

云备份最佳实践

云备份最佳实践

文档版本 01
发布日期 2020-11-12



版权所有 © 华为技术有限公司 2020。保留一切权利。

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

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

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

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

目录

1 通过自定义脚本实现数据库备份	1
1.1 通过自定义脚本实现 MySQL 一致性备份.....	1
1.1.1 准备工作.....	1
1.1.2 详细步骤.....	2
1.2 通过自定义脚本实现 SAP HANA 一致性备份.....	2
1.2.1 准备工作.....	3
1.2.2 详细步骤.....	3
1.3 通过自定义脚本实现其它 Linux 应用的一致性备份.....	4
1.3.1 场景介绍.....	4
1.3.2 开发冻结脚本.....	5
1.3.3 开发解冻脚本.....	6
1.4 自定义脚本问题定位方法.....	6
1.5 验证数据库备份结果 (Linux).....	7
1.6 验证数据库备份结果 (Windows).....	8
1.7 保护 Failover Cluster 模式下的 SQL Server.....	9
1.8 保护 Always on Availability Groups 模式下的 SQL Server.....	9
A 修订记录	10

1 通过自定义脚本实现数据库备份

- 1.1 通过自定义脚本实现MySQL一致性备份
- 1.2 通过自定义脚本实现SAP HANA一致性备份
- 1.3 通过自定义脚本实现其它Linux应用的一致性备份
- 1.4 自定义脚本问题定位方法
- 1.5 验证数据库备份结果 (Linux)
- 1.6 验证数据库备份结果 (Windows)
- 1.7 保护Failover Cluster模式下的SQL Server
- 1.8 保护Always on Availability Groups模式下的SQL Server

1.1 通过自定义脚本实现 MySQL 一致性备份

1.1.1 准备工作

本章节以SuSE 11 SP3操作系统下MySQL 5.5单机版为例，介绍如何通过自定义脚本来冻结、解冻MySQL数据库，以实现对于MySQL数据库的数据库备份。

场景介绍

某企业购买了云服务器，并在云服务器中安装了MySQL 5.5数据库用于存放业务数据。随着数据量的增加，之前的崩溃一致性保护已经满足不了RTO、RPO的要求，决定采用应用一致性备份，减小RTO与RPO。

数据准备

表 1-1 数据准备

准备项	说明	示例
MySQL用户名	连接MySQL数据库时使用的用户名	root

准备项	说明	示例
MySQL密码	连接MySQL数据库时使用的密码	Example@123

1.1.2 详细步骤

步骤1 加密MySQL密码，供自定义脚本使用

1. 登录MySQL服务器，输入`cd /home/rdadmin/Agent/bin/`，进入Agent目录。
2. 执行`/home/rdadmin/Agent/bin/agentcli encpwd`，回显如下：

```
Enter password:
```

输入MySQL密码，并按“Enter”，屏幕上就会打印出加密后的密码，将其拷贝到剪贴板中。

步骤2 执行`cd /home/rdadmin/Agent/bin/thirdparty/ebk_user`，进入自定义脚本目录，然后执行`vi mysql_freeze.sh`，打开MySQL示例冻结脚本。

将下图所示的MYSQL_USER与MYSQL_PASSWORD修改为实际值，其中MYSQL_PASSWORD为**步骤1**的屏幕输出。

```
#####  
#Importnata note  
#Please change this parameters according to your Mysql system configuration!!!  
MYSQL_USER="root"  
MYSQL_PASSWORD="0000000100000001000000000000000500000001000000017334dcf36ace871b1  
0001000000000000000804000000010000000129678894e3225391233bac37497d3728000000000000  
#####
```

也可以使用sed命令来直接进行修改：

```
sed -i 's/^MYSQL_PASSWORD=.*MYSQL_PASSWORD="XXX"/' mysql_freeze.sh  
mysql_unfreeze.sh
```

，其中XXX为步骤1中打印出的密码。

此操作会同时修改冻结解冻脚本，所以无需再执行步骤3。

步骤3 执行`vi mysql_unfreeze.sh`，打开MySQL示例解冻脚本，修改此脚本中的用户名和密码。

`mysql_unfreeze.sh`与`mysql_freeze.sh`脚本实现了基本的数据库冻结与解冻操作，如果你在冻结、解冻时有其它额外步骤需要执行，可以自行在其中进行修改。详细说明请参见 [1.3 通过自定义脚本实现其它Linux应用的一致性备份](#)。

注意

MySQL的冻结是通过FLUSH TABLES WITH READ LOCK指令来实现的，此指令不会触发bin log刷盘操作，如果开启了bin log，且sync_binlog参数不为1，则可能出现保存的备份映像中部分SQL操作未记录到bin log的情况，如果bin log也需要完整保护，请设置sync_binlog=1。

----结束

1.2 通过自定义脚本实现 SAP HANA 一致性备份

1.2.1 准备工作

本章节以 SuSE 11 SP4 for SAP操作系统下HANA 2.0单机版为例，介绍如何通过自定义脚本来冻结、解冻HANA数据库，以实现对HANA数据库的数据库备份。

场景介绍

某企业购买了云服务器，并在上面安装了HANA 2.0单机版数据库，用于存放业务数据，随着数据量的增加，之前的崩溃一致性保护已经满足不了RTO、RPO的要求，决定采用应用一致性备份，减小RTO与RPO。

数据准备

表 1-2 数据准备

准备项	说明	示例
HANA用户名	连接HANA SYSTEMDB 数据库时使用的用户名	system
HANA密码	连接HANA SYSTEMDB 数据库时使用的密码	Example@123
HANA实例编号	连接HANA数据库时使用的实例编号	00
HANA SID	连接HANA数据库时使用的SID	WXJ

1.2.2 详细步骤

步骤1 加密HANA用户密码，供自定义脚本使用

1. 登录HANA服务器，输入`cd /home/rdadmin/Agent/bin/`，进入Agent目录。
2. 执行`/home/rdadmin/Agent/bin/agentcli encpwd`，回显如下：

```
Enter password:
```

输入HANA用户的密码，并按“Enter”，屏幕上就会打印出加密后的密码，将其拷贝到剪贴板中。

执行`cd /home/rdadmin/Agent/bin/thirdparty/ebk_user`，进入自定义脚本目录，执行`vi hana_freeze.sh`，打开HANA示例冻结脚本。

步骤2 将下图所示的HANA_USER HANA_PASSWORD INSTANCE_NUMBER DB_SID修改为实际值，其中HANA_PASSWORD 为步骤1的屏幕输出。

```

#*****
#Importnata note
#Please change this parameters according to your HANA system configuration!!!

HANA_USER="system"
HANA_PASSWORD="0000000100000001000000000000005000000010000000161b3258428fbdf
00100000000000003040000000100000001c9562ef9b7f838e984dc3d1080975be00000000000
INSTANCE_NUMBER="00"
DB_SID="WXJ"

#*****

```

也可以使用sed命令来直接进行修改：

```
sed -i 's/^HANA_USER=.* /HANA_USER="XXX" /' hana_freeze.sh  
hana_unfreeze.sh , 其中XXX为数据库用户名。
```

```
sed -i 's/^HANA_PASSWORD=.* /HANA_PASSWORD="XXX" /' hana_freeze.sh  
hana_unfreeze.sh , 其中XXX为步骤1中打印出的密码。
```

```
sed -i 's/^INSTANCE_NUMBER=.* /INSTANCE_NUMBER="XXX" /' hana_freeze.sh  
hana_unfreeze.sh , 其中XXX为数据库实例编号。
```

```
sed -i 's/^DB_SID=.* /DB_SID="XXX" /' hana_freeze.sh hana_unfreeze.sh , 其中  
XXX为数据库SID。
```

此操作会同时修改冻结解冻脚本，所以无需再执行[步骤3](#)。

步骤3 执行vi hana_unfreeze.sh，打开HANA示例解冻脚本，修改此脚本中的用户名、密码、实例编号与SID

hana_freeze.sh与hana_unfreeze.sh脚本实现了基本的数据库冻结与解冻操作，如果你在冻结、解冻时有其它额外步骤需要执行，可以自行在其中进行修改。详细说明请参见 [1.3 通过自定义脚本实现其它Linux应用的一致性备份](#)

警告

冻结SAP HANA数据库时，按照SAP官方建议，需要冻结Data卷的XFS文件系统，否则可能出现数据不一致的问题。在此示例脚本中，将会查询出HANA使用的Data卷挂载点，并用xfs_freeze 命令进行冻结。

如果HANA系统未按照SAP官方建议使用一个独立分区来存放Data卷数据，而是与系统卷共用一个分区，则请修改hana_freeze.sh脚本，注释掉xfs_freeze相关行，防止整个系统都被冻结，但此时可能出现备份数据不一致的问题。

----结束

1.3 通过自定义脚本实现其它 Linux 应用的一致性备份

1.3.1 场景介绍

在Linux下，如果有其它应用需要一致性备份，可以编写自己的冻结、解冻脚本，来实现应用的保护。自定义脚本需放置在/home/rdadmin/Agent/bin/thirdparty/ebk_user目录中，供Agent在备份过程中调用。

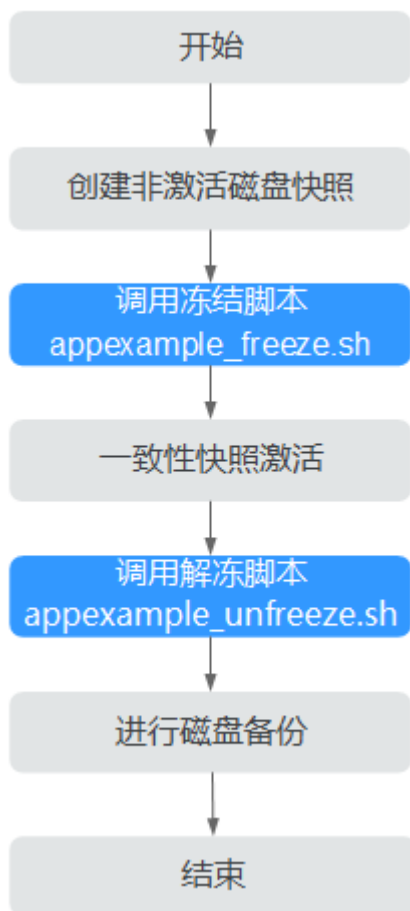
下面以一个虚构的应用appexample为例，来进行说明。

appexample是一款新的数据库，它对外提供了appexample -freeze与appexample -unfreeze两个命令来实现冻结与解冻。

用户需要开发自己的appexample_freeze.sh与appexample_unfreeze.sh脚本，供备份Agent调用以实现一致性备份。在备份过程中，会先调用appexample_freeze.sh脚本来冻结IO，冻结成功后，会进行磁盘的一致性快照激活，保证备份的数据是一致的，最后再调用appexample_unfreeze.sh脚本解冻IO。

整体流程如[图1-1](#)所示：

图 1-1 数据库备份流程图



1.3.2 开发冻结脚本

appexample_freeze.sh示例如下：

```
#!/bin/sh
AGENT_ROOT_PATH=$1 #Agent程序调用脚本时，传入的根目录，日志函数等会使用此变量，请不要改名
PID=$2 #Agent程序调用脚本时，传入的PID数字，用于结果的输出，请不要改名
. "${AGENT_ROOT_PATH}/bin/agent_func.sh"#引用脚本框架，提供了日志，加解密等功能
#结果处理函数，用于将结果写入到文件中，供脚本调用者获取返回值。
#入参 $1: 0表示成功，1表示失败
#无返回值
#RESULT_FILE在agent_func.sh中进行了定义
function ExitWithResult()
{
    Log "[INFO]:Freeze result is $1."
    echo $1 > ${RESULT_FILE}
    chmod 666 ${RESULT_FILE}
    exit $1
}
function Main()
{
    Log "*****"
    Log "[INFO]:Begin to freeze appexample."
    #查找appexample是否存在，如果appexample不存在，则返回0，退出
    #在冻结IO步骤中，Agent程序会依次调用每个冻结脚本，如果一个失败，总体就会失败。所以为了防止干扰
    #其他程序的冻结过程，找不到appexample时，应返回0
    which appexample
    if [ $? -ne 0 ]
    then
        Log "[INFO]:appexample is not installed."
```



```
        ExitWithResult 0
    fi
    #调用实际的冻结命令
    appexample -freeze
    if [ $? -ne 0 ]
    then
        Log "[INFO]:appexample freeze failed."
        #冻结失败，记录结果并退出
        ExitWithResult 1
    fi
    Log "[INFO]:Freeze appexample success."
    #冻结成功，记录结果并退出
    ExitWithResult 0
}
Main
```

1.3.3 开发解冻脚本

appexample_unfreeze.sh示例如下：

```
#!/bin/sh
AGENT_ROOT_PATH=$1 #Agent程序调用脚本时，传入的的根目录，日志函数等会使用此变量，请不要改名
PID=$2 #Agent程序调用脚本时，传入的PID数字，用于结果的输出，请不要改名
. "${AGENT_ROOT_PATH}/bin/agent_func.sh"#引用脚本框架，提供了日志，加解密等功能
#结果处理函数，用于将结果写入到文件中，供脚本调用者获取返回值。
#入参 $1: 0表示成功，1表示失败
#无返回值
#RESULT_FILE在agent_func.sh中进行了定义
function ExitWithResult()
{
    Log "[INFO]:Freeze result is $1."
    echo $1 > ${RESULT_FILE}
    chmod 666 ${RESULT_FILE}
    exit $1
}
function Main()
{
    Log "*****"
    Log "[INFO]:Begin to freeze appexample."
    #查找appexample是否存在，如果appexample不存在，则返回0，退出
    #在解冻IO步骤中，Agent程序会依次调用每个解冻脚本，如果一个失败，总体就会失败。所以为了防止干扰
    #其他程序的解冻过程，找不到appexample时，应返回0
    which appexample
    if [ $? -ne 0 ]
    then
        Log "[INFO]:appexample is not installed."
        ExitWithResult 0
    fi
    #调用实际的解冻命令
    appexample -unfreeze
    if [ $? -ne 0 ]
    then
        Log "[INFO]:appexample freeze failed."
        #解冻失败，记录结果并退出
        ExitWithResult 1
    fi
    Log "[INFO]:Freeze appexample. success"
    #解冻成功，记录结果并退出
    ExitWithResult 0
}
Main
```

1.4 自定义脚本问题定位方法

如果自定义脚本存在缺陷，可能导致数据库备份失败，此时可以打开/home/rdadmin/Agent/log/thirdparty.log，查看日志进行定位。

图1-2为一个冻结MySQL数据库失败时的日志样例

图 1-2 日志示例

```
18-09-13--22:30:10:[30243][root] *****
18-09-13--22:30:10:[30243][root] [INFO]:Begin to freeze mysql.
Id User Host db Command Time State Info
20 root localhost test123 Sleep 1063 NULL
21 root localhost test123 Sleep 1066 NULL
24 root localhost NULL Query 23 User sleep select 1 and sleep(60)
27 root localhost NULL Query 0 NULL show processlist
18-09-13--22:30:10:[30243][root] [ERROR]:MySQL already been frozen
18-09-13--22:30:10:[30243][root] [INFO]:mysql freeze result is 1.
18-09-14--10:07:54:[71621][root] *****
```

第一列 18-09-13--22:30:10 为日志记录时间

第二列 [30243] 为脚本的PID编号

第三列 [root] 为脚本的执行用户

第四列 [INFO] 或 [ERROR] 为日志级别

一般脚本调用失败时，打开日志文件，找到相应时间点的ERROR即可初步确定问题原因。例如图1-2中的错误就是因为MySQL已经处于冻结状态，再次冻结，就会出错。

1.5 验证数据库备份结果 (Linux)

使用自定义脚本实现数据库备份完成后，可以通过如下操作验证数据库备份结果是否成功。本章节以MY SQL数据库为例进行验证。

步骤1 登录MY SQL数据库，创建新的数据库。

步骤2 创建数据库成功后，创建存储过程，可以参考图1-3。

图 1-3 创建存储过程

```
DELIMITER //
CREATE DEFINER='root'@'localhost' PROCEDURE `test_insert_xuwei3`()
BEGIN
declare i int;
declare v float;
set i = 0;
while i < 10000000
do
select RAND()*100 into v;
insert into xuwei1_test values(i, 'xxxxxx', now());
set i = i+1;
end while;
END
//
DELIMITER ;
```

步骤3 进入云服务器备份控制台，对目标弹性云服务器创建数据库备份，并勾选数据库备份。

步骤4 待备份完成后，进入/home/radmin/Agent/log/rdagent.log，查看冻结、解冻日志，确定冻结解冻时间。

步骤5 使用新创建的数据库备份恢复目标弹性云服务器。恢复成功后，登录云服务器和数据库，查看表中最后一条插入数据对应的时间。

步骤6 对比**步骤5**日志显示的VSS冻结成功时间和**步骤4**的时间。冻结成功之前会停止插入数据，所以**步骤5**的时间比**步骤4**早。若**步骤5**的时间比**步骤4**早，则表示引用一致性备份成功。

----结束

1.6 验证数据库备份结果 (Windows)

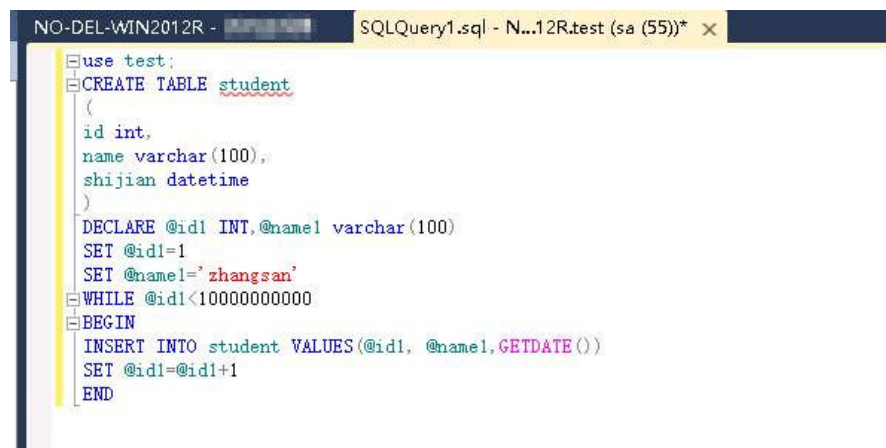
使用自定义脚本实现数据库备份完成后，可以通过如下操作验证数据库备份结果是否成功。本章节以SQL_SERVER数据库为例进行验证。

操作步骤

步骤1 登录SQL_SERVER数据库，创建新的数据库。

步骤2 创建数据库成功后，创建存储过程，可以参考图1-4。

图 1-4 创建存储过程



```
use test;
CREATE TABLE student
(
    id int,
    name varchar(100),
    shijian datetime
)
DECLARE @id1 INT,@name1 varchar(100)
SET @id1=1
SET @name1='zhangsan'
WHILE @id1<10000000000
BEGIN
    INSERT INTO student VALUES(@id1, @name1,GETDATE())
    SET @id1=@id1+1
END
```

步骤3 进入云备份控制台，对目标弹性云服务器创建数据库备份，并勾选数据库备份。

步骤4 待备份完成后，进入Cloud Server Backup Agent-WIN64\log\ rdagent.txt文件，查看冻结、解冻日志，确定冻结解冻时间。如图中所示的17:28:51。

图 1-5 查看日志

```
[2018-11-14 17:28:46][0x0000531600001536][2052][SYSTEM][INFO][Requester.cpp,1369]Start snap shot set.
[2018-11-14 17:28:46][0x0000531600001536][2052][SYSTEM][INFO][Requester.cpp,1372]Add to snapshot set.
[2018-11-14 17:28:46][0x0000531600001536][2052][SYSTEM][INFO][Requester.cpp,1375]Prepare for backup.
[2018-11-14 17:28:46][0x0000531600001536][2052][SYSTEM][INFO][Requester.cpp,1261]Begin prepare for backup.
[2018-11-14 17:28:46][0x0000531600001536][2052][SYSTEM][INFO][Requester.cpp,1272]Prepare for backup succ.
[2018-11-14 17:28:46][0x0000531600001536][2052][SYSTEM][INFO][Requester.cpp,1378]Do snapshot set.
[2018-11-14 17:28:46][0x0000531600001536][2052][SYSTEM][INFO][Requester.cpp,1278]Begin create the shadow (Do SnapShot Set).
[2018-11-14 17:28:51][0x0000531600001536][2052][SYSTEM][INFO][Requester.cpp,1317]Create the shadow (Do SnapShot Set) succ.
[2018-11-14 17:28:51][0x0000531600001536][2052][SYSTEM][INFO][Requester.cpp,227]Freeze volume succ.
[2018-11-14 17:28:51][0x0000531600001536][2052][SYSTEM][INFO][Requester.cpp,180]Freeze file sys, succ.
[2018-11-14 17:28:51][0x0000531600001536][2052][SYSTEM][INFO][App.cpp,383]Vss freeze success.
[2018-11-14 17:28:51][0x0000531600001536][2052][SYSTEM][INFO][AppPlugin.cpp,157]Freeze app succ.
[2018-11-14 17:28:51][0x0000531600001536][4872][SYSTEM][INFO][MessageProcess.cpp,1034]Json key "loop_time" does not exist.
[2018-11-14 17:28:51][0x0000531600001536][4872][SYSTEM][INFO][FTExceptionHandler.cpp,849]Update monitor obj freeze begin time
[2018-11-14 17:28:52][0x0000531600001536][544][SYSTEM][INFO][Communication.cpp,400]End accept fcgx
[2018-11-14 17:28:52][0x0000531600001536][544][SYSTEM][INFO][Authentication.cpp,104]strClientCertDN: CN=BCManager eBackup Cl
[2018-11-14 17:28:52][0x0000531600001536][544][SYSTEM][INFO][Authentication.cpp,130]Client IP address 100.125.1.142 Auth suc
[2018-11-14 17:28:52][0x0000531600001536][544][SYSTEM][INFO][Communication.cpp,390]Begin accept fcgx
[2018-11-14 17:28:53][0x0000531600001536][2052][SYSTEM][INFO][AppPlugin.cpp,168]Begin unfreeze app.
[2018-11-14 17:28:53][0x0000531600001536][2052][SYSTEM][INFO][App.cpp,392]Begin vss unfreeze.
[2018-11-14 17:28:53][0x0000531600001536][2052][SYSTEM][INFO][Requester.cpp,275]Begin unfreeze all.
[2018-11-14 17:28:53][0x0000531600001536][2052][SYSTEM][INFO][Requester.cpp,1703]Begin wait for async ex.
[2018-11-14 17:28:53][0x0000531600001536][2052][SYSTEM][INFO][Requester.cpp,1733]End wait for async ex, return 0x0004230a (V
[2018-11-14 17:28:53][0x0000531600001536][2052][SYSTEM][INFO][Requester.cpp,1579]VSS async finished.
[2018-11-14 17:28:53][0x0000531600001536][2052][SYSTEM][INFO][Requester.cpp,303]End unfreeze all.
[2018-11-14 17:28:53][0x0000531600001536][2052][SYSTEM][INFO][App.cpp,415]VSS unfreeze success.
[2018-11-14 17:28:53][0x0000531600001536][2052][SYSTEM][INFO][App.cpp,424]Begin vss endbakup.
[2018-11-14 17:28:53][0x0000531600001536][2052][SYSTEM][INFO][Requester.cpp,311]Begin end bakcup.
[2018-11-14 17:29:05][0x0000531600001536][2052][SYSTEM][INFO][Requester.cpp,333]End end bakcup.
[2018-11-14 17:29:05][0x0000531600001536][2052][SYSTEM][INFO][App.cpp,445]Vss endbakup success.
[2018-11-14 17:29:05][0x0000531600001536][2052][SYSTEM][INFO][App.cpp,342]Unfreeze all apps success.
[2018-11-14 17:29:05][0x0000531600001536][2052][SYSTEM][INFO][AppPlugin.cpp,185]Unfreeze app succ.
[2018-11-14 17:29:05][0x0000531600001536][4872][SYSTEM][INFO][MessageProcess.cpp,1034]Json key "loop_time" does not exist.
```

步骤5 使用新创建的数据库备份恢复目标弹性云服务器。恢复成功后，登录云服务器和数据库，查看表中最后一条插入数据对应的时间(17:28:49)的记录。

步骤6 对比**步骤5**日志显示的VSS冻结成功时间和**步骤4**的时间。冻结成功之前会停止插入数据，所以**步骤5**的时间比**步骤4**早。若**步骤5**的时间比**步骤4**早，则表示引用一致性备份成功。

---结束

1.7 保护 Failover Cluster 模式下的 SQL Server

当前云服务器备份只支持单个虚拟机的一致性备份，对于集群数据库暂不支持，完整支持将在后续版本中推出。

在Failover Cluster模式下，SQL Server服务只在主节点上是启动的，故在创建云服务器备份时，只需要将主节点加入策略进行备份。在主备发生切换后，及时调整策略，确保始终对主节点进行备份。在恢复时，请先停止所有备节点，然后还原主节点。

1.8 保护 Always on Availability Groups 模式下的 SQL Server

当前云服务器备份只支持单个虚拟机的一致性备份，对于集群数据库暂不支持，完整支持将在后续版本中推出。

在Always On模式下，SQL Server服务在主备节点上都是启动的，数据由主复制到备，主上拥有全部的数据。故在创建云服务器备份时，只需要将主节点加入策略进行备份。在主备发生切换后，及时调整策略，确保始终对主节点进行备份。

由于SQL Server自身的机制，在恢复主时，可能会触发同步，使备节点上的数据也被覆盖，导致备份时刻之后新产生的数据丢失，所以建议只有在主备节点均不可用时才进行整机恢复，防止非预期的数据丢失。

A 修订记录

发布日期	修订记录
2020-11-06	第一次正式发布。