EsgynDB 安装部署指南 2.8.0

2021 年 04 月

目录

前言

本文简介

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

安装EsgynDB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EsgynDB系统管理员。

修订历史

大版本	小版本	日期	摘要
2.8	2.8.0	2020年4月	删除原安装 Operational Management 删除原 11.7 已知问题及解决办法 删除原 15.高可用指南 EsgynDB 安装之前检查集群时间是否同步命令 ntp 改成 chrony 5.5 页面部署 CDH 添加注意事项 安装 EsgynDB 中更新软件包下载地址
2.7	2.7.0	2020年7月	更新 EsgynDB2.6.3 升级到 R2.7.0 的相关操作指南
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 EsgynDB 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 EsgynDB 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 客户端安装 EsgynDB，在安装之前，建议您装好以下或功能类似的软件：

- 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 http://www.realvnc.com/）
- 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

EsgynDB 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 8≤ 每节点的 CPU 核数≤16
Memory	64GB 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，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，并且根据业务量按照每多一个 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 EsgynDB 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 EsgynDB 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
      ```

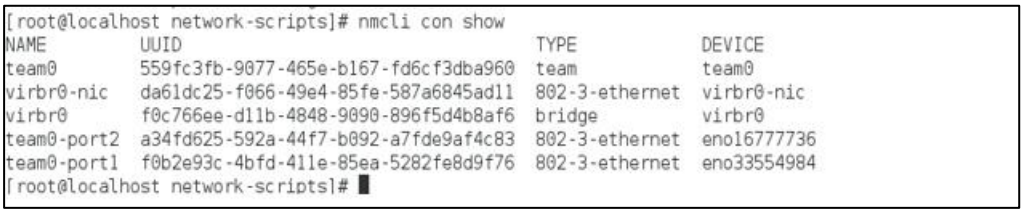
      运行结果如下图所示：

      ```
      [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  team0
      lo           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：



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 EsgynDB 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 4 GB	1.5 倍内存大小
> 64GB	至少 4 GB	不建议开启休眠功能

对于安装 EsgynDB 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EsgynDB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EsgynDB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。

EsgynDB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EsgynDB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EsgynDB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:

安装 EsgynDB 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 5.13
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)
$ mkdir cloudera-scm-server-db
$ chmod 777 cloudera-scm-server-db
$ cd /var/lib
$ ln -s <new drive>/cloudera-scm-server-db
```

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.

EsgynDB 与 Cloudera 和 Hortonworks 发行版兼容。

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

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

EsgynDB 命令行安装程序需要在 EsgynDB 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 EsgynDB 节点必须安装 HBase。

注意

EsgynDB 安装之前请检查集群中各节点时间是否同步，使用命令 chronyc sources，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 chrony 服务后再进行数据库安装!

```
[root@ESGZB-qa-n110 ~]# chronyc sources
210 Number of sources = 2
MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================
^* ESGZB-qa-n109.esgyn.local     9   3   377     5    +11us[  +15us]
+/-  627ms
```

5.2 准备工作

在拿到新的系统之后，需要做下列的准备工作，才能开始相关的预安装工作.

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@esgzb-qa-n144 pre_install]# ll
total 534684
-rw-r--r-- 1 root root  61836298 Apr  9 09:47
EsgynDB_pyinstaller-2.8.0-RH7.tar.gz
drwxr-xr-x 9 root root      4096 Apr  9 12:10 python-
installer
[root@esgzb-qa-n144 R2.8.0_D_20210408]# cd python-
installer/
[root@esgzb-qa-n144 python-installer]# ll
total 408
-rwxr-xr-x 1 root root 14936 Apr  8 19:09 add_nodes.py
-rwxr-xr-x 1 root root 38675 Apr  8 19:09 all_install.py
-rwxr-xr-x 1 root root  8068 Apr  8 19:09
all_uninstall.py
-rwxr-xr-x 1 root root  9586 Apr  8 19:09 auto_config.py
-rwxr-xr-x 1 root root 18569 Apr  8 19:09 cdh_install.py
drwxr-xr-x 3 root root  4096 Apr  8 19:09 cm_api
drwxr-xr-x 2 root root  4096 Apr  8 19:09 configs
```

```
-rwxr-xr-x 1 root root 47001 Apr  8 19:09 db_install.py
-rwxr-xr-x 1 root root  7317 Apr  8 19:09 db_uninstall.py
-rwxr-xr-x 1 root root  8522 Apr  8 19:09 delete_nodes.py
drwxr-xr-x 2 root root  4096 Apr  8 19:10 ha_jars
-rwxr-xr-x 1 root root  9989 Apr  8 19:09 inspector.py
-rw-r--r-- 1 root root 14403 Apr  8 19:10 LICENSE
drwxr-xr-x 4 root root  4096 Apr  8 19:09 lock_setup
-rw-r--r-- 1 root root 18168 Apr  8 19:09
nodes_replace.py
-rw-r--r-- 1 root root   291 Apr  8 19:10 NOTICE
drwxr-xr-x 5 root root  4096 Apr  8 19:10 omclient
-rwxr-xr-x 1 root root 10981 Apr  8 19:09 pre_install.py
-rw-r--r-- 1 root root 54204 Apr  8 19:09 prettytable.py
-rw-r--r-- 1 root root 52543 Apr  9 11:49 prettytable.pyc
-rw------- 1 root root    36 Apr  8 19:10 PyInstallerVer
-rw-r--r-- 1 root root  3672 Apr  8 19:09 README.md
drwxr-xr-x 4 root root  4096 Apr  9 11:49 scripts
-rwxr-xr-x 1 root root 16127 Apr  8 19:09 secure_setup.py
drwxr-xr-x 2 root root  4096 Apr  8 19:09 templates
-rwxr-xr-x 1 root root 13679 Apr  8 19:09
upgrade_nodes.py
```

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 ~]# ll cdh/cm_rpms/
total 690312
-rw-r--r--. 1 root root   9805556 Apr 19 11:45 cloudera-
manager-agent-5.13.3-1.cm5133.p0.6.el7.x86_64.rpm
```

```
-rw-r--r--. 1 root root 697039048 Apr 19 11:45 cloudera-
manager-daemons-5.13.3-1.cm5133.p0.6.el7.x86_64.rpm
-rw-r--r--. 1 root root      8696 Apr 19 11:38 cloudera-
manager-server-5.13.3-1.cm5133.p0.6.el7.x86_64.rpm
-rw-r--r--. 1 root root     10608 Apr 19 11:38 cloudera-
manager-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 ~]# ll 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 mysql-
connector-java-5.1.34.jar
```

7. MySQL的jdbc driver

```
[root@host-10-10-23-62 ~]# ll 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 mysql-
connector-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-n013
10.10.23.64 esggy-qa-n014.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 chrony 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_64-dvd.iso
```

--填写 centos 的依赖包镜像全路径

```
Enter the gateway of the LAN(used for chrony 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 chrony 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 |
| chrony_gateway  | 10.10.23.1
|
| chrony_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              |  o   |     -      |
|          Linux distro          |  o   |     -      |
|        Firewall status         |  o   |     -      |
|         Selinux status         |  o   |     -      |
|       chrony service status       |  o   |     -      |
|       chrony 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]# ll
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-qa-n012.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

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

TASK: Deploy Mysql
**********************************************************
**************

Host [esggy-qa-n014.esgyncn.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 [esggy-qa-n013.esgyncn.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

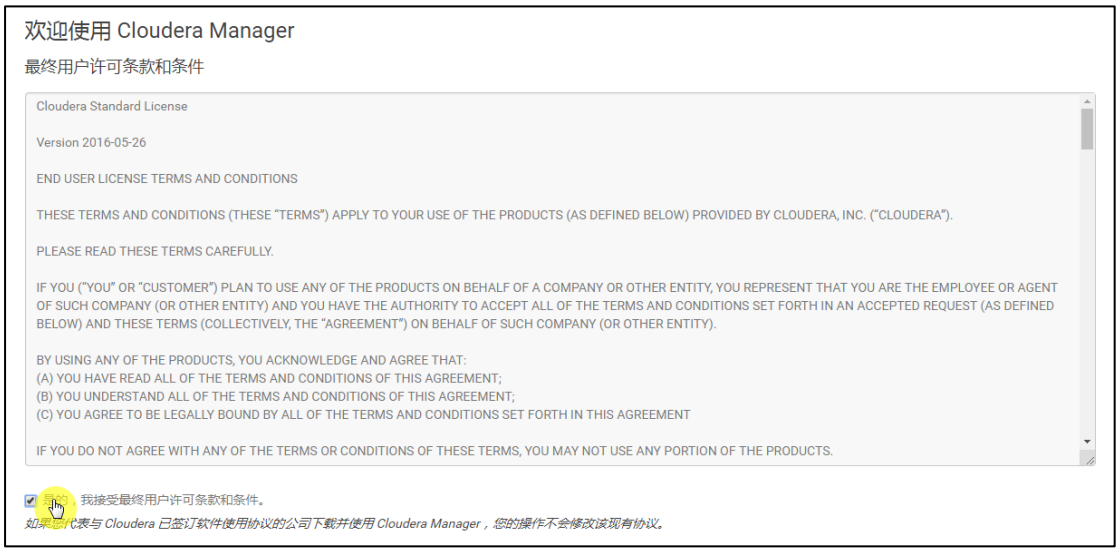
前面的两个步骤都成功完成后就可以到Cloudera网页上部署CDH了。

注意：由于R2.8.0中python installer会自动安装HA addon，用于实现HA快速恢复，安装时需要搭配CDH 5.16.2版本，所以这里建议使用CDH 5.16.2。如果使用其它版本的CDH，并且不配置HA addon，可以在安装EsgynDB数据库时，使用python installer的--not-enhance-ha参数。

5.5.1 登录 Cloudera 网页

打开浏览器，进入 http://10.10.23.62:7180 网页，这是安装 Cloudera

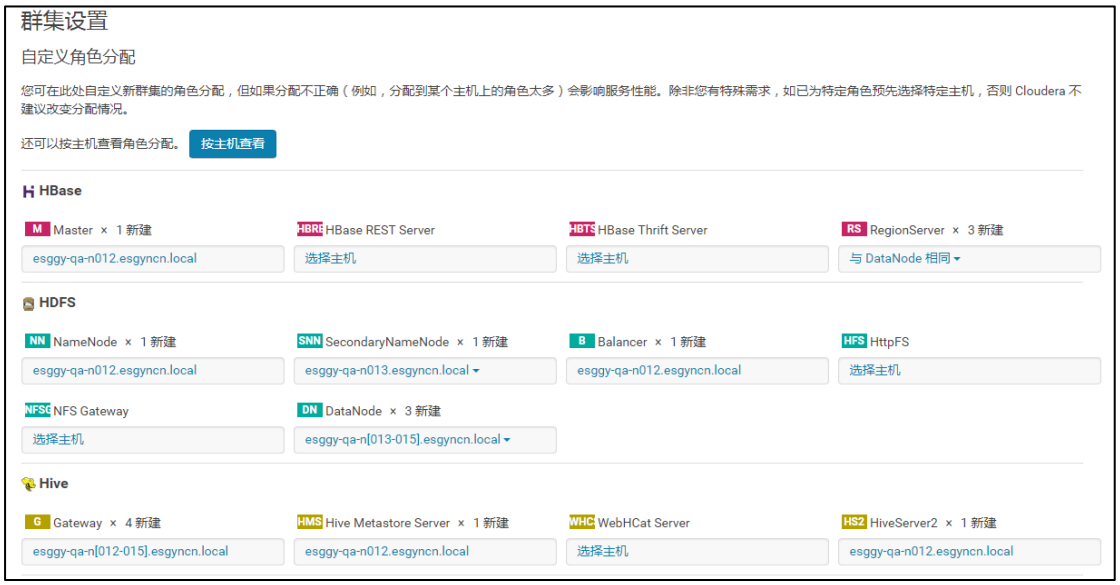
Manager Server 的节点的 IP 地址，默认登陆用户密码是 admin/admin，用户可用此账号信息登录后自行修改密码：



5.5.2 选择部署节点

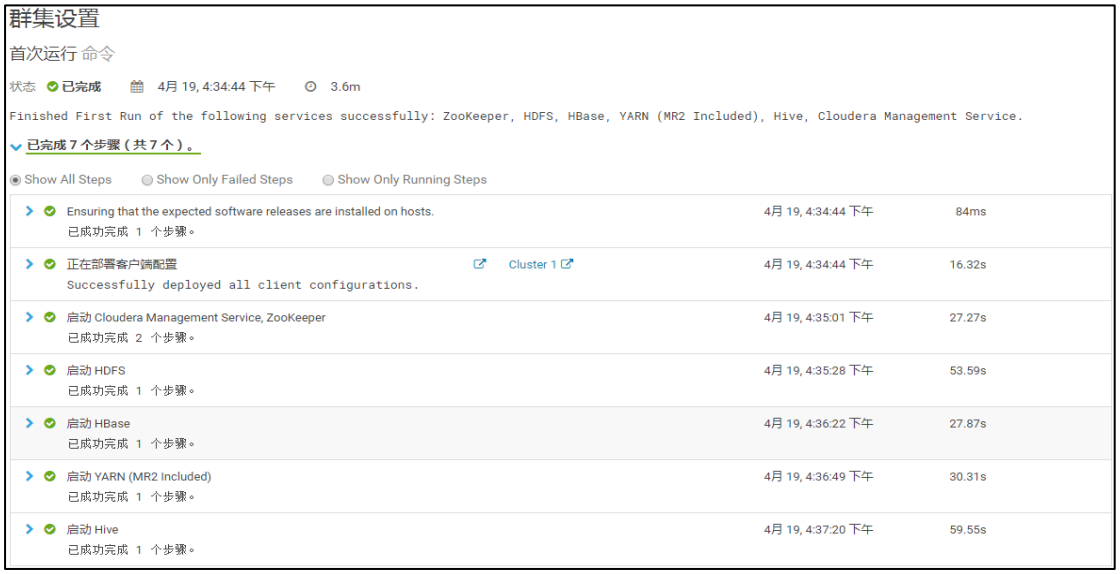
5.5.3 选择存储库

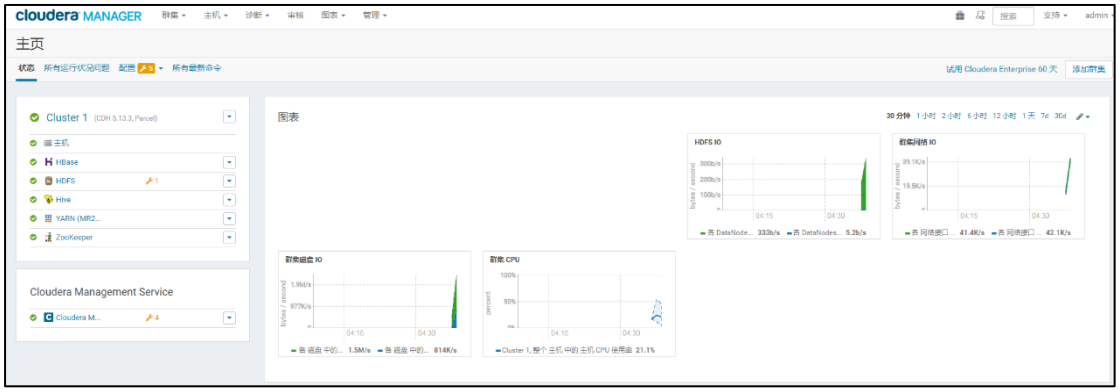
5.5.4 角色分配



5.5.5 数据库设置

5.5.6 安装成功





CDH 就部署成功。

5.6 Hadoop 配置 HA

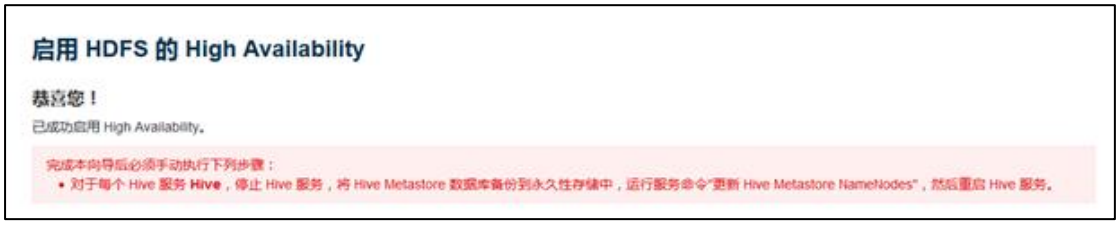
5.6.1 HDFS 配置高可用配置

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

2) 制定 HA 名字的名称

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

4) 制定多个磁盘的多个目录



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

2) 指定 RM 的 2 个主机名

5.6.3 HBASE HA 配置

1) Hbase 服务--》实例--》添加角色实例

2) 点击 Master 下的选择主机

3）选择多个主机

添加角色实例

继续下一步

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。

5.6.4 HIVE HA 配置

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

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 Heap Size	4 GiB	在大型配置中应用此设置。
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 堆栈大小（字节）	1G	1G

5.7.3 HBase 参数调优

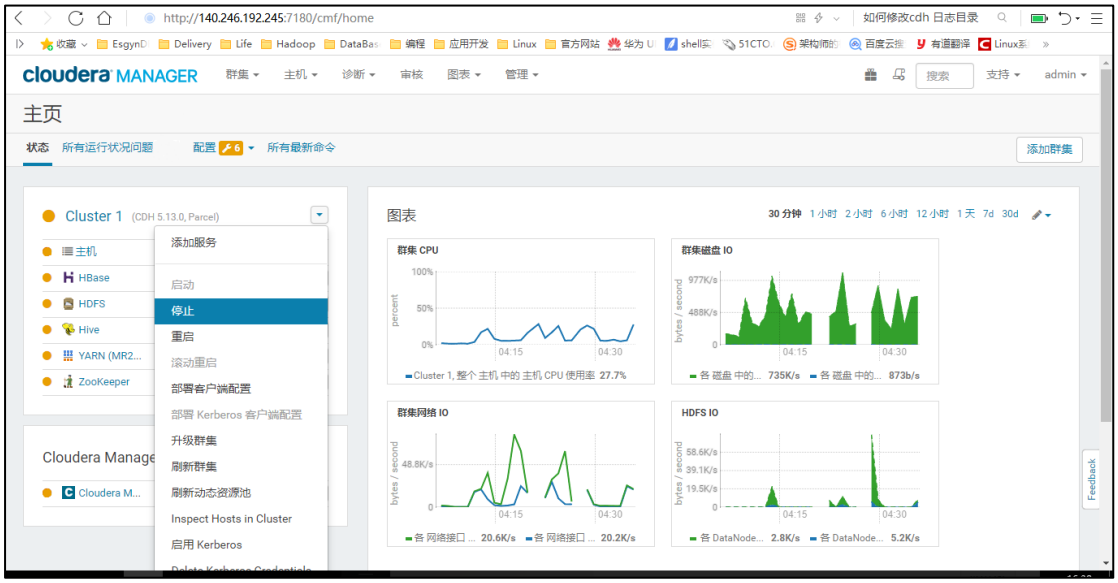
HBase 配置属性	推荐设置	备注
hbase.rpc.timeout	10 分钟	此设置取决于表的大小。默认值是 60 seconds。对于大表需要调高这个值。此值需保持跟参数 hbase.client.scanner.timeout.period 一样。我们发现把这个值调为 600 秒会减少超时相关的

		错误，譬如 OutOfOrderNextException 错误。
hbase.regionserver.lease.period, hbase.client.scanner.timeout.period	60 分钟	跟上面的配置类似，默认值是 60 秒。根据用户表大小的不同，在做 count(*) 和 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.timeout.millis hbase.snapshot.master.timeoutMillis hbase.snapshot.region.timeout	10 分钟	HBase 默认设置是 60000 毫秒。如果您使用 Trafodion Bulk Loader 或其他语句时遇到与 HBase 快照相关的超时问题，可以设置这两个 HBase 的属性为 10 分钟（600,000 毫秒）的值。
hbase.hregion.max.filesize	10737418240 bytes	HBase 默认设置是 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.blockingStoreFiles	10	CDH 5.4 默认值是 10。安装 EsgynDB 之后 Installer 会改成

		200.这也是 HBase 早期版本的默认值。
hbase.regionserver.handler.count	参见备注	此设置需要跟 mxosrvr 的并发会话数目设置匹配。默认值是 30。
HBase Master 的 Java 堆栈大小（字节）	2G	缺省 1G
HBase RegionServer 的 Java 堆栈大小（字节）	31G	缺省 4G
客户端 Java 堆大小（字节）	256M	缺省 256M
HBase Memstore 刷新大小 hbase.hregion.memstore.flush.size	512M	缺省 128M
hbase.hregion.majorcompaction	0	缺省 7 天
hbase.regionserver.handler.count	200	缺省 10

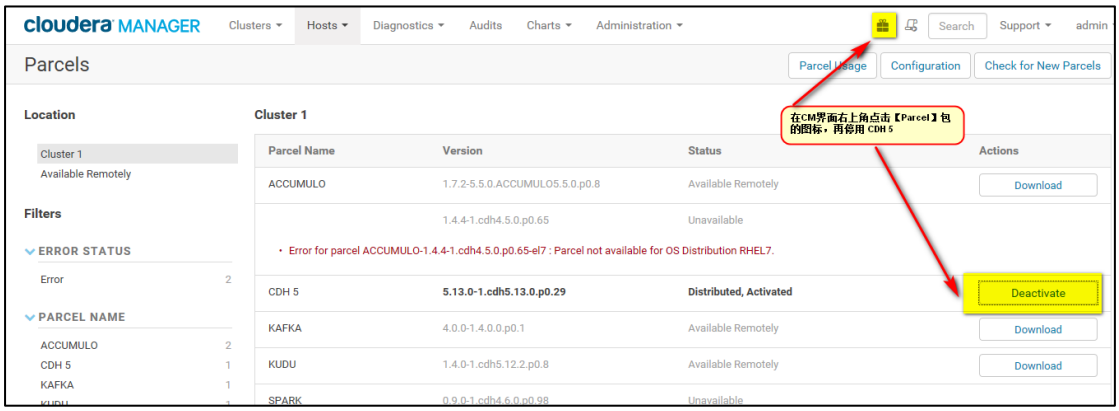
6. 卸载 CDH 及 Hadoop 服务

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

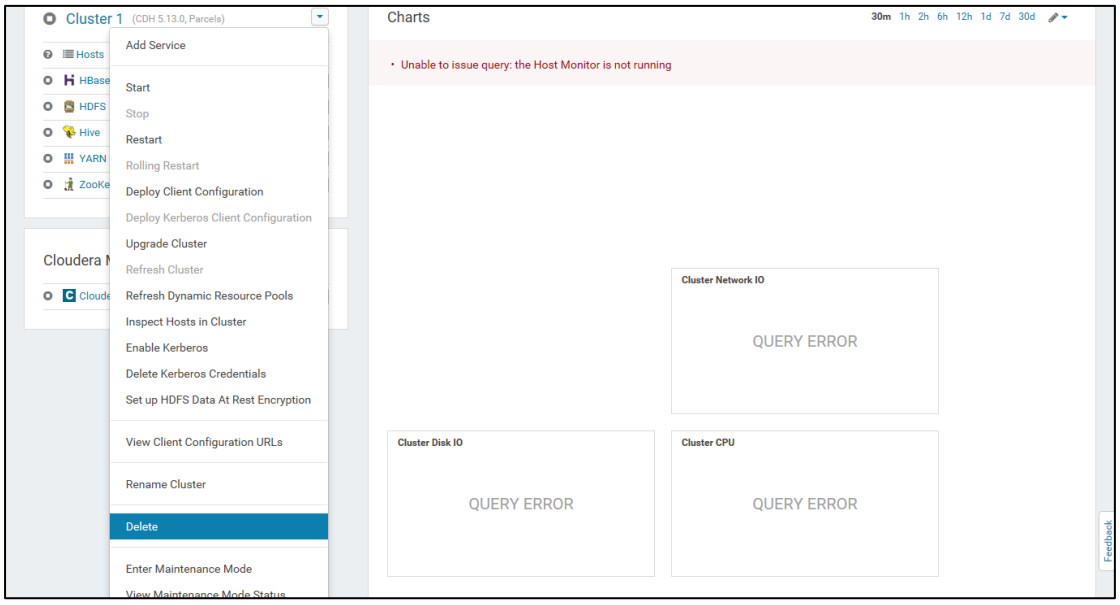


2. 移除parcel包

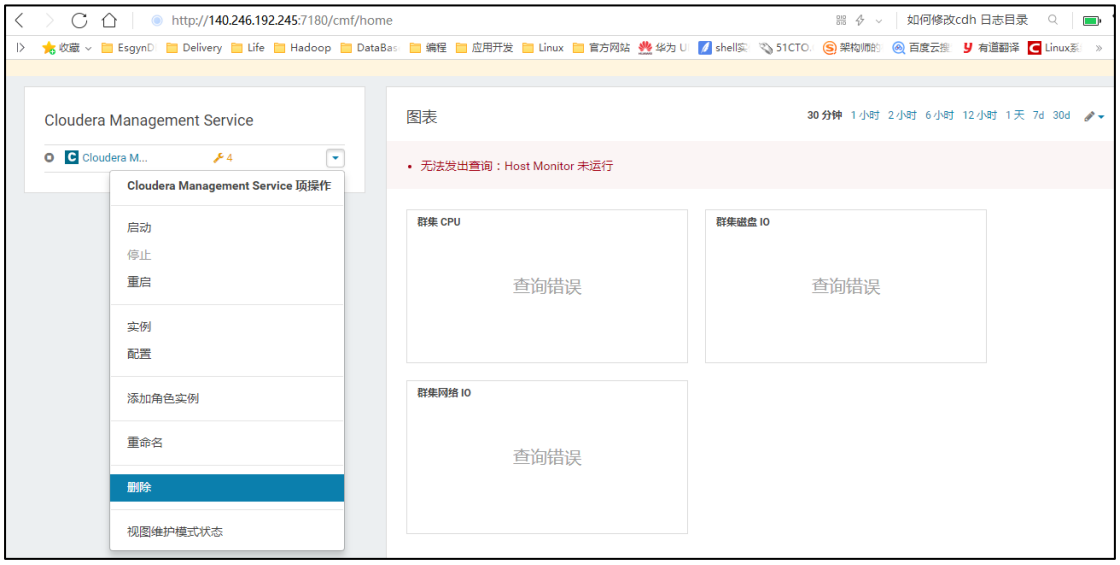
假如是通过 packages 来安装的，那么请跳过这个步骤。本文是针对parcel包方式进行安装，在CM界面右上角点击Parcel包的图标：



3. 通过CM删除集群Cluster 1



4. 删除Cloudera Management Service



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

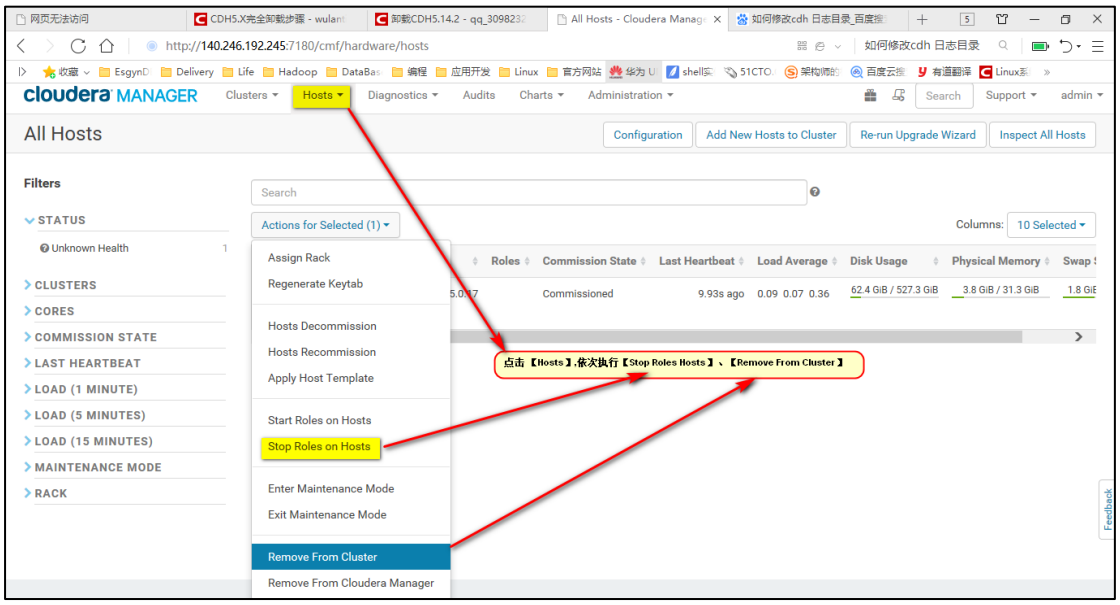
ps -ef|grep cmf|grep supervisord;ps -ef|grep cmf|grep flood

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



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

```
[root@sjtl-n001 ~]# service cloudera-scm-server stop  --
-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 ~]#
```

3) 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/*`

7. 准备安装 EsgynDB

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）

EsgynDB 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
$ cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
$ echo localhost $(cat /etc/ssh/ssh_host_rsa_key.pub) >>
$HOME/.ssh/known_hosts
$ echo "NoHostAuthenticationForLocalhost=yes" >>
$HOME/.ssh/config
$ chmod 600 $HOME/.ssh/config
```

2. 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 EsgynDB 验证（Authentication）功能，您需使用 LDAP Identity Store 进行验证。EsgynDB 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 EsgynDB 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 12.

用户 ID 和密码

以下为安装 EsgynDB 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 网页登录	`admin`（默认）	`admin`（默认）	• 安装 Cloudera 后，系统会提示您登录 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权限的用户	`<sudo-username>`	`<password>`	• 安装时，您可能需要 sudo 或 sudo 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。
EsgynDB 登录	`trafodion`	`traf123` （默认）	• 安装 EsgynDB 时，该用户 ID 由 EsgynDB 安装程序**自动创建**。 • 请勿手动创建该用户。

7.3 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 Red Hat Linux 6.5 ~ 6.9, 7.2 ~ 7.4	pdsh	apr
	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	openldap-clients
	libcgroup	snappy

7.4 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的数量	8	指定每个节点并行会话的数量。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 EsgynDB 上升级安装或全新安装	无	如果在新集群上安装，安装程序将执行额外操作。
`trafodion` 用户 ID 和密码	用户 ID： `trafodion` 密码： ******	建议您不要更改用户 ID。 • python 命令行安装程序会提示您输入密码。
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 的根目录路径前缀	`/home`	如果 `trafodion` 用户 ID 的根目录为`/opt/home/trafodion`，则前缀为`/opt/home`。

[1] 更多信息，请参阅《EsgynDB DCS 安装指南》。

（续前表）

信息	默认	备注
JDK 的路径	无	JDK 的绝对路径。 **示例** `/usr/java/jdk1.8.0_112-cloudera`
EsgynDB 安装程序压缩文件的路径	无	指定 EsgynDB 安装程序的完整路径。

Hadoop 发行版 URL	无	在表单中指定:
		`<IP-address>:<port>`
		或
		`<node name>:<port>`
		
		示例
		`vm-1.yourcompany.local:7180`

Hadoop 发行版详情 • 管理员界面用户 ID、密码 • 集群名称 • HDFS 用户 ID • HBase 用户 ID、组 • HBase 服务名称	 • 依赖于发行版 • `Cluster1` • `hdfs` • `hbase, hbase` • `hbase`	

（续前表）

信息	默认	备注
EsgynDB 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 EsgynDB

本章讲述以下内容:

8.1 命令行安装程序

EsgynDB 命令行安装工具为 db_install.py，它是一个独立的安装包。

> **注意**
>
> EsgynDB 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 EsgynDB，执行以下步骤：

1. 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：
     - EsgynDB_server-2.8.0-RH7-x86_64.tar.gz
   - 相应的安装软件包如下：
     - EsgynDB_pyinstaller-2.8.0-RH7.tar.gz

```
$ mkdir $HOME/esgyndb_downloads
$ mv <your-download-path>/ EsgynDB_server-2.8.0-RH7-
x86_64.tar.gz $HOME/esgyndb_downloads
$ mv <your-download-path>/ EsgynDB_pyinstaller-2.8.0-
RH7.tar.gz $HOME/esgyndb_downloads
$ cd $HOME/esgyndb_downloads
```

3. 解压下载的安装文件。

```
$ tar -xzf EsgynDB_pyinstaller-2.8.0-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
引导设置	推荐新手使用。 安装时，系统将提示所需信息。	执行./db_install.py
专家设置	推荐以下用户使用: • 资深用户 • 使用无人值守安装（Unattended Setup）方式的用户 配置文件是一个预编辑好的文本文件，它包含安装所需信息。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。	在安装配置文件中输入所需参数后，调用安装程序 ./db_install.py -config_file <*installer-config-file*>

	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 EsgynDB 安全功能的详细信息，请参阅 12. 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 EsgynDB 发行版文件复制至集群所有节点。

(6) 生成启动文件。

(7) 启动 EsgynDB、数据库连接服务[2]和 EsgynDBManager 服务。

[2] 即 Database Connectivity Services, DCS。

6. 安装完成后，您将看到以下消息:

```
*************************
  Installation Complete
*************************
```

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

   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 EsgynDB 需要以 trafodion 用户身份登录。

以下为管理子系统的脚本。

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

示例

启动 EsgynDB。

```
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. 如需将客户端应用程序连接至 EsgynDB，您可以下载并安装 EsgynDB JDBC 和/或 ODBC 驱动程序（安装程序名称为 `EsgynDB_clients-2.8.0-RH7-x86_64.tar.gz`[3]）。

[3] 更多信息，请参阅《EsgynDB 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 EsgynDB 以及如何运行示例程序测试连接。

9. 卸载 EsgynDB

本章讲述以下内容:

卸载 EsgynDB 前，确定数据已保存。

使用 Trafodion 配置用户 ID 卸载 EsgynDB。

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

> **注意**
>
> 如需更新 EsgynDB，无需卸载现有 EsgynDB。

9.1 停止 EsgynDB

执行以下命令:

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

9.2 卸载 EsgynDB

EsgynDB 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 EsgynDB 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 EsgynDB 集群中的任一节点:
2. esgynDB_pyinstaller-2.7.0-RH7.tar.gz
3. 解压安装包

```
tar zxf esgynDB_pyinstaller-2.7.0-RH7.tar.gz
```

4. 执行 db_uninstall.py 卸载 EsgynDB

(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)

```
[root@esggy-qa-n025 ~]# cd python-installer/

[root@esggy-qa-n025 python-installer]# ./db_uninstall.py
******************************
Trafodion Uninstall Start
```

```
*****************************
Enter Trafodion node list to uninstall(separated by
comma): 10.10.23.76,10.10.23.77,10.10.23.78
--输入 EsgynDB 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 EsgynDB 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 EsgynDB
***[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
*********************************
```

10. 升级 EsgynDB

本章以 EsgynDB R2.7.2 升级到 EsgynDB R2.7.0 为例介绍了如何升级 EsgynDB 并保证数据的完整性,内容如下,

0

环境检查

10.2 备份配置文件

10.3 在线备份数据

10.4 备份 metadata

10.5 升级 EsgynDB

10.6 版本回退

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 0

环境检查)、

备份数据和文件（见章节 10.2 & 10.3 & 10.4）

升级 EsgynDB（见章节 10.5）

升级后：在线备份数据库（见章节 10.3）

注意

升级到 EsgynDB2.8.0 之前，请确认 CDH 版本是否为 5.16.2。如果已经是该版本，请手动将 zookeeper jar 包替换成 zookeeper-3.4.5-cdh5.16.2.jar，如果不是，请先将 CDH 版本升级至 5.16.2 或者联系易鲸捷技术人员获取相对应 CDH 版本的 zookeeper jar 包。

步骤一：获取相应的 zookeeper jar 包；

步骤二：备份/opt/cloudera/parcels/CDH/jars/中原有 zookeeper jar 包到其他目录；

步骤三：将步骤一中获取的 jar 放在/opt/cloudera/parcels/CDH/jars/中，并且保持和原有 jar 包同样的名字；

步骤四：通过 pdcp 将新的 zookeeper jar 包同步到所有节点；

步骤五：为保障所有组件都使用到新的 zk 包，重启整个集群包含 hadoop 和数据库。

10.1 环境检查

升级前检查当前环境是否符合升级条件。

1) 停止 EsgynDB:

```
[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 EsgynDB，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: 47.333333333333336
```

10.2 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) EsgynDB R2.7.2 ms.env文件：位于/opt/trafodion/esgynDB_server-2.6.3/etc/ms.env

EsgynDB R2.7.0 ms.env文件：位于/var/lib/trafodion/ms.env

```
[trafodion@node01 ~]$ cp /opt/trafodion/esgynDB_server-2.6.3/etc/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。

3) EsgynDB R2.7.2 dcs conf：位于$DCS_INSTALL_DIR/conf/*

EsgynDB R2.7.0 dcs conf：位于/etc/trafodion/conf/dcs/*

```
[trafodion@node01 ~]$ cp -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 ~]$ cp /etc/trafodion/trafodion_config ~/backup_conf
```

5) EsgynDB R2.7.2 ldap conf:

位于 /opt/trafodion/esgyndb/sql/scripts/.traf_authentication_config

EsgynDB R2.7.0 ldap conf:

位于 /etc/trafodion/conf.traf_authentication_config

```
[trafodion@node01~]$ cp /etc/trafodion/conf/.traf_authentication_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)

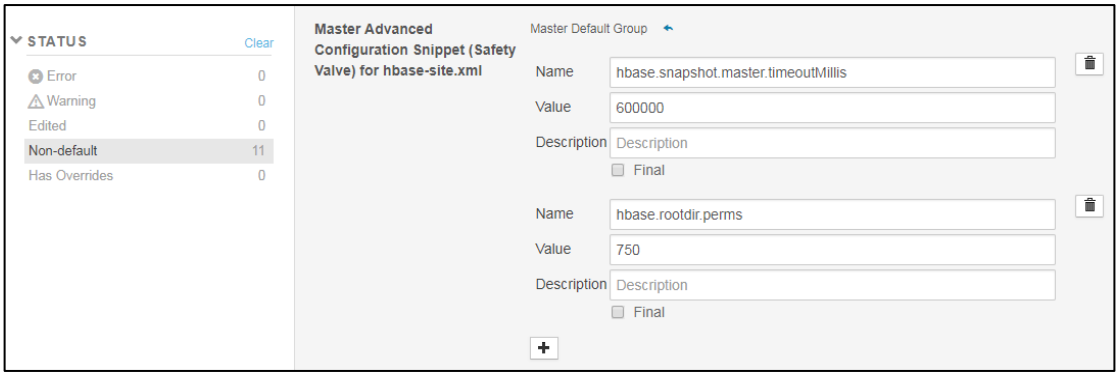
6) sqenvcom.sh：位于$TRAF_HOME/sqenvcom.sh

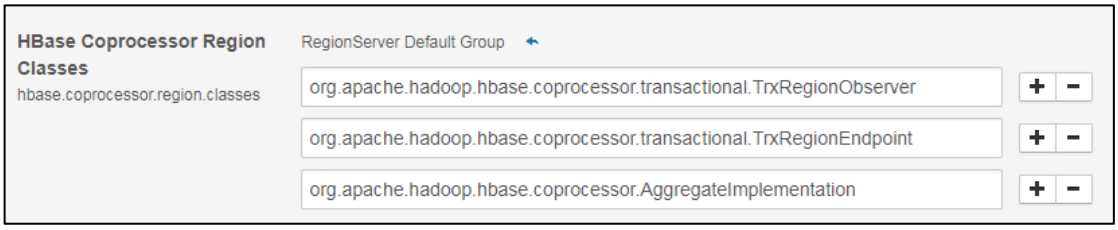
```
[trafodion@node01 ~]$ cp $TRAF_HOME/sqenvcom.sh ~/backup_conf
```

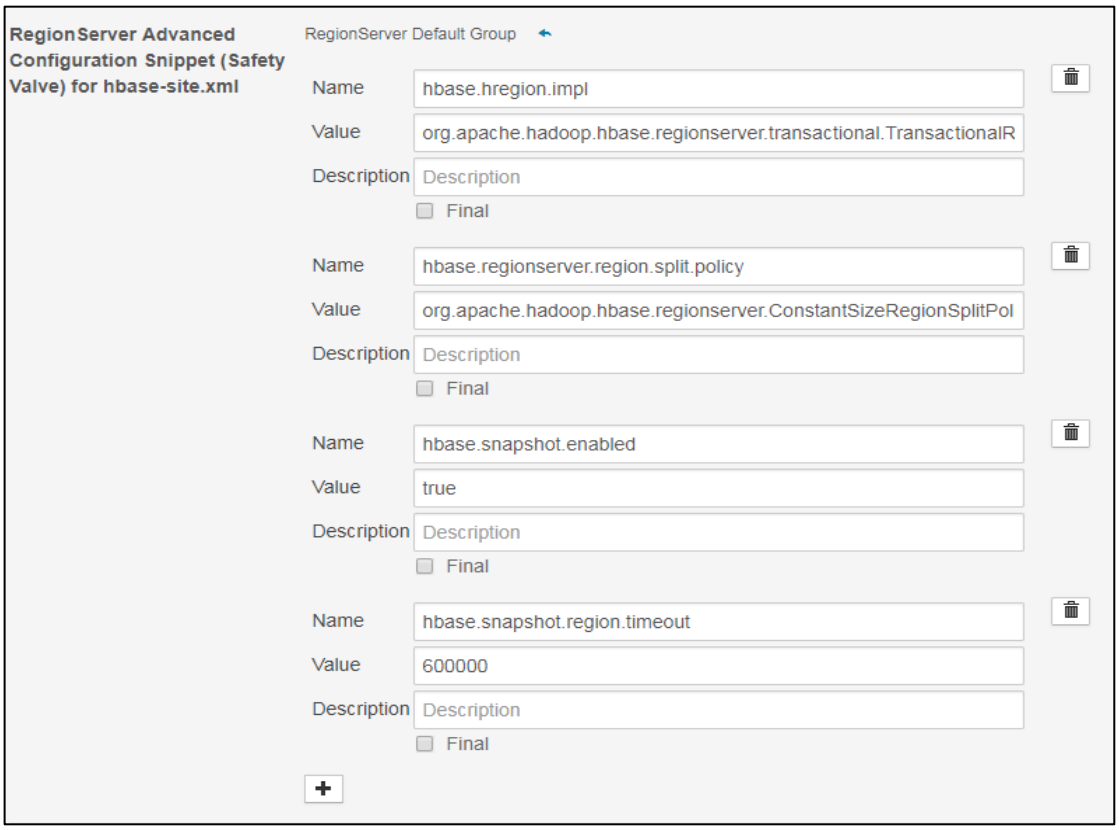
7) 记录 hadoop server 的部分参数值（在 Cloudera manager 或 ambari 网页上查找），如下表所示，加粗文本是参数值:

```
<HBase config>
"hbase-site": {
        "hbase.snapshot.master.timeoutMillis": "600000",
        "hbase.table.sanity.checks": "false",
        "hbase.hregion.impl":
"org.apache.hadoop.hbase.regionserver.transactional.Trans
actionalRegion",
        "hbase.regionserver.region.split.policy":
"org.apache.hadoop.hbase.regionserver.ConstantSizeRegionS
plitPolicy",
        "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
Error 0
Warning 0
Edited 0
Non-default 11
Has Overrides 0
Master Advanced Configuration Snippet (Safety Valve) for hbase-site.xml
Master Default Group
Name
hbase.snapshot.master.timeoutMillis
Value
600000
Description
Description
Final
Name
hbase.rootdir.perms
Value
750
Description
Description
Final


HBase Coprocessor Region Classes
hbase.coprocessor.region.classes
RegionServer Default Group
org.apache.hadoop.hbase.coprocessor.transactional.TrxRegionObserver
org.apache.hadoop.hbase.coprocessor.transactional.TrxRegionEndpoint
org.apache.hadoop.hbase.coprocessor.AggregateImplementation


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

10.3 在线备份数据

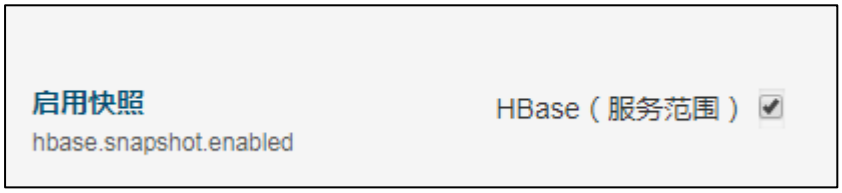
10.3.1 备份简介

EsgynDB_R2.7.2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EsgynDB失败导致数据丢失，故先在R2.7.2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R2.7.2。

10.3.2 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:



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>cleanup metadata,check,return details;

DDL_OUTPUT
--------------------------
Metadata Cleanup: started, check only

Start: Cleanup Orphan Objects Entries
Entry #1(OBJECT): TRAFODION.ABC.B1[4]
End:   Cleanup Orphan Objects Entries (1 entry found)

Start: Cleanup Orphan Hbase Entries
Entry #1(OBJECT): TRAFODION.ABC.A1[5]
End:   Cleanup Orphan Hbase Entries (1 entry found)
```

[5] 本例中 A1 表为不一致数据，已经从 EsgynDB 中删除，却还存在于 HBase 中。

[6] 本例中 B1 表为不一致数据，已经从 HBase 中删除，却还存在于 EsgynDB 中。

```
 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[6]，清除语句如下：

```
trafodion@node01 ~]$ trafci
 > 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
```

[6] 建议不清除，不影响升级，知道有这些不一致数据即可。

```
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
……
```

10.3.3 在线备份 EsgynDB

EsgynDB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.
```

10.3.4 检查在线备份数据

1) 监控备份进度[7]

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
……
```

[7] 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

```
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:

```
[trafodion@node01 ~]$ trafci
SQL> get all backup snapshots;
BackupTag       BackupTime       BackupStatus
BackupOperation
=====================================================
bkfulldb_00212413689918115206[8]   2019-01-08:14:52:30
VALID         REGULAR
--- SQL operation complete.
```

10.3.5 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
```

[8] 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;

```
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 2020-07-20
15:58
/user/trafodion/trafodion_backups/TRAFODION.SCH0501.T1IN
CR1
drwxr-xr-x   - trafodion trafodion          0 2020-07-20
15:57
/user/trafodion/trafodion_backups/TRAF_inctest1_TRAFODIO
N.NASCH1.T1INCR2
drwxr-xr-x   - trafodion trafodion          0 2020-07-20
16:08
/user/trafodion/trafodion_backups/TRAF_inctest2_TRAFODIO
N.NASCH2.T1INCR3
drwxr-xr-x   - trafodion trafodion          0 2020-07-20
16:12
/user/trafodion/trafodion_backups/bkfulldb_0021241368991
8115206
```

3) 从hdfs中导出到linux上

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

10.4 备份 metadata

10.4.1 手动备份 snapshot

1) 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EsgynDB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[9]
……
TRAF_RSRVD_1:TRAFODION._MD_.TABLES
……
> snapshot 'TRAF_RSRVD_1:TRAFODION._MD_.TABLES' ,
'SNAP_TRAF_RSRVD_1_TRAFODION._MD_.TABLES'
>list_snapshots[10]
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)
……
```

[9] 查出 R2.7.2 EsgynDB 所有的元数据表（当有多租户时，还有'TRAF_RSRVD_1:TRAFODION._TENANT_MD_.*'）

[10] 查看备份的 metadata 快照

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

```
[trafodion@node01 ~]$ hdfs dfs -mkdir
/trafodion_backups/backup_metadata
[trafodion@node01 ~]$ hdfs dfs -cp /hbase/.hbase-
snapshot/* /trafodion_backups/backup_metadata/
```

10.5 升级 EsgynDB

10.5.1 停用 EsgynDB

升级前确保 Esgyn DB 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
……
```

10.5.2 解压 Installer

1) 解压已下载的R2.7.0 python installer文件

```
[root@node01 r2.5]# tar -zxf esgynDB_pyinstaller-2.7.0-
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 228
-rwxr-xr-x 1 root root 14124 May 20 10:26 add_nodes.py
-rwxr-xr-x 1 root root  8325 May 20 10:26 auto_config.py
-rwxr-xr-x 1 root root  6231 May 20 10:26 cdh_install.py
drwxr-xr-x 2 root root  4096 May 20 10:26 configs
-rwxr-xr-x 1 root root 30270 May 20 10:26 db_install.py
-rwxr-xr-x 1 root root  6374 May 20 10:26 db_uninstall.py
-rwxr-xr-x 1 root root  8127 May 20 10:26 delete_nodes.py
-rwxr-xr-x 1 root root  9724 May 20 10:26 inspector.py
-rw-r--r-- 1 root root 14403 May 20 10:27 LICENSE
-rw-r--r-- 1 root root   291 May 20 10:27 NOTICE
-rwxr-xr-x 1 root root  9876 May 20 10:26 pre_install.py
```

```
-rw-r--r-- 1 root root 54204 May 20 10:26 prettytable.py
-rw------- 1 root root    36 May 20 10:27 PyInstallerVer
-rw-r--r-- 1 root root  3672 May 20 10:26 README.md
drwxr-xr-x 3 root root  4096 May 20 10:27 scripts
-rwxr-xr-x 1 root root 12831 May 20 10:26 secure_setup.py
drwxr-xr-x 2 root root  4096 May 20 10:26 templates
-rwxr-xr-x 1 root root  7750 May 20 10:26 upgrade_nodes.py
```

10.5.3 升级 EsgynDB[11]

执行安装脚本，填写相应的集群信息，开始升级EsgynDB，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]:
……
```

10.5.4 检查 EsgynDB

升级成功后，对新版本的 EsgynDB 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 EsgynDB 版本:

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

[11] 如果升级 EsgynDB R2.7.0 时遇到问题，可参考错误!未找到引用源。错误!未找到引用源。，如果有相同问题，对应解决即可。

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;
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,'augkhf1934o
rh43'),(4,'qliuh2389'),(5,'8976489'),(6,'adkjhfb');
--- 6 row(s) inserted.

SQL>select * from a1;
A           B
----------- -------------
      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 EsgynDB 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

10.6 版本回退

如果 EsgynDB 升级失败，需回退到 R2.7.2 版本，参考以下步骤：

10.6.1 收集日志

如果升级 EsgynDB 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。

10.6.2 回退准备

1）停止EsgynDB

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

2) 将10.2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R2.7.2正确的环境变量，每个节点都要复制：

```
[trafodion@node01 ~]$ cp ~/backup_conf/.bashrc  ~/
```

10.6.3 回退安装

回退安装时，与正常安装一样

1) 解压 R2.7.2 的 python installer：

```
[root@node01 CBanl1.1]#tar -xzf
esgynDB_pyinstaller-2.7.2-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。

……

10.6.4 恢复 metadata 数据

当回退R2.7.2，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：

1) 利用10.4备份metadata中备份的快照进行恢复，恢复前，需在hbase中disable所有的R2.7.2 EsgynDB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

>disable_all 'TRAFODION._DTM_.*'
Y
```

2) 利用快照恢复metadata表，恢复时，需恢复10.4备份metadata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EsgynDB，检查恢复后的表

```
[trafodion@node01 ~]$ trafci
SQL>get schemas;
Schemas in Catalog TRAFODION
============================
BR_TEST
SEABASE
VOLATILE_SCHEMA_MXID11001042442212382418436620695000000000
02
_LIBMGR_
_MD_
_PRIVMGR_MD_
_REPOS_
--- SQL operation complete.

SQL>select * from <xx_table>;
……
```

10.6.5 在线恢复数据

回退安装成功后，利用 10.3.3 在线备份 EsgynDB 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[12]   2020-07-20:14:52:30
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 EsgynDB, 参照 10.5.4 检查 EsgynDB 检查检查数据；然后根据 10.6.1 收集日志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

[12] 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;

11. 故障排除

1. 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 EsgynDB 进程是否正常工作。

2. 如果进程未正常运行，执行以下操作：
   (1) 使用 `sqstop` 命令关闭 EsgynDB。如果无法关闭部分 EsgynDB 进程，使用 `ckillall` 命令。
   (2) 使用 `sqstart` 命令重启 EsgynDB。

3. 如果问题仍然存在，检查日志 `$TRAF_LOG: EsgynDB logs`。

12. 启用安全功能

本章讲述以下内容:

12.1 配置 LDAP

EsgynDB 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，EsgynDB 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。EsgynDB 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[13]。

注意

如果 EsgynDB 未启用 AD/LDAP，则客户端需输入用户名和密码才能连接至 EsgynDB，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。

[13] 更多关于**向用户和角色授予权限**的信息，请参阅《EsgynDB SQL 参考手册》的 **GRANT Statement** 章节。

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：

```
[root@esggy-qa-n026 python-installer]# ./secure_setup.py
**********************************
  Trafodion Security Setup Script
**********************************

TASK:                    Environment                    Discover
************************************************************

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

12.2 安装与配置 OpenLDAP 服务器

EsgynDB 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。

12.2.1 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
```

12.2.2 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/openldap-servers/slapd.conf.obsolete 这个文件了，所以现在配置都在/etc/openldap/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/openldap/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=auth"
  read by dn.base="cn=admin,dc=esgyn,dc=local" read by *
none

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: organizationalPerson
objectClass: inetOrgPerson
userpassword: DB_ROOT_PASSWORD #明文密码

# db_admin
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
```

12.2.3 配置 OpenLDAP HA

如下步骤，在两个节点均要执行；

12.2.3.1 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（chronyd）

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）

12.2.3.2 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。

12.2.3.3 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 openldap 是不会有任何报错的，但添加 olcMirrorMode 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。

12.2.3.4 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

12.2.3.5 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;

12.2.4 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 Openldap 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 Openldap 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 Openldap 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，EsgynDB 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。

12.2.4.1 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

12.2.4.2 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 openldap 自带的命令(例如 ldapsearch)对已死锁的 openldap 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 openldap 服务，并脚本返回 1。

12.2.4.3 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。

12.2.4.4 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"
```

-S0 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

12.2.4.5 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。

12.2.5 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/openldap/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

12.2.5.1 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
```

12.2.6 开启 OpenLDAP 的密码策略

12.2.6.1 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

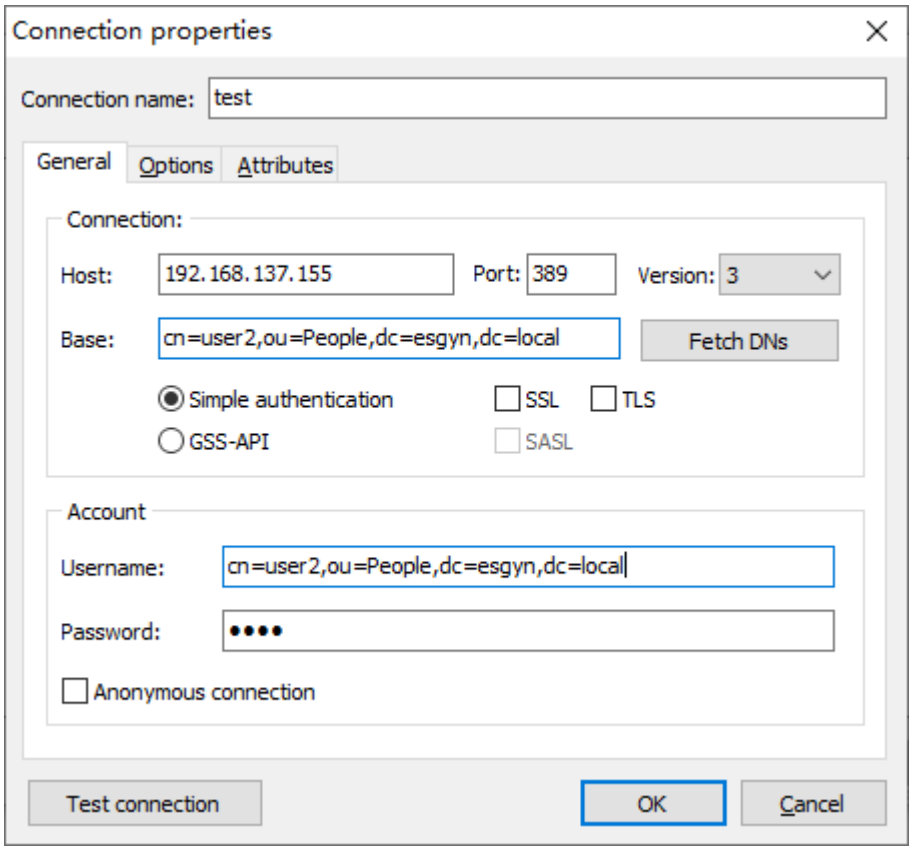
2．除了自身节点外其它节点该用户无法访问的

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。



12.2.6.2 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

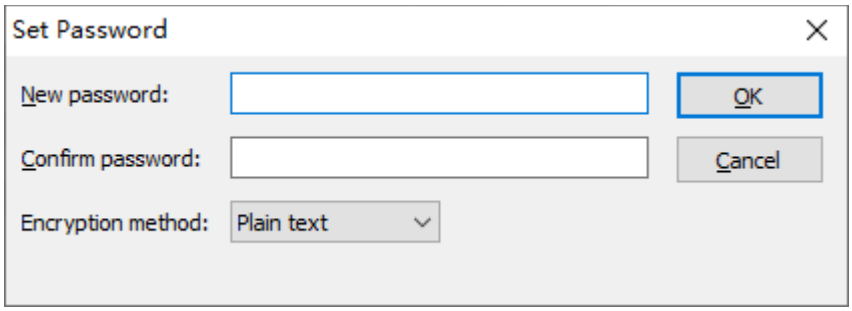
是明文密码

```
{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



phpLDAPAdmin

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。

12.2.6.2.1 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。

12.2.6.2.2 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

12.2.6.2.3 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(cn=test_ppolicy,ou=Pwpolicy,dc=esgyn,dc=local)有访问权。

12.2.6.2.4 已存在的条目变更密码策略方案

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)。

12.2.6.2.5 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。

12.2.6.2.6 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 。

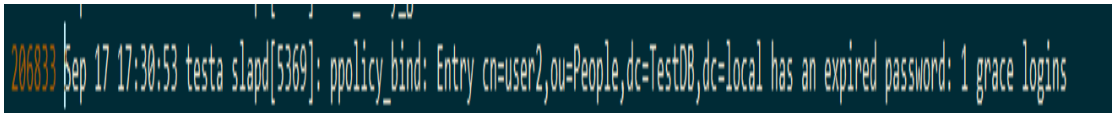
pwdReset 用于"解锁"账户。当设置为 true 时,允许用户再登陆后立即修改自己密码,设置为 false 时,用户解锁可继续使用。

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。

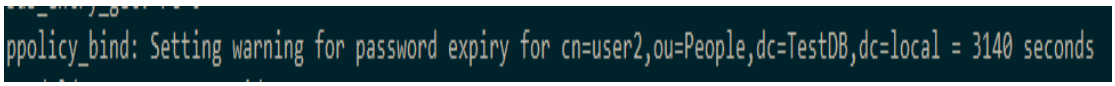
※如果 pwdLockout 设定为 FALSE 或设定了错误的 pwdPolicySubentry 时，即使出现了 pwdAccountLockedTime 用户也不会被锁定。

※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



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



12.2.6.3 同步密码策略

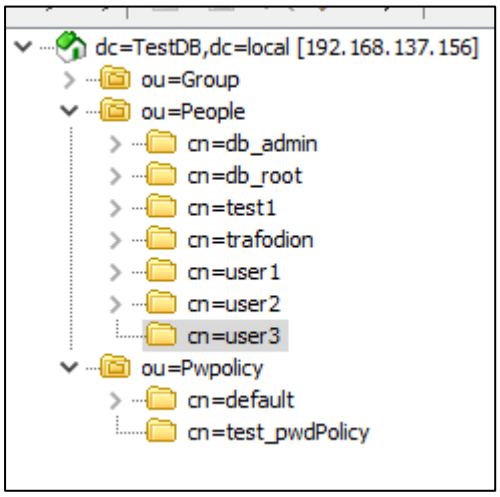
在多主环境下，同步双方均配置好 ppolicy 模块后(此过程需手动进行)，当一台主机修改了密码策略或触发密码策略(比如输错密码)后这份变化会被同步至其它主机之上，则条目在其中一台主机被锁定，其它主机上也被锁定。但对 ppolicy 模块(overlay)的修改是不能同步的，需要手动进行。

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
```

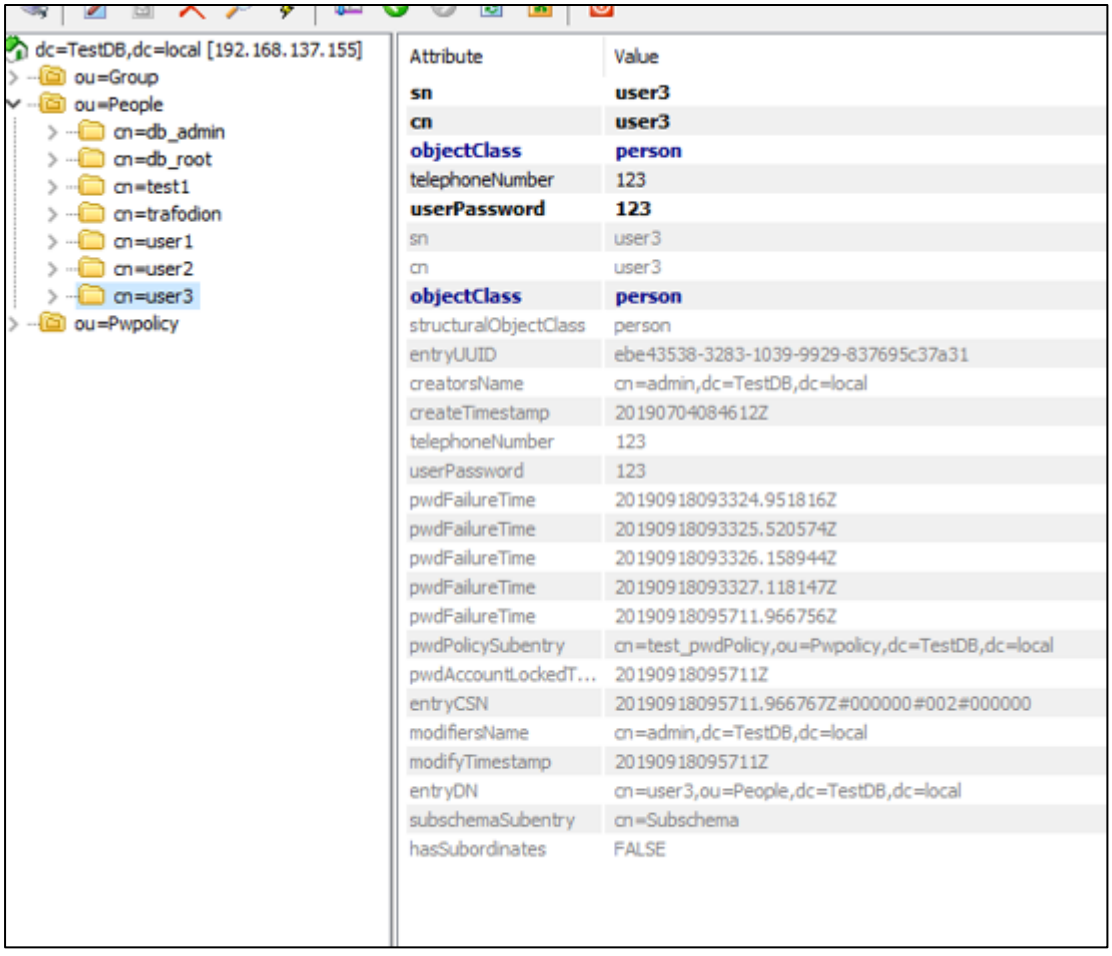
加粗部分为关键点，其余属性依实际环境不同而不同。没有 attrs 时,默认属性为”*,+”表示所有的属性都作为同步的属性。

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。



※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。

12.2.6.4 额外的密码检查

openldap 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 openldap 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 openldap-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/openldap-2.4.44 JAVAHOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.181-7.b13.el7.x86_64 PARAMDIR=/etc/openldap/pqchecker libdir=/usr/lib64/openldap

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 18 9 月 16 16:37 libpqchecker.so -> pqchecker.so.1.2.2
lrwxrwxrwx. 1 ldap ldap 18 9 月 16 16:37 pqchecker.so -> pqchecker.so.1.2.2
-rwxr-xr-x. 1 ldap ldap 78872 9 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
ln -s /usr/lib64/openldap/pqchecker.so.1.2.2 /usr/lib64/openldap/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)

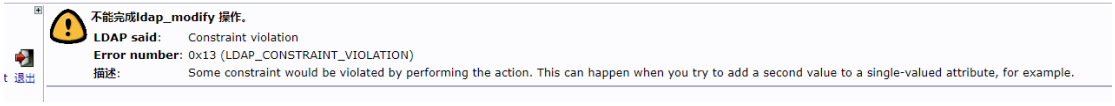
※此模块同 ppolicy，无法对已加密的密码进行检查。

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



ldappasswd 提示

```
[root@testb openldap]# ldappasswd -H ldap:/// -D cn=user2,ou=People,dc=TestDB,dc=local -w aA1_ -S
New password:
Re-enter new password:
Result: Constraint violation (19)
```

Log 中体现为

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

12.2.6.5 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

12.2.6.6 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

12.2.7 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，Openldap 使用 ldap:ldap 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。

Openldap 关联的也更新了其它可选属性。

12.2.8 开启 OpenLDAP access log

此模块记录用户对 Openldap 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>=2018070908345
7Z))
```

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>=201807090834
57Z))"
```

12.2.8.1 将 access log 用于记录

下例为一次 modify 操作的记录:

```
dn: reqStart=20190718095715.000000Z,dc=TestDB,dc=local
objectClass: auditModify
reqStart: 20190718095715.000000Z
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
reqOld: 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。

12.2.8.2 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
 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。

12.2.9 如何删除 overlay

停止 openldap 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。

12.2.10 如何删除数据库

停止 openldap 服务:

以 cn= accesslog 为例，进入/etc/openldap/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。

12.2.11 如何卸载 OpenLDAP

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

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

12.2.12 ldapconfigcheck 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。

12.2.12.1 语法

```
ldapconfigcheck  [<option>]...
<option> ::= --help|-h : display usage information
             -file <config-filename>
```

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:

1. 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。

示例

```
ldapconfigcheck －file myconfigfile
File myconfigfile is valid.
```

如果发现错误，ldapconfigcheck 会显示错误及行号。

12.2.12.2 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：*filename* line-number 行缺少值。
4	文件：*filename* line-number 行中的值超出范围。
5	文件：*filename* 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。

12.2.13 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。

12.2.13.1 语法

```
ldapcheck [option]...
option ::= --help|-h
```

```
            --username=<LDAP-username>
            --password[=<password>]

  or        --groupname=<LDAP-groupname>
            --confignumber=<config-section-number>
            --configname=<config-section-name>
            --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
```

12.2.14 故障排除

问题: 长时间使用后，Openldap 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: Openldap 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/openldap 和 数据库存储目录权限为 ldap:ldap

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

回答:检查/etc/openldap 及其下属目录 和 数据库存储目录权限是否为 ldap:ldap。

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。

12.2.14.1 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/openldap/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 openldap 服务，还原时注意将文件夹\文件的所有权改为 ldap:ldap

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 openldap 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

-l 默认输出为标准输出,-l 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*))" -l /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 openldap 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
/^createTimestamp: /d
/^modifiersName: /d
/^modifyTimestamp: /d
/^structuralObjectClass: /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
```

12.2.14.2 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

6. 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。

12.3 生成服务器证书

EsgynDB 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。

12.3.1 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，EsgynDB 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`。

12.3.2 生成 CSR

1. 使用服务器秘钥生成 CSR[14]，运行命令 sqcertgen gen_csr。
2. 发送 CSR 至 CA[15]。

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。

12.3.3 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。

安装 EsgynDB 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。

12.4 管理用户

AD/LDAP 对任何连接至 EsgynDB 的用户实施强制验证。EsgynDB 支持数据库

[14] 即 Certificate Signing Request，证书注册请求。

[15] 即 Certificate Authorit，证书授权中心。

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

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

启用权限功能后，EsgynDB 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 EsgynDB 时设置）相关。请以用户 DB__ROOT 或 DB__ADMIN 身份登录 EsgynDB，创建所需的 Schema、用户[16]、角色和权限[17]。

[16] 更多关于**如何注册用户**的信息，请参阅《EsgynDB SQL 参考手册》的 **REGISTER USER Statement** 章节。

[17] 更多关于**如何向对象和角色授予权限**的信息，请参阅《EsgynDB SQL 参考手册》的 **GRANT Statement** 章节。

13. 提高安全性

本章讲述以下内容:

13.1 提高 Linux 安全性

更多关于如何提高 Linux 安装过程的安全性，请参阅以下信息。

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

13.2 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
Cloudera CDH 5.x	• CDH 5 安全指南 https://www.cloudera.com/documentation/cdh/5-0-x/CDH5-Security-Guide/CDH5-Security-Guide.html

13.3 提高 Jetty Server 安全性

EsgynDB 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
  http://www.eclipse.org/jetty/documentation/current/configuring-security.html

13.4 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 0

用户 ID 和密码。

13.5 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，EsgynDB 验证配置文件保存在

$TRAF_HOME/sql/scripts/.traf_authentication_config 中。

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 服务器的主机名	`ldap.master.com`	如果提供两个及以上的 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 服务器的端口号	`345`	必须为数值。与 LDAPSSL 相关。 OpenLDAP 标准端口号： • 非安全：389 • 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，请指定搜索用户的区别名（Distinguished Name）。	cn=aaabbb, dc=demo, dc=net	如果本地服务器允许匿名搜索，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，指定下一个 LDAPHostName 的超时时间（单位：秒）。 该参数与 ldap_conf(5)中的 NETWORK_TIMEOUT 类似。 默认值为 30 秒。	20	该参数的值必须为正数或-1，-1 表示无限超时。
LDAPTimelimit	指定在 LDAP 服务器上执行用户名搜索的等待时间。 该参数的值必须为正数。 该参数与 ldap_conf(5)中的 TIMELIMIT 类似。 默认值为 30 秒。	15	在搜索过程中，服务器端可能使用比 LDAPTimelimit 更短的超时时间。
LDAPTimeout	指定在调用同步 LDAP API 后，未收到回应到终止调用之间的超时时间	15	该参数的值必须为正数或-1，-1 表示无限超时。

	（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
RetryCount	建立成功 LDAP 连接的尝试次数。 默认值为 5。如果重试 5 次后全部失败，将返回错误。	10	重试失败操作时，EsgynDB 将尝试连接每个配置 LDAP 服务器，直到操作成功或超过允许的重试次数。
RetryDelay	指定重试之间延迟的时间。 默认值为 2 秒。 更多信息，请参阅 RetryCount。	1	/
PreserveConnection	操作完成后，指定 LDAP 服务器连接保留（YES）或关闭（NO）。 默认值为 NO。	YES	/
RefreshTime	指定重新读取配置文件之前必须等待的时间。 默认值为 1800 秒（30	3600	如果设置为零，则不会读取配置文件。 如果该值为零，则必须重启服务器连接才能使更改

	分钟)。		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 服务器证书文件的位置。文件名可以是绝对路径名或与 $CACERTS_DIR 相关。	`cert.pem`	该属性适用于两种配置。如果一种配置无需证书，则忽略该属性。 该属性必须在 DEFAULTS 组定义。
DefaultSectionName	如果未指定验证类型，则指定由 REGISTER USER 命令分配给用户的配置类型。 初始 trafodion 版本仅支持一种配置。	`LOCAL`	如果指定了 DefaultSectionName 属性，则必须在 .traf_ldapconfig 中定义使用该名称（或等效值）的部分。 合法值为 LOCAL 和 ENTERPRISE。该语法可能会更改。

附录 2. Inspector 工具

安装前，EsgynDB 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 EsgynDB 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       |  o   |    -     |
|             CPU cores           |  w   |    4     |
|            Disk numbers         |  x   |    4     |
|   Free data File System spaces  |  x   |  1000GB  |
|         Free System spaces      |  x   |  200GB   |
|         Total memory size       |  x   |   64GB   |
```

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

附录 3. EsgynDB 和 Hbase 参数优化

1. EsgynDB 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_ESP_PARALLELISM','OFF','ATTEMPT_ESP
_PARALLELISM',1); --关闭 esp 并发
insert into "_MD_".defaults
values('EXPLAIN_IN_RMS','OFF','EXPLAIN_IN_RMS',1);
--关闭 rms 的 query plan 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_CACHE_BLOCKS','ON','HBASE_CACHE_BLOCK
S',1);
insert into "_MD_".defaults
values('HBASE_REGION_SERVER_MAX_HEAP_SIZE','31744',
'HBASE_REGION_SERVER_MAX_HEAP_SIZE',1);
insert into "_MD_".defaults
values('MDAM_SCAN_METHOD','OFF','MDAM_SCAN_METHOD',
1); --关闭 MDAM
insert into "_MD_".defaults
```

```
values('MODE_COMPATIBLE_1','ON','MODE_COMPATIBLE_1'
,1);--rownum 和 rowid 支持
insert                into                "_MD_".defaults
values('VARCHAR_PARAM_DEFAULT_SIZE','8000','VARCHAR
_PARAM_DEFAULT_SIZE',1); --解决参数过长的问题
insert                into                "_MD_".defaults
values('QUERY_TEXT_CACHE','ON','QUERY_TEXT_CACHE',1
); --prepare 编译时间问题
insert                into                "_MD_".defaults
values('QUERY_CACHE','65536','QUERY_CACHE',1); --调
大 query cache, 默认 16MB
insert                into                "_MD_".defaults
values('CANCEL_QUERY_ALLOWED','OFF','CANCEL_QUERY_A
LLOWED',1);
insert                into                "_MD_".defaults
values('HBASE_DATA_BLOCK_ENCODING_OPTION','FAST_DIF
F','HBASE_DATA_BLOCK_ENCODING_OPTION',1); --建表默认
encoding
insert                into                "_MD_".defaults
values('HBASE_COMPRESSION_OPTION','SNAPPY','HBASE_C
OMPRESSION_OPTION',1); --建表默认压缩格式
insert                into                "_MD_".defaults
values('HBASE_MEMSTORE_FLUSH_SIZE_OPTION','10737418
24','HBASE_MEMSTORE_FLUSH_SIZE_OPTION',1); --建表默
认 flush 大小
insert                into                "_MD_".defaults
values('TRAF_DEFAULT_COL_CHARSET','UTF8','TRAF_DEFA
ULT_COL_CHARSET',1); --建表默认编码为 UTF8
insert                into                "_MD_".defaults
values('TRAF_COL_LENGTH_IS_CHAR','OFF','TRAF_COL_LE
```

```
NGTH_IS_CHAR',1); --建表默认为 bytes 而非 chars
insert                into                "_MD_".defaults
values('DYNAMIC_PARAM_DEFAULT_CHARSET','UTF8','DYNA
MIC_PARAM_DEFAULT_CHARSET',1);  --支持中文参数
insert                into                "_MD_".defaults
values('TRAF_ENABLE_METADATA_LOAD_IN_CACHE','ON','T
RAF_ENABLE_METADA',1);  -- preload metadata
```

3) dcs-site.xml

添加:

```
    <property>
                    <name>dcs.server.user.program.s
tatistics.enabled</name>
        <value>false</value>
    </property>
    <property>
                    <name>dcs.master.port.range</na
me>
        <value>200</value>
    </property>
```

2. JDBC 端修改内容

JDBC URL 配置如下

```
jdbc:t4jdbc://10.10.12.25:23400/:schema=V7FAT;maxSt
atements=400;connectionTimeout=7200
```

3. HBase 配置修改

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

附录 4. EsgynDB 在线增加节点

1. 在 HDP/CDH 中手动增加 HBase Regionserver 节点、HDFS Datanode 节点和 Yarn/MapReduce 节点[18]。

2. 增加 EsgynDB 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
```

[18] 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) 2 minute(s) 26 second(s)
***[INFO]: Running dcs setup on all node(s)
[suyan02,suyan03,
suyan04,suyan05,suyan06,suyan07,suyan08,suyan17,suyan1
8,suyan19,suyan20,suyan21,suyan22,suyan23,suyan24] ...
TASK: DCS/REST Setup
**********************************
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

```
*** Checking Trafodion Environment ***
Checking if processes are up.
Checking attempt: 1; user specified max: 2. Execution
time in seconds: 0.
The Trafodion environment 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

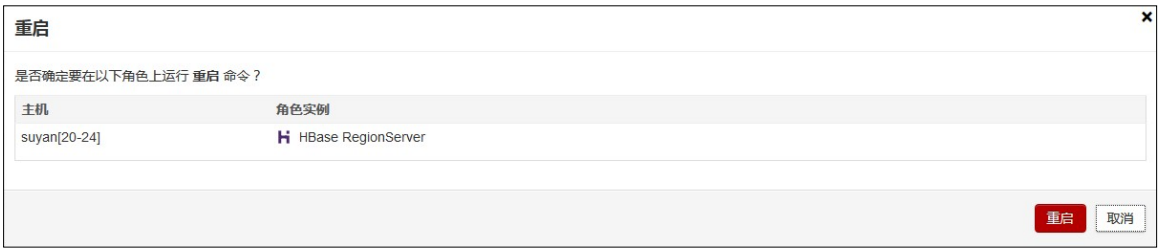
```
[$Z000EJ2] Shell/shell Version 1.0.1 EsgynDB Release
2.5.x (Build release [centos], date 09Nov17)
[$Z000EJ2] %node info
[$Z000EJ2] Logical Nodes       = 23
[$Z000EJ2] Physical Nodes      = 23
[$Z000EJ2] Spare Nodes       = 0
[$Z000EJ2] Available spares= 0
[$Z000EJ2] NID Type      State  Processors     #Procs
```

```
[$Z000EJ2]      PNID     State    #Cores     MemFree
SwapFree  CacheFree   Name
[$Z000EJ2] --- ---- --- ---- ------- ---------   ----
[$Z000EJ2] 000 Any    Up        2        109
[$Z000EJ2]      000    Up       40 5761536  66999072
277981484 suyan02
[$Z000EJ2] 001 Any    Up        2        106
[$Z000EJ2]      001    Up       40 2597588  66975220
281136344 suyan03
[$Z000EJ2] 002 Any    Up        2        106
[$Z000EJ2]      002    Up       40 1839660  66939476
280069928 suyan04
…
[$Z000EJ2]      009  Up   40   2377144 66977628
280325836 suyan11
[$Z000EJ2] 010 Any  Up    2        106
[$Z000EJ2]      010  Up   40   1250020    67036444
279768248 suyan12
[$Z000EJ2] 011 Any  Up    2        106
[$Z000EJ2]      011  Up   40   1271592    66985060
280740440 suyan13
[$Z000EJ2] 012 Any  Up    2        106
[$Z000EJ2]      012  Up   40   2748056    66971736
280183032 suyan14
[$Z000EJ2] 013 Any  Up    2        106
[$Z000EJ2]      013  Up   40  41488500 67079948
282424384 suyan15
[$Z000EJ2] 014 Any  Up    2        106
[$Z000EJ2]      014  Up   40   1519468    66960032
281095480 suyan16
[$Z000EJ2] 015 Any  Up    2        106
[$Z000EJ2]      015  Up   40   1548692    66978064
280175188 suyan17
[$Z000EJ2] 016 Any  Up    2        106
```

```
[$Z000EJ2]       016  Up    40    3712684     66914132
281733864 suyan18
[$Z000EJ2] 017 Any  Up     2         106
[$Z000EJ2]       017  Up    40    2621536     66980652
280225512 suyan19
[$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。



附录 5. EsgynDB 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@esgvm-test scripts]$ sqcheck
*** Checking Trafodion Environment ***
Checking if processes are up.
Checking attempt: 1; user specified max: 2. Execution
time in seconds: 4.
The Trafodion environment is not up at all, or
partially up and not operational. Check the logs.
Process         Configured    Actual   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

```
node-id=0;node-name-suyan02;cores=0-
39;processors=2;roles=connection,aggregation,storage
node-id=1;node-name-suyan03;cores=0-
39;processors=2;roles=connection,aggregation,storage
node-id=2;node-name-suyan04;cores=0-
39;processors=2;roles=connection,aggregation,storage
node-id=3;node-name-suyan05;cores=0-
39;processors=2;roles=connection,aggregation,storage
node-id=4;node-name-suyan06;cores=0-
39;processors=2;roles=connection,aggregation,storage
node-id=5;node-name-suyan07;cores=0-
```

```
39;processors=2;roles=connection,aggregation,storage
node-id=6;node-name-suyan08;cores=0-
39;processors=2;roles=connection,aggregation,storage
node-id=7;node-name-suyan09;cores=0-
39;processors=2;roles=connection,aggregation,storage
node-id=8;node-name-suyan10;cores=0-
39;processors=2;roles=connection,aggregation,storage
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：

```
node-id=0;node-name-suyan02;cores=0-
39;processors=2;roles=connection,aggregation,storage
node-id=1;node-name-suyan03;cores=0-
39;processors=2;roles=connection,aggregation,storage
node-id=2;node-name-suyan04;cores=0-
39;processors=2;roles=connection,aggregation,storage
node-id=3;node-name-suyan05;cores=0-
39;processors=2;roles=connection,aggregation,storage
node-id=4;node-name-suyan06;cores=0-
39;processors=2;roles=connection,aggregation,storage
node-id=5;node-name-suyan07;cores=0-
39;processors=2;roles=connection,aggregation,storage
node-id=6;node-name-suyan08;cores=0-
39;processors=2;roles=connection,aggregation,storage
node-id=7;node-name-suyan09;cores=0-
39;processors=2;roles=connection,aggregation,storage
node-id=8;node-name-suyan10;cores=0-
39;processors=2;roles=connection,aggregation,storage
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
```

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
,suyan09,suyan10,suyan11,suyan12,suyan13,suyan14,suyan1
5,suyan16,suyan17,suyan18,suyan19

Creating directories on cluster nodes
/usr/bin/pdsh -R exec -w
suyan02,suyan03,suyan04,suyan05,suyan06,suyan07,suyan08
,suyan09,suyan10,suyan11,suyan12,suyan13,suyan14,suyan1
5,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
,suyan09,suyan10,suyan11,suyan12,suyan13,suyan14,suyan1
5,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
,suyan09,suyan10,suyan11,suyan12,suyan13,suyan14,suyan1
5,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
,suyan09,suyan10,suyan11,suyan12,suyan13,suyan14,suyan1
5,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
,suyan09,suyan10,suyan11,suyan12,suyan13,suyan14,suyan1
5,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
,suyan09,suyan10,suyan11,suyan12,suyan13,suyan14,suyan1
5,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 processes are up.
Checking attempt: 1; user specified max: 2. Execution
time in seconds: 0.

The Trafodion environment is up!

Process      Configured      Actual      Down
---------   -----------     -------    ------
DTM          18              18
RMS          36              36
DcsMaster     1              1
Dcsserver    18              18
mxosrvr      1764            1226        538
Restserver   1               1

Startup time 0 hour(s) 7 minute(s) 57 second(s)
```

10. 检查 Trafodion 运行状况。

输入

```
sqcheck
```

输出

```
*** Checking Trafodion Environment ***

Checking if processes are up.
Checking attempt: 1; user specified max: 2. Execution
time in seconds: 1.
```

```
The Trafodion environment is up!

Process     Configured       Actual      Down
---------   -----------     -------     --------
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>
    <name>dcs.zookeeper.property.clientPort</name>
    <value>2181</value>
  </property>
   <property>
    <name>dcs.zookeeper.quorum</name>
    <value>gy08.esgyncn.local,gy07.esgyncn.local,gy09.esg
yncn.local</value>
  </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</nam
e>
    <value>10.10.12.252</value>
  </property>
```

(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-clu-n011 conf]$ dcscheck

Cluster Configuration     : HA
DcsMaster virtual address : 10.10.23.120

Configured  DcsMaster(s)      :  esggy-clu-n011.esgyn.cn
esggy-clu-n002.esgyncn.local     esggy-clu-n012.esgyn.cn
esggy-clu-n013.esgyn.cn
Active DcsMaster(s)       : esggy-clu-n011
DcsMaster listen port     : 23400

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 dcs-site.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目录】
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

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

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】

主机组件	hadoop-hdt01	hadoop-hdt02	hadoop-hdt03	hadoop-hdt04	hadoop-hdt05	hadoop-hdt06	hadoop-hdt07	hadoop-hdt08	hadoop-hdt09	hadoop-hdt10	hadoop-hdt11
HDFS NN/SNN	NN	NN									
HDFS Journal Node	Y	Y	Y								
HDFS DN				Y	Y	Y	Y	Y	Y	Y	Y
Hbase Master	Y	Y									
Hbase RS				Y	Y	Y	Y	Y	Y	Y	Y
Zookeeper	Y	Y	Y								
Yarn Master	Y	Y									
Yarn NM				Y	Y	Y	Y	Y	Y	Y	Y
Hive Metastore	Y	Y									
Hive2 Server	Y	Y		Y	Y	Y	Y	Y	Y	Y	Y

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】

2. HDSF角色配置规划

HDSF 角色分配:

1) Balancer Default Group 1
2) DataNode Default Group 1→N
3) NameNode Default Group 1
4) SecondaryNameNode Default Group 1

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	角色组
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*/

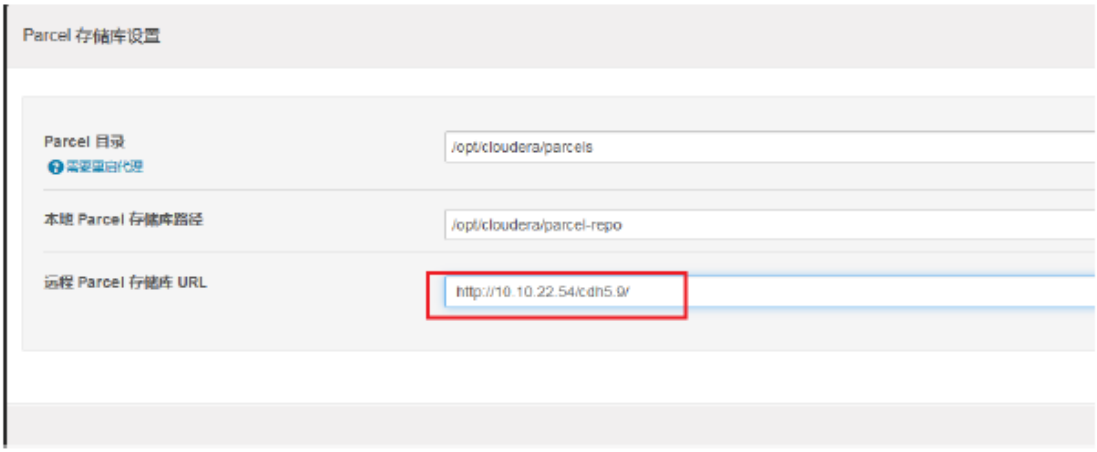
4. 安装CDH 节点

登陆 CDH CM，进行安装配置：http://IP:7180/

1) 为群集制定主机

2)指定要安装的主机

3) 远程 Parcel 存储库 URL：指定上面配置的 CDH 或者远程资源库



4）填写 Cloudera 的地址

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。

输入 root 的密码，或者具有 sudo 权限用户的密码

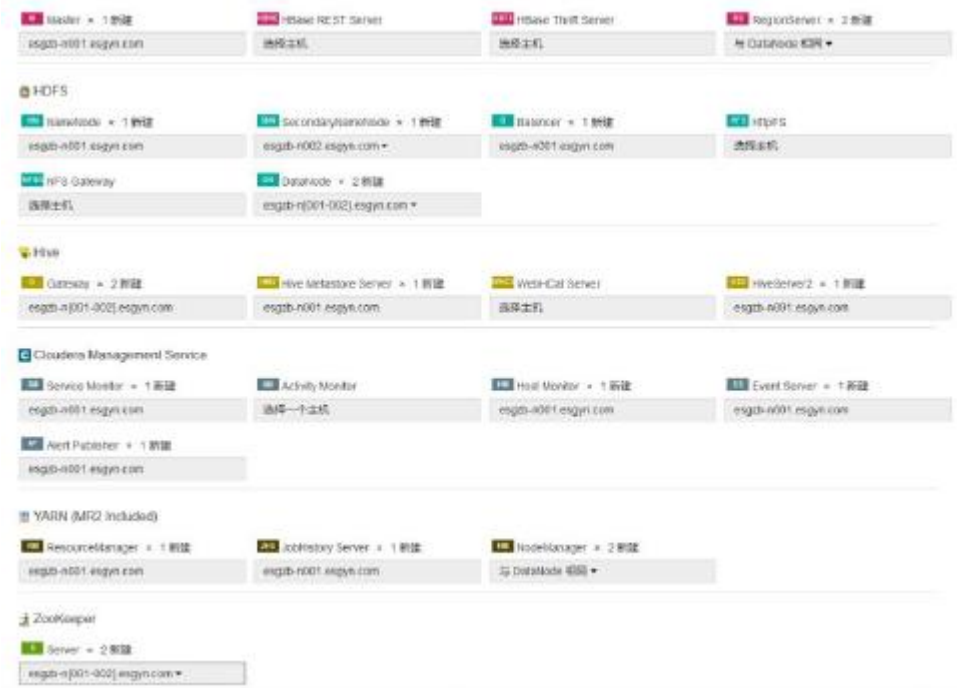
自定义安装 Zookeeper，HDFS，Hbase，Yarn，Hive 服务

选中自定义服务

zookeeper 选择奇数个节点，推荐 3 个。

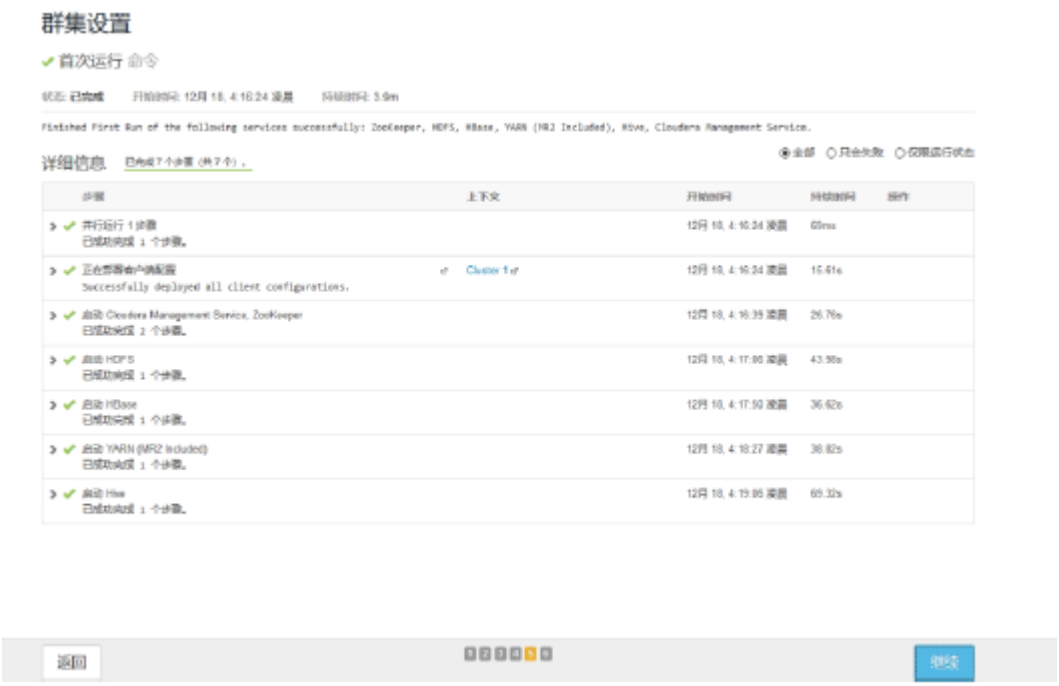
数据库设置【选择之前安装的 mysql 或则系统自带数据库】

添加 HDFS、HBase、Hive、Yarn Master、Zookeeper 等服务



注意

此处为告警，尽量做到没有告警提示，否则可以忽略



查看界面展现效果

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，EsgynDB 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. Predload 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。

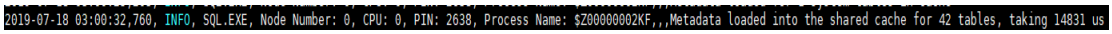
```
create schema sch1 stored desc
```

4) Preload 生效

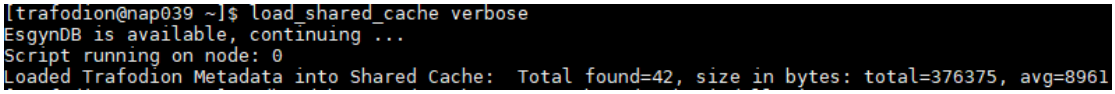
有两种生效方式:

方式 1）重启 EsgynDB，执行 sqstop 然后执行 sqstart。

可通过 sql 日志 trafodion.sql_#.log 查看，



方式 2）使用 trafodion 用户登录，在终端执行“load_shared_cache verbose”



5) 检查是否生效

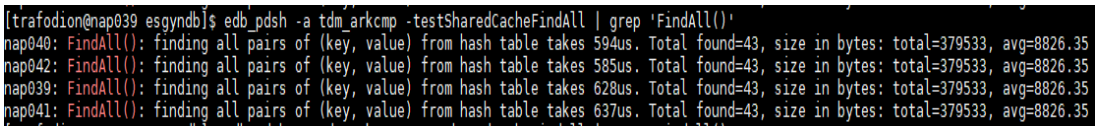
使用 trafodion 用户登录，在终端执行以下命令:

“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。



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=TRAFODION.SEABASE.USER_TEST1
nap042: key=TRAFODION.SEABASE.USER_TEST1
nap041: key=TRAFODION.SEABASE.USER_TEST1
nap040: key=TRAFODION.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 EsgynDB 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 EsgynDB。

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. EsgynDB端口列表

产品	组件	端口号	是否向外部网络开放	是否向内部网络EsgynDB节点间开放	说明
EsgynDB	DCS Master Listen	23400	Yes	Yes	
EsgynDB	DCS Master UI	24400	Yes	Yes	
EsgynDB	mxsosrvr	23401 : 23499	Yes	Yes	基于并发要求和最大配置mxosrvr 数的端口范围
EsgynDB	REST Server	4200/4201	Yes	Yes	
EsgynDB	DB Manager	4205/4206	No	Yes	
EsgynDB	Esgyn exporters	23300/23301	No	Yes	
EsgynDB	Filebeat	5044	No	Yes	
EsgynDB	mds	8989	No	Yes	
EsgynDB	Monitor (MONITOR_ COMM_PORT)	23399	No	Yes	
EsgynDB	Monitor (MONITOR_ SYNC_PORT)	23380	No	Yes	
EsgynDB	Monitor (NS_ COMM_PORT)	23370	No	Yes	
EsgynDB	Monitor (NS_M2N_ COMM_PORT)	23360	No	Yes	
EsgynDB	Monitor (NS_M2N_ COMM_PORT)	23350	No	Yes	
EsgynDB	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	30002	No	Yes	部署在管理节点上
OM HA	logstash	30003	No	Yes	部署在管理节点上

2. 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	HTTP	10110
Telemetry Publisher	HTTP (Debug)	10111
Cloudera Manager Server	Avro (RPC)	7182
Cloudera Manager Server	Embedded PostgreSQL database	7432
Cloudera Manager Server	Peer-to-peer parcel distribution	7190, 7191
Cloudera Manager Agent Listening Port	Custom protocol	9000
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	HTTP	7186
Cloudera Navigator Audit Server	HTTP (Debug)	8089