 服务器, 直到
	次后全部失败,将返回错		操作成功或超过允许的重
	误。		试次数。
RetryDelay	指定重试之间延迟的时	1	/
	间。		
	默认值为2秒。		
	更多信息,请参阅		
	RetryCount <sub>o</sub>		
PreserveConnection	操作完成后,指定 LDAP	YES	/
	服务器连接保留 (YES)		
	或关闭 (NO)。		
	默认值为 NO。		
RefreshTime	指定重新读取配置文件之	3600	如果设置为零,则不会读
	前必须等待的时间。		取配置文件。
	默认值为 1800 秒 (30 分		如果该值为零,则必须重
	钟)。		启服务器连接才能使更改
			生效。
			该属性不特定于任何配
			置,且必须在
			DEFAULTS 组定义。

(续前表)

属性名称	用途	示例值	注意
TLS_CACERTFilename	指定 LDAP 服务器证书	cert.pem	该属性适用于两种配置。
	文件的位置。文件名可以		如果一种配置无需证书,
	是绝对路径名或与		则忽略该属性。
	\$CACERTS_DIR 相关。		该属性必须在
			DEFAULTS 组定义。
DefaultSectionName	如果未指定验证类型,则	LOCAL	如果指定了
	指定由 REGISTER USER		DefaultSectionName 属
	命令分配给用户的配置类		性,则必须在
	型。		.traf_ldapconfig 中定义使
	初始 trafodion 版本仅支持		用该名称(或等效值)的
	一种配置。		部分。
			合法值为 LOCAL 和
			ENTERPRISE。该语法可
			能会更改。

## 附录 2. Inspector 工具

安装前, QianBase Python Installer 将检查集群的状态,例如,检查和记录硬件、 固件和软件的所有必要组件,以及所有子系统配置。

以下步骤为 QianBase Python Installer 调用预安装检查:

1. 显示检查结果概览。

\$ python-installer/inspector.py

2. 显示所有检查结果。

\$ python-installer/inspector.py --all

3. 无密码 ssh 运行检查程序。

\$ python-installer/inspector.py --all --enable-pwd

4. 使用指定远程用户, 在远程节点上无密码 ssh 运行检查程序。

```
$ python-installer/inspectors.py --all --enable-pwd --
remote-user <user>
```

5. 如果使用--enable-pwd 参数, 输入远程主机 ssh 密码。

Input remote host SSH Password:

6. 调用检查程序脚本时,输入目标节点列表。

```
Enter list of Nodes separated by comma, support numeric RE, i.e. n[01-12]:
```

# ┛注意

- 如果未指定--remote-user 选项,检查程序将使用目前登录用户作为远程用户。--remote-user 应在远程节点上具有 root 或 sudo 权限。
- 运行检查程序的节点需知晓目标节点的主机名和 IP 地址。
- 如需使用--enable-pwd 参数,您需安装 sshpass 工具,安装命令为 yum
   install -y sshpass。

#### ₽ 示例

```
[centos@esgvm-test python-installer]$ ./inspector.py
Enter list of Nodes separated by comma, support numeric RE,
i.e. n[01-12]: developer-[1-2]
```

TASK: Environment Discover

\*\*\*\*\*\*

Time Cost: 0 hour(s) 0 minute(s) 8 second(s)

Discover results

Hosts:test-1,test-2

+		-+-		•+-		-+
	OverView		Stat		Expected	
+		-+-		+-		-+
	CPU architecture		0		-	
	CPU cores		W		4	
	Disk numbers		Х		4	
	Free data File System spaces		Х		1000GB	
	Free System spaces		Х		200GB	
	Total memory size		Х		64GB	

	Current free memory size		W		8GB	
	Swap/Mem percentage		Х		25%	
	Network Card bandwidth		Х		10Gbps	
	Linux distro		0		_	
	FQDN		0		-	
	Localhost setting in /etc/hosts		0		—	
	Ntp service status		Х		-	
	Firewall status		0		-	
	Kernel pid max		0		—	
	Kernel tcp keep alive time		W		240	
	Kernel tcp keep alive interval		W		15	
	Kernel tcp keep alive probes		W		4	
	NFS on /home		0		-	
	Sudo access		0		-	
	NetworkManager service status		0		-	
	SSH PAM settings		0		-	
	Default java version		0		-	
	HBase version		0		-	
	HDFS version		0		-	
	Hive version		0		-	
	Leftover Trafodion process		0		-	
	license status		Х		-	
+		-+-		+		+

在上述命令输出中, Stat 列输出值的意义如下:

o:ok

- w: warn
- x : error

183

# 附录 3. QianBase 和 Hbase 参数优化

1. QianBase 数据库端参数设置

1) ms.env

添加:

JVM\_MAX\_HEAP\_SIZE\_MB=128 ESP\_JVM\_MAX\_HEAP\_SIZE\_MB=128 TM\_JAVA\_THREAD\_POOL\_SIZE=128 TM\_JAVA\_CP\_THREAD\_POOL\_SIZE=128 TMCLIENT\_POOL\_PUT\_SIZE=128 RMS\_SHARED\_SEG\_SIZE\_MB=256

注: RMS\_SHARED\_SEG\_SIZE\_MB 表示把 RMS 共享内存设置为 256MB, 默认 64MB

2) "\_MD\_".defaults

添加:

insert	into	"_MD_".defaults
values('ATTEMPT_P	ESP_PARALLELISN	4','OFF','ATTEMPT_ESP_PAR
ALLELISM',1);;	关闭 esp 并发	
insert	into	"_MD_".defaults
values('EXPLAIN_	IN_RMS','OFF',	'EXPLAIN_IN_RMS',1);关
闭rms 的query pla	in 日志,无法执行。	explain for qid 查看执行计划
insert	into	"_MD_".defaults
values('GENERATE	EXPLAIN', 'OFF	', 'GENERATE_EXPLAIN',1);
insert	into	"_MD_".defaults
values('HBASE_CAG	CHE_BLOCKS','ON	V', 'HBASE_CACHE_BLOCKS', 1
);		
insert	into	"_MD_".defaults
values('HBASE_REG	GION_SERVER_MAX	Leap_size','31744','HBA
SE_REGION_SERVER	MAX_HEAP_SIZE	',1);
insert	into	"_MD_".defaults
values('MDAM_SCAN	N_METHOD','OFF'	', 'MDAM_SCAN_METHOD', 1);
关闭 MDAM		
insert	into	"_MD_".defaults
values('MODE_COM	PATIBLE_1','ON'	', 'MODE_COMPATIBLE_1',1);
rownum 和 rowid	支持	
insert	into	"_MD_".defaults
values('VARCHAR_I	PARAM_DEFAULT_S	SIZE','8000','VARCHAR_PAR

AM_DEFAULT_SIZE',1);	解决参数过长的问题	
insert	into	"_MD_".defaults
values('QUERY_TEXT_CA	CHE','ON','QUERY_TE	XT_CACHE',1); -
-prepare 编译时间问题		
insert	into	"_MD_".defaults
values('QUERY_CACHE',	'65536', 'QUERY_CACH	E',1); 调 大
query cache, 默认16MB		
insert	into	"_MD_".defaults
<pre>values('CANCEL_QUERY_ ED',1);</pre>	ALLOWED', 'OFF', 'CAN	CEL_QUERY_ALLOW
insert	into	"_MD_".defaults
values('HBASE_DATA_BI	JOCK_ENCODING_OPTION	','FAST_DIFF','
HBASE_DATA_BLOCK_ENCC	DING_OPTION',1);	建表默认 encoding
insert	into	"_MD_".defaults
values('HBASE_COMPRES	SION_OPTION','SNAPP	Y', 'HBASE_COMPR
ESSION_OPTION',1);	-建表默认压缩格式	
insert	into	"_MD_".defaults
values('HBASE_MEMSTOF	RE_FLUSH_SIZE_OPTION	','1073741824',
'HBASE_MEMSTORE_FLUSH	H_SIZE_OPTION',1); -	建表默认 flush 大
'HBASE_MEMSTORE_FLUSE	H_SIZE_OPTION',1); -	建表默认 flush 大
'HBASE_MEMSTORE_FLUSH 小 insert	H_SIZE_OPTION',1); -	建表默认 flush 大 "_MD_".defaults
'HBASE_MEMSTORE_FLUSH 小 insert values('TRAF_DEFAULT_	H_SIZE_OPTION',1); - into _COL_CHARSET','UTF8'	建表默认 flush 大 "_MD_".defaults ,'TRAF_DEFAULT_
'HBASE_MEMSTORE_FLUSH 小 insert values('TRAF_DEFAULT_ COL_CHARSET',1);建	H_SIZE_OPTION',1); · into _COL_CHARSET','UTF8' 表默认编码为UTF8	建表默认 flush 大 "_MD_".defaults ,'TRAF_DEFAULT_
<pre>'HBASE_MEMSTORE_FLUSH 小 insert values('TRAF_DEFAULT_ COL_CHARSET',1);建 insert</pre>	H_SIZE_OPTION',1); · into _COL_CHARSET','UTF8' 表默认编码为UTF8 into	建表默认 flush 大 "_MD_".defaults ,'TRAF_DEFAULT_ "_MD_".defaults
<pre>'HBASE_MEMSTORE_FLUSH 小 insert values('TRAF_DEFAULT_ COL_CHARSET',1);建 insert values('TRAF_COL_LENG</pre>	H_SIZE_OPTION',1); - into _COL_CHARSET','UTF8' 表默认编码为UTF8 into ;TH_IS_CHAR','OFF','	建表默认 flush 大 "_MD_".defaults ,'TRAF_DEFAULT_ "_MD_".defaults TRAF_COL_LENGTH
<pre>'HBASE_MEMSTORE_FLUSE 小 insert values('TRAF_DEFAULT_ COL_CHARSET',1);建 insert values('TRAF_COL_LENGIS_CHAR',1);建表默</pre>	H_SIZE_OPTION',1); · into COL_CHARSET','UTF8' 表默认编码为UTF8 into TH_IS_CHAR','OFF',' 认为 bytes 而非 chars	建表默认 flush 大 "_MD_".defaults ,'TRAF_DEFAULT_ "_MD_".defaults TRAF_COL_LENGTH
<pre>'HBASE_MEMSTORE_FLUSE 小 insert values('TRAF_DEFAULT_ COL_CHARSET',1);建 insert values('TRAF_COL_LENGIS_CHAR',1);建表默 insert</pre>	H_SIZE_OPTION',1); into COL_CHARSET','UTF8' 表默认编码为UTF8 into STH_IS_CHAR','OFF',' 认为 bytes 而非 chars into	建表默认 flush 大 "_MD_".defaults ,'TRAF_DEFAULT_ "_MD_".defaults TRAF_COL_LENGTH "_MD_".defaults
<pre>'HBASE_MEMSTORE_FLUSH 小 insert values('TRAF_DEFAULT_ COL_CHARSET',1);建 insert values('TRAF_COL_LENG _IS_CHAR',1);建表默 insert values('DYNAMIC_PARAM</pre>	H_SIZE_OPTION',1); - into _COL_CHARSET','UTF8' 表默认编码为UTF8 into STH_IS_CHAR','OFF',' 认为 bytes 而非 chars into 1_DEFAULT_CHARSET','	建表默认 flush 大 "_MD_".defaults ,'TRAF_DEFAULT_ "_MD_".defaults TRAF_COL_LENGTH "_MD_".defaults UTF8','DYNAMIC_
<pre>'HBASE_MEMSTORE_FLUSH 小 insert values('TRAF_DEFAULT_ COL_CHARSET',1);建 insert values('TRAF_COL_LENG _IS_CHAR',1);建表默 insert values('DYNAMIC_PARAM PARAM_DEFAULT_CHARSET</pre>	<pre>H_SIZE_OPTION',1); - into _COL_CHARSET','UTF8' 表默认编码为UTF8 into GTH_IS_CHAR','OFF',' 认为bytes 而非 chars into 1_DEFAULT_CHARSET',' '',1);支持中文参数</pre>	建表默认 flush 大 "_MD_".defaults ,'TRAF_DEFAULT_ "_MD_".defaults TRAF_COL_LENGTH "_MD_".defaults UTF8','DYNAMIC_
<pre>'HBASE_MEMSTORE_FLUSH 小 insert values('TRAF_DEFAULT_ COL_CHARSET',1);建 insert values('TRAF_COL_LENG _IS_CHAR',1);建表默 insert values('DYNAMIC_PARAM PARAM_DEFAULT_CHARSET insert</pre>	<pre>H_SIZE_OPTION',1); - into COL_CHARSET','UTF8' 表默认编码为UTF8 into TH_IS_CHAR','OFF',' 认为bytes 而非 chars into 1_DEFAULT_CHARSET',' '',1);支持中文参数 into</pre>	建表默认 flush 大 "_MD_".defaults ,'TRAF_DEFAULT_ "_MD_".defaults TRAF_COL_LENGTH "_MD_".defaults UTF8','DYNAMIC_ "_MD_".defaults
<pre>'HBASE_MEMSTORE_FLUSH 小 insert values('TRAF_DEFAULT_ COL_CHARSET',1);建 insert values('TRAF_COL_LENG _IS_CHAR',1);建表默 insert values('DYNAMIC_PARAM PARAM_DEFAULT_CHARSET insert values('TRAF_ENABLE_M</pre>	<pre>H_SIZE_OPTION',1); - into _COL_CHARSET','UTF8' 表默认编码为UTF8 into STH_IS_CHAR','OFF',' 认为bytes 而非 chars into 4_DEFAULT_CHARSET',' '',1);支持中文参数 into 1ETADATA_LOAD_IN_CAC</pre>	建表默认 flush 大 "_MD_".defaults ,'TRAF_DEFAULT_ "_MD_".defaults TRAF_COL_LENGTH "_MD_".defaults UTF8','DYNAMIC_ "_MD_".defaults HE','ON','TRAF_

3) dcs-site.xml

添加:

```
<property>
</property>
</property>
</property>
<property>
<property>
<property>
<property>
<property>
<property>
</property>
```

2. JDBC 端修改内容

JDBC URL 配置如下

jdbc:t4jdbc://10.10.12.25:23400/:schema=V7FAT;maxStatem ents=400;connectionTimeout=7200

3. HBase 配置修改

hbase.master.handler.count 100 hbase.regionserver.handler.count 200

# 附录 4. QianBase 在线增加节点

- 在 HDP/CDH 中手动增加 HBase Regionserver 节点、HDFS Datanode 节点和 Yarn/MapReduce 节点<sup>19</sup>。
- 2. 增加 QianBase 节点。

在现有节点上输入:

```
[root@esgvm-test python-installer]# ./add_nodes.py --
nodes=suyan20,suyan21,suyan22,
suyan23,suyan24
```

#### 输出

***************************************
Trafodion Elastic Add Nodes script
* * * * * * * * * * * * * * * * * * * *
***[INFO]: Creating trafodion packages of /opt/trafodion/
esgyndb, this will take awhile
***[INFO]: Copying trafodion files to new nodes, this will
take a while
****************
AddNode sub scripts start
***************
***[INFO]: Running add node setup on new node(s) [suyan20,
suyan21,suyan22,suyan23,suyan24]
TASK: Add nodes Setup ************************************
TASK: Install Trafodion dependencies ************************************
Time Cost: 0 hour(s) 2 minute(s) 26 second(s)
***[INFO]: Running dcs setup on all node(s)
[suyan02, suyan03,
suyan04,suyan05,suyan06,suyan07,suyan08,suyan17,suyan18,suy
an19,suyan20,suyan21,suyan22,suyan23,suyan24]

<sup>&</sup>lt;sup>19</sup> 更多关于**如何在 HDP/CDH 中增加这些节点**的信息,请参阅《如何在 Hadoop 和 Cloudera 集 群增加节点》。

```
Time cost: 0 hour(s) 0 minute(s) 51 second(s)
***[INFO]: Trafodion instance is up, adding node in
sqshell ...
***[INFO]: Trafodion instance is up, adding node in
sqshell ...
***[INFO]: adding node [suyan20] in sqshell ...
***[OK]: Node [suyan20] added!
***[INFO]: adding node [suyan21] in sqshell ...
***[OK]: Node [suyan21] added!
***[INFO]: adding node [suyan22] in sqshell ...
***[OK]: Node [suyan22] added!
***[INFO]: adding node [suyan23] in sqshell ...
***[OK]: Node [suyan23] added!
***[INFO]: adding node [suyan24] in sqshell ...
***[OK]: Node [suyan24] added!
***[INFO]: starting DCS on new nodes ...
***[INFO]: Run sqregen ...
AddNode complete
```

3. 检查 Trafodion 运行状况。

### 输入

sqcheck

## 输出

*** Checki	ng Trafodion	Environmen	t ***			
Checking i	f processes	are up.				
Checking a	ttempt: 1; u	ser specifi	ed max:	2. E	xecution	time
in seconds	: 0.					
The Trafod	ion environm	ent is up!				
Process	Configured	Actual	Down			
DTM	23	23				
RMS	46	46				
DcsMaster	1	1				
Dcsserver	23	23				
mxosrvr	2254	2254				
Restserver	1	1				

4. 检查每个节点的运行状况。

### 输入

sqshell -a

#### 输出

[\$Z000EJ2]	Shell/shell Ver	sion 1.0	.1 EsgynDB 1	Release 2.5.x
(Build rel	ease [centos], d	ate 09Nov	<i>s</i> 17)	
[\$Z000EJ2]	%node info			
[\$Z000EJ2]	Logical Nodes	= 23		
[\$Z000EJ2]	Physical Nodes	=	23	
[\$Z000EJ2]	Spare Nodes	= 0		
[\$Z000EJ2]	Available spare	s = 0		
[\$Z000EJ2]	NID Type	State	Processors	#Procs
[\$Z000EJ2]	PNID	State	#Cores	MemFree
SwapFree	CacheFree Name			

[\$Z000EJ2]										
[\$Z000EJ2]	000	Any U	q		2	109	9			
[\$Z000EJ2]		00	0	Up	)	40	5763	1536	669	99072
277981484 :	suyar	n02								
[\$Z000EJ2]	001	Any	Up		2	106	5			
[\$Z000EJ2]		00	1	Up	)	40	259	7588	669	75220
281136344 :	suyar	103								
[\$Z000EJ2]	002	Any	Up		2	106	5			
[\$Z000EJ2]		00	2	Up	)	40	1839	9660	669	39476
280069928 :	suyar	n04								
[\$Z000EJ2]		009	Up		40	2377	144	66977	628 2	80325836
suyan11										
[\$Z000EJ2]	010	Any	Up		2		106			
[\$Z000EJ2]		010	Up		40	1250	020	6703	86444	279768248
suyan12										
[\$Z000EJ2]	011	Any	Up		2		106			
[\$Z000EJ2]		011	Up		40	1271	592	6698	5060	280740440
suyan13										
[\$Z000EJ2]	012	Any	Up		2		106			
[\$Z000EJ2]		012	Up		40	2748	056	6697	1736	280183032
suyan14										
[\$Z000EJ2]	013	Any	Up		2		106			
[\$Z000EJ2]		013	Up		40	41488	500	67079	948 2	82424384
suyan15										
[\$Z000EJ2]	014	Any	Up		2		106			
[\$Z000EJ2]		014	Up		40	1519	468	6696	50032	281095480
suyan16										
[\$Z000EJ2]	015	Any	Up		2		106			
[\$Z000EJ2]		015	Up		40	1548	692	6697	8064	280175188
suyan17										
[\$Z000EJ2]	016	Any	Up		2		106			
[\$Z000EJ2]		016	Up		40	3712	684	6691	4132	281733864
suyan18										
[\$Z000EJ2]	017	Any	Up		2		106			
[\$Z000EJ2]		017	Up		40	2621	536	6698	80652	280225512
suyan19										

190

[\$Z000EJ2]	018	Any	Up	2	108		
[\$Z000EJ2]		018	Up	40	75295732	66950316	335096680
suyan20							
[\$Z000EJ2]	019	Any	Up	2	108		
[\$Z000EJ2]		019	Up	40	35179380	67051084	306800276
suyan21							
[\$Z000EJ2]	020	Any	Up	2	108		
[\$Z000EJ2]		020	Up	40	64271476	66985376	332396772
suyan22							
[\$Z000EJ2]	021	Any	Up	2	108		
[\$Z000EJ2]		021	Up	40	53754092	67105416	312541096
suyan23							
[\$Z000EJ2]	022	Any	Up	2	108		
[\$Z000EJ2]		022	Up	40	46462700	67097284	310962924
suyan24							

5. 重启 HBase RegionServer。

在 Cloudera Manager 管理界面上,重启新增节点的 HBase RegionServer。

重启		×
是否确定要在以下角色上运行	重启命令?	
主机	角色实例	
suyan[20-24]	HBase RegionServer	
		重启取消

## 附录 5. QianBase 离线删除节点

1. 停止数据库。

输入

sqstop

#### 输出

```
Stopping DBMgr...
2017-11-16_12:36:51: Stopping EsgynDB Manager pid (32532)
2017-11-16_12:36:54: stopped EsgynDB Manager
Stopping Bosun...
2017-11-16_12:36:59: Bosun process is not started
SQ shutdown (normal) from
/opt/trafodion/esgyndb/sql/scripts successful
```

2. 检查 Trafodion 运行状况。

#### 输入

sqcheck

#### 输出

[trafodion@es	gvm-test s	cript	s]\$	sqche	ck	
*** Checking '	Trafodion	Envir	onme	nt **	*	
Checking if p	rocesses a	re up	•			
Checking atter	mpt: 1; us	er sp	ecif	ied m	ax: 2.	Execution time
in seconds: 4						
The Trafodion	environme	nt is	not	up a	t all,	or partially up
and not opera	tional. Ch	eck t	he l	ogs.		
Process	Configured	dActu	al	Down		
DTM	0		0			
RMS	0		0			
DcsMaster	1		0		1	
Dcsserver	23		0		23	
mxosrvr	2254	0		2254		

Restserver 0 0

The Trafodion environment is down.

3. 修改 servers 文件。

输入

pwd

#### 输出

/opt/trafodion/esgyndb/dcs-2.5.x/conf

#### 输入

cat servers

#### 输出

suyan02	98
suyan03	98
suyan04	98
suyan05	98
suyan06	98
suyan07	98
suyan08	98
suyan09	98
suyan10	98
suyan11	98
suyan12	98
suyan13	98
suyan14	98
suyan15	98
suyan16	98
suyan17	98
suyan18	98
suvan19	98

4. 将 servers 文件复制至 pdcp 'trafconf -wname' servers \$PWD/ (所

有节点)的当前目录。

#### 输入

pdcp \$(trafconf -wname) servers \$PWD/

5. 修改 sqconfig 文件, 删除待删除节点的信息。

#### 输入

pwd

#### 输出

/opt/trafodion/esgyndb/sql/scripts

#### 输入

vi sqconfig

#### 输出

<pre>node-id=0;node-name-suyan02;cores=0-</pre>
39;processors=2;roles=connection,aggregation,storage
<pre>node-id=1;node-name-suyan03;cores=0-</pre>
39;processors=2;roles=connection,aggregation,storage
<pre>node-id=2;node-name-suyan04;cores=0-</pre>
39;processors=2;roles=connection,aggregation,storage
<pre>node-id=3;node-name-suyan05;cores=0-</pre>
39;processors=2;roles=connection,aggregation,storage
<pre>node-id=4;node-name-suyan06;cores=0-</pre>
39;processors=2;roles=connection,aggregation,storage
<pre>node-id=5;node-name-suyan07;cores=0-</pre>
<pre>39;processors=2;roles=connection,aggregation,storage</pre>
<pre>node-id=6;node-name-suyan08;cores=0-</pre>
39;processors=2;roles=connection,aggregation,storage
<pre>node-id=7;node-name-suyan09;cores=0-</pre>
39;processors=2;roles=connection,aggregation,storage
<pre>node-id=8;node-name-suyan10;cores=0-</pre>
<pre>39;processors=2;roles=connection,aggregation,storage</pre>

node-id=9;node-name-suyan11;cores=0-39; processors=2; roles=connection, aggregation, storage node-id=10;node-name-suyan12;cores=0-39; processors=2; roles=connection, aggregation, storage node-id=11;node-name-suyan13;cores=0-39; processors=2; roles=connection, aggregation, storage node-id=12;node-name-suyan14;cores=0-39; processors=2; roles=connection, aggregation, storage node-id=13;node-name-suyan15;cores=0-39; processors=2; roles=connection, aggregation, storage node-id=14;node-name-suyan16;cores=0-39;processors=2;roles=connection,aggregation,storage node-id=15;node-name-suyan17;cores=0-39; processors=2; roles=connection, aggregation, storage node-id=16;node-name-suyan18;cores=0-39; processors=2; roles=connection, aggregation, storage node-id=17;node-name-suyan19;cores=0-39; processors=2; roles=connection, aggregation, storage node-id=18;node-name-suyan20;cores=0-39; processors=2; roles=connection, aggregation, storage node-id=19;node-name-suyan21;cores=0-39; processors=2; roles=connection, aggregation, storage node-id=20;node-name-suyan22;cores=0-39; processors=2; roles=connection, aggregation, storage node-id=21;node-name-suyan23;cores=0-39; processors=2; roles=connection, aggregation, storage node-id=22;node-name-suyan24;cores=0-39; processors=2; roles=connection, aggregation, storage

删除待删除节点的信息后, sqconfig 文件为:

<pre>node-id=0;node-name-suyan02;cores=0-</pre>
<pre>39;processors=2;roles=connection,aggregation,storage</pre>
node-id=1;node-name-suyan03;cores=0-
<pre>39;processors=2;roles=connection,aggregation,storage</pre>
node-id=2;node-name-suyan04;cores=0-
39;processors=2;roles=connection,aggregation,storage

<pre>node-id=3;node-name-suyan05;cores=0-</pre>
39;processors=2;roles=connection,aggregation,storage
node-id=4;node-name-suyan06;cores=0-
<pre>39;processors=2;roles=connection,aggregation,storage</pre>
node-id=5;node-name-suyan07;cores=0-
<pre>39;processors=2;roles=connection,aggregation,storage</pre>
node-id=6;node-name-suyan08;cores=0-
<pre>39;processors=2;roles=connection,aggregation,storage</pre>
node-id=7;node-name-suyan09;cores=0-
<pre>39;processors=2;roles=connection,aggregation,storage</pre>
node-id=8;node-name-suyan10;cores=0-
<pre>39;processors=2;roles=connection,aggregation,storage</pre>
node-id=9;node-name-suyan11;cores=0-
<pre>39;processors=2;roles=connection,aggregation,storage</pre>
node-id=10;node-name-suyan12;cores=0-
<pre>39;processors=2;roles=connection,aggregation,storage</pre>
node-id=11;node-name-suyan13;cores=0-
39;processors=2;roles=connection,aggregation,storage
<pre>39;processors=2;roles=connection,aggregation,storage node-id=12;node-name-suyan14;cores=0-</pre>
<pre>39;processors=2;roles=connection,aggregation,storage node-id=12;node-name-suyan14;cores=0- 39;processors=2;roles=connection,aggregation,storage</pre>
<pre>39;processors=2;roles=connection,aggregation,storage node-id=12;node-name-suyan14;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=13;node-name-suyan15;cores=0-</pre>
<pre>39;processors=2;roles=connection,aggregation,storage node-id=12;node-name-suyan14;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=13;node-name-suyan15;cores=0- 39;processors=2;roles=connection,aggregation,storage</pre>
<pre>39;processors=2;roles=connection,aggregation,storage node-id=12;node-name-suyan14;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=13;node-name-suyan15;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=14;node-name-suyan16;cores=0-</pre>
<pre>39;processors=2;roles=connection,aggregation,storage node-id=12;node-name-suyan14;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=13;node-name-suyan15;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=14;node-name-suyan16;cores=0- 39;processors=2;roles=connection,aggregation,storage</pre>
<pre>39;processors=2;roles=connection,aggregation,storage node-id=12;node-name-suyan14;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=13;node-name-suyan15;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=14;node-name-suyan16;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=15;node-name-suyan17;cores=0-</pre>
<pre>39;processors=2;roles=connection,aggregation,storage node-id=12;node-name-suyan14;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=13;node-name-suyan15;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=14;node-name-suyan16;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=15;node-name-suyan17;cores=0- 39;processors=2;roles=connection,aggregation,storage</pre>
<pre>39;processors=2;roles=connection,aggregation,storage node-id=12;node-name-suyan14;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=13;node-name-suyan15;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=14;node-name-suyan16;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=15;node-name-suyan17;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=16;node-name-suyan18;cores=0-</pre>
<pre>39;processors=2;roles=connection,aggregation,storage node-id=12;node-name-suyan14;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=13;node-name-suyan15;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=14;node-name-suyan16;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=15;node-name-suyan17;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=16;node-name-suyan18;cores=0- 39;processors=2;roles=connection,aggregation,storage</pre>
<pre>39;processors=2;roles=connection,aggregation,storage node-id=12;node-name-suyan14;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=13;node-name-suyan15;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=14;node-name-suyan16;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=15;node-name-suyan17;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=16;node-name-suyan18;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=16;node-name-suyan18;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=17;node-name-suyan19;cores=0-</pre>
<pre>39;processors=2;roles=connection,aggregation,storage node-id=12;node-name-suyan14;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=13;node-name-suyan15;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=14;node-name-suyan16;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=15;node-name-suyan17;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=16;node-name-suyan18;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=16;node-name-suyan18;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=17;node-name-suyan19;cores=0- 39;processors=2;roles=connection,aggregation,storage</pre>
<pre>39;processors=2;roles=connection,aggregation,storage node-id=12;node-name-suyan14;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=13;node-name-suyan15;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=14;node-name-suyan16;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=15;node-name-suyan17;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=16;node-name-suyan18;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=17;node-name-suyan19;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=17;node-name-suyan19;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=18;node-name-suyan20;cores=0-</pre>
<pre>39;processors=2;roles=connection,aggregation,storage node-id=12;node-name-suyan14;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=13;node-name-suyan15;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=14;node-name-suyan16;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=15;node-name-suyan17;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=16;node-name-suyan18;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=17;node-name-suyan19;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=17;node-name-suyan19;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=18;node-name-suyan20;cores=0- 39;processors=2;roles=connection,aggregation,storage</pre>
<pre>39;processors=2;roles=connection,aggregation,storage node-id=12;node-name-suyan14;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=13;node-name-suyan15;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=14;node-name-suyan16;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=15;node-name-suyan17;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=16;node-name-suyan18;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=17;node-name-suyan19;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=18;node-name-suyan20;cores=0- 39;processors=2;roles=connection,aggregation,storage node-id=18;node-name-suyan20;cores=0-</pre>

6. 备份 sqconfig.db 文件。

### 输入

mv sqconfig.db sqconfig.db.bak

7. 生成启动脚本和配置数据库。

#### 输入

sqgen

### 输出

Checking for the configuration file (sqconfig.db).
suyan02,suyan03,suyan04,suyan05,suyan06,suyan07,suyan08,suya
n09, suyan10, suyan11, suyan12, suyan13, suyan14, suyan15, suyan16,
suyan17,suyan18,suyan19
Creating directories on cluster nodes
/usr/bin/pdsh -R exec -w
suyan02,suyan03,suyan04,suyan05,suyan06,suyan07,suyan08,suya
n09, suyan10, suyan11, suyan12, suyan13, suyan14, suyan15, suyan16,
suyan17,suyan18,suyan19 -x suyan02 ssh -q -n %h mkdir -p
/opt/trafodion/esgyndb/etc
/usr/bin/pdsh -R exec -w
suyan02,suyan03,suyan04,suyan05,suyan06,suyan07,suyan08,suya
n09, suyan10, suyan11, suyan12, suyan13, suyan14, suyan15, suyan16,
suyan17,suyan18,suyan19 -x suyan02 ssh -q -n %h mkdir -p
/opt/trafodion/esgyndb/logs
/usr/bin/pdsh -R exec -w
suyan02,suyan03,suyan04,suyan05,suyan06,suyan07,suyan08,suya
n09, suyan10, suyan11, suyan12, suyan13, suyan14, suyan15, suyan16,
suyan17,suyan18,suyan19 -x suyan02 ssh -q -n %h mkdir -p
/opt/trafodion/esgyndb/tem
/usr/bin/pdsh -R exec -w
suyan02,suyan03,suyan04,suyan05,suyan06,suyan07,suyan08,suya
n09, suyan10, suyan11, suyan12, suyan13, suyan14, suyan15, suyan16,
suyan17,suyan18,suyan19 -x suyan02 ssh -q -n %h mkdir -p
/opt/trafodion/esgyndb/sql/scripts

```
The SQ environment variable file
/opt/trafodion/esgyndb/etc/ms.env exists.
The file will not be re-generated.
Copying the generated files to all the nodes in the cluster
Copying /opt/trafodion/esgyndb/etc/ms.env to
/opt/trafodion/esgyndb/etc of all the nodes
/usr/bin/pdcp -R ssh -w suyan02, suyan03, suyan04, suyan05,
suyan06, suyan07, suyan08, suyan09, suyan10, suyan11,
suyan12, suyan13, suyan14, suyan15, suyan16, suyan17,
suyan18, suyan19| -x suyan02
/opt/trafodion/esgyndb/etc/ms.env /opt/trafodion/esgyndb/etc
Copying /opt/trafodion/esgyndb/etc/seamonster.env to
/opt/trafodion/esgyndb/etc of all the nodes
/usr/bin/pdcp -R ssh -w
suyan02, suyan03, suyan04, suyan05, suyan06, suyan07, suyan08, suya
n09,suyan10,suyan11,suyan12,suyan13,suyan14,suyan15,suyan16,
suyan17, suyan18, suyan19 -x suyan02
/opt/trafodion/esgyndb/etc/ms,env
/opt/trafodion/esgyndb/etc
Copying /opt/trafodion/esgyndb/etc/seamonster.env to
/opt/trafodion/esgyndb/etc of all the nodes
/usr/bin/pdcp -R ssh -w
suyan02,suyan03,suyan04,suyan05,suyan06,suyan07,suyan08,suya
n09, suyan10, suyan11, suyan12, suyan13, suyan14, suyan15, suyan16,
suyan17,suyan18,suyan19 -x suyan02
/opt/trafodion/esgyndb/etc/seamonster.env
/opt/trafodion/esgyndb/etc
pdcp@suyan02: can't stat
/opt/trafodion/esgyndb/etc/seamonster.env
```

8. 验证环境变量 MY NODES 是否正确。

输入以下命令后将返回集群中节点的信息,此时,应不包含被删除的节

点,否则,需重新登录验证环境变量是否正确。

### 输入

su -	trafodion	
echo	\$(trafconf	-wname)

#### 9. 启动 Trafodion。

输入

sqstart

#### 输出

Checking if p	processes are up			
Checking atte	empt: 1; user spe	ecified max:	2. Execution	time
in seconds: (	).			
The Trafodior	n environment is	up!		
Process	Configured	Actual	Down	
DTM	18	18		
RMS	36	36		
DcsMaster	1	1		
Dcsserver	18	18		
mxosrvr	1764	1226 53	8	
Restserver	1	1		
Startup time	0 hour(s) 7 mir	ute(s) 57 s	econd(s)	

10. 检查 Trafodion 运行状况。

输入

sqcheck

#### 输出

\*\*\* Checking Trafodion Environment \*\*\*

Checking if processes are up.

Checking at	tempt: 1; us	er spe	ecified	max:	2.	Execution	time
in seconds:	1.						
The Trafodio	on environme	nt is	up!				
Process	Configured		Actua	1	Dow	'n	
DTM	18		18				
RMS	36		36				
DcsMaster	1	1					
Dcsserver	18		18				
mxosrvr	1764		1764				
Restserver	1		1				

# 附录 6. 安装后配置 DCS Master 的 HA

- 1. 配置服务端文件。
  - (1) 复制以下内容至 DCS\_INSTALL\_DIR/conf/dcs-site.xml 文件。

<property></property>
<name>dcs.zookeeper.property.clientPort</name>
<value>2181</value>
<property></property>
<name>dcs.zookeeper.quorum</name>
<value>gy08.esgyncn.local,gy07.esgyncn.local,gy09.esgyncn.</value>
local
<property></property>
<name>dcs.dns.interface</name>
<value>eth1</value>
<property></property>
<name>dcs.master.floating.ip</name>
<value>true</value>
<property></property>
<name>dcs.master.floating.ip.external.interface</name>
<value>eth1</value>
<property></property>
<name>dcs.master.floating.ip.external.ip.address</name>
<value>10.10.12.252</value>

(2) 复制以下内容至 DCS INSTALL DIR/conf/masters 文件。

gy09.esgyncn.local gy07.esgyncn.local

gy10.esgyncn.local

(3) 复制以下内容至 vi /home/trafodion/.bashrc 文件。

```
export ENABLE_HA=true
then have to re-login all terminates
```

(4) 复制以下内容至 MY\_SQROOT/dbmgr-2.5.x/conf/config.xml 文

件。

jdbc:t4jdbc://suyan02:23400/:

2. 配置客户端。

在客户端的 Connection String 中指定浮动 IP (10.10.12.252:23400),而不能 DCS 节点的 IP。

3. 配置操作系统。

复制以下内容至/etc/sudoers.d/trafodion。

## Trafodion Floating IP commands Cmnd\_Alias IP = /sbin/ip Cmnd\_Alias ARP = /sbin/arping ## Allow Trafodion id to run commands needed to configure floating IP %trafodion ALL = NOPASSWD: IP,ARP ## Allow trafodion id to run commands needed for backup and restore %trafodion ALL = (hbase) NOPASSWD: /usr/bin/hbase

4. 重新启动 DCD Master。

### 输入

dcsstop			
dcsstart			
dcscheck			

## 附录7. 内外网映射指南

下文演示以下测试环境为例:

Trafodion Cluster: 10.10.23.19, 10.10.23.11, 10.10.23.20, 10.10.23.21, 10.10.23.22 HA DcsMaster virtual address : 10.10.23.120

DcsMaster list: 10.10.23.11, 10.10.23.20, 10.10.23.21, 10.10.23.22

Active DcsMaster:10.10.23.20

# ▲注意

10.10.23.19, 是管理节点没有安装 trafodion

dcscheck 如下:

[trafodion@esggy	y-clu-n011 conf	]\$ dcscheck	
Cluster Configu	ration : HA	Α	
DcsMaster virtua	al address : 10	0.10.23.120	
Configured DcsMa	aster(s) : es	ggy-clu-n011.esgyn	.cn esggy-clu-
n002.esgyncn.loo	cal esggy-o	clu-n012.esgyn.cn	esggy-clu-
n013.esgyn.cn			
Active DcsMaster	r(s) : es	sggy-clu-n011	
DcsMaster lister	n port : 23	3400	
Process	Configured	Actual	Down
DcsMaster	4	4	
DcsServer	4	4	
mxosrvr	32	32	

网络的映射配置:

测试环境中, client 端是没有办法直接通过 jdbc 的 url 上写的内网地址 (10.10.23.120)去访问 trafodion 的数据库,所以对数据库的内网地址和外网的地 址做了如下映射:

- 192.167.1.90---→10.10.23.11
- 192.167.1.91---→10.10.23.19
- 192.167.1.92---→10.10.23.20
- 192.167.1.93---→10.10.23.21
- 192.167.1.94---→10.10.23.22
- 192.167.1.95---→10.10.23.120

这样映射之后我们可以通过在 jdbc 的 url 上填写 192.167.1.95 来通过策略访问到

内部 ip 10.10.23.120 的 trafodion。

Server 端配置:

1. 配置 dcs-site.xml 文件。

该文件的目录为 \$DCS\_INSTALL\_DIR/conf/dcs-site.xml, 需要在该文件中增加一个属性 dcs.default.ip.mapping, value 可以自定义,此处

为 default

<property>
<name>dcs.default.ip.mapping</name>
<value>default</value>
</property>

2. 配置 ipmapping.conf 文件。

该文件的目录为\$DCS\_INSTALL\_DIR/conf/ipmapping.conf,如果没有,则需要新建。

具体的配置规格如下:示例的文件中有3列,列名分别为 innerIP, default,

mapping1, innerIP 表示内网不能访问的目标 ip 地址, 列值可以为

10.10.23.[19-22] 或 10.10.23.11 或 10.10.23.120。default 和 mapping1 都表示自己

定义的映射名,其中 default 这个名字应该和 dcs-site.xml 的

dcs.default.ip.mapping 的 value default 是对应的。

```
[trafodion@esggy-clu-n011 conf]$ vi ipmapping.conf
# @@@ START COPYRIGHT @@@
#
#
# (C) Copyright 2015-2018 Esgyn Corporation
#
```

# @@@ END COPYRIGHT @@@	
#config	
innerIP, default,	mapping1
10.10.23.11, 192.167.1.80,	192.167.1.90
10.10.23.19, 192.167.1.81,	192.167.1.91
10.10.23.20, 192.167.1.82,	192.167.1.92
10.10.23.21, 192.167.1.83,	192.167.1.93
10.10.23.22, 192.167.1.84,	192.167.1.94
10.10.23.120, 192.167.1.85,	192.167.1.95

#表示注释, innerIP和 default和 mapping1必须要用逗号分割,这几个列名下面的 ip 一行就是一组映射关系。此次演示的每一行就是两个对应关系即 default→innerIP 和 mapping1→innerIP, 当 client 的 url 传过来的是 ipMapping=default 就对应 default→innerIP 的映射规则,当传过来的是 ipMapping=mapping1则对应的 mapping1→innerIP 映射规则。

# ▲注意

ipmapping.conf 文件中不允许有空格行,否则该文件会失效。

192.167.1.8x 的 IP 是不能成功映射到对应的 innerIP 的。

Client 端配置:

配置 client,目前仅仅是支持 jdbc, client 的配置主要是增加了一个 ipMapping 的 一个属性,该 ipMapping 的 value 要对应 server 端的 ipmapping.conf 里面的 mapping1 或 default 才会使用到配置文件里的映射规则,例如: String URL = "jdbc:t4jdbc://192.167.1.95:23400/:ipMapping=mapping1";

具体的使用:

**URL** = "jdbc:t4jdbc://192.167.1.95:23400/:"

1. 当 client 端的 url 上没有传入 ipMapping 这个属性和对应的 value,并且 dcssite.xml 的 property 里没有 dcs.default.ip.mapping 这个属性仅仅存在已经配置好的 ipmapping.conf 文件时不能连接。
- a. client 端不传 ipMapping
- b. dcs-site.xml 不配置 dcs.default.ip.mapping
- c. 存在配置好的 ipmapping.conf

结果:链接失败

2. 当 client 端的 url 上没有传入 ipMapping 以及 value 并且 dcs-site.xml 的 property 里已经配置了 dcs.default.ip.mapping=default 这个属性,也存在已经配置好的 ipmapping.conf 文件时, client 则会是根据 dcs-site.xm 中 dcs.default.ip.mapping 的 value default 去走 ipmapping.conf 的 default 的映射关系。根据上面的 ipmapping.conf 的配置,映射规则 default 是不能连接的。(黄色部分不是很通顺)

a. client 端不传 ipMapping

- b. dcs-site.xml 配置 dcs.default.ip.mapping=default
- c. 存在配置好的 ipmapping.conf

结果:链接失败。

3. 当 client 端的 url 上增加 ipMapping=mapping1 的属性,该 value 等于 ipmapping.conf 的关键字 mapping1(client 的 url 上配置了 ipMapping 的属性之后 dcs-site.xml 的 property 里 dcs.default.ip.mapping=default 这个属性是否存在也没有 影响),也存在已经配置好的 ipmapping.conf 文件时, client 则会走 ipmapping.conf 的 mapping1 的映射关系 19。根据上面的 ipmapping.conf 的配置, 映射规则 mapping1 是能连接的。

- a. client 端传 ipMapping=mapping1
- b. dcs-site.xml 配置 dcs.default.ip.mapping 无关
- c. 存在配置好的 ipmapping.conf
  - 结果:链接成功

4. 当 client 端的 url 上增加 ipMapping 的属性的 value 不匹配 ipmapping.conf 的关键字是,比如 ipMapping=mapping2 时则也会连接失败。

5. 在 ipmapping.conf 的文件中也可以定义多个映射规则,比如多一个

mapping2,mapping3 等等

6. 目前 ipmapping.conf 的文件中不能用空格行。

7. ipMapping这个属性名是区分大小写的。不可以写为 IPMAPPING 或 ipmapping。

# 附录8. Hadoop 官方安装方法

## 1. CDH 安装准备依赖配置

1. 制作本地yum源

上传所需依赖包或者 Mount iso 文件到本地某个指定目录。

```
vi /etc/yum.repos.d/iso.repo
[iso]
name=iso
baseurl=file:///media/cdrom 【假定/media/cdrom月录】
enable=0
gpgcheck=0
```

然后执行命令: yum clean all 这样就可以本地通过 iso 文件进行安装软件了,示例: yum -y install httpd

2. 安装JDK

1): rpm安装包

rpm -ivh jdk-8u181-linux-x64.rpm

2): 修改系统默认 JDK

```
$ sudo update-alternatives --install /usr/bin/java java
/usr/java/jdk1.8.0_181/bin/java 300
$ sudo update-alternatives --install /usr/bin/javac javac
/usr/java/jdk1.8.0_181/bin/javac 300
$ sudo update-alternatives --config java
$ sudo update-alternatives --config javac
```

3): 检查验证 JDK

```
检测, 输入 java -version
java version "1.8.0_181"
Java(TM) SE Runtime Environment (build 1.80_181b18)
Java HotSpot(TM) 64-Bit Server VM (build 23.6-b04, mixed
mode)
如果是 tar 解压方式安装 JDK, 需要手工修改 PATH 变量。
vi /etc/profile
export JAVA_HOME=/usr/java/jdk1.8.0_181
export PATH=$JAVA_HOME/bin:$PATH
source /etc/profile
```

#### 3. MySQL安装

1) 安装依赖包及 MySQL

yum install libaio\*

备注:安装该包 否则不能正常安装 MySQL

Centos6.x: yum -y install mysql-server service mysqld start chkconfig mysqld on

CentOS7: yum -y install mariadb-server service mariadb start chkconfig mariadb on

2) MySQL root 密码修改

```
mysqladmin -uroot password 'traf123'
或
$ sudo /usr/bin/mysql secure installation
[...]
Enter current password for root (enter for none):
OK, successfully used password, moving on...
[...]
Set root password? [Y/n] y
New password:traf123
Re-enter new password:traf123
Remove anonymous users? [Y/n] Y
[...]
Disallow root login remotely? [Y/n] N
[...]
Remove test database and access to it [Y/n] Y
[...]
Reload privilege tables now? [Y/n] Y
All done!
```

3) MySQL JDBC Driver 安装

```
方法一: yum install mysql-connector-java
方法二: 手工 copy 文件:
下载地址: https://dev.mysql.com/downloads/connector/
拷贝 mysql jdbc 驱动程序 mysql-connector-java-5.1.34.jar 到
/usr/share/java/
ln -s mysql-connector-java-5.1.34.jar /usr/share/java/mysql-
connector-java.jar
```

#### 4) 创建 CDH 数据库

```
mysql -u root --password='traf123' -e 'create database hive
default character set utf8'
mysql -u root --password='traf123' -e "CREATE USER 'hive'@'%'
IDENTIFIED BY 'traf123'"
mysql -u root --password='traf123' -e "GRANT ALL PRIVILEGES
ON hive. * TO 'hive'@'%'"
mysql -u root --password='traf123' -e 'create database amon
default character set utf8'
mysql -u root --password='traf123' -e "create user 'amon'@'%'
identified by 'traf123'"
mysql -u root --password='traf123' -e "grant all privileges
on amon.* to 'amon'@'%'"
mysql -u root --password='traf123' -e "create user 'rman'@'%'
identified by 'traf123'"
mysql -u root --password='traf123' -e 'create database rman
default character set utf8'
mysql -u root --password='traf123' -e "grant all privileges
on rman.* to 'rman'@'%'"
mysql -u root --password='traf123' -e "create user 'cm'@'%'
identified by 'traf123'"
mysql -u root --password='traf123' -e 'create database cm
default character set utf8'
mysql -u root --password='traf123' -e "grant all privileges
on cm.* to 'cm'@'%'"
验证用户登陆:
mysql -u amon -p
Enter password:
ERROR 1045 (28000): Access denied for user 'amon'@'localhost'
(using password: YES)
mysql -u amon -p -h esgyn1
验证可以正常登陆
```

#### 4. 制作httpd安装源

CentOS6x版本 yum -y install httpd /\*安装httpd \*/ service httpd start /\*启动httpd \*/ chkconfig httpd on /\*开机启动httpd\*/

```
cd /var/www/html /*进入html目录*/
ln -s /media/cdrom /*将已经挂载的ISO在该目录做链接,方便其他机器通过
yum源配置访问*/
Centos7用以下方式
Systemctl start httpd.service /*启动*/
systemctl enable httpd.service /*开机启动*/
```

## 5. CDH配置

#### 1) CDH CM 和 Parcel 下载

备注:下载哪个版本因客户现场需求而定。
CentOS 6x
下载CM
http://archive.cloudera.com/cm5/repo-as-
tarball/5.9.3/cm5.9.3-centos6.tar.gz
下载CDH parcels:
http://archive.cloudera.com/cdh5/parcels/5.9.3/CDH-5.9.3-
1.cdh5.9.3.p0.4-el6.parcel
http://archive.cloudera.com/cdh5/parcels/5.9.3/CDH-5.9.3-
1.cdh5.9.3.p0.4-el6.parcel.sha
http://archive.cloudera.com/cdh5/parcels/5.9.3/manifest.json
CenOS 7x
下载CM
http://archive.cloudera.com/cm5/repo-as-
tarball/5.9.3/cm5.9.3-centos7.tar.gz
下载CDH parcels:
http://archive.cloudera.com/cdh5/parcels/5.9.3/CDH-5.9.3-
1.cdh5.9.3.p0.4-el7.parcel
http://archive.cloudera.com/cdh5/parcels/5.9.3/CDH-5.9.3-
1.cdh5.9.3.p0.4-el7.parcel.sha
http://archive.cloudera.com/cdh5/parcels/5.9.3/manifest.json

2) 配置 CDH 资源库

1、解压、拷贝安装包到目录
tar -xvf cm5.9.3-centos7.tar.gz -C /var/www/html
mkdir /var/www/html/cdh5.9
cp CDH-5.9.3-1.cdh5.9.3.p0.7-el7.parcel /var/www/html/cdh5.9
cp manifest.json /var/www/html/cdh5.9
2、给两个目录添加权限
chmod -R ugo+rX /var/www/html/cm
chmod -R ugo+rX /var/www/html/cdh5.9

3) 配置新的CDH安装yum源

```
vi /etc/yum.repos.d/cdh.repo
[cdh]
name=cdh
baseurl=http://IP/cm/5
enabled=true
gpgcheck=false
yum clean all /*手工清除*/
```

4) 验证 CDH 是否成功读取



Ĩ	Parent Directory	-
2	CDH-5.9.3-1.cdh5.9.3>	2018-12-20 01:23 1.4G
?	manifest.json	2018-12-20 01:32 63K

如果出现没法读取 cdh5.9 目前的情况时候,可以将文件直接拷贝到主机的 /opt/cloudera/parcel-repo 目录下面,使用本地安装即可 (centos7 未出现该情况, centos6 会出现这种情况) 5): 验证 cm 是否成功读取

(i) 10.10.22	2.54/cm/5/			
质-依赖包 🖨 cdh/5.	9.3/ 🚯 Esgyn Corporat	ion	🚺 易鲸捷知识库 笔记本	*
Index of	/cm/5			
Name	Last modified	<u>Size</u> <u>I</u>	<u>Description</u>	
Parent Direct	ory	-		
RPMS/	2017-06-27 23:02	-		
<b>mirrors</b>	2017-07-13 09:12	75		

# 2. Hadoop 集群角色规划

• 集群规划信息:【11个节点的 hdsf,hbase,hive, Zookeepe 等角色规划】

入 主	hado										
人机	op-										
组	hdt0	hdt1	hdt1								
件	1	2	3	4	5	6	7	8	9	0	1
HDFS	NN	NN									
NN/SN											
Ν											
HDFS	Y	Y	Y								
Journal											
Node											
HDFS				Y	Y	Y	Y	Y	Y	Y	Y
DN											
Hbase	Y	Y									
Master											

Hbase				Y	Y	Y	Y	Y	Y	Y	Y
RS											
Zooke	Y	Y	Y								
eper											
Yarn	Y	Y									
Master											
Yarn				Y	Y	Y	Y	Y	Y	Y	Y
NM											
Hive	Y	Y									
Metast											
ore											
Hive2	Y	Y		Y	Y	Y	Y	Y	Y	Y	Y
Server											

1. HBASE角色配置规划

Hbase 角色组分配:

- 1) Master Default Group 2
- 2) RegionServer Default Group 1+N【节点数】

已选定的	的操作▼ 添加角色实例	前角色组			
<b>—</b> ÷	角色类型	状态 🕴	主机	♦ 授权状态	♦ 角色组
	Master (备份)	已启动	esgzb-del-n005.esgyn.com	已授权	Master Default Group
	Master (活动)	已启动	esgzb-del-n004.esgyn.com	已授权	Master Default Group
	RegionServer	已启动	esgzb-del-n005.esgyn.com	已授权	RegionServer Default Group
	RegionServer	已启动	esgzb-del-n007.esgyn.com	已授权	RegionServer Default Group
	RegionServer	已启动	esgzb-del-n006.esgyn.com	已授权	RegionServer Default Group

2. HDSF角色配置规划

HDSF 角色分配:

- 1) Balancer Default Group 1
- 2) DataNode Default Group  $1 \rightarrow N$

- 3) NameNode Default Group 1
- 4) SecondaryNameNode Default Group 1

已选定的操作	乍▼ 迁移角色 添加角色:	实例 角色	组		
□ + 角€	色类型	状态  ≑	主机    ≑	授权状态	角色组
🔲 🌒 Bal	lancer	不适用	esgyn1.local.cn	已授权	Balancer Default Group
🔲 🔴 Dat	taNode	已启动	esgyn1.local.cn	已授权	DataNode Default Group
🔲 🌒 Nai	meNode (活动)	已启动	esgyn1.local.cn	已授权	NameNode Default Group
🔲 🔴 Sec	condaryNameNode	已启动	esgyn1.local.cn	已授权	SecondaryNameNode Default Group

3. HIVE角色配置规划

Hive 角色分配:

- 1) Gateway Default Group 1
- 2) Hive Metastore Server Default Group 3-5
- 3) HiveServer2 Default Group 1—2

已选定的	已选定的操作 ▼ 添加角色实例 角色组								
÷	角色类型	状态  ∜	主机	♦ 授权状态	♦ 角色组				
	Gateway	不适用	esgzb-del-n005.esgyn.com	已授权	Gateway Default Group				
	Gateway	不适用	esgzb-del-n007.esgyn.com	已授权	Gateway Default Group				
	Gateway	不适用	esgzb-del-n006.esgyn.com	已授权	Gateway Default Group				
	Gateway	不适用	esgzb-del-n004.esgyn.com	已授权	Gateway Default Group				
	Hive Metastore Server	已停止	esgzb-del-n005.esgyn.com	已授权	Hive Metastore Server Default Group				
	Hive Metastore Server	已停止	esgzb-del-n006.esgyn.com	已授权	Hive Metastore Server Default Group				
•	Hive Metastore Server	已停止	esgzb-del-n004.esgyn.com	已授权	Hive Metastore Server Default Group				
	HiveServer2	已停止	esgzb-del-n005.esgyn.com	已授权	HiveServer2 Default Group				
•	HiveServer2	已停止	esgzb-del-n006.esgyn.com	已授权	HiveServer2 Default Group				
	HiveServer2	已停止	esgzb-del-n004.esgyn.com	已授权	HiveServer2 Default Group				

### 4. YARN角色配置规划

YARN 角色分配:

- 1. JobHistory Server Default Group 1
- 2. NodeManager Default Group 3-N
- 3. ResourceManager Default Group 1

已选定	的操作▼ 添加角色实例	角色组			
\$	角色类型	状态  ∜	主机	◆ 授权状态 ◆	<b>角色组</b>
	JobHistory Server	已停止	esgzb-del-n004.esgyn.com	已授权	JobHistory Server Default Group
	NodeManager	已停止	esgzb-del-n005.esgyn.com	已授权	NodeManager Default Group
	NodeManager	已停止	esgzb-del-n007.esgyn.com	已授权	NodeManager Default Group
	NodeManager	已停止	esgzb-del-n006.esgyn.com	已授权	NodeManager Default Group
	ResourceManager	已停止	esgzb-del-n004.esgyn.com	已授权	ResourceManager Default Group

5. ZooKeeper角色配置规划

ZooKeeper 角色分配:

```
Server Default Group 3 【默认为奇数, 最小 3 个节点】
```

已选定的操作▼	添加角色实例	角色组			
□ 🕴 角色类	型 🕈 状态	\$	主机	授权状态	角色组
🗌 🔴 Server	已启	司动	esgzb-del-n005.esgyn.com	已授权	Server Default Group
Server	已启	动	esgzb-del-n007.esgyn.com	已授权	Server Default Group
Server	已启	司动	esgzb-del-n006.esgyn.com	已授权	Server Default Group

## 3. Hadoop 集群安装配置

1. 安装Cloudera管理器服务器

```
yum -y install cloudera-manager-daemons cloudera-manager-
server
```

#### 2. 为Cloudera管理器配置外部数据库

```
初始化:
/usr/share/cmf/schema/scm_prepare_database.sh -h esgzb-
n001.esgyn.com mysql cm cm traf123
注释:
scm_prepare_database.sh -h esgyn1--机器名或 IP mysql---数据库类型 cm--数据库 cm--数据库用户名 traf123---数据库密
```

3. 启动Cloudera管理器服务器

```
service cloudera-scm-server start
查看 Cloudera 管理器服务是否启动成功:
```

```
service cloudera-scm-server status
```

```
安装详细日志: /var/log/cloudera-scm-server/cloudera-scm-
server.log /*用于定位问题*/
```

4. 安装CDH 节点

登陆 CDH CM, 进行安装配置: <u>http://IP:7180/</u>

1) 为群集制定主机

## 为 CDH 群集安装指定主机。

应使用主机用于标识目身的同一主机名称 (FQDN) 来指定主机。 Cloudera 建议包括 Cloudera Manager Server 的主机。这样还将对该主机进行运行状况监控。 提示: 使用 <u>模式</u> の搜索主机名称和/或 IP 地址。	
esgyn[1-3]	
	SSH 鎮□: 22 Q 搜索

## 2)指定要安装的主机

## 为 CDH 群集安装指定主机。

应 C 摄	t使用主机用于标识目身 loudera建议包括 Clou 际: 使用模式 e 複素主	的同一主机名称 (FQDN) 来撤定主机 dera Manager Server 約主机。这样 机名称和/或 IP 地址。	1。 还将对读主机进行运行;	<b>庆</b> 况监控。	
			已扫描3个主机,其4	₱3个正在运行	SSH。 Q 新提索
	已扩展查询	主机名称 (FQDN)	IP teta	当前受管	结果
	esgyn1	esgyn1.cn	192 168.1.2	杏	✔ 主机准备就请:2 笔砂峭应时间。
•	esgyn2	esgyn2.cn	192 168 1.3	否	✓ 主机准备就着:3 笔秒峭应时间。
	esgyn3	esgyn3.cn	192 168 1.4	否	✓ 主机准备就清:4 電砂崎应时间。

3) 远程 Parcel 存储库 URL: 指定上面配置的 CDH 或者远程资源库

Parcel 目录 /opt/clow ● 需要單合理 本地 Parcel 存储库器径 /opt/clow 运程 Parcel 存储库 URI	udera/parcels
本地 Parcel 存储库留径 /opt/clos	uders/instriel_repo
运程 Parcel 存储库 URI	uno approven oper
http://1	0.10.22.54/cdh5.9/

#### 4) 填写 Cloudera 的地址

#### 群集安装

Cloudera 建议使用 parcel 来代善软件包进行安装,因为 parcel 可以使服务二进制文件的部署和升级自动化,让 Cloudera Manager 经松地管理群集上的软件。如果选择不使用 parcel,当有软件更新可用时,将需要包干动升级群集中所有主机上的包,并会阻止忽使用 Cloudera Manager 的滚动升级功能。 	选择存储库	
満評方法 ○ 使用数据包 ●   ● 使用 Parcei (建议) ● 更多協項 代理设置   选择 CDH 的版本   ● CDH-5.9.3-1.cdn5.9.3.p0.4	Cloudera 建议使用 parcel 新 parcel,当有软件更新可用的	R代菁软件包进行安装,因为 parcel 可以使服务二进制文件的部署和升级启动化,让 Cloudera Manager 经松培管理期集上的软件。如果选择不使用 f,将需要您于动开级群集中所有主机上的包,并会租止您使用 Cloudera Manager 的滚动开级功能
选择 CDH 的版本 ③ CDH-5.9.3-1.cdh5.9.3.p0.4	选择方法	○ 使用数据包 ● ● 使用 Parcel (建议) ● 亜多脂類 代理设置
对于此 Cloudera Manager 版本 (5.9.3) 太勤的 CDH 版本不会显示。	选择 CDH 的版本	● CDH-5.9.3-1.cdh5.9.3.p0.4 划于此 Cloudera Manager 版本 (5.9.3) 太新的 CDH 版本不会显示。
选择想要安装在主机上的 Cloudera Manager Agent 特定发行版。	选择您要安装在主机上的 Ci	oudera Manager Agent 特定拨行版。
<ul> <li></li></ul>		<ul> <li>○ 此 Cloudera Manager Server 的匹配发行版</li> <li>● 自由文育構成</li> <li>● 自由文育構成</li> <li>● 内田文育構成</li> <li>● ND 210.10.22.54/cm/5/</li> <li>以 SLES、Redhat 成員他 RPM 为基础的分布示例:</li> <li>● https://archive.cloudera.com/cm5/redhat/6/x86_64/cm/5/</li> <li>以 Ubuntu 或其他 Debian 为基础的分布示例:</li> <li>● bttps://archive.cloudera.com/cm5/whentu/lucid/eedP4/cm/_lucid-cm5_contrib.</li> </ul>

## 5) 完成安装

### 不要勾选单一用户模式,否则会遇到很多权限问题。

#### 群集安装

N

- P.O					
	**	-	-	1++ _ AL	
EH	ш	н			
	æ	-	-	100-2-6	

#### 仅受 CDH 5.2 及更高版本支持。

默认情况下,服务进程以不同用户身份在系统中运行。例如,HDFS DataNode 以用户"hdfs"的身份运行,HEase RegionServer 以用户"hbase"的 身份运行。启用"单个用户模式"将 Cloudera Manager 配置为以单个用户的身份运行服务进程,默认为"cloudera-scm",从而使受管服务与系统其 他服务之间的分离优先于受管服务之间的分离。

此选项的主要益处是 Agenr 不作为根运行。但是,此模式将使安装变得复杂,这将在<u>文档</u>中作完整介绍。最明显的是,在常规模式下由 Agent 自 动创建的目录必须通过相应权限在每个主机上手动创建,并且必须为配置的用户设置 sudo(或同等)访问权限。

在单个用户模式与常规模式之间来回切换不受支持。

单用户模式

0			
0			
0			
0			
0			
Θ	 _	_	
Θ			
υ			
9			
•••			
· •			
~		-	

输入 root 的密码,或者具有 sudo 权限用户的密码

#### 提供 SSH 登录凭据。

安装 Cloudera 包需要有主机 root 用户的无密码 sudo/pb	机的 root 访问权限。此安装程序将通过 SSH 连接到您的主机,然后直接以 root 用户身份登录,或者以另一个具有支为 run 权限的用户身份登录。
登录到所有主机,作为:	€ root
对以上选定的用户,您可通	过恶弱或公明身份验证重接。
身份验证方法	④ 所有主机接受相同密码
	○ 所有主机接受相同私期
输入密码	******
備认应的	
SSH MACT	22
同时安装的数量	10 (同时运行多个安装时将托费大量的网络带宽和其他其他系统资源)

#### 群集安装

and the second second				
主体的例	17 1812	送来	(te	
esg25-n001 esgys.com	11.90.22.54	8	日常功能成果例。	7-80.0
esigt0-n002 esigner com	10.10.22.60		已成功完成安装。	Pese

#### 群集安装

正在安装选定 Parcel					
協能的 Parcel 正在下級升份的在群論的時代	LaRLE.				
♥ 0DH 5.9.3-1.cdt5.9.3 p0.4	已下载 100%	已分配 212 (07.2 mB/s)	EME 22	已務派 22	

## 自定义安装 Zookeeper, HDFS, Hbase, Yarn, Hive 服务 选择您要在群集上安装的 CDH 5 服务。

										1.1				
退	择要安于	装的服务组合。												
0	核心 H HDFS,	<b>ladoop</b> 、YARN(包括 Ma	pReduce 2),	ZooKeeper,	Oozie,	Hive,	Hue ¥	E Sqoop						
0	合 HB HDFS	<b>ase 的内核</b> 、YARN(包括 Ma	pReduce 2),	ZooKeeper,	Oozie,	Hive,	Hue,	Sqoop	RD HBase					
0	A Imp HDFS,	<b>ala 的内核</b> 、YARN(包括 Ma	pReduce 2)	ZooKeeper,	Oozie,	Hive,	Hue,	Sqoop	€] Impala					1.4
0	A Sea	arch 的内核 , YARN (包括 Ma	pReduce 2),	ZooKeeper,	Oozie,	Hive,	Hue,	Sqoop i	和 Solr					
0	A Spa HDFS,	ark 的内核 、YARN(包括 Ma	pReduce 2),	ZooKeeper,	Oozie,	Hive,	Hue,	Sqoop	10 Spark					
0	所有服 HDFS,	<del>135</del> 、YARN(包把 Ma	pReduce 2)	Вкеерег,	Oozie,	Hive,	Hue,	Sqoop,	HBase,	impala,	Solr.	Spark #OBI	/值 Store Ind	lexer
۲	自定义 选择您	<b>服务</b> 自己的服务。将自	同物自然活动	需要的服务。	只有在	设置了	初始郡	慎之后?	才能添加!	Flume,				
			HDFS )											
-	+ 70	ookeeper	Anache	ZooKeeper §	用于油	护印度	北部市		重中影务,					

## 选中自定义服务

遗	<sup>梁悠</sup> 目己的服务。将高动き	1余乐选服务需要的服务,只有在设置了初始群集之后才能添加 Fume,
	服务关型	除期
Ð	H HBase	Apache HBase 提供对大型数据集的模机。实时的误写访问权限(需要 HDFB 和 Zookeeper)。
2	HDPS	Apache Hadoop 分布式交件系统(HDFS)量 Hadoop 应用程序使用的主要存储系统。HDFS 创建多个数据块副本并将它们分布在整个群集的计算主机上,以应用可靠且依据实施的计算句呢。
Ø	😧 Hive	Hive 是一种都蛋白片系统,提供名为 HiveQL 的 SQL 美面言。
	🖶 Hue	Hue 是与包括 Apache Hadoop 的 Cloudera Distribution 配合视频的塑形用户界面(需要 HDFS、MapReduce 和 Hive)。
	9 Impala	Impaila 为存储在 HDFS 和 Hbase 中的教授提供了一个实时 SQL 詹纳姆囗。 Impaila 需要 Hive 服务,并与 Hue 共享 Hive Metastore。
	🕼 Islion	EMC islion is a distributed filesystem.
	Kafka	Apache Kafka is publish-subscribe messaging rethought as a distributed commit log. Before adding this service, ensure that either the Kafka parcel is activated or the Kafka package is installed.
	Rey-Value Store Indexer	離論 Store Indexer (3時 HSase 中裕含物內的設置做化。并使用 Sor 为此的融放的1。
	NapReduce	Apache Hadoop MapReduce 支持对整个群集中的大型数据集进行分布式计算(需要 FDFS)。建论或用 VARN(包括 MapReduce 2)、包括 MapReduce 用于和后来容性。
	Oozie	Coze 是群集中管理数据处理作业的工作院协调服务。
	🔑 Sair	Solir 是一个分布式服务,用于调制存储在 HDFS 中的数据的类引并模型这些数据。
	🔯 Spark	Apache Spark is an open source cluster computing system. This service runs Spark as an application on YARM.
	👼 Sqoop 2	Sqoop 是一个设计用于在 Apache Hadoop 和四特代政策符種(如关系政策库)之间竟然地传输大规量取强的工具。Cloudera Manager 支持的版本 为 Sqoop 2。
	YARN (MR2 Included)	Apache Hadoop MapReduce 2.0 (NRv2) 或 YARN 是也将 MapReduce 应用程序的股级计算程序(需要 HDFS)。
Ø	200Keeper	Apache Zookkeper 显用于维护和同步配置数据的集中服务。
۶D		

## zookeeper 选择奇数个节点, 推荐 3 个。

## 自定义角色分配

Cloudera Management Servic	ce		
Service Monitor + 1 Milit	Activity Monitor = 1 mit	Host Monitor = 1 Milit	Reports Manager + 1 Mill
esgynt.cn	esgynt on •	esgynt cn	esgyn1.cn
Event Server = 1 Bill	Alert Publisher * 1 新聞		
esgyn1 cn	esgyn1.cn		
ZooKeeper			

数据库设置【选择之前安装的 mysql 或则系统自带数据库】

#### 数据库设置

Activity Monitor					✓ Success!
当前被分配在 esgyn1.sn 上运行, 数据库主机名称:	数据库关型		教服体名称:"	用户名: "	<b>坡</b> 段
esgyn1 on	MySQL	٣	arron	amon	Esgyn618
Reports Manager					✓ Successf
当前地分配在 esgyn1.cn 上运行。 数据库主机名称: <sup>-</sup>	教服有关型:		数据本名称:*	用户名:*	未驻
esgyn1.cn	MySQL	-	man	man	Esgyn618
					記錄示者

## 添加 HDFS、HBase、Hive、Yarn Master、Zookeeper 等服务

taster + 1998	HISalet REST Safett	THE HEASE THEIR Server	RegionServer * 1 mill
asgen-millit asgyn enm	8923.	3824	H Database KIR +
O HOFS			
INANOTODO + 1 ME	socondaytemoteda × 1 #88	Hatercer + 1 Milt	EEE Amperis
esgab-n001 esgyn zom	esgab-ri002 esgyn.com •	esgan-a001 esgyn com	<b>法</b> 国家和
ITT NF8 Gateway	Datahice + 2 Mila		
选择生化	esgati-n(001-002) esgyn con *		
- Hue			
Carrier + 2 Will	mive Metastore Server + 1 Milt	WEH-Cal Server	Hveitetver2 = 1 Mill
esgan-a (001-002), esgyn com	esgab-n001 esgan.com	高祥士和	esgtb-n001 esgyn com
Couders Management Service			
🔜 Service Monitor - 1 新聞	Activity Monitor	Hold Monitor + 1 Britt	Event Server + 1888
eogen-noon esgyn com	AI#1-1#1	esges-a001 esgyn com	esgib-n001.esgyn.com
Avent Publisher + 1 818			
esgio-into'r esgyn com			
# YARN (MR2 included)			
ResourceManager + 1 MM	and interfering Server + 1 Milt	Nodelikanager + 2802	
eegan-n001 eegyn com	esgib-n001 segyn com	Sp Dostatilocis 4060 +	
2 ZooKeeper			
5erver = 2.852			

talan i		8 <mark>8</mark> 8888		in the second se
自定义角色分配				
经可有此处理会交易管理的承兑分配。 假如常	RS配不正确(例如,S配加某个主机上的角点)	(多) 全影成服务性能、除全型有特殊需求,1	81日为将金角色预先通停将金工机,否则 Cloudera	
不遵迫攻支分配库见。	-			
还可以按主机直要角色分配。 医土和血液				
H HBase				
Master × 1 2012	HBase REST Server	HBase Thrift Server	RegionServer + 2 #F#	
engzb-s001.engym.com	地缘主机	造释主机	Jy Dataliode (10) *	
O HDFS				
Marweliode × 1 Milt	🐜 SecondaryfiameNoda × 1 108	Dalancer + 1 Wilk	HT HEPTS	
esgzb-x001.esgym.com	esgab-n002.esgyn.com +	esgab-n001.esgyn.com	选择主机	
MF3 Gateway	DataNode × 2 #98			
选择主机	esgab-nj001-002).esgyn.com =			
P HW0				
Galeway + 2 mile	Hive Metastore Server × 1 #Eliz	WebHCat Server	HveServer2 = 1 Friz	
entre don ovel estàs con	engennen augusten	ALCONT PO	estimate estist con	
Cicudera Management Service				
Service Montor × 1 新建	Activity Monitor	Host Manitor × 1 👬	an event server × 1 thit	
esgzb-e001 esgan.com	道祥—个主机	esgab-n001.esgyn.com	esgzb-m001 esgyn.com	
Alert Publisher × 1818				
esgzb-#001.esgyn.com				
样集安装				
ANTE CONTRACTOR				
<u>w</u> ur.				
✓ 检查端在所有 2 个王机上运行。	-			
✓ 下班主机上時間解除了自己的主机构 本に現在は各部の時間をはたいです。	17. 11			
✓ 並用好任/F用的設備算手段不及影響 ▲ 給養 (my/set) 計畫的問題	n,			
<ul> <li>Matting for the state of the state</li> <li>新教士記述版 [praired matter state</li> </ul>	101.			
<ul> <li>检查过的所有主机均正确只及时影响;</li> </ul>	析了彼此的主机实际。			
✔ 主机时种几字两步(10分钟布)。				
✔ 整个群集中的主机时区一致。				
✓ 尤用中感相缺失。				
✓ 软件包和 parcel 之同未检测别冲突。				
<ul> <li>没有存在已知論說的內核版本在因行。</li> </ul>				
Claudera Eliciti /procisys/www.swapp 80 / Claudera Managera 2000/2007	piness 記書為離大値 10、当前記録为 30、( Linux File)の正式の構成できたがある。	部用 syscel 命令在協行社業改成資源 CELETATION	#倫理 /etc/syscil.conf. 以作量向気体な	
<ul> <li>A 日本部編集主要工作を目的になる。</li> </ul>	n and a feature of the second state of the se	enalitentrasenati heraran bidar	Werks near > hus/kanal/modewarausi >	a na na na kao ina mana ina mana ina mana ina mana kao amin' na man
<ul> <li>E. 然后均同一会令添加到 Ascircle</li> </ul>	cal等机能化器中中, 以使在系统重点分子	の旧香、以下土机将受到影响: >	en even regel « rokong region in a obsidar".	elebolis connectional
<ul> <li>已満足 CDH 6 Hate Python 版本依赖</li> </ul>	关系。			
✓ ◎ 魚主規正在运行 CDH 4, 2 角主机	正在运行 CDH 5。			
▲ 系统之间存在不匹配的版本,这将导	微失败,有关在哪个主机上在运行哪个版本	的祖忤乾绅慎,清参见下离的绅母俱息。		
✓ 所有托管的主机管则有不一致的 Java	四本.			
メ 所設高的所有 Cicedera Management	t Deemon 關本這般的書一面。			
✓ 前检查的所有 Cloadwa 管理代理版法	に与服何最一致。			

# ▲注意

此处为告警,尽量做到没有告警提示,否则可以忽略

#### 群集设置

✔ 首次运行 命令			
RE: 已完成 开始的标记 12月 18, 4, 16.24 波具 环绕的时间: 3.9m			
Finished First Run of the following services successfully: ZooGeeper, H	DPS, HEase, YARN (HR2 Included), H	Sve, Clouders Ranagement Servic	π.
洋细信息   回告#7个#夏(洪7个),		01	1章 ○見会大敗 ○夜開返行式台
PH .	上下文	FIGNOR	HILING SH
> プ 并行运行10項 已成初先成1个分裂。		12月 10, 4:10:24 液調	65ms
> ✓ 正在部署由户通転置 Successfally deployed all client configurations.	e Cluster 1 e	12月 10, 4 位 24 波昌	15.61e
> ✓ 創語 Clouders Management Service, ZooKeeper 日記記的記述 2 个话题。		12月 18, 4, 16:39 道圖	26.76s
> ✓ Anni HOrs 已知知何知 1 个世界。		12月 18, 4:17:16 波興	43.30s
2 ✔ 品版10000 日間以完成 1 个标准。		12月 10, 4:17:50 激調	36.628
> ✓ 治动 YARN (MR2 Included) 已成功必成 1 个分圈。		12月 18, 4:19:27 遺農	38.425
> ✓ 麻油How Edutated i 个分型。		12月 18, 4 19:16 漢農	69.32s

1600	1052

### 查看界面展现效果

cloudera MANAGER Ins J	us. ee.				🚔 65 💷 339 - 4
主页					今天,建築1点2
ALE ANALOUSIER () NOR CO - A	CARNOO				5 a
					MRI Clouders Enterprise Data Hab Edition 6
<ul> <li>Cluster 1 (CDH 5.9.3, Parcel)</li> </ul>		图表			38.5996 1 산태 2 산태 8 산태 12 산태 1天 74 30년
• <b>H</b> 25.		ENE CPU	29480 IO	TISHON IO	HDF5 ID
H HDate	-	1005.	7 999	2 cm	
B HDF2	-	10			
· ····	-	<u>1</u> 10%			1 mil
III TARM (MR2 In		21.00	A 101-144 102+5	20 AL 101.414 101.415	
2 Zookeeper 26 Z		-CLOSE 1.007 235 000 245 CPU 0700 2476	-5 dia +010	-5 R840 +	-Boston 1304 -Bostone+ the
Cloudera Management Service					
😆 🖬 Clauders Mana	-				

## 5. 安装后检查

- 1、缺省检查
  - 1)Parcels目录是否是缺省设置: /opt/cloudera/parcels/
  - 2)Hbase shell 检查 scan 一个表,检查 HBase 是否正常
  - 3)Hbase RegionServer 是否在同一个 RegionServer Default Group

# 附录9. Preload 功能使用说明

## 1. Preload 功能简介

Preload 对数据库中的表的 meta 信息进行预加载,以减少 sql 语句的第一次编译 时间。具体是指,QianBase 启动时(也可以手动加载),将所有表(需生成 descriptors)的 meta 信息加载至 shared cache(共享内存)中。当 sql 进行第一 次编译时,先从 shared cache 中拷贝 meta 信息至 mxosrvr 本地缓存中,再进行 语句编译。

如果没有 preload 功能,第一次编译时则先需要从磁盘上的 meta 表中读取 meta 信息,再拷贝至本地缓存中,这降低了第一次编译的效率。使用 preload 功能 后, sql 语句第一次编译时间由 3~5 秒,降为 0.5~2 秒。

## 2. Preload 使用步骤

1) 增加 cqd "TRAF ENABLE METADATA LOAD IN CACHE"

upsert into "\_MD\_".defaults values
('TRAF\_ENABLE\_METADATA\_LOAD\_IN\_CACHE','ON', 'enable
preload', 0);

#### 2) 增加 cqd "TRAF ENABLE METADATA LOAD IN SHARED CACHE"

upsert into "\_MD\_".defaults values
('TRAF\_ENABLE\_METADATA\_LOAD\_IN\_SHARED\_CACHE','ON', 'enable
shared cache', 0)

3) 给表增加 desc 属性

可以通过以下三种方式给表添加 desc 属性:

方式1) 创建表时添加 desc 属性。

create table tb1 (a int) attribute stored desc;

方式2) 创建好表后(在所有 DDL 语句后), 添加 desc 属性。

create table tb1 (a int); alter table tb1 generate stored desc;

方式3) 创建 schema 时添加 desc 属性,则 schema 下的所有表都具有 desc 属性。此方式不用对单独的表添加 desc 属性。

create schema sch1 stored desc

4) Preload 生效

有两种生效方式:

方式1) 重启 QianBase,执行 sqstop 然后执行 sqstart。

可通过 sql 日志 trafodion.sql\_#.log 查看,

2019-07-18 03:00:32,760, INFO, SQL.EXE, Node Number: 0, CPU: 0, PIN: 2638, Process Name: \$200000002XF,, Metadata loaded into the shared cache for 42 tables, taking 14831 us

方式2)使用 trafodion 用户登录,在终端执行"load\_shared\_cache verbose"

[trafodion@nap039 ~]\$ load\_shared\_cache verbose EsgynDB is available, continuing ... Script running on node: 0 Loaded Trafodion Metadata into Shared Cache: Total found=42, size in bytes: total=376375, avg=8961

5) 检查是否生效

使用 trafodion 用户登录, 在终端执行以下命令:

"edb pdsh -a tdm arkcmp -testSharedCacheFindAll"。

该命令输出所有被加载的 shared cahce 表信息,按照实现,每个节点都会有一份拷贝,所以每个节点的信息都是一样的。如果表存在输出信息当中且存在于每个节点,则说明 preload 成功。反之,则 preload 失败。

例如:

下图表示集群各个节点 preload 成功了 43 张表。

[trafod	ion@nap039	esgyndb]	\$ edl	o_pdsh	-a	tdm_a	rkcmp -	testS	nared	CacheF:	indAll	grep	'Find	All()'				
nap040:	FindAll()	finding	all	pairs	of	(key,	value)	from	hash	table	takes	594us.	Total	found=43,	size	in bytes:	total=379533,	avg=8826.35
nap042:	FindAll()	finding	all	pairs	of	(key,	value)	from	hash	table	takes	585us.	Total	found=43,	size	in bytes:	total=379533,	avg=8826.35
nap039:	FindAll()	finding	all	pairs	of	(key,	value)	from	hash	table	takes	628us.	Total	found=43,	size	in bytes:	total=379533,	avg=8826.35
nap041:	FindAll()	: finding	all	pairs	of	(key,	value)	from	hash	table	takes	637us.	Total	found=43,	size	in bytes:	total=379533,	avg=8826.35

下图表示 table USER\_TEST1 成功 preload 到该集群的所有节点上。

[trafodion@nap039 ~]\$ edb\_pdsh -a tdm\_arkcmp -testSharedCacheFindAll | grep USER\_TEST1; nap039: key=TRAF0DION.SEABASE.USER\_TEST1 nap042: key=TRAF0DION.SEABASE.USER\_TEST1 nap041: key=TRAF0DION.SEABASE.USER\_TEST1 nap040: key=TRAF0DION.SEABASE.USER\_TEST1

## 3. 附加信息

1. 怎么删除表的desc属性?

使用以下SQL命令即可删除表的desc属性。

alter table tb1 delete stored desc

也可以删除整个 schema 的 desc 属性。

alter schema sch1 delete stored desc;

2. 怎么关闭Preload功能?

删除两个 cqd, 再重启 QianBase 即可。

delete from "\_MD\_".defaults where ATTRIBUTE='TRAF\_ENABLE\_META DATA LOAD IN CACHE'

delete from"\_MD\_".defaults where ATTRIBUTE='TRAF\_ENABLE\_METAD ATA LOAD IN SHARED CACHE'

- 3. 怎么使用老机制的Preload功能?
- a. 只使用一个 cqd "TRAF\_ENABLE\_METADATA\_LOAD\_IN\_CACHE",如果 另外一个 cqd 被设置了,则需要删除掉。
- b. 给表增加 desc 属性。
- c. restart dcs 或者 restart QianBase。
- d. 使用如下 sql 查询表是否加载至 cache 中。

select catalog\_name, schema\_name, object\_name from table(nata blecacheentries('all','local')) order by 1,2,3;

# 附录10.端口列表

# 1. QianBase端口列表

产品	组件	端口号	是否向 外部网	是否向内部网 络 QianBase	说明
			络开放	节点间开放	
QianBase	DCS Master Listen	23400	Yes	Yes	
QianBase	DCS Master UI	24400	Yes	Yes	
QianBase	mxsosrvr	23401 : 23499	Yes	Yes	基于并发要 求和最大配 置 mxosrvr 数的端口范 围
QianBase	REST Server	4200/4201	Yes	Yes	
QianBase	DB Manager	4205/4206	No	Yes	
QianBase	Esgyn exporters	23300/23301	No	Yes	
QianBase	Filebeat	5044	No	Yes	
QianBase	mds	8989	No	Yes	
QianBase	Monitor (MONITOR_COMM_PORT)	23399	No	Yes	
QianBase	Monitor (MONITOR_SYNC_PORT)	23380	No	Yes	
QianBase	Monitor (NS_COMM_PORT)	23370	No	Yes	
QianBase	Monitor (NS_M2N_COMM_PORT)	23360	No	Yes	
QianBase	Monitor (NS_M2N_COMM_PORT)	23350	No	Yes	
QianBase	Monitor (MON2MON_COMM_PORT)	23340	No	Yes	
OM HA	Offline DBMgr	30005(http)/30006(https)	Yes	No	部署在管理

					节点上
OM HA	ElasticSearch	<b>2</b> 000 <b>2</b>	N	N/	部署在管理
		30002	NO	Yes	节点上
OM HA	logstash	20002	No	Vac	部署在管理
		50005	NO	105	节点上

## 2. Hadoop端口列表

			-			
组件	服务	说明	端口号	访问	配置	注释
	DataNode		50010	External	dfs.datanode.address	DataNode HTTP
						server port
	DataNode	Secure	1004	External	dfs.datanode.address	
	DataNode		50075	External	dfs.datanode.http.addres	
					8	
	DataNode		50475	External	dfs.datanode.https.addre	
					<b>SS</b>	
	DataNode	Secure	1006	External	dfs.datanode.http.addres	
					S	
	DataNode		50020	External	dfs.datanode.ipc.address	
Hadoop	NameNode		8020	External	fs.default.name 或	fs.default.name
HDFS					fs.defaultFS	
	NameNode		8022	External	dfs.namenode.	HDFS daemons 程序
					servicerpc-address	使用的可选端口,以
						避免共享客户端使用
						的 RPC 端ロ
						(8020) ° Cloudera
						建议使用端口
						8022。
	NameNode		50070	External	dfs.http.address 或	dfs.http.address

HBase

				dfs.namenode.http-	
				address	
Secondary		50090	Internal	dfs.secondary.http.addre	dfs.secondary.http.add
NameNode				SS	ress
Secondary	Secure	50495	Internal	dfs.secondary.https.addr	
NameNode				ess	
JournalNode		8485	Internal	dfs.namenode.	
				shared.edits.dir	
JournalNode		8480	Internal	dfs.journalnode.http-	
				address	
JournalNode		8481	Internal	dfs.journalnode.https-	
				address	
Failover		8019	Internal		用于 NameNode HA
Controller					
NFS		2049	External		Nfs 端ロ
gateway					(nfs3.server.port)
NFS		4242	External		mountd 端口
gateway					(nfs3.mountd.port)
NFS		111	External		portmapper 或 rpcbind
NFS		50079	External	nfs.http.port	CDH 5.4.0 及更高版
gateway					本。NFS 网关
					daemon 程序使用此
					端口提供指标服条
					而二极八祖孙派力。
					以端口任 5.10 及文问
NES	Sacura	50570	Extornal	nfa https port	成本上了癿且。 CDU510刀五言比
gateway	Secure	50579	External	ms.mups.port	CDH 5.4.0 及更尚版
gaue in al					本。NFS 网天
					daemon 程序使用此
					端口提供指标服务。
					该端口在 5.10 及更高
					版本上可配置。
HttpFS		14000	External		
HttpFS		14001	External		
Master		60000	External	hbase.master. port	IPC
Master		60010	External	hbase.master. info.port	НТТР
RegionServe		60020	External	hbase. regionserver. port	IPC
r			1	1	

	RegionServe r		60030	External	hbase. regionserver. info.port	НТТР
	HQuorumPe		2181	Internal	hbase. zookeeper.	HBase 管理的
	er				property. clientPort	ZooKeeper 模式
	HQuorumPe		2888	Internal	hbase. zookeeper.	HBase 管理的
	er				peerport	ZooKeeper 模式
	HQuorumPe		3888	Internal	hbase. zookeeper.	HBase 管理的
	er				leaderport	ZooKeeper 模式
	REST	Non-	8080	External	hbase.rest.port	HBase 中的默认
		Cloudera				REST 端口是 8080。
		managed				因为这是一个常用的
		managea				端口,所以 Cloudera
						Manager 将默认值设
						置为 20550。
	REST	Cloudera	20550	External	hbase.rest.port	HBase 中的默认
		Manager -				REST 端口是 8080。
		managed				因为这是一个常用的
						端口,所以Cloudera
						Manager 将默认值设
						置为 20550。
	REST UI		8085	External		
	Thrift Server	Thrift	9090	External	Pass -p <port> on CLI</port>	
	<b>T1</b> :00	Server	0005			
	Thrift Server	Auro	9095	External	Daga part (part) on	
		server	9090	External	CLI	
	hbase-solr- indexer	Lily Indexer	11060	External		
	Server (with	macher	2181	External	clientPort	客户端端口
	CDH 5 or					
	Cloudera					
	Server (with		2888	Internal	X in server N =host·X·Y	Peer
	CDH 5		2000	Interna		
	only)					
ZooKeeper	Server (with		3888	Internal	X in server.N =host:X:Y	Peer
	CDH 5					
	Server (with		3181	Internal	X in server.N =host:X:Y	Peer
	CDH 5 and			litteritur		
	Cloudera					
	Manager 5)		4101	T / 1	X7 ' XT 1 . X7 Y	
	Server (with CDH 5 and		4181	Internal	A in server.N =host:X:Y	Peer

Cloudera Manager 5)			
ZooKeeper JMX port	9010	Internal	ZooKeeper 还会为 RMI 使用另一个随机 选择的端口。

注:

"Internal"表示端口仅用于组件之间的通信;"External"表示端口可用于内部 或外部通信。

## 3. CDH端口列表

产品	组件	端口号	
Cloudera Manager Server	HTTP (Web UI)	7180	
Cloudera Manager Server	HTTPS (Web UI)	7183	
Cloudera Navigator Metadata Server	HTTP (Web UI)	7187	
Backup and Disaster Recovery	HTTP (Web UI)	7180	
Backup and Disaster Recovery	HTTPS (Web UI)	7183	
Backup and Disaster Recovery	HDFS NameNode	8020	
Backup and Disaster Recovery	HDFS DataNode	50010	
Telemetry Publisher	НТТР	10110	
Telemetry Publisher	HTTP (Debug)	10111	
Cloudera Manager Server	Avro (RPC)	7182	
Cloudera Manager Server	Embedded	7432	
Cioudera manager Server	PostgreSQL database	7132	
Cloudera Manager Server	Peer-to-peer parcel distribution	7190,	
Croudera manager Server		7191	
Cloudera Manager Agent Listening	Custom protocol	9000	
Port		2000	
Event Server	Custom protocol	7184	
Event Server	Custom protocol	7185	
Event Server	HTTP (Debug)	8084	
Alert Publisher	Custom protocol	10101	

Service Monitor	HTTP (Debug)	8086
Service Monitor	Custom protocol	9997
Service Monitor	Internal query API (Avro)	9996
Activity Monitor	HTTP (Debug)	8087
Activity Monitor	Custom protocol	9999
Activity Monitor	Internal query API (Avro)	9998
Host Monitor	HTTP (Debug)	8091
Host Monitor	HTTPS (Debug)	9091
Host Monitor	Custom protocol	9995
Host Monitor	Internal query API (Avro)	9994
Reports Manager	Queries (Thrift)	5678
Reports Manager	HTTP (Debug)	8083
Cloudera Navigator Audit Server	НТТР	7186
Cloudera Navigator Audit Server	HTTP (Debug)	8089